計算機システムＩ・第3回

2022 年 4 月 27 日

今回の内容

3.1　論理回路

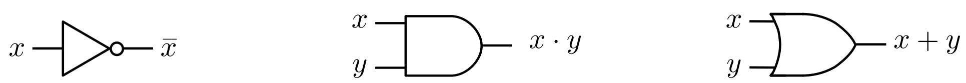
電圧や電流の大きさや向きなどで 0 と 1 を表現することで、電子的 (電気的) に論理演算を行う回路を**論理回路**といいます。論理回路には、(内部状態を持っておらず) 論理式中の各論理変数の値を入力とし、その入力値だけで決まる論理式全体の値を常に計算して出力し続けるものと、過去の入力に依存した内部状態を持っていて、現在の入力値と内部状態によって、出力値が決まるものの 2 つに大別されます。前者は**組み合わせ (論理) 回路**、後者は**順序 (論理) 回路**と呼ばれます。この科目ではまず組み合わせ論理回路について解説します。

メモ

3.2　論理素子

組み合わせ回路にせよ順序回路にせよ、論理回路は、単純な論理演算を行う基本的な部品を組み合わせることで構成します。この基本的な部品のことを**論理素子**と呼びます。

もっとも基本的な論理素子は、それぞれ、否定、論理積、論理和の論理演算を行う **NOT ゲート**、**AND ゲート**、**OR ゲート**の 3 つで、回路図では次のような記号で書き表します。

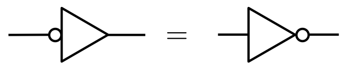


NOT ゲート (否定)　　AND ゲート (論理積)　　OR ゲート (論理和)

各論理素子から伸びている線は、電線 (信号線) を表しており、(この図で) 各論理素子の左側の (1 つあるいは 2 つの) 線が入力となり、右側の (ただ 1 つの) 線が出力となります。これらの線では、たとえば、その電線の電位が (基準となるある電位よりも) 高いか低いかで 0 と 1 を区別して表現します。電圧 (電位の差) で真理値を表す場合、電圧が低いことで 0 を、高いことで 1 を表すものを**正論理**、電圧が低いことで 1 を、高いことで 0 を表すものを**負論理**と呼びます。

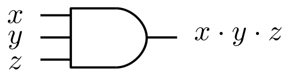
メモ

NOT ゲートについては、次のような別の記号で表すこともあります。こちらの記号で書いた場合でも、論理素子としての働きは変わりません。

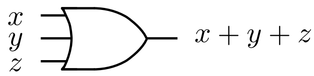


NOT ゲート
(もう 1 つの記号)

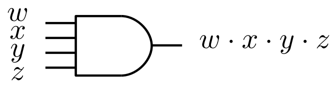
また、AND ゲートや OR ゲートは、次の図のように、3 つ以上の入力を持つものを使用することもあります。



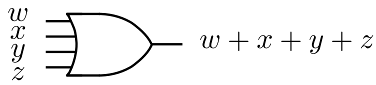
3 入力 AND ゲート



3 入力 OR ゲート



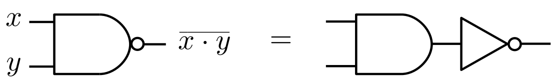
4 入力 AND ゲート



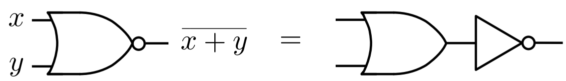
4 入力 OR ゲート

メモ

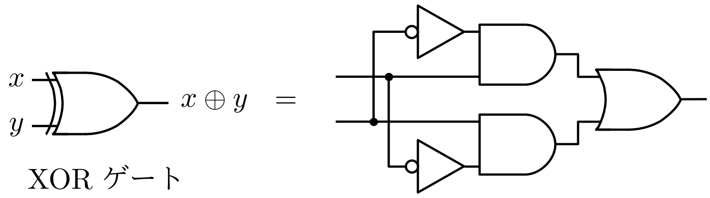
その他の論理素子　論理回路を構成する際には、NOT、AND、OR の 3 つのゲートの他に、次の **NAND** ゲート、**NOR** ゲート、**XOR** ゲートがよく用いられます。



NAND ゲート



NOR ゲート



XOR ゲート

上に示した[1]ように、これらのゲートは、NOT、AND、OR を組み合わせることで構成できますので不可欠な論理素子とは言えませんが、NAND ゲートと OR ゲートは、そのどちらか一方だけあ

[1]回路図では、2 つの線が交差しただけでは、それら 2 つの線の接続を意味しません。(XOR ゲートと等価な論理回路図にあるように) 黒い点 (塗り潰された小さな円) を書くことで、2 つの線がそこで接続されていることを表します。

れば、NOT、AND、OR を含むすべてのゲートを構成することができますので、便利な論理素子としてよく用いられます。

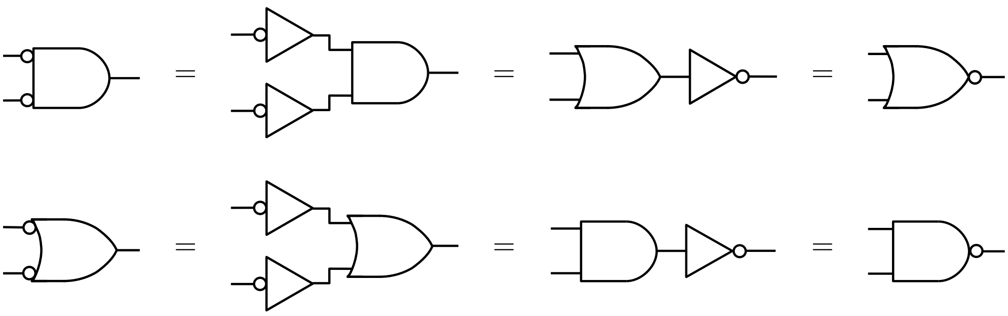
XOR ゲートは**排他的論理和**を計算する論理素子です。第 1 回で解説した (ふつうの) 論理和は「少なくとも一方が 1(真)」のときに 1(真) となる論理演算がですが、排他的論理和は「**どちらか一方だけ**が 1(真)」のときに 1(真) となる論理演算です。論理式では x と y の排他的論理和を $x \oplus y$ で表し、その真理値表は次のようになります[2]。

排他的論理和

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

メモ

否定を表す丸印　NAND ゲートや NOR ゲートの記号の出力側に現れている丸印は、そこでの否定 (NOT) を表しています[3]。この丸印は、次のように、AND ゲートや OR ゲートの入力側に書くこともあります。ド・モルガン律により、これらの論理素子は NOR ゲートや NAND ゲートと等価になります。

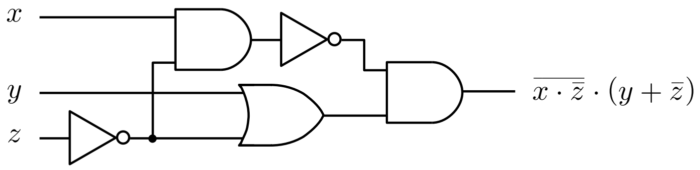


メモ

[2] $x_1 \oplus x_2 \oplus x_3 \oplus \ldots \oplus x_n$ の値は、$x_1, x_2, x_3, \ldots, x_n$ のうち 1 であるものの個数が奇数個のとき 1 となり、偶数個のとき 0 となります。

[3] 正論理に対する負論理 (あるいは負論理に対する正論理) を表していると考えることもできます。

組み合わせ論理回路の例　論理素子の出力を他の論理式の入力に接続することを繰り返し、いくつかの論理素子を組み合わせることで、複雑な論理式の値を計算する論理回路を作ることができます。たとえば、$\overline{x \cdot \bar{z}} \cdot (y + \bar{z})$ という論理式の値は次のような論理回路で計算することができます。



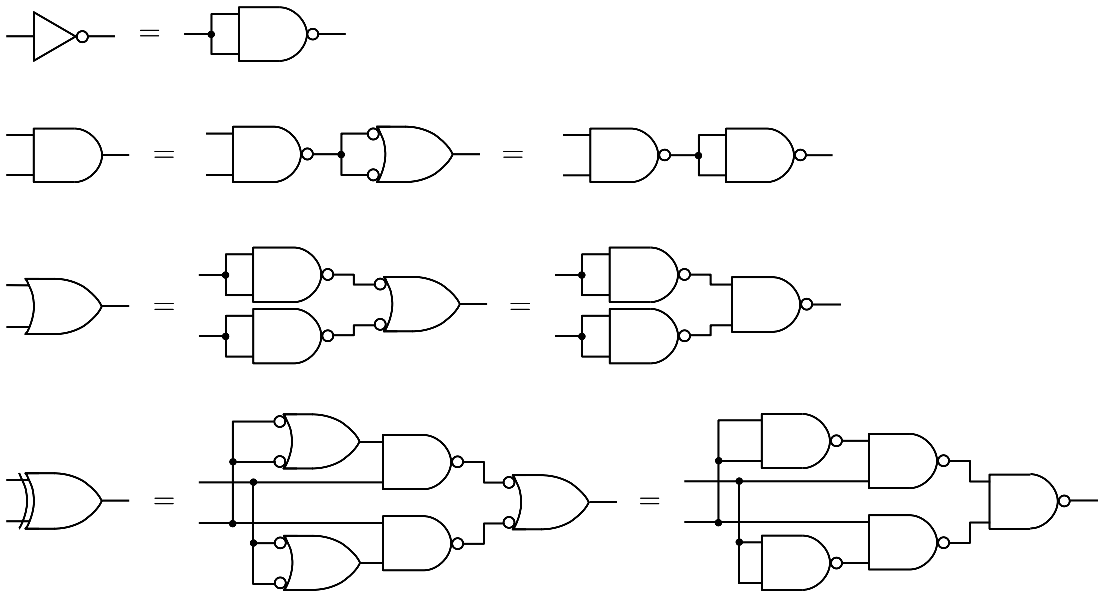
この論理式の真理値表は次のようになります。

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

メモ

計算機システムＩ・第３回・終わり

付録: **NAND** ゲートによる各種ゲートの実現



付録: **NOR** ゲートによる各種ゲートの実現