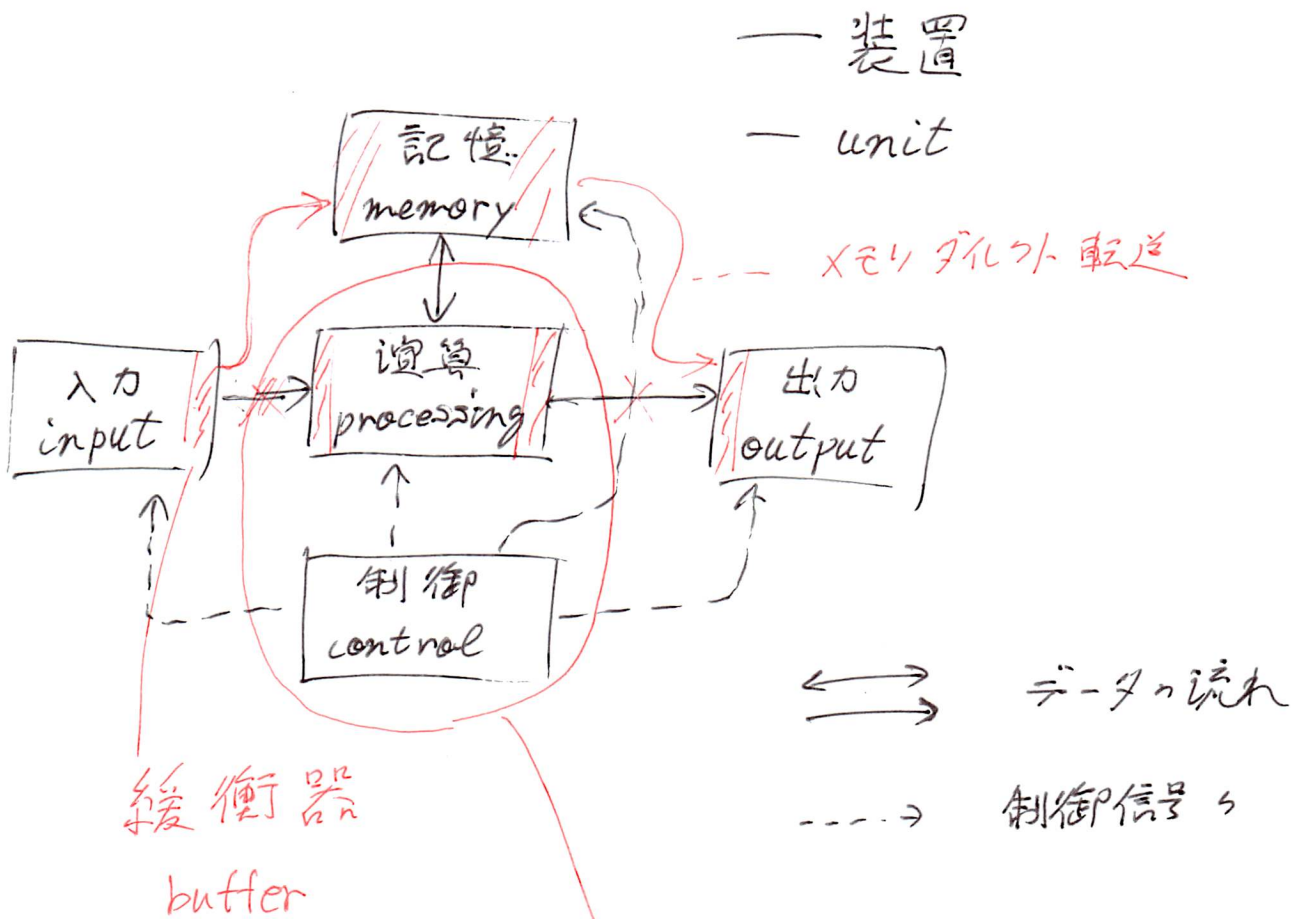
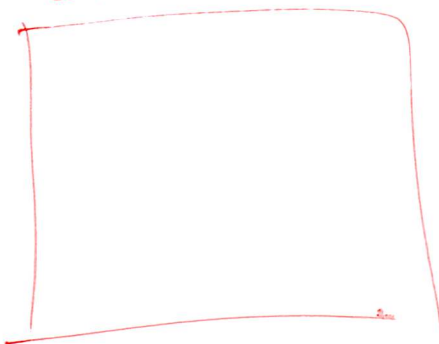


計算機の基本構成 (ブロック図)



中央処理装置 CPU
Central Processing Unit

全体を 1 個の IC に



⇒ マイクロ・プロセッサ

組み込み機器

(例)

PIC

AVR

1 個の IC に

マイクロ・プロセッサ

MPU

JAVA

言語

address
番地 アドレス

メモリ memory

大容量, 低速



情報(量)の単位

○ bit binary digit
ビット

○ 1 byte = 8 bit

1B バイト

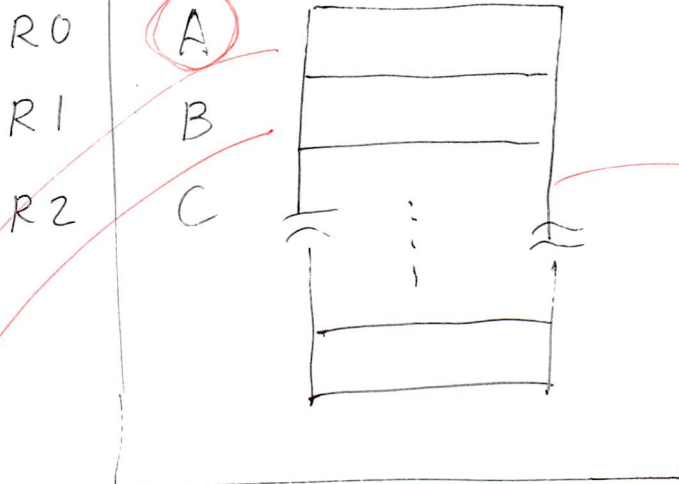
(例) ハードディスクの容量

2TB

↑
補助単位 10^{12}

演算・制御 CPU
置数器

レジスタ register



小容量, 高速

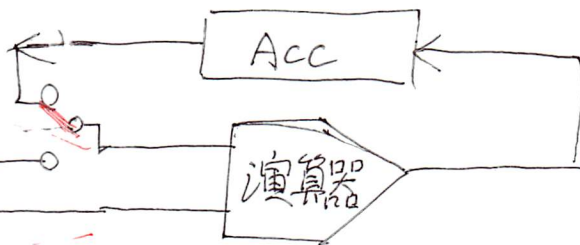
累算器

Accumulator

アキュムレータ

③ 抵抗器

レジスタ



n 進法

$n = 2$ --- 計算機

10 --- 人

8
16) --- 計算機

$n = 3$ --- 問. + 進表記へ変換せよ

$$(121)_3 = ?$$

人類の使用している n 進数
(いくつ)

12 進法 通貨

(24)

60 進

60 分 = 1 時間

60 秒 = 1 分

8 進

部屋番号 など

0 1 2 3 5 6 7 8

$\left(\begin{array}{cc} 4 & \times \\ 9 & \times \end{array} \right.$ 死 苦

10進数	2進数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

10進数 $\Rightarrow$ 十進数

（'0' ~ '9' の 10 種類の
符号（記号）を用いて表記した数）

⑤ 十進表記

2進数（あるいは 2進表記）

（'0' と '1' との 2 種類の符号と
用いて表記（した数））

⑥ 位取り記号 $\Rightarrow$ 「0 の発見」

符号の位置による

値の重みが異なる
加重

4桁表記で統一。

（例）十進数の場合

（例）2進数の場合

1010

2³ 2² 2¹ 2⁰
8 4 2 1

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 8 + 0 + 2 + 0$$

$$= 10$$

$$4 \times 10^2 + 1 \times 10^1 + 8 \times 10^0$$

$$= 400 + 10 + 8$$

$$= 418$$

418

100 -- 10 -- 1 -- 加重
10² 10¹ 10⁰

（参考）漢数字による表記

10 --- 十

11 --- 十一

12 --- 十二

100 --- 百

418 --- 四百十八

四コ、百

$$\sum (\text{表記した数}) \times (\text{基数})^{\text{(指数)}}$$

（問）ローマ数字で 418 を表わせ
（ハ）他の表記もいみよ

(問) つぎの 2進表記 を十進数に + 進表記で表わせ

$$\textcircled{1} \quad (101011)_2 =$$

$$\textcircled{2} \quad (10010)_2 =$$

$$\textcircled{3} \quad (1111111)_2 =$$

(解) ① 指 5 4 3 2 1 0

$$\begin{aligned} 101011 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= \underbrace{32}_{\text{重 } \textcircled{32}} + \underbrace{8}_{\text{重 } \textcircled{8}} + \underbrace{2}_{\text{重 } \textcircled{2}} + 1 \\ &= 43 \end{aligned}$$

$$\begin{aligned} \textcircled{2} \quad 10010 &= 16 + 2 \\ &= \underbrace{16}_{\text{重 } \textcircled{16}} + \underbrace{2}_{\text{重 } \textcircled{2}} = 18 \end{aligned}$$

$$\begin{aligned} \textcircled{3} \quad (1111111)_2 &= (10000000)_2 - 1 \\ &= \underbrace{128}_{\text{重 } \textcircled{128}} - 1 \\ &= 127 \end{aligned}$$

① 2進 + 進変換

② + 進 2進変換

$$\begin{aligned} 418 &= 256 (= 2^8) + 162 \\ &= 2^8 + 128 (= 2^7) + 34 \\ &= 2^8 + 2^7 + 32 (= 2^5) + 2 (= 2^1) \\ &= 2^8 + 2^7 \quad \textcircled{0} + 2^5 \quad \textcircled{000} + 2^1 \quad \textcircled{0} \\ &= (110100010)_2 \end{aligned}$$

商が0で終了

	0	---	1
2	$\overline{) 1}$	---	1
2	$\overline{) 3}$	---	0
2	$\overline{) 6}$	---	1
2	$\overline{) 13}$	---	0
2	$\overline{) 26}$	---	0
2	$\overline{) 52}$	---	0
2	$\overline{) 104}$	---	1
2	$\overline{) 209}$	---	0
2	$\overline{) 418}$		

110100010

418

209	---	0
104	---	1
52	---	0
26	---	0
13	---	0
6	---	1
3	---	0
1	---	1
0	---	1

(問) 2進数で表記せよ。

$$\textcircled{1} \quad 100 = (0110\ 0100)_2 = (64)_{16}$$

$$\begin{aligned} \textcircled{2} \quad 2014 &= 2048 - 34 \\ &= 2^{11} - (100010)_2 \quad \left(\begin{array}{l} 34 = 32 + 2 \\ = 2^5 + 2^1 \end{array} \right) \\ &= (1000\ 0000\ 0000)_2 \\ &\quad - (0\ 0000\ 010\ 0010)_2 \\ &= (0111\ 1101\ 1110)_2 \\ &= (7DE)_{16} \end{aligned}$$

2014.04.23 補足

○ 2進数の加減算

○ 16進表記 (目的: 2進表記との変換が容易で、かつ桁数が少ない表記)

⑧ 8進表記

2進1桁の呼称

$$\textcircled{\text{答}} \quad 2^{10} = 1024 \div 1000$$

1K 1k

bit ビット

負数の表現

～～～
その前に、

計算機内部で表現できる情報(量)は

有限である。 ← 最大(容量)を気にして扱う。

～～～
8bitで表わせる 最大値は？

⇒ 8bitで 何通りの情報を表わせるか？

の情報量

(答) $2^8 = 256$ 通り

256 通り	0000	0000	00H	---	△	何に割合てる かは自由 ↓ たが 標準化 standard が必要
	0000	0001	01H	--	□	
	0000	0010	02H	-	☆	
	1111	1110	0FEH	---	Σ	
	1111	1111	0FFH	-	□	

Hexadecimal 16進の

256通り	128通り	0FFH	---	255	127	---	111 1111
		0FEH	--	254	126	---	111 1110
		...		...	...		
		80H	---	128	0		
		7FH	--	127	-1		
	128通り	02H	---	2	-2		
		01H	--	1	-127		
		00H	--	0	-128		

正(1) 大小関係を保つ
 負(0)
 正の整数の割当て 正負の整数の割当て
 符号bit (最上位bit) を反転

計算機では

先頭の0/1で正負の符号を表わすものとした。

0 — 正
 1 — 負

(負)	0FFH	-1
	0FEH	-2
	...	...
	080H	-128
	07FH	127
(正)	02H	2
	01H	1
	00H	0

素直 =>

並べ替

127	07FH	0111 1111
...	...	...
1	01H	0000 0001
0	00H	0000 0000
-1	0FFH	1111 1111
-2	0FEH	1111 1110
...	...	...
-128	80H	1000 0000

↓

補数表現

1 補数 9 補数
2 補数 10 "

補数とは

補う (足す) ことで、ちょうどキリの良い数になる数

(例)

$$7 \text{ の 補数 } = 3$$

$$7 + 3 = 10$$

キリの良い数

負数を符号と補数で表す

$$-7 \implies -, 3$$

符号bit

$$7 = (\boxed{0000} \ 0111)_2$$

「2 補数」

$$+) -7 = (\boxed{1111} \ 1001)_2$$

$$\hline (1 \ 0000 \ 0000)_2 \quad \leftarrow \text{キリの良い数}$$

もうひとつの例

「1 補数」

$$7 = (0000 \ 0111)_2$$

$$+) -7 = (1111 \ 1000)_2$$

bit 反転 (0↔1)

$$\hline (1111 \ 1111)_2 \quad \leftarrow \text{キリの良い数}$$

② 61 の ¹⁰⁰⁰100 補数 は ⁹³⁹39 である。 (代表)

^(a)100 に対する補数

10 補数

↓

2進では

2 補数

$$\begin{array}{r} 100 \\ - 61 \\ \hline 39 \end{array}$$

② 61 の ⁹⁹⁹99 補数 は ⁹³⁸38 である。 (代表) 9 補数

99 に対する補数

↓

2進では

1 補数

$$\begin{array}{r} 99 \\ - 61 \\ \hline 38 \end{array}$$

(一般化して)

y の a に対する補数を x とすると,

$$x = a - y$$

あるいは

$$x + y = a$$

補数 x --- 補うことで 「キリの良い数」 a になる数
 y に

2補数と1補数のどちらを使えばいい?

(注意: その前に、種々の負数表現の長所・短所を理解しているか?)

○ 大小関係などを扱うときは ... オフセット表現

② (0と) 正数のみを扱う場合と同じであることが大切
できるわけ $\Rightarrow$ (同じハードウェアが利用できる)

演算のしやすさ

(問) 2つの補数表現において、正数・負数の
加算をしてみて、正数のみの場合と比べてみよ。

(例)

$$a + b$$

(さ312)

$$|a| = 3, |b| = 5$$

5bit

$$\begin{cases} 3 + 5 \\ (-3) + 5 \\ (-3) + (-5) \\ 3 + (-5) \end{cases}$$

$$3 = \overset{1}{0}\overset{1}{0}\overset{1}{0} \overset{1}{0}\overset{1}{0} \\ (000\ 11)_2$$

$$5 = (00\ 101)_2$$

1補数を用いて

~~implement~~
インクリメント

$$-3 = (111\ 00)_2$$

$$-5 = (110\ 10)_2$$

2補数

increment
インクリメント

$$-3 = (111\ 01)_2$$

$$-5 = (110\ 11)_2$$

[5/2 修正]

complement

コンプリメント

(補う, 補数)

(bit毎)
反転

+1

$$3 + 5 = 8$$

$$-3 + 5 = 2$$

$$\begin{array}{r} 00011 \\ +) 00101 \\ \hline (01000)_2 = 8 \end{array}$$

$$\begin{array}{r} \textcircled{1} \quad 11100 \\ +) 00101 \\ \hline 1(00001)_2 \neq 2 \end{array}$$

← 2進33

補正(回路)が必要

$$\begin{array}{r} \textcircled{2} \quad 11101 \\ +) 00101 \\ \hline 1(00010)_2 = 2 \end{array}$$

つまり、
正数用回路が
使える

$$3 + (-5) = -2$$

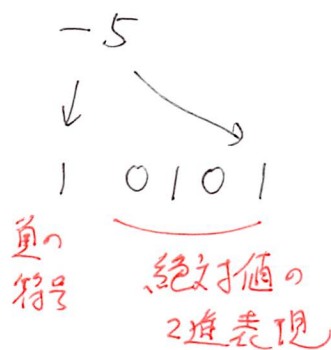
$$\begin{array}{r} \textcircled{1} \quad 00011 \\ +) 11010 \\ \hline (11101)_2 = (\text{反転 } (00010)_2 = 2) \\ = -2 \end{array}$$

$$\begin{array}{r} \textcircled{2} \quad 00011 \\ +) 11011 \\ \hline (11110)_2 = -2 \end{array}$$

反転 反転 ↓
00001
+1 ↓
(00010)₂ = 2

(補足) 別、負数表現

(例) 符号と絶対値



文字符号 (文字コード)

character code

- 文字も 2進数で表現される,
- 表現方法も 数種類ある。

(例) ◦ ASCII
EBCDIC

文字 'A' = $(41)_{16}$
= $(0100\ 0001)_2$

- JIS
- Shift-JIS
- EUC
- Uni-Code

(課題) 各自の氏名を 各種の文字コードで表わせ。
(16進表記せよ。)

制御(プログラム)も 2進表記する。

→ 機械語

(例) モデル計算機

論理演算

論理積

x : 「(ある人が) 男である」

y : 「, 大人, 」

z : 「, 大人, 男である」

大人であり、かつ
男である

x, y, z の値 (真理値) を一覧表にしてみる。

(1 : 真
0 : 偽)

真理値表

x	y	z	
0	0	0	女の子
0	1	0	女の大人
1	0	0	男の子
1	1	1	男の大人

「 z は、 x と y との論理積である。」

AND

(書き方) [記法]

$$z = x \text{ and } y$$

または、

$$z = x \wedge y$$

$$z = x \cdot y$$

$$(z = xy)$$

論理記号の一種

(演算記号)

[参考] 数の積と比較

→ (形式的には) 同一である。

$$0 \cdot x \longrightarrow 0$$

論理和

x, y は 同じ (同左)

w : 「~~夫~~女 大人であるか、
または、男である」

x	y	w
0	0	0
0	1	1
1	0	1
1	1	1

 ←

「 w は、 x と y との 論理和 である。」
OR

$$w = x \text{ or } y$$

$$w = x \vee y$$

$$w = x + y$$

[参考] 数の和と比較

→ 一部が異なる

$$1 + 1 = \underline{1} \iff 1 + 1 = \underline{2}$$

論理否定

x : 「男である」

y : 「男でない (女である)」

x	y
0	1
1	0

「 y は x の論理否定である」

NOT

$$y = \text{not } x$$

$$y = \bar{x}$$

○ 論理積, 論理和, 論理否定などを総称して,
論理演算という。

○ ^(真値) 論理値, 論理変数, 論理演算を用いて,
表現した式を論理式という。

組み合わせるもの

$$x + y \cdot z$$

$$x + 1$$

$$z = x \cdot y$$

(問) 空欄を埋めよ。

$$(\text{積の否定}) = (\text{否定の和})$$

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

ド・モルガンの法則

$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

(和の否定)

(否定の積)

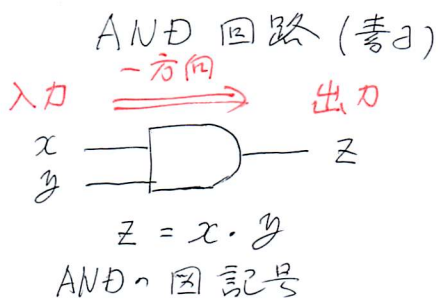
x	y	$\bar{x}$	$\bar{y}$	$x + y$	$\overline{x + y}$	$\bar{x} \cdot \bar{y}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

(ハードウェア基礎)

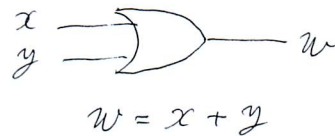
論理回路

(回路)

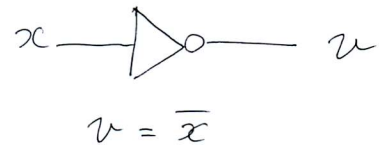
○ 基本的な回路素子 $\Rightarrow$ 論理素子



OR回路 (・)



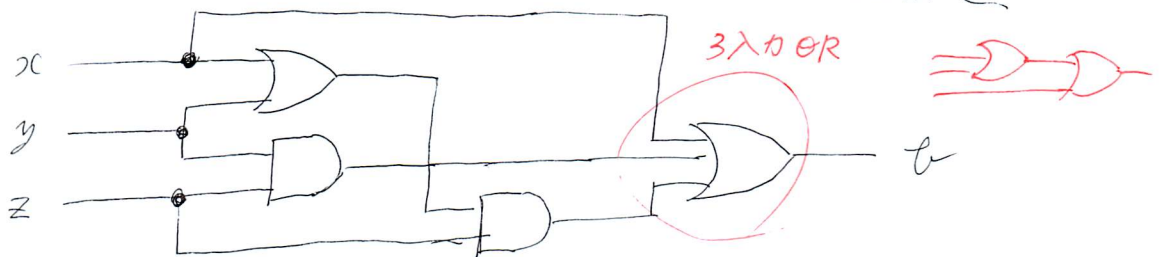
NOT回路 (')



(問) $a = \overline{x + y}$ を回路図で表わせ。



(問) $b = x + \overline{z} \cdot y + (x + y) \cdot \overline{z}$ を回路図で表わせ



(注1) 交差点と分岐点の区別

(注2) 多入力素子

④ 簡単化

(例)

$$x + x \longrightarrow x$$

$$x \cdot x \longrightarrow x$$

$$1 + x \longrightarrow 1$$

$$0 \cdot x \longrightarrow 0$$

例12

係数 a, b, c を与えて,
2次方程式の解を求める(プログラムを作れ)。

$$ax^2 + bx + c = 0$$

(例)

$$\begin{cases} a = 2 \\ b = 5 \\ c = 3 \end{cases}$$

$$2x^2 + 5x + 3 = 0$$

$$\therefore (x+1)(2x+3) = 0$$

$$\therefore x = -1, -\frac{3}{2}$$

のとき,

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(解の公式)

=

~~~~~  
プログラムでは

double a = 2 ;

double b = 5 ;

double c = 3 ;

double x ;

System.out.println(x);

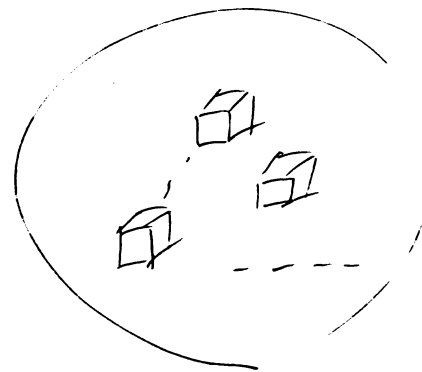
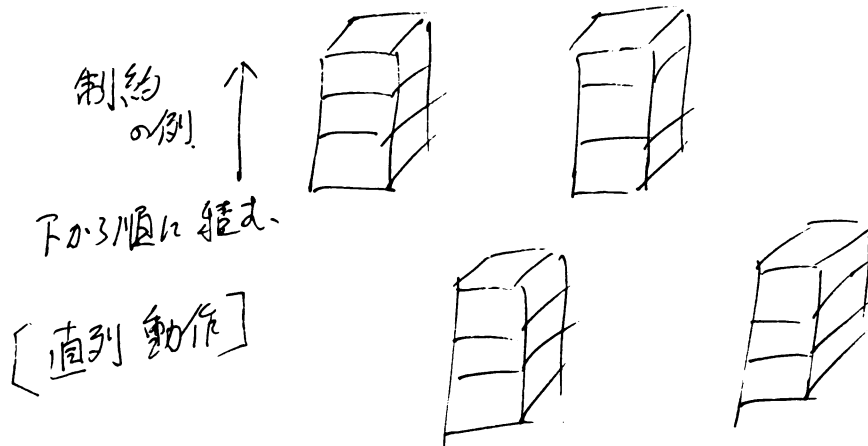
$x = \dots ;$



- 制御 ----- 繰返し文 (for文, while文, ...)
- データ ----- 配列 array

(例) 箱を積む仕事

[並列動作]  
(並行)



メモリ ----- ランダムアクセス ↔ シーケンシャルアクセス

(読書き)

(例) テーポ 入出力装置

主記憶装置

補助記憶装置

ハードメモリ、ユニット

PFL2

00H: INDF

04H: FSR

30H

30H

78H

MOV F INDF, W

W 78H

## 間接アドレッシング・モード

(例)

メモリの向き



LD A, (12H)

A [78H]

レジスタ  
直接

メモリ  
間接

12H

34H

⋮

34H

78H

(参) LD A, 12H

直接

A [34H]

LDI A, 12H

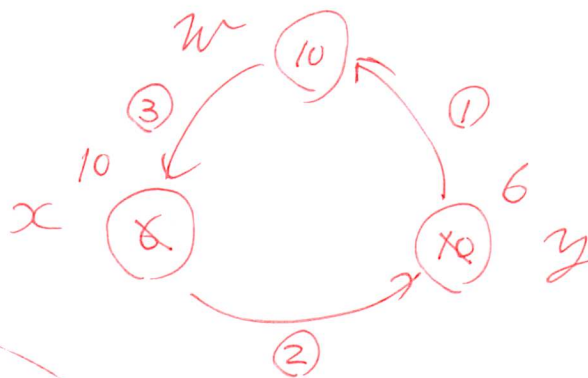
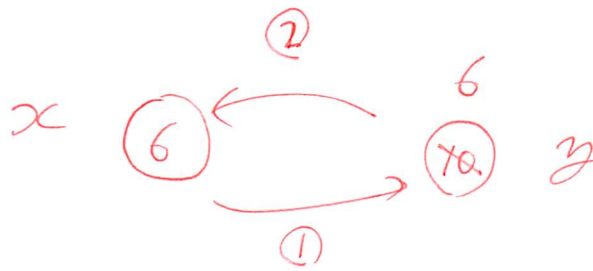
即値

(イミディエート)

A [12H]

間接 ... ~~メモリ~~ ブロックの処理に用いる

(例) ブロック転送



$$x = 3$$

前置

$$y = ++x;$$

值: 4

后置

$$y = x++;$$

值 3

副作用

$x$  4

$$y = x++;$$

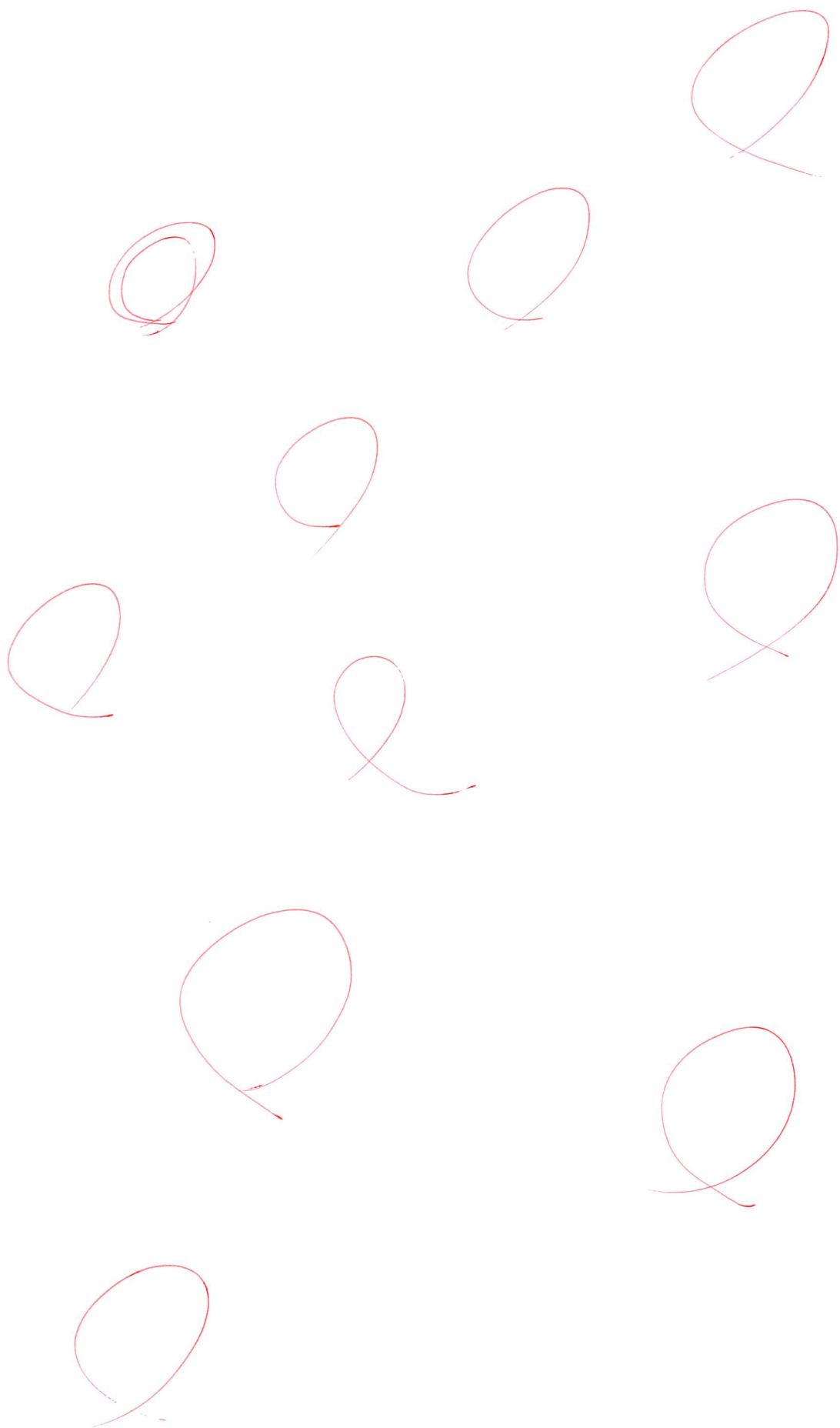
2013

$x$  2014  $y$  2013

$$z = ++y;$$

2014

$y$  2014  $z$  2014



2014.05.30

$x = (y = 30);$   
②      ①  
        └───┘  
        値: 30  
└──────────┘

代入演算子の優先順位  
右から左へ

代入も評価 evaluation  
される。

---

$x = 30;$

$y = (x++);$   
        └───┘  
        30

- 代入の優先順位は最下位
- ++は最優先

$x++$  が  $x$  の内容を +1 するのは、  
評価の結果ではない。  
→ 「副作用」とみえる  
 $x++$  の評価結果は「 $x$  の値」

# Search 探<sup>サウ</sup>

## ○ 線形探○

片はしから順に探す。

## ○ <sup>ブン</sup>2分探<sup>サウ</sup>

条件 : 整列していること。  
(ソート済)



12コのボールがあり、

1コだけ他とは重さが異なる。

(他の11コは同じ重さ)

天びんを3回だけ使って、

重さの異なるボールを探せ。

24通りの可能性

$$3^3 = 27 \neq 24$$

~~~~~

13コの場合 ---- 26通り

$$3^3 > 26$$

理論的に「解けることが分かる」

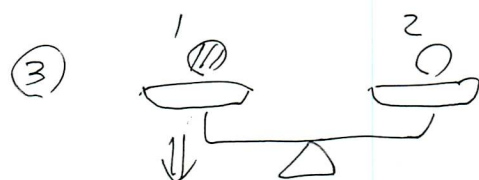
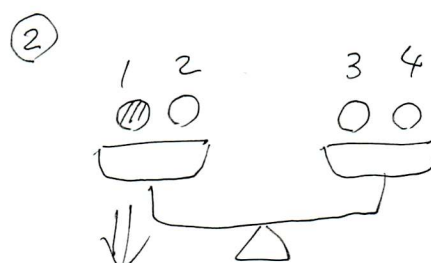
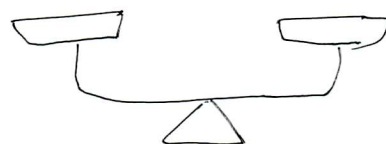
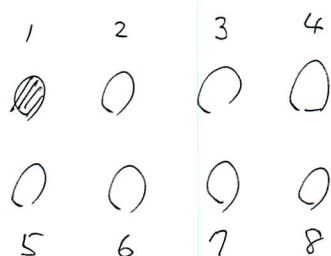
(可能性がある)

アルゴリズム論

(例) NP問題

8コのボールがあり、
1コだけ他より重い。
(他の7コは同じ重さ)

天秤を3回だけ使って、
重いボールを探せ。



$$2^3 = 8 \text{ 通り}$$

情報量 : 可能性を $\frac{1}{n}$ に絞る 情報量 I
 $= n^{-1}$

3通り

$$\log_2 3$$

=

$$I = -\log_2 \frac{1}{n}$$

$$= \log_2 n$$

$$= -\log_2 p (> 0)$$