4.3　加算回路

2 章で述べたように、コンピュータの算術演算の基本は加算とシフト演算です。シフト演算の回路を構成するのは簡単なので省略します。ここで重要なのは加算回路で、論理素子を使って構成されます。

最初に、1 桁の加算[1]$S = A + B$ を考えてみましょう。$A = 1$ かつ $B = 1$ のとき繰り上がりが起きることに注意します。繰り上がりの有無を変数 C とし、A と B の組み合わせを全て求めると表 4.1 になります。

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$\Longleftrightarrow$

$$S = A \oplus B$$

$$C = A \cdot B$$

表 4.1: 1 桁の加算

従って、S と C の関係式が求まるので、図 4.30 のような回路によって 1 桁の加算回路を構成することができます。ただし、任意の桁の加算を行うときには下の桁からの繰り上がりを考慮する必要があり、この回路で実現されるのは繰り上がりだけです。そのため、**半加算器**(half adder, HA) と呼ばれ、図 4.31 のようにブロック図に置き換えて表します。

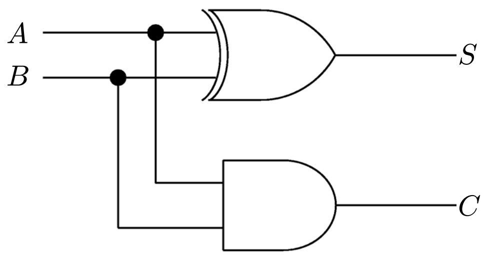

図 4.30: 1 桁の加算回路

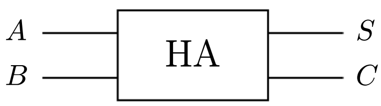

図 4.31: 半加算器のブロック図

全加算器(full adder, FA) は 2 つの半加算器を使って作ることができます。半加算器と同じように考えて、i 桁目の全加算器を考えてみましょう。i 桁目の A_i, B_i と $i-1$ 桁目の繰り上がり C_{i-1} の全ての組み合わせから S_i と C_i の値は表 4.2 となります。

[1]論理和ではないことに注意しなさい。

C_{i-1}	A_i	B_i	S_i	C_i
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

$$\Longleftrightarrow \quad \begin{aligned} S_i &= (A_i \oplus B_i) \oplus C_{i-1} \\ C_i &= (A_i \oplus B_i) \cdot C_{i-1} + A_i \cdot B_i \end{aligned}$$

表 4.2: i 桁目の全加算

従って、S_i と C_i の関係式が求まるので、図 4.32 のような回路によって i 桁目の全加算器を構成することができます。

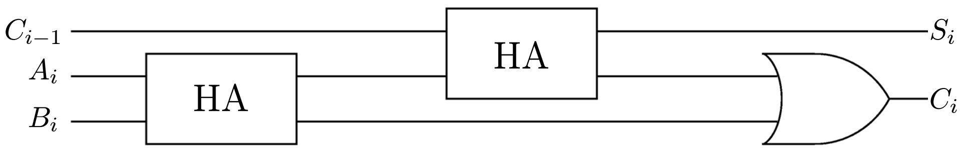

図 4.32: i 桁目の全加算器

また、図 4.33 のようにブロック図に置き換えて表します。全加算器を使って 4 ビットの加算器を作ると図 4.34 のようになります。

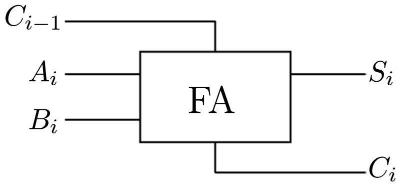

図 4.33: 全加算器のブロック図

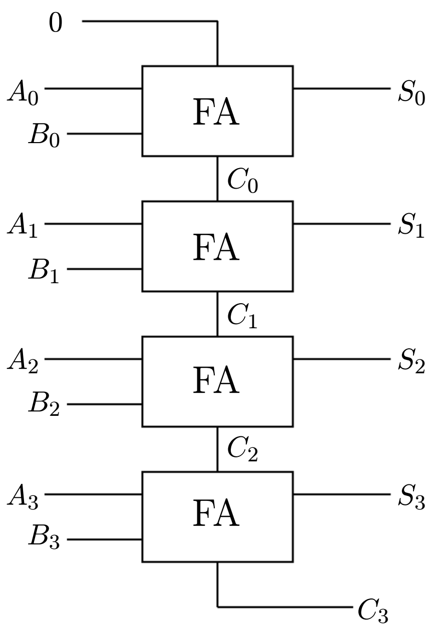

図 4.34: 4 ビットの全加算器

4.4　フリップフロップ

コンピュータを構成する上で、演算回路の他にレジスタやプログラムカウンタなどの記憶回路が必要となります。実際、**フリップフロップ**(flip flop) によって実現されていて、図 4.35 のような回路で実現されます。

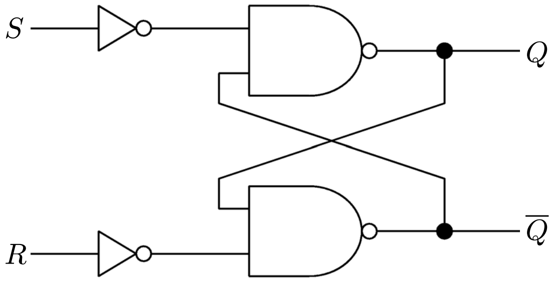

図 4.35: フリップフロップ

図 4.35 の回路は、$S=1$ かつ $R=0$ のとき $Q=1, \overline{Q}=0$ の安定状態となります (この回路は対称なので $S=0$ かつ $R=1$ のとき $Q=0, \overline{Q}=1$ の安定状態になります)。また、$S=1$ かつ $R=0$ の状態から S の値を $1 \to 0 \to 1 \to \cdots$ に変化させても $Q, \overline{Q}$ の値は変化しないことがわかります。逆に、$S=1, R=0$ または $S=0, R=0$ の状態から $S=0, R=1$ の状態にすると $Q=0, \overline{Q}=1$ の状態になり値が変化します。以上をまとめると、表 4.3 のように S と R の状態によって $Q, \overline{Q}$ の値が変化します。ただし、$S=1, R=1$ のときは状態が不安定になるので使いません。このように、フリップフロップ回路は S と R の値によって Q の値を保持や変更が可能になります。

S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0

表 4.3: R, S による $Q, \overline{Q}$ の状態

実際のコンピュータでは**クロック**(clock) と同期を取って値の変更を行います (図 4.36 参照)。なお、クロック C は周期的に 1 と 0 の状態を交互に作り出します。

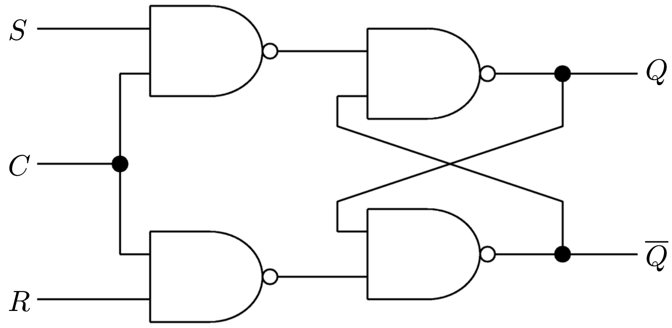

図 4.36: クロックを用いたフリップフロップ

初期状態を $S = 1, R = 0$ として、時間的な流れによる変化を見ていきましょう。

クロックだけで Q の値を変えない場合 ($S = 0, R = 0$)

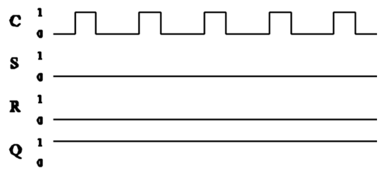

図 4.37: $S = 0, R = 0$

Q の値を 0 にセットする場合 ($S = 0, R = 1$)

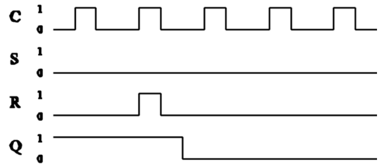

図 4.38: $S = 0, R = 1$

Q の値を 1 に再セットする場合 ($S = 1, R = 0$)

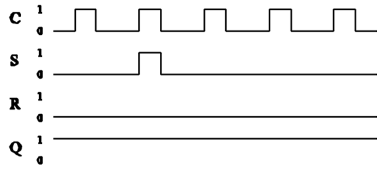

図 4.39: $S = 1, R = 0$