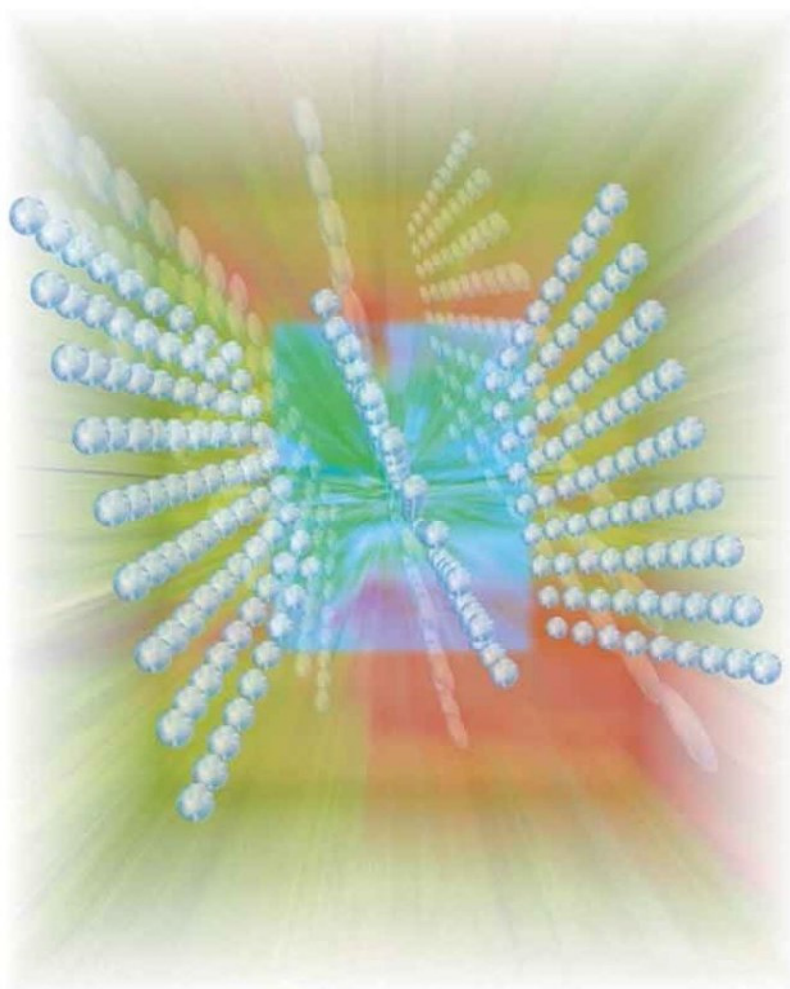


Webで学ぶ 情報処理概論

改訂版



上山 清二 著

ふくろう出版

まえがき

2011年度末、パーソナルコンピュータの所帯保有率は77.4%、インターネットの所帯普及率は79.1%に達しました*3。10軒のうち約8軒にパーソナルコンピュータがあり、インターネットを利用していることとなります。

パーソナルコンピュータは「情報機器」ですが、近頃はむしろ「文房具」と思うことが多くなりました。鉛筆やノートであり、絵筆であり、定規やコンパスでもあります。あるいはまた電卓であったり、書籍や辞書でもあったり…。このような傾向は今後ますます強くなりそうで、パーソナルコンピュータのない生活は考えられなくなりつつあります。

本書はそのコンピュータのしくみを、特に「なぜ機械に計算ができるのか」ということに重点を置いて説明しています。コンピュータの歴史のページでは、当時のコンピュータの開発者たちが、メモリを作るのにどんな工夫をしたかということにも触れています。コンピュータでいちばん重要なのは、演算処理を司る中央処理装置（CPU）と主記憶装置（メモリ）だからです。

そのため特に「第2章 コンピュータの原理」などは、文科系の学生には難しく見えるかもしれません。実際にはそうでもないのですが、論理回路の図などは近寄りがたく思えます。分かりやすく説明しているつもりですが、往々にして文章の説明は取っつきにくく、分かってしまえば「何だ」というようなことも、なかなか理解できないものです。

そこで理解を助けるために、ウェブページを用意しました。

ウェブページ「Web で学ぶ 情報処理概論」（<http://www.infonet.co.jp/ueyama/ip/>）

を開いて（「情報処理概論」で検索できます）「目次」のページから、たとえば「2進数と16進数、10進数」に進んでください。このページの2進数をマウスでクリックすると、0は1に、1は0に変わっているような2進数を作ることができます。同時にそれが10進数や16進数*4に変換されて表示されますから、何度か試しているうちに自然に2進数が理解できるようになります。

この動く教材は、Java というプログラミング言語で書かれたプログラム（Java アプレット）で、ウェブページに貼り付けて動かせるという特徴があります。ウェブ上に動く図や動画はたくさんありますが、インタラクティブに動くものは希です。まして教材となればなおさらですが、ウェブページ「Web で学ぶ 情報処理概論」には、これがたくさんあります。

書籍の文章や図による説明ではなかなか理解できず、これまで「理数系は苦手」、「難しい」、「分かるはずがない」と思いこんでいた人も、Java アプレットを使うと容易に理解できることに驚くかもしれません。「何だ、そういうことか」と、ひとつ分かればしめたものです。

*3 出典：総務省 平成23年通信利用動向調査。

*4 「16進数」（21ページ参照）。

本書では関連するウェブページを次のような形で案内しています。枠内をクリックするとウェブページが開きますから、試して下さい。

Web で学ぶ 【2 進数と 16 進数、10 進数】



Web ページ「2 進数と 16 進数、10 進数」では、数字が 2 進数、16 進数、10 進数で表示されます。2 進数の数字をマウスでクリックして数値を変えたり、自動的に数値を大きくしたり小さくしたりすることもできます。2 進数とはどういうものか、直感的に理解できます。

ところで、本書はほぼ 100%、パーソナルコンピュータを使って作成しました。

文章は言うまでもなくテキストエディタ*5 で書いています。ああでもない、こうでもない。何度も書いたりけしたりするのにコンピュータはもってこいです。図の大部分はグラフィックソフト*6 で描きました。きちんとした図も下手なマンガも、いずれもマウスをぐるぐり動かして描きました。電気信号波形の図は、オシロスコープ*7 をパーソナルコンピュータにつないで作りしました。グラフは表計算ソフトです。写真はデジタルカメラで撮ったり、スキャナで読み込んだり、ウェブサイトから引用させて戴いたり…。顕微鏡写真も、顕微鏡に USB カメラを取り付けてコンピュータで処理しています。

そして、それらすべてを本書の体裁にまとめてくれたのは、DTP ソフト*8 です。組版はもちろん、章や節、図や表の番号の管理から目次、索引の作成にいたるまで、雑事は全部引き受けてくれました。

つくづく、コンピュータってすごいです。これらがなければ本書は完成していません。

私が情報処理概論の講義をはじめたとき、テキストとして金谷信之先生の「情報処理論」（晃洋書房）を使わせて戴きました。やがて先生がご退職され、しばらく経つと、「情報処理論」も実情に合わないが増えてきましたので、重い腰を上げて本書の執筆に取りかかりました。「情報処理論」からはあらゆることを参考にし、あるいは引用させていただきました。先生のご好意とお励ましがなければ、本書はやはり完成していません。厚く御礼申し上げます。

2012 年 11 月 上山 清二

*5 テキストエディタは文字データを入力・編集するためのソフトウェア。使用したのは「秀丸」（シェアウェア）。

*6 グラフィックソフトは画像データを編集するソフトウェア。99 ページ参照。使用したのは「Paint Shop Pro」。

*7 たとえば「図 3.6 リフレッシュ」（52 ページ）。オシロスコープは、電圧の変化を目で見るための測定器です。

*8 DTP は机上出版（desk top publishing）。使用したのは「L^AT_EX」。特に理工系の文書の作成に適したソフトで、フリーウェアです。「デスクトップパブリッシング」（101 ページ）参照。

目 次

第 1 章 コンピュータ	1
1. コンピュータとは	1
1.1. コンピュータは計算機	1
1.2. 計算機からコンピュータへ	2
1.3. コンピュータはスイッチでできている	2
1.4. コンピュータは左脳の	3
1.5. コンピュータは情報処理装置	4
2. コンピュータの種類	5
2.1. メインフレーム	5
2.2. スーパーコンピュータ	5
2.3. ミニコンピュータ	6
2.4. ワークステーション	7
2.5. パーソナルコンピュータ	7
2.6. マイクロコンピュータ	7
3. コンピュータの構成	8
4. 情報	10
4.1. 情報量	10
4.2. アナログとデジタル	12
第 2 章 コンピュータの原理	14
1. 2進数	14
1.1. なぜ 2 進数なのか	14
1.2. 2 進数	16
1.3. 10 進数から 2 進数への変換	17
1.4. 2 進数から 10 進数への変換	18
1.5. 符号付き 2 進数	18
1.6. 2 進数の加算	19
1.7. 2 進数の減算	20
1.8. 16 進数	21
2. キャラクタコード	22

2.1.	ASCII	22
2.2.	JIS コード (JIS X 0201)	23
2.3.	JIS 漢字コード (JIS X 0208)	24
2.4.	シフト JIS 漢字コード	25
3.	論理回路	28
3.1.	基本論理回路	28
3.2.	論理回路の応用	31
4.	半導体	35
4.1.	半導体とは	35
4.2.	N型半導体と P 型半導体	35
4.3.	ダイオード	37
4.4.	トランジスタ	38
4.5.	MOS トランジスタ	40
4.6.	集積回路	42
5.	コンピュータのしくみ	43
5.1.	手動式電子計算器	43
5.2.	仮想コンピュータ	45
5.3.	プログラム	47
第 3 章	ハードウェア	49
1.	中央処理装置	49
1.1.	クロック	50
1.2.	CPU のビット数	51
2.	主記憶装置	52
2.1.	DRAM	52
2.2.	ROM	53
3.	補助記憶装置	54
3.1.	ハードディスク	54
3.2.	CD	57
3.3.	DVD、ブルーレイ・ディスク	60
3.4.	USB メモリ、メモリカード	61
4.	入力装置	62
4.1.	キーボード	62
4.2.	マウス	63

4.3.	イメージスキャナ	65
4.4.	タッチパネル	65
4.5.	バーコードリーダー	66
5.	出力装置	68
5.1.	ディスプレイ	68
5.2.	プリンタ	72
5.3.	フォント	76
第4章	ソフトウェア	79
1.	プログラミング言語	79
1.1.	低水準言語	80
1.2.	高水準言語	80
2.	プログラム	83
2.1.	VBA	83
3.	オペレーティングシステム (OS)	90
3.1.	OS とは	90
3.2.	OS の起源	91
3.3.	OS の機能の進化	92
4.	アプリケーションプログラム	95
4.1.	アプリケーションプログラムの種類	95
4.2.	主なアプリケーションプログラム	96
第5章	コンピュータの歴史	106
1.	機械式計算機	106
1.1.	パスカルの計算機	106
1.2.	ライプニッツの計算機	107
1.3.	オドナーの計算機	107
1.4.	バベジの階差機関	108
1.5.	解析機関	109
1.6.	ツェーの Z1	109
2.	チューリングマシン	111
3.	電気 (リレー) 式計算機	113
3.1.	Z3	113
3.2.	ハーバード Mark-I	114
4.	電子計算機の誕生	115

4.1. ABC	115
4.2. Colossus	116
4.3. ENIAC	117
4.4. The Baby	118
4.5. EDSAC	119
4.6. メモリの比較	120
4.7. ノイマン型コンピュータ	121
4.8. 最初のコンピュータ	121
5. 第2次世界大戦とコンピュータ	124
6. 商用コンピュータの出現	126
6.1. UNIVAC I	126
6.2. IBM 701, 650, System/360	126
7. 米ソ冷戦、宇宙開発競争とコンピュータ	127
7.1. スプートニク・ショック	127
7.2. アポロ計画	128
7.3. アポロ計画と半導体技術の発達	129
8. 電卓戦争	130
8.1. 黎明期の電卓	130
8.2. 電卓のIC化	130
8.3. マイクロプロセッサの誕生	131
8.4. 電卓戦争の終焉 ^{しゅうえん}	132
9. パーソナルコンピュータの誕生	134
9.1. ALTAIR 8800	134
9.2. Apple I	135
9.3. TK-80	135
9.4. Apple II	136
9.5. PC-8001	136
9.6. IBM PC	137
9.7. PC-9801	137
9.8. Macintosh	138
10. インターネットの成立	140
10.1. ARPANET	140
10.2. WWW	141
10.3. WWW から Mosaic、Netscape へ	142

10.4. ブラウザ戦争	143
10.5. インターネットの普及	143
第6章 ネットワークとインターネット	145
1. ネットワーク	145
1.1. LAN	145
1.2. WAN	150
1.3. PAN	150
2. インターネット	151
2.1. インターネットの現況	151
2.2. インターネットへの接続	154
2.3. IP アドレス	159
2.4. ドメイン	159
2.5. DNS	163
2.6. ルータ	164
2.7. WWW	165
2.8. 電子メール	169
第7章 生活と情報技術	171
1. 家中どこにも、コンピュータ	172
1.1. 電気炊飯器	172
1.2. シーリングライト	173
1.3. おもちゃ	175
1.4. ガスメーター	177
1.5. 身の周りはコンピュータだらけ	178
2. 携帯電話	179
2.1. 携帯電話の現況	179
2.2. 携帯電話のしくみ	180
2.3. 携帯電話の歴史	181
2.4. 携帯電話は何処へ	184
3. 交通とコンピュータ	185
3.1. GPS とナビゲーションシステム	185
3.2. VICS	187
3.3. ETC	190
3.4. PTPS	191

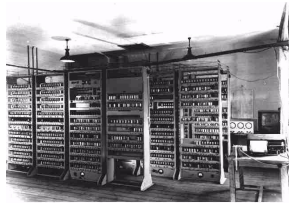
4.	電子商取引と電子マネー	193
4.1.	電子商取引	193
4.2.	ネットショッピングの決済の現状	194
4.3.	電子マネー	195
5.	ユビキタス・コンピューティング	197
5.1.	ユビキタス・コンピューティングとは	197
5.2.	IC タグ	198
5.3.	アドホックネットワーク	199
第 8 章	コンピュータセキュリティ	201
1.	システムダウン	202
1.1.	システムダウンの予防	202
1.2.	被害を最小に	203
2.	不正アクセス	205
3.	コンピュータウイルス	206
3.1.	コンピュータウイルスとは	206
3.2.	主なコンピュータウイルス	208
3.3.	コンピュータウイルス被害の状況	210
3.4.	コンピュータウイルス対策	212
4.	個人情報の保護	215
4.1.	パスワード	215
4.2.	暗号化	216
4.3.	主な暗号	219
付 録		226
1.	数の接頭語と単位	226
1.1.	数の接頭語	226
1.2.	主な単位	227
2.	エクセルのマクロと VBA	229
2.1.	マクロとは	229
2.2.	VBE	229
2.3.	マクロの実行	231
2.4.	マクロの保存と実行	232

第 1 章 コンピュータ

1. コンピュータとは

1.1. コンピュータは計算機

コンピュータ (computer) の日本語訳は電子計算機、すなわち「計算機」です。



EDSAC (1949年)



パーソナルコンピュータ

図 1.1 初期の電子計算機 (EDSAC) とパーソナルコンピュータ

コンピュータの歴史は 1946 年に始まるとされていますが、1949 年に作られた最初の本格的なノイマン型コンピュータ、^{エドサック}EDSAC *1 の正式名称も、“electronic delay storage automatic calculator” ですから、「遅延線メモリ式自動電子計算機」です。

コンピュータの性能はその後飛躍的に高まりましたが、基本的なしくみは EDSAC 以来ほとんど変わっていません。コンピュータの頭脳に当たる中央処理装置 (CPU: central processing unit) *2 はたくさんの命令を実行できますが、その大部分は算術演算や論理演算命令で、他にはデータを転送したり、プログラムの実行を制御するための命令があるばかりです。コンピュータの命令の大部分は計算のための命令だといえます。つまりコンピュータは、本質的に、計算しかできない機械です。

しかし、私たちはコンピュータを単に「計算機」として使っているわけではなく、レポートを書いたり、音楽を聞いたり、デジタルカメラで撮った写真を印刷したり、インターネットで情報を検索したり、テレビ番組を録画したり… もしています。

「計算しかできない機械」に、何故こんなことができるのでしょうか。

*1 「EDSAC」(119 ページ) 参照。ノイマン型コンピュータは、現在普通に使われている、プログラムをメモリに読み込んで実行するコンピュータ。121 ページ参照。

*2 「中央処理装置」(49 ページ) 参照。

1.2. 計算機からコンピュータへ

計算機をコンピュータに変えたのはシャノン (Claude E. Shannon) (1916~2001) *3 です。コンピュータは2進数しか扱えない機械ですが、文字を数字に置き換えることによって、文字データも扱うことができるようになります。

イチローに‘51’という背番号があるように、文字‘A’にも‘65’という背番号をつければ、文字‘A’は‘65’というコード化された数字によってコンピュータで処理できます。

静止画像も画面全体を無数の小さい画素 (Pixel) に分割して個々の画素の色を数値で表せば、無数の数値の集合として静止画像をコンピュータで処理することができます。動画は無数の静止画像の集合だし、音声も瞬間瞬間の音の強さを数値で表し、そうした一連の数値の集合を音声データとすれば、やはりコンピュータで取り扱うことができます。

こうしたことを最初に系統的に研究したのがシャノンでした。

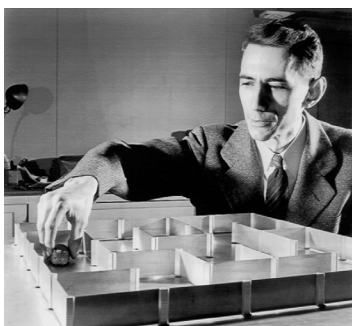


図 1.2 クロード・シャノン

1948年、論文「通信の数学的理論 (A Mathematical Theory of Communication)」などを発表し、情報や通信を数学的に扱う情報理論を創設したシャノンは、情報量の単位としてビット*4 (bit) を定義し、すべての情報は bit に置き換えられることを示しました。こうして、電子計算機を単に計算だけでなく、文書や画像、動画、音声など、あらゆるデータを処理する情報処理装置、いわゆる「コンピュータ」として利用する道を開きました。

1.3. コンピュータはスイッチでできている

コンピュータの主要部は、何億個もの「スイッチ」を組み合わせてできています。

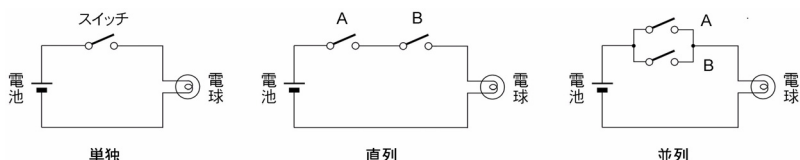


図 1.3 スwitchの性質

*3 「電気 (リレー) 式計算機」 (113 ページ) 参照。

*4 「情報量」 (10 ページ)、「ビット」 (227 ページ) 参照。

スイッチといえば、もちろん、電灯を点けたり消したりするときに使う、あのスイッチです。図 1.3 の左は普通の回路で、スイッチを ON-OFF すると電球が点いたり消えたりします。中央は 2 個のスイッチが直列につながれているので、両方のスイッチを ON にしないと電球が点きません。消すときはひとつでも OFF にすれば消えます。右は並列で、スイッチがひとつでも ON になれば電球が点きますが、両方のスイッチが OFF でないと電球は消えません。

スイッチが直列に接続されると、電球は「俺、スイッチがみんな ON でないと点かないよ」という回路になるし、並列だと「ひとつでも ON なら点くわ、私」となって、スイッチのつながり方によって回路にも個性が現れてきます。たくさんのスイッチを直列や並列につないで組み合わせればもっと複雑な、計算する回路などを作ることができ、それらを更に組み合わせれば、コンピュータを作ることできます。

ただし、普通のスイッチは一個一個手で操作しますが、スイッチが極端に多くなるとそういうわけにもいきません。ON-OFF が自動的に行えるように、スイッチにはトランジスタを使います。普通のスイッチとトランジスタでは姿形は異なりますが、働きは同じです。トランジスタは IC に大量に集積されます。

コンピュータは無数の「スイッチ」を組み合わせられて作られています。

1.4. コンピュータは左腦的

人類は有史以前から道具を用い、18 世紀後半の産業革命以降は機械を用いてきました。しかし、道具といい機械といい、それらはすべて人間の手足の働きを補助するものです。中でも機械は、人間の手足では不可能な、複雑で精密な動作と、大きな力で人間の手足の働きを補助します。しかし、それは所詮^{しょせん}手足の働きの補助であり、頭腦の働きを補助するものではありませんでした。

コンピュータの出現によって、人間は初めて頭腦の働きを補助するものを得ることができました。コンピュータは①高速処理、②大量記憶、③正確性という、3 つの点において人間より優れていて、人間の頭腦を補助することができます。

しかし、コンピュータは人間以上の能力は持っていない、あくまで人間のアシスタントです。機械がどのように人間以上の能力を持っていても、あくまで人間のアシスタントであるのと同様に、コンピュータもまた人間の命令に従って動いて人間を補助するものであって、人間を支配するものではありません。

人間の頭腦の右半分と左半分はその機能が異なっていて、左脳では計算・推論などの論理的な仕事が行われており、右脳では直感・連想・概念形成・パターン認識・類似性認識などの非論理的な仕事が行われるとされています。

これに従えば、現在のコンピュータは主として左腦的機能のものであって、右腦的な仕事に対してはまだまだほとんど無力です。

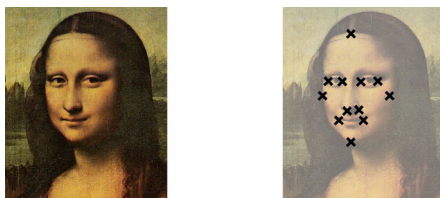


図 1.4 コンピュータ的「パターン認識」

近頃は顔認識によってマンションの住人かどうかを識別するセキュリティシステムや、指紋認証のできる携帯電話も珍しくありません。そういうところで使われているコンピュータは、パターン認識や類似性認識を行っているように思えます。しかし残念ながら、コンピュータはやはり計算機です。まずカメラで撮影した画像から、人の顔の特徴をもっともよく表す点を抽出し、それらの座標値が登録されているデータと一定の誤差の範囲内で一致するものを探すために、膨大な計算をしているのです。

右脳の働きこそ、あるいは人間にしかできない、最も人間的なものといえるかもしれません。

1.5. コンピュータは情報処理装置

コンピュータは情報処理 (information/data processing) を行うための装置、すなわち「情報処理装置」です。



図 1.5 情報処理装置

コンピュータは、電卓のような単なる計算機ではありません。もちろん数値の計算も行いますが、それは情報処理の中の1つの分野の仕事として行っているに過ぎません。「情報処理」とは、素材となる情報 (データ) を処理加工して、利用する者のニーズに応じた形に変換することをいいます。たとえば、素材となるデータから全体の平均値を求めるとか、それをグラフの形で表す、などはその代表的な例です。

したがって情報処理装置は、入力 (input)、処理 (processing)、出力 (output) の3つの要素で構成されています。

2. コンピュータの種類

私たちが普段目にするコンピュータは、たいていパーソナルコンピュータ (personal computer) ですが、この他にも超大型から超小型まで、さまざまなコンピュータがあります。

2.1. メインフレーム

かつては単に「コンピュータ」といえば、メインフレーム (mainframe) (大型汎用コンピュータ) のことでした。コンピュータが発明されて以来、コンピュータの技術はメインフレームを中心に発達してきたからです。

メインフレームは企業の基幹業務や大学での研究などに使われるため、中央処理装置や補助記憶装置は二重化され、電源も無停電電源装置 (UPS : uninterruptible power supply) によってバックアップされるなど、信頼性に対しても十分な配慮がされています。



図 1.6 メインフレームの例

出典：(社) 情報処理学会「コンピュータ博物館」(<http://museum.ipsj.or.jp/>)

図 1.6 は 1990 年に発表された ACOS システム 3800 (日本電気) です。利用者はメインフレームに接続された端末*5 (terminal) を通じてコンピュータを操作します。何台もの端末が接続されますが、処理はすべてメインフレームが行なう集中処理的なシステムです。

近年はメインフレームに代わって、小型のコンピュータをネットワークで接続した分散型のシステムが使われるようになりました。これをダウンサイジング (downsizing) といいます。

2.2. スーパーコンピュータ

気象解析や構造解析、流体力学シミュレーションなど、複雑で膨大な科学技術計算のために用いられるコンピュータです。

気象データを解析するのに一週間かかっていたのは天気予報に使えませんが、コンピュータが 100 倍速くて 2 時間ですむのなら、話は違ってきます。コンピュータの処理が速いと、情報の利用範囲も広げることができます。

*5 キーボード、CRT ディスプレイ、ネットワーク通信機能などからなる大型コンピュータの入出力機器。



図 1.7 スーパーコンピュータの例

写真提供：独立行政法人海洋開発研究機構 地球シミュレータセンター

図 1.7 はスーパーコンピュータの一例、地球シミュレータです。地球大気や海洋、地球内部の変動を定量的に評価・予測することなどを目的に作られたものですが、最近では分子設計や遺伝子解析など、材料科学や生命科学の分野にも幅広く使われています。

地球シミュレータは 50m × 65m もある体育館のような建物の中に、写真の計算ノード (processor node) が 640 台設置されている並列計算機です。ピーク性能は 40T FLOPS^{テラ フロップス}、1 秒間に 40 兆回の浮動小数点演算ができ、運用開始 (2002 年) 時点では世界最高速の性能を持っていましたが、2004 年に IBM の Blue Gene (70T FLOPS) に首位の座を譲りました。

2011 年には 10P FLOPS^{ペタ}、1 秒間に 1 京回の浮動小数点演算ができるスーパーコンピュータ「京」が開発され再び世界最速になりましたが、2012 年 6 月に IBM の Sequoia (16P FLOPS) に追い越されました。

2.3. ミニコンピュータ

初期のコンピュータはすべてメインフレームでしたが、半導体技術の進歩と共にコンピュータも IC 化されるようになると、大学の研究室などでも比較的気軽に使える小型で安価なコンピュータ、ミニコンピュータ (mini computer) が作られるようになりました。

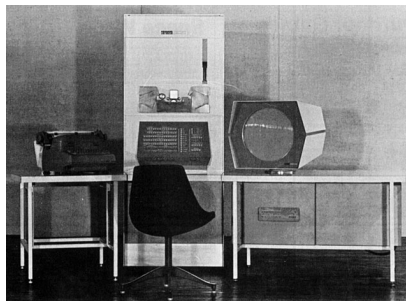


図 1.8 最初のミニコンピュータ、DEC PDP-1

当初は主として科学技術分野、工場の設備機器や通信の制御などに利用されましたが、次項

のワークステーションと共に、コンピュータネットワークの研究やダウンサイジングなどにも貢献しました。

2.4. ワークステーション

もともと、メインフレームのインテリジェント端末として作られたコンピュータです。特に高精細度ディスプレイやマルチウィンドウ、GUI 環境などの機能が整っているものをエンジニアリング・ワークステーション (engineering workstation) といい、主に CAD (computer aided design) による設計やグラフィックデザインなどの分野で利用されました。現在はパーソナルコンピュータの機能が向上したため、ワークステーションとの境界は曖昧になっています。OS には UNIX が採用されているものが多く、安定性やセキュリティ面で優れているため、インターネットのサーバ*6 (server) としても広く利用されています。

2.5. パーソナルコンピュータ

半導体技術の進歩によって、コンピュータの電子回路を 1 チップの IC に集積した MPU が開発されると、個人が所有できる程度の価格のコンピュータ、すなわちパーソナルコンピュータ (パソコン) が生まれました*7。パーソナルコンピュータの機能が次第に向上して普及し、また LAN に接続されて業務にも使われるようになった現在では、「コンピュータ」といえばパーソナルコンピュータを指すことが多くなりました。

2.6. マイクロコンピュータ

冷蔵庫、電子レンジ、電気炊飯器、湯沸器、オーブントースター、食器洗い機、掃除機、洗濯機、乾燥機、アイロン、扇風機、エアコン、テレビ、ビデオ、CD コンポやラジカセ、ファックス、ゲーム機、電卓、電子辞書、携帯電話、デジタルカメラ、ビデオカメラ、自動車…。

思いつくままに、身のまわりにある家電製品などのうち、内部にコンピュータが入っているようなものを並べてみました。これらの製品に組み込まれているのがマイクロコンピュータで、IC チップの上にコンピュータの機能がほとんど全部集積されています。前項では「コンピュータといえばパーソナルコンピュータ」と書きましたが、実はいちばん普及しているのはパーソナルコンピュータではなく、マイクロコンピュータです。これらのコンピュータは、冷蔵庫が冷えすぎるとコンプレッサを止めたり、洗濯が終わるとバルブを開いてすぎの水を入れるなど、それぞれの機器の制御のために使われています。姿や役割は違いますが、パーソナルコンピュータの中で使われているものと、本質的にはまったく同じです。

気がつけば私たちの身のまわりはコンピュータだらけ、いつの間にか、すっかりコンピュータに囲まれて生活しているのです*8。

*6 ネットワーク上で、クライアントに対して自身の機能や情報を提供するコンピュータ。

*7 「パーソナルコンピュータの誕生」(134 ページ) 参照。

*8 「家中どこにも、コンピュータ」(172 ページ) 参照。

3. コンピュータの構成

コンピュータは中央処理装置、主記憶装置、入力装置、出力装置、補助記憶装置の5つの装置からなっています。

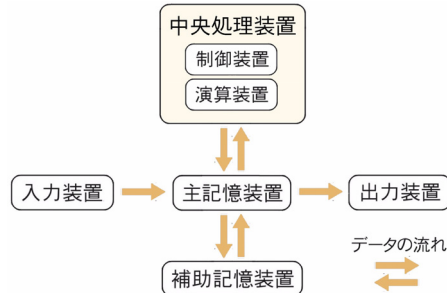


図 1.9 コンピュータの構成

中央処理装置 (CPU: central processing unit) はコンピュータの頭脳に相当する最も重要な装置で、制御装置と演算装置からなっています。演算装置は文字通り算術演算や論理演算などを行い、制御装置は命令を解釈して演算装置に伝えたり、コンピュータ内部の各装置の動作のタイミングを制御したりします。

コンピュータの性能は、ほぼ CPU で決まります。現在の CPU はワンチップの IC に集積されています。これを MPU (micro processing unit) といいます。近年、MPU は機能面でも速度面でも飛躍的に性能を高めており、小型のパーソナルコンピュータでも、かつてのスーパーコンピュータを凌ぐ性能を持つようになりました。

主記憶装置 (main memory) はデータやプログラムを記憶するメモリです。CPU は必要なデータやプログラムをメモリから受け取り、処理済みのデータはまたメモリに記憶されます。こうして、CPU とメモリは常にデータのやりとりをしながら、二人三脚で走るように処理を続けていきます。二人三脚のコツは二人の呼吸をピッタリ合わせることですが、そのへんはコンピュータも同じで、CPU とメモリのスピードが揃っていないと性能を十分に発揮できません。しかし現在のコンピュータは、通常 CPU は十分高速ですが、メモリ (DRAM) は遅いため、二人三脚のバランスがとれていません。コンピュータの性能を下げないよう、様々な工夫がされています。

補助記憶装置 (auxiliary storage) としてもっともよく使われているのはハードディスクです。これはソフトウェアをインストールしておいて、必要に応じて起動 (メモリに読み込んで実行) したり、作成した文書データなどをファイルとして保存しておくためのものです。主記憶装置のようなスピードより、むしろ十分な記憶容量が求められます。

これらの他に、コンピュータにデータ等を入力するための**入力装置** (input device : キーボード、マウスなど)、処理結果を人が見たり記録することができるようにするため、**出力装置** (output device : ディスプレイやプリンタなど) も必要です。

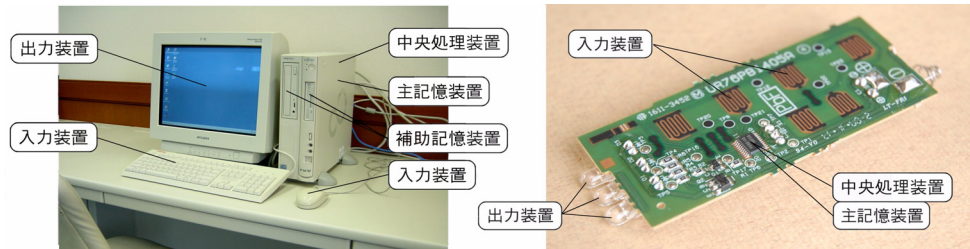


図 1.10 パーソナルコンピュータとマイクロコンピュータ（リモコン）

図 1.10 左はパーソナルコンピュータの構成の例です。コンピュータ本体内に中央処理装置、主記憶装置の他、ハードディスクドライブや CD-ROM、DVD-ROM ドライブなどの補助記憶装置も収められています。入力装置としてキーボードとマウス、出力装置には液晶ディスプレイ、写真には含まれていませんがプリンタなどがよく使われます。

右はマイクロコンピュータの例で、リモコンの内部にある基板です。中央やや左の IC がマイクロコンピュータで、ここに CPU やメモリが集積されています。

楕形の電極パターンはスイッチです。リモコンのボタンを押すと導電性ゴムが接触し、スイッチが ON になります。このコンピュータの入力装置です。

出力装置は左下の赤外線 LED。点滅を繰り返して 0, 1 の信号を送り出します。

リモコンには何種類ものプログラムをインストールしておく必要はないので、ハードディスクのような補助記憶装置はありませんが、これも立派なコンピュータです。

照明の明るさを変えたりテレビのチャンネルを切り替えるのにも、私たちはコンピュータのお世話になっています。

4. 情報

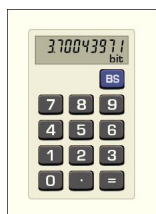
コインを投げると、やがて落ちてきて、表か裏のいずれかを上にして静止します。このとき、「表だったよ!」という知らせは情報です。トランプのカードの束から一枚を引いた時の「ハートのエースだよ」も情報ですが、このふたつの情報の価値（あるいは重み）は同じでしょうか。

コインの表裏と、カードの束からの一枚。同じでないとすれば、どこが違うのでしょうか。

4.1. 情報量

シャノン (Claude E. Shannon) ^{*9} はこの問題を数字で評価するために、「情報量 (information)」を定義しました。シャノンによると、ある事象（ここではコインの表裏や引かれたトランプのカード）についての情報量は、それが起こるとき確率 (probability) で決まります。

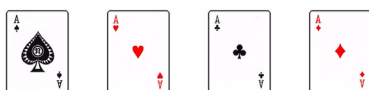
Web で学ぶ 【情報量】



情報量 (I) は、確率を P とすると $I = -\log_2 P$ で表されますが、 \log_2 の計算はかなり厄介です。

しかし Web ページ「情報量」には、確率 P (または確率の逆数 $\frac{1}{P}$) をキーインするだけで情報量が計算できる電卓があります。ここでは、情報量の計算にはこの「情報量電卓」を使うことを前提にして話を進めることにします。

コインを投げたときは、表が出る確率も裏が出る確率も $\frac{1}{2}$ ですから、情報量電卓に 0.5 と入力^{*10}して「=」をクリックすると、情報量は 1 bit であることが分かります^{*11}。コインを投げ、表か裏か、どちらが出たかという場合の情報量は 1 ビットです。



次に、エースばかりを集めた 4 枚のトランプのカードから、1 枚引くことを考えてみます。

カードは 4 枚なので、どのスーツ (suit) を引く確率も $\frac{1}{4}$ です。情報量電卓に 0.25 を入力すると、情報量は 2 ビットになります。4 種類のうちのいずれか、という情報を伝えるには 2 ビットあればいいわけです。2 枚のコインを投げたらどちらも表だったという場合も同様で、2 枚のコインがどちらも表を向く確率は $\frac{1}{4}$ ですから、情報量は 2 ビットです。

今度はエースからキングまで 13 種類のカードを集めて、この中から 1 枚を引きます。どの

^{*9} 「計算機からコンピュータへ」(2 ページ) 参照

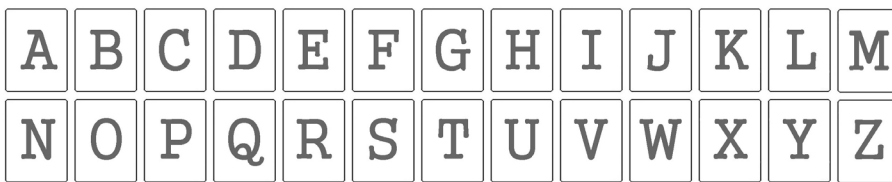
^{*10} 2 を入力も同じ結果が得られます。1 より大きい値が入力されると、確率の逆数と判断して計算します。

^{*11} 情報量は確率だけから決まる値ですから、本来単位はありませんが、 $-\log_2 P$ で計算したときはビット (bit) という単位を使います。bit は binary digit (2 進数の 1 桁) から作られた言葉です。



カードを引く確率も $\frac{1}{13}$ です。これを小数にするのは厄介ですが、情報量電卓はその逆数、13を入力しても大丈夫です。情報量は約 3.7 ビットであることが分かります。端数を切り上げて、4 ビットあれば、どんなカードを引いたかという情報を伝えることができます*12。

トランプのすべてのカード（52 枚）の場合、あるカードを引く確率は $\frac{1}{52}$ です。情報量は約 5.7 ビットになります。やはり端数を切り上げて、6 ビットあればこの情報を伝えることができます。



カードにトランプの絵柄ではなく、文字を書き込んでみました。図では省略していますが小文字のカードもあります。カードは全部で 52 枚、トランプと同じです。したがって、このカードを 1 枚引いて、どんな文字が現れたかという場合の情報量も当然約 5.7 ビットです。端数を切り上げて 6 ビットあれば、この情報を伝えることができます。

カードにはトランプの絵柄でも文字でも、何が描かれていてもいいのです。52 枚のうちのどれかという情報は、いずれも 6 ビットあれば伝えることができます。

ただし、6 ビットで伝えることができるのは、何枚目のカードかという「数字」だけです。ハートのエースであるとか、A という文字であるとか、カードに描かれている内容まで伝えることはできません。これらは、情報を伝える側と受ける側で、同じ順序で並べたトランプや文字のカードを共有しておいて、そうしたデータを参照してはじめて可能になることです。

文字の場合も、同じ順序で並べた文字のカードを共有していれば、6 ビットで 52 種類の英文字の情報を扱うことができます。しかし、英文にはアルファベット 52 文字の他に、数字や記号類も必要です。ASCII*13 でコード化されている文字・記号は全部で 94 種類あり、これを表すのに必要な情報量は約 6.6 ビットですから、本格的な英文の文字情報を扱うには 7 ビットの情報量が必要になります。

このことから、7 ビット、のちにはコンピュータのデータ長として扱いやすくて区切りのいい 8 ビットが、ひとつの文字を表す情報量の単位とされるようになりました。これを 1 バイト (byte) といいます。

*12 4 ビットあれば、16 枚までのカードについての情報を伝えることができます。

*13 american standard code for information interchange. コンピュータで使用されている最も代表的な文字コード。「ASCII」(22 ページ) 参照。

4.2. アナログとデジタル

コンピュータで取り扱われているデータは、すべてデジタル (digital) です。

数字や文字はもちろん、静止画、動画、音声などもコンピュータで処理されています。そうした静止画、動画、音声などのデータは本来アナログ (analog) 信号ですが、コンピュータで処理するに先立ってデジタル信号に変換されます。

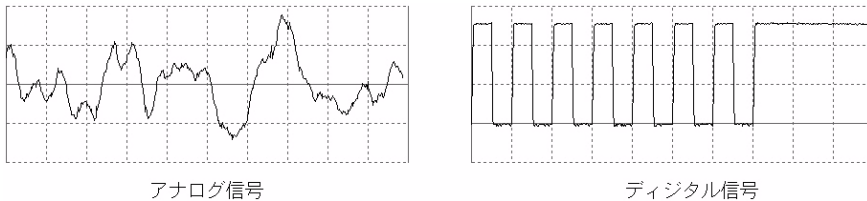


図 1.11 アナログ信号とデジタル信号の例

「アナログ」とは“連続して変化するもの”をいう語で、「デジタル」は“不連続に変化するもの”です。自然物は音であれ風景であれ時の流れであれ、すべて切れ目なく連続的に変化しています*14。このように連続的に変化するものを、自然物に類似したという意味でアナログ (analogue = 類似物) と呼びます。

一方、人工物の中には、しばしば不連続に変化し、その変化の数を指折り数えることができるものがあります。たとえば、階段や梯子は高さ^{はしご}が不連続に変化し、カレンダーやデジタル時計では数字が不連続に変化します。このように不連続に変化するものを、指折り数えることができるという意味でデジタル (digit = 指) と呼びます。

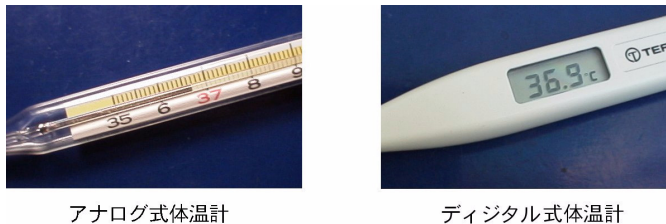


図 1.12 体温計

図 1.12 はアナログ式体温計とデジタル式体温計の例です。「アナログ」という用語は、温度や音や光など連続的に変化するもの、また、それらを連続的に変化する物理量 (水銀柱の長さなど) で表現すること、さらに、それらを用いて目的を達成するための装置 (体温計など) に対して使われます。一方、「デジタル」という用語は、温度や時間などを数字で表現すること、あるいは数字で表現するための装置 (体温計など) に対して使われます。

図 1.13 は写真をデジタル化した例です。A はふつうの写真ですが、B は白、明るい灰色、暗い灰色と黒の 4 階調、C は白黒 2 階調で表されているデジタル画像です。

*14 原子のレベルでは、電子のエネルギーなどは不連続に変化します。

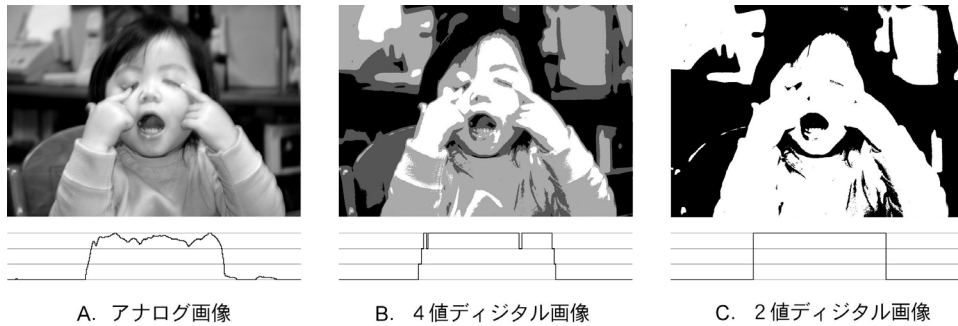


図 1.13 アナログ画像とデジタル画像

写真の下のグラフは天地中央部の明るさを表しています。顔の部分は明るくて背景が左右とも暗いので、明るさのグラフは帽子のような形になります。このように、写真の明暗の変化は本来連続的なもので、灰色には明るさが微妙に異なる無数の灰色がありますが、階調の数を有限化すれば写真もデジタル化されます。

デジタル化された写真は階調数が少ないと不自然ですが、256 階調（8 ビット）あればほぼアナログ写真と同じように見えますので、一般的なコンピュータでは 8 ビットでデジタル化して画像データを処理しています*15。

Web で学ぶ 【アナログとデジタル】



Web ページ「アナログとデジタル」には、図 1.13 のように、写真の明るさをグラフ化するアプレットがあります。

2 階調、4 階調、8 階調、16 階調、256 階調の写真について、明るさの変化をグラフにします。

アナログとデジタルの違いを目で理解できます。

*15 図 1.13 A は「アナログ画像」としてありますが、これもコンピュータで処理された写真ですから、やはり 256 階調のデジタル画像です。カラー写真の場合は光の三原色、赤・緑・青の明るさをそれぞれ 8 ビット、計 24 ビットで表します。したがってコンピュータのディスプレイでは $2^{24} \approx 1,678$ 万色で表示されています。「ディスプレイ」(68 ページ) 参照。

第2章 コンピュータの原理

1. 2進数

1.1. なぜ2進数なのか

コンピュータの内部では、なぜ2進数 (binary number) が使われているのでしょうか。

しかしその前に、私たちはなぜ10進法を使っているのか、まずこれを考えてみましょう。

それは、人間の手の指が5本だからだ、という説に異を唱える人はまずいないでしょう。ものを指折り数えるとき、1、2、3、4、5と、右手の指を全部折ってしまったら、次の6、7、8…は指を開いていきます。10まで数えたら左手の指をひとつ折る…、これで100まで数えることができます。両手の指の数、10を基数とする10進法が使われるようになるのはごく自然なことです。

もし人間の指が4本だったら、どうだっでしょう。4本指でも物は掴めるので、文明の発達には特に支障なさそうです。だとすると、4本指文明の社会ではものの数え方は必然的に8進法になるでしょう。私たちは当たり前のように10進法を使っていますが、それはたまたま、指が5本だったからだとはいえそうです。

しかし私たちは同時に12進数も使っています。鉛筆一箱には12本入っていますが、これを特に不思議には思いません。10進法でなくても、12本が1ダース、12ダースが1グロス、慣れれば何でもありません。

お医者さんで処方された薬はPTPシートにたいてい10錠入っていますが、中には14錠というのもあります。高血圧や糖尿病など慢性薬の薬は2週間単位で処方されることが多いので、薬も7の倍数で梱包されている方が都合がいいのです。

また、時計や角度の分や秒は60進法です。このように、私たちの身の回りの数字は必ずしも10進法ばかりではありませんが、しかしこれらも慣れてしまえば何でもありません。

ならば2進法はどうか。2進数には0と1しかないので、10進数 (decimal number) に慣れている私たちにの目は奇異に映ります。たとえば123,456,789という数字を2進数で表すと、

$$111010111011100110100010101$$

になります。0と1ばかりで、読みとるのも一苦勞です。

なぜコンピュータではこんなものを使うのでしょうか。

10進数で演算するコンピュータを作ることも不可能ではありません。現に、1946年に作られた世界で最初のコンピュータ、^{エニヤック}ENIAC*1 は10進演算方式でした。

しかし、10進数演算のコンピュータは必要以上に複雑になってしまいます。ENIACには18,800本もの真空管が使われていましたが、1949年に作られた2進数演算のコンピュータ、^{エドサック}EDSAC*2 は3,000本の真空管で作られています。

1830年代、バベジも機械式の自動計算機の開発を企てましたが、^{ことごと}悉く失敗しました。歯車の回転角度を0~9の数字に対応させた10進演算方式だったため、あまりにも複雑になってしまったのです。一方ツェーゼは1938年、同じ機械式でも2進数演算方式を採用し、カムを使って計算機を作ることに成功しています*3。

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	10	12	14	16	18
3	0	3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5	0	5	10	15	20	25	30	35	40	45
6	0	6	12	18	24	30	36	42	48	54
7	0	7	14	21	28	35	42	49	56	63
8	0	8	16	24	32	40	48	56	64	72
9	0	9	18	27	36	45	54	63	72	81

	0	1	2	3	4	5	6	7	8	9	
0	0	0	1	2	3	4	5	6	7	8	9
1	1	1	2	3	4	5	6	7	8	9	10
2	2	2	3	4	5	6	7	8	9	10	11
3	3	3	4	5	6	7	8	9	10	11	12
4	4	4	5	6	7	8	9	10	11	12	13
5	5	5	6	7	8	9	10	11	12	13	14
6	6	6	7	8	9	10	11	12	13	14	15
7	7	7	8	9	10	11	12	13	14	15	16
8	8	8	9	10	11	12	13	14	15	16	17
9	9	9	10	11	12	13	14	15	16	17	18

図 2.1 10進数の掛け算と足し算

上図は10進数の掛け算と足し算をするための、左はいわゆる「九九表」です*4。右はあまり見かけませんが、あえて言えば、足し算のための「九九表」です。掛け算のように丸暗記しなくてもいいので、足し算の「九九表」は必要ありませんが、いずれにしても10進数の計算をするにはこのような知識が必要です。小学生の頃、何度も何度も復唱して「九九」を覚えました。

	0	1
0	0	0
1	0	1

	0	1
0	0	0
1	1	10

図 2.2 2進数の掛け算と足し算

さて、2進数ならどうでしょうか。2進数には0と1しかないので、掛け算にしる足し算にしる、その組み合わせは4種類しかありません。「九九表」は「一一表」になります。

計算機を作る場合も、10進数演算なら私たちが「九九表」を使って計算するのと同じ複雑な演算装置が必要ですが、2進数なら「一一表」の演算ができればOKです。

*1 「ENIAC」(117ページ) 参照。

*2 「EDSAC」(119ページ) 参照。

*3 「バベジの階差機関」(108ページ)、「解析機関」(109ページ)、「Z1」(109ページ) 参照。

*4 ゼロの項も加えてあるので、一般的な「九九表」とは少し違います。

2進数の数字は0と1だけですから、これを電圧や電流の“あり・なし”で表すことができます。0から9まで10種類の数字を電気信号に置き換えるのは厄介ですが、“あり・なし”だけなら簡単です。

コンピュータに2進数を使わない手はありません。

1.2. 2進数

私たちが普段使っているのは10進数です。0~9の、10種類の数字を使って数値を表します。「423円」と書けば、百円硬貨4枚、10円硬貨2枚と1円硬貨3枚の、「四百二十三円」のことです。10進数の各桁にそれぞれ 10^2 、 10^1 、 10^0 の重みがあるからで、1桁左に書かれた数字は、右にある数字よりも10倍の重みを持っています。この、重み付けの基本になる数を基数 (radix) といいます。10進数の基数は10です。

$$423 = 4 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

↑基数
 ↑基数
 ↑基数

2進数は0と1の2種類の数字だけで数値を表します。2進数の各桁にもそれぞれ重みがあり、1桁左に書かれた数字は、1桁右の数字よりも2倍の重みを持っています。2進数の基数は2です。2進数で「1101」と書けば、

$$1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

↑基数
 ↑基数
 ↑基数
 ↑基数

ですから、 $(1101)_2 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 8 + 4 + 1 = (13)_{10}$ です*5。

0から15までの数字を2進数で表すと、表2.1のようになります。

10進数	2進数	10進数	2進数	10進数	2進数	10進数	2進数
0	0	4	100	8	1000	12	1100
1	1	5	101	9	1001	13	1101
2	10	6	110	10	1010	14	1110
3	11	7	111	11	1011	15	1111

表 2.1 10進数と2進数

表2.1で、2進数1桁では0と1の2種類の数字を、2進数2桁では0、1、10、11の4種類の数字を表すことができることが分かります。同様にして2進数3桁では0~111の8種類、4桁では0~1111の16種類、一般的には2進数n桁で 2^n 種類の数字を表すことができます。2進数の桁数と表せる数字の数の関係は、表2.2にまとめてあります。

*5 以後、2進数の「10」は $(10)_2$ 、10進数の「2」は $(2)_{10}$ と表すことにします。

2進数の桁数	表せる数字の数	2進数の桁数	表せる数字の数
1	2	7	128
2	4	8	256
3	8	9	512
4	16	10	1,024
5	32	11	2,048
6	64	12	4,096

表 2.2 2進数の桁数と表せる数字の数

Web で学ぶ 【2進数と16進数、10進数】



Web ページ「2進数と16進数、10進数」では、数字が2進数、16進数、10進数で表示されます。2進数の数字をマウスでクリックして数値を変えたり、自動的に数値を大きくしたり小さくしたりすることもできます。2進数とはどういうものか、直感的に理解できます。

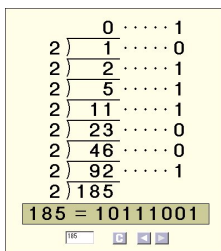
1.3. 10進数から2進数への変換

10進数を2進数に変換するには、順次2で割ってその余りを並べます。

$$\begin{array}{r}
 0 \dots 1 \\
 2 \overline{) 1 \dots 1} \\
 \underline{2} \quad 3 \dots 0 \\
 2 \overline{) 6 \dots 1} \\
 \underline{2} \quad 13
 \end{array}
 \quad \downarrow$$

たとえば13を2で割ると商が6で余りが1ですが、更に2で割っていくと、余りは順に0、1、1となります。この余りを逆に並べると2進数になります。すなわち、 $(13)_{10} = (1101)_2$ です。

Web で学ぶ 【10進数から2進数への変換】



Web ページ「10進数から2進数への変換」では、255までの任意の10進数を2進数に変換することができます。

マウスでクリックするたびに2で割って余りを出すという手順が繰り返して表示され、10進数への変換方法がすぐに理解できます。

1.4. 2進数から10進数への変換

2進数の各桁にはそれぞれ2倍の重みがありますから、10進数に変換するには各桁の係数を掛けたものを加えます。

たとえば2進数の1101は

$$\begin{aligned} (1101)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = (13)_{10} \end{aligned}$$

となります。

Web で学ぶ 【2進数から10進数への変換】

10111001 (2進数)

$= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 128 + 0 + 32 + 16 + 8 + 0 + 0 + 1$

$= 185$ (10進数)

BIN DEC HEX

Web ページ「2進数から10進数への変換」では、8桁までの任意の2進数を10進数に変換できます。いろんな数字で試すと、すぐに理解できます。

1.5. 符号付き2進数

負の数を取り扱う場合は符号付き2進数を使用します。

「符号付き」といっても、数字の前に + や - の記号をつけるわけではありません。符号付き2進数では最上位桁 (MSB: most significant bit) を符号ビット (sign bit) といい、負の数を表します。4桁の2進数では表 2.3 (下線部は符号ビット) のように、符号ビットが0であれば符号なし2進数と同じですが、これが1になると全く異なる値になります。

符号なし2進数		符号付き2進数		符号なし2進数		符号付き2進数	
2進数	10進数	2進数	10進数	2進数	10進数	2進数	10進数
0000	0	<u>0</u> 000	0	1000	8	<u>1</u> 000	-8
0001	1	<u>0</u> 001	1	1001	9	<u>1</u> 001	-7
0010	2	<u>0</u> 010	2	1010	10	<u>1</u> 010	-6
0011	3	<u>0</u> 011	3	1011	11	<u>1</u> 011	-5
0100	4	<u>0</u> 100	4	1100	12	<u>1</u> 100	-4
0101	5	<u>0</u> 101	5	1101	13	<u>1</u> 101	-3
0110	6	<u>0</u> 110	6	1110	14	<u>1</u> 110	-2
0111	7	<u>0</u> 111	7	1111	15	<u>1</u> 111	-1

表 2.3 符号なし2進数と符号付き2進数

4桁の2進数の場合、たとえば1101は符号なし2進数であれば

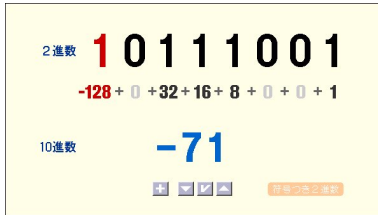
$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = (13)_{10}$$

ですが、符号付き2進数なら

$$(1101)_2 = \underline{-1} \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = \underline{-8} + 4 + 1 = (-3)_{10}$$

となります。符号付き2進数と符号なし2進数には見かけの違いはありませんが、どちらで扱うかによって値は全く異なってしまふことがあるので、注意しなければなりません*6。

Web で学ぶ【符号つき2進数】



Web ページ「符号つき2進数」では、8桁の2進数が符号なし2進数、または符号付き2進数として10進数に変換して表示されます。2進数の数値を自由に変えて、符号なし2進数と符号付き2進数を比べることができます。

1.6. 2進数の加算

2進数は0と1だけですから、加算の組み合わせは図2.2(15ページ)の通り、次の4種類しかありません。

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= (1)0 \quad (1) \text{ は桁上げ} \end{aligned}$$

最後の行は少し異様に見えるかもしれませんが、2進数には0と1しか数字がありませんから、10進数の「1 + 1 = 2」は、2進数では「1 + 1 = 10」と桁上がりします。

一例として110110と010111を加えると、次のようになります。

$$\begin{array}{r} 110110 \\ 110110 \\ + 010111 \\ \hline 1001101 \end{array}$$

上の例で、いちばん上の行に小さい数字で示しているのは桁上げ(carry)です。

2桁目も3桁目も同じ1 + 1なのに、和はそれぞれ0と1になっています。これは2桁目の計算の結果生じた桁上げを3桁目で加えたため、10進数の加算では7 + 3 = 10ですが、75 + 38 = 113になるのと同じことです。

*6 人は区別できませんが、符号付き2進数として扱うか、符号なし2進数として扱うかはプログラムに明示されているので、コンピュータが戸惑うことはありません。

Web で学ぶ 【2進数の加算】

C	A	B	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Web ページ「2進数の加算」では、任意の8桁の2進数の加算ができます。1桁ずつ順に加算でき、また戻すことも自由なので、十分理解できるまで繰り返し試すことができます。

1.7. 2進数の減算

2進数の減算も、次の4種類しかありません。

- $0 - 0 = 0$
- (1) $0 - 1 = 1$ (1)は桁借り
- $1 - 0 = 1$
- $1 - 1 = 0$

2行目は上位から桁借り (borrow) することを表しています。これは10進数の $(1) 0 - 7 = 3$ などと同じと考えればいいでしょう。たとえば $1011 - 0110$ を計算すると、次のようになります。

$$\begin{array}{r}
 0 \\
 \cancel{1} 0 1 1 \\
 - 0 1 1 0 \\
 \hline
 0 1 0 1
 \end{array}$$

Web で学ぶ 【2進数の減算】

A	B	D
0	0	0
0	1	1 (borrow)
1	0	1
1	1	0

Web ページ「2進数の減算」では、加算と同様、任意の8桁の2進数の減算ができます。桁借りが発生すると、減算は少し難しくなります。特に何桁にもわたる桁借りは厄介ですが、何度か繰り返して試してみれば、すぐに理解できます。

1.8. 16進数

この章のはじめ（14ページ）で例にあげた通り、123,456,789 という数字を2進数で表すと111010111011100110100010101です。2進数は桁数が多くなるとたいへん読みづらくなります。そこで、これを扱いやすくするために16進数が使われています。

2進数	10進数	16進数	2進数	10進数	16進数
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

表 2.4 2進数、10進数、16進数

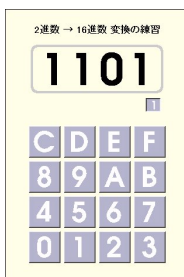
表 2.4 は4桁の2進数と10進数、16進数の関係を表したものです。10進数では、9より大きい数字は1桁で表すことができませんが、10～15をA～Fと書くことにすると、4桁の2進数0000～1111をすべて1桁で表すことができます。これが16進数（hexadecimal number）で、0からFまで16種類の数字があり、15までは1桁で表せますが、16になると桁がひとつ繰り上がって $(10)_{16}$ になります。基数は16です。16進数では、A～Fは10～15を表す数字です。

16進数を10進数に変換するには、次のようにします。

$$\begin{aligned}
 (4D2)_{16} &= 4 \times 16^2 + 13 \times 16^1 + 2 \times 16^0 \\
 &= 4 \times 256 + 13 \times 16 + 2 \times 1 \\
 &= 1024 + 208 + 2 = (1234)_{10}
 \end{aligned}$$

2進数を16進数に変換するには、0000～1111を機械的に0～Fに置き換えます。たとえば“1011”なら“B”という具合です。“101 0010 0110 1110”なら“526E”です。

Webで学ぶ【16進数】



Web ページ「16進数」に、16進数変換の練習用アプレットがあります。

スタートボタンをクリックすると、4桁の2進数がランダムに表示されます。対応する16進キーをクリックすると、正否がブザーで知らされます。10問で終了し、経過時間が表示されます。ゲームのような感覚で、16進への変換を習得できます。

2. キャラクターコード

コンピュータが直接扱えるのは2進数だけですが、文字を数字に置き換えれば、コンピュータで文字データを扱うことができます*7。

文字ひとつひとつに番号をつければいいのですが、番号のつけ方は無数にありますから、各自が勝手に決めたのでは文書データの互換性が失われます。そこで、コンピュータの利用者が共通に使用できるように、規格化された文字コードのセットをキャラクターコード (character code) といいます。

キャラクターコードには文字や数字、記号だけでなく、コンピュータのディスプレイでカーソルを行頭に戻す‘CR’ (carriage return)、改行する‘LF’ (line feed)、一文字戻す‘BS’ (back space) など、ハードウェアを制御するための符号 (機能キャラクター) も含まれています。

2.1. ASCII

ASCII (american standard code for information interchange) はANSI *8 が1963年に定めたキャラクターコードです。アルファベット、数字、記号など94個の図形文字と、34個の機能キャラクターの計128文字を7ビットで表しています。図2.3はASCIIの表ですが、たとえば‘A’という文字は4列目の1行目にありますから、‘A’のコードは“41”です。同様にして‘z’のコードは“7A”です。

		上位3ビット								
		000	001	010	011	100	101	110	111	
		0	1	2	3	4	5	6	7	
下位4ビット	0000	0	NUL	DLE (SP)	0	@	P	.	p	
	0001	1	SOH	DC1	!	1	A	Q	a	q
	0010	2	STX	DC2	"	2	B	R	b	r
	0011	3	ETX	DC3	#	3	C	S	c	s
	0100	4	EOT	DC4	\$	4	D	T	d	t
	0101	5	ENQ	NAK	%	5	E	U	e	u
	0110	6	ACK	SYN	&	6	F	V	f	v
	0111	7	BEL	ETB	'	7	G	W	g	w
	1000	8	BS	CAN	(8	H	X	h	x
	1001	9	HT	EM)	9	I	Y	i	y
	1010	A	LF	SUB	*	:	J	Z	j	z
	1011	B	VT	ESC	+	:	K	[K	{
	1100	C	FF	FS	,	<	L	\	l	
	1101	D	CR	GS	-	=	M]	m	}
	1110	E	SO	RS	.	>	N	^	n	~
	1111	F	SI	US	/	?	O	_	o	DEL

機能キャラクター

図 2.3 ASCII

*7 もちろんこの場合も、コンピュータが扱っているのは、あくまでも数字です。

*8 American National Standards Institute: 米国規格協会。

ASCII は 1973 年に ISO 646 として国際規格化されましたが、もともと米国生まれの規格 (american standard code …) ですから、英語にはよくてもドイツ語の ä、ö、ü、ß、フランス語の á、â、ç など、他の国の言語で使われている文字がありません。そこで “5B” ~ “5E”、 “7B” ~ “7E” など、図 2.3 で太い線で囲っている 12 文字は、各国が独自に定めてよいことになっています。日本では 5C の ‘\’ (バックスラッシュ) を ‘¥’ (円記号) に、同じく 7E の ‘˘’ (チルダ) を ‘¯’ (オーバーストリーク) に変えて使用しています*9。

2.2. JIS コード (JIS X 0201)

		上位 4 ビット																
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
		O	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
7 ビット	0000	O	NUL	DLE (SP)	O	@	P	'	p			。 (機能)	一	タ	ミ			
	0001	1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
	0010	2	STX	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
	0011	3	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
	0100	4	EOT	DC4	\$	4	D	T	d	t			、	エ	ト	ヤ		
	0101	5	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
	0110	6	ACK	SYN	&	6	F	V	f	v			ラ	カ	ニ	ヨ		
	0111	7	BEL	ETB	'	7	G	W	g	w	未定義	未定義	ア	キ	ヌ	ラ	未定義	未定義
	1000	8	BS	CAN	(8	H	X	h	x			イ	ク	ネ	リ		
	1001	9	HT	EM)	9	I	Y	i	y			ウ	ケ	ノ	ル		
	1010	A	LF	SUB	*	:	J	Z	j	z			エ	コ	ハ	レ		
	1011	B	VT	ESC	+	:	K	[K	{			オ	サ	ヒ	ロ		
	1100	C	FF	FS	.	<	L	¥	l				ヤ	シ	フ	ワ		
	1101	D	CR	GS	-	=	M]	m	}			ユ	ス	ヘ	ン		
	1110	E	SO	RS	.	>	N	^	n	¯			ヨ	セ	ホ	°		
	1111	F	SI	US	/	?	O	_	o	DEL			ツ	ソ	マ	°		

■ 機能キャラクタ

図 2.4 JIS コード

図 2.4 は JIS コード*10 です。ASCII に準拠していますが、8 ビットに拡張してカナ文字と記号を追加し、上述の通り ‘˘’ と ‘\’ を ‘¯’ と ‘¥’ にしてあります。

次項の JIS 漢字コード (JIS X 0208) のカナはディスプレイで正方形に表示されるのに対して、JIS コード (JIS X 0201) のカナは半分の幅で表示されることから、「半角カナ」と呼ばれることもあります。コンピュータの処理能力が低く、漢字が扱えなかった時代には重宝しましたが、ISO 646 (ASCII) などの国際規格との整合性がないため、コンピュータ間のデータ交換の機会が増えた今日では、文字化けを起こす原因のひとつになっています。

*9 ウェブページの URL にはチルダがよく使われるので、日本のパーソナルコンピュータでもこれを入力、表示できるようにしています。JIS では、オーバーストリークの字形は ‘¯’ を本則とするが、送信者と受信者の間で明示的な合意がある場合はチルダと同じ字形を用いてもよいとされています。ただしこの場合でも、文字の名称は「オーバーストリーク」でなければならないそうです。

*10 JIS X 0201: 「7 ビット及び 8 ビットの情報交換用符号化文字集合」。

図 2.5 は JIS 漢字コード表の一部分です。たとえば「亜」という文字は「30' 行の '21' 列にありますから、文字コードは“30 21”、同様にして、「漢」は“34 41”です。

さて、「JIS 漢字コード」という文書データを JIS コード表と JIS 漢字コード表を使って文字コードに置き換えると、“4A 49 53 34 41 3B 7A 25 33 21 3C 25 49”となります。次にこれを元の文字に戻すと、同じコード表で“4A 49 53”は「JIS」です。ところが、次の“34 41 3B 7A …”は、そのまま JIS コード表を使うと「4A;z …」になってしまいます。ここから JIS 漢字コード表に切り替えればちゃんと「漢字コード」になりますが、文字コードのデータだけでコード表を切り替えるという判断はできません。

そこで、漢字データの前には“1B 24 42”、漢字データの後は“1B 28 42”というコード列を挿入することによって、JIS コードと JIS 漢字コードを区別することになっています。

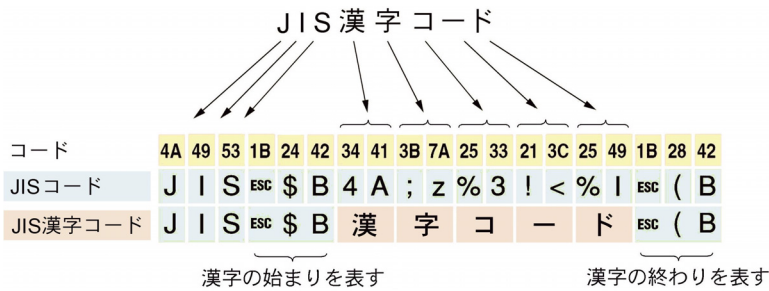
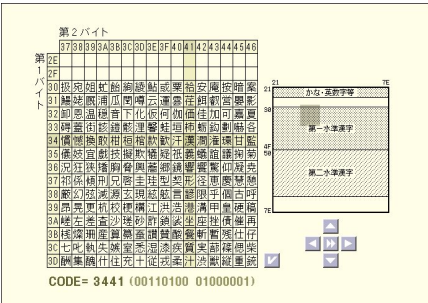


図 2.6 JIS 漢字コードの例

Web で学ぶ 【JIS 漢字コード】



第2バイト
0708090A0B0C0D0E0F10111213141516

第1バイト
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46

CODE= 3441 (00110100 01000001)

Web ページ「JIS 漢字コード」では、JIS 漢字コード表のすべての文字のコードを表示することができます。

拡大表示窓内でマウスカーソルを移動させるとその位置の文字コードが表示され、クリックすると文字コードの表示が固定されます。

2.4. シフト JIS 漢字コード

しかし、漢字を使用する度に“1B 24 42”などのデータ列を挿入しなければならないのも面倒ですから、その必要がないシフト JIS 漢字コードが作られました。

シフト JIS 漢字コードは、第1バイトが JIS コードでは使用されていない 81~9F、または E0~EF になるように、一定の規則で文字の位置を移してあります。

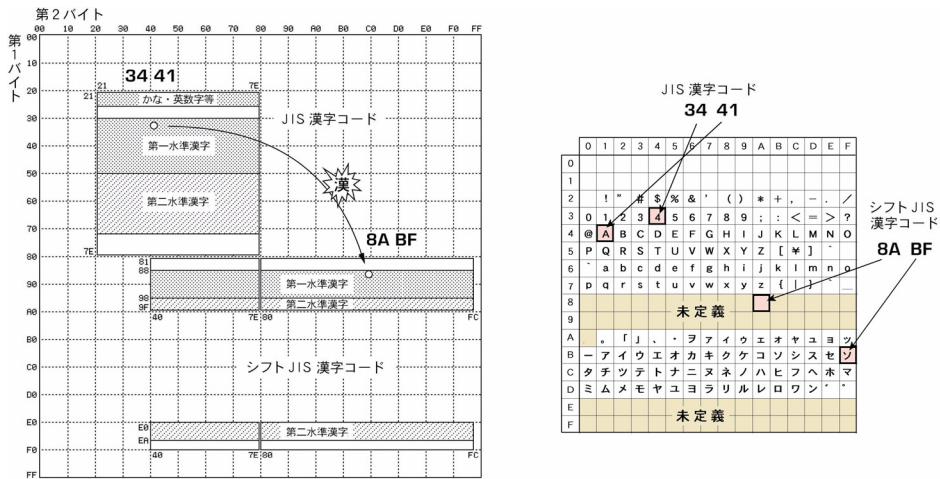


図 2.7 シフト JIS 漢字コード

「漢」という文字のコードはシフト JIS 漢字コードでは“8A BF”に変わります。JIS コード表 (図 2.7 右)^{*12} では“8A”は未定義ですから、本来文書データ中に現れるはずがありません。したがって、JIS コードで未定義のコードがあればシフト JIS 漢字コードの第 1 バイトだと判断できるので、“1B 28 42”といったデータ列は不要になります。

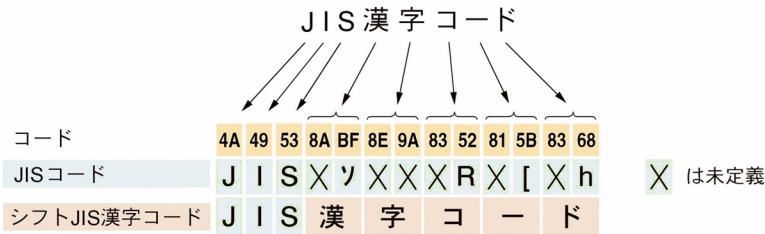


図 2.8 シフト JIS 漢字コードの例

シフト JIS 漢字コードは主にパーソナルコンピュータで使われていますが、ワークステーション^{*13}では同じように区切り文字列が不要な EUC^{*14} (extended UNIX code) が使われています。いずれもコンピュータ内部で自動的に処理されますから、コンピュータのユーザーがこれらを意識する必要はほとんどありません。しかし、インターネットのように様々のコンピュータがネットワークで接続されている環境では、幾つもの文字コード体系が混在することによって、文字が正しく表示されないというトラブルが発生する原因にもなっています。

また、世界には中国語をはじめアラビア語、インド語、ハンゲル語などの多文字言語がたくさんあり、それぞれ独自に文字コードを制定して使用しています。しかしこの場合も、世界の

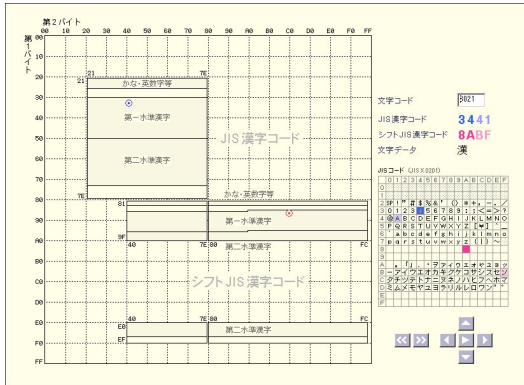
*12 漢字コード表と対応させるため、図 2.4 の図を回転してあります。

*13 主としてエンジニアやデザイナーなどの業務や、ウェブサーバとして使われている小型コンピュータ。

*14 JIS 漢字コードの各バイトの MSB を 1 にしたもの。“3441”は“B4C1”になります。

国々のコンピュータがネットワークで接続されるようになると、やはり文字が正しく表示できないということが起こります。このため米国の情報関連企業によって、世界中の主な言語の文字をすべて表現する国際標準文字コード (unicode) が提唱され、1993年に ISO *15 で標準化されて普及しています。

Web で学ぶ 【シフト JIS 漢字コード】



Web ページ「シフト JIS 漢字コード」では、すべての文字について、図 2.7 のように、JIS 漢字コードとシフト JIS 漢字コードの関係が表示されます。

シフト JIS 漢字コードの第一バイトが、常に JIS コード表の未定義の位置にあることがすぐに理解できるでしょう。

文字コードは、コンピュータが「文字をどのような数字に置き換えて処理するか」という約束事にすぎないので、コンピュータの利用者は文字コードを意識する必要はありません。

しかし、コンピュータが広く普及し、文書はコンピュータで作成するのが当たり前になり、作家はコンピュータで執筆し、原稿は電子メールや CD-R でやり取りされ、書籍の電子化も急速に進んできました。レポートや論文も電子媒体で提出する機会が増えています。

文字コードはかつての活字に代わって、文化の根幹を支えていかなければなりません。現在は日本語だけをとってみても、コンピュータですべての文字を扱い、手軽にデータを交換することができません。JIS 漢字コード (JIS X 0208) には 7 千文字程度しかないので、古典文学の参考書も、漢和辞典の原稿も作ることができません。unicode も収容文字数が足りないなど、問題が少なくありません。

世界中のすべての国のすべての文字が、コンピュータで一様に扱えるようになるのはいつの日でしょうか。

*15 国際標準化機構: international organization for standardization

3. 論理回路

さて、いよいよコンピュータの計算のしくみについて考えることにします。

計算には演算回路が必要ですが、演算回路は論理回路を組み合わせで作られます。

論理回路にはさまざまな種類のものがありますが、中でも特に重要な AND、OR、NOT 回路の三つを「基本論理回路」といいます。

3.1. 基本論理回路

3.1.1. AND 回路

AND 回路（論理積回路）は、複数の入力信号のいずれもが 1 のときのみ出力が 1 になります。図 2.9 は入力信号が二つの場合の AND 回路について、左からその回路記号と論理式、真理値表^{*16} (truth table)、スイッチによる近似的な回路を表しています。スイッチ A、B が直列に接続されていると、どちらも ON でないと電球はつきませんが、AND 回路の出力も同様に、入力 A、B がいずれもが 1 のときだけ出力 X が 1 になります。論理式は $X = A \cdot B$ という、積の形になります。

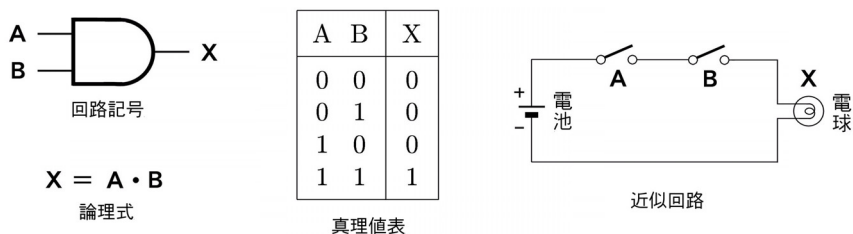


図 2.9 AND 回路

このように、実際にスイッチを直列に接続して使うことがあるのでしょうか。…あまり多くはありませんが図 2.10 はその一例、裁断機です。



図 2.10 スイッチの直列接続例

何百枚もの紙を一度に裁断することができる印刷屋さんの必需品ですが、ひとつ間違えば大怪我をしかねない危険な機械でもあります。そのため、直列に接続された 2 個のスイッチを使っています。

^{*16} 論理回路の出力を、入力条件のすべての組み合わせについて示した表。

ボディの左右にある押しボタンスイッチ（○印）がそれで、機械を操作する人は両手でふたつの押しボタンスイッチを押さないとカッターの刃は動きません。2個のスイッチがいずれもONのときだけカッターが作動するので、誤って指を切断するといった事故を未然に防いでいます。

3.1.2. OR回路

OR回路（論理和回路）は複数の入力信号の少なくともひとつが1のときに出力が1になります。図2.11は入力信号が二つの場合のOR回路の回路記号と論理式、真理値表、近似的な回路です。この回路ではスイッチA、Bのどちらか、あるいは両方がONのときに電球が点きます。OR回路の出力も同様に、入力A、Bの少なくともひとつが1のときに出力Xが1になります。論理式は $X = A + B$ です。

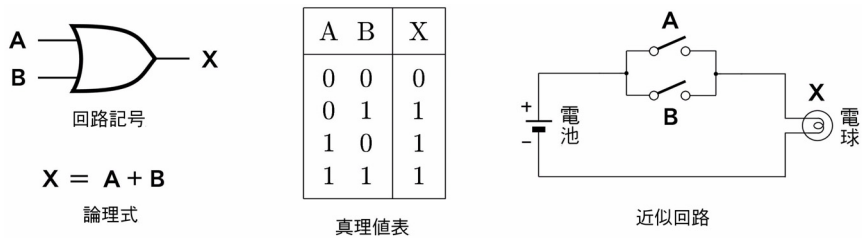


図 2.11 OR回路

OR回路のようなスイッチの接続の例としては、バスの押しボタンスイッチがあります。



図 2.12 スイッチの並列接続例

バス停が近づくと、乗客は近くにあるスイッチを押します。この信号は運転席に伝えられて、次のバス停は止まらなくてはならないことを知らせます。いずれかのスイッチが押されるとバスの運転席でブザーが鳴るのと同様に、OR回路も入力の少なくともひとつが1になると出力が1になります。

Web で学ぶ 【AND 回路、OR 回路】

Web ページ「AND 回路」や「OR 回路」では、図の入力信号やスイッチの状態をマウスで変えることができます。AND 回路、OR 回路の働きがすぐに理解できます。

3.1.3. NOT 回路

NOT 回路（論理否定回路）は入力された信号を反転して出力します。0 が入力されれば 1 を、1 が入力されれば 0 を出力する回路で、インバータ (inverter) ともいいます。

NOT 回路の記号は三角形のうしろに小さい円がついた形で表しますが、否定の働きを表しているのは後ろの小さい円の方です。論理式は $X = \bar{A}$ と書き、 $\bar{\quad}$ は信号の反転を表しています。

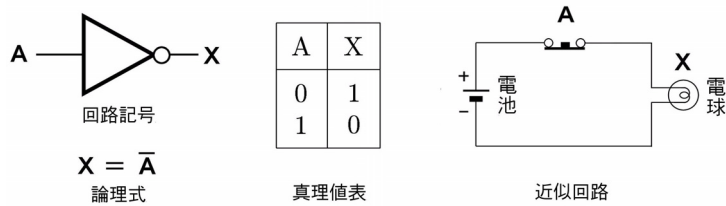


図 2.13 NOT 回路

図 2.13 の近似回路で使っているのは押しボタンスイッチですが、普通の押しボタンスイッチと違って、通常（押されていないとき）は ON になっていて電球は点灯しています。スイッチを押すと OFF になって電球が消えます。押されていないときを 0、押したときを 1 と考えれば NOT 回路と同じ動作と考えることができます。

Web で学ぶ 【その他の論理回路】

基本論理回路以外の論理回路については、Web ページ「NAND 回路」や「NOR 回路」、「EXOR 回路」で、それぞれの回路の働きを分かりやすく説明しています。

3.2. 論理回路の応用

これまで学んできた基本論理回路を組み合わせると、もっと具体的な働きをさせることができるようになります。以下、いくつかの例を紹介します。

3.2.1. 車のうっかりミス防止

以前は、自動車の中にキーを残したままドアをロックしたり、ライトを消し忘れていたりすることがよくありました。近頃の自動車はキーを付けたまま、あるいはライトを点けたまま運転席のドアを開けるとブザーが知らせてくれます。このような機能は実際にはマイクロコンピュータにプログラムされて組み込まれていますが、同じような警報装置を論理回路で作ることができます。

まず、キーの抜き差し、ライトの ON-OFF、ドアの開閉などを 0 と 1 で表します。たとえばエンジンがかかっているときは 1、止めると 0 という風に、全ての条件を 0 と 1 に置き換えると次のような表ができます*17。

キー	ライト	エンジン	ドア	ブザー
0: 抜いている	0: 消灯	0: 停止	0: 閉	0: 鳴らない
1: 挿入	1: 点灯	1: 回転	1: 開	1: 鳴る

この場合、キーかライトのどちらか、あるいは両方が 1 のときに、エンジンが 0 でドアが 1 であればブザーが鳴ればいのですから、論理回路は次のようになります。

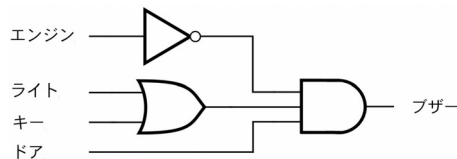
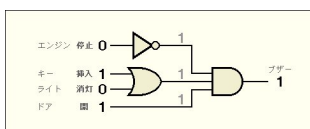


図 2.14 車のうっかりミス防止回路

エンジンを止めると NOT 回路の出力が 1 になります。キーを抜き忘れているか、ライトが点いたままであれば OR 回路の出力も 1 になります。この状態でドアを開けると AND 回路の入力信号はすべて 1 になりますから、出力が 1 になってブザーが鳴ります。

Web で学ぶ 【論理回路の応用】



Web ページ「車のうっかりミス防止」では、図 2.14 の回路がマウスで動き、キーを閉じこめたり、ライトを消し忘れて車を降りそうになるとブザーが鳴ります。

*17 ここでの 0、1 は自由に決めてかまいません。条件に応じて NOT 回路を入れたり外したりすればいいのです。

3.2.2. 半加算器

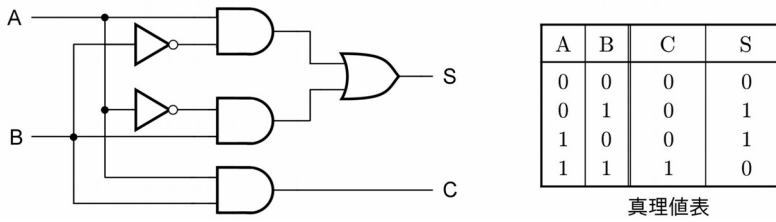
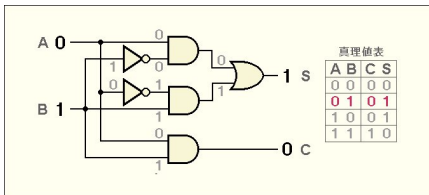


図 2.15 半加算器

図 2.15 は半加算器 (Half Adder) です。一桁の 2 進数、A と B を加算して和は S に、加算の結果生じるキャリーが C に出力されます。

半加算器の真理値表をよく見ると、出力 S と C は 2 進数の加算の規則 (19 ページ) と全く同じであることが分かります。したがってこのような半加算器を使えば、電子回路に 2 進数の加算をさせることができます。半加算器はコンピュータの演算装置を構成する、最も基礎的な論理回路です。

Web で学ぶ 【半加算器】



Web ページ「半加算器」では、内部のすべての信号が 0, 1 で表示されています。入力信号 A, B が変わると、出力 S, C がどのように変わるかがよく分かります。

3.2.3. リレーによる半加算器

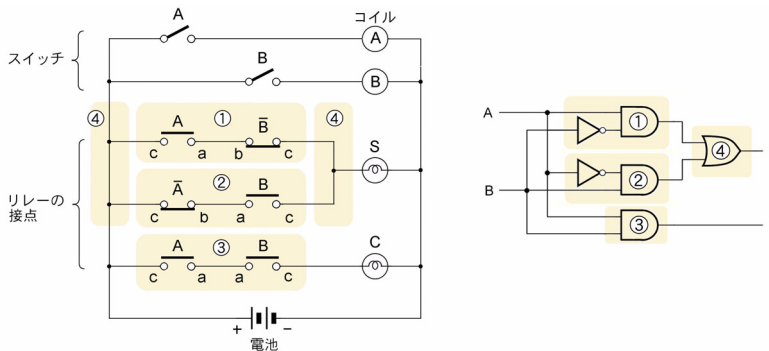


図 2.16 リレーによる半加算器

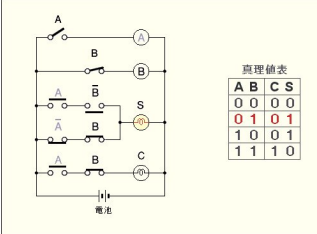
スイッチを組み合わせれば論理回路と同じ働きをしますから、半加算器もスイッチで作るこ

とができます。図 2.16 上部の A、B はリレー*18で、スイッチ A、B を ON にすると、コイル A、B に電流が流れて電磁石になり、下部にある接点 A、B 各 3 個が同時に動きます。

スイッチ A を ON にすると接点 A も ON になりますが、接点 \bar{A} は OFF になります。スイッチ B の場合も、接点 B について同じことが起こります。したがって、スイッチ A だけ、あるいはスイッチ B だけを ON にするとランプ S が点灯し、スイッチ A、B をどちらも ON にするとランプ C が点灯します。スイッチ A、B がどちらも OFF のときはもちろんどのランプも点かないので、この回路は半加算器と全く同じ動きをします。

図 2.16 で①～④の着色部は、右の論理回路による半加算器の①～④に対応します。

Web で学ぶ【リレーによる半加算器】



Web ページ「リレーによる半加算器」では、スイッチ A、B を ON/OFF すると、それに応じてリレーの接点が動きます。

スイッチで作られている論理回路ですから、半加算器の動作が直感的に理解できます。

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

論理回路は同じ働きをするものをいくつかの方法で作ることができます。それらの中から、状況に応じて、シンプルなもの、演算速度の速いもの、安価なものなどを選びます。半加算器にも様々なバリエーションがあり、図 2.17 はその一例です。

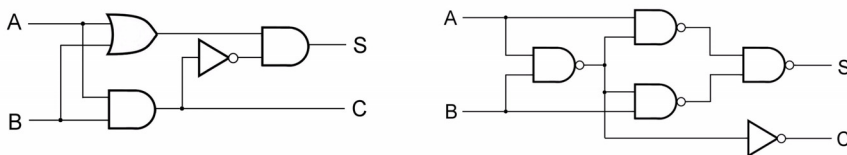


図 2.17 半加算器のいろいろ

3.2.4. フリップ・フロップ

フリップ・フロップ (flip-flop) は 1 ビットのデータを記憶する回路で、ラッチ*19 (latch) ともいいます。

図 2.18 で使っている論理回路は^{ナンド}NAND回路といい、AND 回路に NOT 回路を加えたのと同じ働きをします。すなわち、二つの入力のいずれれもが 1 のときだけ出力が 0 になります。

*18 電磁石とスイッチを組み合わせたもので、コイルに電流を流すと電磁石に引きつけられてスイッチが働きます。「電気 (リレー) 式計算機」(113 ページ) 参照。

*19 latch には「掛け金」という意味があります

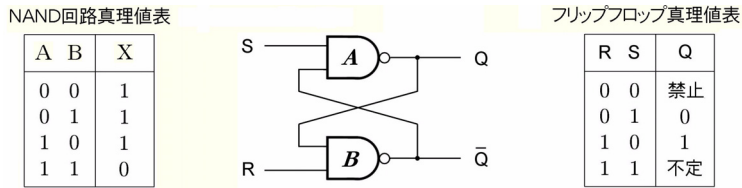


図 2.18 フリップ・フロップ

R (reset) と S (set) の入力は普段はどちらも 1 しておきます。図 2.18 にこの回路の真理値表がありますが、どちらも 1 の場合出力 Q は不定で、Q は 0 にも 1 にもなることができ、その状態を永久に保持します。

入力 S を 0 にすると (フリップフロップ真理値表 3 行目に相当)、Q は 1 になります。NAND 回路は二つの入力のいずれもが 1 のときだけが出力 0 ですから、S が 0 になると A の出力 Q は 1 になります。Q は B の入力に接続されていますから、Q が 1 になると B の出力 \bar{Q} は 0 になります。そしてこの後 S を 1 に戻しても、NAND 回路の片方の入力信号は 0 なので、Q は 1 のままです。

つぎに入力 R を 0 にすると、B の出力 \bar{Q} が 1 になり、それが A の一方の入力につながっていて S とともに入力がすべて 1 ですから、出力 Q は 0 に変わります (真理値表 2 行目)。同じくこの後 R を 1 に戻しても、Q は 0 のままです。

フリップフロップは、S または R が一瞬でも 0 になれば出力 Q が 1 または 0 になって、その状態をいつまでも保持する「記憶回路」です。

このように、論理回路を使えば記憶回路も作ることができます。

さて、論理回路で演算回路や記憶回路を作ることができました。演算と記憶は、コンピュータの最も基本的な機能です。

AND 回路と OR 回路、そして NOT 回路。一つ一つは単純ですが、これらがたくさん集まれば、コンピュータのような凄いものを作ることができます。

Web で学ぶ 【フリップ・フロップ】

Web ページ「フリップ・フロップ」では、R, S が 0 になったとき、それがどのように影響していくかが、少し時間をおきながら表示されます。少し分かりにくいフリップ・フロップの動きですが、容易に理解できます。

4. 半導体

論理回路の中で、実際にスイッチとして働いているのは半導体素子 (semiconductor device) です。ここでは半導体や半導体素子のしくみや性質について調べていくことにします。

4.1. 半導体とは

私たちの身の回りの金属、銅や鉄、金、アルミニウムなどには電流が流れます。金属ではありませんが炭素 (たとえば鉛筆の芯) などにも電流が流れます。これらはいずれも導体 (conductor) です。一方、ガラスやプラスチック、陶器などには電流が流れません。これらは絶縁体 (insulator) です。

このように電流の流れやすい導体と流さない絶縁体がありますが、その他に少しだけ電流を流す「半導体」もあります。電流の流れにくさは抵抗率*20 (resistivity) で表します。導体、たとえば銅の抵抗率は約 $1.7 \times 10^{-6} \Omega \text{ cm}$ で、絶縁体、たとえばガラスの抵抗率は約 $10^{12} \Omega \text{ cm}$ です。それに対して、純粋なシリコン (珪素: silicon) は約 $10^3 \Omega \text{ cm}$ という、導体と絶縁体の中間の抵抗率を持っているため、半導体といえます。半導体には高温になると抵抗率が下がる性質もあります。さまざまな電子機器に使われている IC (integrated circuit) やトランジスタ (transistor)、ダイオード (diode) など、半導体素子の材料はほとんどがシリコンです。

4.2. N型半導体とP型半導体

ほとんどの半導体素子は高純度のシリコンで作られていますが、これを真性半導体 (intrinsic semiconductor) といいます。半導体は導体と絶縁体の中間の抵抗率を持っていますが、実際には電流はごく僅かしか流れません。しかし、これに微量の燐や硼素などを不純物として加えると、添加量に応じて抵抗率が下がって電流が流れやすくなり、「半導体」が「導体」に近づいていきます。



図 2.19 硼素、シリコン、燐の原子

シリコンは 4 価の元素で、燐は 5 価、硼素は 3 価の元素*21 です。電子は原子核の周りの定められた軌道上を回っていますが、最も外側の軌道の電子を価電子 (または原子価電子) といい、物質の電気的・化学的な性質と密接な関係があります。シリコンは全部で 14 個ある電子のうち、4 個が価電子です。また燐は 5 個、硼素は 3 個の価電子を持っています。

*20 長さ $l \text{ cm}$ 、断面積 $a \text{ cm}^2$ の物質の電気抵抗は $R = \rho l/a$ (Ω) で表され、この ρ を抵抗率といえます。抵抗率の単位は $\Omega \text{ cm}$ (オームセンチメートル) です。

*21 炭素、ゲルマニウムなども 4 価の元素の仲間です。5 価の元素は燐の他に砒素、アンチモンなど。3 価の元素は硼素の他にガリウム、インジウムなど。

シリコンは互いに隣り合うシリコン原子と電子を共有して結晶を作ります。シリコンには価電子が4個しかありませんが、共有しあっているとりの原子のもの（図2.20の破線で結んだ電子）を含めると、どの原子も8個の電子を持っているように見えます。この状態の結晶は非常に安定で、炭素が同じような形で結晶になったものが天然の鉱物の中で最も堅いダイヤモンドです。電子は原子の結合に寄与しているため容易に動くことができません。電子が容易に動けない——ということは電流が流れにくいということです。「半導体」という名前でも、純粋なシリコンの結晶には電流はほとんど流れないのです。

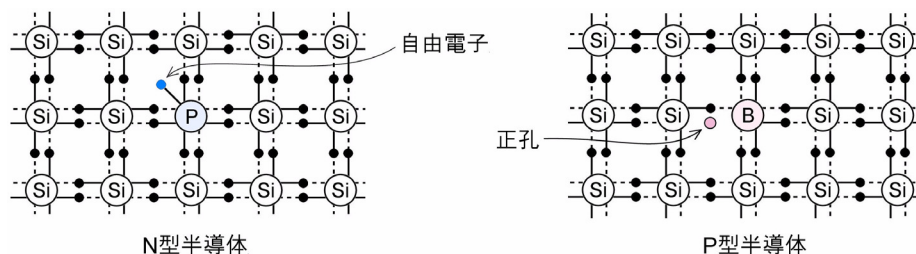


図 2.20 N 型半導体と P 型半導体

しかしシリコンに燐 (P) や硼素 (B) などを不純物として加えると、図 2.20 のような状態になります。

燐は5個の価電子を持っていますから、4個を共有結合のために使っても電子が1個余ります。この余った電子は束縛が強くないので、結晶に電圧がかかるとプラス極の方に向かって動いていきます。電子が動く——ということは、電流が流れるということです。マイナス (negative) の電気を持った電子が動くことによって電流が流れる半導体を N 型半導体 (N-type semiconductor) といいます。

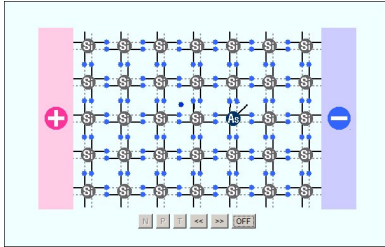
3個の価電子を持っている硼素が不純物として加えられると、今度は逆に電子のない空席ができます。この状態の結晶に電圧をかけると、空席の近くにある電子がプラス極の方向に移動して空席を埋めようとします。このとき、その電子がももいた場所は新たな空席になります。こうして新しくできた空席をまた別の電子が埋める…ということを順に繰り返すことによって、結果的に空席がマイナス極の方に向かって動いていきます。

小学校の学芸会や音楽会などで、前列の席が空くと後ろにいた人が前に移動しますが、そのあとにできた空席にさらに後ろの人が座る、という光景に似ています。

電子はマイナスの電気をもっていますからプラスの電極に引かれますが、空席はマイナスの電極の方向に動いていくので、あたかも空席自身がプラスの電気をもっているかのように見えます。それでこの空席、電子のない孔を「プラス (positive) の電気を持っているかのようにふるまう孔」という意味で正孔 (hole) といいます。正孔が動くことによって電流が流れる半導体を P 型半導体 (P-type semiconductor) といいます。また、N 型半導体や P 型半導体で、電流を流す役割を担っている電子や正孔を、キャリア (carrier) といいます。

N 型、P 型半導体は、電子や正孔によって電流が流れやすくなった半導体です。これらを組み合わせると、ダイオードやトランジスタといった「半導体素子」を作ることができます。

Web で学ぶ 【半導体】



Web ページ「半導体」では、N 型半導体の電子や P 型半導体の正孔が、アニメのように動きます。

電子や正孔が動けば、電流が流れます。

N 型半導体、P 型半導体に電流が流れるしくみが、目で見て理解できます。

4.3. ダイオード

ダイオード (diode) は P 型半導体と N 型半導体を接合*²²した半導体素子で、電流を一方に流す整流作用を持っています。電流がある方向に流れるときには ON、逆の方向に流れようとするときには OFF になるスイッチ、と考えることもできます。

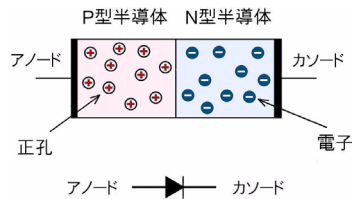


図 2.21 ダイオード

P 型半導体側の電極をアノード (anode)、N 型半導体側の電極をカソード (cathode) といいます。回路記号は三角形と線を組み合わせたもので、三角形は電流が流れる方向を表しています。

P 型半導体では正孔が、N 型半導体では電子が電流を流す役割を担っています。ダイオードに電圧をかけると、マイナスの電気を持っている電子はプラス極に、プラスの電気を持っているかのようにふるまう正孔はマイナス極に向かって動きます。

カソードに電池のプラス極、アノードにマイナス極をつなぐと、電子はカソード、正孔はアノードの方に向かって動きますが、電子や正孔が外部に流れ出そうとしても、新たにキャリアが供給されませんから、電子や正孔は電極部に引き寄せられた状態で止まってしまいます。したがってこの方向に電池をつないでも電流は流れません (図 2.22 左)。

*²² 接合 (junction) とは、半導体結晶のある面を境にして片方が P 型、他方が N 型になっているような状態をいいます。

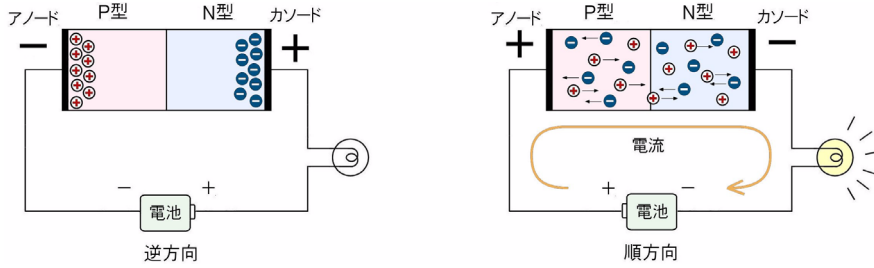
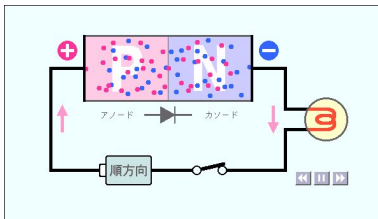


図 2.22 ダイオードの逆方向接続と順方向接続

逆に、カソードに電池のマイナス極、アノードにプラス極をつなぐと、N型半導体の電子はアノードに、P型半導体内の正孔はカソードに向かって動きます。電子が接合部に到着してP型領域に入ると、電子は新たに外部から次々に供給されますから、P型領域を経て外部に流れ出すことができます。正孔についても同様で、この場合は電子や正孔が動き続けて、電流が流れます(図 2.22 右)。

電池を接続する方向によって電流が流れるか流れないかが決まり、電流が流れる方向の電池の接続を順方向接続、流れない方向の接続を逆方向接続といいます。

Web で学ぶ【ダイオード】



Web ページ「ダイオード」でも、電子や正孔が動きます。

実際に電子や正孔がこのページの図のように動くわけではないのですが、ダイオードのしくみが直感的に理解できます。

4.4. トランジスタ

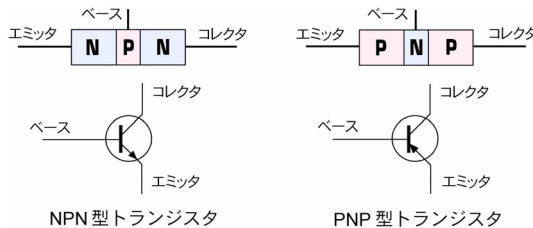


図 2.23 トランジスタ

接合型トランジスタは、P型半導体とN型半導体をNPN、またはPNPのサンドイッチ状に接合した半導体素子で、エミッタ (emitter)、ベース (base)、コレクタ (collector) という

3つの電極があり、増幅作用やスイッチング作用を持っています。

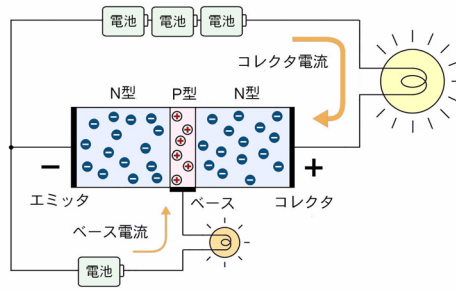


図 2.24 トランジスタの増幅作用

トランジスタのコレクタ-エミッタ間およびベース-エミッタ間に図 2.24 のように電池をつないで、ベースからエミッタに流す電流をわずかに変化させると、コレクタ-エミッタ間を流れる電流が大きく変化します。これがトランジスタの増幅作用です。また、ベースの電流を ON-OFF させるとコレクタ-エミッタ間の電流も ON-OFF するので、トランジスタをスイッチとして使うこともできます。

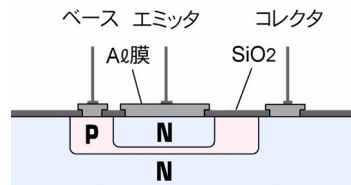


図 2.25 プレーナ型トランジスタ (NPN)

接合型トランジスタの理論は 1949 年にショックレー (Shockley : 1910~1989) によって作られました。ベースの層の厚さは数 μm 程度でなければなりません。当時はそんなに薄いベース層を挟んだ接合トランジスタを実際に作ることはできませんでした。これまで成長型、合金型、メサ型など、さまざまなタイプのトランジスタが考案されましたが、現在は図 2.25 のような構造のプレーナ型トランジスタ (planar transistor) が作られています。

プレーナ型トランジスタは大きな半導体の円板 (ウェハ: wafer) に多数のトランジスタを形成し、完成後に切り離します。品質の揃ったトランジスタを大量に作ることができますが、プレーナ型トランジスタは集積回路 (IC: integrated circuit) へと発展する可能性を持っているのも大きな特徴です。切り離せば単なるトランジスタですが、ウェハ上にはトランジスタが並んでいますから、ここにダイオードや抵抗、コンデンサなど他の電子部品を作り込んで結線すれば、半導体ウェハ上に電子回路を作りこむことができます。これが集積回路です。

Web で学ぶ 【トランジスタ】



Web ページ「トランジスタ」でも、電子や正孔が動いて表示されます。
トランジスタがなぜスイッチとして使うことができるのか、ダイオードのしくみと合わせて考えれば、これも容易に理解できます。

4.5. MOS トランジスタ

接合型のトランジスタはベースに流す電流によってコレクタの電流を制御しますが、電流が流れると熱が発生します。

論理回路に使われるトランジスタの場合、ベースに流す電流はごくわずかなので、トランジスタ 1 個あたりの発熱は微々たるものです。しかし、コンピュータの機能が高くなるにつれて使用するトランジスタの数が増え、小さい IC チップに大量のトランジスタを集積するようになると、発熱の問題が無視できなくなります。

ひとつの IC チップに何万個、何千万個という数のトランジスタが集積されると、個々のトランジスタの発熱はわずかであっても、チップ全体で発生する熱は膨大になります。大規模な集積回路 (LSI: large scale integration) を作るには、トランジスタの発熱、言い換えればトランジスタに流れる電流を極力抑えなくてはなりません。

MOS (金属酸化膜半導体: metal oxide semiconductor) トランジスタは、電流を流さなくても動作するトランジスタです。

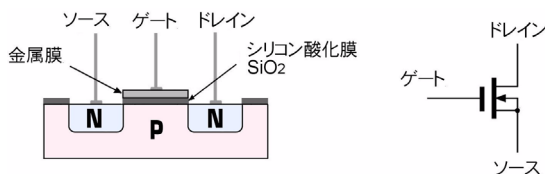


図 2.26 N チャネル MOS トランジスタ

図 2.26 は N チャネル MOS トランジスタです。ソース (source)、ゲート (gate)、ドレイン (drain) という三つの電極があります。ゲートの部分は上から順に金属膜 (Metal)、シリコン酸化膜 (Oxide)、半導体 (Semiconductor) の三層構造になっているので、MOS といいます。シリコン酸化膜 (SiO_2) は絶縁体ですからゲートに電圧をかけても電流は流れませんが、電圧がかかると強い電界ができ、半導体中の電子がゲートの近くに集められて、チャネル (channel) と呼ばれる電流の流れやすい領域ができます。これによってソースからドレインに

電流が流れるようになります。ゲートの電圧を取り去ると電流は流れません。ゲートの電界によってドレインの電流をコントロールできるので、電界効果型トランジスタ（FET: field effect transistor）ともいいます。

トランジスタにNPN型とPNP型があるように、MOSトランジスタにもNチャネルMOS（NMOS）トランジスタとPチャネルMOS（PMOS）トランジスタがあります。そして、NMOSトランジスタとPMOSトランジスタを組み合わせたものを^{シーモス}CMOS（complementary MOS）といい、これを図2.27のように接続するとインバータ（NOT回路）ができます。

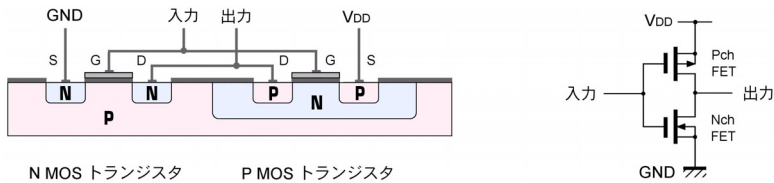


図 2.27 CMOS インバータ

トランジスタはスイッチとして使われているので、スイッチの記号で書くと図2.28のようになります。入力を0にするとPチャネルFETはON、NチャネルFETはOFFになるので出力は1に、逆に入力を1にするとPチャネルFETはOFF、NチャネルFETはONになるので出力は0、まぎれもなくNOT回路です。

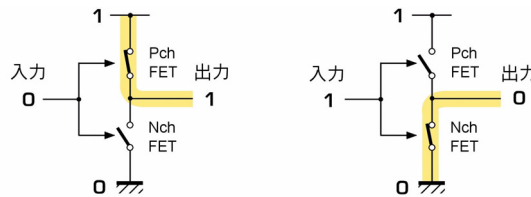


図 2.28 CMOS インバータの等価回路

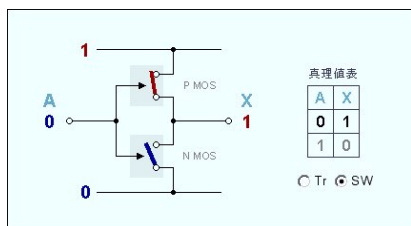
インバータには二つのトランジスタが使われていますが、いつもどちらかのトランジスタがOFFになっているので、電流は流れません。また、ゲートには電圧がかかるだけで電流は流れませんから、CMOSで構成した回路は、原理的にどこにも電流が流れません*23。

電流が流れなければ、熱も発生しません。大量のトランジスタを集積するICには、消費電力や発熱の面からもCMOSが適しています。電卓がボタン電池ひとつで長期間使用できたり、小さい太陽電池で動くのも、CMOSの「低消費電力」という特長を生かしているからです。

コンピュータのCPUに使われるマイクロプロセッサも、近年はほとんどCMOSプロセスで作られています。

*23 実際にはスイッチング時（1から0、0から1に状態が変わるとき）に瞬間的に電流が流れます。このため高速に動作させればさせるほど消費電力（発熱）が大きくなります。

Web で学ぶ【CMOS 論理回路】



Web ページ「CMOS 論理回路」では、MOS トランジスタとこれを使った論理回路について説明されています。

MOS トランジスタはスイッチの形で表示することができるので、論理回路のしくみが容易に理解できます。

4.6. 集積回路

集積回路 (IC: integrated circuit) は半導体チップの上にトランジスタ、ダイオード、抵抗、コンデンサーなどの素子を組み込んで種々の電子回路を構成したものです。

図 2.27 の CMOS インバータは、P チャネル FET と N チャネル FET のゲートとゲート、ドレインとドレインを接続していますが、電極間を電線ではなく、金属の薄い膜を生成して接続すれば、半導体チップ上に種々の電子回路を作ることができます。これが集積回路 (IC) です。

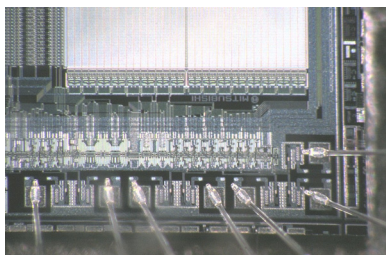


図 2.29 集積回路

IC の製造コストは主としてチップの大きさで決まります。チップに集積される素子の数とは無関係なので、素子を小さくしてたくさん作り込めば作り込むほど製造コストが下がります。初期の IC はチップ上の配線に使われるパターン幅が $10\mu\text{m}$ 程度でしたが、最近では $0.03\mu\text{m}$ 以下のものもあります。パターンの幅が約 $\frac{1}{300}$ なので、面積ではざっと $\frac{1}{10000}$ になっていることになります。つまり、かつてトランジスタを 1 個作っていた面積のチップに、現在は 10 万個のトランジスタを集積できるわけです。

またウェハの直径が大きければ大きいほど一度にたくさんのチップができるので、これもコストが下がる要因になります。

5. コンピュータのしくみ

演算回路や記憶回路は論理回路で作れることが分かりましたので、いよいよコンピュータのしくみについて考えていくことにします。しかし本物のコンピュータは非常に複雑で、しくみを理解しようという目的には相応しくありません。そこで、ここではコンピュータのしくみを学ぶための、究極の「単純コンピュータ」を使うことにします。もちろんこれは実在しない、仮想のコンピュータです。

しかしその前にまず、計算について考えてみましょう。

5.1. 手動式電子計算器

たとえば、「 $177 + 248$ を計算しなさい」と言われたら、どうしますか？

暗算が得意な人ならすぐ仕事にかかれますが、そうじゃない大部分の人はメモ用紙とペンがほしいですね。電卓がほしいという人もいるかも知れませんが、それは反則。

問題をメモ用紙に書き写して、準備が整えばひと桁ずつ計算していきます。「図 2.1 10 進数の掛け算と足し算」(15 ページ)の知識を使って、まず $7 + 8 = 15$ だから和は5、桁上げの1もメモして…という具合に、足し算の知識と筆記用具があれば OK です。

ならば機械にも、演算回路とメモ用紙代わりのメモリがあれば、計算ができそうです。

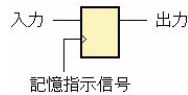


図 2.30 レジスタ

上図はレジスタという記憶装置で、177 とか 248 といった数字を一つ書き留める、メモ用紙のようなものです。左の入力端子にデータを与えておいて記憶指示信号を送ると、そのデータが記憶され、右の出力端子に現れます*24。

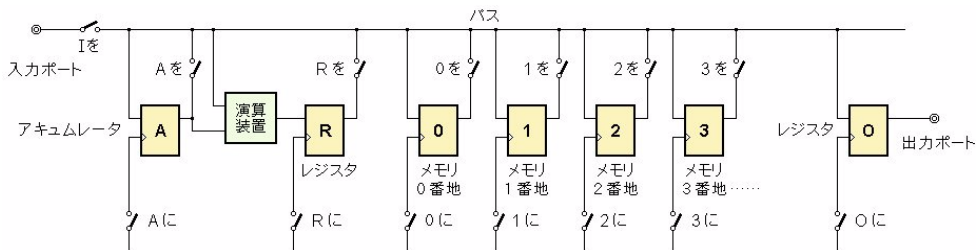


図 2.31 手動式電子計算器

図 2.31 は演算装置とレジスタからなる手動の計算機です。電子そろばんのようなものですが、プログラムがあって自動的に動くわけではないので、これはコンピュータではありません。

*24 もちろん、2 進数です。 $(248)_{10} = (11111000)_2$ を記憶するには、8 ビット以上のレジスタが必要です。

第2章 コンピュータの原理

黄色の長方形 A, R, 0, 1, 2, 3, O はレジスタですが、役割が違うのでアキュムレータやレジスタ、メモリなど、いろんな名前がついています。特にアキュムレータ（左端）はいちばん使い易いメモ用紙、コンピュータの中で最も頻繁に使われるレジスタです。

計算自体は図 2.31 の左半分（アキュムレータ、演算回路、レジスタ）だけでできますが、もう少し高等な計算、たとえば英・数・理・国 4 科目の合計点を計算する場合などは、あらかじめ各科目の点数をメモリに記憶させておいて一気に集計したいので、4 個のメモリをつけてあります。メモリは多い方がいい*25 のですが、コンピュータのしくみを理解するにはこれで十分です。メモリはレジスタがたくさん集まったもので、いわばメモ用紙の厚い束です。それぞれを識別できないといけないので番号をつけます。この番号をアドレス（番地：address）といいます。

レジスタ類やメモリの出力は、スイッチを介して図のいちばん上の線、バスに接続されています。これらのスイッチはすべて OFF で、バスには普段レジスタ類の出力信号は何も繋がっていませんが、データをバスに送り出すときだけスイッチを ON にしてバスに繋がります。ここではこのスイッチを「～のデータをバスに接続する」という意味で「～を」のスイッチと呼ぶことにします。

一方、レジスタ類の入力は常にバスに繋がっています。記憶指示信号のスイッチを ON にすれば、データが書き込まれ、記憶されます。このスイッチは「～に記憶させる」という意味で「～に」のスイッチと呼ぶことにします。

入力ポートと出力ポートは外部に開かれた扉で、この出入り口からコンピュータにデータやプログラムを入力したり、演算の結果を出力したりします。たとえば 177 という数値を入力するには、入力ポートに 10110001 *26 というデータをセットして「I を」のスイッチを ON にします。次にアキュムレータの「A に」のスイッチを ON にすれば、10110001 という値がアキュムレータに記憶されます。データが記憶されたら、「I を」、「A に」のスイッチはすぐ OFF に戻します。

アキュムレータの 10110001 というデータをメモリの 0 番地に書き込むには、「A を」のスイッチを ON にして、「0 に」のスイッチを ON にします。「～を」のスイッチのどれかを ON にしてデータをバスに乗せ、「～に」のスイッチを ON にしてそのデータをレジスタに記憶させる、これがこの計算機の操作の基本です。

177 (2 進数では 10110001) という数値データはいま、0 番地のメモリに記憶されています。これに 248 (2 進数では 11111000) を加えるには、まず入力ポートに 11111000 をセットして「I を」を ON にし、「A に」のスイッチを ON にすると、アキュムレータに 11111000 が記憶されます。これで準備ができたので、次に 0 番地のメモリのデータとアキュムレータのデータを加算します。「0 を」のスイッチを ON にして「R に」のスイッチを ON にするとレジスタ R に加算結果が記憶されます。

*25 通常コンピュータにはアキュムレータが数個、レジスタは数十個なのに対して、メモリは数十億個もあります。

*26 $(177)_{10}$ は $(10110001)_2$ です。

このように、スイッチを操作してデータをバスにのせ、スイッチを操作してレジスタに記憶させることを繰り返して計算を進めます。

Web で学ぶ 【コンピュータのしくみ】

Web ページ「コンピュータのしくみ」では、図 2.31 の各スイッチをマウスで実際に動かすことができます。データが流れる経路も分かりやすく表示されます。

5.2. 仮想コンピュータ

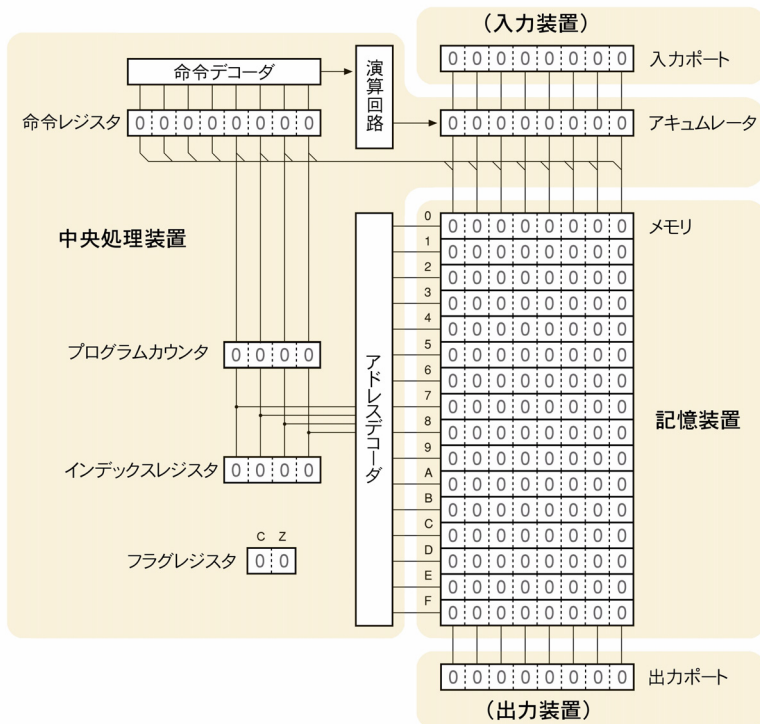


図 2.32 仮想コンピュータ

どのデータをバスに乗せるかをスイッチで切り替え、どこに記憶させるかもスイッチで指示

して演算処理を進めていくと、「計算」ができます。しかしたくさんのスイッチを人が操作していたのでは間違いが起こるし、時間もかかります。そこで操作の手順を2進数のコードにしてこれもメモリに記憶させておき、順に読み出してスイッチを自動的に動かすようにすれば、とても具合がよさそうです。

図 2.32 は、図 2.31 のような計算機にスイッチを自動的に動かすしくみを付け加えた「コンピュータ」です。自動的に動くしくみが加わったので、単純ではあっても、これは「コンピュータ」です。しかし単純すぎて実用的な仕事は殆どできないので、実際に世の中にこんなコンピュータがあるわけではなく、コンピュータのしくみを学ぶための「仮想のコンピュータ」です。これを動かすために、一連の操作の手順を2進数でコード化したものが「プログラム」です。

それではこのコンピュータについて、もうすこし詳しく見ていくことにしましょう。

1. 入力ポート

外部からプログラムやデータを入力するための、データの入り口です。このコンピュータには1個しかありませんが、通常は入力ポートがいくつもあって、キーボードやマウスなど、さまざまな入力装置が接続されます。

2. 中央処理装置

コンピュータはメモリに記憶されている命令（プログラム）をひとつひとつ実行していきます。命令が格納されているメモリのアドレスは、プログラムカウンタが数えています。プログラムカウンタの値は命令を実行する度に増えていきますが、分岐命令などがあると急に変わることもあります。このカウンタの値は、アドレスデコーダ（address decoder）によって個々のメモリへの信号に変換されます。

メモリに記憶されているプログラムは、命令レジスタ（instruction register）に読み込まれます。命令は命令デコーダ（instruction decoder）によって解釈され、演算回路に伝えられます。

3. 記憶装置（メモリ）

メモリ（memory）はプログラムや、処理すべきデータを記憶しています。プログラムやデータは特に区別する必要はなく、どこに記憶させても構いません*27。

この仮想コンピュータにはメモリが16バイトしかありませんから、複雑なプログラムやデータは記憶できません。コンピュータの性能はCPUの機能や演算速度と、メモリの容量やアクセスに必要な時間などで決まります。

4. 出力ポート

外部に処理結果を出力するための、データの出口も必要です。通常、ここにディスプレイやプリンタなどの出力装置が接続されます。

*27 ウェブページの仮想コンピュータは0番地からスタートする仕様になっていますから、少なくとも0番地はプログラムでなくてはなりません。

コンピュータにはこの他に、補助記憶装置（ハードディスクなど）が必要ですが、コンピュータの基本的なしくみを理解するだけの目的なら、なくても差し支えありません。

Web で学ぶ【仮想コンピュータ】

Web ページ「仮想コンピュータ」には、ここで説明した超単純コンピュータがあり、プログラムを書くと実際に動きます。プログラムの例がたくさんあり、メモリやレジスタの値、データの流れなども明示されるので、コンピュータのしくみが容易に理解できます。

5.3. プログラム

図 2.32 には描かれていませんが、コンピュータにはもうひとつ、重要なものがあります。それはプログラム (program) です。コンピュータはメモリ上に 2 進数で書かれている命令をひとつずつ取り出して、その指示通りに動くように作られています。したがって、メモリ上に「2 進数で書かれた命令」がないと動くことができません。これをプログラムといいます。

5.3.1. 機械語プログラム

この仮想コンピュータで 1 から 10 までの整数を加えるといくつになるかを計算するには、メモリに次のような 2 進数のプログラムとデータを書き込みます。

番地	プログラム
0	00000001
1	00111111
2	10011111
3	11010001
4	00000011
番地	データ
F	00001010

これが 2 進数で書かれた機械語のプログラム (machine language program) です。この場合は 0~4 番地がプログラム、F 番地の 00001010 はデータです。コンピュータのメモリには、このようにプログラムとデータが、特に明示して区別されることなく同居しています。

5.3.2. アセンブリ言語

機械語のプログラムは、コンピュータの命令一覧表から命令コードを拾い出せば作ることができますが、短いものならともかく、何千ステップもある本格的なプログラムを、人が機械語（2進数）で書くのは大変です。そこで加算なら **ADD**、出力なら **OUT** というように、分かりやすい記号でプログラムを書くようになりました。これをアセンブリ言語（assembly language）といいます。人はこれを用いてプログラムを作り、コンピュータに入力する前に機械語に変換します。機械語への変換もコンピュータが行います。その変換プログラムをアセンブラ（assembler）、変換することをアセンブル（assemble）といいます。

先の1から10までの加算のプログラムを、アセンブリ言語で書くと次のようになります。

```

STORE    $F, 10    ; 10 -> M(15)

CLR                      ; 0 -> Acc
LOOP:  ADD    $F      ; Acc + M(15) -> Acc
      DEC_M  $F      ; M(15) - 1 -> M(15)
      JNZ   LOOP    ; if M(15) <> 0, jump to LOOP
      OUT                   ; Acc -> Output Port

END

```

機械語のプログラムに比べれば、たしかに分かりやすくなっていますが、それでもまだ、まるで暗号のようです。

上のプログラムで、最初の1行は本来のプログラムではなく、メモリのF番地にデータを書き込む指示で、実際のプログラムは2行目の **CLR** からです。プログラム中、セミicolon (;) より右側はコメントです。プログラムを作成してしばらくたつと、作った本人にも意味が分からなくなることがよくあるので、注釈を書き込めるようになっていきます。

このプログラムは、ウェブページの仮想コンピュータ上で動きます。この通りにタイプ*28して実行すると、00110111 という値が出力されて止まります。これは1から10までの整数の和の計算結果で、10進数にすると55です。

この計算は、もちろん43ページの「電子式手動計算機」でもできます。ただし「～を」や「～に」のスイッチを、嫌と言うほど切り替えなくてはなりません。

スイッチの操作の手順をプログラムにしてメモリに記憶させておいて、それを実行すると無数の「～を」や「～に」のスイッチが素早く正確に操作されて、自動的に計算が実行される…。これが「コンピュータ」です。

*28 Web ページのプログラム例からコピー&ペーストで OK です。

第3章 ハードウェア

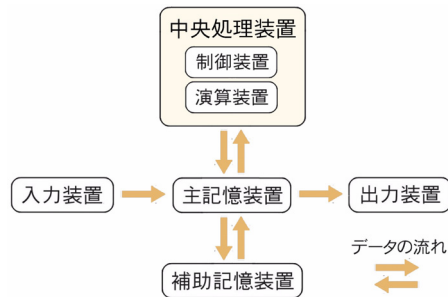


図 3.1 コンピュータの構成

コンピュータは図 3.1 のように中央処理装置、主記憶装置、入力装置、出力装置、補助記憶装置の 5 つの部分から成っています。この章ではそれぞれの装置について、そのしくみなどを調べていくことにします。

1. 中央処理装置

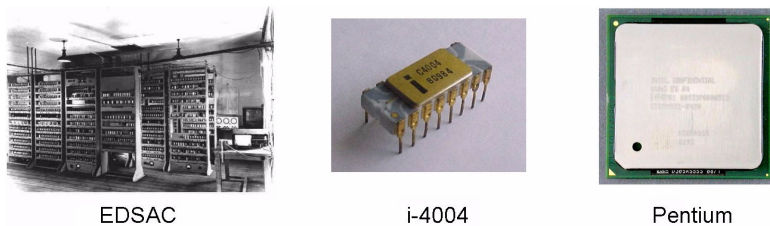


図 3.2 初期のコンピュータと MPU

図 3.2 の左は 1949 年に作られたコンピュータ、EDSAC *¹ です。写真に写っているものほとんどが中央処理装置と考えていいと思います。初期のコンピュータは真空管で作られていましたので、ラック何個分もの大きさがあります。

やがてトランジスタや集積回路 (IC) が発明されると、コンピュータもトランジスタ化、IC 化され、更に集積回路の技術の進歩とともに、CPU 全体が 1 個の大規模集積回路 (LSI) に作り込まれるようになりました。これを MPU (micro processing unit) といいます。図 3.2 の中央は 1971 年に初めて作られた MPU (i-4004)、右はパーソナルコンピュータに使われている MPU (Pentium 4) です。

*1 「コンピュータは計算機」(1 ページ)、「EDSAC」(119 ページ) 参照。

ICにはトランジスタなどの半導体素子が集積されていますが、トランジスタの寸法を小さくすればするほど同じ面積のICチップ上にたくさんの素子が集積できます。そしてまた、トランジスタを小さくすればするほど、動作速度が速くなります*2。

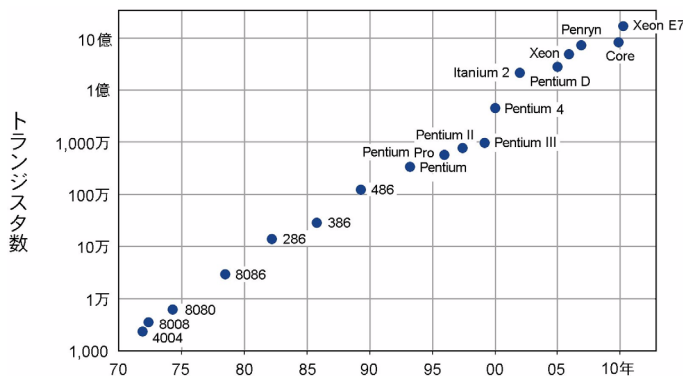


図 3.3 MPUの集積度の推移

ICの製造技術の進歩に伴ってMPUは高機能・高速になり、コンピュータの性能も飛躍的に高まりました。図3.3はインテル製のMPUに集積されているトランジスタ数の推移を表しています。初めて作られたMPU、i-4004（1971年）には2,300個のMOSトランジスタが集積されていましたが、Xeon E7（2011年）では26億個にも達しています。

1965年、インテル社のムーア（Gordon E. Moore）（1929～）は、「半導体の集積度は1年半～2年で倍増する」という経験則を提唱しました。これをムーアの法則（Moore's law）といいますが、図3.3を見ると、40年もの間、半導体技術がこの法則通りに進歩し続けてきたことが分かります。

1.1. クロック

CPUの中には演算回路やレジスタ、カウンタなどがたくさん組み込まれています。これらの回路がそれぞれタイミングを無視して勝手に動く、CPUは正しく動作することができません。オーケストラのメンバーがめいめい勝手に演奏したのでは音楽にならないのと同じように、CPUでも、ある信号はとうに変わっているのに他はまだ元のまま、ということが起こります。オーケストラに指揮者が必要なのに同じように、CPUにもタイミングを揃えるための信号が必要です。これをクロック（clock）といいます。

クロック信号は図3.4のように、正確に0と1とを繰り返すデジタル信号です。たとえばクロック信号が1から0に変化する（図の矢印）瞬間に、CPU内部の各回路はいつせいにその状態を変化させます。これによって内部の各信号の同期がとられます。

*2 半導体の材料が同じであれば、電子が動く速度自体はほとんど変わりませんが、電子がトランジスタを通り抜ける時間はトランジスタが小さいほど短くなるからです。

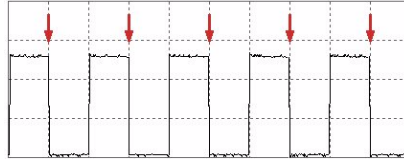


図 3.4 クロック信号

世界で最初に作られたコンピュータ、ENIAC のクロック周波数は 5 kHz^{キロヘルツ}、最初のマイクロプロセッサ i-4004 のクロック周波数は 108kHz でした。それぞれ 1 秒間に 5,000 回、あるいは 108,000 回、CPU 内部の回路が動いたことになります。

今日のマイクロプロセッサのクロック周波数は 3 GHz^{ギガヘルツ}に達しています。これは CPU が 1 秒間に 30 億回も動くスピードです。すなわち、現在の CPU は 30 億分の 1 秒に 1 回、何らかの仕事をしていることになります。30 億分の 1 秒といえ、1 秒間に 30 万 km 進む光でさえ、わずか 10cm しか進めないほどの「一瞬」です。CPU がクロックの度に行う仕事は極めて単純ですが、猛スピードで繰り返すため、複雑な処理もあつという間に片づくように見えます。

クロック周波数が高ければ高いほど CPU は速く動きますが、使用されている真空管やトランジスタ等によって決まる限界があり、クロック周波数だけを無闇に高くしても CPU は追従できません。指揮者がどんなに速くタクトを振っても、オーケストラには演奏できる速度の限界があるのと同じです。

1.2. CPU のビット数

CPU が一度に演算処理できるデータ（2 進数）の桁数を CPU のビット数といいます。大型コンピュータのビット数は 32 ビットや 64 ビットですが、マイクロプロセッサでは最初の i-4004（1971 年）が 4 ビット、i-8008（1972 年）は 8 ビット、8086（1978 年）は 16 ビット、80386（1985 年）は 32 ビット、Itanium（2001 年）は 64 ビットと、徐々にそのビット数を多くしてきました。

$$\begin{array}{r}
 01010110\ 10000100\ 11100111\ 10011001 \\
 +\ 00100011\ 00011110\ 01001010\ 10000000 \\
 \hline
 01111001\ 10100011\ 00110010\ 00011001
 \end{array}$$

32 ビットの CPU は、上のような 32 ビットの加算を 1 度の演算処理で計算することができますが、8 ビットの CPU はこれを 8 ビットずつ、4 回に分けて処理しなければなりません*3。したがって、同じ加算をするにも時間がかかります。CPU のビット数が大きいほどコンピュータは高速になります。

*3 8 ビットの CPU は本来 8 ビットの 2 進数（0～255）しか扱えませんが、このような場合は先ず最下位の 8 ビットを加算し、その最上位桁で発生したキャリーを次の 8 ビットの加算時に加えれば、大きな数値の計算もできます。

2. 主記憶装置

コンピュータで、CPU に次いで重要なのは主記憶装置（メモリ）です。CPU が処理を進めるのに必要なプログラムやデータを記憶しています。CPU が「次の命令は何？」と尋ねてくるたびに、メモリは即座に答えなくてはなりません。コンピュータの処理速度はおおむね CPU のスピードで決まりますが、コンピュータは CPU とメモリが二人三脚のようにデータのやりとりをしながら処理を進めますから、CPU だけがどんなに速くても、メモリが遅いとコンピュータ全体としての処理速度は遅くなってしまいます。また、複雑なプログラムや大量のデータを扱うには記憶容量が大きくなってはなりません。

主記憶装置には CPU に劣らない速さと、十分な記憶容量が求められます。

2.1. DRAM

主記憶装置（main memory）はプログラムやデータを記憶しています。かつては磁気を利用したコアメモリが使われていましたが、現在は半導体メモリ、中でも^{ダイナミック}DRAM（dynamic random access memory）が主に使用されています。

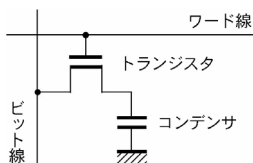


図 3.5 DRAM のメモリセル

1 ビットのデータを記憶する回路をメモリセル（memory cell）といいます。図 3.5 は DRAM のメモリセルで、1 個の MOS トランジスタとコンデンサ（condenser）でできています。トランジスタはスイッチの働きをしています。ワード線が ‘0’ のときはトランジスタが OFF、‘1’ のときは ON になります。

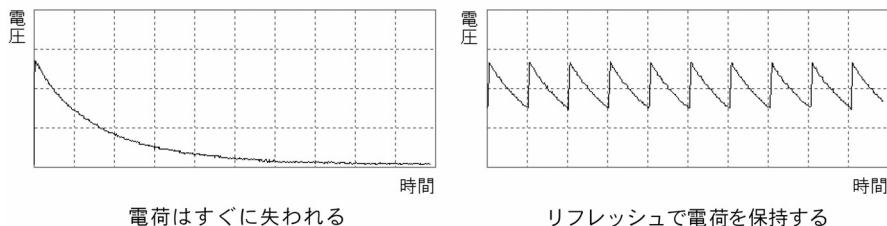


図 3.6 リフレッシュ

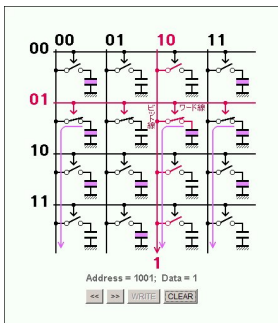
コンデンサは電荷を一時的に蓄える性質があり、メモリセルの記憶データはコンデンサに電荷があれば ‘1’、電荷がなければ ‘0’ です。トランジスタが ON になるとビット線に信号が現れます。しかしコンデンサの電荷は時間とともに失われて、図 3.6 左のように電圧は次第に下がっていきます。

コンデンサの電荷は静電気と同じです。誰にも子供だった頃、下敷きで静電気を起こして、友だちの髪の毛を吸いつけて遊んだ思い出があるでしょう。しばらくすると静電気が失われて吸いつかなくなるので、何度も下敷きを擦らなくてはなりません。コンデンサの電荷も同じで、そのままでは記憶は失われてしまいますから、再充電を繰り返すことによって記憶を保持します (図 3.6 右)。これをリフレッシュ (refresh) といい、 $100\mu\text{sec}$ ($1/10,000$ 秒) 程度の間隔で繰り返されています。

DRAM のメモリセルは図 3.5 の通り、1 個の MOS トランジスタとコンデンサだけでできています。構造が簡単なので、集積度を高めてメモリのコストを下げることができます。しかしリフレッシュが必要なため動作速度が遅く、消費電力も大きいという短所があります。

DRAM は CPU に比べると遅いので、コンピュータにとってはこれが処理速度のボトルネックになります。現在は CPU の内部に高速・小容量のメモリを設け、データを一括して読み込んでおいて、普段はそちらを参照するようにしています。これをキャッシュメモリ (cache memory) といいます。

Web で学ぶ 【DRAM】



Web ページ「DRAM」では、16 ビットのメモリセルが表示されます。

コンデンサに蓄えられた電荷はリフレッシュによって保たれていますが、これは色の変化で表されています。

任意のセルをクリックすると、セルのトランジスタが ON になって、データがビット線上に現れる様子が表示されます。

2.2. ROM

DRAM はスイッチを切ると記憶が消えてしまいます。この性質を揮発性 (volatile) といいます。スイッチを切るとメモリは空っぽになりますから、次にコンピュータのスイッチを入れたとき、プログラムがないのでコンピュータは動くことができません。

そこで主記憶装置の一部を、スイッチを切っても記憶が消えない不揮発性 (nonvolatile) のメモリにしておき、そこに書き込まれている「メモリにプログラムを読み込むプログラム」を実行します。そしてハードディスクから OS *4 (operating system) のプログラムをメモリに読み込んで、OS を起動します。これではじめてコンピュータが使用できる状態になります。このような場合に使われる不揮発性メモリを、ROM (読み出し専用メモリ: read only memory) といいます。

*4 基本ソフト。「オペレーティングシステム」(90 ページ) 参照。

3. 補助記憶装置

主記憶装置は揮発性のため、コンピュータの電源を切るとメモリのデータは消失します*5。次にスイッチを入れたとき、速やかに主記憶装置にプログラムを読み込むことができるようにするには、別途不揮発性（nonvolatile）の「補助記憶装置」を用意する必要があります。

不揮発性とは、メモリの電源が切れても記憶が消滅しない性質です。補助記憶装置には大量のデータやプログラムを長期間保存しようというのですから、電源を切ると記憶も消える、では使い物になりません。

また、コンピュータは様々なプログラムを実行し、大量のデータを扱います。それらのプログラムやデータをすべて記憶しておくためには、補助記憶装置の記憶容量は十分大きいものでなくてはなりません。

今日用いられている補助記憶装置は主として磁気ディスクと光ディスクです*6。中でも特にハードディスクドライブ（HDD: hard disk drive）とCD-ROM/R、DVD-ROM/RAM、ブルーレイディスクドライブなどの光ディスクがよく使われています。ハードディスクは記憶容量、速度、価格などの点でバランスのよい、補助記憶装置に相応しい特徴を持っています。また最近では半導体技術の進歩に伴ってフラッシュメモリの大容量化、低価格化が進んでいます。これを利用したUSBメモリは手軽なため、よく利用されるようになりました。

名 称	記録方式	記憶容量（代表値）
ハードディスク	磁気ディスク	さまざま（～数100GB）
USBメモリ／メモリカード	半導体	さまざま（～数10GB）
ブルーレイ・ディスク	光ディスク	25GB
DVD-ROM/RAM	光ディスク	4.7GB
CD-ROM/R/RW	光ディスク	650MB

表 3.1 主な補助記憶装置

3.1. ハードディスク

ハードディスクは、金属（またはガラス）製の、磁性物質を塗布したプラッタ（platter）と呼ばれる円盤にデータを記録します。データは磁気によって記録されます。ビデオテープやカセットテープに動画や音楽を録画・録音するのと、基本的な原理は同じです。

図 3.7 の上部の円盤がプラッタで、3.5 インチ型と呼ばれるものの直径は約 95mm です。鏡のように仕上げられた金属製円盤の表面に磁性体が塗布されています。

左下から斜めに伸びている鉄のようなものがアームで、この先端に磁気ヘッドがあります。

*5 昔のコンピュータには、「磁気コアメモリ」という不揮発性メモリが使用されていました。120 ページ参照。

*6 大量のデータのバックアップの目的で磁気テープも使われますが、一般的ではありません。



図 3.7 ハードディスクの内部

アームは左下の軸を中心に回転し、磁気ヘッドはディスク面を円弧状に素早く移動します。磁気ヘッドはデータを読み書きする、最も重要な部品のひとつです。

3.1.1.1. トラックとセクタ

プラッタは円盤ですから、データを保存する場所を管理するために、まず同心円状に細かく分割します。これをトラック (track) といいます。複数のプラッタが組み込まれたときの*7、トラックの仮想的な円筒をシリンダ (cylinder) といいます。

トラック一周に記録できるデータは膨大で、このままで使用すると無駄が多くなるため、トラックは更に分割します。これをセクタ (sector) といいます。セクタはハードディスクに情報を記録する最小の単位で、ここに通常 512 バイトのデータが記録されます。

図 3.7 のハードディスクは 54,982 トラック、外周部は 950、内周部は 486 のセクタに分割されています。トラック密度は 56,170 tpi (tracks per inch) なので、トラック一本の幅は $25.4\text{mm} \div 56,170 \approx 0.45\mu\text{m}$ 、セクタの長さは約 0.3mm にすぎません。幅 $0.45\mu\text{m} \times$ 長さ 0.3mm のディスクの表面に、512 バイトのデータが記録されます。

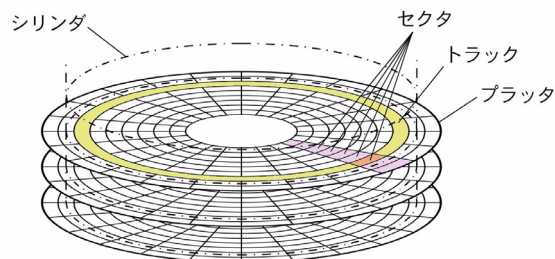


図 3.8 トラックとセクタ

*7 図 3.7 のハードディスクには 2 枚のプラッタが使用されています。

3.1.2. 面記録密度

単位面積に記録できるデータ量を面記録密度 (surface recording density) といいます。ハードディスクの場合、通常1平方インチに書き込むことができるビット数 (ビット/インチ²) で表しますが、図 3.9 はその推移を表したものです。●は研究室レベル、○は製品としてのハードディスクの面記録密度を表しています。最近ではほぼ5年で10倍に近いペースで増え続けていることが分かります。

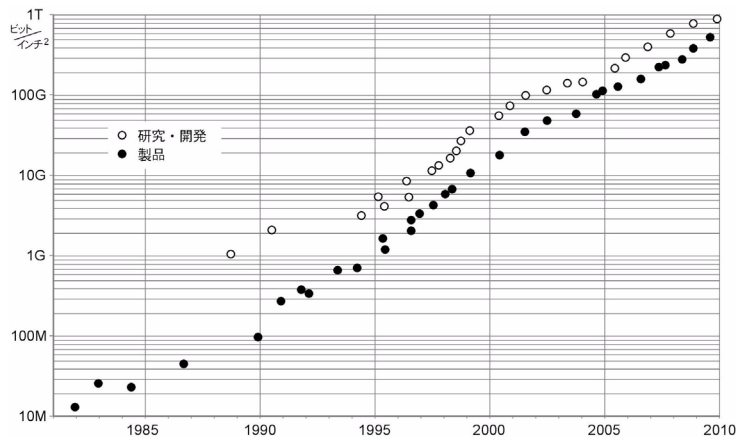


図 3.9 ハードディスクの面記録密度の推移

図 3.7 のハードディスクの面記録密度を計算すると約 20G ビット/(インチ)² になります。最近のハードディスクの面記録密度はこれより 2 桁ほど大きいので、トラック幅もセクタの長さも、単純に考えるとそれぞれ $\frac{1}{10}$ 、幅 45nm × 長さ 30 μ m になっていることになります。

紙に文字を書く場合でも、文字を小さくすればするほどたくさん書けます。しかし文字が小さいと読みづらいのと同じように、ハードディスクも面記録密度を高めればデータの記録面積が小さくなるので、読み出し時の電気信号も弱くなります。面記録密度を高めると高感度の磁気ヘッドが必要になり、磁気ヘッドの位置の精度もなお一層要求されます。また、あまりにも記録面積が小さくなりすぎると、時間の経過と共にデータが失われるという現象も起こります。これらのさまざまな問題を克服するために、GMR*⁸ (giant magnetoresistive)、TMR (tunneling magnetoresistive)、垂直磁気記録方式などの技術が次々に開発され、面記録密度は現在も高め続けられています。

3.1.3. 高速回転と磁気ヘッドの浮上

ハードディスクのプラッタは 7,200 rpm*⁹ という高速回転をしています。このときプラッタは、外周部では時速 120km 以上という猛スピードで動いています。カセットテープやビデオ

*⁸ GMR (巨大磁気抵抗効果) を発見したアルベール・フェール (Albert Fert) とペーター・グリュンベルク (Peter Grünberg) は、2007 年ノーベル物理学賞を受賞しました。

*⁹ rpm: 1 分間当たりの回転数 (revolution per minute)。「rpm」(227 ページ) 参照。

オテープなどのように、磁気ヘッドと磁性体が接触していたのでは、すぐにヘッドが摩耗してしまいます*10。そこで、摩擦を避けるために、ディスクが回転すると磁気ヘッドはごく僅か(0.01 μ m程度)浮き上がるようになっています。ディスクが回転すると、ディスクの近くでは空気も一緒に動くので、ディスク近傍には空気の流れができます。この空気の流れによる浮力によって、磁気ヘッドは浮上します。

ハードディスクはこのようなデリケートな機構を持っているので、振動や衝撃は禁物です。特にディスク回転中は細心の注意が必要です。

3.1.4. フォーマット

製造されたばかりのハードディスクのプラッタは、トラックもセクタもない、のっぺらぼうの円盤です。これをどれだけトラックやセクタに分割するかを決め、その情報を書き込むことを物理フォーマット (physical format)、または低レベルフォーマット (low level format) といいます。

さらに、実際にデータを書き込む論理的な位置を決め、目次に相当するデータを設定する、などの作業を論理フォーマット (logical format) といいます。

通常フォーマットはディスクを使用する前に行いますが、使用中のハードディスクを再度フォーマットすると、記録されていたデータは全て消えてしまいます。事前にデータのバックアップを取るなどの注意が必要です。

3.2. CD

3.2.1. CDのしくみ

CD (compact disc) は、光の波の性質を利用してデータを読み出します。1977年にソニーとフィリップスによって開発されました。



波の性質にもいろいろありますが、よく目にするのは干渉 (interference) です。

水面で二つの波が出会うと特徴のある模様ができたり、楽器をチューニングするときに「うわ〜んうわ〜ん」という唸^{うな}りを利用したり、しゃぼん玉を大きくふくらませるときれいな虹色の模様が現れたり*11、これらはいずれも波の干渉によるものです。

*10 カセットテープの速度は時速約 170m、ビデオテープの場合でもテープとヘッド間の速度は時速約 9km です。

*11 この現象は、しゃぼん玉の膜の表面で反射した光と裏面で反射した光が干渉することによって起こります。

干渉は二つの波が出会ったときに起こります。



図 3.10 波の干渉

図 3.10 で、左は二つの波の山 (●印) と山の位置が一致しています。これを「位相が揃っている」といいます。位相がそろった二つの波が出会うと、山と山、谷と谷が干渉によって強めあって波の振幅が大きくなります。その結果、光の波の場合は明るくなります。

図 3.10 の右は、上の波の山のところは下の波では谷 (○印) になっています。山から山まで (●から●まで) や谷から谷までの長さを波長 (wave length) といいますが、この場合は二つの波はちょうど波長の半分だけ位相がずれています。このような二つの波が出会うと、山と谷、谷と山が干渉によって打ち消し合って波は消えてしまいます。光の場合は暗くなります。

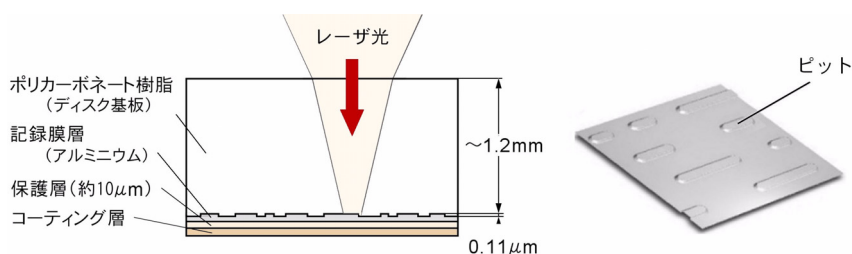


図 3.11 CD の構造

CD の情報はピット (pit) にデジタル信号として記録されています。ピットは幅 $0.5\mu\text{m}$ 、長さ $0.83\mu\text{m}\sim 3.56\mu\text{m}$ ^{*12}、深さ $0.11\mu\text{m}$ のくぼみです。この $0.11\mu\text{m}$ という深さは、レーザ光の波長の約 $\frac{1}{4}$ になるように選ばれています。図 3.11 ではピットは長円形の突起のように見えますが^{*13}、CD は直径 12cm、厚さ 1.2mm のポリカーボネート樹脂製の円板で、ピットはその表面に作られた微細なくぼみです (これにレーザの光を反射させるためのアルミニウムの記録膜層をつけて、ポリカーボネート樹脂板の方から見ると突起のように見えます)。情報を読み出すときは矢印の方向からレーザの光を当てます。レーザ光は記録膜層で反射されて戻ってきますが、このときピットに当たった光は、ランド (ピットでないところ) に当たった光よりも手前で反射されます。

図 3.12 は CD の記録膜層でのレーザ光の反射の様子を表したものです。記録膜層に向かって進む位相が揃った二つの光をうすいグレーで描いてありますが、図 3.12 の左はランドで反射された光 (赤色) も位相が揃っています。この場合反射光は明るく見えます。

*12 $0.3\mu\text{m}$ ずつ長さが違うものが 9 種類あります。

*13 ピットの図 出典: 情報機器と情報社会のしくみ素材集 (<http://www.sugilab.net/jk/joho-kiki/index.html>)



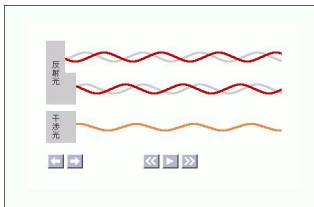
図 3.12 CD の記録膜層での反射と反射光の干渉

一方、ピットは波長の約 $\frac{1}{4}$ 凹んでいるので、ピットに当たった光が通る経路は、ランドに当たった光の経路よりも往復で $\frac{1}{2}$ 波長短くなっています。その結果、図 3.12 の右のようにランドに当たった光とピットに当たった光の反射光は、波長の $\frac{1}{2}$ だけ位相がずれます。この二つの光が干渉すると山と谷、谷と山が打ち消し合って、反射光は見えなくなります。

ピットの深さは $0.11\mu\text{m}$ 、ごく僅かですが、光の干渉によって大きい明暗差ができます。CD は光のこの性質を利用して、データを読み出しています。

CD-ROM は CD のデータ誤り検出機能を強化してコンピュータの補助記憶装置に利用したもので、650MB の記憶容量があります。

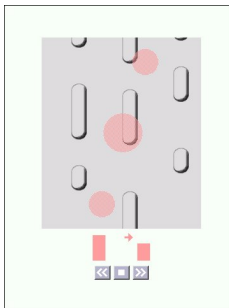
Web で学ぶ 【CD】



Web ページ「CD」には、光の干渉の様子を示す Java アプレットがあります。

ピットの深さが変わると、反射光が干渉して光の強さがどう変わるのが動画で表示されるので、光の干渉や CD のしくみが容易に理解できます。

Web で学ぶ 【CD の制御機構】



レコードの溝に針を下ろすと、溝に記録された音を再生しながら、針は溝に導かれてレコードのお終いまで再生されます。レコードなら当たり前ですが、溝も針もない CD で、なぜ同じことができるのでしょうか。

本書では触れていませんが、Web ページ「CD の制御機構」にはこれを説明する Java アプレットの動画があります。上の「CD」が理解できていれば、これもきっと理解できます。

3.3. DVD、ブルーレイ・ディスク

コンピュータの処理能力が向上するにつれ、動画もコンピュータで処理するようになってきました。しかし動画のデータ量は膨大です。CD はもともと音楽のデータを扱うために作られたメディアで、動画のデータを記録するには容量が十分ではないため、DVD (digital versatile disc) やブルーレイ・ディスク (blu-ray disc) が開発されました。いずれも基本的な原理は CD (650MB) と同じで、ディスクの直径も CD と同じ 12cm ですが、DVD は 4.7^{ギガバイト}GB、ブルーレイ・ディスクは 25GB の記憶容量があります。

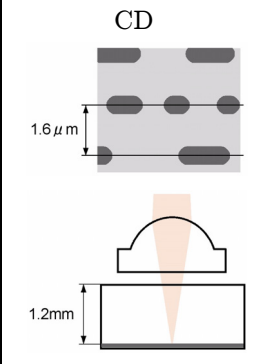
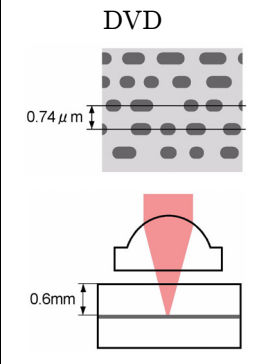
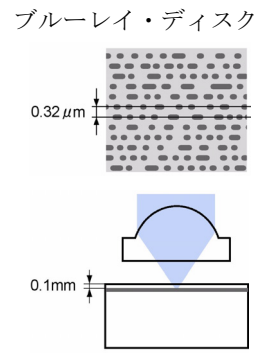
メディア	CD	DVD	ブルーレイ・ディスク
			
トラックピッチ	1.6 μm	0.74 μm	0.32 μm
レンズ開口数	0.45	0.65	0.85
使用レーザー波長	780nm	650nm	405nm
カバー層厚さ	1.2mm	0.6mm	0.1mm
記憶容量	650MB	4.7GB	25GB

表 3.2 各種光ディスクの比較

限られた面積のディスクにたくさんの情報を記憶させるには、ピットを小さくするのが早道です。ハードディスクもトラックピッチなどを小さくすることによって面記録密度を高めてきましたが*14、光ディスクでも事情は同じで、CD には波長 780nm の赤外線レーザーが使われているのに対して、DVD には 650nm の赤色レーザー、ブルーレイ・ディスクには 405nm の青色レーザーが使われています。波長が短くなればなるほど、小さいピットを読みとることができるからです。また、開口数*15 (NA : numerical aperture) の大きいレンズを使うことによってレーザービームを小さく絞り込むようにします。CD のレンズの開口数は 0.45、DVD は 0.65 なのに対して、ブルーレイ・ディスクには 0.85 という開口数のレンズが使われています。

なお、コンピュータの補助記憶装置としては、読み出し専用の CD-ROM、DVD-ROM、BD-ROM のほか、データを書き込むことができる (消去はできない) CD-R や DVD-R、BD-R などの追記型 (recordable)、消去が可能で書き換えもできる CD-RW、DVD-RW、BD-RE

*14 「面記録密度」(56 ページ) 参照。

*15 レンズの集束能力を表す指数。NA が大きいほどレンズの分解能は高い。

(rewritable) など、さまざまなタイプのディスクがあります。

3.4. USBメモリ、メモリカード

USBメモリ (USB memory) もメモリカードも、記憶素子にフラッシュメモリ (flash memory) を使っています。フラッシュメモリは不揮発性*16 の半導体メモリで、近年技術革新がめざましく、急激に大容量化・低価格化が進んでいます。



図 3.13 USBメモリと各種メモリカード

USBメモリは、USB (universal serial bus) コネクタに接続して使用するメモリです。USBにはハードディスクやフロッピーディスクなどの補助記憶装置を直接接続でき、ドライバなどをインストールする必要もありません。ハードディスクのような機構部がないので小型で軽量、寿命も長く、安価で手軽な補助記憶装置として普及しています。

メモリカード (memory card) はデジタルカメラの画像データ保存用として普及しはじめました。コンパクトで記憶容量もさまざまなものがあるので、デジタルカメラや携帯音楽プレーヤー、携帯電話などのほか、一般的な補助記憶装置としても使われています。

メモリカードの規格はさまざまですが、代表的なものにはコンパクトフラッシュやスマートメディア、SDメモリーカード、メモリースティックなどがあります。

*16 「補助記憶装置」(54 ページ) 参照。

4. 入力装置

入力装置はコンピュータにデータやプログラム等を入力するためのもので、キーボードやマウスなどがその代表です。その他タッチパネル、タブレット、イメージスキャナ、バーコードリーダー、カメラ、マイクなども入力装置の仲間です。

4.1. キーボード

キーボードは文字や数字の入力をするために使う、最も身近な入力装置です。

キーが押されると、内蔵されているマイクロコンピュータ^{*17} が、どのキーが押されたかを調べ、そのキーに相当する文字コードを出力します。シフトキーやコントロールキーと組み合わせて押されたときの処理、キーを長時間押し続けると自動的に連続出力される機能なども、このマイクロコンピュータによるものです。

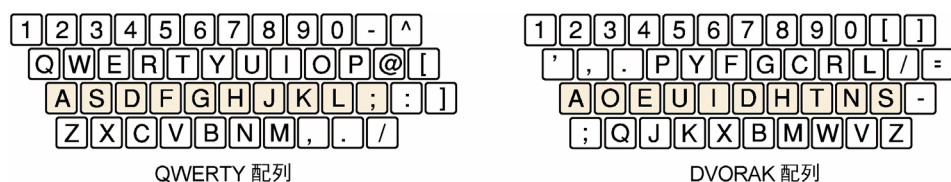


図 3.14 QWERTY 配列と DVORAK 配列

一般的に使われている図 3.14 左のようなキーボードの文字配列を、英字上段の配列が「QWERTY…」であることから、^{クワージー}QWERTY配列とといいます。コンピュータのキーボードはタイプライターのキーボードをそのまま継承したのですが、タイプライターは速くキーを打ちすぎると活字のバーが絡まってしまうので、速く打ちにくいように工夫されています。

文字	文字数	%	文字	文字数	%	文字	文字数	%
A	8,240	7.79	J	277	0.26	S	6,584	6.22
B	1,715	1.62	K	831	0.79	T	9,709	9.17
C	2,284	2.16	L	4,976	4.70	U	3,759	3.55
D	3,934	3.72	M	3,358	3.17	V	1,109	1.05
E	12,818	12.12	N	6,468	6.11	W	2,519	2.38
F	2,030	1.92	O	8,805	8.32	X	125	0.12
G	1,838	1.74	P	1,551	1.47	Y	2,623	2.48
H	6,787	6.41	Q	65	0.06	Z	32	0.03
I	6,909	6.53	R	6,488	6.13	記号	6,646	—

表 3.3 英文（ロミオとジュリエット）中の文字の使用頻度

表 3.3 は「ロミオとジュリエット」に使われている文字数を数えたものです^{*18}。この表で分かるように、英文で最もよく使われる文字は E、T、O、A、I、H などですが、QWERTY 配

^{*17} 「マイクロコンピュータ」(7 ページ) 参照。

^{*18} 出典：“The Complete Works of William Shakespeare” (<http://the-tech.mit.edu/Shakespeare/>)
数字は四捨五入しているため、合計は 100% にはなりません。記号の数は計算に含まれていません。

列のキーボードではいずれも上段にあるか、左手や、小指、薬指に割り当てられていたり、打ちやすい文字は H だけです。

図 3.14 右は、あまり見かけませんが、^{ドボラック}DVORAK配列といます。よく使われる文字がホームポジション*19 や打ちやすいキーに割り当てられていたり、母音と子音が左右に分かれているので両手で交互に打つ機会が多くなる、といった工夫がされています。

Two households, both alike in dignity,
In fair Verona, where we lay our scene,
From ancient grudge break to new mutiny,
Where civil blood makes civil hands unclean.

QWERTY 配列

Two households, both alike in dignity,
In fair Verona, where we lay our scene,
From ancient grudge break to new mutiny,
Where civil blood makes civil hands unclean.

DVORAK 配列

図 3.15 指の上下移動量の比較

図 3.15 は「ロミオとジュリエット」の冒頭部ですが、これをそれぞれのキーボードで入力したとき、指を上下に動かさなくてはならない文字には色をつけてあります。ピンクの文字は上段、ブルーの文字は下段にあります。ロミオとジュリエット全文を入力した場合、QWERTY 配列ではホームポジションで打てる文字は 34%にすぎませんが、DVORAK 配列だと 66%がホームポジションで打てます。どちらが打ちやすいかは言うまでもありません。

タイプライターには不合理な配列もやむをえなかったかもしれませんが、コンピュータの入力装置として使われるようになったときにも、合理的な配列に変更されることはありませんでした。一度デファクトスタンダード (de facto standard) *20 となったものは、多少の不都合があってもなかなか覆りません。QWERTY 配列のキーボードはその好例です。

4.2. マウス

画面上のカーソルの位置を指定する入力装置をポインティングデバイス (pointing device) といいます。マウス (mouse) は最も代表的なポインティングデバイスで、手に収まる本体とケーブルが尾の長いネズミに似ているのでその名があります。

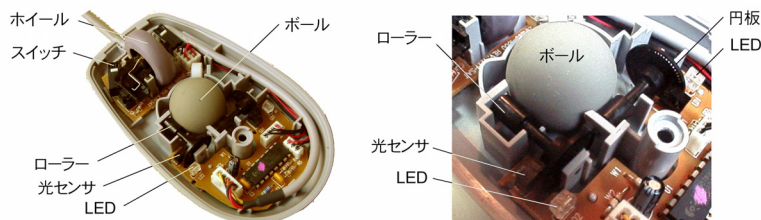


図 3.16 マウスの内部

図 3.16 はマウスの内部です。このタイプの、中央にボールがあるマウスは最近ほとんど見かけませんが、マウスのしくみはこちらの方が直感的に理解できます。

*19 タッチタイピングで、指が常に置かれている位置。図 3.14 の網かけ部。

*20 業界標準、事実上の標準。

第3章 ハードウェア

このボールにローラーが接していて、マウスを動かすとボールが回転し、それに接しているローラーも回転します。ローラーには穴があいた円板が取り付けられていて、これも回転します。円板の両側には赤外線 LED（発光ダイオード : light emitting diode）と光センサが向かい合っていて、円板が回転すると LED の光を遮ったり通したりします。光の ON-OFF の回数を数えれば、ボールの回転の様子が分かります。ローラー、円板、光センサは 90° の角度で 2 組取り付けられているので、あらゆる方向へのマウスの動きを知ることができます。

最近使われているのは光学式マウスです。机やマウスパッドなどの表面にある微細な模様を画像処理することによってマウスの移動量を算出します。

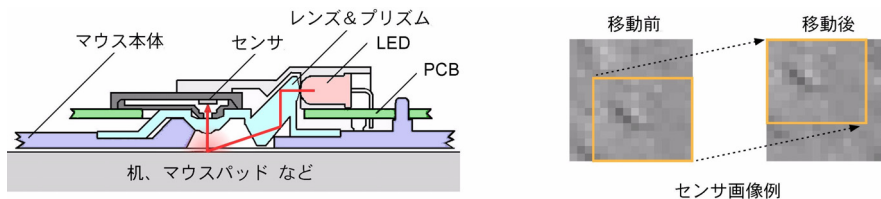
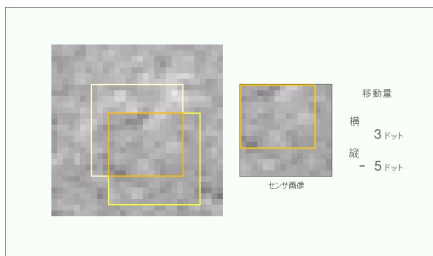


図 3.17 光学式マウスのしくみ

光学式マウスの主要部は図 3.17 のようになっています。LED から出た光はプリズムで 2 度曲げられて斜めに機の表面に当たり、ここで乱反射します。反射光のうち、真上に反射したものがレンズを通してセンサに入ります。機の表面には細かい不規則な模様がありますから、センサはそれを拡大して画像データとして取り込みます。図 3.17 の右はその画像例ですが、マウスが移動するとこの画像も変わります。移動前の画像と比較すると、線で囲んだ共通部分が左下から右上に移動しているのので、マウスが左下に動いたことが分かります。光学式マウスはこのような画像処理を、1 秒間に数千回繰り返して行っています。

机やマウスパッドの表面に光沢があると、斜めに当たった光は反射の法則通り前方に進んで、真上のセンサには光がほとんど入らなくなるので、マウスは操作できなくなります。

Web で学ぶ【光学式マウス】



Web ページ「光学式マウス」では、マウスの光センサの画像がどのように変化するかを表す Java アプレットがあります。マウスが動く前と動いた後の画像を比較することによって、マウスの移動量を算出するしくみが理解できます。

4.3. イメージスキャナ

イメージスキャナ (image scanner) は、イラストや写真を画像データとしてパーソナルコンピュータに入力するための入力装置です。コピー機のように原稿を走査 (scan) し、反射光を CCD (電荷結合素子 : charge coupled device) で読み取ってデジタルデータに変換します*21。



図 3.18 イメージスキャナ

コンピュータでは画像は画素 (pixel) の集まりとして表現されますが、イメージスキャナの画像のきめ細かさを表すには dpi (dot per inch) という単位が使われます。たとえば 200dpi だと 1 インチ (25.4mm) を 200 分割して画像を読み取ります。この値が大きければ大きいほど精細な画像データが得られますが、画像データが膨大になったり、処理に長い時間がかかるようにもなります。必要以上に細かくしても無意味で、目的に応じた精細度を選ぶことが大切です。

ネガフィルムを読み取ってコンピュータでポジ画像に変換して写真を印刷したり、新聞や書籍の文字部分を OCR (optical character reader : 光学式文字読み取り装置) ソフトで文字データに変換する場合などにも利用されます。

4.4. タッチパネル

タッチパネル (touch panel) は画面に指で触れて操作する入力装置で、銀行の ATM (automatic teller machine) やカーナビゲーション、最近ではゲーム機や携帯電話、タブレット端末などにも使われ、すっかりお馴染みになりました。

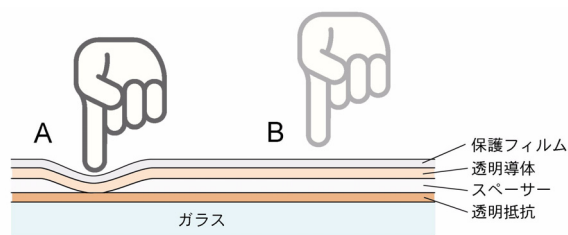


図 3.19 タッチパネルのしくみ

*21 イメージスキャナでは直線状の CCD センサが使われています。

タッチパネルには静電容量、超音波、光、抵抗膜を使ったものなど、さまざまな方式のものがあります。それぞれ一長一短があり、目的に応じて最もふさわしいものが使われています。

図 3.19 は抵抗膜式タッチパネルの構造を模式的に描いたものです。ガラス基板の上に極めて薄い透明抵抗の膜とスペーサ、透明電極、保護フィルムが重ねられています。指でタッチパネルを押すと、その部分の透明導体と透明抵抗膜が接触して電流が流れます。A 点を押したときと B 点を押したときでは透明抵抗膜の長さが違いますから、電流値を測れば押された位置を知ることができます。

携帯電話で使われているのは静電容量方式です。表面型と投影型の 2 種類があり、後者は透明フィルムに電極パターンを形成し、指が近づいたときの電極間の静電容量の変化を検出します。多点の接触を検出できるので、2 本の指で写真を拡大・縮小するような、直感的な操作ができるという特徴があります。

4.5. バーコードリーダー

バーコードリーダー (bar code reader) はスーパーやコンビニで精算するとき、レンタルビデオ店でビデオを借りたとき、学生証に印刷されているバーコードで出席を取るときなどに見かけますからお馴染みですが、レーザの光でバーコードをスキャンして読み取るという原理はどれも同じです。

バーコードリーダーの赤い光は一見「線」に見えますが、実際は高速で動いている赤い光の「点」です。それが人の目には残像によって線に見えます。

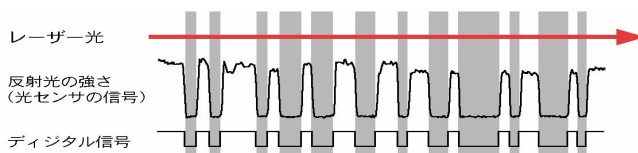


図 3.20 バーコードの読み取り

光の点がバーコードを横切るとき、黒く印刷されているバーの部分からの反射光は弱くなりますから、反射光は図 3.20 のように変化します。これを電気信号に変換して、一定のレベル以上の電圧は ‘1’、それ以下の電圧の場合は ‘0’ とすると、2 値デジタル信号になります。このデジタル信号の 0 と 1 の幅を測れば、バーとスペースの幅が分かります*22。



バーコードリーダーの光はバーと直角の方向にスキャンされるのが理想的ですが、必ずしも

*22 CCD センサを使用して、画像処理によって読みとるバーコードリーダーもあります。

その必要はありません。傾いていても、逆方向でも、いずれの場合もソフトウェアで処理できますから、レーザの光が全てのバーを横切ってさえいれば読み取ることができます。



図 3.21 主なバーコード

バーコードにはたくさんの種類がありますが、図 3.21 はそのうちの主なものです。

左のJAN (japan article number) は最もおなじみの共通商品コード用バーコードで、たいていの商品にはこれが印刷されています。ヨーロッパのEAN (european article number)、米国の UPC (universal product code) とも互換性があり、1985年に JIS X 0501 として JIS 規格化されました。バーとスペースには4種類の太さがあるため、その太さにも情報が含まれているため、小さいバーコードでも効率よく情報を表すことができます。しかしその反面、高い印刷精度と読取り精度が要求されます。

JANの最初の2桁は国コードで、日本は49または45です。次の5桁^{*23}(図では「12345」)は商品メーカーコードで、(財)流通システム開発センターで管理されています。次の5桁(図では「67890」)はメーカーが独自に決める商品アイテムコードです。最後の1桁(図では「4」)はバーコードの読み取りが正しいかどうかを調べるためのチェックデジットです。

図 3.21 の中央はCODABAR (JIS X 0506) です。Narrow と Wide の2種類の太さのバー4本、スペース3本の計7本で1キャラクタが表されることから、NW-7とも呼ばれています。バーの太い細いが判別できればいいので、JANほど高度な印刷精度は要求されません。図書館の本の貸し出し管理、各種会員カードや学生証などで広く利用されています。

図 3.21 の右はQR (quick response) コード (JIS X 0510) です。QR コードは2次元バーコードで、数字なら最大7089文字、英数字は4296文字、漢字の場合1817文字を表すことができます。たくさんの情報を書き込むことができるので、日本自動車工業会などで早くから使用されていましたが、QR コード読み取り機能がある携帯電話の普及に伴って、次第に目にする機会が増えてきました。

*23 2001年1月より新規に割り当てられる商品メーカーコードは7桁に、商品アイテムコードは3桁になっています。この場合の国コードは45が使われます。

5. 出力装置

コンピュータの処理結果は、出力されなければ人は知ることができません。そのため、ディスプレイの画面に表示したりプリンタで印刷したり、出力装置の出番です。

5.1. ディスプレイ

ディスプレイ (display) は画面に文字や図、写真などを表示する出力装置です。かつてはディスプレイといえば CRT (cathode ray tube) でしたが、液晶ディスプレイの生産技術、品質の向上と低価格化によって、パーソナルコンピュータのディスプレイはほとんどが薄くて軽い液晶に置き換わりました。

5.1.1. ディ스플레이のサイズ

ディスプレイでは文字や図を小さな点の集合として表示します。この点を画素 (ピクセル: pixel *²⁴) といいます。

ディスプレイにはさまざまなサイズのものがありますが、サイズが大きくなると表示できる画素数が増えます。ディスプレイの画素数には次のようなさまざまな規格があります。



図 3.22 ディスプレイ画素数の種類

VGA (video graphics array) は初期のパーソナルコンピュータに採用されていたもので、640 × 480 画素で縦横の比が 3 : 4、従来のテレビと同じ比率です。現在パーソナルコンピュータで主として使われているのは XGA (extended graphics array) や SXVGA (super extended VGA) で、横長のものも増えてきました。

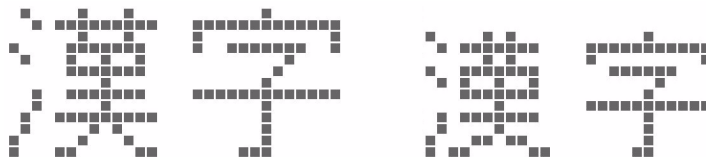


図 3.23 ディスプレイでの文字の表示

ディスプレイに表示されている文字を拡大すると、文字は図 3.23 のように画素の集合として表現されていることが分かります (小さい正方形が画素です)。

画素の大きさはディスプレイによって決まっていますから、文字を小さく表示するには、文

*²⁴ picture の複数形 pix と element からの造語。

字の画素数を少なくしなければなりません。注意して見ると、図 3.23 の「漢」のように、画数の多い文字は小さく表示すると字画が省略されていることが分かります。

5.1.2. 色の表現

ディスプレイでは色も自在に表示できますが、これらのさまざまな色は光の三原色、赤 (R)・緑 (G)・青 (B) を重ね合わせることによって作られています。各色の明るさを 8 ビットで表すことにすると、それぞれ $2^8 = 256$ 種類の階調 (明るさ) で表示できます。画面の色は RGB 3 色を組み合わせていますから、各色がそれぞれ 256 種類の階調を持っていると、

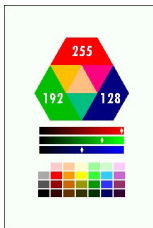
$$256 \times 256 \times 256 = 16,777,216 \text{ (色)}$$

が表示できることとなります。1 ピクセルの色を表示するのに RGB 各 1 バイト、計 3 バイトのデータが必要ですから、XGA の画面を表示するには、

$$1024 \text{ ピクセル} \times 768 \text{ ピクセル} \times 3 = 2,359,296 \text{ バイト} \approx 2.3 \text{ MB}$$

のメモリが必要になります

Web で学ぶ 【色の表示】



Web ページ「色の表示」では、光の三原色 RGB の組み合わせによって、いろんな色が表示される Java アプレットがあります。R、G、B 各色の明るさをマウスで簡単に変えることができます。RGB 三色の組み合わせで 1667 万色もの色が表示できることが、すぐに理解できます。

5.1.3. 液晶ディスプレイ

液晶とは有機物の一種で、流動性があるため液体であると同時に分子配列が規則的で、あたかも結晶のような性質をもっている物質です。オーストリアの植物学者ライニツァー (F. Renitzer) によって 1888 年に発見され、1968 年に RCA 社のハイルマイヤー (Heilmeyer) によって液晶の性質を応用した表示装置が試作されました。この試作品は短寿命であるなど実用性が乏しいものでしたが、1973 年、シャープが電卓の表示装置として液晶ディスプレイ (LCD: liquid crystal display) の実用化に成功しました*25。

液晶ディスプレイも光の、波としての性質を巧みに利用しています。

ピンと張ったロープを想像して下さい。ロープを持った手を上下に振ると、その振動は波となってロープを縦方向に振動させます。手を横に振るとロープは横方向に振動します。斜めに

*25 「電卓戦争の終焉」(132 ページ) 参照。

振っても同様です。ロープはどんな方向にでも振動させることができます。

光も波ですが、ロープの波と違うのは、普通の光にはあらゆる方向に振動している波が混ざり合っていることです。光はあらゆる方向に振動している無数の波の束、と考えればいいでしょう。

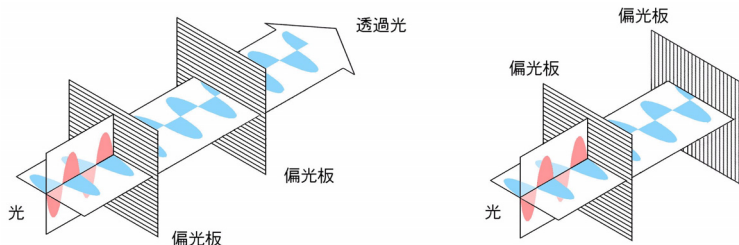


図 3.24 偏光

図 3.24 のように、光が偏光板 (polarizer) に当たると、偏光板の方向と一致している振動方向の光だけが偏光板を通り抜けます*26。もう一枚の偏光板を平行に置くと光は通り抜けますが、直角に置くと光は全く通らなくなります。



図 3.25 ロープの波と柵

ロープの波の場合、ロープの振動方向 (図 3.25 では縦) に平行な柵があっても支障ありませんが、直角に置かれた柵は通り抜けることができないのと同じです。

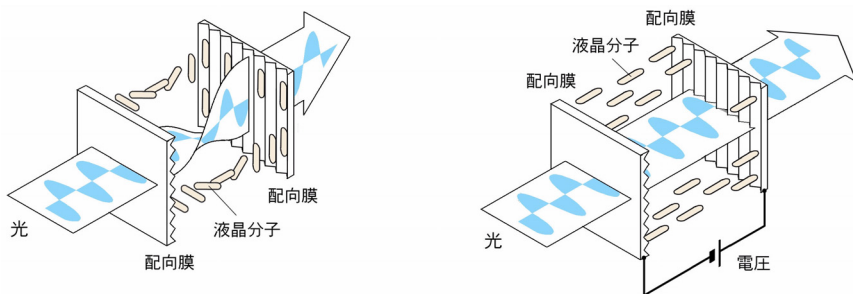


図 3.26 液晶

液晶は細長い分子がゆるやかな規則で並んだものです。液晶分子は微細な溝の付いた配向膜

*26 図 3.24 では便宜的に水平と垂直に振動している波だけを描いていますが、実際の光にはあらゆる方向に振動している波が含まれています。

(alignment layer) に接触すると、溝にそって並ぶ性質を持っています。溝の方向を 90° の角度で配置した配向膜の間に液晶を入れると、液晶の分子も 90° ねじれます。そして、偏光板を通りぬけた光がこの液晶の中を通過するとき、光の振動方向も液晶分子のねじれにそって 90° ねじれます (図 3.26・左)。一方、液晶に電圧をかけると分子は電界に沿って並び、光は振動方向がねじれることなく進むようになります (図 3.26・右)。

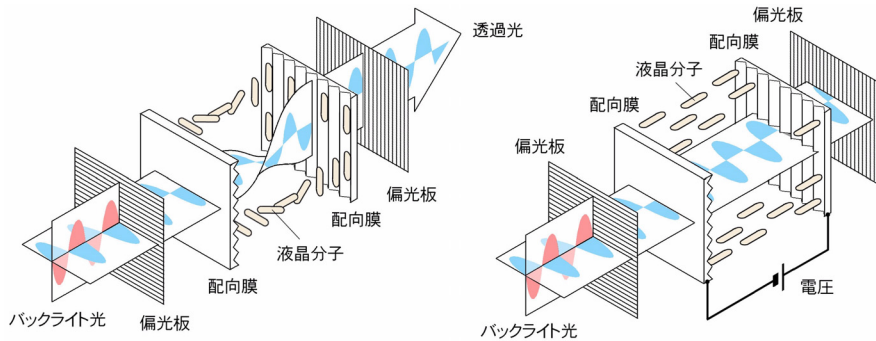


図 3.27 液晶ディスプレイ

図 3.27 は液晶ディスプレイの原理です。

2 枚の偏光板の間に液晶を置いてあります。液晶に電圧をかけなければ、バックライト光は液晶の中で 90° ねじれて偏光板を通過しますが、電圧をかけると直交している偏光板によって光が遮られます。液晶に電圧をかけるかかけないかによって光を ON-OFF するスイッチができます。1 画素毎にこのようにして光を ON-OFF すれば、ディスプレイとして使用することができます。これをカラー表示にするには各画素を更に RGB 3 色に分割し、各色ごとに個別に ON-OFF します。

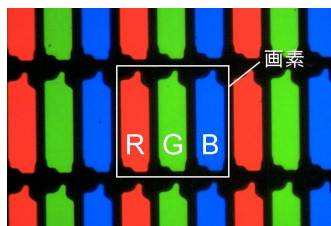


図 3.28 液晶ディスプレイのカラーフィルタの拡大写真

XGA ^{*27} サイズのディスプレイには $1024 \times 768 \approx 80$ 万個の画素がありますが、カラーディスプレイにはその 3 倍、240 万個の光のスイッチが必要です。これをガラス基板上に形成したトランジスタで行うものを、TFT (thin film transistor) 液晶といいます。10 数インチもの大きいガラス基板にトランジスタを作り込むには、高度な技術が要求されます。初期の

*27 「図 3.22 ディスプレイ画素数の種類」(68 ページ) 参照。

液晶ディスプレイは歩留まりが悪くて高価でしたが、生産技術の進歩とともに低価格化が進み、2001年には液晶ディスプレイの出荷台数がCRTディスプレイを上回るようになりました。

5.2. プリンタ

プリンタにはインパクト型と非インパクト型の2種類があります。

インパクト型はタイプライタと同じように、インクリボン越しに活字やワイヤーを用紙に打ちつけて印刷します。活字式のプリンタは文字はきれいですが、印刷できる文字の種類が少ないため、日本語の印刷には適していません。一方、文字の形を点の集合で表して印刷するドットインパクトプリンタはその制約がなく、図形の印刷もできることから一時普及しましたが、印刷速度が遅く、騒音も大きいため次第に敬遠されるようになりました。

現在主に使われているのは非インパクト型のプリンタです。

非インパクト型のプリンタには、感熱プリンタ、熱転写プリンタ、インクジェットプリンタ、レーザープリンタ、昇華型プリンタ、銀塩プリンタなど、さまざまなものがあります。

ここでは非インパクト型のレーザープリンタとインクジェットプリンタ、インパクト型のドットインパクトプリンタについて、そのしくみなどを見ていくことにします。

5.2.1. レーザプリンタ

レーザープリンタ (laser printer) は印刷品質が良く、静かで印刷速度も速いので、オフィスを中心に広く使われています。印刷方式はコピー機と同じ電子写真方式です。

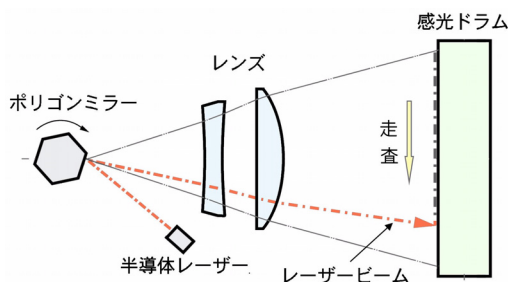


図 3.29 レーザプリンタのしくみ (光学系)

コピー機は原稿の像を感光ドラムに当てるのに対して、レーザープリンタはコンピュータの信号によって半導体レーザーの光を ON-OFF させ、これを感光ドラム上で走査することで画像を作ります。1 ページ単位で印刷データの処理が行われるので、ページプリンタ (page printer) ともいいます。

レーザービームは、まず回転しているポリゴンミラーに当たって反射し、ミラーの回転に応じて反射角度を変えながらレンズを経て感光ドラムに達します。感光ドラム上を図 3.29 では上から下に走査することになり、感光ドラムには点滅する光で線が引かれます。

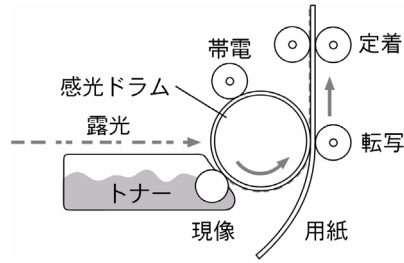


図 3.30 レーザプリンタのしくみ (印刷機構)

感光ドラムの表面は半導体できていて、あらかじめマイナスに帯電されています。光が当たると導電率が高くなって静電気がなくなり、感光ドラム上に目には見えない静電気の有無による画像が作られます。ここでマイナスに帯電させたトナー*28 (toner) を近づけると、光が当たらなかったところはマイナスの静電気同士が反発しあうためトナーが付きませんが、光が当たった部分にはトナーが付きまます。こうして感光ドラム上に、目に見える画像ができます。あとはこれを紙に転写すればいいのですが、ここでも静電気の性質を利用します。紙の裏から強いプラスの電圧をかけると、マイナスに帯電しているトナーはプラスの電気に引かれて紙の上に移ります。熱と圧力をかけてトナーを紙に定着させれば印刷完了です。

5.2.2. インクジェットプリンタ

コンピュータ周辺の技術は年々著しく進歩していますが、インクジェットプリンタ (ink jet printer) の技術の進歩もその最たるもののひとつです。

かつては高品位のカラー印刷には昇華型や銀塩方式といった特殊なプリンタが使われていましたが、高価でメンテナンスが複雑なため、とても個人ユーザーが使用できるものではありませんでした。一方その頃のインクジェットプリンタの印字品質は、お世辞にもいいといえませんでした。現在は写真と見分けがつかないほどのカラー印刷ができるようになりました。



図 3.31 インクジェットプリンタ

インクジェットプリンタの原理は、^{たと}喩えていえば水鉄砲にインクを詰めて文字や絵を描くようなものです。小さい文字や写真をきれいに描くにはインク滴はできるだけ小さい方がよく、

*28 炭素などでできた黒色の微粉末。

カラーで描くにはカラーインクを詰めた水鉄砲をいくつか用意すればいいわけです。シアン (cyan)、マゼンタ (magenta)、イエロー (yellow) の3原色の他に黒 (key plate) を加えて、CMYK の4色のインクで印刷します*29。

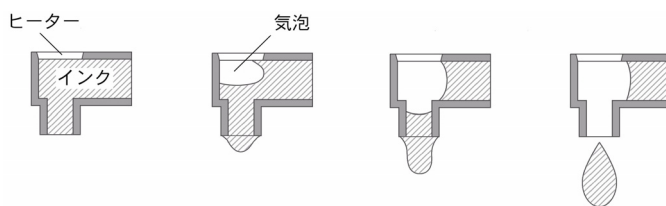


図 3.32 インクジェットプリンタの原理

インクジェットプリンタの印字品質は、いかにして小さいインク滴を正確に吐出するかにかかっています。図 3.32 はインクジェットプリンタのノズルです。ノズルにはヒーターが組み込まれています。ヒーターに電流を流して加熱すると、周辺のインクの温度が上がって気泡ができ、インクが押し出されます*30。インク滴の大きさはノズルの形状で決まりますから、常に一定量のインク滴を吐出することができます。インクジェットプリンタのインク滴の容量は数ピコリットル程度ですが、1pl は1l の1兆分の1にすぎません*31。

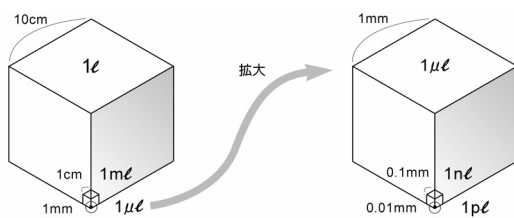


図 3.33 1pl とは？

1pl や1兆分の1といわれてもなかなかピンときませんが、1l といえば一辺が10cm の立方体の体積です。そして各辺がそれぞれ $\frac{1}{10}$ の1cm になると、体積は $\frac{1}{1000}$ の1ml になります。一辺がさらに $\frac{1}{10}$ の1mm になれば体積はまた $\frac{1}{1000}$ になって $1 \mu\text{l}$ (マイクロリットル)、さらに一辺が $\frac{1}{10}$ の0.1mm (たとえば髪の毛の太さくらいです) の立方体の体積はそのまた $\frac{1}{1000}$ の 1nl (ナノリットル) です。一辺がさらに $\frac{1}{10}$ の0.01mm (10 μm) という、台所のラップやアルミホイルの厚さがほぼこの程度です。このサイズの立方体の体積が、ようやく1pl、1兆分の1l です。

インクジェットプリンタは、こんなにも微量のインクを一滴一滴打ち出して、文字や画像を描いているのです。

インクジェットプリンタは、ごく微量の液体を正確に吐出することができる機械ですから、

*29 雑誌やパンフレットなどの一般のカラー印刷も通常はこの4色刷です。やや薄いシアンやマゼンタ、あるいは鮮やかな赤や緑などを加えて、5色や6色のインクで印刷するインクジェットプリンタもあります。

*30 電圧をかけると変形する「ピエゾ素子」を使ってインクを吐出させるプリンタもあります。

*31 $1 \text{pl} = 10^{-12} \text{l}$ 。なお、 $1 \mu\text{l}$ は 10^{-6}l 、 1nl は 10^{-9}l 。「数の接頭語」(226ページ) 参照。

たとえば導電性インクで電子回路のパターンを印刷してプリント基板を作ったり、液晶ディスプレイのカラーフィルターやDNAチップをインクジェットプリンタの技術を使って作るなど、エレクトロニクスやバイオテクノロジーへの応用も検討されています。



図 3.34 インクジェットプリンタによる賞味期限やロット No. の印字例

逆に、インク滴を十分大きくすれば、印刷対象物から多少離れていても、あるいは曲面にでもインクを吐出して印刷することができます。食品工場などでは、製品に賞味期限やロット No.などを印刷するのにインクジェットプリンタが使われています。インクジェットプリンタにはむしろ、こういう用途で生まれ、長い間使われてきたという歴史と実績があります。

5.2.3. ドットインパクトプリンタ

タイプライタがインクリボン越しに「活字」を用紙に打ち付けて文字を印刷するのに対して、ドットインパクトプリンタ (dot impact printer) は、「点 (dot)」の集合によって文字や図形を印刷するプリンタです。

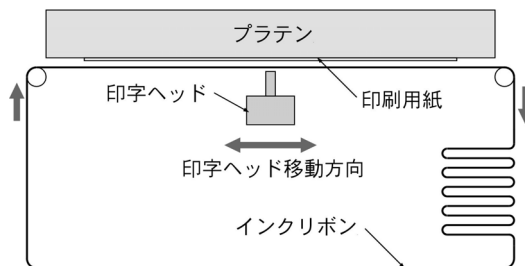


図 3.35 ドットインパクトプリンタ

構造はタイプライタにそっくりで、印刷用紙はプラテンと呼ばれるゴムのローラーで上下に動くようになっています。用紙の前にはループ状のインクリボンがあり、印字ヘッドは左右に動きながらドットワイヤを制御して文字や図形を印刷します。一行印刷が終わると、定められた角度だけプラテンが回転し、用紙を一行分だけ送ります。

図 3.36 はドットインパクトプリンタの印字ヘッドを表しています。12 個ずつ 2 列に並んでいるのはドットワイヤで、対応する端子に電流を流すと電磁石の働きでワイヤが飛び出して用紙に衝突し、ドットが印刷されます。

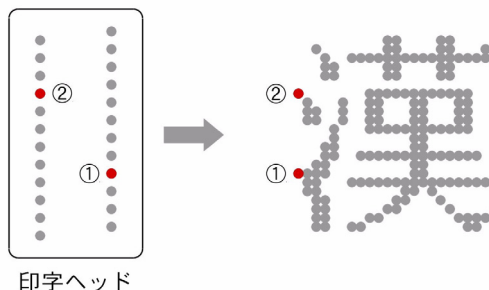
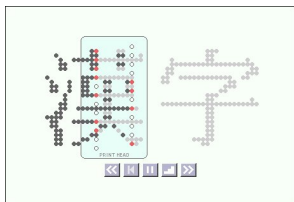


図 3.36 ドットインパクトプリンタの原理

いま、「漢」という文字を印刷しようとして印字ヘッドが右に動いているものとします。この場合、最初に印刷しなければならないのは「漢」の左端の2点、①と②ですが、印字ヘッドの1列目のワイヤがこの位置に来たとき、まず①のドットが印刷されます。②のワイヤはまだ文字の位置に達していませんから、印字ヘッドが更に右に動いて、2列目のワイヤが文字の位置に来たとき、電流を流して②のドットを印刷します。このようなヘッドの移動や印字ヘッドに電流を流すタイミングなどは、プリンタに内蔵されているマイクロコンピュータ*32 が制御しています。

ドットインパクトプリンタは、騒音が大きい、印字速度が遅いなどの欠点があり、プリンタの主流はレーザープリンタやインクジェットプリンタに移りました。しかしワイヤを物理的に打ち付けて印刷するという他の方式のプリンタにはない特徴があるため、複写帳票を連続用紙に印刷するオフィスなどでは現在も重宝されています。

Web で学ぶ 【ドットインパクトプリンタ】



Web ページ「ドットインパクトプリンタ」には、文字がドットによって印刷される様子を動的に表示する Java アプレットがあります。

交互に配置された2列のワイヤによって、「漢字」という文字が印刷される様子を見ることができます。

5.3. フォント

ディスプレイに文字を表示させるにしても、プリンタで文字を印刷するにしても、ディスプレイやプリンタはもちろん、コンピュータも、最初から文字の「形」を知っているわけではありません。

「B」という文字は ASCII では単に“42”というコードで表されますが*33、これは「ASCII

*32 「マイクロコンピュータ」(7ページ) 参照。

*33 「ASCII」(22ページ) 参照。

BBBBB BBBB BB B

の 66 番目の文字である」ということを表しているだけで、文字の形を表しているわけではありません。一口に「B」といっても、「B」にはさまざまなデザインの「B」がありますが、キャラクタコードではどれも“42”です。したがって「B」を実際に表示・印刷するには、「B」とはどんな形なのかというデータが別途必要です。もちろん B 以外にも、すべての文字の形状のデータが必要で、これをフォント (font) といいます。フォントは元来、同一の書体・サイズの大文字や小文字、数字などの欧文活字のセットの意味でしたが、最近ではコンピュータが表示・印刷に使う文字の形のデータセットもフォントと呼ぶようになりました。

フォントにはドットフォント (dot font) とアウトラインフォント (outline font) の 2 種類があります。

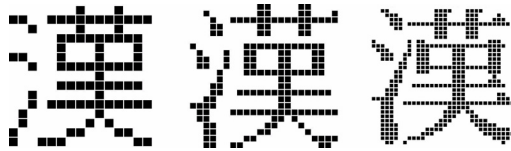
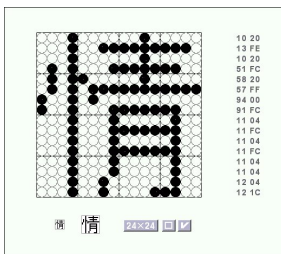


図 3.37 ドットフォント

図 3.37 はドットフォントの例で、文字をドットの集合で表します。表示や印刷時の処理が簡単なのでよく使われましたが、文字の形を美しく表そうとするとたくさんのドットが必要になる、文字の拡大・縮小や変形が困難、拡大すると斜線や曲線部のギザギザが目立つ、などの短所があります。

Web で学ぶ【ドットフォント】



Web ページ「ドットフォント」には、マウスでクリック／ドラッグすることによって簡単にドットフォントを作成できる Java アプレットがあります。作成されたドットフォントは即座に 16 進数のフォントデータに変換されます。ドットフォントのしくみが楽しく理解できます。

アウトラインフォントは文字通り、文字の輪郭 (outline) をデータとして記憶しています。図 3.38 はアウトラインフォントの例ですが、□印は直線または曲線の端点、○印は曲線の制御点です。複雑な文字でも、このような何本かの直線や曲線の組み合わせで表せますから、こ

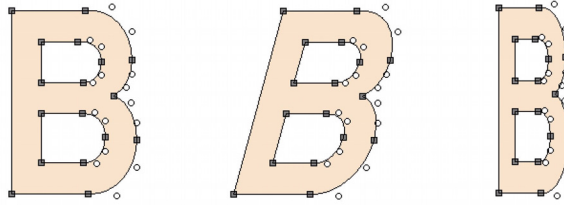
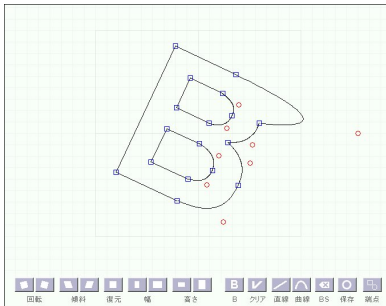


図 3.38 アウトラインフォント

これらの端点や制御点の座標値を記憶しておけばよいことになります。文字の形が座標値で与えられているので、計算によって拡大・縮小したり、変形することも自由自在です。図 3.38 では斜体、長体に変形させた例を示しています。

Web で学ぶ 【アウトラインフォント】



Web ページ「アウトラインフォント」には、アウトラインフォントのデータをマウスで編集できる Java アプレットがあります。曲線の形状を変えたり、傾斜や回転も思いのままです。新たにどんな文字でも作れるので、アウトラインフォントをゲームのように楽しみながら理解できます。

第4章 ソフトウェア

コンピュータは、猛烈な計算速度と巨大な記憶能力をもっている。しかし、世間が信じているのと反対に、それはまったくのばかであり、あらかじめ組み込まれた規則に、奴隷的に従っているにすぎない。コンピュータがプログラマーの助けを借りないでできる最高に知的なことといえば、空気調整なしに仕事をしろといわれたとき、ストライキを起すことぐらいのものである。

ペートル・ベックマン*1

現在のコンピュータは、ベックマンがこれを書いた時代のものであって、エアコンのない部屋で仕事をしろといわれても、ストライキを起こさないかもしれません*2。しかしそれはとにかく、間違いなく言えることは、コンピュータがプログラマーの助けを借りないでできる知的なことは、昔も今も、ひとつもないということです。

コンピュータは、どんなに高性能なものでも、プログラムがなければ動きません。

1. プログラミング言語

プログラムを作るには、プログラミング言語 (programming language) を使用します。

プログラミング言語は、プログラムを記述するために作られた人工的な言語で、機械語に対応した記号で記述する低水準 (低級) 言語と、人間の思考に適した概念と構造を持つ高水準 (高級) 言語の2種類があります。


ここで低水準・高水準とは、優劣をいうのではなく、コンピュータの機械語と人が使う自然言語のどちらに近いかを表しています。低水準言語は機械語に一对一に対応しているため木目の細かいプログラムを作ることができる反面、ハードウェアに関する詳しい知識が要求され、プログラムの生産性、保守性、移植性、汎用性に劣るという特徴があります。一方、高水準言語は自然言語に近い記法が取り入れられ、制御構文や関数なども豊富なため、効率よくプログラミングできます。現在では、ハードウェアを直接制御するプログラムのような特殊な場合を除いて、ほとんどのプログラムは高水準言語によって作られています。

*1 「 π の歴史 (A History of PI)」 田尾陽一・清水昭光訳 筑摩書房 (1973年)、ちくま学芸文庫 (2006年)。

*2 一定の規模以上になると、現在のコンピュータもエアコンが必要です。しかし、エアコンのない部屋で仕事をしろといわれたとき、ストライキを起こすのはまわりの人間の方が先かもしれません。

1.1. 低水準言語

コンピュータのメモリには、2進数の、機械語のプログラムが読み込まれています。コンピュータが理解できるのはこの機械語のプログラムですが、人が直接2進数で機械語のプログラムを書くのは困難なため、これを記号に置き換えたアセンブリ言語 (assembly language) が開発されました。このような、機械語と一対一で対応しているプログラミング言語を低水準言語 (low-level language) といいます。

<pre>00000001 00111111 10011111 11010001 00000011</pre>		<pre>CLR ; 0 -> Acc LOOP: ADD 15 ; M(15) + Acc -> Acc DEC_M 15 ; M(15) - 1 -> M(15) JNZ LOOP ; if M(15) <> 0, jump to LOOP OUT ; Acc -> Output Port</pre>
機械語		アセンブリ言語

上の例は第2章の「5.3.2. アセンブリ言語」(48ページ)のプログラムと同じものです。機械語に比べれば分かりやすくなっていますが、アセンブリ言語でプログラムを書くにしても、ハードウェアを熟知している必要があり、どのレジスタを使うか、プログラムやデータをメモリの何番地に格納するか、といったことを考慮しなければなりません。同じプログラムを別のコンピュータ (CPU) で動かすためには、そのコンピュータ用のアセンブリ言語で書き直さなくてはならないという不便もあります。

アセンブリ言語で書かれたプログラムも、これを実行するには事前に機械語 (2進数のプログラム) に変換しなくてはなりません*3。コンピュータが直接理解できるのは、2進数の機械語プログラムだけだからです。

1.2. 高水準言語

次の例は高水準言語 (high-level language) のひとつ、C言語を使って書いた、円の面積を計算して表示するプログラムの一部です。

```
double s; // 変数 s (面積) の宣言
double r = 5.0; // 変数 r (半径) の宣言と値の代入
s = 3.14 * r * r; // 面積の計算
printf("面積 = %7.2lf", s); // 面積の表示
```

最後の行の“%7.2lf”などはまるで暗号ですが、これはプログラムを簡潔に書くために工夫されたC言語特有の書式です。面積 *s* や半径 *r* のように、変数は名前を付けて宣言するだけでいいので、プログラマはメモリ管理の気苦労からも解放されます。

高水準言語はCPUに依存せず、複雑な処理や演算を日常言語に近い表現や数式の形で簡単に記述できるなど、すぐれた特徴を持っています。

*3 アセンブリ言語で書かれたプログラムを機械語に変換することを、アSEMBルといいます。

高水準言語で作ったプログラムも、実行するに当たって機械語に変換しなくてはなりません。これにはインタープリタ (interpreter) とコンパイラ (compiler) という、ふたつの方式があります。前者はその都度プログラムを一行ずつ変換しながら実行していくのに対して、後者はあらかじめ一括して変換しておいたものを実行します。インタープリタはコンピュータとの対話性に優れているのでプログラムの作成が容易ですが、一行ずつ変換しながら実行するため実行速度が遅いのにに対して、コンパイラは実行速度が速いという特徴があります。

一般に市販されて利用されているソフトウェアは、ほとんどがコンパイラで作られています。

1.2.1. 主な高水準言語

1. ^{フォートラン}FORTRAN

formula translator の略で、1956年にIBM社によって作られた最初のプログラミング言語で、科学技術計算に向いています。

2. ^{コボル}COBOL

common business oriented language の略で、1959年に米国防省が政府・民間の関係者を集めて設立した“CODASYL” (conference on data system language : データシステム言語協議会) によって作られた、事務処理向きの言語です。

3. ^{ベーシック}BASIC

beginner's all purpose symbolic instruction code の略で、1965年にダートマス大学のケムニ (John G. Kemeny) とクルツ (Thomas E. Kurtz) によって作られました。FORTRANの流れを汲んだ、初心者向きのインタープリタ言語です。1975年にはビル・ゲーツ (Bill Gates) らによってパーソナルコンピュータ用に手直しされ、パーソナルコンピュータの標準言語として普及しました。現在は改良版の“Visual BASIC”が使われています。

4. ^{パスカル}PASCAL

世界で初めて機械式の計算機を作ったパスカル*4 にちなんで名づけられたもので、1971年にチューリッヒの連邦工科大学のヴィルト (Niklaus Wirth) らによって作られました。部品になるプログラムを作り、それらを組み合わせて全体のプログラムを作る、「構造化プログラミング」という方法を用いているので、効率的にプログラミングできます。

5. ^CC

1972年、米国のベル研究所で開発された言語で、これも構造化プログラミング指向です。現在最も広く一般的に使用されているプログラミング言語で、最近ではオブジェクト指向*5 を取り入れた“C++”が使われています。

*4 「パスカルの計算機」(106ページ) 参照。

*5 データと、それを扱う手続きをすべてオブジェクト (object) とみなし、オブジェクトの組み合わせによって記述していくプログラミング技法。

6. Perl^{パール}

ラリー・ウォール (Larry Wall) が 1987 年に開発したインタプリタ方式の言語です。小規模のプログラムを簡単に記述するためのプログラミング言語をスクリプト言語*6 (script language) といいますが、Perl もそのひとつです。ウェブページの CGI (common gateway interface) のプログラムを作る場合などによく使われています。

7. Java^{ジャバ}

サン・マイクロシステムズ社が 1995 年に開発したプログラミング言語です。オブジェクト指向とセキュリティ機構を持っていて、ネットワーク環境で利用されることを前提にして設計されています。

Java で作られたソフトウェアはどんなコンピュータでも動作するという特長があり、インターネットのウェブページに組み込まれている Java で書かれたソフトウェアをアプレット (applet) といいます。

以上、代表的な高水準言語について簡単に紹介しましたが、最も古いものは 1956 年に作られた FORTRAN、次いで 1959 年の COBOL です。現在では FORTRAN や COBOL で新たにコンピュータシステムを構築することはまずありませんが、これまでにこれらのプログラミング言語で作られたシステムは現在も使われていますから、FORTRAN や COBOL 自身も現役です。コンピュータのハードウェアは 10 年もたてばすっかり陳腐化するのに対して、プログラミング言語の寿命の長さは驚きです。

なお、現在、基本情報技術者試験でプログラミング能力を問う問題は、C、COBOL、Java 及びアセンブリ言語の 4 言語、テクニカルエンジニア (情報セキュリティ) 試験では C++、Java、Perl の中からひとつを選択する形で出題されています。

*6 Perl の他には PHP、JavaScript、Ruby、VBScript などがあります。

2. プログラム

プログラムとは、コンピュータがある仕事を遂行するために必要な宣言や文、命令などで構成されたもので、プログラミング言語を使って記述されます。

コンピュータやプログラムを理解するには、実際にプログラムを作ってコンピュータを動かしてみるのがいちばんの近道です。ここではパーソナルコンピュータで、誰でも手軽に試すことができる VBA を使って、プログラムの具体的な説明をしていきます。

2.1. VBA

VBA は Visual Basic for Applications の略で、Visual Basic*7 はコンピュータのプログラミング言語、Application はアプリケーションプログラムを指しています。ワード (Word) やエクセル (Excel) もアプリケーションプログラムのひとつで、VBA はこれらのアプリケーションに対して指示を与えるプログラム、マクロ*8 (macro) を記述するツールです。

VBA はたいていのパーソナルコンピュータにインストールされていますから、誰でも試すことができます。ここではエクセルのマクロ機能を使って作る簡単なプログラムについて説明します。VBA の起動の仕方や使い方等については、付録「エクセルのマクロと VBA」(229 ページ) を参考にしてください。

2.1.1. メッセージの表示

まず、最も簡単な、メッセージを表示するプログラム (マクロ) を紹介します。

```
Sub hello()
  MsgBox "Hello !!"
End Sub
```

VBA のプログラムは **Sub** **マクロ名** **()** で始まり、**End Sub** で終わります。1 行目の **hello** はマクロにつけた名前です。名前ですから、**test** でも **message** でも、原則として何でも構いません*9。また、**こんにちは** などのように、日本語も使えます*10。

2 行目の **MsgBox "Hello !!"** が本来のプログラムです。ここで **MsgBox** はメッセージボックスを表示させる命令で、VBA で「関数」と呼ばれるものの一つです。**Hello !!** はメッセージボックスに表示させる「メッセージ」です。したがってこれも、**こんにちは** や **パスワードが間違っています** など、どんな文字列でも、あるいは数字でも構いません。このプログラムを実行させると、図 4.1 のようなメッセージボックスが表示されます。

*7 「Basic」(81 ページ) 参照。

*8 ワープロや表計算ソフトなどで、操作手順を記録またはプログラムとして記述して複数の操作を自動化する機能。

*9 Sub や End、MsgBox などのように、VBA 自身が使用している語はマクロ名や変数名には使えません。これらを予約語といいます。

*10 VBA では日本語のマクロ名や変数名を使えますが、多くのプログラミング言語では日本語は使用できません。



図 4.1 “hello” の実行結果

2.1.2. 変数

変数 (variable) とは、データを記憶するメモリに名前をつけたものです。

```
A = 3  
Name = "山田"
```

これは変数 (A や Name) に値を代入するプログラム例で、A という変数の値は 3 という数字、Name という変数の値は 山田 という文字列です

```
Sub area()  
R = 5  
Pi = 3.14  
S = Pi * R * R  
MsgBox S  
End Sub
```

変数を使ったプログラム例です。R に半径 5、Pi に円周率 3.14 を代入しておき、円の面積 S を計算して表示しています。次のように、マクロ名や変数名を日本語にしても構いません。

```
Sub 面積計算()  
半径 = 5  
円周率 = 3.14  
面積 = 円周率 * 半径 * 半径  
MsgBox 面積  
End Sub
```

実行するとメッセージボックスに計算された面積が表示されます。



図 4.2 「面積計算」の実行結果

もうひとつ、今度は変数に文字列を代入するプログラムを作ってみます。

```
Sub myname()
  姓 = "山田"
  名 = "太郎"
  MsgBox "私は " + 姓 + 名 + " です。"
End Sub
```

姓 という変数の値は 山田、名 という変数の値は 太郎 です。これを実行すると図 4.3 のように表示されます。



図 4.3 “myname” の実行結果

2.1.3. オブジェクト、プロパティ、メソッド

VBA ではエクセルのシートやセルなど、エクセルの部品をオブジェクト (object) といいます。また、オブジェクトの属性をプロパティ (property)、オブジェクトに対する操作をメソッド (method) といいます。

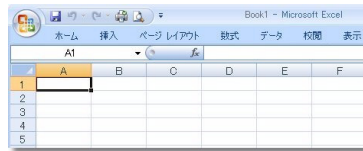


図 4.4 A1 セル

たとえば、`Cells(1, 1).Value` *¹¹ と書くと、`Cells(1, 1)` が「オブジェクト」で、エクセルワークシートの 1 行 1 列目のセル、すなわち A1 セルを表しています。`Value` は「プロパティ」で、セルの属性、この場合はセルの値のことで、`Cells(1, 1).Value` は「A1 セルの値」という意味です。

`Cells(1, 1).Activate` と書いた場合、`Activate` が「メソッド」です。`Activate` はアクティブにするという操作を表し、A1 セルをクリックした状態にします。

```
Sub 小遣い帳 ()
  Cells(1, 1).Value = "チーズケーキ"
  Cells(1, 2).Value = 1180
  Cells(2, 2).Activate
End Sub
```

`Cells(1, 1).Value = "チーズケーキ"` の `Cells(1, 1)` はオブジェクトで A1 セルのこ

*¹¹ `Cells(row, column)` は行・列の順でセルを指定します。通常セルを A1 セル、B3 セルなど、列・行の順で呼ぶのとは逆ですから注意してください。

と、Value はプロパティ、その値です。A1セルに **チーズケーキ** という文字列を代入しています。

Cells(1, 2).Value = 1180 も同じで、こちらの値は **1180** という数値です。

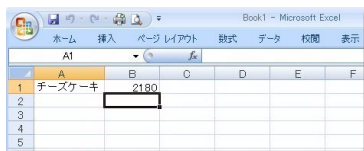


図 4.5 「小遣い帳」の実行結果

これを実行すると、セル A1 には **チーズケーキ**、B2 には **1180** が入力され、アクティブなセルは B2 に変わります。

2.1.4. 条件分岐

コンピュータが「賢い」ように見える*12 理由のひとつが、この条件分岐 (conditional branch) にあります。コンピュータのプログラムは、データの値や演算結果によって、処理を変えることができます。

条件分岐もいろいろな書き方がありますが、よく使われるのが、

```
If ~ Then ~ Else ~ End If
```

です。「もしお天気だったら公園に行き、そうでなければ映画でも見よう」という構文に似ています (Else ~ は省略できます)。

```
Sub 条件分岐 ()
    点数 = Cells(1, 1).Value
    If 点数 < 60 Then
        判定 = "不合格"
    Else
        判定 = "合格"
    End If
    Cells(1, 2).Value = 判定
End Sub
```

これは「もし 60 点未満なら不合格、そうでなければ合格だよ」というプログラムです。実行すると、A1セルの値に応じて、A2セルに **合格** または **不合格** という判定が書き込まれます。まず、`点数 = Cells(1, 1).value` で A1セルの値が `点数` という変数に代入され、`If ~ Then ~ Else ~` という構文のプログラムによって、`判定` という変数の値が、**合格** になったり **不合格** になったりします。

*12 ベックマンも言っているように (79 ページ)、コンピュータは「まったくのばか」ですが。

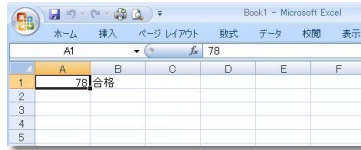


図 4.6 「条件分岐」の実行結果

2.1.5. 多重分岐

先ほどは「もしお天気だったら公園に行って、でなければ映画でも見よう」という例を使いましたが、天気は晴れでなければ雨、というわけではありません。曇りや雪だったらどうするか。そういう場合には `ElseIf` を使います。

```
Sub 多重分岐 ()
    点数 = Cells(1, 2).Value
    If 点数 < 60 Then
        判定 = "不合格"
    ElseIf 点数 < 70 Then
        判定 = "可"
    ElseIf 点数 < 80 Then
        判定 = "良"
    Else
        判定 = "優"
    End If
    Cells(1, 2).Value = 判定
End Sub
```

これを実行すると図 4.7 のようになります。もちろん、A1 セルの値が 83 であれば優に、61 であれば可に、43 であれば不可になります。

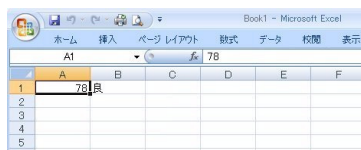


図 4.7 「多重分岐」の実行結果

2.1.6. 繰り返し

繰り返し処理はコンピュータの得意技です。

ヒトは同じことを何度も繰り返すとすぐに飽き、疲れ、眠くなってしまうますが、コンピュータは疲れを知りません。しかも超高速です。何千回、何万回と繰り返しても、ほとんど一瞬で終わってしまいます。

繰り返しにも色々な方法がありますが、VBA でよく使われるのは `For ~ Next` です。

```

Sub 繰り返し ()
SUM = 0
For N = 1 To 10
    SUM = SUM + N
Next N
Cells(1, 1).value = SUM
End Sub

```

これは1から10までの整数を加算するプログラムです。

3行目の For N = 1 To 10 によって、N は1から10になるまで、ひとつずつ値を大きくしながら Next N までの処理、ここでは SUM = SUM + N を繰り返します。SUM の値は最初は0で、SUM = SUM + N の度に1、2、3、4と増えていくNの値を加えていきます。最終的にSUMの値は55 (= 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10) となります。

SUM = SUM + N は数学的には矛盾していますが、コンピュータプログラムでは「変数 SUM の値を SUM + N に置き換える」という意味です。

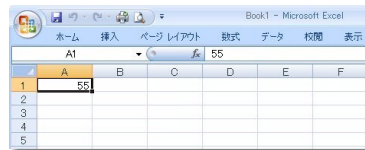


図 4.8 「繰り返し」の実行結果

2.1.7. 繰り返しと多重分岐

繰り返し処理の方法が分かりましたので、先ほどの多重分岐を組み合わせてみます。

```

Sub 成績判定 ()
For Row = 1 To 12
    点数 = Cells(Row, 2).Value
    If 点数 < 60 Then
        判定 = "不合格"
    ElseIf 点数 < 70 Then
        判定 = "可"
    ElseIf 点数 < 80 Then
        判定 = "良"
    Else
        判定 = "優"
    End If
    Cells(Row, 3).Value = 判定
Next Row
End Sub

```

For Row = 1 To 12 …… Next Row によって、変数 Row が1から12になるまで、処理が繰り返されます。Row は、ここではエクセルのワークシートの行を表わしています。

`点数 = Cells(Row, 2).Value` は、セルの値を **点数** という変数に代入する命令で、行 (**Row**) と列 (**2**) の値でセルを指定します。**Row** は最初 **1** ですから、1 行目の 2 列目、すなわち B1 セルを指しています。処理が進んで **Row** の値が増えるに従って、B2、B3 と変わっていきます。

	A	B	C	D	E	F
1	鶴野	62	可			
2	憐部	72	良			
3	宮澤	92	優			
4	細河	88	優			
5	羽田	64	可			
6	卯山	77	良			
7	橋元	73	良			
8	小淵	42	不可			
9	杜	56	不可			
10	古泉	60	可			
11	阿倍	70	良			
12	副田	67	可			
13						
14						

図 4.9 「成績判定」の実行結果

マクロを実行する前に、セル B1 から B12 まで、0~100 の間の点数を入力しておきます。

実行すると、図 4.9 のように、C1 から C12 に「優」「良」「可」「不可」が書き込まれます。A 列には名前を書き込んでおくと、より一層成績表らしく見えます。

エクセルのマクロのプログラムをいくつか紹介しましたが、これらはほんの一例にすぎません。VBA を使えばもっともっといろいろなことができます。参考になる書籍やウェブサイトもたくさんありますから、ぜひ本格的なプログラム作りを試みてください。

3. オペレーティングシステム (OS)

ごく初期の、単純なコンピュータを除けば、コンピュータのソフトウェアはオペレーティングシステム (OS : operating system) とアプリケーションプログラム (AP : application program) とで構成されています。

AP はワープロソフト、ウェブブラウザなどの、実際の仕事のプログラムです。

OS はこの AP とハードウェアとを橋渡しして繋ぐプログラムで、「基本ソフト」とも呼ばれています。

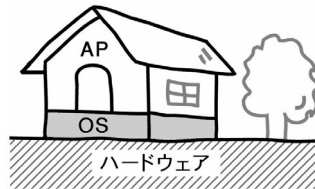


図 4.10 OS と AP

図 4.10 のように、建物を建てる場合、地面の上にまず基礎を作り、その上に建物の本体を作ります。この地面に当たるのがハードウェア、基礎に当たるのが OS、建物本体に当たるのが AP、と考えればいいでしょうか。

3.1. OS とは

AP が仕事そのものをするためのプログラムであるのに対して、OS は、荒っぽい言い方をすれば、ハードウェアを動かすためのプログラムです。



図 4.11 OS (Windows) の画面例

パーソナルコンピュータは、電源スイッチを入れると自動的に OS が起動するようになっています。図 4.11 は OS (Windows) の準備が完了したときの画面の一例です。

この状態のとき、OS は、主にマウスの操作 (入力) を待っています。画面左下の「スタートボタン」がクリックされると利用できるプログラムのメニューが表示され、デスクトップのアイコン (icon) がダブルクリックされると対応するソフトウェアが起動^{*13} されます。

図 4.12 はワープロソフト、ワードを起動して「あ」を入力したところです。

*13 起動とは、プログラムを主記憶装置に読み込んで、これを実行させることをいいます。

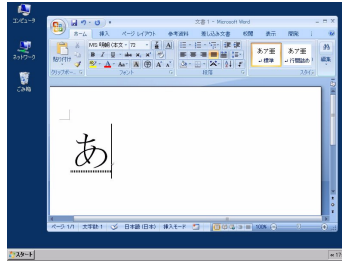


図 4.12 ワードプロソフト

ハードウェアである「マウス」をクリックすると、指示された AP（ここではワード）のプログラムが、ハードウェアである「ハードディスク」から同じくハードウェアである「メモリ」に読み込まれます。そしてプログラムが実行されて、やはりハードウェアである「ディスプレイ」に表示されます。ハードウェアを動かすためのプログラム、OS が、マウス操作の指示通りに AP をメモリに読み込んで実行し、それをディスプレイに表示する処理を行ったのです。

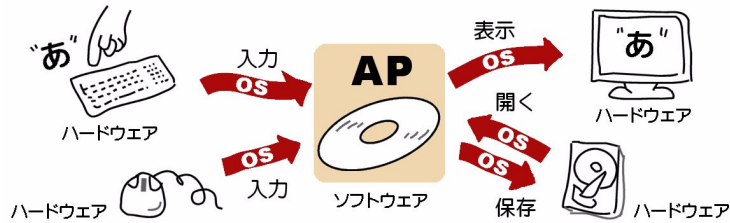


図 4.13 OS は AP とハードウェアを橋渡しするプログラム

図 4.12 には、「あ」という文字が入力されています。これもまず、ハードウェアである「キーボード」で押されたキー、「あ」に相当するコードを OS が AP（ワードプロソフト）に伝えます。AP はそれを受けて、OS に「これこれの位置に「あ」という文字を表示して下さい」と要請します。OS はそれに従って、指定された位置に「あ」という文字を表示するのです。

繰り返すようですが、OS は AP とハードウェアとを橋渡しして繋ぐプログラムです。

3.2. OS の起源

かつては単に「コンピュータ」といえば、大型汎用コンピュータのことを意味していました。大型汎用コンピュータは非常に高価でしたから、コンピュータを導入した企業や役所などでは、1 台のコンピュータですべての仕事进行处理しなければなりませんでした。

1 台の大型汎用コンピュータで、内容の異なる種々の仕事を、複数のユーザーのために実行するには、それらの仕事 (job) をできるだけ効率よく、時間的なロスも少なくなるように割り振らなくてはなりません。

しかし、この割り振りを人間が行ったのでは充分なことではできません。そこで、これもコンピュータに行わせようということが考えられました。まず行わせるべき一連の仕事の名前や、

それに必要な装置やデータファイルなどを一定の様式で記述します。これをジョブ制御言語 (JCL: job control language) といいます。この、ジョブ制御言語を解釈して、すべて自動的に処理されるようにしたプログラムが、大型汎用コンピュータにおける OS の始まりでした。高価なコンピュータを、効率よく使用するために生まれました。

また、ハードディスクのような補助記憶装置に、プログラムはもちろんデータも保存して随時読み書きするような使い方をするには、まずディスクをフォーマットしたり、データを保存したり開いたり、不要なものは削除したりするための複雑なプログラムが必要です。他にも、キーボードから文字データを読み込んだり、それを画面に表示したり、あるいは通信回線とのデータの入出力など、周辺装置に関する様々なプログラムも必要です。

しかしこれらのプログラムはあらゆる AP から共通に使用できる性格のもので、ハードウェアを動かすこれらのプログラムをまとめて、OS として独立させるようになりました。

3.3. OS の機能の進化

コンピュータのハードウェアが年々進歩するに伴って、OS の機能も進化してきました。

コンピュータのハードウェアとソフトウェアは車の両輪のようなもので、ハードウェアが進歩しても、ソフトウェアが追従しなければコンピュータの総合的な性能は向上しません。こうして OS にも、次々と新しい機能がつけ加えられていきました。

3.3.1. ネットワーク機能

コンピュータを単独で使う状態をスタンドアローン (stand alone) といいます。

パーソナルコンピュータは、かつてはスタンドアローンで使われましたが、LAN *14 (構内情報通信網: local area network) の技術が発達してくると、パーソナルコンピュータも LAN のメンバーに組み入れられることが多くなりました。パーソナルコンピュータが普及した現在では、家庭でも何台かのコンピュータを LAN で接続することが珍しくなくなってきました。

OS には、コンピュータを通信回線で結んでネットワークを構成するための、ネットワークの機能が加えられています。

3.3.2. GUI

初期のコンピュータの操作はキーボードで行われました。コンピュータを操作するにはコマンドをタイプし、プログラムを実行するにはプログラム名を正確に入力しなければなりません。コンピュータを扱うには、ハードウェアやソフトウェアに関するかなりの知識が要求され、誰にもできるというものではありませんでした。

これに対して、アイコン (icon) をディスプレイに表示し、これをマウスでクリックすることによってコンピュータを操作する GUI (graphical user interface) は、誰にも直感的に、容易に操作することができます。GUI の機能を持った OS が普及した現在では、小学生からお年

*14 「LAN」(145 ページ) 参照。

寄りまで、誰でもコンピュータを扱うことができるようになりました。

3.3.3. マルチタスク

図 4.14 は、表計算ソフト（エクセル）で作ったグラフを、ワープロソフト（ワード）で作成中の文書に貼り付けたところです。

表計算ソフトでグラフを「コピー」し、ワープロソフトの文書に「貼り付け」ればいいので、この例の他にもペイントソフトで描いた絵、デジタルカメラで撮った写真などを含む文書を簡単に作ることができます。

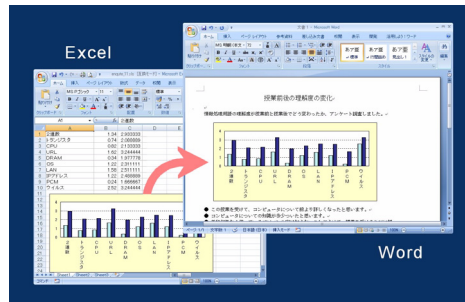


図 4.14 マルチタスク

こういうことができるのは、表計算ソフトとワープロソフトが「同時」に動いているからです。複数のアプリケーションプログラムが同時に実行されることをマルチタスク (multi task) といいます。

かつての OS はシングルタスク (single task)、ただひとつのプログラムだけしか動かすことができませんでした。同じことをシングルタスクの OS でしようと思えば大変です。まず表計算ソフトでグラフを作って保存し、ひとまず表計算ソフトを終了させます。つぎにワープロソフトでそれを読み込み、グラフ以外の不要な部分を削除して文書データを作っていきます。もし途中で「折れ線グラフに替えたい」と思ったら、ワープロソフトは終了して表計算ソフトに戻って…。とても厄介なことになりそうです。

マルチタスクは、コンピュータの使い勝手を飛躍的に高めました。

ウェブページで見つけた資料をワープロで作成しているレポートに引用する。メールが届けばすぐに返事を書いて送信し、お気に入りの CD を聴きながらレポートを仕上げる。こんなことが簡単にできるのも、すべてマルチタスク機能のおかげです。

3.3.4. 仮想記憶

コンピュータは、主記憶装置（メモリ：main memory）に読み込まれているプログラムを実行します。中央処理装置（CPU）は、メモリに記憶されているプログラムをひとつずつ読み出しては実行し、処理結果をまたメモリに書き込みます。CPU とメモリとの間で、プログラムやデータがたえずやりとりされながらコンピュータの処理が進みます。

したがって、マルチタスクを十分に活用するには、複数のプログラムとそれぞれが処理するデータ等がすべてメモリに読み込まれていなくてはなりません。静止画や音声、動画の処理をしようとするとデータ量は膨大で、メモリはいくらあっても足りません。といってメモリは高価ですから、おいそれと記憶容量を増やすこともできません*15。

そこで、仮想記憶 (virtual memory) が考え出されました。

マルチタスクで複数のプログラムが実行されているといっても、必ずしもすべてのプログラムが「同時に」動いているわけではありません。先の例でいえば、グラフを作っているときに動いているのは表計算ソフトでワープロは休んでいますし、ワープロに貼り付けているときは表計算ソフトがひと休みしています。

ここで更に別のソフトのプログラムを読み込むのにメモリが足りないという状況が起これば、表計算ソフトがどういう状態で休んでいるかを記録しておいて、そこに新しいプログラムを読み込んでもいいわけです。

プログラムばかりでなく、データについても同じことがいえます。画像処理のソフトなどでは大容量のデータを扱いますが、処理が終わって写真をワープロソフトに貼り付けてしまえば、画像ソフトで使われていたメモリ領域は空けることができます。

このように、プログラムやデータをハードディスクに一時的に保存してメモリを空ける、あるいは逆にハードディスクからメモリに復帰させることをスワップ (swap) といいます。ハードディスクに空きスペースがある限り、ほぼ無尽蔵にメモリがあるかのようにコンピュータを使うことができます。

こういうといいことづくめのようなのですが、残念ながらハードディスクは DRAM に比べるとデータ転送速度が圧倒的に遅いので、スワップが増えるとコンピュータの処理速度が目に見えて遅くなります*16。

大容量の画像データを大量に編集していると、マウスをクリックする度にハードディスクのアクセスランプが点きつきりになって、画像が切り替わるのにうんと時間がかかるようになることがあります。このような状態がしばしば起こるようであれば、メモリの容量に対してコンピュータの使い方が過酷すぎるということです。メモリを増設して記憶容量を増やすと、これはすぐに解決します。仮想記憶は、処理速度よりもメモリの確保、言い換えればコンピュータが動き続けること、を優先して実現されています。

*15 記憶容量当たりの単価を比較すると、DRAM はハードディスクの 200 倍程度の価格になります。

*16 DRAM とハードディスクの速度を単純に比較することはできませんが、DRAM の方が 100~1,000 倍くらい速いと考えればいいでしょう。

4. アプリケーションプログラム

4.1. アプリケーションプログラムの種類

大型コンピュータの場合は利用者自身、あるいは依頼された業者がプログラムを作ることが多いのに対して、パーソナルコンピュータではもっぱらパッケージソフトが使われています。

パーソナルコンピュータも初期の頃は、利用者自身がアセンブリ言語や BASIC *17 を用いて自分でプログラを作っていました。しかし、簡単なサンプルプログラム程度のものならともかく、実用に耐えるプログラムとなると一般の人にはハードルが高く、そのため「パーソナルコンピュータは難しい」とされていました。しかしやがてパッケージソフトが登場し、パーソナルコンピュータの普及を促しました。今日ではパーソナルコンピュータを使いこなすということは、プログラムを作ることではなく、これらのレディーメイドのプログラムをうまく使いこなすことになりました。

レディーメイドのプログラムには、市販されているパッケージソフトの他に、インターネットなどによってダウンロードするオンラインソフトもあります。

オンラインソフトには有料のシェアウェア (shareware) と無料のフリーウェア (free software) があります。一般のパッケージソフトは開封すると返品できないことが多いのに対して、シェアウェアは一定期間 (たとえば1ヶ月) 試用した後、引き続き使用したければ料金を支払うものが多いので、類似のソフトウェアをいくつか試して、気に入ったものを選べるというメリットがあります。価格もたいていは安く設定されています。

フリーウェアは文字通り無料です。「タダのソフトなんか大丈夫？」と心配する人もいますが、ソフトウェアの質や機能は必ずしも金額に比例するわけではありません。フリーウェアやシェアウェアには永年多くの人々に使われて定評のあるものも多く、逆に市販のソフトウェアにも使い勝手が悪い、機能が乏しいなど、感心しないものが少なくありません。

アプリケーションプログラム (AP) には次のような、様々な分野のものがあります。

1. ビジネス
2. 教育・学習
3. 科学・技術
4. 趣味・家庭・生活
5. 画像・音楽
6. インターネット・通信
7. ゲーム・アミューズメント
8. ユーティリティ・プログラミング

以下、これらのうちから、主なものをいくつかを紹介します。

*17 「アセンブリ言語」(80 ページ)、「BASIC」(81 ページ) 参照。

4.2. 主なアプリケーションプログラム

4.2.1. ワードプロソフト

欧米には古くからタイプライタを使う習慣がありましたが、日本語は文字が多いため専ら手書きでした。和文タイプもあるにはありましたが、何千という活字がセットされた文字盤から文字を探しだして印字する構造のため熟練が必要で、公文書の作成などに用いられる程度でした。

コンピュータ技術が発達してくると、欧米ではいち早くワードプロソフト (word processor software) が登場しました。単なる清書機であるタイプライタと違って、ワードプロソフトには文書の編集ができるという利点がありますが、日本ではやはり漢字の壁に阻まれて、なかなか実用化されませんでした。

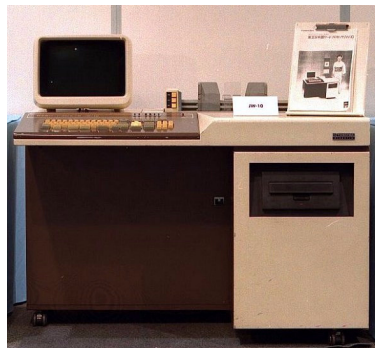


図 4.15 日本初のワードプロセッサ (JW-10)

出典：(社) 情報処理学会「コンピュータ博物館」(<http://museum.ipsj.or.jp/>)

日本語を処理できるワードプロセッサは、1978年に初めて東芝が開発しました。図 4.15 は日本初のワードプロセッサ JW-10 *¹⁸ です。発売当時の価格は 630 万円でした。

1980年代になると価格も半分程度にまで下がって大手企業に導入され始め、その後も低価格化、小型化が進み、やがて個人用ワープロ専用機として各社が様々な機種を発売するに至りました*¹⁹。しかし 1990年代後半にはパーソナルコンピュータやワードプロソフトが普及し、ワープロ専用機の製造は 2001年までに相次いで中止されました。

ワードプロソフトは、専用機であるワードプロセッサの行う仕事を、汎用機であるコンピュータで行うことができるようにしたソフトウェアです。代表的なものにはマイクロソフト社のワード、ジャストシステム社の一太郎などがあります。

日本語のワードプロソフトは、キーボードから入力された仮名もしくはローマ字を漢字に変換する日本語入力システム (IME : input method editor) の部分と、文章となったものを編集・校正・保管・記憶・印刷する、ワープロ本体の機能の部分とからなっています。

*¹⁸ 文節指定入力によるかな漢字自動変換方式。文書記憶容量 200 ページ (磁気ディスク 10MB)。

*¹⁹ たとえば Rupo (東芝)、OASYS (富士通)、書院 (シャープ)、文豪 (日本電気) など。

IME *20 はワープロソフト以外の AP でも使いますから、常にメモリに読み込まれていて、他のプログラムからいつでも共通に利用できるようになっています。Windows にはマイクロソフトの MS-IME が標準で付属していますが、変換効率の良さと定評のあるジャストシステムの^{エイトック}ATOK を愛用する人も少なくありません。

4.2.2. 表計算ソフト

事務処理では、表を作ってタテ・ヨコに合計などを計算することがよくあります。表計算ソフト (spreadsheet software) は、このような業務を行うためのソフトウェアです。

1979 年にソフトウェア・アーツ社から、Apple II *21 で動くビジカルク (VisiCalc) という表計算ソフトが発売されましたが、これがパーソナルコンピュータの最初のビジネス用パッケージソフトでした。現在はマイクロソフトのエクセルが代表的です。

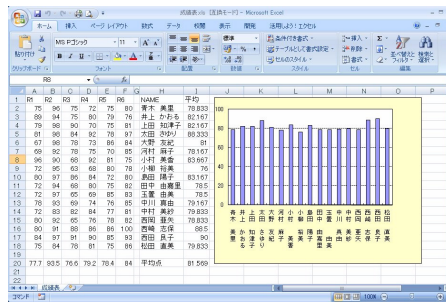


図 4.16 表計算ソフト (エクセル)

表計算ソフトは、ディスプレイに 1 枚の紙を広げたような感覚で作業するため、スプレッドシート (spread sheet) とも呼ばれています。単に計算するばかりでなく、ソート*22 (sort) やマージ*23 (merge)、あるいはその結果を様々な形式のグラフにする機能などをもっています。

「情報処理」とは、素材となる情報 (データ) を処理加工して利用する者のニーズに応じた形に変換すること*24 ですが、表計算ソフトはそのための、手軽で強力なツールです。

4.2.3. データベース

データベース (data base) とは、大量の情報をコンピュータの記憶装置に蓄積し、必要に応じて条件に合うものを検索して取り出せるようにしたものです。

データベースソフト (DBMS : data base management system) はデータの蓄積、検索、管理をするためのソフトウェアで、小規模な住所録などを扱うのに適したことから、本格的な販売管理や在庫管理にも対応できるもの、あるいはインターネットの検索サーバなどで用いられ

*20 FEP (front end processor) ともいいます。

*21 アップル社製の初期の 8 ビットパーソナルコンピュータ。「Apple II」(136 ページ) 参照。

*22 データの表示順序を数値の大小、ABC、アイウエオ順などに並び替える機能。

*23 2 つ以上の表を 1 つの表に合体して統合する機能。

*24 「情報処理装置」(4 ページ) 参照。

第4章 ソフトウェア

ているような超大規模なものにいたるまで、さまざまなものがあります。

検索サーバが扱うウェブページのデータベースなどでは、10億件以上ものウェブページの中から、キーワードを含んでいるページを瞬時に検索することができます。データベースソフトは、コンピュータを「最もコンピュータらしく使う」ソフトウェア、といえるでしょう。しかしデータベースソフトは、表計算ソフトでも処理できるような小規模な例外を除けば、データベースを管理するために専用のプログラムを作って運営するのが常ですから、一般のパーソナルコンピュータのユーザーがデータベースソフトを直接扱うことはほとんどありません。

主なDBMSにはオラクル社のオラクル (Oracle)、小規模データベース向きのものとしてマイクロソフト社のアクセス (Access) などがあります。

4.2.4. ウェブブラウザ

ウェブブラウザ (web browser) はウェブページを閲覧するためのアプリケーションプログラムです。パーソナルコンピュータのユーザーが、いちばん利用するソフトウェアのひとつでしょう。HTML *25 (hypertext markup language) ファイルを解析して必要な画像ファイルや音楽ファイルなどをダウンロードし、指定されたレイアウトに従って文字や画像を表示、あるいは音楽の再生なども行います。



図 4.17 ウェブブラウザ (Internet Explorer)

初めて作られたブラウザ、WWW *26 は文字情報だけしか表示できませんでしたが、モザイク (Mosaic) によって画像が表示できるようになり、更にネットスケープ・ナビゲータ (Netscape Navigator) が好評を博して、インターネット普及のきっかけとなりました*27。

現在はマイクロソフト社のインターネット・エクスプローラ (Internet Explorer) が約 80% という圧倒的なシェアを占めていますが、オープンソースプロジェクト Mozilla Foundation によるファイアフォックス (Firefox)、アップル社のサファリ (Safari)、オペラソフトウェア社のオペラ (Opera)、ネットスケープ・ナビゲータなども使われています。

*25 ウェブページを記述するためのマークアップ言語。

*26 「WWW」(141 ページ) 参照。

*27 「WWW から Mosaic、Netscape へ」(142 ページ) 参照。

4.2.5. 電子メールソフト

インターネットで、ウェブページの閲覧に次いで利用されているのは電子メールでしょう。

電子メールソフト*28 (e-mail software) は、電子メールの作成、メールの送受信、メールやメールアドレスの保存や管理を行なうソフトウェア (MUA : mail user agent) です。メールアドレスを管理するアドレス帳の機能や、あらかじめ設定された受信箱に送信者名によって振り分けて保存する機能、データベースに基づいて複数のメールを自動的に送信する機能を持ったものなどもあります。

よく使われている電子メールソフトは、ブラウザ同様マイクロソフト社のアウトルック・エクスプレス (Outlook Express) です。他にはネットスケープ・メッセンジャー (Netscape Messenger) やユードラ (Eudora) などがあります。

4.2.6. グラフィックソフト

コンピュータを用いて図形を作成するコンピュータグラフィックス (CG : computer graphics) のためのもので、ドローソフト (drawing software) やペイントソフト (painting software)、3次元の作図に使用するものなどがあります。

ドローソフトは、定規やコンパスを使う感覚で直線や曲線を描き、これを組合わせて図形にします。正確な図を描いたり、地図を作ったりするのに向いています。データは、線の端点の座標、線の太さや色などの情報として記録されます。

一方ペイントソフトは「お絵かきソフト」ともいわれますが、画素 (ピクセル) 毎に色をつけることで絵を描きます。データは各画素の色の情報として記録されます。

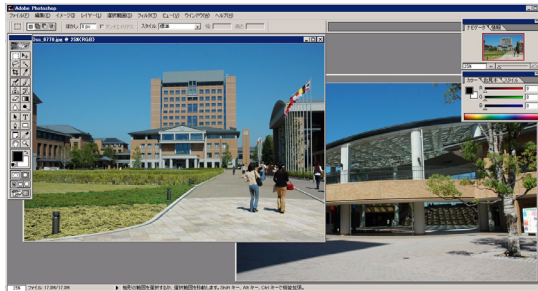


図 4.18 ペイントソフト (Adobe Photoshop)

したがって、ドローソフトで作られた画像データは、点や線、長方形や円などといった図形の量、すなわち図形の複雑さに応じてデータ量が増えるのに対して、ペイントソフトの画像のデータ量はピクセル数、すなわち画像の面積が大きければ増える傾向があります*29。

ペイントソフトはピクセル毎の色のデータを編集するソフトウェアですから、写真を加工す

*28 メール、e-mail ソフト、メールクライアントなど、様々な呼称があります。

*29 画像データを保存するときはデータを圧縮することが多いので、その場合のファイル容量は様々な要因によって変化します。

第4章 ソフトウェア

るのにも向いています。写真加工の機能に特化したペイントソフトを、フォトレタッチソフト (photo retouch software) ということもあります。

代表的なグラフィックソフトには、ドローソフトではアドビイラストレータ (Adobe Illustrator)、ペイントソフトではアドビフォトショップ (Adobe Photoshop) があります。いずれもアドビシステムズ社 (Adobe Systems) の製品で、マッキントッシュ (Macintosh) で動くソフトウェアとしてクリエイターの間で定評がありましたが、IBM PC でもこのようなソフトウェアが動くような環境が整ってくると、Windows 版の製品も現れました。

ペイントソフトには他に、コーレル社 (Corel) のペインター (Painter)、ジャスク・ソフトウェア社 (Jasc Software) のペイントショップ・プロ (Paint Shop Pro) があります。またドローソフトには、他にマクロメディア社 (Macromedia) のマクロメディアフリーハンド (macromedia FREEHAND) があります。

4.2.7. CAD

^{キヤド}CAD (コンピュータ支援設計 : computer aided design) はその名の通り、コンピュータを使って設計するためのソフトウェアです。マウスを使って図面を描くところはドロー系グラフィックソフトに似ていますが、グラフィックソフトは図形を作ること自体が目的であるのに対して、CAD はあくまでも手段として図面を描きます。したがって、CAD には描いた図に自動的に寸法線をつけるなど、設計を助けるためのさまざまな機能が備えられています。また、一口に設計といっても、機械設計や建築、電子回路、プリント基板、集積回路などから服飾にいたるまで、さまざまな分野があるので、それぞれに対して特化された CAD が用意されています。

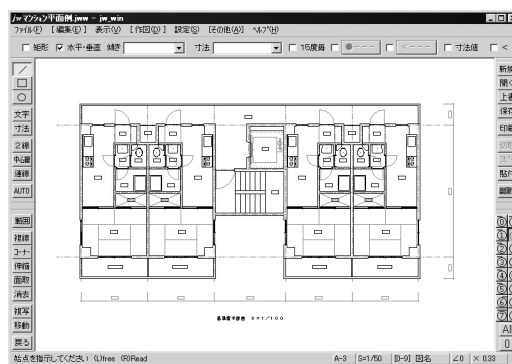


図 4.19 CAD (Jw_cad)

CAD で設計することの利点は、単に図面を効率よく作成するばかりでなく、製品そのものを効率よく作る、あるいは品質を高めるといった点にもあります。たとえば、図 4.19 はマンションの平面図ですが、ここには流し台や浴槽、便器などがたくさん描かれています。こういったものを1つ1つ手で書き込むのはたいへんですが、CAD ならあらかじめ部品として登録され

ているものを呼び出して配置するだけですみます。図面の中にそれらが何個使用されているかは容易に算出できますから、それを見積書や資材発注の資料として利用することもできます。また CAD のデータから工作機械の制御プログラムを自動的に作る、ということも広く行われています*30。

さらに、そのマンションを実際に建築すると季節によってどのような日陰ができるか、あるいは車が時速 80km/h で走行するとどのような空気抵抗があるか、電子回路のどの部分にどの程度の熱が発生するかなど、従来は模型や試作品を作って検討していたことも容易にシミュレーションすることができます。結果が思わしくなければ設計を変更してシミュレーションを繰り返すことによって品質を高め、製品の開発期間を短縮することができ、コストダウンにも寄与しています。

前述の通り、CAD にはさまざまな分野に特化したものがたくさんありますが、パーソナルコンピュータ用の一般的な機械系のものにオートデスク社 (Autodesk) の AutoCAD、あるいは図 4.19 に示した Jw_cad などがあります。Jw_cad は業務用ソフトウェアには珍しいフリーウェアです。清水治郎、田中善文両氏によるもので、CAD の定番ソフトウェアとして広く使われています。

4.2.8. デスクトップパブリッシング

デスクトップパブリッシング (DTP : desk top publishing) とは絵、図、表などを含む高品位な印刷物の原版をパーソナルコンピュータで作作り、CD-ROM などに収めた電子的なデータで出稿して印刷・出版することをいいます。

従来は印刷物の原版は専門の業者でないと作ることができなかったもので、パンフレットや同好会誌など、比較的小部数の出版の場合は 1 部当たりの製作費が割高になり、やむをえず手書きのまま印刷することも少なくありませんでした。DTP は高品質の印刷原版を手軽に作ることができるため、現在ではほとんどの印刷物が DTP によって作成されています。

グラフィックソフト同様、DTP も当初は Macintosh の^{どくせんじょう}独壇場でした。現在は Windows パソコンでも可能になっていますが、商業出版物には現在でも Macintosh が使われることが多いようです。DTP ソフトにはクォーク社 (Quark) の QuarkXPress、アドビ社の PageMaker などがあります。また、CAD に Jw_cad というフリーウェアがあったように、DTP にも^{ラテックス}L^AT_EX というフリーウェアがあります。L^AT_EX は、スタンフォード大学のドナルド・クヌース (Donald E. Knuth) が作った組版システム、T_EX を基礎にして、レスリー・ランポート (Leslie Lamport) が構築した文書処理システムです。きめ細かい組版の指示ができるほか、複雑な数式の記述が簡単なことから、理工系の文書の作成には特に広く利用されています*31。

*30 このようなシステムを CAD/CAM (computer aided design/manufacturing) といいます。

*31 このテキスト、「Web で学ぶ情報処理概論」も、すべて L^AT_EX によって作成されています。

4.2.9. 音楽ソフト

MIDI形式^{*32}で音楽データを作成、演奏するためのソフトウェアで、シーケンサソフト (sequencer software) ともいいます。

音の高さや長さ、強さ、音色といった楽譜のデータを音にして出力するために、かつては専用のハードウェア (音源) が必要でしたが、CPU の処理能力が飛躍的に高くなった現在では、外付けのハードウェアがなくても演奏できるようになりました^{*33}。

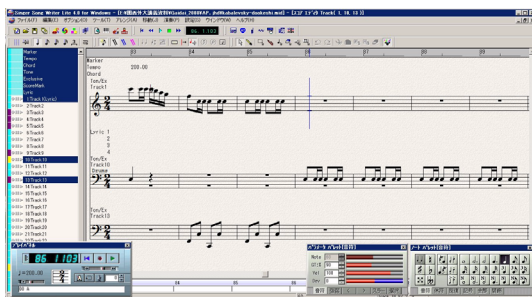


図 4.20 シーケンサソフト (Singer Song Writer)

データは電子楽器などをパーソナルコンピュータに接続して楽器を演奏し、あたかも録音するかのように入力することも、楽譜のデータをマウスやコンピュータのキーボードで直接入力することもできます。入力したデータは、もちろん自由に編集できます。データは音の高さや音色 (の番号) などですから、データ量は CD のように録音されたものに比べると極めて小さくなります。カラオケや着メロなどには、このようにして作成された音楽が使われています。

4.2.10. 翻訳ソフト

1960~1980年代にかけて、盛んに人工知能 (AI : artificial intelligence) の研究が行われましたが、結果的には人工知能自体は実現できず、逆に本格的な人工知能を作るのは不可能という結論が得られています。

しかしその研究過程で、たとえばファジー理論^{*34} (fuzzy theory) など多くの成果があり、現在はこうした技術が家庭のエアコンや洗濯機などにまで応用されています。

自動翻訳も当時の課題のひとつで、これも人工知能ではありませんが、コンピュータの処理能力の向上もあって一応の成果が得られ、今日では翻訳ソフト (translation software) として利用されています。

とはいえコンピュータは単なる超高速計算機にすぎませんから、人間のように直感的に構文を解釈したり、適切な訳語を選ぶことはできません。「文学的に」という注文がつけば尚更で

^{*32} musical instruments digital interface. 電子楽器やシンセサイザとパーソナルコンピュータ間で音楽のデータをやりとりするための規格。

^{*33} もちろん、音源を準備した方が、いい状態で再生できます。

^{*34} あいまいなものをコンピュータで取扱うための理論。あいまいさをメンバーシップ関数 (membership function) で表します。

す。現在の翻訳ソフトの実力と限界は、次のような翻訳例によって理解できるかと思います。

The considerations which follow deal with the structure of a very high speed automatic digital computing system, and in particular with its logical control. Before going into specific details, some general explanatory remarks regarding these concepts may be appropriate.

これはノイマンの“First Draft of a Report on the EDVAC”^{*35}の冒頭です。ある翻訳ソフトに翻訳させてみました。

次に続く考慮は非常に高速の自動的なデジタルの計算システムの構造、そして特にその論理的な制御について論ずる。特定の細部を詳しく述べる前に、これらの概念に関しての若干の一般的な説明的な発言が適切であるかもしれない。

まづまづの訳ですが、次はブラウニング (Robert Browning) の Pippa's Song^{*36} です。

The year's at the spring
And day's at the morn;
Morning's at seven;
The hill-side's dew-pearled;
The lark's on the wing;
The snail's on the thorn:
God's in his heaven —
All's right with the world!

下の左はある翻訳ソフトの訳です。比べては気の毒ですが、上田敏は海潮音に「春の朝」と題して、右のように訳しています^{*37}。

年は春にある
そして日が朝にある；
朝が7時にある；
山腹は露 - 玉のようにたれた；
ヒバリは飛んでである；
カタツムリはとげの上にある；
神は彼の天国にいる—
すべては世界と一緒に正しい！

時は春、
日は朝、
朝は七時、
片岡に露みちて、
揚雲雀なのりで、
蝸牛枝に這ひ、
神、そらに知ろしめす。
すべて世は事も無し。

^{*35} 「EDVAC に関する報告書の初稿」。Springer-Verlag “The Origins of Digital Computers” p.383.
「ノイマン型コンピュータ」(121 ページ) 参照。

^{*36} James F. Loucks “Robert Browning's poetry: authoritative texts, criticism” p.28

^{*37} 岩波書店「上田敏全訳詩集」(1983年) p.77。

4.2.11. 音声認識

音声認識 (speech recognition) ソフトは、人が話す音声 を直接コンピュータに入力して解析し、テキストデータに変換するソフトウェアです。

人の声には個人差があり、話し方も違うため、たいていの音声認識ソフトはあらかじめ一定の文章を読み上げさせ、声や話し方の特徴などを登録するようになっています。これをエンロール (enroll) といいます。

読み上げる文章、話した言葉をほぼリアルタイムで変換できるため、キーボード入力と違って文字入力の効率は飛躍的に高まりますが、時には誤認識や漢字の誤変換があるため、入力後の編集作業は欠かせません。

音声を認識するためには高度な処理が必要ですが、パーソナルコンピュータや音声認識ソフトの性能が高くなるにつれ、認識精度も年々向上しています。音声認識のアルゴリズム*38 を開発したのは IBM で、音声認識ソフトも同社の ViaVoice がよく知られています。

4.2.12. 音声合成

音声合成 (speech synthesis) とは人が話す音声を人工的に合成して作ることで、こうした技術そのものは 1960 年代にすでに研究されていました。1980 年代になると、操作方法を音声で案内する「喋る家電」や、時刻を音声で告げる時計なども現れました。今日では電話窓口で自動応答を行なう音声応答システム (IVR : interactive voice response) や、駅での案内放送、カーナビゲーションシステムの音声案内など、音声合成の応用例は私たちの身の回りにたくさんあります。

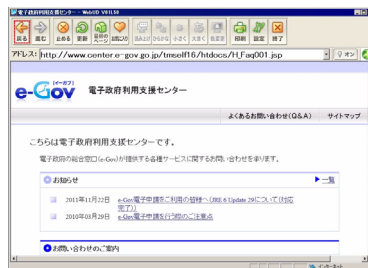


図 4.21 WebUD

コンピュータのソフトウェアにも、たとえば高齢者や視覚障害者がウェブの情報を利用するときに使う、音声読み上げブラウザなどがあります。図 4.21 はその一例 WebUD で、経済産業省「平成 14 年度障害者等用情報通信機器等開発事業」の成果物です*39。WebUD に対応しているウェブページのテキストを読み上げてくれます。

*38 目的を達成するための処理手順をアルゴリズムといいます。ここで使われているのは「隠れマルコフモデル (hidden Markov model)」です。

*39 電子政府利用支援センターのページ (<http://www.center.e-gov.go.jp/webud/download.jsp>) から無料でダウンロードできます。

WebUD はブラウザですが、もちろん音声合成単体のソフトウェアもあります。よほど注意して聴かないと合成音声だと気づかないほど自然なものから、合成音声だとすぐに分かるものまで様々ですが、不自然なのはたいていアクセントや抑揚なので、読み上げている文章の意味が分からないということは少ないようです。

いろいろの音声合成ソフトウェアがありますが、ある程度の不自然さを我慢するならフリーウェアもあります*40。コンピュータで何かやってみたいとき、まずフリーウェアを試してみると気軽に始められます。フリーウェアに飽き足りなくなったら本格的なソフトウェアを探せばいいのだし、時には市販のソフトウェアより優れているものに出会うこともあります*41。

ところで、最後に紹介した翻訳と音声認識、音声合成の三つの機能を組み合わせると、自動通訳システムになります。たとえば日本語で話した音声を認識してテキストデータにし、翻訳ソフトで英文にして英語の音声に変換すればいいわけです。現状では音声認識には認識率、翻訳には構文の解析、音声合成にも抑揚の不自然さなど、いくつかの問題は残されていますが、そこそこのレベルには達しています。自動通訳機の実現に向けて研究が続けられていますが、これだけの機能を携帯端末に実装するのは、やはりまだ難しいようです。

*40 たとえば「ドキュメントトーカー」、「棒読みちゃん」など。いずれも Vector (<http://www.vector.co.jp/>) からダウンロードできます。

*41 残念ながら、劣悪なもの、コンピュータウイルスのような悪意のあるものに出会うこともあります。説明をよく読んで信頼できるサイトからダウンロードする、などの注意が必要です。

第5章 コンピュータの歴史

20世紀の科学技術の歴史の中で、最も重要な発明といえばコンピュータ、トランジスタ、レーザなどでしょう*1。現代社会は高度情報化社会といわれていますが、それを支えているのもコンピュータや半導体、レーザです。現在コンピュータはいたるところで利用されていますが、コンピュータが発明されたのはわずか70年前、コンピュータが身近なものになったのは、ほんの十数年前のことです。

コンピュータはどのようにして生まれ、どのようにして情報化社会が出現したのでしょうか。

1. 機械式計算機

人は古来より計算を助けるために、算木やジュトン*2、^{そろばん}算盤など、さまざまな道具を使用してきました。しかし、少なくとも計算機、「機械」と呼べるものが現れるには、17世紀まで待たなくてはなりません*3。

1.1. パスカルの計算機

現存する最古の歯車式計算機は、パスカルの原理やパンセ、あるいは「中心気圧は960ヘクトパスカル…」などでお馴染みの、パスカル (Blaise Pascal) (1623~1662) が1642年に作った「パスカリーヌ」です。



図 5.1 パスカルの計算機「パスカリーヌ」

パスカルの父は税務官吏で、毎日膨大な計算に追われている父の姿を見て計算機の製作を思い立ったといわれています。50台程の計算機が作られました。図 5.1 は後期のもので、8個並んでいる歯車のうち左の6個は10進数、右からふたつ目は20進数、右端は12進数で計算

*1 核兵器なども？

*2 jeton. カウンターの意。

*3 1900年にエーゲ海で、古代ギリシャの沈没船の中から、紀元前一世紀の天文学的時計（あるいは計算機）と思われる機械が発見されています。(J. ディビッド・ボルター著「チューリング・マン」1995年 みすず書房 p.22)

をするようになっていきます。当時のフランスの貨幣制度が複雑であったため*4、このような変則計算機も製作されたようですが、広く普及するには至りませんでした。

1.2. ライプニッツの計算機

パスカルの計算機は加減算だけしかできませんでしたが、ライプニッツ*5 (Gottfried W. Leibnitz) (1646～1716) は 1674 年、これを加減乗除が行えるものに改良しました。段付歯車を使用した機構や、加減算を繰り返すことによって乗除算を行うライプニッツの計算機の基本的な考え方は、後々の機械式計算機に永く受け継がれました。

機械式計算機はその後、1820 年にトーマス (Charles X. Tothomas) が改良を加えて商品化し、1875 年にはボールドウィン (Frank S. Baldwin) らによって更に改良されました。

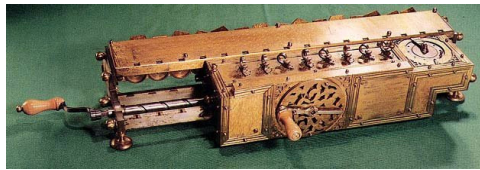


図 5.2 ライプニッツの計算機

1.3. オドナーの計算機

スウェーデンのオドナー (Willgodt T. Odhner) は、1874 年にピン付きの歯車を使って計算機をコンパクトにすることに成功しました。図 5.3 は 1940 年頃に作られたものです。

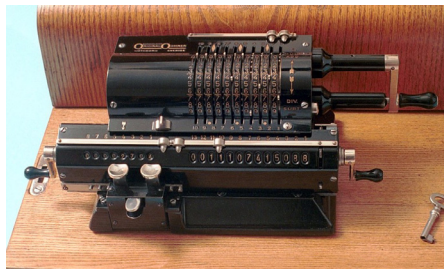


図 5.3 オドナーの計算機

日本では大本寅治郎 (1887～1961) が 1923 年 (大正 12 年) に国産計算機を完成、「虎印計算器」と名づけて販売し、後にタイガー手廻計算器として長く親しまれましたが、1969 年に LSI を使った電卓が登場*6したため、翌 1970 年に製造が中止されました。

*4 12 ドゥニエが 1 スー、20 スーが 1 リーブル。

*5 ニュートンと並んで、微積分学を確立したことで有名な数学者。

*6 早川電機 QT-8D。「電卓の IC 化」(131 ページ) 参照。

1.4. バベジの階差機関

パスカルの計算機以来 300 年余りの間、人々はこれまで見てきたような、機械式の卓上計算機を工夫し、使ってきました。これらの計算機に共通しているのは、いずれも卓上の小型計算機だということです。歯車やハンドルを操作して簡単に手軽に計算しようというもので、現在の電卓と発想は同じです。

そうした中であって、時代は少しさかのぼりますが、バベジ (Charles Babbage) (1791~1871) は、自動計算機の夢を追っていました。



図 5.4 バベジ

当時イギリスは海運が盛んでしたが、船の位置を算出するのに使う数表の誤りによる海難事故に悩まされていました。船の位置は星を観測して三角関数で計算しますが、その三角関数の数表に誤りが多かったのです。1833 年、バベジはそのような計算を自動的に行い、同時に計算結果を印刷する階差機関 (difference engine) を作ろうと考えました。

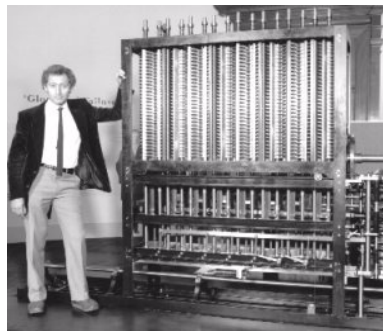


図 5.5 階差機関

数 列	①	4	9	16	25	36	49	64	81	100
第一階差	③	5	7	9	11	13	15	17	19	
第二階差		②	2	2	2	2	2	2	2	2

上の例で、1、4、9、16… という数列はそれぞれ 1^2 、 2^2 、 3^2 、 4^2 …、すなわち一辺が 1、2、3、4… の正方形の面積ですが、互いの差 (第一階差) を取ると 3、5、7、9…、さらにその差 (第二階差) は 2、2、2、2… となって、一定になります。このように、何度か階差を取れば一

定になるような数列であれば、1、3、2 という初期条件さえ与えれば、あとは加算だけで自動的に計算することができます。階差機関はこれを実現しようとしたものです。

残念ながら当時は機械加工精度が十分でなく、政府の資金援助がうち切られたこともあって、完成させることはできませんでした。図 5.5 の階差機関は 1991 年、バベジの生誕 200 年を記念してイギリスの科学博物館が作ったものです*7。

1.5. 解析機関

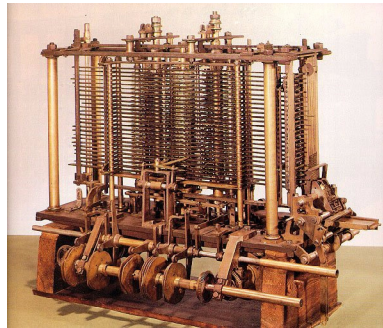


図 5.6 解析機関の一部

しかし階差機関は、階差法が使えるものしか計算できない、いわば専用機です。1834 年、バベジは新たに、プログラムによってどのような計算でも行える解析機関 (analytical engine) の開発に着手しました。解析機関は演算装置 (mill) と記憶装置 (store) からなっています。データやプログラムを入力するパンチカード読み取り装置や、計算結果を印刷する出力装置もあって、機械式とはいえ今日のコンピュータとほぼ同じ構成になっています。しかしこれも、部分的に動作を確認するための試作機は作られましたが、装置全体としては完成に至りませんでした。

エイダ *8 (Augusta Ada Byron) (1815~1852) は、バベジがイタリアで行った解析機関に関する講演の記録を英訳し、膨大な訳注をつけています。ここには解析機関のプログラムについても触れられており、ループや条件分岐など、今日もコンピュータのプログラムに不可欠な概念が記されています。

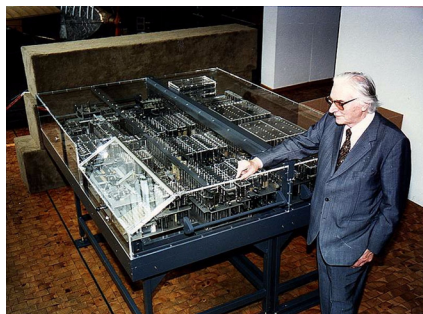
1.6. ツーゼの Z1

バベジの解析機関以後 100 年間、新たに機械式の自動計算機を試みた者は現れませんでした。機械式ではあまりにも複雑になって、仮に製作できたとしても、費用やサイズ、重量、保守や信頼性の問題などを考えると、とうてい現実的ではありません。

*7 バベジの図面には若干の誤りがありましたが、いずれも簡単に修正できる程度のものでした。この複雑な機械が、バベジの図面通りに作って正常に動作したそうです。

*8 詩人 George G. Byron の娘で後のオーガスタ・ラヴレース伯爵夫人。バベジと親交があり、数学や解析機関についてバベジから多くを学びました。

コンラッド・ツーゼ (Konrad Zuse) (1910~1995) は軍用機などを生産していたヘンシェル社の技術者でしたが、航空力学に必要な、膨大で単調な計算に辟易^{へきえき}していました。そこで自動計算機の製作を計画し、1936年から38年にかけて、10進数ではなく2進数を、歯車ではなく薄い金属板製のカムを採用した計算機、Z1を製作しました。



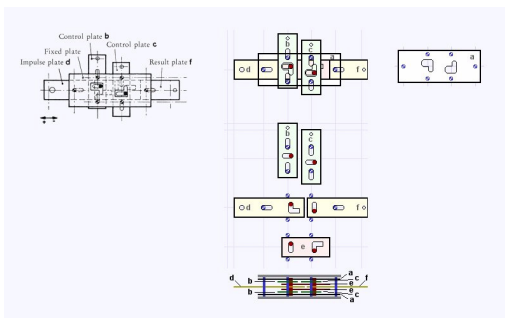
演算ユニット	22ビット浮動小数点演算
プログラム	テープ読み取り式
クロック周波数	1Hz
計算速度	乗算約5秒
消費電力	約1kW(モーター用)
重量	約500kg

図 5.7 Z1 (レプリカ) とその仕様

Z1はパンチテープでプログラムを与えて22ビットの浮動小数点演算ができる、言うまでもなく世界で最初の、本格的な科学技術計算ができる計算機でした。Z1のクロック周波数は1Hz、乗算に要する時間は約5秒です。バベジの夢は100年後に、Z1によって初めて実現されました。

Z1の論理素子^{*9}は金属板のカムを組み合わせで作られています。入力側のカムの動きが出力側に伝わるかどうかを、制御用のカムの位置によって決めることができるようにしてあります。金属板のカムの位置を2進数の0と1に対応させ、様々な論理素子を組み合わせで演算装置が作られています。

Web で学ぶ 【Z1】



Z1はカムを使った機械式の自動計算機です。Web ページ「Z1」では、ツーゼが独自に考案した論理素子のひとつを画面上で動くようにしてあります。

カムを動かすとピンや他のカムがそれに伴って動くので、ツーゼの論理素子のしくみがよく分かります。

*9 論理演算を行うための回路を構成している素子。通常はトランジスタなどの半導体素子が使われる。

2. チューリングマシン

チューリング*¹⁰ (Alan M. Turing) (1912~1954) はイギリスの数学者で、チューリングマシンは 1936 年に発表された彼の論文、「計算可能数についての決定問題への応用」*¹¹ の中で、計算を数学的にモデル化するために使われた仮想の計算機です。



図 5.8 チューリング

チューリングマシンは無限に長いテープと、テープの記号を読み書きできるヘッドから成っています。ヘッドはその内部状態と読み出した記号（入力）に応じてテープに書くべき記号を出力し、右か左にひとつ移動して内部状態を変えます。これを停止するまで繰り返します。

内部状態	入力	出力	移動方向	次の内部状態
0	空白	1	右	1
0	0	1	右	1
0	1	0	左	0
1	0	0	右	1
1	1	1	右	1
1	空白	空白	左	H
H	—	—	停止	H

表 5.1 チューリングマシンの動作例

チューリングマシンを理解するために、ここでは 2 進数の値に 1 を加える、たとえば 101 なら 110 に、111 なら 1000 にするという簡単な「計算」を考えます。この場合、ヘッドは 0、1、H（停止）の 3 種類の内部状態を取ります。

ヘッドの内部状態が 0 のとき、テープから 0 または空白を読み込むと、テープに 1 を書き込んで右に動いて内部状態を 1 に変え、1 を読み込むと 0 を書き込んで左に動いて内部状態は 0 のままにします。また、内部状態が 1 のとき、テープの記号が 0 または 1 であれば右に動き、空白であれば左に動いて停止します。以上をまとめたものが表 5.1 です。

*¹⁰ チューリングは第 2 次世界大戦中、ドイツの暗号エニグマ (220 ページ参照) の解読機「ボンベ」(bombe) を開発し、戦争の早期終結に貢献したことも知られています。

*¹¹ On computable numbers, with an application to the Entscheidungsproblem.

いま、テープには101という2進数が書かれていて、最初ヘッドの内部状態は0で最下位ビットの上にあるものとします（図5.9）。

このとき、テープの記号は1ですから、ヘッドはテープに0を書き込んで左に動きます（表5.1の3行目）。次のステップでは内部状態が0、記号も0ですから、テープには1を書き込んで右に動き、内部状態を1に変えます（表5.1の2行目）。次のステップは内部状態が1、記号が0なのでテープに0を書き込んで右に動き、内部状態は1のままという風に、定められた規則に従って処理を進めていくと、ヘッドが停止したときにはテープの2進数は110になっています。

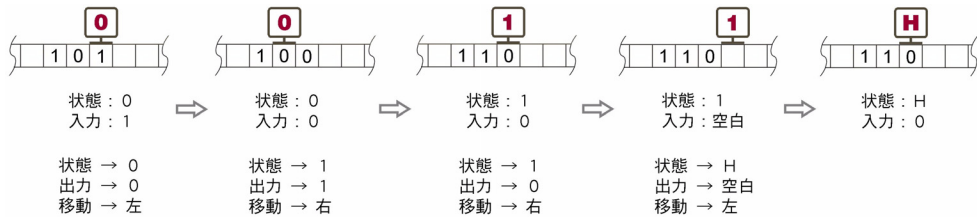
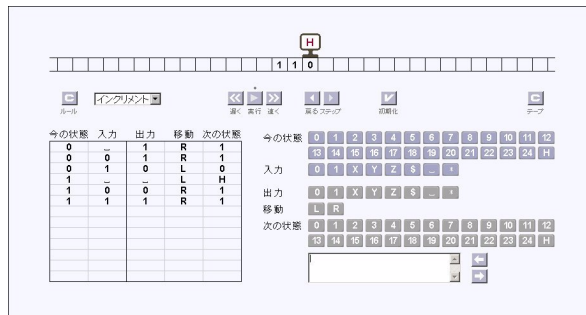


図 5.9 チューリングマシンの計算例

これは極めて簡単な例ですが、チューリングマシンに無限の長さのテープがあって、プログラムに相当する表5.1のようなヘッドの遷移規則が書けるようなものであれば、どんなに複雑なものでも計算することができます。ヘッドの遷移規則そのものを記号化してテープに書き込んでおき、ヘッドがそれを読み込んで処理を始めることもできます。現在のコンピュータはメモリに記憶されているプログラムを随時読み込みながら処理を進めていきますが、チューリングマシンはそのような計算機の可能性も示唆しています。

Web で学ぶ 【チューリングマシン】



Web ページ「チューリングマシン」には、動くチューリングマシン（Java アプレット）があります。いくつかのプログラムがあらかじめ用意されているので、計算処理が行われていく様子を見れば、チューリングマシンが容易に理解できます。データやプログラムの編集もできます。

3. 電気（リレー）式計算機

1937年、シャノン（Claude E. Shannon）（1916～2001）は、スイッチの組み合わせによって論理回路を作ることができるという論文を書き^{*12}、スイッチで AND、OR、NOT 回路を構成し、それらを組み合わせればあらゆる論理回路を実現できることを示しました。これによって、歯車をスイッチに置きかえ、リレーで計算機を作る道が開かれました。

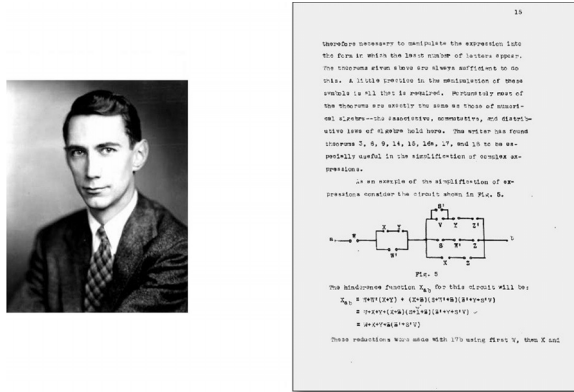


図 5.10 シャノンと論文の一部

リレーは電磁石とスイッチを組み合わせたもので、図 5.11 のように、接点は何個か鉄片に取り付けられています。普段はスプリングによって接点 c は接点 b と接触していて、c-b 間は ON ですが、c-a 間は OFF になっています。コイルに電流を流すと鉄片は電磁石に引きつけられて、c-b 間は OFF、c-a 間は ON になります。何個かの接点を同時に ON-OFF させることができるので、論理回路にはよく利用されます。

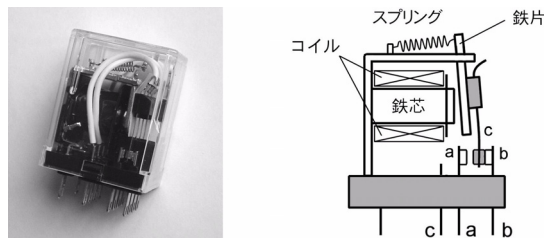


図 5.11 リレー

3.1. Z3

Z1 は機械式のため演算装置がやや不安定でしたので、ツーゼは演算装置をリレー化した Z2 を試作（1939）した後、1941年に本格的な電気（リレー）式計算機、Z3 を完成させました。Z3 にはメモリ（22 ビット× 64 語）を含め、2,600 個のリレーが使用されています。加算を 0.7 秒、乗算と除算を 3 秒で実行できる、世界で最初の電気（リレー）式計算機が誕生しました。

^{*12} A Symbolic Analysis of Relay and Switching Circuits. マサチューセッツ工科大学での修士論文。

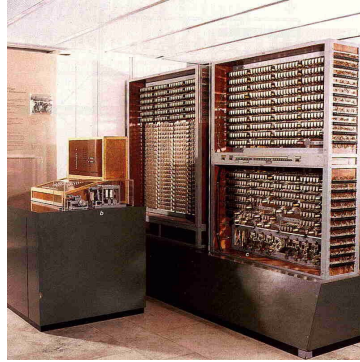


図 5.12 Z3

残念ながら Z1 も Z3 も第 2 次世界大戦中、連合軍の爆撃（1944 年）によって破壊されてしまいました。図 5.12 は 1960 年に復元されたレプリカで、1967 年にモントリオール博覧会に、現在はミュンヘンのドイツ博物館に展示されています。

その後会社を設立したツェーゼは、演算精度を上げた Z4（1945）、光学機器設計のために Z5（1952）、商業的にも成功した真空管式の Z22（1957）、トランジスタ式の Z23（1961）など、多くのコンピュータを世に送り出しました。こうした数々の業績にもかかわらず、今日ツェーゼがあまり知られていないのは、彼が敗戦国ドイツの人であり、Z1 や Z3 など初期のコンピュータがごとごとく爆撃で破壊されてしまったことなども理由の一つでしょう。

3.2. ハーバード Mark-I

米国では 1943 年、ベル研究所のスタイビッツ（George R. Stibitz）（1904～1995）が弾道計算のためにリレー式計算機を、エイケン（Howard H. Aiken）（1900～1973）も 1944 年、IBM の協力を得て、3,300 個のリレーを使った ASCC（automatic sequence controlled calculator）という計算機を作りました。ASCC は 50 フィートもある巨大な計算機で、加減算を 0.3 秒、乗算を 4 秒、除算を 11 秒で計算できました。後にハーバード大学に寄付され、ハーバード Mark-I と呼ばれました。

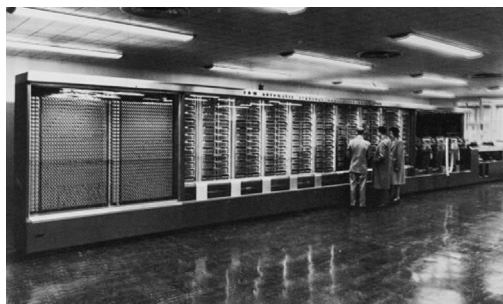


図 5.13 ハーバード Mark-I

写真提供：International Business Machines Corporation.

4. 電子計算機の誕生

電気（リレー）式計算機によって、自動計算機は実用的な機械となりました。しかし、リレーは電磁石によってスイッチを ON-OFF するため、高速動作は望めません。リレーを組み合わせで作った計算機は、1 秒間に数回しか命令を実行できないので、これでは計算を何度も繰り返す高度な処理には向きません。

大量の計算を短時間で行うために、リレーに代わって、真空管をスイッチとして使った高速の計算機、すなわち電子計算機が作られるようになりました。

真空管を使うことによって処理速度の問題はある程度解決できますが、メモリをどうして作るかという難問がまだ残っています。

コンピュータには高速に読み書きできる大容量のメモリが必要ですが、メモリを真空管で作ると大量の真空管が必要になるため、装置の規模が膨大になってしまいます*13。大容量のメモリをコンパクトに作るために、当時のコンピュータ開発者達がいかに苦労したか、そのような点にも注目しながら、コンピュータ誕生の歴史を眺めていくことにします。

4.1. ABC

1942 年、アイオワ州立大学のアタナソフ（John V. Atanasoff）（1903～1995）は、ベリー（Clifford E. Berry）（1918～1963）と共に、ABC（Atanasoff-Berry computer）という電子計算機を作りました。

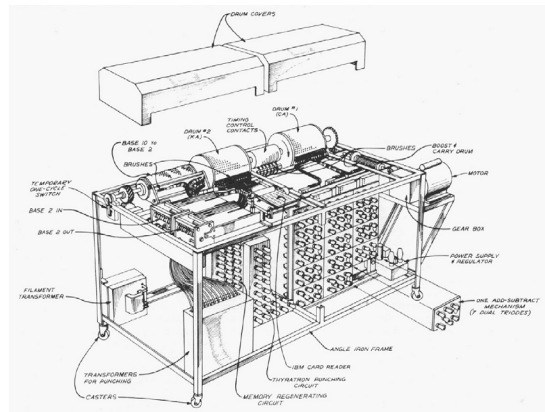


図 5.14 ABC

ABC は連立方程式を解くための、やや大きめの机くらいの計算機です。2 進数演算で、約 300 本の真空管を用いています。

*13 1 ビットを記憶するのに 2 本の真空管が必要です（1 本の真空管に 3 極真空管の要素が 2 個封入された「双 3 極管」を使えば、見かけ上 1 本の真空管で済ませることができます。ABC や ENIAC にもこのような真空管が随所に使用されています）。

ABCは連立方程式を解くための専用機ですから、プログラムはハードウェアで固定されています。写真の左上にあるドラムがABCのメモリですが、このメモリは演算データだけを記憶します。図5.15はメモリの断面の模式図です。ドラム上に6°間隔で50個、それが32列並んだ接点にコンデンサが接続されています。コンデンサには1ビットのデータを記憶させることができます。ドラムは2個あるので、全部で3,200ビットのメモリです。

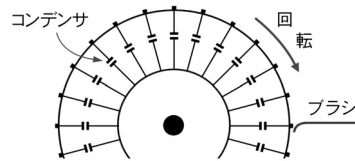


図 5.15 ABCのメモリ

ドラムは1秒で1回転し、コンデンサがプラスに充電されているか、マイナスに充電されているかをブラシで読み取ります。このように、一列に並んでいるデータをその順序で読み書きするメモリをシーケンシャルアクセスメモリ (sequential access memory) といいます。

現在私たちが使っているメモリ、DRAMもコンデンサを使って情報を記憶させていますが、ABCのドラムは1回転するのに1秒かかりますから、先頭のデータがやってくるのを待つのに平均して0.5秒、50個のコンデンサのデータを全部読むのに約0.8秒かかります。これでは真空管を使ったせっかくの高速演算回路も台無しで、計算機全体としての性能は機械式計算機並みになってしまいます。

とはいえ、世界で最初の、電子回路によるデジタル式計算機であったこともまた事実です。

4.2. Colossus

Colossusは第2次世界大戦も最盛期の1943年、ドイツ軍のローレンツ暗号*14を解読するために、ニューマン (Maxwell H. A. Newman) (1897~1984) らによってイギリスで作られた電子計算機です。

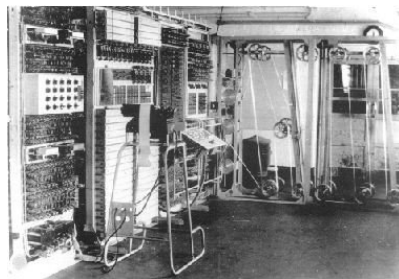


図 5.16 Colossus

*14 ドイツの暗号といえばエニグマ (220ページ参照) が有名ですが、ヒトラーや将官との間の通信には、更に複雑なローレンツ暗号機が使用されていました。

Colossus に使われている真空管は 2,500 本で、データのメモリとして使われている 5 台の 5 ビットのシフトレジスタ*15 も真空管で作られています。プログラムはハードウェアで与えられ、一部は固定、一部はプラグの差替えによって変更できましたが、いずれにしても暗号解読のための専用機です。

第二次世界大戦では暗号解読によって戦争の早期終結に貢献しました。本格的に稼働した最初の電子計算機でもあります。

4.3. ENIAC

米国ではその頃、弾道計算のための高速の計算機を必要としていました。

米国はヨーロッパの諸国より遅れて、日本の真珠湾奇襲（1941 年）を契機として第 2 次世界大戦に参戦したため、当初は軍備も十分に整っていませんでした。そこで陸軍弾道研究所は、ペンシルバニア大学ムーア校のモークリー（John W. Mauchly）（1907～1980）とエッカート（John P. Eckert）（1919～1995）が計画していた電子計算機、ENIAC（electronic numerical integrator and computer）に資金を提供して開発を進めることにしました。

ENIAC は使用した真空管が 18,800 本、消費電力は 150kW、重量 30 トン、100m² の部屋を専有する巨大な計算機でした。10 進演算方式のため、リングカウンタというやや特殊なカウンタを演算装置として使用し、10 組のリングカウンタからなる「アキュムレータ」*16 が 20 台ありました。ENIAC のプログラムはアキュムレータ間等をハードウェア的に接続することで与えられ、プログラムを変更するにはそれらのケーブルをつなぎ変えなくてはなりませんでした。

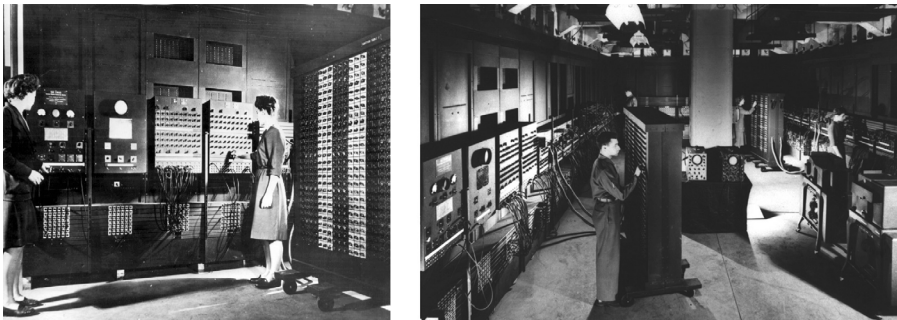


図 5.17 ENIAC

ENIAC の開発目的は弾道計算でしたが、完成は終戦後の 1945 年秋になり、翌 1946 年に機密解除されて公表されました。最初に行われたプログラムは水爆開発のための計算であったといわれています。

*15 データを右、又は左に移動させられるレジスタ。たとえばレジスタの値がいま 11100 であるとき、右にシフトさせると 01110、左にシフトさせると 11000 になります。

*16 るいさんき 累算器。今日アキュムレータという用語は、演算の結果が格納される汎用レジスタの意味で使いますから、ENIAC の「アキュムレータ」とは概念が異なっています。ENIAC の「アキュムレータ」は約 500 本の真空管で構成されていました。

4.4. The Baby

The Baby は、マンチェスター大学のウィリアムス (Freddie C. Williams) (1911~1977) とキルバーン (Tom Kilburn) (1921~2001) によって作られ、1948年6月にプログラムの実行に成功した、最初のプログラム内蔵方式*17 のコンピュータです。

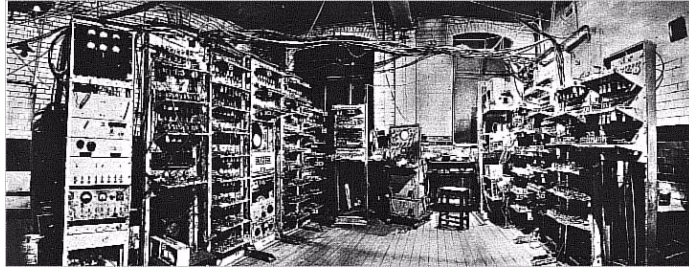


図 5.18 Baby Mark 1 prototype

“Baby” はニックネームで、正式な名称は SSEM (small scale experimental machine) といいます。ウィリアムスは 1946 年に CRT (cathode ray tube) をメモリとして使用する研究に着手し、1 ビットのメモリとして動作させることに成功しました。これを蓄積記憶管といますが、1947 年には 2,048 ビットを記憶できるようになり、コンピュータのメモリとして実際に使えるどうかを確認するための実験機として Baby が製作されました。

蓄積記憶管のデータは CRT の蛍光面上の電荷によって記憶されます。CRT のガラスも蛍光物質も絶縁体ですが、それでも電荷はすぐに漏れて失われるので、電荷を読み出しては再書き込みすることによって記憶が保たれるように工夫されました。このプロセスは “regeneration” と呼ばれましたが、ABC のコンデンサドラムメモリにも、現在の DRAM にも同様の方式が使われています。現在、私たちはこれをリフレッシュ (refresh) と呼んでいます。

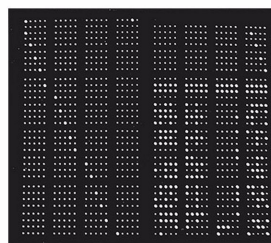
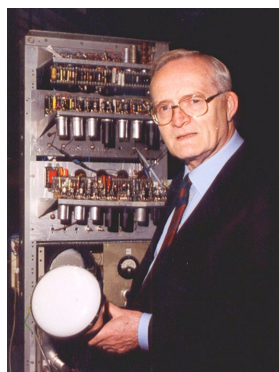


図 5.19 左. キルバーンと CRT 右. CRT 画面

図 5.19 の左はキルバーンと CRT、右は蓄積記憶管のデータの様子を別の CRT に表示させ

*17 「ノイマン型コンピュータ」(121 ページ) 参照。

たものです*18。明るい点は‘1’、暗い点は‘0’を表しています。

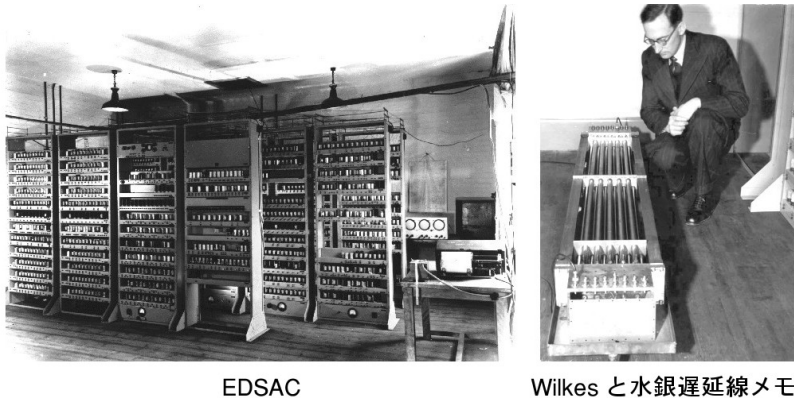
ABCのメモリはコンデンサドラム式のシーケンシャルアクセスで1回転/秒であったのに対して、BabyのメモリはCRTを使ったランダムアクセスメモリで、しかも電子管ですから高速です。Babyは約1.2msecで演算できましたが、これもメモリが高速だったからに他なりません。しかし、メモリ容量は1kビット（128バイト）しかありませんでした。これでは到底、満足な仕事はできません。その名の通り“Baby”、「小規模実験機」でした。

4.5. EDSAC

EDSAC (electronic delay storage automatic calculator) は1949年、ケンブリッジ大学のウィルクス (Maurice V. Wilkes) (1913~2010) によって作られました。

EDSACの正式名称中、“delay strage”とは「遅延線メモリ」のことです。

コンピュータの信号を超音波のパルスにして水銀を満たした管の片側から与えると、超音波は水銀中を伝わって反対側に達します。これを電気信号に変えて、また超音波のパルスにして管に戻してやると、水銀の中を流れ続ける超音波のパルスとしてデータを記憶させることが出来ます。これを遅延線メモリといいます。



EDSAC

Wilkes と水銀遅延線メモリ

図 5.20 EDSAC

出典：Computer Laboratory, University of Cambridge

図 5.20 右の、パイプが収められている長い箱が水銀遅延線メモリです。パイプの長さは 5 フィートあります。水銀中の音速は 1,450m/sec ですから、約 1msec でデータが循環します。EDSAC のクロック周波数は 500KHz ですから、CPU は $2\mu\text{sec}$ 毎に動くこととなりますが、メモリアクセスが 1msec と遅いため、実際に実行できる命令は平均すると 650 回/秒でした。

1本のタンク（水銀を満たした管）に、1語（ワード）17ビットのデータが32ワード記憶でき、そのタンクが図 5.20 の写真の箱に16本納められています。ただし17ビットのうち1ビットは符号として使われているので、このメモリの実質的な記憶容量は $16 \times 32 \times 16 = 8,704 = 8\text{k}$ ビット（1kバイト）と考えてよさそうです。

*18 Babyのメモリは32ビット×32語ですが、写真は後に作られた40ビット×32語のものです。

わずか1kバイトのメモリには大きすぎるように思えますが、もしこれを真空管で作るとすると、1ビット当たり2本の真空管が要るので、1kバイトでは $2 \times 1,024 \times 8 = 16,384$ 本もの真空管が必要になります。EDSACに使用されている真空管は約3,000本ですが、メモリだけでENIAC（約18,800本の真空管を使用）ほどの真空管が必要になってしまいます。

メモリにどのような方式を採用するかということが、当時コンピュータを設計する上で、いかに重要な問題であったかが分かります。

4.6. メモリの比較

	ABC	Colossus	ENIAC	The Baby	EDSAC
完成年	1942	1943	1946	1948	1949
メモリ素子	コンデンサ	真空管	真空管	蓄積記憶管	水銀遅延線
記憶容量 (bit)	3,200	25	2,000	1,024	17,408
コスト	○	△	×	○	△
アクセス方式	△ (順次)	-	-	○ (ランダム)	△ (順次)
アクセス時間	×	○	○	○	△

表 5.2 メモリ方式の比較

表 5.2 にそれぞれのコンピュータのメモリを比較してみました*19。こうしてみると、記憶容量が少ないことをのぞけば The Baby の蓄積記憶管が優れているものの、総合的には水銀遅延線メモリがいちばんバランスがよさそうです*20。

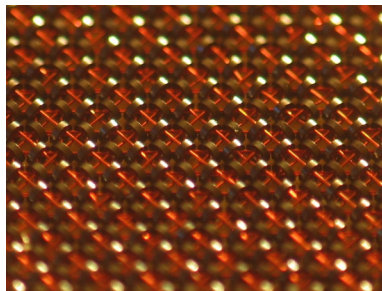


図 5.21 磁気コアメモリ
写真提供：Brian P. Walenz

その後しばらく、コンピュータには蓄積記憶管や水銀遅延線メモリが使われていましたが、やがて安価で小型、高速の磁気コアメモリが現れると、これらのメモリは姿を消しました。しかし磁気コアメモリもまた、1970年に半導体メモリ (DRAM) *21 が発売されると、次第に主役の座を譲っていきます。

*19 ABC、Colossus、ENIACはプログラム内蔵方式ではないので、メモリは演算データの記憶のみに使われます。

*20 真空管でメモリを作るには通常1ビット当たり2本の真空管が必要ですが、ENIACには10本の双三極管でできているリングカウンタが200組あるものとして計算しました。○、×、△の判断に明確な基準はありません。

*21 「DRAM」(52ページ)参照。

4.7. ノイマン型コンピュータ

数学、物理学者として有名なノイマン（John Louis von Neumann）（1903～1957）は、原爆開発プロジェクトであったマンハッタン計画の一員でしたが、1944年、偶然 ENIAC の開発計画を知り、その後 ENIAC のプロジェクトにも加わりました。原爆の開発には膨大な計算が必要で、それを処理できる高速の計算機を求めていたからです。

一方 ENIAC を開発していたモークリーとエッカートは、① ENIAC は 10 進演算方式のためハードウェアが複雑、② プログラムがハードウェアによって与えられているため変更が困難、などの問題があるため、ENIAC の開発を進めるかわら、EDVAC (electric discrete variavle automatic computer) と名づけた次期電子計算機の計画も進めていました。プログラムをメモリに格納するには大容量のメモリが必要です。メモリに水銀遅延線を利用しようというのは、かつてレーダーの研究をしたときに、これを使って成功させたエッカートのアイデアでした。

1945年、ノイマンは EDVAC についてのこうした計画を「EDVAC に関する報告書の初稿 (First Draft of a Report on the EDVAC)」として発表しました。以来、ここに示された 2 進数演算でプログラム内蔵方式のコンピュータは「ノイマン型」と称されるようになりました。

EDVAC の計画はもともとモークリー、エッカートが進めていたもので、ノイマンはそれに数学的な根拠を与えました。しかし常識的にはモークリー、エッカート、ノイマン連名の論文とすべきところを、ノイマン単独の論文として発表されたため、以後ノイマンばかりがコンピュータの発明者扱いされるようになりました。EDVAC 開発プロジェクトには人間関係のもつれが生じ、EDVAC の開発遅れの要因のひとつになりました。

4.8. 最初のコンピュータ

これまで初期の電子計算機をいくつか見てきましたが、それではこのうち「最初のコンピュータ」はどれか、という問題について考えてみます。

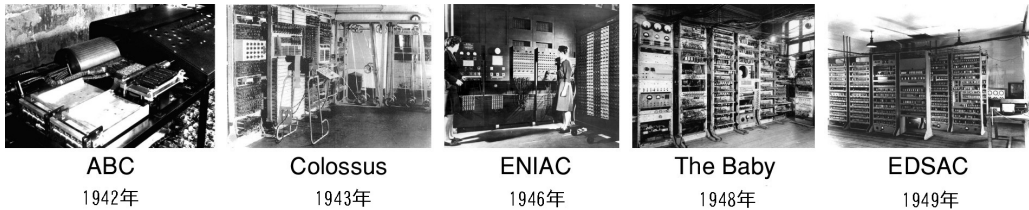


図 5.22 最初のコンピュータはどれ？

一般的には、最初のコンピュータは ENIAC で、コンピュータの歴史は 1946 年に始まるというのが定説になっています。ENIAC は第 2 次世界大戦中、軍事的で開発されていたので、厳重な機密下におかれていましたが、終戦後の 1945 年秋に完成し、1946 年 2 月に機密解除されて公開されました。ENIAC は初めて公表された電子計算機でした。

一方、ABC は 1942 年に作られていますから最初の電子式計算機であることに間違いあり

ませんが、出力装置の動作が一部不安定なまま解決せず、未完成の状態でした。その後アタナソフもベリーも軍の研究に従事するため ABC の開発を中断、特許の申請手続きも戦時の混乱に紛れてうやむやになってしまいました。ABC の存在が知られるようになったのは 1967 年に始まった ENIAC 特許の裁判*22 によってですが、すでに「世界で最初のコンピュータは ENIAC」という常識が定着していました。

しかし ABC は連立方程式を解くための、また Colossus は暗号解読のための専用機で、プログラムは固定、あるいは一部が変更できるだけです。ENIAC は汎用機ですが、プログラムを変更するには大量のケーブルの接続を変更しなくてはなりません。いずれも、現在私たちが使っているコンピュータ（ノイマン型）のイメージとはかけ離れています。

ENIAC が 1946 年 2 月に公表されると、ペンシルバニア大学には技術的な問い合わせが殺到したので、同年 7 月に技術セミナーを開くことにしました。この頃には既に次期コンピュータとして構想されていた EDVAC の仕様も固まりつつあったので、セミナーでは ENIAC ばかりでなく、EDVAC の技術動向もあわせて講義が行われました。EDSAC を作ったウィルクスや、The Baby を作ったウィリアムズもセミナーを受講、あるいは講師であったエッカートを訪れて技術的な指導を受けています。したがって、The Baby も EDSAC も、EDVAC 同様、プログラム内蔵方式の電子計算機として開発・設計されました。

1917/89 Kilburn Highest Factor Routine (amended)

Instruction	C	26	26A	27	Line	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		
-26 C	C_1	-	-	-	1	00011	010																				
-26A C	C_1	-	$-C_1$	-	2	01011	110																				
-26A C	C_1	-	C_1	-	3	01011	010																				
-26A C	C_1	-	C_1	-	4	11011	110																				
-26A C	C_1	-	C_1	-	5	11011	010																				
Sub 27	$a - C_1$				6	11011	001																				
Sub 26	C_1				7	-	011																				
Sub 26	C_1				8	00101	100																				
Sub 26	C_1				9	01011	001																				
Sub 26	C_1				10	10011	110																				
Sub 26	C_1				11	10011	010																				
Sub 26	C_1				12	-	011																				
Sub 26	C_1				13	01011	010																				
Sub 26	C_1				14	10101	001																				
Sub 26	C_1				15	11011	110																				
Sub 26	C_1				16	11011	010																				
Sub 26	C_1				17	01011	110																				
Sub 26	C_1				18	01011	000																				
Sub 26	C_1				19	01011	000																				

20	-3	1011100	23	-a	25	10101	100
21	1	10000	24	C_1	26	-	C_1
22	4	00100			27	-	C_1

or 10100

図 5.23 The Baby の最初のプログラム

プログラム内蔵方式は、ノイマン型コンピュータの中でも最も重要な概念です。プログラムはメモリの中にデータと区別されることなく記憶されていて、コンピュータはメモリに格納さ

*22 ENIAC 特許の権利を持っていたスパーランド社が莫大な特許使用料を請求したため、ハネウエル社は ENIAC の特許は無効であるとして 1967 年に提訴しました。この特許紛争の過程で、モークリーがアタナソフを訪ねて ABC を見学したことが明らかになり、ハネウエル社が勝訴、ABC の存在が世に知られるようになりました。

れたプログラムを読み取りながら逐次的に処理を進めます。別のプログラムを実行するには、新たにプログラムをメモリに読み込むだけで変更できます。これがノイマン型コンピュータの最大の特徴です。そのため「プログラム内蔵方式でないものはコンピュータとはいえない」という考え方もできます。これにしたがえば、ABC も Colossus も ENIAC も、コンピュータではありません。最初のコンピュータは The Baby だということになります。

「電子回路によって計算できるものであればコンピュータと考えていい」とするのなら、最初のコンピュータは ABC ですし、「ABC も Colossus も専用機である。コンピュータとは汎用機でなくてはならない」のであれば、最初のコンピュータは ENIAC でしょう。

しかし、最初のコンピュータはどれかという議論はひとまずおいて、後世に与えた影響がいちばん大きかったのは何かといえば、明らかに ENIAC です。

当時は、信頼性に問題のある真空管を大量に使ってコンピュータを作っても、到底使い物にならないと考えられていました。ENIAC も真空管の故障によるトラブルは少なくありませんでしたが、それでも実用可能なレベルにまとめ上げた功績は特筆すべきで、これによって電子計算機を真空管で作ることも夢ではないことが示されました。

ENIAC の製作中、プログラム内蔵方式、水銀遅延線メモリ、2進数演算を骨子としたノイマン型コンピュータ、EDVAC の基本的な構想をまとめたのもモークリーとエッカートです。The Baby や EDSAC などに影響を与えたペンシルバニア大学ムーア校の技術セミナーの貢献もはかりしれませんし、モークリーとエッカートは後に会社を興して商用コンピュータ UNIVAC を開発し、コンピュータ産業の^{さきがけ}魁ともなりました。

こうしたことも含めて考えれば、「最初のコンピュータ」と呼ぶにふさわしいものは、やはり ENIAC ではないでしょうか。

5. 第2次世界大戦とコンピュータ

これまで見てきたように、リレー式計算機や黎明期の電子計算機の多くは、第2次世界大戦（1939～1945）中、戦争目的のために開発されました。スタイビッツのリレー式計算機やENIACは弾道計算、Colossusは暗号解読が目的でした。

大砲の砲弾が描く軌跡（弾道：trajectory）は、砲身の角度や弾丸や火薬の量はもちろん、その日の温度や風などによっても微妙に変わります。この計算を当時の機械式、リレー式、あるいはアナログ式計算機で行うと、速いものでも数10分かかりました。戦場で計算するわけにはいきませんから、あらかじめ条件を変えて計算した表を作っておかなくてはなりません。新しい大砲や砲弾ができる度に膨大な計算が必要になるので、短時間で弾道計算ができる計算機が求められていました*23。この時代、コンピュータは兵器として開発されたのです。

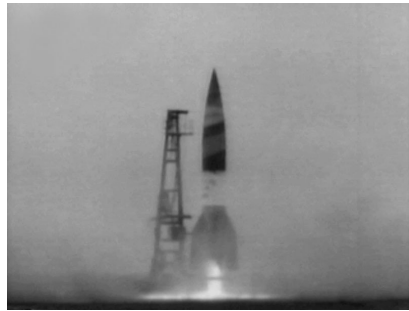


図 5.24 V2

戦争によって科学技術が発達した例は少なくありません。コンピュータ以外にも、第2次世界大戦中に発明、あるいは実用化されたものにレーダー、ロケット、原子爆弾などがあります。

ロケットは古くから中国で武器として使用されていましたが、ドイツのブラウン（Wernher von Braun）（1912～1977）は実用的な液体燃料ロケットV2を開発、1944年には主としてベルギーとイギリスに向けて、爆弾を積んだ数千発のV2が発射されました。正確な誘導ができなかったため爆撃効果は大きくありませんでしたが、超音速で飛来するロケットは警報を出すことも迎撃することもできなかったため、心理的な効果は大きかったといわれています。ドイツ敗戦後、V2の開発に携わった科学者達は米国と旧ソ連に移ってロケットの研究を続け、後の宇宙開発に貢献することになります。

一方、シュレーディンガー（Erwin Schrödinger）（1887～1961）らによる量子力学、アインシュタイン（Albert Einstein）（1879～1955）の相対性理論などの近代的な物理学の理論体系は20世紀初頭に確立されていました。1938年、ドイツのハーン（Otto Hahn）（1879～1968）とシュトラスマン（Fritz Straßmann）（1902～1980）はウランの核分裂を発見、その際質量の一部がエネルギーに変換されるため、何らかの方法で核分裂の連鎖反応を起こすことができれば、莫大なエネルギーを一気に放出できることが分かりました。これが原子爆弾で

*23 ENIACは約20秒で弾道計算を行うことができました（4秒という説もあります）。



図 5.25 広島に投下された原子爆弾
米軍撮影／広島平和記念資料館提供

す。ナチスドイツが原子爆弾を開発することを危惧したアインシュタインをはじめとする科学者達は、ドイツより先に原子爆弾を開発すべきであるという書簡をルーズベルト (Franklin Delano Roosevelt) (1882~1945) 大統領に送りました。こうしてオッペンハイマー (J. Robert Oppenheimer) (1904~1967) をリーダーとする原子爆弾開発プロジェクト、マンハッタン計画がスタートしました。1945年7月16日に人類初の原子爆弾の実験に成功、8月6日に広島にウラン型原子爆弾、8月9日には長崎にプルトニウム型原子爆弾が投下され、8月15日、日本の無条件降伏によって第2次世界大戦は終了しました。

このようにして第2次世界大戦中に生まれた核兵器やロケットの技術は、戦後のコンピュータの歴史にも、少なからぬ影響を及ぼしていくことになります。

6. 商用コンピュータの出現

6.1. UNIVAC I

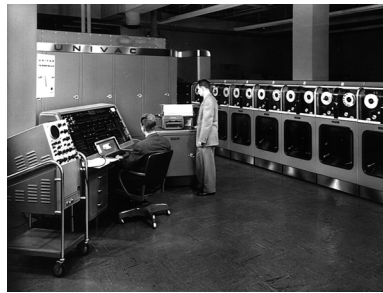


図 5.26 UNIVAC I

ENIAC を開発したモックリーとエッカートは、ペンシルバニア大学を去って*24 エッカー
ト・モックリー社を設立し、コンピュータの開発を続けました。しかし二人は元来学者で、資金
繰りや売買契約の交渉などは苦手です。会社は 1950 年にレミントン・ランド社 *25 に買収され
ました。翌 1951 年に世界初の商用コンピュータ UNIVAC I (universal automatic computer)
を開発し、国勢調査局に納入、以後 56 年までに 39 台が出荷されました。

6.2. IBM 701, 650, System/360

一方、IBM は 1952 年に IBM 701、53 年にはコンパクトな IBM 650 を開発しました。
UNIVAC I の 39 台に対して IBM 650 は 566 台もが出荷され、コンピュータ時代の幕が開き
ました。



図 5.27 IBM 701 (左)、 IBM 650 (中)、 IBM System/360 (右)

写真提供：International Business Machines Corporation.

1964 年、IBM は商用計算、科学技術計算、リアルタイム・アプリケーションなど様々な用
途に適応できる汎用コンピュータ、System/360 を開発、以後数十年間、メインフレーム市場
をほぼ独占する「IBM の時代」を築きました。

*24 大学から ENIAC 特許の譲渡を求めましたが、それを断ったため、大学での研究を続け難くなったようです。

*25 後にスペリー・ランド社となり、現在はユニシス。

7. 米ソ冷戦、宇宙開発競争とコンピュータ

第2次世界大戦の終了後、1949年にはソ連も原爆実験に成功して米国に次ぐ核保有国となり、世界は米国を中心とする自由主義陣営と、ソ連を中心とする共産主義陣営とに二分され、半世紀におよぶ対立が続きました。米ソが直接武力を行使する戦争には至りませんでしたので「冷戦」と呼ばれましたが、朝鮮戦争（1950～1953）やベトナム戦争（1965～1973）のほか、ベルリン危機（1948）やキューバ危機（1962）など、一触即発の事件も少なくありませんでした。

こうした社会背景のもと、コンピュータの歴史もまた、米ソ冷戦の影響を大きく受けることになります。

7.1. スプートニク・ショック

1957年10月4日、ソ連は人類初の人工衛星スプートニク（sputnik）の打ち上げに成功しました。

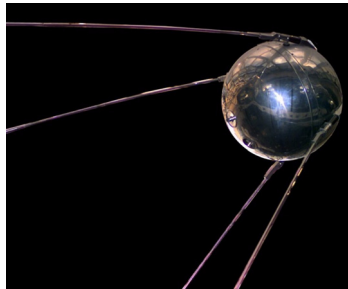


図 5.28 スプートニク

人工衛星は長年の人類の夢を実現させたものではありませんが、同時にまた世界中、とりわけ米国を震撼させました。なぜなら、その打ち上げ用ロケットと制御の技術は、容易に大陸間弾道ミサイル（ICBM : intercontinental ballistic missile）に転用することができるからです。



図 5.29 SAGE のコンソール
写真提供：The MITRE Corporation.

この頃米空軍は、全米をカバーする防空管制システム、SAGE（semi-automatic ground

environment)を開発していました。

各地のレーダーサイトやミサイル哨戒艦などの情報を集め、迎撃用ミサイルや戦闘機に即座に指示を出せるオンラインシステムが完成に近づいていました。しかし、SAGEのコンピュータはニュージャージー州マグワイア空軍基地に設置されていましたが、もしこれをミサイルで攻撃されたら、せっかくの防空システムも一瞬で使用不能になってしまいます。

この年、国防総省に ARPA (高等研究計画局: advanced research projects agency) が設立され、核攻撃にも耐えるネットワークの研究が始まりました。また翌 1958 年には軍事以外の宇宙開発を推進するため、NASA (米国航空宇宙局: national aeronautics and space administration) も設立されました。

7.2. アポロ計画

米ソのロケット開発の中心になっていたのは、主として第2次世界大戦中にドイツでV2を開発していた科学者達ですが、当時の米国のロケット技術はソ連に比べて数年後れていました。スプートニクの翌 1958 年 1 月、米国も人工衛星エクスポラ (explorer) の打ち上げに成功しますが、ソ連は 1959 年にルナ 3 号が月の裏側の写真撮影、1961 年にはガガーリンによる初の有人宇宙飛行に成功するなど、米国に先駆けて画期的な成果を上げていました。

ケネディ大統領 (John F. Kennedy) (1917~1963) は 1961 年、国家の威信をかけたアポロ計画 (Apollo program) を発表し、「1960 年代中に、人類を月に送り込もうではないか」と呼びかけました。

しかし米国のロケットはソ連に比べて推力が劣っていて、重いものを打ち上げることができません。もちろん強力なロケットの開発が必要ですが、積み込む荷物、とりわけコンピュータは小型軽量で高性能なものが求められました。これがコンピュータの小型化を推し進める原動力となりました。

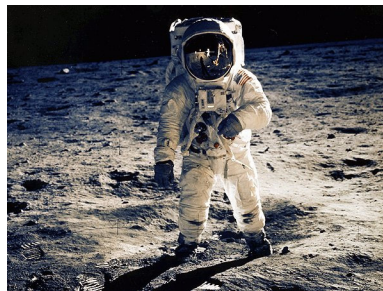


図 5.30 月面に立つアームストロング船長

アポロ計画は 1969 年 7 月 21 日、アポロ 11 号が「静かの海」への着陸に成功し*²⁶、米国の面目は保たれましたが、これを陰で支えたのはコンピュータでした。

*²⁶ 月面に人類としての第1歩を印したアームストロング船長は、“That’s one small step for a man, one giant leap for mankind.” という有名な言葉を残しました。

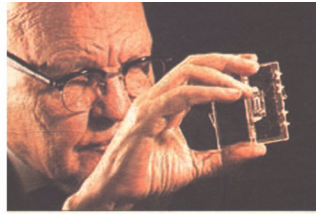
7.3. アポロ計画と半導体技術の発達

1948年、ベル研究所のバーディーン (John Bardeen) (1908～1991) とブラッテン (Walter H. Brattain) (1902～1987) は点接触型トランジスタを発明、翌1949年には同研究所のショックレー (William B. Shockley) (1910～1989) によって接合型トランジスタの理論が発表されました。

当初は安定した品質のトランジスタを作ることができませんでしたが、やがてそれも解決し、1958年にはキルビー (Jack S. Kilby) (1923～2005) とノイス (Robert Noyce) (1927～1990) がそれぞれ集積回路 (IC : integrated circuit) を発明しました。



バーディーン ショックレー ブラッテン



キルビー



ノイス

図 5.31 トランジスタと IC の発明者達

アポロ計画が発表された頃 (1961年) の IC はまだまだ単純なものでしたが、軽くて高性能のコンピュータが必要な NASA にとって、IC はまさにうってつけでした。IC を開発するには多額の開発費がかかるため、出来上がったばかりの IC は非常に高価でしたが、新しい製品は NASA や軍に「金に糸目をつけずに」採用されました。

半導体メーカーにとっては、願ってもない追い風が吹いていたのです。

8. 電卓戦争

ICは、米国では主として軍事・宇宙開発向けの技術として進歩しました。しかしそれだけでは大量の需要は望めず、せつかくのICも広く一般に普及することはありません。自動車もそうでしたが、工業製品は大量生産によって製造コストが下がり、価格が下がることによってさらに需要が増える、という循環が生じてはじめて本格的に普及します。

ICの場合、それは米国ではなく、日本の「電卓戦争」によって始まりました。

8.1. 黎明期の電卓

世界で最初の電卓（電子式卓上計算機）は、1961年にイギリスのBell Punch社が発表したAnita Mark 7です。しかしこれは真空管式の電卓で、生産台数も少なかったようです。




機種	アレフゼロ 101	コンペット CS-10A	キャノーラ 130
			
メーカー	大井電気	早川電機	キャノン
演算素子	パラメトロン (1700 個)	トランジスタ (530 個)	トランジスタ (600 個)
寸法	55 × 52 × 38cm	42 × 44 × 25cm	35.5 × 46.7 × 22.5cm
重量	17.5kg	25kg	15kg
価格	80 万円	53.5 万円	39.5 万円

表 5.3 黎明期の各種電卓

日本でも1964年、大井電気、早川電機（現シャープ）、キャノン各社から相次いで電卓が発売されました。アレフゼロ 101 はパラメトロン*27 を使用していましたが、コンペット CS-10A とキャノーラ 130 はともにトランジスタを使用した最初の電卓でした。しかしそのサイズは4～50cm、重量は20kg程度と大きくて重く、価格も40～80万円と高価なものでした*28。

8.2. 電卓のIC化

早川電機やキャノンが発売したトランジスタ式電卓が評判になり、予想以上に売れはじめること、他のメーカーも続々と電卓市場に参入しました。

しかし電卓を小型・低価格化するするためには、部品点数を減らさなくてはなりません。当初より電卓の小型化構想を持っていた早川電機は、1966年に世界で初めてICを28個使った

*27 東京大学の後藤英一（1931～）により1954年に開発された論理素子。フェライト（酸化鉄を成分とする磁性材料）を用いたもので、寿命は長いが消費電力が大きい。

*28 当時、大卒者の初任給は21,500円でした。

電卓、CS-31A を発売しました。しかしこれにはまだ 553 個のトランジスタが使われていました。同じく早川電機は 1969 年、米ロックウェル社の協力を得て、電卓の機能を 4 個の LSI に集積した LSI 電卓 QT-8D を開発しました。また、1971 年にはビジコンが、米モステック社と共同開発したワンチップ LSI を使った LE-120 を発売しました。

機種	CS-31A	QT-8D	LE-120
発売年	1966 年	1969 年	1971 年
メーカー	早川電機	早川電機	ビジコン
半導体	IC (28 個) + トランジスタ	MOS-LSI (4 個)	MOS-LSI (1 個)
寸法	40 × 48 × 22cm	13.5 × 24.7 × 7.2cm	6.4 × 12.3 × 2.2cm
重量	13kg	1.4kg	300g
価格	35 万円	9.98 万円	8.98 万円

表 5.4 電卓の IC・LSI 化

表 5.4 に示した通り、IC 化・LSI 化されて部品点数が少なくなるにしたがって、小型化、軽量化、低価格化が進んでいく様子が分かります。

こうして電卓の価格は 10 万円を切るまでに下がりましたが、大卒の初任給が 46,400 円 (1971 年) という時代ですから、まだまだ個人が気軽に買えるという価格ではありませんでした。

8.3. マイクロプロセッサの誕生

1971 年に世界初のワンチップ電卓 LE-120 を世に出したビジコンは、1966 年にもコアメモリを採用したメモリ付き電卓、ビジコン 161 *²⁹ を発売するなど、小さいながらもユニークな会社でした。

ビジコンの社長小島義雄はまた、仕様の異なった電卓をそのつど設計するのではなく、電卓の機能は ROM に書き込んでおいて、ROM のデータ、すなわちプログラムを書き換えることによって様々な仕様の電卓を作ることができないかと考えました。新しい電卓を作る度に新たにハードウェアを設計するのは、人材に恵まれた大企業なら可能であっても、ビジコンのような小企業では負担が大きいき、製品価格を下げるのも難しいからです。

この電卓用のマイクロプロセッサ (MPU) の開発を引き受けてくれる半導体メーカーは国内にはありませんでしたので、1970 年にインテル (Intel) と協同開発および独占使用契約を結びました。当時インテルは、IC を発明したノイスが、半導体メモリを作るために 1969 年に

*²⁹ この頃電卓はまだ 40 万円の時代でしたが、コアメモリの採用によってコストダウンされたビジコン 161 は、298,000 円という価格が不当なダンピングだとして通産省が発売中止を申し入れたほどでした。

作っただけの会社でした。まだ経営が軌道に乗らず仕事も少なかったため、日本の中小企業にすぎないビジコンと契約することになりました。しかしインテルもビジコンも、そのときは、このMPUがどれほど大きな意味を持っているかということには気づいていませんでした。



図 5.32 i-4004 と、それを使った電卓 141-PF

ビジコンのアイデアをもとに、インテルのテッド・ホフ (Marcian E. “Ted” Hoff) (1937～) が4ビットのマイクロプロセッサを考案、ビジコンから派遣された嶋正利*³⁰ (1943～) が論理設計を行い、ファジン (Federico Faggin) (1941～) が回路の設計を担当して、初のマイクロプロセッサ i-4004 が1971年3月に誕生しました。10月には i-4004 を搭載したプリンタ付き電卓、141-PF (159,800 円) がビジコンから発売されました。

ビジコンとインテルの契約には、両社が共同でMPUを開発すること、開発費としてビジコンが10万ドル支払うこと、製品はビジコンが独占販売権を持つこと、製品出荷のスケジュールや価格などが取り決められていました。しかしその後、ビジコンは熾烈な価格競争に疲れ、オイルショックによる円高で輸出が激減したことなどから、資金繰りに苦しむようになりました。一方インテルは、開発したマイクロプロセッサは電卓以外の用途に広く使えることから、開発費の返却と製品の値下げを条件に、ビジコンから i-4004 の外販許可を得ました。

ビジコンは1974年に倒産*³¹しましたが、独占販売権の放棄がなければ、今頃はインテル同様「世界のビジコン」の可能性もあっただけに、不運なできごとでした。

8.4. 電卓戦争の終焉^{しゅうえん}

この頃、電卓の標準的な価格は4万円前後になっていました。しかし当時の大学卒の初任給は46,400円です。電卓はまだ業務用、オフィスで計算をするためのものでした。

電卓を一般の人々に、^{そろばん}算盤がわりに使ってもらいたいと考えたのはカシオでした。そのために、電卓の価格を1万円にしようという目標がたてられました。1万円電卓を実現するために、機能も最小限に絞って開発されたのが1チップの6桁電卓「カシオミニ」でした。個人が計算に使うのなら、6桁でも100万円まで計算できるんだからまあいいんじゃないか、というわけ

*³⁰ 嶋正利は i-4004 を開発した後も、インテルから請われて8ビットのMPU i-8080、ファジンが設立したザイログ社に移ってZ80、16ビットのZ8000など、数多くのMPUの開発に携わりました。

*³¹ 一時不渡り手形を発行して倒産しましたが、ビジコンはその後債務を返済して再建されています。

機種	カシオミニ	EL-805	LC-78
発売年	1972年	1973年	1978年
メーカー	カシオ	シャープ	カシオ
特徴	低価格	初の液晶表示	名刺サイズ
寸法	146 × 77 × 42mm	78 × 118 × 20mm	91 × 55 × 3.9mm
重量	315g	200g	39g
価格	12,800円	26,800円	6,500円

表 5.5 電卓の低価格化、小型・薄型化

です。発売価格は12,800円。1万円という目標は達成できませんでしたが、それでもこれまでの $\frac{1}{3}$ の価格です。カシオミニは大ヒットし、他の電卓メーカーに大きな衝撃を与えました。

この頃シャープは、薄型電卓への道を探っていました。当時の電卓には表示素子として蛍光表示管やLEDが使われていました。いずれも消費電力が大きいため単三電池が使われていて、電卓の小型化、特に薄型化の制約となっていました。

シャープが省電力化をはかるために注目したのは液晶^{*32}でした。液晶は材料の選択や配合が難しく、寿命も短いため実用化は望めないと考えられていましたが、電卓の表示装置として液晶ディスプレイの開発に成功し、EL-805が発売されました。必ずしも低価格ではありませんでしたが、単三乾電池一本で100時間も使える画期的な特徴によって、これも大ヒットしました。電卓の小型軽量化、低価格化競争はますます激しくなり、1978年にはついに名刺サイズで厚さ3.9mmというLC-78（カシオ）が現れました。

そして電卓が、半導体技術、生産技術、低価格化、小型軽量化、省電力化など、あらゆる面で行き着くところまで行ったときに、気がつけばこれまで50社以上が参入していた電卓業界に、残ったのはシャープとカシオだけでした。こうして、熾烈^{しれつ}を極めた電卓戦争が終わりました。

米国ではもっぱら軍事・宇宙開発に使われていたICですが、日本では電卓という民生品に大量に使われて半導体産業を育て、同時にマイクロプロセッサも生み出しました。日本の電卓戦争は、結果的に、コンピュータの歴史に大きな影響を与えました。

*32 「液晶ディスプレイ」(69ページ) 参照。

9. パーソナルコンピュータの誕生

インテルとビジコンが開発した i-4004 は 4 ビットのマイクロプロセッサでした。数値計算にはこれで十分ですが、文字データ (7~8 ビット) を扱うには無理があります。インテルはコンピュータの端末に使用するために、8 ビットのマイクロプロセッサ 8008 を 72 年に、74 年にはこれを改良した 8080 を開発しました。一方モトローラ^{*33} (Motorola) はミニコンピュータのアーキテクチャを MPU 化した 8 ビットの MC6800 を 1974 年に発表、ザイログ^{*34} (Zilog) も 1976 年に 8080 に上位互換性のある Z80 を開発し、1970 年代に主要な 8 ビットマイクロプロセッサが出揃いました。

それ以前は個人がコンピュータを所有することなど到底考えられませんでした。マイクロプロセッサの出現によって、「パーソナルコンピュータ」も夢ではない時代がやってきました。

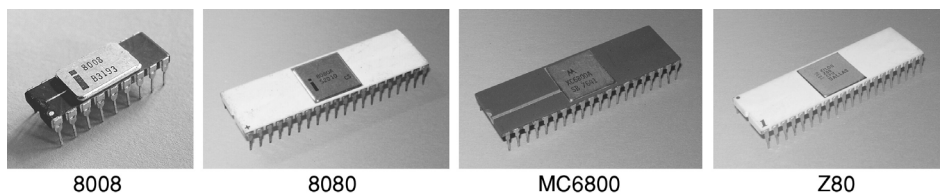


図 5.33 初期の 8 ビットマイクロプロセッサ

9.1. ALTAIR 8800

世界で最初のパーソナルコンピュータ ALTAIR 8800 は 1975 年、MITS (Micro Instrumentation and Telemetry Systems) 社から発売されましたが、最初は組み立てキットでした。インテルのマイクロプロセッサ 8080 の他、コンピュータに必要な部品一式がキットになっていて、ユーザーは電子部品を一個一個半田付けして組み立てなくてはなりません。

本体正面のパネルにはスイッチと LED がついているだけで、2 進数の 0 はスイッチを OFF、1 は ON にして、直接 2 進数でプログラムやデータを入力するというものでした。



図 5.34 ALTAIR 8800

本体は 439 ドルですが、オプションを含めると 1,000 ドル^{*35} 程度となり、個人の買い物と

^{*33} 米国の通信機器、半導体メーカー。1979 年に 16 ビットの MC68000、84 年に 32 ビットの MC68020 を発売。

^{*34} インテルのファジンらがスピンアウトして 1974 年に興した半導体メーカー。

^{*35} 当時の 1 ドルはほぼ 300 円だったので、1,000 ドルは約 30 万円。

してはかなり高額でしたが、それでも ALTAIR はマニアの間で好評を博し、3 カ月で 4,000 台の注文があったといわれています。

ビル・ゲーツ (William 'Bill' H. Gates) (1955～) とポール・アレン (Paul G. Allen) (1953～) は、BASIC *³⁶ を ALTAIR 8800 上で動くように改造し、これが大ヒットしました。彼らはマイクロソフト社を設立、BASIC は初期のパーソナルコンピュータの標準プログラミング言語となり、広く普及しました。

9.2. Apple I

1976 年、ウォズニアック (Steve Wozniak) (1950～) は自作のコンピュータを出入りしていたアマチュアコンピュータクラブ*³⁷ に持ち込みました。ボード一枚のコンピュータです。キーボードや CRT ディスプレイに接続することができ、BASIC も装備されていたので評判になり、たちまちコンピュータクラブの仲間に広まりました。ジョブス (Steve P. Jobs) (1955～2011) はこれを事業化しようとして Apple 社を設立しました。ジョブスは愛用のフォルクスワーゲンを、ウォズニアックもヒューレットパッカードの高級電卓を売って、プリント基板を作るための資金 1,300 ドルを作ったという話は、シリコンバレーの伝説になっています。

最初はボードのみでしたが、後には図 5.35 のような木製ケースに収められて、のべ 200 台近くが出荷されたといわれています。



図 5.35 Apple I

出典 : Computer History Museum 撮影 : Jean-Baptiste Queru

9.3. TK-80

Apple I が作られた 1976 年、日本では日本電気がマイクロコンピュータのトレーニング用組立キット、TK-80 *³⁸ を発売しました。これもワンボードのコンピュータで、モニタという、簡単な OS のようなプログラムが ROM に書き込まれていました。ボード上の 16 進キーボードから入力したり、7 セグメント LED 表示器*³⁹ に 16 進数で表示ができましたので、ALTAIR

*³⁶ 初心者向けのプログラミング言語。「BASIC」(81 ページ) 参照。

*³⁷ homebrew computer club (Palo Alto, California)

*³⁸ 価格は 88,500 円。当時の大卒初任給は 94,300 円。

*³⁹ 日の字型に配置された 7 つの LED の発光を制御して数字を表示するデバイス。電卓の表示器と同じ形式ですが、*R. b. C. d. E. F.* を使えば 16 進数を表示させることができます。

と Apple I の中間くらいの仕様だったと考えればいいのでしょうか。

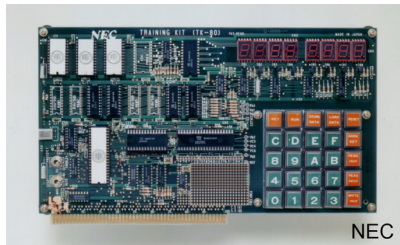


図 5.36 TK-80
写真提供：NEC

9.4. Apple II

1977年には Apple II が発売されました。Apple I はマニア向けのワンボードコンピュータでしたが、Apple II はキーボードが本体に組み込まれていて、モニターまたは家庭用 TV を接続するとカラー表示ができるなど、使いやすい工夫がされていました。カラーといっても、40 × 48 ピクセルの表示モードで 15 色、280 × 192 ピクセルの高解像度モードでも黒、白、紫、緑の 4 色表示ですから、現在のパソコンや携帯電話とは比べものにもなりません、これでも当時としては画期的な仕様でした。Apple II 以後、パーソナルコンピュータ本体にはキーボードがつくというスタイルが定着しました。



図 5.37 apple II

9.5. PC-8001

1979年、日本電気は初のパーソナルコンピュータ PC-8001*40 を発売しました。キーボード付きの本体に、オプションとして CRT ディスプレイ、プリンタ、フロッピーディスクドライブ、カセットテープレコーダ*41などを接続することができました。スイッチを入れると

*40 本体定価は 168,000 円。この年の大卒初任給は 109,500 円。

*41 当時、フロッピーディスクはまだまだ高嶺の花でしたので、個人ユーザーはもっぱらプログラムをオーディオ用カセットテープに保存（録音）していました。カセットテープは遅くて不便で不安定でしたので、プログラムだけならともかく、保存されたデータを参照するような本格的な使い方には向きませんでした。

BASIC が起動するようになっていて、BASIC のコマンドを入力して操作する、というのが一般的な使い方でした。PC-8001 は 2 年間で 12 万台を出荷するヒット商品となり、日本でのパーソナルコンピュータブームに火をつけました。



図 5.38 PC-8001

しかし当時の 8 ビットマイクロプロセッサ（クロック周波数 4MHz、最大メモリ容量 64k バイト）では、できることは限られています。ほとんどの場合、ゲームを中心とした、ホビー用コンピュータの域を出ませんでした。

9.6. IBM PC

Apple II が成功し、パーソナルコンピュータが脚光を浴びてくると、大型汎用コンピュータの雄 IBM も、1981 年、急遽ビジネス用 16 ビットパーソナルコンピュータを発売しました。



図 5.39 IBM PC

しかし開発を急いだためか、マイクロプロセッサはインテルの 8088、OS はマイクロソフトの MS-DOS と、主要部分はいずれも他社製品を採用した、IBM としては異例の製品でした。しかし IBM PC は技術情報が公開され、他のメーカーが互換機や拡張ボードなどの市場に参入できたこともあって、後にパーソナルコンピュータの標準機となりました。そのため、IBM PC はインテルとマイクロソフト両社の躍進のきっかけとなりました。

9.7. PC-9801

PC-9801 は、日本電気が 1982 年に開発した 16 ビットパーソナルコンピュータ、PC-9800 シリーズの初代機です。日本語が扱いやすかったため、ビジネス市場を中心に広く受け入れら

れました。一時は「国民機」と言われるほどの圧倒的なシェアを占め、他のメーカーによって「PC98互換機」さえ作られました。ソフトウェアや周辺機器メーカーをはじめとするパーソナルコンピュータ産業の形成にも貢献しましたが、一方では圧倒的なシェアを占めたことによって、機能や価格面での企業間の競争が正常に機能しないといった弊害も生じました。



図 5.40 PC-9801

しかしソフトウェアの進歩とともに日本語処理上の優位性が次第に失われ、1997年、日本電気自身もIBM PC互換のパーソナルコンピュータを発売、事実上PC-9800シリーズの歴史の幕を閉じました。

9.8. Macintosh

Macintoshはアップルが1984年に発表した、GUI (graphical user interface) を採用した最初のパーソナルコンピュータ^{*42}です。



図 5.41 Macintosh

これまでのパーソナルコンピュータのOSはCUI^{*43} (character user interface) で、コンピュータを使うには定められたコマンドをキーボードから入力しなくてはなりません。コンピュータやOSについての知識が必要で、誰にでも気軽に使えるものではありませんでした。

^{*42} 最初にGUIを採用したコンピュータは、1973年にXEROXのパロアルト研究所で作られたワークステーション、Altoです。スティーブ・ジョブズは1979年にこれを見学、Macintosh開発の参考にしました。なお、Macintoshに先だって、初めてGUIを採用したLisaというパーソナルコンピュータが1983年にアップルから発売されましたが、約1万ドルという価格設定もあって、商業的には失敗しています。

^{*43} コンピュータの操作をキーボードで行なうユーザインターフェース。「GUI」(92ページ)参照。

Macintosh はマウスでアイコン (icon) をクリックするだけで操作でき、コンピュータは難しいもの、という常識を覆くつがえしました。

MS-DOS を供給していたマイクロソフトも、GUI 化を図った OS、Windows 1.0 を 1985 年に発表しましたが、本格的に普及しはじめたのは Windows 3.1 (1992 年) や Windows 95 (1995 年) が発売された頃からでした*44。

かつてはコンピュータといえば、ハードウェアやソフトウェアに精通した技術者にしか扱えないものでしたが、OS の進歩によってその障壁が次第に取り払われ、小学生からお年寄りまで、誰にでも使えるパーソナルコンピュータが生まれてきました。

*44 Macintosh に比べて、Windows が実質的に 10 年近くも後れをとったのは、当時インテルの CPU には大容量のメモリアクセスへの制約が多かったことも要因のひとつです。

10. インターネットの成立

10.1. ARPANET

1957年、ソ連が人工衛星スプートニクの打ち上げに成功すると、ソ連からの核ミサイル攻撃の脅威に震撼した米国防総省は、^{エアパ}ARPA（高等研究計画局）を設立して通信やコンピュータネットワークの研究に着手しました。

また、1961年にはユタ州の電話中継局がテロにより爆破されました。軍用回線にも被害が及び、米軍の防空管制システム SAGE *45 で採用されているような集中型のネットワークでは、中心のコンピュータが破壊されるとネットワーク全体の機能が瞬時に停止することが問題になりました。こうして、「核戦争にも耐えるネットワーク」の研究が始まりました。

1964年、ランド社のバラン（Paul A. Baran）（1926～2011）は、集中型でも非集中型でもなく、データが多数の中継局を経由して送られていく分散型のネットワークに関する論文、“On Distributed Communications”を発表しました。分散型ネットワークでは、中継局がいくつか破壊されても、情報は生き残っている中継局を経由して目的地まで届きます。アナログ通信方式では中継を繰り返すと信号が劣化するため、デジタル通信方式を採用し、データを一定の長さに分割して取り扱うパケット通信の概念も折り込まれました。

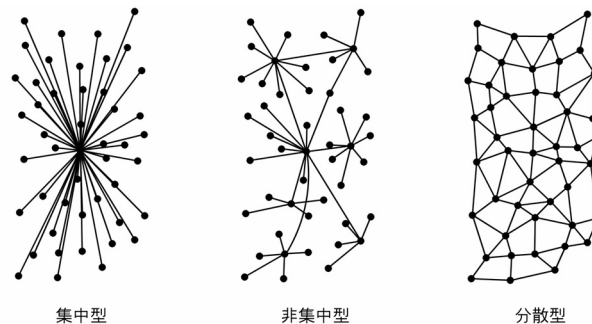


図 5.42 ネットワークの種類

しかし、コンピュータのネットワークといっても、同一メーカーの同一機種同士ならともかく、メーカーが違い、機種が異なると、それらのコンピュータの間でデータを交換するのは容易ではありません。外国語のできない日本人と中国人、スペイン人、ドイツ人が集まって、いきなり会議をしようというようなものです。

そこで、IMP（interface message processor）と名づけられたミニコンピュータをそれぞれのコンピュータの前に置いて、そうしたネットワーク上の諸問題を一手に引き受けて解決させることにしました。各国語に堪能な通訳を一人づつつけたわけです。

また、分散型ネットワークでは、送られてきたパケットを中継しなくてはなりませんが、こ

*45 semi-automatic ground environment. 「スプートニク・ショック」（127 ページ）参照。

の仕事も IMP が行います。パケットには目的地を表すデータが付けられていて、IMP はこれによって次はどの中継局に送ればよいかを調べて送り出します。それぞれの IMP が次の中継局へパケツリレーのようにパケットを中継することによって、パケットは自動的に目的地に到着します。

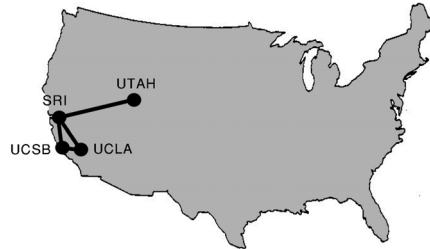


図 5.43 ARPANET (Dec. 1969)

こうして 1969 年 12 月、カリフォルニア大学ロサンゼルス校 (UCLA: University of California Los Angeles)、同サンタバーバラ校 (UCSB: UC Santa Barbara)、スタンフォード大学 (SRI: Stanford Research Institute)、ユタ大学 (University of Utah) の 4 つの大学のコンピュータがネットワークで結ばれ、インターネットの前身、ARPANET が誕生しました。

1971 年になると ARPANET には 23 台のホストコンピュータが接続され、1973 年には初めて国外、ロンドン大学 (UCL: University College of London) のコンピュータが接続され、その後もホストコンピュータの数は 1981 年に 213 台、1984 年に 1,000 台を越え、1987 年には 10,000 台を突破するなど、すさまじい勢いで成長し始めました*46。

10.2. WWW

インターネットに接続されているホストコンピュータの数は順調に増え続けましたが、情報閲覧するには、ファイルが保存されているコンピュータのアドレスやファイル名、コマンドなどを正確に入力しなければならないなど、当時のインターネットの使い勝手は必ずしも良くありませんでした。

1989 年、^{セルジュ}CERN *47 のティム・バーナーズ・リー (Tim J. Berners-Lee) (1955～) は、そうした煩わしさを解消するために、新たに情報閲覧システムを開発して WWW (world wide web) と名付けました。利用者がファイル名などを入力しなくても、あらかじめ文書の中にそうした情報を埋めこんでおいて、利用者がマウスで指示するだけで、あたかもそれが入力されたのと同じように扱われる仕組みを作ったのです。このような情報が埋めこまれた文書を、ハイパーテキスト (hyper text) といいます。

*46 「ホストコンピュータ数の推移」(144 ページ) 参照。

*47 欧州核物理学研究所 (European organization for nuclear research)



図 5.44 バーナーズ・リー

これによって、リンク^{*48} (link) をマウスでクリックすれば即座にその情報が表示される、現在のウェブページ閲覧のスタイルが生まれました。

10.3. WWW から Mosaic、Netscape へ

WWW は、マウスでクリックするだけで関連文書が表示されるという点では画期的でしたが、処理できるのは文字データだけでした。しかも WWW は NeXT ^{*49} という、あまり普及していないコンピュータで動くように作られていました。

イリノイ大学 NCSA^{*50} のマーク・アンドリーセン (Marc Andreessen) (1971～) は、WWW を参考にして、画像データも扱えるようにしたブラウザソフト、モザイク (Mosaic) を開発し、1993 年に公開しました。



図 5.45 モザイク (version 3.0 for Windows)

画像が表示できるモザイクはたちまち普及しましたが、あまりにも反響が大きかったため、それまで学生たちが自主的に作っていたモザイクに大学が関与するようになりました。これを嫌ったマーク・アンドリーセンは、卒業後 NCSA を去って、ジム・クラーク ^{*51} (James H. Clark) (1944～) と共にネットスケープ・コミュニケーションズ社を作り、1994 年、モザイ

^{*48} 文書内に埋め込まれた他の文書へのアドレス情報。通常ブラウザ画面上では、アンダーラインで示されます。

^{*49} Apple 社を作ったスティーブ・ジョブズが興した会社製のコンピュータ。GUI インターフェイスを採用。

^{*50} (米) 国立スーパーコンピュータ応用研究所 (national center for supercomputing applications)

^{*51} 1982 年にコンピュータグラフィックス専用コンピュータの会社、シリコン・グラフィックスを設立。後に同社を引退し、1994 年にネットスケープ・コミュニケーションズを設立しました。

クに代わる新たなブラウザソフト、ネットスケープ・ナビゲータ (Netscape Navigator) を開発しました。

ネットスケープ・ナビゲータは大好評を博し、たちまち 80% のシェアを獲得、ブラウザの定番ソフトとなりました。

10.4. ブラウザ戦争

WWW からモザイク、ネットスケープ・ナビゲータと、ブラウザソフトが進歩するにつれて、ウェブの情報は次第に利用しやすくなりました。また、ウェブに蓄積された情報も次第に充実してくると、インターネットの重要性はますます高くなっていきました。

ブラウザはインターネットを利用する上で最も重要なソフトウェアです。インターネットへの進出に遅れをとっていたマイクロソフトは、急遽ブラウザソフトの spyglass 社を買収し、1995 年にインターネット・エクスプローラ (Internet Explorer) を開発しました。しかしそれだけでは、すでに 80% 以上のシェアを持っているネットスケープ・ナビゲータには敵いません。何としてもインターネットでのイニシアティブを、と考えたマイクロソフトがとった戦略は、圧倒的な資金力にものをいわせてインターネット・エクスプローラを無料配布することと、Windows に統合された形で同梱することをコンピュータメーカーに強制することでした。^{*52}

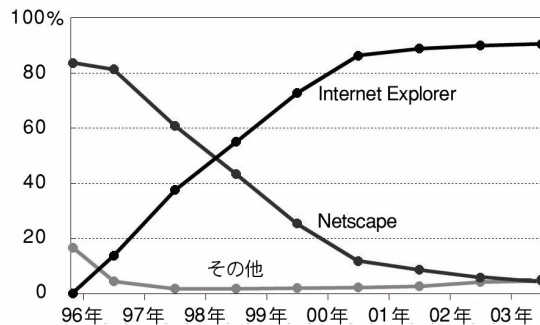


図 5.46 ブラウザのシェアの推移

「無理を通せば道理が引込む」のは世の常、ブラウザのシェアは図 5.46 ^{*53} の通り、1998 年に主客逆転したまま現在に至っています。この当時のブラウザのシェア争いは、「ブラウザ戦争」とも呼ばれました。

10.5. インターネットの普及

インターネットは国防総省の下部組織、ARPANET からスタートしましたが、1986 年に NSF (national science foundation : (米) 国立科学財団) が NSFNET を新設し、ARPANET

^{*52} そのような強引なやり方は独禁法に違反するとして、米司法省などから提訴されました。

^{*53} 出典：第 17 回 CSJWWW 利用者調査結果 (<http://www.csj.co.jp/www17/>)。

は本来の軍用ネットワークに戻りました。そして1991年、それまで学術研究専用であったインターネットの商用使用禁止制限を廃止し、一般の企業などもインターネットを利用できるようになりました。

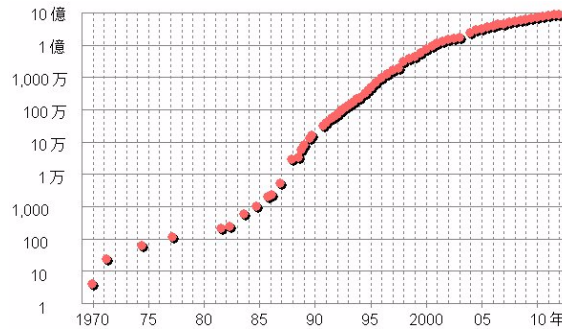


図 5.47 ホストコンピュータ数の推移（対数）

1989年にバーナーズ・リーが開発したWWWを1991年にCERNがリリース、1993年にNCSAがモザイクを公開、1994年にはネットスケープ・ナビゲータの開発と、この頃はインターネットに関する重要な出来事、ソフトウェアの開発が集中しています。1995年にはWindows 95が発売されて、パーソナルコンピュータの普及が本格化しました。図5.47はインターネットに接続されているホストコンピュータ数の推移を対数グラフで、図5.48は通常のグラフで表したものです*54。ホストコンピュータの数が、この時期から急激に増え続けている様子が分かります。

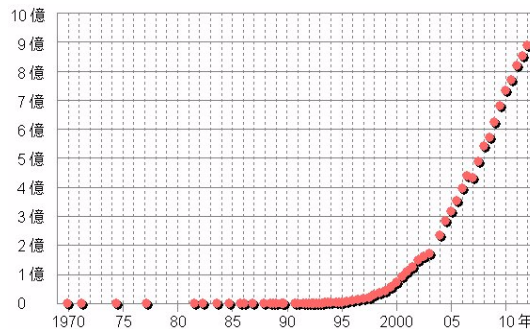


図 5.48 ホストコンピュータ数の推移

以後もホストコンピュータ数はほぼ一定の割合で増え続け、2012年には約8.9億台ものホストコンピュータが接続されに至りました。

*54 出典：Hobbes' Internet Timeline (<http://www.zakon.org/robert/internet/timeline/>)
The ISC Domain Survey (<https://www.isc.org/solutions/survey>)

第 6 章 ネットワークとインターネット

1. ネットワーク

現在はコンピュータをネットワーク（network）に接続して使うのが当たり前ですが、以前はそうではなく、コンピュータは単独で使われていました。そのようなコンピュータの利用形態をスタンドアローン（stand alone）といいます。

しかし、コンピュータにどれだけ処理能力があり、記憶容量が大きくても、一台のコンピュータにできることには限りがあり、複雑な業務や大量のデータをスタンドアローンのコンピュータで効率よく処理するには無理があります。コンピュータは次第にネットワークを構成し、互いに通信回線で接続して使われるようになりました。

1.1. LAN

LAN（^{ラン}local area network：構内情報通信網）は、同一の敷地内に置かれているコンピュータの間を私設の通信回線で結んで、データの伝達・交換ができるようにしたものです。

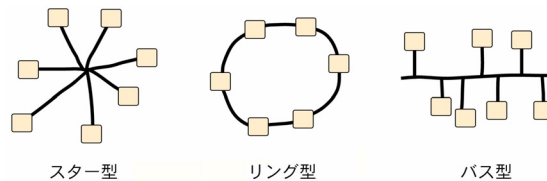


図 6.1 LAN のトポロジー

LAN の接続形態（トポロジー：topology）にはスター型、リング型、バス型などがあります。いずれの形態をとるにしても、ひとつの通信回線を多くのコンピュータで共用することになるので、コンピュータがめいめい勝手にデータを送信したのでは、ケーブルの中でデータが衝突して通信できなくなります。これを防ぐために、何らかの方法でデータの交換を制御しなければなりません。これをメディアアクセス制御（MAC：media access control）といい、代表的なものに CSMA/CD（carrier sense multiple access with collision detection：搬送波感知多重アクセス／衝突検出方式）や、トークンパッシング（token passing）などがあります。

1.1.1. イーサネット

イーサネット（ethernet）は、現在最も普及している LAN の制御方式です。1973 年にゼロックス、インテル、DEC の 3 社が共同開発した、CSMA/CD という通信線制御方式が使わ

れています。

CSMA/CD は早い者勝ち方式で、データを送信しようとするコンピュータは、通信線が使用中かどうかをチェックし、空いていればデータを送り出すことができます。たまたま複数のコンピュータが、同時に空いていると判断して送信するとデータが衝突 (collision) しますが、そのときは互いに送信を止め、乱数で決めた待ち時間後に再度送信します。

イーサネットはデータの衝突を敢えて容認し、データが衝突しても、その影響をできるだけ少なくしようという通信方式です。衝突が起これば通信効率は下がりますが、ネットワークの構築や管理が簡単なため、現在イーサネットはほとんどの LAN で採用されています。

イーサネットのトポロジーにはバス型とスター型の2種類があります。大規模な LAN では両方を併用することがありますが、スター型は細くて扱いやすいツイストペアケーブル (twisted pair cable)*¹ を使用し、ハブ (hub) と呼ばれる集線装置を使って容易にネットワークを構築できることから、現在では家庭内でもイーサネットの LAN が使われています。

1.1.2. トークンパッシング

イーサネットがデータの衝突を容認するのに対して、トークンパッシングはデータの衝突そのものを避けようという考え方です。

鉄道で、列車の衝突を防ぐために、タブレットと呼ぶ通行票を使うことがあります。線路を駅ごとなどの一定の区間に区切り、タブレットを持った列車だけがその区間を走ることができます。タブレットはその区間には1個しかないので、ひとつの区間を2台以上の列車が走ることはありません*²。

トークンパッシング方式のネットワークでは、通信回線の中を常時トークン (token) と呼ばれるパッケージが回っています。トークンは鉄道のタブレットのような、データを送信してもよいという「送信権」です。データを送信したいコンピュータは、自分のところにトークンが回ってくるのを待って、トークンにデータを付けて送信します。送信が終わって相手のコンピュータにデータが届いたことを確認したら、トークンだけにして再びネットワークに流します。

ネットワーク中にトークンはひとつしかないので、データの衝突が起こることはありません。通信回線を効率よく使用できますが、何らかの原因でトークンが消滅すると、ネットワーク全体が停止してしまうという欠点もあります。

トークンパッシング方式は、コンピュータをリング状に接続したトークンリング (token ring) や、光ファイバーを使って高速通信ができる FDDI (fiber-distributed data interface) などで使用されています。

*¹ 2本の電線を撚りあわせた通信用ケーブル。撚りあわせることによって、平行線よりもノイズの影響を受けにくくなります。電話線に使われているのもツイストペアケーブルです。

*² 現在でもローカル線の駅などで、タブレットの交換を見ることができます。

1.1.3. クライアントサーバシステム

LANの目的はデータの伝達・交換ですが、LANを構築するとプリンタやハードディスク、あるいはその中に保存されているデータベースなどを共用することができます。したがって、LANを構成するコンピュータ群の中に、ハードディスクを管理する専用のコンピュータや、プリンタを管理する専用のコンピュータを設け、これらが他のコンピュータのために、その仕事を一括して行うようにしたものをクライアントサーバシステム (client server system) といいます。ここで、ハードウェアやデータなどの資源を管理するコンピュータをサーバ (server)、その資源を利用するコンピュータをクライアント (client) といいます。

また小規模のLANで、サーバを置かず、すべてのコンピュータがサーバでありクライアントでもあるものをピアツーピア (peer to peer) *³ といいます。

1.1.4. パケット

コンピュータネットワークは何台ものコンピュータが通信回線を共用しているので、あるコンピュータとコンピュータとの間でデータ通信が行われている間は、他のコンピュータが通信回線を利用することはできません。回線の使用中は、他のコンピュータは通信が終わるまで待たなくてはなりません。このような不都合を防ぐために、データを小さいブロックに分割して通信する方法が考えられました。小さいブロックそれぞれに宛先や送り主などを示す情報を付け加えたものをパケット (packet)、データをパケットに分割して送受信する通信方式をパケット通信 (packet communication) といいます。

パケット1個の通信に要する時間はごくわずかなので、パケットが送信されても通信回線が独占されることはありません。たくさんのコンピュータが通信回線を共有できるので、回線の利用効率は飛躍的に向上します。

何両も連結した列車が専用の線路を走れば、多数の乗客をダイヤ通りに運ぶことができますが、「線路」そのものの利用効率は必ずしも良くありません。一方、乗客がバスや車に分乗して道路を走れば、予定通りの時刻に到着しないかもしれませんが、同じ道を他の車も走れるため、「道路」の利用効率は高くなります。

情報を乗客にたとえれば、電話のような回線交換方式*⁴ は電車に、パケット通信はバスや車で乗客を運ぶ方式に似ています。

1.1.5. 無線LAN

LANの通信回線には一般的にツイストペアケーブル*⁵ (twisted pair cable) が使われていますが、多数のコンピュータが接続されるようになると太いケーブルの束が張り巡らされたり、

*³ peer は仲間、対等者の意。ここではデータを交換する装置をピアといいます。

*⁴ 通信を始める前にあらかじめ回線を接続して情報を伝送する方式。一度接続すれば回線を独占できるので通信が途切れることはありませんが、データ通信の場合情報は間欠的に発生することが多いので、情報が流れていないときも回線を占有しているため、回線の使用効率が悪くなります。

*⁵ 2本の電線を撚り合わせた通信用ケーブル。撚り合わせることによってノイズの影響を受けにくくなります。

第6章 ネットワークとインターネット

ノート型コンピュータの移動が面倒など、ケーブル接続の煩わしさを感じる事が多くなりました。

無線 LAN (wireless LAN) は、無線 (電波) を利用してデータ通信を行う LAN システムです。当初は通信速度が 2^{メガビット・秒} *6 と遅く、機器も高価だったためなかなか普及しませんでした。が、^{アイティプロバイダー} IEEE *7 によって高速の規格が順次標準化され、低価格の機器も現れたため次第に普及してきました。

規格	策定	最大伝送速度	周波数
IEEE802.11a	1999 年	54Mbps	5.2GHz 帯
IEEE802.11b	1999 年	11Mbps	2.4GHz 帯
IEEE802.11g	2003 年	54Mbps	2.4GHz 帯
IEEE802.11n	2009 年	300Mbps	2.4GHz/5GHz 帯
IEEE802.11vht	策定中	1Gbps	6GHz 以下/60GHz 帯

表 6.1 無線 LAN の各種規格

伝送距離はいずれも 50~100m 程度で、電波の性質上周波数が高いほど伝送距離が短く、また障害物の影響を受けやすくなります。また 2.4GHz 帯は電子レンジなど無線 LAN 以外の機器にも使用されているため、混信やノイズにより伝送速度が低下しやすい傾向があります。どのメーカーの機器とも相互に接続できることが認定された無線 LAN 機器には、^{ワイ・ファイ} Wi-Fi (wireless fidelity) という名称がつけられています。

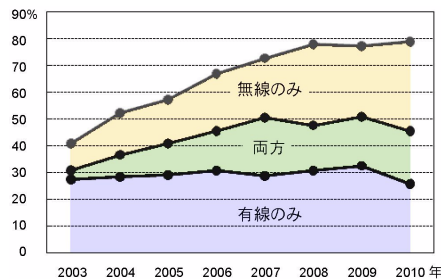


図 6.2 家庭内 LAN の構築率の推移

図 6.2 はパーソナルコンピュータを 2 台以上保有している世帯での、家庭内 LAN の構築率の推移を表しています*8。有線のみ LAN の構築率はほぼ一定で推移して、2010 年にはやや減少しているのに対して、無線 LAN は伸びが目立ちます。2010 年の無線のみの利用者は 33.3%、両方は 19.6%なので、無線 LAN の利用者は合わせて 52.9%、有線のみ 25.7%の 2 倍以上になっています。

*6 bps (bits per second) は通信速度の単位。「主な単位」(227 ページ) 参照。

*7 電気電子学会： Institute of electrical and electronic engineers

*8 出典：総務省 平成 22 年通信利用動向調査。

近年スマートフォンやタブレット型コンピュータなど Wi-Fi 対応機器の普及が著しいことから、LAN の無線化傾向は今後も進むものと思われます。

1.1.6. プロトコル（通信規約）

コンピュータのネットワークを介してデータの通信を行うには、あらかじめ取り決めておかなければならないことがたくさんあります。たとえば、どんなコネクタやケーブルを使うか、相手のコンピュータをどうして識別するか、通信データの誤りをどう検出して修復するか、電子メールの件名や本文などをどんな風に送信するか、などです。このような、コンピュータネットワークでデータ通信を行う際の様々の取り決めをプロトコル (protocol *9) といいます。

プロトコルは、ネットワークを利用しているすべてのコンピュータで共通でなくてはなりません。ワークステーション同士ではうまくいっても、パーソナルコンピュータとは通信できないとか、あるメーカー製のコンピュータにはつながらない、というようなことがあつては困ります。もし乾電池がメーカー毎に規格がバラバラで、ある懐中電灯には A 社の乾電池、ある器具には B 社の専用乾電池しか使えなければ、とても不便なことは容易に想像できます。初期のコンピュータのデータ通信は、まさにそのような状態でした。

そこで国際標準化機構、ISO (international organization for standardization) は、異なる機種間でもデータ通信ができるようにするために OSI (open system interconnection : 開放型システム間相互接続) と呼ばれるプロトコルの規格を定めました。

ネットワークに多数のコンピュータが接続され、その間で相互にパケット通信を行うには複雑なプロトコルが必要になります。そこでプロトコルを階層化し、機能を分割して単純化することになりました。これを OSI 基本参照モデル (OSI basic reference model) といいます。

層	名 称	機 能
7	アプリケーション層	アプリケーションに特化した処理
6	プレゼンテーション層	データフォーマットの設定
5	セッション層	通信の管理、コネクションの確立と切断
4	トランスポート層	確実にデータを送るための管理
3	ネットワーク層	アドレスの管理と通信経路の選択
2	データリンク層	データフレームの識別と転送
1	フィジカル層	機械的・電氣的な物理的条件の設定

表 6.2 OSI 基本参照モデル

OSI 基本参照モデルは表 6.2 のような階層構造になっています。それぞれの層は下位層からサービスを受け、上位層にサービスを提供します。こうすれば、アプリケーション層のプロト

*9 protocol はもともと、外交儀礼や条約議定書といった意味です。たとえば 1997 年の地球温暖化防止のための協定、京都議定書は Kyoto protocol といいます。

コルは、そのアプリケーションについてのプロトコル、たとえばメールソフトは電子メールの件名や本文を送信するためのプロトコルだけを考えればよく、下位の、プレゼンテーション層でのデータフォーマットや、セッション層での通信そのものの管理などまで考慮する必要はなくなります。

1.2. WAN

LAN が比較的狭い、同一の敷地内で利用される局所的なネットワークであるのに対して、WAN (wide area network : 広域通信網) は異なる敷地間、たとえば大学の各キャンパス、企業の本社と支社、銀行の本店と支店などに構築された LAN 同士をつなぐ、広域のコンピュータネットワークです。

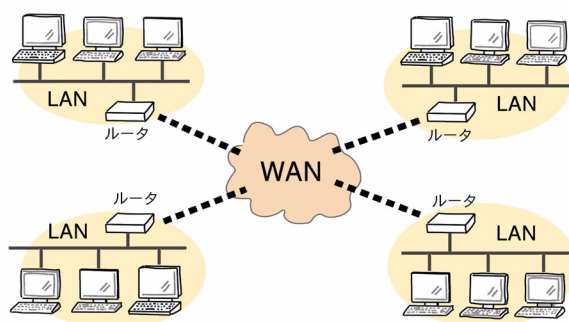


図 6.3 WAN

同一の建物や敷地内にあるコンピュータでネットワーク (LAN) を作るのなら、私設の通信ケーブルを自由に張りめぐらせることができますが、たとえ道ひとつでも離れた敷地には勝手にケーブルをつなぐことはできません。同じネットワークでも、同一敷地内でない場合は電気通信事業者*10 の設備や通信回線を利用することになります。こうして大学であれば各キャンパス、企業であれば本社、支社、工場などをつないだ広域のネットワークができます。これを WAN といいます。

1.3. PAN

PAN は personal area network の略で、個人が使用する携帯電話やデジタルカメラなどの機器をパーソナルコンピュータと接続するための、通常は数メートル程度の範囲のネットワークです。接続には USB (universal serial bus) などの有線や、ブルートゥース (bluetooth) や IrDA*11 (infrared data association) といった近距離無線技術が利用されます。

*10 電気通信事業を営む事業者として登録されるか、届出をした者。NTT や KDDI、携帯電話や CATV の事業者など。

*11 同名の団体 (IrDA) が定めた、赤外線を利用した近距離のデータ通信の規格。

2. インターネット

こうして大規模なネットワーク、WAN が作られるようになりましたが、WAN はあくまでも大学や企業など、特定の組織の LAN をつなぐ「閉じた」ネットワークです。やがてさらに、これらの LAN や WAN 同士をつなぐ「開かれた」広域ネットワーク、インターネット (Internet) が生まれました。

2.1. インターネットの現況

1969 年、米国の国防総省の ARPA^{*12} は、^{アーパネット}ARPANET と呼ばれるコンピュータの広域ネットワークを構築しました。最初は 4 つの大学^{*13} のコンピュータを結ぶことからスタートし、次第に全米の大学、研究所を結ぶネットワークへと発展しました。この ARPANET がインターネットの前身ですが、現在 (2011 年) は表 6.3 の通り、世界中で 22 億余りの人々に利用されるまでに成長しました^{*14}。

地 域	利用者数 (万人)	普及率 (%)
アジア	101,679.9	26.2
アフリカ	2,392.7	13.5
ヨーロッパ	50,072.4	61.3
中東	7,702.1	35.6
北アメリカ	27,306.8	78.6
中南米	23,582.0	39.5
オセアニア	2,392.8	67.5
計	226,723.4	32.7

表 6.3 インターネットの利用者数、普及率

表 6.3 によると、地球人口 70 億人の約 $\frac{1}{3}$ がインターネットを利用していますが、最も普及率の高い北アメリカと、最も低いアフリカでは普及率に約 6 倍の格差があります。そして国別に普及率を見ていくと、普及率の格差は更に大きくなります。

図 6.4 はインターネットの普及率を国別に表した地図です。インターネットの普及率が高いのは北アメリカやヨーロッパ、オセアニアとアジアのごく一部の地域だけで、その他の地域では全般的に遅れていること、普及率が 1% に満たない国や地域も少なくないことが分かります。

国別に見ると普及率のトップはフォークランド諸島で 95.8%、2 位はアイスランドで 95.0%、3 位はノルウェーの 93.4%、以下オランダ 90.7%、ルクセンブルク 90.6%、スウェーデン 90.0%、デンマーク 88.7%、フィンランド 86.9%、イギリス 85.0%、バミューダ諸島 84.2% と

^{*12} advanced research projects agency. 高等研究計画局。「ARPANET」(140 ページ) 参照。

^{*13} UCLA、UCSB、スタンフォード大学、ユタ大学。「ARPANET」(141 ページ) 参照。

^{*14} 出典：Internet World Stats (2011 年) (<http://www.internetworldstats.com>)

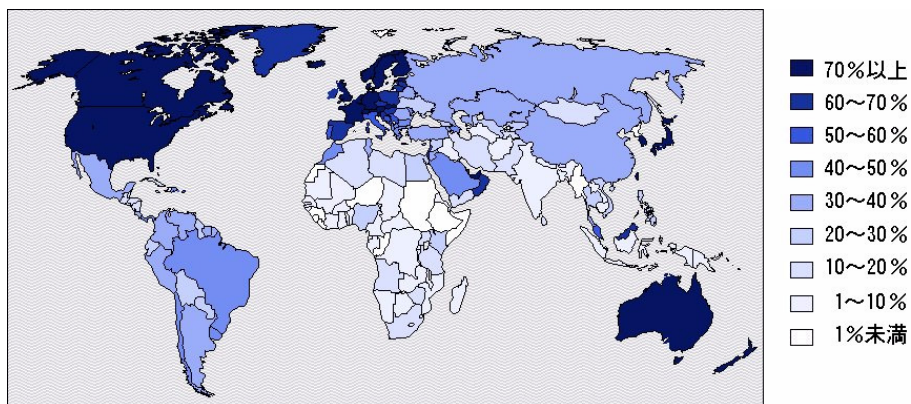


図 6.4 インターネットの国別普及率（2007 年）

続きます。これがインターネット普及率のベストテンで、日本は 76.8%、19 位です*15。

主な国々のインターネット普及率の最近の推移を見たのが図 6.5 です。日本を含めて普及率が高いインターネット先進諸国と、普及率は半分以下ですが、近年経済成長が著しい BRICs 諸国*16 を選んであります。

2007 年にオーストラリアの普及率が 20%以上も下がっているなど、集計方法の変更のためか不自然な箇所がいくつか目につきますが、全体としては各国とも、普及率が年々順調にのびています。

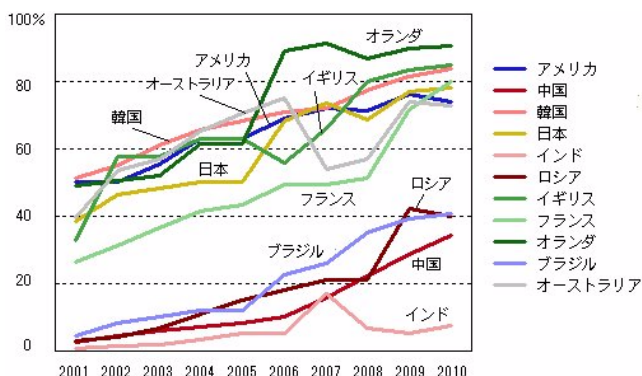


図 6.5 主な国のインターネット普及率の推移

図 6.6 はこれを、2001 年の各国の普及率を 1 として表したものです。こうすると、2001 年を基準にして、各国の普及率がどれだけ増えたかがよく分かります。

図 6.5 と比べてみると、米国、韓国、日本など、インターネット先進諸国は 2001 年時点ですでに普及率が 20%~50% に達しているため、増加率は数倍が限度です。それに対して、BRICs

*15 出典：ITU (2010 年) (<http://www.itu.int/ITU-D/icteye/Indicators/Indicators.aspx#>)。表 6.3 とは集計年度に差があります。

*16 ブラジル (Brazil)、ロシア (Russia)、インド (India)、中国 (China) の頭文字を並べた名称。

諸国は普及率こそ高くありませんが、この期間に 10 倍程度に増加しています。

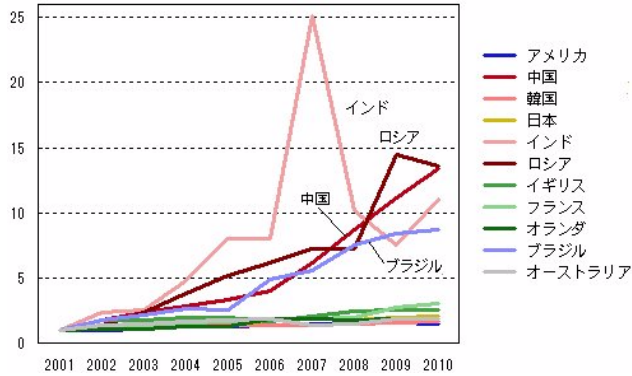


図 6.6 2001 年を 1 とするインターネット普及率の推移

中国の人口は 13 億人余で世界一、インドは約 12 億人で第 2 位、ブラジルは 1.8 億人で第 5 位、ロシアも 1.4 億人で第 7 位の大国です。こうした国々がインターネットの普及率を着実に伸ばしていることから、世界のインターネットの地図は大きく塗り替えられつつあることが分かります。

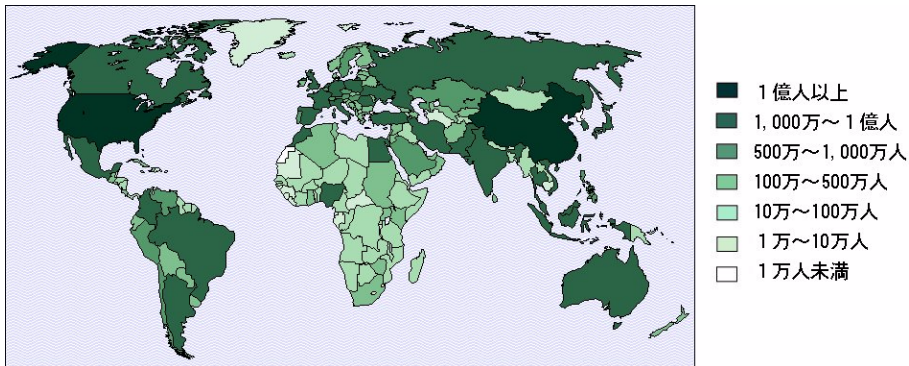


図 6.7 インターネットの国別利用者数 (2007 年)

利用者数で塗り分けてみたのが図 6.7 です。

利用者数では中国が 3.84 億人でトップ、米国が 2.4 億人で 2 位、日本は 9,914 万人で 3 位です。ブラジルが 7,594 万人、ドイツが 6,512 万人、インドは 6,130 万人、ロシア 5,970 万人、イギリス 5,144 万人、フランスは 4,463 万人でナイジェリアが 4,398 万人。これが利用者数で見たベストテンです。

図 6.5 同様、主な国々の利用者数の推移を表したものが図 6.8 です。

インターネットの利用者数は長年アメリカが首位でしたが、2008 年に中国が追い越しました。日本は 2002 年までアメリカに次いで 2 位でしたが、2003 年に中国に抜かれて 3 位になり

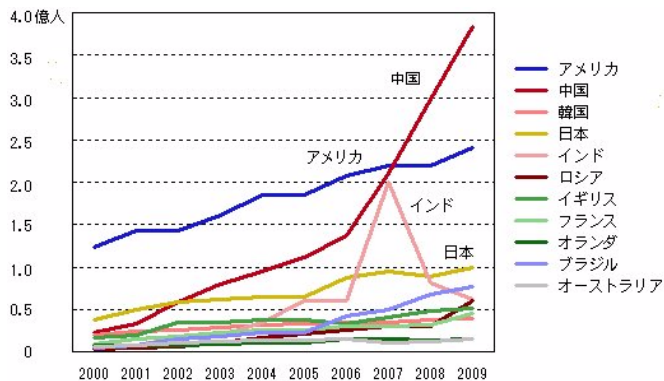


図 6.8 主な国のインターネット利用者数の推移

ました*17。中国やインドの人口はアメリカの約4倍、日本の約10倍もあります。このような国々では、インターネットの利用者がまだまだ増え続ける可能性があります。国別のインターネット普及率や利用者数の地図は、今後どんどん塗りかえられていくことと思われます。

しかしその一方で、普及率も利用者数もまだまだ低い国々が少なくありません。インターネット利用者数が0という国や地域もあります。多くは南太平洋のサモア、ナウル、パラオ、北マリアナ諸島など、中央アメリカのオランダ領アンティルやタークス・カイコス諸島などの島国ですが、アジアでは北朝鮮もそうです。さらに、西サハラ*18は諸国から国家として承認されておらず、国連でも「自治政府を持たない領域」とされているため、このような統計に上ってくることもありません。

WWWが本当の意味での“World Wide Web”になる日が、一日も早くやってきてほしいものです。

2.2. インターネットへの接続

コンピュータをインターネットに接続するには、基本的にはプロバイダ*19 (ISP: Internet services provider) と契約し、何らかの通信線によってプロバイダのサーバと接続します。

プロバイダと接続する通信線には電話、ISDN、CATV、ADSL、光ファイバなど、多くの選択肢があります。したがって、逆に、まず通信会社を選択して契約し、プロバイダは通信会社が推薦するものの中から選ぶのが一般的になりました。

通信線で情報を送るには、放送と同じような搬送波*20 (carrier) を使いますが、この搬送波には一定の幅が必要です。たとえば「800kHzのラジオ放送」と言いますが、この800kHzの電波(搬送波)に音声信号を乗せて放送するには、搬送波の両側に数kHzの幅が必要です。

*17 2007年にはインドにも抜かれていますが、これはインドの集計方法の一時的な変更のためと思われます。

*18 アフリカ大陸の左肩部、図6.7で利用者数1,000人未満となっている地域。

*19 家庭や企業のコンピュータをインターネットに接続するサービスを業務とする会社。

*20 情報を乗せて伝送するための信号で、数100kHz～数GHzの電磁波の他、光も利用されます。

これを帯域幅 (bandwidth) といいます。これがないと、ラジオでチューニングはできても肝心の音が出せません。情報を送るには搬送波に帯域幅が必要です。

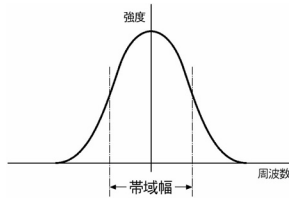


図 6.9 帯域幅

電話やラジオなどの音声信号に比べて、情報量が多いテレビは帯域幅も広くなりますが、通信の場合も通信速度が速くなればなるほど広い帯域幅が必要になります。ADSL や CATV、FTTH などがこれに相当し、これらを総称してブロードバンド (broadband) といいます。

2.2.1. 電話回線、ISDN 回線

電話や ISDN (integrated services digital network : 統合デジタル通信網) などの公衆回線を使ってインターネットに接続することを、ダイヤルアップ接続 (dial-up connection) といいます。

電話回線はアナログ回線なので、コンピュータのデジタル信号はそのままでは電話回線で通信できません。そのため、モデム (MODEM : modulator-demodulator) を使ってアナログ信号に変換します。モデムの最高通信速度は 56 ^{キロビット毎秒} kbps ですが、通信線のノイズ等の環境に応じて自動的に通信速度を下げて、通信が安定するように調整しているため、必ずしも最高速度で通信できるわけではありません。

ISDN は、FAX やコンピュータ間のデータ通信はもちろん、アナログである電話の音声信号も、すべてデジタル信号で取り扱います。64kbps の通信回線が 2 回線同時に使用できるので、電話をかけながらインターネットに接続したり、あるいは 2 回線を束ねて 128kbps で通信することもできます。

日本で ISDN のサービスが始まった当時 (1988 年)、高速のデジタル通信として脚光を浴びました。しかし電話回線も ISDN も回線交換方式のため、料金は回線接続時間に対して課金されます*21。インターネットが普及しはじめた頃はよく利用されましたが、ウェブページを時間を気にせず、ゆっくり読むのには適していません。ADSL、ケーブルテレビなどのブロードバンド通信が普及するにつれて、利用者は次第に減少しました。

*21 「パケット」(147 ページ) 参照。月々定額料金を支払うことによって、通信料金が定額になるサービスもあります。

2.2.2. ADSL

ADSL (asymmetric digital subscriber line : 非対称デジタル加入者線) は、加入者線 (subscriber line) を利用したデジタル信号の伝送方式です。

「加入者線」とは家庭から電話局までの電話線のことで、ADSLは電話線という電線を利用して通信します。電話やISDNのように電話回線をつぐのではなく、搬送波に乗せたコンピュータの信号を多重化して通信します。そのためインターネットを利用していても通話料金はかかりません。コンピュータのスイッチを入れると、いつでもインターネットが利用できる常時接続になり、もちろん同時に電話も使用できます。

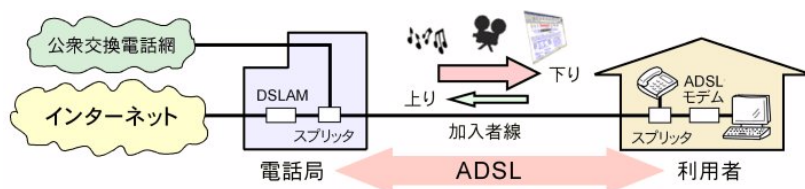


図 6.10 ADSL

「非対称」とは、電話局から利用者への方向（下り）の通信速度と、利用者から電話局への方向（上り）の通信速度が対称（同じ）でないことを意味しています。家庭のコンピュータをインターネットに接続する場合、利用者から送り出される情報（上り）は主として閲覧したいページの URL *22 などですが、それに対して送られてくる情報（下り）は、テキスト、静止画、音声、動画と多種多様で、データ量は上りに比べてはるかに多くなるのが常です。そのため、上下方向の通信速度を同じにしておくよりも、下り方向の速度を速くする方が回線を効率的に利用できます。道幅が限られているなら、交通量が多い下り車線を太くするのがいい、というわけです。

ADSL のもうひとつの特徴は、ISDN の数百倍という通信速度です。当初は 1.5 Mbps^{メガビット毎秒}でしたが、次第に 12Mbps、24Mbps と高速化され、50Mbps というサービスも始まっています。しかし、電話線はもともと音声通話のためのもので、このような高速のデータ通信を保証するものではありません。最良の状態でも最高速度の 80% 程度、通常はもっと遅くなると考えた方がよさそうです。電話局から遠いほど通信速度が落ちる傾向があり、長距離になると通信できないというケースもあります。また、ADSL は加入者線という電線に信号を多重化していますから、電話局までの通信線の中に光ファイバーが含まれている場合も利用できません。

2.2.3. CATV

ケーブルテレビ (CATV : cable television) の通信線を利用してインターネットにアクセスすることもできます。

本来 CATV とは community antenna television であって、たとえば山のかげや離島などテ

*22 インターネットの情報の格納場所を指定する表記法。「URL」(167 ページ) 参照。

テレビの電波が弱い地域で、受信条件の良い地点に共同のアンテナを設け、ケーブルで各家庭にテレビ信号を届ける難視聴地域対策でした。

しかし、ここでいう CATV は都市型 CATV (city CATV) ともいわれ、多チャンネルのテレビ番組の他、インターネット接続サービスなども提供するものを指しています。CATV の通信線はテレビ信号伝送用の同軸ケーブルを使用しているため、ADSL のように距離によって通信速度が変わることはありません。しかし数十 Mbps の回線を何名かで共同で使用するため、利用者が増えると通信速度が遅くなる場合があります。

2.2.4. FTTH

FTTH (fiber to the home) は光ファイバ (optical fiber) による情報通信です。

“Fiber to the home” とは、1991 年に米国のゴア上院議員が提唱した「情報スーパーハイウェイ構想」に基づくもので、一般家庭にまで光ファイバを引いて、高速の情報通信網を構築しようという概念を表すキャッチフレーズでした。

日本では 2001 年に光ファイバによる常時接続サービスがスタートしました。通信速度は主として 100Mbps *23、他の通信回線よりも高速でノイズの影響を受けないなど、数々の特徴があります。新たに光ファイバの回線網を構築しなければならないため、既存の電話回線を利用できた ADSL に比べると普及のペースは速くありませんでしたが、現在は着実に利用者が増えつつあります。最終的には、通信は将来、すべて光ファイバに移行するものと考えられています*24。

光ファイバのしくみ

光ファイバといえば、胃カメラを連想する人も多いでしょう。光はまっすぐ進むのが常識なのに、曲がっているファイバを通して胃の内部が見えるのは本当に不思議です。

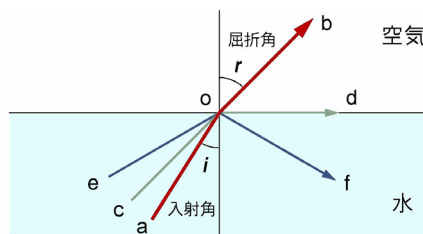


図 6.11 光の屈折と全反射

光ファイバは、光の全反射 (total reflection) の性質を利用して光を伝えます。光は空気と水、水とガラスのように、異なる媒質の境界面で進行方向を変えます。これを屈折 (refraction) といいます。図 6.11 のように、光が水中の a から上方に進んでいるものとする、界面 o で

*23 10Mbps から、速いものでは 1Gbps というものもあります。

*24 「ブロードバンドサービスの利用状況」(158 ページ) 参照。

屈折してbに向かって進みます*25。このとき、入射角*i*と屈折角*r*の正弦(sin)の比は屈折率と呼ばれる一定の値になります。

入射角*i*が大きくなると屈折角*r*も大きくなりますが、屈折角*r*が90°になるとき(c-o-d)の入射角*i*を臨界角といいます。水の、空気に対する臨界角は約48.6°です。入射角が臨界角より大きくなると、光は水面で全て反射します(e-o-f)。これを全反射といいます。

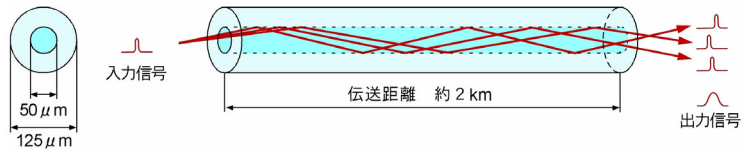


図 6.12 マルチモードファイバ

図 6.12 は通信用の光ファイバのひとつ、マルチモードファイバ (MMF : multi mode fiber) です。直径は 125 μm 、髪の毛ほどの太さのガラス繊維ですが、屈折率が異なる二種類のガラスからなっています。屈折率が大きい内側の部分をコア (core)、屈折率が小さい外側の部分をクラッド (cladding) といいます。光はコアとクラッドの境界で全反射を繰り返しながら進むので、ファイバ全体が曲がっていても、光はファイバに沿って進みます。光がコアから洩れることはありません。

2.2.5. ブロードバンドサービスの利用状況

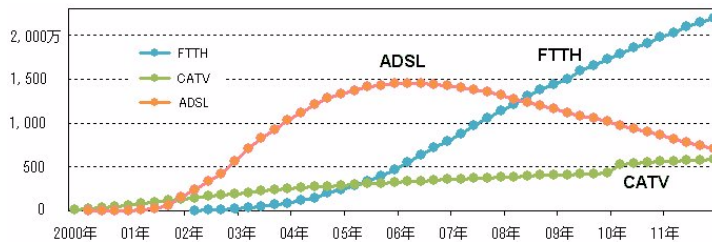


図 6.13 ブロードバンドサービス契約数の推移

出典：総務省情報通信統計データベース

図 6.13 はブロードバンドサービスの契約数の推移を表しています。ブロードバンドの先鞭をつけたのは CATV ですが、ADSL の利用料金が 2001 年 9 月に大幅に下がったため、一気に ADSL が普及しました。しかし FTTH が本格的に普及し始めると 2006 年 3 月をピークにして減少に転じ、2008 年 6 月には FTTH が逆転、現在 (2011 年末) は FTTH アクセスサービスの契約数が約 2,200 万件に達しています。

*25 光の一部は反射します。



2.3. IP アドレス

IP アドレス (internet protocol address) は、インターネットに接続されているコンピュータにつけられている番号です。家庭の電話や携帯電話に電話番号がつけられているのと同じこと、と思えばいいでしょう。

世の中に、同じ番号の電話がいくつもあると困りますが、IP アドレスも同じです。世界中で IP アドレスが重複することがないように、^{アイキャン}ICANN (internet corporation for assigned names and numbers) によって管理されています*26。

IP アドレスは 32 桁の 2 進数です。たとえば、11011011011110100011110110010011。

これを、人がこのままで扱うのは大変ですから、8 桁毎に区切って 10 進数に置き換えます。最初の 8 桁、11011011 は 10 進数では 219、次の 01111010 は 122、以下同様に 00111101、10010011 は 61、147 となります。

11011011011110100011110110010011

 11011011 . 01111010 . 00111101 . 10010011

 219 . 122 . 61 . 147

この 219 . 122 . 61 . 147 が、人が扱うときの IP アドレスです。

IP アドレスは 32 ビットですから、約 43 億台のコンピュータを識別することができます*27が、すでに約 8.9 億台のホストコンピュータがインターネットに接続されています*28。32 ビットでは到底十分とはいえませんので、現在 IP アドレスをもっと確保するために、128 ビットに拡張した新しい IP プロトコル、IPv6 の開発、運用が始まっています。

2.4. ドメイン

さきほどの「219 . 122 . 61 . 147」は、関西外国語大学のウェブサーバの IP アドレスです。2 進数より親しみやすくなりましたが、やはり数字では覚えにくいし、紛らわしくて不便です。そこで数字ではなく、分かりやすいように、コンピュータにつけられた名前がドメイン (domain) です。

たとえば関西外国語大学のウェブサーバのドメインは www.kansai.ac.jp です。これを使って、ブラウザのアドレス欄に <http://www.kansai.ac.jp/> と入力すれば、関西外国語大学のウェブページにアクセスできます。

ドメインもピリオドによっていくつかの部分に分けられていますが、右から順にトップレベルドメイン、第 2 レベルドメイン、第 3 レベルドメイン… といいます。

*26 日本では JPNIC (Japan network information center) がこれを管理しています。

*27 n 桁の 2 進数は 2^n 種類の数字を表すことができます。 $2^{32} = 4,294,967,296$ 。「2 進数」(16 ページ) 参照。

*28 「インターネットの普及」(143 ページ) 参照。



2.4.1. トップレベルドメイン

トップレベルドメイン (TLD : top level domain) には、分野別トップレベルドメイン (gTLD : generic TLD) と国名コードトップレベルドメイン (ccTLD : country code TLD) の2種類があります。

a. 分野別トップレベルドメイン (gTLD)

分野別トップレベルドメインは、ドメインを取得する組織の分野によって定められています。

com	商業組織	edu	米国教育機関
net	ネットワーク管理組織	gov	米国政府機関
org	非営利団体	mil	米国軍事機関

表 6.4 主な分野別トップレベルドメイン

たとえば、スタンフォード大学は stanford.edu、ホワイトハウスは whitehouse.gov、米国防総省なら defenselink.mil、というわけです。

表 6.4 は主な分野別トップレベルドメインですが、他にも aero (航空運輸)、biz (ビジネス)、cat (カタロニア文化コミュニティ)、coop (協同組合)、info (制限なし)、int (国際機関)、jobs (人事管理業務)、mobi (モバイル関係)、museum (博物館、美術館等)、name (個人)、nato (北大西洋条約機構関係)、pro (弁護士、医師、会計士等)、travel (旅行関連業界) などがあります*29。

b. 国名コードトップレベルドメイン (ccTLD)

国名コード (country code) は ISO 3166-1 (日本では JIS X 0304) で規定されている、国や地域を表すコードです。2文字コードと3文字コード (たとえば日本は jp と jpn) がありますが、2文字のコードがトップレベルドメインに使われています。

アメリカ合衆国の国名コードは us ですが、米国のドメインにはほとんど使用されません。また、イギリスの国名コードは gb ですが ccTLD は uk *30、などの例外もあります。

*29 新たに post (郵便事業)、tel (IP 電話番号)、xxx (アダルトサイト)、asia (アジア太平洋地域の法人、団体等)、mail (スパムフリーメールの送受信者) なども登録されようとしています。

*30 イギリスの正式国名はグレートブリテンおよび北部アイルランド連合王国 (The United Kingdom of Great Britain and Northern Ireland)。

au	オーストラリア	fr	フランス共和国	ph	フィリピン共和国
be	ベルギー王国	hk	香港特別行政区	ru	ロシア連邦
ca	カナダ	in	インド	se	スウェーデン王国
ch	スイス連邦	jp	日本	to	トンガ王国
cn	中華人民共和国	kr	大韓民国	uk	イギリス
de	ドイツ連邦共和国	lk	ラオス人民民主共和国	vn	ベトナム社会主義共和国
es	スペイン	nl	オランダ王国	za	南アフリカ共和国

表 6.5 ccTLD の例

2.4.2. 第 2 レベルドメイン

日本の第 2 レベルドメイン (SLD: second level domain) にも「属性型ドメイン」と「地域型ドメイン」の 2 種類があります。

a. 属性型ドメイン

属性型ドメイン (organizational type JP domain names) は、組織の種類によって、表 6.6 のように定められています。

ac	高等教育機関等の学校法人
ad	JPNIC 会員
co	企業等の営利法人
ed	18 歳未満を対象とする教育機関
go	政府機関、特殊法人
gr	任意団体
lg	地方公共団体
ne	不特定の利用者へのネットワークサービス
or	非営利法人

表 6.6 属性型ドメインの種類

たとえば高等教育機関の第 2 レベルドメインは ac ですから、関西外国語大学は kansaigaidai.ac.jp、同様にして首相官邸なら kantei.go.jp などとなります。なお、米国には mil という gTLD がありますが、日本には建前上、mil に相当するものではありません^{*31}。

b. 地域型ドメイン

地方公共団体やその機関等は地域型ドメイン (geographic type JP domain names) をつけることができます。

^{*31} 自衛隊のドメインは jda.go.jp です。インターネットはもともと米国防総省が軍事目的で開発したものですから、米国には当然 mil があります。「ARPANET」(140 ページ) 参照。

pref	道府県	例：pref.nagano.jp（長野県）
metro	東京都	例：metro.tokyo.jp（東京都）
city	政令指定都市 東京都特別区 市	例：city.osaka.jp（大阪市） 例：city.setagaya.tokyo.jp（世田谷区） 例：city.chikuma.nagano.jp（千曲市） 例：city.chino.lg.jp（茅野市） higashiosaka.osaka.jp（東大阪市）等も可
town	町	例：town.karuizawa.nagano.jp（軽井沢町）
vill	村	例：vill.hakuba.nagano.jp（白馬村）

表 6.7 地域型ドメイン（地方公共団体）の例

また一般の企業や団体も、組織ラベル・市町村ラベル・都道府県ラベル.jp の形で地域型ドメインが使用できます。たとえば大阪府枚方市の〇×株式会社なら、ox.hirakata.osaka.jp などとすることができます。

2.4.3. 第3 レベルドメイン

第3 レベルドメイン（third level domain）は、申請する者が決める組織の名称です。

第3 レベルドメインは自由に決めることができますが、ドメイン名登録機関（レジストリ：registry）に登録申請しなくてはなりません。ドメイン名もIP アドレス同様重複があってはいけないので、原則として早い者勝ち方式で受け付けられます。

IP アドレスやドメイン名の標準化や割り当てはICANNが行っていて、その登録機関、レジストリはTLD ごとにあります。日本のccTLDを管理しているのは株式会社日本レジストリサービス（JPRS：Japan registry service）ですが、指定事業者（レジストラ：registrar）を経由して申請することもできます。

2.4.4. 第4 レベルドメイン

第4 レベルドメイン（fourth level domains）はサーバ名です。関西外国語大学の場合、ウェブサーバのドメイン名はwww.kansaigaidai.ac.jpですが、この“www”がウェブサーバにつけられた名前、第4 レベルドメインです。ウェブサーバの名前はwwwにする例が多いようですが、もちろん他の名前でも構いません。

2.4.5. 汎用jpドメイン

第2 レベルドメインは属性型ドメインでは組織の属性、地域型ドメインでは都道府県名が使われていますが、汎用jpドメインではここに任意の名称を登録することができます。登録できるドメイン名の数や内容に制限はなく、組織名の他にも商品名やイベント名でも登録でき、日本語も使えます。

汎用JPドメイン名は2001年に導入された新しい制度で、これまでドメインの登録は1組

織 1 ドメインに限られていましたが、こうした制約も大幅に緩和されています。

2.5. DNS

ドメイン名が登録されると、たとえば“`http://www.kansai.ac.jp/`”^{*32} をウェブブラウザのアドレス欄に入力すれば関西外国語大学のウェブサーバにアクセスでき、ウェブページが表示されます。しかし、ドメインは単に人が扱いやすいようにつけた名前前で、コンピュータは IP アドレスでないと理解できません。実際にサーバにアクセスするには、これをコンピュータに付けられている番号、IP アドレスに変換しなくてはなりません。

これを変換するのが DNS (domain name system) です。

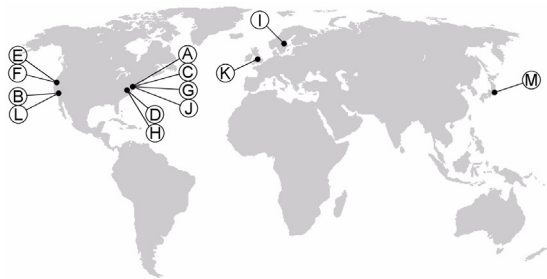


図 6.14 ルートサーバ

ドメイン名から IP アドレスに変換することを名前解決 (name resolution) といい、ドメイン名を IP アドレスに変換するコンピュータを DNS サーバ (DNS server)、またはネームサーバ (name server) といいます。ドメイン名が第 1 ドメイン、第 2 ドメイン…と階層的になっているように、DNS サーバも同じような階層構造になっています。

最上位の DNS サーバをルートサーバ (root server) といいます。図 6.14 のように、世界中に 13 台 (うち 1 台は東京) 配置されていて、トップレベルドメイン (com、edu、jp、uk など) の DNS サーバの IP アドレスを管理しています。

図 6.15 は名前解決の手順を表しています。ここではルートサーバを ① としてあります。

次の階層の DNS サーバ、たとえば jp ドメインの DNS サーバ (図 6.15 では ②) は、ac.jp や ne.jp などの DNS サーバを管理し、ac.jp ドメインの DNS サーバ ③ は、kansai.ac.jp や kwansei.ac.jp などの DNS サーバ ④ の IP アドレスを管理しています。

関西外大のウェブページにアクセスするために、ウェブブラウザに“`http://www.kansai.ac.jp/`”と入力すると、ユーザのネットワークまたはプロバイダの DNS サーバ ⑤ は、IP アドレスをまずルートサーバ ① に問い合わせます。ルートサーバはこれに対して、「それは jp ドメインの DNS サーバが管理しているから、そちらに照会しなさい」という意味で、jp ド

*32 これを URL といいます。「URL」(167 ページ) 参照。

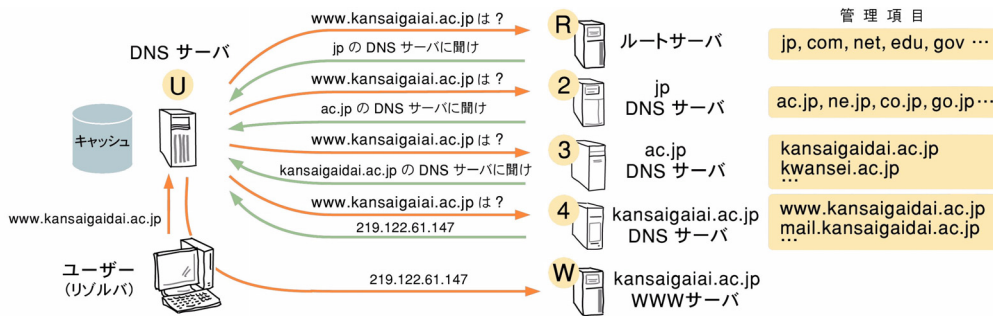


図 6.15 名前解決

メインの DNS サーバ②の IP アドレスを返します。

DNS サーバ①は次に jp ドメインの DNS サーバ②に問い合わせますが、jp ドメインの DNS サーバは、ac.jp ドメインの DNS サーバ③の IP アドレスを返してきます。

今度は ac.jp ドメインの DNS サーバ③に問い合わせると、kansai.ac.jp の DNS サーバの IP アドレスが返ってきます。次に DNS サーバ④に問い合わせると、ようやく www.kansai.ac.jp の WWW サーバ、図 6.15 では⑤の IP アドレス、219.122.61.147 が返ってきます。

ルートサーバに全てのドメイン名が登録されていれば問い合わせは一度で済むところを、ずいぶん回りくどい処理をしているように見えます。しかし、インターネットの規模が小さかった頃ならともかく、8億台以上のホストコンピュータが接続されている現在では、それを一元的に管理し、運用することは不可能です。データを複数の DNS サーバで分散して管理することにより、サーバや通信回線の負荷が分散されます。

また DNS サーバはこうして得た IP アドレスを一時的に記憶しておくキャッシュ機能を持っているので、一度問い合わせたドメイン名をその都度問い合わせることはありません。よくかける電話の番号は分厚い電話帳で調べるのではなく、電話機に登録しておくのに似ています。

2.6. ルータ

インターネットでパケットの交通整理をしているのはルータ (router) です。パケットを受け取ったルータは、目的地の IP アドレスに応じた最適の経路を選んで、パケットを最寄りのルータに転送します。次にこのパケットを受け取ったルータも同じことを繰り返せば、パケットは、いつかは相手のウェブサーバに届きます。

このように、ルータが最適の経路を選んでパケットを転送することをルーティング (routing) といいます。ルーティングを行うためには経路の情報が必要です。この経路情報は、人が入力して設定する (スタティックルーティング: static routing) こともできますが、通常は自動的に更新されます (ダイナミックルーティング: dynamic routing)。そのため、その経路にあるルータがたまたまダウンしていても、自動的にそれを迂回してパケットが転送されます*33。

*33 インターネットは ARPA によって、「核戦争にも耐えうるネットワーク」として研究・開発されました。一部

2.7. WWW

WWW (world wide web) はインターネット上にあるウェブページ (ホームページ) の情報閲覧システムで、インターネットの最も代表的なサービスのひとつです。CERN^{*34} のバーナーズ・リー (Tim J. Berners-Lee) によって 1989 年に開発されました^{*35}。

ウェブページの情報を提供する WWW サーバには、HTML (hypertext markup language) という言語で書かれた文書ファイルや、画像ファイル、音声ファイルなどが保存されていて、外部から自由に閲覧することができるようになっています。

2.7.1. HTML

HTML は、ウェブページの記述言語です。関連する他のページが容易に参照できるしくみ、リンク (link) が最大の特徴で、文書構造や書式、他の文書や画像へのリンクをタグ (tag) によって指示します。このように、様々な情報に関連づけられている文書をハイパーテキスト (hypertext) といいます。

タグはブラウザソフトへの表示の指示や命令を `< >` で挟んだもので、指示の開始と終了を表すために、通常 `<タグ> …… </タグ>` という形で記述されます。

以下は HTML 文書の例です^{*36}。

```

<html> (1)
  <head><title>Root Server</title></head> (2)
  <body bgcolor="#abcdef"> (3)
    <center> (4)
      <h1><font color="#0000cc">ルートサーバ</font></h1> (5)
       (6)
      <br> (7)
      <a href="http://www.infonet.co.jp/ueyama/ip/">情報処理概論</a> (8)
    </center> (9)
  </body> (10)
</html> (11)

```

1 行目の `<html>`、11 行目の `</html>` は、それぞれ HTML 文書の開始と終了を表しています。通常、HTML 文書の冒頭と末尾にはこれが書かれます。

2 行目はこの HTML 文書のヘッダ (header) で、ここではタイトル `Root Server` が記述されています。このタイトルはウェブブラウザのタイトルバーに表示され、ブックマークに登録したときのタイトルなどとしても使われます。

3 行目の `<body bgcolor="#abcdef">` と 10 行目の `</body>` は、この HTML 文書の本文の開始と終了を表しています。ここでは `body` タグの属性として、背景色 (`bgcolor` : background color) も指定しています。`bgcolor="#abcdef"` の `abcdef` は 16 進数で、`ab` は

のルータが破壊されても、パケットが届くように設計されています。「ARPANET」(140 ページ) 参照。

^{*34} 欧州核物理学研究所 (European center for nuclear research)。

^{*35} 「WWW」(141 ページ) 参照。

^{*36} この例ではタグを小文字にしていますが、大文字でも構いません。また、行頭のインデントはタグの対応を分かりやすくするためにつけています。なくても構いません。行番号も説明のために付加したものです。

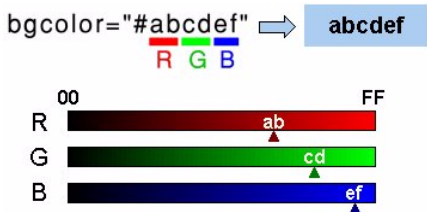


図 6.16 色の指定

赤、cd は緑、ef は青の階調を表しています。ここでは赤の階調の値は ab ですから、10 進数では 171、同様に緑の cd は 205、青の ef は 239 です。00 はいちばん暗い階調の値なので黒、ff は最も明るい値ですから、それぞれいちばん明るい赤 (R)、緑 (G)、青 (B) として表示されます。abcdef だと赤よりは緑、緑よりは青が明るい値なので、背景は水色に見えます。

Web で学ぶ【タグ (色の指定)】

ABCDEF

Web ページ「タグ (色の指定)」には、カラーコードを自由に設定できる Java アプレットがあります。マウスで簡単に色を変えることができますから、カラーコードと色の関係がすぐに理解できます。

4 行目の `<center>` と 9 行目の `</center>` は、文字や画像を中心に揃えて表示する指示です。この例では、見出しも図もリンクも、全て中心に整列して表示されます。

5 行目でページの見出し、「ルートサーバ」を表示させています。`<h1>` の h は heading の略です。一番大きい見出し `<h1>` から一番小さい `<h6>` まで、6 段階の大きさの見出しを選べます。ここでは `` も併用して文字に色をつけています。`color="#0000cc"` という指定では、見出しの文字は藍色になります。

6 行目は画像を表示しています。`` として表示する画像のファイル名を指定します。img は image、src は source の略です。画像ファイルを原寸で表示させる場合は width、height の指定を省略できますが、サイズを明示した方がウェブブラウザでの表示が速くなります。画像の表示サイズの単位はピクセルです。alt は alternative の略で、ここに書かれたテキストは画像が表示できない環境で表示され、また視覚障害者のための音声ブラウザ*37 ではこれが読み上げられます。誰にも分かりやすいページにするために、省略せずに丁寧に書いておく配慮が必要です。

*37 「音声合成」(104 ページ) 参照。

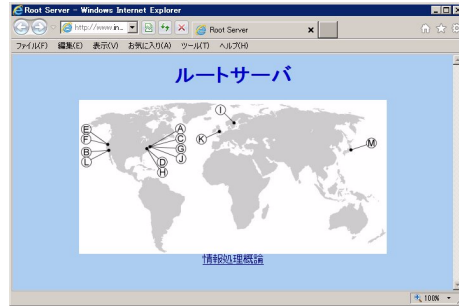


図 6.17 ウェブページの例

7行目の `
` は break の略で、改行を意味しています。ウェブブラウザで改行して表示させるには、`
` タグで明示的に改行を指示しなければなりません。

8行目がリンクの指定です。この場合は「情報処理概論」という文字がアンダーライン付きで表示され、これをマウスでクリックすると <http://www.infonet.co.jp/ueyama/ip/> にジャンプします。a は anchor の、href は hyper reference の略です。

この HTML ファイルをウェブブラウザで表示させると、図 6.17 のようになります。

2.7.2. FTP

こうして完成した HTML ファイルは、コンピュータのディスクに保存されています。しかし、そのままでは単なる HTML ファイルであって、「ウェブページ」ではありません。これをウェブサーバに転送して初めて、インターネットで誰もが見ることができる「ウェブページ」になります。

プロバイダと契約すると、インターネットへの接続やメールアドレスと共に、ウェブページのためのディスクスペースが提供されます。ここに HTML ファイルを転送します。

このとき、ファイルをサーバに転送することをアップロード (upload)*38、転送に使うプロトコルを FTP (file transfer protocol)、そのためのソフトウェアを FTP クライアント (FTP client)、あるいは FTP ソフトといいます。

2.7.3. URL

165 ページの HTML ファイルは、root_server.html というファイル名でウェブサーバ www.infonet.co.jp に実際にアップロードしてあります。このウェブページの URL (uniform resource locator) は http://www.infonet.co.jp/ueyama/ip/root_server.html です。したがって、これをウェブブラウザに入力すれば、ウェブページ “Root Server” が表示されます。

URL はインターネットの情報の住所のようなもので、図 6.18 の例では、左から順に、スキーム名、ドメイン名、フォルダ*39名、ファイル名で構成されています。

*38 サーバから手元のコンピュータに転送することをダウンロード (download) といいます。

*39 ディレクトリ (directory) や、パス (path) という場合もあります。

http://www.infonet.co.jp/ueyama/ip/root_server.html

スキーム名 ドメイン名 フォルダ名 ファイル名

図 6.18 URL の構成

最初のスキーム (scheme) はインターネットの情報にアクセスする手段を表していて、ウェブページを閲覧する場合は HTTP というプロトコルが使われます。http の他にはファイル転送に使われる ftp、ローカルディスクのファイルを表す file などがあります。

ドメイン名はその情報が置かれているサーバの名前、フォルダ名はその情報が保存されているフォルダの名前、ファイル名はその HTML ファイルの名前です。

この URL、http://www.infonet.co.jp/ueyama/ip/root_server.html は、「www.infonet.co.jp というウェブサーバの、ueyama/ip/というフォルダにある root_server.html というウェブページを、http プロトコルを使って閲覧したい」という意味になります。

また、http://www.kansai.gaidai.ac.jp/ は関西外国語大学のウェブページの URL ですが、このように、フォルダ名がない、あるいはファイル名を省略できる場合もあります*40。

2.7.4. HTTP

ウェブブラウザがサーバとの間で情報を交換するときに使われるプロトコルが HTTP (hypertext transfer protocol) です。

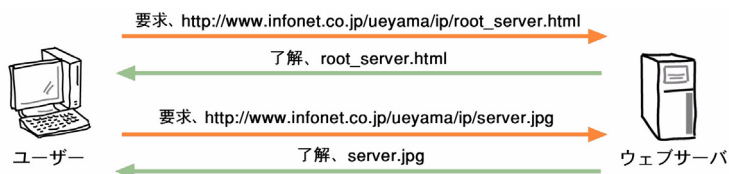


図 6.19 HTTP のしくみ

先の URL、http://www.infonet.co.jp/ueyama/ip/root_server.html がウェブブラウザのアドレス欄に入力されると、このリクエストを受け取ったウェブサーバは、まず、root_server.html ファイルを送り出します。

ユーザーが最初に受け取るのは、HTML ファイル、すなわちテキストファイルです。タイトルは何々、画面の背景は何色、見出しの色や文字の大きさはこうして… そのようなことが書かれています。ウェブブラウザは背景や文字を指示通りの色で表示します。そういったことは、文書の指示だけで処理することができます。

しかし HTML ファイルを順に見ていくと、6 行目は `` となっています。ここに、縦横こういうサイズで画像を表示しなさいとい

*40 通常、ファイル名が省略された場合、index.html や welcome.html という名前のファイルが表示されます。

う指示ですが、画像のデータはないので `server.jpg` ファイルを要求します。ウェブサーバはそれに応じて画像ファイルのデータを送信し、ウェブブラウザはそれを受け取って表示します。

この例では使用されている画像ファイルはひとつだけですが、イラストや写真を多用したページには `img` タグが大量にあります。あらかじめウェブサーバが `HTML` ファイルをチェックして、必要なものを一括して送信してもよさそうなものですが、実際はそうではなく、ブラウザがその都度、必要に応じてデータの送信を要求します。

ウェブブラウザは、こうして受信したデータをキャッシュファイルとしてすべて一時的に保存しています。新たにウェブページにアクセスするときは、まず自身のキャッシュファイルの中に同じものがないかチェックします。もし手元があれば、わざわざ送信してもらうより素早く表示できるし、ネットワークのトラフィックを増やすこともないからです。

2.8. 電子メール

電子メール (electronic mail) は、ウェブページの閲覧に次いでよく利用されるサービスです。手軽にメールが送信でき、即座に届くので大変便利です。

郵便に似ていますが、大きく違うところは、郵便のように自宅まで配達されるのではなく、電子メールはメールサーバのメールボックスに保管されていることです。郵便局の私書箱のようなもの、と考えればいいでしょう。

2.8.1. SMTP と POP

電子メールを配送するプロトコルを `SMTP` (simple mail transfer protocol)、受信するプロトコルを `POP` (post office protocol) といいます。

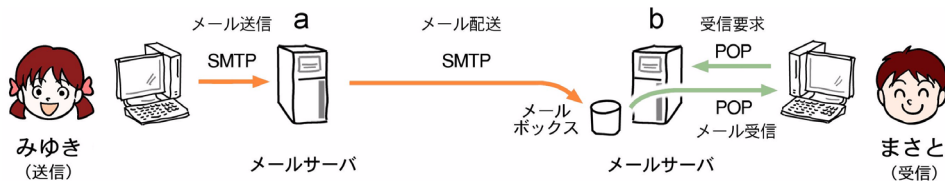


図 6.20 電子メールのしくみ

いま、みゆきがまさとに電子メールを送るものとします。メールを送信すると、まずみゆきのコンピュータが接続されているネットワーク (学校や会社、あるいはプロバイダなど) のメールサーバ **a** に届きます。このときに使われるプロトコルが `SMTP` です。次にメールサーバ **a** は、まさとが属するネットワークのメールサーバ **b** にメールを配送しますが、このときにも `SMTP` が使われます。メールはメールサーバ **b** のメールボックスに保管されます。

まさとがメールを受信するときのプロトコルは `POP` です。まさとのメールソフトは、ユーザー ID やパスワードを送信してユーザー認証を受けた後、着信メールがあればそれを受信し、

第6章 ネットワークとインターネット

メールボックスのメールを消去します。

電子メールは、受信者がメールの着信をチェックすることによって、はじめて手元に届きます。郵便のように直接配達しようにも、送信先のコンピュータの電源スイッチが OFF になっていれば配達できないからです。書留や宅配便が留守中に配達されたときはメモを置いていってくれますが、コンピュータの電源が OFF では、それもできません。そこで、メールボックスに保管しておく方法、POP が考案されました。これをポーリング型電子メールといいます。

これに対して、携帯電話は常に電源が入っているので、直接端末までメールを届けることができます。こちらはプッシュ型電子メールといいます。

2.8.2. MIME

電子メールは本来、ASCII *41 で定められた文字だけしか送受信できません。したがって、日本語は JIS 漢字コード (JIS X 0208) なら大丈夫ですが、パーソナルコンピュータで使われているシフト JIS 漢字コードは、ASCII にはないコードが含まれているため使えません*42。また電子メールも郵便のように写真などを同封 (添付) して送れば便利ですが、写真のデータには文字コード以外のデータが含まれているため、これも不可能でした。

たとえば画像ファイルのデータの一部を 16 進数で表すと、

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 48 00 48 00 00 FF DB 00 ...
```

などとなっています。下線をつけた 4A、46、49、48 はそれぞれ J、F、I、H という「文字」ですが、それ以外は文字ではありません*43 から、これらのコードを含んでいる画像ファイルなどは、電子メールに添付して送信することができませんでした。

MIME (multipurpose internet mail extension) はこれを文字に変換して送信できるようにしようという規格です。同じ画像ファイルを MIME でエンコードすると、

```
AgIDAwQDAwMDAwICAwQDAwQEBAQEAgMFBQQEBQQEBAT/wA...
```

となります。まるで暗号のようですが、すべて「文字」に置き換えられています。これなら電子メールのプロトコルで送信することができます。

現在は MIME が一般的になったため、漢字はもちろん、写真でも音楽でも動画でも、電子メールには何でも添付して送信することができます。

*41 american standard code for information interchange. 「文字コード」(22 ページ) 参照。

*42 「JIS 漢字コード」(24 ページ)、「シフト JIS 漢字コード」(25 ページ) 参照。

*43 00 は NUL (Null)、01 は SOH (Start of Heading)、10 は DLE (Data Link Escape) という制御符号で、「文字」ではありません。また ASCII は 7 ビットなので、コードの範囲は 00~7F です。FF、D8、E0 などは存在しません。

第 7 章 生活と情報技術

コンピュータは 20 世紀半ばに誕生しましたが、半導体技術の飛躍的な進歩と共に、みるみるうちに小型化、高機能化、低価格化を果たし、当初おそらく誰も想像しなかったほどに普及しました。気がついてみれば私たちの身のまわりはコンピュータだらけ、1 日といえどもコンピュータに接しない日はありません。

もともとコンピュータは計算機で（今でもそうですが）、大量の計算を短時間で処理するために作られました。そしてデータやプログラムを効率よく入力するためにキーボードがつながれ、処理結果を即座に表示できるようにディスプレイがつながれて、現在のパーソナルコンピュータの「かたち」ができあがりました。

しかし、キーボードの代わりにセンサを、ディスプレイの代わりにモーターをつないでエアコンに組み込めば、コンピュータはエアコンの制御装置になります。入力された温度センサのデータと設定温度を比較して、室温が設定温度より高ければモーターを回せばいいのです。実際はこんなに単純ではありませんが、基本的にはこれで温度調節ができます。かつてのエアコンには「強・中・弱」の切り替えスイッチしかありませんでした。部屋が暑ければ「強」にし、冷えすぎれば「弱」に、人が調節していたのです。

入力装置としてスイッチやセンサ、出力装置としてモーターやバルブなどのアクチュエータをつないだコンピュータは、人に代わって機械の制御をすることができます。このようなコンピュータは、一度仕様が定まればプログラムを変更することもほとんどないので、プログラムは ROM *1 に格納され、専用機として働きます。コンピュータとしての機能はすべてひとつの IC チップに集積され、コンピュータ自体はほとんど目立たなくなります。これをマイクロコンピュータ、あるいは組み込みコンピュータといいます*2。

現在では、コンピュータのない生活は考えられません。この章では私たちの身の回りにおいて、あまり目立つことなく活躍しているコンピュータやコンピュータシステムについて考えていくことにします。

*1 read only memory. 「ROM」(53 ページ) 参照。

デジタルカメラや携帯電話などでは、購入後もプログラムを更新できるものがあります。このような製品ではプログラムはフラッシュメモリなどに格納され、プログラムを読み込む何らかのしくみも備えています。

*2 「マイクロコンピュータ」(7 ページ) 参照。

1. 家中どこにも、コンピュータ

現在ではたいいてい家電製品、テレビにもエアコンにも洗濯機にも、小型のコンピュータが組み込まれていて、コンピュータの制御によって動いています。

洗濯機は、適量の水が入ると洗剤を加え、洗濯槽を回転させ、濯いで、脱水までやってくれます。もちろん、各種センサから与えられる情報と、それを処理するコンピュータ、その処理結果によって動くモーターやバルブなどが、システムとして総合的に働いているからです。

私たちの身の回りには、コンピュータで制御されているものがたくさんあります。コンピュータはいつの間にか、私たちが考えているよりも、もっともっと身近な存在になっています。ここでは、どの家庭にもありそうなもので、「えっ、こんなものにも？」と思えるものをいくつか、どんな働きをしているのかを見ながら紹介することにします。

1.1. 電気炊飯器

電気炊飯器は1955年、「自動式電気釜」という名前で初めて商品化されました。

お米を磨いで、約20分間沸騰させるとご飯が炊けますが、だからといって、タイマーをくっつけただけではうまくいきません。米や水の量、気温の変化などによって沸騰するまでの時間が変わるため、焦げついたり、芯のあるご飯ができたりします。そこで「自動式電気釜」は釜を二重構造にし、内釜と外釜の間に20分で蒸発する分量の水を入れることでこの問題を解決しました。沸騰して20分たつて、水が蒸発してしまうと温度が上がり、サーモスタット*3が作動してスイッチが切れるようになっていました。

現代の“自動式電気釜”は、温度センサとマイクロコンピュータによってこの問題を解決しています。玄米でもお粥でも何でも炊けて、予約や保温もできます。



図 7.1 電気炊飯器の例

図 7.1 は電気炊飯器の一例で、外観とその操作部、およびプリント基板です。プリント基板のほぼ中央に斜めに取り付けられているのが、1チップのマイクロコンピュータです。

炊飯器そのものの制御はあまり複雑ではないので、ここで使われているのは4ビットのコンピュータですが、ICにはセンサの信号をデジタル化するA/D変換器やタイマー、液晶ディ

*3 熱膨張率の違う金属を張り合わせた「バイメタル」とスイッチを組み合わせたもの。温度が変わるとバイメタルが変形し、スイッチが作動する。

スプレイの駆動回路など、炊飯器に必要な機能がすべて集積されています。

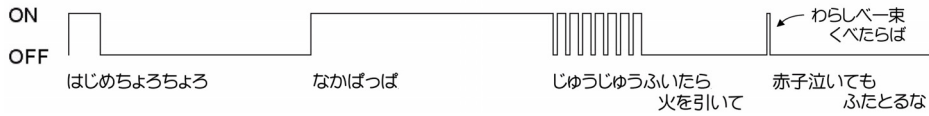


図 7.2 電気炊飯器の炊飯例

図 7.2 は電気炊飯器の炊飯例*4 です。

おいしいご飯を炊く秘訣は、昔から「はじめちよろちよろなかばっぱ、じゅうじゅうふいたら火を引いて、赤子泣いても蓋とるな」と言い伝えられていますが、ここでもそれがきちんと守られています。

スイッチを入れるとまずヒーターが ON になって加熱が始まり、水温が約 50℃になると OFF になります。これがお米に水分を吸わせるための「はじめちよろちよろ」期。約 10 分後に「なかばっぱ」、本格的にご飯を炊き始めます。「じゅうじゅうふいたら火を引いて」ではヒーターの ON と OFF を繰り返して火力が約 $\frac{3}{4}$ になるように調整されます。ご飯が炊きあがると、蒸らしのための「赤子泣いても蓋とるな」に移ります。一説ではこの間に「藁しべ一束くべたらば」が挿入されるそうですが、ここでの約 5 秒間のヒーター ON が「藁しべ一束」なのかもしれません。

おいしいご飯を炊くのも慣れるまではたいへんですが、お米を磨いでスイッチを押すだけ、40 分ほど待てば炊きあがります。マイクロコンピュータと温度センサが、一切を引き受けてくれているからです。

1.2. シーリングライト



照明器具は電球や蛍光灯を灯す器具ですから、コンピュータがなくても大丈夫、ちゃんと点灯します。しかし ON-OFF や調光をリモコンでしようと思えば、やはりコンピュータが必要になります。

1.2.1. シーリングライト本体

図 7.3 は、コンピュータ内蔵のシーリングライの例です。

左は器具のカバーを外した状態で、蛍光灯の内側左寄りに凹みがあって、その中に小さい黒い点が見えます。ここにリモコンからの赤外線を受けるセンサがあります。

*4 図 7.1 の炊飯器のものではありません。



図 7.3 シーリングライの例

中央の写真は制御用のプリント基板ですが、この電子回路の大部分は蛍光灯を点灯させるためのインバータ (inverter) です。通常、蛍光灯は 50Hz または 60Hz の交流電源で点灯させますが、大きくて重い安定器が必要で、雑音やちらつきも出やすい傾向があります。それに対してインバータは、ひとまず交流を整流して直流に変え、それをまた高い周波数の交流に変換します。周波数を高くすることによって大型の安定器が不要になり、雑音やちらつきがなくなり、発光効率も高めることができます。コンピュータはリモコンの信号を受信するために使われていて、この写真では隠れていて見えませんが左上の奥の方にあります。

右の写真はプリント基板裏面のコンピュータ部で、やはり斜めに取り付けられている IC が 1 チップのマイクロコンピュータです。

1.2.2. リモコン

リモコン (remote control) にもコンピュータが使われています。シーリングライに限らず、テレビやエアコンはもちろん、ビデオやオーディオ機器から扇風機にいたるまで、家の中にはいくつものリモコンがあり、それぞれにコンピュータが組み込まれています。照明を点けたり消したり、テレビのチャンネルひとつ変えるにもコンピュータだなんて、信じられますか。

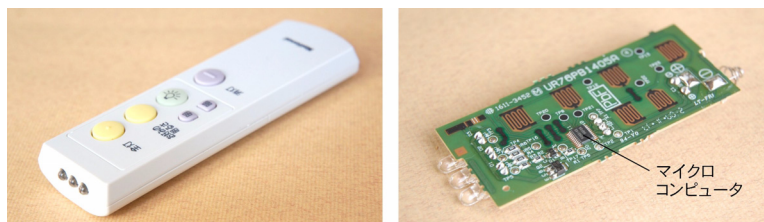


図 7.4 リモコン

図 7.4 はリモコンの外観 (左) と内部のプリント基板 (右) です。プリント基板のほぼ中央手前に 1 チップマイクロコンピュータがあります。

リモコンを「コンピュータ」として見たとき、その入力装置はボタンです。点けたり消したり、明るくしたり暗くしたり、ボタンにはそれぞれ機能が割り当てられていて、ボタンが押される度にコンピュータにその信号が入力されます。

出力装置はこの場合赤外線 LED だけですが、エアコンのリモコンなどのように液晶ディスプレイが付いているものは、もちろんそれも出力装置の一部です。マイクロコンピュータは、押されたボタンに応じて、定められた光の ON-OFF のパターンを作る仕事をしています。

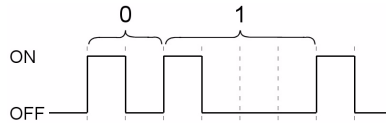


図 7.5 リモコンの信号

図 7.5 はリモコンの赤外線信号の例で、‘0’ と ‘1’ は光の ON-OFF の時間で表しています。一定時間（この場合は約 $\frac{1}{2000}$ 秒）光が ON *5 にされ、同じ時間だけ OFF にされると ‘0’、OFF の時間が 3 倍だと ‘1’ です。

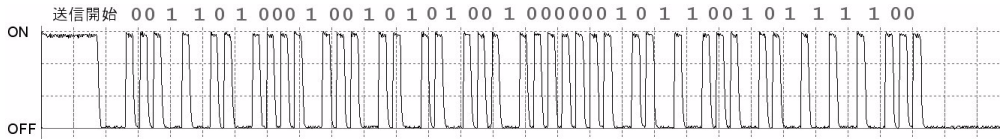


図 7.6 リモコンの赤外線信号例

図 7.6 は実際にリモコンから出力された赤外線信号です。この例では、送信開始を表す長い ON-OFF のあと、“00110100 01001010 10010000 00101100 10111100” という 40 ビットのデータが送信されています。前半 3 バイト (24 ビット) はメーカー名や機種を表す固定のデータ、後半 2 バイトの各ビットに蛍光灯の ON や OFF、明るく、暗くといった意味が割り当てられています。

現在、多くのリモコンの通信仕様は、各メーカー独自のものです。他のリモコンとの混信を防ぐために、通常冒頭にこのような機種を表すデータが加えられていますが、国際的な規格ではないため誤作動の可能性があります。テレビのリモコンで電気ストープがついた例*6 もあり、場合によっては非常に危険なため、早期の対策が望まれます。

1.3. おもちゃ

コンピュータがワンチップ化されて家電品に大量に使用されると、マイクロコンピュータの価格もどんどん下がって手頃になり、やがておもちゃにもコンピュータが使われるようになりました。

図 7.7 はデジキュー (DigiQ)、赤外線でコントロールするリモコンカーで、車の全長は約 5cm です。こんな小さい車の中に、モーターやバッテリーはもちろん、コンピュータまで組み

*5 リモコン以外から出る赤外線によって誤動作するのを防ぐために、実際にはこの間に 16 回も素速く ON-OFF を繰り返しています。

*6 「TV つけたらストープついた リモコン 危ない誤作動」(2007 年 4 月 1 日付朝日新聞 第 1 面)。



図 7.7 DigiQ とその内部

込まれています。図 7.7 の右はその内部で、中央下部の薄いグリーンの部分バッテリー、その後ろにモーターやギヤがあります。プリント基板上の IC が 4 ビットのマイクロコンピュータ、その後ろで L 字形に折れ曲がっているのが赤外線センサです。

この車は、左右のタイヤを別々のモーターで駆動して車の動きをコントロールします。どちらも同じスピードで回転すれば車は直進、右のモーターが左よりも早ければ車は左に、逆なら右にカーブします。そのため、左右のモーターの回転速度を個別に赤外線の信号でコントロールしています。

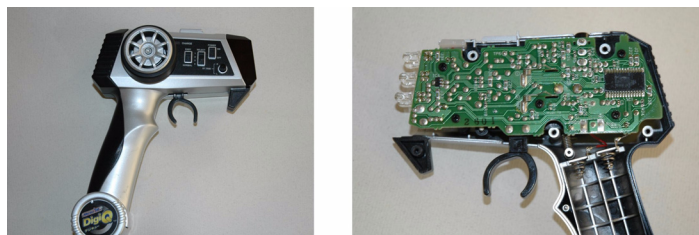


図 7.8 コントローラ（左）とその内部（右）

図 7.8 はコントローラです。ピストルの引き金のように見えるスロットトリガーを引くと車は前進し、押すと後退します。車の速度は引く強さ、押す強さで調整できます。タイヤのように見えるのはステアリングホイールで、これを回すと車は右や左にカーブします。内部の基板の右端にある IC が、やはり 4 ビットのマイクロコンピュータです。

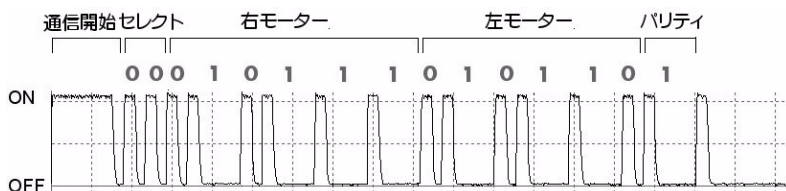


図 7.9 コントローラの光信号例

図 7.9 はコントローラから出力された赤外線信号の一例です。ここでも光の ON と OFF の時間の比で 0 と 1 を表しています。通信開始の合図に続く 2 ビットは、4 台のコントローラ

を識別するためのセレクト信号です*7。左右のモーターの回転方向と速度の信号がそれぞれ6ビット、それにパリティビットの計15ビットからなっています。

パリティビット (parity bit) は偶奇性検査のためのビットで、データ全体に含まれる‘1’の数をいつも偶数または奇数と定めておけば、データ通信時のエラーの有無をチェックすることができます。図7.9の例では、データ“00 010111 010110”に含まれている‘1’は7個なので、パリティビットを‘1’にして、‘1’の数を8個、偶数にしています。データを受信したとき、もし‘1’の数が奇数であれば、通信に誤りがあったことが分かります。

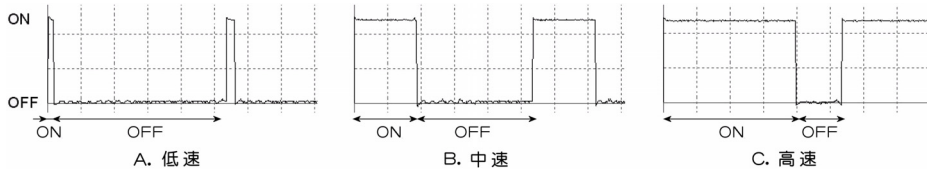


図 7.10 モーターの制御信号

モーターの制御信号は、左右のモータにそれぞれ6ビット、うち1ビットは前進・後進を表していますから、速度の情報は5ビットです。車のコンピュータは、この5ビットの速度情報から、図7.10のようなモーターの制御信号を作ります。

低速で走るときは、モーターに電流を流して、直ちにOFFにします。こうすると約5%の時間だけモーターに電流が流れます。車は5%のパワーで、ゆっくりと動き出します*8。

スロットルトリガーをもう少し引くと、モーターの制御信号は図7.10の中央、Bのような状態になります。この場合は約35%、もっと強く引くとCのようになって、約75%のパワーで走るようになります。

電気炊飯器の「じゅうじゅうふいたら火を引いて」*9 のときも、ヒーターのONとOFFを繰り返して火力を調整していました。電気炊飯器のヒーターなら秒という単位でON-OFFさせても構いませんが、モーターの場合はもっと早くしないと動きがぎくしゃくします。しかしコンピュータや半導体にとって、 $\frac{1}{1000}$ 秒の単位でON-OFFを繰り返すことなど朝飯前です。このようにして、ON-OFFの時間の比を変えることによってモーターの回転速度や電力を制御することを、パルス幅変調 (PWM : pulse width modulation) といいます。

1.4. ガスメーター

図7.11はガスメーター (マイコンメーター) です。普段はほとんど気に留めることもありませんが、ここにもコンピュータが組み込まれています。ガスメーターは、ガスの使用量をチェックして料金を請求するためにだけあるのではなく、私たちの安全を守るためにも働いています。

大きな地震があると、火災が発生します。1995年の阪神・淡路大震災でも285件の火災が

*7 4台のコントローラを識別できるので、4人でカーレースができます。

*8 モーターがON-OFFされる間隔は $\frac{1}{100}$ 秒程度です。

*9 「電気炊飯器」(173ページ)参照。



図 7.11 ガスメーター

発生し*¹⁰、6,982棟が全焼、9,017世帯が罹災しました。マイコンメーターにはこうした火災の発生を防ぐために、震度5強相当以上の地震を感知するとガスを遮断する機能があります。ガス管が破損するなどして多量のガスが流れた場合や、ガス管にトラブルがあって圧力が低下したときもガスを遮断します。いずれも地震発生直後ばかりでなく、ガスや電力の供給が再開されたときに発生しがちな火災の防止にも役立ちます。またガス漏れを検出したり、ガスが長時間流量の変動なしに流れ続けた場合も、異常と判断してガスを止めます。遮断するガスの流量や時間は、普段のガスの使用状況から学習して自動的に決められています。

1.5. 身の周りはコンピュータだらけ



冷蔵庫、電子レンジ、電気炊飯器、湯沸器、食器洗い機、掃除機、洗濯機、アイロン、扇風機、エアコン、テレビ、録画機、CDコンポ、ファックス、ゲーム機、電子辞書、携帯電話、デジタルカメラ、ビデオカメラ…。思いつくままに、身の周りにおいて、内部にコンピュータが組み込まれていそうなものを並べてみました。もちろん、この他にもたくさんあります。

私たちの身の周りはコンピュータだらけです。こうした機器は高性能に、私たちの生活は便利になりました。もしコンピュータがなかったら…。いえ、もはやコンピュータなしの生活は考えられませんね*¹¹。

*¹⁰ 発火源が判明しているのは139件で、うち「ガス油類を燃料とする道具」による火災が24件あります。

*¹¹ 原子力発電所はなくなってもやっていけそうですが…。

2. 携帯電話

携帯電話自身も「コンピュータ」ですが、携帯電話（cellular telephone）の通信システムももちろんコンピュータによって管理されています。ここでは、携帯電話のしくみや歴史、携帯電話は今後どう変わっていくのか、などについて考えてみることにします。

2.1. 携帯電話の現況

携帯電話の契約数は、2011年12月末現在で12,176万件です*¹²。これを同時期の人口12,615万人*¹³で割ると、普及率は96.5%になります。しかし携帯電話の実質的な利用者は10～74歳だと考えれば、10歳未満と75歳以上を除いた人口は10,224万人ですから、普及率は119%に達していることになります。企業などがまとめて契約しているものも少なくないので、実質的な普及率はもう少し低いのですが、契約数の推移を見る限りでは、まだまだ契約数が増え続けそうな勢いです。

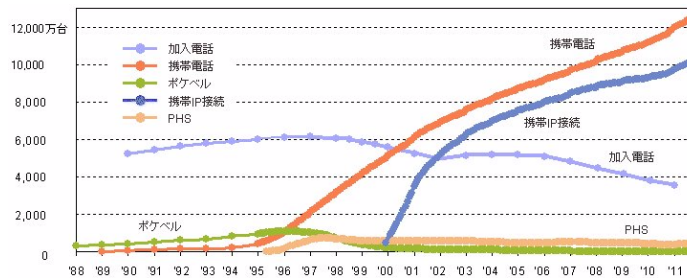


図 7.12 固定電話・携帯電話等の契約数の推移

図 7.12 は固定電話や携帯電話の累計契約数の推移を表しています。

携帯電話は1995年頃から、本格的に普及し始めました。同じ頃、手軽な PHS（personal handyphone system）のサービスも開始されましたが、携帯電話の使用料金が下がったこともあって、PHS の契約数は頭打ちから減少傾向となって現在に至っています。

当時、最も普及していたのは言うまでもなく固定電話ですが、これも同時期から減少に転じ、2000年には携帯電話と固定電話の契約数が逆転しました。

携帯電話がこのように驚くべき普及をした理由のひとつは、1999年に携帯電話がインターネットへの接続機能を持ったことでしょう。図 7.12 にも示されているとおり、携帯 IP 接続はまたたくうちに普及しました。他にも音楽再生、デジタルカメラ、Java プログラム実行、キャッシング、ワンセグ機能などが次々と搭載されました。2008年にはスマートフォンが発売され、携帯電話はついに高機能の携帯情報端末となりました。

*¹² 出典：社団法人 電気通信事業者協会 (<http://www.tca.or.jp/>)

*¹³ 出典：総務省 統計局 (<http://www.stat.go.jp/index.htm>)

2.2. 携帯電話のしくみ

固定電話は相手の電話が文字通り固定されているので、市外局番、市内局番、電話番号ですぐにつながりますが、携帯電話の場合は、相手の端末が今どこにあるのか分かりません。それでもちゃんと、携帯電話がつながるのは何故でしょうか。



図 7.13 携帯電話基地局の各種アンテナ

携帯電話の通信はセルラー方式 (cellular system) によっています。セルラー方式では、サービスエリアを小さいセル (cell) に分割し、それぞれのセルごとに基地局を設け、ユーザ (移動局) は現在位置しているセルの基地局と通信します*14。

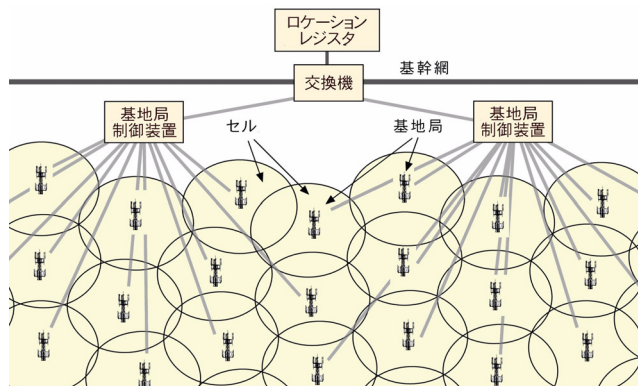


図 7.14 携帯電話のしくみ

基地局は数 10 台程度のグループにまとめられ、基地局制御装置によって管理されています。また、基地局制御装置は交換機を介して基幹網と呼ばれるネットワークに接続されています。

携帯電話は定期的に最寄りの基地局と交信し、現在どのセルの中にいるという情報がロケーションレジスタ (location register) と呼ばれる記憶装置に登録されています。携帯電話のスイッチが ON にされたときや、別のセルに移ったときも、直ちに位置情報が登録・更新されます。ロケーションレジスタを検索すれば、どの携帯電話がどのセルの中にいるかが分かるようになっています。

*14 相手の携帯電話がすぐ近くにいっても、トランシーバーのように、携帯電話同士が直接通信することはありません。

携帯電話から発信するときは、相手の電話番号を先ず基地局に送信します。この番号は交換機に送られ、交換機は相手のユーザー ID と、相手の携帯電話が属している基地局をロケーションレジスタに問い合わせます。基地局が分かると、そこに相手のユーザー ID を送ります。基地局がこのユーザー ID を送信すると、セル内にある全ての携帯電話がこれを受信しますが、ユーザー ID が一致した端末だけが応答します。こうして発信元と着信先の携帯電話の通話用の回線が設定されます*15。

携帯電話で通話中、隣のセルに移動すると、通信する基地局は隣のセルの基地局に切り替えられます。通話中にセル間を移動しても、交信する基地局が自動的に切り替わって、途切れることなく通話できる機能をハンドオーバー (hand over) といいます。

2.3. 携帯電話の歴史

2.3.1. 軽量化の時代

移動式電話は、自動車電話からスタートしました。この電話機の重量は 7kg、容積が 6.6ℓ といえますから、2ℓ のペットボトル 3 本分余りの重さと嵩かさです。本体は自動車のトランク内に取り付けて使われました。

一般的なサービスは 1979 年 12 月にまず東京で、次いで 1980 年 11 月には大阪でも始まりましたが、当初は保証金が 20 万円、新規加入料が 72,800 円、基本料金が月額 3 万円、通話料が 6 秒で 10 円という料金設定でしたので、これでは利用できる人は限られていますし、また普及するはずもありません*16。



図 7.15 ショルダーホン (左) と初代携帯電話 TZ-802 (右)

1985 年 9 月には「ショルダーホン」ができました。こちらは重量が約 3kg で容積は 2ℓ。名前の通り、肩にかけて持ち歩くことができますが、正式名称は車外兼用型自動車電話です。重量 3kg ではやはり、「携帯電話」とは呼べなかったようです。

最初の携帯電話、TZ-802 は重量 900g、容積 500ml でした。まだカバンに入れて持ち歩くのは躊躇ためらわれる重さですが、ともあれ 1987 年 4 月、携帯電話のサービスが開始されました。

図 7.16 は、携帯電話機 TZ-802 以降、携帯電話がどのように軽くなっていったかを表してい

*15 携帯電話と固定電話、あるいは異なる通信事業者の携帯電話と通話する場合、回線設定はもっと複雑になります。

*16 普及の障壁となる保証金は 1987 年に 20 万円から 10 万円に値下げされ、1993 年には保証金制度そのものが廃止されました。新規加入料も 1991 年に 45,800 円、1992 年に 36,000 円、1994 年に 21,000 円、1995 年に 9,000 円、6,000 円と段階的に値下げされ、1996 年 12 月によりやく廃止されました。

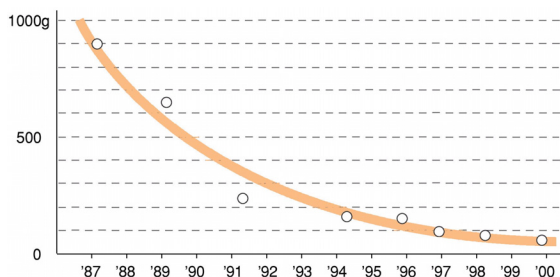


図 7.16 携帯電話の軽量化

ます。最初は 900g であったものが毎年約 20% ずつ軽くなっていき、1999 年 11 月にはわずか 57g しかない P208 HYPER が発売されました。

軽量化のレースはこのあたりでひとまず終了し、1997 年に始まったショートメールサービスを契機として、携帯電話は機能で競う時代に移りました。

2.3.2. 多機能化の時代

ショートメールは、50 字までのメールを送受信できるサービスです。通話一辺倒であった携帯電話が、50 字とはいえデータ通信機能を持ったことで利便性が高まりました。しかし文字数が制限されている上、携帯電話各社による独自仕様のサービスのため、提携していない会社間ではメールが交換できないなど、不便な点も少なくありませんでした。

1999 年 2 月、i モードサービスがスタートし、携帯電話でインターネットに接続してホームページを見たり、電子メールが利用できるようになりました。2000 年 6 月には i モードの利用者が早くも 1,000 万人に達し、図 7.12 (179 ページ) の通り、その後も利用者は増え続けています。



図 7.17 日本初のカメラ付き携帯電話、J-SH04

写真提供：シャープ株式会社

2000 年 6 月、韓国のサムスン社 (Samsung Electronics) が世界に先駆けてデジタルカメラ搭載の携帯電話を発売、日本でも同年 11 月に J-フォンからカメラ付きカラー携帯電話 J-SH04 が発売されました*17。

*17 カメラは 11 万画素、液晶画面は 256 色表示でした。

2000年7月には16和音対応の着信メロディがダウンロードできるようになりました。着信メロディそのものは以前(1996年)からあり、着信音作曲機能を持つ機種もありましたが、これまでは4和音までの単純な音楽しか扱えませんでした。

2001年1月には、携帯電話でJava^{*18} プログラム(iアプリ)が実行できるようになりました。これで携帯電話は、ますます携帯情報端末に近づきました。

2001年の10月にFOMA(freedom of mobile multimedia access)サービスが開始されました。FOMAは第1世代のアナログ式携帯電話、第2世代のデジタル通信方式に次ぐ第3世代、IMT-2000(international mobile telecommunication 2000)による通信方式です。固定電話なみの音質と、下り384kbps、上り64kbpsの高速データ通信、およびそれを生かした動画や音声の配信などが特徴で、TV電話の機能を持つものも現れました。

2004年7月、非接触ICカードFeliCa^{*19}を内蔵した携帯電話、P506iCが発売されました。携帯電話を加盟店の読み取り装置にかざすだけで支払いができます。携帯電話が財布の役割を果たすようになりました。



図 7.18 ワンセグ対応携帯電話、W33SA

2005年12月には移動体向けの地上デジタルテレビ放送サービス、ワンセグ^{*20}(1 segment broadcasting)対応の携帯電話が発売され、携帯電話はこうして、テレビにもなりました。

携帯電話の進化はまだまだ続きます。従来からPDA(Personal Digital Assistant: 携帯情報端末)^{*21}に電話・通信機能を持たせたものがいくつか提案されていましたが、広く普及するには至りませんでした。しかし米国で2007年に発売されたアップルのiPhoneは、複数の指で画面の縮小・拡大などができるマルチタッチスクリーン(multi-touch screen)の操作性の良さもあって、たちまち世界中で受け入れられ、日本でも2008年7月に発売されました。

図 7.19 は携帯電話の総出荷台数に対するスマートフォン出荷台数の割合です^{*22}。最初の2008年は数%程度に過ぎませんでしたが、以後前年度比2~3倍という驚異的なペースの成長

*18 「Java」(82ページ)参照。

*19 不揮発性メモリと無線通信機能を内蔵し、読み取り装置からの電波を電力として動作する非接触ICカード。JR東日本のSuicaや、電子マネーEdyにも採用されています。「電子マネー」(195ページ)参照。

*20 地上デジタル放送では、1つのチャンネルは13の帯域(セグメント)に分割され、ハイビジョン(HDTV)は12、通常放送(SDTV)は4つのセグメントで放送します。セグメントを1つだけ使って放送するのがワンセグで、解像度は通常放送の $\frac{1}{4}$ 、QVGA(320×240画素。ディスプレイ(68ページ)参照)なので、携帯電話で受信するのに適しています。本放送開始は2006年4月1日。

*21 スケジュール、住所録、メモなどの情報を携帯して扱うための小型機器。

*22 出展: 株式会社MM総研(<http://www.m2ri.jp/>)

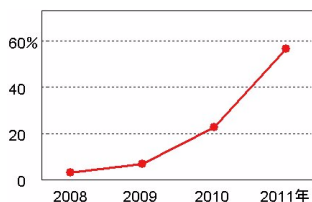


図 7.19 スマートフォンの急成長

を続け、2011年にはスマートフォンの出荷割合は56.6%に達しました。2016年には83.4%になるという予測もあります。

2.4. 携帯電話は何処へ

それでは、携帯電話は今後どうなっていくのでしょうか。

もちろん、半導体技術の進歩と共に、携帯電話の機能は今後も進化を続けるでしょう。それに伴って扱うデータ量も増えることから、低消費電力化や電池の性能向上も必要になります。あるいは燃料電池*23が実用化されて充電の必要がなくなる、という可能性も考えられます。

携帯電話が軽量化を競っていた頃、携帯電話の中身は「電話」のためのハードウェアで占められていました。しかし多機能化が始まって、デジタルカメラやICカード、ワンセグの機能を搭載するとなると、そうはいきません。限られた大きさの携帯電話の中に種々の機能を組み込むには、必然的に電話用のハードウェアを小さくしなくてはなりません。電話のためのハードウェアを小型化して、空いたスペースにデジタルカメラを、ICカードを、GPSを、もっと小さくしてテレビのチューナーを組み込んできたのです。

そしてついに、「電話」のハードウェアは1チップのICに集積されようとしています。

かつては一部屋を占領していた巨大なコンピュータが、真空管からトランジスタになり、ICに変わり、さらに1チップのMPUとなったとき、コンピュータに何が起こったか——。それは第5章 コンピュータの歴史 や、前節「家中どこにも、コンピュータ」で見てきた通りです。

同じことが携帯電話にも起ころうとしています。

1チップ化された携帯電話の通信モジュールは、あらゆる機器に容易に組み込むことができます。たとえば電気・ガス・水道のメーターは、各世帯にほぼ1個ずつあると考えていいでしょう。月に一度、使用量のデータを送信させれば膨大な検針作業が自動化できます。他にも組み込めそうなものはたくさんあります。自動車やカーナビ、自動販売機、ノートパソコン、ゲーム機、携帯音楽プレーヤ、デジタルカメラ…。

携帯電話は姿を消して、もっともっと活躍しそうです。

*23 水素と酸素から水が生成される化学反応 ($2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$) などから電力を取り出す電池。携帯機器用燃料電池としては、エタノールを燃料としたものなどが研究されている。燃料を供給すれば継続的に電力が取り出せ、二次電池のように充電する必要がない。

3. 交通とコンピュータ

現在はあらゆる分野で情報化が急速に進展していますが、道路交通も例外ではありません。ITS（高度道路交通システム：intelligent transport systems）は、最先端の情報通信技術を用いて人と道路と車両との間で情報を交換することにより、交通事故、渋滞などといった道路交通上の諸問題を解決するための新しい交通システムです。1997年に策定された「高度情報通信社会推進に向けた基本方針」を受け、国土交通省、総務省、経済産業省、警察庁を中心に次の9つの分野について開発が進められています。

1. ナビゲーションシステムの高度化 VICS*²⁴によるナビゲーションの高度化等
2. 自動料金収受システム ETC*²⁵による料金所のノンストップ化等
3. 安全運転の支援 AHS*²⁶、DSSS*²⁷、ASV*²⁸等による危険警告・自動運転等
4. 交通管理の最適化 UTMS*²⁹による経路誘導、公共交通優先信号制御等
5. 道路管理の効率化 工事情報等の提供、特殊車両管理等
6. 公共交通の支援 TDM*³⁰等の公共交通の運行状況の提供
7. 商用車の効率化 効率的な配車計画の支援等
8. 歩行者等の支援 歩行者等の危険防止や経路・施設案内の提供等
9. 緊急車両の運行支援 緊急時自動通報、緊急車両の経路誘導等

いずれも複雑な情報収集、通信、情報処理を要する大規模なシステムで、まだ開発が緒に付いたばかりのものもあれば、すでにかなり普及しているものもあります。ここではこの中から、比較的普及して目にする機会も多くなったナビゲーションシステムと道路交通情報通信システム（VICS）、自動料金収受システム（ETC）、公共交通優先信号制御（PTPS）などを紹介します。

3.1. GPS とナビゲーションシステム

GPS（全地球測位システム：global positioning system）は人工衛星を使って正確な位置を求めるシステムで、1973年にスタートした米国の国防航法衛星システム（defense navigation satellite system）がその前身です。1978年に最初の衛星が打ち上げられて運用が始まり、1993年に民間利用が認められました。GPSの受信機は次第に小型化され、ナビゲーションシステム（car navigation system）や、携帯電話にも内蔵されて利用されるようになりました。

*24 道路交通情報通信システム：vehicle information and communication system

*25 ノンストップ自動料金支払いシステム：electronic toll collection system

*26 走行支援道路システム：advanced cruise-assist highway systems

*27 安全運転支援システム：driving safety support systems

*28 先進安全自動車：advanced safety vehicle

*29 新交通管理システム：universal traffic management system

*30 交通需要マネジメント：transportation demand management

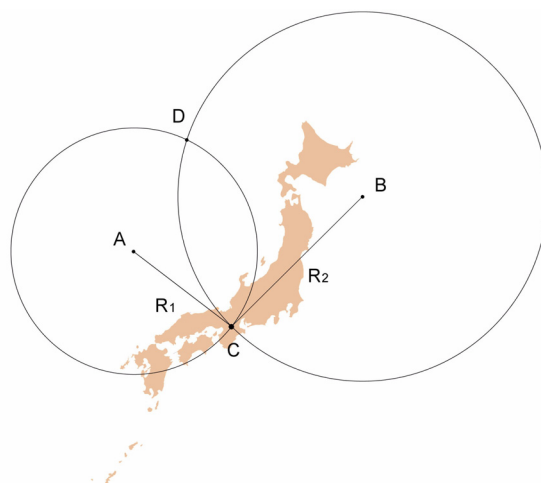


図 7.20 GPS の原理

GPS は大がかりで複雑なシステムですが、基本的な原理はそれほど難しくありません。

試しに、コンパスで円を二つ描いてみます。図 7.20 の A という位置に針を立てて、半径 R_1 の円を描きます。次に B に針を立てて半径 R_2 の円を描くと、二つの円は C と D で交わります。C と D の位置は、A, B の位置と半径 R_1, R_2 で決まります。したがって、A と B の位置が分かって R_1 と R_2 が分かれば、C (と D) の位置は計算できます。これが GPS の原理です。A と B には人工衛星があり、C には道に迷って「ここは何処？」と途方に暮れている車があります。

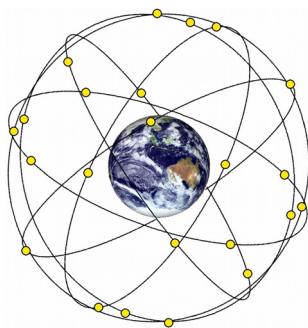


図 7.21 GPS 衛星

実際には地球は球形で山や谷があり、しかも人工衛星は地上 2.1 万 km の軌道にあります。高さも含めて三次元で計算しなければなりませんから、話はもう少し複雑になります。また R_1 と R_2 の長さは、それぞれの人工衛星から発射された電波がどれだけ遅れて届いたかを測って計算しますが、そのためには A、B、C、すなわち人工衛星と車が、非常に正確で、しかもピッタリ合っている時計を持ってはなりません。人工衛星にはセシウムまたはルビジウム原子時計などの超正確な時計が何台か積み込まれていますが、車にはそんな時計はありません。時間合わせから始めなくてはならないので、更に複雑な計算が必要になります。

しかし計算ならコンピュータが得意です。人工衛星の軌道の情報から電波が送信されたときの人工衛星の位置と距離を計算し、現在地 C の位置を教えてください。

地球が平面で、車の時計も正確なら図 7.20 のように人工衛星は 2 個あればいいのですが、3 次元で、時間も合わせて位置を計算するには、4 個の衛星からの電波が必要です。そのため GPS 衛星は図 7.21 のように、6 種類の軌道に 4 個ずつ、計 24 個配置されていて、通常同時に 4 個以上の衛星の電波が受信できるようになっています。



図 7.22 カーナビゲーションの画面例

しかし、こうして得られた位置情報は、たとえば東経 $135^{\circ} 39' 59.7''$ 、北緯 $34^{\circ} 49' 04.3''$ という数字です。このままでは人にはチンプンカンプンですから、これを地図に表示しなくてはなりません。カーナビゲーションの地図データはベクター形式で保存されていて、直線なら両端点の緯度経度の値などの情報で構成されています。

また地図のデータには、道路そのもののデータの他に建物、駅やガソリンスタンドなどの目録物 (landmark)、地名や施設名を表す文字データなどが個別に管理され、必要に応じて重ね合わせて表示されます。こうして自車の位置を計算して、現在地を画面の中心にして表示されたカーナビゲーションの画面例が図 7.22 です。

地図データの記憶装置も最初は CD-ROM でしたが、やがて DVD、ハードディスク、メモリーカードへと進化し、より詳細な地図が表示できるようになりました。音声で操作できる機能、目的地の検索機能などの他、音楽再生やテレビ放送が受信できるものもあります。バック時にバックカメラの画面に切り替わるもの、インターネットで地図データを更新できるもの、他の車の走行状況を受信して渋滞の回避やルート誘導に利用するものなど、カーナビゲーションも携帯電話のそれに似て、ますます機能を高めつつあります。

3.2. VICS

3.2.1. VICS とは

VICS (道路交通情報通信システム : vehicle information and communication system) と

は、渋滞や道路工事、交通規制などの道路交通情報を、カーナビゲーションなどの車載機に文字や図形で表示するシステムです。ITSの一環として警察庁、総務省、国土交通省などが共同で推進しており、財団法人道路交通情報通信システムセンター（略称：VICSセンター）がシステムの開発・運用にあたっています。

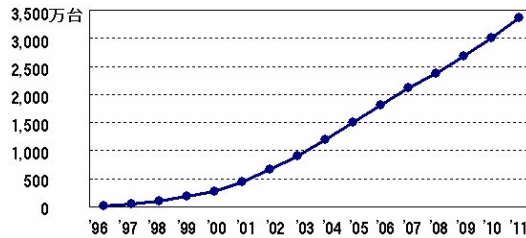


図 7.23 VICS 車載機の累積出荷台数

図 7.23 は VICS 車載機の累積出荷台数の推移を表しています^{*31}。VICS のサービスは 1996 年 4 月にスタートしましたが、当時カーナビゲーション自体がまだ 190 万台しか出荷されておらず、VICS 対応機はその 7%、わずか 13 万台でした。しかしその後徐々に普及が進み、2011 年度には累計 3,364 万台にも達しています。

3.2.2. VICS のしくみ

道路交通情報は、高速道路からは道路管理者が、一般道路からは都道府県の警察が収集し、最終的に財団法人日本道路交通情報センター（JARTIC：Japan road traffic information center）に集められます。この情報はテレビやラジオ、新聞などに提供されていますが、VICS センターではこれをカーナビゲーション用に編集・処理してリアルタイムに送信します。

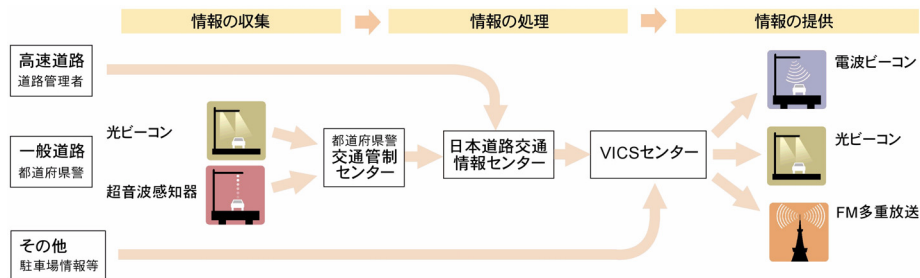


図 7.24 VICS のしくみ

道路交通情報の収集

一般道路の道路交通情報は、主として光ビーコン（optical beacon）や超音波式車両感知器（ultrasonic vehicle detector）で収集されています。

光ビーコンは通常交差点の出口側の各車線に設置されていて、車両感知器としての機能と、

*31 出典：(財) 道路交通情報通信システムセンター（VICSセンター）

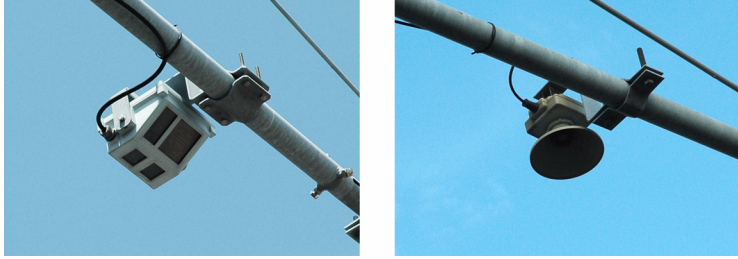


図 7.25 光ビーコン（左）と超音波式車両感知器（右）

車載機と赤外線で双方向通信を行い、より正確な渋滞情報を収集する機能を持っています。超音波式車両感知器は超音波を断続的に発振し、反射波が戻ってくる時間を計測して車両の存在と速度を感知します。この他、画像式車両感知器やマイクロ波式車両感知器、旅行時間計測端末装置なども使用されています。

道路交通情報の表示

現在市販されているカーナビゲーションの多くは VICS に対応しているため、それらのカーナビゲーションを設置すれば VICS 情報を利用できます。

道路交通情報は高速道路に設置された電波ビーコン（radio beacon）や、主要幹線道路に設置された光ビーコン、FM 多重放送（FM multiplex broadcasting）などで送信されます。

電波ビーコンからは、主として高速道路の渋滞や所要時間情報、事故・故障車情報、速度・車線などの規制情報が、光ビーコンからは一般道路についての同様の情報、FM 多重放送からは広いエリアの情報が送信されています。

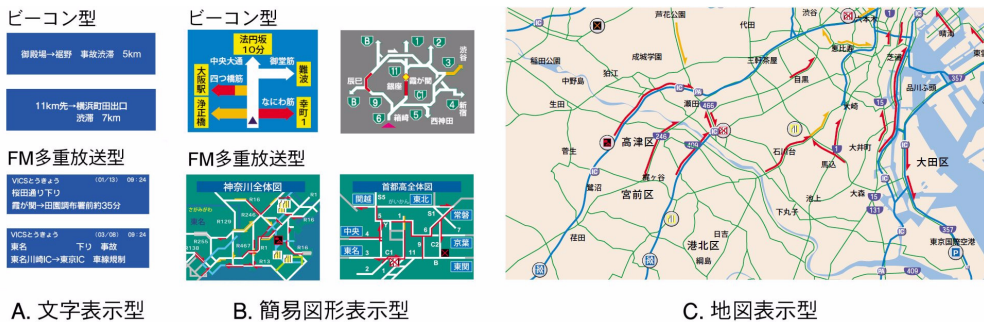


図 7.26 VICS の道路交通情報

VICS の道路交通情報は、A. 文字表示型、B. 簡易図形表示型、C. 地図表示型の 3 種類で提供されています。車載機の機能や走行場所により利用できる情報は異なりますが、情報技術の進歩、車載機の低価格化等により、情報量が多い地図表示型、あるいはすべてのタイプの情報を選択して表示できるものが主流になっています。

3.3. ETC

ETC (electronic toll collection system) は有料道路の料金自動支払いシステムで、料金所のETCゲートのアンテナと、自動車に搭載した車載器との間で利用者情報や通行情報を交信し、利用料金をクレジットカード機能をもっているETCカードで決済します。ETC利用車は料金所をノンストップで通過できます。



図 7.27 ETC ゲート

高速道路で渋滞が起こり易いのは坂道やトンネル、合流部などですが、何といたってもいちばん多いのは料金所です。原因はもちろん、料金支払いのために停車するからです。したがって、ノンストップで料金を支払うことができれば、渋滞の要因の大半を取り除くことができます。

料金所の有人レーンを1時間に通過できる車は約230台ですが、ETCレーンは約800台です。ETCは料金所のレーンを3.5倍に増やすのと同じ効果があります。

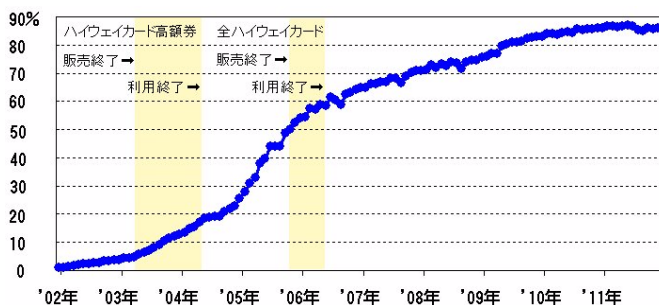


図 7.28 ETC 普及率の推移

ETCは2001年11月末に、全国的に運用が開始されました。図7.28は、ETCの普及率の推移を表しています^{*32}。当初は料金所のETC対応レーン自体が少なく、車載器の費用負担もあって、ETCはなかなか普及しませんでした。

また、当時は高速道路の料金割引システムとしてハイウェイカードが利用されていました。これはテレホンカードなどと同じ磁気カードで、3千円から5万円まで5種類が発行されましたが、他の磁気カード同様偽造が多発しました。ハイウェイカードでは割引率が高い高額

^{*32} 出典：国土交通省道路局 | ETC利用状況。

券に偽造が集中し、社会問題化した*³³ ことから廃止されることになり、同時に一般の高速道路利用者向けの料金割引制度が ETC に移ったこともあって、ETC が一気に普及しました。

ETC を利用するには、先ずクレジットカード会社に ETC カードの発行を申請します。次に ETC 車載器を購入して取り付け、車検証情報や暗号鍵情報を車載器に登録してセットアップします。これで準備は完了で、あとは ETC カードを車載器に入れるだけです。

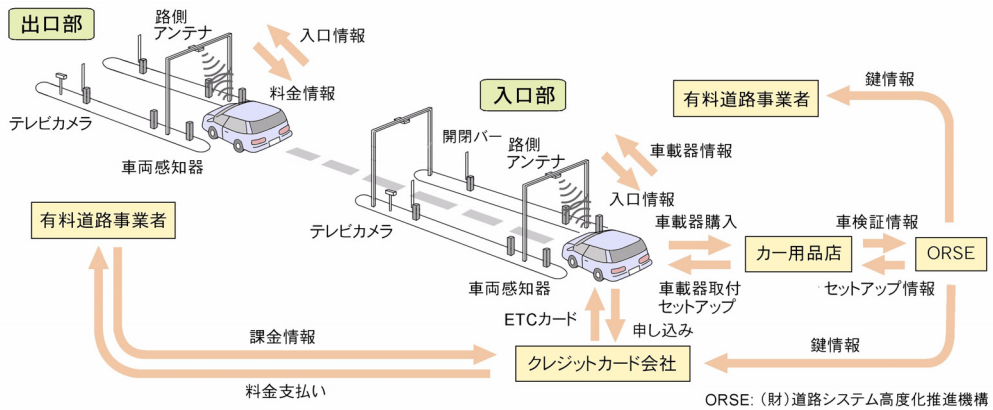


図 7.29 ETC のしくみ

高速道路入口の料金所に近づくと、車載器に登録されている車載器情報が送信され、ETC ゲートの路側アンテナからは通過時刻、料金所番号、TV カメラで撮影したナンバープレート情報などの「入口情報」が送信されます。この入口情報は ETC カードに書き込まれます。

出口では車載器から車載器情報や入口情報が送信され、路側アンテナからは入口・出口の料金所番号、通過時刻、および計算された通行料金などの利用明細情報が送信されます。自動的に計算された通行料金はクレジットカード会社に請求され、有料道路事業者を支払われます。

ETC による料金所渋滞の緩和・解消の経済効果は年間約 3,100 億円と見積もられています。首都高速道路では、ETC 普及率が 50% に達した時点で、本線料金所での渋滞がほぼ解消しています。渋滞の解消は経済効果ばかりでなく、有料道路の料金所周辺の騒音や排気ガスの軽減といった福次効果もあり、車両の停止・発進が減るため燃費の向上、排気ガスや二酸化炭素 (CO₂) 排出量の削減も期待でき、社会や地球環境保全にも貢献する、とされています。

3.4. PTPS

PTPS (公共車両優先システム : public transportation priority systems) は UTMS*³⁴ のサブシステムで、光ビーコンを利用してバスなどの公共車両を優先的に走行させます。バスの通行を円滑にすることによって定時運行を確保し、バス利用者の利便性を向上させ、バスの

*³³ 偽造ハイウェイカードによる累計被害額は約 250 億円に達するといわれています。

*³⁴ 新交通管理システム : universal traffic management system. 「交通とコンピュータ」(185 ページ) 参照。

利用を促して交通量を抑制することがその目的です。PTPS が初めて導入されたのは 1995 年（札幌市）ですが、2010 年度末では全国 41 の都道府県で PTPS が導入、運用されています。

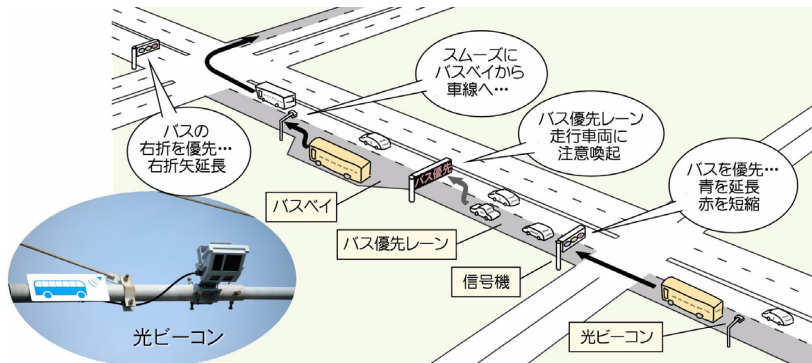


図 7.30 公共車両優先システム

バスには光ビーコンと通信するための車載機が搭載され、バス専用の ID を送信します。光ビーコンが受信したバスの走行情報は交通管制センターに送られ、進行方向の信号機の青時間を延長させるなど、バスができるだけ信号機で停車しないように、あるいは停車時間が短くなるように信号機を制御します。バスが右折する場合は右折矢表示時間を長くしたり、バス専用車線を走っている一般車両には、警告メッセージを表示して車線変更を促したりもします。

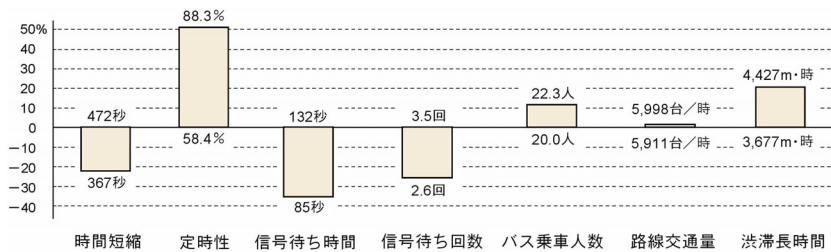


図 7.31 PTPS の導入効果調査例

図 7.31 は PTPS の導入効果の調査例^{*35} で、PTPS 導入前と、導入 1 ヶ月後を比較したものです。バスの運行所要時間は導入後 22%短縮、定時性（ここでは運行ダイヤ 7 分の路線に対して 1 分以内の遅れで到着する割合）は約 50%増加、バスの信号待ちの時間と回数もそれぞれ 35%、25%減少しています。バス乗車人数も 10%程度増加しています。また一般交通に対しては、20%の渋滞長時間^{*36} の増加などの影響があります。

この調査例は、千葉市のバス路線 2km について、導入前と 1 ヶ月後の 3 日間のデータを平均したものです。導入効果を正しく評価するにはもっと長期かつ広範囲の調査事例が必要ですが、概ねの傾向は把握できそうです。

*35 出典：千葉県警察 新交通管理システム PTPS 調査報告。

*36 渋滞の長さ×渋滞時間の積（m・時）。

4. 電子商取引と電子マネー

4.1. 電子商取引

インターネットを利用した電子商取引、ネットショッピングやオークションが盛んになってきました。

電子商取引 (electronic commerce) は、企業間の取引である「B to B」(business to business)、企業と消費者間の「B to C」(business to consumer)、消費者同士の「C to C」(consumer to consumer) の三つに分類することができます。企業間の電子商取引、B to B は従来より小規模ながら専用線や VAN^{*37} などのネットワークを用いて行われていましたが、インターネットの普及とともに、一般消費者を対象にした B to C、オークションなどに代表される C to C などが急激に成長しました。

電子商取引は販売業者と消費者の双方にメリットがあります。販売業者のメリットは

1. 店舗や在庫費、人件費、広告費などのコストが削減できる。
2. 商品回転率や一人当たり売上高などの生産性が向上する。
3. 納期短縮や豊富な商品数など、顧客サービスを充実できる。

などであり、一方消費者にも、

1. 安価に購入できる場合が多い。
2. 商品や価格の比較検討が容易。
3. 店まで行かなくてもよい、地方の地理的な制約がない。

などのメリットがあります。

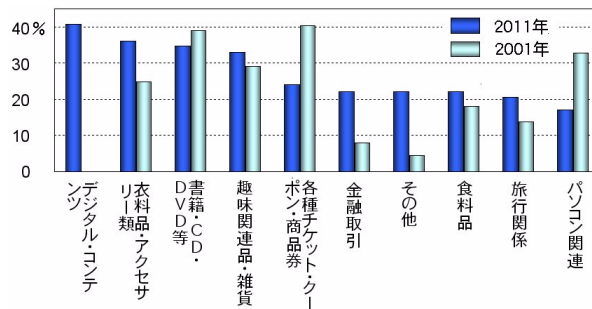


図 7.32 インターネットにより購入・取引した商品・サービス

図 7.32 はネットショッピングの内訳です^{*38}。音楽のダウンロードなど、10年前にはなかったデジタルコンテンツの購入が一番多いのが目を引きます。かつてはチケットや書籍CD、パソコン関連商品、趣味・服飾雑貨などを中心に利用されていたのに対して、近年はさまざまな品目がネットを通じ広く購入されるようになりました。しかしネット詐欺や個人情報漏洩などの問題点が残されていて、ネットショッピングに不安を抱く人も少なくありません。

^{*37} value added network. 付加価値通信網。

^{*38} 出展：平成 23 年版 情報通信白書、平成 13 年 通信利用動向調査。

4.2. ネットショッピングの決済の現状

図 7.33 は、ネットショッピングの決済がどのような形で行われているかを表しています*39。クレジットカード決済が 44.3%、代金引換 15.6%、あとは銀行振込と郵便振替が 15.3%、コンビニ決済 9.3%。これらはどこでも使われている従来型の決済方法で、合計すると 85%になります。目新しいといえばオンライン音楽配信などで使われている課金でしょうが、まだ 6.3%にすぎません。

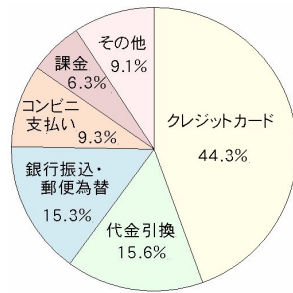


図 7.33 ネットショッピング決済法

ネットショッピングを物品（食品・雑貨・趣味用品など）と、サービス（旅行・チケット・金融など）、そしてデジタルコンテンツ（映像・音楽・電子書籍など）に分けて決済方法を見たのが図 7.34 です（色分けは図 7.33 と同じ）。

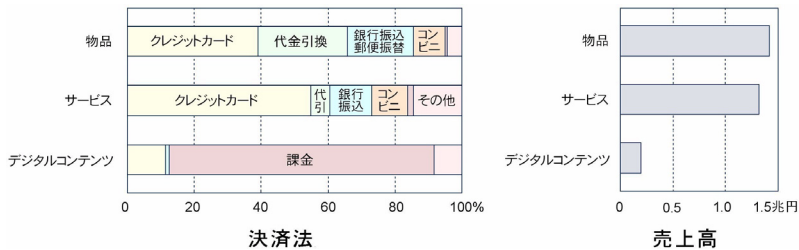


図 7.34 品目別の決済法と売上高

目を引くのはデジタルコンテンツの決済の 80%近くが課金だということですが、売上高は 1,970 億円、ネットショッピング全体の 6.7%です。近年スマートフォンやタブレット型端末の普及が著しいことから、今後このような決済方法が増えるものと思われます。しかし課金にしても、最終的にはたいてい預金口座から引き落としたりクレジットカードで決済することになるので、銀行振込の変形版といえなくもありません。

ショッピングは電子化されましたが、決済の電子化はもう少し先ようです。

*39 平成 21 年 消費者向け電子商取引実態調査（経済産業省）

4.3. 電子マネー

商品を注文して商品受け取ると代金を支払わなくてはなりません。これを決済 (settlement) といいます。

決済はもちろん「お金」で行われますが、お金はもともと家畜や穀物などの実用品 (物品貨幣: material money) であったものが、小さくて丈夫で便利な金属 (鑄造貨幣: coined money) へ、さらにそれ自身にはほとんど価値がない紙幣 (信用貨幣: faith money) へと変わってきました。

紙幣はただの紙切れにすぎませんが、誰でも紙幣と商品を交換するのは当たり前と考えています。これは紙幣にはその価値が変わらない、つまり今日 1,000 円の本を 1 冊買った千円札は、明日も 1,000 円の本を 1 冊買えるという性質と、紙幣はどこでも通用する、1,000 円の本を千円札と交換した本屋さん、スーパーに行けばその千円札で 1,000 円の牛肉を買える、という性質があるからです。

したがって、そのような性質さえあれば、紙幣でなくても、お金のように通用します。電子マネー (electronic money) も、その価値が変わらず、だれでも「それなら商品と交換してもよい」と思えるものであれば、カードに組み込まれた IC チップに記録されている 1 万円という「データ」でも構わないのです。お店の機械にカードを入れて 1,000 円支払いの操作をすると、カードの残高が 9,000 円になって、お店の機械に 1,000 円のデータが増えるのなら、これで決済が行われたことになります。ただし、このデータは簡単にコピーしたり偽造したりできず、価値も変わらない、今日の 1,000 円は明日も 1,000 円として使える、などの性質が必要なのは言うまでもありません。このような性質が整っていれば、物品貨幣から信用貨幣に変わったのと同じように、「電子貨幣」も通貨として使うことができます。

しかし、現在はまだ、電子貨幣がこのような意味で、紙幣と同じ様に流通するまでには至っていません。

4.3.1. Edy

^{エディ}Edy は上述のような、厳密な意味での「電子貨幣」ではありませんが、現金やクレジットカードなどから入金するプリペイド型の「電子マネー」です。すでに 6,420 万枚以上のカードが発行され、加盟店は 267,000 店と、現在最も広く利用されています*40。10cm 程度の近距離通信ができる^{フェリカ}FeliCa*41 という非接触 IC カード技術を利用していますから、読み取り機にかざすだけで料金の精算ができます。FeliCa は読み取り機の電波の電力で動くため、内部に電池などがなく、したがって充電の手間も、電池が切れて使えなくなることもありません。会員証や学生証に Edy の機能を組み込んだ一体型もあります。2012 年 6 月に楽天 Edy という名称に変わりました。

*40 2011 年 4 月現在。

*41 13.56MHz の電波で 100~400kbps の双方向通信ができる非接触 IC カード。

4.3.2. ICカード乗車券

Suica、PASMO、ICOCA、PiTaPaは、いずれも首都圏と関西のJRおよび私鉄各社で利用できるICカード乗車券です。

Suicaは非接触方式ICカードを利用した乗車券の先駆けとして、2001年11月に首都圏でサービスが開始されました。プリペイド方式のカードで、乗車券の購入や精算、定期券機能の他、駅構内や加盟店舗で利用できる電子マネー機能などがあります。

ICOCAも定期券と電子マネー機能を持つICカードで、2003年11月から近畿圏のJR西日本で使用が開始されました。PASMOは首都圏の私鉄（2007年3月）、PiTaPaは関西の私鉄（2004年8月）のICカード乗車券、電子マネーです。PiTaPaはポストペイ方式のため、クレジットカードのような与信審査があり、この点では誰でも利用できる他のカードと異なっています。



図 7.35 ICカード乗車券

図 7.35 は、主な公共交通系の IC カードです。いずれも非接触 IC カード (FeliCa) が採用されていますから、基本的な通信方法などは Edy と同じです。しかし、システムの導入時期や自動改札機のメーカーによってプログラムや仕様に差があるため、1 枚のカードで首都圏から関西まで、JR も私鉄も乗り継いで「スイスイ行こか」とはいかないようです。

4.3.3. iD

iD は 2005 年 12 月に開始された、NTT ドコモの携帯電話を用いた決済サービスです。会員数は 1,650 万人、スーパーやコンビニ、自動販売機などに約 55 万台のリーダー・ライターが設置されています*42。同様の電子マネーに QUICPay、Smartplus、Visa Touch などがあります。

4.3.4. nanaco

nanaco は 2007 年春にスタートした、セブン&アイ・ホールディングスの電子マネーです。全国の 1 万以上の店舗で利用できるだけでなく、nanaco の決済機は Edy や Suica、iD など、他の電子マネーや携帯電話にも対応し、店舗内のセブン銀行 ATM でもチャージできるように、と検討されています。電子マネーとしては後発組ですが、3 年後の 2010 年 3 月に発行件数が 1,000 万件を越えています。

*42 2012 年 3 月現在。

5. ユビキタス・コンピューティング

5.1. ユビキタス・コンピューティングとは

ユビキタス (ubiquitous) とは、「いたるところに存在する」という意味で、ラテン語の“ubique”が語源です。“ubi”は英語の where のような意味で、and に相当する“que”が結びつくと、「神はどこにでも遍在する」という意味の宗教用語になるそうです。

ユビキタス・コンピューティング (ubiquitous computing) は、「利用者に意識されないコンピュータやセンサがいたるところにあって、互いにネットワークで結ばれ、ユーザーがいつでも、どこでも利用できるコンピューティング環境」をいいます。1988年に、ゼロックス社パロアルト研究所のマーク・ワイザー (Mark Weiser) (1952~1999) によって提唱された概念です。



図 7.36 マーク・ワイザー

彼はこれを「多くのコンピュータを物理的な環境全体で利用できるように、しかしユーザーには実質的に見えないようにして、コンピュータの使い方をより良くする方法」*43 であると説明しています。具体的には「タブ」という手のひらサイズのコンピュータ、「パッド」というペン入力ノート型コンピュータ、「ライブボード」という黒板型のディスプレイなどを用いて研究を進めました。また、「アクティブバッジ」という赤外線の出すバッジを各自が身につけておき、今誰がどの部屋にいるかがディスプレイに表示されるシステムの研究などもよく知られています。

一方、東京大学の坂村健 (1951~) は 1984 年、身の回りの環境にコンピュータを組み込んだ「賢い機器」を遍在させ、それらをネットワークで結ぶことによって人々の生活を助ける「どこでもコンピュータ環境」を構築するために、TRON (the real-time operating system nucleus) プロジェクトを立ち上げました。坂村健の研究はマーク・ワイザーに先駆けて着手され、またその成果も少なくありませんが、名前としては“computing everywhere”よりも“ubiquitous computing”の方がよく知られています*44。IBM の pervasive computing、

*43 Mark Weiser: Some computer science issues in ubiquitous computing. (1993)

<http://www.ubiq.com/hypertext/weiser/UbiCACM.html>

*44 坂村氏は後に、「YRP (Yokosuka research park) ユビキタスネットワーキング研究所」を作るときも、「どこでもコンピュータ研究所」にしたかったけれども、「ドラえもんみたい」と反対されて断念したそうです。

MITの TTT (things that think)、ドナルド・ノーマン (Donald A. Norman) の invisible computer など、ほぼ同様の概念を表しています。

この章の第1節、「家中どこにも、コンピュータ」(172ページ)でも述べた通り、たいていの家電製品にはコンピュータが組みこまれています。その意味ではすでに、どこにでも「利用者に意識されないコンピュータ」があります。しかしそれらは全て単独 (stand alone) にそれぞれの機器を制御しているにすぎません。これらのコンピュータがネットワークに接続され、さまざまなセンサもネットワークに接続されて、情報の交換が自由にできなければユビキタス・コンピューティングとはいえません。

一方、前節の VICS (187ページ) や PTPS (191ページ) は、多数のセンサによって収集されコンピュータで処理された交通情報が、車載機に表示されたり、交通信号機等を制御する大規模な情報処理システムでした。VICSの利用者も PTPS で走っているバスも、コンピュータの存在を意識することなく渋滞情報を利用し、バスを運行しています。道路に設置された多数のセンサとコンピュータがネットワークで結ばれることによって、交通情報がいつでも、どこでも利用できる環境が整いつつあります。

5.2. IC タグ

タグ (tag) とは荷札のことで、したがって IC タグは「IC チップに荷札のような情報を書き込んだもの」ということになります。バーコードもタグの一種ですが、バーコードで表せる情報はせいぜい 10 桁の英数字なのに対して、IC タグには 16~256 バイト程度のメモリが内蔵されています。16 バイトでも、現在普及しているバーコード、たとえば JAN *⁴⁵ に比べて、その 1 兆倍の 1 兆倍以上もの大量の情報を表すことができます。

IC タグは RFID (radio frequency identification) ともいい、多くは電池等を内蔵せず、読み取り機の電波の電力を利用して情報を送信します。電池の交換や充電が不要で、寿命も半永久的です。

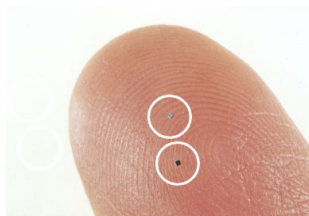


図 7.37 IC タグ用 IC チップ

出典：日立製作所 Web ページ「吹けば飛ぶようなチップが社会を変える」

図 7.37 は IC タグの一例、ミューチップです。指先についているゴマ粒のようなもの (円内) がそれです。この $0.4 \times 0.4\text{mm}$ のチップにアンテナをつけて使います。

*⁴⁵「バーコードリーダー」(66ページ) 参照。

ミューチップは愛知万博の入場券に使われて広く知られるようになりましたが、紙幣に^す漉き込んでおいて偽造を防ぐ、などという応用も期待できそうです。また、食品につけておくと、外部のデータベースと照合して、食品個々の生産経歴や賞味期限などを調べることができます。IC タグのついた食品を冷蔵庫に入れておけば、庫内の食品を賞味期限順に表示させたり、それらでできる料理のレシピを表示させることもできるでしょう。薬瓶につければ投薬の間違いを防げるばかりでなく、同時に飲んではいけない瓶同士が近づくと警告を出す、という応用も考えられています。

回転寿司で IC タグが実用化されている例もあります。回転寿司業界は高付加価値化が進み、商品の種類が次第に増えています。裏面に IC タグを貼り付けた皿を使えば、何種類もの皿を一度に読み取って短時間で精算でき、間違えることもありません。一定時間経過した皿を自動的に排除するなど、コンベア上にある寿司の管理にも使われています。

様々な分野で期待されている IC タグですが、現在の IC タグのコストは数十円程度です。回転寿司のように何度も繰り返して使うなら問題ありませんが、バーコード代わりに商品につけて使うには、もっと単価が下がらなくてはなりません。

もうひとつの課題はプライバシーの問題でしょう。何十種類もの IC タグが非接触で一度に読みとれますから*46、IC タグに直接個人情報を書き込むような使い方をした場合、気付かないうちに、望まない形で読み取られると個人情報漏洩のおそれもあります。

5.3. アドホックネットワーク

アドホック (ad hoc) は、「一時的な」とか「その場しのぎの」という意味のラテン語です。したがって、アドホックネットワーク (ad hoc network) は「一時的なネットワーク」という意味になります。

携帯電話は気軽に通話したり、インターネットに接続したりできますが、これは基地局と通信できるエリア内にいる場合に限られます。携帯電話のネットワークは大規模な通信基盤 (communication infrastructure) によって支えられているシステムです。これに対してアドホックネットワークは、そのような既存の通信基盤を必要とせず、端末間で自律的に相互に接続して通信します。端末の電波自体はそれほど遠くまで届きませんが、それぞれの端末がバケツリレーのように中継することによって、遠くまでデータを送ることができます。

アドホックネットワークは携帯電話やインターネットなど、既存のインフラに依存しない、「中継機能を持つ移動端末のみで構成された一時的なネットワーク」です。普段は心拍、血圧、体温等を測定しているセンサが、異常を検出するとアドホックネットワークによって緊急通報したり、地震などの大災害時には温度、ガス、化学物質、画像等の各種センサをヘリコプターで上空から散布してアドホックネットワークを構築し、建物倒壊、火災、ガス漏れなどの被災地の情報を収集するシステムなどが検討されています。

*46 IC タグの読み取り距離は、数 mm から数 m まで、方式や使用周波数、用途によって様々なものがあります。

第7章 生活と情報技術

しかしアドホックネットワークは参加しているすべての端末がデータの中継機能を持っているため、もし悪意を持った端末が存在すると、盗聴や改ざんといった問題が生じる可能性もあります。データの中継はサーバーが行っているインターネットに比べると、アドホックネットワークでのセキュリティの確保はより困難という課題もあります。

なおアドホックネットワークは、元来 DARPA^{*47} が軍事目的で開発した技術です。コンピュータ技術は多くの場合軍事目的で開発されてきましたが^{*48}、アドホックネットワークも例外ではありません。先の例、「地震などの大災害時、各種センサを上空から散布…」を「戦場で…」と読みかえれば、この技術が軍事上如何に魅力的であるか、想像できるでしょう。

^{*47} 防衛高等研究計画局：defense advanced research projects agency. 米国防総省の研究、開発部門。1972年に「ARPA」から改称されました。「スプートニク・ショック」(128ページ)参照。

^{*48} コンピュータは弾道計算のため(「ENIAC」117ページ)、インターネットは「核戦争に耐えるネットワーク」として開発された(「インターネット」140ページ)等。

第 8 章 コンピュータセキュリティ

コンピュータセキュリティ (computer security) とは、コンピュータシステムを故障や事故、災害、犯罪などの各種の脅威から守り、その安全性 (security)、信頼性 (reliability) を確保することをいいます。

通信、交通、電力などの分野で使われているコンピュータシステムでは、万一トラブルが発生するとその影響が極めて大きいため、予備のコンピュータを待機させておくデュプレックスシステム (duplex system) や、2 台のコンピュータで並行して処理し、結果を照合して誤動作をチェックするデュアルシステム (dual system) などが採用されています。航空管理のシステムでは 3 台のコンピュータで同時処理が行われ、処理結果が一致しない場合は多数決によって判断されています。病院で使われているコンピュータシステムは、停電で止まるようなことがあれば人命に関わる場合もあります。エレベータを制御しているコンピュータシステムは、地震の時こそ的確に動いてもらわなくては困ります。

コンピュータシステムが使われている場所や役割、その重要性や緊急性等によって、コンピュータセキュリティの考え方は自ずから変わります。システムの重要性に応じて、様々なセキュリティ対策が施されています。

さらに、コンピュータセキュリティの問題を一層複雑にしているものに、コンピュータ犯罪があります。特に近年、インターネットとブロードバンド環境の普及に伴い、不正アクセスやコンピュータウイルスなどのコンピュータ犯罪が深刻化しています。

インターネットはもともと学術研究目的のネットワークでしたので、セキュリティに対してそれほど深く考慮されていません。その後商用使用禁止制限が廃止され*1、一般のインターネット利用者の増加と共に悪意ある利用者も増え、その犯罪はますます多様に、巧妙になってきています。残念ながら今日では、無防備のコンピュータでインターネットにアクセスするのは暴挙と言っても過言ではなくなっています。

ここでは、コンピュータの個人的な利用者が、日常遭遇するセキュリティ上の問題を中心に考えていくことにします。そうすると、セキュリティ対策として考えておかななくてはならない課題には、主としてコンピュータの機械的不具合による故障と、不正アクセスやコンピュータウイルスなどのコンピュータ犯罪対策、個人情報の保護、などがあります。

*1 1991 年。「インターネットの普及」(143 ページ) 参照。

1. システムダウン

コンピュータシステムが事故や故障などによって機能しなくなることを、システムダウン (system down) といいます。システムダウンを防ぐ最も効果的な対策は上述のデュプレックスシステムやデュアルシステムですが、これらはパーソナルコンピュータの利用者向きではありません。そこで、できるだけシステムダウンを起こさないように注意しながら、万一システムダウンが起こった場合でも、できるだけ被害を小さくするような措置を講じておくのが現実的な対策ということになります。

1.1. システムダウンの予防

システムダウンはソフトウェアの不具合、ハードウェアの故障、ユーザの誤操作、予期せぬ負荷の集中などによって起こります。したがって、システムダウンを防ぐには、これらの要因をできる限り遠ざけなくてはなりません。

1.1.1. ソフトウェア

パーソナルコンピュータのソフトウェアは、ほとんどの場合、市販のパッケージソフトです。通常これらのプログラムは出荷前に十分チェックされていますから、ソフトウェアの不具合が原因でシステムダウンにいたる可能性は少ないといえます。しかしながら、「ソフトウェアにバグ*2 (bug) はつきもの」というのもまた事実です。市販のソフトウェアでも、特定の操作をしたとき、あるいは突然に、コンピュータが反応しなくなることはあります。こうなると強制終了するしかありませんが、作業中のデータは消失してしまいます。このようなソフトウェアはできるだけ使用しない、やむを得ず使用する場合はデータをこまめに保存する、などの注意が必要です。

ソフトウェアには、フリーウェアやシェアウェアなどの、オンラインソフトもあります。これらのソフトウェアの品質は千差万別で、ソフトウェアの規模もごく単純なものから本格的なものまで、作者も経験の少ないアマチュアから技術者や企業まで、実にさまざまです。したがってシステムダウンを引き起こす可能性もまたさまざまです。フリーウェアやシェアウェアには優れたものも少なくないので、これらを上手く利用することはコンピュータを活用する近道でもあります。ダウンロードしてインストールするに当たって、慎重な選定もまた欠かせません。

1.1.2. ハードウェア

コンピュータのハードウェア、特に MPU 周辺の半導体部分は信頼性が高く、あまり壊れることはありませんが、電源回路や液晶ディスプレイのバックライト*3、挿抜を繰り返すコネク

*2 コンピュータプログラムの間違いをバグといいます。初期のコンピュータで、小さな虫が原因でコンピュータが誤動作したことが語源とされています。

*3 冷陰極蛍光ランプの寿命は約5万時間です。

タなどは比較的寿命が短い部品です。ハードディスクなど、機構部を含む部品も壊れやすいものの代表です。

しかし、たとえば液晶のバックライトならだんだん暗くなる、コネクタなら^{かんごう}嵌合が甘くなる、少し動かすと動作しなくなるなどの兆候があって、それが次第に進行し、やがては使えなくなるという経過をたどることがあります。このような場合は、早めに修理に出して部品を交換しておくことによってトラブルを防ぐことができます。

一方、ハードディスクはある日突然故障する例が少なくありません。急に起動しなくなるなど深刻な症状を示すことが多く、しかも修理に出すと、コンピュータは直っても記憶されていたデータが失われることが少なくありません。「ハードディスクはいつ壊れてもおかしくない」と考えて、次節の対策を講じておくことが必要です。

1.2. 被害を最小に

コンピュータを使う上で、いちばん貴重なのは蓄積されている「データ」です。メールやアドレス帳のデータ、ワープロで作成した文書、デジタルカメラで撮影した写真など、どれも大切なものばかりです。オフィスで使用している表計算ソフトのファイルやデータベースならなおさらです。仮にコンピュータが壊れても、新しく購入すれば作業は続けられますが、失われたデータは決して戻りません。

1.2.1. データのバックアップ

データが万一壊れたり紛失した場合に備えて、別のハードディスクやCD-ROM、USBメモリなどに定期的にコピーしておくことをバックアップ (back-up) といいます。

バックアップは定期的に行う習慣をつけることが大切です。コンピュータの利用度にもよりますが、できれば毎日、少なくとも毎週バックアップするようにします。ハードディスクのトラブルがあった場合でも、毎日バックアップを取っていれば失われるデータは平均して半日分で済みますが、週に一度のバックアップなら損失はその7倍です。もしバックアップの習慣がなければ、被害は計りしれません。卒業論文の提出前日にハードディスクがクラッシュしたら…。考えただけでもゾッとしますが、こういうことは得てして起こるものです。

とはいえ、バックアップを定期的に行うのは結構面倒で、ともすれば怠りがちです。自動的にバックアップをしてくれるソフトウェアがあるので、そういうものを利用するのもひとつの手です。対象フォルダや期間を設定しておく、後は自動的にやってくれます。

1.2.2. ミラーリング

データを定期的に別の記憶装置にコピーしておくのがバックアップですが、これを常時自動的に行うものをミラーリング (mirroring) といいます。ハードディスクの信頼性を向上させる「RAID-1 (redundant arrays of inexpensive disks 1)」^{レイドワン}と呼ばれる技術で、2台のハード

ディスクに同じデータを同時に書きこむため、片方が故障しても、もう1台からデータを取り出して復旧することができます。

最近ではミラーリング専用のハードディスクやミラーリングのソフトウェアが市販されているので、これらを利用すると簡単にデータの保護ができます。しかし誤操作でファイルを削除したり、ウイルスに感染した場合には、どちらのハードディスクにも影響が及ぶので、このような場合、ミラーリングだけではデータの保護ができません。バックアップと組み合わせて使用する必要があります。

1.2.3. 無停電電源装置

コンピュータのメモリにはDRAM*4が使われていますが、これは揮発性メモリですから、停電するとメモリのデータは消失します。したがってコンピュータの使用中に停電になると、それまでに入力したデータは、保存していないものはすべて失われます。

また、運悪くハードディスクに保存している最中に停電した場合は、ファイルが破損する、最悪の場合はハードディスクのデータがすべて読めなくなる可能性もあります。

ノート型のパーソナルコンピュータはこの点に関しては安心ですが、一般的にはこのような事故を防ぐために無停電電源装置（UPS：uninterruptible power supply）を使用します。UPSはバッテリーを内蔵していて、停電になってもしばらくの間はコンピュータを使用することができます。通常は瞬停と呼ばれる数分の1秒程度の停電が多いので、UPSを備えておけば安心です。停電がしばらく続くようであれば、バッテリーから電力が供給されている間に作業中のデータを保存するなどして、システムを安全に終了させます。特に信頼性が求められる場合は、長時間の停電にも対応している発電機を備えたUPSを使用します。

*4 「DRAM」（52 ページ）参照。

2. 不正アクセス

コンピュータやコンピュータネットワークが社会の隅々にまで普及すると、残念ながら、コンピュータを犯罪に利用しようという者が現れます。そして、その被害はしばしば、社会に重大な影響を与えるようになってきました。

不正アクセス (unauthorized access) とは、「システムを利用する者が、その者に与えられた権限によって許された行為以外の行為をネットワークを介して意図的に行うこと」*5 をいいます。

個人のコンピュータがダイヤルアップによってインターネットに接続されていた頃は、インターネットへの接続時間が短いのので、外部からの不正アクセスを考慮する必要はほとんどありませんでした。しかし ADSL や CATV、FTTH などが普及し、インターネットにつながりっぱなしという使用形態が一般的になると事情は変わります。コンピュータが常にインターネットに接続されていればパスワードの解読や、OS のバージョン、侵入口になるポート情報の解析などがいつでもでき、不正アクセスが容易になるからです。

こうして、メールやアドレス帳などの個人情報の盗聴、データの改竄^{かいざん}や破壊、別人を装ってクレジットカードを使うなどのなりすまし、不正プログラムの埋め込みなどの行為が横行するようになりました。侵入の形跡を消したり、次に訪れるのを容易にするためのバックドアを作る場合もあります。

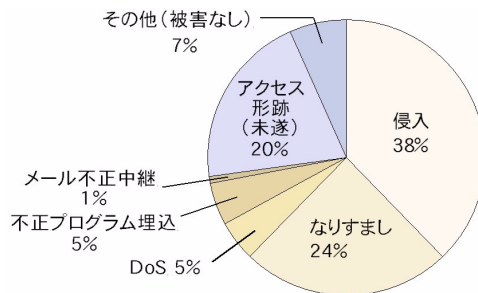


図 8.1 不正アクセス届出件数 (2011 年)

図 8.1 は、IPA *6 (情報処理推進機構 : information-technology promotion agency, Japan.) に届けられた不正アクセス件数の内訳です。法人からの届け出が 33%、個人ユーザーからは 32%、残り 35% は教育・研究・公的機関からの届け出です。常時接続環境の普及によって、個人ユーザーも不正アクセスの攻撃対象になっていることが分かります。

不正アクセス対策としては、ファイアウォール*7 (firewall) 機能を搭載しているブロードバンドルーターや、パーソナルファイアウォール (ソフトウェア) があります。

*5 コンピュータ不正アクセス対策基準 (平成 8 年 通商産業省告示第 362 号)。

*6 「コンピュータウイルス被害の状況」(210 ページ) 参照。

*7 許可のないアクセスを阻止するシステムで、文字通りインターネットの防火壁の役割を果たします。

3. コンピュータウイルス

3.1. コンピュータウイルスとは

コンピュータウイルス (computer virus) は、コンピュータのプログラムやファイルに寄生して破壊的な活動をする不正なプログラムです。

コンピュータウイルス対策基準*8 によるウイルスの定義は、「第3者のプログラムやデータベースに対して意図的に何らかの被害を及ぼすように作られたプログラムであり、次の機能をひとつ以上有するもの。」となっています。

- (1) 自己伝染機能 自らの機能によって他のプログラムに自らをコピーし又はシステム機能を利用して自らを他のシステムにコピーすることにより、他のシステムに伝染する機能
- (2) 潜伏機能 発病するための特定時刻、一定時間、処理回数等の条件を記憶させて、発病するまで症状を出さない機能
- (3) 発病機能 プログラム、データ等のファイルの破壊を行ったり、設計者の意図しない動作をする等の機能

コンピュータのプログラムは、コンピュータが実行可能な範囲内のことであればどんなことでも (たとえ有害であっても)、させることができます。たとえば画面にメッセージを表示したり、一定の時間に音楽を演奏したりすることはもちろん、ファイルを削除したり、コンピュータ内のプログラムやコンピュータそのものを使用不能にしたり、他のプログラムを部分的に修正して自分自身を感染させたり、自分自身のコピーをハードディスクに書き込んだり、あるいは自分自身のコピーをメールに添付して自動的に送信したり…。

3.1.1. コンピュータウイルス

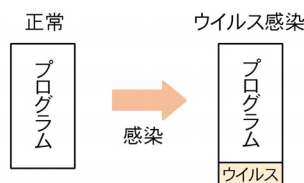


図 8.2 ウイルスの感染

コンピュータウイルスは本来、他のプログラムに寄生するように作られていました。自然界のウイルスが単独では生存できず、人や動物に寄生するように、コンピュータウイルスも OS や他のアプリケーションのプログラムに寄生します。これを「感染」といいます。感染してい

*8 通商産業省 (現: 経済産業省) 告示 第 429 号 (平成 7 年 7 月 7 日)

ることに気付かないままこれらのプログラムを使用すると、ウイルスが活動を始めます。

したがって、かつては「コンピュータウイルスはプログラムに寄生するもので、データには感染しない。プログラムをダウンロードしたり、コピーするときだけにウイルスに注意すればよい。」というのが、ウイルス対策の常識でした。

3.1.2. マクロウイルス

しかし、1995年に文書ファイルに感染する「マクロウイルス」が出現しました。マクロとは、表計算やワープロなどのソフトウェアで、何度も実行する定型的で複雑な操作の手順を登録しておいて、必要な時に呼び出して自動的に実行させる機能です*9。



図 8.3 マクロウイルス

エクセルやワードで使われるマクロは Visual Basic for Applications というプログラミング言語で記述されたプログラムで、文書ファイルに含まれて保存されています。このプログラムにウイルスの機能を持たせたものがマクロウイルスです。インターネットが普及し、電子メールに添付して文書ファイルが交換されるようになると、安心だったはずの文書ファイルもウイルスに感染するようになりました。

3.1.3. ワーム・トロイの木馬

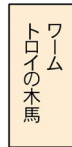


図 8.4 ワーム・トロイの木馬

一方プログラムや文書ファイルに寄生するのではなく、不正なプログラム単独で存在しているものをワーム (worm)、また、自分自身の増殖活動をしないう不正なプログラムをトロイの木馬 (Trojan horse) といいます。これらもコンピュータプログラムですから、実行しない限り実害はありませんが、たいていは自らを無害なプログラムに見せかけて実行させる、あるいはセキュリティホール*10 (security hole) を悪用して、自動的に実行されるような工夫が施されています。

*9 「マクロとは」(229 ページ) 参照。

*10 ソフトウェアのバグ (欠陥) などによって生じたセキュリティ上の弱点。バグのない完全なソフトウェアを作るのは困難で、ソフトウェアの規模が大きくなればなるほどバグも増える傾向があります。

「コンピュータウイルス」とは、本来プログラムや文書ファイルに寄生するものことでしたが、現在ではワームやトロイの木馬なども含めて、不正なプログラムをすべて「ウイルス」と呼ぶことが多いようです。

3.2. 主なコンピュータウイルス

^{ブレイン}
Brain (1986年)*¹¹

パキスタンで作られた最初のウイルスです。IBMのパーソナルコンピュータに感染します。ウイルスというよりはソフトウェアの無断コピーに警告する意図で作られた自己増殖するプログラムで、“Beware of this VIRUS...Contact us for vaccination”というメッセージが電話番号と共に表示されます。

^{カスケード}
Cascade (1987年)

発病すると表示されている文字が滝のように落ちることからこの名前が付けられました。このウイルス以来、コンピュータウイルスという言葉が広く知られるようになりました。

^{コンセプト}
Concept (1995年)

初のマクロウイルス。Word文書からWord文書へとみずからをコピーします。感染するだけで、ファイル破壊などの実害はありません。「文書ファイルに感染させることもできる」ということを証明するために作られた、概念証明 (concept-proof) 型ウイルスです。

^{ラルー}
Laroux (1997年)

初のExcelマクロウイルス。最も繁殖したマクロウイルスのひとつです。

^{ハッピー99}
Happy99 (1999年)

メールに添付されて感染を広げるワーム。感染して発病したコンピュータでメールを送信すると、同じ宛て先に自分自身を添付したメールを送信します。

^{メリッサ}
Melissa (1999年)

Wordのマクロウイルスで、感染するとOutlookのアドレス帳に登録されているメールアドレス50件に、ウイルスを添付したメールを自動的に送信します。アドレス帳を使っていますから、受け取った人には知人から送られてきたメールのように見えます。

^{ラブレター}
LOVELETTER (2000年)

有名なマクロウイルスで、感染するとOutlookのアドレス帳のアドレスすべてにメールを送ります。件名は“ILOVEYOU”、本文は“kindly check the attached LOVELETTER coming from me.”、添付ファイル名は“LOVE-LETTER-FOR-YOU.TXT.vbs”、つい添付ファイルを開いてみたくなるような工夫が随所に施されています。海外では大流行し、大きな被害が発生しました。

*¹¹ 以下、() 内は発見年。

ハイブリス (2000年)

閲覧したウェブページにメールアドレスが書かれていると、そのアドレスに宛てにウイルス添付メールを送信します。Melissa や LOVELETTER のようにアドレス帳を利用する方法では、アドレス帳は頻繁に更新されませんから効果的なのは最初だけですが、閲覧したウェブページからなら新しいアドレスを無尽蔵に入手できます。ウイルスを蔓延させるにはいいアイデアです。

バッドトランス (2001年)

Internet Explorer のセキュリティホール を悪用したワームで、Outlook ではメールを開くだけで、OutlookExpress ではプレビューただけでワームが動きだします。「添付ファイルをダブルクリックしなければウイルスには感染しない」という常識が覆されました。

クレズ (2001年)

これも Internet Explorer のセキュリティホールを利用したワームで、ワクチンソフトを強制的に停止させてしまいます。感染すると Outlook のアドレス帳のメールアドレスすべてに、送信者を詐称したウイルス添付メールを送信します。

フォートナイト (2003年)

メールの本文に、見えない形で、自動的にウイルス本体のウェブページが参照・実行されて感染する仕組みが書き込まれています。Windows のセキュリティホールを利用しています。

エムエスブラスター (2003年)

メールや添付ファイルとは関係なく、インターネットに接続しているだけで感染する可能性があります。これも Windows のセキュリティホールを利用しています。

スウェン (2003年)

セキュリティホールを利用したウイルスがあまりにも多くなったので、それを逆に利用しています。マイクロソフトからの修正プログラムの案内メールを装っていますが、添付されているのは修正プログラムではなく、もちろんワームです。

ネツトスカイ (2004年)

メールの添付ファイルを開くと偽のエラーメッセージが表示されます。感染すると Windows の起動時にウイルスが実行されます。29種類もの亜種*12 が短期間に出現し、これまでで最も蔓延したワームです。

サッサー (2004年)

これもインターネットに接続しているだけで感染するワームです。Windows 2000/XP のセキュリティホールを利用し、メールの受信やウェブページの閲覧をしていなくても感染します。

*12 新しいウイルスが現れると、それを一部改造した類似のウイルスが現れることがよくあります。これを亜種といえます。

^{ボフラ}
Bofra (2004 年)

このウイルスには添付ファイルがなく、メール本文のリンクをクリックすると感染します。添付ファイルがなくても、気は許せません。

^{マイトブ}
Mytob (2005 年)

感染したコンピュータを外部から操ることを目的として作成されたウイルスで、ネットワークを通じて外部の指令サーバと通信し、外部からの指示によってスパムメールを送信したり DoS 攻撃^{*13} などをを行います。このような機能を持ったウイルスをボット (bot) といいます。

^{オートラン}
Autorun (2008 年)

USB メモリ等に自分自身をコピーして感染を広げるウイルスです。ウイルスに感染した USB メモリをパーソナルコンピュータに接続すると、コンピュータにウイルスが感染し、さらに別の USB メモリを接続すると、その USB メモリも感染します。このようにして、USB メモリを介して次々と感染が広がります。

3.3. コンピュータウイルス被害の状況

コンピュータウイルスに感染したり発見したときは、ウイルスの被害の拡大と再発を防ぐために、IPA (情報処理推進機構) に届け出る制度があります^{*14}。図 8.5 は IPA へのウイルス被害の届出件数の推移を表したものです。

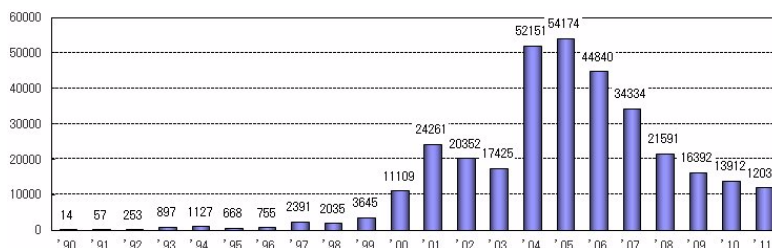


図 8.5 ウイルス感染・発見届出件数の推移

制度が始まってしばらくの間、届け出はごく僅かでしたが、1997 年に届出件数が増えました。これはマクロウイルスの増加が原因です。マクロウイルスは 1995 年に発見されましたが、当時はインターネットが普及し始めたばかりで、電子メールはまだ一般的ではありませんでした。インターネットが次第に普及し、電子メールの利用者が増えるにつれて、メールに添付されて送られてくるマクロウイルスの被害も多くなりました。

それまではプログラムをフロッピーディスクからコピーしたり、ダウンロードしたときにウ

^{*13} サービス拒否攻撃 (denial of service attack)。インターネット上の特定のサーバに大量のパケットを送ることによってサーバの機能を停止させたり、ほかのユーザーとの通信を妨害する攻撃。サービス妨害攻撃、サービス不能攻撃ということもあります。

^{*14} コンピュータウイルスに関する届出制度は、経済産業省のコンピュータウイルス対策基準に基づいて 1990 年 4 月にスタートしました。コンピュータウイルスを発見、または感染した場合は、被害の拡大と再発を防ぐために必要な情報を IPA (<http://www.ipa.go.jp/>) に届け出ることになっています。

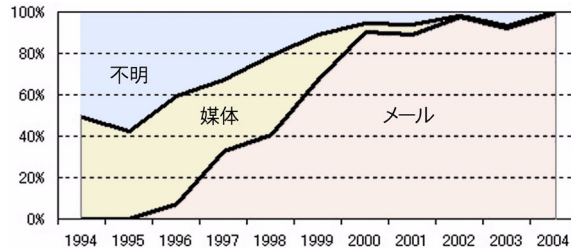


図 8.6 ウイルス感染経路の推移

イルスに感染していましたが、図 8.6 の通り、1997 年からはメールによる感染が急増しています。マクロウイルス増加の時期とも一致しています。

2000 年は前年に比べて 3 倍の感染届出がありますが、この原因は主としてメールの機能を悪用したウイルスの増加です。メールに添付されて感染するばかりではなく、自らウイルスを添付したメールを自動的に送信する機能を持ったものが増えました。

ウイルスもワームもプログラムですから、たとえウイルス添付メールが届いても、アイコンをダブルクリックするなどしてウイルスプログラムを実行しない限り実害はありません。しかし、2001 年には Windows や Internet Explorer、Outlook などのセキュリティホールを利用して、添付ファイルをダブルクリックしなくても自動的に実行されるものが現れました。2001 年に届出件数が激増しているのは、このタイプのものが増加したためです。

2002 年からは届出件数が減少した後、2004 年に再び激増しています。これは Netsky ウィルスに多数の亜種が短期間に発生して大流行したのが主な原因です*15。

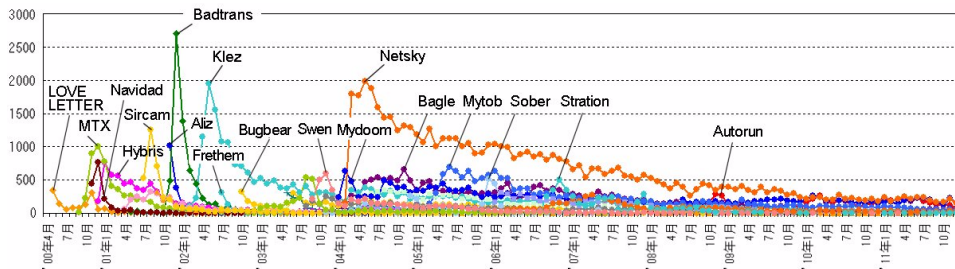


図 8.7 ウィルス別届出件数の推移

図 8.7 は、IPA への届出件数が多いものについて、ウイルス別の届出件数の推移を表しています。この図で目立つのは Badtrans や Klez、Netsky などです。特に Badtrans は、2001 年 12 月に 2701 件もの感染届けがありました。以後は 1381 件、649 件、448 件… と減少していますが、このように、新種のウイルスが現れると一気に被害が広まり、やがて情報や対策が知れ渡ると共に、次第に減少するのが一般的なパターンです。

Klez はピーク時の届出件数こそ 1943 件と Badtrans には及びませんが、その後も数 100 件

*15 IPA へのウイルス感染届けの様式が簡素化されたことも、届けが増加した要因のひとつです。

程度を維持したまま現在に至っています。Netskyはその傾向が更に顕著で、漸減傾向を示しながらも届出件数では5年近く首位を保ち続けています。届出総数で見るとBadtransは7,477件なのに対してKlezは27,350件、Netskyは60,677件もあります*16。このように、いつまでたっても終息しない傾向は年々強まっています。その背景にはおそらく、興味本位でウイルス亜種を作る者と、セキュリティ対策を施さないコンピュータの個人ユーザーの存在（次節参照）があります。

3.4. コンピュータウイルス対策

3.4.1. コンピュータウイルス対策の目的

コンピュータウイルス対策の目的は、

1. 自分のコンピュータへのウイルス感染を防ぐ。
2. ウイルスに感染することによって、他のコンピュータへの加害者となることを防ぐ。

の2点です。そして、全てのコンピュータが、きちんとウイルス対策を施せば、ウイルスそのものを撲滅することができます。もちろん実際はそれほど単純ではありませんが、ウイルス対策の励行はウイルス撲滅の第一歩で、これに寄与するという視点も欠かすことはできません。

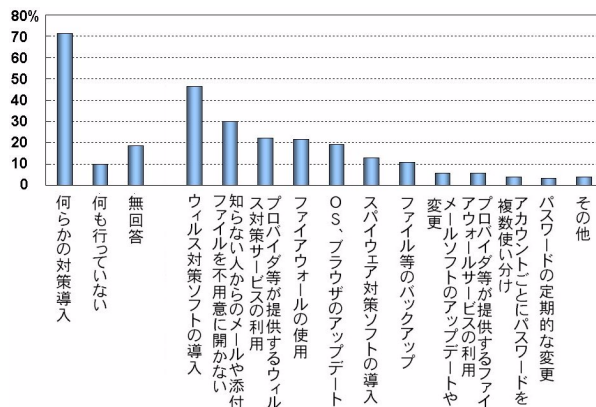


図 8.8 世帯におけるセキュリティ対策の実施状況（複数回答）

図 8.8 は、インターネット利用者のセキュリティ対策の実施状況を表したもので、「ウイルス対策ソフトの導入」から右はその内訳（複数回答）です*17。71.3%の所帯が何らかの対策を実施している一方で、セキュリティ対策の未実施所帯は10%、無回答が18.7%となっています。

図 8.9 はセキュリティ対策実施状況の推移です。図 8.8 の「何らかの対策を導入」、「何も行っていない」、「無回答」が、どのように推移してきたのかを表しています。これによると「何らかの対策を導入」している所帯はおおむね漸増、「何も行っていない」、「無回答」の所帯は漸減の傾向にあります。セキュリティ対策未実施や無回答をあわせて約30%もあるということは、3~4軒に1軒は鍵もかけずに外出しているようなものです。山奥や無人島でない限り玄関

*16 2011年12月までの届出件数累計。

*17 出典：総務省 平成22年 通信利用動向調査。

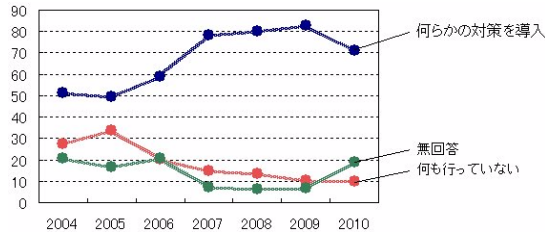


図 8.9 セキュリティ対策実施状況の推移

に鍵は当たり前なのと同じように、ネットワークに接続されているコンピュータにも鍵が必要です。すべてのコンピュータがウイルス対策を施してあれば、ウイルスも容易に感染できません。すべてのコンピュータがウイルス対策を施せば、ウイルスを撲滅することも不可能ではありません。

3.4.2. コンピュータウイルス対策

コンピュータウイルスの対策として、次のようなものが考えられます。

1. ワクチンソフトをインストールし、常に最新の定義ファイルに更新する。

ワクチンソフト (vaccine software) はウイルスを検出し駆除するためのソフトウェアで、ウイルス対策の定石です。今までに発見されたウイルスの特徴を収録したデータベース (定義ファイル: definition file)*¹⁸ を参照しながら検査します。

新種のウイルスは毎日のように現れています。したがって、定義ファイルを常に更新していなければ「頭隠して尻隠さず」になりかねません。最近のワクチンソフトは自動的に定義ファイルを更新してくれますから、面倒な操作は要りません。

2. ソフトウェアのセキュリティホールをふさぐ。

たとえウイルスメールがやって来ても、「添付ファイルをダブルクリックしなければ発病しない」というのは昔の話。現在はセキュリティホールを悪用したものが増え、ダブルクリックしなくても発病することが少なくありません。また、インターネットに接続しているだけで感染するウイルスもあります*¹⁹。セキュリティホールはソフトウェアのバグですから、対策はこれを修正することです。「セキュリティパッチをあてる」ともいいます。

セキュリティホールはマイクロソフト社の Windows、Internet Explorer、Outlook などに特に多く、Internet Explorer の「ツール」メニューから Windows Update を実行すれば、修正プログラムをダウンロードしてインストールすることができます。

3. セキュリティホールが多いソフトウェアを使わない。

*¹⁸ パターンファイル (pattern file)、署名ファイル (signature file) ともいいます。

*¹⁹ MSBlaster、Sasser など。「主なコンピュータウイルス」(209 ページ) 参照。

前項の「セキュリティホールをふさぐ」は、見つかったセキュリティホールをその都度修正する、いわば「臭いものにフタ」的対策です。本格的な対策は「元から絶つ」、すなわちセキュリティホールがあるソフトウェアを使わないことです。Internet Explorer や Outlook はセキュリティホールだらけにもかかわらず、多くの人が使っているため、ほとんどのウイルスはこれらをターゲットにして作られています。したがって、セキュリティホールが少なくなるまでは危険なソフトウェアを使わないのが、最も簡単でかつ効果的な対策です*20。

4. 定期的にデータをバックアップする。

データのバックアップは、万一ウイルスに感染しても被害を最小限に止めるのに役立ちますが、ハードディスクが故障したときにデータの消失を防ぐためにも必要です。普段から定期的にデータをバックアップする習慣をつけておくことが大切です*21。

5. その他

以上の他に、メールに添付されてきたファイルは不用意にダブルクリックしない、メール本文中に書かれているウェブサイトへのリンクも信頼できるもの以外はクリックしないなども基本的な注意です。パーソナルコンピュータの動きがなんとなく遅くなった、勝手に再起動されたなど、異常の兆候が現れたときはウイルスチェックを試みるのもいいでしょう。最近プロバイダのウイルスメールのチェックサービスが増えてきましたので、念のため利用するのもいいかもしれません。しかし、新種のウイルスへの対応が遅れる例もあり、これで安心していかえって大怪我をすることもあります。

いずれにしても、上記1~4の対策が励行されていれば、これらは必ずしも必須ではありません。

*20 Windows も使わなければ理想的。Windows 以外の環境で感染するウイルスはほとんどありません。

*21 「データのバックアップ」(203 ページ) 参照。

4. 個人情報の保護

インターネットを利用する機会が増えるに伴って、私たちの生活は便利になった反面、これまで考えられなかった新たなリスクも生まれました。前節のコンピュータウイルスをはじめ、暴力・ポルノ・覚醒剤などの有害サイトやフィッシング詐欺、スパムメール、個人情報の漏洩など、さまざまなものがあります。

ここでは、インターネットというオープンなネットワークの中で、個人情報をいかに守るか、いかに守られているか、という点について考えていきます。

4.1. パスワード

正式の利用者以外の者がコンピュータを扱うことによって、データが改竄^{かいざん}されたり不正使用されることを防ぐために、ID (identifier) やパスワード (password) が使われています。

ID は個人を識別するための符号で、コンピュータシステム上の「名前」です。コンピュータは、誰がログイン (login) しようとしているのかを ID によって知ります。

一方パスワードは、正式の利用者本人であることを証明するための文字列です。パスワードは本人だけが知っていて、正しい ID とパスワードが入力されると、コンピュータは正式の利用者であると判断してシステムへのアクセスを許可します。パスワードはコンピュータシステムの扉を開く鍵のような役割を果たしますから、パスワードが他人に知られることは、鍵が盗まれたのと同じです。厳重に管理しなければなりません。

通常パスワードには 8 桁程度の英数字文字列が使用されますが、盗難を防ぐには、

1. 許される範囲で、できるだけ長い文字列にする。
2. 名前や、辞書に載っている単語は使用しない。
3. 数字、アルファベット (大文字・小文字)、記号などをランダムに組み合わせる。
4. 複数のユーザ ID を持っている場合は、それぞれ異なるパスワードを設定する。
5. パスワードは随時変更する。

などが効果的です。とはいえ、自分でも覚えられないようなパスワードを作って、手帳や携帯電話などにメモしないといけないようではかえって危険です。覚えやすく他人に類推されにくいパスワード作るには、フレーズの頭文字を並べる方法があります。たとえば、“Twinkle, twinkle, little star, How I wonder what you are !” → “TlsHIwya” など。

キャッシュカードなどに用いられている暗証番号 (PIN : personal identification number) もパスワードの一種ですが、多くは 4 桁の数字で構成されているため、しばしばパスワードの盗難事件が発生します。生年月日や電話番号、自動車のナンバー、1234 や 7777 といった特徴のある数字列などは覚えやすいので安易に使われるようですが、覚えやすいということは同時にまた類推しやすいということでもあり、たいへん危険です。

パスワードの他に、指紋や顔、虹彩^{*22}、指先や手のひらの静脈のパターンなどを利用する更

*22 瞳孔の周りの円形の膜。

に進んだ方法も使われています。このような、本人しか持たない身体的特徴によって本人であることを確認するものを生体認証 (biometrics) といいます。

4.2. 暗号化

インターネットの情報は、ルータからルータへ、バケツリレーのようにして伝えられますから、途中でパスワードや電子メールなどの個人情報が漏洩したり、改竄されたりする可能性がないとはいえません。そのため、重要な情報は第3者に理解できないように、暗号化 (encryption) されています。暗号といえば軍事や外交、あるいは映画や小説の世界の話の思いがちですが、インターネットや携帯電話が広く普及している今日では、数十年前には考えることもできなかった高度な暗号技術が日常的に使用されています。

暗号 (cipher) とは、文字を一定の規則に基づいて他の文字や記号に置きかえることで、この「一定の規則」を暗号化アルゴリズム (encryption algorithm)、暗号化に当たって与えるパラメータを暗号鍵 (encryption key) といいます。

暗号には、大きく分けると秘密鍵暗号方式 (private key encryption system) と公開鍵暗号方式 (public key encryption system) の2種類があります。

4.2.1. 秘密鍵暗号方式

秘密鍵暗号方式は従来からある暗号方式で、共通鍵暗号、対称鍵暗号、慣用暗号方式ともいいます。暗号化するための暗号鍵と復号化^{*23} するための復号鍵が同じです。暗号文で通信する場合、鍵はあらかじめ双方で共有しておかなければなりません。鍵が漏洩する可能性があり、絶対に安全ということはできません。

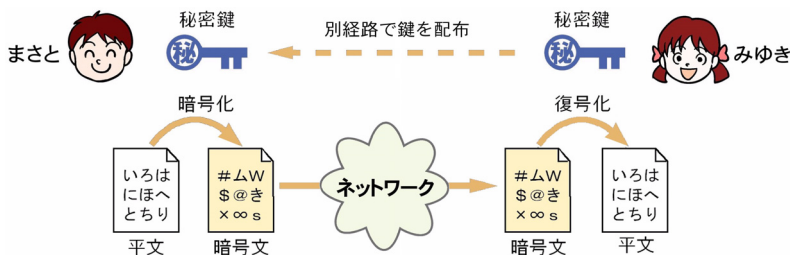


図 8.10 秘密鍵暗号方式

4.2.2. 公開鍵暗号方式

1976年、ホイットフィールド・ディフィー (Whitfield Diffie) とマーティン・ヘルマン (Martin Hellman) によって、暗号の歴史に革命が起きました。公開鍵暗号^{*24} です。

これまで使われてきた暗号はどれも鍵はひとつ (共通) で、秘密にしなければなりません

^{*23} 暗号文をもとの平文 (ひらぶん) に戻すこと。

^{*24} 非対称鍵暗号 (asymmetric key cryptosystem) ともいいます。

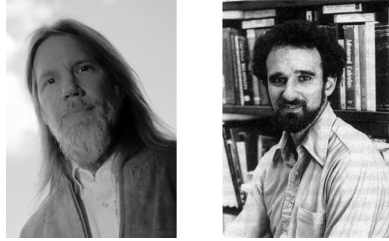


図 8.11 ディフィー（左）とヘルマン

が、公開鍵暗号方式は数学的に互いに関係のある 2 種類の鍵、公開鍵（public key）と秘密鍵（secret key）を使用します。公開鍵はその名の通り、公開します。

公開鍵暗号には、公開鍵で暗号化した文書は対応する秘密鍵でしか復号化できず、逆に、秘密鍵で暗号化した文書は公開鍵でしか復号化できない、という画期的な特徴があります。秘密鍵を文字通り秘密に、誰にも知られないように保管しておけば、公開鍵で暗号化された文書は、秘密鍵を持っている人以外は誰も復号化することはできません。

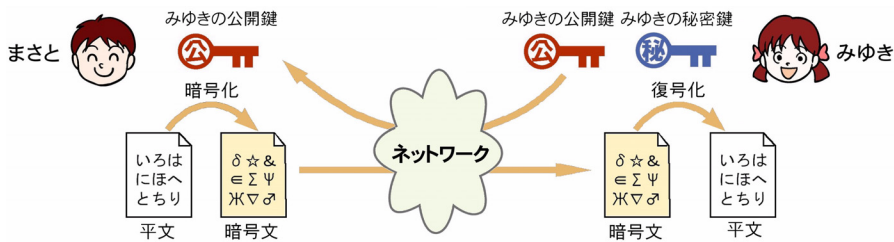


図 8.12 公開鍵暗号方式

公開鍵は公開されているので、誰でも入手できます。もちろん、メールに添付して送ることもできます。図 8.12 ではまさとが「みゆきの公開鍵」で暗号化したメールを送信しています。秘密鍵はみゆきだけが持っています。送られてきた暗号文はこの秘密鍵で復号化します。秘密鍵暗号方式のように、鍵が漏洩する心配がないので安全性がきわめて高くなります。

4.2.3. 電子署名

インターネット上では、電子商取引をする相手が本当に本人なのかどうかわかりません。実社会では相手の顔も見えるし、契約書や領収書には署名や押印がありますが、電子メールの注文書には印鑑も筆跡もありません。このようなとき、公開鍵暗号は情報を秘匿するばかりでなく、相手が本人に間違いのないことを確かめる、署名や認証のために使うことができます。

公開鍵暗号には、公開鍵で暗号化したものは秘密鍵で、秘密鍵で暗号化したものは公開鍵でないと復号化できないという性質があります。誰でも公開鍵で暗号化できますが、復号化できるのは秘密鍵を持っている人だけです。逆に秘密鍵での暗号化は秘密鍵を持っている人にしか

できませんが、この復号化は誰でもできます。



図 8.13 電子署名

いま、まさとがメールを自分の秘密鍵で暗号化してみゆきにするものとします。

みゆきがこのメールを「まさとの公開鍵」で復号化して読むことができたとすると、その暗号化は「まさとの秘密鍵」を持っているまさとにしかできないのですから、このメールは間違いなくまさとから送られてきたものだ、ということが分かります。

秘密鍵で暗号化した文書は、署名や押印と同じ効果があります。これを電子署名 (electronic signature) といいます。

4.2.4. 電子認証

しかし、第三者であるよしおが、自分の公開鍵を「まさとの公開鍵」と偽って公開しておくこともできます。みゆきがもし、よしおの公開鍵をまさとの公開鍵と勘違いすれば、よしおのメールをまさとから来たものと思いこんでしまいます。こうして、よしおはまさとになりすますことができます。ネットワーク社会では相手の顔が見えませんが、このような「なりすまし」を防ぐ仕組みが必要です。

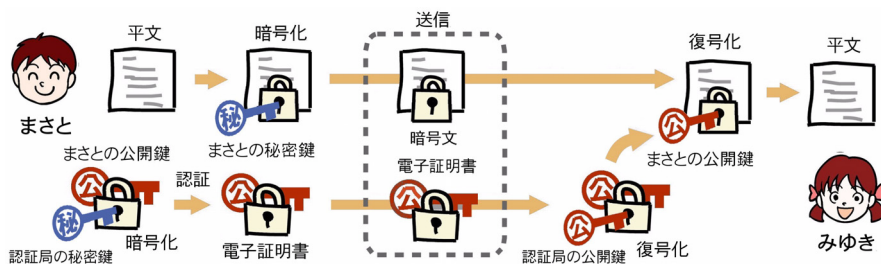


図 8.14 電子認証

そこでまず、まさとは認証局 (certificate authority) に自分の公開鍵を届け出ます。認証局とは電子証明書を発行する機関で、市役所や法務局などのほか、信頼のできる企業などがその任に当たります。認証局は公開鍵がまさとのものであることを確認した上で、認証局の秘密鍵で「まさとの公開鍵」を暗号化します。これが「電子証明書」です。

まさとは自分の秘密鍵で暗号化したメールに、電子証明書を添付して送信します。これを受け取ったみゆきは、まず電子証明書を「認証局の公開鍵」で復号化してまさとの公開鍵をとりだし、それを使って暗号化メールを復号化します。「まさとの公開鍵」を第三者が偽って認証局に届け出ることはできませんから、こうすればまさとから送られたメールであることを証明

できます。これを電子認証 (electronic authentication) といい、電子証明書は実社会の印鑑証明書に相当する役割を果たします。公開鍵暗号を使うことによって、相手の姿が見えなくても、安心してネット上で商取引をすることができます。

4.3. 主な暗号

4.3.1. シーザー暗号

伝えられている最古の暗号は、古代ローマ時代にジュリアス・シーザーが使用したシーザー暗号 (Caesar cipher) といわれています。アルファベットを何文字かずらせることによって暗号化します。

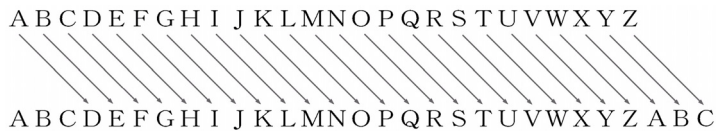


図 8.15 シーザー暗号

図 8.15 の例は 3 文字ずらせたものですが、これを使って “I LOVE YOU” を暗号化すると、“L ORYH BRX” になります。この場合のアルゴリズムは「文字をずらす」、鍵は「3」です。

これなら暗号文が誰かに盗まれても安心ですが、もしこの暗号化アルゴリズムがシーザー暗号であるということが分かれば、これを解読するのは簡単です。アルファベットは 26 文字ですから、暗号として意味のある文字のずらせ方は 25 通りしかありません。ずらせ方を変えながら試していけば、どんなに運が悪くても 25 回目には解読に成功します。

4.3.2. 換字暗号

文字を単純にずらすのではなく、ランダムに並び替えたアルファベットや記号などで置きかえたものを換字暗号 (substitution cipher) といいます。

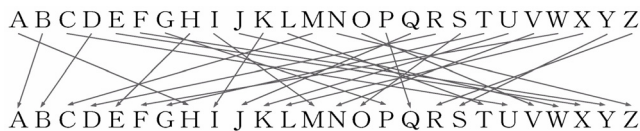


図 8.16 換字暗号

たとえば図 8.16 のように置きかえると、“I LOVE YOU” は “M TVNX SVK” になります。シーザー暗号には 25 種類の鍵しかなかったのに対して、この場合の鍵の種類 (文字の並び換え方) は、なんと 403,291,461,126,605,635,584,000,000 通りもあります*25。

しかし、“M TVNX SVK” では無理ですが、暗号文にある程度の長さがあれば、解読は難しくありません。英語の文章の中では、アルファベット 26 文字がすべて同じ確率で使われて

*25 $26! = 1 \times 2 \times 3 \times 4 \cdots 23 \times 24 \times 25 \times 26 = 403,291,461,126,605,635,584,000,000$

いるわけではなく、表 3.3 (62 ページ) のように、E や T、O といった文字は頻繁に使われるのに対して、Z や Q はほとんど使われていません。これらの文字が暗号文中でどんな文字に置き換えられていようと、E が置き換えられた文字は暗号文中でも使用頻度が高いし、Z が置き換えられた文字は少ないはずです。また使用頻度の高い単語や、…tion のような特徴のある綴りなど、言語が持っている特徴もあわせて解析すると、意外なほど簡単に解読することができます。

Web で学ぶ 【暗号解読】



実際にやってみると、暗号解読は難しくないので納得できます。

Web ページ「暗号解読」では、一文字解読すると暗号文中の同じ文字が一斉に置き換えられます。ゲームのような感覚で暗号の解読ができます。

4.3.3. エニグマ

エニグマ (enigma) は第 2 次世界大戦でドイツ軍が使っていた暗号機です。

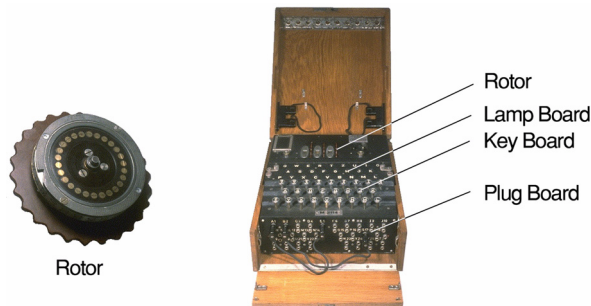


図 8.17 エニグマ

図 8.17 の左にあるのはロータで、円周上に並んだ 26 個の接点が見えていますが、裏面にも同じ数の接点があり、表裏の接点が互いに不規則に接続されています。これによって文字が別の文字に置き換えられます。このロータがエニグマの奥の方に 3 個セットされていて、一文字暗号化すると右端のロータが一文字分回転し、これが一回転すると中央のロータが一文字分回転します。このため、右から順に fast rotor、medium rotor、slow rotor という名前がつけら

れています。

キーボードのキーを押すと、暗号化された文字のランプが光るしくみになっています。

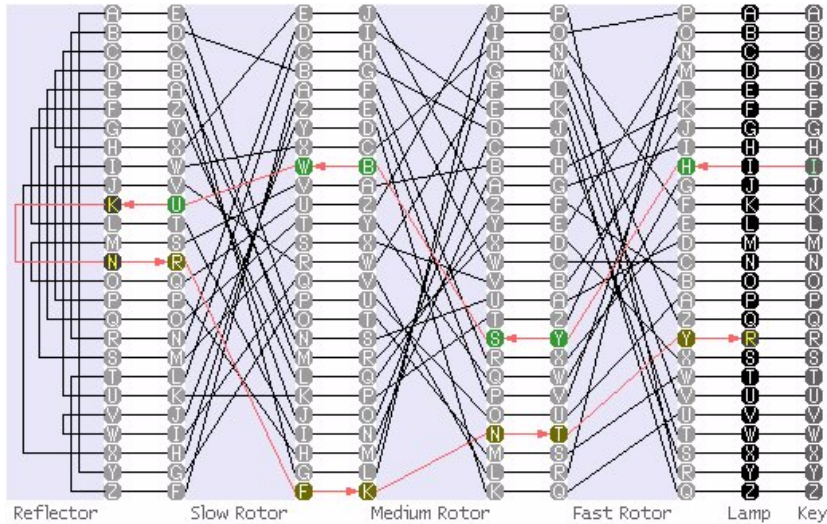


図 8.18 エニグマのしくみ

図 8.18 で、たとえば I のキーを押すと、電流はまず fast rotor に入り、Y の接点から出てきます。次に medium rotor で B の接点へ、slow rotor で U へと変換され、リフレクタ (reflector) で N に移り、こんどは逆方向に F、N を経て、最終的に R のランプを光らせます。こうして、I は R に暗号化されました。一文字ごとにロータが回転するため、同じ文字を続けてタイプしても、次々に違う文字に暗号化されます。いま I は R に変換されましたが、この場合、続けて I をタイプすると Z に、次は Q になります。

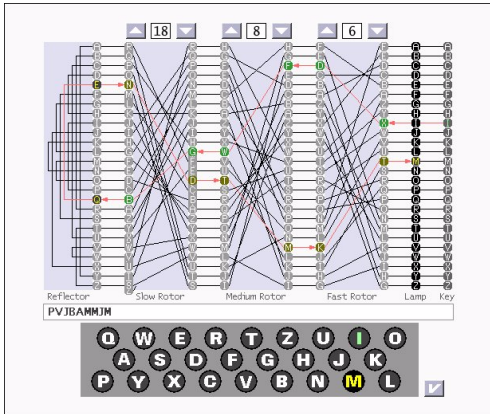
ロータは全部で 5 個あり、その中から定められた 3 個*26 を定められた順序でエニグマにセットし、定められたロータの設定位置から暗号化するほか、結線の一部を振り替えるプラグボードという機構もあって、全体として文字の置換の組み合わせは 10^{20} 通りもあります。考えられる組み合わせを風潰しにチェックする総当たり式では、これを解読するには現在のコンピュータでもかなりの時間がかかります。

ドイツ軍は「エニグマは絶対に解読されない」という自信を持っていましたが、エニグマも結局はチューリング*27 らによって解読され、戦局は連合軍に有利に展開するようになりました。1944 年のノルマンディ上陸作戦の成功や、戦争の早期終結にも、エニグマの解読が果たした役割は大きかったといわれています。

*26 ローターを 4 個セットできるタイプのももあります。

*27 「チューリングマシン」(111 ページ) 参照。

Web で学ぶ 【エニグマ】



Web ページ「エニグマ」の Java アプレットは図 8.18 と同じデザインで、キーをクリックすると暗号化された文字のキーが黄色で表示されるようになっています。電流の流れる経路が表示され、ローターも一文字暗号化するたびに回転します。エニグマのしくみが容易に理解できます。

4.3.4. DES

戦後、コンピュータが現れると暗号もコンピュータ化されてますます複雑になり、コンピュータネットワークが普及してくると、ネットワーク上の情報を暗号で保護しようという気運が高まりました。

DES (data encryption standard) は 1977 年に米国商務省が連邦情報処理標準 (FIPS: federal information processing standard) として採用した暗号化アルゴリズムです。

これまでの暗号は文字単位で置換していたのに対して、DES は図 8.19 のように、まず 8 文字ごとにビット単位でデータを変換します。“I LOVE YOU” は ASCII*28 で表すと「49 4C 4F 56 45 59 5F 55」ですが、DES の初期転置でシャッフルされると「EE 0D 9F 7C 2B 00 FF 00」というデータに変わります。そのあと暗号鍵データとの EXOR 演算など、複雑な処理を 16 回繰り返して、最後に再びビット置換を行ないます。

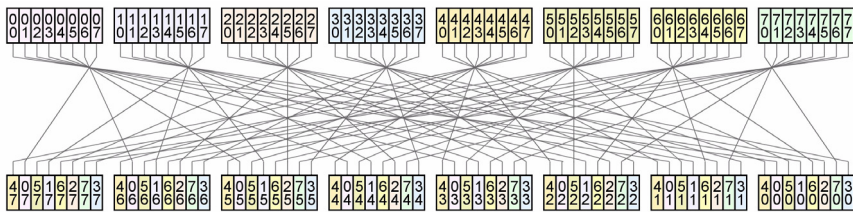


図 8.19 DES の初期転置

DES の暗号化アルゴリズムは公開されています。アルゴリズムを公開すると解読される可能性が高くなりますが、ネットワークを通じて不特定多数の人々の間で暗号化通信をする場合は、標準化されたアルゴリズムを使う方が暗号化のコストが下がります。暗号の安全性は鍵で

*28 「ASCII」(22 ページ) 参照。

保とう、というわけです。

しかし DES の鍵は 56 bit なので、総当たり式でも $2^{56} = 7.2 \times 10^{16} = 7.2$ 京^{けい} 回試せば解読することができます。DES が作られた頃 (1977 年) はこれでも安全と考えられたのかもしれませんが*29、コンピュータの性能が良くなった現在では、これでは不十分です。

三菱電機の松井充^{みつる}は、1993 年に線形解読法を開発して DES を解読しました。また、1997 年から 1999 年にかけて、RSA 社によって何度か DES の解読コンテストが行われましたが、1999 年にはインターネットを通じて約 10 万台のコンピュータを動員したグループが、22 時間あまりで解読に成功しています*30。

4.3.5. RSA

これまで紹介した暗号はすべて秘密鍵暗号方式でしたが、RSA は公開鍵暗号方式です。

公開鍵暗号方式は 1976 年にディフィーとヘルマンによって考案されましたが、彼らの論文には公開鍵と秘密鍵、二つの鍵を使うという概念のみが書かれていて、具体的にそれをどう実現するかについては触れられていませんでした。

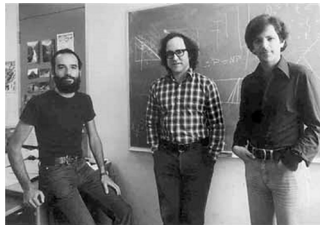


図 8.20 左から、リヴェスト、シャミル、エイドルマン

そこでロナルド・リヴェスト (Ronald Rivest)、アディ・シャミル (Adi Shamir)、レナード・エイドルマン (Leonard Adleman) の三人は、1977 年にこれを実用化し、それぞれの頭文字をとって RSA と命名しました。

RSA は「大きい数の素因数分解は困難である」という性質を利用しています。たとえば 65 という数字は 5 と 13 という素数*31 同士を掛け合わせたものです。この例の場合 65 は 5×13 だということは誰にもすぐに分かりますが、それは 65 が十分小さい数だからで、4,294,967,297 なら、何と何を掛け合わせたものか、すぐに分かる人はまずいないでしょう*32。

4,294,967,297 は 6,700,417 と 641 とを掛けたものですが、RSA は、この例でいえば 6,700,417 を公開鍵、4,294,967,297 と 641 を秘密鍵としています。公開鍵から秘密鍵を推測することはできません。もちろんコンピュータを使えば、即座に $4,294,967,297 = 6700417 \times 641$ と答えが出ますが、それはこれらの数がコンピュータにとってはまだ「小さい数」だからです。もっ

*29 当初は 128 ビットの鍵が検討されていましたが、NSA (国家安全保障局 : national security agency) の要請によって 56 ビットに短縮されたといわれています。

*30 distributed.net: Project DES (<http://www.distributed.net/des/>)

*31 1 と自分自身以外に約数をもたない正の整数。

*32 レオンハルト・オイラー (Leonhard Euler) (1707~1783) には、これができたそうですが…。

ともっと大きな数、200桁、300桁という途方もなく大きい数の素因数分解となると、どんなコンピュータでもこれを解くことは事実上不可能です*33。

4.3.6. PGP

PGP (pretty good privacy) は1991年、フィリップ・ジーマーマン (Philip R. Zimmermann) によって開発された暗号化ソフトウェアです。彼は反核運動の活動家でしたが、プライバシーを守るために暗号技術の必要性を感じて PGP を作り、インターネットで公開しました。



図 8.21 ジーマーマン

PGP は本文の暗号化には秘密鍵暗号方式を使い、その暗号鍵を公開鍵暗号方式で暗号化しています。秘密鍵暗号方式は処理は速いが鍵の管理に問題があり、一方公開鍵暗号方式は鍵は安全だが処理に時間がかかるという、両方式の特徴が巧みに組み合わされています。

当時米国の司法当局は、犯罪者やテロリストに高度な暗号を使われることを防ぐために、暗号技術の輸出を規制していました。しかしジーマーマンは、市民権とプライバシーを守るには暗号が必要で、暗号の規制はテロリストの脅迫に屈することだと考えていました。PGP の公開後、ジーマーマンは暗号輸出規制違反の疑いで召還されましたが、司法省は後にこれを撤回し、結果的に、米国の暗号輸出規制は次第に緩和されていきました*34。

PGP はインターネットでダウンロードでき、個人が非営利目的で使用する場合は無償で使うことができます*35。

4.3.7. SSL

インターネットでショッピングをしたり、チケットの予約をしたりする機会が増えてきましたが、クレジットカード番号などの個人情報の通信は、安全でなくてなりません。

SSL (secure socket layer) は、1994年に Netscape Communications 社が開発した、インターネット上でデータを暗号化して送受信するためのプロトコルです。公開鍵暗号や秘密鍵暗号、認証などのセキュリティ技術を組み合わせたもので、盗聴や改竄、なりすましを防ぐことができます。

*33 解読には宇宙の年齢 (約 137 億年) の数億倍の時間がかかると見積もられています。ただし量子コンピュータが実用化されれば、素因数分解の問題は簡単に解けるかもしれません。

*34 日本や EU、オーストラリアなど向けの暗号輸出規制が事実上撤廃されたのは、2000年7月のことです。

*35 The International PGP Home Page (<http://www.pgpi.org/>)



図 8.22 ブラウザの鍵アイコン

オンラインショッピングなどで個人情報を入力する画面では、図 8.22 のように、ブラウザの鍵アイコンが変化することがあります。これは、データは SSL で暗号化して通信されることを表しています。

 付 録

1. 数の接頭語と単位

1.1. 数の接頭語

コンピュータの処理速度はあまりにも速く、処理時間は極めて短いので、扱う数字は日常生活で用いているものと比べて極端に大きい、あるいは小さいものになります。また記憶装置が大量のデータを記憶するようになると、その記憶容量の値なども膨大になります。そのような大きい数、あるいは小さい数を表すために、次のような接頭語が使われます。

接 頭 語	意 味	記憶容量を表す場合	例
^{キロ} k (kilo)	$10^3 = 1,000$	$2^{10} = 1,024$	km, kHz
^{メガ} M (mega)	$10^6 = 1,000,000$	$2^{20} = 1,048,576$	MB, Mbps
^{ギガ} G (giga)	$10^9 = 1,000,000,000$	$2^{30} = 1,073,741,824$	GB, GHz
^{テラ} T (tera)	$10^{12} = 1,000,000,000,000$	$2^{40} = 1,099,511,627,776$	TB

表.1 大きい数の接頭語

メモリのアドレスは2進数で指定されるため、記憶容量も2進数が基準になります。たとえば1 ^{キロバイト}kB は1,000 バイトではなく、その近似値1,024 (2^{10}) バイトです。10進数の1,000は2進数では1111101000 となって半端な数字ですが、1,024なら2進数では10000000000 なので、区切りがいいのです。

T より大きい接頭語に ^{ペタ}P (peta : 10^{15}) *1 があります。1,000 兆という途方もない大きさですが、日本の最新スーパーコンピュータ「京」は10 ^{けい}ペタフロップス余の演算性能があることから「京」と命名されています。更に大きい接頭語には ^{エクサ}E (exa : 10^{18})、^{ゼタ}Z (zetta : 10^{21}) などがあります。

接 頭 語	意 味	例
^{ミリ} m (milli)	$10^{-3} = 1/1,000$	mm, mg, msec
^{マイクロ} μ (micro)	$10^{-6} = 1/1,000,000$	μ m, μ sec
^{ナノ} n (nano)	$10^{-9} = 1/1,000,000,000$	nm, nsec
^{ピコ} p (pico)	$10^{-12} = 1/1,000,000,000,000$	pℓ
^{フェムト} f (femto)	$10^{-15} = 1/1,000,000,000,000,000$	

表.2 小さい数の接頭語

*1 $10^{15} = 1,000,000,000,000,000 \div (2^{50} = 1,125,899,906,842,624)$

1.2. 主な単位

^{ビービーエス}**bps** (bits per second) : bps は、1 秒間に何ビットのデータを送受信できるかを表す通信速度の単位です。ADSL の通信速度は 24Mbps、などという場合に使われます。

^{ティビーアイ}**dpi** (dots per inch) : dpi は 1 インチ (25.4mm) 当たりのドット数の意味で、プリンタやイメージスキャナの精細度などをあらわすのに使われます。ここでの「ドット」は文字や画像を小さい点の集合で表現するためのもので、1 インチ当たりのドット数が多ければ多いほど鮮明に印刷できます。

^{フロップス}**FLOPS** (floating point operations per second) : FLOPS は 1 秒間に浮動小数点演算命令が実行できる回数を表しています。浮動小数点演算とは小数点がある実数の計算のことで、コンピュータは整数の演算は得意ですが、実数は苦手です。そこでコンピュータの演算性能を表すのに、1 秒間に実数の演算が何回できるかを指標として使います。高性能のパーソナルコンピュータは ^{ギガ フロップス}5 G FLOPS 程度、海洋研究開発機構のスーパーコンピュータ、地球シミュレータは ^{テラ フロップス}40 T FLOPS です。それぞれ 1 秒間に 50 億回、40 兆回の浮動小数点演算ができる、という意味です。

^{ヘルツ}**Hz** (hertz) : Hz は周波数の単位です。たとえばピアノの弦が 1 秒間に 440 回振動すると 440Hz の音になります。ラジオの電波は ^{キロヘルツ}666 kHz、FM 放送なら ^{メガヘルツ}88.1 MHz、コンピュータでは CPU のクロック信号*2 の周波数が ^{ギガヘルツ}3 GHz、といった風に使われます。

^{rpm}**rpm** (revolution per minute) : rpm は 1 分当たりの回転数を表す単位です。昔の LP レコードの回転数は $33\frac{1}{3}$ rpm、ハードディスクのプラッタ*3 の回転数は 7,200 rpm、などといっています。

バイト (byte) : 情報量の単位。通常 8 ビットを 1 バイトとし*4、記憶容量を表す場合に「ハードディスクの記憶容量は ^{ギガバイト}240 GB」といった使い方をします。

ピクセル (pixel) : picture (画像) の複数形 pix と element (要素) からの造語。ディスプレイで文字や画像を表示する最小の点、画素。68 ページ参照。

ビット (bit) : 情報の最小単位。binary digit (2 進数の 1 桁) の略。等確率の 2 つの選択肢から 1 つを定めるのに必要な情報量を 1 ビットといっています。2 進数 1 桁の情報量も 1 ビットで

*2 「クロック」(50 ページ) 参照。

*3 ハードディスクの磁気ディスク円盤。「ハードディスク」(54 ページ) 参照。

*4 7 ビットを 1 バイトとするシステムもあります。

付 録

す。

秒 (second) : 言うまでもなく時間の単位です。人間にとって 1 秒はほんの一瞬ですが、コンピュータにとっては十分に長い時間です。1GHz のクロック信号で動作する CPU は何らかの仕事を 1 秒間に 10 億回もこなし、10Mbps の通信回線は 1 秒間に 1,000 万ビットのデータを伝送しています。

2. エクセルのマクロと VBA

2.1. マクロとは

コンピュータを理解するにはプログラミングを体験するのが近道ですが、通常パーソナルコンピュータにはプログラミング言語などの開発環境はないので、新たにインストールしなくてはなりません。無料のソフトも多いのですが、それでも、誰でも今すぐ、というわけにはいきません。

しかし、エクセルにはマクロ (macro) というしくみがあります。

エクセルで集計作業などをするとき、いつも決まった手順を繰り返すことがあります。たとえば「毎月小遣い帳で費目別の小計を計算してグラフを作る」などですが、こんなとき、この手順を記録できて、毎度毎度同じ作業を繰り返さなくてもすめば便利です。マクロとは複数の操作を記録しておき、複雑な操作でもマクロを実行するだけで済むようにしたものです。

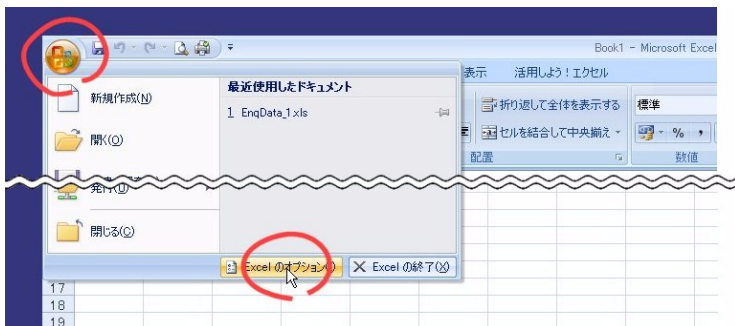
マクロはエクセルの操作を記録すれば作れますが、出来上がったマクロ自体は VBA (Visual Basic for Applications) というプログラミング言語で書かれたプログラムです。したがって、逆に VBA でプログラムを書いてエクセルのマクロを作ることもできます*5。

VBA はアプリケーションに対して指示を与えるプログラム (マクロ) を、Visual Basic という言語で記述するツールです。VBA はエクセルに付属していて、エクセルはたいていのコンピュータにインストールされていますから、VBA なら誰でも、すぐにプログラムを作ってみることができます。

ここでは VBA でプログラム作る方法について、簡単に説明します。

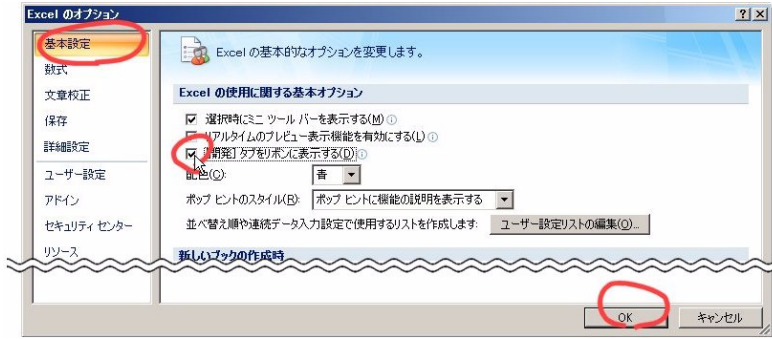
2.2. VBE

VBA でマクロを作るには、まず VBE (Visual Basic Editor) を起動します。Editor (エディタ) は文字データを入力・編集するためのソフトウェアで、VBE は特に、Visual Basic のプログラムを書くのに便利なように工夫されています。

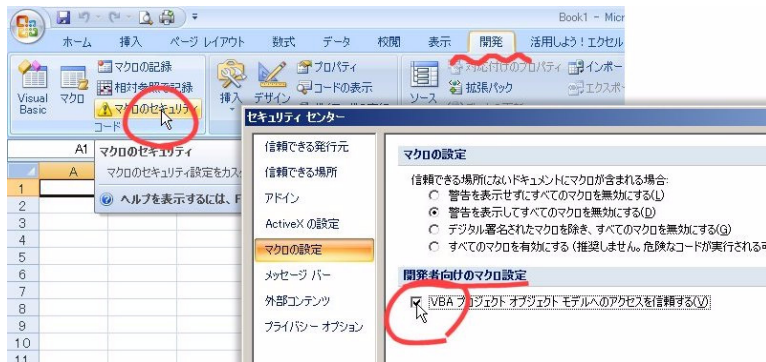


まずエクセルの Office ボタンをクリックし、開いたメニューから「Excel のオプション」をクリックします。

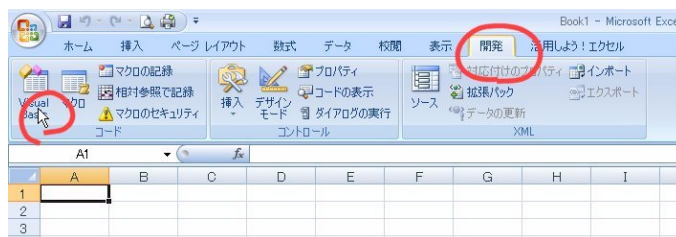
*5「プログラム」(83 ページ) 参照。



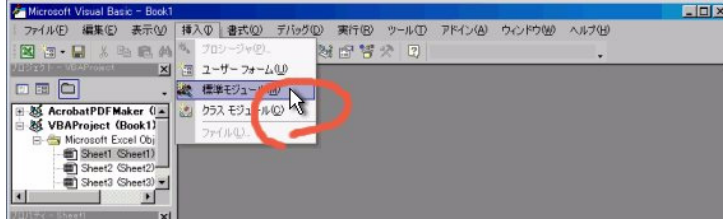
「基本設定」カテゴリーの「[開発] タブをリボンに表示する」チェックボックスを ON にして、[OK] をクリックします。



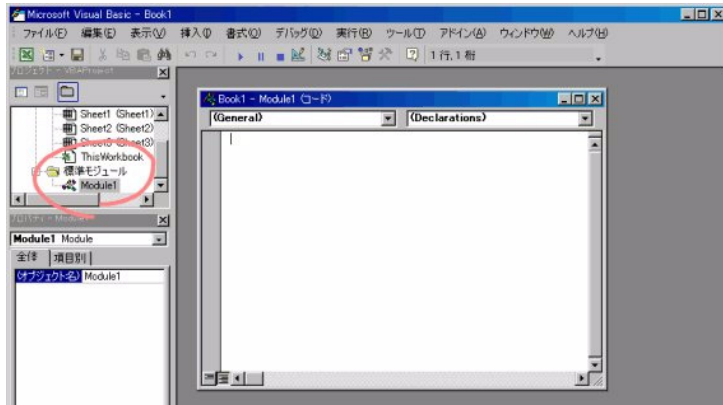
[開発] タブの中の「マクロのセキュリティ」をクリックして開いた「セキュリティセンター」の「開発者向けのマクロ設定」で [VBA プロジェクト オブジェクト モデル へのアクセスを信頼する] のチェックボックスを ON にします。



[開発] タブの「Visual Basic」をクリックします。
これで VBE が起動されます。



VBE が起動されたら、標準モジュールを追加します。VBE の「挿入」メニューの「標準モジュール」をクリックします。

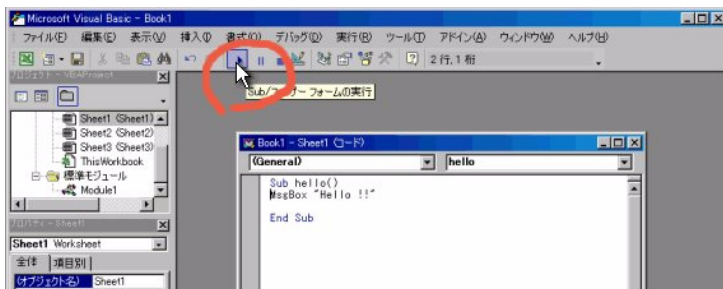


これで準備完了です。ここに開いたウィンドウに、プログラムを書いていきます。

2.3. マクロの実行

マクロのプログラム作成については第 4 章 ソフトウェアの第 2.1 節、VBA (83 ページ) を参考にしてください。

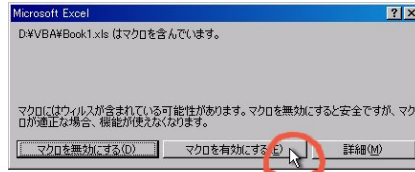
出来上がったマクロを実行するには、「Sub/ユーザーフォームの実行」ボタンをクリックします。



2.4. マクロの保存と実行

マクロは、普通に「名前を付けて保存」すれば、ワークシートのデータと一緒にエクセルのブックに保存されます。

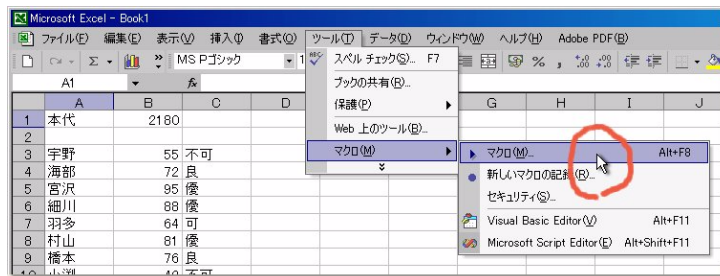
次にエクセルを起動してファイルを開くと、下図のようなダイアログが表示されます。



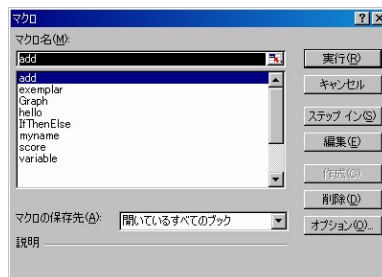
VBA には高い機能があり、ファイルも操作できます。もしマクロで重要なファイルを削除したり変更するようなプログラムを書けば、コンピュータを起動できないようにすることも不可能ではありません。現在はやや下火になりましたが、かつてはこのようにして作られたマクロウイルスが流行し、大きな被害が出ました。

この警告ダイアログボックスは、不用意にマクロを実行してウイルスに感染するのを防ぐためのものです。素性の分からないマクロは危険ですが、自分で作ったものなら安心ですから、ここでは「マクロを有効にする」をクリックします。

マクロを実行するには、「ツール」メニューから、「マクロ」 → 「マクロ」をクリックします。



保存されているマクロのリストが表示されます。実行するマクロを選んで「実行」をクリックします。



図版出典

本書の図版や写真は、次の方々からご提供や撮影許可をいただき、また書籍や Web ページから引用・転載させていただきました。ご協力下さった方々に厚く御礼申し上げます。

- 図 1.1, 5.20 *Computer Laboratory, University of Cambridge.*
<http://www.cl.cam.ac.uk/>
- 図 1.1, 3.2, 5.32, 表 5.4, 5.5 電卓博物館 <http://www.dentaku-museum.com/>
- 図 1.6, 4.15 (社) 情報処理学会 Web サイト「コンピュータ博物館」
<http://museum.ipsj.or.jp/>
- 図 1.7 独立行政法人海洋研究開発機構 地球シミュレータセンター
<http://www.jamstec.go.jp/esc/index.html>
- 図 1.8 *The origin of Spacewar*
<http://www.wheels.org/spacewar/creative/SpacewarOrigin.html>
- 図 3.11 情報機器と情報社会のしくみ素材集
<http://www.sugilab.net/jk/joho-kiki/index.html>
- 図 5.1 内山 昭：計算機械の歴史，スペクトラム Vol. 1 No.1-6, 1988
- 図 5.2, 5.6 *Information Technology History - Outline*
<http://www.tcf.ua.edu/AZ/ITHistoryOutline.htm>
- 図 5.3 *Four Function Mechanical Calculators*
<http://www.hpmuseum.org/ffhand.htm>
- 図 5.4, 5.5, 5.8 *THE HISTORY OF COMPUTING* <http://ei.cs.vt.edu/~history/>
- 図 5.7, 5.12 *The Life and Work of Konrad Zuse*
<http://www.epemag.com/zuse/default.htm>
- 図 5.9 *xTuringMachine Lab*
<http://math.hws.edu/TMCM/java/labs/xTuringMachineLab.html>
- 図 5.10 *Computer History Museum* <http://www.computerhistory.org/>
- 図 5.13, 5.27 *IBM Archives: Exhibits* <http://www-03.ibm.com/ibm/history/exhibits/>
- 図 5.14 *ISU, Department of Computer Science*
<http://www.cs.iastate.edu/homepage.html>
- 図 5.16 *Colossus Rebuild* <http://www.cranfield.ac.uk/ccc/bpark/colossus.htm>
- 図 5.17 *John W. Mauchly and the Development of the ENIAC Computer*
<http://www.library.upenn.edu/exhibits/rbm/mauchly/jwminintro.html>
- 図 5.18, 5.19, 5.23 *50th Anniversary of the Manchester Baby computer*
<http://www.computer50.org/>
- 図 5.21 *Magnetic Core Memory from a Litton navigation system*
<http://www.walenz.org/bri/museum/litton/index.html>
- 図 1.2 *RIP Claude Shannon* <http://plus.maths.org/issue15/features/shannon/>
- 図 5.24 *Flight tests of the V-2 rocket*
<http://www.jirzy.webzdarma.cz/indexe.html>

- 図 5.25 広島平和記念資料館 <http://www.pcf.city.hiroshima.jp/>
- 図 5.26, 5.41 *Lexikon's History of Computing*
<http://www.computermuseum.li/Testpage/02HISTORYCD-Index.htm>
- 図 5.28 *APOD: 2007 October 4 -50th Anniversary of Sputnik: Traveling Companion*
<http://apod.nasa.gov/apod/ap071004.html>
- 図 5.29 *Semi-Automatic Ground Environment (SAGE)*
http://www.mitre.org/about/photo_archives/sage_photo.html
- 図 5.30 *NASA Langley Research Center's Contributions to the Apollo Program*
<http://oea.larc.nasa.gov/PAIS/Apollo.html>
- 図 5.31 *Walter Houser Brattain* <http://chem.ch.huji.ac.il/history/brattain.htm>
- 図 5.31 *DATAMATH CALCULATOR MUSEUM*
<http://www.datamath.org/Story/JackKilby.htm>
- 図 5.31 *NSF PR 03-122 - October 22, 2003*
<http://www.nsf.gov/od/lpa/news/03/pr03122.htm>
- 表 5.3 大井電気株式会社
- 表 5.3, 5.4, 5.5 シャープ株式会社 技術本部 歴史ホール/技術ホール
- 表 5.3 東京理科大学 近代科学資料館
<http://www.tus.ac.jp/info/setubi/museum.html>
- 表 5.5 カシオ計算機株式会社
- 図 5.33 *CPU-Museum* <http://www.cpu-museum.com/>
- 図 5.34 *Computer Closet Collection*
<http://www.computercloset.org/MITSAltair8800.htm>
- 図 5.35 *Computer History Museum: Get In Touch With History*
http://www.osnews.com/story/4102/Computer_History_Museum_Get_In_Touch_With_History/page2/
- 図 5.36, 5.38, 5.40 日本電気株式会社
- 図 5.37, 5.41 アップルジャパン株式会社
- 図 5.39 *The DigiBarn Computer Museum* <http://www.digibarn.com/>
- 図 5.44 *The 14th International World Wide Web Conference 2005*
<http://www2005.org/index.html>
- 図 5.46 第 17 回 CSJWWW 利用者調査結果 <http://www.csj.co.jp/www17/>
- 図 5.47, 5.48 *Hobbes' Internet Timeline*
<http://www.zakon.org/robert/internet/timeline/>
The ISC Domain Survey <http://www.isc.org/solutions/survey>
- 図 6.2 総務省 平成 18 年通信利用動向調査
http://www.johotsusintokei.soumu.go.jp/statistics/data/070525_1.pdf
- 図 6.2 インターネット白書 2005 (C) Access Media/impress, 2005
- 表 6.3, 図 6.5, 6.6, 6.8 *ICT Statistics* <http://www.itu.int/ITU-D/ict/statistics/>
- 図 6.13 総務省情報通信統計データベース
<http://www.johotsusintokei.soumu.go.jp/>

- 図 7.12 携帯電話・PHS 契約数 | 社団法人 電気通信事業者協会 (TCA)
<http://www.tca.or.jp/database/index.html>
- 図 7.17 シャープ株式会社
- 図 7.23 (財) 道路交通情報通信システムセンター <http://www.vics.or.jp/>
- 図 7.28 国土交通省道路局 | ETC 利用状況
<http://www.mlit.go.jp/road/yuryo/riyou.pdf>
- 図 7.31 千葉県警察 新交通管理システム PTPS 調査報告
http://www.police.pref.chiba.jp/safe_life/UTMS/ptps_report.php
- 図 7.32, 8.8 総務省 平成 19 年 通信利用動向調査
http://www.soumu.go.jp/s-news/2008/pdf/080418_4_bt.pdf
- 図 7.36 *In memory of Dr. Mark Weiser*
<http://www2.parc.com/csl/members/weiser/>
THE LAST WEISERAMA
<http://www-sul.stanford.edu/weiser/weiserama/default.html>
- 図 7.37 株式会社日立製作所「吹けば飛ぶよなチップが社会を変える」
http://www.hitachi.co.jp/inspire/hakken/blue/04_mu_chip.html
- 図 8.1, 8.5, 8.6, 8.7 情報処理推進機構 <http://www.ipa.go.jp/>
- 図 8.11 *neverendingbooks; lieven le bruyn's blog*
<http://www.neverendingbooks.org/index.php/tag/rationality>
- 図 8.17 *The Crypto Drop Box Master Index*
<http://www.und.nodak.edu/org/crypto/crypto/>
- 図 8.20 *Mathematics and Internet Security*
<http://www.ams.org/featurecolumn/archive/internet.html>
- 図 8.21 *Philip Zimmermann*
<http://www.philzimmermann.com/EN/background/index.html>

索引

【Symbol】

10 進数..... 16
16 進数..... 21, 166
2 進数..... 14, 159

【A】

ABC..... 115
Access..... 98
ADSL..... 156
AI..... 102
ALTAIR 8800..... 134
AND 回路..... 28
AP..... 90, 95
Apple I..... 135
Apple II..... 136
ARPA..... 128, 140
—NET..... 140
ASCC..... 114
ASCII..... 22
ATM..... 65
ATOK..... 97
AutoCAD..... 101
Autorun[†] *1..... 210

【B】

Badtrans[†]..... 209
BASIC..... 81, 95, 135
BD..... 60
Bofra[†]..... 210
bps..... 227
Brain[†]..... 208
BRICs..... 152

【C】

C..... 81
CAD..... 100
Cascade[†]..... 208
CATV..... 156
CCD..... 65
ccTLD..... 160
CD..... 57

—ROM..... 59
CERN..... 141, 165
CG..... 99
CMOS..... 41
COBOL..... 81
CODABAR..... 67
Colossus..... 116
Concept[†]..... 208
CPU..... 1, 8, 93
CRT..... 118
CS-10A..... 130
CS-31A..... 131
CSMA/CD..... 145

【D】

DBMS..... 97
DES..... 222
DNS..... 163
—サーバ..... 163
DoS 攻撃..... 210
dpi..... 65, 227
DRAM..... 52
DTP..... 101
DVD..... 60
DVORAK 配列..... 63

【E】

EAN..... 67
EDSAC..... 119
EDVAC..... 121
Edy..... 195
EL-805..... 133
ENIAC..... 51, 117
—特許..... 122
ETC..... 185, 190
EUC..... 26

【F】

FDDI..... 146
FeliCa..... 183
FEP..... 97

*1 索引中、[†]印はコンピュータウイルス、ワーム名。

FET 41
 FLOPS 227
 FM 多重放送 189
 FOMA 183
 Fortnight[†] 209
 FORTRAN 81
 FTP 167
 FTTH 157

【G】

GMR 56
 GPS 185
 gTLD 160
 GUI 92, 138

【H】

Happy99[†] 208
 HDD 54
 HTML 98, 165
 HTTP 168
 Hybris[†] 209
 Hz 227

【I】

i-4004 50, 132, 134
 IBM 114, 126
 —PC 137
 —System/360 126
 IC 42, 49, 129
 —タグ 198
 —カード乗車券 196
 ICANN 159, 162
 ICBM 127
 ICOCA 196
 ID 215
 iD 196
 IEEE 148
 IME 96
 IMP 140
 IPA 205, 210
 IPv6 159
 IP アドレス 159
 ISDN 155
 ISO 27, 149
 ISP 154
 ITS 185
 i モード 182

【J】

JAN 67
 Java 82
 JCL 92
 JIS 漢字コード 24
 JIS コード 23
 Jw_cad 101

【K】

Klez[†] 209

【L】

LAN 92, 145
 Laroux[†] 208
 L^AT_EX 101
 LC-78 133
 LCD 69
 LE-120 131
 LOVELETTER[†] 208
 LSI 49

【M】

MAC 145
 Macintosh 138
 Melissa[†] 208
 MIDI 102
 MIME 170
 MITS 社 134
 MMF 158
 MOS トランジスタ 40, 52
 MPU 8, 49, 131
 MS-DOS 137
 MS-IME 97
 MSB 18
 MSBlaster[†] 209
 Mytob[†] 210

【N】

nanaco 196
 NASA 128
 Netsky[†] 209, 211
 NOT 回路 30
 NSFNET 143
 NW-7 67
 N 型半導体 35

【O】

OCR 65
 Oracle 98

OR 回路	29
OS	53, 90
OSI	149
—基本参照モデル	149

【P】

PageMaker	101
PAN	150
PASCAL	81
PASMO	196
PC-8001	136
PC-9801	137
PDP-1	6
Perl	82
PGP	224
PHS	179
PIN	215
PiTaPa	196
pixel	65
POP	169
PTPS	185, 191
PWM	177
P 型半導体	35

【Q】

QR コード	67
QT-8D	131
QuarkXPress	101
QWERTY 配列	62

【R】

RAID	203
RFID	198
ROM	53
rpm	56, 227
RSA	223

【S】

SAGE	127
Sasser [†]	209
SMTP	169
SSEM	118
SSL	224
Suica	196
Swen [†]	209
SXVGA	68

【T】

TFT 液晶	71
The Baby	118
TK-80	135
TLD	160
TMR	56
tpi	55
TRON	197

【U】

unicode	27
UNIVAC I	126
UNIX	7
UPC	67
UPS	5, 204
URL	167
USB	150
—メモリ	61, 210
USB メモリ	54
UTMS	185, 192

【V】

V2	124
VBA	83, 229
VGA	68
ViaVoice	104
VICS	185, 187

【W】

WAN	150
Wi-Fi	148
Windows	90
Word	96
WWW	98, 141, 165
—サーバ	165

【X】

XGA	68, 71
-----	--------

【Z】

Z1	109
Z3	113

【あ】

アイコン	90, 92, 139
アインシュタイン* ²	124
アウトラインフォント	77
アウトルック・エクスプレス	99

*² 索引中、*印は人名または社名。

アキュムレータ	44
アクセス	98
アセンブリ言語	48, 80
アタナソフ*	115
アップロード	167
アドビイラストレータ	100
アドビフォトショップ	100
アドホックネットワーク	199
アナログ	12
アプリケーションプログラム	90, 95
アプレット	82
アポロ計画	128
アレン*	135
暗号	216
—化	216
—解読	116, 221
換字—	219
シーザー—	219
暗証番号	215

【い】

イーサネット	145
一太郎	96
イメージスキャナ	65
インクジェットプリンタ	73
インターネット	140, 151
—エクスプローラ	98
—の普及率	151
インタープリタ	81
インテル*	131, 137
インバータ	30, 41

【う】

ウィリアムス*	118
ウィルクス*	119
ウェブブラウザ	98
ウェブページ	98, 167
ウォズニアック*	135
宇宙開発競争	127

【え】

エイケン*	114
エイダ*	109
エイドルマン*	223
液晶	69
—ディスプレイ	69, 133
TFT—	71
エクセル	83, 97, 229

エックカート*	117, 126
エニグマ	220
演算装置	8, 32
エンジニアリング・ワークステーション	7

【お】

大型汎用コンピュータ	5
オープンハイマー*	125
オドナー*の計算機	107
オブジェクト	85
オペラ	98
オペレーティングシステム	90
おもちゃ	175
オラクル	98
音楽ソフト	102
音声合成	104
音声認識	104
オンラインソフト	95

【か】

階差機関	108
解析機関	109
回線交換方式	147, 155
顔認識	4
ガガーリン*	128
カシオミニ	133
ガスメーター	177
画素	65, 68
仮想記憶	93
価電子	35
換字暗号	219
干渉	57

【き】

キーボード	62
記憶容量	54
機械語	47, 80
基数	16
揮発性	53, 54
基本ソフト	90
基本論理回路	28
キャッシュメモリ	53
キャノーラ 130	130
キャラクタコード	22
キャリア	36
キルバーン*	118
キルビー*	129

【く】	
偶奇性検査	177
クライアントサーバシステム	147
クラッド	158
グラフィックソフト	99
繰り返し	87
クロック	50

【け】	
携帯電話	179
—のしくみ	180
—の歴史	181
桁上げ	19
桁借り	20
ケネディ*	128

【こ】	
コア	158
コアメモリ	52
公開鍵暗号方式	216
光学式マウス	64
公共車両優先システム	191
高水準言語	80
高等研究計画局	128, 140
構内情報通信網	92
国際標準化機構	149
国際標準文字コード	27
国名コード	160
—トップレベルドメイン	160
小島義雄*	131
コンデンサ	52, 116
コンパイラ	81
コンピュータ	1
—ウイルス	206
—グラフィックス	99
—セキュリティ	201
—のしくみ	43
コンペット CS-10A	130

【さ】	
サーバ	7
坂村健*	197
サファリ	98

【し】	
シーケンサソフト	102
シーケンシャルアクセスメモリ	116, 119
シーザー暗号	219

シーリングライト	173
シェアウェア	95, 202
磁気コアメモリ	120
磁気ディスク	54
システムダウン	202
自動車電話	181
自動翻訳	102
シフト JIS 漢字コード	25
ジーマーマン*	224
嶋正利*	132
シミュレーション	5, 101
ジム・クラーク*	142
指紋認証	4
シャノン*	2, 113
シャミル*	223
集積回路	42, 49, 129
主記憶装置	8, 52, 93
出力装置	8, 68
シュトラースマン*	124
シュレーディング*	124
条件分岐	86
常時接続	156
情報処理推進機構	210
情報量	10
照明器具	173
ショックレー*	39, 129
ジョブス*	135
ジョブ制御言語	92
ジョルダーホン	181
シリコン	35
シリンダ	55
真空管	115, 120
シングルタスク	93
人工衛星	127, 185
人工知能	102
真性半導体	35
真理値表	28

【す】	
水銀遅延線メモリ	119
垂直磁気記録方式	56
スイッチング作用	39
スーパーコンピュータ	5, 227
スキーム	167
スタイビッツ*	114
スタンドアロン	92, 145
スプートニク	127, 140
スプレッドシート	97

スワップ	94
【せ】	
制御装置	8
正孔	36
生体認証	216
整流作用	37
セキュリティホール	207, 213
セクタ	55
接合	37
—型トランジスタ	38, 129
セルラー方式	180
全地球測位システム	185
【そ】	
走査	65
増幅作用	39
ソート	97
属性型ドメイン	161
ソフトウェア	79
【た】	
帯域幅	155
ダイオード	35, 37
タイガー手廻り計算機	107
第2次世界大戦	124
ダイヤルアップ接続	155
大陸間弾道ミサイル	127
ダウンサイジング	5
ダウンロード	95
タグ	165
多重分岐	87
タッチパネル	65
端末	5
【ち】	
地域型ドメイン	161
遅延線メモリ	119
地球シミュレータ	6, 227
蓄積記憶管	118
中央処理装置	1, 8, 49, 93
チューリング*	111
—マシン	111
超音波式車両感知器	188
【つ】	
ツイストペアケーブル	146
通信規約	149

ツーゼ*	109, 113
【て】	
抵抗率	35
デジタル	12
低水準言語	80
ディスプレイ	68
ディフィー*	223
低レベルフォーマット	57
データベース	97
デスクトップパブリッシング	101
デファクトスタンダード	63
電界効果型トランジスタ	41
電気式計算機	113
電気炊飯器	172
電子	35
—計算機	1, 115
—商取引	193
—署名	217
—認証	218
—マネー	195
—メール	169, 210
—メールソフト	99
点接触型トランジスタ	129
電卓	130
—戦争	130
電波ビーコン	189
電話回線	155
【と】	
統合デジタル通信網	155
道路交通情報通信システム	187
トークンパッシング	146
トークンリング	146
ドットインパクトプリンタ	75
ドットフォント	77
トップレベルドメイン	160
国名コード—	160
分野別—	160
トナー	73
ドメイン	159, 167
属性型—	161
第2レベル—	161
第3レベル—	162
第4レベル—	162
地域型—	161
トップレベル—	160
—ネームシステム (DNS)	163

汎用 jp—	162
—名登録機関	162
トラック	55
トランジスタ	35, 38
トロイの木馬	207
ドローソフト	99

【な】

ナビゲーションシステム	185
名前解決	163
波の干渉	57

【に】

日本語入力ソフト	96
ニューマン*	116
入力装置	8, 62

【ね】

ネームサーバ	163
ネットスケープ・ナビゲータ	98
ネットスケープ・メッセンジャー	99
ネットワーク	92, 145

【の】

ノイズ*	129, 131
ノイマン*	103, 121
—型コンピュータ	1, 121

【は】

バーコードリーダー	66
パーソナルコンピュータ	7, 134
バーディーン*	129
ハードディスク	54
バーナーズ・リー*	141, 165
ハーバード Mark-I	114
ハーン*	124
配向膜	71
バイト	11, 227
ハイパーテキスト	141, 165
ハイルマイヤー*	69
パケット	140, 147
—通信	140
パスカル*の計算機	106
パスワード	215
パソコン	7
パターン認識	3
波長	58
バックアップ	203, 214

パッケージソフト	95, 202
ハブ	146
バベジ*	108
パラメトロン	130
バラン*	140
パリティビット	177
パルス幅変調	177
半加算器	32
半導体	35
—素子	37
ハンドオーバー	181

【ひ】

ピアツーピア	147
ビーコン	
—電波—	189
—光—	188
光ディスク	54
光ファイバ	157
ピクセル	68, 227
ビジュアル	97
ビジコン*	131
非接触 IC カード	183, 195
ビット	2, 10, 227
ピット	58
秘密鍵暗号方式	216
秒	228
表計算ソフト	97
ビル・ゲーツ*	81, 135

【ふ】

ファイアウォール	205
ファイバ	157
—光—	157
—マルチモード—	158
ファジー理論	102
フォーマット	57
—物理—	57
—論理—	57
フォトレタッチソフト	100
フォント	76
不揮発性	53, 54
符号付き 2 進数	18
符号ビット	18
不正アクセス	205
物理フォーマット	57
浮動小数点演算	227
ブラウザ戦争	143

ブラウン*	124
フラッシュメモリ	61
ブラッタ	54
ブラッテン*	129
フリーウェア	95, 202
フリップ・フロップ	33
プリンタ	72
ブルーレイ・ディスク	54, 60
プレーナ型トランジスタ	39
ブロードバンド	155, 158
プログラミング言語	79, 135
プログラム	47, 83
—内蔵方式	118, 122
フロッピーディスク	54
プロトコル	149
プロバイダ	154
プロパティ	85
分野別トップレベルドメイン	160
【へ】	
米国航空宇宙局	128
米ソ冷戦	127
ペイントソフト	99
ページプリンタ	72
ベリ-	115
ヘルツ	227
ヘルマン*	223
偏光板	70
変数	84
【ほ】	
ポインティングデバイス	63
補助記憶装置	8, 54
ポット	210
翻訳ソフト	102
【ま】	
マーク・アンドリーセン*	142
マーク・ワイザー*	197
マージ	97
マイクロコンピュータ	7, 62, 171, 172
マイクロソフト*	135, 137
マイクロプロセッサ	131, 134
マイコンメーター	177
マウス	63
マクロ	83, 229
—ウイルス	207
松井 充*	223

マルチタスク	93
マルチモードファイバ	158
マンハッタン計画	121, 125

【み】

ミニコンピュータ	6
ミラーリング	203

【む】

ムーアの法則	50
無線 LAN	147
無停電電源装置	5, 204

【め】

メインフレーム	5
メソッド	85
メディアアクセス制御	145
メモリ	44, 46, 93
メモリカード	54, 61
メモリセル	52
面記録密度	56

【も】

モックリー*	117, 126
モザイク	98, 142
モジラ	98
モデム	155

【ゆ】

ユードラ	99
ユビキタス・コンピューティング	197

【ら】

ライニツァー*	69
ライブニッツ*の計算機	107
ランダムアクセスメモリ	119
ランド	58
ランド*	140

【り】

リクエスト*	223
リフレッシュ	53, 118
リレー	32
—式計算機	113
リンク	142, 165

【る】

ルーズベルト*	125
ルータ	164
ルートサーバ	163

【れ】			
レーザ	58, 66	論理フォーマット	57
—プリンタ	72	論理和回路	29
レジスタ	44	【わ】	
レジストリ	162	ワークステーション	7
レミントン・ランド*	126	エンジニアリング—	7
【ろ】		ワード	96
ログイン	215	ワープロソフト	96
論理回路	28, 113	ワーム	207
論理積回路	28	ワクチンソフト	213
論理否定回路	30	ワンセグ	183