3－4 ALU の構成回路

ALU（算術論理演算装置）は，算術演算や論理演算を行う装置である。この装置は，さまざまな回路により構成されている。

3-4-1 論理回路

論理回路とは，コンピュータが命令を論理的に処理するための**論理演算**を実現する回路である。論理演算とは，真(1)又は偽(0)の２通りの真理値しかもたない演算である。

〔代表的な論理演算〕

① **論理積演算（AND 演算）**
　二つの入力値がともに真(1)のときに真(1)，そうでない（どちらか一方でも偽(0)の）ときに偽(0)を出力する。

② **論理和演算（OR 演算）**
　二つの入力値がともに偽(0)のときに偽(0)，そうでない（どちらか一方でも真(1)の）ときに真(1)を出力する。

③ **否定演算（NOT 演算）**
　入力値が真(1)のときに偽(0)を，偽(0)のときに真(1)を出力する。

④ **排他的論理和演算（XOR 演算又は EOR 演算）**
　二つの入力値が同じときに偽(0)を，二つの入力値が異なるときに真(1)を出力する。

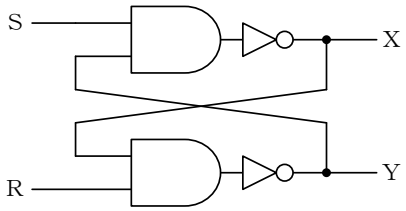
それぞれの論理演算の入力と出力の関係を表にまとめると，次のようになる。このような表のことを**真理値表**という。

入力		論理積	論理和	否定	排他的論理和
X	Y	（X AND Y）	（X OR Y）	（NOT X）	（X XOR Y）
真(1)	真(1)	真(1)	真(1)	偽(0)	偽(0)
真(1)	偽(0)	偽(0)	真(1)	偽(0)	真(1)
偽(0)	真(1)	偽(0)	真(1)	真(1)	真(1)
偽(0)	偽(0)	偽(0)	偽(0)	真(1)	偽(0)

なお，論理積は“X AND Y”のほかに“X・Y”や“X∧Y”，論理和は“X OR Y”のほかに“X＋Y”や“X∨Y”，否定は“NOT X”のほかに“$\overline{X}$”や“￢X”，排他的論理和は“X XOR Y”のほかに“X⊕Y”や“X≢Y”と表すこともある。

(1)　順序回路

順序回路は，そのときの入力と，それ以前の状態によって出力が決まる論理回路のことである。代表的な順序回路であるフリップフロップ回路は，二つの安定状態をもつ回路であり，SRAM の記憶セルに使用されている。参考までに，図 1-25 にフリップフロップ回路の回路図と真理値表を示す。図 1-25 で使用されている回路の記号は，電子回路を表現する方式の一つで MIL 記号（MIL：Military Specifications and Standards）という。



S	R	X	Y	動作
1	1	X	Y	保持
1	0	0	1	リセット
0	1	1	0	セット
0	0	—	—	(禁止)

図 1-25　フリップフロップ回路の回路図と真理値表

(2)　組合せ回路

組合せ回路は，そのときの入力によって出力が決定する論理回路のことである。基本的な論理演算（論理積演算，論理和演算，否定演算）を実現するための基本回路から，これらの基本回路を組み合わせることで構成される回路まで，さまざまな種類がある。

①　AND 回路

論理積演算（AND 演算）を行う回路である。AND 回路の真理値表と MIL 記号は，次のとおりである。

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

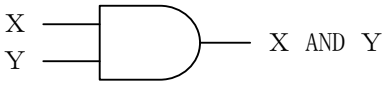


図 1-26　AND 回路

②　OR 回路

論理和演算（OR 演算）を行う回路である。OR 回路の真理値表と MIL 記号は，次のとおりである。

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

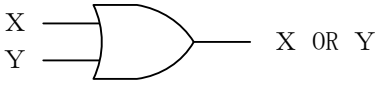


図 1-27　OR 回路

③　NOT 回路

否定演算（NOT 演算）を行う回路である。NOT 回路の真理値表と MIL 記号は，次のとおりである。

X	NOT X
0	1
1	0

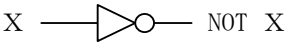


図 1-28　NOT 回路

④　NAND 回路

否定論理積演算（NAND 演算）を行う回路である。否定論理積演算は，論理積演算の結果を否定する演算である。NAND 回路の真理値表と MIL 記号は，次のとおりである。NAND 回路は，AND 回路と NOT 回路を組み合わせた構成の回路である。

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

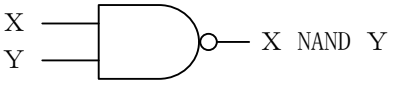


図 1-29　NAND 回路

⑤　NOR 回路

否定論理和演算（NOR 演算）を行う回路である。否定論理和演算は，論理和演算の結果を否定する演算である。NOR 回路の真理値表と MIL 記号は，次のとおりである。NOR 回路は，OR 回路と NOT 回路を組み合わせた構成の回路である。

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

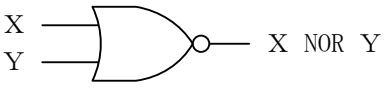


図 1-30　NOR 回路

⑥　XOR 回路

排他的論理和演算（XOR 演算）を行う回路である。XOR 回路の真理値表と MIL 記号は，次のとおりである。

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

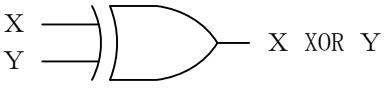


図 1-31　XOR 回路

XOR 回路も，基本回路の組合せで構成される。しかし，NAND 回路や NOR 回路のように単純な回路の組合せでは実現できない。このようなときに，目的とする結果（真理値）を得るための基本回路の組合せを考えることを論理設計又は**回路設計**という。

論理設計では，真理値表から得られた論理式（**論理関数**）を，論理演算の法則（論理法則）などを利用して簡素化し，最適な基本回路の組合せ方法を求める（“最適”とは，性能，効率，コストなどを考慮して設計することを意味する）。

〔代表的な論理法則〕

※論理積演算を∧，論理和演算を∨，否定演算を¬で表記する。

・**べき等則**

A∨A＝A

A∧A＝A

・**交換則**

A∨B＝B∨A

A∧B＝B∧A

・**結合則**

A∨(B∨C)＝(A∨B)∨C

A∧(B∧C)＝(A∧B)∧C

・**分配則**

A∨(B∧C)＝(A∨B)∧(A∨C)

A∧(B∨C)＝(A∧B)∨(A∧C)

・**吸収則**

A∨(A∧B)＝A

A∧(A∨B)＝A

・**ド・モルガンの法則**

¬(A∨B)＝(¬A)∧(¬B)

¬(A∧B)＝(¬A)∨(¬B)

・その他

A∨0＝A

A∧0＝0

A∨1＝1

A∧1＝A

A∨(¬A)＝1

A∧(¬A)＝0

¬(¬A)＝A　　（二重否定により元に戻る）

【XOR 回路の論理設計】

1)　真理値表の出力が１となる部分に着目し，そのときの入力で１となる論理積演算を求める。

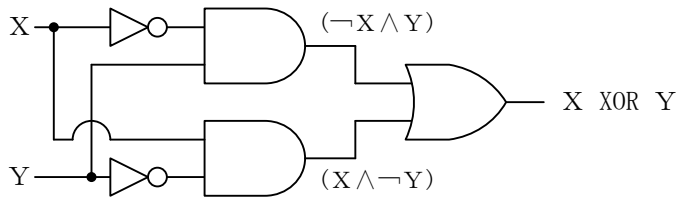
・（X＝0，Y＝1）のとき：¬X∧Y
・（X＝1，Y＝0）のとき：X∧¬Y

2)　求めた論理積演算のどちらかが１であれば出力が１となる回路を作成するので，二つの論理積演算を論理和演算で結んだ論理式Ｆを求める。

論理式Ｆ＝(¬X∧Y)∨(X∧¬Y)

3)　論理式Ｆを表すように，基本回路を組み合わせて回路図を求める。

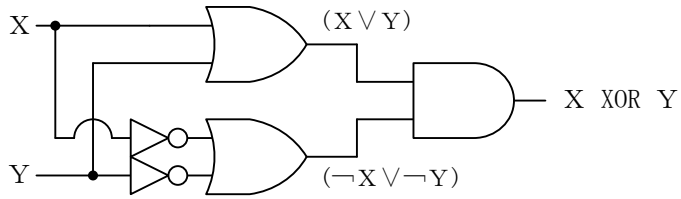
【回路図】



このとき，【XOR 回路の論理設計】で求めた論理式Ｆを，論理法則を利用して展開すると異なる論理回路を求めることができる。

論理式Ｆ＝(¬X∧Y)∨(X∧¬Y)
＝((¬X∧Y)∨X)∧((¬X∧Y)∨¬Y)　・・・ 分配則
＝((¬X∨X)∧(Y∨X))
　　∧((¬X∨¬Y)∧(Y∨¬Y))　・・・ 分配則
＝(１∧(Y∨X))∧((¬X∨¬Y)∧１)　・・・ A∨(¬A)＝１
＝(X∨Y)∧(¬X∨¬Y)　・・・ A∧１＝A，交換則

【回路図】



なお，実際の論理設計では，NAND 回路，NOR 回路，XOR 回路も，そのまま一つの回路（基本回路）として利用するほうが一般的である。