

第4章 コンピュータの構成

本章では、コンピュータの全体構成について、いくつかの視点から取り上げます。具体的には、コンピュータの概念モデルが、コンピュータの製品へどのように実現されているのか、コンピュータの動作原理や仕組みはどうなっているのか、といったことに関して明らかにします。これによって、コンピュータの内部を、ある程度ホワイトボックス化することを目指します。

1.1 コンピュータの概念モデル

「コンピュータとは一体何なのか」という問いに対しては、その概念モデル - つまり、メンタルモデル - を明らかにするによって答えることができます。メンタルモデルとは、実際の物や事象を理解する過程で形成される知識や概念を抽象化したものといえます。

コンピュータの概念モデルを把握することによって、コンピュータの内部をある程度ホワイトボックスとして捉えることができるようになります。こうして、コンピュータに対して、統制感を持った健全な利用者という立場を取ることが可能になり、高度情報化社会の一員として積極的に活躍することができると言えます。

(1) コンピュータとは

コンピュータという言葉は、英語の computer をそのまま片仮名で表したものです。当初は、「電子計算機」と日本語に訳されて使われていましたが、単なる計算しかできない道具というような非専門家らの誤解が生じたりしました。しかし、コンピュータの技術革新が進展するにつれて、高度に複雑な処理も実現できるようになってきました。このため、現在では日本語に訳すことなく、そのまま使われていることの方が多いといえます。

このように、コンピュータの世界では、コンピュータという言葉以外に、実に多くの用語が片仮名のまま使用されています。これには、あえて日本語に訳すと上述のような誤解が生じることや、次から次へと新しい用語が登場するため訳語が定着しない、などの状況が理由としてあげられます。それだけでなく、英語の略称も多いと言えます。単語の先頭文字だけをまとめたもの(たとえば、LAN: Local Area Network, CSS: ClientSever System など)です。

このようなことから、情報処理の世界では、英語の理解も必要になります。このような状況になっている背景には、コンピュータ関連技術の多くが、英語文化圏から我が国へ流入してきていると事実があげられます。これを打開するためにも、教育機関における情報(処理)教育のあり方が問われることになるわけです。

さて、コンピュータは、一般製品(家電や日用品など)と同様に、工場で生産され出荷されています。つまり、コンピュータも一つの製品であるわけです。しかし、一般製品とコンピュータには、決定的な相違があります。それは、利用目的の違いという点です。

通常、一般製品は、何らかの利用目的を前提に作られます。たとえば、電子レンジは食品を解凍したり短時間に料理するといった利用目的が、乗り物は人間や物を移動させるといった利用目的が、それぞれあります。これに対して、コンピュータは、汎用的な利用目的を前提にしています。汎用的であるということは、どんな目的にも利用できることです。そのために、部品(ハードウェア)やプログラム(ソフトウェア)を組み換えることができるよ

うになっています。これによって、我々は万能機械を得ることができたわけです。

ただし、万能機械といっても、何から何までできるということではありません。具体的には、プログラミングができる問題に対してしか対応できないという点があげられます。プログラミングとは、人間(一般には、プログラマ)がコンピュータに対して指示命令する手順を記述する作業のことを意味しています。このため、人間が逐次考えた手順にしたがって、プログラミングが行われるわけです。一方、人間が考えられないような問題に対しては、プログラミングができないことになります。これには、たとえば停止問題(あるプログラムが有限時間内に停止するかどうかを判別する問題)などがあげられます。

また、理論的にはプログラミングできても、実用的ではない問題に対しても対応できません。これには、たとえば巡回セールスマン問題(指定された都市をあるセールスマンが1回ずつ訪問するうちの最短経路を求める問題)があげられます。

あるいは、発見的(ヒューリスティック)な問題に対しても、万能ではありません。これには、チェス・囲碁・将棋といったプログラムがあります。最終的な結果に到達するまでに試行錯誤を繰り返すことによって、解法を発見するような手順です。

このようなことから、映画「2001 年宇宙の旅」に登場した木星探索機に搭載されていた HAL のような知的コンピュータは、まだまだ先の話になりそうです。

(2) ハードウェアとソフトウェア

汎用的な利用を実現するために、コンピュータは、ハードウェアとソフトウェアをうまく協調させて、与えられた処理を遂行する機能が組み込まれています。

このうちのハードウェアは、コンピュータを構成する装置や部品のことです。皆さんが目にする、あるいは、触ることができる装置や筐体そのものです。デスクトップパソコン(卓上という意味です)を例にすると、箱型の本体、キーボード、ディスプレイ、マウスなどが相当します。また、本体の前面についている CD-ROM やフロッピーディスクの読取り装置や、本体の中にあるハードディスク装置やマザーボードに接続している CPU やメモリなどもそうです。

それに対して、ソフトウェアは、目に見えないものといえます。これらは、プログラムやデータという形で、ハードディスク上に磁気的に記憶されたり、CD-ROM に工学的に記憶されています。このため、我々は物理的に触れることもできないわけです。あるいは、エディタなどを通して、プログラムやデータを覗いたとしても、プログラミング言語やデータの意味がわからないと、やはり何が何だかわからない代物になってしまいます。それだけでなく、プログラムを作成するための考え方(理論、概念、方法論、技法など)やドキュメント(仕様書、設計書、操作手引書など)も含まれます。アイデアなどは、まさしく見ることができないわけです。

以上のハードウェアとソフトウェアの特性を対比すると、次のようになります。

ハードウェア：有形、可視性、物理的、定量的、工業生産

ソフトウェア：無形、不可視性、抽象的、定性的、手工業的

昔は、ハードウェアを「金物」と訳したり、ソフトウェアの一つであるプログラムを「算譜」と訳した時代もありました。

また、ハードウェアの中の中央処理処理は、LSI を構成する回路図を最も基本的な仕様の枠組みとしています。回路図からその装置の動作まで把握できることから、その可視性が

保証されています。それに対して、ソフトウェアの中のプログラムは、基本的な仕様の枠組みな存在していないといえます。たとえば、流れ図を眺めてみても、残念ながら、それがどのような機能を実現するプログラムなのか理解することがほとんどできません。このことから、ソフトウェアは不可視性であると言えます。

これらのハードウェアとソフトウェアを適宜組み合わせることによって、コンピュータに対してさまざまな利用を実現することができます。ハードウェアに関して言えば、装置や部品を利用目的に合わせて組み換えることができます。たとえば、写真を取り込んでマルチメディア的なコンテンツを制作するためには、イメージスキャナかデジタルカメラを新たに接続することになります。また、それらのコンテンツをカラーで印刷する際に、インクジェットプリンタではなく高解像度のレーザープリンタに切り換えることもできます。

ソフトウェアに関して言えば、アプリケーションソフトを組み合わせる使うことができます。スキャンした画像データを画像処理ソフトで加工したり、さらに DTP(卓上電子編集)ソフトで編集することができます。それだけでなく、自分でプログラミング言語を用いて、思い通りの処理をプログラミングする(つまり、命令文の組合せ)こともできます。

コンピュータは、ハードウェアとソフトウェアの協調作業によって、一連の処理を実行しています。その場合のハードウェアとソフトウェアの位置づけについては、

コンピュータ ::= ハードウェア + ソフトウェア とします。

そのうちのハードウェアについては、

ハードウェア ::= 入力装置 + 中央処理装置 + 主記憶装置 + 補助記憶装置 + 出力装置

とします。一方、ソフトウェアについては、

ソフトウェア ::= 基本ソフトウェア + 応用ソフトウェア

基本ソフトウェア ::= 入出力管理 / プロセス管理 / (言語管理) / 記憶管理 / データ管理プログラム

応用ソフトウェア ::= アプリケーションプログラム + 利用者プログラム

であるということです。

以上の関係を、図 1 に示します。なお、ここで留意すべき点は、これらの各装置やプログラムはそれぞれ独自にではなく相互に関連し合いながら、動作を進めていくということです。

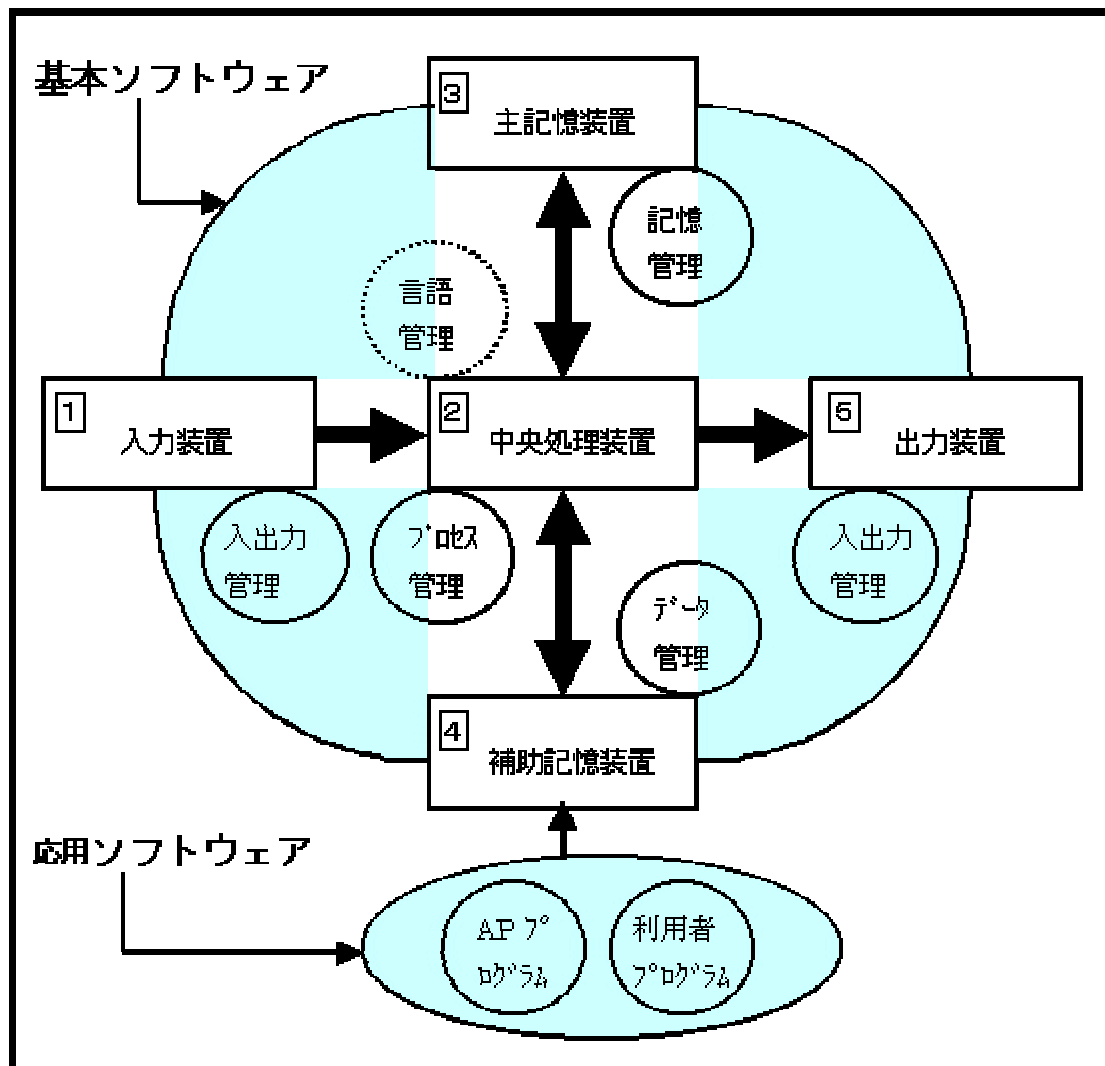


図1．コンピュータの構成

1.2 ノイマン型コンピュータ

1.1で取り上げた論理機械は、あくまでも抽象的な仮想モデルといえます。この論理的な仮想モデルを、実際の物理的素材(真空管、トランジスタ、集積回路など)を用いて、情報処理装置として実現したものが、(電子計算機と呼ばれる)コンピュータなわけです。その中でも、ノイマン(John.von Neumann)によるアイデアを実装したノイマン型コンピュータは、今日最も普及しているものといえます。ここでは、そのノイマン型コンピュータについて取り上げます。

ノイマン型コンピュータは、次の三つの方式を実現したコンピュータといえます。以下から、それぞれについて説明します。なお、ここからは、1.1で取り上げた論理機械の操作装置に記録されている操作規則(つまり、個々の遷移関数)を、プログラムと呼ぶことにします。および、1.1で取り上げた論理機械の入出力テープに記録されている記号を、データと呼ぶことにします。

(1) 2元符号方式

2元符号化方式とは、「プログラムとデータを、ともに2進数列として扱うこと」を意味しています。

一般的には、 i 桁の n 進数の数値(N)をべき乗を用いて表すと、

$$N = C_{i-1} \times n^{i-1} + C_{i-2} \times n^{i-2} + \dots + C_1 \times n^1 + C_0 \times n^0$$

になります。ここで、 C は仮数を、 n は基数を、 i は指数を、それぞれ表します。また、 C については、0から $(n-1)$ までのどれかの値をとります。

これより、2進数は「 $n=2$ 」に相当します。したがって、例えば6桁の2進数 $abcdef$ は、

$$(a b c d e f)_2 = a \times 2^5 + b \times 2^4 + c \times 2^3 + d \times 2^2 + e \times 2^1 + f \times 2^0$$

と表すことができます。当然ながら、 a から f までは、0か1のどちらかになります。

さて、コンピュータの内部では、すべての情報(数値、文字、図表、画像、映像、音声、音楽など)を2進数として扱います。

数値については、例えば10進数値ならば、

$$(21)_{10} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (10101)_2$$

となります。

文字については、例えば2進数8桁を用いる場合は、

$(0000000)_2$ に文字1を割り当てる

$(0000001)_2$ に文字2を割り当てる

:

$(1111111)_2$ に文字256を割り当てる

となります。これより、256種類の文字を割り当てることができます。

1	2	3	4	
			■	1
		■		2
	■			3
■				4

図2 . 4 × 4の2進数構成

イメージデータの中では、例えば4桁×4桁の2進数の組合せで扱う場合を考えてみます。例えば、斜線(/)を表すと、図2のようになります。ここで、1行目のマス目の中で黒を2進数の"1"、白を2進数の"0"に対応させると、

第1行目：(0 0 0 1)

第2行目：(0 0 1 0)

第3行目：(0 1 0 0)

第4行目：(1 0 0 0)

のようになります。つまり、 $(0001001001001000)_2$ のような2進数列として、斜線というイメージデータを扱うことができるわけです。

以上のことから、コンピュータでは、プログラムであろうと、プログラムが処理するデータであろうと、いずれも任意の2進数列(ビット列とも呼びます)に変換されているわけです。

このように、すべての情報を2進数で扱う理由としては、コンピュータに実装されている装置の部品の動作機構に依存しています。現在のコンピュータは、いずれも電子回路に

よって構成されています。電子回路の詳細については1．3で説明しますが、最も基本となる論理回路は、電流がオンの状態かオフの状態の組合せによって、その動作が決定されます。その電流のオン状態を"1"，オフ状態を"0"とすることによって、それぞれ2進数に対応することができるわけです。

(2) プログラム内蔵方式

プログラム内蔵方式とは、「プログラムとデータは、コンピュータのメモリに記憶すること」を意味しています。

ここで言っているメモリとは、記憶装置のことです。これには、主記憶装置と補助記憶装置があります。主記憶装置は、おもに実行するプログラムとデータを記憶するための装置であり、比較的容量(数十M^{メガ}バイト)が限られています。それに対して、補助記憶装置は、さしあたって実行しないプログラムとデータを記憶するための装置であり、大容量のもの(数G^{ギガ}バイト)が多くなります。

実際には、いろいろなプログラムとデータ(基本ソフトウェアであるオペレーティングシステムから、応用ソフトウェアであるワープロソフトや表計算ソフト等まで)を補助記憶装置に記憶します。このことを、インストールと呼びます。その上で、実行したいプログラムを、アイコンをクリックすることによって選びます。クリックすると、選択されたプログラムとデータが、主記憶装置に移ります。このことを、ローディングと呼びます。これによって、実行したいプログラムがコンピュータで稼動する状態になります。以上の関係を、図3に示します。

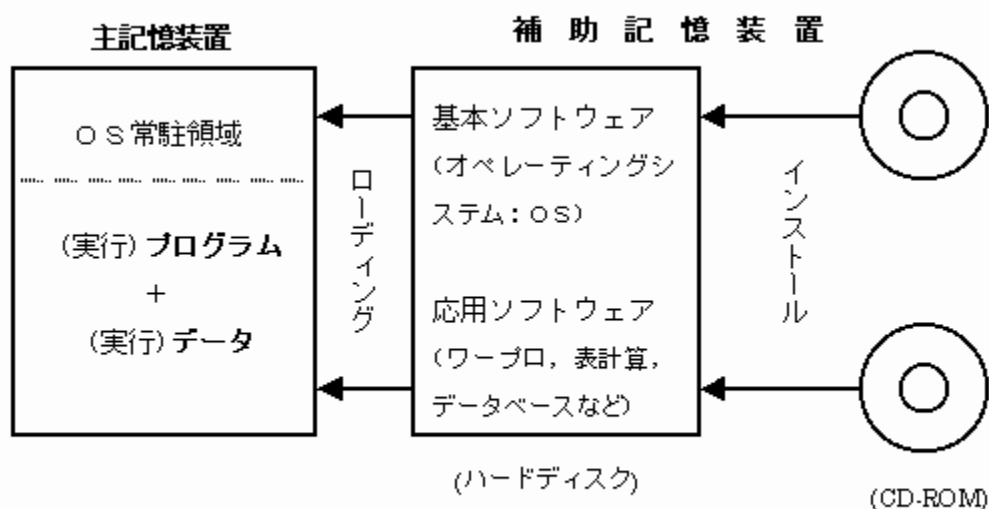


図3．プログラム内蔵の構成

ノイマン型コンピュータより以前のコンピュータは、メモリに相当する機能がほとんどなかったため、コンピュータに指示する手順は、逐次人間が外部から与えていました。たとえば、配線の組み替えやスイッチの組み合わせなどの物理的な機構を用いていたわけです。その結果、非常に複雑で煩雑な操作を、人間の方が強いられていました。

一方、プログラムとデータをメモリに記憶することによって、人間(ただし、利用者)はこ

ういった煩雑な作業から開放されるようになりました。それとともに、利用者から見ると、コンピュータはある意味で自律的に動作することが可能になったといえます。つまり、1.1 で取り上げた JIS によるコンピュータの定義にある「...人手の介入なしに遂行することのできる...云々」に相当する機能を持っていることになるわけです。

ただし、「ある意味で自律的に動作する」としたのは、あらかじめ人間(ただし、プログラマ)がプログラムを作成して、それを記憶装置に記録しているという前提条件があるからです。このため、プログラマは、あいかわらずプログラミングという複雑で煩雑な作業を繰り返していることになります。

以上より、コンピュータが、自己判断のもとに処理を遂行しているわけではなく、すべての手順を人間(プログラマ)がプログラミングして与えているということです。その結果、(利用者が)外から見ると、まるで勝手に動作しているように見えるわけです。

(3) 逐次実行制御方式

逐次実行制御方式とは、「プログラムの実行は、一度に一つの命令を順番に進めること」を意味しています。このことは、1.1 で取り上げた論理機械で言えば、操作規則が順次一つずつ処理されることに相当します。逆に言うと、複数の操作規則を、同時に並列的に実行することはできないということです。

このような実行制御になる理由として、ノイマン型コンピュータでは実行するプログラムをただ一つしか指定できないことがあげられます。

図3によると、実行するプログラムとデータがローディングされる(実行するソフトウェアの起動用アイコンをクリックすることによって、主記憶装置に記憶されます。その主記憶装置のすべての領域に対して、番地がふられています。つまり、番地によって、主記憶装置上のすべての場所が修飾されていることになります。

実行するプログラムを指定するということは、そのプログラムが格納されている主記憶装置上の最初の番地に位置づけることでもあります。実際のコンピュータでは、プログラムカウンタというカウンタの中に、その番地が設定されます。そのプログラムカウンタは、中央処理装置の中に一つしかありません。この結果、ただ一つの命令しか取り出せないわけです。

【練習演習】

いま、次のような文字を、2進数として符号化するとします。そのためには、2進数が何桁必要になるでしょうか。

- ・数字：0 から 9 までの 10 種類
- ・アルファベット：A から Z までの大文字と小文字 52 種類
- ・記号：空白や特殊記号など 38 種類

1.3 IPO モデル

IPO モデルとは、入力(Input)と処理(Process)：制御 / 演算 / 記憶と出力(Output)の関係を抽象化したモデルです。図4に、その構成を示します。

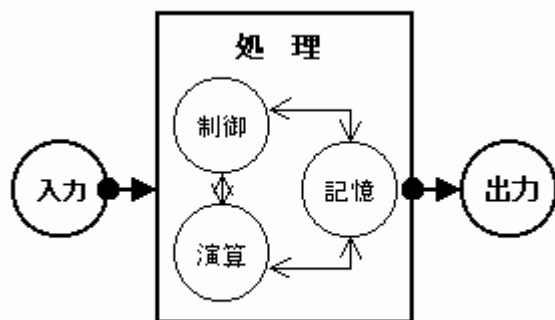


図4 . IPO モデル

ここでは、ノイマン型コンピュータを、IPO モデルという視点から見直します。ただし、それぞれの部分(図4の円に相当)については、個別の装置としてではなく、機能単位としてとらえます。このため、個々の機能単位に取り上げることにします。

(1) 入力機能

コンピュータにどのように情報が入力されるのか、その原理について取り上げます。コンピュータには、さまざまな情報が入力されますが、1.2で述べたようにすべて2進数に変換されます。そこで、ここでは具体的な例として、数と文字に関する2元符号化について取り上げます。

数の2元符号化

数の2元符号化については、1.2で取り上げたようにすべて2進数表示を用います。

・正の数について

正の数に関しては、次のようになります。たとえば、正の整数(i)において、0から $2^i - 1$ までの整数は、

$${}_{i-1}2^{i-1} + {}_{i-2}2^{i-2} + \dots + {}_12^1 + {}_02^0$$

より、 ${}_{i-1}{}_{i-2}\dots{}_1{}_0$ と表すことができます。ただし、 ${}_j = 0, 1 (j = 0, 1, \dots, i - 1)$ になります。たとえば、 $i = 3$ として、0から7までの整数は、

整数0 : $0(0 \times 2^0)$

整数1 : $1(1 \times 2^0)$

整数2 : $10(1 \times 2^1 + 0 \times 2^0)$

整数3 : $11(1 \times 2^1 + 1 \times 2^0)$

整数4 : $100(1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0)$

整数5 : $101(1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)$

整数6 : $110(1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0)$

整数7 : $111(1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)$

となるわけです。

・負の数について

負の数に関しては、符号ビットを付加する絶対値表現と補数表現とがあります。絶対値表現とは、絶対値を2進数に変換したものに、符号ビットを先頭に1桁付加する方式です。

一方、補数表現とは、1の補数と2の補数を用いて表します。

一般的には、補数というものを、次のように定義します。

いま、Aをr桁のn進数とすると、Aの補数には、nの補数とn - 1の補数があり、

$$A'(n \text{ の補数}) = n^r - A$$

$$A''(n - 1 \text{ の補数}) = n^{r-1} - A$$

になります。

これより、10進数(n = 10)1桁(r = 1)の「8」の補数については、

$$8'(10 \text{ の補数}) = 10^1 - 8 = 2$$

$$8''(9 \text{ の補数}) = 10^1 - 1 - 8 = 1$$

がそれぞれ対応します。

これより、イメージ的には、「11 - 8」の減算を、「8」に対して10の補数(つまり、「2」)をとって「11 + 2」の加算に置き換えてから、桁上がりを見捨てるために「13」の上位桁を除き「3」とすることになります。これによって、減算を、補数を用いた加算に変えることができるわけです。つまり、補数を用いることによって、正負や値の大小に関係なく加算することができることになります。

2進数r桁のAの補数については、

$$A'(2 \text{ の補数}) = 2^r - A$$

$$A''(1 \text{ の補数}) = 2^{r-1} - A$$

がそれぞれ対応します。

1の補数については、 $A' + A = 2^r - 1$ になることから、基数の数と補数を加算するとr桁がすべて1になります。これより、1の補数を求めるためには、Aの各桁についてそれぞれ1から引くことによって求められます。ただし、1の補数は+0と-0が生じるため、実際のコンピュータでは、次の2の補数を用います。

2の補数については、 $A' + A = 2^r$ になることから、基数と補数を加算するとr桁すべてが0で、(r + 1)桁が1になります。これより、2の補数を求めるためには、Aの各桁についてそれぞれ1から引いた値の最下位桁に1を加える(便宜的には、各桁の0と1を反転し1を加える)ことによって、求められます。

以上をもとに、絶対値表現と補数表現の対応を、表1に示します。ただし、2進数については8桁とします。

表1．数の表現

10進数表現	絶対値表現	1の補数表現	2の補数表現
+3	00000011	00000011	00000011
+2	00000010	00000010	00000010
+1	00000001	00000001	00000001
+0	00000000	00000000	00000000
-0	10000000	11111111	(+0と同じ)
-1	10000001	11111110	11111111
-2	10000010	11111101	11111110
-3	10000011	11111100	11111101

・ 2 進化 10 進符号

10 進数を 2 元符号化するとき、10 進数 1 桁毎に符号化を行う方式もあります。10 進数の性質をそのまま利用したいような場合に有効です。この方式で代表的なものに、2 進化 10 進符号(BCD)があります。これは、10 進数の 1 桁(0 から 9 まで)を、2 進数の 4 桁(つまり、4 ビット)として表す方式です。

文字の 2 元符号化

数値だけでなく、文字そのものも 2 元符号化します。これには、すでに規格制定されている体系が存在しています。というのも、情報交換を国際間で円滑に進めるためには、標準化が必要になるからです。

国際的な標準規格としては、ISO の IS646(7 単位符号)があります。これに準拠する形で、米国では ASCII が規格制定されています。同様に、我が国では JIS C6220「情報交換用符号」が規格制定されています。これには、

- ・ ローマ文字用 7 単位符号
- ・ 片仮名用 7 単位符号
- ・ ローマ文字及び片仮名用 7 単位符号
- ・ ローマ文字及び片仮名用 8 単位符号

がそれぞれ含まれています。

また、漢字の 2 元符号化の標準規格として、1978 年に制定された JIS C6226「情報交換用漢字符号」があります。これには、非漢字(？や+などの記号群)として 453 文字、第 1 水準(使用頻度の高い漢字群)として 2965 文字、第 2 水準(比較的使用頻度の低い漢字群)として 3384 文字が含まれます。合わせると、6802 文字分になります。これだけの文字を割り当てるので、2 進数 16 桁(つまり、16 ビット = 2 バイト)を用います。このため、我々のような漢字文化圏のことを、2 バイト文化圏と呼んだりします。

JIS C6226 は、1983 年に名称を X0208 に変更し、その後も改訂が続いています。現在では、JIS X0208:1997 になっています。なお、1990 年には JIS X0212「補助漢字文字セット」が追加されました。これは、印刷業界からの要望によるもので、X0208 の漢字だけでは足りないということから、6067 の漢字と非漢字を新たに追加した規格です。

(2) 記憶機能

コンピュータには、情報を記憶する機能があります。ここでは、その記憶をどのような機構によって実現しているのか、コンピュータで記憶機能を持つ各装置の構成について、それぞれ取り上げます。

記憶の仕組み

コンピュータは、電子回路で構成されています。その最も基本となる電子回路が、基本論理回路です。

基本論理回路には、AND 回路、OR 回路、NOT 回路があります。以下に、それぞれについて概説します。

・ AND 回路

AND 回路は、二つの集合の論理積に相当します。具体的には、二つの入力がいずれも"1"(電流がオン)のときだけ出力が"1"(電流がオフ)になり、それ以外は"0"になる回路です。

図 5 に ,MIL 記号(電子回路の表現として規格制定されている記号)を用いて ,その構成を表します。

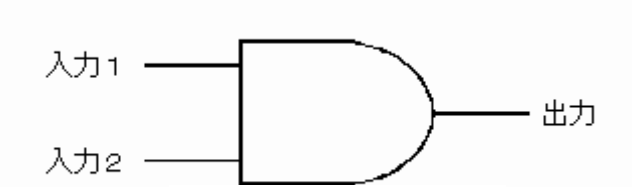


図 5 . AND 回路の構成

・ OR 回路

OR 回路は ,二つの集合の論理和に相当します。具体的には ,一つ以上の入力が" 1 "のときに出力が" 1 "になり ,ともに" 0 "のときに出力は" 0 "になる回路です。図 1 2 に , MIL 記号を用いて ,その構成を表します。

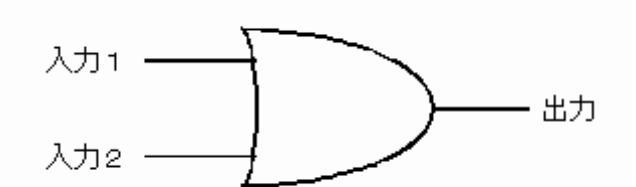


図 6 . OR 回路の構成

・ NOT 回路

NOT 回路は ,補集合に相当します。具体的には ,一つの入力に対して ,つねに逆の出力 (" 0 "ならば" 1 " , " 1 "ならば" 0 ")になる回路です。図 7 に , MIL 記号を用いて ,その構成を表します。

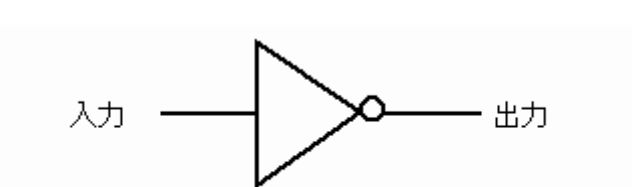


図 7 .NOT 回路の構成

以上の入力と出力の関係を表すと ,表 2 のようになります。この表のことを , 真理値表とも呼びます。これによって ,入力と出力の関係が一目瞭然になります。

表 2 . 基本論理回路の真理値表

入力 1	入力 2	AND 回路	OR 回路	入力	NOT 回路
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1		
1	1	1	1		

これらの基本論理回路を組み合わせることによって、いろいろな論理回路を構成することができます。

論理回路には、組合せ回路と順序回路があります。組合せ回路は、入力された(電流の)状態に応じて出力の状態が決定される論理回路です。順序回路は、過去の状態とある時点での状態によって、次の出力の状態が決定される論理回路です。このため、直前の入力状態を記憶することができる機構を持つことになり、一時的な記憶回路になります。

順序回路の一つに、フリップフロップ回路があります。フリップフロップ回路とは、外部からの入力状態が変わらない限り、0と1の出力状態を保つ論理回路です。このため、一つのフリップフロップ回路で、2進数1桁の記憶ができるわけです。

フリップフロップ回路の構成を、図8に示します。この図では、RとSが入力であり、

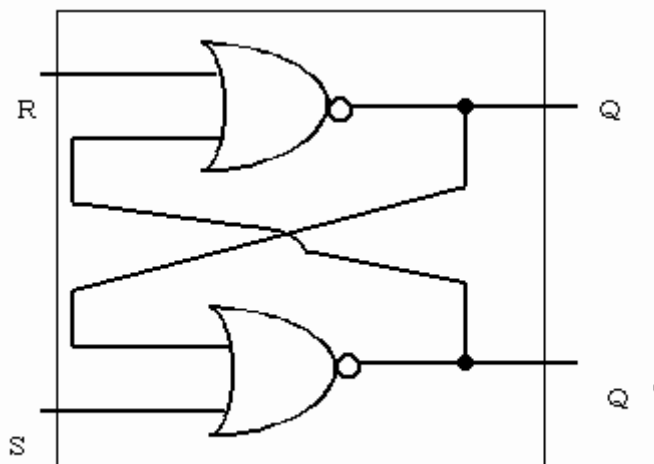


図8．フリップフロップ回路の構成

QとQ'が出力になります。QとQ'は、入力状態が変わらない限り、同じ状態を維持します。一方、入力状態が変わったときに、QとQ'はそれぞれ"0"か"1"のどちらかの値を持ちます。

具体的には、 $S = 1$ (電流が流れている状態)かつ $R = 0$ (電流が流れていない状態)の場合(セット状態)は、以前の状態に関係なく、 $Q = 1$ かつ $Q' = 0$ になります。それに対して、 $S = 0$ かつ $R = 1$ の場合(リセット状態)は、同様に以前の状態に関係なく、 $Q = 0$ かつ $Q' = 1$ になります。

$S = 0$ かつ $R = 0$ の場合は、QおよびQ'に変化が生ぜずに、そのままの状態が維持されます。一方、 $S = 1$ かつ $Q = 1$ の場合は、 $Q = Q' = 1$ になりますが、次の状態で $S = 0$ かつ $R = 0$ になった場合、QおよびQ'の値が確定せずに、どの状態になるか特定できません。このため、未定になります。以上の関係を、真理値表として、表3に示します。

表3．フリップフロップ回路の真理値表

R	S	Q	Q'
0	0	Q	Q'
0	1	1	0
1	0	0	1
1	1	未定	

フリップフロップ回路を組み合わせることによって、記憶装置を作成することができます。それらの中に、レジスタやカウンタがあります。

レジスタには、フリップフロップ回路をいくつか(処理する2進数の桁数分)並列に並べて接続したものや、直列に並べたもの(シフトレジスタとも呼ばれる)などがあります。カウンタは、入力されたパルス数を累計した値を記憶するためのものです。これらは、いずれも一時的な記憶装置として、中央処理装置の中に組み込まれています。

主記憶装置の構成

主記憶装置は、半導体メモリによって構成されています。代表的な半導体メモリには、ROMとRAMがあげられます。

ROM(Read Only Memory)は、読み出しだけが可能なメモリです。つまり、あらかじめ記憶された内容しか読むことができません。その一方で、電源を切っても記憶された内容は消去されないで、不揮発性メモリと呼ばれます。このため、コンピュータの初期化プログラム、入出力を行うBIOS(Basic Input Output System)、あるいは、各種フォントデータなどを、ROMとして組込んでいる製品が多いわけです。

RAM(Random Access Memory)は、読み出しだけでなく、書き込みも可能なメモリです。ただし、電源を切った時点で、記憶されている内容は消去されるので、揮発性メモリと呼ばれます。

さて、図3にも示したように、主記憶装置には、実行するプログラムとデータがローディングされることになります。それらは、いずれも任意のビット列(2進数の並び)で構成されているとともに、それらのビット列は番地(アドレス)によって修飾されています。

番地は、プログラムを構成する命令文を一つずつ取り出すため、および、その命令が使用するデータを取り出すため、メモリ上の格納場所を指定するためのポイントです。たとえば、実行したいプログラムの起動アイコンをクリックすると、メモリ上に配置された該当するプログラムの1行目の命令文が格納されている番地が割り当てられます。これによって、該当のプログラムの実行が始まるわけです。

番地の値は、内部的には2進数になっています。このため、その都度メモリ上の位置を2進数で計算する必要が生じます。この煩雑さを防ぐために、プログラミングの世界では変数を用います。つまり、番地の修飾を、記号化しているわけです。該当する番地を示す変数を用いることによって、番地を求めるための計算が不要になり、その分プログラミングが簡単になるという利点が生じます。

補助記憶装置の構成

補助記憶装置は、その記憶方式によっていくつかの種類があります。

磁気による方式とは、磁性体をその表面に塗り付けた媒体に、磁気ヘッドにより磁気を変化させることで記憶をする方式です。これには、(大型コンピュータでは)磁気ディスクや

磁気テープ、(パーソナルコンピュータでは)ハードディスク、フロッピーディスクなどがあります。いずれの装置も、使用する前に、磁性体の初期化(フォーマット)を行う必要があります。

光磁気による方式とは、磁気とレーザーを併用して記憶をする方式です。これには、光磁気ディスク(MO: Magneto-Optic disk)があります。

光による方式とは、レーザー光を用いて、記憶した内容を読み出す方式です。これには、CD-ROM(Compact Disk - Read Only Memory)、DVD があります。通常の CD-ROM は、前二者と異なり、書き込みができずに読み込みしかできません。ただし、CD-R(CD-Recordable)は追記可能、CD-E(CD-Erasable)は書き換え可能になります。同じく、DVD にも、DVD-ROM のほか、DVD-R、DVD-RAM があります。

(3) 制御機能

ここでは、コンピュータが内部的にプログラムの実行をどのように制御しているかについて、ハードウェア的な側面、および、ソフトウェア的な側面から取り上げます。

ハードウェア的な側面

コンピュータが目的の処理を(自律的に)実行するということは、その処理をプログラムとしてあらかじめ与えている(プログラム内蔵方式)からです。そこで、プログラムの実行における中心的な役割を担う中央処理装置(CPU: Central Processing Unit)のメカニズムについて明らかにします。

コンピュータの本体には、中央処理装置、主記憶装置、(内蔵型)補助記憶装置、電源などが組込まれています。そのうちの中央処理装置が、おもにコンピュータ全体の実行制御をつかさどっているわけです。図9を参考にして、以下を説明します。

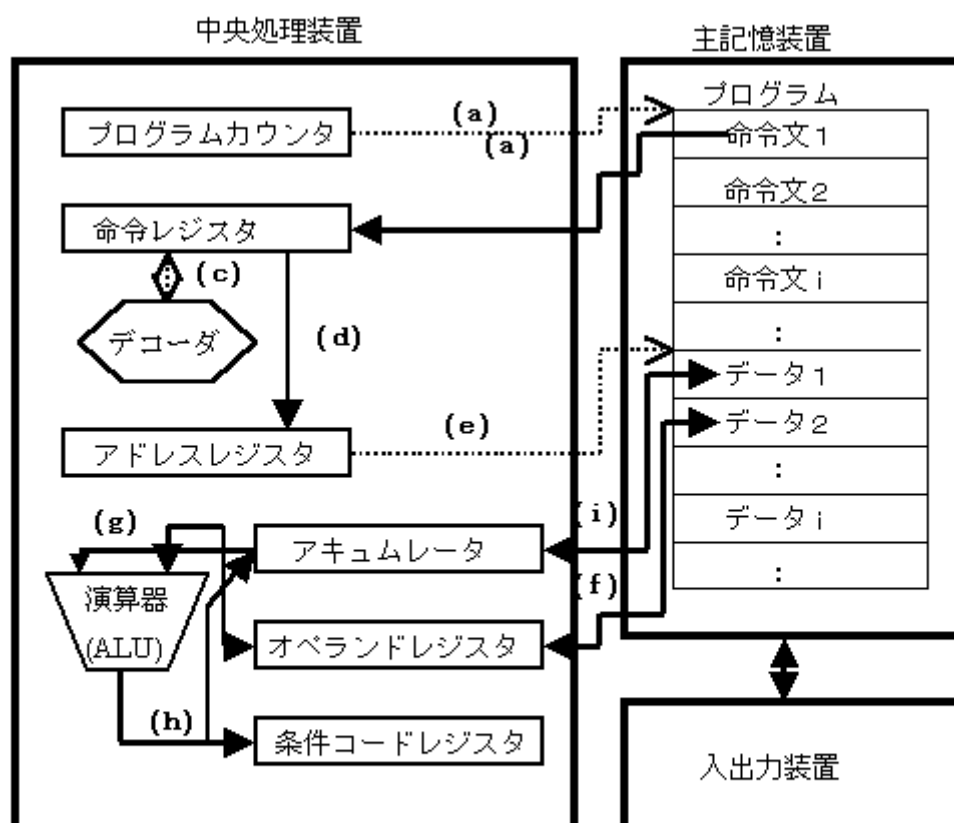


図9．プログラムの実行制御

中央処置装置には、レジスタやカウンタといった一時的な記憶装置が組込まれています。そのうちの一つに、プログラムカウンタがあります。プログラムカウンタには、実行すべきプログラムが記憶されている主記憶装置の先頭番地が設定されます。これによって、実行するプログラムが主記憶装置のどこに格納されているかがわかります。

プログラムカウンタの値にもとづいて、主記憶装置から該当プログラムの第1行目の命令文を取り出します(a)。その命令文を、命令レジスタというレジスタに転送します(b)。命令レジスタに格納された命令文は、命令部とオペランド部によって構成されています。そのうちの命令部は、デコーダによって、命令の種類が解読されます(c)。オペランド部には、その命令が使用するデータの番地が格納されています。それを、アドレスレジスタに転送します(d)。

アドレスレジスタの値にもとづいて、主記憶装置の該当する番地に格納されているデータを取り出します(e)。そのデータを、アキュムレータ(必要ならば、オペランドレジスタ)に転送します(f)。ここで、命令が演算命令の場合、アキュムレータとオペランドレジスタにあるデータを、算術論理演算装置(ALU)に転送して演算を行います(g)。演算結果は、アキュムレータに戻されるとともに、条件コードレジスタに状態コードも戻されます(h)。さらに、アキュムレータにある演算結果のデータは、該当する主記憶装置の番地に戻されます(i)。なお、命令の種類が入出力命令の場合は、入出力装置にアクセスして、データを読み出したり書き込んだりします。

以上の一連の処理が終了すると、プログラムカウンタは、次の命令文 2 の番地を指すように変わります。これは、(命令文 1 の番地) + (命令文の長さ)によって、自動的に計算されます。ただし、命令文が分岐命令や無条件分岐命令の場合は、その分岐先の番地が設定されます。こうして、順番に一つずつ命令が処理され(状態遷移、逐次実行制御)、最終的には停止命令によって、処理が終了することになります。

ソフトウェア的な側面

ソフトウェアを大別すると、基本ソフトウェアと応用ソフトウェアに分けることができます。このうちの基本ソフトウェアが、コンピュータ全体における実行制御を、ソフトウェアによって管理していることになります。

基本ソフトウェアを具体的な製品として見ると、オペレーティングシステムになります。オペレーティングシステムについては、1.3 で取り上げるので、ここでは多少抽象化して、基本ソフトウェアとします。

基本ソフトウェアは、コンピュータ資源を効率的に行うこと、利用者にとって使いやすい環境を提供すること、といった目的のもとに作られています。コンピュータは、さまざまな装置から構成されています。それらの装置を有効的に、かつ、効率的に利用するための仕組みを提供することが、第一の役割としてあげられます。その上で、利用者にとって、コンピュータの詳細な仕組みや動作原理を知らなくても簡便に使いこなせるインタフェースを提供することが、第二の役割としてあげられます。これらの役割を実現することによって、コンピュータは我々にとって快適な道具(電子文房具)になるわけです。

その基本ソフトウェアには、代表的な機能として、タスク管理機能、メモリ管理機能、入出力管理機能、データ管理機能、言語処理機能などが用意されています。以下に、それぞれについて概説します。

・タスク管理機能

タスク管理機能とは、タスクという処理単位を、効率よく管理するための機能です。実行するプログラムは、コンピュータの中でタスクという仕事の単位に分割されます。複数のタスクがある場合は、それらを並行的に実行するために、さらにプロセスという単位で管理します。また、それらのプロセスを実行するときに、実行しているプロセスと実行待ちのプロセスを単位時間毎に分けた単位をスレッドと呼びます。

コンピュータでは、一つの中央処理処置に、同時に一つのプロセスしか実行できません。そこで、複数のプロセスに割り当てる CPU 時間を、それぞれのスレッドに割り当てて、切り換えながら処理を進めるマルチタスク方式も開発されています。

・メモリ管理機能

メモリ管理機能とは、主記憶装置の領域を、効率よく管理するための機能です。主記憶装置には、実行プログラムを一時的に記憶することになりますが、その領域には限界があります。このため、実行したいすべてのプログラムを主記憶装置に格納することは、不可能といえます。そこで、現在では仮想化記憶方式が採用されています。これは、補助記憶装置のある領域が、あたかも主記憶装置にあるかのように仮想的に管理する方式です。主記憶装置だけでなく、補助記憶装置上にも実行プログラムの部分を格納し、必要なときに、主記憶装置に転送して実行を行うことになります。その場合、実行プログラムの分割単位がセグメントかあるいはページかによって、セグメンテーション方式とページング方式が

あります。

- ・入出力管理機能

入出力管理機能とは、各種の入力装置および出力装置に対してアクセスを管理するための機能です。オペレーティングシステムの BIOS の機能に相当します。これには、直接入出力制御方式(初期のコンピュータに実装)、そして、チャンネル制御方式と DMA(Direct Memory Access)方式があります。

チャンネル制御方式は、チャンネルという専用の装置を用いて、主記憶装置と入出力装置間のデータ転送を実現する方式です。その場合において、中央処理装置により、チャンネルは制御(割り込み制御)されることになります。それに対して、DMA 方式は、DMAC という専用の装置を用いて、中央処理装置の介入なしに、直接的に主記憶装置と入出力装置間での高速なデータ転送を実現する方式です。現在では、大型コンピュータがおもにチャンネル制御方式を、パソコンがおもに DMA 方式を採用しています。

- ・データ管理機能

データ管理機能とは、おもに補助記憶装置上に記憶するデータ(ファイルとも呼びます)のアクセスを管理するための機能です。具体的には、各種ディスクの初期化、ファイルの操作(新規作成、表示、印刷、複写、移動、削除、属性変更、圧縮、復元など)、アクセス制御(機密保護)、障害復旧などがあげられます。

- ・言語処理機能

言語処理機能とは、プログラム言語を扱うための機能です。コンピュータへの指示命令は、すべてプログラムによって行います。このために、各種のプログラム言語、および、プログラム言語を処理するための言語プロセッサがあります。

プログラム言語には、機械語やアセンブリ言語といった低水準言語(計算機向き言語とも呼ばれる)から、FORTRAN や COBOL といった高水準言語(問題向き言語とも呼ばれる)まで、数多くのものがあります。低水準言語とは、プログラム言語体系が人間(プログラマ)よりも計算機(コンピュータ)に適しており、人間から見て水準が低い言語といえます。その逆が、高水準言語と言えます。たとえば、COBOL といったプログラム言語は、英語の文を記述するようにプログラミングが行えます。

言語プロセッサとは、プログラム言語をコンピュータが実行可能なように変換する処理系のことです。通常、高水準言語で記述したプログラムは、原始プログラムと呼びます。しかし、原始プログラムのままでは、コンピュータが解釈できません。そこで、トランスレータかインタプリタにより、原始プログラムを解釈します。図 10 に、その構成を示します。

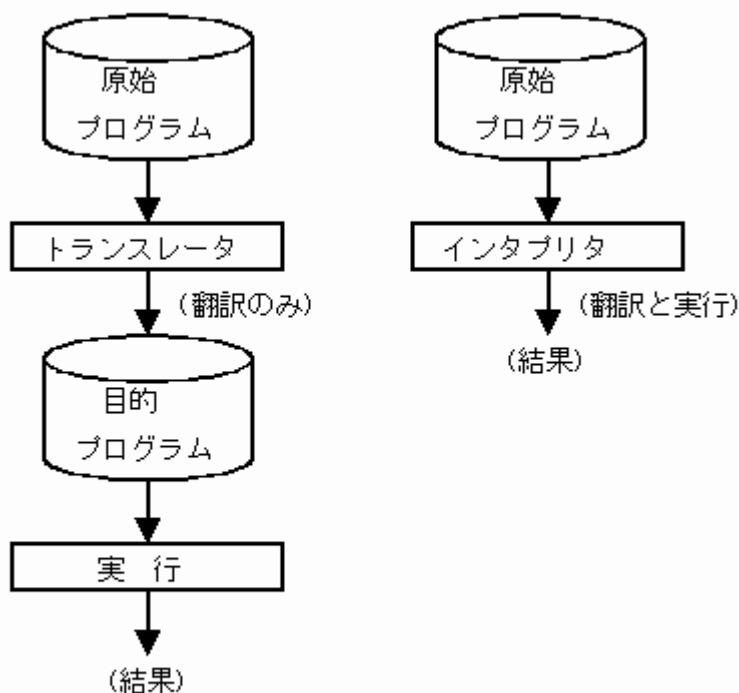


図10．トランスレータとインタプリタ

トランスレータは、原始プログラムを翻訳して、目的プログラムを生成するプログラムです。また、対象とする原始プログラムの種類によって、コンパイラ(高水準言語を対象)とアセンブラ(アセンブリ言語を対象)があります。作成された目的プログラムでは、まだ実行できません。このため、連係編集処理を行う必要があります。その結果、実行可能プログラム(ロードモジュールとも呼ぶ)が生成されます。この実行可能プログラムは、すべて機械語に変換されています。つまり、任意のビット列になっているわけです。これによって、はじめてコンピュータが理解できる言語になるわけです。したがって、どんなプログラムでも、機械語に変換されていることになります。この関係を、図11に示します。

これに対して、インタプリタは、原始プログラムを翻訳すると同時に実行を行うプログラムです。このため、トランスレータよりも簡単に、プログラムの実行ができるわけです。ただし、同時に二つの処理(翻訳と実行)を行っているため、トランスレータよりも処理効率が多少悪くなります。

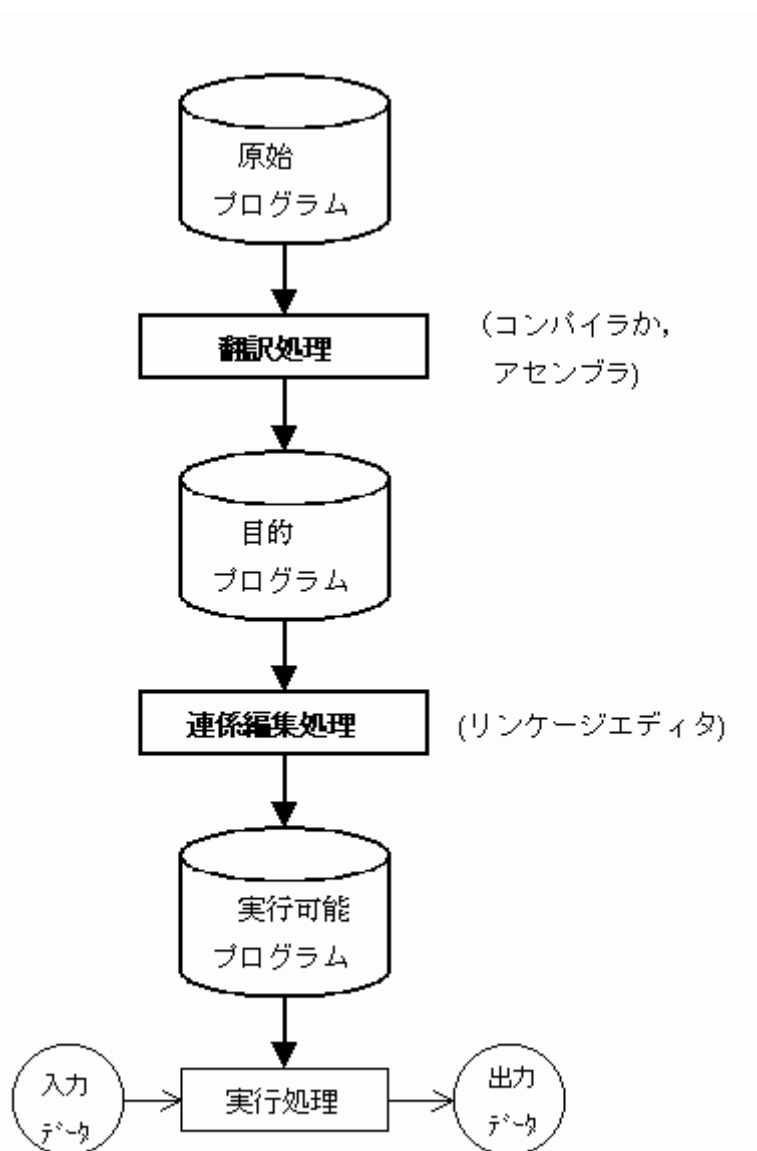


図 1 1 . プログラム処理方式

(4) 演算機能

コンピュータには、数値データを演算する機能があります。その演算には、四則演算(和、差、積、商)だけでなく、論理演算やシフト演算などがあります。ここでは、四則演算のいくつかについて取り上げます。

・和演算

2 進数 1 桁同士の和には、次のパターンがあります。

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10 \text{ (桁上がり)}$$

このように、演算数と被演算数がともに "1" のときだけ、上位への桁上がりが生じます。

・差演算

2進数1桁同士の差には、次のパターンがあります。

$$0 - 0 = 0$$

$$0 - 1 = -1 \text{ (借り)}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

このように、演算数よりも被演算数の方が大きいときには、負の値になります。このため、(10進数と同様に)上位の桁から"1"を借りて演算を行うことになります。ただし、実際のコンピュータでは、2の補数を用いた演算を行います。

・積演算

2進数1桁同士の積には、次のパターンがあります。

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

このように、演算数と被演算数がともに"1"のときだけ、"1"になります。これは、論理積と同じパターンになっているわけです。また、実際には(積演算だけでなく、商演算も)シフト演算を用います。シフト演算では、2進数の各桁を左へn桁移動することによって、演算数を2nで乗算したことになります。同じく右へ桁移動することによって、演算数を2nで除算したことになります。

以上の演算を、コンピュータは電子回路によって処理します。(2)の記憶機能で取り上げた基本論理回路の組み合わせた論理回路には、組合せ回路と順序回路があります。そのうちの組合せ回路の中に、半加算回路や全加算回路、あるいは、補数回路があります。これらの論理回路を用いて、コンピュータは演算処理を行っています。

半加算回路は、和演算を行う際に、下位桁からの桁上りを無視して、上位桁への桁上がりだけに対応した論理回路です。そのため、2進数1桁の加算しかできないので。半加算回路と呼ばれます。半加算回路の構成を、図12に示します。また、半加算回路の真理値表を、表4に示します。

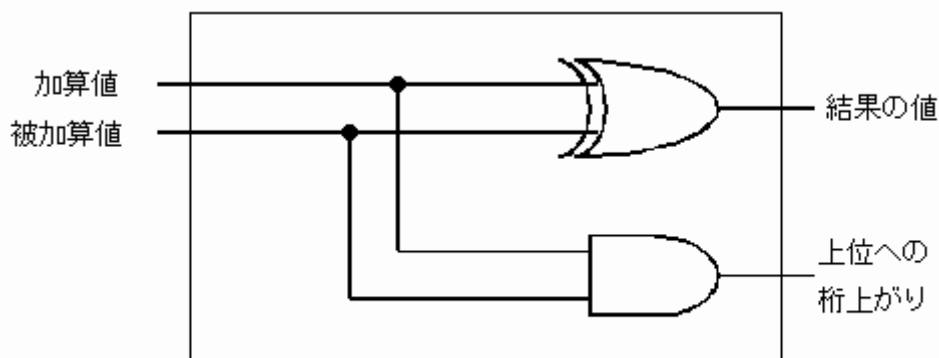


図12．半加算回路の構成

表 4 . 半加算回路の真理値表

加算値	被加算値	結果の値	上位への 桁上がり
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

これに対して，全加算回路は，上位桁への桁上がりだけでなく，下位桁からの桁上がりまで対応できる論理回路です。このため，2進数n桁の加算ができるので，全加算回路と呼ばれます。全加算回路の構成を，図 13 に示します。また，全加算回路の真理値表を，表 5 に示します。

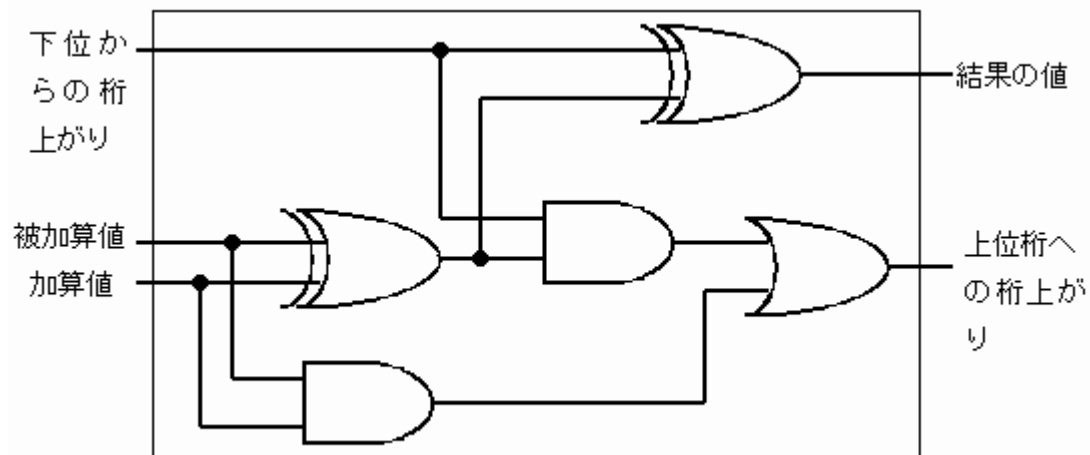


図 13 . 全加算器の構成

表 5 . 全加算回路の真理値表

加算値	被加算値	下位からの桁 上がり	結果の値	上位への桁 上がり
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

(5) 出力機能

コンピュータで処理した結果は，我々が理解できる形で表示されます。ここでは，フォントシステムと色の出力について取り上げます。

フォントシステム

個々の文字が持つ形のことを、字形と呼びます。英数字や片仮名あるいは平仮名や漢字などの字種に対して、統一的に決めた字形の集まりを、字体(フォント)と呼びます。これには、明朝体やゴシック体や教科書体などがあります。通常、明朝体は本文に、ゴシック体は見出しや強調箇所に、それぞれ使用することが多いといえます。多くのワードプロセッサでは、これらの字体を1字毎に指定できるようになっています。

字体の大きさのことを、ポイントと呼びます。1ポイントは、1/72.27 インチの大きさになります。このため、72 ポイントの字体が、縦と横それぞれ約1インチの大きさになります。

日本語フォントについては、いくつかの方式が開発されています。代表的なものに、ビットマップ方式とベクトル方式があります。

ビットマップ方式は、字体を縦横のドットに分けて、各ドットを黒色で塗るか否かで構成する方式です。これらのドットのマトリックスを、ビットマップフォントと呼びます。たとえば、1.2の図8は、4×4ドットのビットマップフォントに相当します。多くのパソコンのディスプレイでは、16×16ドット(高解像度ディスプレイでは、24×24ドット)を用いています。ただし、この方式では、字体を拡大すると、斜線にギザギザが生じて見にくくなる場合があります。そこで、次のベクトル方式が開発されています。

ベクトル方式は、文字の輪郭をベクトル曲線によって表す方式です。文字の輪郭線であるアウトラインを、線や弧や曲線の集合として数学的に表したもので、アウトラインフォントと呼びます。これによって、あらゆる字体の大きさ(ポイント)と解像度(ドット)に対応できます。

これらのフォントシステムによって、出力装置の解像度が決定されます。ここでは、ディスプレイとプリンタに分けて取り上げます。

・ディスプレイ

ディスプレイの解像度は、画面の「横方向のドット総数×縦方向のドット数」によって表します。たとえば、16×16ドットの日本語フォントでは、640×400という解像度は、横40字(640÷16)で縦25行(400÷16)の表示特性(ノーマルモード)を表しています。これに対して、1024×768という解像度は、横64行(1024÷16)で縦48行(768÷16)の表示特性(ハイレゾリューションモード)を表しています。

・プリンタ

プリンタの解像度は、1インチ当たりのドット総数によって表します。これには、dpi(dot per inch)という単位を用います。たとえば、パーソナルプリンタの解像度は、240dpiから400dpi程度になっています。これに対して、レーザプリンタの解像度は、600dpiから720dpi程度であり、より鮮明な印刷が可能になっています。

カラーの表示

カラーを表示する機構は、ディスプレイとプリンタでは基本的に異なる部分があります。それは、ディスプレイは光の3原色(赤色：R，緑色：G，青色：B)を用いてカラーを表示するのに対して、プリンタは色の3原色(水色：C，紫色：M，黄色：Y)を用いてカラーを表示する点です。そこで、ここでも分けて説明します。

・ディスプレイ

光の3原色に相当する RGB 信号を用いて、カラーを表示します。これらの3色を混合することによって、8色まで表すことができます。表6に、その関係を示します。

表6．RGB の混色

赤	緑	青	混色
×	×	×	黒色
×	×		青色
×		×	緑色
×			黄色
	×	×	赤色
	×		紫色
		×	水色
			白色

これは、それぞれの色を1ビット(か×か)に対応させると、3色で3ビットになることから、

$$2^3 = 8$$

より、8色まで扱うことを示しています。

また、各色の階調(発光するか、しないか)を取る(つまり、色の明るさ)ことによって、表示できる色数の範囲が多くなります。たとえば、赤色と青色の明るさを5ビット(つまり、5段階の明るさ)に、緑色の明るさを6ビット(合計で、16ビット)で表すとすると、

$$2^5 \times 2^5 \times 2^6 = 65536 \text{ 色}$$

になります。ただし、これでは、写真の色をきれいに表示することが難しいといえます。

そこで、各色の階調を8ビット(つまり、256段階の明るさ)に設定すると、

$$(2^8)^3 = 2^{24} = 16777216$$

となり、約1677万色の色(フルカラー)を表示することになります。こうなると、写真を十分きれいに表示することができます。

・プリンタ

色の3原色に相当する CMY、あるいは、CMYK(すべての色を混ぜると黒：Kに)、をそれぞれ用いてカラーを印刷します。ただし、CMYK については、カラー用のインクと、黒色専用のインクを分けることもできます。これは、黒をよりはっきり印刷するため、ある

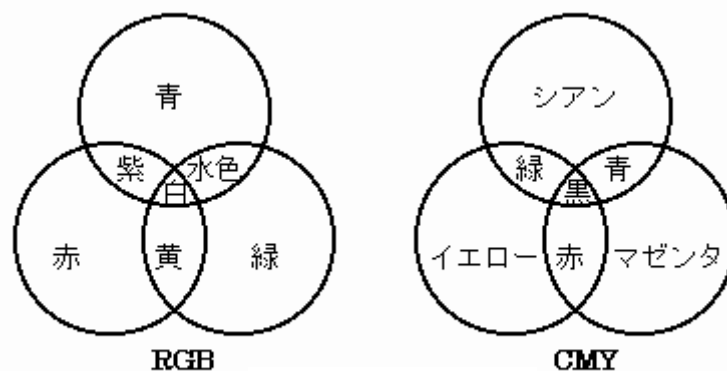


図14．色の比較

いは、黒色での印刷(文字中心)が多く黒色のインクの減りが早いことからこれだけを差し換えるためなどの理由によります。

なお、RGB と CMY の関係は、図 1 4 のようになります。これより、色調を補正する場合には、補色(反対色)を考慮する必要があります。たとえば、CMY のマゼンタはの補色は、シアンとイエローを混ぜた緑になります。また、プリンタを購入すると、プリンタドライバというソフトウェアが添付されてきます。これは、ディスプレイで表示されている RGB から CMYK へ変換する処理を行うソフトウェアなのです。

【練習問題】

次の漢字フォントの容量(キロ Byte)で計算せよ。

- ・JIS 第 1 水準：3,000 字とする
- ・JIS 第 2 水準：3,500 字とする
- ・フォントは、24×24 ドットとする

1.4 コンピュータシステム

1.3 までは、コンピュータの機能要素について説明しましたが、ここでは、それらの機能が実際の製品にどのように対応しているのかについて取り上げます。具体的には、図 1 5 のような 1 台のパソコンを想定して、ハードウェアおよびソフトウェアの両面から対応づけることにします。

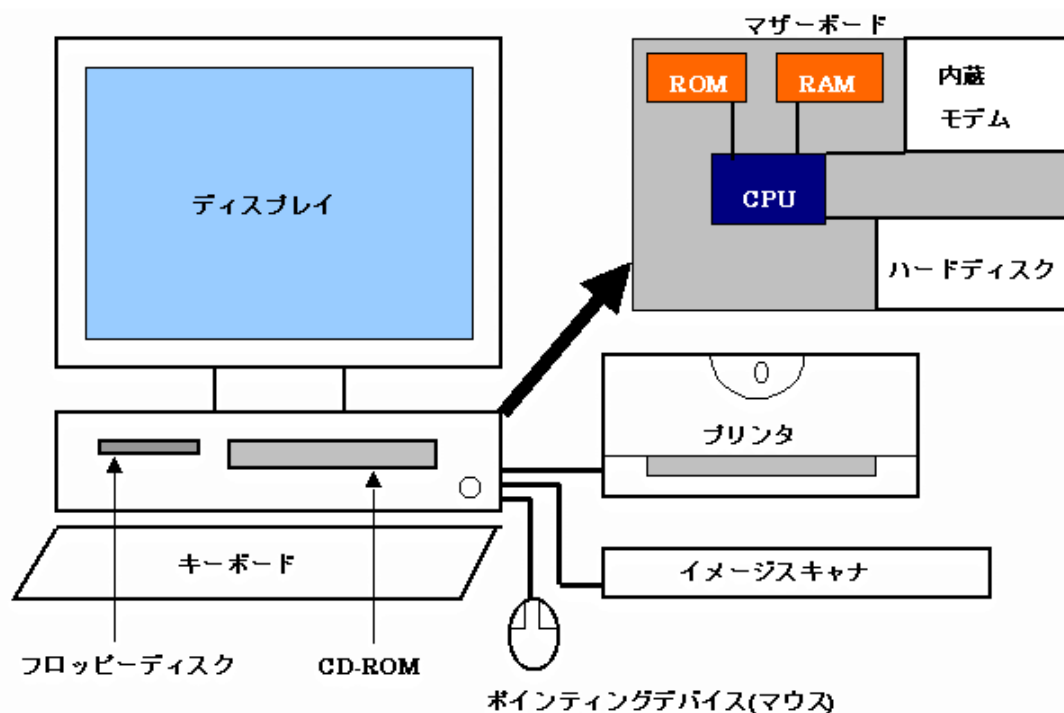


図 1 5 . パソコンの構成

(1) ハードウェアシステム

コンピュータシステムは、複数の装置群から構成されています。その基本的なモデルとして論理機械が位置づけられます。それは、状態と記号によって、その動作が決定されることでもあります。その論理機械に対して、実装のためのアイデアを取り込んだものがノイマン型コンピュータです。それとともに、ノイマン型コンピュータを、IPO というモデルによって機能単位に分けることができます。以上の各機能を、コンピュータのハードウェアを構成する個々の装置として実現したものが、パソコンのハードウェアシステムに相当します。以下に、それぞれについて取り上げます。

中央処理装置(CPU)

中央処理装置は、IPO モデルの制御機能と演算機能をハードウェア的に実現するための装置です。図 15 に示したように、中央処理装置にはレジスタやカウンタ、デコーダや演算器(ALU)などがあります。これらを用いて、コンピュータのさまざまな動作機構を制御するわけです。

CPU を構成するマイクロプロセッサについては、いくつかの企業から製品が出荷されています。これらマイクロプロセッサの製品の物理的な特性が、コンピュータシステム全体としての性能を決定づける要因になります。

その物理的な特性の一つに、マイクロプロセッサを構成するビット数(データを表現するビット幅のこと)があります。現在は 32 ビットが主流ですが、今後は徐々に 64 ビットに移行していくことでしょう。もう一つに、マイクロプロセッサがもつクロック周波数もあげられます。クロック周波数とは、CPU が一動作を行うサイクル時間の単位を決定する周波数のことです。この周波数が高いほど、性能が向上していることになります。現在では、百数十 MHz(メガヘルツ)のものが主流です。ただし、技術的には 1 GHz(ギガヘルツ)までが限界だろうと言われています。

主記憶装置(RAM と ROM)

主記憶装置は、IPO モデルの記憶機能の一部を実現するための装置です。CPU が直接アクセスするプログラムやデータを記憶しているところが、主記憶装置になります。

主記憶装置の多くは、DRAM により構成されます。DRAM(Dynamic Random Access Memory)とは、コンデンサとトランジスタを 1 個ずつ用い、その電荷の有無によって 1 ビットの記憶を行う RAM です。集積度が高く大容量の記憶が可能な反面、再充電やリフレッシュ(自己放電のための再書き込み)動作が必要になります。

この DRAM は、揮発性メモリであり、電源を切ると記憶内容が消去されます。このため、コンピュータの初期化プログラム(BIOS)などは ROM に格納されています。これによって、コンピュータの電源を入れると、ROM にある BIOS を読み込みシステムの初期化を行ってから、RAM をアクセスすることになります。

主記憶装置に関しても、コンピュータシステムの性能を決定づける特性があります。それは、主記憶装置そのものの容量です。この容量が大きいほど、より大きなサイズの実行可能プログラムをより多く格納することができるからです。最近の傾向として、基本ソフトウェアを始め各種応用ソフトウェアのプログラムのサイズは、豊富な機能を実現するために大きくなる一方です。このため、メモリの大きさも、コンピュータシステムの性能を表す尺度基準の一つになります。現在では、64Kbyte(キロバイト)が主流になりつつありま

す。

バス

バス(bus)は本来「乗物」という意味ですが、コンピュータの世界では「通路」という意味になります。つまり、CPU と各装置間の転送路ということです。このため、バスの数は、CPU のビット数に対応します。たとえば、32 ビットのマイクロプロセッサをもつ CPU には、1 ビットの情報を転送するバスが 16 本あることになります。この結果、2 の 16 乗の情報を転送することになります。

バスには、内部バス(CPU 内部の通路)と外部バス(CPU と外部装置間の通路)があります。このうちの外部バスには、アドレスバス、データバス、コントロールバスがあります。アドレスバスは、主記憶装置や入出力装置の番地を転送する通路です。データバスは、データを転送する通路です。コントロールバスは、制御信号を転送する通路です。

補助記憶装置

補助記憶装置は、IPO モデルの記憶機能の一部を実現するための装置です。これらの装置に記憶してあるプログラムやデータは、主記憶装置に読み込まれることによって、はじめて実行されます。これには、その記憶方式や容量の大きさによって、次のような装置があります。

・ハードディスク

磁気による記憶方式を持つ補助記憶装置です。その記憶容量は大きくなる一方で、数 GByte(ギガバイト)のオーダーになっています。

・フロッピーディスク

磁気による記憶方式を持つ補助記憶装置です。比較的容量の小さいもので、1.44MByte あるいは 1.2MByte の容量を持つ 2HD(両面高密度倍トラック)フォーマットが主流になっています。

・MO(Magneto-Optic)ディスク

磁気とレーザー光を併用した記憶方式を持つ補助記憶装置です。その記憶容量は、数百 MByte であり、マルチメディアなどのコンテンツを記憶するのに適した装置といえます。

・CD-ROM(Compact Disk-Read Only Memory)と DVD

レーザー光による記憶方式を持つ補助記憶装置です。MO よりも容量が多く、より大容量な補助記憶装置として利用できます。最近のソフトウェアのほとんどは、CD-ROM によって流通しています。このため、多くのパソコンには CD-ROM ドライブが装備されるようになっています。

CD-ROM は、オーディオ用 CD と同じ仕様を持つ Yellow Book 仕様を採用しています。これには、追記が可能な CD-R(Recordable)、書き換え可能な CD-E(Erasable)、カメラで撮影したフィルムの画像を記憶できる Photo CD、ビデオを CD 上に記憶できる Video CD などもあります。

DVD(以前は Digital Video Disk の略称としたが、現在は固有名詞)は、CD や CD-ROM との互換性を持ち、動画像をレーザーディスク以上の品質で記憶できる媒体です。これには、パソコン用の ROM ドライブである DVD-ROM、書き換え可能な DVD-RAM、追記型の DVD-R などがあります。

入力装置

入力装置は、IPO モデルの入力機能を実現するための装置です。いずれもコンピュータの外部から入力される情報を、内部的には2進数に変換して処理することになります。これには、入力方式の違いによって、次のような装置があります。

・キーボード

欧米を中心に用いられていたタイプライタが、その原型といえます。該当するキーを入力すると、ある電気信号に変換されて、コンピュータの本体で対応する2進数に符号化されます。キーの配列には、いくつかの種類があります。最も普及しているのが米国のQWERTY配列を踏襲した(旧)JIS配列であり、そのほかに仮名キーを4段ではなく3段にまとめた(新)JIS配列や、富士通が独自に開発した親指シフト配列、五十音順に仮名を配列した五十音配列などがあります。

・ポインティングデバイス

最も代表的なものが、マウスです。ねずみのような形状をしていることから、このような名前がつけられました。マウスは、マルチウィンドウシステムの普及とともに、アイコンのクリックやプルダウンメニューの選択などの入力用装置として使われるようになりました。マウス以外には、トラックボールやタッチスクリーン、ペン入力などがあります。

・イメージスキャナ

写真や絵などの画像データを、光学的に読み取ってデジタルデータ(通常は、ビットマップ形式)に変換する装置です。そのときの性能を示す単位に、1インチに何ドット表示できるかを表すDPI(Dot Per Inch)という解像度があります。

出力装置

出力装置は、IPO モデルの出力機能を実現するための装置です。いずれもコンピュータの内部で処理されたデータを、我々人間にわかるように表示することになります。これには、出力方式の違いによって、次のような装置があります。

・ディスプレイ

電子ビームを走査させ蛍光体を発色させることで情報を表示するCRT(Cathode Ray Tube)ディスプレイと、偏光板ではさんだ液晶に電圧をかけて偏光させることで情報を表示する液晶ディスプレイがあります。

・プリンタ

紙媒体に印刷する装置です。これは、印刷の際に物理的な接触があるかないかで、インパクトプリンタとノンインパクトプリンタに分類されます。前者では、ドットインパクト式があります。後者には、インク液をノズルから噴射して印刷するインクジェット式、サーマルヘッドで感熱紙に焼きつけて印刷する感熱式、感熱インクリボンを用いて普通用紙に焼きつけて印刷する熱転写式、光導電体を表面に塗った感光ドラムにレーザービームをあててトナーを付着させて印刷するレーザービーム式、レーザービームの代わりに発行ダイオードを用いるLED(Light Emitting Diode)式などがあります。

通信装置

通信装置は、コンピュータ同士で情報を送受信するための機能を実現する装置です。これには、次のような装置があります。

・モデム

日本語では、変復調装置と呼びます。コンピュータなどのコンピュータは、すべてデジタル方式です。それに対して、一般に使われる電話回線は、アナログ方式です。このため、電話回線を用いてパソコン通信やインターネットを使用しようとする、デジタルとアナログを相互に変換する必要が生じます。デジタル信号をアナログ信号に変換することを変調(modulator)、アナログ信号をデジタル信号に変換することを復調(demodulator)と呼びます。このため、モデムは、これらの言葉を合成した造語になるわけです。

モデムの性能を示す尺度としては、通信速度があります。1秒間に何ビット転送できるかを表すbps(bit per second)です。最近では、33.6Kbps から 56Kbps までもが製品化されています。なお、パソコンとモデムを接続するには、RS-232C インタフェースを用います。

・DSU と TA

最近では、電話回線そのものをデジタル化されています。そのような回線にパソコンを接続する場合、モデムは不要になりますが、DSU(Digital Service Unit)というデジタル回線接続装置と、TA(Ternimal Adapter)が必要になります。これらを用いることによって、インターネット上のホームページをより高速に閲覧することが可能になります。

(2) ソフトウェアシステム

ノイマン型コンピュータを IPO モデルによって機能化した際に、制御機能に相当する部分が、パソコンのソフトウェアシステムになります。ここでは、基本ソフトウェアとしてのオペレーティングシステム、応用ソフトウェアとしての各種アプリケーションソフトウェアについて取り上げます。

オペレーティングシステム

オペレーティングシステムは、IPO モデルの制御機能をソフトウェア的に実現するためのソフトウェアです。オペレーティングシステムとしての製品には、さまざまなものが販売されているが、一般的には中核となるシェル(コマンドプロセッサ)システム、タスク管理システム、メモリ管理システム、ファイルシステム、デバイス管理システム、GUI(Graphical User Interface)システム、言語処理システムなどから構成されます。これらは、いずれも機械語に変換された実行プログラムとして、主記憶装置のオペレーティングシステム常駐領域にあらかじめローディングされています。

シェルシステムは、コマンドの入力により対応する処理プログラムを起動する働きを持つプログラム群です。タスク管理システムは、オペレーティングシステムのタスク管理機能を実現するプログラム群です。メモリ管理システムは、オペレーティングシステムのメモリ管理機能を実現するプログラム群です。ファイルシステムは、オペレーティングシステムのデータ管理機能を実現するプログラム群です。デバイス管理システムは、オペレーティングシステムの入出力管理機能を実現するプログラム群です。GUI システムは、利用者にとって操作性のよいインタフェース(マルチウィンドウ、アイコン、プルダウン/ポップアップメニューなど)を提供する機能を実現するプログラム群です。言語処理システムは、オペレーティングシステムの言語処理機能を実現するプログラム群です。

アプリケーションソフトウェア

利用者が実現したいさまざまな適用業務の処理を，ソフトウェアとして提供するものです。ソフトウェア技術がまだそれほど進展していなかった時代では，利用者は自分でプログラミングをしない限りコンピュータを利用することができない状況にありました。その後，多くのソフトウェアが開発され，これらを目的に合わせて選択し利用することによってプログラミングの必要がなくなったわけです。これが，パソコンを普及させた原動力になったわけです。

アプリケーションソフトウェアには，数多くのものがあります。ここでは，最もよく利用されている代表的なソフトウェアだけを取り上げます。これら以外にも，特定の適用処理処理用のソフトウェア(多くは，パッケージ化)などもあります。

・日本語フロントエンドプロセッサ

仮名あるいはローマ字をキーボードから入力することによって，漢字に変換するソフトウェアです。これが開発され安価に提供されたことによって，欧米でアルファベットが簡単に処理できたように，我が国でも仮名漢字が簡単に処理できるようになったわけです。

あらゆるソフトウェアに組込む(利用者から見ると，そのソフトウェアの前に位置するので，フロントエンドと呼ぶ)ことによって，そのソフトウェアで日本語を扱うことができるようになっています。

・ワードプロセッサ

日本語の文章の編集処理を行うソフトウェアです。最近では，単なる文書整形だけではなく，推敲機能や校正機能，あるいは，DTP(DeskTop Publishing)機能などが付加されています。

・表計算ソフト

行と列で構成される表形式のスプレッドシート上に，データを入力してさまざまな計算(関数，マクロ使用)やグラフ化する機能を持つソフトウェアです。

・データベースソフト

データを格納して検索する機能を実現するソフトウェアです。パソコンのデータベースソフトでは，カード型あるいは関係型データベースが主流になっています。関係型データベースは，ある意味では表計算の表と同じ様式になりますが，異なる点として関係データベース言語 SQL が扱えることがあげられます。

・描画ソフト(ペイント系，ドロー系)

絵や画像を作画したり編集加工する機能を持つソフトウェアです。このうちのペイント系ソフトは点描形式で，ドロー系ソフトは線形式で，それぞれ作画し編集するものです。

・通信ソフト

パソコン通信やインターネットといった通信を実現する機能を持つソフトウェアです。インターネットに関しては，電子メールを送受信するソフトウェアや，ホームページを閲覧するソフトウェアなどがあります。