

**本時の目標**

コンピュータにおける演算の基本的な内容を身に付ける。

**(1) 論理回路**

コンピュータは情報を ( 0 ) と ( 1 ) で表しており、( 論理回路 ) と呼ばれる回路を構成して ( 論理演算 ) を行っている。

基本  
論理回路

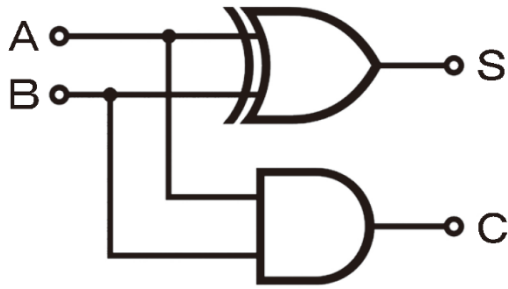
AND 論理 ( 積 )  
OR 論理 ( 和 )  
NOT 否定

NAND ( 否定 ) 論理積  
NOR ( 否定 ) 論理和  
XOR ( 排他的 ) 論理和

論理演算	MIL 記号	ベン図	真理値表															
AND 論理積			<table border="1"> <thead> <tr> <th>入力 A</th> <th>入力 B</th> <th>出力 Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0	0	0	1	0	1	0	0	1	1	1
入力 A	入力 B	出力 Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR 論理和			<table border="1"> <thead> <tr> <th>入力 A</th> <th>入力 B</th> <th>出力 Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0	0	0	1	1	1	0	1	1	1	1
入力 A	入力 B	出力 Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT 否定			<table border="1"> <thead> <tr> <th>入力 A</th> <th>出力 Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	入力 A	出力 Y	0	1	1	0									
入力 A	出力 Y																	
0	1																	
1	0																	
NAND 否定的論理積			<table border="1"> <thead> <tr> <th>入力 A</th> <th>入力 B</th> <th>出力 Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0	1	0	1	1	1	0	1	1	1	0
入力 A	入力 B	出力 Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR 否定的論理和			<table border="1"> <thead> <tr> <th>入力 A</th> <th>入力 B</th> <th>出力 Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0	1	0	1	0	1	0	0	1	1	0
入力 A	入力 B	出力 Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR 排他的論理和			<table border="1"> <thead> <tr> <th>入力 A</th> <th>入力 B</th> <th>出力 Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0	0	0	1	1	1	0	1	1	1	0
入力 A	入力 B	出力 Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

## (2) 半加算器

A,B,C,S に入る値を考えてみよう

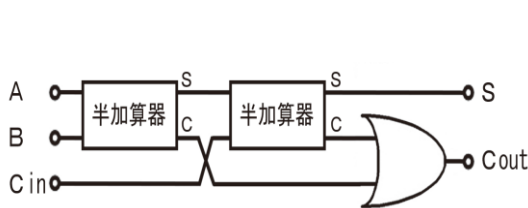


0	+	0	=	0	0
0	+	1	=	0	1
1	+	0	=	0	1
1	+	1	=	1	0
A		B		C	S

## (3) 全加算器

下位からの桁上がりも加えられるようにしたものを ( 全加算器 ) という。

加算する二つの2進数と下位からの桁上がり ( Cin ) が入力され、三つの数を加算した結果の1桁目Sと上位への桁上がり ( Cout ) が出力となる。



入力 A	入力 B	入力 Cin	出力 Cout	出力 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

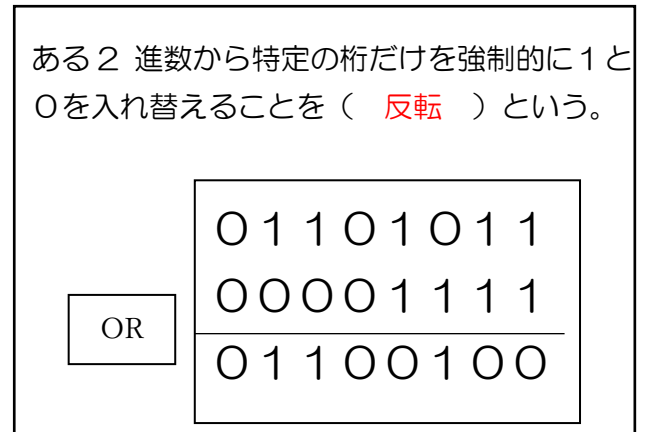
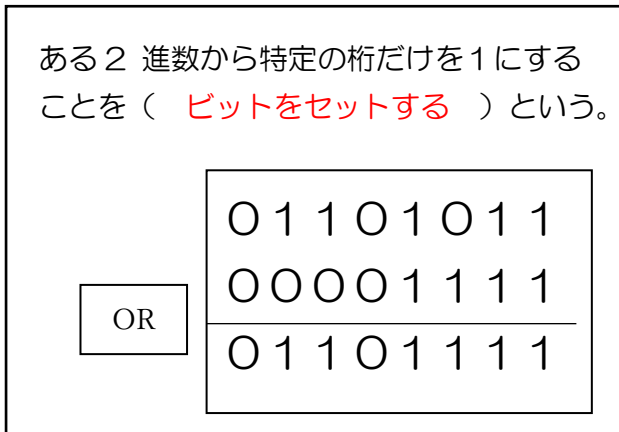
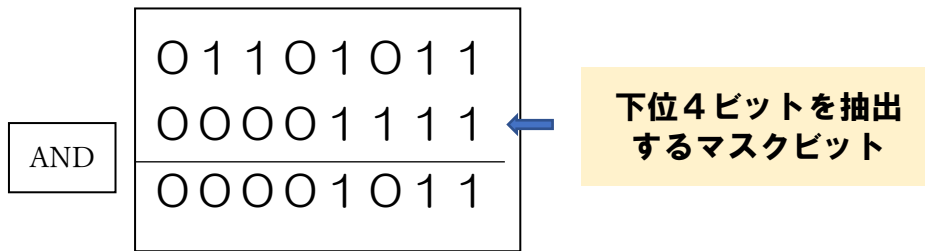
## (4) 論理演算

問題② (01011111)<sub>2</sub>と(00110011)<sub>2</sub>のAND, OR, NAND, NOR, XORを求めなさい。

AND	$\begin{array}{r} 01011111 \\ 00110011 \\ \hline 00010011 \end{array}$	OR	$\begin{array}{r} 01011111 \\ 00110011 \\ \hline 01111111 \end{array}$	NAND	$\begin{array}{r} 01011111 \\ 00110011 \\ \hline 11101100 \end{array}$
NOR	$\begin{array}{r} 01011111 \\ 00110011 \\ \hline 10000000 \end{array}$	XOR	$\begin{array}{r} 01011111 \\ 00110011 \\ \hline 01101100 \end{array}$		

## (5) マスクビット

ある2進数から特定の桁だけを抽出するときに（ **マスクビット** ）を用いる。



## (6) シフト

乗算をするときは、（ **左** ）に1ビットシフトすると（ **2** ）倍される仕組みを利用する。

除算をするときは、（ **右** ）に1ビットシフトすると（ **1/2** ）倍される仕組みを利用する。

(00000001) <sub>2</sub> = (1) <sub>10</sub>	↓ × 2	↑ ÷ 2
(00000010) <sub>2</sub> = (2) <sub>10</sub>	↓ × 2	↑ ÷ 2
(00000100) <sub>2</sub> = (4) <sub>10</sub>	↓ × 2	↑ ÷ 2
(00001000) <sub>2</sub> = (8) <sub>10</sub>	↓ × 2	↑ ÷ 2
(00010000) <sub>2</sub> = (16) <sub>10</sub>	↓ × 2	↑ ÷ 2
(00100000) <sub>2</sub> = (32) <sub>10</sub>	↓ × 2	↑ ÷ 2
(01000000) <sub>2</sub> = (64) <sub>10</sub>		
(10000000) <sub>2</sub> = (128) <sub>10</sub>		

◇本時の目標への自己評価 演算の基本的な内容の理解	A : よくできた、B : できた、C あまりできなかった A B C
◇感想、今後の学習への活用	