

はじめに

本書の前著である「マイクロコンピュータ技術入門」の初版から18年が経過した。卒業して社会に出てから本書の初版本に再度目を通した、などのうれしい声を聞かせてもらったこともあったが、この18年間の技術進歩は大きく、また初版には未熟で至らない説明も多々あることに気づき永らく気になっていた。

この度、共著者を迎え、共著者とともに初版における説明の未熟な点を直し、また現在の学習環境に合わせて内容も構成も少し変えて、新編として再発行することとなった。

コンピュータは人に代わって思考できるところが、それまでの道具とは異なる。コンピュータは現在、将棋のプロ棋士に勝ったり、地球規模で気象予測ができるまでになった。またコンピュータは、10万馬力もの大形ジェット機の頭脳となったり、日常の身の周りにおいては、携帯電話、炊飯器、テレビ、自動車など多くのものに組み込まれている。現代はコンピュータ文明と言ってよい。コンピュータは人間を超える思考力を発揮したり、人間の代わりに思考したりするようになったのである。今や人間がコンピュータや機械に勝てるのは、夢をみる（ビジョンをもつ）こと、祈る（思念する）ことだけになったと思われる。

本書は、コンピュータがいったいどのような仕組みで思考することができるのかをわかりやすく説明することを目的としている。

コンピュータの思考は、論理、算術、記憶を基にして行われる。論理は、3つの基本論理：AND、OR、NOTのどれか、またはそれらの組み合わせで行われ、算術は、これらの論理回路の組み合わせで行われ、また記憶も、SRAMと呼ばれる記憶回路では算術回路同様にこれらの組み合わせを基にして行われる。これら論理、算術、記憶を行う電子回路はIC化されている。コンピュータはこのような電子回路を基にした思考できる機械である。

この機械が思考するとき、論理、算術、記憶の操作命令とそれらが扱うデータ、すなわち情報は電気信号となって回路を伝わる。第1部では、これら情報（おもにデータ）がどのようにコンピュータの言葉（2進数）に置き換えられるのかがまず説明され、次に、論理、算術、記憶がどのような回路で行われるのかが説明されている。ただし、第1部の記憶回路に関しては論理回路を基にしたSRAMだけが取り上げられている。それは、論理回路で記憶できるということは、電荷の蓄積などを基に記憶できる、という原理よりも理解しにくく不思議で興味深いと思えるからである。

第1部で説明されたところの各種データについてとそれら进行处理する回路をふまえて、第2部ではプログラムとプログラム処理の仕組みについて説明される。ここでの各種回路や装置の中身はブラックボックスとして扱われている。プログラムとデータはソフトウェアとよばれ、プログラム処理を行うための各種装置はハードウェアとよばれ、これらは車の両輪のようなものなので最初はほぼ同時進行で説明されるが、その後は順に、マイクロプロセッサのハードウェア、プログラム、具体的マイクロコンピュータと説明されている。最後の10章の具体的マイクロコンピュータにおいては、現在、機器への組込みでよく実用され、また、教育現場でもよく使われているマイクロコントローラ（シングルチップコンピュータ）PICの具体的な仕組みが説明されている。

本書は、高専3年生に行った講義経験を基に、高専3年生または大学1年生あたりの、コンピュータの専門予備知識をほとんど持たないで学習を始めた人を対象に書かれている。

コロナ社には原稿の隅から隅まで目を通していただき多くのご指摘を受け、また著者の度重なる追加や訂正にも付き合ってくださいました。この場を借りて感謝致します。

2014年12月

松 田 忠 重

目 次

第1部 デジタル技術の基礎

1 デジタルコードとビット

1.1	デジタルコード	1
1.2	自然数と正の有理数の2進コード	3
1.2.1	アラビア数字による自然数の各種コード	3
1.2.2	正の有理数の2進コード	5
1.2.3	自然数の各種コードの基数変換	6
1.3	ビット	8
1.4	エンコーダ, デコーダ	13

2 文字, 画素のデジタルコード

2.1	文字, 数字, その他の記号のデジタルコード	18
2.2	画素とそのデジタルコード	21
	1~2章の演習問題	25

3 いろいろな数の2進コード

3.1	ストレートバイナリ, オフセットバイナリ, 2の補数バイナリ	26
3.2	2の補数2進数, 1の補数2進数	29

3.3 固定小数点2進数	33
3.4 2進数による浮動小数点数	35
3.5 2進化10進数	40
3章の演習問題	41

4 A-D変換, D-A変換

4.1 量子化	43
4.1.1 量子化誤差と分解能	43
4.1.2 量子化ノイズとSN比	46
4.2 正負電圧の量子化	48
4.3 標本化定理	49
4.4 D-A変換	53
4.5 アナログとデジタルの比較	55
4章の演習問題	55

5 基本論理回路

5.1 AND, OR, NOT	57
5.2 正論理, 負論理	62
5.3 論理回路の入出力回路と信号	66

6 加算, 記憶, その他の代表的回路

6.1 加算回路	74
6.2 記憶回路	75
6.3 その他の代表的論理回路	79
5~6章の演習問題	81

第2部 マイクロコンピュータ

7 コンピュータの構成と働きの概説

7.1 プログラムとプロセッサ	83
7.1.1 データと命令とプログラム	83
7.1.2 内部メモリとプロセッサ	85
7.1.3 プログラム処理	87
7.1.4 リセット, 割込み	89
7.2 ハードウェア基本構成	90
7.3 ソフトウェア基本構成	96
7.3.1 プログラムとデータ	96
7.3.2 アプリケーションとオーエス	100
7.4 ハーバード・アーキテクチャ	102
7.5 並列処理	104
7.6 マイクロコントローラ	106
7章の演習問題	107

8 マイクロプロセッサのハードウェア

8.1 基本構成	108
8.1.1 クロック発生器	108
8.1.2 バス制御部	109
8.1.3 命令解読部	109
8.1.4 算術論理演算装置	109
8.1.5 レジスタ部	109
8.1.6 バスインタフェース部	110
8.1.7 キャッシュ	110
8.2 各種バス	111
8.2.1 アドレスバス	112
8.2.2 データバス	112
8.2.3 制御バス	113
8.3 各種レジスタ	117

8.3.1 汎用レジスタ	118	8.3.2 専用レジスタ	119
8章の演習問題	126		

9 命令セットとプログラム

9.1 命令セットとアドレッシング	127		
9.2 アセンブリ言語	133		
9.2.1 はじめに	133	9.2.2 語彙	134
9.2.3 構文	137	9.2.4 擬似命令	137
9.2.5 オペコード, オペランドをアセンブリ言語で	141		
9.3 アセンブリ言語でのプログラム構成	142		
9.3.1 はじめに	142	9.3.2 定義文	143
9.3.3 メインルーチン	144	9.3.4 サブルーチン	146
9.3.5 割込みルーチン	147		
9章の演習問題	148		

10 PIC

10.1 PIC 中間性能グループの大まかな特徴	150		
10.2 ハードウェア構成概要	151		
10.2.1 プロセッサ	153	10.2.2 プログラムメモリ	156
10.2.3 ファイルレジスタ	156	10.2.4 各種周辺装置	157
10.3 いくつかの専用レジスタ	159		
10.4 命令セット	169		
10章の演習問題	178		
演習問題の解答	181		
索引	189		

第1部

デジタル技術の基礎

情報はデータ（物事、知識）とそれら进行处理する命令に分けることができる。第1部では、各種データのデジタルコード（数で表したもの）の説明から始める。次に、それらを取り扱う論理・算術・記憶などの回路について説明する。

コンピュータにはデジタル式とアナログ式があるが、本書ではデジタル式だけを扱うので、以下、デジタルコンピュータのことを単にコンピュータと記す。

1 デジタルコードとビット

情報の2進コード化（2進符号化）、およびビットについて説明する。

1.1 デジタルコード

物事は、その物事とは異なる別の抽象物（点と棒、文字、数字など）で表現することができる。それらの抽象物を体系化したものをコード（code）という。コードは**符号**と翻訳されているが、符号という言葉は単なる記号を表す mark や sign にも使われている。また実際コンピュータ工学では符号は sign にも使われており、以下それらの意味ではないことをはっきりさせるため、コードには符号の代りにコードという言葉そのものを使うようにする。

コード例には以下のようなものがある。

- (1) 文字、数字、その他の記号は、物事を表すためのコードといえる。
- (2) 音波をマイクロホンで電圧にした電圧波形も、その音を表すためのコー

表 1.1 モールスコード例

A	• —
B	— •••
C	— • — •
⋮	⋮

ドといえる。

(3) 表 1.1 のように、文字、数字、その他の記号を点と棒で表現したものはモールスコードと呼ばれる。

(4) 文字、数字、その他の記号を、太さの異なる線によって表現したコードはバーコードと呼ばれ、2次元にドットを配置してつくられたコードはQRコードと呼ばれる。

コードにはアナログ式とデジタル式がある。

アナログ (analog) は「相似の」という意味である。漢字のようにものの名称をその形をまねて表したり、あるいは音波を波形で表したり、などしてできたコードがアナログ式コードである。上記(1)(ただし数字を除く)～(4)は、情報を絵模様で表しているのでアナログ式コードといえる。

デジタル (digital) は、「数字を用いた」という意味で使われる。digit は「数字」のことである。もともとは、数量を計算したり順番を付けたりするとき「指を使うことによる」という形容詞の意味である。

アラビア数字や漢数字などは、もともとは大きさや量を表すためのものである。デジタル技術においては、電圧などのように大きさや量に関係あるものだけでなく、文字や色や論理条件の状態(真, 偽)などのように大きさや量に関係ないものも、それぞれある約束で「数字」に置き換えられて表される。

「数字」で置き換えられたコードは**デジタルコード** (digital code) と呼ばれ、そのときの数字には10進数, 2進数, 16進数がよく用いられる。コードは用途によって変換されて使い分けられる。人間にはキーボードに並んでいるような文字, 数字, 記号が使われるが, 商品ラベル読取り器などには, 高速で間違いなく行えることから, バーコードやQRコードが広く使われている。

コンピュータが情報処理をするときデータ(処理対象情報)はデジタルコードで, これらはコンピュータ内部ではすべて0,1からなる2進コードである。また, コンピュータが情報処理をするためのプログラム(手続き, あるいは命令が順に並んだもの)も, **マシンコード** (machine code) と呼ばれる2進コードである。これは, 通常まず人間に分かりやすい文字や記号などを使ったコン

コンピュータ用**プログラミング言語** (programming language) (**コンピュータ言語**ともいう) で書かれる。書かれたものは**ソースコード** (source code) と呼ばれる。次にこれが変換されてマシンコードがつくられる。

1.2 自然数と正の有理数の2進コード

数値計算に使われる整数や浮動小数点数を表すコードについては後回しにして、ここでは0を含めた自然数のコードの体系：数字とその使い方、の基礎について述べる。

1.2.1 アラビア数字による自然数の各種コード

数とは、民族に無関係な普遍的なもので、自然数、有理数、無理数、整数、また近代では複素数も含めたものことである。数を表すコードには、アラビア数字、漢数字、ローマ数字などがあるが、通常はアラビア数字が使われる。

< 例 > アラビア数字による10進数によるコード：2014

これは、漢数字では：二千十四 (貳阡拾四) となる。

アラビア数字には**ゼロ**のコード0があるが、漢数字にはない[†]。

数字の「位置」は**桁** (place, または position) と呼ばれる。アラビア数字では0を使うことですべての数字に**順番に桁**ができる。このとき桁はまた、並んだ数字の個数の単位にも使われる。例えば2014は4桁の数である、のようにも使われる。

位置としての桁はそれぞれ重みをもっていて、桁に置かれた数にその重みが乗ぜられてその桁の数が表す値になる。そのとき乗ぜられる数は**位** (place value) と呼ばれる。例えば10進数2014では、位は最下位から上位向きに 10^0 , 10^1 , 10^2 , ... となり、次のように桁ごとの和に分解できる。

$$2014 = 2 \cdot 10^3 + 0 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0$$

10^0 , 10^1 , 10^2 , ... は、**底** (base) が10で、**指数** (exponent) は最下位の桁

[†] 日本の数え年年齢では生まれた瞬間に1歳。家の2階は米語では the second floor で日本語と同じ、しかし英語では the first floor。

から上位向きに 0, 1, 2, …になっている。またついでに書くと、小数点付き 10 進数の場合では、小数点の右側の桁から順に右向きに 10^{-1} , 10^{-2} , …となる。

アラビア数字ではすべての数字の桁が分かるのでそれらの数字の位も分かり、特に漢数字で使う十、百、千などを数字に付ける必要がない。

このような位の「底 10」の 10 のことをコンピュータ用語では**基数** (radix, または base) と呼び、「**基数 10 (base-10)**」のコードのことを「**10 進数** (decimal number)」と呼んでいる。アラビア数字の 10 進数では一つの桁に置ける数は、0, 1, 2, …, 8, 9 の「10 種類」あり、この数は基数に一致している。

基数 2 (base-2) のコードは「**2 進数** (binary number)」と呼ばれる。自然数の場合、基数 2 の数の桁の位は右端から左向きに 2^0 , 2^1 , 2^2 , …を表し、それぞれ順に 2^0 の位, 2^1 の位, 2^2 の位, …と呼ばれる。それらの桁は**ビット** 0 の桁, ビット 1 の桁, ビット 2 の桁, …と呼ばれる。基数 2 の数字にはアラビア数字 0, 1 の 2 種類の数字が使われる。またついでに書くと、小数点付き 2 進数では、小数点の右側の桁から右向きに、 2^{-1} , 2^{-2} , …を表し、それぞれ順に 2^{-1} の位, 2^{-2} の位, …と呼ばれる。

漢数字の場合の数字には、十、百、千など位の名前を付けることが必要である。漢数字の場合、基数を変えたら位の名前が新たに必要になるが、アラビア数字では 0 のおかげで数字に位の名前を付ける必要がないので、基数を変えて数を表すことも容易である。

基数 16 (base-16) のコードは「**16 進数** (hexadecimal number)」と呼ばれる。自然数を表す 16 進数では、桁の位は右端から左向きに 16^0 , 16^1 , 16^2 , …を表し、それぞれ順に 16^0 の位, 16^1 の位, 16^2 の位, …と呼ばれる。なお、16 進数では小数点付きは使われていないようである。16 進数ではアラビア数字と英文字 0, 1, 2, …, 8, 9, A, B, C, D, E, F の 16 種類が使われる。

明らかに何進数か分かる場合は別にして、分かりにくい場合には数の後に添字、2 進数であれば b, B あるいは 2 を、8 進数 (octal number) であれば o, O あるいは 8, 10 進数であれば d, D あるいは 10 を、16 進数であれば h, H あるいは 16 を付ける。例えば 10 進数の 10 は、10 進数では 10_{10} , 2 進数では

1010_b, 16進数ではA_hのように表される。

本書でよく使うのは2進数, 10進数, 16進数である。表1.2は自然数(ゼロを含む自然数)の一部分の10進コード, 2進コード, 16進コードによる桁上がりの様子を示すものである。

表1.2 ゼロを含む自然数の一部の3種類のコード

10進コード	2進コード	16進コード
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
⋮	⋮	⋮
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	1 0000	10
17	1 0001	11
18	1 0010	12
⋮	⋮	⋮
255	1111 1111	FF

1.2.2 正の有理数の2進コード

10進数では10倍すると最下位に0が付いて全体が左にずれるように, 2進数では2倍すると同様のことが, 16進数では16倍すると同様のことが起こる。例えば2進数では図1.1のようになる。これから2進数同士の掛け算は, 例えば図1.2のようになることがわかる。

10進数では10分の1すると全体が最下位方向に1桁下がるように, 2進数では2分の1すると同様のことが, 16進数では16分の1すると同様のことが起こる。例えば2進数では, 図1.3のようになる。

$$\begin{array}{ccc} 1_b & \xrightarrow{\times 2} & 10_b \\ & & 2 \end{array} \quad \begin{array}{ccc} & \xrightarrow{\times 2} & 100_b \\ & & 4 \end{array}$$

$$\begin{array}{ccc} 11_b & \xrightarrow{\times 2} & 110_b \\ 3 & & 6 \end{array} \quad \begin{array}{ccc} & \xrightarrow{\times 2} & 1100_b \\ & & 12_d \end{array}$$

図1.1 2進数では2倍すると最下位に0が付き1桁左向きにずれる

$$\begin{array}{r} 110_b \\ \times 11_b \\ \hline 110 \\ 110 \\ \hline 10010_b \end{array}$$

図1.2 2進数での掛け算例

また2進数同士の割り算は, 例えば図1.4のようになることが分かる。図1.4では8/3の2進数による割り算である。小数点以下3桁までなら, 割られ

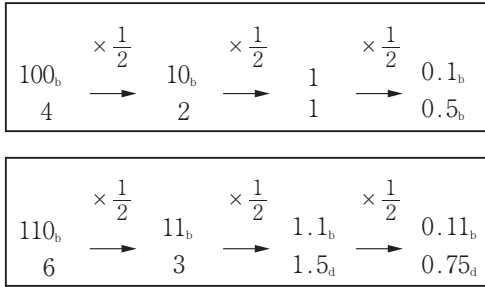


図 1.3 2進数では1/2すると1桁右向きにずれる

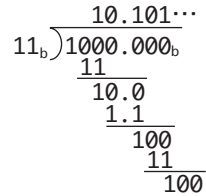


図 1.4 2進数同士の割り算例

る数を 2^3 倍して下桁に 0 を 3 個付けて 11_b で割り算を行い、後で $1/2^3$ すると考えればよい。

1.2.3 自然数の各種コードの基数変換

例えば 4 桁自然数の 10 進数は、次のように位ごとの和に分解できる。

$$(a_3a_2a_1a_0)_d = (a_3 \cdot 10^3 + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0)_d \quad a_n = 0 \sim 9$$

同様に 4 桁の基数 2 の自然数（絶対値型 2 進数）では、次のように位ごとの和で基数 10 のコード（10 進数）に変換できる[†]。

< 絶対値型 bin → dec >

$$(a_3a_2a_1a_0)_b = (a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0)_d \quad a_n = 0 \text{ or } 1$$

例えば次のようになる。

$$1000 \ 0010_b = (2^7 + 2^1)_d = 130_d$$

$$0111 \ 1110_b = (2^7 - 2^1)_d = 126_d$$

< 絶対値型 bin → hex >

16 進数では、4 桁の 2 進数 0000_b から 1111_b まだが 0, 1, 2, ..., 7, 8, 9, A, B, C, D, E, F に対応している。この性質を使えば基数 2 から基数 16 への変換ができる。例えば次のようになる。

$$1000 \ 0010_b = 82_h$$

$$0111 \ 1110_b = 7E_h$$

[†] 電卓では、モードのメニューの中で基数変換ができる。ただし、電卓による基数変換では整数型数を対象とし絶対値型数ではないことに注意（1.3 節参照）。

索 引

【あ】	
アキュムレータ	118, 131
アセンブリ言語	133
後入れ先出し	125
アドレス空間	92
アドレスバス	87, 112
アドレッシング	127
アプリケーションソフト	
ウェア	100
【い】	
色深度	21
インデックスレジスタ	123
【う】	
打ち切り誤差	12
【え】	
エイリアシング誤差	49
エンコーダ	13
【お】	
オーエス	100
オーバーフロー	28
オーバーフローフラグ	119
オフセットバイナリ	27
オペコード	127
オペランド	127
オペレーションコード	127
オペレーティングシステム	
	100
【か】	
解像度	21

仮数	35
画素	21
【き】	
偽	57
記憶容量	11
擬似命令	135
基数	4
キャッシュ	111
キャラクタコード	18
キャリアー	74
キャリアーフラグ	120
【く】	
組合せ論理	75
組込み	106
位	3
クロック	91
【け】	
桁	3
ゲート	61
【こ】	
語	13, 85
固定小数点数	33
コード	1
【さ】	
サブルーチン	97, 122
算術論理演算	84
【し】	
識別子	96
思考	84

指数	3, 35
システム制御	84
実行する	88
周辺装置	157
16進数	4
主記憶装置	93
10進数	4
順序論理	75
真	57
シングルチップマイクロ	
コンピュータ	106
シンボル	136
真理値表	59
【す】	
スタック	87
スタックポインタ	123
ステータスレジスタ	119, 159
ストレートバイナリ	27
3-state	67
【せ】	
制御バス	87, 113
整数	27
ゼロフラグ	119
専用レジスタ	119
【そ】	
ソフトウェア	83, 85, 96, 100
【た】	
代入演算子	118
タイムシェアリング	99
立上り時間	71
立下り時間	71

単精度浮動小数点数	37	ハーバード・アーキテクチャ	102	ポップ	124
【て】		ハーフキャリーフラグ	120	ポート	158
底	3, 35	半加算器	74	【ま】	
デジタル	2	バンク	157	マイクロコントローラ	106
デコーダ	14	汎用レジスタ	118	マイクロプロセッサ	90, 108
データ	83	【ひ】		マシン語	85
データ転送	84	比較	175	マスク	116, 148
データバス	87, 112	ピクセル	21	マルチコア	104
伝搬遅延時間	71	ビット	8, 10	マルチタスキング	99
【と】		標本化定理	51	丸め	12, 35, 45
トゥルーカラー	23	【ふ】		【め】	
【な】		ファイルレジスタ	156	命令	84
ナイキスト周波数	51	フェッチ	88	命令サイクル	155
内部記憶装置	86	復号	14	命令セット	85, 127, 169
7セグメントLED	16	符号	1	メインルーチン	99, 144
【に】		符号ビット	27, 32	メカトロニクス	106
2進化10進数	40	プッシュ	123	メモリ空間	92
2進数	4	ブート	101	【も】	
2の補数バイナリ	27	浮動小数点数	35	文字コード	18
ニーモニック	134	ブートローダ	94, 101	【ゆ】	
入出力装置	95	フラグ	119, 147	有効数字	12
【ね・の】		フルカラー	23	【よ】	
ネステイング	100	プログラム	85	呼出し	97
ノイズ	72	プロセッサ	86	読出し / 書込み	113
ノイズマージン	70	分解能	45	【ら】	
ノイマンアーキテクチャ	83	分岐	84, 97	ラベル	136
【は】		【へ】		【り】	
倍精度浮動小数点数	37	並行処理	99, 104	リアルタイム	98
バイト	13	ページ	156	リセット	89, 115
パイプライン	104	ヘッダファイル	144	リセットベクトル	89, 115, 122
バス	11, 86, 92	ベン図	58	リターン	97, 98
バス制御部	109	変数	118	量子化	43
発熱	73	【ほ】		量子化誤差	45
ハードウェア	83, 85, 90, 108	ポインタ	123, 129, 163		
		暴走	100		
		補数	30		

	【わ】	割込み 割込みフラグ	89 116	割込みベクトル 90, 98, 116, 122, 142 割込みルーチン 98, 116, 122
ワード		13		

	【A】		【I】		【P】
A-D 変換	43	IC	12	PC	121
ALU	109	IOR	175	PIC	134
AND	57	IO インタフェース	94, 163	program counter	121
	【B】	IX	123		【R】
BCD	40		【L】	RAM	92
	【C】	LSB	44	ROM	94
CPU	86		【M】		【S】
	【E】	MSB	27	SP	123
EEPROM	94		【N】		【T】
EMC	72	NOT	60	True	57
EXOR	60		【O】		【X】
	【F】	OR	59	XOR	175
False	57	OS	100		

— 著者略歴 —

松田 忠重 (まつだ ただしげ)

1970年 姫路工業大学工学部電気工学科卒業
1971年 神戸市立工業高等専門学校助手
1996年 神戸市立工業高等専門学校教授
2002年 博士(理学)(甲南大学)
2011年 神戸市立工業高等専門学校名誉教授

佐藤 徹哉 (さとう てつや)

1986年 豊橋技術科学大学工学部電気・電子工学課程卒業
1988年 豊橋技術科学大学大学院工学研究科修了(電気・電子工学専攻)
1988年 松下電器産業(現 パナソニック)株式会社勤務
2001年 博士(工学)(豊橋技術科学大学)
2010年 神戸市立工業高等専門学校准教授
2012年 神戸市立工業高等専門学校教授
現在に至る

新編 マイクロコンピュータ技術入門

An Introduction to Microcomputer Technologies (New Edition)

© Tadashige Matsuda, Tetsuya Sato 2015

2015年2月27日 初版第1刷発行



検印省略

著者 松田 忠重
佐藤 徹哉
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 萩原印刷株式会社

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話 (03) 3941-3131(代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-02490-6 (高橋) (製本: 愛千製本所)

Printed in Japan



本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられております。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めておりません。

落丁・乱丁本はお取替えいたします