

## I. デジタル情報と半導体産業（つづき）

### D. コンピュータの構成と働き

コンピュータ（PC、パーソナル・コンピュータ、パソコン）は、「ハードウェア」と「ソフトウェア」から構成される。ソフトウェアは、OS（「基本ソフト」、オペレーティング・システム）と、AP（「応用ソフト」、アプリケーションズ・システム）に分かれる。

#### 1. ハードウェア

##### a. ハードウェアの構成

###### (i) 内部装置と外部装置

###### 内部装置

CPU、メモリー、レジスタ、クロックなど諸コントローラ（チップセット）、  
外部装置用入出力ポート・バス（内部装置を結合）

###### 外部装置

キーボード、マウス、ディスプレイ、プリンタなど

###### (ii) 主要機能別分類

CPU（中央演算処理装置） ビット列の変換を行う

CPUの素子 電圧の有無（0と1）で作用する一種のスイッチ。多数のスイッチ（数百個ないし数百万個）を接続し、必要なビット列の変換を行う。

記憶媒体（ビット列のデータ及びプログラムを記憶する）

ビット列の変換に必要なデータあるいは作業手順（プログラム・ソフトウェア）を記憶し、必要に応じてCPUに情報を与える。またCPUからの情報を蓄積する。

内部記憶（内部メモリー、CPUと同程度のスピードで動く高速記憶）

外部記憶（磁気ディスク、磁気テープ、光ディスク、ICメモリー、他）

入力装置（入力情報をビット列に変換してCPU・メモリーに伝える）

出力装置（CPUからのビット列を出力情報に変換し、印刷物・映像などの形で出力する）

##### b. ハードウェア（内部装置）の動作

クロックにしたがって各機器が動作する。

クロックの各ステップごとに、CPUが命令レジスタに入った「命令」を「実行」する。

機器間の「通信」はバス方式

すべての機器にビット列を送る。取捨は宛先情報で選別する。

クロックの速さによって、ハードウェア全体の動作スピードが決まる。

CPUは自分の（内部）クロックを持っている。

機器間の連携はコントローラ（チップセット）が担当

c. ハードウェア（外部装置）の動作

各外部装置は、固有のデータ伝送方式により入出力ポートを經由して本体のバスに連結される。

2. ソフトウェア（OS、BIOS）

a. BIOS

入出力ポートを經由する外部装置とのデータの流を変換・制御する。

b. OS

コンピュータ・ハードウェア全体の動作を制御する。AP から指示を受け取り、これを各ハードウェアに伝える。そのとき、API（アプリケーション・プログラム・インタフェース）という「標準語」が使われる。つまり、API は、AP ベンダーに（したがって一般に）公開されている。また OS は、「ドライバ」と呼ばれるソフトウェアを經由してハードウェアを動かす。

OS は上記の他に、複数の AP（等のプログラム）の起動・終了を制御し、またユーザに対して「ウインドウズ」や「マウス・クリック入力」などグラフ画面による分かりやすいインタフェースを提供する。

3. アプリケーションソフトウェア（AP）

a. 種別（例）

文書作成（MS ワード、一太郎）

表計算（MS エクセル）

Web 閲覧（MS-IE、ネットスケープ社コミュニケーター）

メール（MS アウトルック）

他多数

b. AP の働らき

それぞれの目的にしたがって、ユーザの指示（キーボード、マウスなど）により、OS を通じてハードウェアを動かす（直接にハードウェアを動かすことは通常はない）。

#### 4. コンピュータはどのように動くか

##### a. 人間との類似点

人間:

頭脳: 処理と記憶が一体化 (詳細についてはまだ分かっていない)

処理と内部記憶 (暗記)

外部メモリー (大量情報を処理・記憶)

ノート、紙、書物  
情報媒体

コンピュータ

CPU (処理) 装置  
メモリー (記憶) 装置... 内部記憶 (半導体)

外部記憶装置 { CD-ROM  
ディスク、テープ

##### b. フォン・ノイマン型コンピュータ (人間との相違点)

CPU と主メモリーを分離

ソフトウェアをメモリー上に展開

ソフトウェアの「命令」を 1 個ずつ取り出して順次実行

(つまりコンピュータの根本である CPU のところでは、同時並行的情報処理はおこなわれていない。しかし、他ハードウェア間では並行処理がなされる。)

「フォン・ノイマン・ボトルネック」 あい路

CPU とメモリー間の通路 (バス)

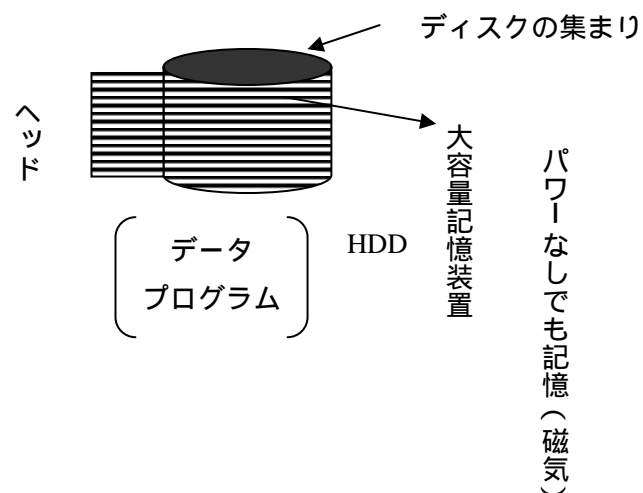
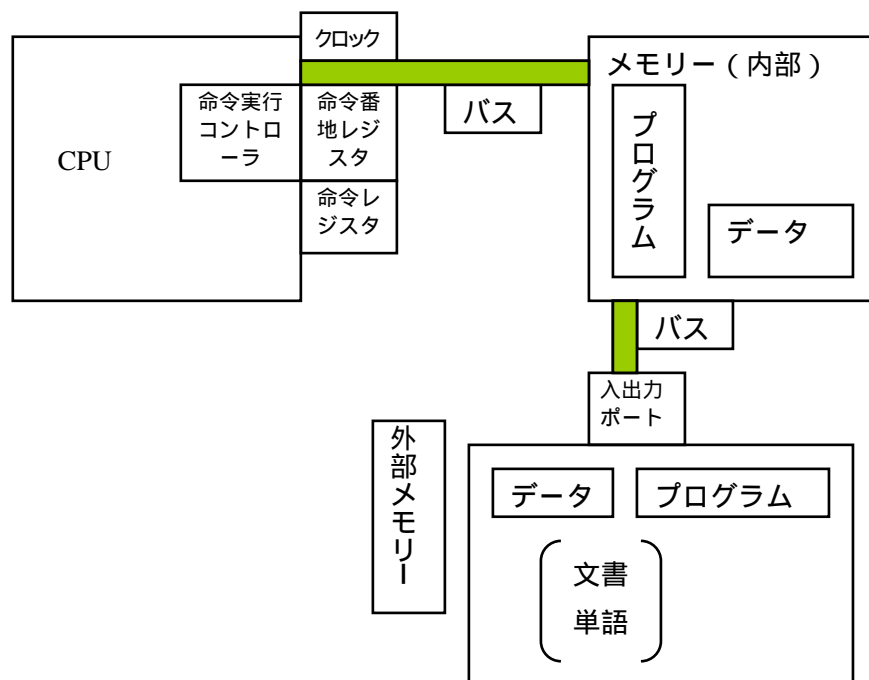
CPU が命令を一個ずつ順に処理しなければならない (同時並行処理ができない)

ボトルネックを避ける例: 複数 CPU を持つコンピュータ

宇宙天文のための多数コンピュータの結合 (地球規模、インターネットを利用)

von Neumann: コンピュータの基本を設計した (1950年代)

## ソフトウェア（プログラム）によるコンピュータの仕事（情報処理）

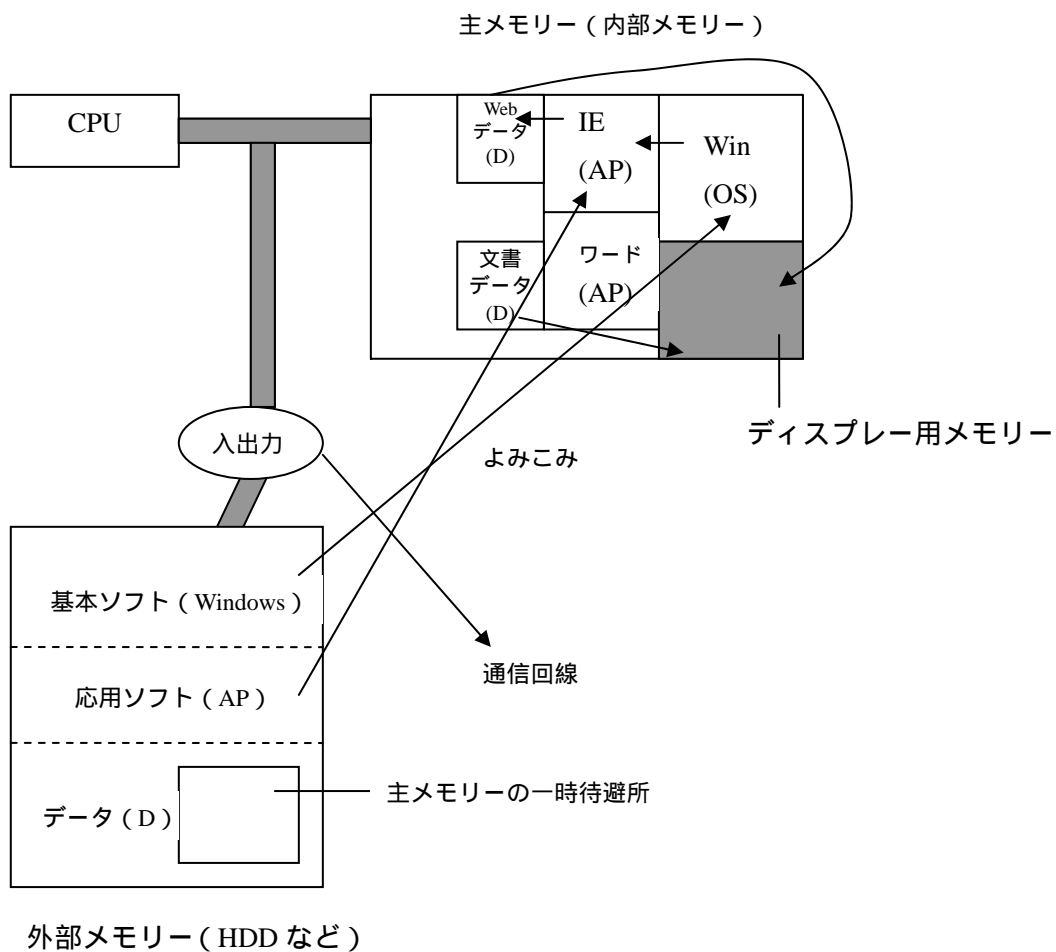


- (1) 命令実行コントローラが命令番地レジスタに入っている主メモリーの番地（アドレス）を読み、そのアドレスに入っているメモリー中の「命令」を「命令レジスタ」に入れる。命令番地レジスタの値に1を加えて、次に実行するべき命令の場所を示す。
- (2) CPUは、「命令レジスタ」に入っている命令を読んで、これを実行する。命令の種類は数十～数百個。（命令のうちには、「命令番地レジスタ」を変更するものもある（分岐命令、そのときは、命令番地がジャンプし、新たな番地から命令が読み込まれ、実行される）。

(3) 以下、上記(1)と(2)をくり返す。つまり、CPUの作業は上記2ステップで1サイクルになっており、内部クロックによって進行する。(その速度は、数百MHzから数GHz、つまり1秒間に数億から数十億回に及ぶ。)

c. ソフトウェアによる情報の重層・並行処理の実現

コンピュータ内部メモリと外部メモリにおけるソフトウェア配置



各プログラムがCPUの作業能力を少しずつ時分割方式（タイムシェアリング タイムスライス）で利用し、仕事を進める。CPUの命令の1つ1つは単純な仕事（例：キーが押されたか否かをチェックする、ディスプレイの一部を指定された色で塗る、など）。プログラムは、これを次々に組織的に組み合わせて使い、目的とする仕事をおこなう。

多数のプログラム（ソフト）の「同時進行」（OS により実行）

それぞれ、分業により、専門のプログラムが担当。プログラム自体が重層構造になっている。

タスク（コンピュータの側から見たプログラム（の各部分）の呼び方）

制御タスク

マネージャーが担当

アプリケーションタスク

アプリケーション自体

各種マネージャー（OS 中の「スペシャリスト」）

タスク（仕事）マネージャー：

全体の仕事（タスク）進行をコントロールする

資源マネージャー

CPU マネージャー（割り込みコントローラ）

メモリ・マネージャー

メモリ・ページ・マネージャー

ディスク・マネージャー

キーボード入力マネージャー

通信入出力マネージャー

など

各ハードウェアの状態を管理し、命令を実行する。

これらのマネージャーの仕事自体も、OS の一部としてメモリー内のプログラムによる CPU 利用として実現されている。ただし一部は、ハードウェアに組み込まれている。

舞台ステージとコンピュータの比較

舞台ステージ

表方：観衆に見える人間・装置

裏方：観衆に見えないところで人間・装置を用意し、舞台を進行させる。

コンピュータ

表方：AP などで実現される入出力、ユーザに見える。

裏方：各マネージャーによるサポート

舞台監督・進行係：タスク・マネージャー

装置係、出演者の世話：資源マネージャー

進行のパワー・エネルギー：

直接的には電気エネルギー

情動的にはCPUの「パワー」

コンピュータの「待機」状態

マネージャーが同じ仕事（無意味な）をくり返している

舞台とコンピュータの相違点：

舞台はあらかじめ決められた筋書に沿ってシナリオが進む

コンピュータはユーザ他との「やりとり」で進み方がちがってくる（シナリオが分岐する）（ゲーム機はコンピュータの一種）

（観衆の注文によって演技や筋書きを変える舞台のようなもの）

裏方が働かなければ舞台は進行しない

全体システムの「協調」「相互関連」が重要

シナリオ進行方式（2種類）

