第 1 章

コンピュータによる情報の処理と表現

第1節
コンピュータと情報処理

2
コンピュータの動作のしくみ2

＜前時の振り返り＞

コンピュータが扱うディジタルデータは、

- □量を□的な数値として表現する形式
- □進数で表現される
- 最小単位＝1□(1□)(1b)
- □＝1□(1□) (1B)

＜前時の振り返り＞

- [　　]
  コンピュータの機械そのものや
  コンピュータに接続されたさまざまな装置
  物理的実体を持つ
- [　　]
  ハードウェアを動作させるプログラム
  物理的実体を持たない

- コンピュータの五大機能
  [　　]・[　　]・[　　]・[　　]・[　　]

＜本時のめあて＞

コンピュータ内部で行われているデータ処理のしくみについて理解するために、

・3つの論理回路

・CPUとメインメモリの動作

について理解する。

論理回路

↓

論理演算や記憶を行なう電子回路

CPUはいくつかの論理回路を組み合わせてできている

論理回路
＝論理演算を行なう電子回路
→演算と演算の違いは?

	演算	計算
広辞苑 第7版 定義	数式の示す通りに所要の 数値を計算すること。	演算をして 結果を求め出すこと。
相違点	数値を計算する行為	演算で求められた結果
用例	演算子	計算式

論理回路

- デジタル信号を処理して論理演算や記憶などを行うための電子回路。
- 単純な論理演算を行う回路を膨大な数組み合わせれば
  CPU（MPU/マイクロプロセッサ）のような
  複雑な装置を作ることができる。
- 信号の状態は論理的には
  2進数の「0」と「1」あるいは真偽値の「真」と「偽」に対応し、
  物理的には電圧の高低で表わすことが多い。
- 二状態のいずれかを取るデジタル信号を入力および出力する
  論理演算を行なう機能を持った単体の素子である論理素子を
  配線で結び、様々な論理演算や記憶を行う回路を構成する。

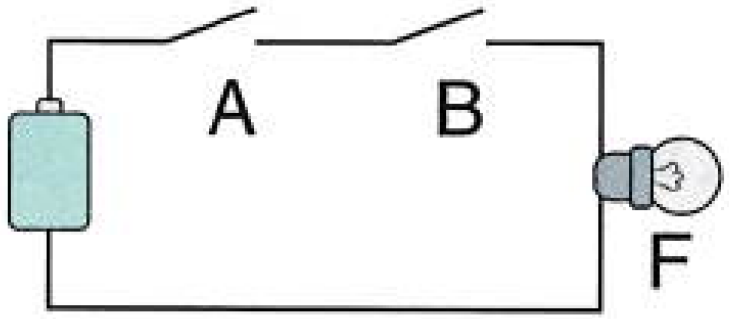
3つの論理回路

- AND回路
- OR回路
- NOT回路

AND回路
(論理積回路)

2つの入力が
共に1の時だけ
出力が1になる

真理値表↓

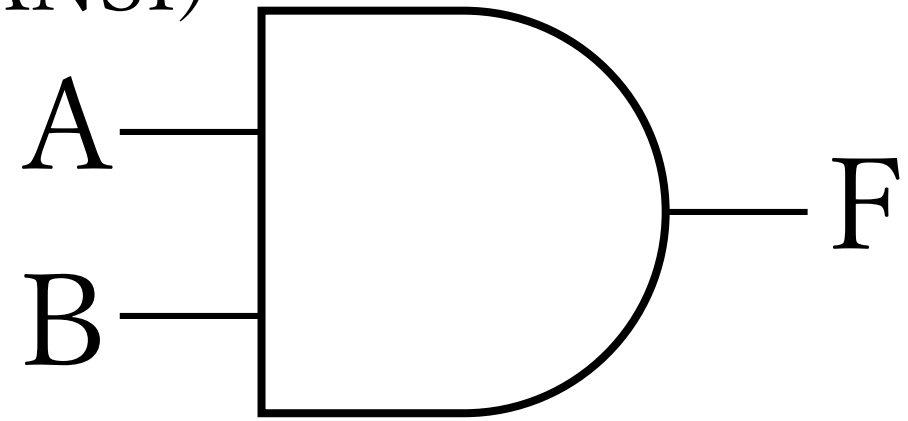


入力		出力
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

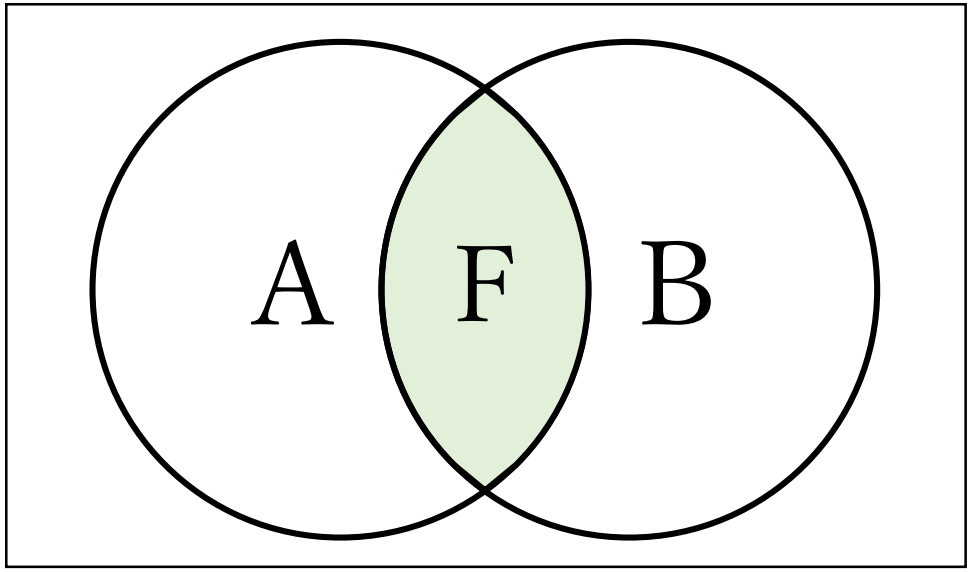
論理式

$$F = A \cdot B$$

回路記号
(ANSI)



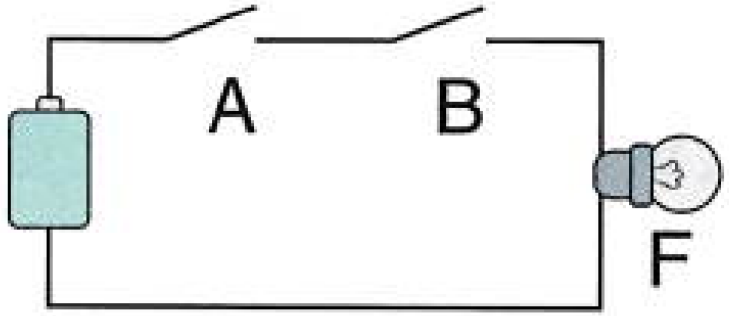
ベン図



AND回路
(論理積回路)

2つの入力が
共に1の時だけ
出力が1になる

真理値表↓

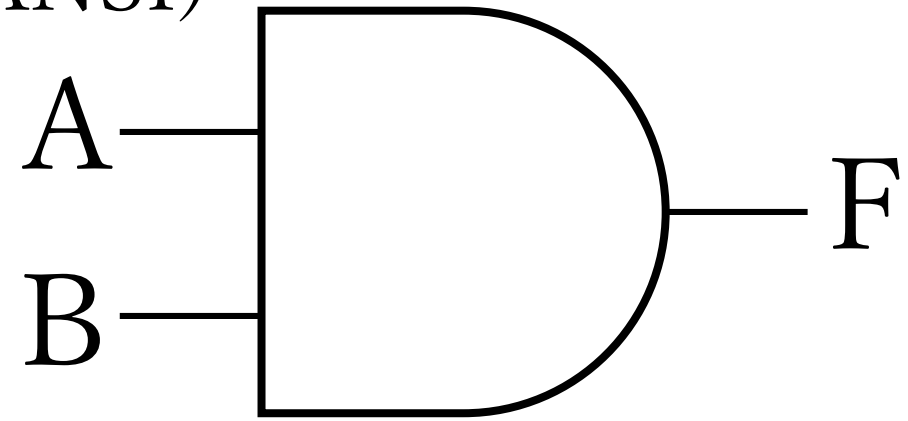


入力		出力
A	B	F
×	×	×
×	○	×
○	×	×
○	○	○

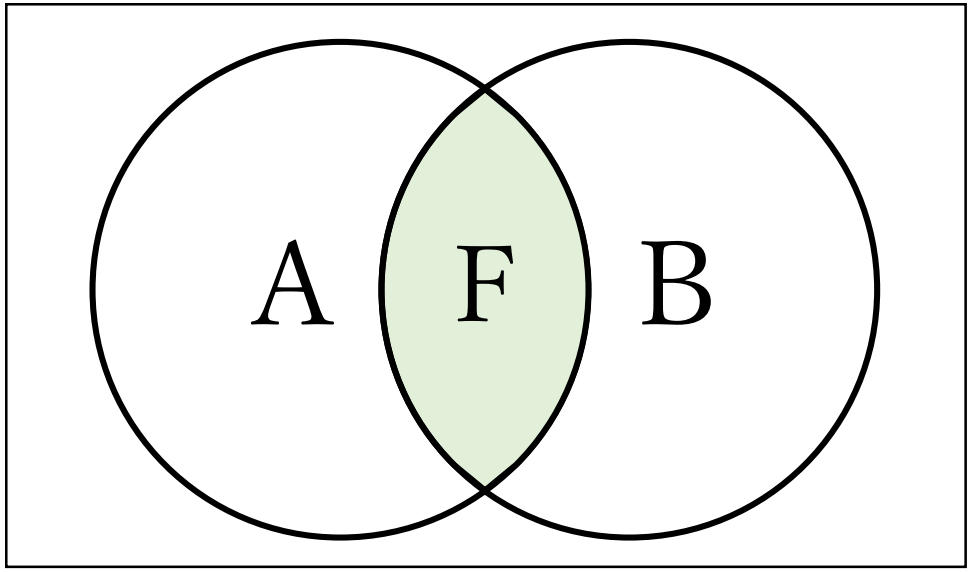
論理式

$$F = A \cdot B$$

回路記号
(ANSI)



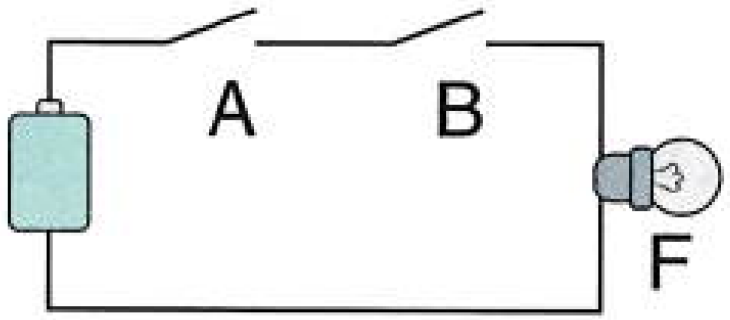
ベン図



AND回路
(論理積回路)

2つの入力が
共に1の時だけ
出力が1になる

真理値表↓

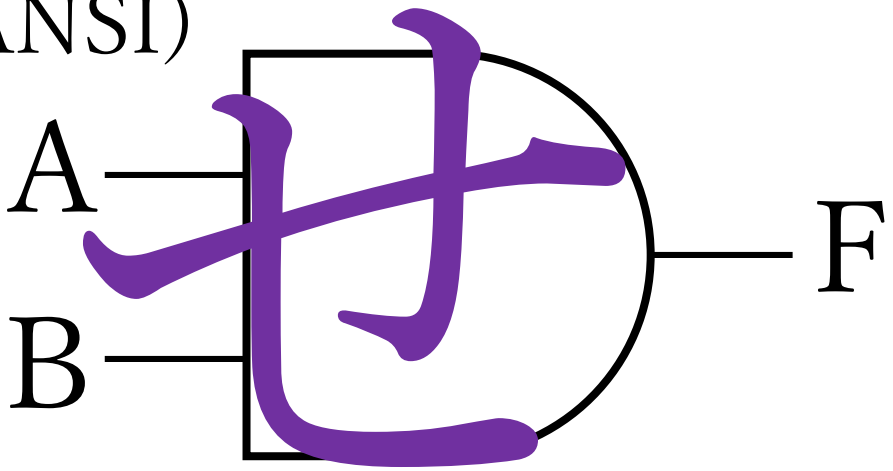


入力		出力
A	B	F
×	×	×
×	き	×
せ	×	×
せ	き	○

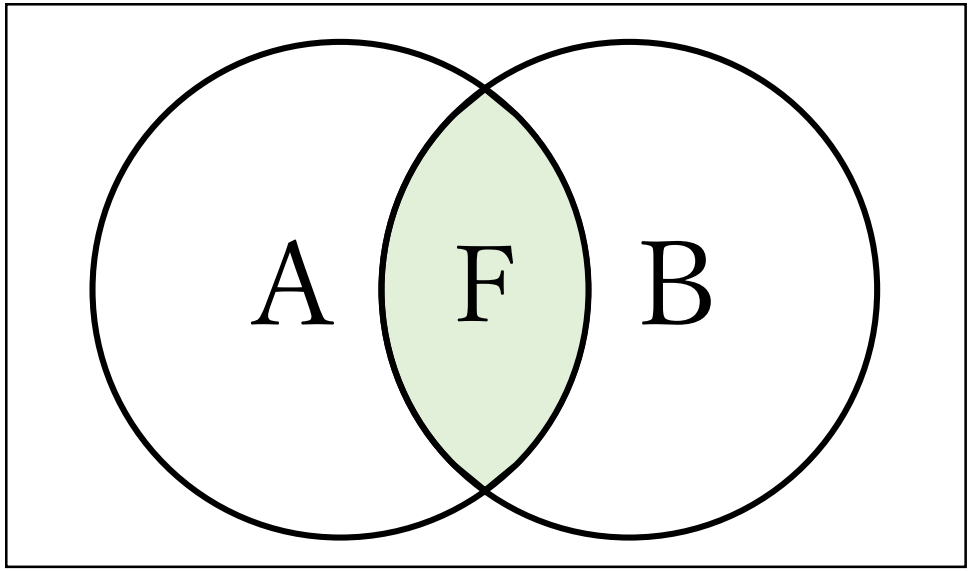
論理式

$$F = A \cdot B$$

回路記号
(ANSI)



ベン図



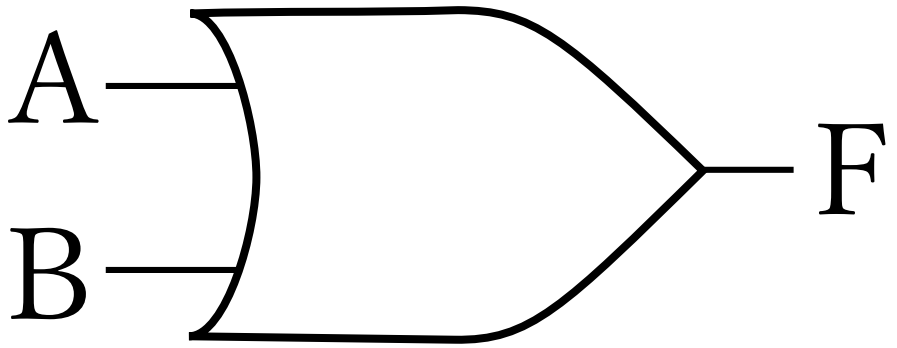
OR回路
(論理和回路)

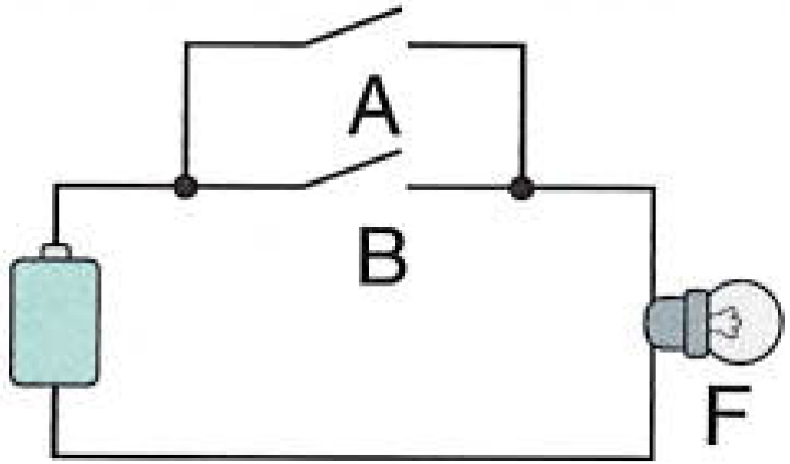
2つの入力の
一方が1であれば
出力が1になる

真理値表↓

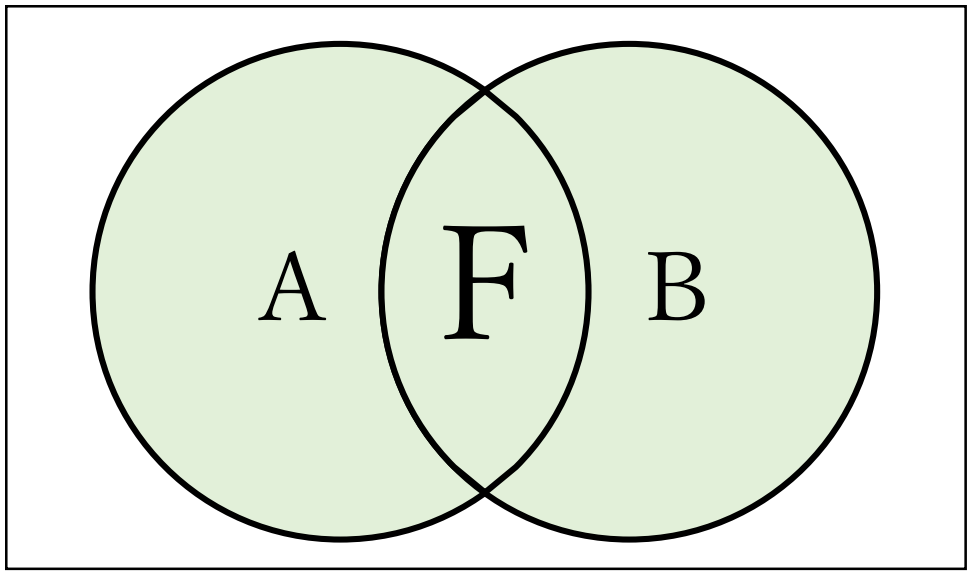
入力		出力
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1 !?

回路記号
(ANSI)





ベン図



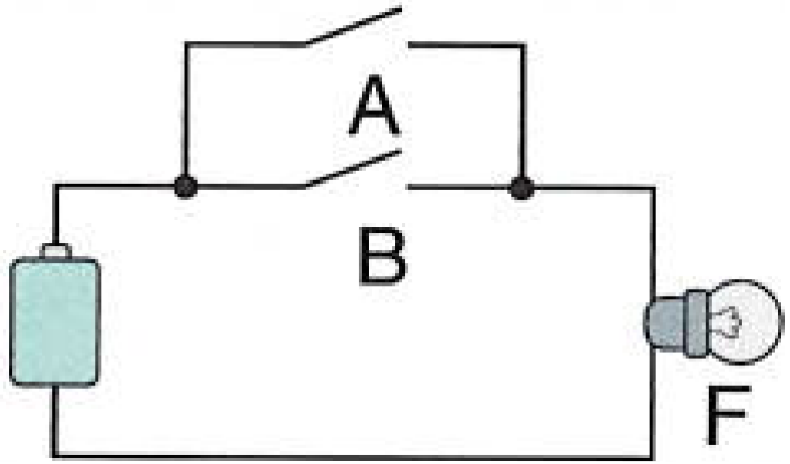
論理式

$F = A + B$

OR回路
(論理和回路)

2つの入力の
一方が1であれば
出力が1になる

真理値表↓

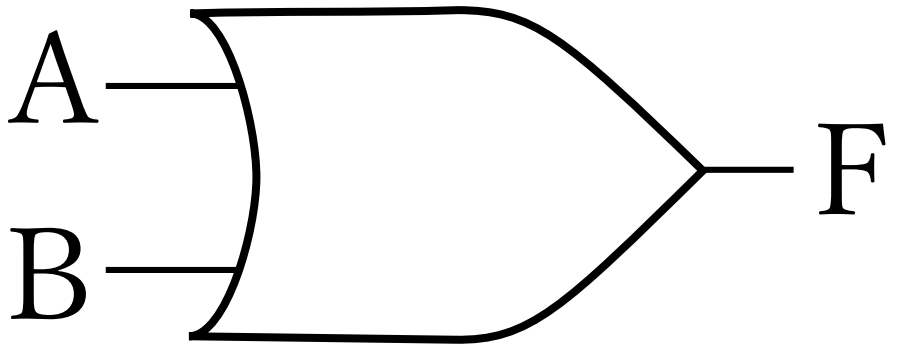


入力		出力
A	B	F
×	×	×
×	○	○
○	×	○
○	○	○

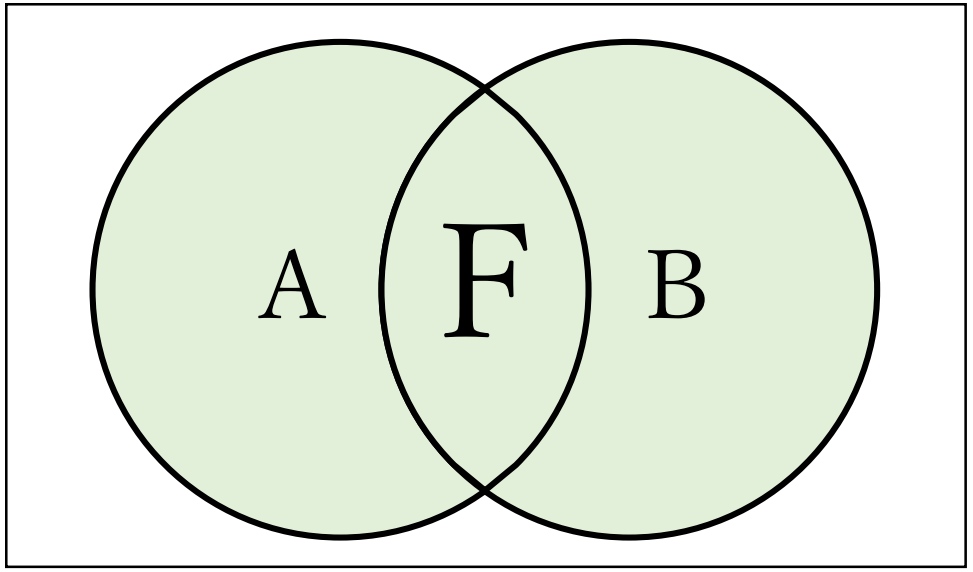
論理式

$$F = A + B$$

回路記号
(ANSI)



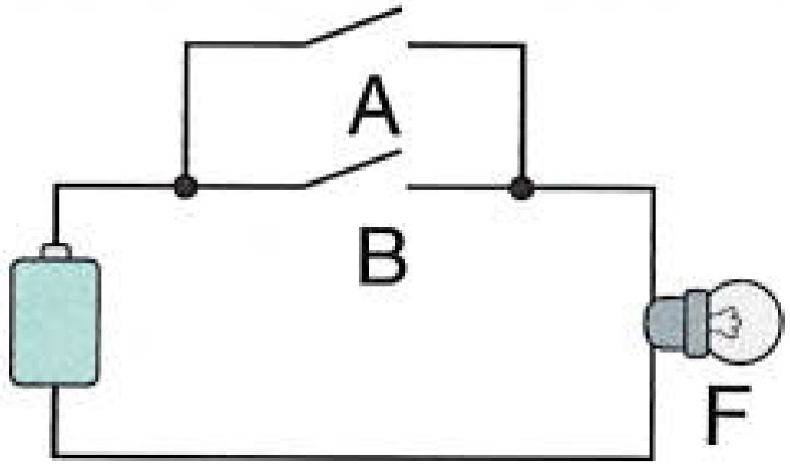
ベン図



OR回路
(論理和回路)

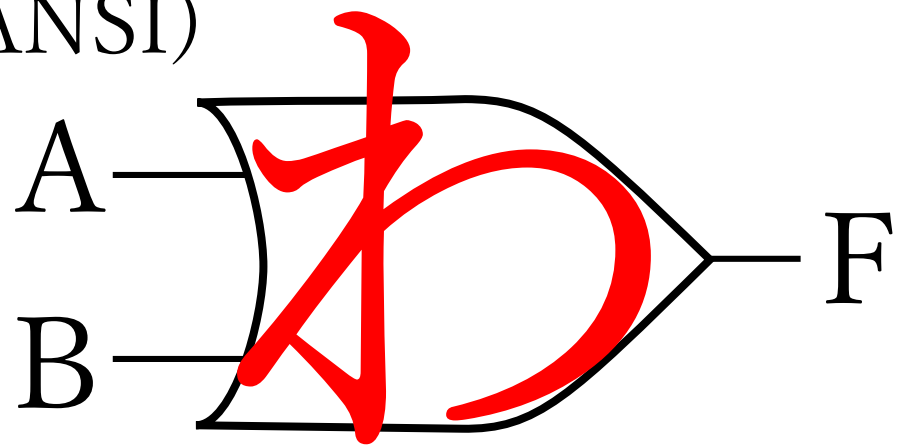
2つの入力の
一方が1であれば
出力が1になる

真理値表↓

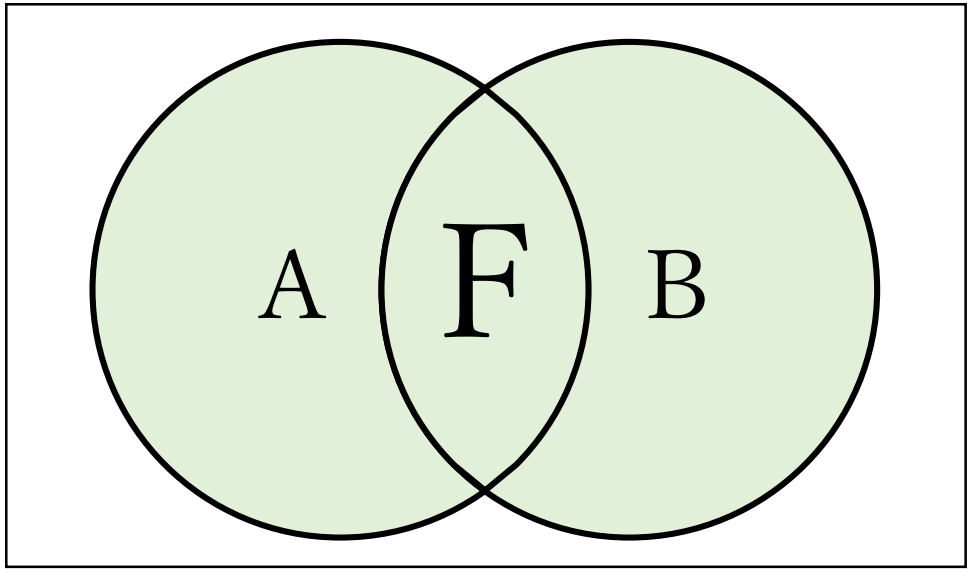


入力		出力
A	B	F
×	×	×
×	わ	○
わ	×	○
わ	わ	○

回路記号
(ANSI)



ベン図



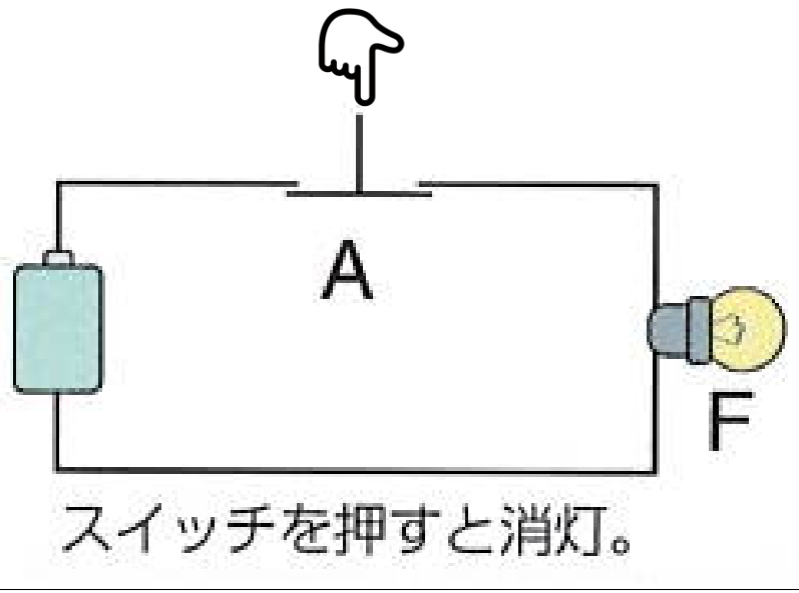
論理式

$$F = A + B$$

NOT回路
(否定回路)

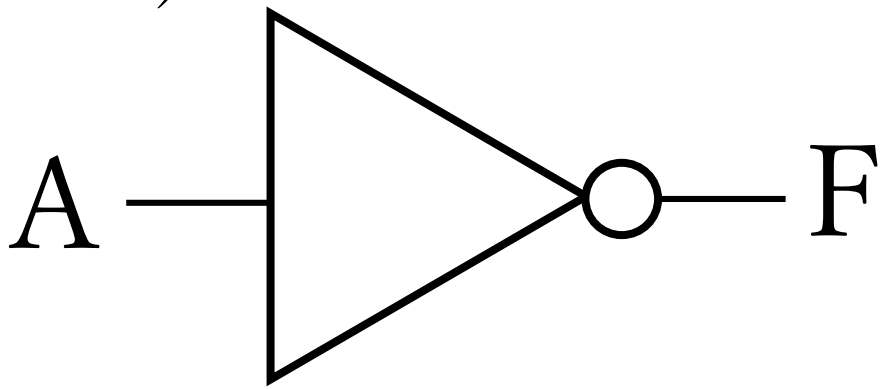
1つの
入力した信号を
反転した値を出力

真理値表↓

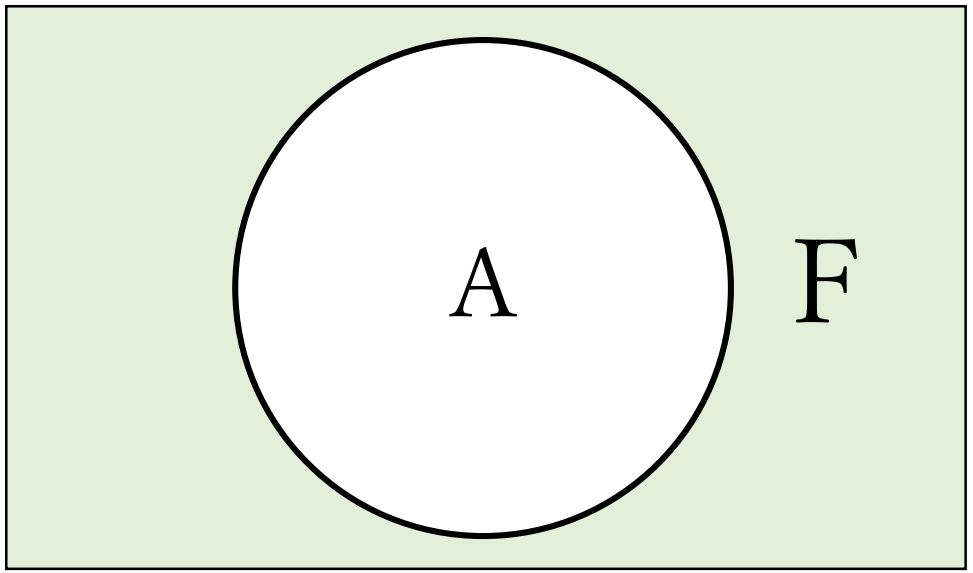


入力	出力
A	F
0 →	1
1 →	0

回路記号
(ANSI)



ベン図



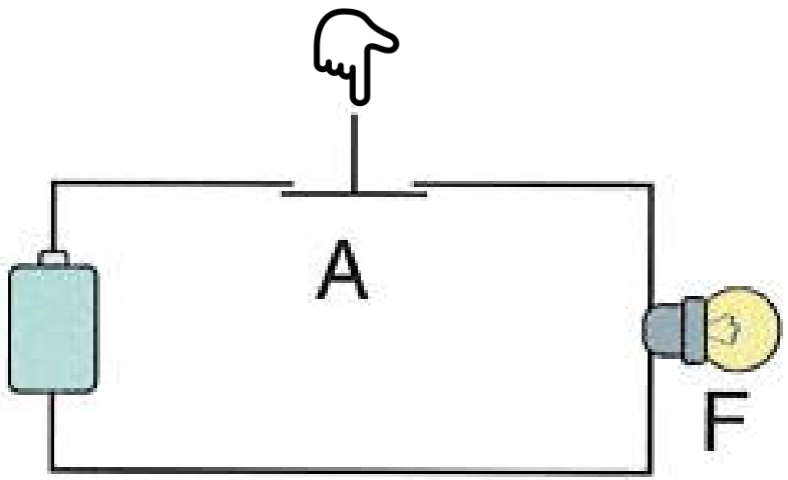
論理式

$$F = \overline{A}$$

NOT回路
(否定回路)

1つの
入力した信号を
反転した値を出力

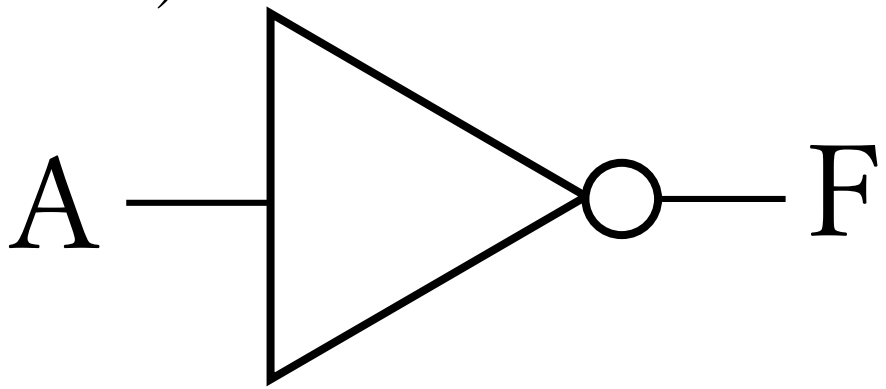
真理値表↓



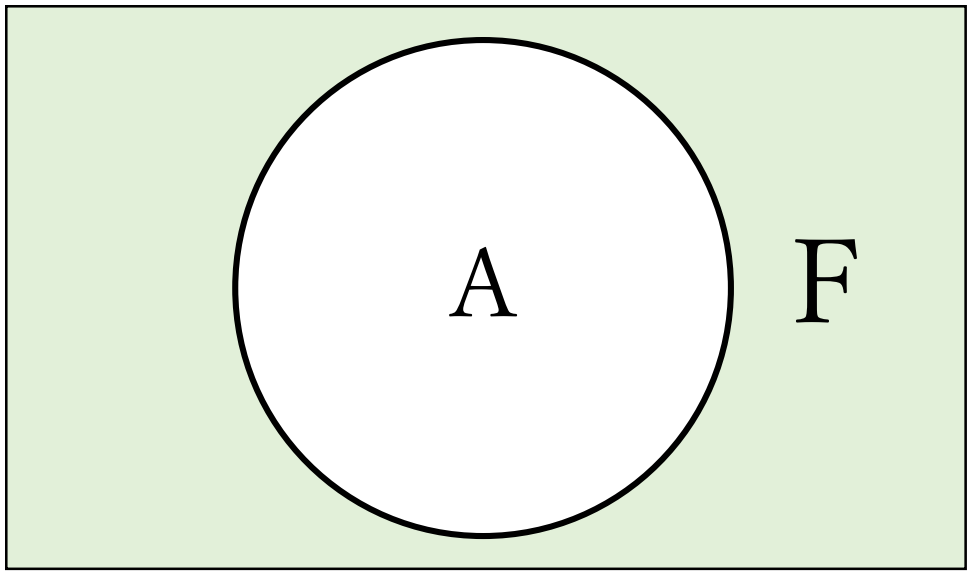
スイッチを押すと消灯。

入力	出力
A	F
× →	○
○ →	×

回路記号
(ANSI)



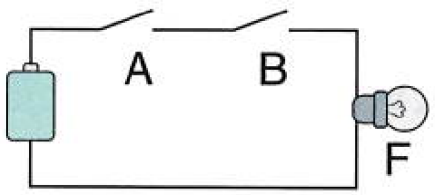
ベン図



論理式

$$F = \overline{A}$$

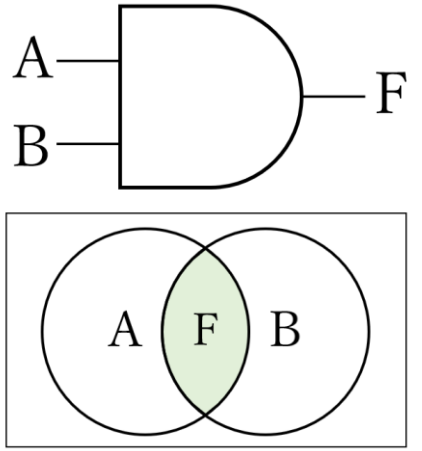
AND回路(論理積回路)



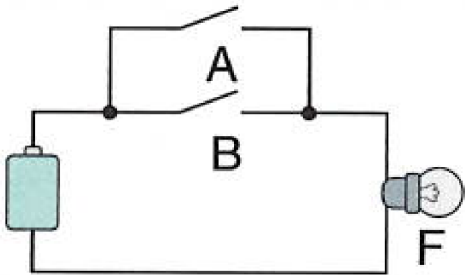
2つの入力が共に1の時だけ
出力が1になる

$F = A \cdot B$

入力		出力
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



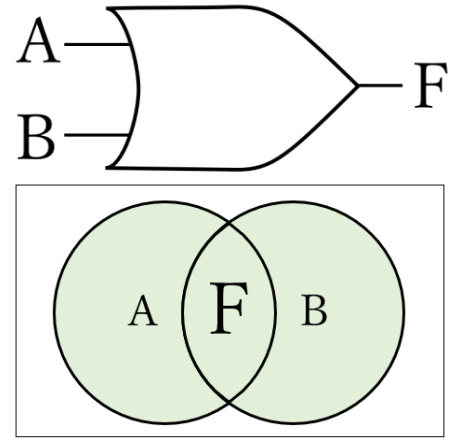
OR回路(論理和回路)



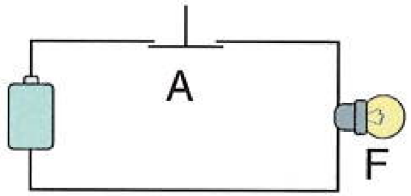
2つの入力の一方が1であれば
出力が1になる

$F = A + B$

入力		出力
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



NOT回路(否定回路)

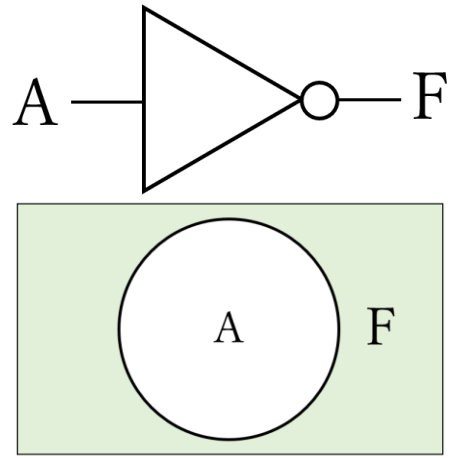


スイッチを押すと消灯。

1つの入力した信号を
反転した値を出力

$F = \overline{A}$

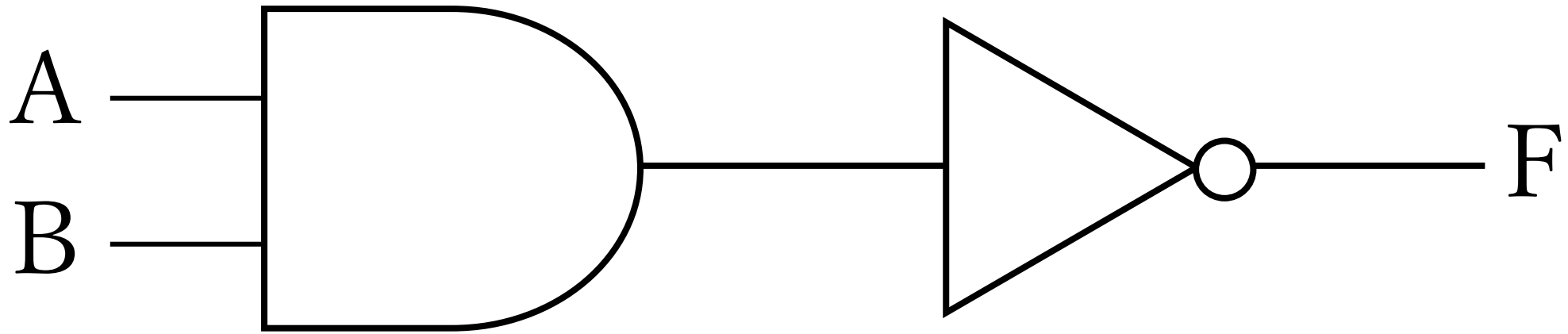
入力	出力
A	F
0	1
1	0



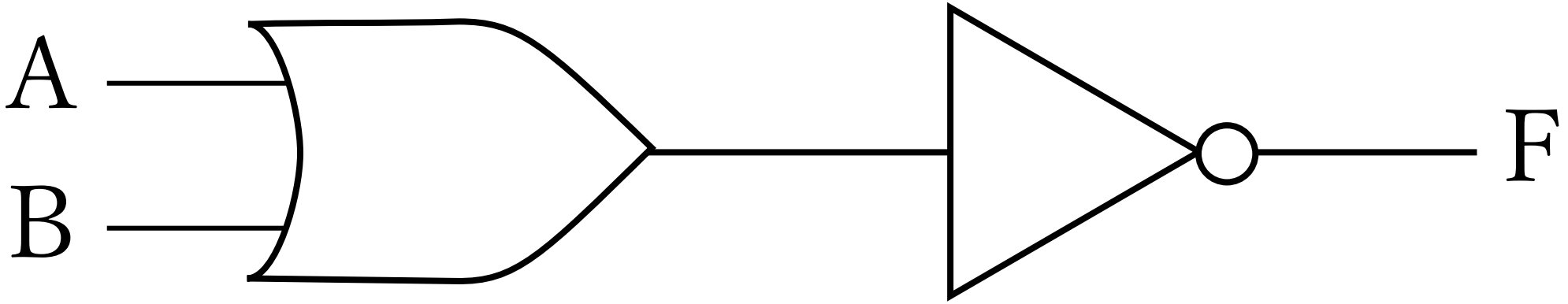
Q1

Aの入力が「0」、Bの入力が「1」のとき、
出力Fを答えなさい

①

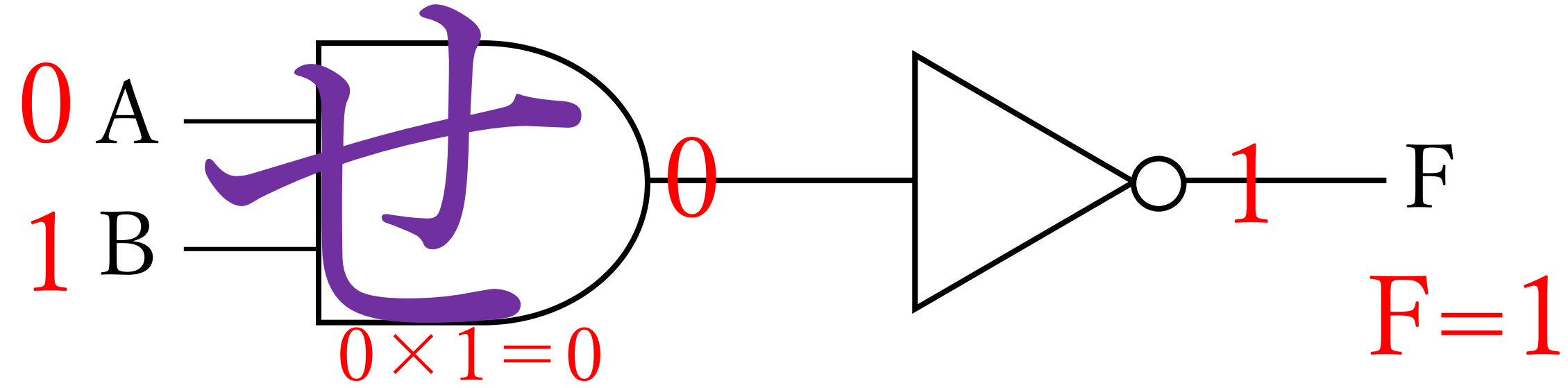


②



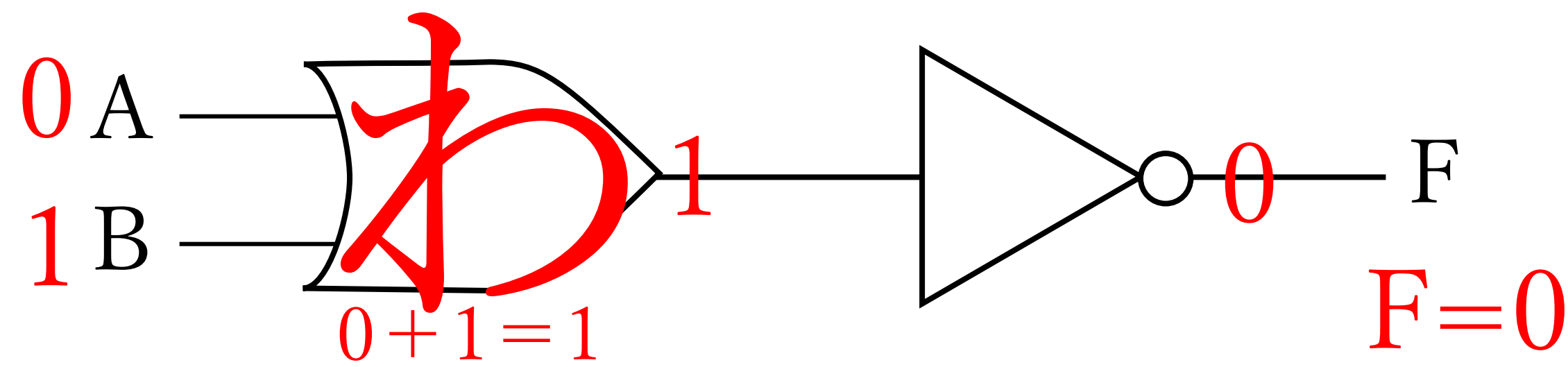
Aの入力が「0」、Bの入力が「1」のとき、
出力Fを答えなさい。

①



F＝1

②



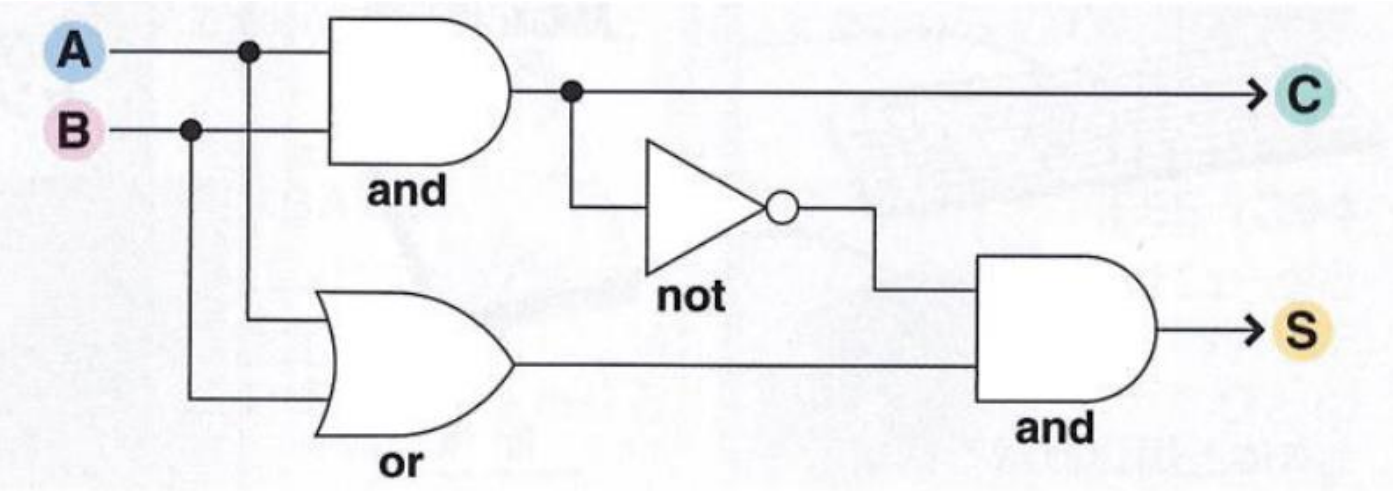
F＝0

＜応用＞
加算回路

- 半加算回路
- 全加算回路

半加算回路

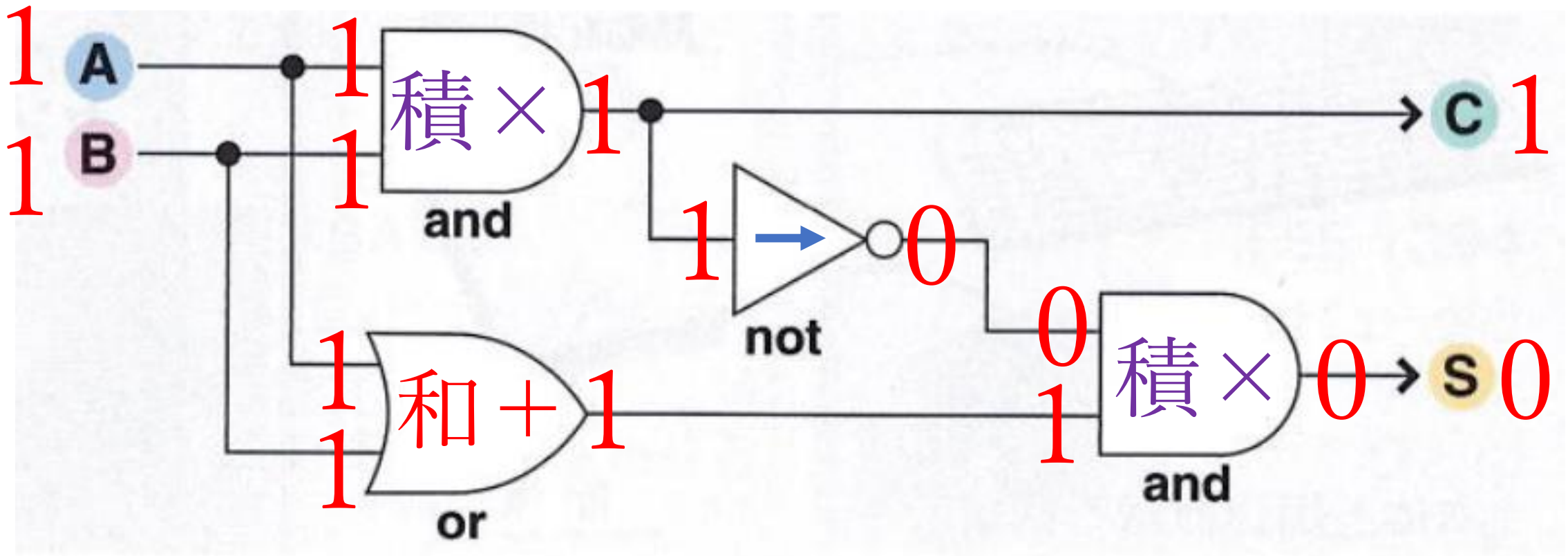
- AND回路、OR回路、NOT回路を組み合わせて、
  2進法の1桁の加算を実行する半加算回路がつくられている。
- 加算する2進法1桁の数値をA、Bとして、
  和(Sum)をS、桁上げ(Carry)をCとすれば、
  A、BとC、Sの関係はこのようになる



半加算回路の真理値表

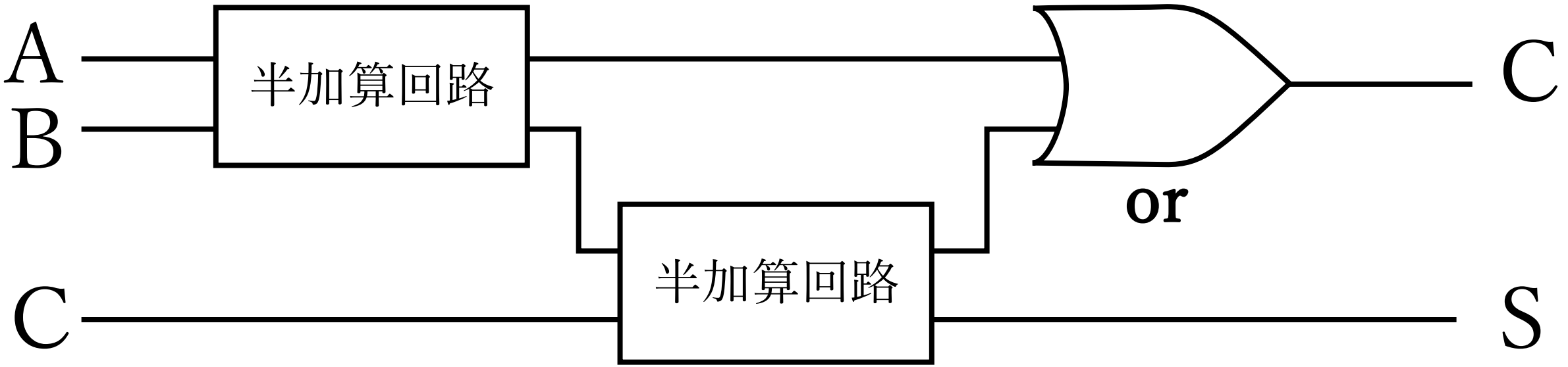
		A	B	C	S
0＋0＝	0→	0	0	0	0
0＋1＝	1→	0	1	0	1
1＋0＝	1→	1	0	0	1
1＋1＝1	0→	1	1	1	0

半加算回路　例:(A,B)=(1,1)



全加算回路

- 半加算回路は1桁の加算しかできないが、これを組み合わせて桁上げできるようにした回路を全加算回路という。
- コンピュータの中ではこれらを組み合わせ、桁数の多い演算もできるようにしている。

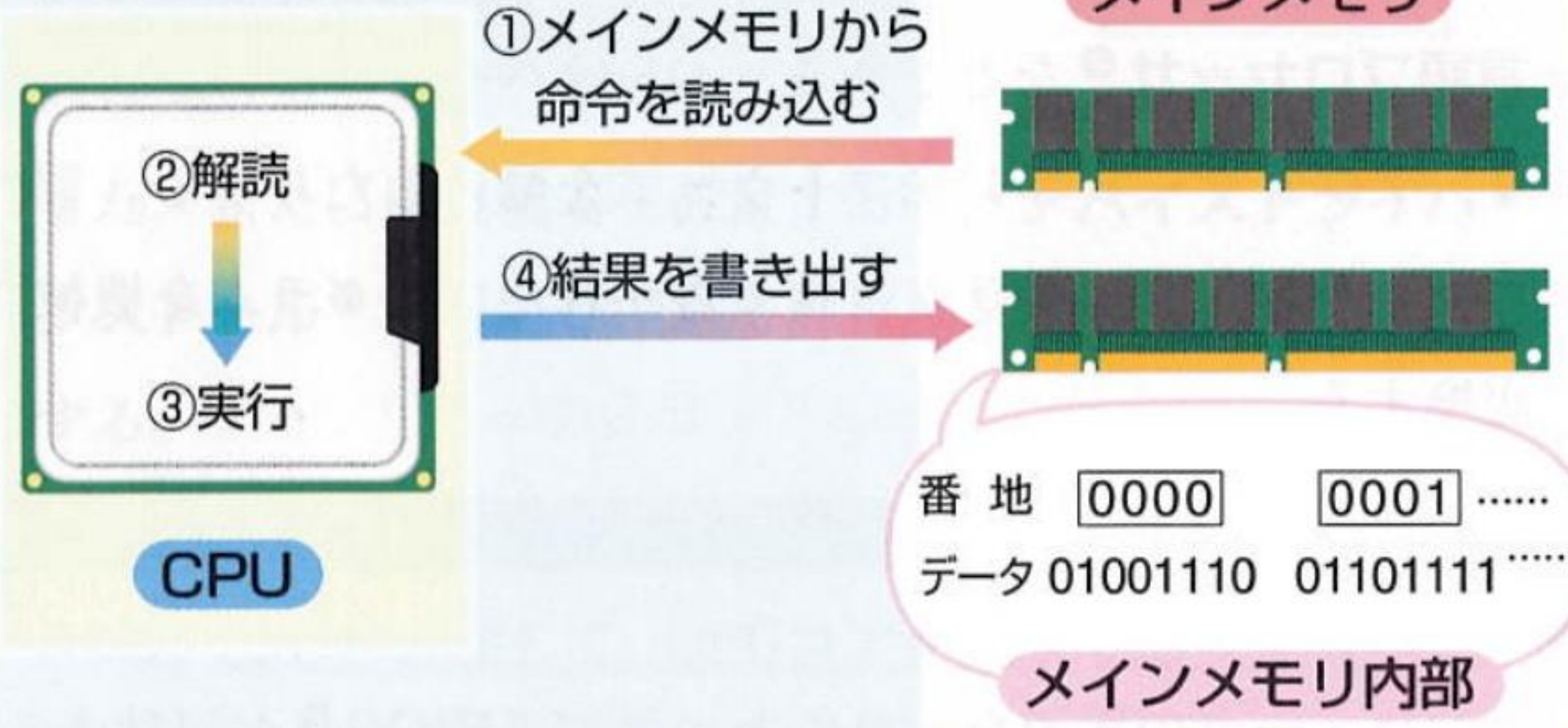


CPUとメインメモリの動作

- コンピュータが扱う命令やデータは、8ビットつまり1バイトをひとまとまりとしてメインメモリに記憶される。
- メインメモリの内部は1バイトごとに区分けされ、0からはじまる「アドレス(番地)」がつけられている。
- CPUは、次の4つの動作を繰り返して命令を処理する

1. メインメモリのアドレスを指定して命令を読み込む。
2. 読み込んだ命令を解読する。
3. 解読された命令を実行する。
4. 処理した結果をメインメモリなどに書き出す。

CPUとメインメモリ


メインメモリ
①メインメモリから
命令を読み込む
②解読
③実行
④結果を書き出す
CPU
番 地 0000 0001 ……
データ 01001110 01101111 ……
メインメモリ内部

CPU内部では

- データの受け渡しのタイミングを合わせるために、
  CPUクロックという信号が使われている。
  これが1秒間に発生するクロック信号の数を
  クロック周波数と呼び、単位はHzであらわされる。
- これまでクロック周波数を高めることで、CPU性能を高めてきた。
  しかし、クロック周波数が高まると消費電力も増え、熱も多く発生する。
  近年ではマルチコアCPUや64ビットCPUなど、
  クロック周波数を高めずに性能を高めたCPUが登場した。

＜本時のまとめ＞

- AND回路…2つの入力が共に1の時だけ出力が1になる
- OR回路…2つの入力の一方が1であれば出力が1になる
- NOT回路…1つの入力した信号を反転した値を出力

- CPUとメインメモリの動作
  →命令やデータは1バイト単位でメインメモリで処理される
  ※メインメモリには「アドレス」が振られている

  1. メインメモリのアドレスを指定して命令を読み込む
  2. 解読する
  3. 命令を実行する
  4. 結果をメインメモリなどに書き出す

＜課題＞

- 自分で論理回路を組み合わせて
  問題を作り、
  真理値表に結果をまとめること。