解　説

メモリと計算機アーキテクチャ†

相　磯　秀　夫††

1. まえがき

過去 30 年余にわたる計算機開発の歴史をメモリとそれに関連する計算機アーキテクチャの観点から眺めると，いつの時代でもメモリの“高速化”，“大容量化”，“高信頼化”そして“使い易さ”への追求がなされていたことが分かる．それらは新しい原理あるいは素子に基づくメモリの開発，メモリを実用化し懸案の問題解決を図るための工夫，新しい応用分野の展開に伴って提起される問題に対する有効な解を得る努力，あるいは新しい機能をもつメモリ開発への要求といった形で表われている．いずれの場合も，メモリは計算機アーキテクチャとの関連において初めて実用としての意味をもち，計算機ユーザにインパクトを与えている．このようにメモリと計算機アーキテクチャとの関係は不可分のものであるが，多くの場合，半導体を主体とするハードウェア技術の進歩と応用面からの必要性に促がされて両者は発展している．

ここではメモリ技術の進歩とそれをとりまく計算機アーキテクチャの関係について，いくつかの視点から概観してみたい．

2. 半導体技術のインパクト

メモリ開発の歴史を通して，半導体技術の進歩ほど大きな影響をメモリ技術に与えたものはないであろう．1950 年代から 1960 年代初めにかけて種々のメモリが開発され，実用化の試みがなされた．その結果，フェライト磁性体がメモリとして最も適していることが明らかになり，1970 年代前半までコアメモリが主メモリの主流を占める時代が到来する[1]．しかしながら，1960 年初頭から研究開発が進んでいた論理回路の高集積化はその後急速に進展し，1970 年半ばにはコアメモリは半導体 (IC) メモリに取って代わられる時代がくる．実際に，1975 年には主メモリ・ビットの約 50% が，また 1977 年には 85～90% が半導体素子に代わったといわれる[2),3)]．

このように主メモリが IC メモリへと移り変わった理由には，コアメモリよりも，(i)高速化できる，(ii)小型化できる，(iii)大容量メモリが実現できる，(iv)価格が安くなる，(v)電力消費が小さくなるなどの他に，(vi)複雑な論理機能を盛込めるといった魅力にあるといえる．実際に最近の技術傾向は実装密度にして年 2 倍の改善，価格は年約 40% の低下を示している．

メモリに論理機能を直接的に付加する試みは古くから行われていたが，従来のメモリ素子は情報の論理処理を行う論理素子とは本質的に異質でことごとく不成功に終っていた．IC メモリはその点では問題は殆んどない．記憶内容からそれを蓄えるアドレスを探索する連想メモリ[4)]は論理メモリの一種として既に実用になりつつある．大規模な連想メモリが実用化すれば，多くの応用分野でデータ処理の仕方が本質的に変わり，処理効率が大幅に改善できると期待されている[5)]．

IC メモリの実用化が定着するまでには，IC メモリがもつ欠点を技術的に解決する努力がなされている．それらは(a)符号理論あるいは多重化などの冗長技術による信頼性の改善，(b)停電時における記憶内容の確保などに関するものである．特に後者(b)に関しては不揮発性半導体メモリ ROM (Read Only Memory) の開発と実用化に拍車をかけた．また，論理回路とメモリ素子の概念を組合わせた PLA (Programmable Logic Array)[6)] の開発へと進展し，論理設計の分野にも大きな影響を与えている．

半導体技術のインパクトは主メモリだけではなく，大容量ファイル・メモリの分野にも及んでいる．磁気バブルや電荷結合素子 (CCD) などによる機械的な機構を含まない，いわゆる電子ディスク[7)] の普及である．恐らく数百 MB 程度の容量の高速・高信頼ファイル・メモリとして重要な位置を占めるであろう．バ

† Memories and Computer Architecture by Hideo AISO (Department of Electrical Engineering, Faculty of Engineering, Keio University).

†† 慶応義塾大学工学部電気工学科

ーソナル・ファイルあるいは数 MB]程度の主メモリと数十 GB クラスの通常の磁気ディスク・ファイル・システムとのギャップをうめる役割を果すものと思われる.

高速で安価な IC メモリの普及はシステム・アーキテクチャにも大きな変化を与えている. ダイナミック・マイクロプログラム方式の普及, 高機能パーソナル・コンピュータの実用化, 分散処理システムあるいは専用コンピュータの開発を容易にしている原動力となっている.

3. 高速化と大容量化の追求

計算機ユーザにとって, メモリの高速化と大容量化はいつの時代でも最大の願望であった. しかしながら, メモリの速度と容量に関する特性には相反する要素が含まれ, 高速化と大容量化は同時に両立することはない. このため, どちらかの特性に重みづけを行った種々のメモリが開発されることになった. その結果, 小容量・高速 IC レジスタ・ファイルから大容量・低速ストレージ・システム MSS (Mass Storage System)[8] に至るまで, 相互に特性の弱点を補いつつ, 複数種のメモリを実装する, いわゆる記憶の階層という概念をもたらした. 実際に階層化されたメモリ・システムではメモリ間のデータ転送が要求されるが, その頻度は小容量高速メモリほど一般に高く, その管理はユーザにとって厄介な問題となっていた. これはプログラミング上の単一レベル論理アドレス空間と実際の複数レベル物理アドレス空間との間のギャップに起因するものである. ユーザにとっては, メモリの物理的階層に関係なく, 論理的には容量無限大の単一レベルに見えることが望ましい. 仮想記憶システム[9] はこのような問題解決の一手法として考案されたものである. この考え方は現在 MSS レベルまで拡張されている.

一つのメモリを分割し, モジュール化して並列に動作させ, 見かけ上の高速化を図る試みも古くからある. プログラムとデータ用のメモリを別々に設ける方式, 一つのメモリを独立して動作するメモリ・バンクに分割し, 一つずつ順番にアドレス付けをするインタリーブ方式, 分散処理システムにおけるローカル・メモリやバッファ・メモリの設置などがその例である.

従来の計算機では, CPU のスループットは主メモリの速度に依存することが多く, その意味では主メモリへのアクセスを減らすことが重要となる. このため多数のレジスタを設けることが常識になりつつある. 当然のことながら, CDC 6600 計算機などに見られるように, 高速計算を目的とする場合にはレジスタのみを対象とする演算命令体系が生まれる.

データ処理の効率を上げるために, マイクロプログラム制御方式を採用することが普及してきたが, 使用頻度の高い中間言語やサブルーチンをファームウェアの形で高速マイクロプログラム・メモリに格納する方式がとられている. また, 従来の ROM に代えて, 書換え可能な RAM (Random Access Memory) をマイクロプログラム・メモリに用い, ユーザ・マイクロプログラミングを可能にし, 仮想計算機の実現を容易にしている. これらも半導体技術の進歩に負うところが多い.

4. 使い易さへの追求

従来のノイマン型計算機の特徴の一つはメモリが連続した一次元のアドレス, すなわち線形アドレスをもっていることである. このような単純構造のメモリに対して, 複雑な構造をもつユーザのプログラムやデータをいかに対応づけるかは重要な問題になっている. 線形構造をもつセグメント (プログラムやデータの集合) の対応づけ (マッピング) はプログラミング技術の基本として早くから議論され, 直接, 間接, 相対, インデックス修飾, イミディエイトといったアドレス方式が確立した. このような基本的なアドレス方式の重要性は最近のマイクロプロセッサのアーキテクチャにおいて再認識されており, プログラミングのし易さ, プログラムの大きさ, プログラムの実行速度などに密接な関係がある.

マルチプログラム処理においては, プログラムやデータを主メモリの任意の位置に実行中に移しかえる必要があり, このための再配置 (リロケーション) 機構は必須とされている. また, 応用が拡大し, プログラムの種類が多くなり, その上個々のプログラムが大規模になるにつれて, 主メモリの大きさによる制約が大きく浮び上がってきた. これらの問題を解決する手法として, 仮想記憶方式が高く評価されている. 仮想記憶方式ではマッピングを伴う主メモリとファイル・メモリ間のデータ転送 (オーバーレイ) はオペレーティング・システムの管理下で行われるため, オーバーレイに必要な入出力処理はユーザ・プログラムに含める必要はない. この考え方を徹底的に拡張して, オンライン・ライブラリに至るまでのすべてのファイルを同

—論理アドレス空間に取り込み，一般のプログラムから入出力処理命令を完全になくす試みが MULTICS システム[10]に見られる．仮想入出力方式ともいうべきこのような方式はプログラミングを容易にする方法として注目されているが，計算機アーキテクチャ・レベルの機能サポートはまだ十分ではない．

応用分野が拡大するにつれて，配列，待行列，スタック，表，木(トリー)，有向グラフといった複雑なデータ構造を線形アドレスのメモリ構造に効率よくマッピングする技法が要求されるようになった．これを助ける基本的な機構としては，ディスクリプタ[11]，スタック[11]，ハッシュ機構[12]あるいは連想メモリなどが考案されている．特に，スタックはプログラムの動的リンケージや数式の計算に極めて有用な役割を果すことが明らかになっている．スタックを用いれば，ユーザは記憶場所の管理あるいはオペランド・アドレスの指定などの煩わしさから開放される．

プログラムの多重処理においては，記憶領域の独立性を保証することが求められる．このため種々の記憶保護方式が提案されるようになった．また，セグメントの性格をプログラム実行の過程で細かく定義し，その状況に応じてアクセス権を設定する制御方式も実現されている．

ノイマン型計算機のもう一つの特徴は固定長のデータを取扱うことを原則としていることである．可変長データを扱えるメモリ構造が提供されれば，処理効率ならびにメモリの利用率が著しく改善できる可能性がある[13]．

5. 応用面からの要求

最近の応用分野は計算機を主として計算処理のために使うよりはメモリ・システムの機能を利用するために使うことが多い．したがって，応用分野が拡大するにつれて，メモリ・システムに対する要求も厳しくなっている．

先ず第一に大量のデータを取扱う応用が増えたことである．大規模な科学計算，画像処理，情報検索，自然言語処理などがその例である．これらの場合には，大容量主メモリの他にそれをバック・アップするバッファ・メモリの役割が重要になり，メモリ・システムの構成，データ転送，アクセス法などが全体の処理効率の鍵をにぎることになる．また，ある分野では大容量で安価な光映像ファイル・メモリやマイクロ・フィッシュなど新しいメモリ・システムの利用を望んでいる．

オンライン実時間処理の分野では，高速性と高信頼性をメモリ・システムに要求する．高速 LSI メモリの使用，キャッシュ・メモリの適用，モジュール構造とマイクロプログラム制御による適応機能の実現，冗長技術の駆使などによってこれらの問題に対処している．

マンマシン・インタフェースで主役を果すグラフィック・システムなどの分野では複雑なデータ構造を扱う上にデータ構造の変換という極めて複雑な操作を行う．このため最近では特殊な機能を備えたメモリ・システムならびにデータ構造変換ハードウェアを開発している例が見られる．

6. 最近の研究課題

メモリならびにそれに関連する計算機アーキテクチャに関して最近話題になっている課題を簡単に列挙すれば，次のとおりである．

(1) 新しいメモリ：新しい記憶原理に基づく磁性体メモリ，光技術を利用したファイル・メモリ，化合物半導体あるいはジョセフソン結合素子による超高速メモリ，アナログ情報用ファイル・メモリなどの開発が進んでいる．

(2) 論理メモリ：半導体メモリの特性を活かして，メモリの中に応用に適した論理機能を含める試みがある．連想メモリに関しては小規模な LSI 素子が実用になっているが，本格的な実用化はこれからである．連想機能のほかに，分類，変換，特殊演算などの機能を対象としたセル構造論理あるいは配列論理メモリなどの研究がされている．ICL 社の DAP[14] はその一例といえよう．

(3) 高級プログラミング言語：高級プログラミング言語で定義される種々の抽象データ構造を計算機アーキテクチャ・レベルの機能でいかに効率よくサポートするかは重要な研究課題である．この場合，すべてのデータ構造はもとよりデータの型，データの操作，正当なアクセス権，メモリ領域，共用変数，データ識別子あるいはサブプログラムの管理，デバッギングなどに関しても配慮することが望ましい．

(4) 機能分散処理：主メモリやファイル・メモリを機能的に分散し，それぞれを専用プロセッサで処理する方式が次第に普及しつつある．特にファイル・メモリを OS のバック・エンド・プロセッサとして独立させることを目的としたファイル・プロセッサの開発

が問題になっている.

(5) データベース・マシン: リレーショナル, 属性, 階層あるいはネットワークといった形体のデータベースを効率よく取扱うための専用マシンの開発が急速に進展している. いずれの場合も, メモリの構造, アクセス法, データ転送, 集合演算, 連想処理, セキュリティなどに関する機能の検討が重要である.

(6) データフロー・マシン: 非ノイマン型計算機の代表的な例として注目されているデータフロー・マシン[15]とそれをとりまく新しいプログラミング言語には, ノイマン型計算機の大きな特長であるメモリ・セルの概念を打破しようとする考え方が含まれている. メモリ・セルの概念は順序機械の基本で極めて有用であるが, プログラムの並列処理やプログラム開発の生産性改善などの観点からは適さない面がある. そのような意味で, 従来のメモリの概念から外れた新しい動きがある. 人工知能や知識工学の考え方に基づく新しい問題解決手法にもデータフロー・マシンの概念は密接な関係があるように考えられ, 今後の研究成果に期待するところが多い.

7. あとがき

広義にとらえた計算機アーキテクチャの立場から過去の計算機開発を眺めると, 時代とともにその主題は移り変わっている. それらはCPU, メモリ, 入出力, そしてシステムに主題を置いたアーキテクチャへと移行していることが分かる. ここで概観したメモリに関連するアーキテクチャの諸問題は多くの経験を通して, かなり明確になりつつあるが, プログラミング言語の観点から見た諸問題に対しては計算機アーキテクチャ・レベルでの解決はほとんどなされていないように思う. 計算機をより使い易くするためにも, また人間に適した新しい問題解決手法を可能にするためにも, 高級プログラミング言語の観点からメモリ・システムと計算機アーキテクチャのあり方を見直すべきではなかろうか. これからは言語アーキテクチャ (language architecture) が重要になるような気がする.

参考文献

1) 石井 治: メイン・メモリの動向, 情報処理, Vol. 16, No. 4, pp. 258-274 (Apr. 1975).
2) Juliussen, J. E., et al.: Problems of the 80's: Computer System Organization, Proc. Conference on Computing in the 1980's, pp. 14-23 (Mar. 1978).
3) Petschauer, R. J.: Semiconductor Storage-A Look at the Future, ibid., pp. 244-247 (Mar. 1978).
4) 飯塚 肇: 論理メモリ, 情報処理, Vol. 16, No. 4, pp. 275-285 (Apr. 1975).
5) Feldman, J. D., et al.: RADCAP-An Operational Parallel Processing Facility, Proc. NCC, 43, pp. 7-15 (June 1974).
6) Fleisher, H., et al.: An Introduction to Array Logic, IBM J. Res. and Develop., Vol. 19, No. 2, pp. 98-109 (Mar. 1975).
7) 石井 治: 電子ディスク, 情報処理, Vol. 17, No. 12, pp. 1160-1168 (Dec. 1976).
8) Johnson, C.: IBM 3850-Mass Storage System, Proc. NCC, 44, pp. 509-514 (June 1975).
9) Kilburn, T., et al.: One-Level Storage System, IRE Trans. EC-11, No. 2, pp. 223-235 (Apr. 1962).
10) Bensoussan, A., et al.: The Multics Virtual Memory: Concepts and Design, CACM, Vol. 15, No. 5, pp. 308-318 (May 1972).
11) The Descriptor-A Definition of the B 5000 Information Processing System, Burroughs Corp. (Feb. 1961).
12) Ida, T., et al.: Performance of a Parallel Hash Hardware with Key Deletion, Proc. IFIP Congress 77, pp. 643-647 (Aug. 1977).
13) Myers, G. J.: Advances in Computer Architecture, John Wiley & Sons (1978).
14) Reddaway, S. F.: DAP-A Distributed Array Processor, Proc. 1st Annual Symp. on Computer Architecture, 1, pp. 61-65 (Dec. 1973).
15) Dennis, J. B.: First Version of a Data Flow Procedure Language, Lecture Note in Computer Science, 19, pp. 362-376, Springer-Verlay (1974).

(昭和55年2月20日受付)