

2026 年度 論理回路

第 3 回講義資料

情報コース 小野寺 優希也

2026 年 10 月 15 日

【本日の内容】

- 整数・小数をコンピュータのメモリ上でデジタル表現する方法を説明できる。
- 基本的な論理演算を行うことができる。
- 基本的な論理演算を組合わせて、論理関数を論理式として表現できる。

§1

絶対値表現と補数表現

コンピュータでは、整数を 0 と 1 からなるビット列としてメモリ上に表現する。正の整数は通常の 2 進数を用いて表すことができるが、負の整数を表すためには、符号をどのようにビット列として表現するかを定める必要がある。本節では、負の整数を表現する代表的な方法として、絶対値表現と補数表現について説明する。

1.1

絶対値表現

絶対値表現は最上位ビットで符号を表現する。以下では 8 ビットを例に説明をする。また、 $(5)_{10} = (101)_2$ を例にした絶対値表現を図 1.1 に示す。図 1.1 にもあるように最上位のビットが 0 であれば正の値を表し、最上位ビットが 1 であれば負の値を表す。また、絶対値に相当する数値部分*1は 7 ビットで表現する。

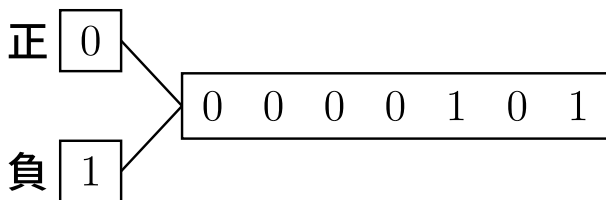


図 1.1 $(5)_{10} = (101)_2$ を例にした絶対値表現

ただし、この表現方法では、表 1.1 のように 0 を表す符号が 2 通り存在してしまうため、実際に使用されることは少ない。

*1 ± 5 であるところの 5 の部分のこと

表 1.1 8 ビットの絶対値表現

10 進数	絶対値表現
-127	11111111
-126	11111110
⋮	⋮
0	10000000 00000000
1	00000001
⋮	⋮
127	01111111

1.2 補数表現

n 桁に固定して表された整数 A の基数 r に対する補数 A' は

$$A' := r^n - A \quad (1.1)$$

で定義される^{*2} ^{*3}. また,

$$A'' := r^n - 1 - A \quad (1.2)$$

を A の $r - 1$ に対する補数, もしくは A の $r - 1$ の補数という. 例えば, $(73)_{10}$ を考えたとき, $(73)_{10}$ は 2 桁であることに注意すると, $(73)_{10}$ の 10 の補数 A' は

$$A' = 10^2 - 73 = 100 - 73 = (27)_{10}$$

となる. また, 同様の議論から, $(73)_{10}$ の 9 の補数 A'' は $(26)_{10}$ であることがわかる.

以下では, 2 進数の場合について, 議論する. すなわち, 式 (1.1) および式 (1.2) における r は $r = 2$ とする. なお, 補数を考える際には, 2 進数を表すビット数を固定する必要がある.

■ **1 の補数** $(7)_{10}$ を 4 ビットの 2 進数で表した $(0111)_2$ を例に考える. 式 (1.2) より, $n = 4$ であるから, $(0111)_2$ の 1 の補数 A'' は

$$A'' = 2^4 - 1 - (0111)_2 = (1111)_2 - (0111)_2 = (1000)_2$$

となる. この結果から, $(0111)_2$ の 0 と 1 を反転したものが 1 の補数となっていることがわかる. したがって, 1 の補数を求める際には, 与えられた 2 進数の全ビットの 0 と 1 を反転させばよい.

■ **2 の補数** ここでも, $(7)_{10}$ を 4 ビットの 2 進数で表した $(0111)_2$ を例に考える. 式 (1.1) と式 (1.2) の違いは 1 を引いているかいないか, すなわち,

$$A' = A'' + 1 \quad (1.3)$$

であった. そのため, 2 の補数を求める場合には, 1 の補数を求め, その結果に 1 を加えればよい. $(0111)_2$ の場合, 1 の補数は $(1000)_2$ であるから, $(0001)_2$ を加えると, $(0111)_2$ の 2 の補数として $(1001)_2$ が得られる. 以上から, 2 の補数は

*2 A を B で定義するとき, $A := B$ と記載することがある.

*3 整数 A の基数 r に対する補数を単に r の補数ということもある

1. 与えられた 2 進数の全ビットの 0 と 1 を反転させる.
2. 手順 1 の結果に 1 を加える.

という手順で求めることができる*4.

■負の数の表現 コンピュータ内部では, 0 と 1 を用いて数表現するため, 負の数表現するためには何らかの規則を定める必要がある. 実際には, 負の整数を表現する方法として, 2 の補数が広く用いられている. すなわち, n ビットで表された正の整数 A に対して, A の 2 の補数を A' とすると,

$$-A := A' \quad (1.4)$$

と表し, A' を $-A$ を表す 2 進数として用いる. このように, 2 の補数を用いて負の整数を表す方法を **2 の補数表現** という.

例題 1.1

$(-7)_{10}$ を 4 ビットの 2 の補数表現で表せ.

解答

$(7)_{10}$ を 4 ビットの 2 進数で表すと, $(0111)_2$ となる. $(0111)_2$ の 2 の補数は $(1001)_2$ であるから,

$$(-7)_{10} = (1001)_2$$

となる. □

また, 先ほどの例からもわかるように, n ビットの 2 の補数表現では, 最上位ビットが 0 の場合は 0 以上の整数を, 最上位ビットが 1 の場合は負の整数を表す.

例題 1.2

4 ビットの 2 の補数表現 $(1001)_2$ が表す 10 進数を求めよ.

解答

$(1001)_2$ の最上位ビットは 1 であるから, 負の整数を表している. $(1001)_2$ の 2 の補数を求めると, $(0111)_2$ となる.

$$(0111)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_{10}$$

であるから,

$$(1001)_2 = (-7)_{10}$$

となる. □

■補数を用いる利点 1 の補数表現では, 0 を表すビット列が 2 種類存在する*5ため, 現在の一般的なコンピュータではほとんど用いられていない. 一方, 2 の補数表現では, 引き算を足し算として

*4 1 の補数と 2 の補数を求める手順を簡単に導出したが, この導出を試験問題とすることはない.

*5 $(+0)_{10} = (0000)_2$ とする. これに対して, $(0000)_2$ の 1 の補数を求めると $(1111)_2$ となるため, 1 の補数表現では $(-0)_{10} = (1111)_2$ と表される.

処理することができる。例えば、 $(5)_{10} - (3)_{10}$ を 4 ビットの 2 進数を用いて計算する場合を考える。

$$(5)_{10} = (0101)_2, \quad (3)_{10} = (0011)_2$$

である。 $(0011)_2$ の 2 の補数は $(1101)_2$ であるから、

$$(-3)_{10} = (1101)_2$$

と表すことができる。したがって、

$$\begin{aligned} (5)_{10} - (3)_{10} &= (0101)_2 + (1101)_2 \\ &= (10010)_2 \end{aligned}$$

となる。4 ビットを超えた最上位の 1 を無視すると、

$$(0010)_2 = (2)_{10}$$

となり、 $5 - 3 = 2$ が得られる。

このように、2 の補数を用いることで、引き算を足し算として処理することができる。

問題 1.3

次の問いに答えよ。

- (1) $(-13)_{10}$ を 8 ビットの 2 の補数表現で表せ。
- (2) 8 ビットの 2 の補数表現 $(11101010)_2$ を 10 進数に変換せよ。
- (3) 8 ビットの 2 の補数表現 $(11111111)_2$ は 10 進数で何を表すか。

問題 1.3

次の問いに答えよ。

- (1) $(-13)_{10}$ を 8 ビットの 2 の補数表現で表せ。
- (2) 8 ビットの 2 の補数表現 $(11101010)_2$ を 10 進数に変換せよ。
- (3) 8 ビットの 2 の補数表現 $(11111111)_2$ は 10 進数で何を表すか。

解答

- (1) $(13)_{10}$ を 8 ビットの 2 進数で表すと、

$$(13)_{10} = (00001101)_2$$

となる。全ビットの 0 と 1 を反転すると、

$$(11110010)_2$$

となり、さらに 1 を加えると、

$$(11110010)_2 + (00000001)_2 = (11110011)_2$$

となる。したがって、

$$(-13)_{10} = (11110011)_2$$

となる。

- (2) $(11101010)_2$ は最上位ビットが 1 であるから、負の整数を表している。 $(11101010)_2$ の 2 の補数を求めると、

$$(11101010)_2 \longrightarrow (00010101)_2 \longrightarrow (00010110)_2$$

となる。また、

$$(00010110)_2 = 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 = (22)_{10}$$

であるから、

$$(11101010)_2 = (-22)_{10}$$

となる。

- (3) $(11111111)_2$ は最上位ビットが 1 であるから、負の整数を表している。 $(11111111)_2$ の 2 の補数を求めると、

$$(11111111)_2 \longrightarrow (00000000)_2 \longrightarrow (00000001)_2$$

となる。したがって、

$$(11111111)_2 = (-1)_{10}$$

となる。

□

問題 1.4 出典：平成 25 年度応用情報技術者試験（AP） 秋期分 午前 問 3

負の整数を表現する代表的な方法として、次の 3 種類がある。

- a.1 の補数による表現
- b.2 の補数による表現
- c.絶対値に符号を付けた表現 (左端ビットが 0 の場合は正、1 の場合は負)

4 ビットのパターン 1101 を a～c の方法で表現したものと解釈したとき、値が小さい順になるように三つの方法を並べたものはどれか。

- (ア)a, c, b
- (イ)b, a, c
- (ウ)b, c, a
- (エ)c, b, a

問題 1.4 出典：平成 25 年度応用情報技術者試験（AP） 秋期分 午前 問 3

負の整数を表現する代表的な方法として、次の 3 種類がある。

- a.1 の補数による表現
- b.2 の補数による表現
- c.絶対値に符号を付けた表現 (左端ビットが 0 の場合は正、1 の場合は負)

4 ビットのパターン 1101 を a～c の方法で表現したものと解釈したとき、値が小さい順になるように三つの方法を並べたものはどれか。

- (ア)a, c, b
- (イ)b, a, c
- (ウ)b, c, a
- (エ)c, b, a

解答

a の 1 の補数による表現では、 $(1101)_2$ の全ビットの 0 と 1 を反転すると、

$$(1101)_2 \rightarrow (0010)_2$$

となる。 $(0010)_2 = (2)_{10}$ であるから、a で表される値は $(-2)_{10}$ となる。

b の 2 の補数による表現では、 $(1101)_2$ の全ビットの 0 と 1 を反転し、その結果に 1 を加えると、

$$(1101)_2 \rightarrow (0010)_2 \rightarrow (0011)_2$$

となる。 $(0011)_2 = (3)_{10}$ であるから、b で表される値は $(-3)_{10}$ となる。

c の絶対値に符号を付けた表現では、左端の 1 が負を表し、残りの 3 ビット $(101)_2$ が絶対値を表す。

$$(101)_2 = 1 \times 2^2 + 1 \times 2^0 = (5)_{10}$$

であるから、c で表される値は $(-5)_{10}$ となる。

以上から、

$$-5 < -3 < -2$$

であるから、値が小さい順に c, b, a となる。したがって、答えは (エ) である。 □

§2 浮動小数点数の表現

コンピュータでは、小数も 0 と 1 からなる有限長のビット列としてメモリ上に表現する必要がある。本節では、小数を表現する代表的な方法である浮動小数点形式について説明し、正規化、

IEEE 754 の倍精度浮動小数点数, および丸め誤差について扱う.

2.1 浮動小数点形式と正規化

例えば, $(12345.6)_{10}$ について, 各桁の値に着目すると,

$$12345.6 = 1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} \quad (2.1)$$

$$= 10^4 (1 + 2 \times 10^{-1} + 3 \times 10^{-2} + 4 \times 10^{-3} + 5 \times 10^{-4} + 6 \times 10^{-5}) \quad (2.2)$$

$$= 1.23456 \times 10^4 \quad (2.3)$$

と表すことができる. ここでは, 最上位の桁が 10^4 の位であるため, 10^4 でくくることによって, 小数点の左側が 1 桁となるように表している. 1.23456 の部分を仮数, 10^4 の 4 を指数という. このような表記を浮動小数点形式という. 2 進数についても同様に考えることができる. 例えば,

$$(1101.101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \quad (2.4)$$

$$= 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-3} \quad (2.5)$$

$$= 8 + 4 + 1 + \frac{1}{2} + \frac{1}{8} \quad (2.6)$$

$$= (13.625)_{10} \quad (2.7)$$

となる. また, 最上位の桁が 2^3 の位であることから, 2^3 でくくると,

$$(1101.101)_2 = 2^3 (1 + 2^{-1} + 0 \times 2^{-2} + 2^{-3} + 2^{-4} + 0 \times 2^{-5} + 2^{-6}) \quad (2.8)$$

$$= (1.101101)_2 \times 2^3 \quad (2.9)$$

と表すことができる. このとき, $(1.101101)_2$ の部分を仮数, 2^3 の 3 を指数という.

2.2 IEEE 754 の倍精度浮動小数点数

浮動小数点形式では仮数部や指数部の長さ, 基数など多様な規格が考えられる. そのため, 1985 年に William Kahan^{*6}が中心となって IEEE754^{*7}が制定された. IEEE 754 の倍精度浮動小数点数では, 正規化された有限の非零の数 x を

$$x = (-1)^s 2^E (1.f_1 f_2 \cdots f_{52})_2 \quad (2.10)$$

の形で表す. ここで, $s = 0, 1$, $f_i = 0, 1$ であり, $i = 1, 2, \dots, 52$ である. また, E は整数である. 式 (2.10) における $1.f_1 f_2 \cdots f_{52}$ を仮数, E を指数という. 式 (2.10) の内部表示を図 2.1 に示す. 図 2.1 より, 符号に 1 bit, 指数部に 11 bit, 仮数部に 52 bit を用いる. 指数部の 11 bit を

$$(e_{10} e_9 \cdots e_0)_2$$

とすると, 正規化数について

$$(e_{10} e_9 \cdots e_0)_2 = E + 1023 \quad (2.11)$$

^{*6} William Kahan は, 浮動小数点演算をはじめとする数値解析への貢献により, 1989 年にチューリング賞を受賞しています. 詳しくは https://amturing.acm.org/award_winners/kahan_1023746.cfm を参照してください.

^{*7} あいとりぶるいい 754 と読みます. あいいい 754 と読まないでください.

と表される．2進数の小数点の位置を移動し，仮数の整数部分が1となるように

$$(1.f_1f_2\cdots f_{52})_2 \times 2^E$$

の形に表すことを**正規化**という．

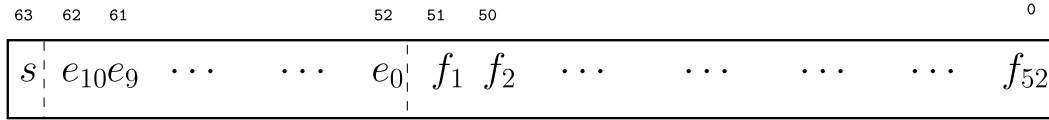


図 2.1 IEEE 754 (倍精度) の内部表示

例えば,

$$(13.625)_{10} = (1101.101)_2 \quad (2.12)$$

$$= (1.101101)_2 \times 2^3 \quad (2.13)$$

であるから, $s = 0$, $E = 3$ となる．また, 正規化された数では仮数の整数部分は常に1となるため, 先頭の1は格納しない．このように, 正規化によって常に1となる先頭の桁を省略して表現する方法を**ケチ表現**という．したがって, $(1.101101)_2$ については, 小数点以下の

$$10110100\cdots 0$$

を52 bitの仮数部として格納する．指数部には E をそのまま格納するのではなく, 1023を加えた値を格納する．このように, 一定の値を加えて指数を表現する方法を**バイアス表現**という．したがって,

$$E + 1023 = (1026)_{10} = (10000000010)_2 \quad (2.14)$$

となる．

2.3 丸め誤差

浮動小数点形式の場合, 仮数 $(1.f_1f_2\cdots)_2$ のうち, 記憶できる桁数には限りがある．例えば, 倍精度では f_{52} までが記憶され, それより下位の桁は丸められる．これによって生じる誤差を**丸め誤差**と呼ぶ．

丸め誤差が実際の計算結果にどのように現れるかを確認する．次のプログラムでは, 0.1 を10000回加算し, 単精度の float 型と倍精度の double 型で計算結果を比較する．

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     /* 変数宣言 */
6     float x = 0.1;
7     float s0 = 0.0; /* 和の結果を格納(float) */
8     double x1 = 0.1;
9     double s1 = 0.0; /* 和の結果を格納(double) */
10    int i; /* ループカウンタ */
11    int n = 10000; /* 繰り返し回数の上限 */
```

```

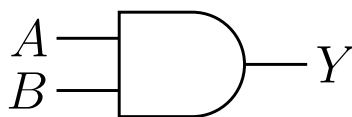
12
13     /* loop(float) */
14     for(i = 1 ; i <= n; i++)
15     {
16         s0 = s0 + x;
17     }
18
19     /* loop(double) */
20     for(i = 1 ; i <= n; i++)
21     {
22         s1 = s1 + x1;
23     }
24
25     /* 結果の出力 */
26     printf("float   : %.9f\n", s0);
27     printf("double  : %.17f\n", s1);
28     return 0;
29 }

```

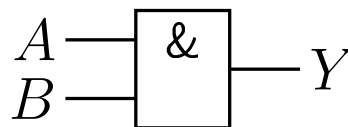
§3 基本的な論理回路

コンピュータでは、0 と 1 の二つの値に対して論理演算を行うことで、さまざまな処理を実現している。本節では、基本的な論理演算と、それらを実現する論理回路について説明する。

論理演算を行うことができる素子を**論理演算素子**という。論理演算素子を表す記号は複数存在するが、本講義では、**MIL 記号**^{*8}を用いる。例として、このあと説明する論理積の MIL 記号を図 3.1(a) に示す。また、MIL 記号以外の記号として、JIS 記号^{*9}がある。JIS 記号の論理積を図 3.1(b) に示す。その他の JIS 記号については例えば [1, 表 3・5] を参照すること。



(a) MIL 記号



(b) JIS 記号

図 3.1 論理積を表す MIL 記号と JIS 記号

以下では、基本論理演算と呼ばれる論理積、論理和、論理否定を扱ったあとに、排他的論理和、否定論理積、否定論理和について説明する。なお、これ以降、 A, B, Y は 0 もしくは 1 の値をとる 2 値変数とし、 A, B を入力、 Y を出力とする。

■**論理積 (AND)** 論理積の MIL 記号を図 3.2 に、真理値表を表 3.1 に示す。

*8 Military Specification Standards Symbol. MIL 規格ということもある。米国軍用規格に用いられる記号である。

*9 日本産業規格による記号。直線で書くことができるため、機械による作図に適している。

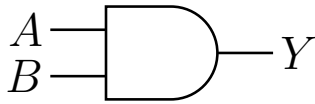


図 3.2 論理積の MIL 記号

表 3.1 論理積の真理値表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

表 3.1 のように、入力のすべての組合せに対する出力を 1 もしくは 0 で表した一覧表を**真理値表**という。

■**論理和 (OR)** 論理和の MIL 記号を図 3.3 に、真理値表を表 3.2 に示す。

表 3.2 論理和の真理値表

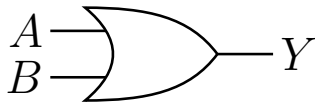


図 3.3 論理和の MIL 記号

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

■**論理否定 (NOT)** 論理否定の MIL 記号を図 3.4 に、真理値表を表 3.3 に示す。

表 3.3 論理否定の真理値表

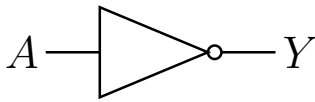


図 3.4 論理否定の MIL 記号

A	Y
0	1
1	0

なお、任意の論理関数は、論理積 (AND)、論理和 (OR)、論理否定 (NOT) の組合せによって表現することができる。この証明のアイディアについては、例えば、Stanford University, *Foundations of Computer Science*, Chapter 12 “Propositional Logic” (<https://cs.stanford.edu/~ullman/focs/ch12.pdf>) を参照すること。

■**排他的論理和 (XOR)** 排他的論理和の MIL 記号を図 3.5 に、真理値表を表 3.4 に示す。

表 3.4 排他的論理和の真理値表

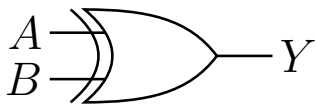


図 3.5 排他的論理和の MIL 記号

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

表 3.4 からわかるように、2つの入力値が同じ場合には 0 を、異なる場合には 1 を出力する。なお、教科書によっては、XOR の MIL 記号を図 3.6 のように表すこともある。今後、論理回路図を記述する問題を出題した場合には、どちらの記号を用いてもよい。また、XOR の MIL 記号を読み取る問題を出題する場合には、どちらの記号を用いているかがわかるように問題文中で明記する。

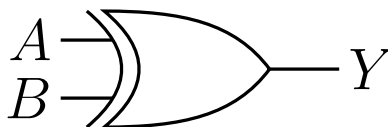


図 3.6 入力を外側の曲線に接続した排他的論理和の MIL 記号

■**否定論理積 (NAND)** 否定論理積 (NAND) は、論理積 (AND) と論理否定 (NOT) を組み合わせた論理演算である。まず、入力 A, B に対する AND の出力を Y とすると、その関係は表 3.5 のようになる。次に、 Y を NOT の入力とすると、その出力 Y_2 は表 3.6 のように Y の値を反転したものとなる。したがって、入力 A, B から出力 Y_2 までの関係をまとめると、表 3.7 に示す NAND の真理値表が得られる。

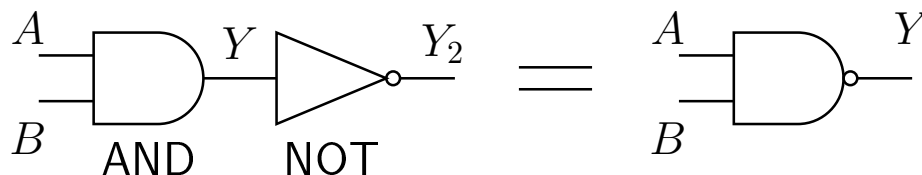


図 3.7

表 3.5 AND の真理値表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

表 3.6 NOT の真理値表

Y	Y_2
0	1
1	0

表 3.7 NAND の真理値表

A	B	Y_2
0	0	1
0	1	1
1	0	1
1	1	0

以上から、否定論理積 (NAND) は、論理積 (AND) の出力を論理否定 (NOT) した論理演算であることがわかる。最後に、NAND の MIL 記号を図 3.8 に、真理値表を表 3.8 に示す。

表 3.8 否定論理積の真理値表



図 3.8 否定論理積の MIL 記号

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

■**否定論理和 (NOR)** NAND の場合と同様に、OR と NOT を組み合わせることで、NOR の真理値表を求めることができる。NOR の MIL 記号を図 3.9 に、真理値表を表 3.9 に示す。

表 3.9 否定論理和の真理値表



図 3.9 否定論理和の MIL 記号

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

例題 3.1

次の論理回路について、真理値表の出力 Y を埋めなさい。

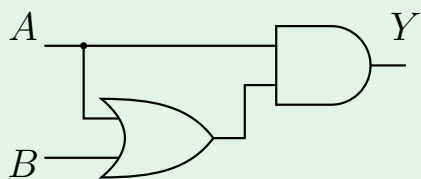


図 3.10

A	B	Y
0	0	
0	1	
1	0	
1	1	

解答

図 3.11 のように Y_1 を置く． Y_1, Y の値を順に求めると、真理値表は表 3.10 となる．

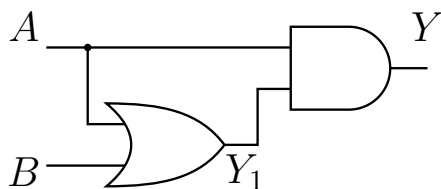


図 3.11

表 3.10

A	B	Y_1	Y
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

□

図 3.10 のように、論理回路図では、配線が接続されている箇所を黒丸で表すことがある。

問題

3.2

図 3.12 を参考に、真理値表の空欄を埋めなさい。

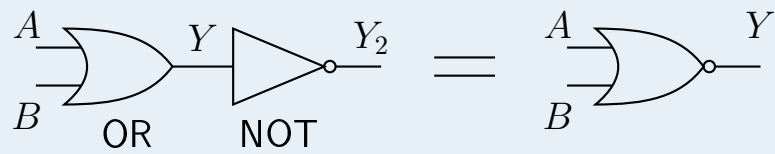


図 3.12

表 3.11 OR の真理値表

A	B	Y
0	0	
0	1	
1	0	
1	1	

表 3.12 NOT の真理値表

Y	Y ₂
0	
1	

表 3.13 NOR の真理値表

A	B	Y ₂
0	0	
0	1	
1	0	
1	1	

A	B	Y
0	0	
0	1	
1	0	
1	1	

Y	Y ₂
0	
1	

A	B	Y ₂
0	0	
0	1	
1	0	
1	1	

問題 3.2

図 3.12 を参考に、真理値表の空欄を埋めなさい。

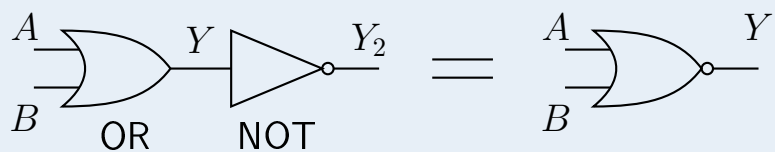


図 3.12

表 3.11 OR の真理値表

A	B	Y
0	0	
0	1	
1	0	
1	1	

表 3.12 NOT の真理値表

Y	Y ₂

表 3.13 NOR の真理値表

A	B	Y ₂
0	0	
0	1	
1	0	
1	1	

解答

表 3.14 OR の真理値表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

表 3.15 NOT の真理値表

Y	Y ₂
0	1
1	0

表 3.16 NOR の真理値表

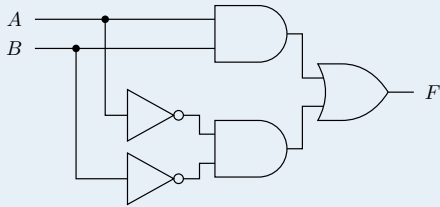
A	B	Y ₂
0	0	1
0	1	0
1	0	0
1	1	0

□

問題 3.3

出典：平成 19 年度ソフトウェア開発技術者試験 春期 午前 問 6

図に示す論理回路と等価な真理値表はどれか。ここで、は論理積、は論理和、は否定を表す。



(ア)

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

(イ)

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

(ウ)

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

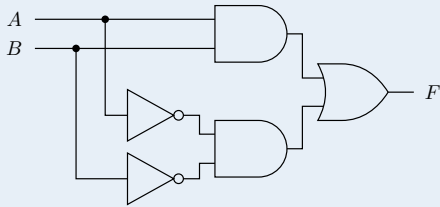
(エ)

A	B	F
0	0	1
0	1	1
1	0	0
1	1	0

問題 3.3

出典：平成 19 年度ソフトウェア開発技術者試験 春期 午前 問 6

図に示す論理回路と等価な真理値表はどれか。ここで、は論理積、は論理和、は否定を表す。



(ア)

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

(イ)

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

(ウ)

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

(エ)

A	B	F
0	0	1
0	1	1
1	0	0
1	1	0

解答

(ウ) □

§4

自主学習課題

問題 4.1 出典：平成 21 年度応用情報技術者試験（AP） 秋期 午前 問 1

2 進数の表現で、2 の補数を使用する理由はどれか。

- (ア) 値が 1 のビットを数えることでビット誤りを検出できる。
- (イ) 減算を、負数の作成と加算処理で行うことができる。
- (ウ) 除算を、減算の組合せで行うことができる。
- (エ) ビットの反転だけで、負数を求めることができる。

問題 4.1 出典：平成 21 年度応用情報技術者試験（AP） 秋期 午前 問 1

2 進数の表現で、2 の補数を使用する理由はどれか。

- (ア) 値が 1 のビットを数えることでビット誤りを検出できる。
- (イ) 減算を、負数の作成と加算処理で行うことができる。
- (ウ) 除算を、減算の組合せで行うことができる。
- (エ) ビットの反転だけで、負数を求めることができる。

解答

(イ)

□

問題 4.2 出典：令和 4 年度基本情報技術者試験（FE）問 2

2 の補数で表された負数 10101110 の絶対値はどれか。

(ア)01010000

(イ)01010001

(ウ)01010010

(エ)01010011

問題 4.2 出典：令和 4 年度基本情報技術者試験（FE）問 2

2 の補数で表された負数 10101110 の絶対値はどれか。

(ア)01010000

(イ)01010001

(ウ)01010010

(エ)01010011

解答

2 の補数表現で表された負数の絶対値を求めるには、与えられたビット列の 2 の補数を求めればよい。

まず、10101110 の全ビットの 0 と 1 を反転すると、

$$10101110 \rightarrow 01010001$$

となる。さらに 1 を加えると、

$$01010001 + 00000001 = 01010010$$

となる。

したがって、答えは（ウ）となる。

□

問題 4.3 出典：平成 19 年度基本情報技術者試験（FE）春期 午前 問 4

浮動小数点表示法における仮数が正規化されている理由として、適切なものはどれか。

- (ア) 固定小数点数とみなして大小関係が調べられるようにする。
- (イ) 四則演算のアルゴリズムが簡素化できる。
- (ウ) 表現可能な数値の範囲を拡大する。
- (エ) 有効数字のけた数を最大に保つ。

問題 4.3 出典：平成 19 年度基本情報技術者試験（FE）春期 午前 問 4

浮動小数点表示法における仮数が正規化されている理由として、適切なものはどれか。

- (ア) 固定小数点数とみなして大小関係が調べられるようにする。
- (イ) 四則演算のアルゴリズムが簡素化できる。
- (ウ) 表現可能な数値の範囲を拡大する。
- (エ) 有効数字のけた数を最大に保つ。

解答

(エ)

□

問題 4.4 出典：平成 17 年度ソフトウェア開発技術者試験 秋期 午前 問 2

規格 IEEE 754 (IEC 60559) による単精度の浮動小数点表示法は、次のとおりである。10 進数 14.75 をこの規格に従って表示したときの指数部 E のビット列はどれか。

【IEEE 754】

$0 < E < 255$ の時に表示される実数

$$(-1)^S \times 2^{E-127} \times (1 + F)$$

ここで、 S は実数の符号 (0: 正、1: 負)

E はけたばき (バイアス付き) の指数

F は純小数

これら S 、 E 、 F の 2 進数表示を並べて元の数を表す。

例えば、2 進数 $(0.011)_2$ は、 $(-1)^0 \times 2^{125-127} \times (1+0.1)_2$ なので、 $S = 0$ 、 $E = 125$ 、 $F = (0.1)_2$ となる。ここで、 $()_2$ 内の数は 2 進数を表す。

(ア)00000010

(イ)00000011

(ウ)10000010

(エ)10000011

問題 4.4 出典：平成 17 年度ソフトウェア開発技術者試験 秋期 午前 問 2

規格 IEEE 754 (IEC 60559) による単精度の浮動小数点表示法は、次のとおりである。10 進数 14.75 をこの規格に従って表示したときの指数部 E のビット列はどれか。

【IEEE 754】

$0 < E < 255$ の時に表示される実数

$$(-1)^S \times 2^{E-127} \times (1 + F)$$

ここで、 S は実数の符号 (0: 正、1: 負)

E はげたばき (バイアス付き) の指数

F は純小数

これら S 、 E 、 F の 2 進数表示を並べて元の数を表す。

例えば、2 進数 $(0.011)_2$ は、 $(-1)^0 \times 2^{125-127} \times (1+0.1)_2$ なので、 $S = 0$ 、 $E = 125$ 、 $F = (0.1)_2$ となる。ここで、 $()_2$ 内の数は 2 進数を表す。

(ア)00000010

(イ)00000011

(ウ)10000010

(エ)10000011

解答

$(14.75)_{10}$ を 2 進数で表すと、

$$(14.75)_{10} = 2^3 + 2^2 + 2^1 + 0 \times 2^0 + 2^{-1} + 2^{-2} \quad (4.1)$$

$$= 2^3 (1 + 2^{-1} + 2^{-2} + 0 \times 2^{-3} + 2^{-4} + 2^{-5}) \quad (4.2)$$

$$= (1.11011)_2 \times 2^3 \quad (4.3)$$

となる。

したがって、問題文の

$$(-1)^S \times 2^{E-127} \times (1 + F)$$

と比較すると、

$$E - 127 = 3$$

であるから、

$$E = 3 + 127 = 130$$

となる。また、

$$(130)_{10} = 2^7 + 2^1 = (10000010)_2$$

である。

したがって、答えは (ウ) となる。

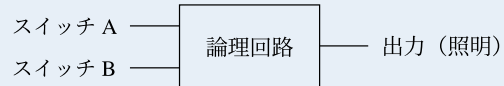
□

問題 4.5 出典：平成 28 年度応用情報技術者試験（AP） 秋期 午前 問 23

次の条件を満足する論理回路はどれか。

【条件】

階段の上下にあるスイッチ A 又は B で、一つの照明を点灯・消灯する。すなわち、一方のスイッチの状態にかかわらず、他方のスイッチで照明を点灯・消灯できる。



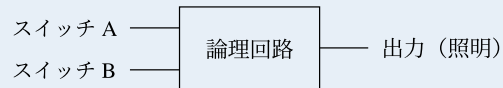
- (ア)AND
- (イ)NAND
- (ウ)NOR
- (エ)XOR

問題 4.5 出典：平成 28 年度応用情報技術者試験（AP） 秋期 午前 問 23

次の条件を満足する論理回路はどれか。

【条件】

階段の上下にあるスイッチ A 又は B で、一つの照明を点灯・消灯する。すなわち、一方のスイッチの状態にかかわらず、他方のスイッチで照明を点灯・消灯できる。



- (ア)AND
- (イ)NAND
- (ウ)NOR
- (エ)XOR

解答

階段の上側にあるスイッチを A，下側にあるスイッチを B とし，それぞれの二つの状態を 0,1 で表す．また，照明の状態を出力 Y とし，消灯を 0，点灯を 1 とする．

XOR の真理値表は次のようになる．

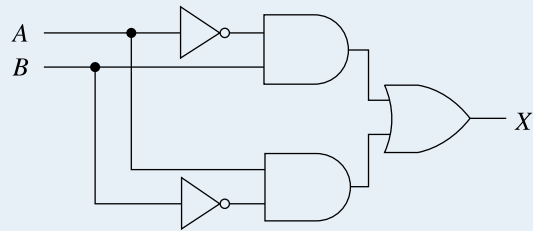
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

この表から，一方のスイッチの状態を固定したまま他方を切り替えると，出力 Y が必ず反転することがわかる．

したがって，一方のスイッチの状態にかかわらず，他方のスイッチで照明を点灯・消灯できるという条件を満たすのは XOR である．よって，答えは（エ）である。 □

問題 4.6 出典：令和元年度基本情報技術者試験（FE） 秋期 午前 問 22

次の回路の入力と出力の関係として、正しいものはどれか。



(ア)

入力		出力
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(イ)

入力		出力
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(ウ)

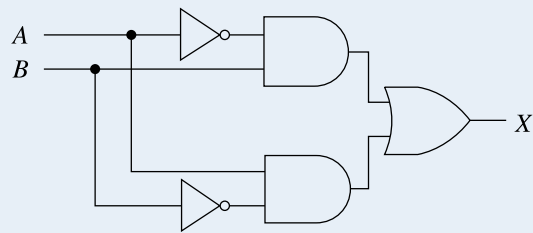
入力		出力
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(エ)

入力		出力
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

問題 4.6 出典：令和元年度基本情報技術者試験（FE） 秋期 午前 問 22

次の回路の入力と出力の関係として、正しいものはどれか。



(ア)

入力		出力
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(イ)

入力		出力
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(ウ)

入力		出力
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(エ)

入力		出力
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

解答

(イ)

□

参考文献

- [1] 藤井 信生, デジタル電子回路 (第 2 版) —集積回路化時代の一—, オーム社, 2020, ISBN 978-4-274-22496-6.