第1編　コンピュータとディジタル情報

第0章　位取記数法
第1章　コンピュータのしくみ
第2章　情報のディジタル表現

1編　コンピュータとディジタル表現　概要

① コンピュータにおけるデータ(0,1)の表現方法
（電圧，磁石，凹凸，電荷等）
② 電子計算機の動作を意識する･･･2進数の処理確認
（加法･シフト･補数･誤差 等）
③ ハードウェア＋ソフトウェア＝コンピュータ
④ コンピュータの内部構造･外部装置･インターフェース
⑤ データの量を表す単位（ビット，バイト，接頭辞）
⑥ コンピュータが計算するしくみ（論理演算と論理回路）
⑦ analogをdigitalへ(量子化)･･･　そして0と1だけに(符号化)･･･
（ディジタル化のデメリット･メリット）

ディジタル化のデメリット･メリット　再掲

関連：pp.14-15

デメリット　ディジタル化するとき
- 情報の劣化：大　（連続量 ⇒ 離散量）

メリット　ディジタル化されたデータは
- 数値，文字，音，画像，動画等，何でもまとめて取り扱える
（すべてのデータが0と1に過ぎないので）　<････方法はのちほど
- 必要に応じてデータ量を調整（圧縮）　<････方法はのちほど
- 漏洩防止が可能（暗号化）　<････方法はのちほど
- 劣化しにくい（修復が容易）
- 情報の編集が容易
- 情報検索が容易

ディジタルデータの量を示す基本単位

関連：pp.28-30

- 実在する，区別できる**「何か」**を2進数と**1対1対応**させる
⇔ **「何か」**を**2進数**で表現する。
- データ(情報)が多い　⇔　2進数で多くの**桁**が必要

ビット(bit)　「何か」を2進数と1対1対応させた時に，必要な2進数の桁数（binary digit）
（コンピュータがデータを扱う時の基本単位）
ビット
最小単位

1ビット（binary digit）のデータとは･･･
- 1桁の2進数で表現（区別）できるデータの量（つまり2通りまで）
- ビットを小文字で **bit** や **b** と書くこともある

nビット のデータとは･･･
- 「n桁までの2進数」で表現（区別）できるデータの量（2^n通りまで）

よく使うデータ量の単位

関連：pp.28-30

ビット(bit)　2進数の桁数（binary digit）
コンピュータがデータを扱う時の基本単位
ビット
最小単位

バイト(byte)　1バイト ＝ 8ビット
バイト
よく使う単位

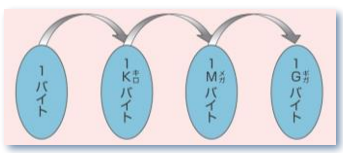
nビット のデータとは･･･
- 「n桁までの2進数」で表現（区別）できるデータの量（2^n通りまで）

1バイト のデータとは･･･
- 「8桁までの2進数」で表現（区別）できるデータの量（2^8通りまで）
- バイトを **B** と書くこともある

1バイト(8ビット)で **256**(＝2^8)通りの状態が表現可能。

バイトとともに使う接頭語（接頭辞）

関連：p.30



注意点
- 単位の表記は10^3倍ではなく2^{10}倍(2進接頭辞)で変化させる場合が多い。(10^3形式も使われる)
- KB(キロバイト)のKは大文字。
- KBを「ケーバイト」と読む場合あり。

- 1B ＝ 8bit
↓ 1024(=2^{10})倍
- 1 KB（　　バイト）
↓ 1024倍
- 1 MB（　　バイト）
↓ 1024倍
- 1 GB（　　バイト）
↓ 1024倍
- 1 TB（　　バイト）
↓ 1024倍
- 1 PB（　　バイト）
↓ 1024倍
- 1 EB（　　バイト）

接頭語（接頭辞）の補足説明

関連：p.30

バイトの単位一覧

SI接頭辞			2進接頭辞			
単位（記号）	慣用値	SI基準	単位（記号）	値	SIとの差(概数)
キロバイト (kB)	2^{10}	10^3	キビバイト (KiB)	2^{10}	2.400000%
メガバイト (MB)	2^{20}	10^6	メビバイト (MiB)	2^{20}	4.857600%
ギガバイト (GB)	2^{30}	10^9	ギビバイト (GiB)	2^{30}	7.374182%
テラバイト (TB)	2^{40}	10^{12}	テビバイト (TiB)	2^{40}	9.951163%
ペタバイト (PB)	2^{50}	10^{15}	ペビバイト (PiB)	2^{50}	12.589991%

図：ウィキペディアより

SI：国際単位系
International System

単位	英語名（省略形）	データ量（情報量）
キビバイト	Kilobinary Byte= Kibi Byte （KiB）	1KiB = 1,024B= 2^{10} = 1,024 Byte
メビバイト	Megabinary Byte=Mebi Byte （MiB）	1MiB = 1,024KiB= $(2^{10})^2$ = 2^{20} = 1,048,576 Byte
ギビバイト	Gigabinary Byte= Gibi Byte （GiB）	1GiB = 1,024MiB = $(2^{10})^3$ = 2^{30} = 1,073,741,824 Byte
テビバイト	Terabinary Byte = Tebi Byte （TiB）	1TiB = 1,024GiB = $(2^{10})^4$ = 2^{40} = 1,099,511,627,776 Byte

参考：情報の記録媒体（記録メディア）の例

名称	代表的な容量の例	種類
フロッピーディスク	1.44MB	磁気ディスク
MO	128MB～2.3GB	光磁気ディスク
CD－R	650MB，700MB	光ディスク
DVD－R	4.7GB，8.5GB (DVD-R DL)	光ディスク
コンパクトフラッシュ	16MB～256GB	フラッシュメモリ
SDメモリカード (SDHC,SDXC)	16MB～512GB	フラッシュメモリ
USBメモリ	256MB～2TB	フラッシュメモリ
ハードディスク	80GB～10TB	磁気ディスク

注意：多くのメディアの容量は，この表以外にもたくさんあり，増え続けている。
（おおよその比較のための参考値としてください）

数のディジタル表現（予備知識）

指数表記

数値の表記例

例　1230000000000 = 123×10^{10} = **1.23×10^{12}**

例　0.0123 = 123÷10^4 = 123×10^{-4} = **1.23×10^{-2}**

$a^{-n} = 1/a^n$
(逆数)

指数表記とは

$\pm a.bcde\cdots \times 10^n$ の形式の数値表現（ただしaは 1～9の整数）

※ コンピュータでは，画面に指数が表示できない場合があるので
例　1.23×10^{12} を 1.23E12
例　1.23×10^{-2} を 1.23E-2
のように 表現することが多い。

コンピュータに関連した指数表記の予備知識

2進数の場合，$\pm 1.bcde\cdots \times 10_2^n$ （bcde･･･,nを構成する数：全て0か1）

数のディジタル表現は3パターン

関連：p.31

固定小数点表示（整数型）
- 符号なし整数（例　0，1，2，3，･･･）
- 符号付き整数（例　･･･，－2，－1，0，1，･･･）

浮動小数点表示（実数型）
- ±仮数部×基数^{指数部} の形式　（例　6.02×10^{23}）

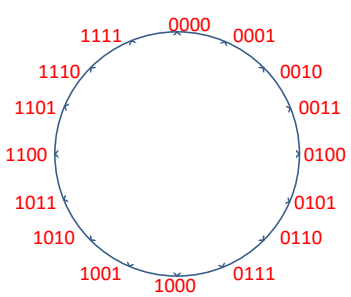
基本は，**0と1の並び**（ビット列）と**表現するもの**との**1対1対応**

問：4bitで，何個の数を表現できるか。

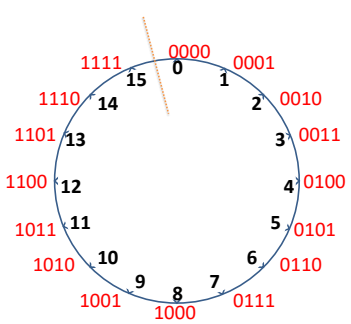
問：32bitで，何個の数を表現できるか。

実習（別紙）：上記3通りの方法で，5ビットの数を表現してみよう。
- 固定小数点表示（符号なし整数，符号付き整数）
- 浮動小数点表示（±仮数部×基数^{指数部} の形式）

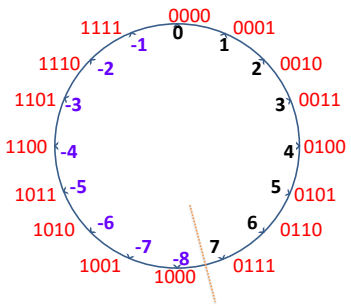
符号なし整数と符号付き整数を図示(4bit)



符号なし整数と符号付き整数を図示(4bit)



符号なし整数と符号付き整数を図示(4bit)



10000

実数(小数)や大きい数(概数)を表現する方法

00000　00001　00010　00011　00100　・・・・・
・・・・・　11011　11100　11101　11110　11111

これらを $\pm 1.\square \times 2^{\square}$ の形として扱う(浮動小数点表示)

符号　仮数 × 基数(底ともいう) 指数

つまり □□□□□□を 符号・指数・仮数(小数部分)に振り分ける

- ±について ・・・・ 0⇔＋　1⇔－
- 「1.」と「×2」 ・・・・ 表現する必要なし
- 0はこの方式で表現できない ・・・・特別ルール必要

順番に注意！

※ 厳密には更に詳細な規則が多く存在するが，ここでは簡略化した実習プリントでざっくりと感覚をつかもう。

数のディジタル表現の実際(具体例)

基本は1対1対応

よく使われる32bitだと次の通り

- 2進数32桁 ⇔ 2^{32}(= 42億9496万7296)通りの状態を表せる。

□□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□

固定小数点表示(整数型)

- 符号なし整数(0～42億9496万7295まで)
- 符号付き整数(-21億4748万3648～+21億4748万3647まで)

浮動小数点表示(実数型)

- 符号部 & 仮数部 × 基数 指数部 の形式(例：-1.23456×10^{-12})

□　□□□□□□□□　□□□・・・23個並ぶ・・・□□□

符号部(1bit)　指数部(8bit)　仮数部(23bit)

- ±ともに 約10^{-45}～約10^{38} まで

メリット：有効数字を適切に保ちながら幅広く実数を表現できる

□□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□

参考：浮動小数点表示の具体的説明1

32bit浮動小数点表示(IEEE754)で表現可能な範囲を求める。

符号部 & 仮数部 × 基数 指数部 の形式

符号部(1bit)　指数部(8bit)　仮数部(23bit)

- 符号部(1bit)　＋⇒0　－⇒1
- 指数部(8bit)　0000 0000～1111 1111
- 仮数部(23bit)　0が23個～1が23個(2^{23}個)

種類	Exp(指数部)	Fraction(仮数部)
ゼロ	0	0
非正規化数	0	0以外
正規化数	1～254	任意
無限大	255	0
NaN	255	0以外の任意

□□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□

考察1：表現できる最大・最小の数(±無限大を除いて)

0 & 1.1111・・・・(23個)・・・・1111 × $2^{11111110}$　☜ 2進数

≒ ＋ 2 × 2^{127} = 2^{128} ＝ 3.4028・・・ × 10^{38}　☜ 10進数

指数部：「**符号なし整数(0～255)－bias(127)**」なので **−127～+128**。

しかし，0と255(bias計算後の−127と128)は0,∞等を意味するので，

実際は **−126～+127** の値をとる。

参考：浮動小数点表示の具体的説明2

32bit浮動小数点表示(IEEE754)で表現可能な範囲を求める。

符号部 & 仮数部 × 基数 指数部 の形式

符号部(1bit)　指数部(8bit)　仮数部(23bit)

- 符号部(1bit)　＋⇒0　－⇒1
- 指数部(8bit)　0000 0000～1111 1111
- 仮数部(23bit)　0が23個～1が23個(2^{23}個)

種類	Exp(指数部)	Fraction(仮数部)
ゼロ	0	0
非正規化数	0	0以外
正規化数	1～254	任意
無限大	255	0
NaN	255	0以外の任意

□□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□

考察2：表現できる0に近い数について

0 & 1.00・・・(23個)・・・000 × $2^{0000\ 0000}$ = 0とする(符号部以外0)。

さらに，1 0000 0000 00・・・(23個)・・・000 も0とみなす。

0 & 1.00・・・(23個)・・・001 × $2^{0000\ 0000}$ (指数部が全部0)は，例外的に

＋0. 00・・・・001 × 2^{-127} ⇒ $2^{-23} \times 2^{-126}$ = 2^{-149} ＝ 1.4012・・・ × 10^{-45}

とする(非正規化数)。2^{-127} とせず 2^{-126} であるのも規則による結果。

数のディジタル表現

参考：データ型の例(BASIC言語)

変数のデータ型

変数の用途のこと。例えば「数値」なのか，「数値」でも「整数」なのか「実数」なのか等。

変数宣言の例

文字列型
Dim A As String　0～2GB

整数型
Dim C As Integer　-32,768～32,767の範囲
Dim D As Long　-2,147,483,648～2,147,483,647の範囲

実数型
Dim E As Single　-3.402823E38～-1.401298E-45(負), 1.401298E-45～3.402823E38(正)
Dim F As Double　-1.79769313486232E308～-4.94065645841247E-324(負),
4.94065645841247E-324～1.79769313486232E308(正)の範囲

通貨型
Dim G As Currency　-922,337,203,685,477.5808～922,337,203,685,477.5807

日付型
Dim J As Date　西暦100年1月1日～西暦9999年12月31日

(1.4012・・・) × 10^{-45}
～
(3.4028・・・) × 10^{38}

コンピュータが行う演算について

関連：p.-

◆算術演算　加減乗除など

◆シフト演算　2進数のビットパターンを右か左にずらす演算

- 例 5ビットの場合

$00010_{(2)}$　左に1ビット　$00100_{(2)}$　右に1ビット　$00001_{(2)}$

$11000_{(2)}$　符号なし整数だと？　左　右

符号付き整数だと？　左　右

◆論理演算　0：偽（false），1：真（true）とみなす演算

- 0,1を論理値（logical value）とか真偽値（boolean）,真理値ともいう。
- 論理演算はブール演算（boolean operation）ともいう。
- AND,OR,NOTを学習するが，他にNAND, NOR, XOR, EQ(XNOR)等もある。

論理演算とは

関連：p.-

論理演算：0偽と1真だけの計算

（該当しない）　（該当する）

1つの値に対する演算F(x)の例

- 「NOT演算」　論理式 $F=\overline{A}$

2つの値に対する演算$F(x_1,x_2)$の例

- 「論理和(OR演算)」　論理式 $F=A+B$
- 「論理積(AND演算)」　論理式 $F=A\cdot B$
- 「排他的論理和(XOR演算)」

右図：Googleの画面コピー

論理演算とは

関連：p.-

論理演算：0偽と1真だけの計算

1つの値に対する演算F(x)の例

- 「NOT演算」　論理式 $F=\overline{A}$

2つの値に対する演算$F(x_1,x_2)$の例

- 「論理和(OR演算)」　論理式 $F=A+B$
- 「論理積(AND演算)」　論理式 $F=A\cdot B$
- 「排他的論理和(XOR演算)」

1つの値に対する演算とは…

A	F0	F1	F2	F3
0	0	0	1	1
1	0	1	0	1

例（真偽が確実なものを扱う）
A: 灘区に住んでいる。
F: ・・・。

論理演算とは

関連：p.-

論理演算：0偽と1真だけの計算

1つの値に対する演算F(x)の例

- 「NOT演算」　論理式 $F=\overline{A}$

2つの値に対する演算$F(x_1,x_2)$の例

- 「論理和(OR演算)」　論理式 $F=A+B$
- 「論理積(AND演算)」　論理式 $F=A\cdot B$
- 「排他的論理和(XOR演算)」

2つの値に対する演算とは…

A	B	F0	F1	F2	F3	F4	F5	F6	F7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

A	B	F8	F9	F10	F11	F12	F13	F14	F15
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

例（真偽が確実なものを扱う）
A: 灘区に住んでいる。 B: 男子である。
F: ・・・。

論理演算の基本（まとめ）

関連：p.-

論理演算：　0（偽）と1（真）だけの計算

1つの値に対する演算　「NOT演算」

論理式 $F=\overline{A}$

2つの値に対する演算の例

「論理和(OR演算)」　「論理積(AND演算)」　「排他的論理和(XOR演算)」

論理式 $F=A+B$　論理式 $F=A\cdot B$

A	B	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

論理演算の計算練習

関連：p.-

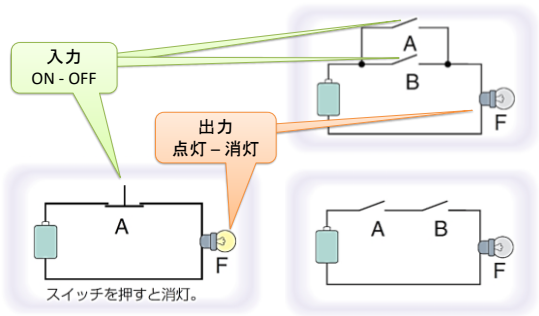
※　かっこ等の計算順序は、今までの計算と同様である。

1. $\overline{A}$
2. $\overline{\overline{A}}$
3. $A+B$
4. $A\cdot B$
5. $\overline{A}+B$
6. $\overline{A\cdot B}$
7. $\overline{A}\cdot B+B$
8. $A+\overline{B}\cdot A$
9. $(A+B)\cdot A$
10. $\overline{(A+B)\cdot A}$
11. $\overline{(A+\overline{A})}$
12. $A\cdot B+\overline{A\cdot B}$
13. $A\cdot\overline{B}+\overline{A}\cdot B$

※　3つ以上の数A,B,Cに対しても、この様な計算を定義することができる。

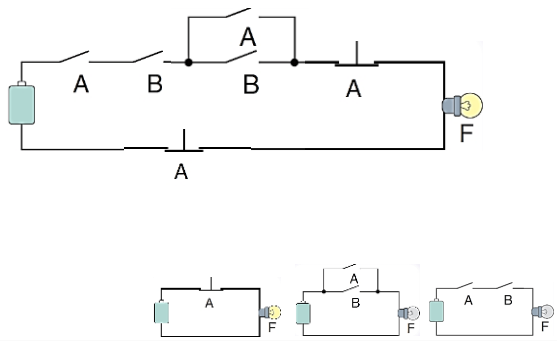
乾電池・豆電球で作ったイメージ・・・

関連：p.-



乾電池・豆電球は「例えに過ぎない」・・・

関連：p.-



CPUが計算するしくみと基本論理回路

関連：p.34

論理演算：

「NOT演算」

「OR演算」

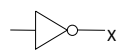
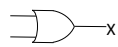
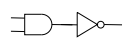
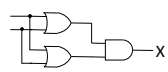
「AND演算」

論理回路を使った計算練習

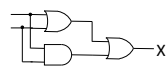
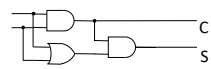
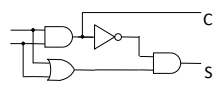
関連：p.-

※ 入力端子は，上がA，下がBとする。

問題1 式で書け（論理演算として記述）。

1. 
2. 
3.
4. 
5. 

問題2 計算せよ。

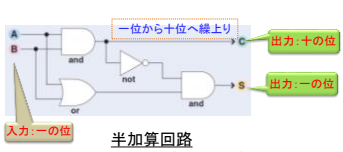
6. 
7. 
8. 

2進数1桁の足し算ができる回路（2桁表示）

関連：p.-

A	B	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

0＋0＝ 0
0＋1＝ 1
1＋0＝ 1
1＋1＝1 0



半加算回路
AND: 2個，OR: 1個，
NOT: 1個 を 繋げればよい

2進数2桁までの足し算ができる回路（3桁表示）

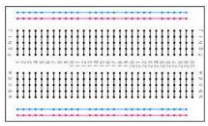
	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
A	00	00	00	00	01	01	01	01	10	10	10	10	11	11	11	11
B	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
X	000	001	010	011	001	010	011	100	010	011	100	101	011	100	101	110

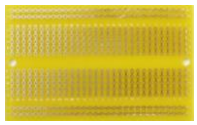
半加算回路3つ＋OR
AND:6個，OR:4個，
NOT:3個 を 繋げばできる

ブレッドボード

はんだを使わずに回路を組める基盤のこと

しくみ ⇒





電子部品等

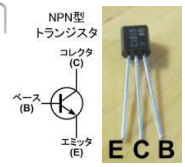
関連：p.34

抵抗器：接続する向きはどちらでもよい

- 茶黒橙金：10kΩ ・・・ 6個
- 茶黒赤金：1kΩ ・・・・ 2個

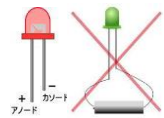
トランジスタ：接続に注意 ・・・ 4個

- Bから（Eに）わずかな電流 ⇒ Cから（Eに）100倍以上の電流が流れる（増幅作用）。
  ベースへの接続(ON,OFF)をスイッチとする



発光ダイオード：接続の向き（慎重！）・・・2個

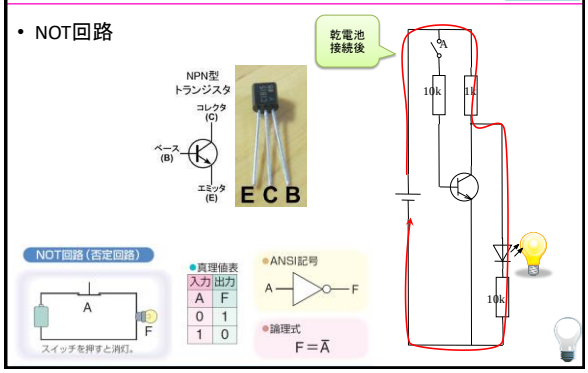
- 出力の確認用
- （好奇心等で）電池に直接つながないこと



基本論理回路の作製（NOT回路　説明1）

関連：p.34

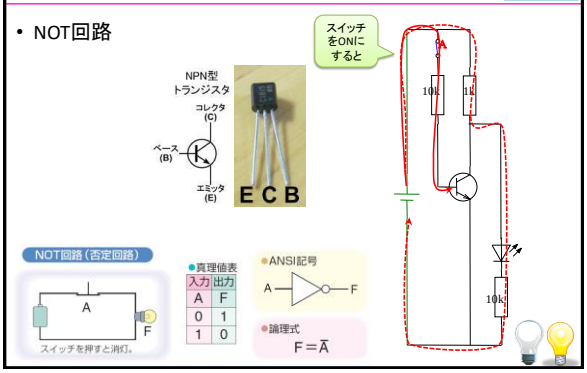
- NOT回路



基本論理回路の作製（NOT回路　説明2）

関連：p.34

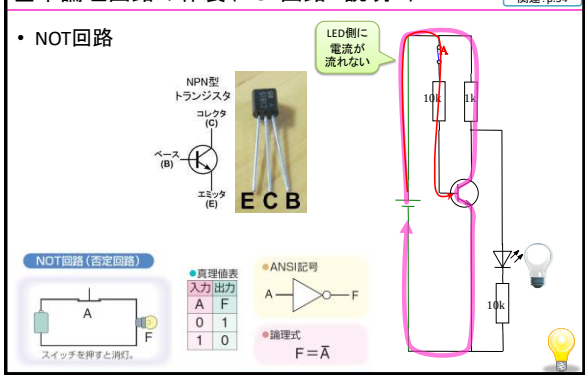
- NOT回路



基本論理回路の作製（NOT回路　説明3）

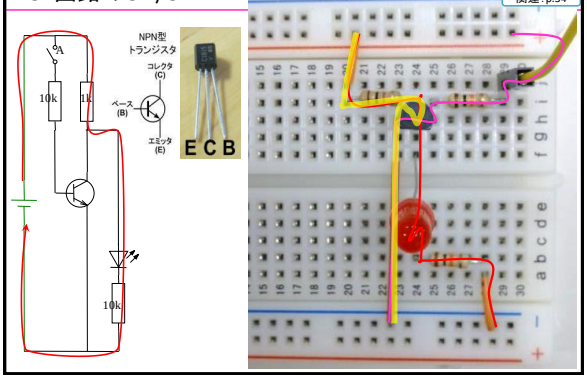
関連：p.34

- NOT回路



NOT回路のON/OFF

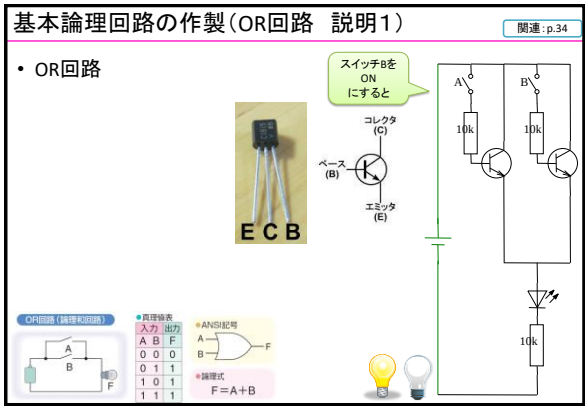
関連：p.34



基本論理回路の作製（OR回路　説明1）

関連：p.34

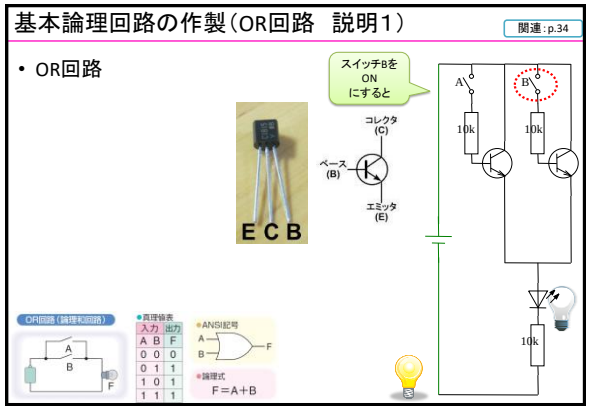
- OR回路



基本論理回路の作製（OR回路　説明1）

関連：p.34

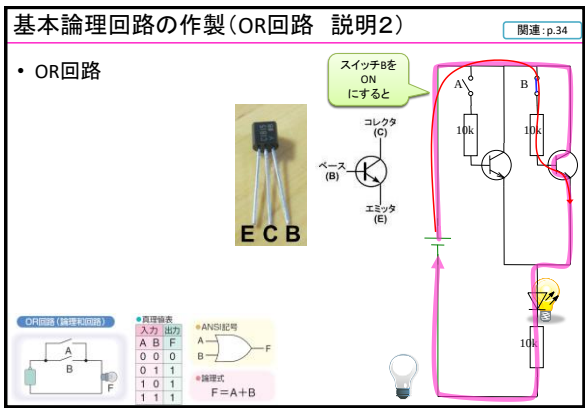
- OR回路



基本論理回路の作製（OR回路　説明2）

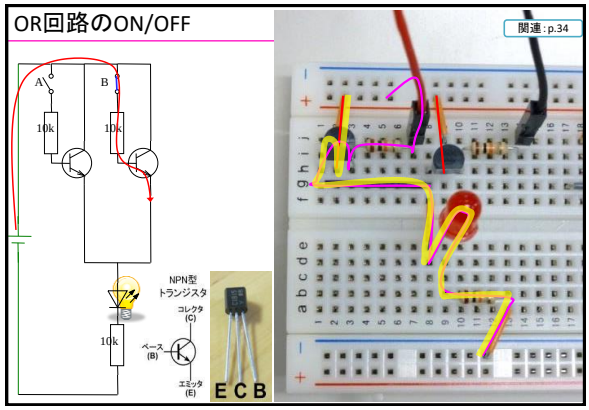
関連：p.34

- OR回路



OR回路のON/OFF

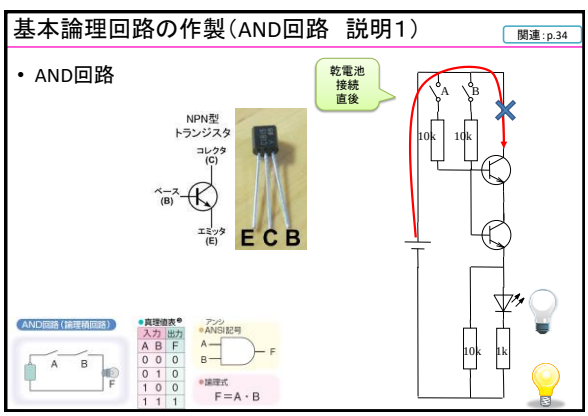
関連：p.34



基本論理回路の作製（AND回路　説明1）

関連：p.34

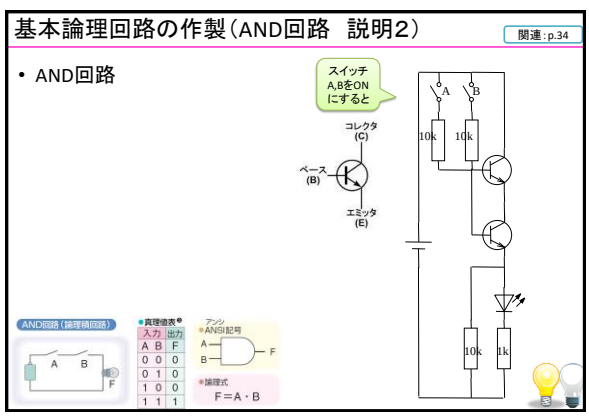
- AND回路



基本論理回路の作製（AND回路　説明2）

関連：p.34

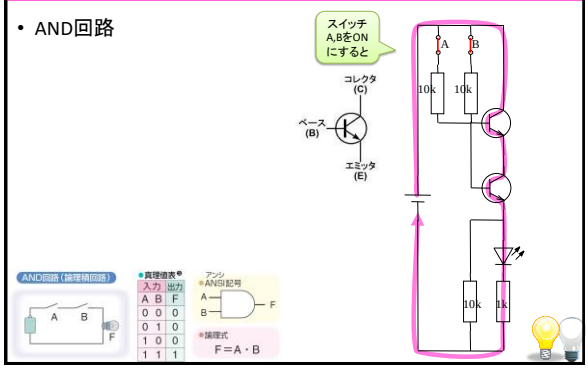
- AND回路



基本論理回路の作製（AND回路　説明2）

関連：p.34

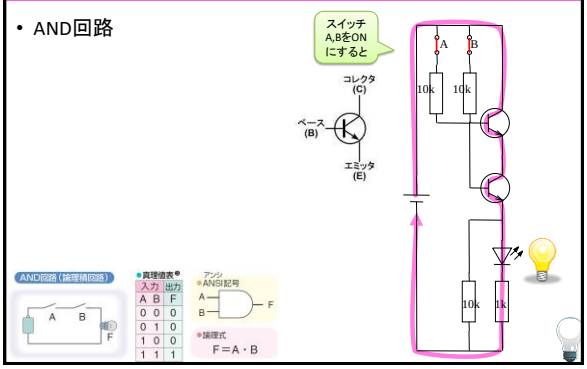
- AND回路



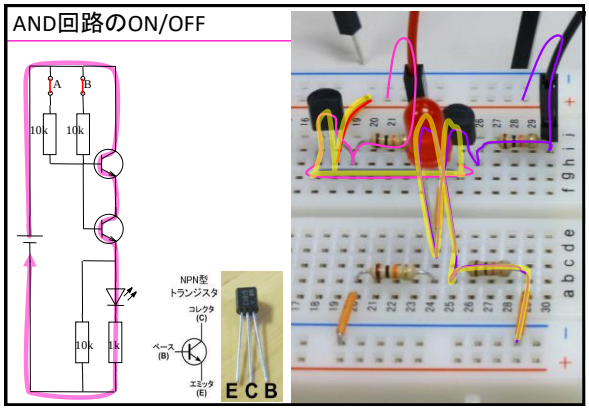
基本論理回路の作製（AND回路　説明2）

関連：p.34

- AND回路



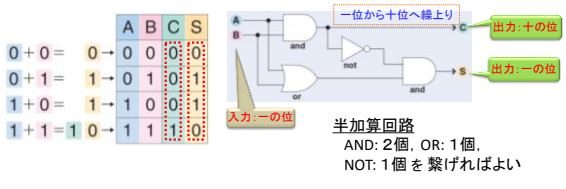
AND回路のON/OFF



2進数1桁の足し算ができる回路（2桁表示）

関連：p.34

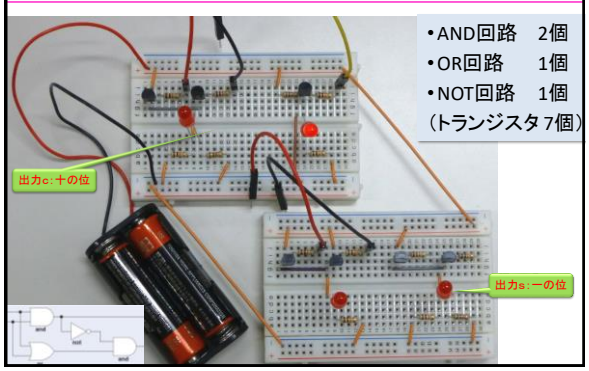
A	B	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1



半加算回路
AND：2個，OR：1個，
NOT：1個 を 繋げればよい

基本論理回路を接続して計算機にする

関連：p.34



- AND回路　2個
- OR回路　1個
- NOT回路　1個

（トランジスタ7個）

2進数2桁までの足し算ができる回路（3桁表示）

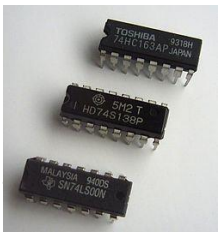
関連：pp.34-35

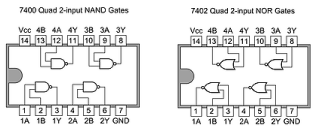
	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
A	00	00	00	00	01	01	01	01	10	10	10	10	11	11	11	11
B	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
X	000	001	010	011	001	010	011	100	010	011	100	101	011	100	101	110

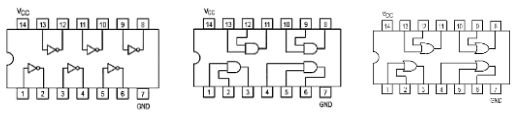
半加算回路3つ＋OR
AND:6個，OR:4個，
NOT:3個 を 繋げばできる

ICの例（論理回路の学習等でよく使われる汎用ロジックIC）

関連：pp.34-35





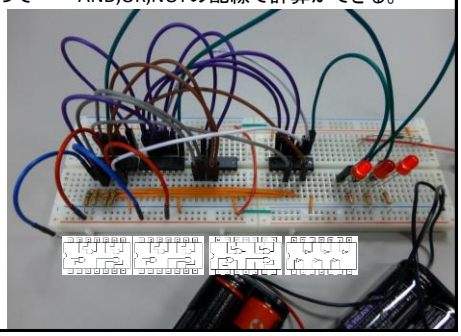


2進数2桁までの足し算ができる回路（3桁表示）

関連：pp.34-35

- ICを4個使って・・・ AND,OR,NOTの配線で計算ができる。

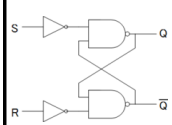
写真は
青：一の位
赤：十の位
LED：3桁表示
10+11=101



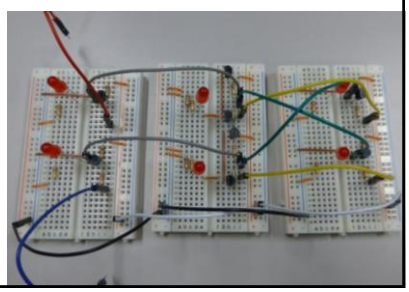
回路で記憶（メモリ）

関連：pp.34-35

- フリップフロップ・・・・
  1ビットの情報を一時的に「0」または「1」として記憶できる，論理回路

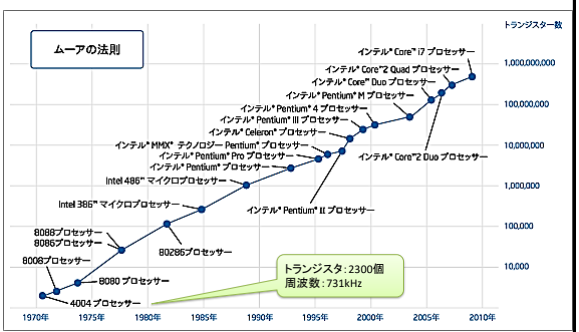


計算だけでなく
記憶も
できる！



トランジスタの集積（IC⇒LSI⇒VLSI）

関連：pp.34-35



- http://japan.intel.com/contents/museum/processor/