4.3 フリップフロップ

フリップフロップ(flip flop) は、1 ビットの情報 (2 つの安定状態) を記憶することのできる論理回路で、中央処理装置内部のレジスタやキャッシュメモリなど、記憶を司る回路に使用されています。ここでは、構造が最もシンプルな図 4.30 の **RS フリップフロップ**[4]について解説します[5]。図中の S (Set) と R (Reset) は入力、Q と $\overline{Q}$ は出力を表します[6]。なお、**出力 Q が記憶される 1 ビットの情報です**。

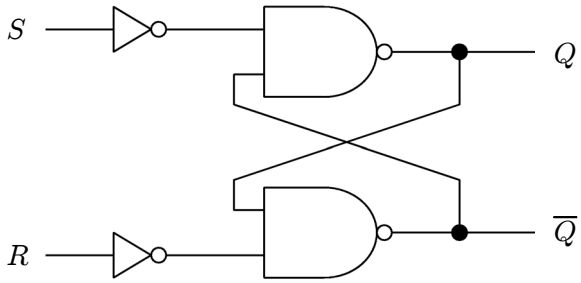


図 4.30: RS フリップフロップ

まず、RS フリップフロップに 1 を記憶させてみましょう (**初期状態**[7])。1 を記憶させるには、**セット** ($S = 1$, $R = 0$) を行います。セットを行うと図 4.31 のように回路は安定状態となり、1 が記憶されます。

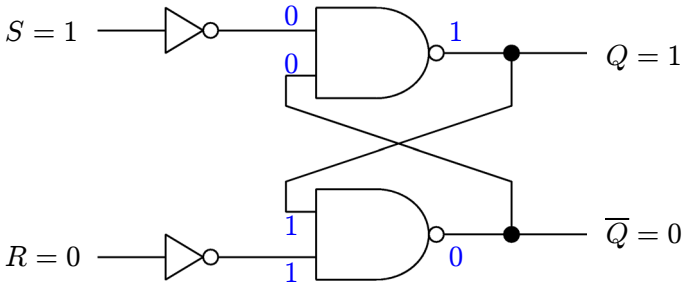


図 4.31: セット (1 を記憶)

逆に、**リセット** ($S = 0$, $R = 1$) を行うことで、0 が記憶されます (付録 B.1 のチェックシートを利用して各自で確認してください)。

次に、セット ($S = 1$, $R = 0$) の状態から S の値を 0 に変化させてみましょう。図 4.32 のように回路の値を順に追っていくと Q と $\overline{Q}$ の値が変化しないことがわかります。すなわち、前の状態が保存され、1 が記憶されたままになっていることがわかります。さらに、**保存状態** ($S = 0$, $R = 0$) からリセット (またはリセット) を行うことで、値を変更することができます。

---

[4]フリップフロップは、前の状態を記憶しておき、その状態が出力に影響を与えることから、順序回路とも呼ばれます。

[5]実際の回路では、構造は複雑になりますが電気的に安定した JK フリップフロップが使用されます。

[6]$\overline{Q}$ は Q の否定で、いつも逆の値が出力されます。

[7]初期状態に設定するには、「セット」または「リセット」のいずれかを行います。

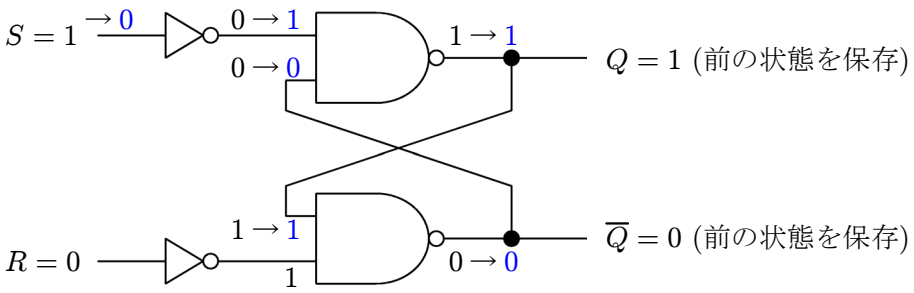


図 4.32: 保存状態 (前の状態を保存)

同様に、**リセット** ($S = 0$, $R = 1$) の状態から R の値を 0 に変化させても、前の状態 ($Q = 0$, $\overline{Q} = 1$) が保存されます (付録 B.1 のチェックシートを利用して各自で確認してください)。

以上をまとめると、表 4.1 のように S と R の値の変化に伴って Q と $\overline{Q}$ の値が変化し、図 4.33 のように状態が推移します。ただし、$S = 1$ かつ $R = 1$ のときは状態が不安定になるので使用しません。

S	R	Q	$\overline{Q}$
0	0	前の Q の値	前の $\overline{Q}$ の値
0	1	0	1
1	0	1	0
1	1	使用禁止	使用禁止

表 4.1: S, R に対する Q, $\overline{Q}$ の値

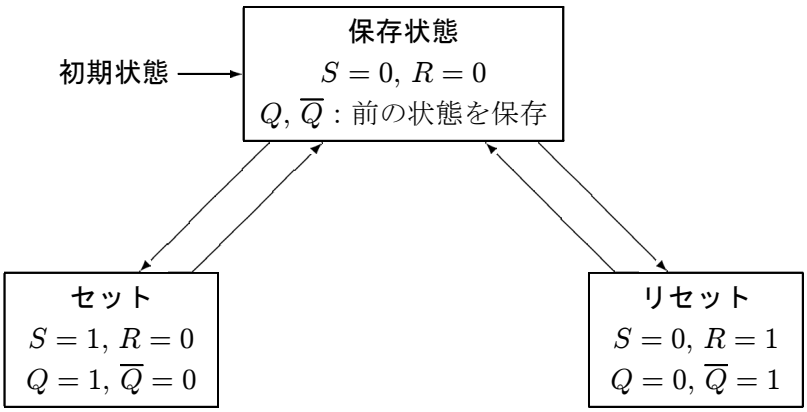


図 4.33: RS フリップフロップ状態推移図

* 付録 B.2 に RS フリップフロップのデジタル回路図を載せておきます。

実際のコンピュータ内部の論理回路では、**クロック**[8](clock) と**同期**を取って値の変更を行います (図 4.34 参照)。なお、クロック C は、コンピュータの動作の基準となる信号 (パルス) で、周期的に 1 と 0 の状態を交互に繰り返します。

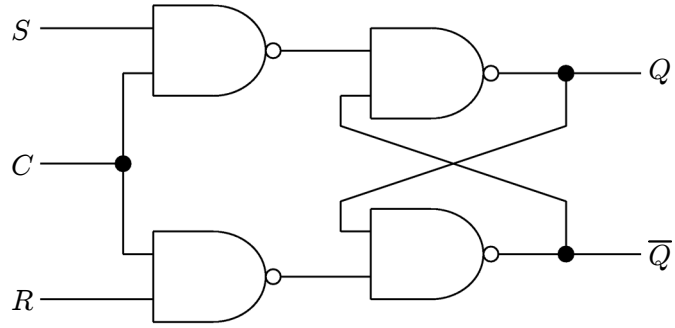
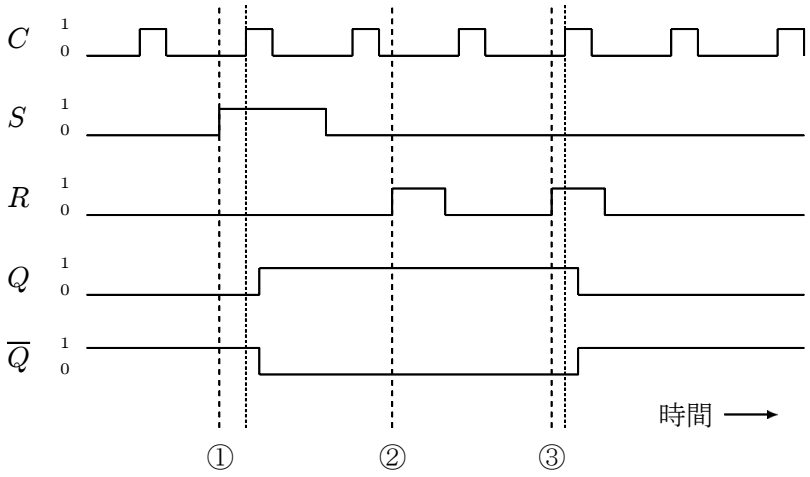


図 4.34: 同期式 RS フリップフロップ

では、クロックと同期した RS フリップフロップの出力の変化を時間的な流れと共に見ていきましょう。ただし、初期状態はリセット ($S=0$, $R=0$, $Q=0$, $\overline{Q}=1$) とします。



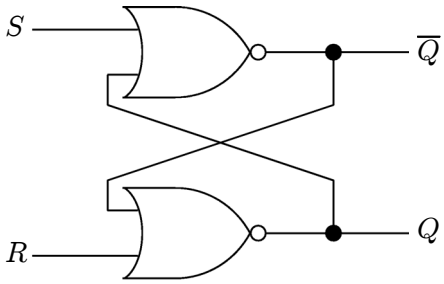
① $Q=0$ の状態からセットを行うと、クロックに合わせて $Q=1$ となります (遅延あり)。
② $Q=1$ の状態からリセットを行うと、クロックと同期しないため Q は変化しません。
③ $Q=1$ の状態からリセットを行うと、クロックに合わせて $Q=0$ となります (遅延あり)。

図 4.35: **タイミングチャート**

この様に、**クロックが 1 の時だけ**、「セット」または「リセット」を行うことができます。クロックが 0 の時は常に保存状態となります。なお、実際の論理回路の出力は、直ぐに値が変わるわけではなく、回路が電気的に安定するまで時間の遅延が生じます。

[8] コンピュータを人間に例えるとクロックは脈のようなもので、一定周期のリズムを刻みながら 1 と 0 の状態を交互に繰り返します。なお、心臓に相当する部分は、クロックジェネレータと呼ばれ、水晶に電気を流すと一定周期のパルスを発生するので、これを合成して方形波を生成します。

問題 1　下図が RS フリップフロップとなることを考察しなさい。



4.4　加算回路

2 章で述べたように、コンピュータの算術演算 (四則演算) の基本は加算 (足し算) とシフト演算です。本節では、論理素子 (MIL 記号) を使って加算回路が構成できること見て行きましょう。

最初に、1 ビット 2 変数の加算を考えると、次の 4 通りになります。

$$\begin{aligned} 0 + 0 &= 0_{(2)} \\ 0 + 1 &= 1_{(2)} \\ 1 + 0 &= 1_{(2)} \\ 1 + 1 &= 10_{(2)} \end{aligned}$$

上記について、2 変数を A, B・和を S・繰り上がりを C として、A と B の全て組み合わせについてまとめると表 4.2 が得られます。この表を真理値表だと考えれば、右下の論理関数と一致することがわかります。

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

$$\Longleftrightarrow \quad \begin{cases} S = A \oplus B \\ C = A \cdot B \end{cases}$$

表 4.2: 1 ビットの加算

この論理関数を論理回路として構成すれば、図 4.36 のような加算回路ができあがります。ただし、この回路は繰り上がりしか考慮されていないため**半加算器**(half adder, HA) と呼ばれています。なお、論理回路の煩雑さを避けるため、図 4.37 のようにブロック図に置き換えて表します。

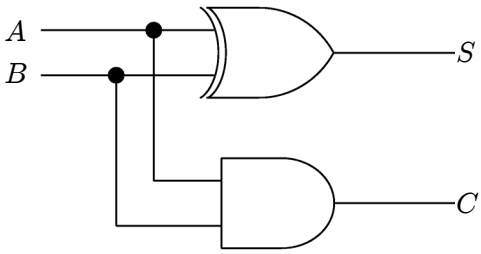


図 4.36: 1 ビットの半加算回路

図 4.37: 半加算器のブロック図

次に、下の桁からの繰り上がりを考慮した加算回路を考えて見ましょう。半加算器に習って、第 i 桁の 2 変数を A_i, B_i・和を S_i・繰り上がりを C_i・下の桁からの繰り上がりを C_{i-1} として、A_i と B_i と C_{i-1} の全ての組み合わせについてまとめると表 4.3 が得られます。第 i 桁の和 S_i は、A_i と B_i の排他的論理和を取ってから、さらに C_{i-1} と排他的論理和をとればよいことがわかります。また、第 i 桁の繰り上がり C_i は、少々複雑ですが、

(「A_i, B_i のどちらか ($A_i \oplus B_i$) が 1」かつ「下の桁からの繰り上がり C_{i-1} が 1」)

または

(「A_i と B_i が共に 1 (C_{i-1} はどちらでもよい)」)

のとき 1 になります。以上より、和 S_i と繰り上がり C_i は右下の論理関数となります。

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

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

表 4.3: 第 i 桁の加算

この論理関数を論理回路として構成すれば、図 4.38 のようになります。この回路は、下の桁からの繰り上がりも考慮され、2 つの半加算器から構成されていることから**全加算器**[9](full adder, FA) と呼ばれています (図 4.39 は半加算器を用いて全加算器を構成したもの)。

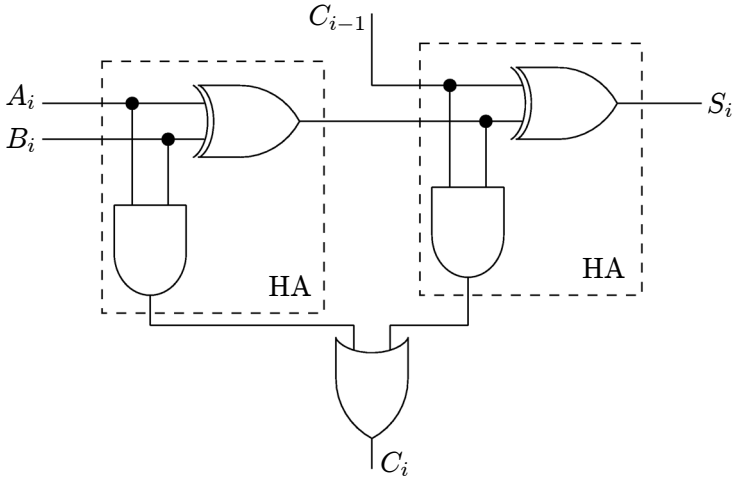


図 4.38: 1 ビットの全加算回路

[9]逆に、半加算器は、機能も回路も全加算器の半分であることからその名がつけられました。

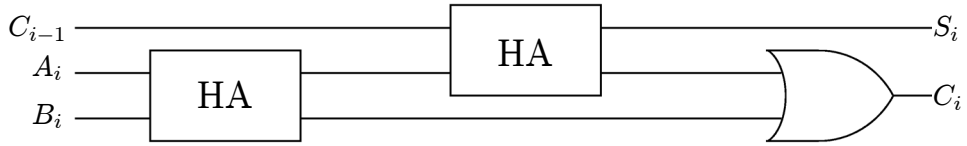


図 4.39: 半加算器によって構成された全加算器

全加算器についても、論理回路の煩雑さを避けるため、図 4.40 のようにブロック図に置き換えて表します。なお、全加算器を 4 つ接続すれば、図 4.41 のように 4 ビット全加算器が構成できます。

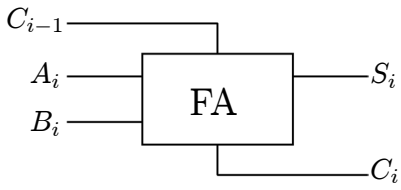


図 4.40: 全加算器のブロック図

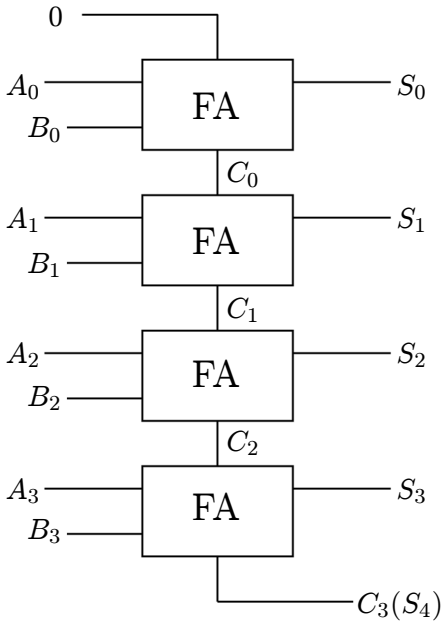


図 4.41: 4 ビット全加算器

* 付録 B.2 に 4 ビット全加算器のデジタル回路図を載せておきます。