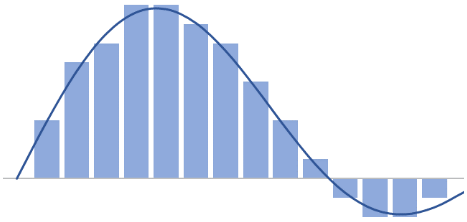
電子工作基礎　デジタル回路編　Part1

デジタル回路とは

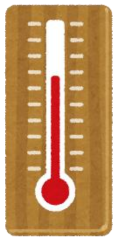
アナログとデジタル



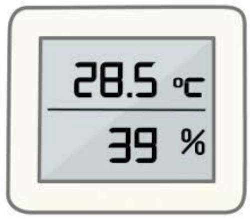
アナログとデジタルの違いをざっくり説明すると、連続しているか飛び飛びであるかという違いです。自然界にある音や光、温度などは連続的な値で変化しており、これを**アナログ**（analog）といいます。それに対して、一般的な時計では時間が更新されるのが 1 秒であるように、データは離散的な（連続的でなく飛び飛びな）値で表現され、これを**デジタル**[1]（digital）といいます。

右のグラフを見てください。波線が元のデータを、棒グラフがそれをデジタルに変換したときのデータを表しています。これをよく見てみると、元のデータとの誤差があるということがわかります。これは、段階的に区切っているが故の誤差です。右図はアナログとデジタルの温度計です。アナログは連続的に変化する温度を動き続ける液面の高さで表しているのに対し、デジタルは温度を数字で表しています。デジタルではデータを段階的に区切っていますから、データを伝達したり保存したりすることが楽になり、ノイズが加わってもデータをもと通り再現できます。その一方で、元のデータは完全に再現できていないという欠点があります。

アナログ



デジタル



2 進法

日常生活において私たちが数を表すには、位取りの基礎を 10 とする **10 進法**（decimal system）を使っています。例えば、10 進法で表された数 1234 は

$$1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 = 1000 + 200 + 30 + 4 = 1234$$

と考えられます。10 進法では位として 10^0 の位，10^1 の位，10^2 の位，……を用い、各位の数字は 0 以上 9 以下の 10 種類の整数を用います。

この 10 進法と同様に考えて、位取りの基礎を 2 とする下図の表し方を **2 進法**（binary system）といいます。2 進法で表された数のことを **2 進数**（binary number）といい、コンピュータの世界では 2 進数が使われ、あらゆる数字や文字が 0 と 1 の組み合わせで表されます。2 進数では、位として 2^0 の位，2^1 の位，2^2 の位，……を用い、各位の数字は 0 または 1 を用います。例えば、2 進法で表された数 1101 は 10 進数に直すといくつでしょうか。これは

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$$

と考えられますから、これは 10 進法の数 13 を表しています。

[1] 英語の di を意識して「ディジタル」と表記されることもあります。

ビットとバイト

0 と 1 の 2 つの状態しか持たない情報量をビット（bit）といい、2 進数の 1 桁に相当します。1 ビットでは 0 か 1 かの 2 通りしか表現できませんが、2 ビットであればどうでしょう。これは、00，01，10，11 の 4 通り表現できます。3 ビットなら 8 通り，4 ビットなら 16 通り，5 ビットなら 32 通り，…となっていきます。すなわち、n ビットであれば2^n通り表現できるということです。

また、8 ビットの情報量を 1 バイト（byte）といい単位は B で表します。1 バイトで 256（$= 2^8$）通りの状態を表すことができます。つまり 1 バイトあれば、英数字（a～z，A～Z，0～9）くらいなら表現できるということです。

IC

IC（integrated circuit，集積回路）は小さな基板上にトランジスタ，ダイオード，抵抗，コンデンサなどの多数の回路素子を高密度に集積した超小型の電子回路のことで、現在ではテレビ，コンピュータなどのあらゆる電子機器に使用されています。例えば、Suica や PASMO などのような IC カードにはデータの記録や演算などのために IC が組み込まれています。詳しい IC の使い方については次章で解説します。

電圧レベル

デジタル方式では、数値や文字など全ての情報を 0 と 1 の組み合わせで表すことができ、コンピュータの内部では、高低 2 種類の信号を 1 と 0 に対応させています。具体的には、電源電圧に近い電圧である H（high，ハイ）レベルと、0（GND）に近い電圧である L（low，ロー）レベルの 2 値で表します。例えば 10 進法で表された数 11 を表したければ、HLHH とすれば表すことができます。電圧のほかにも、磁気やスイッチ，ランプでも 2 進数に対応させることができます。

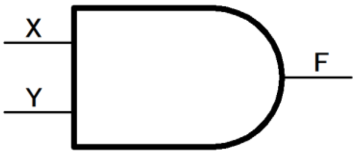
さて、ここで気になる疑問として、H と L の境はどうなるのかということです。電源電圧が 6V のとき、H と L の境は何 V でしょうか。もちろん、0V は L，6V は H ですが、5.5V はどうでしょう。これは、H になります。答えとしては、「具体的に何 V まで L で何 V から H なのか IC によって異なる」です。例えば TC74HC04A はデータシートによると、電源電圧が 6V のとき H は 4.20V 以上，L は 1.80V 以下となっています。また、この間の電圧は判別不能として扱われます。

論理演算

論理演算（logical operation）には AND 演算，OR 演算，XOR 演算，NOT 演算，NAND 演算，NOR 演算などがあります。右側の表は、**真理値表**（truth table）といい、考えられるすべての入力の組み合わせとそれに対応する出力を書き表したものです。この表を丸暗記してもいいですが、意味が理解できれば簡単に覚えることができます。また、中央の記号は MIL 記号といい、回路図などで表記するときに使用します。

AND（論理積）

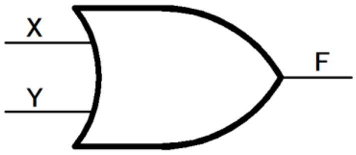
論理積（AND）（logical conjunction）は X と Y の両方に属する部分、すなわち「X かつ Y」の関係を表し、式で表すと$X \cdot Y$です。入力のどちらも 1 の時だけ出力が 1 になります。



X	Y	出力 F
0	0	0
0	1	0
1	0	0
1	1	1

OR（論理和）

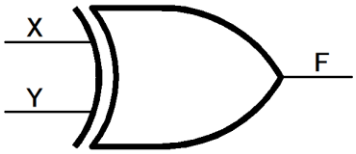
論理和（OR）（logical disjunction）は X と Y の少なくとも一方に属する部分、すなわち「X または Y」の関係を表し、式で表すと$X + Y$です。入力の少なくとも 1 つが 1 の時に出力が 1 になります。



X	Y	出力 F
0	0	0
0	1	1
1	0	1
1	1	1

XOR（排他的論理和）

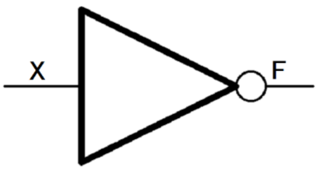
排他的論理和（XOR）（exclusive or）は X と Y のどちらか一方のみに属する部分を表し、式で表すと$\overline{X} \cdot Y + X \cdot \overline{Y}$です。入力が異なる時に出力が 1 になります。



X	Y	出力 F
0	0	0
0	1	1
1	0	1
1	1	0

NOT（論理否定）

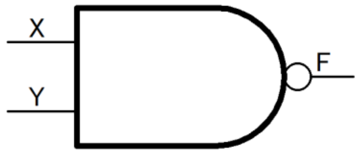
論理否定（NOT）（logical negation）は X を含まない部分、すなわち「X ではない」ことを表し、式で表すと$\overline{X}$です。入力を反転します。



X	出力 F
0	1
1	0

NAND（否定論理積）

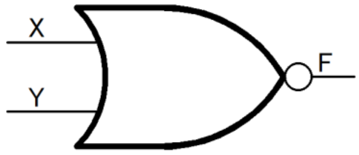
否定論理積（NAND）（non-conjunction）は論理積（AND）の結果を否定（NOT）、すなわち「(X かつ Y) ではない」ことを表し、式で表すと$\overline{\mathrm{X}\cdot\mathrm{Y}}$です。入力のどちらも 1 の時だけ出力が 0 になります。



X	Y	出力 F
0	0	1
0	1	1
1	0	1
1	1	0

NOR（否定論理和）

否定論理和（NOR）（non-disjunction）は論理和（OR）の結果を否定（NOT）、すなわち「(X または Y) ではない」ことを表し、式で表すと$\overline{\mathrm{X}+\mathrm{Y}}$です。入力の少なくとも 1 つが 1 の時に出力が 0 になります。



X	Y	出力 F
0	0	1
0	1	0
1	0	0
1	1	0

また、以下のように NAND や NOR だけで任意の論理演算が可能です。実際、1 つの IC には複数の論理回路が入っているため、NAND や NOR のみで回路を構成することにより、「OR は余っているけど AND は足りない」などといった状況をなくすことができます。

• **A** AND **B**

$$A\cdot B=\overline{\overline{A\cdot B}}=\overline{\overline{A\cdot B}\cdot\overline{A\cdot B}}$$

$$A\cdot B=\overline{\overline{A\cdot B}}=\overline{\overline{A}+\overline{B}}=\overline{\overline{A+A}+\overline{B+B}}$$

• **A** OR **B**

$$A+B=\overline{\overline{A+B}}=\overline{\overline{A}\cdot\overline{B}}=\overline{\overline{A\cdot A}\cdot\overline{B\cdot B}}$$

$$A+B=\overline{\overline{A+B}}=\overline{\overline{A+B}+\overline{A+B}}$$

• **A** XOR **B**

$$\overline{A}\cdot B+A\cdot\overline{B}=\overline{\overline{\overline{A}\cdot B+A\cdot\overline{B}}}=\overline{\overline{\left(\overline{A}\cdot B\right)}\cdot\overline{\left(A\cdot\overline{B}\right)}}=\overline{\overline{\left(\overline{A\cdot A}\cdot B\right)}\cdot\overline{\left(A\cdot\overline{B\cdot B}\right)}}$$

$$\overline{A}\cdot B+A\cdot\overline{B}=\overline{\overline{\overline{A}\cdot B+A\cdot\overline{B}}}=\overline{(A\cdot B)+\left(\overline{A+B}\right)}=\overline{\left(\overline{\overline{A}+\overline{B}}\right)+\left(\overline{A+B}\right)}=\overline{\left(\overline{\overline{A+A}+\overline{B+B}}\right)+\left(\overline{A+B}\right)}$$

• NOT **A**

$$\overline{A}=\overline{A\cdot A}$$

$$\overline{A}=\overline{A+A}$$

• **A** NOR **B**

$$\overline{A+B}=\overline{\overline{\overline{A+B}}}=\overline{\overline{\overline{A}\cdot\overline{B}}}=\overline{\overline{\overline{\overline{A\cdot A}\cdot\overline{B\cdot B}}\cdot\overline{\overline{A\cdot A}\cdot\overline{B\cdot B}}}}$$

• **A** NAND **B**

$$\overline{A\cdot B}=\overline{\overline{\overline{A\cdot B}}}=\overline{\overline{\overline{A}+\overline{B}}}=\overline{\overline{\overline{\overline{A+A}+\overline{B+B}}+\overline{\overline{A+A}+\overline{B+B}}}}$$