

は し が き

本書はコロナ社の新しい大学教科書シリーズの中の1冊として、電子計算機のハードウェアとソフトウェアの基礎的知識を提供する目的で計画された。本書の執筆を東京大学の故宮川洋教授から丁重に依頼されたのは昭和51年ごろであった。それは筆者が東北大学から静岡大学に移って間もないころであったが、電子計算機についてはすでに国立と私立を合わせて六つの大学で講義をした経験があり、講義用のノートも持っていた。したがってそのご依頼にはすぐに応えられるつもりでお引受けしたのだが、いざ改めて新しい教科書を作るとなるとあれこれ盛り込みたい内容が多岐にわたり、また気負いもあった上に校務多忙が重なって、思うように筆が進まなかった。計算機技術の発展は実に急速であり、静岡大学での講義は学期により重点の置き所を変え、アーキテクチャ、オペレーティングシステム、ネットワーク、分散処理、高速演算装置、デジタル信号処理 LSI 等々、その時のトピックスや筆者自身の興味に従って内容を変えて来た。それは新しい教科書の執筆を念頭に置いてのことであったが肝心の原稿の方は抄らない。原稿の下書きは留まっても筆者の気持の方が変化してしまって一つにまとまらないのである。結局、筆者1人では何時まで経っても原稿が手許から離れないと自覚し、思い切って研究室の後輩で学生時代からその才能を尊敬していた名古屋工業大学の石井直宏教授に協力を求めたところ幸いに快諾を得、さらに新進気鋭の岩田彰助教授の協力も得ることができてここによりやく1冊の本として完成した。

本書は工学部電気系（電気・通信・電子・情報等）の学生が電子計算機について必ず持つべき基礎的でかつ系統的な知識で構成した。それは筆者の20年にわたる経験と試行・思惟の結果、流行の話題にあまり振りまわされることなく、地味ながら基礎的な事柄だけから成る簡潔で体系的な知識こそが、これを学んだ学生を将来新しい技術の発展を担う技術者・研究者として大成させ得

ると悟ったからである。

電子計算機の教科書は数多く出版されており、名著も少なくない。ここにさらに1冊を加えることについては一言述べるべきであろう。本書では、ハードウェアの基礎として数の表現法のレベルから説き起こして演算制御等の装置について述べた。ソフトウェアとハードウェアの境界となる機械語やアセンブリ言語によるプログラミングと高級言語によるプログラムを例示して、言語処理ソフトウェアの効用を実感させている。さらにソフトウェアの根幹であるオペレーティングシステムについては紙数の許す限り詳しく述べた。記憶装置に関する章でもソフトウェアとの関連性を述べた。計算機はハードウェアとソフトウェアから成るシステムであるのに拘わらず、それぞれが全く別の視点から書かれた本では教える側も学ぶ側も何となく釈然としないので、両者を一つの本の中で取り扱ってみたかったからである。

電子計算機は1946年のENIACの誕生以来驚異的な勢いで発展し続け、現在では宇宙的な広がりでさまざまな社会機構のあらゆる分野で、個人の日常生活の至る所に浸透している。それは“計算機”としてばかりではなくワープロとかゲームとかセンサとかいろいろの形態で機能を発揮している。そしてその機能や性能は目的に応じて千差万別であり、複雑・精巧で怪奇ですらある。しかしその原理はこの上なく単純であり、容易に理解できる。著者らは、本書によって若い読者が計算機の正体の簡単さ、およびその簡単なものがどうして高い機能と大きな効力を持ち得るかを学んで下さることを期待している。

終わりに、共著者として協力して下さった石井・岩田の両先生に深甚の感謝を捧げたい。もしもこの本がいささかなりとも評価し得る所があればそれは両氏のおかげであり、また不備や誤りがあったらその責めは筆者が負わねばならない。また、本書出版の機会を与えて下さった編集委員各位および長年月にわたって根気良く執筆を督励し続けて下さったコロナ社の編集部に心から感謝する。

1988年3月

著者を代表して 鈴木久喜

目 次



序論：計算機の技術の展開

1.1	計算の手順	1
1.2	電子計算機の技術の歴史	2
1.2.1	第1世代コンピュータ (1946～1954年)	4
1.2.2	第2世代コンピュータ (1955～1964年)	6
1.2.3	第3世代コンピュータ (1965～1974年)	8
1.2.4	第4世代コンピュータ (1975年～)	13
1.2.5	これからのコンピュータ	13
	演習問題	15



情報の表現

2.1	2進数と10進数の変換	16
2.2	負数の表現	18
2.2.1	絶対値表現	18
2.2.2	補数表現	18
2.3	2進数の演算	19
2.4	浮動小数点表現	26
2.5	符号化	28
2.5.1	B C D 符号	28
2.5.2	3余り符号	30
2.5.3	グレイ符号	31
2.5.4	パリティ符号	32

2.5.5	ASCII コードと EBCDIC コード	33
	演習問題	34

3 論理回路

3.1	論理回路の基礎	36
3.2	ダイオードによる基本論理演算	40
3.3	論理関数の標準形	41
3.4	論理関数の簡単化	45
3.4.1	ベイチ・カルノー図法	45
3.5	組合せ回路と順序回路	47
3.6	フリップフロップ回路	48
3.6.1	RS フリップフロップ	50
3.6.2	JK フリップフロップ	52
3.6.3	D フリップフロップ	53
3.6.4	T フリップフロップ	53
	演習問題	54

4 演算回路

4.1	加減算回路	55
4.2	2進化 10進加算回路	61
4.3	算術演算回路	62
4.4	算術論理演算装置	63
4.5	状態レジスタ	66
4.6	乗算回路	67
4.7	除算回路	70
	演習問題	72



コンピュータの命令と動作

5.1	コンピュータシステムの構成	74
5.2	CPU の 構 成	78
5.3	CPU の命令と動作	81
5.4	簡単な CPU の構成と動作	86
5.5	簡単な CPU の制御信号生成回路	89
5.6	SIMCOM の構成と命令	92
5.7	SIMCOM の命令実行動作	97
5.8	SIMCOM の制御信号生成回路	101
5.9	サブルーチン呼出し・復帰命令	107
5.10	マイクロプログラム制御	113
	演習問題	115



入出力制御

6.1	プログラム制御方式	118
6.2	割 込 み	124
6.3	ダイレクトメモリアクセス (DMA)	127
6.4	入出力制御装置	130
	演習問題	132



記憶装置と記憶方式

7.1	記憶装置の種類	133
-----	---------	-----

7.1.1	記憶装置の種類	133
7.1.2	記憶装置のアクセスモード	134
7.1.3	アクセス時間とサイクル時間	135
7.2	主記憶装置	135
7.2.1	主記憶装置のアドレス選択	135
7.2.2	半導体 RAM	136
7.2.3	読出し専用記憶 (read only memory, ROM)	140
7.3	補助記憶装置	142
7.3.1	磁性メモリ	142
7.3.2	光ディスクメモリ	147
7.3.3	固体ファイル記憶装置	150
7.4	記憶のシステム構成	151
7.4.1	記憶装置の階層化と主記憶アクセスの局所性	151
7.4.2	緩衝記憶方式	153
7.4.3	仮想記憶方式 (virtual memory system)	154
7.4.4	連想記憶 (associative memory)	160
7.4.5	記憶保護	161
	演習問題	163



プログラミング

8.1	機械語によるプログラミング	164
8.2	アセンブリ言語によるプログラミング	176
8.3	高級言語によるプログラミング	180
8.3.1	FORTRAN	182
8.3.2	BASIC	184
8.3.3	COBOL	186
8.3.4	C	186
8.3.5	LISP	189
8.3.6	PROLOG	190
	演習問題	194



オペレーティングシステム

9.1	オペレーティングシステム概念	196
9.2	オペレーティングシステム役割	197
9.2.1	処理能力の向上	197
9.2.2	応答時間の短縮	198
9.2.3	RASIS の向上	199
9.2.4	利用者とシステムのインタフェース機能の向上	200
9.2.5	ハードウェア資源の仮想化	201
9.3	計算機システムの運用形態	201
9.4	OS の 構 成	204
9.4.1	OSの階層構造	204
9.4.2	処理プログラムの概要	205
9.5	ジョブ管理	209
9.5.1	ジョブ管理の機能	209
9.5.2	ジョブ制御言語	210
9.5.3	ジョブ管理プログラムの構成	210
9.5.4	ジョブ処理の手順	211
9.5.5	スプールと多重ジョブ処理	213
9.6	データ管理	215
9.6.1	データ管理の機能	215
9.6.2	ファイル	216
9.6.3	ファイルの管理	218
9.6.4	ファイルのアクセス	220
9.7	タスク管理	223
9.7.1	タスクの概念	223
9.7.2	タスク管理の機能	225
9.7.3	タスク制御ブロック	226
9.7.4	タスクの実行状態とその遷移	227
9.7.5	タスクの生成と削除	228
9.7.6	タスク間通信 (タスクの同期)	229
9.7.7	競合とデッドロック	229
	演習問題	230



序論：計算機の技術の展開

1.1 計算の手順

コンピュータが誕生して 40 年を経た今日、その展開には目を見張るものがある。コンピュータとはいったいどんな機能をもった機械であろうか、基本的動作原理は何なのか、それがどのように変わって来ているか、またこれから将来どのように変わる可能性があるのかを、考えることはきわめて大切なことと思う。

コンピュータの動作原理を知る前に、人間による計算の手順を考えてみると、人間は紙と鉛筆で計算を行う。紙の上に書かれた情報は問題を解くアルゴリズムとデータである。人間の脳はアルゴリズムに基づいた計算を紙の上で実行していかなければならない。

さらに実行した計算の結果を紙の上に記録しておく。図 1.1 に人間による手計算の大まかな過程を示す。図 1.1 では、脳に

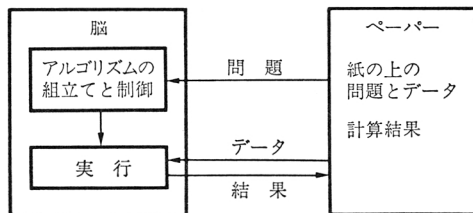


図 1.1 手計算の過程

二つの大きな機能を持たせている。一つは問題のアルゴリズムによってデータの処理を行うように制御することであり、他の一つは計算を実行することである。

図 1.1 の紙の上には問題とデータが与えられており、人間はこの問題を見

て、解決のためアルゴリズムを組み立て、実際の計算の手順を考え、データを計算し実行する。計算機の原理も図 1.1 と似ている。人間の脳に当たるのが図 1.2 の中央処理装置であり、紙に相当するのが入力・出力装置と記憶装置である。中央処理装置のプログラム制御装置は、記憶装置にあるプログラムの命令に従ってプログラムの命令を翻訳し、命令を実行させる制御部分である。

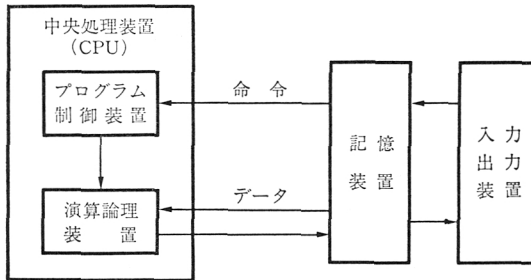


図 1.2 計算機の構成

演算論理装置はプログラムの命令を実行し、計算および論理演算を実行する装置部分である。また、入力・出力装置は計算するデータやプログラムを人間の理解しやすい言語や記号で入力および出力する部分であって、人間と計算機とのインターフェースである。人間による計算と計算機による計算の違いは、命令とデータの表現にある。人間は記号を含めた自然言語を理解し、10進計算を進めるのに対し、計算機は通常0と1の記号を用いる2進数の形式で行われる。しかし、人間も計算機も計算を実行するためには、それぞれ図 1.1 と図 1.2 に示すように、プログラム（思考の手順）を翻訳し実行する部分、プログラムやデータを記録しておく記憶部分、さらにこの記憶部分と処理装置間の情報の伝送の手段が必要となる。

1.2 電子計算機の技術の歴史

電子計算機の原型となる機械計算機の原理は Blaise Pascal (1623 ~ 1662 年) に始まる。Pascal の製作した機械は加算と減算を実行する機械的カウンタであり、歯車を組み合わせて作ったものである。歯車のダイヤルの表示とし

て0から9までの数が刻みこまれており、ダイヤルは $W = w_5w_4w_3w_2w_1w_0$ の10進6けたからなっている。もう一つのダイヤルは $W' = w_5'w_4'w_3'w_2'w_1'w_0'$ で、二つの歯車が連結している。 w_i' が k 刻みに回転すると、 w_i は同時に $w_i \pm k$ だけ回転する。Pascal の機械の新しさは、 w_i が9から0へ回転するとき、 w_{i+1} へ1のけた上げを行ったことである。このようにして8けたの加算機を実現した。1671年 Gottfried Leibniz (1646～1716年) は加減算に加えて乗算と除算のできる四則計算機を発明した。Charles Babbage (1791～1871年) は階差計数器と解析計数器を発明した。ここでの階差計数は数値表の作成を目的とした。しかし多くの関数の値は、この階差法の原理によって計算できる。

いま関数 $f(x)$ を

$$f(x) = \sum_{j=0}^n a_j x^j$$

とおく。 x の値を x_1, x_2, x_3, \dots とおき、おのおのの差分を Δx とおく。 $y_j = f(x_j)$ とすると、 y_j の第 i 番目の階差 $\Delta^i y_j$ は再帰的に

$$\Delta^0 y_j = y_j$$

$$\Delta^i y_j = \Delta^{i-1} y_{j+1} - \Delta^{i-1} y_j, \quad i \geq 1$$

と表される。 $\Delta^n y_j =$ 定数で、 $\Delta^i y_j = 0, i > n$ となる n が存在することから、 y_j の $(n+1)$ 個のゼロでない階差の値を知っていれば、 y_{i+1} の階差が

$$\Delta^i y_{j+1} = \Delta^i y_j + \Delta^{i+1} y_j$$

より計算できる。はじめ y_0 から出発して、 y_j の値を階差の計算をして、算出することができる。また Babbage の解析計数器は記憶装置部と演算装置部から構成されている。演算装置部は加減乗除のいずれかをカードで指定した。この解析計数器は現在の電子計算機にも共通する設計思想を備えていた。しかし、このような機械的計算は機械的部品の力学機構の性質を使って制御されていること、さらに機械的方法による情報伝達はきわめてやっかいで不安定であること、などの欠点がありこれをカバーするプログラム制御の計算機の実用化は電子管を使った計算機の出現まで100年以上も待たなければならなかった。

1.2.1 第1世代コンピュータ (1946～1954年)

最初の真空管を使った電子計算機は米国 ペンシルバニア大学で作られた ENIAC である。1946年に完成したコンピュータであるが、重さ 30t、18000本以上の真空管が使われていた。この計算機は入出力がカードであり、20個のアキュムレータ（累算器）と乗算器、除算器と関数テーブルを有していた。演算制御はスイッチやプラグイン接続で行われた。これを命令の系列（プログラム）という表現で行うこととし、コンピュータの記憶装置部にプログラムとデータを格納するという方式（stored-program concept）を提唱したのが米国の数学者 John von Neumann である。この方式に従い 1951年、EDVAC が動き出した。EDVAC の算術命令の形式は

$$A_1A_2A_3A_4Op$$

であり、 Op は加算、減算、乗算、除算の演算の命令である。主記憶上のアドレス A_1 とアドレス A_2 の内容を Op に従って実行し、アドレス A_3 に格納することを意味する。さらに A_4 がつぎに実行すべき命令のアドレスを指示している。また条件付き分岐命令として、

$$A_1A_2A_3A_4C$$

があり、アドレス A_1 の数が A_2 の数に等しいかあるいは大きければ、つぎの命令はアドレス A_3 を実行し、小さければつぎの命令はアドレス A_4 を実行する。第1世代コンピュータの構造（アーキテクチャ）は図 1.3 のように示される。

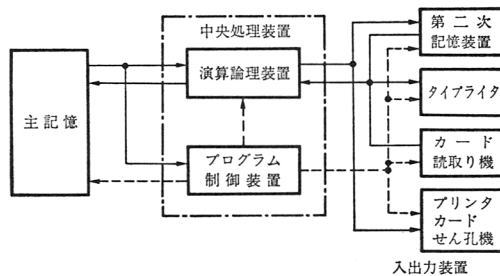


図 1.3 第1世代コンピュータのアーキテクチャ

まず計算機の中央処理装置が1箇所にまとめられている。また図1.3の実線はデータの流れを示し、破線はプログラム制御装置からの制御情報を示す。ただ第1世代のコンピュータでは主記憶装置と入出力装置の命令データの1語長をプログラム制御装置の制御によって情報伝達を行っている。1946年 von Neumann とそのグループはプリンストン高等研究所 (Institute for Advanced Studies) で、その頭文字をとった IAS コンピュータの開発に着手した。1語 40 ビットで主記憶は 4096 語である。EDVAC コンピュータが1語の命令に四つのアドレスの指定があったのに対し、IAS コンピュータでは図1.4のようにアドレスが一つである。また IAS コンピュータは機械を効率よく働かせるために多くのレジスタが採用されている。まず中央処理装置の中の 40 ビットのデータレジスタ (DR) があり、記憶装置部分からの CPU の語の転送に使われている。そして命令の実行中に、命令コードが格納されている命令のアドレスのためのアドレスレジスタ (AR) がある。

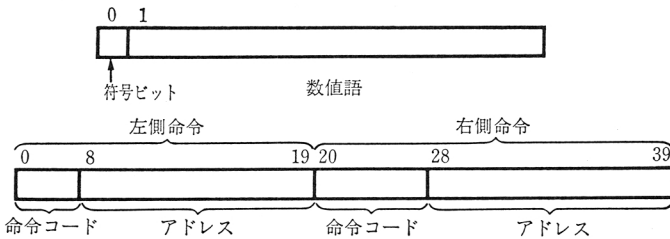


図 1.4 IAS コンピュータの命令データの表現形式

命令コードと結果の一時的保存のためにアキュムレータ (AC) と乗数、商レジスタ (MQ) がある。その他にバッファレジスタなどがある。

第1世代コンピュータに対する批評としてつぎのようなものがあげられる。

- ① アドレス修飾の命令が不十分であり、プログラムの実行が簡単迅速にできない、
- ② サブルーチン形式をとるようなリンケージ機能がない、
- ③ 浮動小数点演算方式のハードウェアがない、
- ④ 入出力命令が不十分であり、全体の効率を下げる結果となる、などである。

そして、真空管からトランジスタ技術に代わる第2世代コンピュータの時代の幕開けとなる。

1.2.2 第2世代コンピュータ (1955～1964年)

1948年、米国ベル研究所の Shockley らによって開発された半導体トランジスタが電子計算機の世界を大きく変える第一歩となった。この世代の代表的コンピュータは IBM 7094 である。1955年、IBM 社は真空管によるコンピュータ IBM 704 を開発し、これはインデックスレジスタと浮動小数点のハードウェアを有していたが、この後続機の IBM 709 は入出力プロセッサを有していた。IBM 709 をトランジスタ化した計算機が IBM 7094 である。したがって第1世代コンピュータにはない機能が付加されて来た。さらに記憶素子として陰極線管メモリ (cathode-ray tube)、遅延メモリ (delay-line memory) に代わってフェライトコアや磁気ドラムのような記憶装置が出現してきた。浮動小数点の演算回路が行きわたるようになり、計算機の手速が上がり

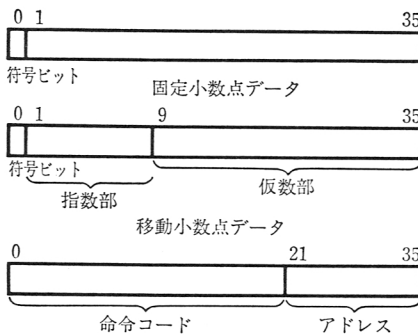


図 1.5 第2世代コンピュータ IBM 7094 の命令、データの表現形式

さらには人間が読み書きするのに理解しやすい FORTRAN, ALGOL および COBOL というプログラミング言語が生まれた。FORTRAN は 1954 年 IBM の Backus らが開発を始め、1957 年に完成した。また COBOL は 1959 年、米国の企業、国防省などの援助で完成された。

IBM 7094 コンピュータの命令、

データの表現形式を図 1.5 に示す。

第2世代コンピュータとしての IBM 7094 にはインデックスレジスタ XR (I) がついており、これがアドレスレジスタ (AR) を修飾し、新しいアドレス

$$AR - XR(I)$$

を指定することによって実効アドレスの内容を主記憶から取り出すことができる。第1世代コンピュータ IAS ではアドレスの修飾レジスタを持っておらず、これらのアドレス修飾も主記憶上で計算していた。このため効率が悪く、

索引

【A】

アドレス 76
アドレスバス 77, 118
アドレス変換表 156
アドレッシング方式 82
アクセス時間 135
アプリケーションプログラ
ム 208
アセンブラ 176
アセンブリ言語 176
アレイ型除算回路 72
アレイ型乗算回路 69
ALGOL 6
AND (論理積) 36
ALU 64
APU 62
ASCII コード 33
Availability(可用性) 200

【B】

バッチ処理 198, 201
バッファ記憶 8
パケットラック 191
バーストモード 131
ベイチ・カルノー図法 45
ベクトル割込み方式 125
ベン図 38
ポリューム 217
分配律 37
分散処理 203
ブロック 217

物理アドレス空間 154
物理レコード 217
BASIC 184
BCD 符号 28
BIOS 215
Blaise Pascal 2
Booth の方法 24
Braun の配列乗算器 69

【C】

遅延メモリ 6
逐次アクセス 134
直列加算器 59
直接アドレッシング 82
直接編成ファイル 220
中央処理装置 74
C 186
carry generate function 58
carry look ahead circuit 57
carry propagate function 58

CCD 151
CPU 74
charles Babbage 3
COBOL 6, 186
CTSS 202

【D】

第1世代コンピュータ 4
第2世代コンピュータ 6
第3世代コンピュータ 8

第4世代コンピュータ 13
第5世代コンピュータ 14
第6世代コンピュータ 15
ダイナミック MOS RAM 139
ダイレクトメモリアクセス 127

データバス 77, 118
データベース 208
データ管理 205, 215
データ転送命令 81
デッドロック 239
ディレクトリ 218
ディスクキャッシュ 154
ド・モルガンの定理 38
動的アドレス変換 156
D フリップフロップ 53

【E】

エミュレーション 13
演算部 80
演算回路 55
演算命令 81
EBCDIC コード 33
EDVAC 4
ENIAC 4
EPROM 141
E²PROM 141

【F】

ファイル 216

ファイル管理 215
 ファームウェア 12
 浮動小数点 26
 復帰命令 107
 フラグメンテーション (断片化) 158
 フレキシブルディスク 143
 フリップフロップ 48
 フロッピーディスク 144
 布線論理 113
 FCB 222
 FORTRAN 6, 182

【G】

外部メモリ 133
 言語処理 205
 減算回路 59
 擬似命令 176
 グレイ符号 31
 Gottfried Leibniz 3

【H】

ハードウェア資源 196
 背景ジョブ 209
 排他的論理和 91
 半加算器 55
 並列加算器 57
 光ディスクメモリ 147
 光磁気メモリ 148
 補数表現 18

【I】

1の補数 8
 陰極線管メモリ 6
 インタフェース回路 120
 インタプリタ 184

インタプリタ方式 207
 一括処理 198, 199, 201
 IAS コンピュータ 5
 IBG 145
 integrity (保安性) 200
 IOCS 215
 IRG 146

【J】

磁気バブルメモリ 150
 磁気ディスク 142
 人工知能 14
 人的資源 196
 10進数 16
 実時間処理 202
 ジョブ 209
 ジョブ管理 205
 ジョブ制御言語 200, 210
 ジョブスケジューラ 210, 211
 ジョブステップ 209
 情報資源 196
 除算回路 70
 乗算回路 67
 状態レジスタ 66
 状態遷移 227
 順編成ファイル 219
 順序回路 48
 順序制御命令 81
 循環アクセス 134
 循環けた上げ 22
 JCL 210
 JK フリップフロップ 52
 John von Neumann 4

【K】

加減算回路 55
 加減算器 60

階差法の原理 3
 書込み専用ファイル 216
 課金 209
 間接アドレッシング 82
 加算回路 55
 仮想記憶方式 154
 級数 26
 カタログ 218
 計算機ネットワーク 203
 けた上げ伝搬関数 58
 けた上げ発生関数 58
 けた上げ先見回路 57
 結合律 37
 記憶保護 161
 機械語 165
 機械命令 77
 機器番号 119, 121
 木構造ディレクトリ 218
 基数 16
 交換律 37
 高級言語 181
 コマンド 210
 コマンドプロセッサ 201
 コンパイラ 181, 207
 コンパイルリンクアンドゴ
 - 182
 コントロールバス 77, 118
 コール命令 108
 故障率 199
 固定小数点 26
 区分編成ファイル 220
 組合せ回路 47
 クロック発生器 80
 キャリーフラグ 66
 キャッシュ記憶 6
 キャッシュメモリ 153

【L】

LISP 189

LSD 16
LSI, 超 LSI 13

【M】

マイクロ命令 10, 113
マイクロプログラミング 9
マイクロプログラム制御 113
マイクロ操作 85
マルチプレクサチャネル 130
マルチタスク処理 225
マスタスケジューラ 210
命令エンクゼクション 77
命令デコーダ 79
命令デコード 77
命令フェッチ 77
命令レジスタ 79
命令サイクル 77
メモリアドレスレジスタ 79
メモリアクセスタイム 76
メモリ直接アドレッシング 82
メモリ間接アドレッシング 83
メモリバッファレジスタ 79
MASS 147
MSD 16
MTBF 199

【N】

内部メモリ 133
内容アクセス記憶 160

ナノプログラム 115
ニモニック 83, 176
2の補数 18
2進化 10 進加算回路 61
2進数 16
ノーマルモード 131
ノンシェアブルファイル (非共用ファイル) 216
入力命令 119
入出力命令 81
入出力プロセッサ 7
入出力制御装置 130
NOT (否定) 36

【O】

オーバフローフラグ 67
オーバヘッド 203
オブジェクトモジュール 207
オブジェクトプログラム 179
オブジェクトライブラリ 180
オフライン処理 202
オンラインリアルタイム処理 199, 202
オペランド 81
オペランド部 167
オペランドフェッチ 77
オペレーティングシステム 196
応答時間 198
親タスク, 子タスク 228
operational code 81
OR (論理和) 36

【P】

パリティフラグ 73

ページング方式 155
ポーリング方式 125
プログラムカウンタ 79
プログラム格納方式 4
プログラム制御方式 118, 123
プロセス 225
program status word 66
PROLOG 190

【R】

ランダムアクセス 134
ランダムアクセスファイル 216
レジスタ直接アドレッシング 82
レジスタ間接アドレッシング 82
レジスタトランスファ論理 85
レコード 217
連想記憶 160
リアルタイム処理 202
リモートバッチ処理 202
リンケージェディタ 207
リターン命令 107
ロードモジュール 207
論理アドレス空間 154
論理演算回路 63
論理関数 41
論理関数の簡単化 45
論理レコード 217
ロールアウト 157
r進教 16
RAS 200
RASIS 199
reliability (信頼性) 199

ROM 140
RS フリップフロップ 50
RWM 134

【S】

サービスプログラム 207
サブルーチン 107
サイクル時間 135
サインフラグ 67
索引順編成ファイル 219
3余り符号 30
算術演算回路 62
算術論理演算回路 63
世代ファイル 219
セグメンテーション方式 155
制御部 79
制御記憶 10
制御命令 82
制御信号生成回路 79
正規化 26
セレクトチャンネル 130
シークンシャルアクセスファイル 216
シェアラブルファイル (共用ファイル) 216
真理値表 36
指数 26
システム生成 207
システム資源 196
装置管理 215
ソフトウェア資源 196
即値アドレッシング 82
操作コード 167
操作コード部 81
ソースプログラム 178
相転移型メモリ 149
垂直型マイクロプログラム

垂直型マイクロプログラム 198
115
水平型マイクロ命令 10
水平型マイクロプログラム方式 115
スーパーバイザコール 201
スーパーバイザコール命令 162
スーパーバイザレベル 161
スプーリング 214
スプール 213
スルーブット 197
スタティックバイポーラRAM 137
スタティック MOS RAM 138
スタック 108
スタックポインタ 108
ステートデコーダ 80
ステートカウンタ 80
主加法標準形 42
修理率 200
出力命令 120
security (機密性) 200
serviceability (保守性) 200

Shockley 6
SPOOL 214
SVC 命令 220
SVC 割込み 220

【T】

多重ジョブ処理 214
多重仮想記憶方式 159
多重プログラム 198
多重プロセッサ処理 202
多重タスク処理 214
ターンアラウンドタイム

タスク 223, 225
タスクディスパッチ 227
タスク管理 205
タスク間通信 214, 229
特権命令 162
特権レベル 161
特性方程式 51
T フリップフロップ 53
TSS 202

【V】

VTOC 217

【W】

ワード 75
ワード長 75
割込み 124
ウィンチェスタディスク 144
Wilkes 9
wired logic 113

【Z】

全加算器 55
ゼロフラグ 67
絶対値表現 18

【Y】

呼出し命令 108
読出し記憶 10
読出し専用記憶 140
読取り専用ファイル 216
ユニフィケーション 191
ユーティリティ 207

— 著者略歴 —

すずき ひさよし
鈴木 久喜

1956年 東北大学工学部通信工学科卒業
1961年 東北大学大学院博士課程修了, 工学博士
1961年 東北大学助手
1964年 東北大学助教授
1975年 静岡大学教授
1997年 静岡大学名誉教授
1997年 東北文化学園大学教授
現在に至る

いし い なお ひろ
石井 直宏

1963年 東北大学工学部電気工学科卒業
1968年 東北大学大学院博士課程修了, 工学博士
1968年 東北大学助手
1975年 名古屋工業大学助教授
1982年 名古屋工業大学教授
2003年 名古屋工業大学名誉教授
2003年 愛知工業大学教授
現在に至る

いわ た あきら
岩田 彰

1973年 名古屋大学工学部電気工学科卒業
1975年 名古屋大学大学院修士課程修了
1975年 名古屋工業大学助手
1981年 工学博士
1985年 名古屋工業大学助教授
1993年 名古屋工業大学教授
2003年 名古屋工業大学副学長
現在に至る

基礎電子計算機

Fundamentals of Digital Computer System

© Suzuki, Ishii, Iwata 1988

1988年5月30日 初版第1刷発行
2003年7月10日 初版第13刷発行

検印省略

著者 鈴木 久喜
石井 直宏
岩田 彰
発行者 株式会社 コロナ社
代表者 牛来辰巳
印刷所 駕籠町印刷所

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 4-339-00134-1
Printed in Japan

(壮光舎印刷, 愛千製本所)



無断複写・転載を禁ずる

落丁・乱丁本はお取替えいたします