

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 () と () で表しており、() と呼ばれる回路を構成して () を行っている。

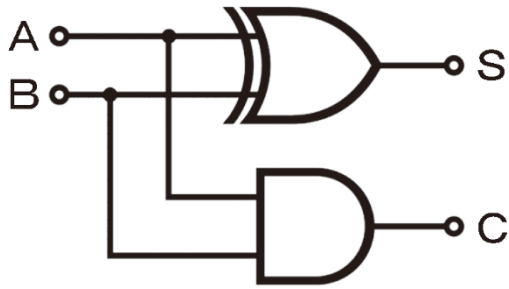
基本
論理回路

AND 論理 () NAND () 論理積
 OR 論理 () NOR () 論理和
 NOT 否定 XOR () 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></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0		0	1		1	0		1	1	
入力 A	入力 B	出力 Y																
0	0																	
0	1																	
1	0																	
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></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0		0	1		1	0		1	1	
入力 A	入力 B	出力 Y																
0	0																	
0	1																	
1	0																	
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></td></tr> <tr><td>1</td><td></td></tr> </tbody> </table>	入力 A	出力 Y	0		1										
入力 A	出力 Y																	
0																		
1																		
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></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0		0	1		1	0		1	1	
入力 A	入力 B	出力 Y																
0	0																	
0	1																	
1	0																	
1	1																	
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></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0		0	1		1	0		1	1	
入力 A	入力 B	出力 Y																
0	0																	
0	1																	
1	0																	
1	1																	
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></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	入力 A	入力 B	出力 Y	0	0		0	1		1	0		1	1	
入力 A	入力 B	出力 Y																
0	0																	
0	1																	
1	0																	
1	1																	

(2) 半加算器

A,B,C,S に入る値を考えてみよう



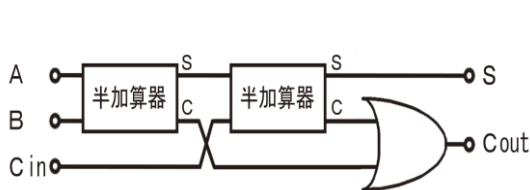
0	+	0	=
0	+	1	=
1	+	0	=
1	+	1	=
A		B	C S

(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（ ）という。
 加算する二つの2進数と下位からの桁上がり（ ）が入力され、三つの数を加算した結果

果

の1桁目Sと上位への桁上がり（ ）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	1			
0	1			
1	0			
1	0			
1	1			
1	1			

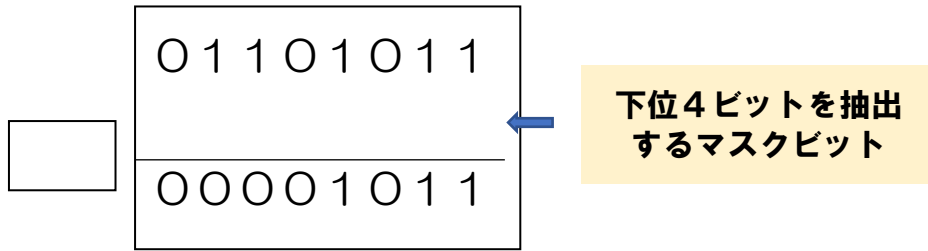
(4) 論理演算

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

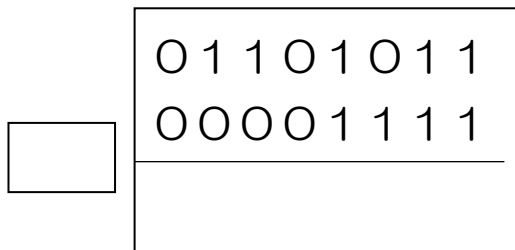
AND	01011111 00110011 <hr/>	OR	01011111 00110011 <hr/>	NAND	01011111 00110011 <hr/>
NOR	01011111 00110011 <hr/>	XOR	01011111 00110011 <hr/>		

(5) マスクビット

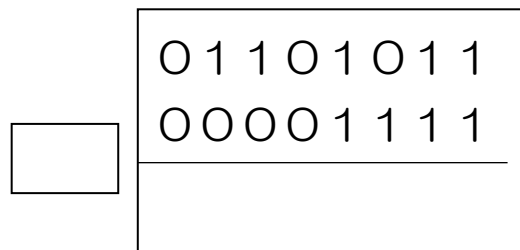
ある2進数から特定の桁だけを抽出するときに（ ）を用いる。



ある2進数から特定の桁だけを1にすることを（ ）という。



ある2進数から特定の桁だけを強制的に1と0を入れ替えることを（ ）という。



(6) シフト

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

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

(00000001) ₂ = () ₁₀	↓ × 2	↑ ÷ 2
(00000010) ₂ = () ₁₀	↓ × 2	↑ ÷ 2
(00000100) ₂ = () ₁₀	↓ × 2	↑ ÷ 2
(00001000) ₂ = () ₁₀	↓ × 2	↑ ÷ 2
(00010000) ₂ = () ₁₀	↓ × 2	↑ ÷ 2
(00100000) ₂ = () ₁₀	↓ × 2	↑ ÷ 2
(01000000) ₂ = () ₁₀		
(10000000) ₂ = () ₁₀		

◇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	