

基礎コンピュータ工学

第2章 情報の表現

(パート5)

コンピュータの基本回路

論理回路を組合せてコンピュータは製作される。

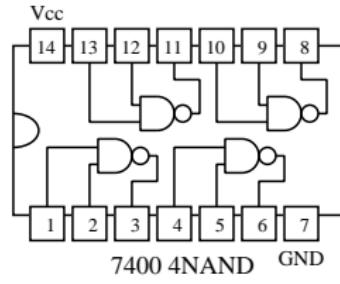
- 電気の OFF と ON だけ用いて情報を表現する。
- OFF/ON をビット 0/1 に対応付ける。
- これは**論理値** (True/False, 真/偽, Yes/No) と同じ。
- 論理値を対象とする演算を**論理演算**と言う。
- 論理演算を計算する回路を**論理回路**と言う。



論理 IC =>

外観

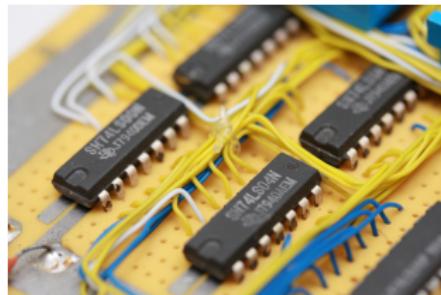
Vcc ----- 5V電源の「+」を配線する。
GND ----- 5V電源の「-」を配線する。



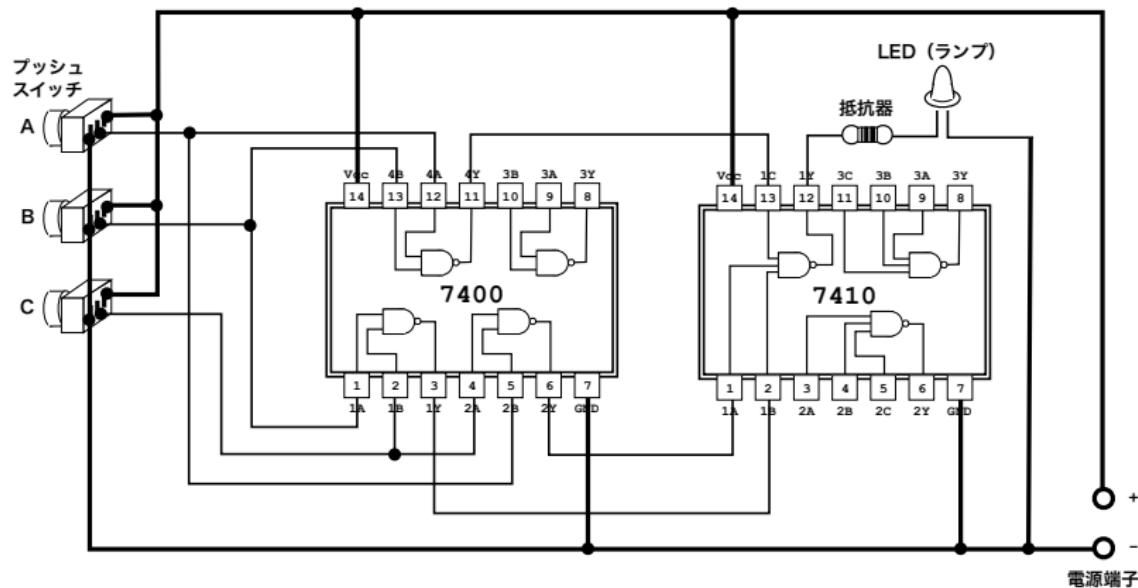
論理ICの内部

論理 IC で作った手作りコンピュータ

徳山高専卒業研究で約 30 年前に製作された手作りコンピュータ

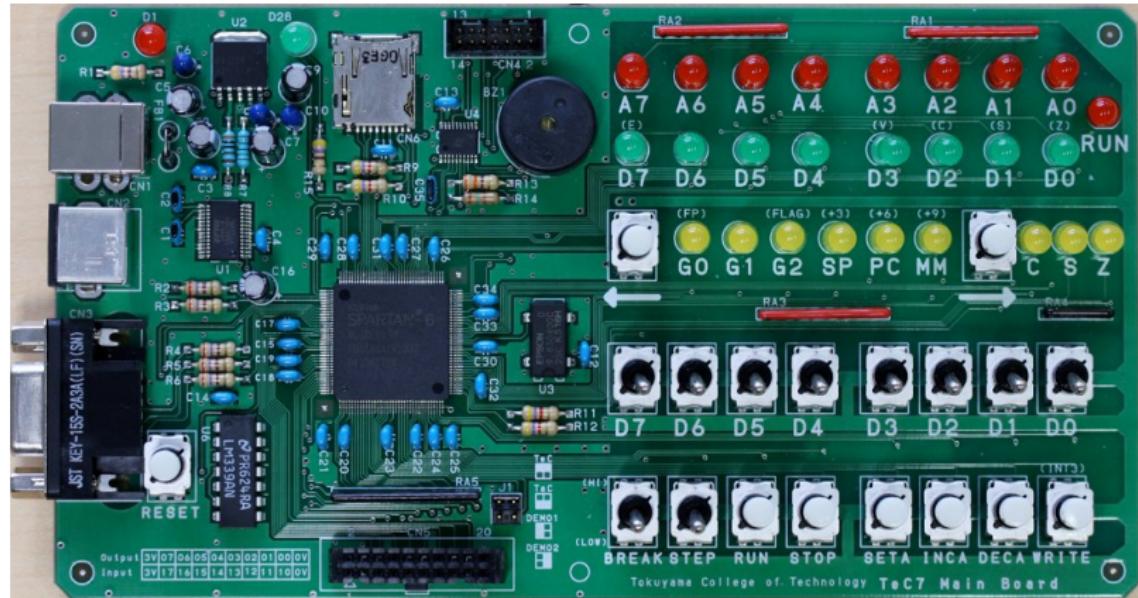


論理 IC で回路を作る



現代の手作りコンピュータ

FPGAの中に論理 IC 数万個に相当する回路が書き込める。



基本的な論理回路（1）

論理積（AND）—「かつ」

2ビット入力し、両方が1のときだけ1を出力する。

入力		出力
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

ANDの真理値表

$$X = A \cdot B$$

ANDの論理式



ANDの回路記号

A かつ B が1なら1

A AND B が1なら1

基本的な論理回路（2）

論理和 (OR) — 「または」

2ビット入力し、どちらかが1のとき1を出力する。

入力		出力
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

ORの真理値表

$$X = A + B$$

ORの論理式



ORの回路記号

A または B が1なら1

A OR B が1なら1

基本的な論理回路（3）

否定 (NOT) — 「ではない」

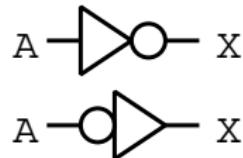
1 ビット入力し、入力とは逆の論理値をを出力する。

入力	出力
A	X
0	1
1	0

NOTの真理値表

$$X = \overline{A}$$

NOTの論理式



NOTの回路記号

A が 1 ではないなら 1

基本的な論理回路（4）

排他的論理和 (XOR) — 「異なる」

2ビット入力し、二つが異なるなら 1 を出力する。

入力	出力	
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

XORの真理値表

$$X = A \oplus B$$

XORの論理式



XORの回路記号

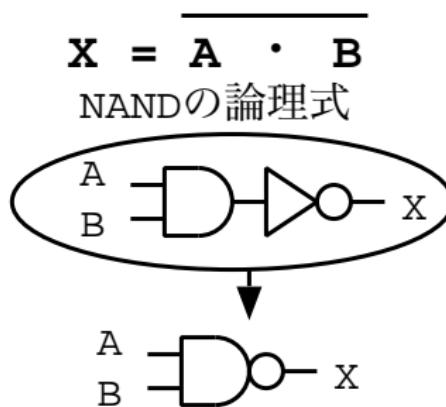
A と B が異なるなら 1

基本的な論理回路（5）

NOTとANDの組合せ (NAND)

入力	出力	
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NANDの真理値表



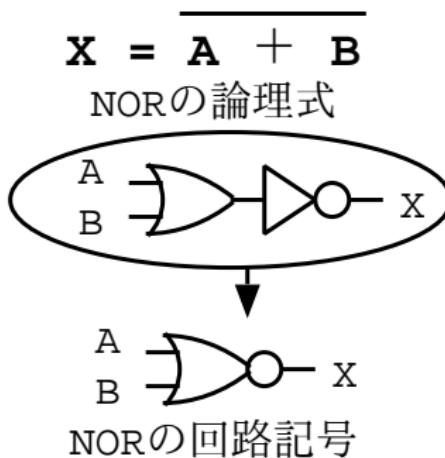
NANDの回路記号

基本的な論理回路 (6)

NOTとORの組合せ (NOR)

入力	出力	
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

NORの真理値表



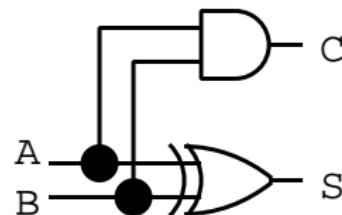
演算回路（1）

半加算器

1桁の2進数を二つ入力し、0, 1, 2のどれかを出力する。

$$\begin{array}{r} \text{A} & 0 & 0 & 1 & 1 \\ + \text{B} & + 0 & + 1 & + 0 & + 1 \\ \hline \text{C} \text{ S} & 0 \ 0 & 0 \ 1 & 0 \ 1 & 1 \ 0 \end{array}$$

入力		出力	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



半加算器の真理値表 半加算器の回路図

演算回路（2）

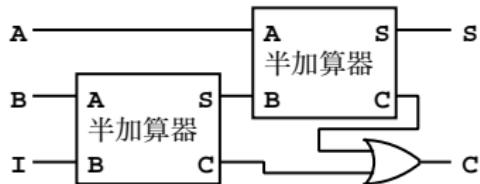
全加算器

1桁の2進数を三つ入力し、0, 1, 2, 3のどれかを出力する。

入力	出力	
A	C	S
0 0 0	0 0	
0 0 1	0 1	
0 1 0	0 1	
0 1 1	1 0	
1 0 0	0 1	
1 0 1	1 0	
1 1 0	1 0	
1 1 1	1 1	

A:計算の入力
B:計算の入力
I:桁上がりの入力

全加算器の真理値表

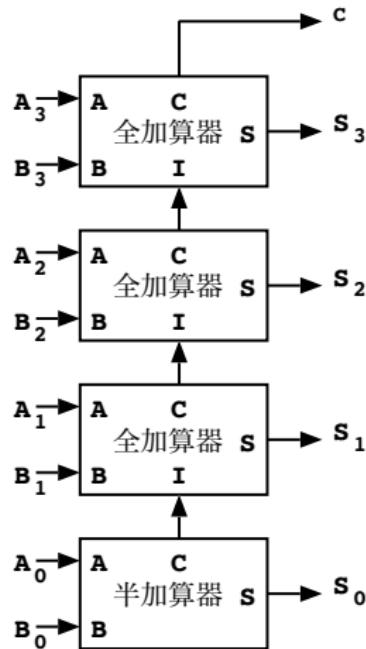


全加算器の回路図

演算回路（3）

4ビット加算器

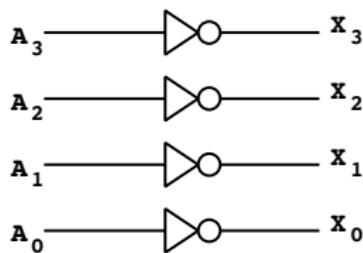
4桁の2進数を二つ入力し、和を計算する



演算回路（4）

4ビット 1の補数器

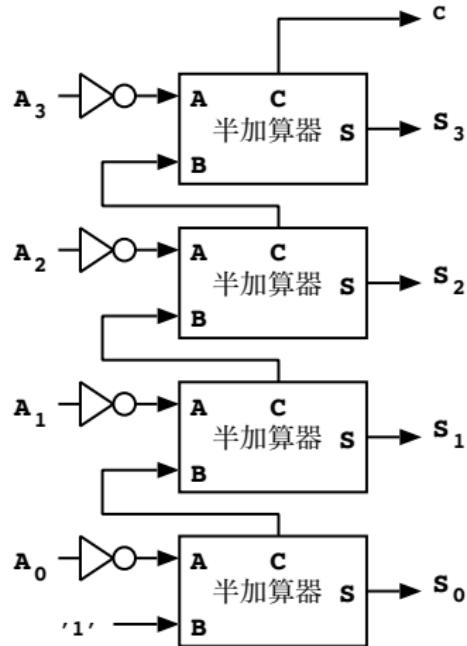
4桁の2進数を入力し、1の補数を計算する



演算回路（5）

4ビット 2の補数器

4桁の2進数を入力し、2の補数を計算する



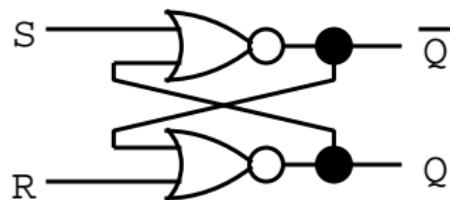
記憶回路

RS フリップフロップ

直前の状態を記憶する回路

入力		出力
S	R	Q
0	0	記憶
0	1	0
1	0	1
1	1	禁止

RS-FFの動作



S: Set
R: Reset

RS-FFの回路