



電子情報通信レクチャーシリーズ C-9

コンピュータ アーキテクチャ

電子情報通信学会◎編

坂井修一 著

コロナ社

▶電子情報通信学会 教科書委員会 企画委員会◀

- 委員長 原島 博 (東京大学教授)
- 幹事 (五十音順)
 - 石塚 満 (東京大学教授)
 - 大石 進一 (早稲田大学教授)
 - 中川 正雄 (慶應義塾大学教授)
 - 古屋 一仁 (東京工業大学教授)

▶電子情報通信学会 教科書委員会◀

- 委員長 辻井重男 (情報セキュリティ大学院大学学長
中央大学研究開発機構教授
東京工業大学名誉教授)
- 副委員長 長尾 真 (情報通信研究機構理事長
前京都大学総長
京都大学名誉教授)
- 幹事長兼企画委員長 神谷武志 (大学評価・学位授与機構教授
東京大学名誉教授)
- 幹事 (五十音順)
 - 原島 博 (東京大学教授)
 - 石塚 満 (東京大学教授)
 - 大石 進一 (早稲田大学教授)
 - 中川 正雄 (慶應義塾大学教授)
 - 古屋 一仁 (東京工業大学教授)
- 委員 122名

(2004年4月現在)

刊行のことば

新世紀の開幕を控えた1990年代、本学会が対象とする学問と技術の広がりとお行きは飛躍的に拡大し、電子情報通信技術とほぼ同義語としての“IT”が連日、新聞紙面を賑わすようになった。

いわゆるIT革命に対する感度は人により様々であるとしても、ITが経済、行政、教育、文化、医療、福祉、環境など社会全般のインフラストラクチャとなり、グローバルなスケールで文明の構造と人々の心のありさまを変えつつあることは間違いない。

また、政府がITと並ぶ科学技術政策の重点として掲げるナノテクノロジーやバイオテクノロジーも本学会が直接、あるいは間接に対象とするフロンティアである。例えば工学にとって、これまで教養的色彩の強かった量子力学は、今やナノテクノロジーや量子コンピュータの研究開発に不可欠な実学的手法となった。

こうした技術と人間・社会とのかかわりの深まりや学術の広がりを踏まえて、本学会は1999年、教科書委員会を発足させ、約2年間をかけて新しい教科書シリーズの構想を練り、高専、大学学部学生、及び大学院学生を主な対象として、共通、基礎、基盤、展開の諸段階からなる60余冊の教科書を刊行することとした。

分野の広がりに加えて、ビジュアルな説明に重点をおいて理解を深めるよう配慮したのも本シリーズの特長である。しかし、受身的な読み方だけでは、書かれた内容を活用することはできない。“分かる”とは、自分なりの論理で対象を再構築することである。研究開発の将来を担う学生諸君には是非そのような積極的な読み方をしていただきたい。

さて、IT社会が目指す人類の普遍的価値は何かと改めて問われれば、それは、安定性とバランスが保たれる中での自由の拡大ではないだろうか。

哲学者ヘーゲルは、“世界史とは、人間の自由の意識の進歩のことであり、…その進歩の必然性を我々は認識しなければならない”と歴史哲学講義で述べている。“自由”には利便性の向上や自己決定・選択幅の拡大など多様な意味が込められよう。電子情報通信技術による自由の拡大は、様々な矛盾や相克あるいは摩擦を引き起こすことも事実であるが、それらのマイナス面を最小化しつつ、我々はヘーゲルの時代的、地域的制約を超えて、人々の幸福感を高めるような自由の拡大を目指したいものである。

学生諸君が、そのような夢と気概をもって勉学し、将来、各自の才能を十分に発揮して活躍していただくための知的資産として本教科書シリーズが役立つことを執筆者らと共に願っ

ii 刊 行 の こ と ば

ている。

なお、昭和 55 年以來発刊してきた電子情報通信学会大学シリーズも、現代的価値を持ち続けているので、本シリーズとあわせ、利用していただければ幸いである。

終わりに本シリーズの発刊にご協力いただいた多くの方々に深い感謝の意を表しておきたい。

2002 年 3 月

電子情報通信学会 教科書委員会
委員長 辻 井 重 男

まえがき

いうまでもなくコンピュータはIT時代の主役である。こんにちのコンピュータは、サーバやパソコンだけでなく、携帯電話、テレビ、エアコン、オーディオ、自動車など、情報を整形・伝達したり、機器を制御したり、画像を表示したりする、あらゆるものに入っている。コンピュータの仕組みを理解することは、情報処理の研究者・技術者だけでなく、理科系のほとんどの人々に（場合によっては文科系の人にも）必要なことであろう。

コンピュータアーキテクチャとは、ソフトウェアとハードウェアのインタフェースのことである。より簡単にいえば、ソフトウェアの立場から見てハードウェアがどう働くかを記述したものである。

本書は、初めてコンピュータアーキテクチャを学ぶ人を対象とし、アーキテクチャの基本を学んでもらうことを主目的としている。本書を読むのには、多量の予備知識を必要としてはいないが、論理回路の入門書を1冊、座右に置いてときどき参照していただきたい。すでに論理回路を学ばれた人には、これも不要であろう。

本書の特徴は、1本の電線からコンピュータまで、その本質を最も分かりやすく簡明に記したことにある。枝葉は個々のマイクロプロセッサのマニュアルを読めばいくらでも学べる。本書では、コンピュータがなぜプログラムを実行できるのか、その一点が分かることを主目的にして、徹頭徹尾単純に書いたつもりである。その上で、性能向上の原点であるパイプライン処理、命令レベル並列処理、記憶階層について説明した。章末の「理解度の確認」もこうした基本の理解を助けるためのものであり、読者はできるだけ全問題を解くことを試みてほしい。

本書を順番に読んでいただければゼロから着実に知識が身につくだろう。あるいは既にいくらか知識のある人は、未知のことがらの書かれている章からはじめて、ときどき前の章を参照する、というやりかたで読んでいただければ十分である。

現実のコンピュータはいまも指数関数的に発展し続けている。しかし、最先端のハイエンドマイクロプロセッサも、本書にしっかり記述されている基礎技術の上にあるとあってよい。読者が本書をきっかけとして、最先端のアーキテクチャ技術を学ばれるところまで進まれることを、著者として切望する次第である。

本書を著すにあたってお世話いただいた電子情報通信学会教科書委員会の原島 博先生、石塚 満先生、村岡 洋一先生にお礼を申し上げたい。本書がほぼ予定どおり刊行できるの

はこの方々のおかげである。また、本書を著すにあたっては細心の注意を払ったつもりであるが、なお内容・字句に不十分なところがあるかもしれない。この点、読者諸賢のご叱責を賜ればありがたい。特に本シリーズの眼目である「図解」の図の部分についてのご意見をうかがわせていただければありがたく思う。

2004年2月

坂井 修一

目 次

1. はじめに

1.1	デジタルな表現	2
1.1.1	1本の線から	2
1.1.2	n 本の線にしてみよう	3
1.1.3	負の数	4
1.1.4	実数	4
談話室	10本の指で数を表現	5
1.2	計算する	6
1.2.1	計算とはなにか	6
1.2.2	1ビットの加算	7
1.2.3	n ビット加算器	8
1.2.4	減算の実現	9
1.2.5	ALU	9
1.3	計算のサイクル	10
1.3.1	フリップフロップ	11
1.3.2	レジスタ	12
1.3.3	レジスタとALUの結合	13
	本章のまとめ	14
	理解度の確認	14

2. データの流れと制御の流れ

2.1	主記憶装置	16
2.1.1	レジスタとALUだけでは計算はできない	16
2.1.2	主記憶装置	16
2.1.3	メモリの構成	17

2.1.4	メモリの分類	19
2.1.5	レジスタファイル	21
2.1.6	主記憶装置の接続	22
2.2	命令とはなにか	23
2.2.1	命令	23
2.2.2	命令実行の仕組み	24
2.2.3	算術論理演算命令の実行サイクル	25
2.2.4	メモリ操作命令の実行サイクル	26
2.3	シーケンサ	27
2.3.1	シーケンサとはなにか	27
2.3.2	条件分岐命令の実行サイクル	29
	本章のまとめ	30
	理解度の確認	30

3. 命令セットアーキテクチャ

3.1	命令の表現形式とアセンブリ言語	32
3.1.1	操作とオペランド	32
3.1.2	命令の表現形式	32
3.1.3	命令フィールド	33
3.1.4	アセンブリ言語	35
3.2	命令セット	35
3.2.1	算術論理演算命令	35
3.2.2	データ移動命令	38
3.2.3	分岐命令	39
3.3	アドレッシング	41
3.3.1	アドレッシングの種類	41
3.3.2	バイトアドレッシングとエンディアン	43
3.3.3	ゼロレジスタと定数の生成	43
3.4	サブルーチンの実現	44
3.4.1	サブルーチンの基本	44
3.4.2	サブルーチンの手順	45
3.4.3	スタックによるサブルーチンの実現	46

3.4.4 サブルーチンのプログラム	47
談話室 CISC と RISC	48
本章のまとめ	49
理解度の確認	49

4. パイプライン処理

4.1 命令パイプライン	52
4.1.1 パイプラインの原理	52
4.1.2 命令パイプラインの基本	53
4.1.3 基本命令パイプラインの実現	54
4.2 基本命令パイプラインの阻害要因	56
4.2.1 オーバヘッド	56
4.2.2 ハザード	57
4.2.3 構造ハザード	57
4.2.4 データハザード	58
4.2.5 制御ハザード	59
4.3 ハザードの解決法	60
4.3.1 フォワーディングによるデータハザードの解消	60
4.3.2 命令アドレス生成のタイミング	61
4.3.3 遅延分岐	62
4.3.4 分岐予測	63
4.3.5 命令スケジューリング	66
本章のまとめ	67
理解度の確認	68

5. キャッシュと仮想記憶

5.1 記憶階層	70
5.1.1 命令パイプラインとメモリ	70
5.1.2 記憶階層と局所性	71
5.1.3 透過性	72

5.2	キャッシュ	72
5.2.1	キャッシュとはなにか	73
5.2.2	ライトスルーとライトバック	74
5.2.3	ダイレクトマップ形キャッシュの機構と動作	75
5.2.4	キャッシュミス	77
5.2.5	フルアソシアティブ形キャッシュと セットアソシアティブ形キャッシュ	78
5.2.6	キャッシュの入った CPU	80
5.2.7	キャッシュの性能	82
5.3	仮想記憶	83
5.3.1	仮想記憶とはなにか	84
5.3.2	仮想記憶の構成	84
5.3.3	ページフォールト	85
5.3.4	TLB	86
5.4	メモリアクセス機構	87
5.4.1	キャッシュと仮想記憶	87
5.4.2	メモリアクセス機構	89
	談話室 透過性と互換性	90
	本章のまとめ	91
	理解度の確認	92

6. 命令レベル並列処理とアウトオブオーダー処理

6.1	命令レベル並列処理	94
6.1.1	並列処理	94
6.1.2	並列処理パイプライン	95
6.2	VLIW	96
6.2.1	VLIW プロセッサの構成と動作	96
6.2.2	VLIW の特徴	97
6.3	スーパスカラ	97
6.3.1	スーパスカラプロセッサの構成と動作	98
6.3.2	並列処理とハザード	99
6.3.3	VLIW とスーパスカラの比較	100

6.4	静的最適化	100
6.4.1	機械語プログラムと命令間依存性	100
6.4.2	ループアンローリング	101
6.4.3	ソフトウェアパイプライン	104
6.4.4	トレーススケジューリング	105
6.5	アウトオブオーダー処理	106
6.5.1	アウトオブオーダー処理とはなにか	106
6.5.2	データ依存再考	108
6.5.3	アウトオブオーダー処理の機構	110
6.6	レジスタリネーミング	111
6.6.1	ソフトウェアによるレジスタリネーミング	111
6.6.2	ハードウェアによるレジスタリネーミング(1) —マッピングテーブル—	112
6.6.3	ハードウェアによるレジスタリネーミング(2) —リオーダーバッファ—	114
6.7	スーパースカラプロセッサの構成	116
6.7.1	アウトオブオーダー処理を行うプロセッサの構成	116
6.7.2	プロセッサの性能	117
	本章のまとめ	119
	理解度の確認	120

7. 入出力と周辺装置

7.1	周辺装置	122
7.1.1	周辺装置の分類	122
7.1.2	液晶ディスプレイ	123
7.1.3	磁気ディスク	125
7.2	入出力の機構と動作	127
7.2.1	ハードウェアインタフェース	127
7.2.2	データ転送の手順	128
7.2.3	割込みの調停	128
7.2.4	DMA	130
7.3	例外処理	132

7.3.1 例外の要因	132
7.3.2 例外処理の手順	132
本章のまとめ	133
理解度の確認	134
引用・参考文献	135
理解度の確認；解説	136
あ と が き	142
索 引	143

1

はじめに

本章では、「デジタル信号とはなにか」から、データの表現、計算するための基本機構である ALU とレジスタの結合までを学ぶ。コンピュータの基礎の基礎であり、既にご存知の学習者とはばしてもよいが、少しでも不安のある読者は一読されることを勧めたい。

1.1 デジタルな表現

データの表現には、アナログ的なやりかたとデジタル的なやりかたがある。現在のほとんどのコンピュータはデジタルな表現をとっている。 n 本の線で n 桁の2進数が表現され、これを移動・整形することで演算が進められる。 n 本の線で、正の整数だけでなく、負の数や実数を表現することができる。

1.1.1 1本の線から

初めに、1本の電線からはじめよう（図1.1）。この電線上に信号が乗る。信号を書き込んだり、遠くへ運んだりたり、読み出したりすることで、データが伝えられる。

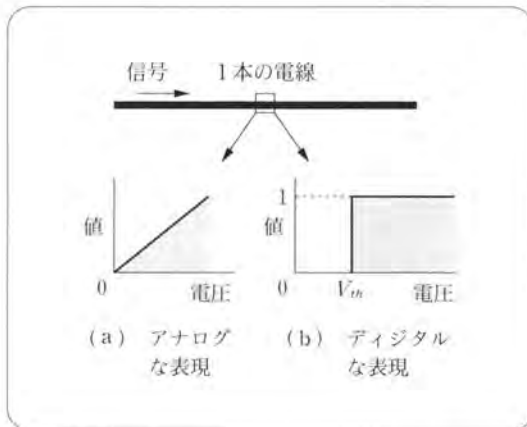


図 1.1 1本の電線と信号

電線上の信号が電圧で表されるとする。ここで、信号の値が電圧に比例したものであるとすると、1本の線で一つの信号が表される。これを、**アナログ** (analogue) な表現と呼ぶ（図(a)）。これに対して、信号の値は、高電圧か低電圧かによって1か0の2値を表現したものと考えることもできる（図(b)）。これを、**デジタル** (digital) な表現と呼ぶ。図(b)では、電圧 V_{th} をしきい値 (threshold) として、これより低い電圧ならば値は0、これより高い電圧ならば値は1と定めている。

アナログな表現は、1本の線で一つの量を表現できる利点があるが、精度を高くするのがむずかしい、雑音に弱い、「記憶」がむずかしい、などの欠点がある。これに対してディジ

タルな表現は、一つの量を表現するのに複数の線が必要になるが、雑音に強い、汎用性が高く、記憶も高密度で正確、などの利点がある。

こんにち、ほとんどのデータは一度デジタルな表現に整形されてから用いられる。このようなデジタルなデータを記憶したり、整形したり、計算したり、入力したり、出力したりする主役が本書で登場するデジタルコンピュータ（以下、単にコンピュータ）である。

1.1.2 n 本の線にしてみよう

データ表現に2進法をとった場合、1本の線で表現できるのは、0か1の2種類のデータだけである。ここに1本の線の情報量を1ビット (bit) と呼ぶ。2進数で n 桁の数を表現するには、 n 本の線を使う（図1.2）。これで n ビットが表現される。ある2進数を10進数で表すとどうなるだろうか。両者の変換のやりかたについては論理回路の教科書にゆずるとして、表1.1に4桁の2進数と10進数の対応表を示す。このように、2進数はある数を表現するのに多くの桁を必要とするが、一つひとつの桁は0か1となって単純である。

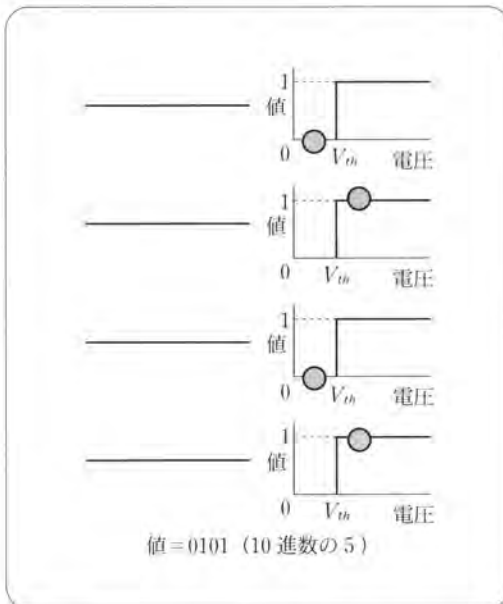


図1.2 n 本の電線（この例では $n=4$ ）

表1.1 2進数と10進数の対応表

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

基準となる桁数 n はコンピュータによって異なるが、近年、パソコンやワークステーションで使われているマイクロプロセッサ (microprocessor) の場合、これは32または64であることが多い。組込み形CPU (central processing unit) の場合、これは8や16のこともある。基準となる n ビットのデータのことを語 (word) と呼び、 n を語長 (word

length) と呼ぶ。 n 本の線では、0 から $2^n - 1$ までの数を表すことができる。

1.1.3 負の数

われわれの扱う数は、正の整数ばかりではない。負の数もあるし、実数もある。

2進数で負の数を表すためには、補数 (complement) 表示を用いる。補数表示とは、最上位のビットを符号を表すものとし、これが0のとき正の数、1のとき負の数とみなすという数の表現法である。補数表示によって、電子計算機の中では、加減算はすべて正の加算とわずかな補正だけで行えるようになる。現在のコンピュータでは、2の補数 (2's complement) によって負の数を表す (図 1.3)。

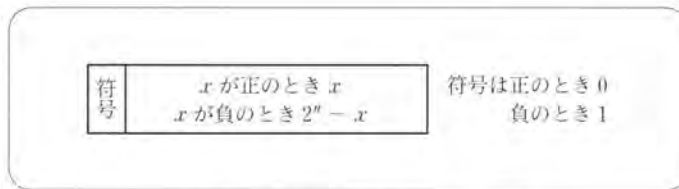


図 1.3 2の補数による負の数の表現

2の補数表示では、負の数 $-x$ を表すのに、 $2^n - x$ を用いる。2の補数は、 x の各桁の1と0を反転し、結果に1を加えたものとなる。

いま、4桁の数を例として考えると、 -6 の2の補数表示は1010となる。

2の補数表示をとった場合、 -2^{n-1} から $2^{n-1} - 1$ までの数を表すことができる。

1.1.4 実数

次に実数の表現法について学ぶ。実数の表現法には、大きく分けて次の二つがある。

[1] 固定小数点による表現 整数の表現と同じであるが、何桁目かに小数点があると約束しておく。特別な回路を用意する必要はないが、演算 (特に乗算と除算) をするたびに小数点の位置合わせのためのシフト (shift, 桁移動) が必要になる。シフトはプログラムがプログラムしてやらなくてはならない。

[2] 浮動小数点による表現 符号、数値、桁数を決められたそれぞれビット数で表現する。通常、演算のために特別な回路を用意する。そうすれば、演算に際して、プログラムによる補正は不要となる。

図 1.4 に、32ビットで有効数字23桁、2進数で ± 127 桁を浮動小数点によって表現したものを示す。

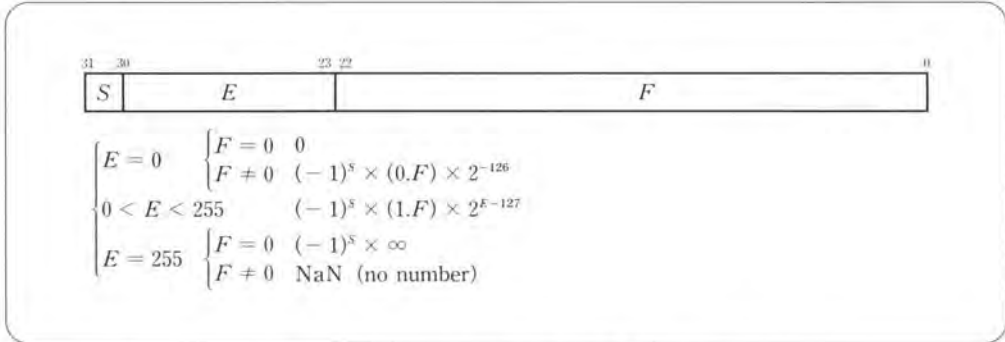


図 1.4 32 ビットの浮動小数点による実数の表現

固定小数点，浮動小数点のどちらをとるにしても，有効桁数以上の精度で実数を表現することはできないため，これを超える数については近似値で表す．近似によって生じる誤差については，プログラムを作るときに十分に神経を使わなければならない．

☒ 談 話 室 ☒

10本の指で数を表現 人間は10本指があるから10進数で物を数えるようになったといわれる．しかし，10本も指を使って， $\log_2 10$ ビットの情報量しか表さないのは，いささかもったいない話である．1本の指の曲げ伸ばしの情報量は1ビットであり，10本あれば10ビット，すなわち1024個の数が表現できるはずである．

実際に，2進数の0から1023までを手の指で表現することができる．手のひらを上にして両方の手を出し，指を1本も立てない（両手ともにじゃんけんのグーの）状態を0とする．ここから右手の親指だけを立てた状態を1，右手の人差し指を立てた状態を2（2進数の10），右手の親指と人差し指を同時に立てた状態を3（2進数の11），右手の中指を立てた状態を4，と数えていけば，指を10本すべて立てた状態で1023を表すことになる．図1.5(a)は5を，図(b)は801を表している．



(a) 5



(b) 801

図 1.5 指による2進数の表現

あとがき

本書では、コンピュータの基本構成と動作原理について述べた。読み通した読者は、コンピュータの中身が何であるのか、ご理解いただけたと思う。

高度な投機処理、ベクトルパイプライン、マルチプロセッサ、省電力アーキテクチャなどについては触れなかった。これらは、先進的なアーキテクチャの教科書や論文によって学んでほしい。同時にコンパイラやOSの技術についても深い知見を身につけてほしい。

コンピュータの進歩を支えているのは、半導体デバイス技術の進歩やコンパイラなどソフトウェア技術の発展とともに、アーキテクチャの技術である。アーキテクチャとは、命令セットの設計と構成法を指す。

アーキテクチャの設計者を**アーキテクト**という。この世界には英雄というべきコンピュータアーキテクトが何人もおり、私なども、彼らを畏敬と感謝の気持ちをもって語ることが大好きな1人である。ただし、この教科書では彼らの名前を1人もあげなかった。人や製品の実名をあげるとバイアスが掛かって、記述の純度が下がる。それを怖れたからだ。

しかし、次のことは強調しておかなければならないだろう。IT全盛の時代にあって見失いがちであるが、コンピュータアーキテクチャは天与のものではない。ここ半世紀の技術者・研究者たちが知恵を振り絞り、究極の手間をかけて成ったものである。コンピュータの専門家をめざす諸君は、アーキテクチャを固定したものと考えてるのではなく、毎年毎月大きな改良が加えられ、ときに革命が起こる分野と理解してほしい。コンピュータアーキテクトをめざす諸氏は、本書を改良・革命のための最初の一步としていただければ幸甚である。

【あ】

アウトオブオーダー完了106
 アウトオブオーダー実行106
 アウトオブオーダー処理106
 アセンブリ言語35
 アドレッシング41
 アドレス17
 アドレス線18
 アナログな表現2
 アービタ129
 アーム126

【い】

依存関係58
 イメージスキャナ122
 インオーダー処理106
 インデックス75

【え】

エイリアス88
 液晶123
 液晶ディスプレイ123
 エッジトリガ形Dフリップ
 フロップ11
 演算結果フラグ29
 演算実行53

【お】

オーバーヘッド56
 オペランド32

【か】

解釈25
 返り値45
 書込み17
 書込み可能ビット85
 加減算器9
 加算7
 仮想アドレス84, 85
 仮想記憶83
 仮想ページアドレス85
 カーネルプログラム133

【き】

記憶装置11

キーボード122
 基本ブロック105
 基本命令パイプライン53
 逆依存108
 キャッシュ73
 キャッシュブロック74
 キャッシュミス77
 キャッシュミス率78
 キャッシュライン74
 競合性ミス77
 共通命令63
 局所性71

【く】

空間的局所性72
 組合せ回路6
 組合せ論理回路6
 クロック11

【け】

計算17
 桁上げ7
 結果の格納53
 減算9

【こ】

語3
 構造ハザード57, 99
 互換性90
 語長3
 固定小数点4
 コーラセーブ方式45
 コーリセーブ方式45
 コンピュータ3

【さ】

最下位のバイト43
 最上位のバイト43
 サブルーチン44
 算術演算命令35
 算術論理演算命令24
 算術論理ユニット9

【し】

時間的局所性72
 しきい値2

磁気ディスク122, 125
 シーク126
 シーク時間126
 シーケンサ27
 実行時間52
 実行履歴105
 実数4
 シフト36
 シフト命令37
 ジャンプ命令28
 周辺装置122
 主記憶80
 主記憶装置17
 10進数3
 出力依存108
 消去64
 条件分岐命令28, 39
 乗算器36
 初期参照ミス77
 除算器36
 ショートカット60
 シリンダ126
 真理値表6

【す】

スタック46
 スタックポインタ46
 ステージ53
 ストア命令38
 ストール57
 スーパスカラ98
 スループット52

【せ】

制御依存59
 制御線18
 制御ハザード59, 100
 制御フローグラフ105
 制御用フラグビット85
 生産者-消費者58
 静的最適化100
 セクタ126
 セット79
 セットアソシアティブ形
 キャッシュ79
 ゼロレジスタ43

【そ】

操作	32
操作コード	24, 32
操作の対象	32
即値	32
ソースオペランド	32
ソフトウェアパイプラインング	104

【た】

大域履歴レジスタ	65
ダイレクトマップ形キャッシュ	75
タグ	75

【ち】

遅延分岐	62
置数器	12
チャンネルプロセッサ	131
直列形物理アドレスキャッシュ	87

【て】

デジタルコンピュータ	3
デジタルな表現	2
デジチェーン	129
ディスク	125
デコーダ	18
デコード	25
デスティネーションオペランド	32
データ依存	58
データキャッシュ	80
データ線	18
データ遅延	108
データハザード	58, 99
データメモリ	24

【と】

透過性	72
動的スケジューリング	106
トラック	126
トレース	105
トレーススケジューリング	105

【な】

74181形ALU	9
-----------	---

【に】

2進数	3
2の補数	4
入出力プロセッサ	131
2レベル適応形予測器	65

【は】

バイトアドレッシング	43
パイパシング	60
パイプライン	52
パイプラインハザード	57
パイプラインレジスタ	55
ハザード	57
番地	17

【ひ】

引き数	45
ビッグエンディアン	43
ビット	3
ヒューズROM	19

【ふ】

フィールド	24
フォワードイング	60
ブックキーピング	105
ブッシュ	46
物理アドレス	84
物理ページアドレス	85
物理レジスタアドレス	114
浮動小数点	4
浮動小数点演算器	36
負の数	4
ブライオリティエンコーダ	129
フラッシュ	64
フラッシュメモリ	20
ブランチ命令	28
フリップフロップ	11
プリンタ	122
フルアソシアティブ形キャッシュ	78
フロー依存	108
プログラム	23
プログラムカウンタ	27
プログラム格納形コンピュータ	27
プロセッサの性能	117
プロファイル	105
分岐命令	39
分岐予測	64
分岐履歴テーブル	65

【へ】

並列形物理アドレスキャッシュ	88
並列処理	94
ページ	84
ページテーブル	84
ページテーブルレジスタ	85
ページ内オフセット	85

ページフォールト	85
ヘッド	125

【ほ】

補数表示	4
ポップ	46
ポート	22
ポーリング	128

【ま】

マイクロプロセッサ	3
マウス	122
マスクROM	19
マスクレジスタ	133
マッピングテーブル	114

【み】

ミスペナルティ	82
三つのC	77

【む】

無条件分岐命令	28, 39
---------	--------

【め】

命令	23
——の表現形式	32
命令ウィンドウ	110
命令キャッシュ	80
命令形式	32
命令語	23
命令スケジューリング	66
命令デコーダ	25
命令デコード	53
命令パイプライン	53
命令フェッチ	24, 53
命令プリデコード	99
命令ポストデコード	99
命令メモリ	24
命令レジスタ	24
メインメモリ	80
メモリ	11
——の語	18
メモリ操作命令	24

【も】

モデム	122
-----	-----

【ゆ】

有効ビット	85
-------	----

【よ】

容量性ミス	77
読出し	17

【ら】

ライトスルー方式74
 ライトバック方式74
 ライトバッファ75
 ライン74
 ランダムアービタ129

【り】

リオーダバッファ114
 リザベーションステーション 111
 リタイア115
 リトルエンディアン43

【A】

A ウェイのセットアソシアティ
 ブ形キャッシュ79
 ALU9

【C】

CD123
 CD-ROM122
 CD-RW123
 CISC48
 CRT123
 CRT/液晶ディスプレイ122

【D】

D フリップフロップ11
 DMA128
 DMA コントローラ130
 DRAM20,21
 DVD123
 DVD-RAM123

【E】

EEPROM20

【る】

ループアンローリング101

【れ】

例外132
 例外処理132
 レジスタ12
 レジスタファイル21
 レジスタリネーミング112
 レーテンシ108
 連想度79
 連想メモリ114

【ろ】

ローテーション126
 ローテーション時間126
 ロード命令38
 論理演算命令35
 論理関数6
 論理レジスタアドレス114

【わ】

ワード18
 割込み128
 割込みベクタ133

EPROM20

【G】

GPS123

【J】

JK フリップフロップ11

【L】

LAN123
 LRU80

【M】

MIPS120

【N】

n ビットレジスタ12

【O】

OS86

【P】

PROM19

【R】

RAM19
 RAW ハザード109
 RDRAM21
 RISC48
 ROM19

【S】

SDRAM21
 SRAM20,21

【T】

TLB86

【U】

UVEPROM20

【V】

VLIW96

【W】

WAR ハザード109
 WAW ハザード109

— 著者略歴 —

坂井 修一 (さかい しゅういち)

1986年 東京大学大学院工学系研究科博士課程修了 (情報工学専門課程)
工学博士 (東京大学)

現在, 東京大学教授

コンピュータアーキテクチャ

Computer Architecture

©一般社団法人 電子情報通信学会 2004

2004年3月31日 初版第1刷発行

2021年5月25日 初版第19刷発行

検印省略

編者 一般社団法人
電子情報通信学会
<https://www.ieice.org/>

著者 坂井 修一

発行者 株式会社 コロナ社
代表者 牛来真也

印刷所 壮光舎印刷株式会社

製本所 株式会社 グリーン

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 <https://www.coronasha.co.jp>

ISBN 978-4-339-01843-1 C3355 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いません。
落丁・乱丁はお取替えいたします。