計算機アーキテクチャ

京都大学情報学研究科
教授　富田眞治

用語知識調査（◎，○，△，×で記入）

Ｍｅｇａ　Ｇｉｇａ　Ｔｅｒａ

フォン　ノイマン

機械命令セット

プログラムカウンタ

レジスタ

スタックマシン

アドレッシングモード

手続き（関数）呼出し

再帰呼出し

浮動小数点数

２の補数

ＣＩＳＣとＲＩＳＣ

プロセススイッチ

割込みと例外

排他的論理和
ＳＲＡＭとＤＲＡＭ
キャッシュメモリ
仮想記憶
連想メモリ
共有メモリ
命令パイプライン
分岐予測
投機実行
スーパスカラ
ＶＬＩＷ
アウトオブオーダ実行
コンパイラ
並列処理

富田眞治

略歴

1945　出雲大社生まれ

1968　京大. 工. 電子工学科卒

1973　同工学研究科電気工学専攻
博士課程修了, 工学博士
合成音声多重出力システムの開発

1973　京大. 工. 情報工学科助手

1978　同助教授
低レベル並列計算機:
ＱＡ－1, ＱＡ－2の開発
グラフィックス用並列計算機
ＥＸＰＥＲＴＳの開発

1986　九州大学教授
　　可変構造型並列計算機の開発
　　スーパスカラ計算機の開発
　　レイトレーシングマシンの開発
　　ベクトル計算機の開発
1991　京都大学教授 (情報工学科)
　　超並列処理マシンの開発
　　超並列処理応用の研究
　　ボリュームレンダリングマシン
　　プロセッサ・アーキテクチャの研究
1998 京都大学情報学研究科教授
2006　同研究科長

著書

ＶＬＳＩコンピュータ，岩波，１９８４

・並列計算機構成論，　昭晃堂，１９８６

計算機システム工学，　昭晃堂，１９８８

並列処理マシン，オーム社，１９８９

並列処理機構，丸善，１９８９

・コンピュータアーキテクチャ　I，丸善，１９９４

・並列コンピュータ工学，昭晃堂，１９９６

・コンピュータアーキテクチャ，第2版，丸善，２０００

情報社会とコンピュータ、昭晃堂、２００５

・：単著

学会

情報処理学会理事 (平成7，8，１０，１１年度)
同関西支部長 (平成１３，１４年度)
電子情報通信学会フェロー (平成１２年度)
情報処理学会フェロー (平成１３年度)
情報処理学会功績賞 (平成１８年度)

コンピュータの動作原理

理解の深さのレベル

	車	コンピュータ	
利用	交通手段	文書作成、数値計算	アプリケーション
操作	運転	プログラム作成	プログラミング
原理、構成要素と相互関連			
	エンジン、ハンドル ブレーキ、車軸	演算装置、制御装置 記憶装置、入出力装置 プログラム制御方式	アーキテクチャ
構成要素の詳細構造			
	エンジンの構造	演算器の構造	論理設計
材料	チタン合金	トランジスタ	デバイス

講義内容

- 基本構造：ノイマン型コンピュータ
- 演算装置の構成
- コンピュータの設計
- 記憶装置の構成
- 制御装置の構成とＣＩＳＣ／ＲＩＳＣ
- 命令レベル並列処理とコンパイラ
- 並列コンピュータ概論

入門的

特論的

地球シミュレータ：５１２０台のプロセッサ
４０TFLOPS

京速コンピュータ：１０PFLOPS

1 基本単位

コンピュータ
電子回路：2つの安定
状態　ONまたはOFF

1．1 ビットとは: binary digit
2進数1桁の数字：0，1
①ビット：b，8ビット：バイトB，

IBM**では**

16ビット：半語HW，32ビット：語W，
64ビット：倍長語DW，128ビット：拡張語EW

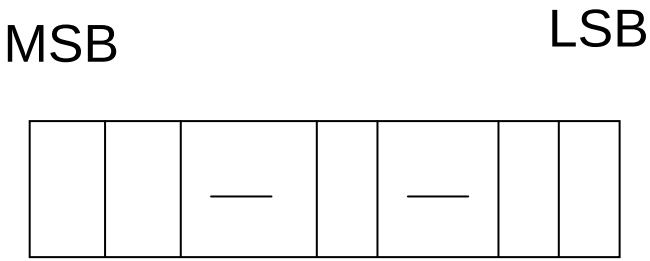
Intel**では**

16ビット：ワード，32ビット：ダブルワード，
64ビット：クワッドワード（Quad　Word），
128ビット：ダブルクワッドワード

②MSBとLSB

MSB：**最上位桁**

LSB：**最下位**



ビット列で表現できるもの

2進数

r進数 $(a_{n-1}a_{n-2}\ldots a_0)_r = a_{n-1}r^{n-1}+a_{n-2}r^{n-2}+\ldots+a_0r^0$

$0 \leqq a_i < r$

10進数：r=10

$(253)_{10}=2*10^2+5*10^1+3*10^0$

2進数：r=2

$(01011010)_2 = 2^6+2^4+2^3+2^1=(90)_{10}$

8ビットの2進数

$(00000000)_2=(0)_{10}\sim(11111111)_2=(255)_{10}$

③2進数

r進数との相互変換

$(a_{n-1}a_{n-2}\ldots a_0)_r = a_{n-1}r^{n-1}+a_{n-2}r^{n-2}+\ldots+a_0r^0$

2進数，8進数の相互変換：例 $(90)_{10}$

$(1011010)_2$ $(1\ 011\ 010)_2$ $(132)_8$

$(a_{n-1}a_{n-2}\ldots a_0)_2 = a_{n-1}2^{n-1}+a_{n-2}2^{n-2}+\ldots+a_02^0 =$ $0 \leqq a_i < r$

・・・・・・・・・

$(a_8 2^2 + a_7 2^1 + a_6 2^0)2^6 +$ ← 8^2

$(a_5 2^2 + a_4 2^1 + a_3 2^0)2^3 +$ ← 8^1

$(a_2 2^2 + a_1 2^1 + a_0 2^0)$ ← 8^0

$(1011010)_2 = 2^6 + 2^4 + 2^3 + 2^1 = 90$

2進数と16進数の相互変換

16進数

$(a_{n-1}a_{n-2}\ldots a_0)_{16} = a_{n-1}16^{n-1}+a_{n-2}16^{n-2}+\ldots+a_0 16^0$

$0 \leqq a_i < 16$

16進数の1~9、10~15の表現：1,2,3,…,9,A,B,C,D,E,F

2進数との対応： 1010：A,1011:B,1100:C,1101:D,1110:E,1111:F

2進数，16進数の相互変換：例 $(90)_{10}$

$(01011010)_2$ $(0101\ 1010)_2$ $(5A)_{16}$

$(a_{n-1}a_{n-2}\ldots a_0)_2 = a_{n-1}2^{n-1}+a_{n-2}2^{n-2}+\ldots+a_0 2^0 =$

・・・・・・・・・

$(a_{11}2^3+a_{10}2^2+a_9 2^1+a_8 2^0)2^8+$

$(a_7 2^3+a_6 2^2+a_5 2^1+a_4 2^0)2^4+$

$(a_3 2^3+a_2 2^2+a_1 2^1+a_0 2^0)$

16^2

16^1

16^0

2進数と10進数

2進数→10進数

$(a_{n-1}a_{n-2}\ldots a_0)_r =$

$a_{n-1}r^{n-1}+a_{n-2}r^{n-2}+\ldots+a_0r^0$

$(1011010)_2 = 2^6 + 2^4 + 2^3 + 2^1 = 90$

１０進整数→２進数

$(N)_{10} = (a_{n-1}a_{n-2}\ldots a_0)_2$

$= (a_{n-1}a_{n-2}\ldots a_1) * 2 + a_0$

$= (a_{n-1}a_{n-2}\ldots a_2) * 2 + a_1) * 2 + a_0$

１０進数／２＝商$_0$　＋余り$_0$

商$_i$／２　＝商$_{i+1}$　＋余り$_{i+1}$

	商	余り	
90÷2	45	0	a_0
45÷2	22	1	a_1
22÷2	11	0	a_2
11÷2	5	1	a_3
5÷2	2	1	a_4
2÷2	1	0	a_5

a_6

１０進小数→２進小数

$(0.N)_{10} = (0.a_{-1}a_{-2}...a_{-m})_2$

$2 * (0.N)_{10} = a_{-1} + (0.a_{-2}...a_{-m})_2$

２＊（１０進数）＝整数部$_{-1}$＋小数部$_{-1}$

２＊小数部$_{-i}$＝整数部$_{-(i+1)}$＋小数部$_{-(i+1)}$

$(0.1)_{10}$は？

2＊0．1＝0．2　　整数部0
2＊0．2＝0．4　　整数部0
2＊0．4＝0．8　　整数部0
2＊0．8＝1．6　　整数部1
2＊0．6＝1．2　　整数部1
2＊0．2＝0．4　　整数部0
2＊0．4＝0．8　　整数部0
2＊0．8＝1．6　　整数部1
2＊0．6＝1．2　　整数部1

繰り返し

循環小数

0．00011

$0.0\dot{0}01\dot{1}$

有限桁の問題（１０／３）＊３を電卓で計算すると？

その他ビット列で表されるもの：符号

④漢字：　2バイト符号(16進4桁)

漢字：第1，2水準

富：4959，田：4544，

真：3F3F，眞：6243，治：3C23

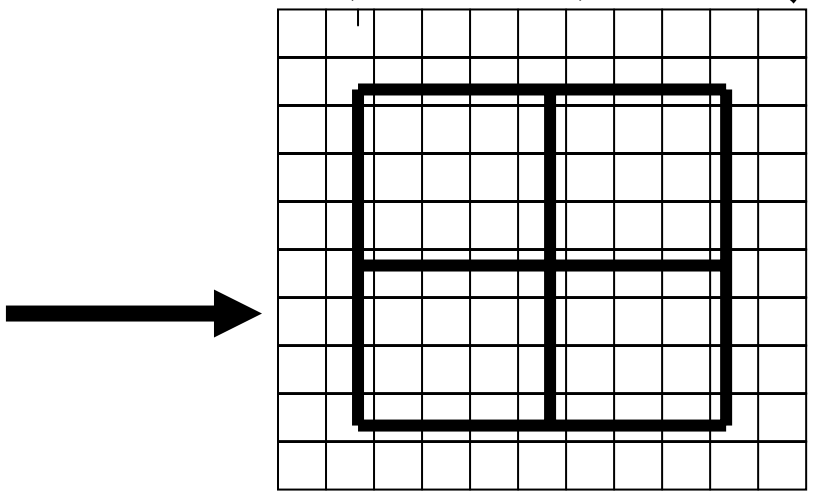
⑤パターン

プリンタでパターン用意

プロセッサ

2バイトデータ

4544



24x24

ビット

表示

⑥情報理論での情報量の「ビット」

確率pで生起する事象が起こったことにより
得られる情報量　$-\log_2 p$　ビット

N個の事象からなる情報源の情報量

$$-\Sigma p_i \log_2 p_i$$

4つの事象0，1，2，3の生起確率
1/4，1/4，1/4，1/4のとき
情報量：2ビット
符号化：0→00，1→01，2→10，3→11
2ビット

4つの事象0，1，2，3の生起確率

1/2，1/4，1/8，1/8**のとき**

情報量：1.75**ビット**

符号化：**0→00，1→01，2→10，3→11**

2ビット

符号化：0→0，1→10，**2→**110，**3→**111

1.75**ビット**

事象列 03201

符号列 0111110010

符号化 ↓

復号化 ↑

1．1．2 各種の量

K：キロ 2^{10}，10^{3}
M：メガ 2^{20}，10^{6}
G：ギガ 2^{30}，10^{9}
T：テラ 2^{40}，10^{12}
P：ペタ 2^{50}，10^{15}

キャッシュメモリ：64KB
メインメモリ：512ＭＢ
パソコン周波数：2ＧＨｚ
ディスク容量：100ＧＢ
地球シミュレータ性能：40TFLOPS
近未来のスーパコン：1PFLOPS

FLOPS:Floating Point Operations／sec

m：ミリ 10^{-3}

μ：マイクロ 10^{-6}

n：ナノ 10^{-9}

p：ピコ 10^{-12}

f：フェムト 10^{-15}

ディスク速度：10msec

LSI設計寸法：0.065μm

光の伝播速度：3.3nsec/m

クロックサイクルタイム：1nsec

ゲート遅延時間：30psec

DRAMのコンデンサ容量：50fF

単位の表現

記号	読み方	SI接頭語	日本語	備考
a	アト	atto-	指（シ）	10^{-18}
f	フェムト	femto-	須（シュユ）	10^{-15}
p	ピコ	pico-	漠（バク）	10^{-12}
n	ナノ	nano-	塵（ジン）	10^{-9}
μ	マイクロ	micro-	微（ビ）	10^{-6}
m	ミリ	milli-	毛（モウ）	10^{-3}
c	センチ	centi-	厘（リン）	10^{-2}
d	デシ	deci-	分（ブ）	10^{-1}
da	デカ	deca-	十（ジュウ）	10^{1}
h	ヘクト	hecto-	百（ヒャク）	10^{2}
k	キロ	kilo-	千（セン）	10^{3} (2^{10})
M	メガ	mega-	百万（マン）	10^{6} (2^{20})
G	ギガ	giga-	十億（オク）	10^{9} (2^{30})
T	テラ	tera-	兆（チョウ）	10^{12} (2^{40})
P	ペタ	peta-	千兆	10^{15} (2^{50})
E	エクサ	exa-	百京（ケイ）	10^{18} (2^{60})

［備考］のカッコ内は、コンピュータのアドレスで用いた場合。

1．2 動作原理

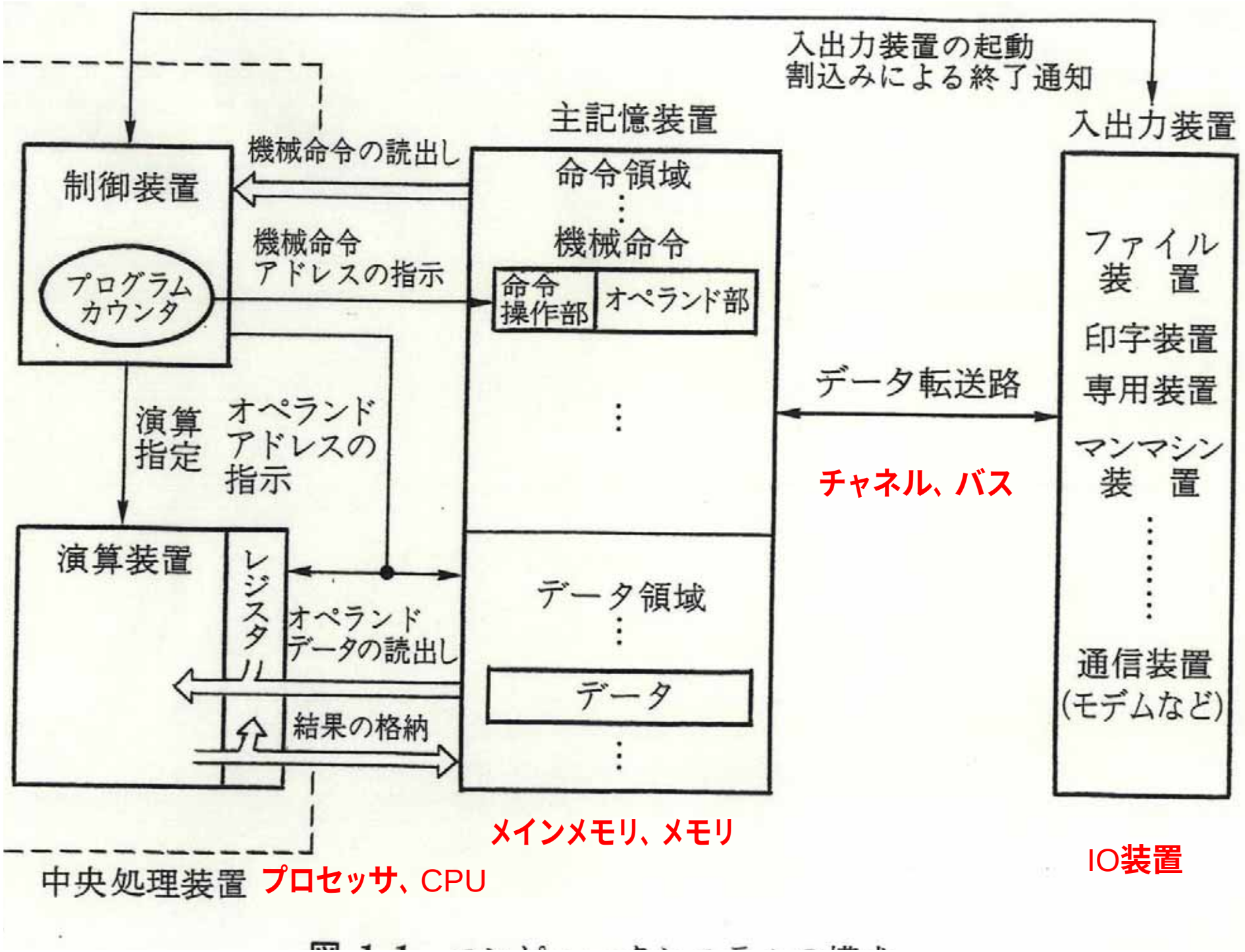
1．2．1 基本原理

(1) 演算装置

(2) 主記憶装置

(3) 制御装置

(4) 入出力装置


入出力装置の起動
割込みによる終了通知
主記憶装置
入出力装置
制御装置
機械命令の読出し
命令領域
機械命令
機械命令
アドレスの指示
プログラム
カウンタ
命令
操作部
オペランド部
ファイル
装　置
印字装置
専用装置
マンマシン
装　置
データ転送路
演算
指定
オペランド
アドレスの
指示
チャネル、バス
演算装置
レジスタ
オペランド
データの読出し
データ領域
データ
結果の格納
通信装置
(モデムなど)
メインメモリ、メモリ
IO装置
中央処理装置 プロセッサ、CPU

主記憶装置（メインメモリ，メモリ）

- バイトアドレッシング
  - リトル／ビッグエンディアン
- プログラムの格納
  - 命令列
  - データ

１ＭＢのメモリ

２０ビットアドレス →

１６進5桁アドレス

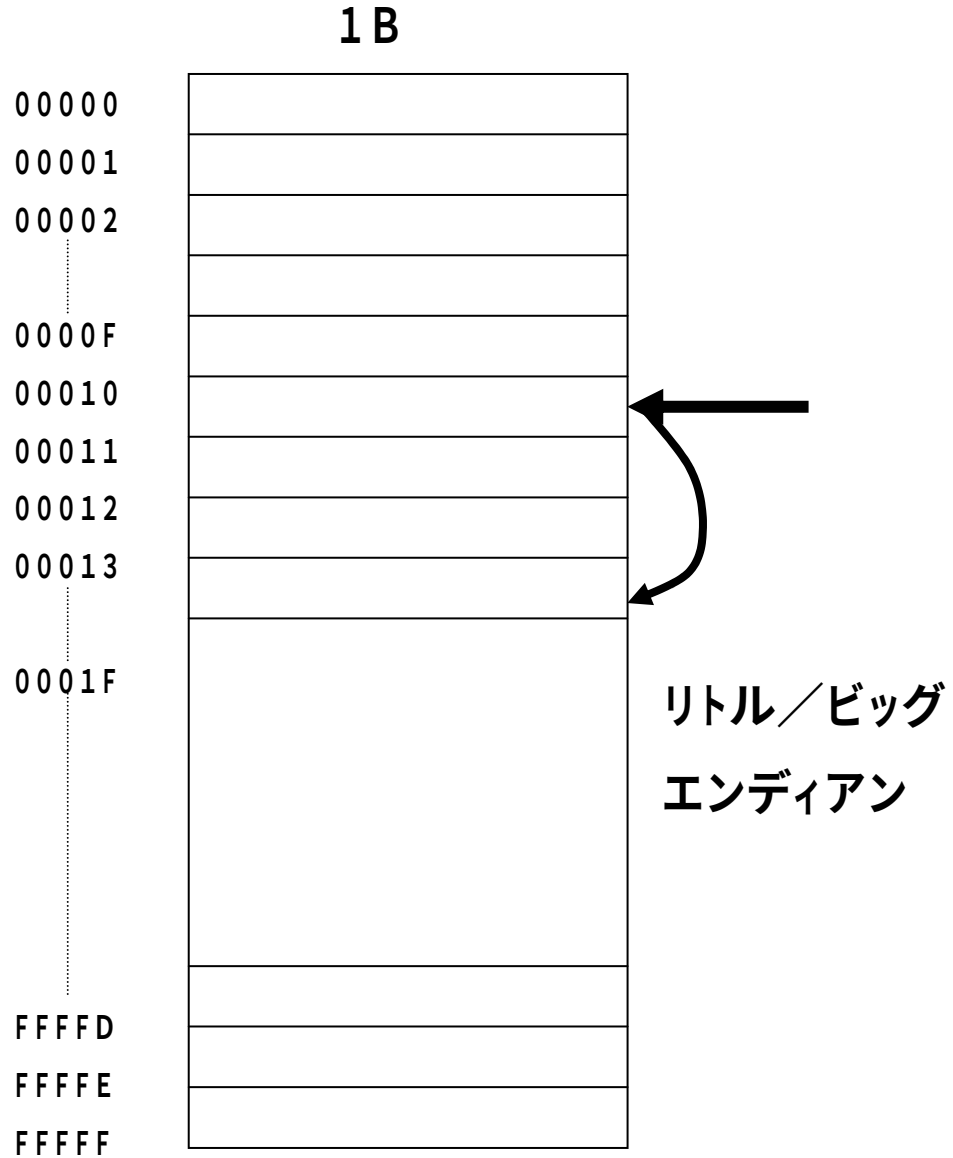
００００〜ＦＦＦＦＦ

Ｒ／Ｗ →

バイト数 →

Read:　Load ←

Write：　Store →



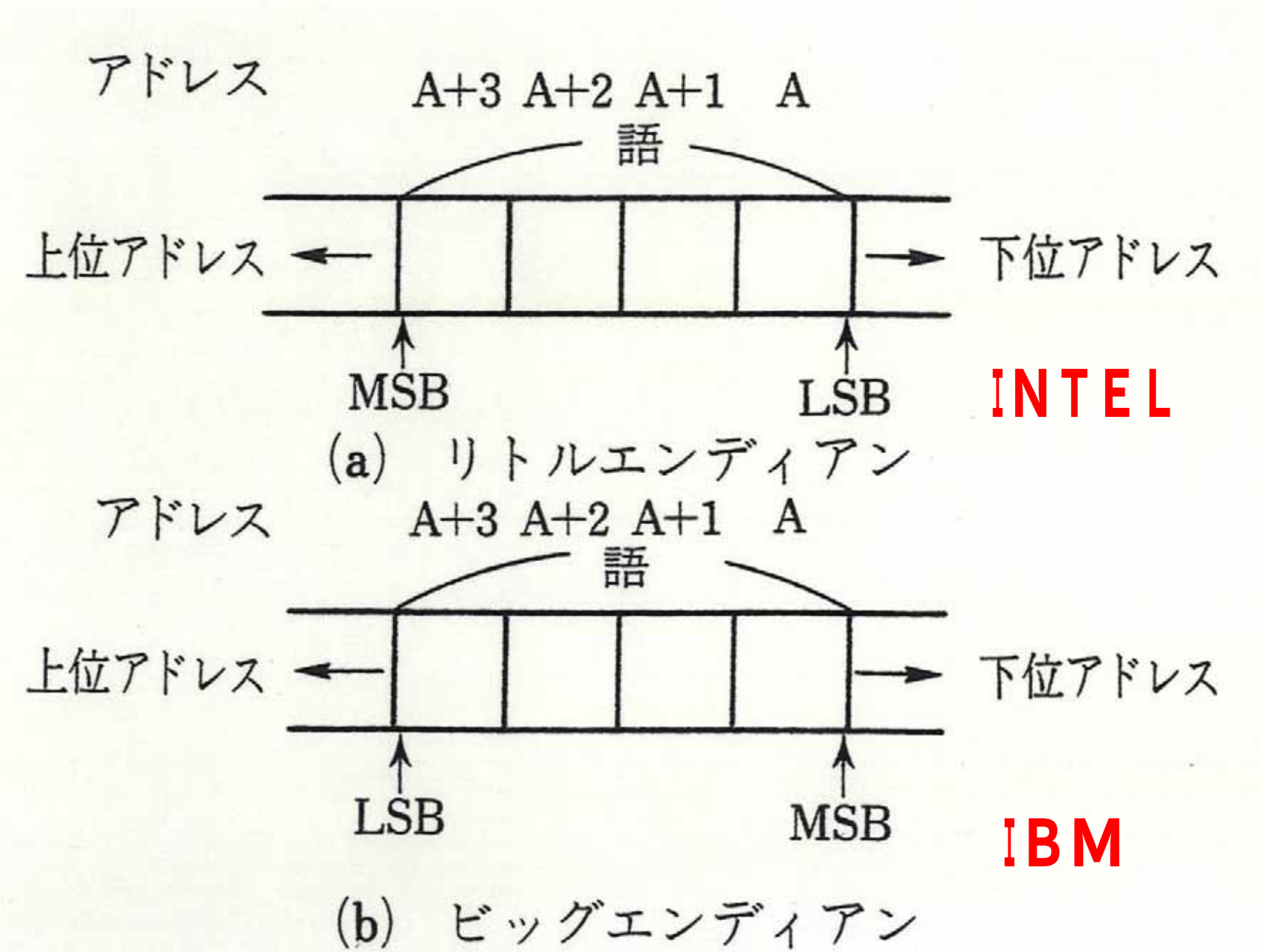


図 1.2 リトル/ビッグエンディアン

実際の主記憶装置

記憶階層方式

例：頭の中，メモ帳，机上，引出し，本箱，倉庫

キャッシュメモリ

仮想記憶方式

線形記憶と連想記憶

アドレス→内容：線形記憶

内容→アドレス：連想記憶

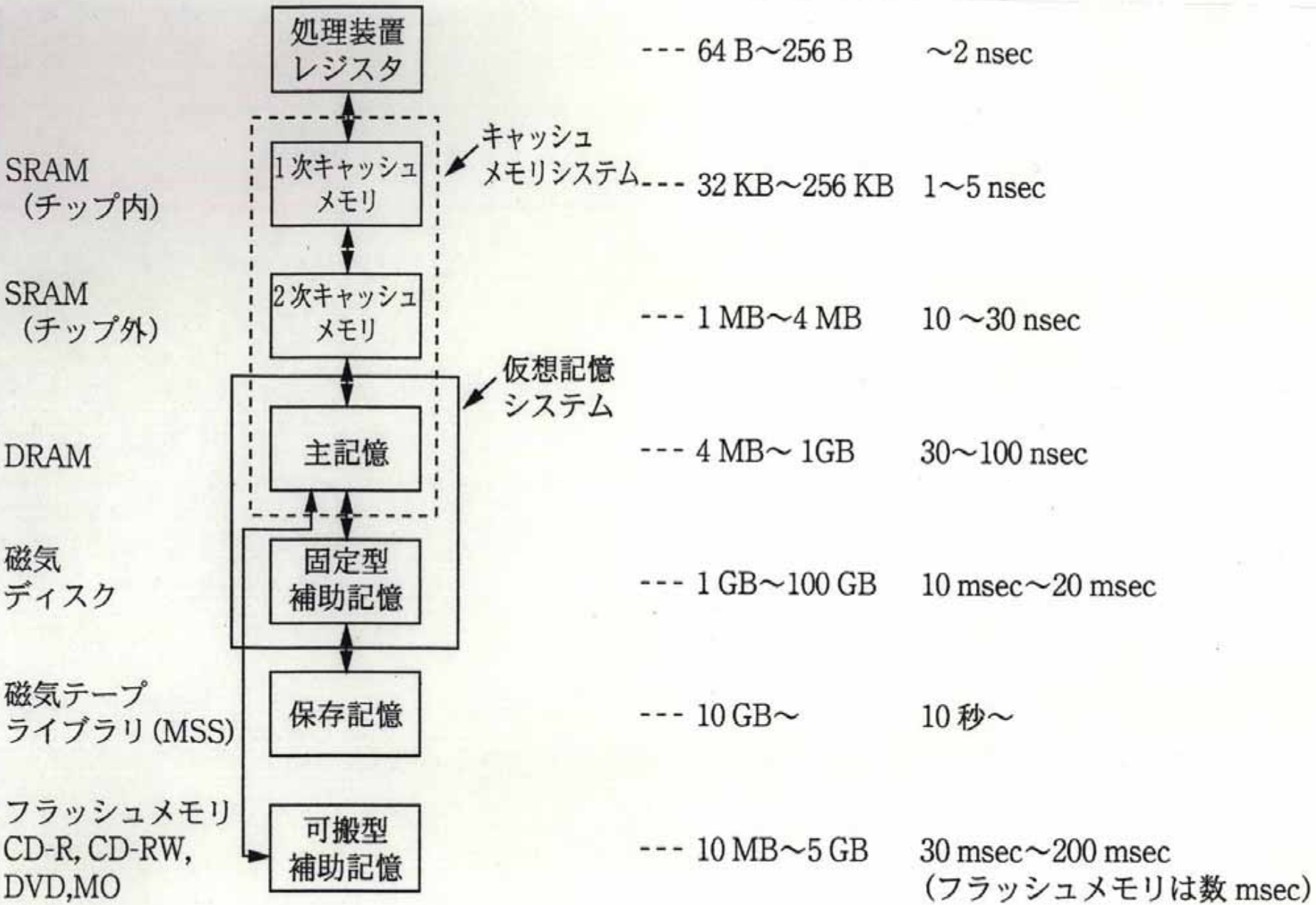

処理装置
レジスタ
--- 64 B～256 B
～2 nsec
キャッシュ
メモリシステム
SRAM
（チップ内）
1次キャッシュ
メモリ
--- 32 KB～256 KB
1～5 nsec
SRAM
（チップ外）
2次キャッシュ
メモリ
--- 1 MB～4 MB
10 ～30 nsec
仮想記憶
システム
DRAM
主記憶
--- 4 MB～ 1GB
30～100 nsec
磁気
ディスク
固定型
補助記憶
--- 1 GB～100 GB
10 msec～20 msec
磁気テープ
ライブラリ (MSS)
保存記憶
--- 10 GB～
10 秒～
フラッシュメモリ
CD-R, CD-RW,
DVD,MO
可搬型
補助記憶
--- 10 MB～5 GB
30 msec～200 msec
（フラッシュメモリは数 msec）

図 4.5 記憶階層方式

DRAMについての私の経験

1Kb DRAM

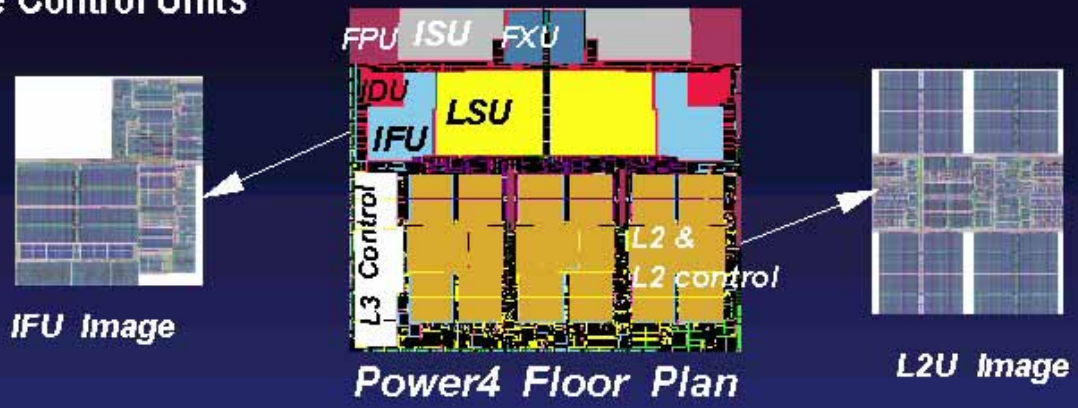
- 1974に購入 (東光株式会社)
- アクセスタイム 350nsec
- 容量256KB
- 価格1000万円

256Mb DRAM

- 2001年にパソコンのアドオンメモリとして購入
- アクセスタイム 70nsec
- 容量128MB
- 価格5000円

Power4 Processor Design

- Power4 Floor Plan and Images of the Instruction Fetch (IFU) & L2 Cache Control Units



- Power4 Processor Chip Features
  - Two microprocessors, 3-level cache hierarchy
  - >30 GB/s (>500 MHz) interconnection fabric
  - 2200 signal I/O; 170M transistors
  - Configured for 8-way system on MCM
  - CMOS 8S2-SOI technology & copper wiring
- Microprocessor core
  - > 1GHz clock frequency; 64KB L1 instruction cache
  - 64-bit PowerPC for RS6000 & AS400 systems
  - Out-of-order, speculative, 8-issue superscalar design
- Memory sub-system
  - 1.5 MB shared L2 caches
  - 32 MB off-chip L3 (on-chip control); Bandwidth > 10GB/s

機械命令

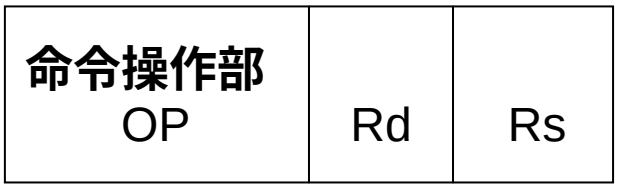
レジスタ：１６個（4ビット指定）

メモリ：１ＭＢ（２０ﾋﾞｯﾄアドレス）

命令操作部：加算などの操作指定

オペランド：レジスタやメモリなど演算をされるデータを指定，0～3個

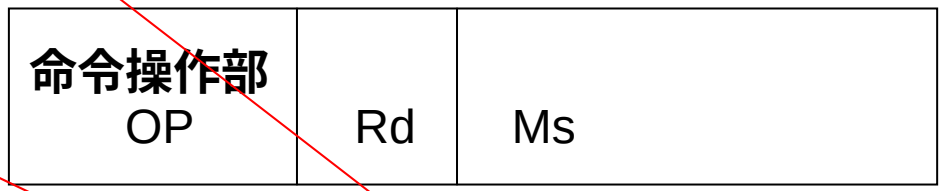
2バイト命令



Rd←Rd OP Rs

例　MUL　R0　R1

4バイト命令



Rd←Rd OP Ms

例　ADD　R0　A

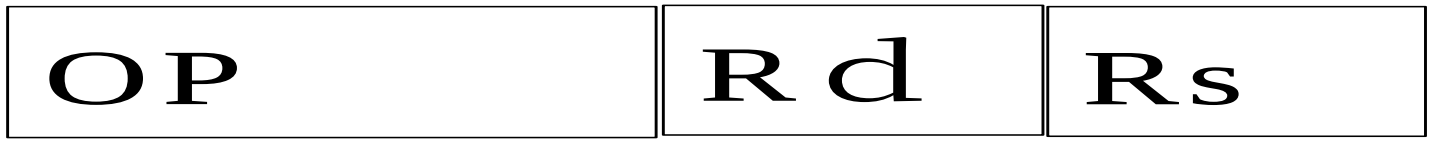
dデスティネーション

Sソース

LOAD:00，STORE：01，ADD:02，MUL:03

データはすべて４Ｂ長

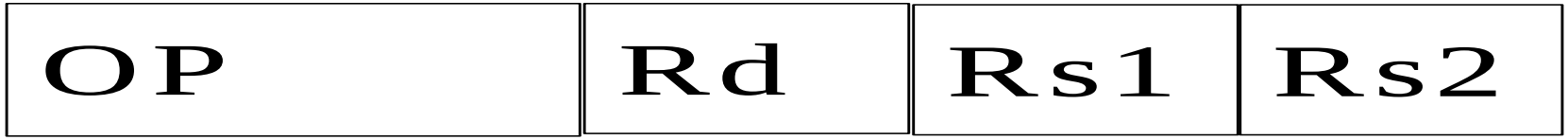
・２オペランド命令



Rd ←― Rd op Rs

Rd の内容書き換えられる

・３オペランド命令



Rd ←― Rs1 op Rs2

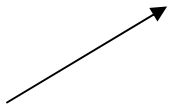
Rs1,Rs2：内容そのまま

記号表現　　MUL　R0　R1

16進表現　　**03**　　**0**　　**1**

記号表現　　ADD　R0　A

16進表現　　**02**　　**0**　**00020**



データAはメモリの00020アドレスに記憶

命令番号	命令			バイトアドレス (16進)	主記憶内での表現 (16進)
1	LOAD	R1	A	00000	00100020
2	ADD	R1	B	00004	02100024
3	LOAD	R2	C	00008	00200028
4	ADD	R2	D	0000C	0220002C
5	MUL	R1	R2	00010	0312
6	STORE	E	R1	00012	01100030

(1.5)

E＝(A＋B)＊(C＋D)

A～E：4Bデータ

データ名		
A	00020	00000010
B	00024	00000003
C	00028	0000000A
D	0002C	00000011
E	00030	?

アセンブリプログラム　　目的プログラム

(3) 制御装置

プログラムカウンタ：PC

逐次制御

コントロール駆動 VS **データ駆動**

プログラムカウンタで指された命令のみ実行

実行可能であっても指されなかったらX

１．２．２ ノイマンモデル

基本モデル：　１９４５年、John von Neumann

①プログラム内蔵：Stored　Program

プログラムの電子的速度での呼び出しと変更

②逐次制御：一つ一つ

③低機能命令：鈍臭いが超速く，八方美人で汎用

④線形記憶：逐次制御と相性極めてよし

存続理由

①プログラムの継続利用：互換性

IBM360：1964，Intel **IA-32，8086：1978**

②デバイス技術の発展

③シンプルな構造：人間の思考方法と一致

Date	Inventor: machine	Capability	Technical innovations
1642	Pascal	Addition, subtraction	Automatic carry transfer; complements number representation
1671	Leibniz	Addition, subtraction multiplication, division	"Stepped reckoner" mechanism for multiplication and division
1827	Babbage: Difference Engine	Polynomial evaluation by method of finite differences	Automatic multistep operation
1834	Babbage: Analytical Engine (never completed)	General-purpose computation	Automatic sequence control mechanism (program)
1941	Zuse: Z3	General-purpose computation	The first operational general-purpose computers
1944	Aiken: Harvard Mark I		

Figure 1.10 Milestones in the development of mechanical computers.

J.P.Hays: Computer Architecture and Organization, McGraw-Hill, 1978

Generation	Technologies	Hardware features	Software features	Representative computers
First (1946–54)	Vacuum tubes; acoustic memories; CRT memories	Fixed-point arithmetic	Machine language; assembly language	IAS; UNIVAC
Second (1955–64)	Discrete transistors; ferrite cores; magnetic disks	Floating-point arithmetic; index registers; IO processors	High-level languages; subroutine libraries; batch monitors	IBM 7094; CDC 1604
Third (1965–74)	Integrated circuits (SSI and MSI)	Microprogramming; pipelining; cache memory	Multiprogramming; multiprocessing; operating systems; virtual memory	IBM S/360; DEC PDP-8
Fourth? (1975–	LSI circuits; semiconductor memories			Amdahl 470 Intel 8748

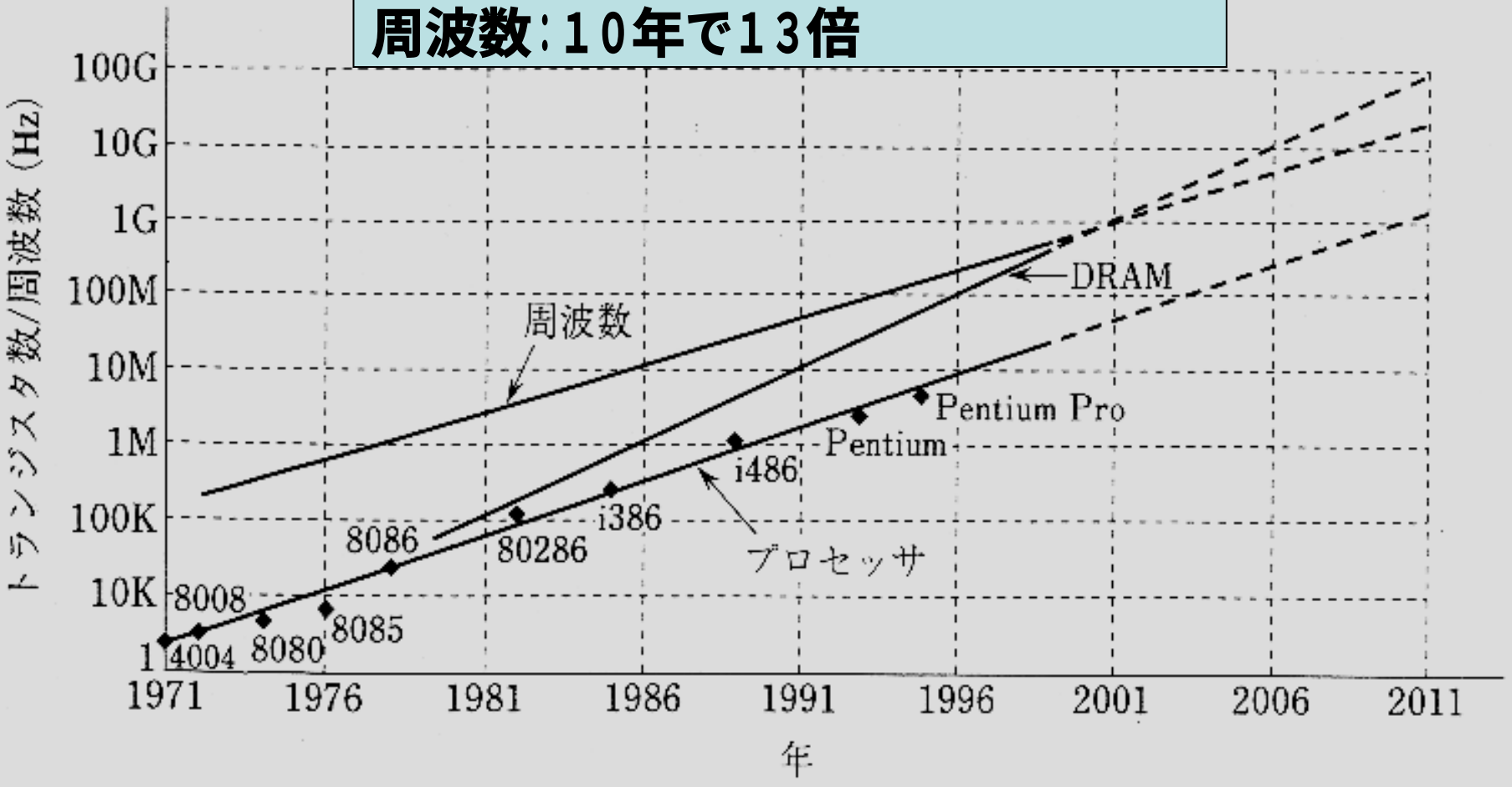
Figure 1.41 Milestones in the development of electronic computers.

J.P.Hays: Computer Architecture and Organization, McGraw-Hill, 1978

表 1.1　デバイス技術の発展[15)]

1946～1958：真空管
　　ENIAC（1946）
1958～1964：トランジスタ
　　Philco 社 Transacs-2000（1958）
1964～1970：SSI/MSI（Small/Medium Scale Integration ～10^3 個/チップ）
　　IBM 360（1964）
1970～1980：LSI（Large Scale Integration 10^3～10^5 個/チップ）
　　IBM 370（1970），Intel 4004（1971），1 KbDRAM（1970）
1980～1990：VLSI（Very Large Scale Integration 10^5～10^7 個/チップ）
1990～　　：ULSI（Ultra Large Scale Integration 10^7～個/チップ）

メモリ容量：10年で100倍
(ムーアの法則：3年で4倍)
周波数：10年で13倍



VLSI ロードマップ

Intelのマイクロプロセッサの発展過程

Intel Processor	Date Intro-duced	Max. Clock Frequency at Intro-duction	Tran-sistors	Register Sizes[1]	Ext. Data Bus Size[2]	Max. Extern. Addr. Space	Caches
8086	1978	8 MHz	29 K	16 GP	16	1 MB	None
Intel 286	1982	12.5 MHz	134 K	16 GP	16	16 MB	Note 3
Intel386 DX Processor	1985	20 MHz	275 K	32 GP	32	4 GB	Note 3
Intel486 DX Processor	1989	25 MHz	1.2 M	32 GP 80 FPU	32	4 GB	L1: 8 KB
Pentium Processor	1993	60 MHz	3.1 M	32 GP 80 FPU	64	4 GB	L1:16 KB
Pentium Pro Processor	1995	200 MHz	5.5 M	32 GP 80 FPU	64	64 GB	L1: 16 KB L2: 256 KB or 512 KB
Pentium II Processor	1997	266 MHz	7 M	32 GP 80 FPU 64 MMX	64	64 GB	L1: 32 KB L2: 256 KB or 512 KB
Pentium III Processor	1999	500 MHz	8.2 M	32 GP 80 FPU 64 MMX 128 XMM	64	64 GB	L1: 32 KB L2: 512 KB
Pentium III and Pentium III Xeon Processors	1999	700 MHz	28 M	32 GP 80 FPU 64 MMX 128 XMM	64	64 GB	L1: 32 KB L2: 256 KB

Intel Processor	Date Intro-duced	Microarchi-tecture	Clock Frequency at Intro-duction	Tran-sistors	Register Sizes[1]	System Bus Band-width	Max. Extern. Addr. Space	On-Die Caches[2]
Pentium 4 Processor	2000	Intel NetBurst Microarchi-tecture	1.50 GHz	42 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	3.2 GB/s	64 GB	12K µop Execution Trace Cache; 8KB L1; 256-KB L2
Intel Xeon Processor	2001	Intel NetBurst Microarchi-tecture	1.70 GHz	42 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	3.2 GB/s	64 GB	12K µop Trace Cache; 8-KB L1; 256-KB L2

Intel Processor	Date Intro-duced	Microarchi-tecture	Clock Frequency at Intro-duction	Tran-sistors	Register Sizes[1]	System Bus Band-width	Max. Extern. Addr. Space	On-Die Caches[2]
Intel Xeon Processor	2002	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology	2.20 GHz	55 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	3.2 GB/s	64 GB	12K µop Trace Cache; 8-KB L1; 512-KB L2
Intel Xeon Processor MP	2002	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology	1.60 GHz	108 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	3.2 GB/s	64 GB	12K µop Trace Cache; 8-KB L1; 256-KB L2; 1-MB L3
Intel Pentium 4 Processor Supporting Hyper-Threading Technology	2002	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology	3.06 GHz	55 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	4.2 GB/s	64 GB	12K µop Execution Trace Cache; 8-KB L1; 512-KB L2
Intel Pentium M Processor	2003	Intel Pentium M Processor	1.60 GHz	77 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	3.2 GB/s	4 GB	L1: 64 KB L2: 1 MB
Intel Pentium 4 Processor Supporting Hyper-Threading Technology at 90 nm process	2004	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology	3.40 GHz	125 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	6.4 GB/s	64 GB	12K µop Execution Trace Cache; 16 KB L1; 1 MB L2
Intel Pentium M Processor 755[3]	2004	Intel Pentium M Processor	2.00 GHz	140 M	GP: 32 FPU: 80 MMX: 64 XMM: 128	3.2 GB/s	4 GB	L1: 64 KB L2: 2 MB
64-bit Intel Xeon Processor with 800 MHz System Bus	2004	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology; Intel Extended Memory 64 Technology	3.60 GHz	125 M	GP: 32, 64 FPU: 80 MMX: 64 XMM: 128	6.4 GB/s	64 GB	12K µop Execution Trace Cache; 16 KB L1; 1 MB L2

Intel Processor	Date Intro-duced	Microarchi-tecture	Clock Frequency at Intro-duction	Tran-sistors	Register Sizes[1]	System Bus Band-width	Max. Extern. Addr. Space	On-Die Caches[2]
64-bit Intel Xeon Processor MP with 8MB L3	2005	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology; Intel Extended Memory 64 Technology	3.33 GHz	675M	GP: 32, 64 FPU: 80 MMX: 64 XMM: 128	5.3 GB/s [4]	1024 GB (1 TB)	12K µop Execution Trace Cache; 16 KB L1; 1 MB L2, 8 MB L3
Intel Pentium 4 Processor Extreme Edition Supporting Hyper-Threading Technology	2005	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology; Intel Extended Memory 64 Technology	3.73 GHz	164 M	GP: 32, 64 FPU: 80 MMX: 64 XMM: 128	8.5 GB/s	64 GB	12K µop Execution Trace Cache; 16 KB L1; 2 MB L2
Intel Pentium Processor Extreme Edition 840	2005	Intel NetBurst Microarchi-tecture; Hyper-Threading Technology; Intel Extended Memory 64 Technology; Dual-core [5]	3.20 GHz	230 M	GP: 32, 64 FPU: 80 MMX: 64 XMM: 128	6.4 GB/s	64 GB	12K µop Execution Trace Cache; 16 KB L1; 1MB L2 (2MB Total)

Pentium4	4004
発表年：2000年	1971年
集積度：4200万TR	2300TR
周波数：1．4GHz	750KHz
ピン数：478	16
電力：55W	？
配線層：6層	？
演算幅：32ビット	4ビット
メモリ256MbDRAM	1KbDRAM(1970)

１．３ 機械命令形式とアドレス方式

１．３．１ 命令形式

命令操作：８ビット，レジスタ：４ビット，メモリ：１２ビット

①２，３アドレス命令形式　　ビット数

Md←Ms1 op Ms2　　OP　Md　Ms1　Ms2 (４４)

Md←Md op Ms　　OP　Md　Ms (３２)

②１アドレス命令形式

レジスターメモリ方式

Rd←Rd op Ms　　OP　Rd　Ms (２４)

(含Rd ←Rd op Rs，　OP　Rd　Rs (１６)

Rd ←Rs1 op Rs2)　　OP　Rd　Rs1　Rs2 (２０)

(a)

```
ADD     B, A
ADD     C, D
MUL     B, C
STORE   E, B
```

(b)

```
LOAD    R1, A
ADD     R1, B
LOAD    R2, C
ADD     R2, D
MUL     R2, R1
STORE   E, R2
```

(c)

```
LOAD    A
ADD     B
STORE   T
LOAD    C
ADD     D
MUL     T
STORE   E
```

(d)

```
LOAD    R1, A
LOAD    R2, B
ADD     R1, R2
LOAD    R3, C
LOAD    R4, D
ADD     R3, R4
MUL     R1, R3
STORE   E, R1
```

(e)

```
PUSH    A
PUSH    B
ADD
PUSH    C
PUSH    D
ADD
MUL
STORE   E
```

E = (A + B) * (C + D)

	(a)	(b)	(c)	(d)	(e)
命令フェッチ回数(実行命令数)	4	6	7	8	8
主記憶データアクセス回数	11	5	7	5	5
必要メモリ量(ビット)	128	136	140	168	124
処理時間	**25**	**13**	**17**	**13**	**13**

図 1.5　各種機械命令形式の評価 (E=(A+B)＊(C+D) の場合)
(a) 2アドレス命令形式, (b) 1アドレス命令形式 (レジスタ-メモリ方式, (c) 1アドレス命令形式 (アキュムレータ方式), (d) 1アドレス命令形式 (ロードストア方式), (e) 0アドレス命令形式

アキュムレータ方式

ACC←ACC op Ms	OP　Ms（20）
Md←ACC	STORE　Md（20）
ACC←Ms	LOAD　Ms（20）

ロードストア方式

Rd←Rs1 op Rs2	OP　Rd　Rs1　Rs2（20）
Rd←Rd op Rs	OP　Rd　Rs（16）
Rd←Ms,　Md←Rs	LOAD　Rd　Ms（24）
	STORE Md　Rs（24）

③0アドレス命令形式

スタックコンピュータ

PUSH　Ms（20）

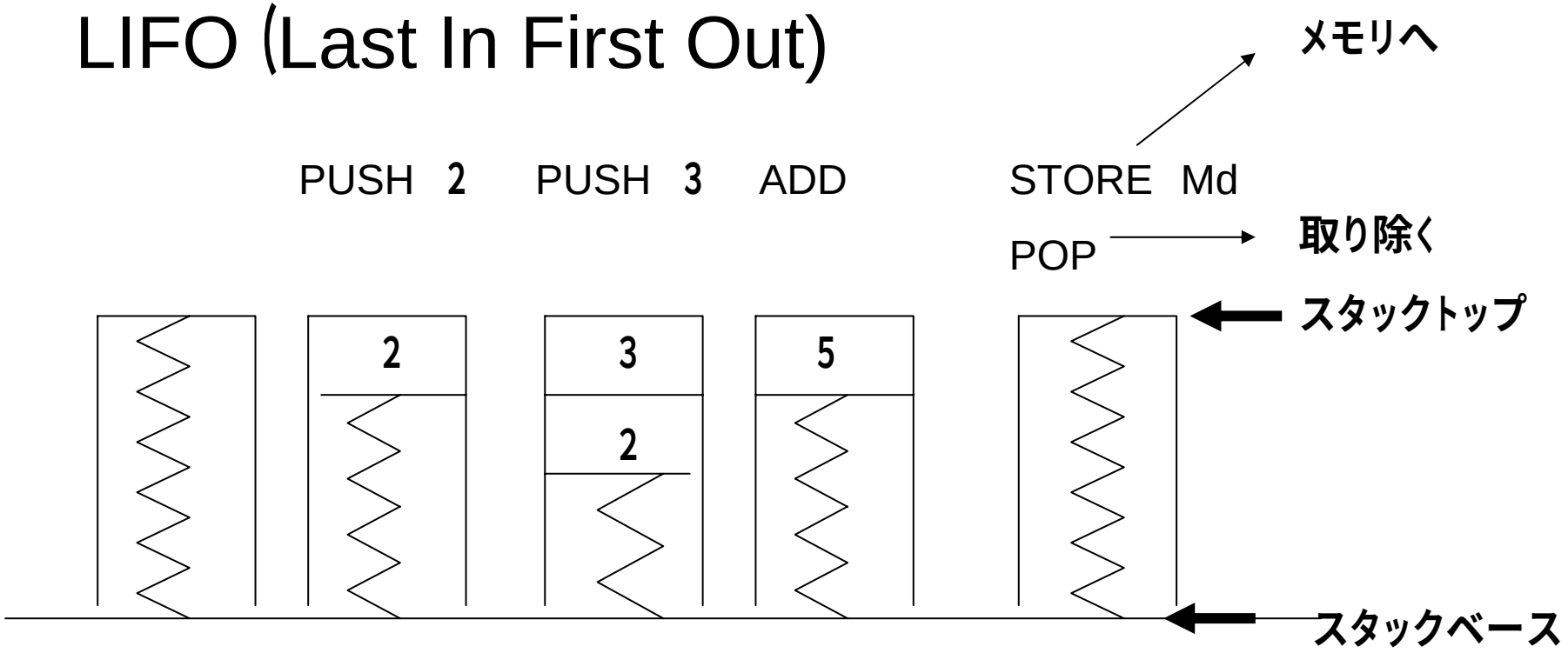
STORE　Md（20）

OP　（8）：**オペランド指定なし**

POP　（8）

スタックの構造

LIFO (Last In First Out)



反対語

FIFO (First In First Out): **キュー**

- インフィックス記法:　２＋３
- プレフィックス記法:＋２３
  (ポーランド記法)
- ポストフィックス記法:２３＋
  (逆ポーランド記法)

スタックによる計算

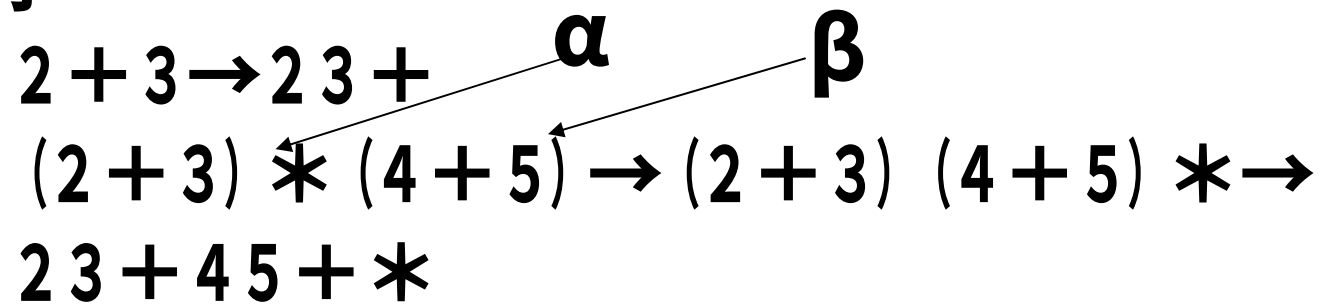
①ポストフィックス記法の数式を左から1回走査

②変数 (数) であればスタックにＰＵＳＨ

③演算であればスタックトップとその下の数に
　演算を施し，結果をプッシュ

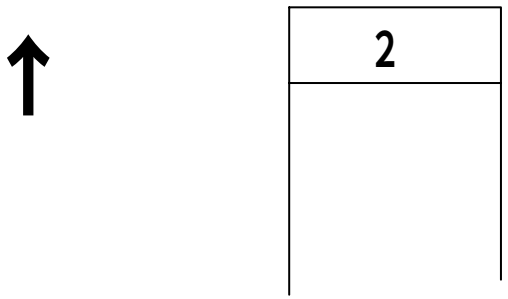
逆ポーランド記法への変換

インフィックス記法 αopβ→
逆ポーランド記法　αβop
部分式α
部分式β
部分式に再帰的に適用
例



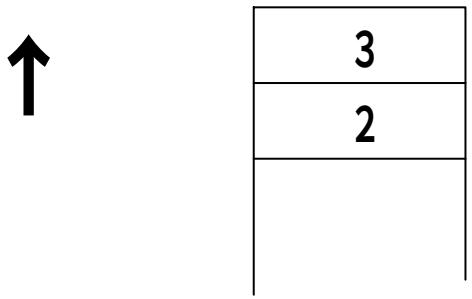
23＋の計算

23＋



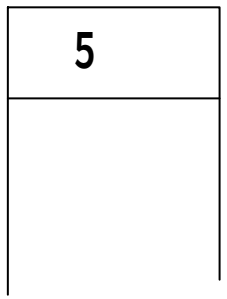
PUSH

23＋



PUSH

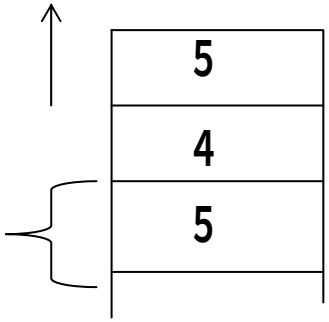
23＋



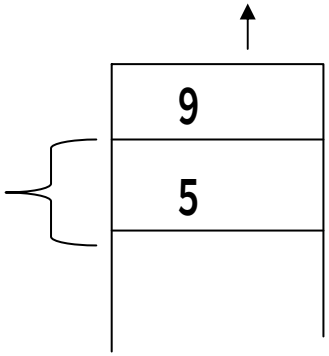
加算

(2+3)*(4+5)の計算

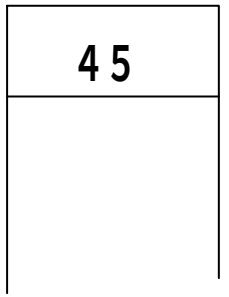
23＋45＋＊



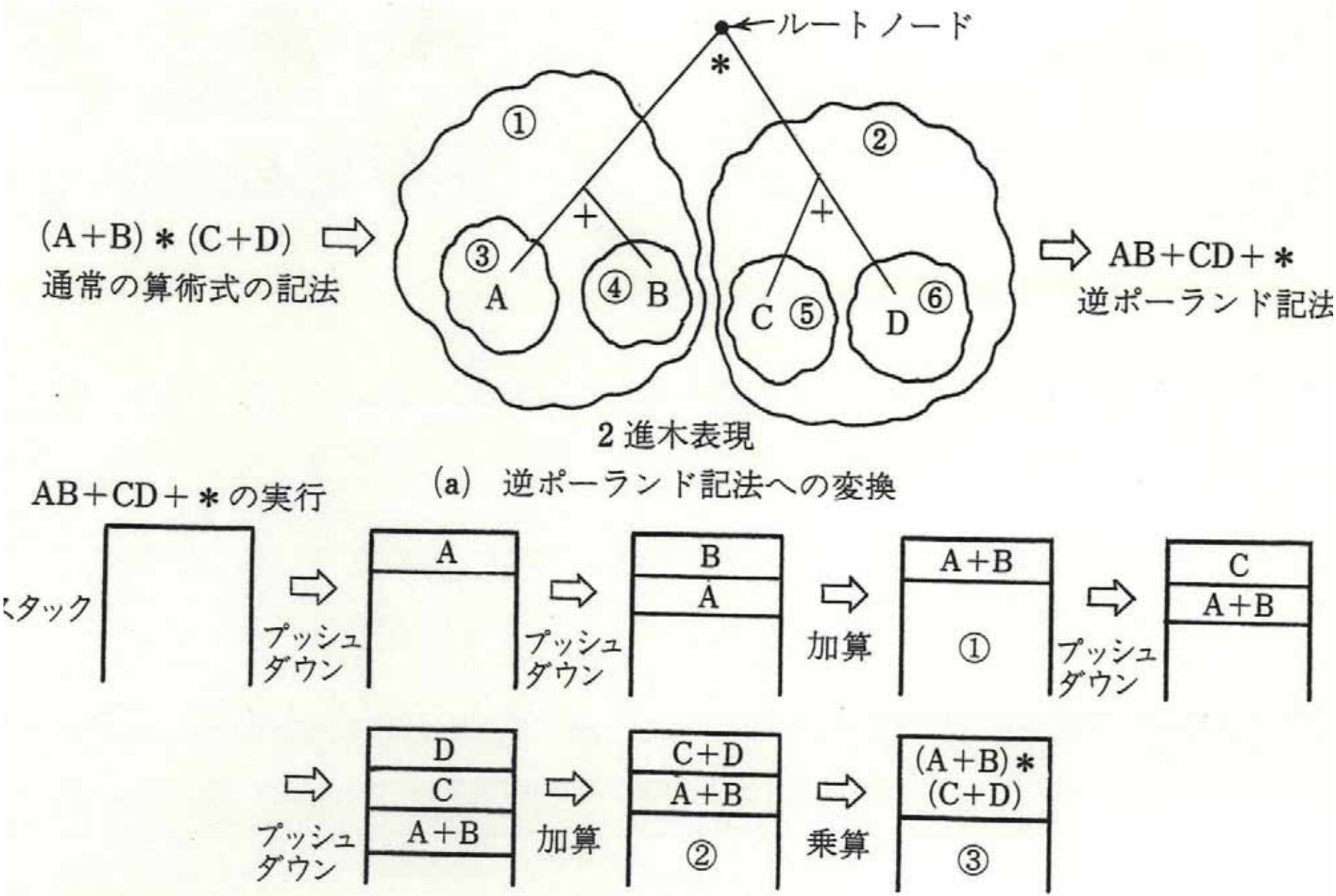
23＋45＋＊



23＋45＋＊



23＋の結果をボトムにおいてその上で45＋の加算.
それらを掛け算


ルートノード
*
①
②
+
+
③
A
④ B
C ⑤
D ⑥
(A+B) * (C+D)
通常の算術式の記法
AB+CD+*
逆ポーランド記法
2進木表現
(a) 逆ポーランド記法への変換
AB+CD+*の実行
スタック
プッシュダウン
A
プッシュダウン
B
A
加算
A+B
①
プッシュダウン
C
A+B
プッシュダウン
D
C
A+B
加算
C+D
A+B
②
乗算
(A+B)*(C+D)
③
(b) AB+CD+*の実行

１．４ 各種機械命令の評価

命令実行数 (IC)

データアクセス回数

必要メモリ量

プログラム実行時間

T＝IC＊CPI＊MC

演算時間1、メモリ2サイクル

逐次　vs　並列

(a)

```
ADD     B, A
ADD     C, D
MUL     B, C
STORE   E, B
```

(b)

```
LOAD    R1, A
ADD     R1, B
LOAD    R2, C
ADD     R2, D
MUL     R2, R1
STORE   E, R2
```

(c)

```
LOAD    A
ADD     B
STORE   T
LOAD    C
ADD     D
MUL     T
STORE   E
```

(d)

```
LOAD    R1, A
LOAD    R2, B
ADD     R1, R2
LOAD    R3, C
LOAD    R4, D
ADD     R3, R4
MUL     R1, R3
STORE   E, R1
```

(e)

```
PUSH    A
PUSH    B
ADD
PUSH    C
PUSH    D
ADD
MUL
STORE   E
```

E = (A + B) * (C + D)

	(a)	(b)	(c)	(d)	(e)
命令フェッチ回数(実行命令数)	4	6	7	8	8
主記憶データアクセス回数	11	5	7	5	5
必要メモリ量(ビット)	128	136	140	168	124
処理時間	**25**	**13**	**17**	**13**	**13**

図 1.5　各種機械命令形式の評価 (E=(A+B)＊(C+D) の場合)
(a) 2アドレス命令形式, (b) 1アドレス命令形式 (レジスタ-メモリ方式, (c) 1アドレス命令形式 (アキュムレータ方式), (d) 1アドレス命令形式 (ロードストア方式), (e) 0アドレス命令形式

E＝(A＋B)＊(C＋D)＋A＊B＊Cの実行

A ロードストアアーキテクチャ 演算は3オペランド

	プログラム				時間
①	LOAD	R1	A		2
②	LOAD	R2	B		2
③	LOAD	R3	C		2
④	LOAD	R4	D		2
⑤	ADD	R5	R1	R2	1
⑥	ADD	R6	R3	R4	1
⑦	MUL	R7	R5	R6	1
⑧	MUL	R2	R1	R2	1
⑨	MUL	R3	R3	R2	1
⑩	ADD	R3	R3	R7	1
⑪	STOR	E	R3		2
				計	16

B　ロードストア　２オペランド

①	LOAD	R1	A	2
②	LOAD	R2	B	2
③	LOAD	R3	C	2
④	LOAD	R4	D	2
⑤	MOVE	R5	R1	1
⑥	ADD	R5	R2	1
⑦	ADD	R4	R3	1
⑧	MUL	R4	R5	1
⑨	MUL	R1	R2	1
⑩	MUL	R1	R3	1
⑪	ADD	R1	R４	1
⑪	STORE	E	R１	2
			計	１７

①	LOAD	R1	A	2
②	LOAD	R2	B	2
③	LOAD	R3	C	2
④	LOAD	R4	D	2
⑤	ADD	R1	R2	1
⑥	ADD	R4	R3	1
⑦	MUL	R1	R4	1
⑧	MUL	R2	R3	1
⑨	LOAD	R3	A	2
⑩	MUL	R2	R3	1
⑪	ADD	R1	R2	1
⑫	STORE	E	R1	2
			計	１８

E＝(A＋B)＊(C＋D)+A*B*C

C　2アドレス命令形式　1リード／1ライトの場合

① ADD　D, C　7
② MUL　C, B　7
③ MUL　C, A　7
④ ADD　B, A　7
⑤ MUL　B, D　7
⑥ ADD　B, C　7
⑦ STOR E, B　4

計 46

D　スタックアーキテクチャ

	プログラム	時間
①	PUSH　A	2
②	PUSH　B	2
③	ADD	1
④	PUSH　C	2
⑤	PUSH　D	2
⑥	ADD	1
⑦	MUL	1
⑧	PUSH　A	2
⑨	PUSH　B	2
⑩	MUL	1
⑪	PUSH　C	2
⑫	MUL	1
⑬	ADD	1
⑭	STORE　E	2
	計	２２

E　レジスターメモリ

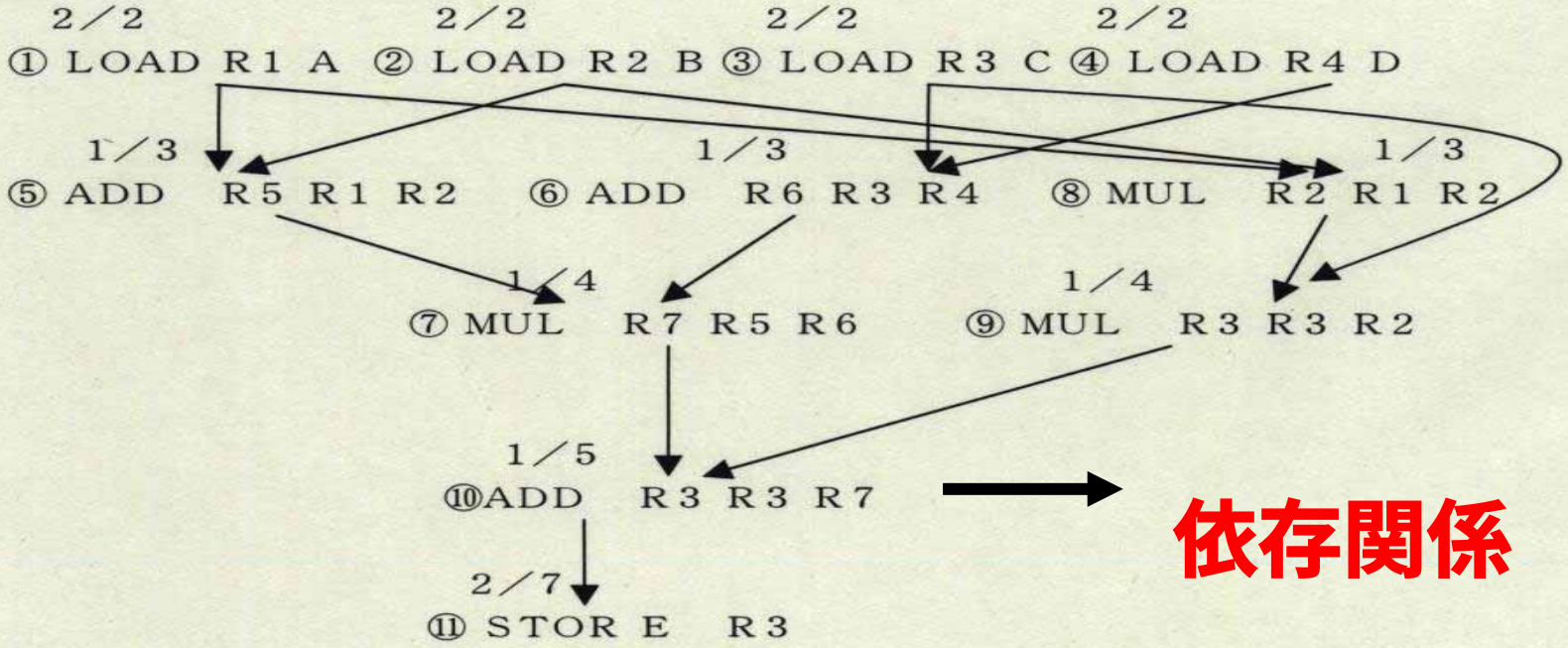
①	LOAD	R1	A	2
②	ADD	R1	B	3
③	LOAD	R2	C	2
④	ADD	R2	D	3
⑤	MUL	R1	R2	1
⑥	LOAD	R3	A	2
⑦	MUL	R3	B	3
⑧	MUL	R3	C	3
⑨	ADD	R1	R3	1
⑩	STORE	E	R1	2

計２２

①	LOAD	R1	A	2
②	MOVE	R2	R1	1
③	ADD	R2	B	3
④	LOAD	R3	C	2
⑤	MOVE	R4	R3	1
⑥	ADD	R4	D	3
⑦	MUL	R2	R4	1
⑧	MUL	R1	B	3
⑨	MUL	R1	R3	1
⑩	STORE	E	R1	2

計１９

命令実行時間／累積実行時間



依存関係

メモリ装置L／S２台，演算装置ALU２台の時（リストスケジューリング）

時間	L／S	L／S	ALU	ALU
9				
8	⑪			
7			⑩	
6			⑨	⑦
5			⑧	⑤
4				⑥
3	①	②		
2	—	—		
1	③	④		

並列処理を入れる

１．３．２ アドレッシングモード

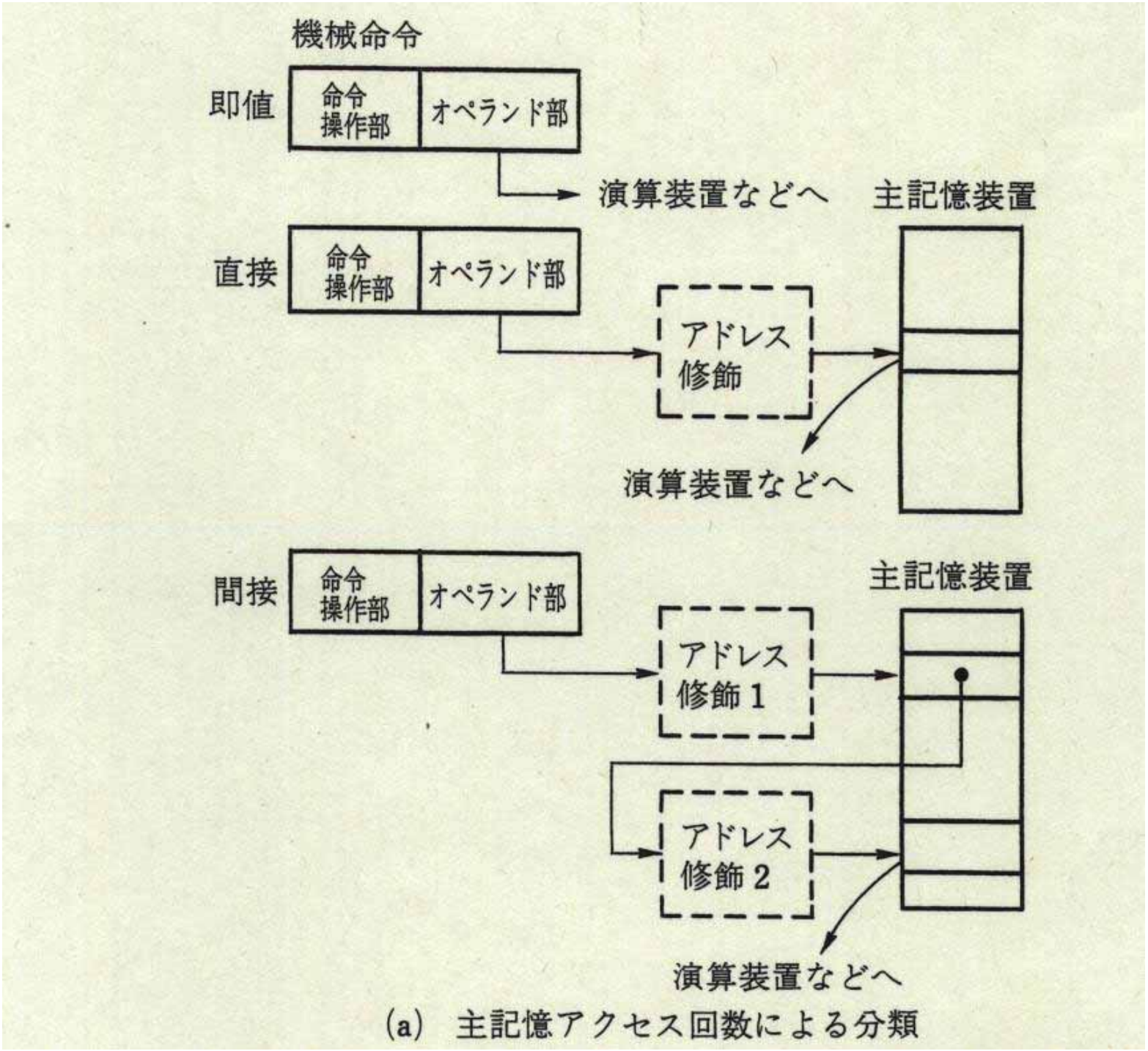
⑴ 即値，直接，間接

⑵ 絶対，相対

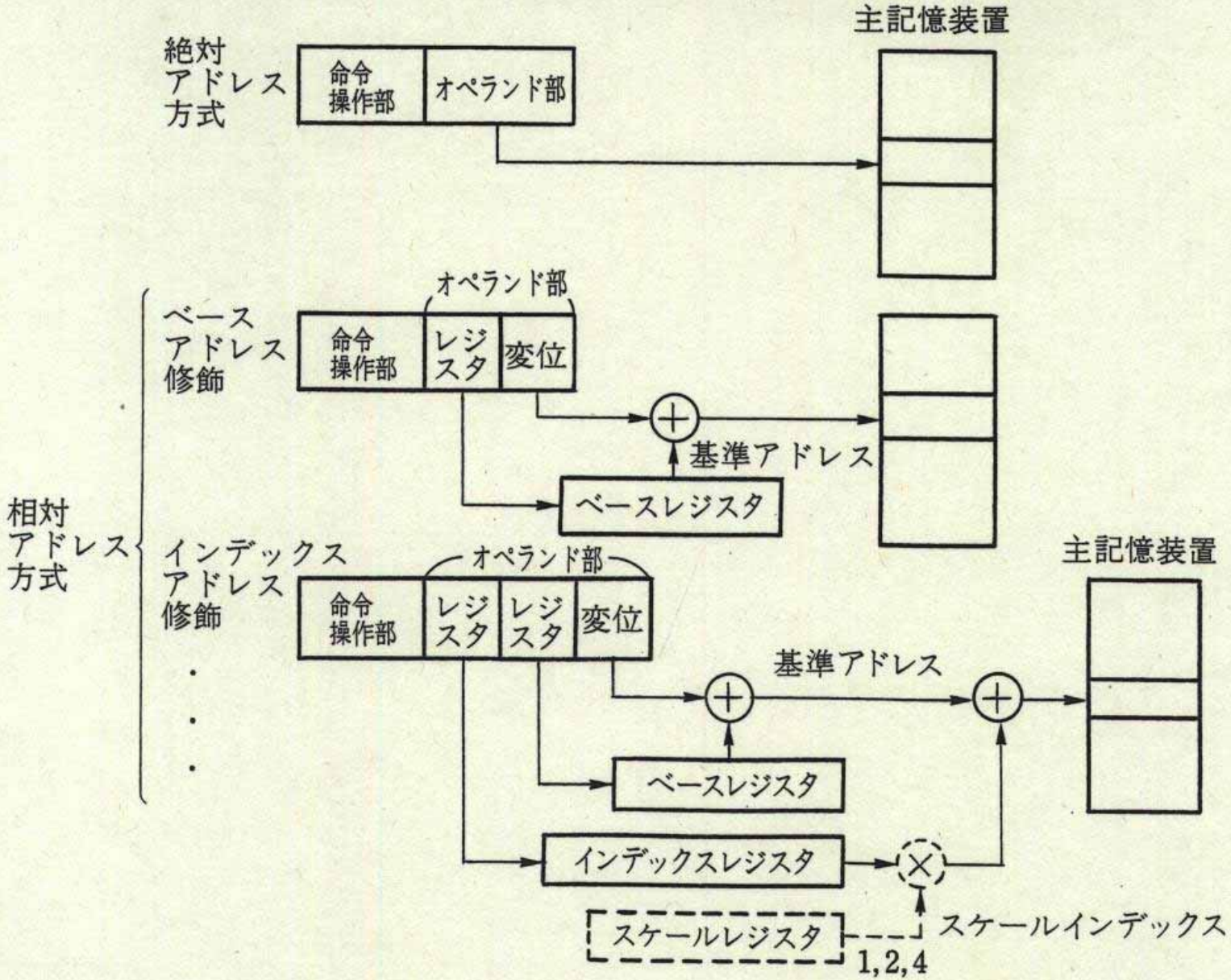
アドレス指定ビット数

参照の局所性：基準アドレスから変位

プログラムでのアドレス変更



(a)　主記憶アクセス回数による分類



(b) アドレス修飾による分類

図 1.4 アドレス方式

(3) アドレッシングモード

レジスタ Ri

即値　#I

利用頻度大

レジスタ間接　M (Ri)

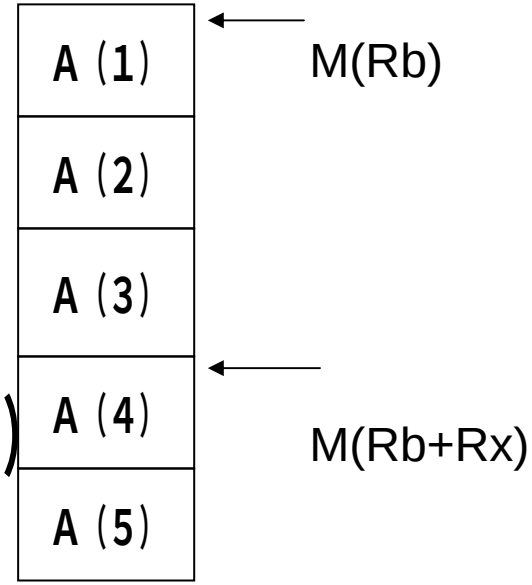
リターンアドレス

ディスプレースメント M (Ri＋D)

絶対　M (D)

インデックス M (Ri＋Rj＋D)

ベクトルや配列へのアクセス



A (i) : 4バイトデータ

Rb : ベースレジスタ

Rx : インデックスレジスタ

＋4ずつ増やす

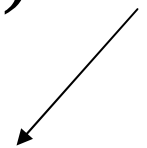
D:0にセット

反復プログラム

```
      S=0.0
      DO 10 I=1,128
   10 S=S+A(I)
```

```
      LOADW   Rs  #FW0.0
      LOADW   Rx  #IW0
    L ADDFW   Rs M(Rb+Rx+10)
      ADDIW   Rx  #IW4
      BRCMPLT Rx  #IW512  M(L)
      STORE   M(Rb+0)  Rs
```

分岐命令



自動インクリメント M (Ri＋)

自動デクリメント M (－Ri)

メモリ上でのスタック実現

自己相対 M (PC+D)

分岐先アドレスの決定

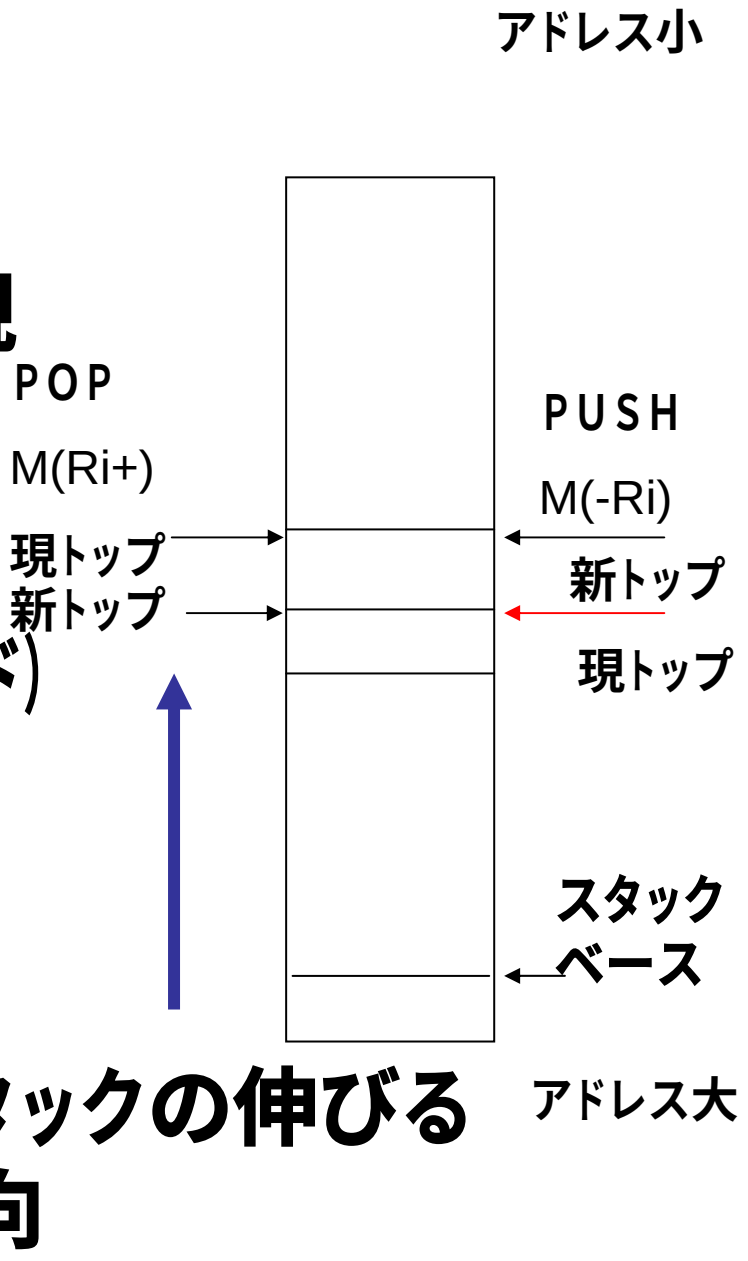
メモリ間接 M (**ベースモード**)

ポインタ変数

PUSH： STORE M(-R1) R0

POP： LOAD R0 M(R1+)

R1:**スタックトップレジスタ**



１．５ データ表現

(1) 固定小数点 (整数)

小数点位置はプログラム実行中固定：

そろばんと同じ

負の数の表現

２の補数と１の補数

(2) 浮動小数点数

単精度 (３２ビット)，倍精度 (６４ビット)

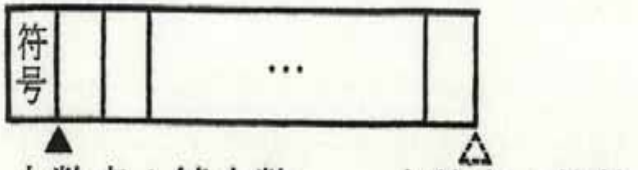
小数点位置を示す情報を個々のデータに保持

(3) １０進数

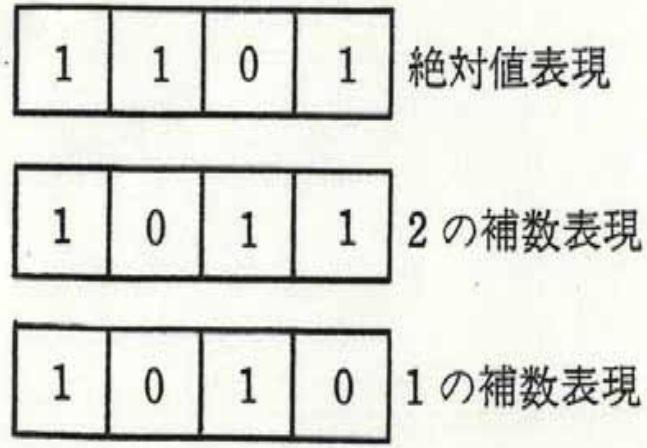
パックとアンパック形式

可変長演算

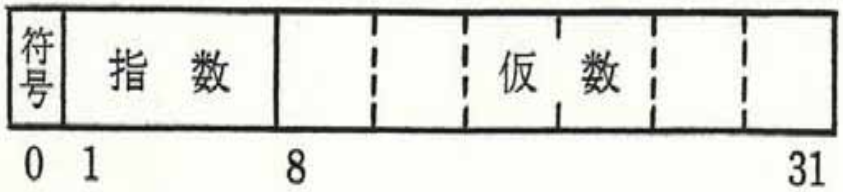
(4) 文字



(a)　固定小数点数の表現

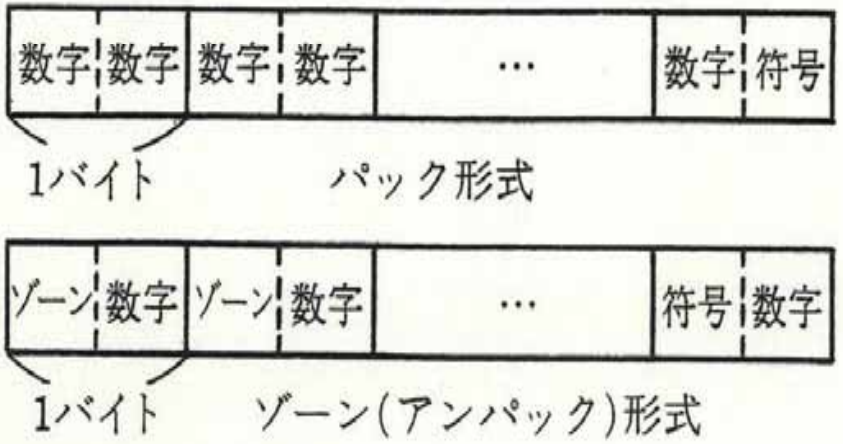


(b)　負数(−5の場合)の表現



符号：仮数の符号
仮数：16進6桁
指数：エクセス64符号

(c)　浮動小数点数の表現(IBM単精度)



(d)　10進数の表現(IBM)

図 1.6　各種の数表現

１０,9の補数

１０の補数　６２２

５３６
－３７８
１５８

536
1000
-378
-1000

５３６
６２２
－1000
１５８

符号桁
数値桁
0536
+9622
10158
無視

536
999
-378
-1000
+　1

9の補数６２１

５３６
＋６２１
１１５７
-１０００
+1
１５８

0536
+9621
10157
+1

2，1の補数

2の補数

Ｌ桁の正整数Ｘの2の補数 $\overline{X}$

$$X+\overline{X}=2^{L}$$

各桁反転＋1

１０１＋０１１＝１０００

1の補数

Ｌ桁の正整数Ｘの1の補数 $\overline{\overline{X}}$

$$X+\overline{\overline{X}}=2^{L}-1$$

各桁反転

１０１＋０１０＝１１１

符号桁

数値桁

+X:0X　0010(+2)

-X:1$\overline{X}$ +1011(-5)

1101(-3)

+X:0X　1101(-2)

-X:1$\overline{\overline{X}}$ +0101(+5)

10010

循環桁上げ

+1

浮動小数点数

１０$^{-100}$～１０^{100}

　２進固定小数点：２＊１００$\log_2 10$：６００ビット？

・ＩＢＭ単精度３２ビット　mr^e

　符号１ビット，指数e：７ビット（-64から+63），

　仮数m：２４ビット純小数（１６進６桁），

　基数r：１６

　正規化：仮数第１桁非零１～Ｆ

　１６$^{-65}$～(1-16^{-6}) １６^{63}

・ＩＢＭ倍精度６４ビット

　仮数：１６進１４桁（５６ビット）

3.7.3 浮動小数点数の標準化

IBM 形式の浮動小数点数の形式

IEEE（アメリカの電気電子学会．
アイトリプルイーと読む)）標準

(1)単精度浮動小数点

符号 s	指数 8 e	仮数 23 f

符号(s)１ビット，

指数(e)８ビット（げたばき１２７）

仮数(f) ２３ビット

基数２

①e=255 で f≠0 なら数の値は NaN（非数）

②e=255 で f=0 なら数の値は$(-1)^s\infty$

③0<e<255 なら数の値は　正規化数

$(-1)^s 2^{e-127}(1.f)$

④e=0、f≠0 なら数の値は

$(-1)^s 2^{-126}(0.f)$　非正規化数

⑤e=f=0 なら数の値は$(-1)^s 0$

(2)倍精度浮動小数点

符号(s)１ビット，

指数（ｅ）11 ビット（げたばき 1023），

仮数（ｆ）52 ビット

(3)IEEE 標準の特徴

①簡約表現

基数が２

仮数の第１桁は暗黙的に１

②非正規化数

正規化表現２^{-126}：最小数

非正規化（denormalize）表現可

③数値としての∞

∞の数値

１／０は∞を値

④非数の導入

$\sqrt{-1}$，0/0，∞-∞：NaN（Not a Number）

⑤丸めの方式の指定

４つの丸めの方式

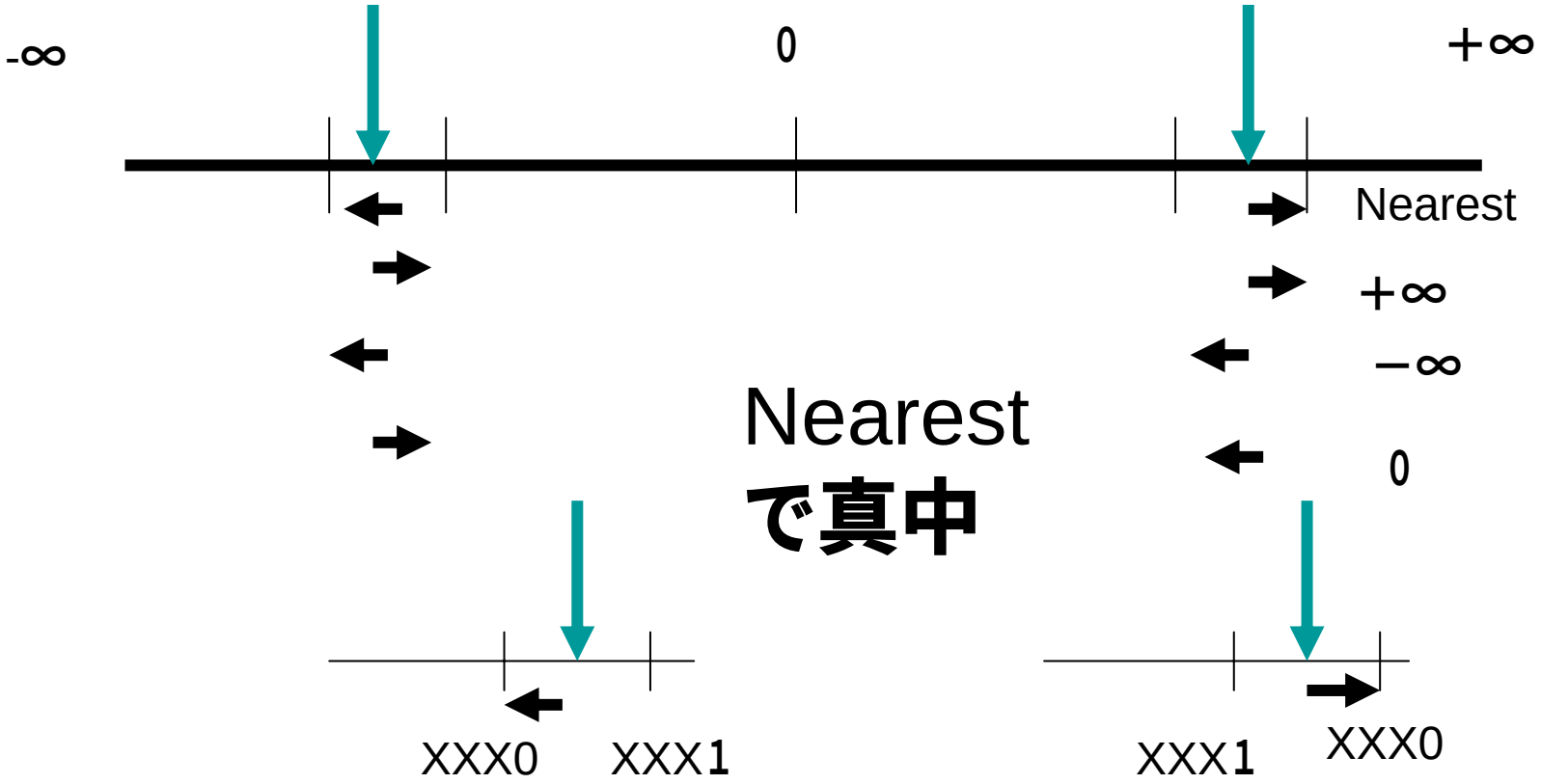
・最も近い数への丸め

・+∞方向への丸め

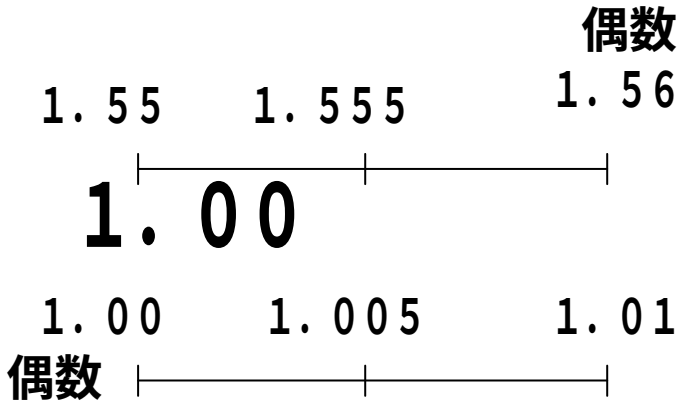
・-∞方向への丸め

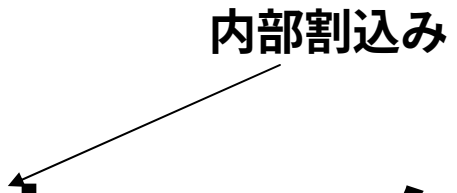
・0 方向への丸め

丸め方式



- $X_0 = X$、$X_1 = (X_0 - Y) + Y$,$X_2 = (X_1 - Y) + Y$
- $X_n = (X_{n-1} - Y) + Y$
- **四捨五入**
- X=1,Y=−0．555, X_0−Y＝1．555　1．56,
- X_1＝1．56−0．555＝1．005　1．01
- X_1−Y＝1．01＋0．555＝1．565　1．57,
- X_2＝1．57−0．555＝1．015　1．02
- **偶数に丸め**
- X_0-Y=1．555　1.56、
- X_1=1.56-0.555=1．005　1．00





⑥例外：マスク可能

- オーバフロー：正規化数より大２^{128}
- アンダフロー：非正規化数発生・演算
- 演算異常：NaN の発生時
- ゼロデバイド
- 精度異常（inexact）：1/３など

１．６ 機械命令セットの機能

⑴ 算術演算命令

演算種類：ADD,SUB,MUL,DIV

データタイプ：I,F,D

データ長：B,HW,W,DW，EW

命令形式：R-R、R-Memory

加算命令だけでも１０個以上

(2) 論理演算・ビット演算命令

ブール演算：AND (**論理積**) ,OR (**論理和**) ,NOT (**否定**)

AND	OR	NOT
0・0　0	0＋0　0	0　1
0・1　0	0＋1　1	1　0
1・0　0	1＋0　1	
1・1　1	1＋1　1	

シフト演算：論理シフト，算術シフト

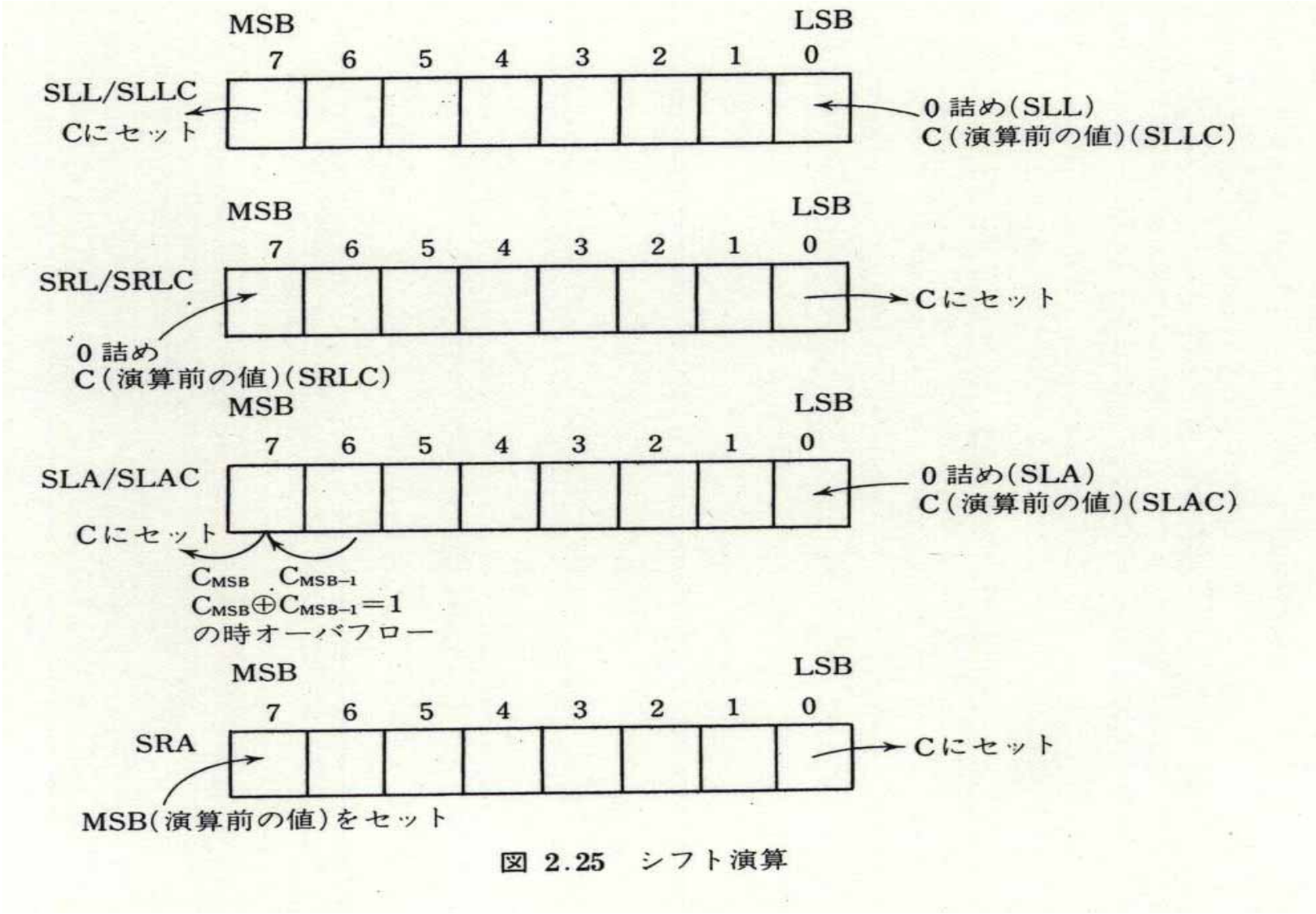
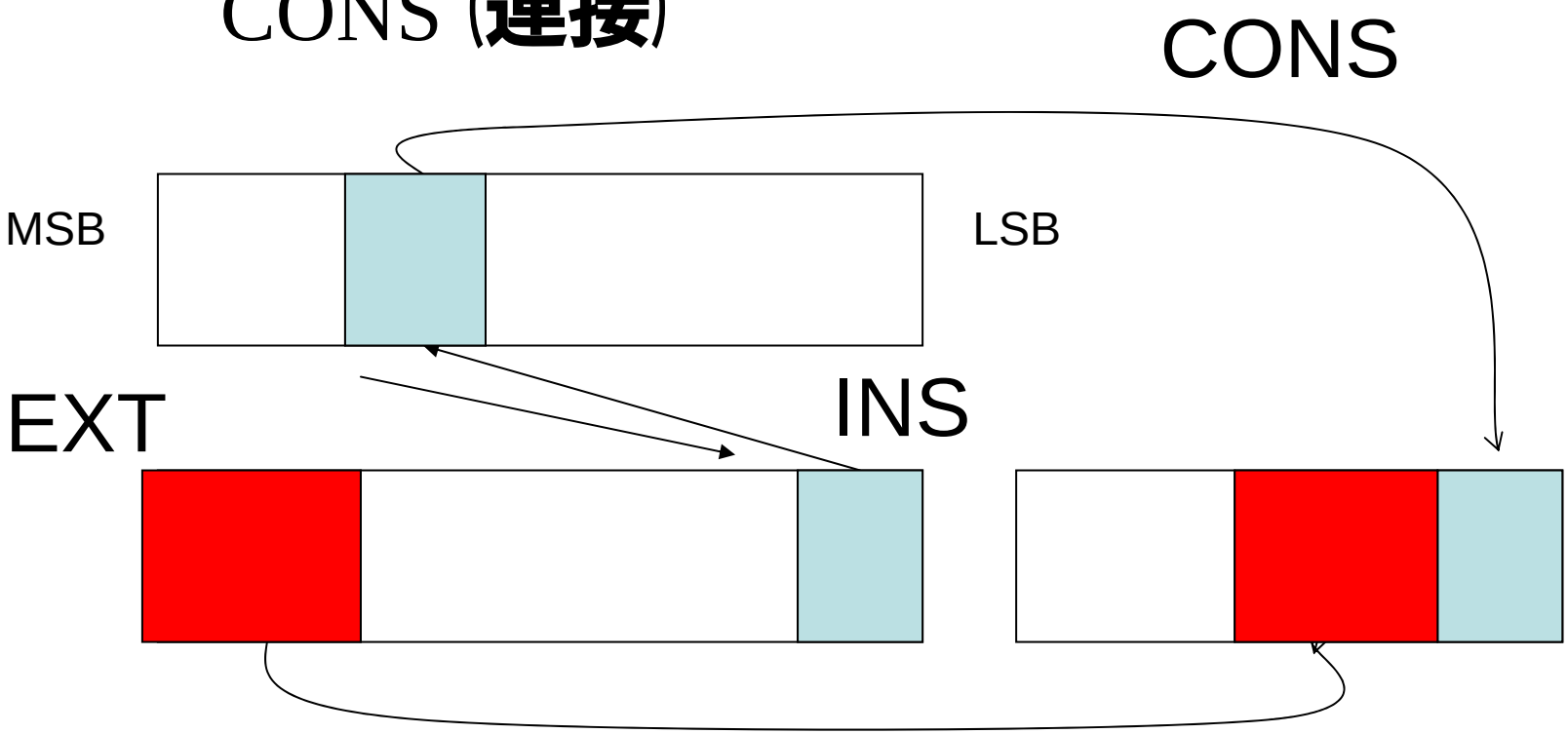


図 2.25　シフト演算

比較演算：E（Equal），LT（Less Than），
GT（Greater Than）

ビット演算：EXT（**抽出**），INS（**挿入**），
CONS（**連接**）



(3) 分岐命令

①無条件分岐

単純な無条件分岐：UCN

②手続き呼び出し：BAL

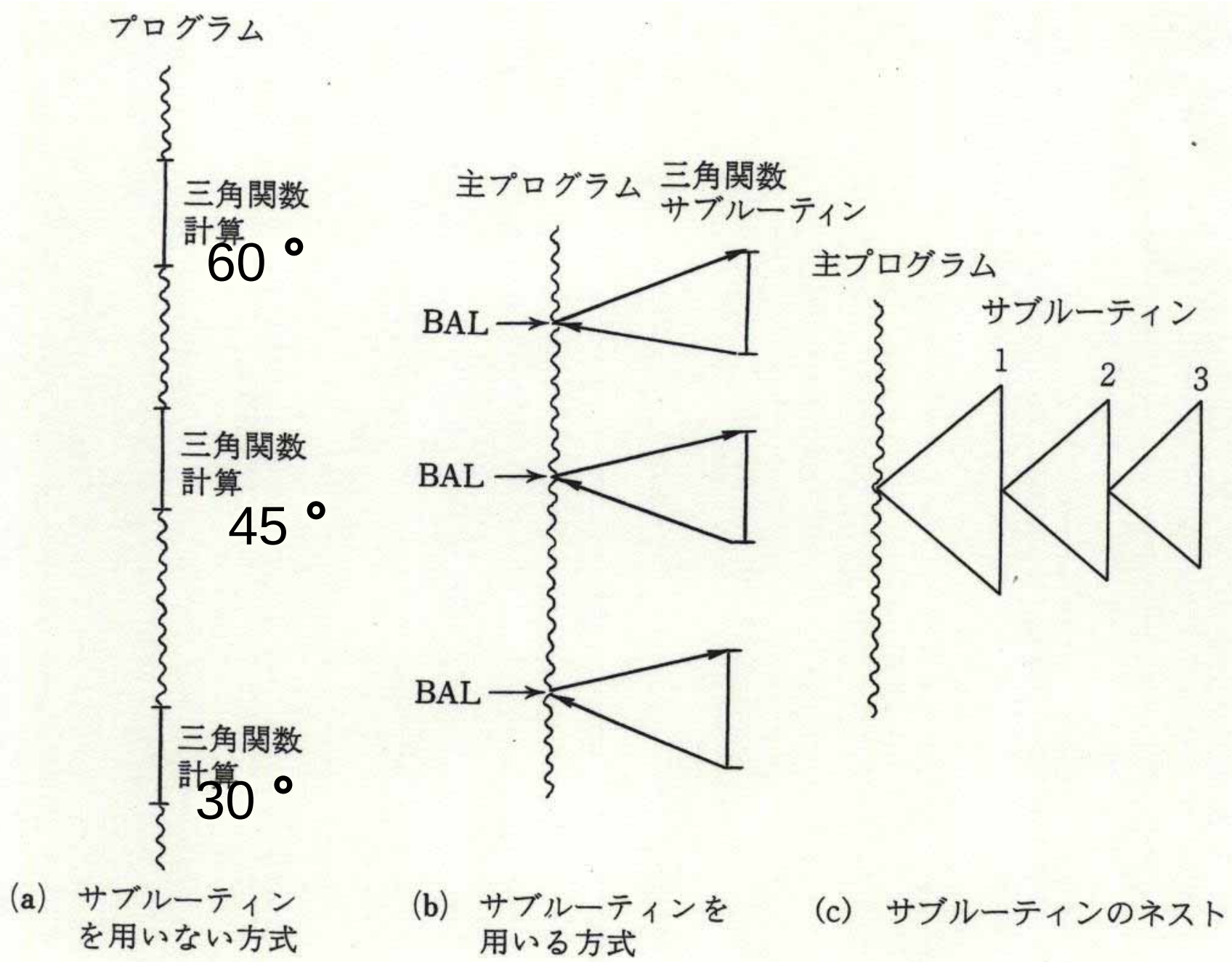
Branch and Link

パラメータパッシング

レジスタ待避と復旧

多重レベル呼出し

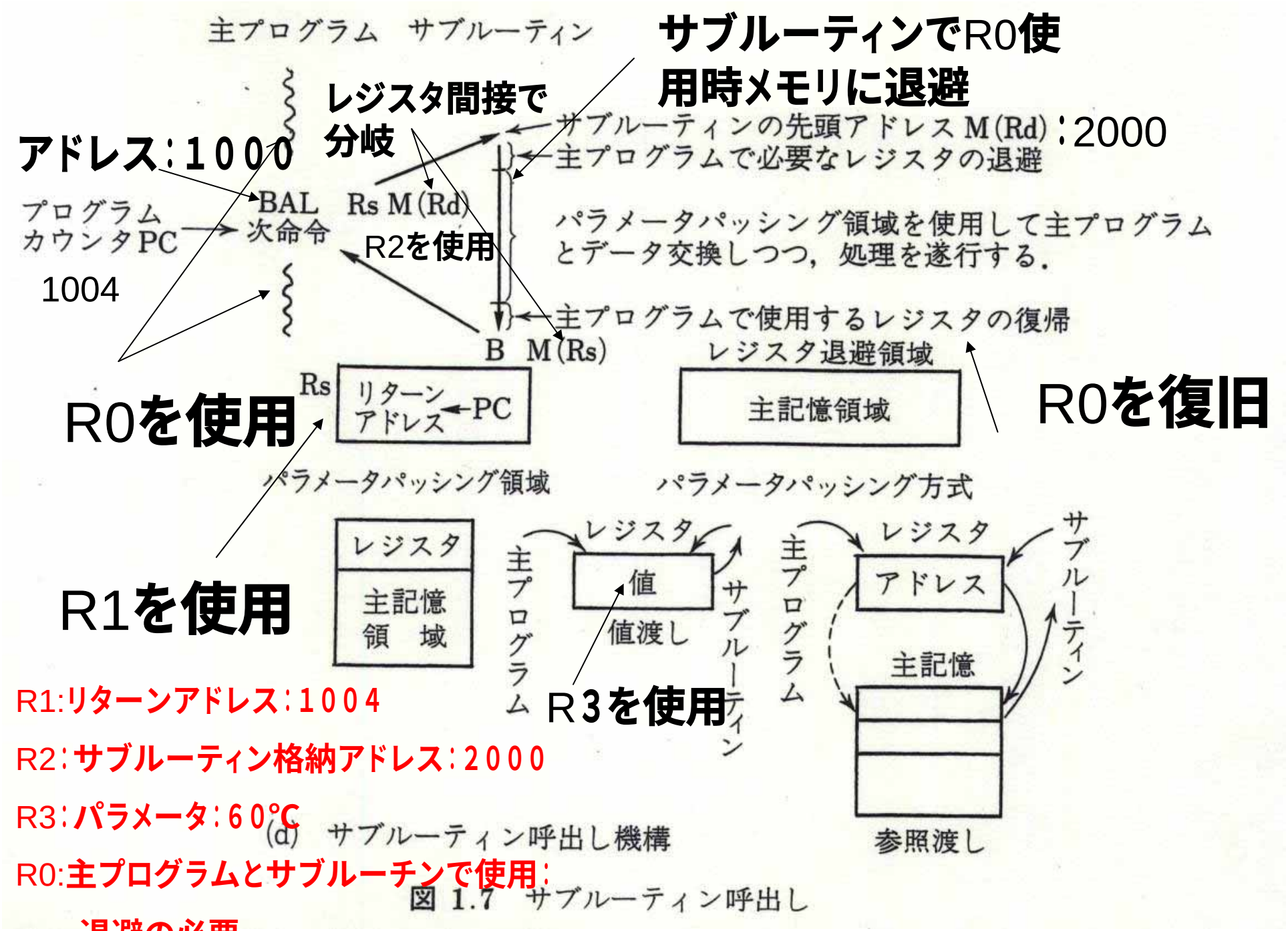
再帰的プログラム



(a) サブルーティンを用いない方式

(b) サブルーティンを用いる方式

(c) サブルーティンのネスト


主プログラム　サブルーティン
サブルーティンでR0使用時メモリに退避
レジスタ間接で分岐
アドレス：1000
サブルーティンの先頭アドレス M(Rd)：2000
主プログラムで必要なレジスタの退避
プログラムカウンタPC
BAL 次命令
Rs M(Rd)
R2を使用
1004
パラメータパッシング領域を使用して主プログラムとデータ交換しつつ，処理を遂行する．
主プログラムで使用するレジスタの復帰
B M(Rs)
R0を使用
Rs
リターンアドレス←PC
レジスタ退避領域
主記憶領域
R0を復旧
パラメータパッシング領域
パラメータパッシング方式
R1を使用
レジスタ
主記憶領域
主プログラム
値
値渡し
サブルーティン
R3を使用
アドレス
主記憶
参照渡し
R1:リターンアドレス：1004
R2：サブルーティン格納アドレス：2000
R3：パラメータ：60℃
R0:主プログラムとサブルーチンで使用：
退避の必要
(d)　サブルーティン呼出し機構
図 1.7　サブルーティン呼出し

再帰プログラム

A(I)の総和計算

```
Proc S (M,N)
 If M=N Then A(M)
 Else S(M,N)＝S (M, (M+N) /2)
          ＋S( (M+N) /2＋1, N)
```

反復プログラム

```
    S=0.0
    DO 10 I=1,128
10  S=S＋A(I)
```

```
/*１００万個のデータの整数加算を１０００回繰り返す．反復１０秒，再帰２１０
    秒*/
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int SUM(int i,int j);
void main(void)
{int i,j,k;
int s;
time_t   start, finish;
double   elapsed_time;
time( &start );
for(k=1;k<=10000;k++)
{
s=SUM(1,100000);
}
time( &finish );
printf("再帰総和=%d¥n",s);
elapsed_time = difftime( finish, start );
printf( "¥n再帰プログラムの実行に %f 秒かかりました。¥n",
    elapsed_time );
```

```
time( &start );
for(k=1;k<=10000;k++)
{s=0;
for(i=1;i<=100000;i++)
s=s+i;
}
time( &finish );
printf("反復総和=%d¥n",s);
elapsed_time = difftime( finish, start );
printf( "¥n反復プログラムの実行に %f 秒かかりました。¥n", elapsed_time );}
int SUM(int i,int j)
{if(i==j) return (i);
else return (SUM(i,(i+j)/2)+SUM((i+j)/2+1,j));
}
```

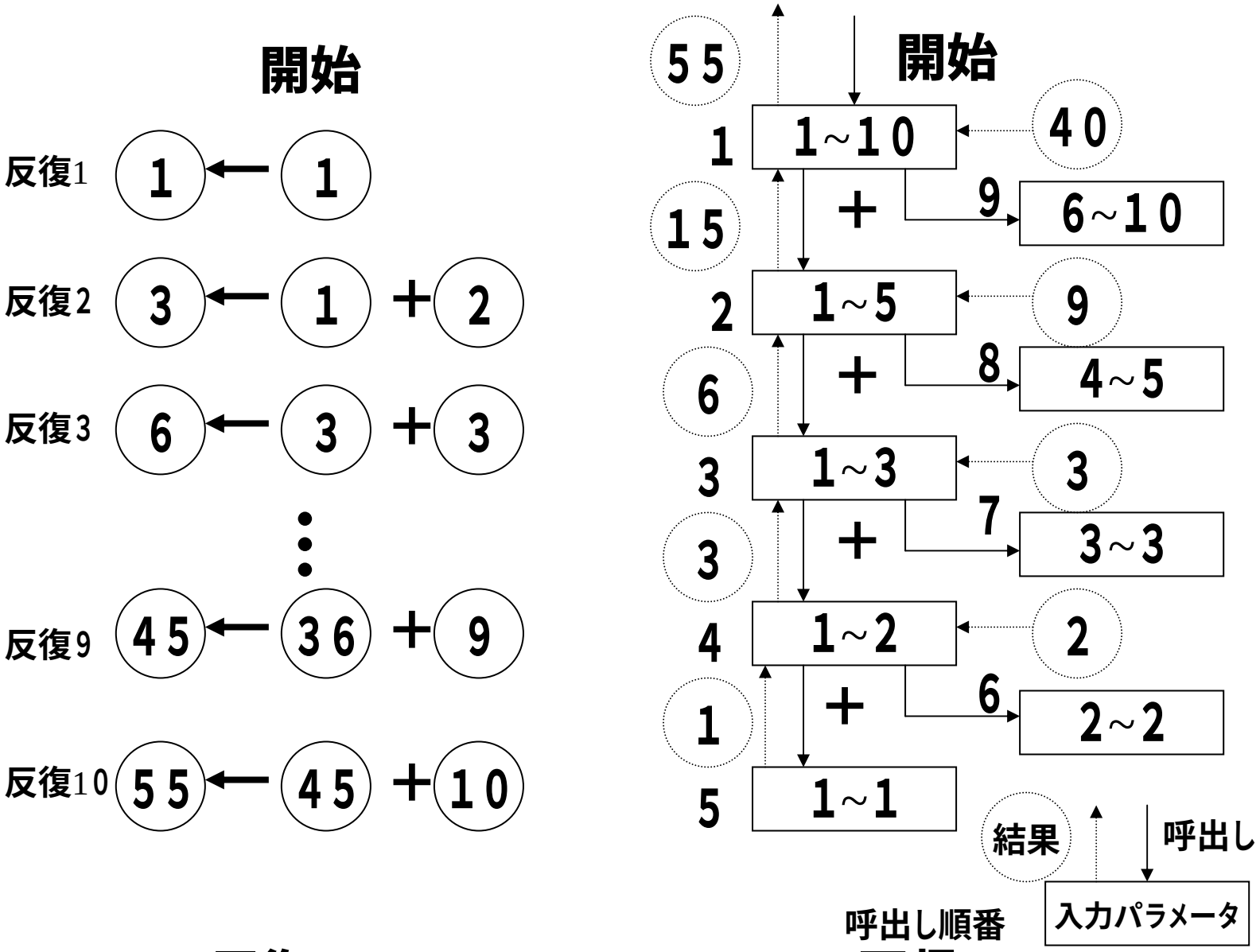


図1.10 (a)　**反復と再帰による1~10の総和計算**

制御スタックによる管理

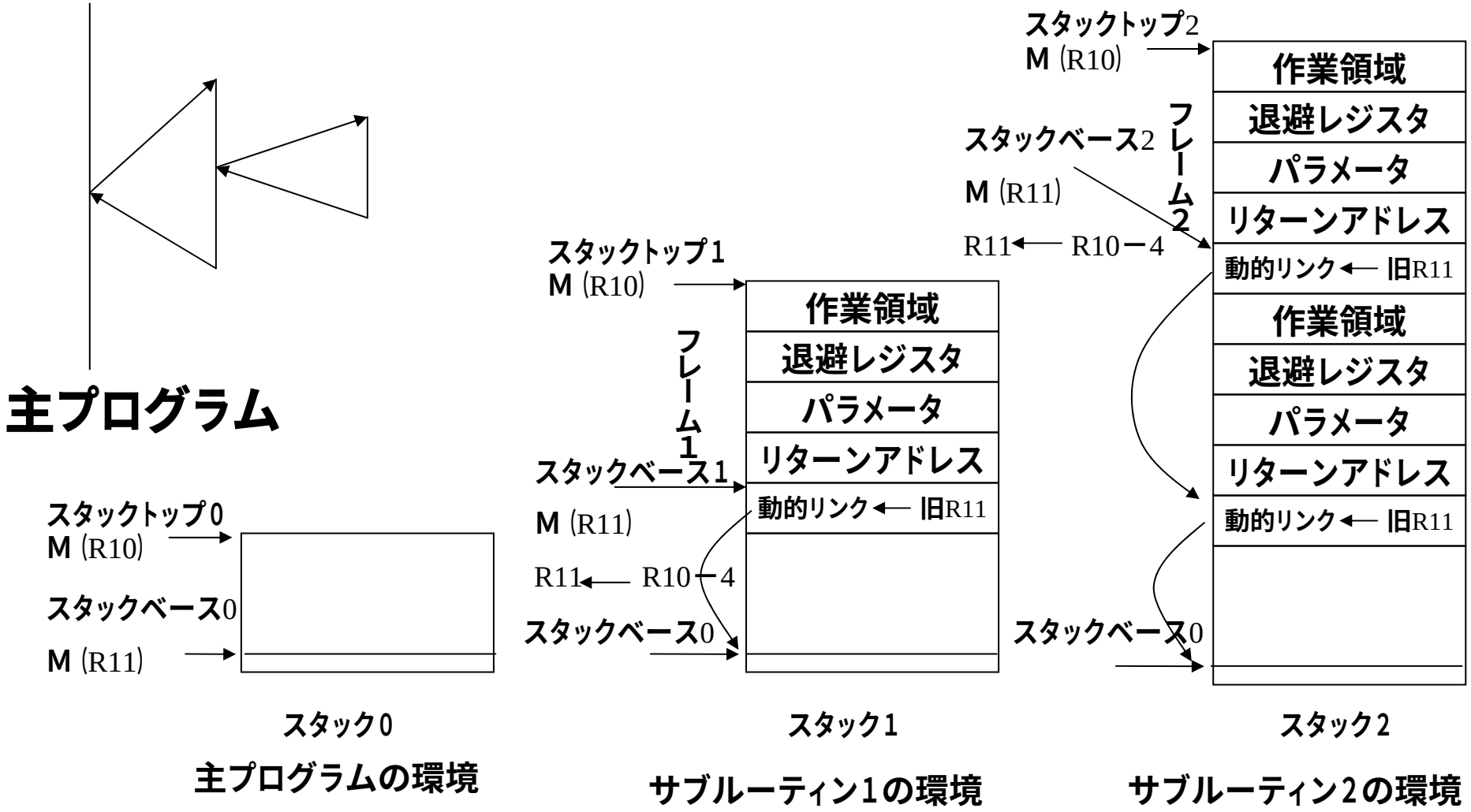


図1.10(b)サブルーティン呼出しの制御

③条件分岐

比較＆分岐：BRCMP　Compare and Branch

BRCMPE R0 R1 A： R0=R1**ならA**へ分岐

条件コード：BRC Branch on Condition

条件：Z：ゼロ，V：オーバフロー

N：負，C：キャリー

SUB R2 R0 R1 R0=R1**ならZ=1となる**

BRCZ A： Z=**1ならA**へ分岐

条件レジスタ：BRCR Branch on Condition Reg

汎用レジスタ内容0，1

CMPE R2 R0 R1： R0=R1**なら**R2=1

BRCR R2 A： R2=1**ならA**へ分岐

ADD R3 R2 R1
挿入されるとどうなるか？

(4) データ転送命令

ロードストア

ムーブ：MOVE

I／O

１．７入出力制御装置と入出力装置

標準入出力バス

インタフェース

パラレル

SCSI：ディスク

Centronics：プリンタ

シリアル

低速：１０Mbps

RS-232C，USB

高速：１００Mbps

Fiber Channel

IEEE１３９４

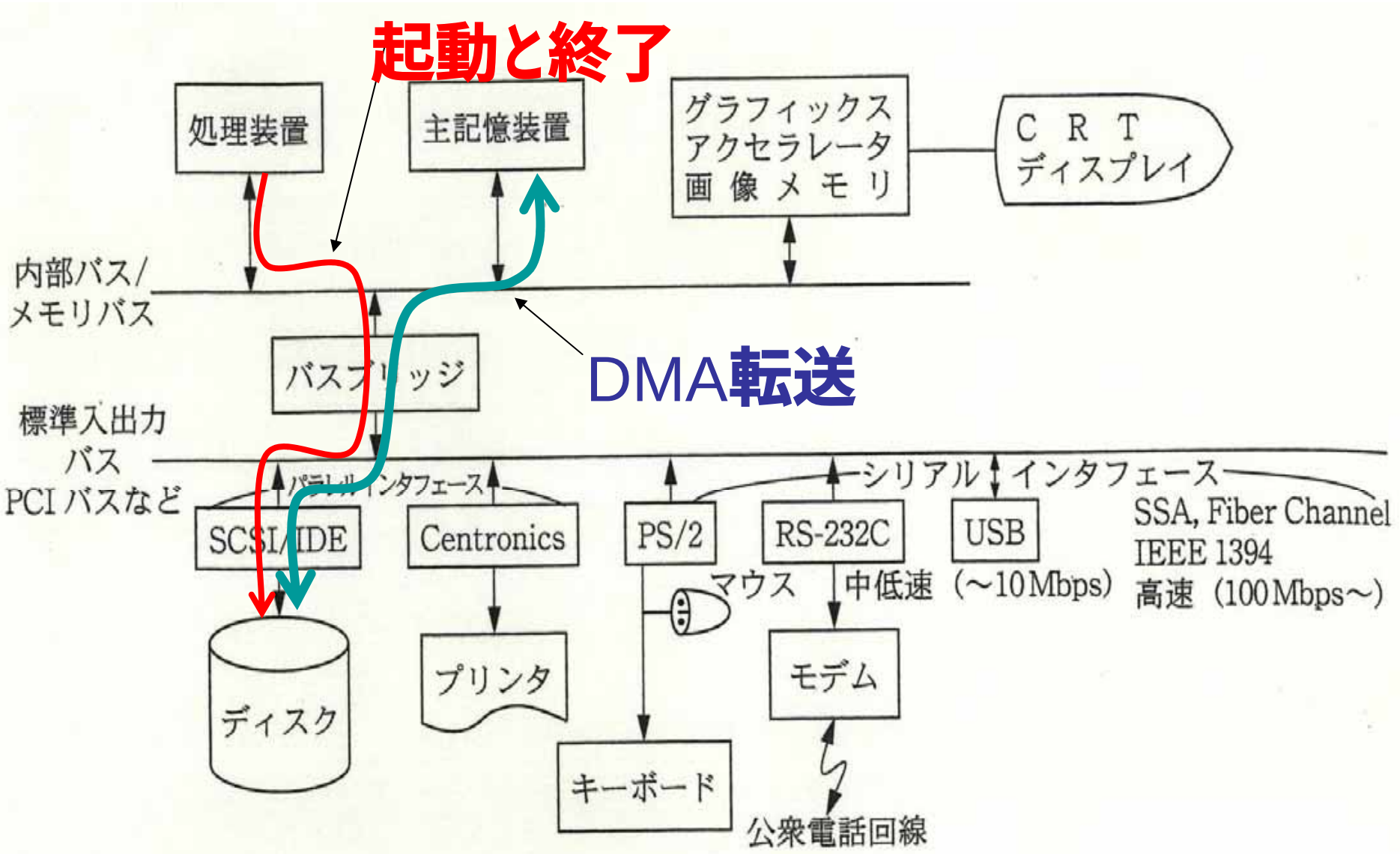


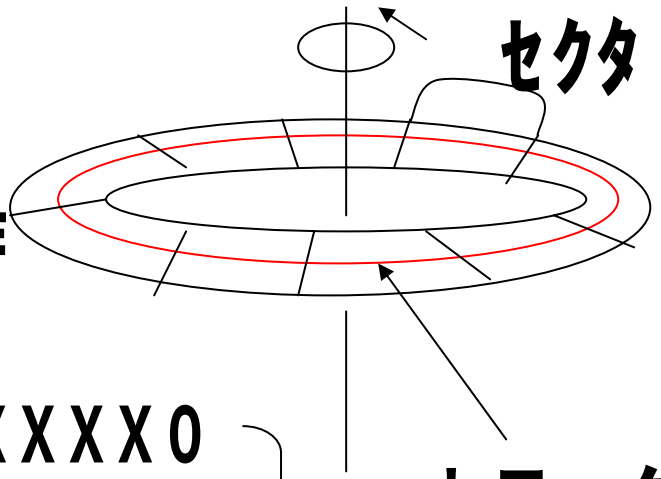
図 1.8 パーソナルコンピュータの入出力制御と装置

ハードディスクの制御

メモリマップドI／O

　メモリへのR／WでI／O操作

起動処理



トラック番号:	アドレスXXXX0	
セクタ番号:	アドレスXXXX1	
転送バイト数:	アドレスXXXX2	Write
転送メモリアドレス:	アドレスXXXX4	
ディスク状態:	アドレスXXXX8	Read

　故障状態

　使用中

　データアクセスOK

- **ハードディスク**
  年率：６０％で容量増加
  トラック数：5,000~30,000
  セクタ数：100~500
  セクタバイト数：**５１２**B
  シーク（seek）**時間**：**５**~12msec
  回転待ち（rotation　latency）
  　3,600~15,000RPM（Rotations　Per Minute）
  　3,600RPM**で**1/（2＊60)=8msec
  　7,200RPM**で４**msec

データ転送
  プログラムモードバス　ＶＳ　ＤＭＡ
    データアクセスＯＫチェック→データ転送
    ＤＭＡ　プロセッサと独立に動作
終了処理
  ポーリング
      終わったかのﾁｪｯｸ
  割込み
      終わりましたの報告

１．８ 割込み

①外部割込み：実行中のプログラムとは無関係な事象で生起

故障

入出力終了

②内部割込み：実行中のプログラムに起因した事象で生起

演算で誤り

記憶領域外の参照

トレース

ＯＳ呼出し（I/Oアクセス要求など）

ページフォールト

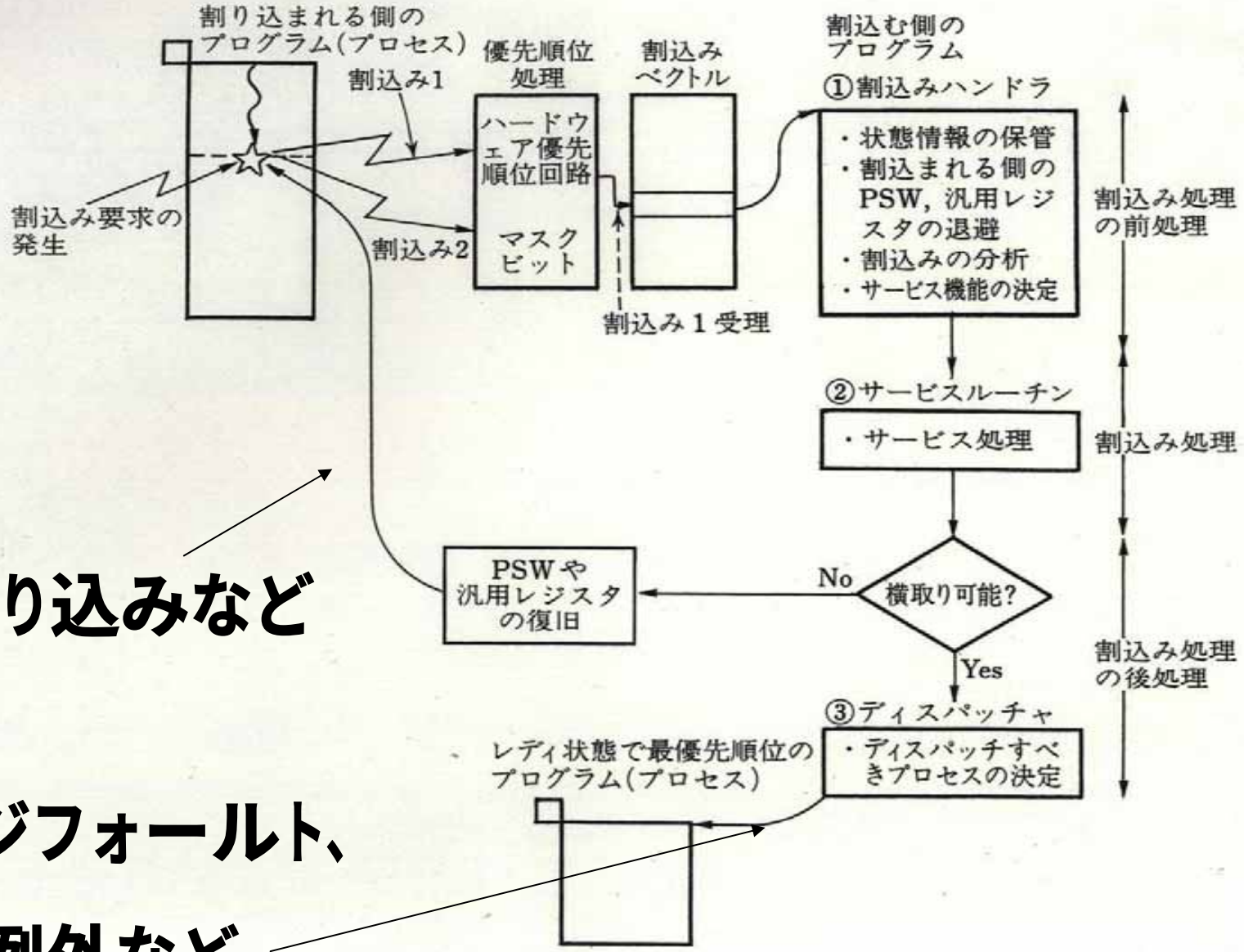


図 5.13　割込み処理の手続

I/O割り込みなど

ページフォールト、

演算例外など

１．９ 階層構造

⑴ レベル

①概念レベル

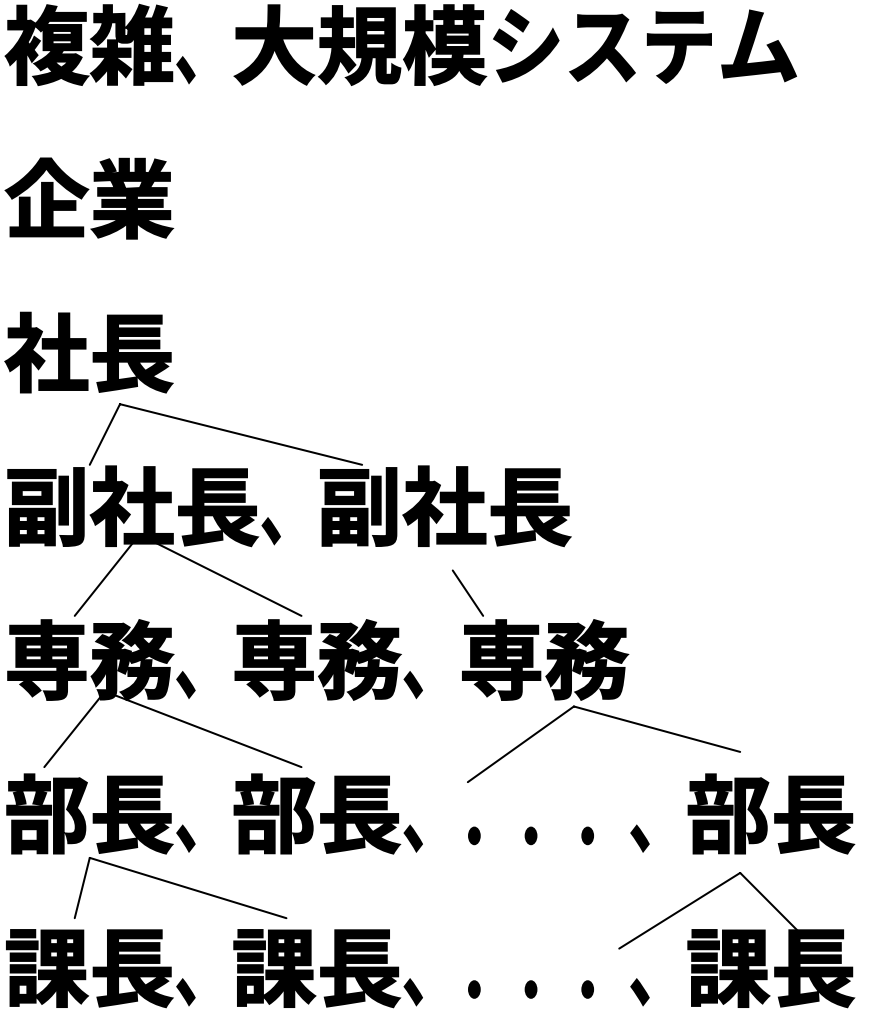
What to do：Σ A(I)

②高級言語レベル

How to do

反復プログラム

```
      S=0.0
      DO 10 I=1,128
10    S=S＋A(I)
```



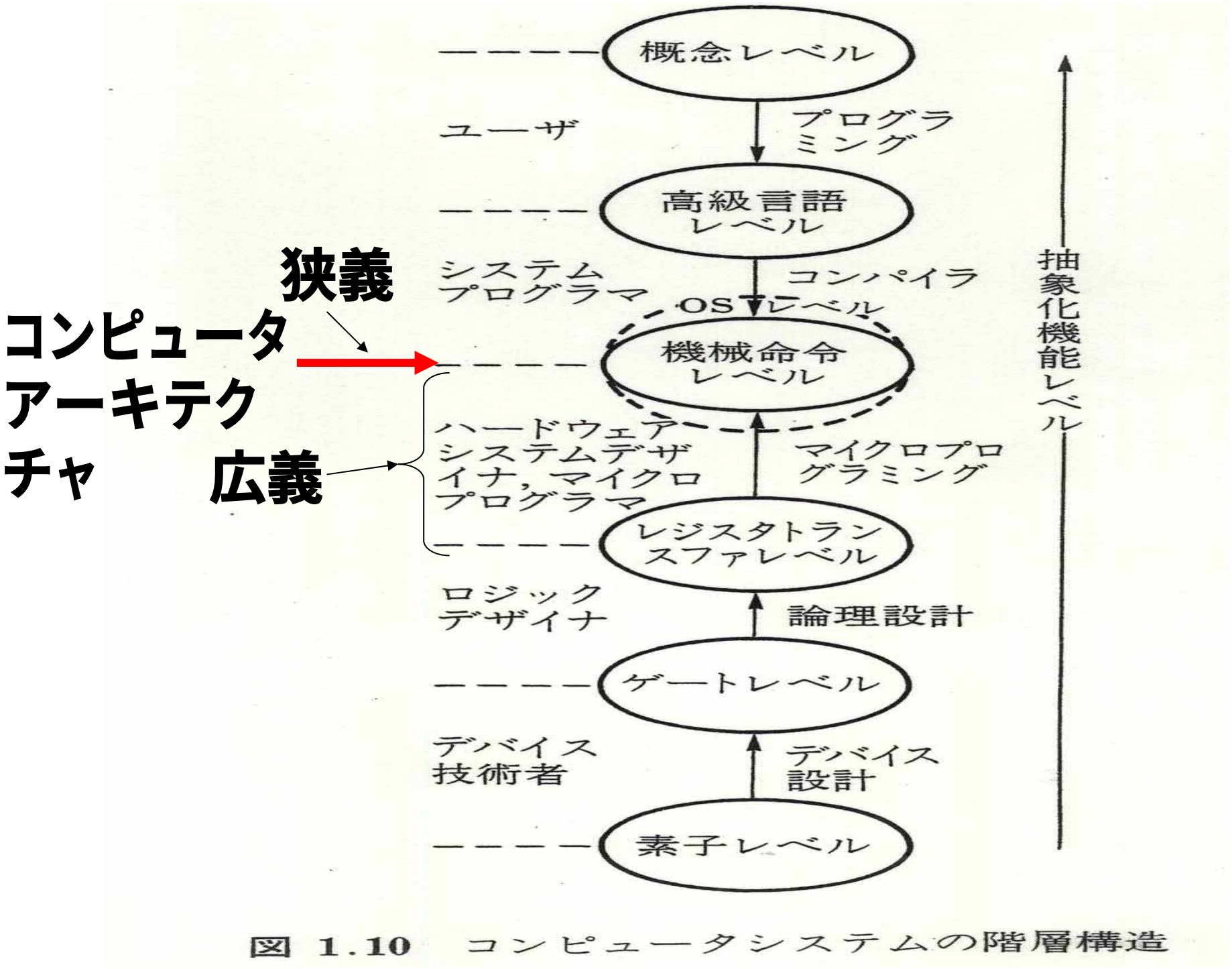


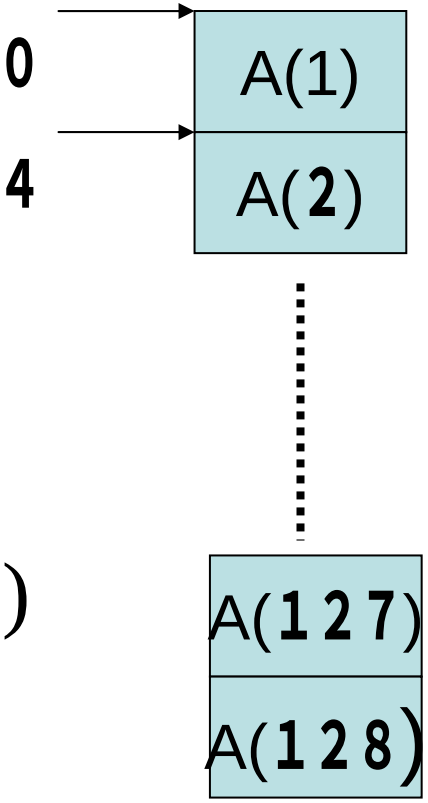
図 1.10　コンピュータシステムの階層構造

③機械命令レベル

コンピュータアーキテクチャ

ハードとソフトのインタフェース

```
  LOADW   Rs  #FW0.0
  LOADW   Rx  #IW0
L ADDFW   Rs M(Rb+Rx+10)
  ADDIW   Rx  #IW4
  BRCMPLT Rx  #IW512  M(L)
  STORE   M(Rb+0)  Rs
```



④RTレベル

機能ブロックでの設計

⑤ゲートレベル

AND, OR, NOTでの論理設計

⑥素子レベル

トランジスタの設計

MOS, CMOS

④RTレベル

1　PCリード, DBUS出力

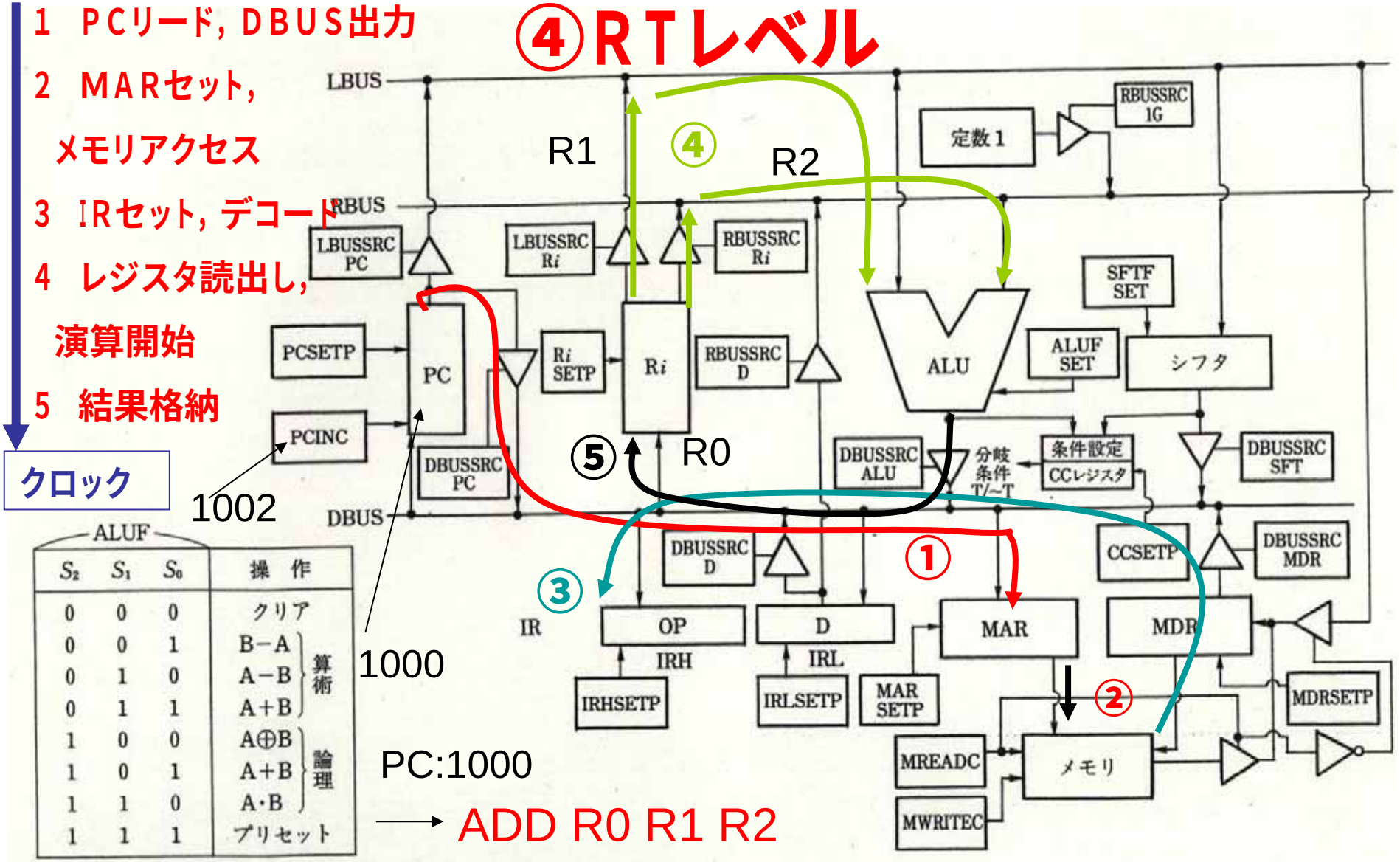
2　MARセット,
　メモリアクセス

3　IRセット, デコード

4　レジスタ読出し,
　演算開始

5　結果格納

クロック



ALUF

S_2	S_1	S_0	操　作	
0	0	0	クリア	
0	0	1	B−A	算術
0	1	0	A−B	算術
0	1	1	A+B	算術
1	0	0	A⊕B	論理
1	0	1	A+B	論理
1	1	0	A·B	論理
1	1	1	プリセット	

算術論理演算装置

図 2.20　データパス

⑤ゲートレベル

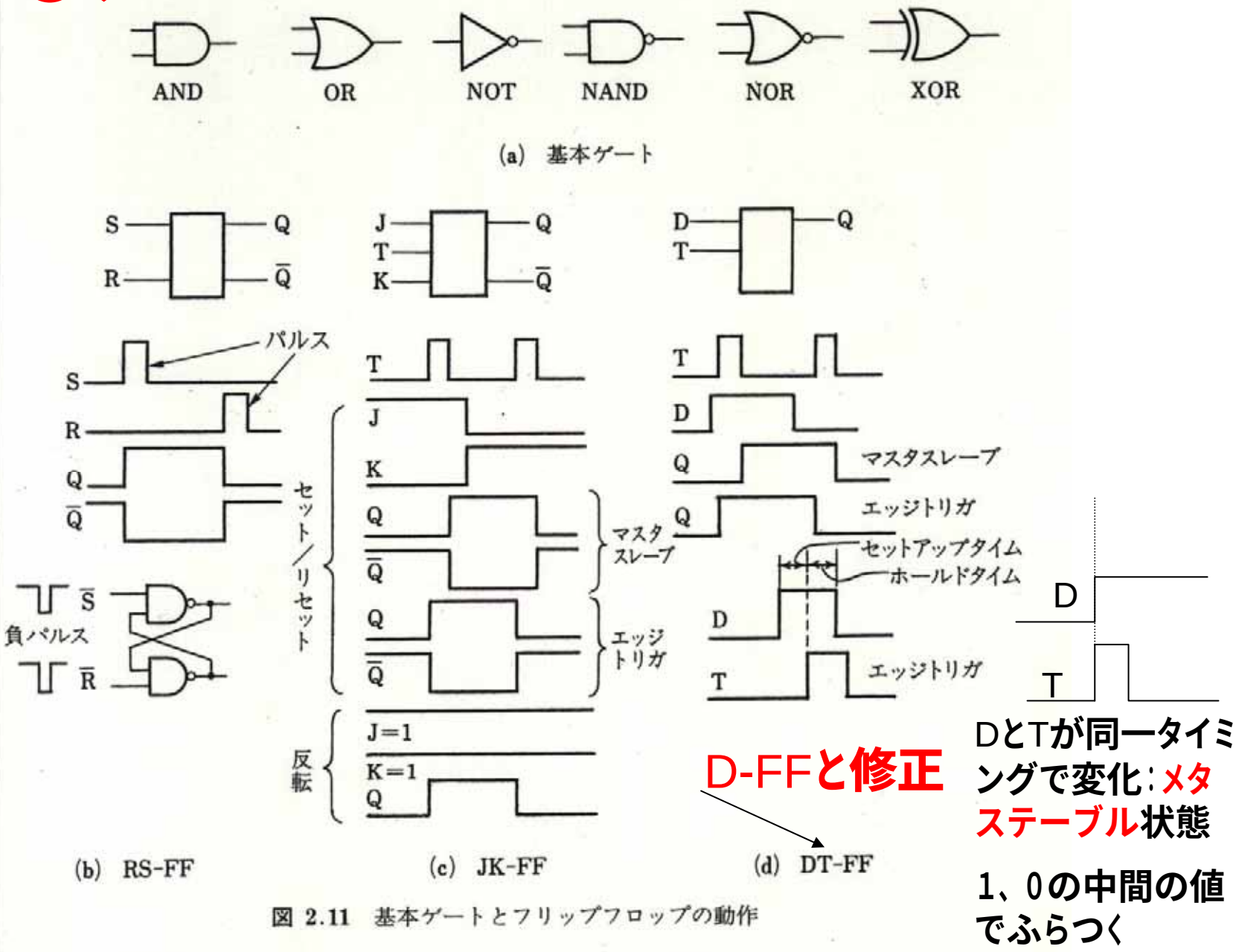


図 2.11 基本ゲートとフリップフロップの動作

D-FFと修正

DとTが同一タイミングで変化：メタステーブル状態

1、0の中間の値でふらつく

組み合わせ回路の例

入力			出力	
X	Y	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

図 2.3 加算器設計のための真理値表

真理値表→積和標準形
→簡単化

$$S=\bar{X}\cdot\bar{Y}\cdot C_{in}+\bar{X}\cdot Y\cdot\bar{C}_{in}+X\cdot\bar{Y}\cdot\bar{C}_{in}+X\cdot Y\cdot C_{in}$$

$$C_{out}=\bar{X}\cdot Y\cdot C_{in}+X\cdot\bar{Y}\cdot C_{in}+X\cdot Y\cdot\bar{C}_{in}+X\cdot Y\cdot C_{in}$$

$$S = X \oplus Y \oplus C_{in}$$

$$C_{out} = (X \oplus Y)C_{in} + XY$$

⊕: 排他的論理和

簡単な順序回路の例

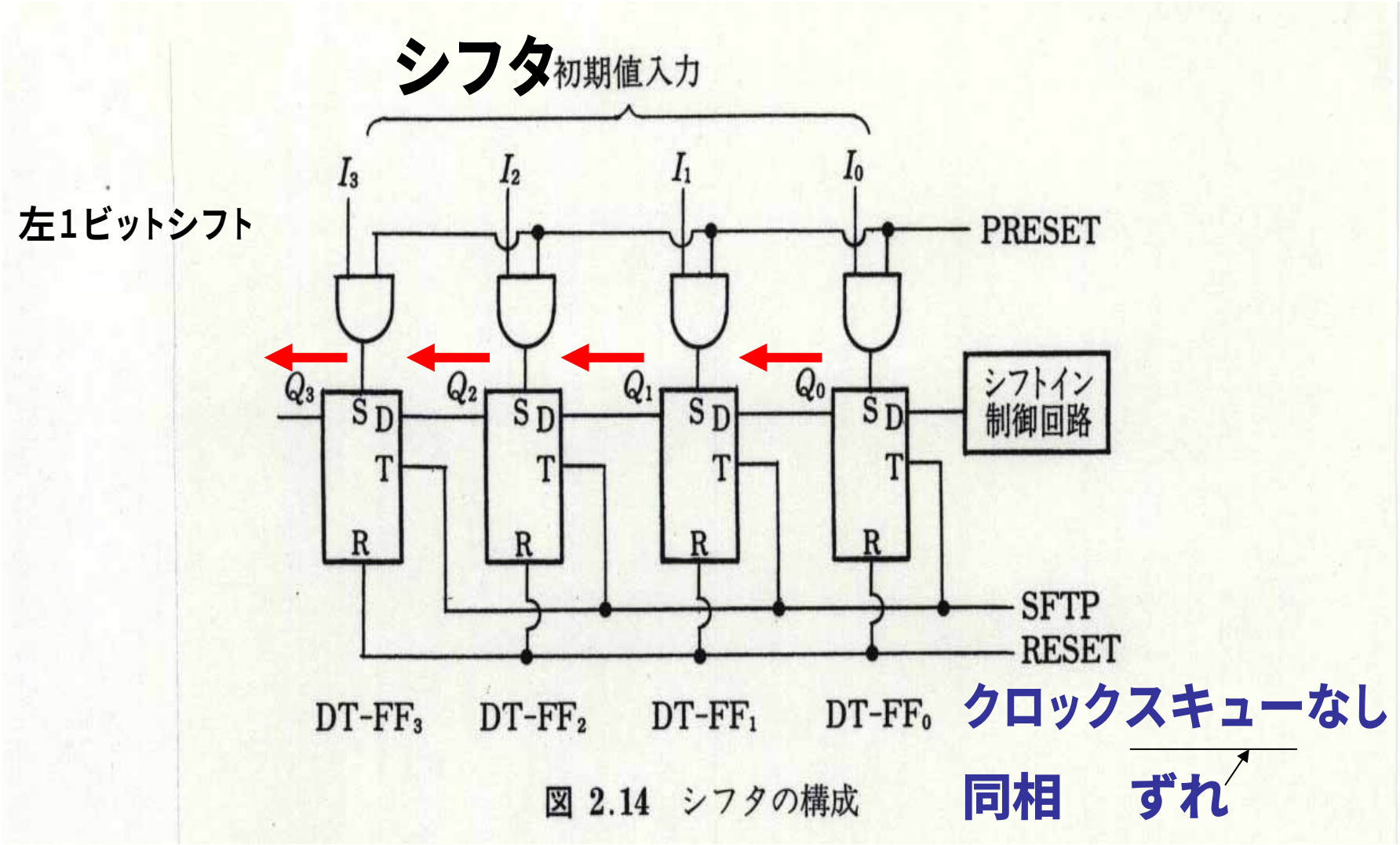


図 2.14　シフタの構成

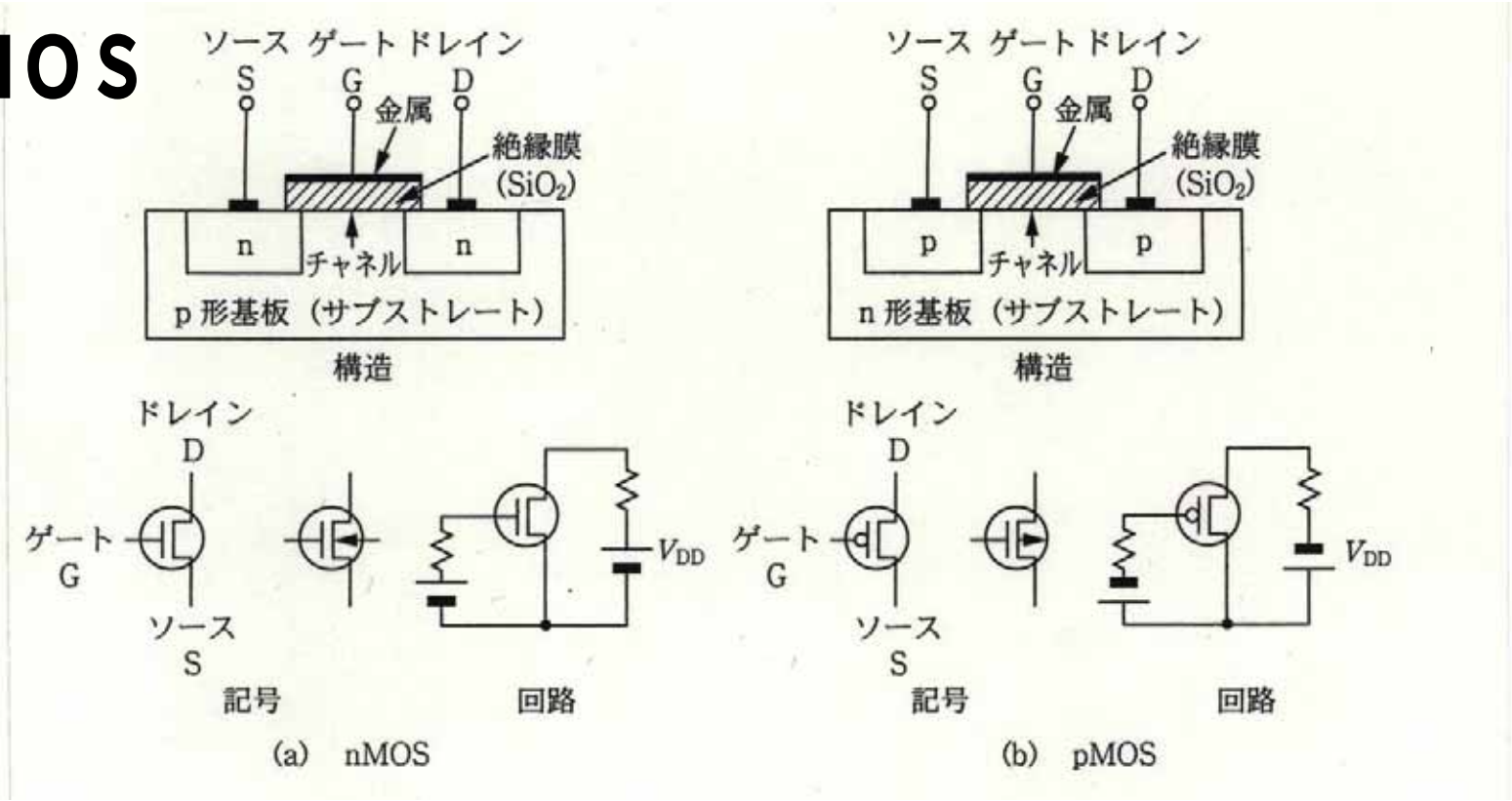
取り込んだ入力値が出力に
現れるまでの時間

$\geq$

ホールドタイム

⑥素子レベル

MOS



電界効果を利用（FET）

電子、正孔いずれか一方が関与

Metal Oxide Semiconductor

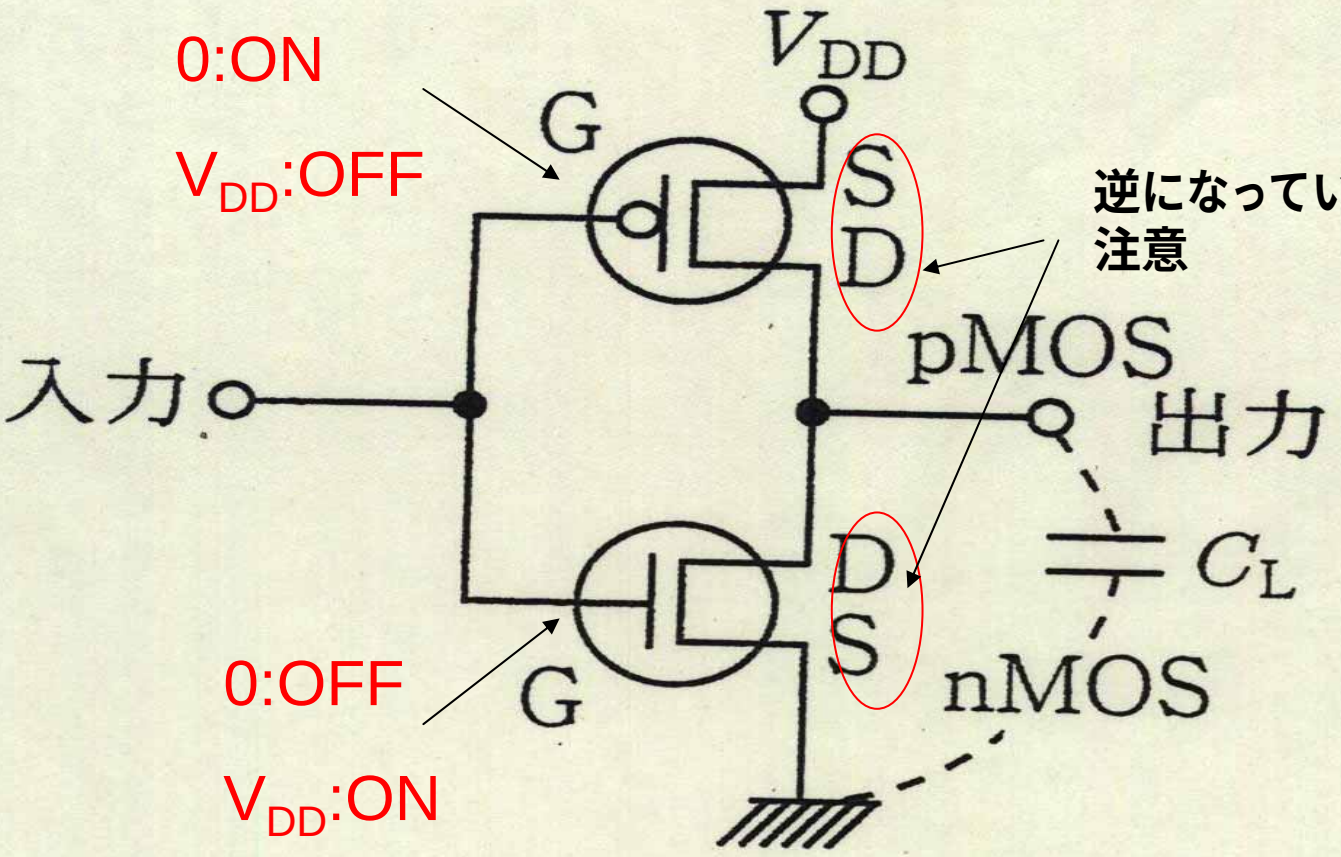
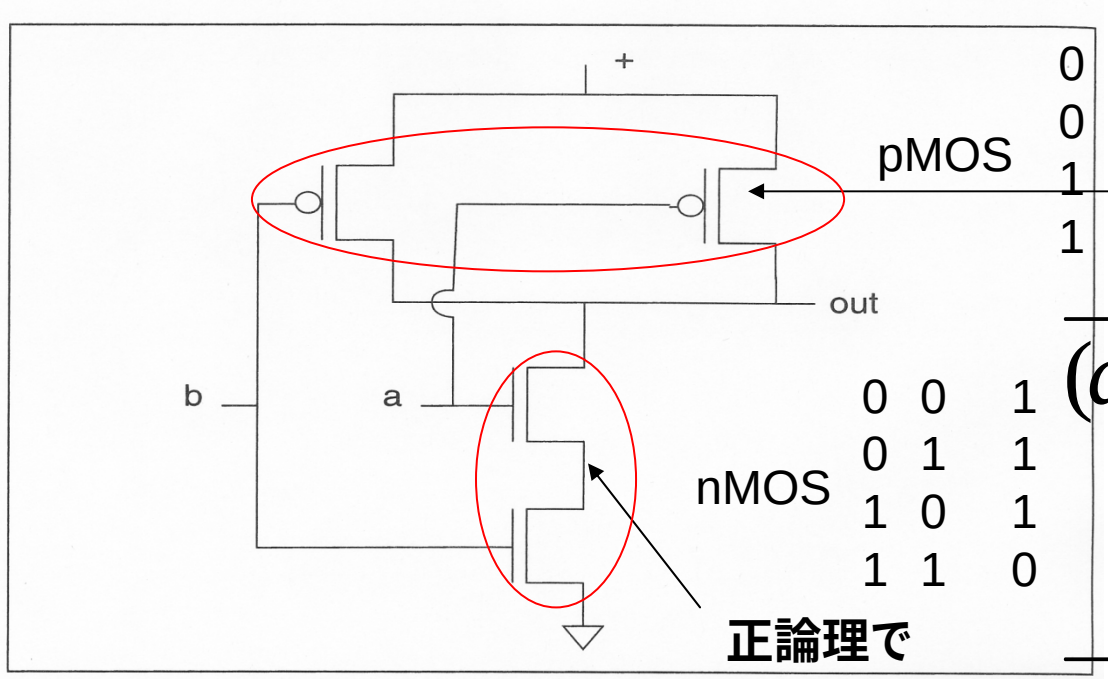


図 2.32 CMOS 否定回路

NAND



pMOS

0	0	1
0	1	1
1	0	1
1	1	0

負論理でNOR

$$\overline{(a \cdot b)} = \bar{a} + \bar{b}$$

nMOS

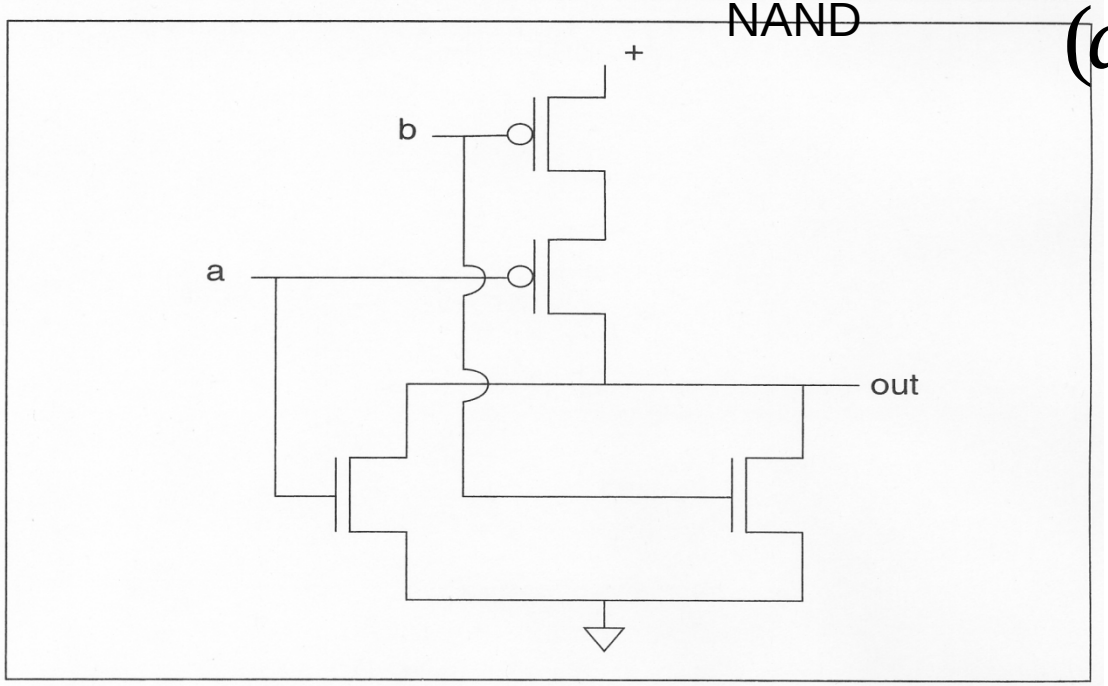
0	0	1
0	1	1
1	0	1
1	1	0

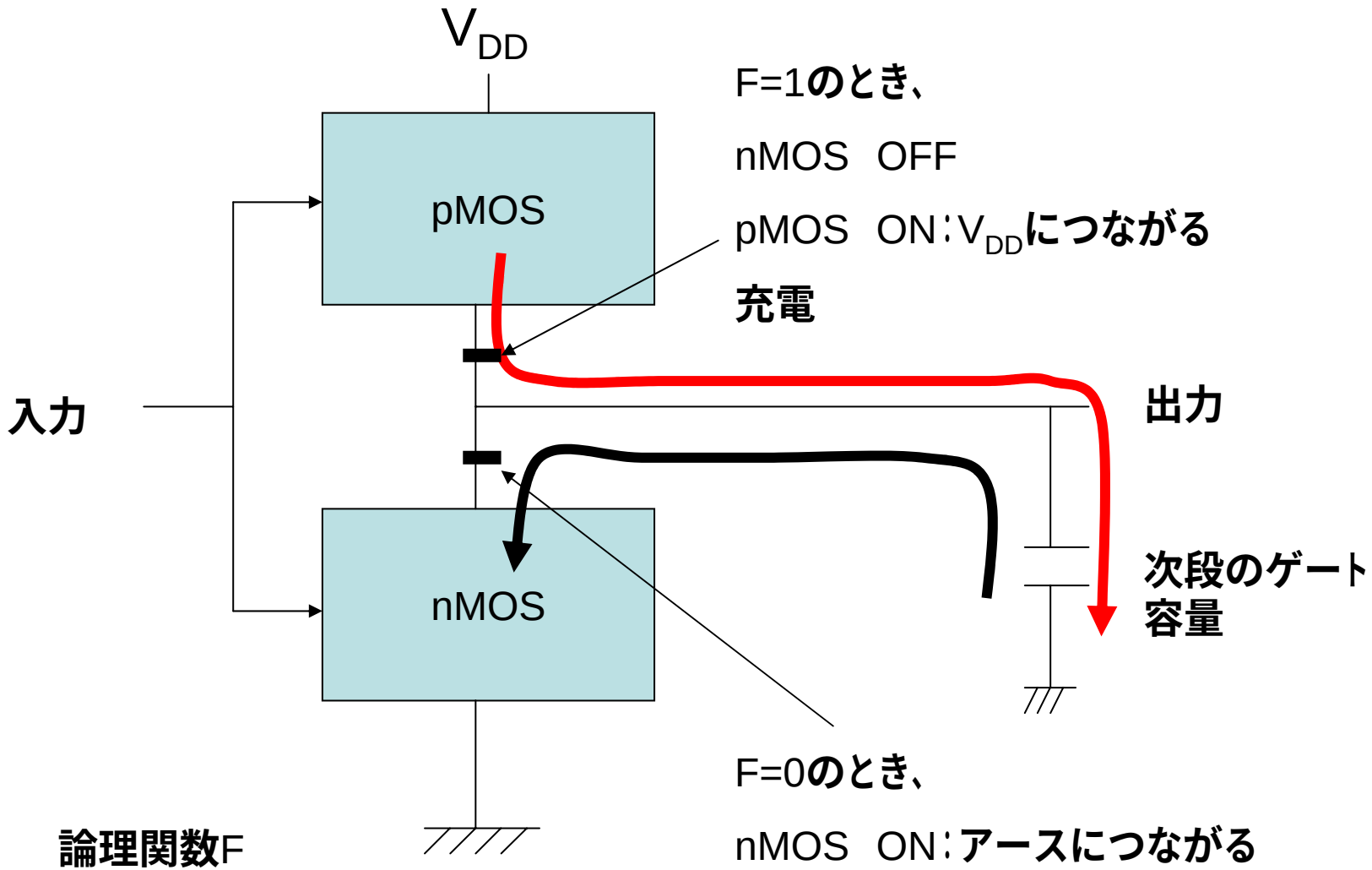
正論理で
NAND

$$\overline{\overline{(a \cdot b)}} = \overline{\bar{a} + \bar{b}}$$

$$= a + b$$

NOR




V_{DD}
pMOS
nMOS
入力
出力
次段のゲート
容量
論理関数F
F=1のとき、
nMOS　OFF
pMOS　ON：V_{DD}につながる
充電
F=0のとき、
nMOS　ON：アースにつながる
放電
pMOS　OFF

(2) 特徴

容易な設計

階層設計，機能分割

(垂直分割と水平分割)

互換性の実現

高級言語：機械命令レベルと独立

機械命令セット：RTと独立

トレードオフの実現

乗算の例

①高級言語：あり

　機械命令：あり

　ＲＴ：あり（ワレストリー）

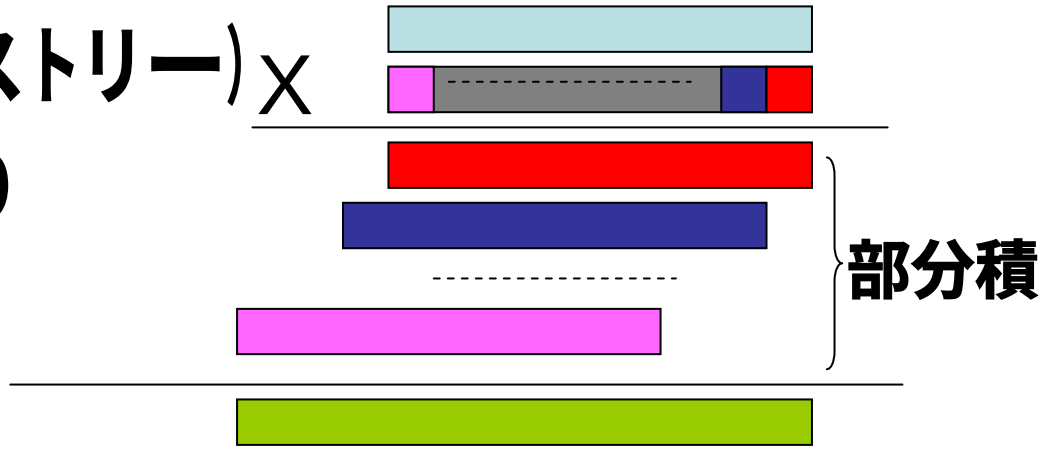
②高級言語：あり

　機械命令：あり

　ＲＴ　　：なし

　（ＲＴレベルでの反復加算：

　　加算とシフト）



③高級言語：あり
機械命令：なし
（機械命令で反復加算
：加算とシフト）
ＲＴ　：なし
仮想化技術

１．１０　ＯＳ

１ ＯＳの必要性

① 複雑な機能の提供：命令セットの拡張

大容量記憶，ファイル操作，

ネットワーク，入出力

②リソースの共有利用：効率的な管理，

セキュリティ

多数のユーザプロセスの実行

多数のサーバプロセスの実行

２ ＯＳの機能

①記憶管理

　多重仮想記憶

②ファイル管理

　ディレクトリ管理，ファイル更新･編集

　効率のよい記憶方式

③入出力管理

　デバイスドライバ

　スケジューリング

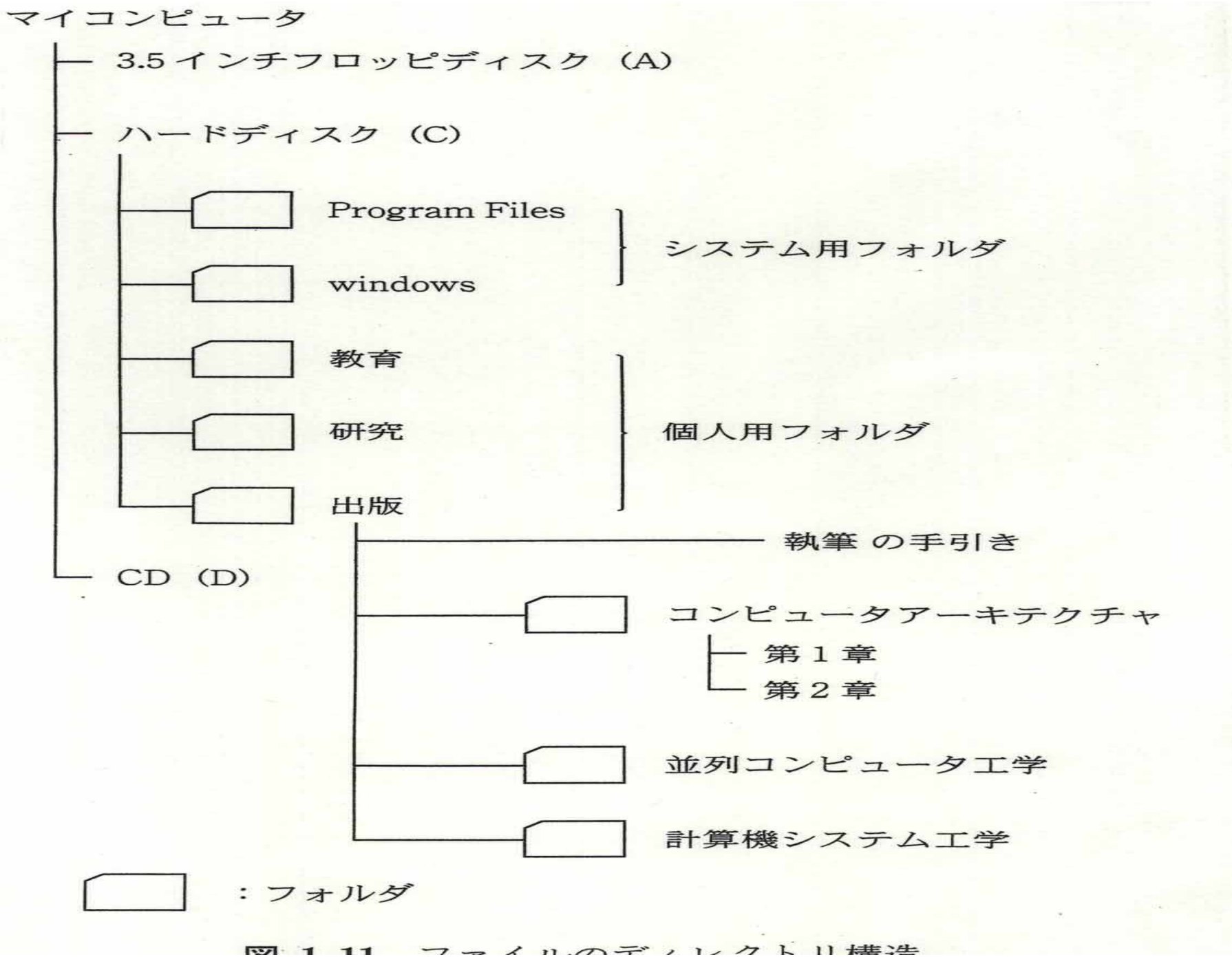


図 1.11　ファイルのディレクトリ構造

④通信管理

アプリケーション層

FTP, Telnet, WWW, SMTP, . .

TCP／UDP層

TCP：コネクション型

バーチャルチャネル

UDP：コネクションレス型

IP層

中継ノード間での通信制御

IPアドレス

インタフェース層

マルチキャスト機能

物理層

イーサネット, ATM

⑤プロセス管理
ＯＳの中のＯＳ

たくさんのプロセスの擬似的な並行実行

プロセス：ＯＳにより管理実行されているプログラム

並列プロセスの例

長時間かかるＣプログラムを実行させながら

ＷＯＲＤで文書作成し，プリントアウト

プリンタが動作し始め

ＥＸＣＥＬで表計算処理を開始していると

時計の表示

電子メールの受信通知の表示

プロセスの状態

実行中：プロセスがＯＳからプロセッサを
　　割り当てらてれ実行中の状態

レディ：プロセスは実行可能状態であるが，
　　ＯＳのプロセッサ割当てがなされて
　　いない状態

待ち：プロセスに必要なデータが揃って
　　いないので，待たねばならない状態

⑴ プロセススイッチ

契機：割込み

外部割込み：入出力，タイマ

マシンチェック

内部割込み：演算例外，

命令例外，ページ

フォルト，トレース

スーパバイザコール

多重プログラミング，

ＴＳＳ（Time Sharing System）

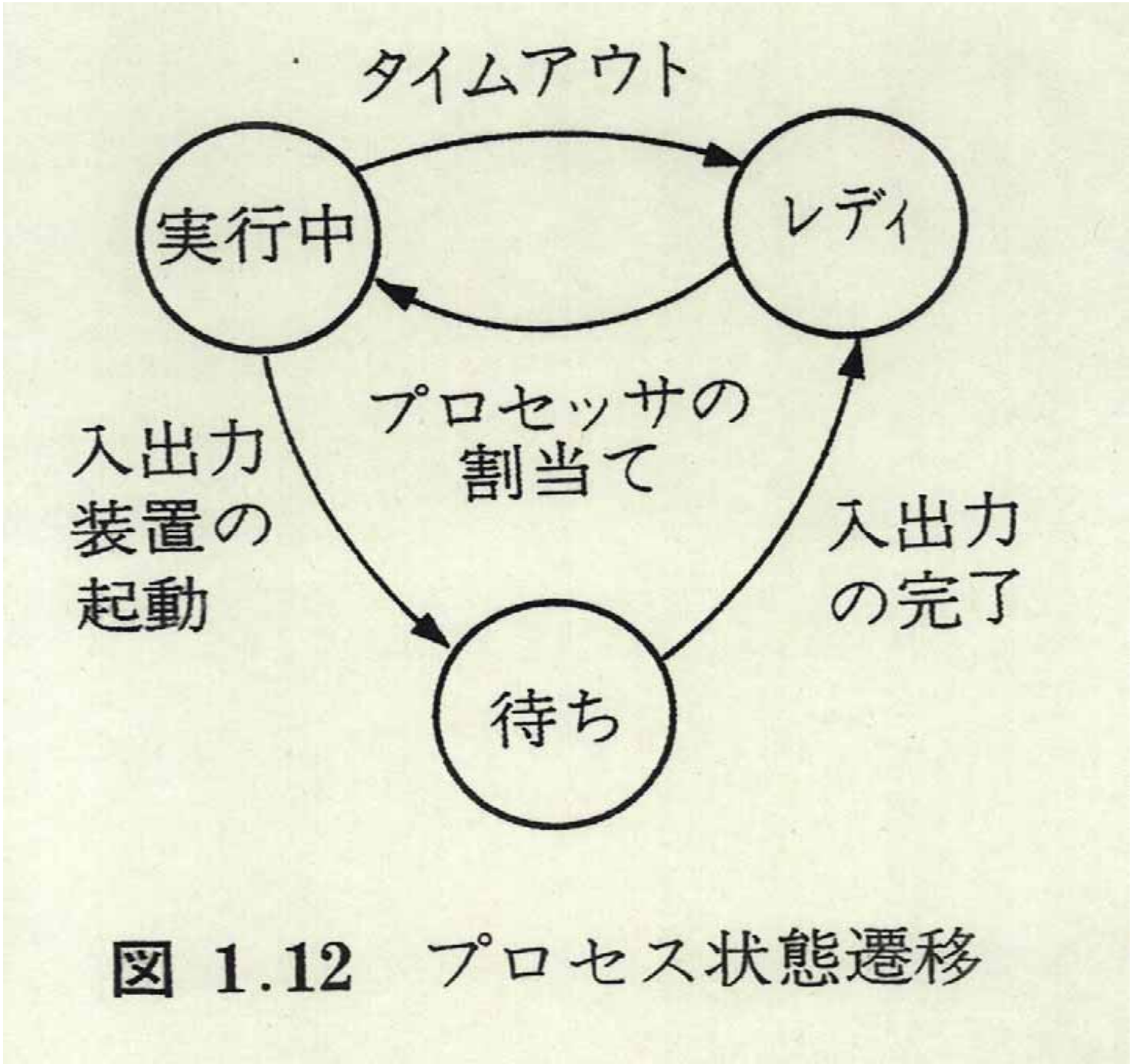
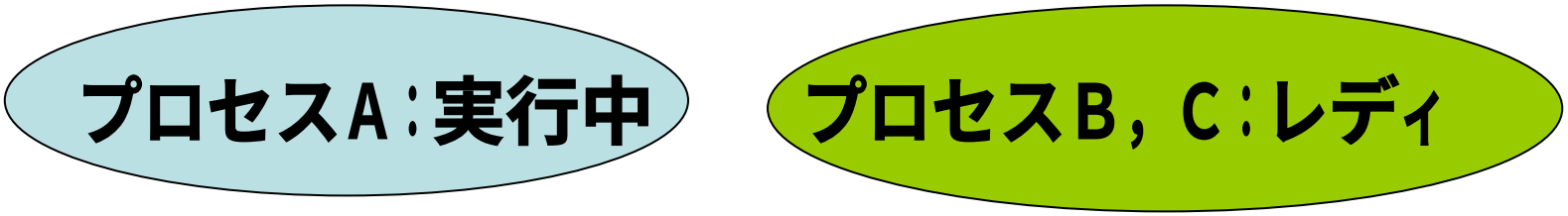
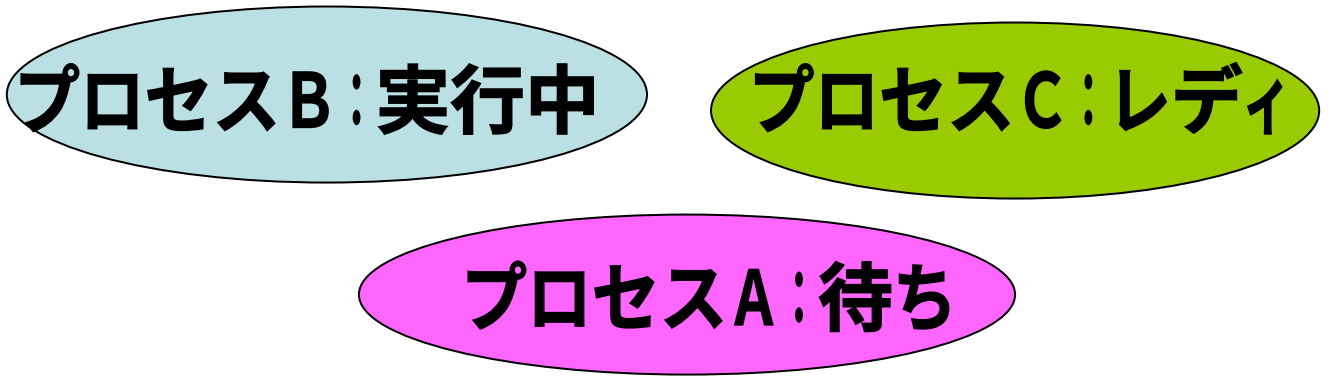


図 1.12　プロセス状態遷移

①プロセスA，B，C：レディ
　ＯＳ：プロセスAを選択し，実行させる



②プロセスA：ディスクアクセス
　Ｉ／Ｏ命令実行でＯＳに割込み
ＯＳ：プロセスAのＩ／Ｏ処理し，Aを待ちへ
　プロセスBを選択し，実行へ

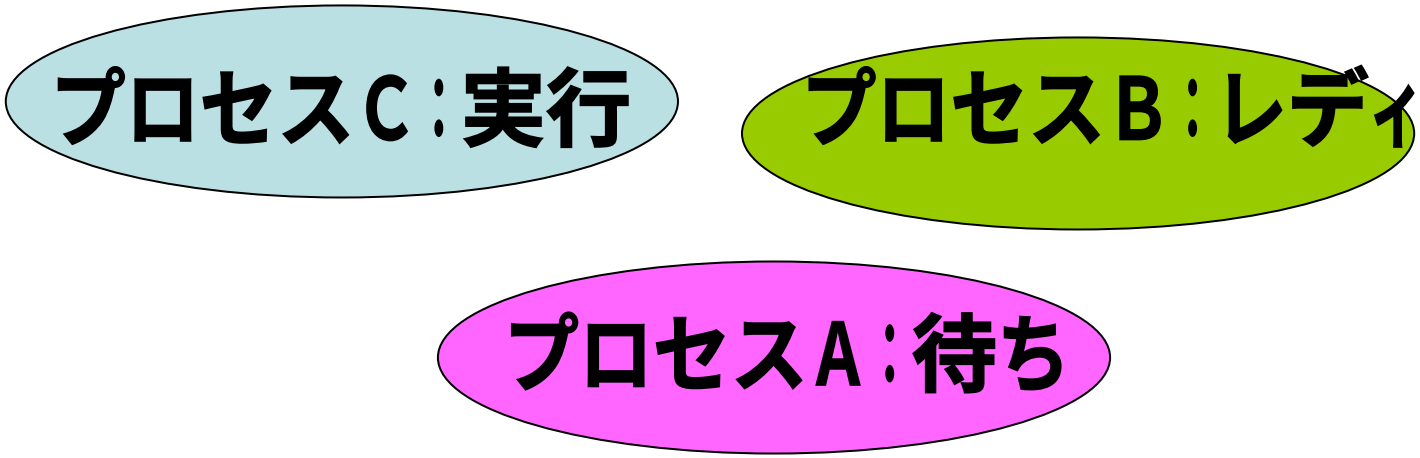


プロセスB：一定時間実行（タイムクアンタム）

１０msec

タイマー割込み

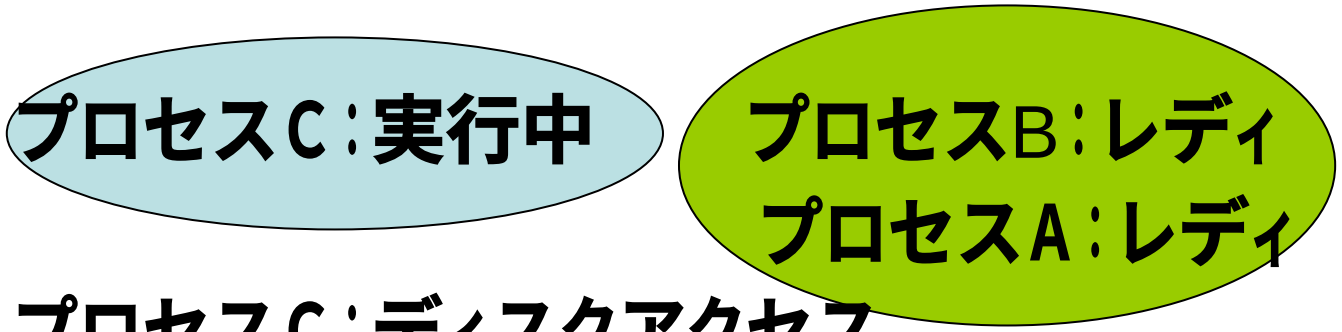
ＯＳがBをレディへ，Cを選択



ディスクからI／O割込み

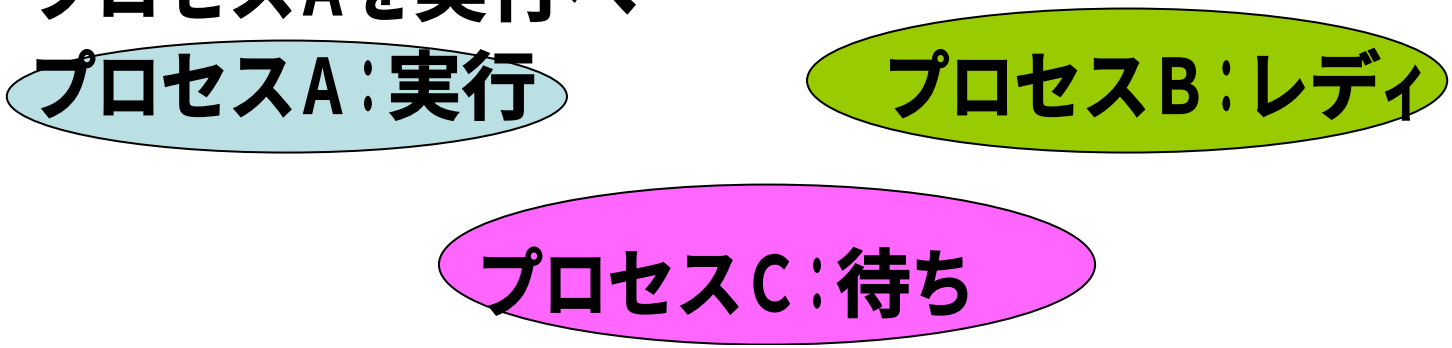
ＯＳがチェックし，プロセスAのディスク
アクセス終了を知る

ＯＳはプロセスAをレディへ



プロセスC：ディスクアクセス

ＯＳ：プロセスCを待ちへ，
プロセスAを実行へ



(2) プロセススケジューリング

(3) プロセス間通信

同期：排他制御，事象待ち

メッセージ交換

メモリ空間共有，非共有

３ ＯＳの構成

シンプルな構成の必要性

マイクロカーネル

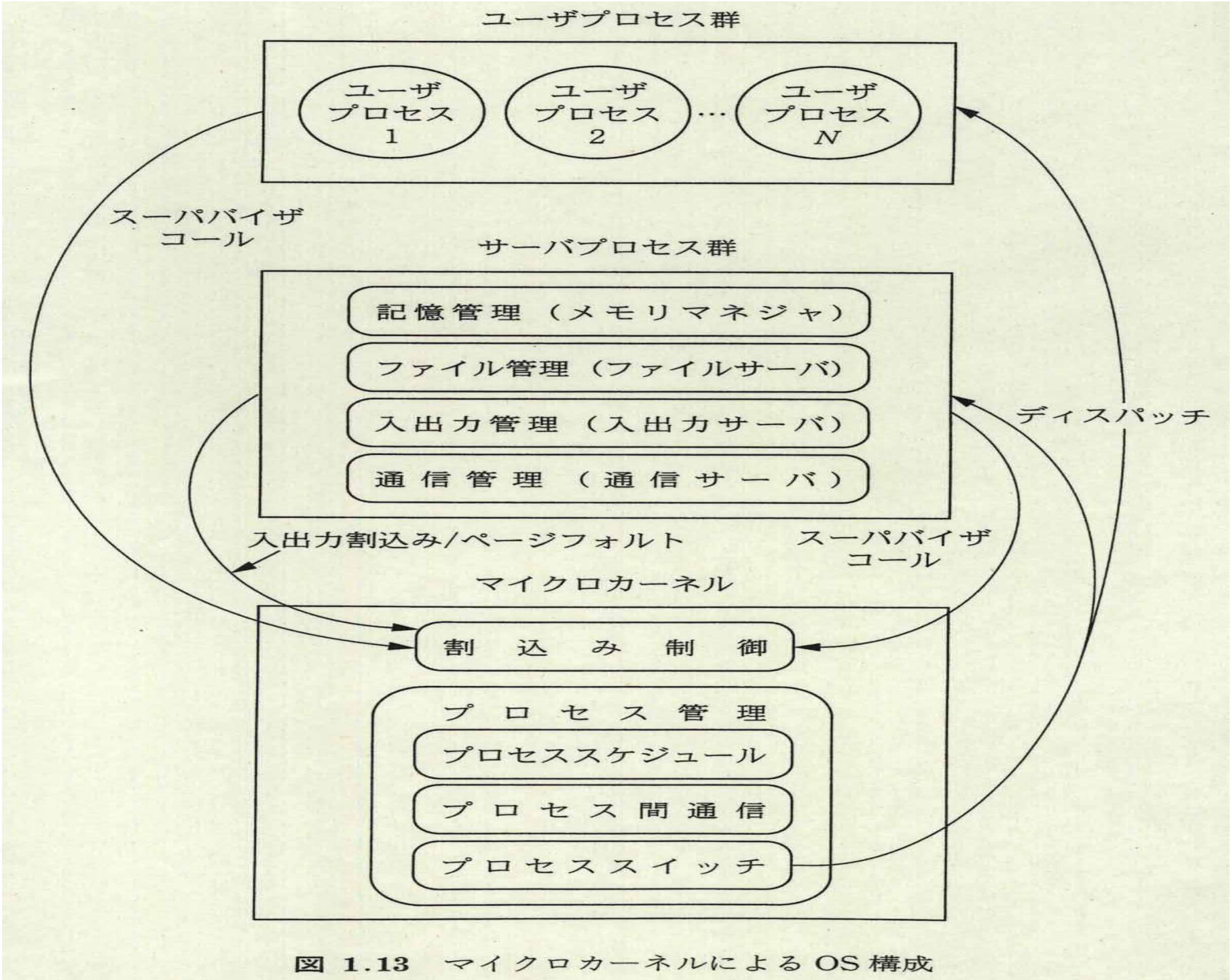


図 1.13 マイクロカーネルによるOS構成

１．１１ コンピュータの分類

１ 基本ハードウェア技術（構成部品）

①機械式：歯車など

②電気機械式：リレー

③電子式：真空管，Tr，集積回路

④光式：レンズなど

⑤DNA：?

⑥量子：？


ST管
MT管
EPROM
Super SPARC™
Microprocessor
TMS390
リレー
TR
LSI
VLSI
真空管
8ビットプロセッサ
6800
32ビットプロセッサ
SPARC

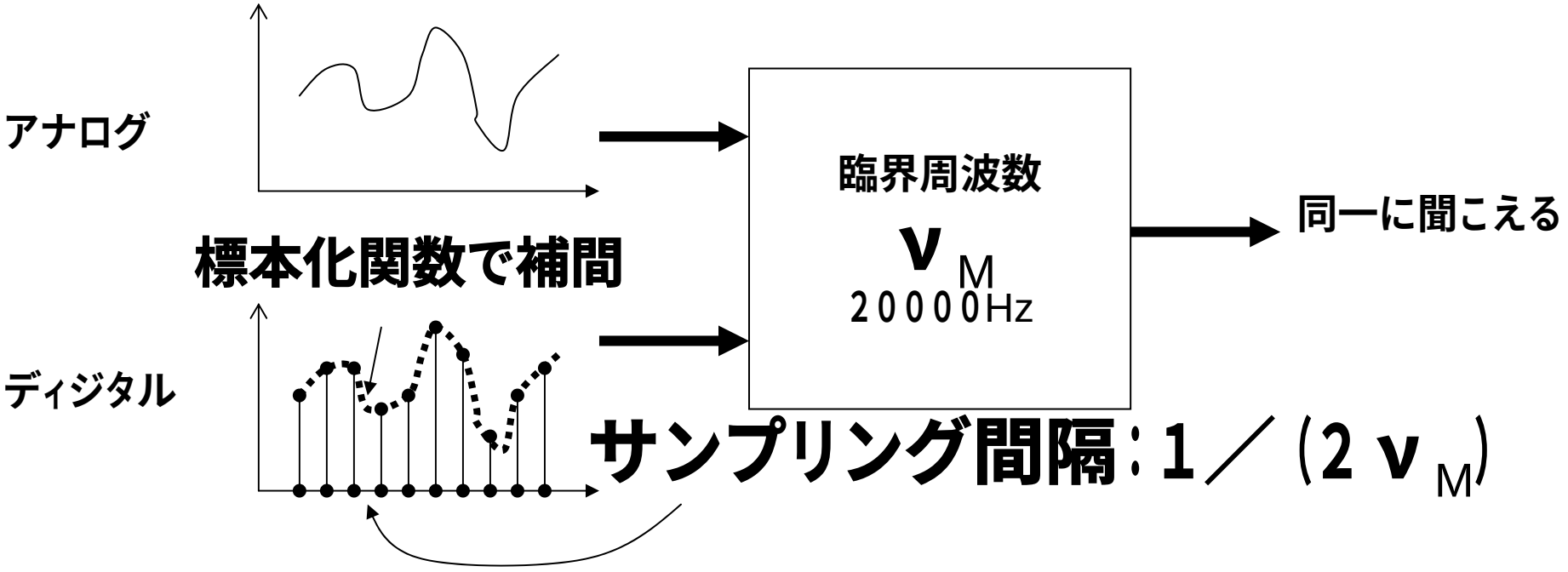
2 取り扱う量;アナログかディジタルか

①アナログ量:連続量

②ディジタル量:離散量

シャノンの標本化定理

ＡＤ変換，ＤＡ変換



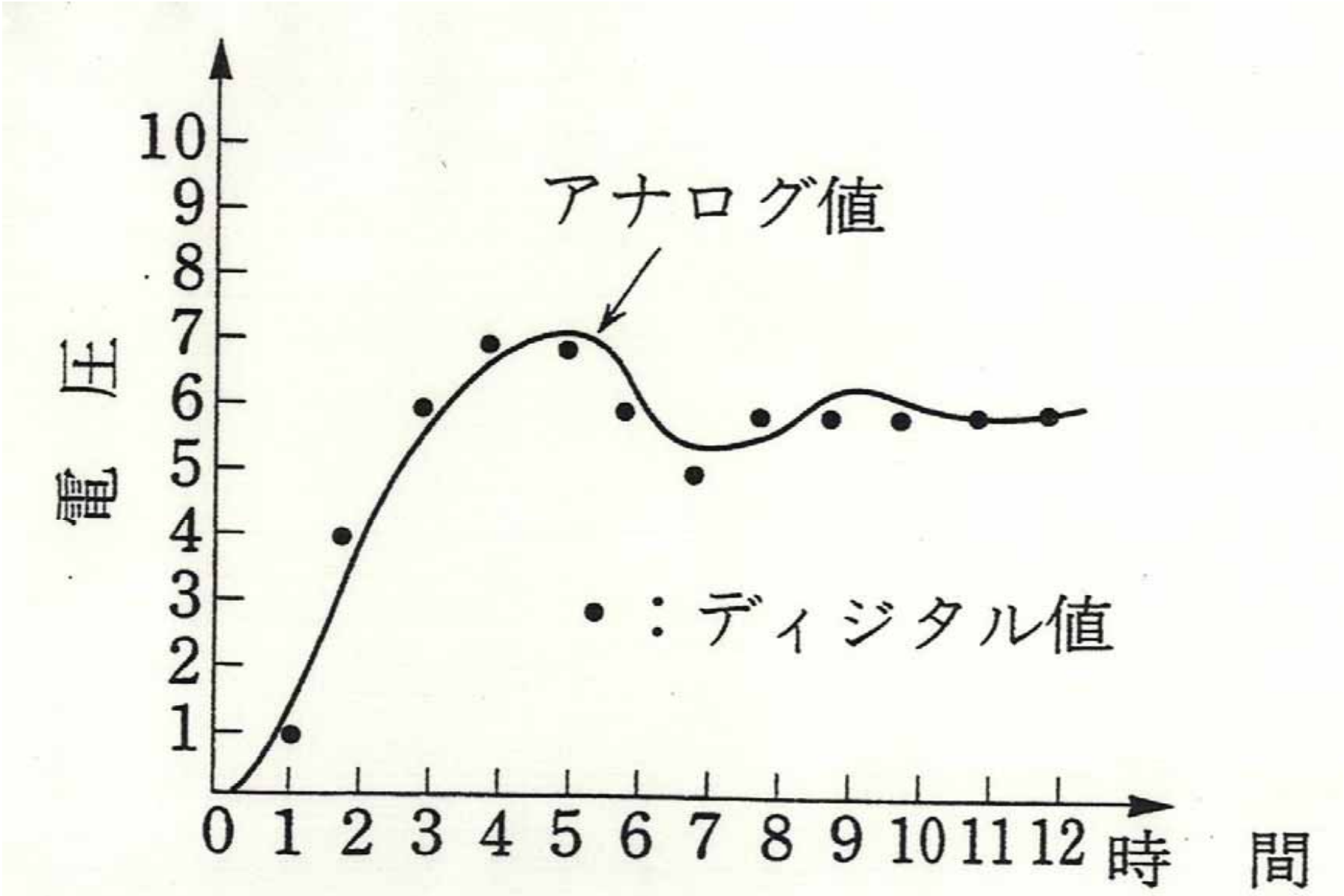


図 1.14　アナログ量とディジタル量

シャノンの標本化定理の証明

フーリエ変換

$$C(\nu) = \int_{-\infty}^{\infty} f(t)\exp(-2\pi j \nu t)dt \quad ①$$

$$f(t) = \int_{-\infty}^{\infty} C(\nu)\exp(2\pi j \nu t)d\nu \quad ②$$

パルス

$$f(t) = 1 : |t| < (1/2)\theta, 0 : |t| > (1/2)\theta \quad ③$$

$$C(\nu) = \theta \frac{\sin \pi\nu\theta}{\pi\nu\theta}$$

標本化関数

$$f(t) = 2\nu_M \frac{\sin 2\pi\nu_M t}{2\pi\nu_M t} \quad ④$$

$$C(\nu) = 1 : |\nu| < \nu_M, 0 : |\nu| > \nu_M$$

$f(t)$：臨界周波数ν_Mの周波数成分のみ

C(ν)は-ν_Mからν_Mまでの成分を有する

C(ν)のフーリエ級数は

$$C(\nu)=\Sigma_k a_k \exp\left(\pi j k \nu / \nu_M\right) \quad ⑤$$

a_kは

$$\int_{-\nu_M}^{\nu_M} C(\nu)\exp\left(-\pi j k \nu / \nu_M\right)d\nu = \quad ⑥$$

$$\int_{-\nu_M}^{\nu_M} \Sigma_{k'} a_{k'} \exp\left(\pi j \nu (k'-k)/\nu_M\right)d\nu$$

$$= 2 a_k \nu_M$$

②式より

$$f(t)=\int_{-\infty}^{\infty} C(\nu)\exp(2\pi j\nu t)d\nu$$

$t=k/(2\nu_M)$とすると⑥

より、

$$f(k/(2\nu_M))=\int_{-\nu_M}^{\nu_M} C(\nu)\exp(\pi j\nu k/\nu_M)d\nu$$

$$=2\nu_M a_{-k} \quad ⑦$$

標本点を知れば、a_{-k}が求まり

$$C(\nu)=\Sigma_k a_k \exp(\pi jk\nu/\nu_M)$$

が定まる。

②,⑤式より

$$f(t) = \Sigma a_k \int_{-\nu_M}^{\nu_M} exp(\pi j k \nu / \nu_M) exp(2\pi j \nu t) d\nu$$

$$= \Sigma a_{-k} \int_{-\nu_M}^{\nu_M} exp(-\pi j k \nu / \nu_M) exp(2\pi j \nu t) d\nu$$

$$= \frac{1}{2\nu_M} \Sigma f(k/(2\nu_M)) \int_{-\nu_M}^{\nu_M} exp(\pi(2\nu_M t - k) j \nu / \nu_M) d\nu$$

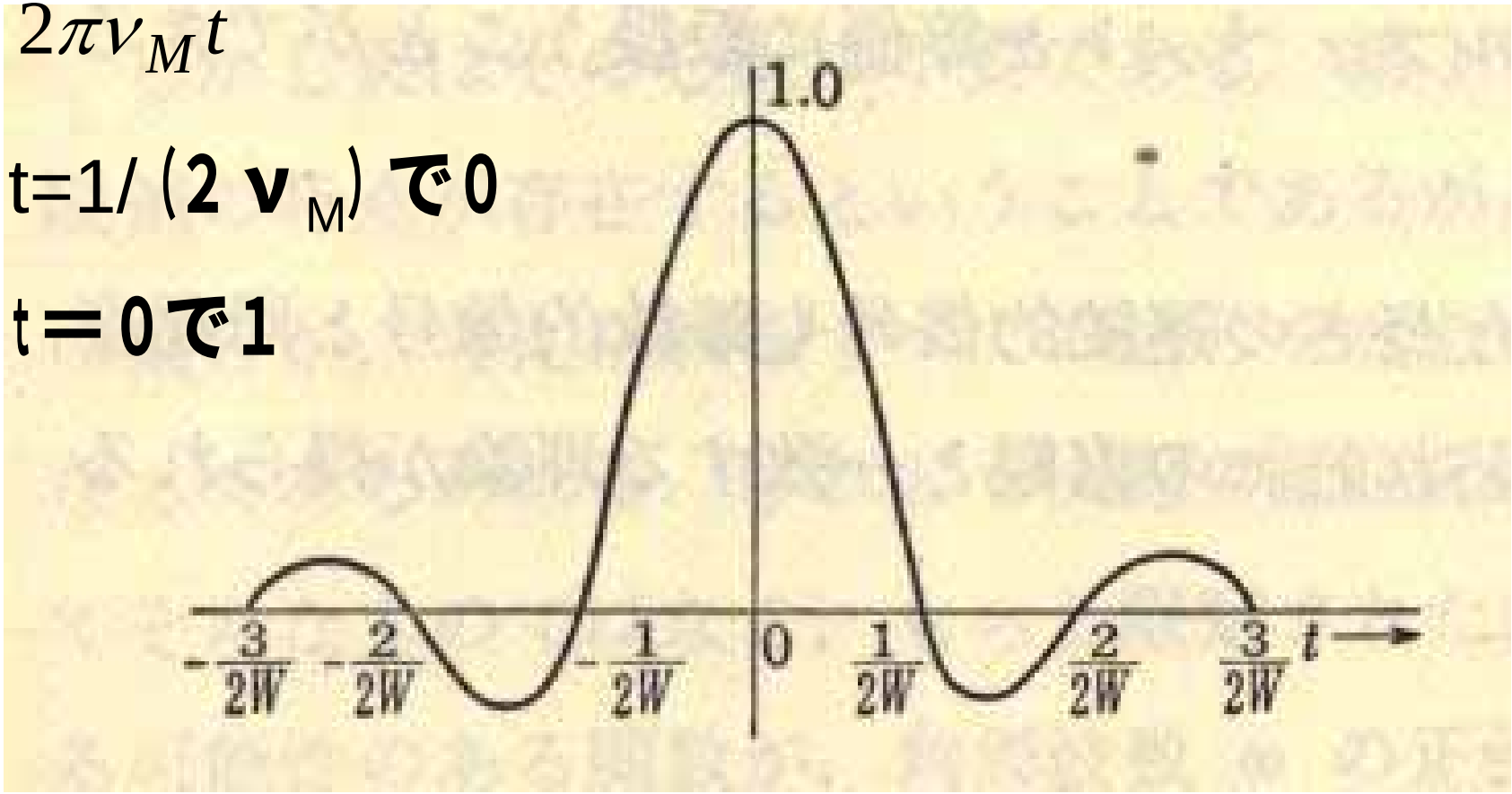
$$= \Sigma f(k/(2\nu_M)) \frac{\sin \pi(2\nu_M t - k)}{\pi(2\nu_M t - k)} \qquad ⑦$$

標本化関数

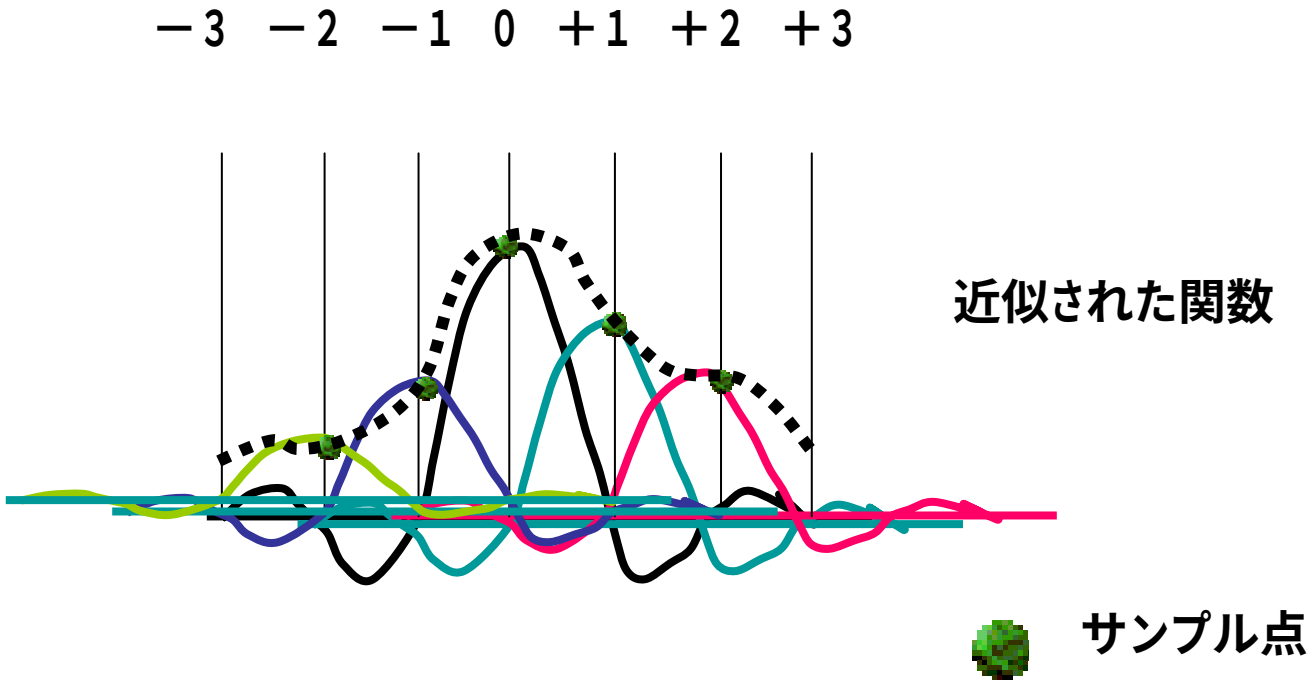
$$\frac{\sin 2\pi\nu_M t}{2\pi\nu_M t}$$

t=1/（2 ν_M）で0

t＝0で1



W＝ ν_M


−3 −2 −1 0 +1 +2 +3
近似された関数
サンプル点

アナログとディジタルは違わない？

- メモリとの親和性
- 精度
- 信頼性
- 汎用性

③アナログコンピュータ

問題解決――＞物理現象･量に写像

例：計算尺

プログラミング

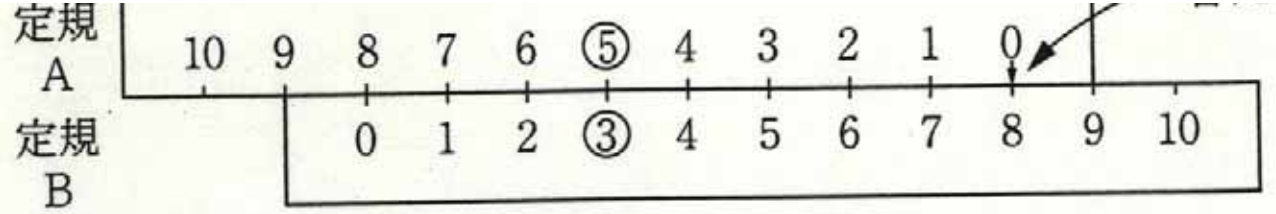
配線

④ディジタルコンピュータ

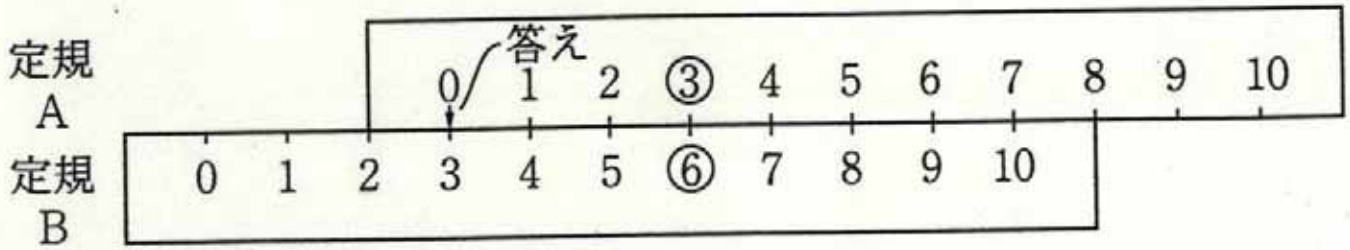
問題解決――＞アルゴリズムによる

記号操作

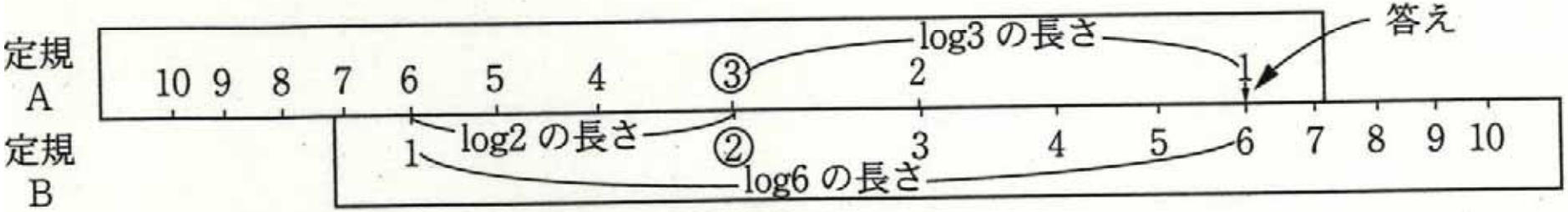
例：そろばん



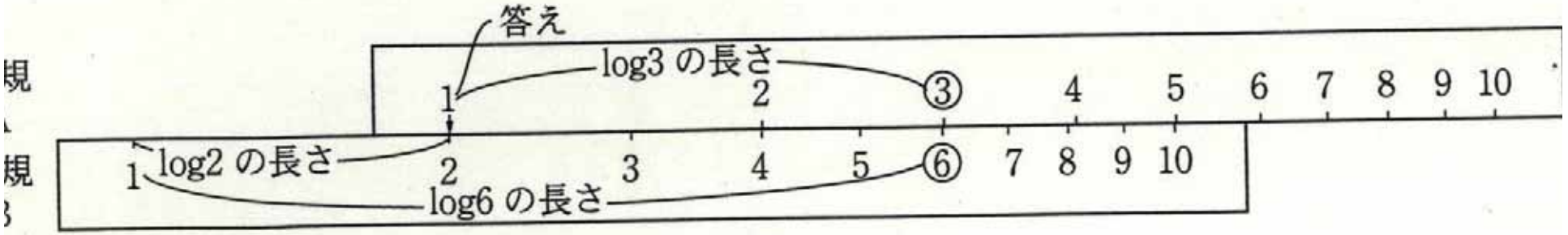
(a)　加算 (3+5)



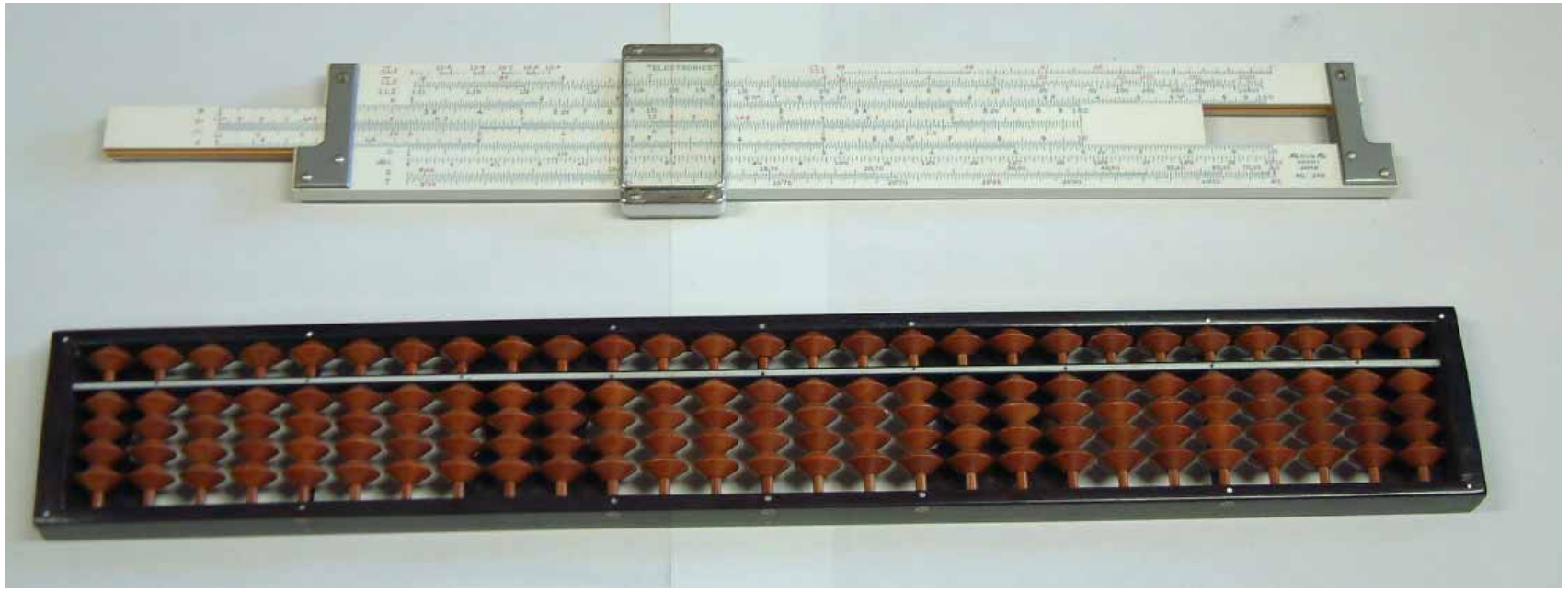
(b)　減算 (6−3)

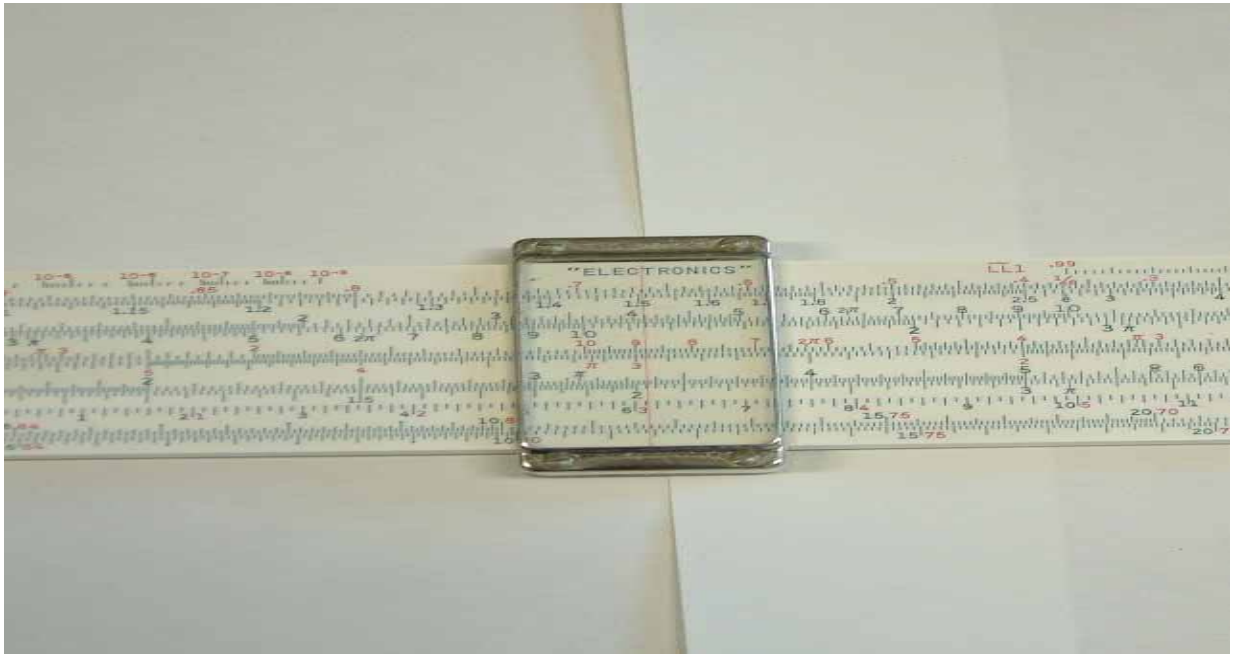


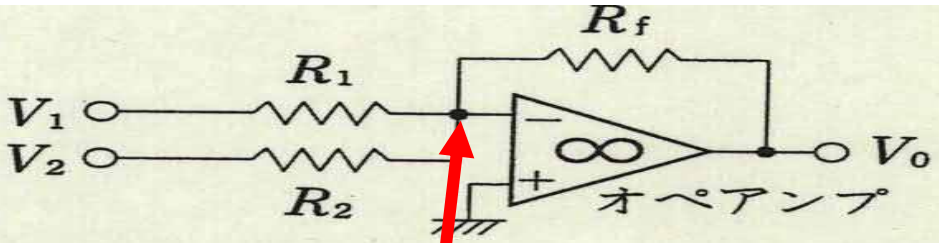
(c)　乗算 (2×3)



(d)　除算 (6÷3)

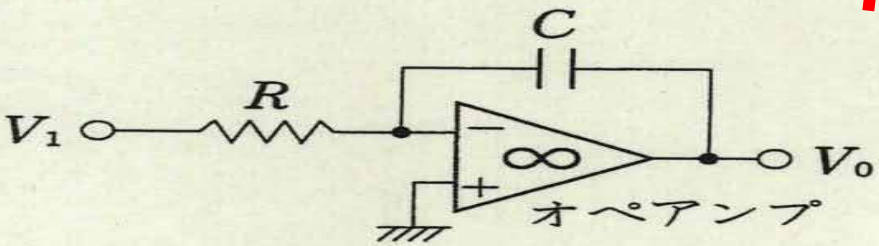



"ELECTRONICS"



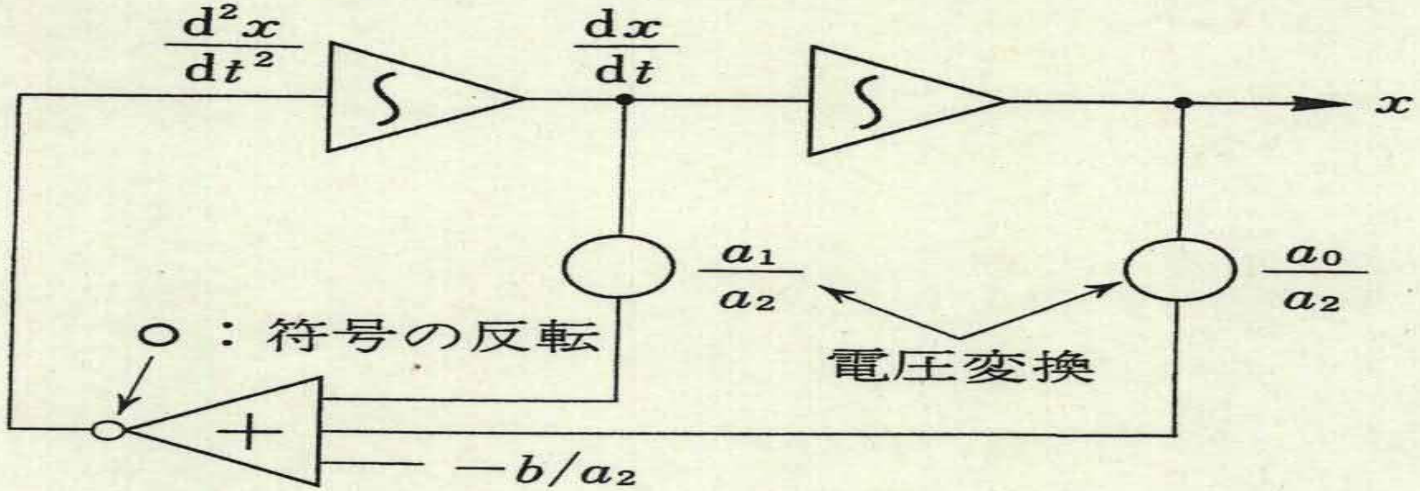
(a) 加算器 $V_0 = -R_f\left(\frac{V_1}{R_1} + \frac{V_2}{R_2}\right)$

仮想接地：0 V



(b) 積分器 $V_0 = -\frac{Q}{C} = -\frac{1}{CR}\int V_1\,dt$

$$a_2\frac{d^2x}{dt^2} + a_1\frac{dx}{dt} + a_0 x = b$$



(c) 常微分方程式を解く回路

3 制御方式

① 集中制御･逐次制御

ノイマン式

② 分散制御･並列制御

非ノイマン式

- データフロー
- ニューラルネット
- 光コンピュータ
- DNAコンピュータ
- 量子コンピュータ