■論理回路

「０」と「１」という２つの信号で演算や制御を行う回路
コンピュータはAND (論理積)回路，　OR (論理和)回路，　NOT (否定)回路
の３つの論理回路の組み合わせですべての計算を行うことができる。

●論理積(AND)回路

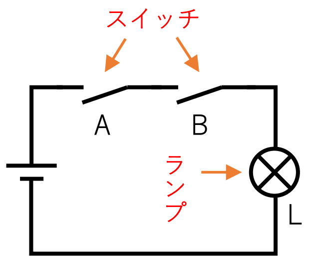


図１ AND回路

表１ AND回路の真理値表

入力		出力
A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

※真理値表…入力と出力の関係を示す表

２つの入力がともに１の時だけ，出力信号が１になる回路

２つのスイッチを直列につなげた回路をイメージする

●論理和(OR)回路

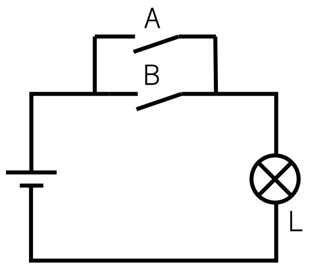


図３ OR回路

表２ OR回路の真理値表

入力		出力
A	B	L
0	0	0
0	1	1
1	0	1
1	1	1

入力のいずれか一方が１であれば，１を出力する回路

２つのスイッチを並列につなげた回路をイメージする

●否定(NOT)回路

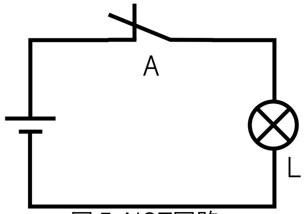


図５ NOT回路

表３ NOT回路の真理値表

入力	出力
A	L
0	1
1	0

入力した信号を反転した値を出力する回路

押したら切れるスイッチを付けた回路イメージする

図記号は次のように表す

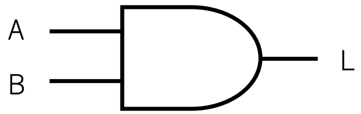


図２ AND回路の図記号

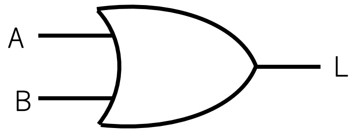


図４ OR回路の図記号

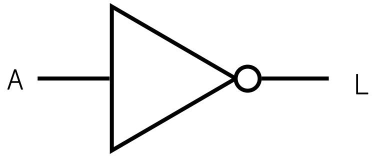


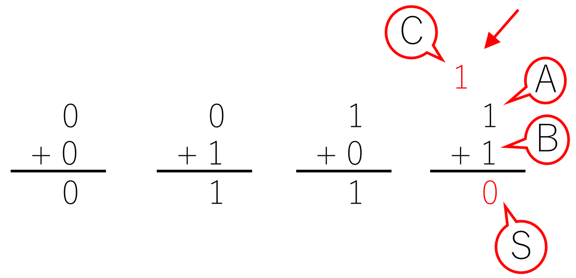
図６ NOT回路の図記号

■演算の仕組み

●２進数の加算

回路を汲む前に、２進数の加算の計算結果
の全パターンを確認する

２進数の加算は，0と1しかないため，
1を超えると桁上げをすることになる



足し合わせる数をA、B、計算結果の１桁目をS、
桁上げの出力をCと置き、本当に半加算回路でこの
結果が得られるか真理値表を作って確認する

●半加算回路（HA）

この回路を参考にして半加算回路の真理値を作成
して本当に計算結果が出るか確認する

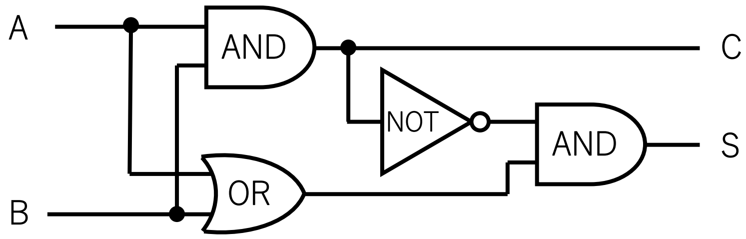


図７　半加算回路

表４ 半加算回路の真理値表

入力		出力	
A	B	C	S
0	0		
0	1		
1	0		
1	1		

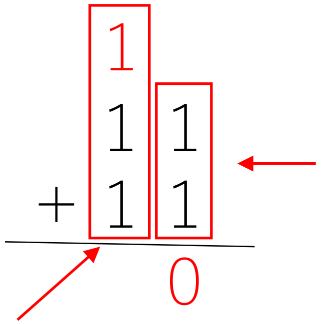
入力		出力	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

２進数の加算の結果が表現できて
いることがわかる

・・・ただし、この回路は１桁の
計算しかできない。

■２桁以上の計算

１１＋１１　の計算を考える



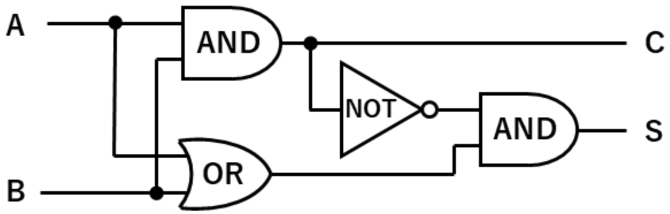
１桁目は
半加算回路で計算できるが

３つの数を足す２桁目は
半加算回路では計算できない

そこで、３入力の全加算回路を使う

●全加算回路（FA）

まず、複雑な半加算回路を、



このように簡単に表す

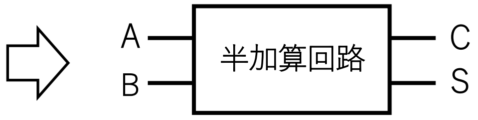
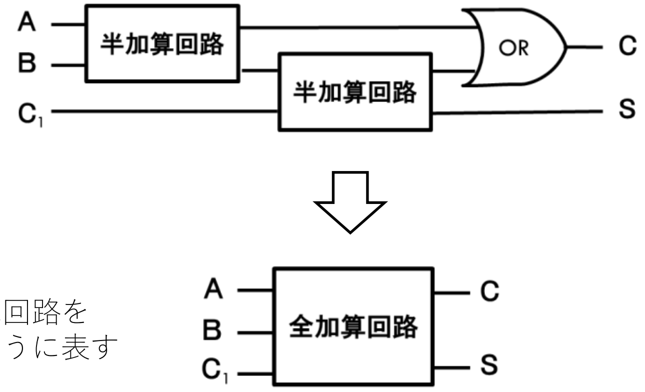


図８ 半加算回路②

半加算回路２つとOR回路１つで回路を使って下記のような回路を作ると、右の真理値表の結果が得られ、２桁目の計算ができる



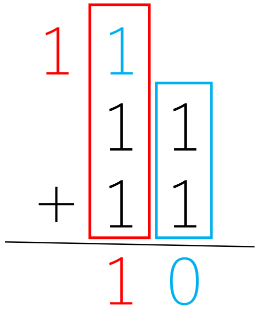
全加算回路を
このように表す

図９　全加算回路

表５ 全加算回路の真理値表

入力			出力	
A	B	C_1	S	C
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

■２桁以上の計算



半加算回路と全加算回路をこのようにつなげば、２桁の計算ができる

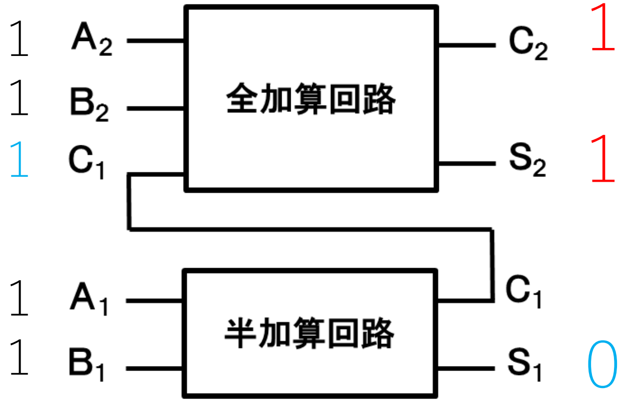
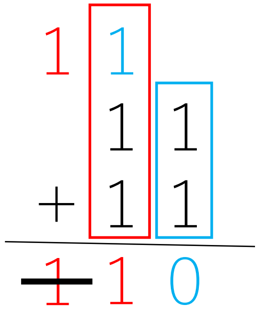


図１０ ２桁の加算をする回路

３桁の計算をするなら、もう一つ全加算回路をつなげれば計算できる

最後に・・・

ここでは２桁の計算で、３桁目を出しているが



指定された桁（ビット）数より、大きい値は、桁あふれとして処理され、無視される

このことについては、補数を使った減算で勉強する。