

まえがき

コンピュータ技術の進歩と、スマートフォンやタブレットなど携帯端末の急速な普及に伴い、われわれの日常生活のありとあらゆるサービスが電子化され、もはやコンピュータなしでは社会が成り立たなくなってきた。一方で、コンピュータには処理の高速化のために高度な並列処理や多階層メモリシステムなどが導入され、その能力を最大限に引き出すためには、さまざまなプログラム上の工夫が必要になってきている。したがって、コンピュータの仕組み、すなわちどのような原理でコンピュータが作られ、中身はどうなっているのかを学ぶことは、コンピュータの設計開発を目指す情報系学生ばかりでなく、コンピュータを活用して制御システムや応用プログラムの開発に従事する理工系学生やデータ分析に従事する医薬・人文・社会系学生にも必須になっている。

本書は、そのような要求に応えるべく、コンピュータの動作原理に関する入門書として、著者がコンピュータを専門としない機械系学生を対象に実施してきた「計算機工学」の講義資料をもとに書いたものである。本書がカバーする範囲は、コンピュータの歴史から、コンピュータの計算原理を与える論理代数、そして、論理代数に基づく回路設計、さらにはシステムソフトウェアを含むシステム構成・制御技術やネットワーク技術と幅広いが、その内容を本編と付録に分け、本編でひととおりコンピュータの基本原則とシステム構成が理解できるようになっている。ウェブ上で配布する付録では、本編で扱った原理・構成要素のさらなる詳細や並列処理などを含む高度なコンピュータ構成技術を扱っており、より深い知識が得られるようになっている。また、章末問題の解答も記載されている。

本書の執筆者は（私を除いて）執筆時点で40歳前後の、まさにコンピュータを活用して応用分野を切り開いてきた新進気鋭の教育・研究者たちである。非情報系の学生目線でまとめられた本書を通じて、多くの学生たちがコンピュータの仕組みを理解し、その得手・不得手を見極めながらコンピュータの能力を最大限に活用できるプログラム開発の一助になればと思っている。

最後に、本教科書を取りまとめるにあたり、ご尽力いただいたコロナ社に深く感謝する。

2015年1月

著者を代表して
小林 広明

目 次

1. 数 の 表 現

1.1	2進数によるデジタル表現	1
1.1.1	デジタル表現	1
1.1.2	2進数	3
1.1.3	10進数, 16進数, 8進数	5
1.1.4	2進数の加減算・乗除算	7
1.2	2進数による符号付き数の表現	8
1.2.1	符号と絶対値法	8
1.2.2	2の補数表現	9
1.2.3	2の補数表現の加減算	11
1.2.4	2の補数表現のビット拡張	13
1.3	固定小数点数と浮動小数点数	13
1.3.1	固定小数点数とその加減乗算	13
1.3.2	浮動小数点数	15
1.3.3	IEEE754 フォーマット	16
1.3.4	浮動小数点数の加減算	18
1.3.5	浮動小数点数の乗算と除算	19
1.3.6	丸め	20
	章末問題	22

2. ブール代数と論理回路

2.1	論理値と論理演算	23
2.2	論理関数	24
2.2.1	論理関数とは	24
2.2.2	真理値表	26
2.2.3	式の変形と公式	26
2.2.4	双対性	29

2.3 標準形	30
2.3.1 論理関数の設計	30
2.3.2 最小項と最大項	30
2.3.3 加法標準形と乗法標準形	32
2.3.4 真理値表から式への変換	33
2.4 論理回路	34
2.4.1 論理演算を行う電子回路	34
2.4.2 回路記号	34
2.4.3 論理関数と論理回路	36
2.5 よく使われる演算子	37
2.5.1 完備性	37
2.5.2 否定論理積 (NAND) と否定論理和 (NOR)	37
2.5.3 排他的論理和	38
章末問題	38

3. 組合せ回路

3.1 論理式の簡単化	40
3.2 カルノー図を使う方法	41
3.2.1 カルノー図	41
3.2.2 論理積の項と隣接するます目の関係	43
3.2.3 論理式の簡単化	43
3.2.4 ドントケア項のある場合	44
3.3 設計の具体例	45
3.4 基本的な組合せ回路	47
3.4.1 デコーダとエンコーダ	48
3.4.2 マルチプレクサとデマルチプレクサ	49
3.4.3 バレルシフタ	51
3.5 計算を行う組合せ回路	52
3.5.1 加算器	52
3.5.2 減算器	54
3.5.3 比較器	54
章末問題	55

4. 順序回路の基礎

4.1 状態機械	57
4.1.1 状態機械の定義	57
4.1.2 状態機械の例	58
4.1.3 同期式と非同期式の状態機械	58
4.2 入力・状態・出力集合	59
4.3 内部状態の遷移	60
4.3.1 状態遷移図	60
4.3.2 状態遷移表と出力表	62
4.4 順序回路	63
4.4.1 順序回路の構成	63
4.4.2 順序回路の動作	64
4.4.3 状態遷移関数と出力関数を表す論理関数	64
4.5 記憶回路	65
4.5.1 フリップフロップ	65
4.5.2 フリップフロップの動作の解析	66
4.5.3 クロック入力付きのフリップフロップ	67
4.5.4 マスタスレーブ型フリップフロップ	69
4.5.5 フリップフロップの種類	71
章末問題	72

5. 順序回路の設計と応用

5.1 順序回路の設計の概要	73
5.2 設計手順の詳細	74
5.2.1 状態遷移図の作成	74
5.2.2 遷移表と出力表の作成	75
5.2.3 状態・入力・出力の符号化	76
5.2.4 状態遷移関数と出力関数の実現	77
5.2.5 組合せ回路の設計	78
5.2.6 順序回路の完成	79
5.3 基本的な順序回路	80

5.3.1 レジスタ	80
5.3.2 シフトレジスタ	80
5.3.3 カウンタ	81
章末問題	83

6. コンピュータの構成とプログラムの実行 (1)

6.1 コンピュータの一般的な構成	84
6.1.1 メモリ	85
6.1.2 プロセッサ	86
6.1.3 命令セット	88
6.1.4 命令セットアーキテクチャの具体例	88
6.2 MIPS の命令セットとアセンブリ言語	89
6.3 MIPS の命令と動作：演算命令	91
6.3.1 レジスタ間の演算	91
6.3.2 レジスタの種類	93
6.3.3 即値演算	95
6.3.4 命令の組合せ例：ビット操作	96
6.4 MIPS の命令と動作：ロード命令・ストア命令	98
章末問題	102

7. コンピュータの構成とプログラムの実行 (2)

7.1 MIPS の命令と動作：分岐命令・ジャンプ命令	103
7.1.1 分岐命令	103
7.1.2 ジャンプ命令	105
7.2 関数呼出し	106
7.3 スタックとメモリマップ	109
7.4 MIPS 命令セットのまとめ	113
7.4.1 命令一覧	113
7.4.2 機械語との対応	114
章末問題	115

8. メモリシステム

8.1	SRAM	117
8.1.1	構 成	117
8.1.2	動 作	119
8.2	DRAM	119
8.2.1	構 成	119
8.2.2	動 作	120
8.3	不揮発性メモリ	121
8.4	記憶階層と参照の局所性	123
8.4.1	記 憶 階 層	123
8.4.2	参 照 の 局 所 性	123
8.5	キャッシュメモリ	125
8.6	仮 想 記 憶	126
	章 末 問 題	127

9. コンパイラ

9.1	高水準プログラミング言語	128
9.2	コンパイラの構成	129
9.2.1	字 句 解 析	131
9.2.2	構 文 解 析	133
9.2.3	意 味 解 析	135
9.2.4	中間コード生成	136
9.2.5	最 適 化	137
9.2.6	コ ー ド 生 成	138
	章 末 問 題	139

10. コンピュータネットワーク

10.1	コンピュータネットワークの基本概念	140
10.1.1	交 換 方 式	140
10.1.2	トポロジ	141

10.2	インターネット	142
10.2.1	インターネットの特徴	142
10.2.2	OSI 基本参照モデルと TCP/IP の階層構造	143
10.2.3	ネットワークインタフェース層	144
10.2.4	インターネット層	146
10.2.5	トランスポート層	149
10.2.6	アプリケーション層	152
10.3	DNS	152
10.3.1	IP アドレスとドメイン名	153
10.3.2	DNS による名前解決	154
10.4	インターネットのセキュリティ	156
	章 末 問 題	158

11. 計算機の歴史

11.1	計算に用いられた古代の道具	159
11.2	歯車による機械式計算機の時代	160
11.3	電気・電子式計算機の時代：近代的計算機の黎明期	162
11.4	集積回路と現代の計算機	165

索 引	170
-----	-----

付 録

以下の Web ページからダウンロード可能である。

<http://www.coronasha.co.jp/np/isbn/9784339024920>

(本書の書籍ページ。コロナ社のトップページから書名検索でもアクセスできる)

A. 順序回路：発展編		C. 入出力システム			
A.1	状態数の最小化	1	C.1	メモリマップ I/O 方式と入出力専用命令方式	10
A.2	非同期式順序回路	3	C.2	割 込 み	11
B. キャッシュメモリの構成		C.3	DMA 転 送	13	
B.1	データ格納位置の決定	5	C.4	入出力とオペレーティングシステム	13
B.2	データ入替え方式	8	C.5	入出力システムの実際	14
B.3	書込みアクセス	8	章 末 問 題	16	
章 末 問 題		9			

D. オペレーティングシステム		F. コンピュータの高性能化	
D.1 オペレーティングシステムの役割	17	F.1 コンピュータの性能	49
D.1.1 ハードウェアの有効活用	18	F.2 コンピュータのコスト	50
D.1.2 使いやすさの向上	19	F.3 さまざまな命令形式	51
D.2 オペレーティングシステムの構成要素	19	F.3.1 演算構成とオペランド指定方式	51
D.2.1 制御プログラム	20	F.3.2 命令の大規模化・複雑化	53
D.2.2 サービスプログラム	22	F.4 布線論理制御とマイクロプログラム制御	54
D.2.3 言語処理プログラム	22	F.5 CISC と RISC	55
D.3 マルチプログラミングと割込み	23	F.5.1 RISC の 登 場	55
D.4 仮 想 記 憶	25	F.5.2 パイプライン処理	57
章 末 問 題	28	F.5.3 パイプラインの実現例	59
E. プロセッサの実現		F.5.4 CISC vs RISC	61
E.1 命 令 セ ッ ト	29	F.6 さ ら な る 高 速 化	62
E.2 全 体 構 成	30	F.6.1 クロックサイクル時間の短縮	62
E.3 構 成 要 素	31	F.6.2 命令レベル並列性	62
E.3.1 基本的な回路部品	31	F.6.3 データ並列性	63
E.3.2 命 令 メ モ リ	33	F.6.4 スレッドレベル並列性	65
E.3.3 汎用レジスタ	34	章 末 問 題	66
E.3.4 ALU	35	G. コンピュータネットワーク：発展編	
E.4 各部の動作と構成	37	G.1 Ethernet による通信	68
E.4.1 IF (命令フェッチ)部	38	G.2 TCP の通信手順	69
E.4.2 ID (命令デコード)部	38	G.3 アプリケーション層プロトコルの例：POP3	71
E.4.3 EX (実行)部	41	H. 計算機の歴史：資料	
E.5 制 御 回 路	42	章末問題解答	
E.6 動 作 例	43	87	
E.7 プロセッサ構成法の一般論	44		
章 末 問 題	47		

1 | 数の表現

複雑な数値計算のみならず、音楽や映像の再生などのさまざまな処理が可能な現在のコンピュータ（計算機）には、0と1の二つの状態を表す、ビットを単位としてデータを表すデジタル方式が用いられている。本章では、数値データのデジタル表現について述べる。

1.1 2進数によるデジタル表現

1.1.1 デジタル表現

『広辞苑』（岩波書店）によれば、デジタル（digital）とは「ある量を有限桁の数字列として表現すること」とある。これに対し、アナログ（analog）とは「ある量を連続的に変化する物理量で表現すること」と説明されている。図 1.1 に示す 2 種類の温度計は、アナログ表現とデジタル表現の例である。

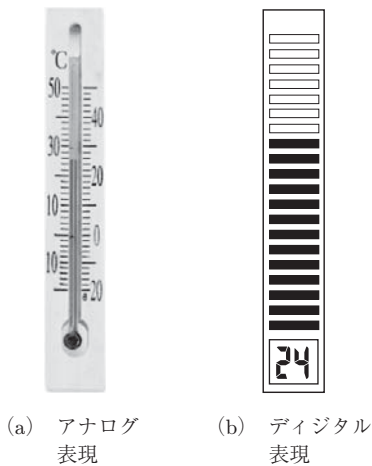


図 1.1 アナログ表現とデジタル表現

図 (a) の水銀温度計では、温度に応じて球部の水銀が熱膨張し、中央部の毛細管中を上昇する。水銀の高さを読み取ることで温度を知ることができる。温度に応じて連続して変化する水銀の高さは、温度のアナログ表現であると言える。一方、図 (b) のデジタル温度計では、

† コンピュータ（computer）と計算機は同じ意味で用いた。ただし、11 章では歴史を扱っているため計算機と記した。

センサにより読み取られた温度情報が、何らかの電氣的処理の後にいくつかのライトの点灯により表現されている。各ライトの点灯と消灯をそれぞれ1, 0と考えれば、これは1, 0の有限桁の数字列による温度のデジタル表現に当てはまることがわかる。これらの二つの表現には決定的な違いがある。アナログ表現の水銀温度計では、読み取ることさえできればごく微小な温度変化を観測できる。一方、デジタル温度計では、ライトの点灯が 0.25°C 単位であるとすれば、それよりも小さな温度変化を知ることはできない。このように、アナログにより表現される値は連続であり中間値をいくらでももちうるのに対し、デジタル表現は離散的であり中間値をもたない。現在のコンピュータは、デジタル方式に基づいて作られている。アナログに比べて表現の限定的なデジタル方式を用いるのはなぜだろうか。それは、与えられたデータの記録・処理・伝送を正しく行う必要があるからである。例えば、銀行のオンラインシステムにおいて、貯金の残高が水銀柱の高さにより記録・処理されていることを想像する。この場合、完璧な温度調節を備えた部屋にでも保管しない限り水銀柱の高さは絶えず変動し、正確な残高がわからなくなる。あるいは、水銀が少しずつ漏れていき、残高が減ってしまうかもしれない。また、口座に入金があった場合、水銀柱の高さを入金額の分だけ正確に増やすことは難しい。さらに、残高情報を送信するために、高さを正確に一定に保ったまま水銀柱を別の支店に運ぶのは至難の業である。これらの問題は雑音や減衰によるアナログ表現の劣化が原因であり、水銀柱の高さのかわりに電子回路における電圧によりアナログで残高情報を表現しても変わることはない。このように、連続な物理量をそのまま用いるアナログ表現では、情報の劣化を避けるのは本質的に困難である。

一方、デジタル表現では、連続情報の一部を切り捨てて離散的に情報を近似表現することにより、情報の劣化を抑えることが可能である。図(b)のデジタル温度計のようにライトの点灯個数により残高を表せば、よほどのことがない限り金額を間違えることはない。もちろん、表現できる金額はライトの数に左右されるが、通常、残高は有限桁の数値により表現できるため、十分な数のライトを用意すれば実用上問題はない。

実際のコンピュータにおいては、デジタル温度計におけるライトの点灯・消灯のかわりに、電圧の高低により表現の基本単位であるビットを表す。例えば、 0V と 1.1V により2通りの状態を表現し、それぞれ0, 1という記号に対応させると、それぞれの電圧が完全に 0V , 1.1V でなくても、ある閾値電圧を境に0, 1を区別できる。このため、情報を記憶する、あるいは遠方に伝送する際に電圧が若干変動しても、元と変わらない0, 1を維持することが可能となる。また、後述のように、数値データを0, 1の列である2進数により表現すれば、1, 0を真偽に見立てた論理操作(2章以降で学ぶ)により値の計算ができる。

いわゆる「数」以外のデータも、コンピュータ内ではすべてデジタル値として表現される。例えば、テキスト処理などは各文字に数値コードを割り振り、そのデジタル値を操作することにより行われる。ASCII(American Standard Code for Information Interchange, アスキー)

は、ほとんどのコンピュータで用いられている規格であり、0~127の文字コード（character code）により、英数字、記号、あるいは改行やタブなどの制御を表す。表 1.1 に ASCII による文字コード表（抜粋）を示す。

表 1.1 ASCII 文字コード表（抜粋）

コード	文字	コード	文字	コード	文字	コード	文字	コード	文字	コード	文字
32	空白	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	DEL

以上のように、デジタル表現は信頼のおけるコンピュータを実現するための基本原理であり、ハードウェア設計からソフトウェア開発、またコンピュータの利用に至るまで欠くことのできない基本知識である。情報の劣化に強いとはいえデジタル表現は万能ではなく、限られた数のビットを用いて目的の情報を効率よく適切に表現することが求められる。以下、コンピュータに用いられる数値表現や、その計算方法について述べる。

1.1.2 2 進 数

1.1.1 項では、今日のコンピュータにおいては 0, 1 の 2 通りの状態を表すビットによりデータを表すことを述べた。2 進数（binary number）は複数のビットにより数を表示する方法である。より正確には、基数（base number, radix）を 2 として数を表示したものが 2 進数であり、その 1 桁（binary digit）をビット（bit）と呼ぶ。われわれが普段使っているのは 10 進数（decimal number）である。図 1.2(a) に示すとおり、10 進数では、0~9 の 10 種類の記号を用いて数を表示している。

このため、各桁では 10 通りの数を数えることが可能である。この表現に用いる記号の種類のことを基数という。10 進数は基数が 10 の数の表現法である。一方、図 (b) の 2 進数では 0 と 1 の 2 種類の記号により数を表示しているため、基数は 2 となる。基数が 2 の場合には、各桁は 2 通りの数のみ数えることができる。したがって、図 1.3 に示すように、1 に 1 を加算

	0, 1,
	10, 11,
	100, 101,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9,	110, 111,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,	1000, 1001,
20, ...	1010, 1011,

(a) 10進数	(b) 2進数

図 1.2 10進数と2進数

$\begin{array}{r} 01 \\ +)01 \\ \hline 10 \end{array}$	$\begin{array}{r} 11 \\ +) 01 \\ \hline 100 \end{array}$
(a)	(b)

図 1.3 2進数の繰上りの例

すると、つぎの桁に繰上がり（桁上がり，キャリー，carry）が生じ、その桁は0となる。

基数が B の数を考える。このとき、 $a_N a_{N-1} \cdots a_2 a_1 a_0$ と表された $(N+1)$ 桁の数は式 (1.1) により計算される数を表している。

$$a_0 B^0 + a_1 B^1 + a_2 B^2 + \cdots + a_{N-1} B^{N-1} + a_N B^N = \sum_{i=0}^N a_i B^i \quad (1.1)$$

10進数の場合には、0 から数えて i 桁目が 10^i の数を表していることを考えればわかりやすい。さて、基数 B が 2 の 2進数の場合には

$$a_0 2^0 + a_1 2^1 + a_2 2^2 + \cdots + a_{N-1} 2^{N-1} + a_N 2^N = \sum_{i=0}^N a_i 2^i \quad (1.2)$$

となり、 a_i が 0, 1 であることから、 i 桁目が 1 なら 2^i を表していることになる。

本来の 2進数には桁数に制限はないが、一般的なコンピュータでは、便宜上ある決まった桁数の 2進数を用いる。この桁数はコンピュータや処理系[†]によりさまざまではあるが、多くの場合、バイト (byte) と呼ぶ単位の倍数の桁数を扱う。現代のほとんどのコンピュータでは 8 ビットを 1 バイトとする。図 1.4(a) に 1 バイトの 2進数の例を示す。

上位 ←	→ 下位	
<u>1</u>	<u>0</u>	<u>1</u>
0	0	1
0	1	0
1	1	1
0	0	0
MSB	LSB	
(a) 1バイトの2進数		(b) 32ビット(1ワード)の2進数

図 1.4 コンピュータにおける有限桁の 2進数の例

2進数において、数の小さな位のことを下位、数の大きな位のことを上位と呼ぶ。また、最も下の位のことを最下位ビット (least significant bit, **LSB**) と呼ぶ。同様に、最も上の位のこ

[†] プログラミング言語が動作するコンピュータ上の環境、またはコンピュータ上で動作できるような準備を行うソフトウェアを言語処理系、略して処理系 (programming language processing system) と呼ぶ。8 章で学ぶコンパイラやインタプリタなどを指す。

とを最上位ビット (most significant bit, MSB) と呼ぶ。バイトのほかによく用いられる単位として、ワード (word) がある。ワードは語と表記することもある。ワードはそのコンピュータや処理系で最も自然に扱うことができる桁数であり、機種により 16 ビット, 32 ビット, 64 ビットなどと異なる。しかしながら、これまでに広く普及している 32 ビットコンピュータでは 32 ビットの固定長データを 1 ワードとすることが多いことから、本書でも特に断りがない限り 1 ワードは 32 ビットであるとする。図 (b) に 32 ビット 1 ワードの 2 進数の例を示す。

プログラミング言語においても、これらの基本的な桁数の 2 進数を表現する方法が用意されている。図 1.5 に示すように、C 言語では、符号なし数 (unsigned number) を表す変数型である unsigned char, unsigned short, unsigned int は、多くの処理系において、それぞれ 8 ビット, 16 ビット, 32 ビットの 2 進数に対応する。これらが表現可能な値の範囲を図 (b) に示す。コンピュータにおいて、演算結果がこの表現可能な値の範囲を超えてしまうことをオーバーフロー (overflow), またはあふれという[†]。オーバーフローはプログラムが正しく動かない原因となりうる。

8 bit (unsigned char) 0 ~ 255 (= $2^8 - 1$)	(signed char) - 128 ~ 127
16 bit (unsigned short) 0 ~ 65535 (= $2^{16} - 1$)	(signed short) - 32768 ~ 32767
32 bit (unsigned int) 0 ~ 4294967295 (= $2^{32} - 1$)	(signed int) - 2147483648 ~ 2147483647
(a) 符号なし	(b) 符号付き

図 1.5 C 言語における整数型と、典型的な処理系で表現可能な値の範囲
(unsigned は符号なし, signed は符号付きを表す)

1.1.3 10 進数, 16 進数, 8 進数

10 進数の数 37 を 2 進数で表現するとどうなるであろうか。また, 1011 という 2 進数は 10 進数ではいくらだろうか。特にプログラムを作成する際に、このような基数の異なる数どうしの変換方法が必要となることがある。1 と 0 だけの 10 進数の数は 2 進数と区別がつかないため、以降、本章では基数を右下に添えることとする。基数を省略した場合には 10 進数として解釈することとする。

2 進数から 10 進数への変換は、式 (1.2) を計算することにより行える。例えば, 1011_2 は

$$1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 1 + 2 + 8 = 11 \quad (1.3)$$

であることから 11 とわかる。一方, 10 進数から 2 進数への変換にはさまざまな方法が考えられるが、筆算により変換する方法として図 1.6 の方法がある。この例では 37 を 2 進数 1001012 に変換している。変換は、2 での割り算を繰り返し行い、余りを並べていくことにより行われ

[†] 符号なし数においては、最上位ビットを超えて繰上がり・繰下がりが生じる場合と一致する。そのため桁あふれとも呼ばれるが、後述する符号付き数も含めて考えると、桁があふれることとオーバーフローは必ずしも一致しない。

$$\begin{array}{r}
 \begin{array}{l}
 \text{2での} \\
 \text{除算}
 \end{array}
 \left.
 \begin{array}{l}
 2 \overline{) 37} \quad \text{余り} \\
 2 \overline{) 18} \quad \cdots 1 \\
 2 \overline{) 9} \quad \cdots 0 \\
 2 \overline{) 4} \quad \cdots 1 \\
 2 \overline{) 2} \quad \cdots 0 \\
 2 \overline{) 1} \quad \cdots 0 \\
 0 \quad \cdots 1
 \end{array}
 \right\} 100101_2
 \end{array}$$

図 1.6 筆算による 10 進数から 2 進数への変換

る。図の例では、まず 37 を 2 で割り、商 18 と余り 1 を下に書く。つぎに、18 を 2 で割り、商 9 と余り 0 を書く。これを商が 0 になるまで繰り返す。最後に、一番下を最上位、一番上を最下位として縦に並んだ余りを並べると、変換後の 2 進数が求まる。

コンピュータの内部では 2 進数がいられるものの、人間が読み書きするには桁数が多すぎで不便である。10 進数は人間が読み書きしやすいが、上述のように 2 進数との相互変換が煩雑である。そこで、8 進数 (octal number) や 16 進数 (hexadecimal number) を用いることがある。8 進数は基数が 8 の数の表現方法であり、16 進数は基数が 16 の数の表現方法である。図 1.7 にこれらの特徴を示す。図 (a) に示す 8 進数では、各桁において 0~7 の 8 個の記号を用いる。各桁が表す 0~7 の数は、3 ビットの 2 進数に対応している。このことから、図のように、2 進数を 3 ビットごとに区切り、それぞれを各桁に変換することにより、2 進数から 8 進数への変換が行える。逆も同様である。

8 個の記号 : 0, 1, 2, 3, 4, 5, 6, 7
2 進数の 3 ビットが 8 進数の 1 桁に対応

2 進数	001	110	101	011 ₂
		↓		
8 進数	1	6	5	3 ₈

(a) 8 進数(基数 = 8)

16 個の記号 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
(F は 10 進数の 15 に相当する)
2 進数の 4 ビットが 16 進数の 1 桁に対応

2 進数	0011	1010	1011 ₂
		↓	
16 進数	3	A	B ₁₆

(b) 16 進数(基数 = 16)

3 桁、4 桁ごとに 10 進数と 2 進数どうしの変換と同様に変換する。

図 1.7 8 進数と 16 進数の特徴

図 (b) に示す 16 進数の場合には、16 種類の記号を用いる。このため、0~9 に加えて、アルファベットの A~F を使用する。このアルファベットは小文字として書かれる場合もある。A, B, C, D, E, F はそれぞれ 10, 11, 12, 13, 14, 15 を表す。すなわち、16 進数では各桁は 15 まで数えることができ、これを超えるとつぎの桁に桁上げを生じることになる。0~15 の数は、4 ビットの 2 進数に対応している。このため、8 進数の場合と同様に 4 ビットごとに区切り、それぞれを変換することにより 2 進数と 16 進数の変換が行える。以上のように、8 進数と 16 進数ではそれぞれ 3 ビット、4 ビットの 2 進数と各桁の対応関係さえ覚えてしまえば、各桁独立に素早く変換が可能となる。

1.1.4 2進数の加減算・乗除算

2進数の加減算・乗除算は、繰上がり繰下がりについて10進数の筆算と同様に行えばよい。図1.8に筆算による2進数の加減算の例を示す。2進数では各桁1までしか数えることができないため、1に1を加算する場合にはつぎの桁に繰り上がり、1を生じる。この点に注意して、あとは10進数の場合と同様に下位から順に各桁の加算を行えばよい。桁上げ1を入れると1+1+1となる桁では、つぎの桁に1を繰り上げ、かつその桁の答も1となる。減算の場合、0-1となる桁では上の桁からの繰下がり（桁下がり、桁借り、ポロー、borrow）を必要とする。この場合、上の桁からは2を借りてくることになるため、2-1=1がその桁の答となる。図の例では、2桁目の上の値1が繰下がりにより0となっている。

$\begin{array}{r} 1111 \\ 01111010_2 \\ +) 00111001_2 \\ \hline 10110011_2 \end{array}$	<p>10進数</p> <p>122</p> <p>57</p> <p>179</p>	$\begin{array}{r} 0 \\ 01111010_2 \\ -) 00111001_2 \\ \hline 01000001_2 \end{array}$	<p>10進数</p> <p>122</p> <p>57</p> <p>65</p>
(a) 繰上がり		(b) 繰下がり	

図 1.8 筆算による2進数の加減算

加減算がわかれば、乗除算も10進数と同様に行うことができる。図1.9に筆算による2進数の乗除算の例を示す。乗算の場合、乗数の各桁を下位より見ていき、1の場合には被乗数をその桁の位置から書き出す。0の場合には0を並べた数を書き出す。乗数のすべての桁についてこれを行った後、最下位より各桁の加算を行う。複数の1を加算する場合には必要な数だけつぎの桁へ繰上げを行う。加算の結果が乗算の答である。図が示すように、乗算の場合には、乗数、被乗数よりも桁が増える場合がある。一般に、 n 桁と m 桁の乗算の結果は最大で $(n+m)$ 桁の2進数となる。例えば、すべての桁が1の数どうしの乗算は、最大の桁数の積を生じる。

$\begin{array}{r} 0111_2 \quad 7 \\ \times) 1101 \quad 13 \\ \hline 0111 \\ 0000 \\ 0111 \\ 0111 \\ \hline 1011011_2 \quad 91 \end{array}$	$\begin{array}{r} 1011_2 \quad 11(\text{商}) \\ 1011_2 \overline{) 01111010_2} \quad (122 \div 11) \\ \hline 1011 \\ \hline 10001 \\ \hline 1011 \\ \hline 1100 \\ \hline 1011 \\ \hline 1_2 \quad (\text{余り } 1) \end{array}$
(a) 乗算	(b) 除算

図 1.9 筆算による2進数の乗除算

除算の場合、被除数の最上位から見ていき、除数を引くことのできる桁数まで下がったところから上に1を書き出す。その位置の下に除数を書き出し、減算を行う。減算結果に被除数の残りの桁を加えていき、除数を引くことができなければ0を書き出していき。引くことができる場合には1を書き出す。以上を被除数の最下位まで行くと、上方に商が、下方に余りが求め

索 引

太字となっているページには、基本的な説明が詳しく書かれている。

「付」が記されているページ数は付録（Web で配布）でのページ数を示している。

<p style="text-align: center;">【あ】</p> <p>アイドル時間 →遊休時間</p> <p>アウトオブオーダー実行 付63</p> <p>アキュムレータ 付52</p> <p>アクセス 86</p> <p>アクセス権限 付21</p> <p>アセンブラ 90</p> <p>アセンブリ言語 90</p> <p>値による呼出し 107</p> <p>値渡し →値による呼出し</p> <p>アドレス 85</p> <p>アドレス空間 →メモリ空間</p> <p>アドレスバス 付11</p> <p>アドレッシングモード 98</p> <p>アナログ 1</p> <p>アプリケーション層 (OSI) 143</p> <p>アプリケーション層 (TCP/IP) 152</p> <p>アプリケーションソフトウェア →応用ソフトウェア</p> <p>あふれ →オーバーフロー</p> <p>アンダーフロー 17</p> <p style="text-align: center;">【い】</p> <p>イーサネット →Ethernert</p> <p>一時レジスタ 92</p> <p>イネーブル信号 付32</p> <p>意味解析 135</p> <p>インオーダー実行 付63</p> <p>インターネット 142</p> <p>インターネット層 146</p> <p>インタプリタ 129</p> <p style="text-align: center;">【う】</p> <p>上向き構文解析法 134</p> <p style="text-align: center;">【え】</p> <p>エッジトリガ型フリップフロップ 70</p> <p>エンコーダ →2 進エンコーダ</p> <p>エンコーディング →符号化</p>	<p style="text-align: center;">【お】</p> <p>応答時間 →レスポンスタイム</p> <p>応用ソフトウェア 付17</p> <p>オーバーフロー 5</p> <p>オーバーフロー (浮動小数点数) 17</p> <p>オフィスコンピュータ 167</p> <p>オブジェクトコード 129</p> <p>オペコード 91</p> <p>オペランド 91</p> <p>オペレーティングシステム 付17</p> <p style="text-align: center;">【か】</p> <p>カーネルモード →特権モード</p> <p>回線交換方式 141</p> <p>外部割込み 付24</p> <p>カウンタ 81</p> <p>加算器 (リップルキャリー型) 53</p> <p>仮数 15</p> <p>仮想アドレス 付26</p> <p>仮想記憶 127, 付25</p> <p>加法標準形 32</p> <p>カルノー図 41</p> <p>関数 (C 言語) 107</p> <p>完 備 37</p> <p style="text-align: center;">【き】</p> <p>偽 23</p> <p>記憶階層 123</p> <p>機械語 90</p> <p>基数 3</p> <p>基本ソフトウェア →オペレーティングシステム</p> <p>キャッシュブロック →キャッシュライン</p> <p>キャッシュメモリ 125</p> <p>キャッシュライン 付5</p> <p>キャリー →繰上がり</p> <p>行アドレスデコーダ 118</p> <p>行選択線 118</p> <p>共通部分式削除 137</p>	<p style="text-align: center;">【く】</p> <p>空間の局所性 124</p> <p>具象構文木 135</p> <p>組合せ回路 40</p> <p>組合せ禁止項 →ドントケア項</p> <p>組込みプロセッサ 88</p> <p>クライアント 150</p> <p>グラフィカルユーザインタフェース 付19</p> <p>繰上がり 4, 52</p> <p>繰下がり 7, 54</p> <p>クリティカルパス 付44</p> <p>クロスバー 142</p> <p>クロック信号 68</p> <p style="text-align: center;">【け】</p> <p>計算機 →コンピュータ</p> <p>ゲート 34</p> <p>桁上がり →繰上がり</p> <p>桁借り →繰下がり</p> <p>桁下がり →繰下がり</p> <p>下駄履き表現 16</p> <p>言語処理系 →処理系</p> <p style="text-align: center;">【こ】</p> <p>語 →ワード</p> <p>高級言語 →高水準言語 90, 129</p> <p>高水準言語 90, 129</p> <p>構造ハザード 付58</p> <p>構文解析 133</p> <p>構文木 134</p> <p>固定小数点数 13</p> <p>コリジョン 146</p> <p>コンパイラ 90, 129</p> <p>コンピュータ 1</p> <p>コンピュータウィルス 157</p> <p>コンピュータ援用設計 付56</p> <p>コンピュータネットワーク 140</p> <p style="text-align: center;">【さ】</p> <p>サーバ 150</p>
---	---	---

最下位ビット	4	ジョブスケジューラ	付20	ソフトウェア割込み	付13, 付24
最近接丸め (偶数)	20	ジョブステップ	付20		
最上位ビット	5	処理系	4	【た】	
最小項	31	真	23	ターンアラウンドタイム	付18
最大項	31	シンボルテーブル	136	待機状態	付23
最適化	137	真理値表	26	ダイナミックメモリ	121
サブネットワーク	147			タイミングチャート	68
サブルーチン	107	【す】		ダイレクトマップ	付6
算術論理演算ユニット	→ALU	スーパーコンピュータ	168	タスク	付20
参照の局所性	124	スーパースカラ	付63	タスクスケジューラ	付21
		スーパーバイザコール		立上がりエッジ	71
【し】		→システムコール		立下がりエッジ	71
時間的局所性	124	スーパーパイプライン	付62	単精度	16
識別子	135	スケリング則	付65	【ち】	
字 句	131	スケジューラ		遅延分岐	付67
字句解析	131	→ジョブスケジューラ,		チップ	34
指 数	15	→タスクスケジューラ		中央処理ユニット	→プロセッサ
システムコール	付14, 付22	スター	141	中間表現	130
システムソフトウェア	付17	スタック	109	抽象構文木	135
下向き構文解析法	134	スタックフレーム	109		
実アドレス	付26	スタックポインタ	101, 110	【つ】	
実 行	87	スタティックメモリ	121	通信プロトコル	142
実効アドレス	98	ストア	98	ツリー	141
実行可能状態	付23	ストール	付58		
実行状態	付23	スプール	付20	【て】	
シフト	8	スループット	付18, 付58	ディジタル	1
シフト回路	→パレルシフタ	スレッド	付25	定数伝搬	137
シフトレジスタ	80	スレッドレベル並列性	付66	定数の畳込み	137
ジャンプ	103			ディレクトリ	付18
集積回路	34	【せ】		データハザード	付59
終端記号	134	正規化浮動小数点数	16	データベース	付11
主加法標準形	32	正規表現	131	データベース	付44
主記憶装置	85	制御回路	91	データ並列性	付63
主乗法標準形	32	制御ハザード	付59	データリンク層	144
出力関数	58, 65	制御バス	付44	デコーダ	→2進デコーダ
出力表	62	制御部	86	デコーディング	→復号化
出力表 (真理値表形式)	78	制御プログラム	付17	手続き	107
出力変数	64	生成規則	134	デバイスドライバ	付14
出力変数関数	65	生存解析	139	デマルチプレクサ	49
循環シフト	51	正の方向への丸め	20		
順序回路	57	セグメント	付27	【と】	
条件分岐	→分 岐	セッション層	143	同期式カウンタ	81
状態機械	57	セットアソシアティブ	付7	同期式状態機械 (順序回路)	59
状態遷移関数	57, 64	セレクタ	→マルチプレクサ	同時マルチスレッディング	付66
状態遷移図	61	全加算器	53	等値比較器	55
状態遷移表	62	センスアンプ	120	動的アドレス変換機構	付26
状態遷移表 (真理値表形式)	78			トークン	→字 句
状態変数	64	【そ】		特権モード	付14
状態変数関数	64	双対性	29	トポロジー	141
冗長項	→ドントケア項	ソースコード	129	ドメイン名	152
乗法標準形	32	即 値	95	ド・モルガンの定理	28
ジョブ	付20	ソケット	151		

- | | | | | | |
|-------------------|----------|------------------|---------|------------------|-------------|
| トランジスタ | 34 | 引数 | 107 | プログラム内蔵 | →プログラム記憶 |
| トランスポート層 (OSI) | 143 | 非終端記号 | 134 | プロセス | 付24 |
| トランスポート層 (TCP/IP) | 150 | 非正規化数 | 17 | プロセッサ | 84 |
| ドントケア項 | 44 | 左シフト | 8 | プロセッサ時間 | 付23 |
| | | ビッグエンディアン | 86 | フロントエンド | 130 |
| 【な】 | | ヒット | 126 | 分岐 | 103 |
| 内部割込み | 付13, 付24 | ビット | 2, 3 | 分岐予測 | 付59 |
| | | ビットごと論理演算 | 92 | | |
| 【に】 | | ビットシフト | →シフト | 【へ】 | |
| 二ーモニック | 90 | ビット線 | →列信号線 | ページ | 付27 |
| 入出力装置 | 85 | 否定 | 23 | ページフォルト | 付27 |
| 入力変数 | 64 | 否定論理積 | 37 | ページモードアクセス | 121 |
| | | 否定論理和 | 37 | ページング | 付27 |
| 【ね】 | | 非同同期式状態機械 (順序回路) | 59 | ベクトルプロセッサ | 付64 |
| ネームサーバ | 154 | | | ヘテロジニアスマルチコア | 付66 |
| ネットワークアダプタ | 付15 | 【ふ】 | | ベン図 | 31 |
| ネットワークインタフェース層 | 144 | ファイル | 付18 | | |
| ネットワーク層 | 144 | ファイルシステム | 付18 | 【ほ】 | |
| 【の】 | | ファットツリー | 141 | ポート | 150 |
| ノード | 140 | フィールド (命令) | 114 | ポートスキャン | 156 |
| | | フィッシング | 157 | ポート番号 | 150 |
| 【は】 | | ブートローダ | 122 | ポーリング | 付11 |
| バーストモードアクセス | 121 | ブール代数 | 23 | 補助記憶装置 | 85 |
| パーソナルコンピュータ | 165 | フェッチ | →命令フェッチ | 補数変換 | →2の補数変換 |
| ハードディスク | 付11 | フォルダ | →ディレクトリ | ポップ | 109 |
| バイアス値 | 16 | 不揮発性メモリ | 122 | ポロー | →線下がり |
| 倍精度 | 16 | 復号化 | 48 | | |
| 排他的論理和 | 38 | 符号化 | 48 | 【ま】 | |
| バイト | 4 | 符号拡張 | 13 | マイクロアーキテクチャ | 88 |
| ハイパーキューブ | 142 | 符号付き数 | 8 | マイクロコンピュータ | 165 |
| パイプライン処理 | 付57 | 符号と絶対値法 | 8 | マイクロプログラム制御 | 付55 |
| バグ | 129, 163 | 符号なし数 | 5 | マイクロプロセッサユニット | →プロセッサ |
| バケット | 141 | 符号ビット | 10 | マクロ命令 | 92 |
| バケット交換方式 | 141 | 布線論理制御 | 付54 | マスタスレーブ型フリップフロップ | 69 |
| バス | 付11 | ブッシュ | 109 | ます目 (カルノー図) | 41 |
| バス (トポロジー) | 141 | 物理アドレス | →実アドレス | マルチコア | 付66 |
| バスアダプタ | →バスブリッジ | 物理層 | 144 | マルチスレッド処理 | 付25, 付66 |
| バスブリッジ | 付14 | 浮動小数点数 | 15 | マルチタスク | →マルチプログラミング |
| バスカス記法 | 133 | 負の方向への丸め | 21 | マルチプレクサ | 49 |
| バックエンド | 130 | 不要コードの削除 | 138 | マルチプログラミング | 付23 |
| パラメトロン | 167 | プライオリティエンコーダ | 49 | マルチプロセス | →マルチプログラミング |
| パレルシフタ | 51 | フラッシュメモリ | 122 | マルチプロセッサ | 付66 |
| 半加算器 | 53 | プリチャージ | 120 | 丸め | 20 |
| 半導体チップ | →チップ | フリップフロップ | 65 | 丸め方式 | 20 |
| 万能 | →完備 | プリプロセッサ | 130 | | |
| 汎用機 | →メインフレーム | フルアソシアティブ | 付5 | 【み】 | |
| 汎用レジスタ | 87 | プレゼンテーション層 | 143 | ミーリー型機械 | 58 |
| | | プレディケート | 付62 | 右シフト | 8 |
| 【ひ】 | | ブロードキャスト | 141 | | |
| 比較器 | 54 | プログラム | 84 | | |
| | | プログラムカウンタ | 87 | | |
| | | プログラム記憶 | 84, 164 | | |

ミス	126
ミスペナルティ時間	126
ミニコンピュータ	165
【む】	
ムーア型機械	58
ムーアの法則	付51
無条件分岐	→ジャンプ
【め】	
命令	85
命令セット	88
命令セットアーキテクチャ	88
命令デコーダ	91, 付39
命令デコード	87
命令フェッチ	87
命令フォーマット	114
命令ポインタ	
	→プログラムカウンタ
命令レベル並列性	付62
メインフレーム	165
メタ言語	133
メニーコア	付66
メモリ	84
メモリ管理ユニット	127
メモリ空間	127
メモリマップ	112
メモリマップ I/O	付10
【も】	
文字コード	3

【ゆ】	
遊休時間	付18
有限状態機械	→状態機械
ユーザモード	付14
【よ】	
四つ組	136
予約語	131
【ら】	
ライトスルー	付8
ライトバック	付8
ライブラリ関数	付22
ラウンドロビン	付24
ラッチ	65
ラベル	103
【り】	
リーク電流	付65
リソース	付18
リップルカウンタ	81
リトルエンディアン	86
リフレッシュ	121
リンカ	130
リング	141
【る】	
ルータ	148
ルーティングテーブル	148
ループ不変コードの移動	138

【れ】	
例外	付13, 付24
レイテンシ	付58
レジスタ	80, 86, 付31
レジスタファイル	付34
レスポンスタイム	付18
列アドレスデコーダ	118
列信号線	118
レベルセンシティブフリップ フロップ	67
【ろ】	
ローテート	→循環シフト
ロード	98
ログ	付19
論理アドレス	→仮想アドレス
論理回路	34
論理関数	25
論理積	23
論理値	23
論理否定	→否定
論理変数	25
論理和	23
【わ】	
ワークステーション	165
ワード	5
ワード線	→行選択線
割込み	付12, 付24
割込みハンドラ	付12
割込みベクタテーブル	付12

【A】	
ALU	86, 付35
AND	→論理積
API	付22
ARP	149
ARPANET	142
ASCII	2
【B】	
BNF	→バックス記法
【C】	
CISC	付54
CPI	付49
CPU	→プロセッサ
CPU 時間	→プロセッサ時間
CRC	付68

【D】	
D フリップフロップ	71, 付31
DAT	→動的アドレス変換機構
DMA	付13
DNS	152
DNS サーバ	→ネームサーバ
DOS 攻撃	157
DRAM	117
【E】	
EPROM	122
Ethernet	144
【F】	
FCS	付68
FIFO	付27
FILO	→LIFO

FLOPS	付50
【G】	
GPGPU	付65
GPU	付65
GUI	→グラフィカルユーザインタ フェース
【I】	
IA-32	→x86
IEEE754	16
IP	146
IP アドレス	146
【J】	
JK フリップフロップ	71

			→パーソナルコンピュータ		
	【L】	PROM	122		【V】
LAN	142			VLIW	付63
LIFO	109	【R】			【W】
LRU	付8, 付27	RAM	117	wait 状態	→待機状態
LSB	→最下位ビット	ready 状態	→実行可能状態	WAN	142
	【M】	RISC	付54		【X】
MAC アドレス	146, 付68	ROM	122	XOR	→排他的論理和
MIL 記号	34	run 状態	→実行状態	x86	88
MIMD	付63				【数 字】
MIPS	89	【S】		0 拡張	13
MIPS (単位)	付50	SDRAM	121	0 方向への丸め	20
MMU	→メモリ管理ユニット	SIMD	付63	1 次記憶装置	→主記憶装置
MPU	→プロセッサ	SR フリップフロップ (エッジトリ		1 の補数	11
MSB	→最上位ビット	ガ型クロック入力付き)	71	2 次記憶装置	→補助記憶装置
mux	→マルチプレクサ	SR フリップフロップ (クロック入		2 次元トーラス	141
	【N】	力なし)	65	2 次元メッシュ	141
NaN	18	SR フリップフロップ (レベルセン		2 進エンコーダ	48
NAND	→否定論理積	シティブクロック入力付き)	67	2 進数	3
NOR	→否定論理和	SRAM	117	2 進デコーダ	48, 付33
NOT	→否 定			2 の補数	9
	【O】	【T】		2 の補数表現	9
OR	→論理和	T フリップフロップ	71	2 の補数変換	11
OS	→オペレーティングシステム	TCP	150	3 番地コード	136
OSI 基本参照モデル	143	TCP/IP	142	8 進数	6
	【P】			10 進数	3
PC	→プログラムカウンタ,	【U】		16 進数	6
		UDP	150		
		ulp	21		
		URL	152		

— 著者略歴 —

鏡 慎吾 (かがみ しんご)

1998 年 東京大学工学部計数工学科卒業
2000 年 東京大学大学院工学系研究科修士課程修了
(計数工学専攻)
2003 年 東京大学大学院工学系研究科博士課程修了
(計数工学専攻), 博士(工学)
2003 年 科学技術振興事業団研究員
2003 年 東京大学助手
2005 年 東北大学講師
2007 年 東北大学准教授
現在に至る

滝沢 寛之 (たきざわ ひろゆき)

1995 年 東北大学工学部機械知能工学科卒業
1997 年 東北大学大学院情報科学研究科博士課程前期
2 年の課程修了(情報基礎科学専攻)
1999 年 東北大学大学院情報科学研究科博士課程後
期 3 年の課程修了(情報基礎科学専攻),
博士(情報科学)
1999 年 新潟大学助手
2003 年 東北大学助手
2004 年 東北大学講師
2009 年 東北大学准教授
現在に至る

小林 広明 (こばやし ひろあき)

1984 年 東北大学工学部通信工学科卒業
1986 年 東北大学大学院工学研究科博士課程前期 2
年の課程修了(情報工学専攻)
1988 年 東北大学大学院工学研究科博士課程後期 3
年の課程修了(情報工学専攻), 工学博士
1988 年 東北大学助手
1991 年 東北大学講師
1993 年 東北大学助教授
2001 年 東北大学教授
2008 年 東北大学サイバーサイエンスセンター長
2012 年 東北大学教育研究評議会委員
2014 年 日本学術会議連携会員
現在に至る
この間, 1995, 1997, 2000~2001 年スタ
ンフォード大学コンピュータシステム研究
所客員准教授

佐野 健太郎 (さの けんたろう)

1997 年 東北大学大学院情報科学研究科博士課程前
期 2 年の課程修了(情報基礎科学専攻)
2000 年 東北大学大学院情報科学研究科博士課程後
期 3 年の課程修了(情報基礎科学専攻),
博士(情報科学)
2000 年 東北大学助手
2005 年 東北大学助教授
2007 年 東北大学准教授
現在に至る

岡谷 貴之 (おかたに たかゆき)

1994 年 東京大学工学部計数工学科卒業
1996 年 東京大学大学院工学系研究科修士課程修了
(計数工学専攻)
1999 年 東京大学大学院工学系研究科博士課程修了
(計数工学専攻), 博士(工学)
1999 年 東北大学助手
2001 年 東北大学講師
2003 年 東北大学助教授
2007 年 東北大学准教授
2013 年 東北大学教授
現在に至る

コンピュータ工学入門

Fundamentals of Computer Engineering

© Kagami, Sano, Takizawa, Okatani, Kobayashi 2015

2015年3月31日 初版第1刷発行

検印省略

著者 鏡 慎 吾
佐 野 健 太 郎
滝 沢 寛 之
岡 谷 貴 之
小 林 広 明
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 三美印刷株式会社

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 978-4-339-02492-0 (新宅) (製本: グリーン)

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います。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