

コンピュータ基礎(10)

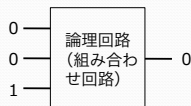
6章 5. 記憶と演算の原理

この章で学習すること

- 論理回路
 - 2進数を電気信号で計算する仕組み
 - 論理回路の基本要素, MIL記号 (AND, OR, NOT)
 - その他の論理回路記号 (XOR, NAND, NOR)
 - 真理値表について
- 記憶のしくみ
 - フリップフロップ回路
- 計算の仕組み
 - 加算回路の基本要素 (半加算器, 全加算器)

論理回路とは

- 0と1の入力から出力を決める回路



例: 多数決回路 (入力のうち0か1の多い方を出力する)

- 論理回路を作るには?

- どのような部品を使うのか?
- どのようにして設計するのか?

詳細は, 2年前期「論理回路」で学習します.

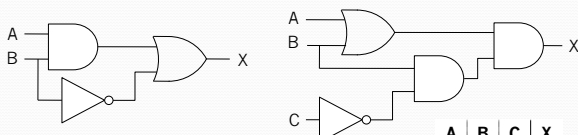
基本論理素子

- どんな論理回路でも, この部品の組み合わせで作ることが出来る

名前	AND	OR	NOT																																				
MIL記号																																							
意味	入力が 両方とも 1のとき, 1を出力	入力の どちらかが 1のとき, 1を出力	入力の 反転 0ならば1, 1ならば0																																				
真理値表	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</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>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</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>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	X	0	1	1	0
A	B	X																																					
0	0	0																																					
0	1	0																																					
1	0	0																																					
1	1	1																																					
A	B	X																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	1																																					
A	X																																						
0	1																																						
1	0																																						

例題

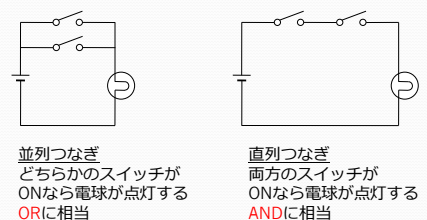
- 以下の論理回路の真理値表を書け.



A	B	X
0	0	
0	1	
1	0	
1	1	

A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

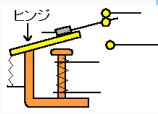
スイッチ回路



- スイッチのつなぎ方で, ANDとORの機能を作る
 - 他の回路からスイッチをON/OFF出来れば, 論理回路を繋いでいくことが出来る
 - 電気によりON/OFF出来るスイッチの利用

基本論理素子の作り方

- リレー（電磁石式のスイッチ）
 - 電磁石に電気を流すとスイッチが ON になる
 - 回路が切れる方もある
- 基本論理素子
 - リレーを2つ、直列つなぎにすると、両方をONにしないと電気が流れない・・・ANDの動作
 - リレーを2つ、並列つなぎにすると、どちらかがONなら電気が流れる・・・ORの動作
 - リレーの回路が切れる側の接点を使うとNOTが作れる
- 実際に、リレーを使った計算機は存在した
 - 1950年代. 計算は遅い.
 - 現在はトランジスタに置き換えられている



オムロンホームページより

その他の論理回路記号

- あると便利
- は反転を表す。
NANDはANDの反転, NORはORの反転

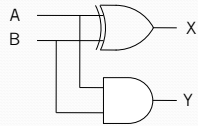
名前	NAND	NOR	XOR																																													
MIL記号																																																
意味	入力が両方とも1のとき, 0を出力	入力のどちらかが1のとき, 0を出力	入力が違えば1を出力 (排他的論理和)																																													
真値表	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <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>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</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>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</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>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
	A	B	X																																													
	0	0	1																																													
	0	1	1																																													
1	0	1																																														
1	1	0																																														
A	B	X																																														
0	0	1																																														
0	1	0																																														
1	0	0																																														
1	1	0																																														
A	B	X																																														
0	0	0																																														
0	1	1																																														
1	0	1																																														
1	1	0																																														

加算回路(1)

- 筆算の1けた分を考える
 - $0+0=0$
 - $0+1=1, 1+0=1$
 - $1+1=0$ で, 1繰り上がる
- $A+B=X$ で, 繰り上がり Y とすると
 - X は A と B の XOR
 - Y は A と B の AND
- これを半加算器という
 - 下の桁からの繰り上がりを計算できないので, 「半」

$$\begin{array}{r} 0\ 0\ 1\ 1 \\ +\ 0\ 1\ 1\ 0 \\ \hline 1\ 0\ 0\ 1 \end{array}$$

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

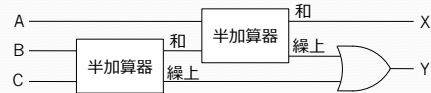


加算回路(2)

- 下の桁からの繰り上がりを考える
 - $0+0+0=0$
 - $0+0+1=0+1+0=1+0+0=1$
 - $0+1+1, 1+0+1, 1+1+0$ のときはその桁は0で, 1繰り上がる
 - $1+1+1$ のときは, その桁は1で, 1繰り上がる
- これを全加算器という
 - 半加算器を2つ使って作れる

$$\begin{array}{r} 0\ 0\ 1\ 1 \\ +\ 0\ 1\ 1\ 0 \\ \hline 1\ 0\ 0\ 1 \end{array}$$

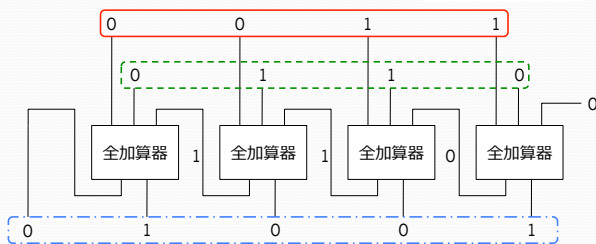
A	B	C	X	Y
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



加算回路(3)

- 筆算の回路を作ることが出来る

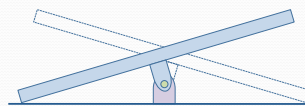
$$\begin{array}{r} 0\ 0\ 1\ 1 \\ +\ 0\ 1\ 1\ 0 \\ \hline 0\ 1\ 0\ 0\ 1 \end{array}$$



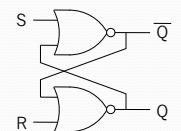
- その他の計算
 - 引き算: 2の補数 (反転して1を足す) して加算
 - 掛け算: シフト (桁ずらし) と加算の繰り返し

記憶回路 (フリップフロップ)

- 記憶素子は, 1か0のどちらかの状態を記憶する



シーソー: 片側に倒れた状態で安定する
Flip-Flop は, シーソーの動きの擬態語.



- フリップフロップの動作
 - S(set)=1のとき, Q=1になる
 - R(reset)=1のとき, Q=0になる
 - S=R=0のとき, 前の状態を保持する
 - \bar{Q} は常に Q の反転である
- フリップフロップは, SRAMの1ビット分の記憶素子である.