

村上ゼミシリーズ IV

# デジタルに親しむ

デジタル計算、デジタル回路、インターネット  
DX、AI まですべてわかる

村上 雅人  
小林 信雄

飛翔舎

## はじめに

いまやデジタル時代 (digital age) と言われています。普段、多くのひとが使っている携帯電話 (mobile phone) は、まさに高度なデジタル機器であり、手のひらに載るコンピュータ (palm-sized computer) と言ってもよいくらいです。さらにデジタル機器はインターネットにもつながり、情報が瞬時に世界を駆け巡ります。

しかし、デジタル技術 (digital technology) や情報通信技術 (ICT: information and communication technology) の進展が急速であったため、利用者の多くは、その原理はさておき、ブラックボックス (black box) として便利な機能を利用しているというのが現状です。そもそも、多くのひとが利用しているパソコン (PC: personal computer) がブラックボックスです。

さらに、自分にはデジタル技術の利用は無理と最初からあきらめているひとも居ます。これを、デジタルデバイド (digital divide) と呼んでいます。つまり、IT 技術を利用できるひとと、そうでないひとに格差が生じているのです。これは不幸なことです。なぜなら、デジタルの恩恵を受けるべきは、地方の過疎地に住むひとや高齢者だからです。よって、政府はデジタル難民が、安心して手軽にデジタル技術を利用できる体制を整備する必要があります。台湾などが、先進事例の参考になるでしょ

う。

一方、デジタルを活用するひとたちは、ブラックボックス化した機器を利用するだけでなく、ある程度、その基礎や原理を理解しておくことも重要です。それが、豊かなデジタル社会を実現する源泉となるからです。一部の専門家が開発を担うという体制では、真のデジタル社会は実現できません。

本書では、**2進法** (binary number system) というデジタルの基本から、機械であるコンピュータが、どのように**四則演算** (four arithmetic operations) に利用されるかまでを振り返ります。そのうえで、**ハード** (hardware)、**ソフト** (software) の両面から基礎技術を概観し、**インターネット** (the Internet) の進展や、**デジタル・トランسفォーメーション** (DX: digital transformation) とは何か、また、いま大きな注目を集めている**人工知能** (AI: artificial intelligence) の意味と意義を、村上ゼミの指導教員とゼミ生が議論を交わしながら、学んでいきます。

みなさんも、ゼミ生と一緒に、デジタル技術に親しんでみてはいかがでしょうか。

2024年 夏  
村上雅人 小林信雄

## もくじ

はじめに . . . . . 3

第1章 デジタルの基本 . . . . . 9

- 1.1. アナログとデジタル 9
- 1.2. 2進数 12
  - 1.2.1. 12進数 15
  - 1.2.2. 16進数 16
  - 1.2.3. 2進数の小数 20
  - 1.2.4. 小数の足し算 24
  - 1.2.5. 2進数の掛け算 26
  - 1.2.6. 2進数の引き算 27
  - 1.2.7. 2進数の割り算 28
- 1.3. ビットとバイト 30
- 1.4. 紙リールへの記録 32
- 1.5. プログラミング言語 36
  - 1.5.1. 機械語 36
  - 1.5.2. プログラミング言語 36
  - 1.5.3. BASIC言語 39
- 1.6. 容量 42
- 1.7. デジタルの世界 44

第2章 ハードウェア . . . . . 47

- 2.1. 半導体とドーピング 50
- 2.2. ダイオード 53
- 2.3. 正孔の移動 55
- 2.4. トランジスタ 58
- 2.5. ON/OFFを制御するトランジスタ 66
- 2.6. 電界効果トランジスタ 67
- 2.7. メモリ 71
  - 2.7.1. 磁気記録 73
  - 2.7.2. フロッピーディスク 77

2. 7. 3. 光磁気ディスク	79
2. 7. 4. 光ディスク	82
2. 7. 5. ハードディスク	86
2. 7. 6. フラッシュメモリ	89
<b>第3章 論理回路</b>	<b>94</b>
3. 1. AND回路	95
3. 2. OR回路	97
3. 3. NOT回路	99
3. 4. ダイオードによるAND回路	102
3. 5. ダイオードによるOR回路	105
3. 6. トランジスタを使ったNOT回路	108
3. 7. 論理回路による計算	109
3. 7. 1. 半加算器	111
3. 7. 2. NAND回路	113
3. 7. 3. 全加算器	115
3. 8. コンピュータによる四則演算	121
<b>第4章 インターネット</b>	<b>127</b>
4. 1. 電波	132
4. 2. アンテナと送受信	140
4. 2. 1. 送信	141
4. 2. 2. 受信用アンテナ	143
4. 2. 3. ラジオ放送	145
4. 2. 4. AM放送の原理	146
4. 2. 5. 検波	147
4. 2. 6. ゲルマニウムラジオ	148
4. 2. 7. FM放送	151
4. 3. 携帯電話の通信	152
4. 4. Wi-Fi ワイファイ	156
4. 4. 1. 有線 LAN	156
4. 4. 2. 無線 LAN	159
4. 5. 有線ケーブル	162

## もくじ

- 4.5.1. 銅線ケーブル 163
- 4.5.2. ADSL回線 164
- 4.5.3. ブロードバンド 166
- 4.5.4. 同軸ケーブル 167
- 4.5.5. ケーブルテレビ 169
- 4.6. 光ファイバー 171
- 4.7. 衛星通信 176
- 4.8. GPSの原理 180
- 4.9. 海底ケーブル 183
- 4.10. インターネット 187
- 4.11. 電子メール 197
- 4.12. ソーシャル・ネットワーキング・サービス 202

## 第5章 デジタル・トランスフォーメーション ···· 208

- 5.1. エドテック 209
- 5.2. 内なるDX 214
- 5.3. 注目されるDX 222
- 5.4. デジタイゼーション 224
- 5.5. デジタライゼーション 229
- 5.6. デジタライゼーションの成功例 231
  - 5.6.1. ネットフリックス 231
  - 5.6.2. アマゾン 233
  - 5.6.3. アントレプレナーシップ 235
- 5.7. レガーシステムの刷新 237
- 5.8. 2025年の崖 242
- 5.9. コロナとDX 245
- 5.10. 大学のDX 246
- 5.11. 大規模公開オンライン講座 253
- 5.12. DXレポート2 257
- 5.13. GIGAスクール構想 258
- 5.14. セキュリティ 262
- 5.15. デジタルガバメント 268
  - 5.15.1. PCの高度化 270

5. 15. 2. サーバによる管理	272
5. 15. 3. データセンター	273
5. 16. クラウドとは	275
5. 17. クラウド導入の例	280
5. 17. 1. 電子カルテ	280
5. 17. 2. 行政サービス	282
5. 18. クラウドサービス	285
<b>第6章 人工知能—AI</b>	<b>291</b>
6. 1. シンギュラリティ	294
6. 2. 機械学習	296
6. 3. 教師あり学習	299
6. 4. 強化学習	301
6. 5. ディープ・ラーニング	304
6. 6. 教師なし学習	312
6. 7. 科学は万能か	315
6. 8. 3体問題	316
6. 9. 多体問題	318
6. 10. 人間の柔軟性	321
6. 11. 自然観察	324
6. 12. 実験	325
6. 13. AIを使いこなす	327
6. 13. 1. チャットボット	328
6. 13. 2. 研究分野への応用	330
6. 14. AIの失敗	333
<b>おわりに</b>	<b>338</b>

## 第1章 デジタルの基本

雅人 今日からのゼミでは、デジタル技術 (digital technology) を取り上げたいと思う。

結美子 いまや、「デジタル」という言葉は世の中のキーワードですね。日本政府も、2021年9月にデジタル庁を設置し、デジタル・ガバメント (digital government) を目指すと宣言しています。

しのぶ ただし、「デジタル」と言っても、範囲が広いですし、具体的になにを指すかがあいまいですよね。そもそも、いろいろな横文字が多すぎます。

### 1.1. アナログとデジタル

雅人 それでは、まず、アナログ (analogue) とデジタル (digital) の違いについて整理しておこう。ここが出発点だね。

和昌 確かにそうですね。デジタルの対極にあるのはアナログですから、これら用語の整理が重要ですね。

雅人 もともとの定義をひも解けば、アナログとは「ある量やデータが連続的に変化する物理量のことで、たとえば、電流や電圧で表現すること」となる。一方、デジタルとは、「ある量やデータの変化を有限桁の数字、たとえば、2進数として表現すること」となる。

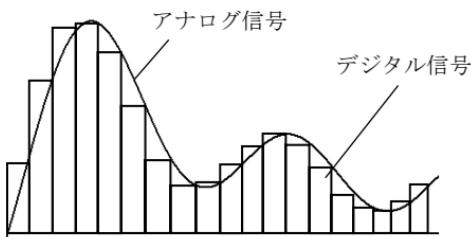


図 1-1 アナログ信号とデジタル信号

結美子 やはり、定義となるとわかりにくいけれど。

雅人 アナログの英語は“analogue”あるいは“analog”で「類似の」という意味がある。つまり、あるデータの変化を、物理量である電圧や電流の変化に「類似させて表示する」というのがもともとの意味となる。

しのぶ “digital”は“digit”的形容詞で、“digit”は「数字」や「桁」という意味でしたね。もともと“digit”には「指」という意味があり、1から10までの数を指で数えたことから、この名がついたと聞いています。

雅人 そうなんだ。“digit”は、正式には「アラビア数字」“Arabian number”的ことを指している。0, 1, 2, ..., 9までの10個の数字のことだ。アラビアという名がついているが、もともとはインド発祥の数字なんだ。インドには「0の発見」という偉大な発明もある。

和昌 確かに、デジタル表示とは数字で表示するという意味ですね。イメージとしては、アナログは連続、デジタルは飛び飛びですか。そう言えば、アナログ人間という言葉もよく聞きます。

雅人 お年寄りが自虐的に使うときの「アナログ人間」とは「デジタル機器の使用が苦手なひと」という意味になる。

和昌 それは、わかります。スマートフォンやパソコンの使い方が苦手なひとが「自分はアナログだから」とよく言いますね。

雅人 もうひとつ、物事を割り切って考えないひとを「アナログ人間」と呼ぶこともある。このときは、「あいまい」という意味合いもあるね。一般的なアナログとデジタルの違いは、時計を考えればわかりやすいかな。



信雄 なるほど。アナログ時計 (analogue clock) は、長針と短針があって、連続的に時間が変化しますね。一方、デジタル時計 (digital clock) では、時間が数字で直接表示されます。

雅人 そして、コンピュータの世界でデジタルという場合、一般には 0 と 1 の 2 個の数字を使うことを意味する。

## 1.2. 2 進数

結美子 それが 2 進数 (binary number) でしたね。これは、確かオンとオフ (ON/OFF) を数字の 1 と 0 に対応させていると聞きました。でも 1 と 0 だけで算用数字を表すには、どうしたらよかつたでしょうか。

雅人 いい機会なので、復習してみようか。まず、最初の 0 と 1 はいいね。

信雄 それは、わかります。問題は 2 ですね。

雅人 そうなんだ。2 進法 (binary number system) では、2 という数字がないから、位が 1 個増えて  $2 = 10$  となる。

しのぶ なるほど、それならば、 $3 = 11$  とすればよいのですね。

雅人 その通りだね。ただし、このままでは 3 が 11 と等しいと誤解するので、それを区別しておこう。ここでは、 $2_{(10)} = 10_{(2)}$  や  $3_{(10)} = 11_{(2)}$  という表記を使うことにする。添え字の ( ) 内の数字

は、10進法と2進法という意味だ。

和昌 とすれば、 $4_{(10)} = 100_{(2)}$ となりますね。

雅人 そうだ。表1-1に、0から5までの10進数(decimal number)と2進数の対応を示してある。

表1-1 10進数と2進数の対応

	$2^2$	$2^1$	$2^0$	換算表
0			0	$2^0 \times 0 = 0$
1			1	$2^0 \times 1 = 1$
2		1	0	$2^1 \times 1 + 2^0 \times 0 = 2$
3		1	1	$2^1 \times 1 + 2^0 \times 1 = 3$
4	1	0	0	$2^2 \times 1 + 2^1 \times 0 + 2^0 \times 0 = 4$
5	1	0	1	$2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 = 5$

結美子 この対応表はわかりやすいですね。この表から、桁数は2の累乗(power)に相当していることがわかります。たとえば、 $101_{(2)}$ は

$$101_{(2)} = 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 = 5_{(10)}$$

となります。以下同様に10進数の10, 15, 16を2進数に変換すれば表1-2のようになります。

表1-2 10進数の10, 15, 16に対応した2進数

	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
10		1	0	1	0
15		1	1	1	1
16	1	0	0	0	0

信雄 この表の対応関係はつぎのようになります。

$$1010_{(2)} = 2^3 + 2^1 = 8 + 2 = 10$$

$$1111_{(2)} = 2^3 + 2^2 + 2^1 + 2^0 = 8 + 4 + 2 + 1 = 15$$

$$10000_{(2)} = 2^4 = 16$$

和昌 2進数どうしの足し算も、2で繰り上がるを考えればよいので

$$1010 + 101 = 1111 \quad 1111 + 1 = 10000$$

$$\begin{array}{r} 1010 \\ + 101 \\ \hline 1111 \end{array} \quad \begin{array}{r} 1111 \\ + 1 \\ \hline 10000 \end{array}$$

となります。これら足し算は10進数では、それぞれ  $10 + 5 = 15$  と  $15 + 1 = 16$  に対応します。

雅人 そうだね。そして、原理的には、どんなに大きな数字でも2進法で表現できることになる。ただし、桁数はとてもなく大きくなるがね。

しのぶ ところで、いまの表は10進法(decimal number system)にも応用できますね。つまり、表1-3のような対応になります。

表1-3 10進法

	$10^3$	$10^2$	$10^1$	$10^0$	
56			5	6	$10^1 \times 5 + 10^0 \times 6$
432		4	3	2	$10^2 \times 4 + 10^1 \times 3 + 10^0 \times 2$
2050	2	0	5	0	$10^3 \times 2 + 10^2 \times 5$

## 1.2.1. 12進数

**結美子** これならば、10のところが、どんな数字に変わっても対応が可能ですね。たとえば、12進法 (duodecimal number system) を考えてみます。表1-3と同じ数字の56, 432, 2050を12進法で表記すると、56は

$$12^1 \times 4 + 12^0 \times 8$$

と計算できるので

$$56_{(10)} = 48_{(12)}$$

という対応になります。同様にして

$$432_{(10)} = 12^2 \times 3 = 300_{(12)}$$

です。ただし、2050を計算すると

$$12^3 \times 1 + 12^2 \times 2 + 12^1 \times 2 + 12^0 \times 10$$

となって、1桁目に10という数字が出てきてしまします。これは12進法だから、ひとつの桁には0から11までの数字が入りうるからですね。無理に表記すれば

$$2050_{(10)} = 122(10)_{(12)}$$

でしょうか。

表1-4 12進法による表記

	$12^3$	$12^2$	$12^1$	$12^0$	
56			4	8	$12^1 \times 4 + 12^0 \times 8 = 56$
432		3	0	0	$12^2 \times 3$
2050	1	2	2	10	$12^3 \times 1 + 12^2 \times 2 + 12^1 \times 2 + 12^0 \times 10$

**雅人** そうなんだ。この対策としては、10にA, 11にBを当てはめることもできる。すると

$$2050_{(10)} = 122A_{(12)}$$

と表記できる。つまり、12進法には、0, 1, 2, ..., 9, A, Bの12個の

数字が必要となることに対応している。

しのぶ この方式ならば、何進法でも対応できますね。そして、20進法ならば、20個の異なる数字表記が、100進法なら100個の異なる数字表記が必要となります。

雅人 それでは、12進数のBBAは、10進数のいくつになるだろう。

結美子 わたしが挑戦してみます。この数字は

$$12^2 \times B + 12^1 \times B + 12^0 \times A$$

となりますので

$$BBA_{(12)} = 144 \times 11 + 12 \times 11 + 10 = 1584 + 132 + 10 = 1726$$

と計算できます。

雅人 正解だ。原理は同様だから、何進法でも対応できるね。それでは、コンピュータでよく利用される 16進数 (hexadecimal numeral) も扱ってみよう。

### 1.2.2. 16進数

信雄 16進数も、基本的な考えは同じですよね。まず、16個の1桁数が必要になりますから、12進数と同じようにアルファベットを使うと、0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F の16個となりますね。10進数との対応は、A=10, B=11, C=12, D=13, E=14, F=15となります。

雅人 変換方法も同じだね。それでは、3AE という 16進数を 10

進数に変換してみよう。

結美子 わたしが挑戦します。A=10, E=14 ですから

$$3AE_{(16)} = 16^2 \times 3 + 16^1 \times 10 + 16^0 \times 14 = 256 \times 3 + 160 + 14 = 942_{(10)}$$

となります。

雅人 正解。では、10進数の2543を16進数に変換するのはどうだろうか。

信雄 ここは、僕が挑戦してみます。 $16^3 = 4096 > 2543 > 16^2 = 256$  ですから、3桁の数になります。ここで、まず $16^2 = 256$ で割って見ます。すると  $2543 \div 256 = 9 \dots 239$  となり、3桁目の数は9となります。つぎに、余りの239を16でわると  $239 \div 16 = 14 \dots 15$  となりますので、16進数では商がEとなり余りがFとなります。したがって、16進数は9EFとなります。

雅人 これも正解だね。つまり

$$2543 = 16^2 \times 9 + 16^1 \times E + 16^0 \times F$$

となる。

しのぶ 先生は、16進数はコンピュータでよく使われると言わっていましたが、どういうことでしょうか。

雅人 正しくは、コンピュータを扱う人間にとって便利と言い換えたほうがいいね。コンピュータが認識できるのは、あくまでも0と1の2進法の世界だ。

ただし、2進数で大きな数字を扱おうとすると、やたらと桁数

が増える。

たとえば、

$$1111_{(2)} = 15_{(10)} = F_{(16)}$$

という対応関係にある。

和昌 確かに、2進数では4桁ですね。

雅人 この程度なら、まだよいが

$$1001011111011011$$

となつたら、どうだろうか。

結美子 すぐには、ピンと来ませんね。地道に定義にしたがつて変換するしかないと思います。

$$\begin{aligned}1001011111011011_{(2)} &= 2^{15} + 2^{12} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^3 + 2^1 + 2^0 \\&= 32768 + 4096 + 1024 + 512 + 256 + 128 + 64 + 16 + 8 + 2 + 1 = 38875_{(10)}\end{aligned}$$

となります。

雅人 そうだね。このように38875という5桁の10進数が16桁の2進数になつてしまふ。コンピュータにとっては、なんでもない数だが、われわれ人間にはどうにも扱いが難しい。ここで登場するのが16進数だ。ヒントは16進数が2<sup>4</sup>進数ということだ。

和昌 そうか。4桁の2進数が1桁の16進数と対応するのですね。対応表をつくつてみると、表1-5のようになります。

雅人 ここからが、重要なところだ。先ほどの2進数を4桁ずつ区切つてみよう。すると

1001 | 0111 | 1011 | 1101

となるね。そのうえで、それぞれの4桁の2進数が、どの16進数と対応するかを見てみよう。

表1-5 1桁の16進数と2進数、10進数の対応

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

信雄  $1001 = 9, 0111 = 7, 1101 = D, 1011 = B$  となりますので、16進数では97DBとなりますね。

しのぶ これを10進数に変換すると

$$\begin{aligned}
 97DB(16) &= 16^3 \times 9 + 16^2 \times 7 + 16^1 \times 13 + 16^0 \times 11 \\
 &= 4096 \times 9 + 256 \times 7 + 16 \times 13 + 11 = 36864 + 1792 + 208 + 11 = 38875
 \end{aligned}$$

となります。確かにありますね。

雅人 このように2進数の大きな数字であっても4桁ずつに分けて16進数を当てはめれば、われわれの認識しやすい数字になる。プログラムをつくるときには、ひとが認識しやすい16進数で表現することが多い。16進数を2進数にするのも簡単だよね。

結美子 はい、表 1-5 の対応表があれば、すぐに変換可能です。ところで、いまは 4 の倍数の桁数でしたが、たとえば 6 桁の 2 進数の場合はどうなのでしょうか。たとえば、111011 の場合の変換です。

雅人 それは、頭に 0 を 2 個付ければ良いんだ。すると

$$111011 = 0011|1011$$

となって、16 進数は 3B となる。

信雄 なるほど、これなら万能ですね。

雅人 実は、16 進数は文字コードにも使われている。たとえば、ASCII<sup>1</sup>では “A” という文字に対応したコードは 41 という 16 進数となる。これは、コンピュータ上では 0100 0001 という 2 進数に対応する。41 ならば覚えやすいよね。そして、必要ならば、すぐに 2 進数に変換できる。

結美子 ただし、41 が 16 進数ということがわからなければ意味不明ですね。それにしても、面白いですね。10 進数と 2 進数以外は、あまり用途がないと思っていましたが、そうではないのですね。

### 1.2.3. 2 進数の小数

和昌 先生、ところで、ずっと疑問に思っていたのですが、2 進

---

<sup>1</sup> American Standard Code for Information Interchange の略で、情報交換用のアメリカ版標準コードとなる。いまでは、多くのコンピュータで使用されている。

法での小数 (decimal) はどのように計算すればよいのでしょうか。イメージが、なかなか湧きません。

雅人 そういう時は基本に戻って考えてみればいいんだ。まず、10進数の小数の 3.625 を考えみよう。これは

$$3.625 = 10^0 \times 3 + 10^{-1} \times 6 + 10^{-2} \times 2 + 10^{-3} \times 5$$

ということを意味している。同じ要領で 2進数も考えればよいんだ。

信雄 なるほど、そうすると 2進数の小数の場合

$$110.1011 = 2^2 \times 1 + 2^1 \times 1 + 2^{-1} \times 1 + 2^{-3} \times 1 + 2^{-4} \times 1$$

と考えればよいのですね。

雅人 その通りだ。

しのぶ 2進数の小数の桁と 10進数の対応を示すと、表 1-6 のようになります。

表 1-6 2進数の小数点の対応

$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$
2	1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125

雅人 それでは、この表を参考にしながら、10進数の小数 3.625 を 2進数の小数に変換してみよう。

結美子 わたしが挑戦します。まず、 $3.625 = 3 + 0.625$  ですが、小数点より大きな 3 は 2進法では 11(2) となります。問題は小数

点以下の 0.625 ですね。すると

$$0.625 = 0.5 + 0.125 = 2^{-1} + 2^{-3}$$

となりますから、結局  $0.625 = 0.101(2)$  と与えられます。よって

$$3.625 = 11.101(2)$$

という対応関係が得られます。

雅人 その通りだ。これで、2進法の場合の小数の扱い方も理解できたと思う。ただし、いまの場合は有限の小数で表現できたが、実は、変換がスムースにいかない場合のほうが多いんだ。

信雄 確かに、2進法の小数点以下第3位は 0.125 となり、第4位は 0.0625、第5位は 0.03125 となって、中途半端な数字が続きますね。

雅人 実は、10進法から2進法への簡単な変換方法があるんだ。例として 0.625 に適用してみよう。まず 2 倍する。すると 1.25 になるが、この小数点以上の 1 を除いた 0.25 を取り出しそれを 2 倍する。そして、同様の操作を繰り返すと

$$0.625 \times 2 = 1.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

という結果になり、小数点以下が 0 となるので、ここで計算は終わる。そのうえで、小数点より上の桁の数を上から並べると 101 となる。実は、2進数の小数がこの値になり

$$0.625 = 0.101(2)$$

という結果が得られるんだ。

しのぶ どうしてこの方法で小数が得られるのでしょうか。

雅人 では、10進数で確認してみよう。この場合は、10倍していけばよいことになる。0.625に適用すると

$$0.625 \times 10 = 6.25$$

$$0.25 \times 10 = 2.5$$

$$0.5 \times 10 = 5.0$$

となり、小数点より上の桁の数字を順に並べれば625となる。よって、小数部分は0.625ということになる。

結美子 なるほど、原理がわかりました。10進法では10倍するごとに桁が1個上がるるので、小数点よりすぐ上の桁が、小数点以下第1位の数字となりますね。同様にして、2進法では2倍するごとに桁が上がるということですね。

雅人 その通り。だから、3進数ならば3倍、12進数なら12倍していけばよいことになる。それでは、この方法を0.875に適用したらどうだろう。

和昌 僕が挑戦してみます。

$$0.875 \times 2 = 1.75$$

$$0.75 \times 2 = 1.5$$

$$0.5 \times 2 = 1.0$$

となります。最後は小数点以下が0となりますので、ここで終わりです。よって、 $0.875 = 0.111(2)$ となります。

雅人 正解。この場合も割り切れているので、ちょうどよい数

字となっているね。それでは、0.6 はどうだろうか。

しのぶ 今度は、わたしが挑戦します。

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$0.8 \times 2 = 1.6$$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$0.8 \times 2 = 1.6$$

となって、循環していきますね。これでは、切りがありません。

信雄 そうか。これは、同じ 1001 が延々と続くことを意味しているのですね。

雅人 そうなんだ。このように収束しない場合も結構ある。そして、0.6 の場合は、0.10011001... (2) のように 1001 の部分が続く循環小数となるんだ。

結美子 なるほど、10 進数の小数を 2 進数に変換するのは結構面倒なのですね。すると、どこかの桁で切って、近似するということでしょうか。

#### 1.2.4. 小数の足し算

雅人 当然そうなるね。いずれ、いま紹介した手法を使えば、1 と 0 からなる 2 進数でも、小数を含めたすべての実数を表現でき

ることになる。その代わり、コンピュータの計算では、筆算と違って端数が出ることもあるんだ。

たとえば、

$$0.1 + 0.1 + 0.1 = 0.3$$

という計算は自明だよね。ところが、0.1を2進数にすると

$$0.1_{(10)} = 0.0001100011\dots_{(2)}$$

のような循環小数となるんだ。これでは、計算できないので、

$$0.1_{(10)} \approx 0.00011_{(2)}$$

と近似してみよう。すると  $0.1 + 0.1 + 0.1$  という計算は、2進法では

$$0.00011 + 0.00011 + 0.00011$$

という計算になる。そして

$$\begin{array}{r} 0.00011 \\ + 0.00011 \\ \hline 0.00110 \end{array} \quad \begin{array}{r} 0.00110 \\ + 0.00011 \\ \hline 0.01001 \end{array}$$

から、2進法では

$$0.00011 + 0.00011 + 0.00011 = 0.01001_{(2)}$$

となるんだ。ここで、最後の2進数を10進数に変換すると

$$0.01001_{(2)} = 2^{-2} + 2^{-5} = 0.25 + 0.03125 = 0.28125_{(10)}$$

となる。つまり、コンピュータの計算では

$$0.1 + 0.1 + 0.1 = 0.28125\dots$$

という結果になる。もちろん、近似精度を上げれば正解に近づくことにはなるが、計算にやたらと時間がかかる。だから、適当なところで切り捨てる。

このため、いまのコンピュータでも、変な表示が出て戸惑うことがよくあるんだ。

しのぶ 面白いですね。人間にとっては簡単な計算がコンピュータには難しいのですね。

雅人 これは、コンピュータでは0と1しか使えないという宿命だね。ここで、2進数の四則演算についてまとめておこう。2進数では、ひとつの桁に0,1の2個の数字しかないので1+1になったときに桁が繰り上がるというのが基本になる。まず、足し算は大丈夫だね。

#### 1.2.5. 2進数の掛け算

しのぶ はい、すでに実行していますね。それでは、掛け算を考えてみます。例として  $11010 \times 101$  を考えます。これも10進法と同じように考えればよいのですね。これは、結局

$$11010 \times 101 = 11010 \times 1 + 11010 \times 100$$

という足し算になりますから

$$11010 + 1101000$$

となりますね。

ここで足し算を実行すれば

$$\begin{array}{r} 11010 \\ + 1101000 \\ \hline 10000010 \end{array}$$

と与えられます。

雅人 正解だ。ただし、掛け算については、10進法と同じように計算することもできる。具体的には

$$\begin{array}{r}
 11010 \\
 \times \quad 101 \\
 \hline
 11010 \\
 11010 \\
 \hline
 10000010
 \end{array}$$

となる。

**結美子** 順序だてて計算していけば、この掛け算も大丈夫と思  
います。

#### 1.2.6. 2進数の引き算

**雅人** それでは、つぎに引き算に挑戦しよう。11010-101に挑戦  
するが、ここで注意するのが、1の位にある0-1という計算だ。  
これは10進法と同じ考え方で、上の位から数字を借りてくること  
になる。そして10-1とできれば1と計算できる。

**信雄** なるほど。そして、ひとつ上の位も0ならば100-1とす  
ればよいのですね。すると  $100-1=11$  となります。

**結美子** それでは、11010-101に挑戦してみます。

$$\begin{array}{r}
 11010 \\
 - \quad 101 \\
 \hline
 10101
 \end{array}$$

となります。念のために、検算をしてみます。

$$\begin{array}{r}
 10101 \\
 + \quad 101 \\
 \hline
 11010
 \end{array}$$

となって、確かに答えがあっていることが確かめられます。

### 1.2.7. 2進数の割り算

**雅人** その通りだね。実は、2進数ではひとつの桁に 0, 1 しかないので計算自体はものすごく簡単になるんだ。それでは、最後は割り算に挑戦してみよう。

$$1100 \div 100$$

はどうだろう。

**和昌** 10進数に変換すれば、 $12 \div 4 = 3$  となるので、答えは 3 ですから、2進数に直せば 11 となります。ですので、 $1100 \div 100 = 11_{(2)}$  となります。

**雅人** まあ、確かに和昌君の方法もひとつのやり方だが、ここでは、2進数で直接割り算をすることを考えてみよう。ヒントになるのが 10進数の割り算だ。 $800 \div 32$  を考えてみよう。このとき、割られるほうの 800 の上の位から見ていくのだったね。まず 100 の位の 8 は 32 では割れない。そこで、桁をひとつ落として 80 とする。すると 32 で割ると 2 となる。つぎに  $32 \times 2 = 64$  を 80 から引いて 16 となる。つぎに 160 として 32 で割ると 5 と割り切れる。よって、答えは 25 となる。

**信雄** いまの計算をたてに表記すると、つぎのようになりますね。

$$\begin{array}{r}
 & & 25 \\
 & & \overline{32) 800} \\
 0 & \rightarrow & 2 \\
 & & \overline{32) 80:0} \\
 & & \quad \overline{64} \\
 & & \quad 16 \\
 & & \overline{160} \\
 & & \quad 0
 \end{array}$$

**しのぶ** この方法を2進法でも、そのまま応用すればよいのですね。

$$1100 \div 100$$

に適用してみます。

1100の上の位から見ていきますと、割る数が100ですから、上の3桁を見ます。すると110となるので、1となります。つぎに110から100を引くと、10となります。つぎの数の0を添えて100となりますので、100で割れば答えは1となり、11が答えとなります。確かに10進法と同じ方法で解が得られますね。

**結美子** いまの計算をたてに表記すれば

$$\begin{array}{r}
 11 \\
 100 \overline{) 1100} \\
 \underline{100} \\
 100 \\
 \underline{100} \\
 0
 \end{array}$$

となります。確かに、数字が1と0しかないので計算はとても楽ですね。

雅人 割り算では、割り切れない場合もあるが、10進法と同じように計算していけば問題ない。このように、2進数の計算は10進数の方法を、そのまま流用すれば簡単にできるんだ。

和昌 はい、よくわかりました。

### 1.3. ビットとバイト

信雄 ところで、コンピュータの容量でビット(bit)やバイト(byte)と聞いたことがあります、こちらも基本は2進数なのでしょうか。

雅人 そうだね。まず、“bit”的語源は“binary information digit”と言われている。これは1か0かの情報という意味だ。そして、1ビットは0か1、2ビットでは(0,0)(0,1)(1,0)(1,1)という4個の情報が使える。

和昌 3ビットでは(0,0,0)(0,0,1)(0,1,0)(0,1,1)(1,0,0)(1,0,1)(1,1,0)(1,1,1)という8個の情報となりますね。そうか3ビットでは $2^3=8$ から8個の情報が可能となるのですね。

雅人 そうだ。4ビットは、 $2^4=16$ 個となり、8ビットでは $2^8=256$ 個の情報が扱えるんだ。

しのぶ つまり8個の(0,1)の数字列が8ビットですね。

雅人 その通りだ。ちなみに、いまの情報の10101001は16進数

では、A9 となる。そして、コンピュータでは、8 ビット (bit) を 1 バイト (byte) という単位で呼んでいる。

表 1-7 8 個の (0, 1) の数字列が 8 ビット

1	0	1	0	1	0	0	1
---	---	---	---	---	---	---	---

和昌 この 1 バイトは 8 ビットですので  $2^8 = 256$  個の情報が使えるのですね。順序だてて考えると、まず最初は 8 個の数字がすべて 0 の場合ですね。つぎが、1 衍目が 1 で他は 0 の場合です。これを続けていくと、表 1-8 のように、のように、256 個の異なるデータを確かに収納できることを確認できます。

表 1-8 1 バイト (8 ビット) で表現できる情報量は 256 個ある。

0	0	0	0	0	0	0	0	0
:	:	:	:	:	:	:	:	:
0	1	0	0	0	1	0	1	69
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	255

しのぶ どうして、8 ビットを 1 バイトとしたのでしょうか。

雅人 これは、アメリカ発祥の単位なんだ。つまり、英語のアルファベットの大文字、小文字や数字や数学記号など、よく使う記号を足すと、7 バイト、つまり  $2^7 = 128$  個よりも多くなってしまう。一方、8 ビットつまり 256 個あれば、十分足りるということがわかったんだ。このため、8 ビットが基本で、それを 1 バイトと呼ぶようになったと言われている。

和昌 そういうことだったのですね。では、日本ではひらがな 46 個、カタカナ 46 個で、これだけで 92 個ありますね。その他アルファベットも必要ですから、1 バイトは 8 ビットでは足りなかったのですね。

雅人 漢字を入れなければ 2 バイト程度だったと言われているね。

しのぶ そうか漢字は多いですよね。常用漢字だけで 2136 個もあります。

#### 1.4. 紙リールへの記録

雅人 実は、わたしが学生のころには、この 8 ビットの数字を紙のリールを使って入力していたんだ。たとえば、図 1-2 に示すようなたて 8 個の列からなるリールにパンチで孔を空けていく。そして、孔が開いた箇所を 1、開いていない箇所を 0 とする。すると、リールの 1 列が 8 ビットの情報量に対応する。

結美子 なるほど。確かに、この方法ならば、たて 1 列に 8 ビットの情報が打てますね。たとえば、00010001 は○○○●○○○●というパンチ列に対応します。

雅人 そうなんだ。そして、プログラムを読むときには、光をあてる。すると孔のある箇所だけが光るので、1 と 0 の区別がつくことになる。

和昌 なるほど。これは賢いですね。

11	10	9	8	7	6	5	4	3	2	1	
○	○	●	○	●	○	○	●	○	○	○	
○	○	○	●	○	●	○	○	○	○	●	
●	●	○	○	○	○	○	○	●	●	●	
○	○	●	○	●	○	○	●	○	○	○	
○	●	○	●	○	●	○	○	●	●	●	
○	○	●	○	●	○	○	○	○	●	○	
○	●	●	●	●	○	●	●	○	●	○	
○	●	●	●	○	●	●	○	●	●	●	

図1-2 8ビットの情報を打ち込んだ紙リール。これをコンピュータに右から1列、2列と順に読み込ませて、プログラムを動かす。

雅人 さらに、アルファベットの A に 0100 0001 ○●○○○○○●を、B に 0100 0010 ○●○○○○●○のように対応させていけば、大文字 26 個、小文字 26 個、ピリオドやコンマなどが全部対応できるので、文章をつくることができるんだ。

結美子 なるほど、これによって紙リールにコンピュータがわかる情報を打ち込むことができるのですね。

雅人 ちなみに、この文字コードは、前に紹介した ASCII コードに対応していて、16 進数では、A は 41、B は 42 となっている。

しのぶ 確かに、表 1-5 の対応表を見ればそうなっていますね。つまり、英語の英数字を、このコードで読めるのですね。

雅人 まさに、その通りなんだ。ちなみに、ASCII コードの対応表を示すと、表 1-9 のようになっている。

信雄 確かに、16進数のほうがわかりやすいですね。しかし、昔は、紙リールでコンピュータに指令を出して計算させていたのですね。

表 1-9 ASCII コードの一例

0011 0000	0	30
0011 0001	1	31
0011 0010	2	32
0100 0001	A	41
0100 0010	B	42
0100 0011	C	43
0010 1011	+	2B
0011 1111	?	3F

雅人 記録した計算プログラムは、光を当てて読み取っていたんだ。すると、明暗が 1, 0 に対応する。そして、プログラムが記録された紙リールは保管ができる、何度でも使える。

和昌 とても賢い方法だと思います。1, 0 の 1 ビットだから、機械的に孔を空けるだけでできる芸当ですね。

雅人 ただし、困ったこともあったんだ。

結美子 それは、なんでしょうか。

雅人 穿孔作業を 1 列でも間違えると、プログラムが動かないという問題だ。とたんにエラー “error” が出て終わりだ。

**信雄** それは、そうですね。“go to”という指令を間違えて“ga to”と入力したのでは、コンピュータは動きません。

**雅人** 短いプログラムならばいいが、長いとどうしても打ち間違いがある。研究室には直径1メートルぐらいの紙リールのプログラムが保管してあったね。先輩たちが苦労してつくってくれたプログラムだ<sup>2</sup>。

**しのぶ** なるほど、先輩たちの所産ですね。リールがあるならば、いつでも使えます。

**雅人** ただし、プログラムは、どんどん進化する。だから、自分たちで打たないといけないことも多い。そして、1列でも打ち間違えれば、最初からやり直しだ。1週間かけて打ち込んだプログラムがダメになったこともある。さらに、作ったプログラム自体にエラーがあっても同様のことが起こる。そこで、ある時から、リールがカードに変わったんだ。

**信雄** カードを積み重ねるという方法ですね。これならば、打ち間違えたカードや、プログラムに誤りがあっても、エラーの部分だけ打ち直せばよいということになりますね。

---

<sup>2</sup> 実は、切り貼りした痕跡のある紙リールも残っていた。つまり、間違った箇所を取り取り、正しく打ち直したものを見付けていたのである。まさに神業であった。

## 1.5. プログラミング言語

### 1.5.1. 機械語

雅人 当時は画期的なことと思ったね。ところで、00010010 や 00010011 は機械語 (machine language) と呼ばれていて、コンピュータは理解できるのだが、人間が機械語を入力するのは結構大変なんだ。

しのぶ わたしは、機械語などまったく知らなかったです。

結美子 マシン語とも呼ばれていますね。そちらは聞いたことがあります。

雅人 機械語の入力は大変なので、ひとのわかる言葉でプログラムをつくり、それをコンピュータのわかる機械語に翻訳できれば、プログラムを作成したり、入力をすることがかなり楽になる。

和昌 なるほど、その言語を僕たちは習っているのですね。

雅人 そうだね。わたしが学生の頃に習ったのが、コボル (COBOL) や フォートラン (FORTRAN) だ。特に、理系の学生にとって、科学技術計算に適したフォートランがいちばんの人気だったね。

### 1.5.2. プログラミング言語

しのぶ プラグラミング言語 (programming language) のことです

よね。今でしたら、JavaScript や Python や C/C++ などが人気ですね。

**信雄** これらプログラム言語は、最後は機械語に翻訳されてコンピュータが処理するのですね。

**雅人** その通り。コンピュータが理解できるのは、(0, 1) の世界だけだからね。そして、機械語に翻訳する方式には 2 種類ある。まず、コンパイル (compile) 方式。これはプログラミング言語を一括して機械語に翻訳するやり方のことだ。たとえば、英語で書かれたプログラムを日本語にいったん翻訳してから実行するようなものだね。

一方、インタープリタ (interpreter) 方式というのは、プログラミング言語の指令をひとつずつ機械語に翻訳しながら実行する方式だ。ちょうど通訳つまり “interpreter” を介して英語と日本語でやりとりするようなものだね。

**しのぶ** なるほど。それでは、なぜ、ふたつの方式があるのでしょうか。

**雅人** それぞれ一長一短があるからだ。コンパイル方式では、一括して翻訳するのでコンピュータの実行時間を短縮できるのだが、どこかにプログラムミスがあれば動かない。一方、インタープリタ方式では、時間がかかるが、エラーがある箇所まで動いてくれる。だから、プログラム開発には向いている。

**和昌** なるほど、それぞれに特徴があるのですね。

**結美子** ところで、プログラミング言語はひとつに統一すればよいのではないでしょか。いろいろな種類があると混乱しますし、学ぶほうも大変です。

**雅人** それもひとつの考えだが、たとえば、FORTRAN は数式に慣れた理系のひとにとっては、とても使いやすい言語なんだ。

一方で、文系のひとには使いにくい。そこで事務作業用に開発された言語が、COBOL だ。これは、“Common Business Oriented Language” の頭文字をとったもので、和訳すれば共通事務処理用言語となる。だから、記述も、自然言語の英語に近いのが特徴だ。ちなみに、FORTRAN の名は “formula translation” に由来している。“formula” は公式、“translation” は翻訳という意味だから、数式翻訳用のプログラム言語となる。

**しのぶ** なるほど。確かに、数式になじみがあるかないかは大きい違いですね。

**雅人** さらに、Python はインタープリタ方式だが、FORTRAN や C/C++ はコンパイル方式という違いもある。

**結美子** プログラミング言語によって、機械語翻訳方式に違いがあるのですね。

**雅人** ただし、Java の基本はコンパイル方式だが、仮想マシンで一部機械語に翻訳されて動くのでインタープリタ方式も取り入れていることになる。

しのぶ ますます、複雑になっていますね。

信雄 ひとことで、コンピュータと言っても、それを動かすソフトのほうは、いろいろな方式や言語があり、それぞれに特徴があるということですね。

### 1.5.3. BASIC言語

雅人 そうなんだ。わたしが学生の頃に重宝したのは、“BASIC”というプログラミング言語だね。なにしろ、当時の容量の少ないPC (personal computer) でも動かすことができたんだ。画期的だったよ。

和昌 BASIC は聞いたことがあります。NEC が開発した N88-BASIC をおじさんが使っていました。

雅人 実は、1982年にNECがN88-BASICを標準装備したPC9801というパソコンを 298000 円で売り出したのだが、飛ぶように売れたんだ。わたしも、博士の学生だったが、バイトで貯めたお金ですぐに購入した。

結美子 当時の 30 万円というのは結構大きいお金だったのではないかでしょうか。

雅人 それ以上に魅力的だったということだ。なにしろ、BASICは扱いやすく、初心者でも、それほど苦労せずに使えたからね。たとえば

```
10 A = 5
20 B = 3
30 C = A + B
40 print C
```

というプログラムでは、 $5 + 3 = 8$  という結果ができるのだが、数式や文字式と言語が対応しているので、実に簡単だったんだ。

信雄 確かに、単純ですね。BASIC 言語を習っていなくともわかります。

雅人 さらに、グラフィック機能にも対応していて、それも簡単なコマンドなんだ。たとえば、円を描く際

```
10 CIRCLE(3, 8), 2
```

と入れれば、中心座標を  $(3, 8)$  として半径 2 の円を描いてくれる。また、線を引くときは

```
20 LINE(3,8)-(5,11)
```

とすれば、座標  $(3, 8)$  から  $(5, 11)$  に線を引けという命令となる。

和昌 確かに簡単なコマンドですね。

雅人 その結果、BASIC を使えば、いろいろなコンピュータゲ

ームを自作することができたんだ。それが、人気の背景にあつたんだ。

**しのぶ** 昔は、自作のゲームを販売することもできたと聞きました。

**雅人** そうなんだ。当時は、カセットテープで販売されていたからね。それで、一躍ヒーローになったひとも結構いたんだ。わたしも簡単なゲームをつくったら、研究室の後輩たちが夢中になってしまって、教授からしかられたことがあったよ。今となっては、懐かしい思い出だね。

**信雄** 結局、コンピュータを扱う場合、コンピュータ本体を機械として扱うハードウェア (hardware) 分野と、それをうまく使いこなすためのツールであるソフトウェア (software) 分野が必要ということですね。

**雅人** そうなんだ。PC9801 は、これら両方を備えていたんだ。そして、ハードとソフトは、車の両輪のように、互いをけん引しながら、コンピュータの世界の発展を支えてきたんだ。

**結美子** それはよくわかります。ハードウェアでは、処理速度の速さやメモリの大きさに進歩がありますが、ソフトウェアでも、数多くのソフトが開発され、機能がどんどん高度になっていきます。

**雅人** そうなんだ。実は、FORTRAN という言語も、1957 年に

IBM704 用に開発された初代のものから FORTRAN II, III, IV, 66, 77, 90 のように、どんどん進化しているんだ。いまでは、FORTRAN2018 もリリースされている。

### 1.6. 容量

和昌 最近テレビコマーシャルでよく、ギガという言葉を聞きますが、あれは、ギガバイトのことでしたでしょうか。

雅人 その通り、まず 1 バイトが 8 ビットで 256 個の情報が入っている。ギガ (giga: G) は接頭語で  $10^9$  という意味だ。順序だてでいくと、キロ (kilo: k) が  $10^3 = 1000$ 、メガ (mega: M) が  $10^6 = 1000000$ 、ギガ (giga: G) が  $10^9 = 1000000000$ 、テラ (tera: T) が  $10^{12} = 1000000000000$  となる。整理すると表 1-10 のようになる。

しのぶ 確か、最近買ったパソコンのハードディスクの容量がギガではなく、テラになっていて、とても驚いたことがあります。年々、容量が大きくなっていますよね。そう言えば、メモリディスクの容量もそうですね。

表 1-10 デジタル情報の単位

バイト [B]	1 [B]	1	情報の基本単位 8 bit
キロバイト [kB]	$10^3$ [B]	1000	短い文章は数 kB
メガバイト [MB]	$10^6$ [B]	100 万	1 分間の音楽は 1[MB]
ギガバイト [GB]	$10^9$ [B]	10 億	4000 枚の写真は 1[GB]
テラバイト [TB]	$10^{12}$ [B]	1 兆	1[TB] で動画 100 時間以上

雅人 いまのテレビでは、動画を記録できるメモリ容量を持っているが、それがテラバイトだね。

和昌 ところで、最近のパソコンでは、よく 32 ビットや 64 ビットと聞きますが、バイトと違いますね。

雅人 ビットを思い出そう。1 ビットとは  $2^1 = 2$  の情報量で 0 と 1 の 2 個だったね。そして、8 ビットとは、 $2^8 = 256$  個の情報量だった。同様にして 32 ビットとは、2 の 32 乗の情報量となる。また、64 ビットは 2 の 64 乗という情報量だ。

結美子  $2^{10}$  が 1024 ですから、 $2^{30} = 2^{10 \times 3} = (1024)^3$  ですね。これは  $1000^3$  とすると、 $10^9$  となって 10 億程度ですね。 $2^{32}$  は、さらに 4 倍ですから、約 40 億の情報量になります。

雅人 その計算はいい線いっているね。実際には

$$2^{32} = 4294967296$$

で、約 43 億となる。

信雄 それは、すごい数ですが、64 ビットは  $2^{64}$  の情報量ですね。よって

$$2^{64} = (2^{32})^2 = 4294967296^2$$

となります。つまり、およそ 43 億  $\times$  43 億ということですね。 $(4.3 \times 10^9)^2$  ですから

$$18.49 \times 10^{18} \approx 1.849 \times 10^{19}$$

のように、 $10^{19}$  のオーダーとなります。

雅人 そうだね。億が 10 の 8 乗 ( $10^8$ )、兆が 10 の 12 乗 ( $10^{12}$ )、そのつぎは京（けい）で 10 の 16 乗 ( $10^{16}$ ) となる。つまり 1849 京という数となる。実際に計算すると

$$2^{64} = 18446744073709551616$$

のように 20 行の数となる。

しのぶ これは、すごい数ですね。確か、OS (operating system) の Windows10 では 32 ビットと 64 ビットの 2 種類がありました、容量が 2 倍程度と思っていました。実際には、これだけ大きな情報量の差があったのですね。

雅人 そして、Windows11 では、64 ビットだけになっている。そう言えば、前に紹介した NEC の PC98 シリーズは 16 ビットだった。それまでの PC88 シリーズが 8 ビットだから、画期的な容量アップだったんだ。そのため、ROM (read only memory) に BASIC を標準装備することができた<sup>3</sup>。それまでは、プログラムはフロッピーディスクを挿入して読み込ませていたからね。

和昌 いまの PC は、ハードディスクの容量が大きいので、それが当たり前ですが、昔は、いちいちフロッピーディスクを挿入していたのですね。

## 1.7. デジタルの世界

雅人 コンピュータは情報をデジタル化しているのだが、画像

---

<sup>3</sup> ROM とは読み出し専用のメモリのことである。パソコンの基本プログラムである Windows のような OS や Word や Excel などの基本ソフトを収容している。

や、音、映像なども 0 と 1 の 2 個の数字の集まりとして扱われている。この 2 個の 2 進数の組み合わせであるビットによって、あらゆる情報を表現することができる。これがデジタルの世界の特徴だ。

たとえば、位置を指定するためには座標が必要になる。そこで、図 1-3 のような 9 個の座標を考える。そして、右表のように、それぞれに 8 ビットの数列を当てはめるんだ。

7	8	9	00000111	00001000	00001001
4	5	6	00000100	00000101	00000110
1	2	3	00000001	00000010	00000011

図 1-3 位置座標と 8 ビットデータの対応

しのぶ なるほど。こうすれば、8 ビットの数値を入れれば、どの座標かがわかるのですね。

雅人 そうだ。たとえば、5 の位置を光らせたい場合、00000101 という数値を入れると

7	8	9
4	5	6
1	2	3

図 1-4 8 ビットの数値の 00000101 を入れると座標 5 が点灯する。

のように、5 の位置の座標が反応することになる。このように、

位置を指定して光らせれば、いろいろな文字や絵を描くこともできる。

**信雄** そう言えば、昔のテレビゲームもそうでしたね。確か、たまごっちの画面もこんな絵でした。

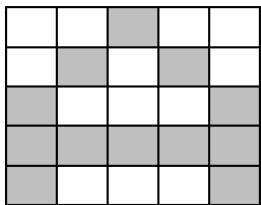


図 1-5 指定した座標の数値を打ち込むとデジタル化された文字(A)が作製できる。

**結美子** 昔のゲームはみんなギザギザがありましたね。インベーダーゲームもそうでした。スーパーマリオもそうですね。

**しのぶ** なにか、そのほうが風情がありますね。いまのゲームは現実世界のようで戸惑うこともあります。

## 第2章 ハードウェア

雅人 機械であるコンピュータは1と0しか認識できない。それでも、高速計算が可能となっている。それでは、コンピュータでは、どうやって1と0を認識しているのだろうか。

信雄 1と0ならば、あるとなしの2種類の動作で済みますね。たとえば、「ランプがついている、ついていない」など、オンオフはいろいろな操作で対応できるのではないでしょか。

雅人 そうだね。ただし、現在のコンピュータでは、「電流のあるなし」をメインに利用している。

しのぶ 電流が流れているか流れていないかを1と0に対応させているのですね<sup>4</sup>。

雅人 その通り。そして、これを高速で処理するために利用されるのが半導体(semiconductor)だ。

---

<sup>4</sup> メモリにおいては、磁場があるかないかを利用している。

信雄 なぜ、半導体なのでしょうか。金属に電流を流すか流さないかでも同じだと思いますが。

雅人 まず、電気伝導性で物質を分類すると、導体 (electric conductor)、半導体 (semiconductor)、絶縁体 (insulator) の 3 種類となる。導体の代表は、金、銀、銅などの金属だね。絶縁体の代表は、陶器やゴムやプラスチックだ<sup>5</sup>。

結美子 これら特性は、電気抵抗 (electric resistance) によって区別できたのでしたね。

雅人 厳密な境界線が定義されているわけではないが、図 2-1 のような分類となっている。ここでは、電気抵抗ではなく抵抗率 (resistivity) が基準だ。

しのぶ 抵抗率というのは、物質定数でしたね。

雅人 そうだ。電気抵抗 ( $R: \Omega$ ) は、物質の断面積 ( $A: \text{m}^2$ ) と長さ ( $L: \text{m}$ ) の影響を受ける。そこで、規格化した抵抗率 ( $\rho: \Omega\text{m}$ ) が使われ

$$\rho = RA / L$$

と与えられる。つまり、ある物質の断面積 1 [ $\text{m}^2$ ] で長さが 1 [ $\text{m}$ ] の電気抵抗ということになる。

和昌 単位を追うと、抵抗率は  $\rho = [\Omega][\text{m}^2] / [\text{m}] = [\Omega\text{m}]$  となるの

---

<sup>5</sup> いまでは、導電性のプラスチックも開発されている。

ですね。

信雄 ところで、半導体の抵抗率を見ると、 $10^{-6}$ から $10^6$  [ $\Omega\text{m}$ ] と  
かなり広範囲にわたっています。

雅人 実は、これが半導体の大きな特徴になる。この変化は、  
温度や印加する電圧でも異なるし、他の元素を添加することで  
も大きく変化する。そして、それを利用すると、多様な素子が  
つくれるんだ。

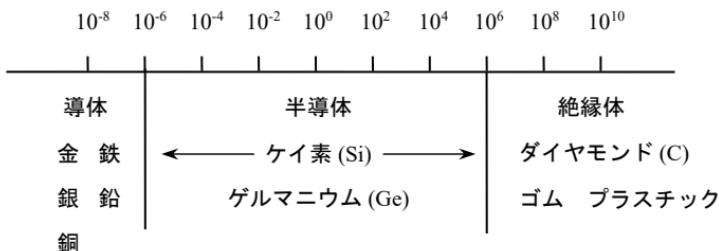


図 2-1 導体、半導体、絶縁体の抵抗率 [ $\Omega\text{m}$ ]

雅人 そして、コンピュータ応用を考えたときに重要なのは、表 2-1 の性質だ。

表 2-1 電気特性と ON/OFF 機能

ON		OFF
○	導体	
○	半導体	○
	絶縁体	○
電流あり		電流なし

和昌 なるほど、金属などの導体は、電流が流れるので ON の状態しか実現できず、絶縁体では、電流が流れないので OFF の状態しかできないのですね。一方、半導体は、この中間なので、ON と OFF の状態がつくれるということですか。面白いですね。

雅人 まさに、その通り。もともと半導体は中途半端な性質で、使い物にならなかつた。たとえば、電流を流すときには導体の金属を使うが、剥き出しのままだと危険なので、絶縁体でまわりを覆つて安全に使える。しかし、半導体は、どちらの用途にも使えない。

しのぶ 確かに、導線にも絶縁層にも使えませんね。

## 2.1. 半導体とドーピング

雅人 ところが、その半導体が一躍エレクトロニクスの主役に踊り出ることになる。その代表がシリコン (silicon: Si) だ。ただし、シリコンだけでは利用価値はほとんどないんだ。

和昌 それは、P型半導体とN型半導体のことでしょうか。

雅人 まさに、その通り。みんなはP型とN型の違いは知っているかな。

しのぶ Pはpositiveつまり正、Nはnegativeつまり負に由来します。それを理解するには Si が基本となるのでした。

和昌 シリコン Si は真性半導体です。炭素 C と同じように、4本の結合手があるので、ダイヤモンド構造をとります。これが基本です。ただし、このままでは電流が流れません。

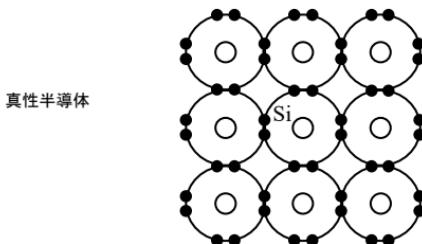


図 2-2 Si は 4 本の結合手があり、M 裂が 8 個の電子で埋まる。そのため、常温では、自由に動ける電子（自由電子）が存在せず、真性半導体となる。

結美子 そこで Si に 3 値の元素であるホウ素 (boron: B) などを添加すると、図 2-3 に示したように、その位置だけ負の電子が 1 個足りなくなります。その結果、あたかも + に帯電しているようにみえます。これが P 型半導体ですね。このとき、電子の欠けた位置を正孔 (positive hole) と呼びます。

雅人 その通り。そして、この正孔が動くことで伝導性が生じ、真性半導体の Si よりも電気が通りやすくなる。

しのぶ さらに、B の添加量を変えることで、電気伝導を担う正孔の濃度を制御することができるのですね。B のように価数の異なる元素を Si に添加するのがドーピング (doping) でしたね。

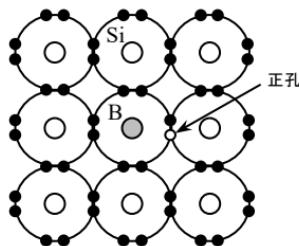


図 2-3 P型半導体: Siに3価のBを添加すると、電子の抜けた孔である正孔が形成される。正孔はプラス+に帯電しているため、正の電荷が注入されたことになる。

信雄 一方、4価のシリコンに、5価の元素のヒ素 (arsenic: As) をドーピングすると、その位置だけ負の電子が1個余分に存在することになります。そのため、1個の Si を As で置換するごとに、動くことのできる電子が1個生成します。これがN型半導体でしたね。

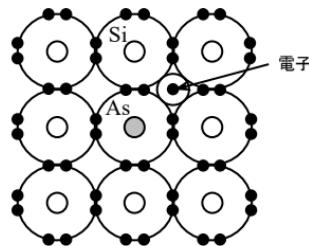


図 2-4 N型半導体: Siに5価のAsを添加すると、負の電荷が注入される。

雅人 その通りだ。そして、P型とN型半導体の組み合わせによって、多くの機能をもった半導体素子が誕生している。

## 2.2. ダイオード

雅人 みんなはダイオード (diode) という半導体素子を知っているかな。

和昌 もちろんです。P型半導体とN型半導体を接合させたものですよね。

しのぶ 整流作用 (rectification) があるので、交流 (alternate current: AC) を直流 (direct current: DC) に簡単に変えることができると習いました。

雅人 実は、PN接合には、整流作用だけでなく、太陽電池 (solar battery) や発光ダイオード (light emitting diode: LED) などの魅力的な応用もあるんだ。ここでは、まず基本となる整流作用の原理を見てみよう。図2-5に電流が流れない場合を示してある。

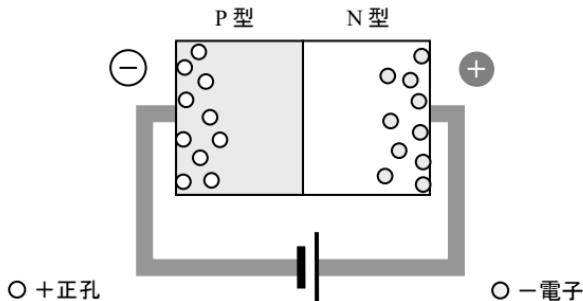


図2-5 PN接合に電流が流れない配置: 逆方向電圧

しのぶ N型半導体側に正極（+側）を、P型半導体側に負極（-側）をつなげていますね。

結美子 すると、N型半導体中の電子が正極側に引き寄せられます。同じようにP型半導体中の正孔が負極側に引き寄せられますので、PN接合部分では電荷が存在せず、電荷の移動が生じないので、電流が流れません。

結美子 これを逆方向電圧 (reverse voltage) と呼ぶのでしたね。

雅人 そうだ。そして、PN接合界面にできる電荷の存在しない領域（電荷欠乏層）は絶縁体となる。

和昌 つぎに、電極の配置を変えたときの状況を考えてみます。すると、図2-6のようになります。N型半導体に負電極から電子が流れてきます。このため、電子の濃度が高くなりますので、電子はP型半導体側に押し出されます。一方、P側にあった正孔は負極に引かれて右側に移動します。

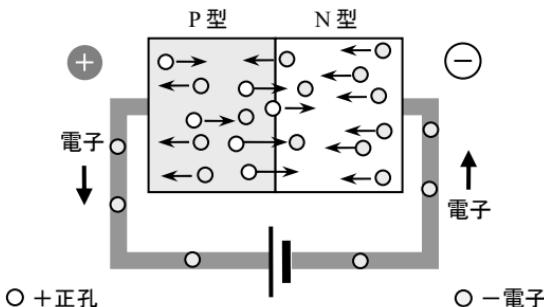


図2-6 PN接合に電流が流れる配置: 順方向電圧

**信雄** P 側に押し出された電子は、N 側に引かれた+の正孔に出会いうと電荷は消えてしまいますね。それでも電子は、どんどん押し寄せるので、電子は P 側を越えて正極に向かうことができるのですね。だから電流が流れます。

雅人 定性的には、その説明で十分だと思う。そして、この配置を順方向電圧 (forward voltage) と呼んでいる。

### 2.3. 正孔の移動

しのぶ 実は、この正孔 (positive hole) のことがよくわかりません。これは、電子が抜けた孔 (hole)のことですよね。原子としては-が足りないから、結果として+になるのはわかります。でも孔はあくまでも孔で、それが図 2-6 のように動くということがよくわからないのですが。

雅人 なるほど、動けるのはあくまでも実体のある電子なはずだからね。では、図 2-7 を使って正孔の移動を説明してみよう。

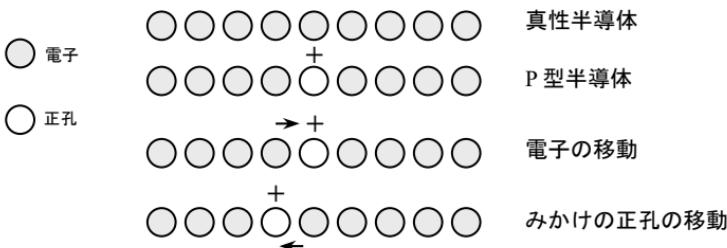


図 2-7 正孔の移動。実際に移動しているのは電子であるが、図のように電子が右に移動すると、あたかも正孔が左に移動したように見える。

しのぶ Si のような真性半導体 (intrinsic semiconductor) では、すべての格子が電子で埋まっています。それがいちばん上の図ですね。そして、P 型半導体は、この状態から電子を取り去った状態なので、そのつぎの図となります。そして、電子の埋まっていない箇所は + に帯電しますので正に帯電した孔、つまり、正孔と呼ぶのでしたね。

雅人 ここで、P 型半導体において、正孔の隣りの電子が移動してきたとしよう。すると、正孔のあった場所を電子が埋め、もとの電子の位置に正孔ができる。つまり、移動しているのは電子なんだが、あたかも正孔が逆方向に移動したように見えるんだ。

しのぶ なるほど。移動するのは実体のある電子ですが、結果として、正孔が移動しているように見えるのですね。

雅人 納得したかな。ちなみに、回路図を描くときのダイオードの記号は図 2-8 のようになる。

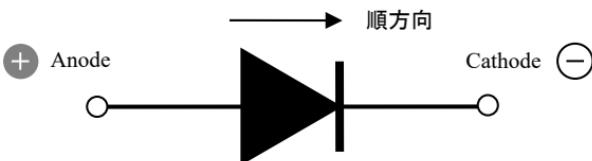


図 2-8 ダイオードの回路記号

しのぶ ダイオードは交流を直流に変換できるのでしたね。

雅人 ああ、交流回路にダイオードを1個挟めばそれで済む。すると図2-9のようだに、交流電流が正の部分だけになる。



図2-9 ダイオードを利用した整流作用: 半波整流

結美子 逆方向では、電流が流れないからこうなるのですね。しかし、このまでは、半分、電力を無駄にしていますね。

雅人 そうなんだ。これを半波整流(half-wave rectification)と呼んでいる。そこで、図2-10のような回路をつくったひとがいる。全波整流回路と呼ばれているものだ。

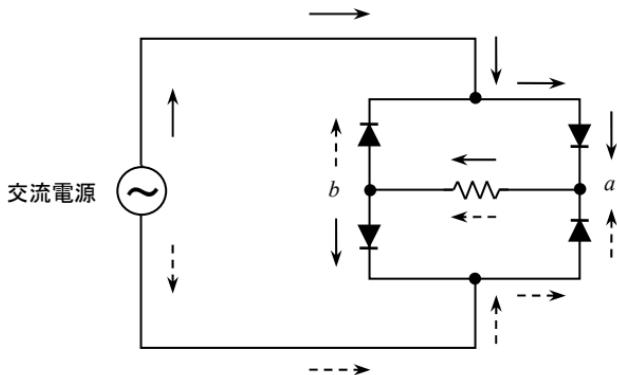


図2-10 全波整流回路: 図の回路のab間に負荷を置けば、常に電流はaからbの一方향に流れる。

和昌 ダイオードを4個使っているのですね。確かに、電流を順に追つていけば、*a*から*b*では、どちらの場合でも同じ方向に電流が流れますね。

雅人 その結果、交流電流は、図2-11のように整流されるんだ。これならばロスはないよね。これを**全波整流**(full-wave rectification)と呼んでいる。



図2-11 全波整流回路を用いた交流電流: 全波整流

結美子 なるほど、うまいことを考へるひとがいるのですね。

雅人 電気回路には、いろいろな知恵がつまつまついていて、人類の科学的所産と思うよ。学生の頃、秋葉原でダイオードを買って整流回路をつくっていたんだが、結構、面倒くさいんだ。そのため、図2-10のような素子に仕上げた整流回路がパッケージとして売られていたんだ。とても重宝したよ。

## 2.4. トランジスタ

雅人 それでは、重要な半導体素子としてトランジスタ(transistor)を取り上げてみよう。これは、PNP接合あるいはNPN接合からなる素子のことなんだ。

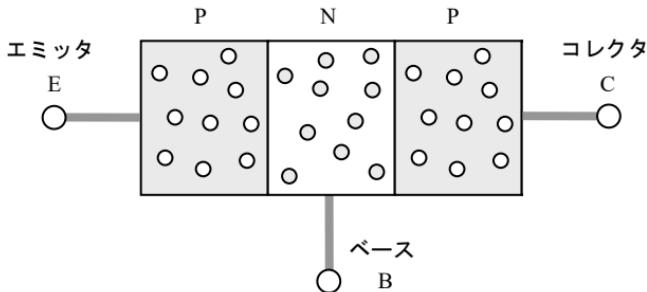


図 2-12 PNP 接合型トランジスタの構造

和昌 トランジスタには3個の足がありましたね。それぞれ、ベース (base: B)、エミッタ (emitter: E)、コレクタ (collector: C)です。

結美子 “base” は、ベースになる「起点」という意味ですね。“emitter” は「放出する」という英語の “emit” の名詞形です。一方、“collector” は「集める」という英語の “collect” の名詞形ですが、機能がよくわかりません。

雅人 なかなか、いい線いっているね。実は、トランジスタには増幅という重要な機能があるんだ。弱いベース電流 ( $I_B$ ) を流すと、エミッタからコレクタに向かって大きなコレクタ電流 ( $I_C$ ) が流れるんだ。エミッタから “emit” つまり出て行く電流を、コレクタで “collect” つまり「集める」という意味となる。このため、この名がついている。これら3端子の機能については後ほど詳しく説明する。

信雄 なるほど、トランジスタのメインの機能は増幅作用

(amplification) のですね。いまの説明でいければ、增幅率 =  $I_C / I_B$  となりますね。

雅人 その通り。正式には直流電流増幅率 (direct current amplification factor) と呼んで  $h_{FE}$  や  $\beta$  という記号を使う。この機能は、なかなか便利で、たとえばラジオに使うと弱い電波信号を増幅してくれる。いわゆるトランジスタラジオ (transistor radio) だ。

しのぶ その名前は聞いたことがあります。

雅人 昔は、増幅回路に真空管 (vacuum tube) が使われていたので、ラジオは大きくて卓上が当たり前だったんだ。それが、トランジスタに替って、コンパクトとなり、携帯できるようになったのだから画期的だったね。

和昌 卓上ラジオですか。茶色い箱形のものですよね。昭和初期の番組では時々見ますね。

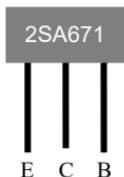


図 2-13 トランジスタの外観

雅人 学生のころは、研究室で使う電気回路を半田ごてで自作

していたので、電子部品を秋葉原に、よく買いに行ったんだ。トランジスタは、図2-13のような3本足の端子で、安いのは壊れていたりしたので、回路に組み込む前はチェックするのが基本だった。産業ごみに捨てられている電化製品を分解して、部品を再利用したりもしていたね。

信雄 そうなんですか。いまは、電気回路実験で少しかじる程度ですね。ところで、もうひとつのトランジスタはNPN型でしたね。

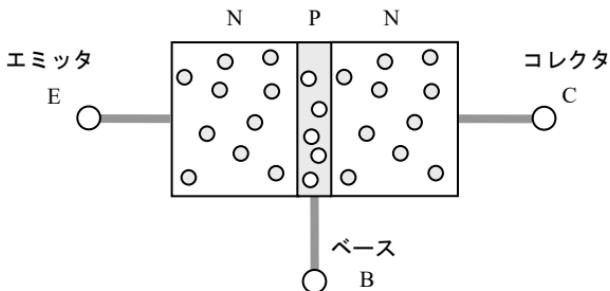


図2-14 NPN型トランジスタの構造：実際のトランジスタでは、この図のように、ベース (B) に対応した半導体部分の厚さは薄くしてある。

雅人 そうだ。図2-14の構造をしている。実際のトランジスタでは、この図のように、ベース (B) に対応した半導体部分の厚さは薄くしてある。

しのぶ この薄いということが重要なのでしょうか。

雅人 そうなんだ。この部分に流れる電流をベース電流 ( $I_B$ ) と呼

ぶが、わずかな電流で NP 接合の電流遮断作用をはずすというのがキーになる。

**結美子** 先ほど、コレクタ電流の話が出てきましたが、図 2-14 を見ると、コレクタ (C) からエミッタ (E) に向けて電流を流そうとしても、途上に NP 接合があるために、電流は流れませんね。

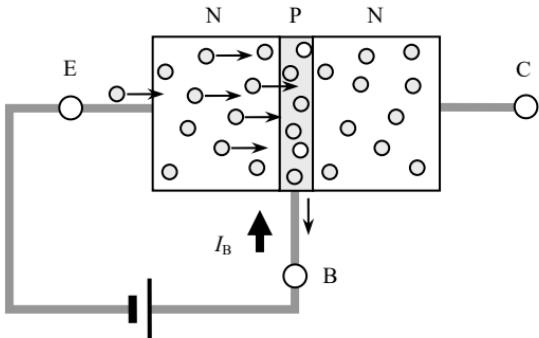


図 2-15 ベース電流を流すと、P 型半導体内の正孔が電子で埋められ、さらに、電子の流れが生じる。

**雅人** ここでベース電流が登場する。図 2-15 のようにベース電流を流すと、左の NP 接合に対しては順方向電圧となる。すると、電子が P 型半導体に注入され電子の流れができる。

**しのぶ** なるほど。その結果、右の PN 接合のブロックがなくなるのですね。あえて言えば、NN 接合のように変わるのでですね。すると、EC 間の導通をブロックしていた接合がなくなり、E から C に向けて電子が移動できるようになります。電流とすれば、

C から E に向けて流れようになるということですね。

雅人 まさに、その通り。ベースとなる P 型半導体の厚さが薄ければ、わずかな電流で、この電流ブロックがはずれるということになる。

結美子 まるで、蛇口のような働きですね。

雅人 その喻えはわかりやすいね。ここで、さらに、図 2-16 に示すように外部電源につなげると、C から E に向けて大きな電流が流れることになる。

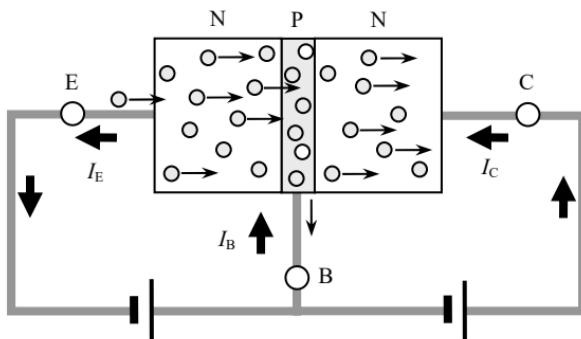


図 2-16 外部電源をつなげると、CE 間に電流が流れれる。  
トランジスタ内部の矢印は電子の移動を示している。

結美子 なるほど。ベース電流を流すことで、NPN の P のバリアがなくなり、CE 間に電流が流れれるのですね。このとき

$$I_E = I_B + I_C$$

という関係になります。

和昌  $I_B = 1 \text{ [mA]}$ ,  $I_C = 100 \text{ [mA]}$  とすると、 $I_E = 101 \text{ [mA]}$  ということになりますね。

しのぶ つまり、増幅率は、 $h_{FE} = I_C / I_B = 100 / 1 = 100$  となります。

信雄 この増幅原理がトランジスタラジオなどに使われているのですね。

雅人 ここで、回路図も示しておこう。PNP と NPN 型トランジスタの回路図における記号は、図 2-17 のようになる。

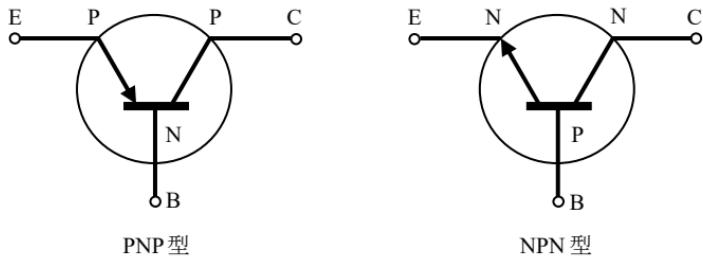


図 2-17 トランジスタの回路図記号

しのぶ 矢印のある端子がエミッタ (E) なのですね。

雅人 その通り。そして、矢印の向きは  $P \rightarrow N$  となっている。こ

れと対称位置にある端子がコレクタ (C) 、そして、真ん中の端子がベース (B) となる。

**結美子** ところで、ラジオに使われる電波は、交流信号ですよね。その場合の増幅はどうなるのでしょうか。

**雅人** そうだね。電波では電場が振動しているため、それをアンテナが受信すると、交流電流が誘導される。これがラジオへの入力となる。この交流電流を、ベース電流として、コレクタ電流も交流となり、その振幅が図 2-18 に示すように増幅されるんだ。

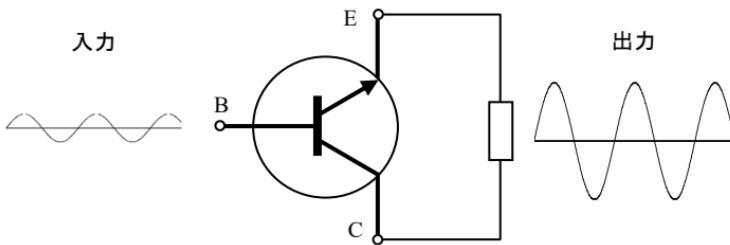


図 2-18 トランジスタを利用した交流信号の増幅

**しのぶ** 先生、ところで主題は高速でオンとオフを制御できる半導体素子の話でしたね。

## 2.5. ON/OFF を制御するトランジスタ

雅人 それが主題だったね。実は、もうすでにスイッチングの原理のヒントは説明しているんだ。それは何だと思う？

信雄 もしかしてベース電流でしょうか。これが蛇口の役目をすると言われていましたよね。

雅人 まさに、その通り。ベース電流を流すと、薄いベースの半導体層に電子が注入され、コレクタ電流が流れる。いままで、定量的な話をしてこなかったが、コレクタ電流が流れるための限界のベース電流が存在する。つまり、このしきい値より、低ければOFF、高ければONという状態がつくれるんだ。

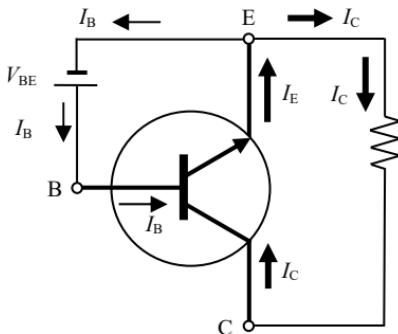


図 2-19 ベースとエミッタ間に流れる電流によってコレクタ電流を ON/OFF することが可能である。実際には、電源  $V_{BE}$  の大きさで制御が可能となる。

しのぶ なるほど、ベース電流によって ON/OFF が制御できるの

ですね。これが、コンピュータで 1 と 0 のビットを作り出しているのですね。

## 2.6. 電界効果トランジスタ

雅人 ただし、いまのコンピュータの ON/OFF に使われているトランジスタはフェット (field effect transistor: FET) と呼ばれているものだ。日本語では、電界効果 (field effect) トランジスタとなる。いまの FET は、MOS (metal oxide semiconductor) つまり金属酸化物を使うことが多いので、MOSFET モスフェットと呼ばれている。ちなみに、すでに紹介した PNP と NPN 型のトランジスタは、バイポーラトランジスタ (bipolar transistor) と呼ばれている。

和昌 電界 (field) ということは、電流で制御しないという意味なのでしょうか。

雅人 その通り。そこが大きなポイントだね。電流を流そうとすると電源の切り替えに時間がかかるんだ。FET は、その必要がないので、かなりの高速で ON/OFF をスイッチできる。

しのぶ 電界で制御するという意味がよくわからないのですが、どうするのでしょうか。

雅人 それでは、まず、FET の構造からみてみよう。それは、図 2-20 のようになる。

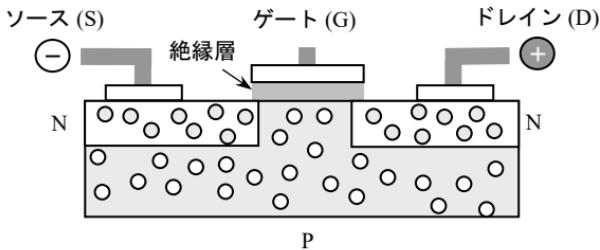


図 2-20 電界効果トランジスタ (FET) の構造: N 型 FET

結美子 バイポーラトランジスタでは、端子はベース (B) 、エミッタ (E) 、コレクタ (C) でしたが、FET では、ゲート (G) 、ドレイン (D) 、ソース (S) となっていますね。

雅人 まず、ベース (B) とゲート (G) が対応していることはわかるね。

和昌 それは、わかります。ただし、図をよく見ると、FET のゲートは本体と絶縁されていますね。

雅人 そうなんだ。絶縁層には  $\text{SiO}_2$  が使われている。そして、ゲートから電流は流れない。その代わり、電圧を加え、電界を印加して制御するんだ。このため、電界効果トランジスタと呼ばれている。

しのぶ つまり、バイポーラでは、ベースの電流で ON/OFF を制御しているのに対し、FET では、ゲートの電圧で制御しているのですね。

雅人 その通り。ここで、ゲートに正の電圧を加えたときの変化を図2-21に示してある。

信雄 面白いですね。ゲートに電圧を加えただけで、電流が流れるようになるのですね。これは、ゲートに正の電圧を加えると、ちょうど、コンデンサのように、絶縁体を介して、半導体側に負の電子が引き寄せられることが原因ですね。

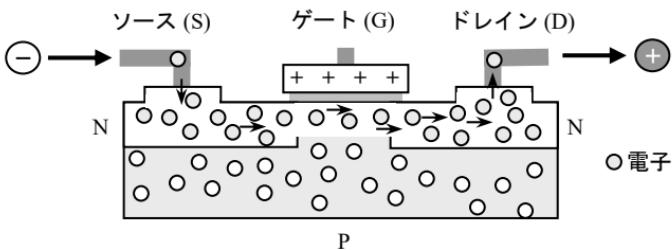


図2-21 FETがONの状態: ソースからドレインに向かって電子の流れが生じる。このとき、電流は+のドレインから、-のソースに流れることになる。

しのぶ 一方では、正の電圧付加によって、正に帯電した正孔が反発されると見ることもできますね。その結果、ソースから供給された電子がドレイン側に移動できるようになりONの状態になるのですね。

雅人 そして、電圧を切れば、ふたたびOFFの状態に戻る。電圧のON/OFFでスイッチングができるので、かなり高速で切り替えが可能となる。これが、FETがコンピュータの1と0つまりON/FFに利用される理由だ。ちなみに、NPN構造のFETはN型

FET と呼ばれている。

結美子 PNP 構造は P 型 FET となるのですね。ただし、この場合は、正孔が移動するのですね。

和昌 でも、前に学習したように、正孔の移動と言っても、実際に移動するのは電子でした。

雅人 その通り。それから、コンピュータのトランジスタに MOSFET が使われる理由にはもうひとつある。それは、集積化が簡単という理由だ。バイポーラ型では、NPN あるいは PNP の 3 種の半導体の積層が必要となる。これが、結構面倒なんだ。一方、MOSFET では、P 型を基板として、平面上に並んだ NN の上に絶縁層とゲート電極を設ける構造なので、集積するのが簡単なんだ。これがコンピュータに採用される大きな理由となっている。

しのぶ なるほど、Si 基板の上に、少し工夫を加えるだけで FET が機能するのですね。

雅人 その通り。ちなみに、MOSFET の回路図記号は図 2-22 のようになる。

和昌 ゲートに対向した点線の部分が、ゲートに電圧を加えると、つながって迂回路ができるという構造ですね。

雅人 その通りだ。

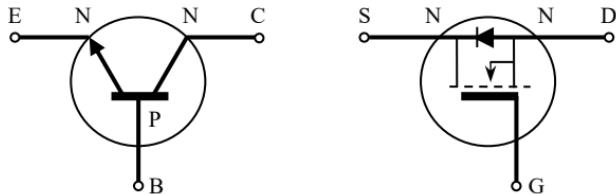


図 2-22 N 型 MOSFET の回路図: ゲート側端子では絶縁されており、電流が流れないことがわかる。

結美子 ところで、1 個のトランジスタでは 1 ビットにしか対応できません。とすると、8 ビットでは、8 個のトランジスタが必要になるのですね。

雅人 そういう計算になる。

信雄 とすれば、32 ビットでは、約 40 億個のトランジスタが必要ということになりますね。ちょっと想像がつかないのですが。

雅人 実際のコンピュータには、このようにとてつもない数のトランジスタが集積されているんだ。そのために、基板にちょっとした工夫で動作する MOSFET が重宝されることになる。

## 2.7. メモリ

和昌 コンピュータがこれだけ進展すると、メモリ (memory) のことが話題になりますね。コンピュータの性能と言えば、記憶装置の容量が引き合いに出されます。

**しのぶ** メモリの進展はものすごいと思います。わたしが購入した電子辞書では、なんと 300 冊以上の分厚い辞書のデータが入っています。

**結美子** 最初の頃は、英語の辞書が入っているものを買っていましたが、いまでは、英和、和英、英英、国語辞典だけでなく、広辞苑や百科事典まで入っていますよね。その進歩はすごいと思います。

**雅人** だからこそ、メモリの市場は毎年成長しているんだ。いまは年間 20 兆円とも言われている。

**信雄** それは、すごい市場ですね。ところで、どのような機構でコンピュータは記録しているのでしょうか。

**雅人** ひとくちにメモリと言っても、いろいろな種類がある。もちろん、コンピュータを動かすためには、いろいろな記憶装置が必要になる。たとえば、ROM と RAM、さらに外付けの記憶装置などもあるね。

**しのぶ** ROM は “read only memory” で、日本語に訳せば「読み取り専用メモリ」ですね。パソコンの基本プログラムが収納されています。電子辞書なども ROM ですね。

**和昌** RAM は “random access memory” の頭韻で「アクセス可能なメモリ」つまり、「書き換えのできるメモリ」という意味です。パソコンを作業するときに必要なデータを一時書き込む場所で

す。ROMはパソコンの書棚で、RAMは作業机にたとえられます。

雅人 その通りだ。以前に紹介した、8ビットのデータを紙リールに穿孔したものも一種のメモリとみなせるが、書き換えはできないからROMのカテゴリに入る。

信雄 紙ではデータ量の増加には限界がありますね。

雅人 ところで、紙リールでは孔があるかないかで1,0に対応させていたね。結局のところ、コンピュータのメモリとは、この1,0をどのように表現するかの問題なんだ。

結美子 それは、わかります。ただし、それを高密度で記録できなければ意味がないということですね。

雅人 その通り。そして、そのためには、磁気と電気を使うのが一般的なんだ。ところで、みんなはカセットテープのことは知っているね。録音や、映画などの録画にも使われていた。

和昌 もちろん、知っています。最近まで使われていましたからね。あれは、確か磁気テープでしたね。

### 2.7.1. 磁気記録

雅人 そうなんだ。磁性を利用した記録装置となる。磁場にはN極とS極があるが、実は、これら2極は分離できないことが知られている<sup>6</sup>。

---

<sup>6</sup> 理論物理の世界では、N極あるいはS極が単独に存在する単極子 (monopole) モノポールの存在が予言されている。

しのぶ N と S の 2 極からなる磁石をどんなに小さな磁石に分解しても、それは、必ず N と S の 2 極ある磁石にしかならないと習いました。

雅人 そこで、磁気記録の世界では、NS と SN という 2 種類の磁石を 1 と 0 に割り当てているんだ。

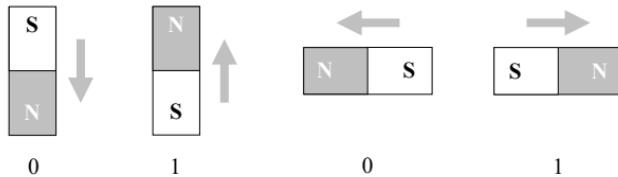


図 2-23 磁気記録では、磁石の向きの違いを、それぞれ 1, 0 に対応させる。図の配置は、それぞれ垂直磁気記録、水平磁気記録と呼ばれている。

しのぶ なるほど、このとき、図 2-23 のように、たて方向に磁石を並べる場合と横方向に並べる場合があるのですね。

雅人 それぞれ垂直磁気記録 (perpendicular magnetic recording) と水平磁気記録 (longitudinal magnetic recording) と呼ばれている。かつてカセットに使われていた磁気テープでは、水平磁気記録が使われていた。テープの構造としては、図 2-24 のようになっている。

しのぶ なるほど、磁性膜の部分を異なる方向に磁化することで、1 と 0 のデジタル信号を得ていたのですね。

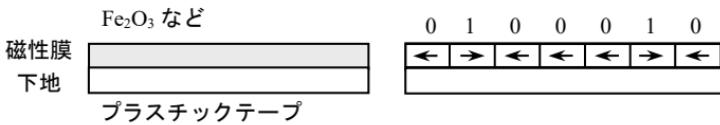


図 2-24 カセットテープの構造。プラスチックできた基板テープ上に磁性膜が塗布されている。磁性膜が磁化されると、その向きによって 1, 0 の信号を読み出すことができる。

雅人 そして、データの書き込みと、読み出しには、コイルを付した磁気ヘッドが用いられていたんだ。

信雄 なるほど、コイルに電流を流せば磁場が発生しますね。それを利用したのですね。

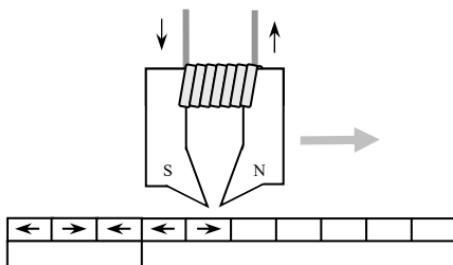


図 2-25 リング型磁気ヘッドにコイルを巻き、書き込む際には、コイルに電流を流してリングヘッドに強い磁場を発生させ記録する。コイルに流す電流を反転させれば、逆向きの磁場を記録することができる。

雅人 その通りだ。図 2-25 のようにコイルに流す電流を制御しながら、リングヘッドを磁気テープ上を走査していくば、磁化

としてデータを記録することができる。

しのぶ なるほど、簡単ですね。

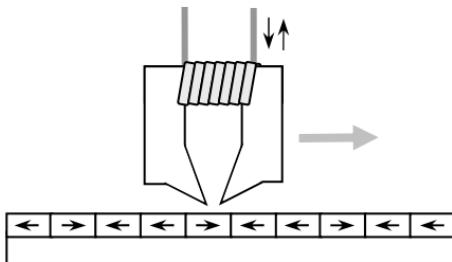


図 2-26 磁気データが磁化として記録された磁気テープ上を、磁気リングヘッドを走査すると、電磁誘導により磁化に対応した方向の電流がコイルに流れる。この電流方向によって、1, 0 の信号を取り出すことができる。

雅人 一方、図 2-26 のように磁気情報が記録された磁気テープ上を磁気ヘッドを走査すれば、電磁誘導によって、磁化に応じた電流が流れるので、今度は、電流の向きによって 1, 0 のデータを取り出すことができるんだ。

和昌 この方法は、実に簡単ですね。さらに、磁気メモリとしての汎用性もあります。

雅人 ただし、音声を録音していた初期のカセットテープは、デジタルではなく、音の高さや強弱を磁化の大きさや方向を変えて記録していて、実際にはアナログだったんだ。

結美子 そうだったんですか。

信雄 もちろん、この方式は、アナログだけではなく、1と0のデジタル記録にも対応可能ですよね。

雅人 もちろん。初期の頃は、テープレコーダーを使って、実際にカセットテープでプログラムの記録もしていた。実は、わたしが学生の頃に購入したシャープの MZ というコンピュータでは、カセットテープが標準メモリとして使われていたんだ。

和昌 いまでは、想像もつきませんね。

### 2.7.2. フロッピーディスク

雅人 ただし、カセットテープでは、プログラムが大きくなればテープを長くしないといけないし、読み取りにも時間がかかっていた。それで登場したのが、フロッピーディスク (floppy disk: FD) なんだ。

しのぶ フロッピーというのは、「柔らかい」という英語の “floppy” から付けられたと聞きました。

雅人 その通り。最初の FD は 8 インチ、つまり、約 20cm 程度の直径の円板だったんだが、実際に柔らかくて、くねくねしていたね。これは、円板上のプラスチック基板に磁性体を塗布したものだ。

結美子 テープが円板に変わったということですね。レコード

のようなものなのでしょうか。

雅人 まさにそうだね。だから、書き込みも読み出しも円板を回転させながら行っていたんだ。これならばカセットテープと違ってコンパクトになる。

信雄 そうは言っても、20 [cm] という直径は、いまから考えると大きいですね。容量はどの程度だったのでしょうか。

雅人 だいたい 200 [kB] から 1.4 [MB] 程度だったんだ。いまなら容量が小さいと感じるかもしれないが、当時としては画期的な技術だったんだ。

和昌 磁性体に記憶させる原理は同じなのでしょうか。

雅人 そうだね。原理的には、図 2-24, 25 と変わらない。記憶させる磁性体がテープ上に載っているか、円板上に載っているのかの違いだ。

しのぶ なるほど。わたしが知っている FD は、もっと小さくて硬かったです。

雅人 それは、3.5 インチのものだね。実は、8 インチの FD は 1980 年代後半まで使われていたんだ。そのつぎが 5.25 インチ、つまり約 13 [cm] のものだ。ここまでくねくねしていて、まさにフロッピーな円板だった。みんなが知っている 3.5 インチからは、プラスチック製の硬いケースに収められるようになった。

さらに、磁性面を接触から守るために、金属製のスライドカバーも付けられるようになったんだ。

**信雄** 記憶容量も大きくなったのでしょうか。

**雅人** いや、最初のものは 400 [kB] 程度だ。しかし、小さくてコンパクトなことが受けたんだ。実は、1984 年に Apple が、あの有名な Macintosh に搭載したのが最初だったんだが、あつという間に広まった。容量も最後は 2.88 [MB] の大きなものまで登場した。

**結美子** FD は、いまは、記憶媒体として、ほとんど見かけませんが、どうしたのでしょうか。そう言えば、研究室の棚に先輩たちの残したデータが FD に収納されていましたね。

### 2.7.3. 光磁気ディスク

**雅人** それは、信頼性と容量の問題だね。記憶容量を上げるためにには、記憶領域の密度を上げる必要がある。さらに、信頼性も高めるとしたら、ナノテクノロジーが必要になる。そこで、登場したのが、光磁気ディスク (magneto-optical disk: MO) だ。一般には MO ディスクと呼ばれている。

**しのぶ** MO ディスクならば聞いたことがあります、あまり普及しませんでしたね。

**雅人** そうなんだ。FD よりも大容量をうたっていたのだが、それほど普及しなかったよね。

和昌 どのような原理で大容量化が図られたのでしょうか。

雅人 それは、二つの要因がある。まず、図 2-25 の方法で磁性膜を磁化するときに、図 2-27 のように、必ず磁場の拡がりが生じるんだ。このため、1, 0 のひとつの単位を狭くすることには限界がある。

しのぶ 磁力線は互いに反発するので、自由空間では拡がる性質があると聞きました。だから、極の大きさを絞っても、その間では磁場は拡がるのですね。

雅人 まさに、その通り。物理の基本だからどうしようもない。

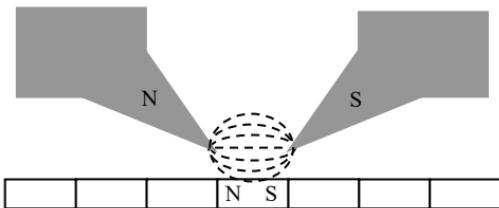


図 2-27 磁気ヘッドの先端で発生する磁場は拡がる。このため、磁化を記録する領域の微細化は限界がある。

結美子 それでは、磁化されている単位の大きさを小さくするのは難しいですね。

雅人 そこで、登場するのがレーザー光 (laser light) だ。この光は、拡がりを絞ることができる。そして、絞ったレーザー光を磁性

膜に照射するんだ。

和昌 それで、なにが起こるのでしょうか。

雅人 レーザーが照射された微小領域が局所的に加熱される。すると、磁性体のキュリーポイント (curie point) よりも温度が上昇する。キュリーポイント以上の温度になると、その部分は強磁性ではなくなるんだ。そして、その領域が冷えるときに、まわりの磁場の影響で、逆方向に磁化される。つまり、もとの生地が 1 とすれば、0 のところだけ照射すればよいことになる。

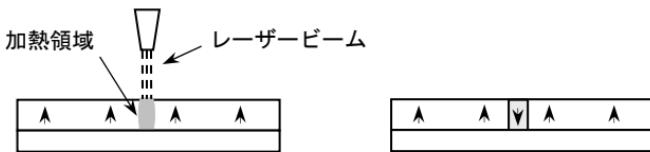


図 2-28 均一方向に磁化された磁性膜にレーザーを照射すると、磁化が反転し、微小領域に磁気記録することが可能となる。

結美子 なるほど、レーザー加熱した領域がデジタル信号の最小単位になるのですね。そして、その密度は磁化の場合よりもはるかに小さい。だから、記憶容量を上げることができるのでですね。

雅人 その通り。

信雄 容量は、どの程度増えたのでしょうか。

雅人 最初は、128[MB]程度だったかな。それでも衝撃的だったんだ。それが、256[MB], 640[MB]と増えていって、最後は、2.3[GB]となった。

しのぶ それは、FD に比べればすごい容量アップですね。でも、いまはほとんどみないです。

雅人 そうだね。2000 年以降は、市場から消えてしまった。それは、CD や DVD が登場したからなんだ。さらに、USB 端子で使えるフラッシュメモリの登場も大きかったね。

#### 2.7.4. 光ディスク

和昌 CD (compact disk) はコンピュータのメモリというよりは、レコードの代替としてのミュージック CD 販売がメインでしたよね。

雅人 その通りだ。レコードは溝に音楽情報が入って、レコード針でその上をなぞると音を再現するという仕組みだ。溝にある凹凸が音を再現してくれる。一度、型をつくれば、それに流し込むことで、いくらでも同じレコードを製造できる。

信雄 CD の原理もレコードに似ているのでしょうか。

雅人 そうだね。ただし、こちらの凹凸は 1 と 0 の 2 種類で、デジタルデータとして音を記録していた。図 2-29 に模式図を示している。

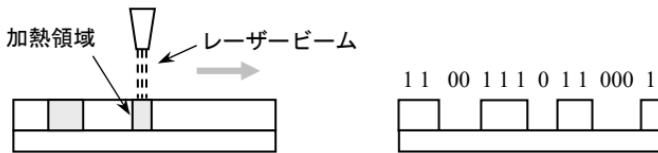


図 2-29 CD の表面にレーザー光線を当てたのち、表面をエッチングすることで 1, 0 に対応した凹凸を形成する。

和昌 なるほど、この図でいけば 0 に対応した領域にレーザー光線を当てて、その後、エッチングで取り除けば、1, 0 に対応した凹凸ができますね。

雅人 ただし、実際には、これは型 (mold) になる。販売用の CD は、この型にプラスチックを流し込んで製品化している。型さえあれば、いくらでも製造が可能となる。

結美子 なるほど、そこはレコードと同じですね。信号を読み込むときには、どうするのでしょうか。

雅人 それにも、レーザー光を利用する。CD の表面に光をあてれば凹凸に対応した 0, 1 のデジタル信号を取り出すことができて、音楽などを再生できる。だから、光ディスクと呼んでいるんだ。今までのような磁性を利用していいのが特長だ。

しのぶ なるほど。確かに、基本はレコードと同じ原理ですね。レコード針がレーザー光に変わったようなものです。

**信雄** とすれば、CD は読み取り専用の、つまり ROM (read only memory) の機能しかないということになりますね。

**雅人** 基本的には、そうなるね。これを CD-ROM と呼ぶこともある。ただし、書き込み可能な CD も売り出されたことがある。記録可能という英語の recordable にちなんで、CD-R と呼ばれている。

**結美子** いったい、どうやって書き込み可能としているのでしょうか。

**雅人** 特殊な素材を使って、レーザー光が当たれば膨張するようしているんだ。すると、凹凸をつくることができる。

**和昌** それでは、一度記録したデータを消すことはできませんね。

**雅人** その通りなんだ。光ディスク上の凹凸は消せないから、データそのものを消すことはできない。そこで、消したい情報は、読み込みができないように設定しているだけなんだ。さらに、CD-R には、凹凸のない未使用の面があるので、新しいデータは、そこに書き込んでいくんだ。

**信雄** とすると、円板のすべての面を書き込み用に使ったら、それ以上は情報を記録できないということですね。

**雅人** その通りだ。

しのぶ ところで、CD の仲間に DVD がありますね。CD よりも容量が大きいので、CD は音楽、DVD は動画などのビデオ録画ができるという印象でした。

結美子 確か、DVD は “digital video disk” の略ではなかったでしょうか。V は “video” つまりビデオのことかと思います。

雅人 そう誤解しているひとも多いね。実際には、DVD は “digital versatile disk” の略なんだ。“versatile” とは聞きなれない単語かもしれないが、「多目的用」という意味になる。

しのぶ それは知りませんでした。それでも、基本は CD と同じ光ディスクで容量が大きくなっただけなのではないでしょうか。

雅人 まさに、その通りなんだ。CD の容量は 700[MB] 程度だが、DVD の容量は 4.7[GB] となっている。そのおかげで映画を記録することができる。結美子さんが V を “video” と誤解しているのも無理はない。

信雄 容量が大きいということは、書き込みの際のレーザービームを絞りこんでいるということですか。

雅人 まさに、その通り。レーザー波長を 760[nm] から 650[nm] まで絞っている。その結果、書き込みできるトラック間の距離が 1.6[μm] から 0.74[μm] まで小さくなるんだ。

和昌 とすれば、CD-R と同じように、DVD-R もあるのですね。

原理は、CD-R と同じように、まだデータのない未使用面に、レーザー光線でデータを書き込んでいく方式でしょうか。

雅人 そうだね。そして、さらに容量の大きい Blu-ray disk も CD や DVD の仲間だ。一般のひとはブルーレイと呼んでいるね。

結美子 そうなんですか。なぜ、ブルーレイと呼ぶのでしょうか。青い光ですよね。

雅人 CD や DVD に使われているのは赤色レーザーだが、Blu-ray に使われているのは青紫色のレーザーで、波長は 405 [nm] とさらに短い。そして、書き込み可能なトラック間の距離は、0.32 [ $\mu\text{m}$ ] まで狭くできる。それだけ、何重にもわたってデータを書き込めることになる。その結果、メモリサイズは 25 [GB] まで増えるんだ。

しのぶ それは、すごいですね。あらためて、ブルーレイの威力を知りました。でも、CD も DVD も Blu-ray もすべて光ディスクで、レーザービームがどれだけ絞れるかで容量が違うということなのですね。

雅人 まさに、その通りだ。

### 2.7.5. ハードディスク

結美子 ところで、ハードディスクは PC の内臓メモリとして使われていますし、いまでは、当たり前のように、外付けハードディスクが販売されていますが、その原理はどのようなものな

のでしょうか。

雅人 ハードディスクは HDD とも呼ばれるね。“hard disk drive”の略だ。そして、磁性を利用した記憶装置なので、その書き込みと読み出しの原理は、図 2-25 や図 2-26 と同じになる。

信雄 それでは、図 2-27 のような磁場の拡がりの問題は避けられないで、容量を大きくするのは、難しいのではないでしょうか。

雅人 そうだね。だから、HDD では、磁気ヘッドと磁気ディスクの間の距離を極限まで小さくしているんだ。面間距離が小さければ、磁場の拡がりを抑えることが可能となる。その結果、書き込めるトラック幅が小さくなり、容量が大きくなる。

しのぶ どの程度なのでしょうか。

雅人  $10[\text{nm}] (10^{-9}\text{m})$  以下と言われている<sup>7</sup>。10 万分の 1mm 以下の大きさだ。ウィルスよりも小さい隙間だ。

結美子 想像もつきませんが、そんな小さい隙間をどうやってコントロールできているのでしょうか。

雅人 もちろん、こんな小さな距離を制御することは難しい。実際には、磁気ディスクを高速で回転させ、空気力学、すなわ

---

<sup>7</sup> nm は nanometer (ナノメートル) のことで、nano は  $10^{-9}$  の接頭辞である。

ち空気が隙間に入る効果を利用して距離を保っているんだ。さらに、回転速度は 5000 から 10000 [rpm] 程度だ。rpm は “revolutions per minute” の略で、1 分間の回転数だ。ディスクの外周でみれば、時速 100 から 200 [km] と言われている。これだけ大きな回転速度なので、読み込みも速い。

和昌 なるほど、回転数が大きいほど、プログラムの立ち上がりも速くなるのですね。最初の待ち時間はいらいらしますからね。しかし、あの小さな HDD の容器の中で、円板がものすごいスピードで回転しているのは驚きです。

雅人 さらに容量を上げるために、図 2-30 に示すように、4 枚の磁気ディスクと 8 個の磁気ヘッドを使っている。しかも、ディスクの両面に情報が記録されているんだ。

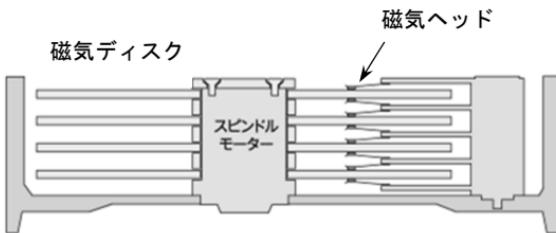


図 2-30 ハードディスク (HDD) 内部の構造

信雄 外からは見えませんが、内部はこうなっているのですね。そう言えば、HDD を起動したときに音が聞こえるのは、ディスクの回転音なのですね。

雅人 回転に使うのはスピンドルモータ (spindle motor) と呼ばれて、高速回転用のモータなんだ。工作機械に使われていたのを HDD に応用しているんだ。

結美子 そう言えば、最近 HDD ではなく SSD が脚光を浴びていますね。

雅人 SSD とは“solid state drive”的略で、フラッシュメモリ (flash memory) を使った記憶装置のことなんだ。HDD と違って回転しないので音も出ない。動作も速いので、いま、かなりの注目を集めている。

#### 2.7.6. フラッシュメモリ

しのぶ フラッシュメモリとは USB (universal serial bus) メモリのことでしょうか<sup>8</sup>。



雅人 そうだね。USB 端子に差し込むことで動作するので、そう呼ばれている。その手軽さと容量の大きさから、いまでは、かなり浸透しているよね。

和昌 そのメモリ機構はどうなっているのでしょうか。

---

<sup>8</sup> USB はパソコンと周辺機器を結ぶインターフェース規格のひとつである。キーボード、マウス、モデムなどの接続に用いられる。1990 年代後半から、シリアルポートに替わって広く普及するようになった。

**雅人** フラッシュメモリでは、磁性ではなく、電荷つまり電子のあるなしを利用しているんだ。だから一瞬にしてメモリを消すことができる。このため、“flash”という名がついている。フラッシュカメラと同じ語源だね。

**信雄** 電子があるかないかで 1 と 0 を表現しているのですね。それは、コンデンサのようなものなのでしょうか。

**雅人** コンデンサとは少し違うんだ。前に MOSFET を紹介したね。このトランジスタでは、ベースのところがゲートになっていたね。

**結美子** はい、絶縁されているのでしたね。これをうまく利用してスイッチングに利用しているのでしたね。

**雅人** フラッシュメモリでは、この絶縁層を通して電子がシリコン基板からゲートに移動して、ゲートに電子が貯まることができる。これがオン状態となり、1 とすることができます。

**しのぶ** 絶縁層を通して、電子が基板からゲートに移動できるのですか。

**雅人** これをトンネル効果と呼んでいるが、実際には、20 [V] の電圧をかけるので、薄い絶縁層であれば、電子が、それを飛び越えるんだ。静電気も、本来は絶縁層である空気を通って飛んでくるよね。

和昌 冬にドアを開けようとしたときバチっと来る電気ですね。

雅人 まあ、絶縁層も薄いし、ごく微小な世界の現象なので、雷ほど激しいものではないが、イメージとしては、それでいいと思うよ。図2-31にフラッシュメモリにおける1と0のイメージを示している。

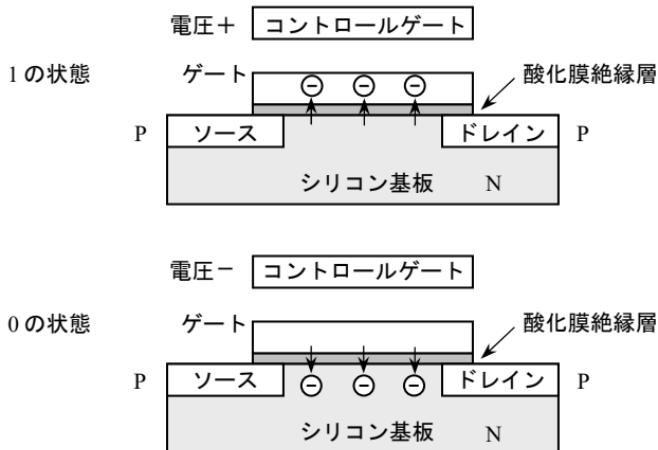


図2-31 フラッシュメモリにおける1と0の状態

信雄 ゲートの電荷を消すためには、逆方向に電圧をかけねばよいのですね。

雅人 その通り。だから、電圧の切り替えだけで1,0を操れる。ちなみに、ゲートはフローティングゲート (floating gate) と正式には呼ばれている。まさに、絶縁層を介して、浮いているようなイメージだね。

結美子 フラッシュメモリの利点は何なのでしょうか。

雅人 電気的な信号で制御できるので、応答が速いんだ。しかも、電気的なメモリの場合、電源を切ればメモリも消えるが、フラッシュメモリでは、電源を切っても電荷は消えないので、不揮発性メモリとなる。

信雄 なるほど。それが利点なのですね。

雅人 さらに、フラッシュメモリは電気的に制御できるので、磁気メモリの HDD よりも電気回路と親和性が良いんだ。いまの電気回路はトランジスタなどが集積したものとなっている。フラッシュメモリは、集積回路と同様につくりこむことができる。しかも HDD では、メモリを読み込むためには高速回転が必要だったが、フラッシュメモリでは必要がない。それが SSD だ。このため、PCに搭載されていた HDD を SSD に変える動きもある。

しのぶ それでは、すべて SSD に移行するのでしょうか。

雅人 いや、そんなことはない。メモリには、それぞれ利点もあれば欠点もある。逆に多様性があるからこそ、今後の進展も望めるんだ。たとえば、いまの HDD では空気が入っているが、これをヘリウム (helium: He) 封入型にする試みがある。空気抵抗が激減して、より高速でディスクを回転できるようになる。

結美子 なるほど。メモリの世界は奥が深いですね。

雅人 その通りと思うよ。それでは、つぎに、コンピュータが、

## 第2章 ハードウェア

どのようにして、複雑な計算をしているのかを見ていきたいと思う。