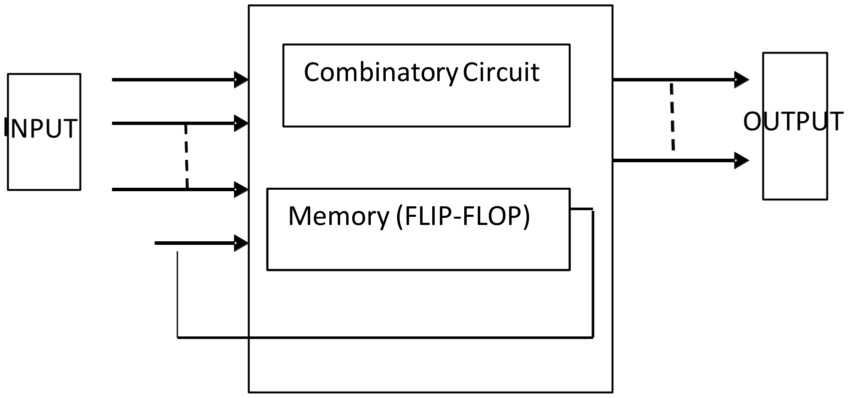
Sequential Circuits

1. Introduction

Combinatory circuits are based on Boolean algebra, while sequential circuits use memory elements called flip-flops in addition to combinatory circuits, adding a time dimension (sequential circuits are often said to remember their previous states).



Sequential Circuit

There are two types of sequential circuits:

-**Asynchronous sequential circuits**: in this type of circuit, the outputs change instantaneously as the inputs change.

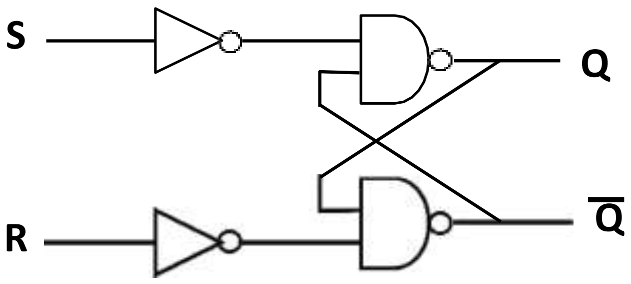
-**Synchronous sequential circuits**: in this type of circuit, the outputs can only change under the control of a clock signal.

2. Bistables (flip-flops)

2.1 RS flip-flop (Asynchronous)

a- The **RS flip-flop** represents a memory equal in size to **one bit**. It has two inputs, **S** and **R**. The S input (**set to one**) sets the flip-flop to one, while the R input (**reset**) sets the flip-flop to zero. The flip-flop also has two complementary outputs $\boldsymbol{Q}$ and $\bar{\boldsymbol{Q}}$. The state of the bit is always Q.

The diagram below illustrates the operation of this flip-flop.

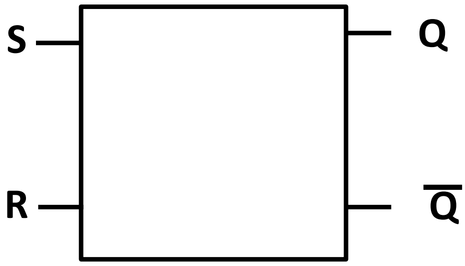


Asynchrone Bistable RS

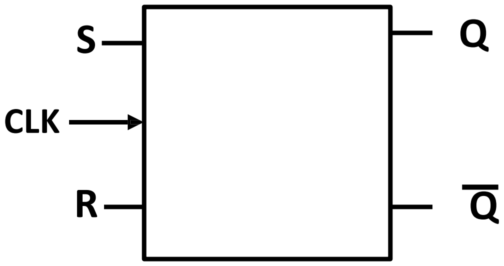
b- How the flip-flop works :

- The future state Q^+ of the RS flip-flop is a Boolean function with parameters **S**, **R** and **Q**.
- If **S=0** and **R=0**, the future state Q^+ of the flip-flop remains equal to **Q** (***bit stored***).
- For **S=0** and **R=1** the future state Q^+ of the flip-flop will be equal to **0** (***clear***)
- For **S=1** and **R=0** the future state Q^+ of the flip-flop would be equal to 1 (***set***)
- For **S=1** and **R=1** this configuration is strictly ***forbidden*** in the latch input.

Note: The synchronous **RS** flip-flop operates under the control of the clock signal, often referred to as "**CLK**". RS flip-flops are generally represented in one of two ways: **synchronous** or **asynchronous**



Bistable RS asynchronous



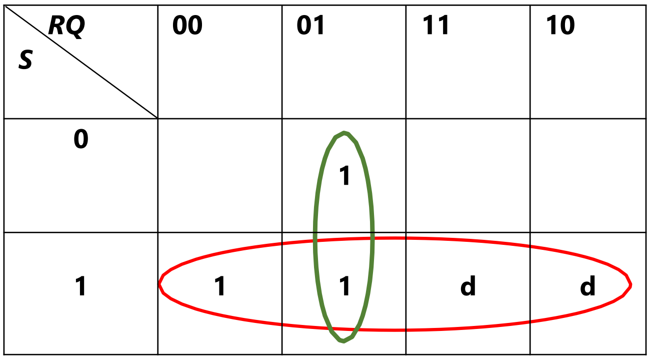
Bistable RS synchronous

c- Characteristic equation of the asynchronous RS flip-flop

The following truth table summarises the operation of the asynchronous RS flip-flop:

S	R	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	d
1	1	1	d

By applying Karnaugh's method for simplifying the Q^+ function



RQ / *S*	00	01	11	10
0		1		
1	1	1	d	d

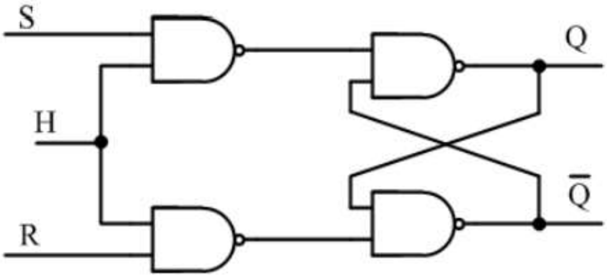
The result is: $Q^+_{RS} = S + \overline{R}Q$

d- Characteristic equation of the synchronous RS flip-flop

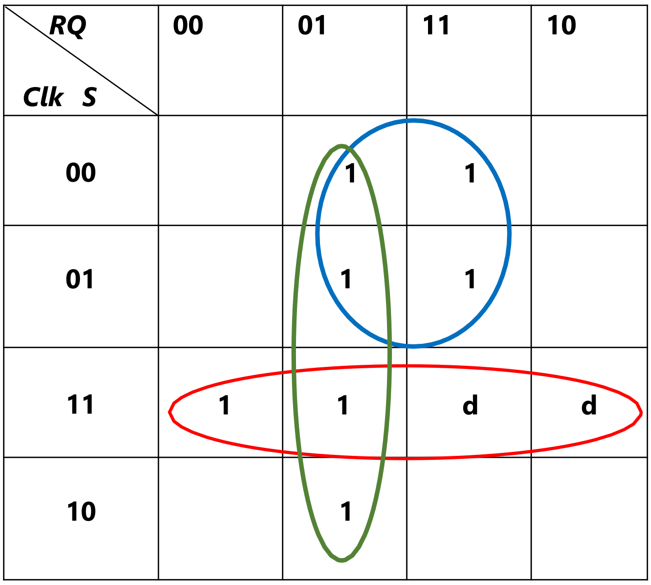
The flip-flop operates under the control of the clock signal **clk**. When the signal **clk=0** the flip-flop operates as a memory element ($Q^+ = Q$) and when the signal **clk=1** the flip-flop operates as an **RS** flip-flop.

The following truth table summarises the operation of the synchronous **RS** flip-flop:

clk	S	R	Q	Q^{+}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	d
1	1	1	1	d



By applying Karnaugh's method for simplifying the function Q^{+}



RQ / Clk S	00	01	11	10
00		1	1	
01		1	1	
11	1	1	d	d
10		1		

The result is : $Q^{+}_{RS} = \overline{clk}\,Q + clk\,S + \overline{R}\,Q$

2.2 JK flip-flop (asynchronous)

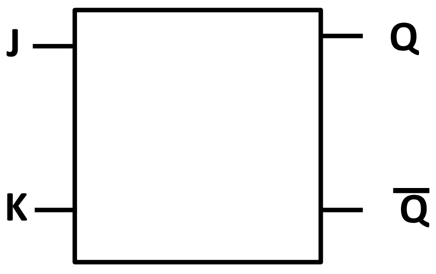
a. The **JK** flip-flop represents a memory of size equal to **one bit**. It has two **inputs J** and **K**, input **J** (**set to one**) for setting the latch to one while input **K** (**reset**) allows the latch to be set to zero. The flip-flop also has two complementary outputs $\boldsymbol{Q}$ and $\boldsymbol{Q}^{+}$. The state of the bit is always $\boldsymbol{Q}$.

b. How the flip-flop works :

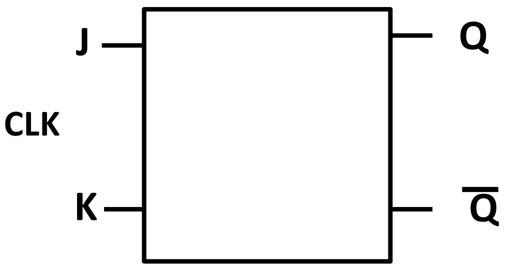
The future state $\boldsymbol{Q}^{+}$of the JK flip-flop is a Boolean function with parameters J, K and $\boldsymbol{Q}$.

- For **J=0** and **K=0**, the future state $\boldsymbol{Q}^{+}$of the flip-flop remains equal to $\boldsymbol{Q}$ (**bit stored**).
- For **J=0** and **K=1** the future state $\boldsymbol{Q}^{+}$of the flip-flop will be equal to **0** (**clear**)
- For **J=1** and **K=0** the future state $\boldsymbol{Q}^{+}$of the flip-flop would be equal to **1** (**set)**
- For **J=1** and **K=1** the future state $\boldsymbol{Q}^{+}$of the latch would be equal to $\bar{\boldsymbol{Q}}$(**invert the bit content**)

Note: **The JK flip-flop presented in this course is simplified for teaching purposes. There are also synchronous JK flip-flops. In general, JK flip-flops are represented in one of two ways: synchronous or asynchronous**.



Bistable JK asynchronous



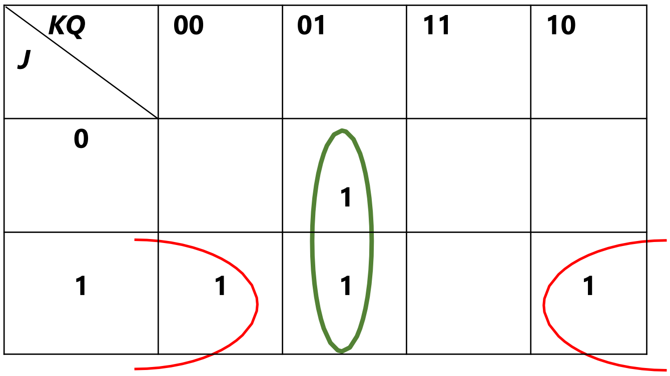
Bistable JK synchronous

C- Characteristic equation of the asynchronous JK flip-flop

The following truth table summarises the operation of the asynchronous JK flip-flop:

J	K	Q	Q^{+}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

By applying Karnaugh's method for simplifying the function Q^{+}



KQ / J	00	01	11	10
0		1		
1	1	1		1

The result is : $Q^{+}_{JK}= J\bar{Q} + \overline{K}\,Q$

d-Characteristic equation of the synchronous JK flip-flop

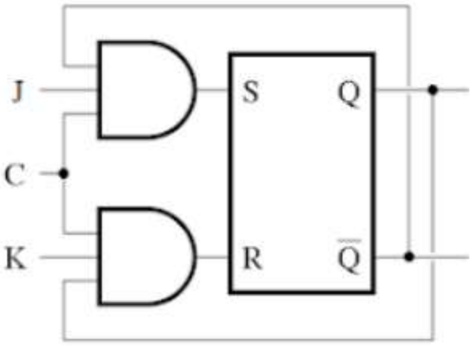
The flip-flop operates under the control of the clock signal **clk**. When the signal **clk=0** the flip-flop operates as a memory element$(Q^{+} = Q)$ and when the signal **clk=1** the flip-flop operates as a **JK** flip-flop.

The following truth table summarises the operation of the synchronous **JK** flip-flop:

clk	J	K	Q	Q^+
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

By applying Karnaugh's method for simplifying the function Q^+

KQ / ***Clk J***	**00**	**01**	**11**	**10**
00		**1**	**1**	
01		**1**	**1**	
11	**1**	**1**		**1**
10		**1**		



The result is : $Q^+_{JK} = \overline{clk}\, Q + clk\, J\overline{Q} \;\; + \overline{K}\, Q$

2-3 T (Trigger) flip-flop

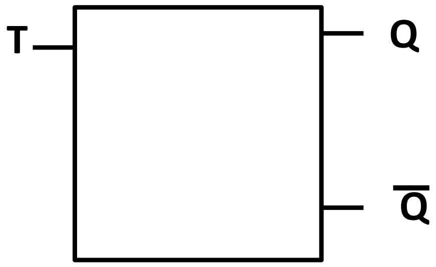
a. The **T** flip-flop represents a memory of size equal to **one Bit**. It has one **input T** and two complementary **outputs** $\boldsymbol{Q}$ and $\bar{\boldsymbol{Q}}$.

b. How the flip-flop works :

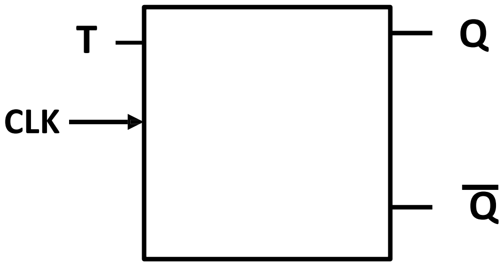
The future state Q^{+} of the flip-flop **T** is a Boolean function of parameters **T** and Q.

- For **T=0** the future state Q^{+} of the latch remains equal to Q (**bit memorised**).
- For T=1 the future state $Q^{+}$of the flip-flop will be equal to $\bar{Q}$(**bit content inverted**)

> Note:
> There are also synchronous T flip-flops. Generally, T flip-flops are represented in one of two ways: synchronous or asynchronous.



Bistable T asynchronous



Bistable T synchronous

c- Characteristic equation of the asynchronous T flip-flop

The following truth table summarises the operation of the asynchronous T flip-flop:

T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

From the truth table we can deduce the characteristic equation of the flip-flop **T** :

$$\boldsymbol{Q^+_T = T\bar{Q} + \bar{T}Q} \quad \text{thus} \quad \boldsymbol{Q^+_T = T \oplus Q}$$

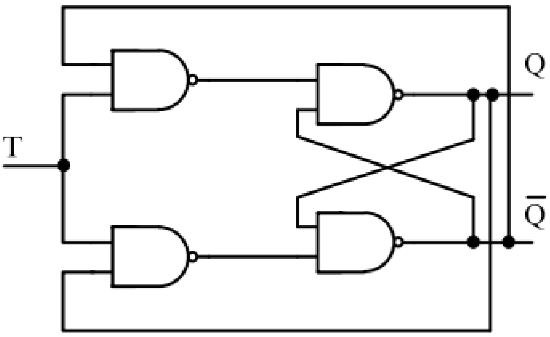
d- Characteristic equation of the synchronous T flip-flop

The following truth table summarises the operation of the asynchronous T flip-flop:

clk	T	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

By applying Karnaugh's method for simplifying the function $\boldsymbol{Q^+}$

TQ / *clk*	00	01	11	10
0		1	1	
1		1		1



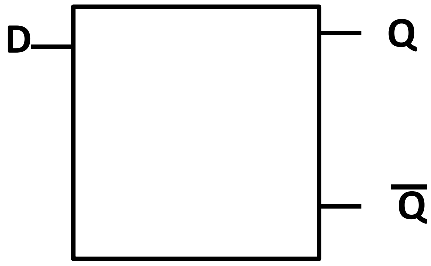
The result is : Q_{T}^{+} = $\bar{T}Q + \overline{clk}\,Q + clk\,T\bar{Q}$

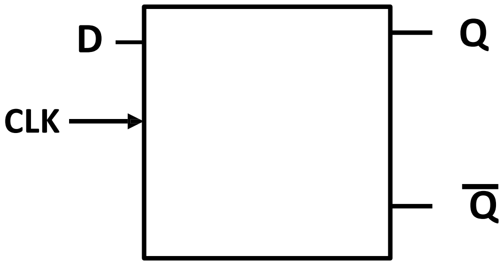
2.4 D (Data) Flip-flop

a. The **D** flip-flop represents a memory of size equal to **one Bit**. It has one **D input** and two complementary outputs Q and $\bar{Q}$.The state of the bit is always Q.

b. How the flip-flop works :

The future state $Q^{+}$of the **D** flip-flop is a Boolean function with parameters **D**,**Q** and **clk** for the **synchronous** flip-flop. On the other hand, the future state $Q^{+}$of the **D** flip-flop depends only on the **D** input for the asynchronous flip-flop.





Bistable T asynchrone

Bistable T synchrone

c- Characteristic equation of the asynchronous D flip-flop

$$Q_D^+ = D$$

d- Characteristic equation of the synchronous D flip-flop

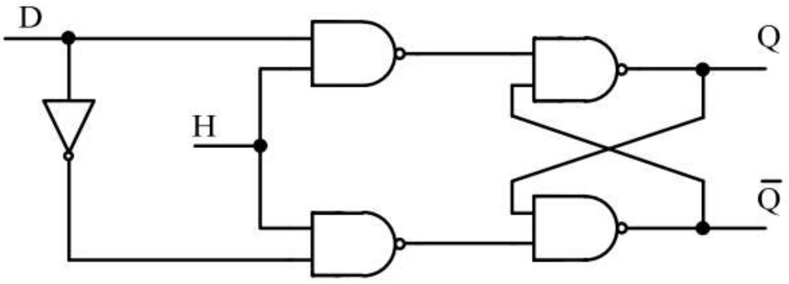
The following truth table summarises the operation of the synchronous D flip-flop:

clk	D	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

By applying Karnaugh's method for simplifying the function Q^+

DQ ***clk***	**00**	**01**	**11**	**10**
0		**1**	**1**	
1			**1**	**1**

The result is : $Q_D^+ = \overline{clk}\,Q + clk\,D.$



3 Sequential circuits

A sequential circuit consists of a combinatory part and memory elements (flip-flops). The study of combinatory circuits is based on Boolean algebra. The study of sequential circuits is based on the theory of finite state automata.

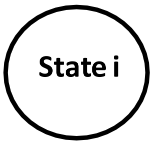
3-1 Finite state automata

In this chapter, we introduce finite state automata for the purpose of synthesising and analysing sequential circuits.

a- Principle:

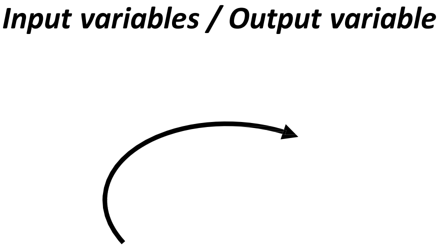
- An automaton is made up of a set of pairs (state, transition): The state of an automaton represents the state (content) of the circuit's memories. Since the memory elements are flip-flops (bits), the maximum **number of states** in a sequential circuit containing **K flip-flops** is **2^K**.

  Graphically, a state is represented by a circle containing a label representing the name of the state.



- A transition in the automaton is a directed arc linking two states (the arc transits from the start state with which it is associated to the end state).

  Graphically, this arc is a one-way arrow labelled with the circuit's input variables and possibly output variables.



b. Illustrative examples:

b-1 - Automaton associated with the T flip-flop

We assume that the sequential circuit is an asynchronous flip-flop T. The truth table associated with this flip-flop is :

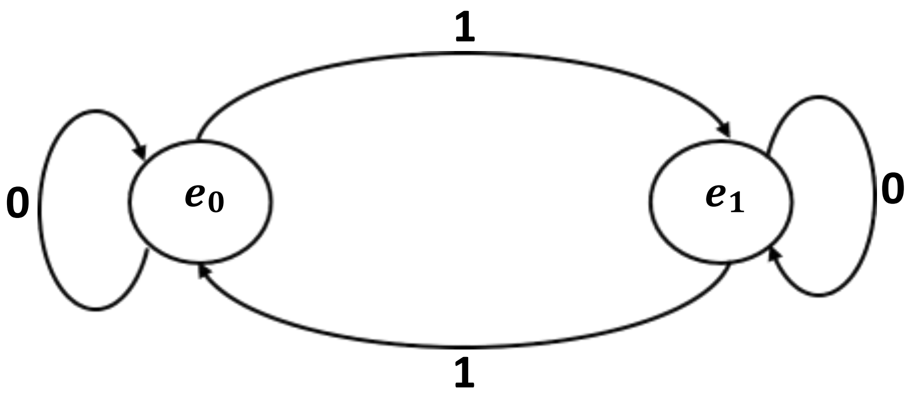
T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

The circuit contains one bit, so the automaton will have a maximum of two states.

We call $\mathbf{e_0}$ the state of the automaton associated with the state **Q = 0** of the flip-flop and $\mathbf{e_1}$ the state of the automaton associated with the state **Q = 1** .

According to the truth tale :

- If the automaton is in state $\mathbf{e_0}$ and **T=0** (input variable) then the future state is $\mathbf{e_0}$. In this case there are no output variables. The output is confused with the future state of the flip-flop.
- If the automaton is in state $\mathbf{e_0}$ and **T=1** (input variable) then the future state is $\mathbf{e_1}$.
- If the automaton is in state $\mathbf{e_1}$ and **T=0** then the future state is $\mathbf{e_1}$.
- If the automaton is in state $\mathbf{e_1}$ and **T=1**, then the future state is $\mathbf{e_0}$.



Automaton describing the operation of the flip-flop **T**

b-2 - Automaton associated with the JK flip-flop

We assume that the sequential circuit is an asynchronous **JK flip-flop**. The truth table associated with this **flip-flop** is :

J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

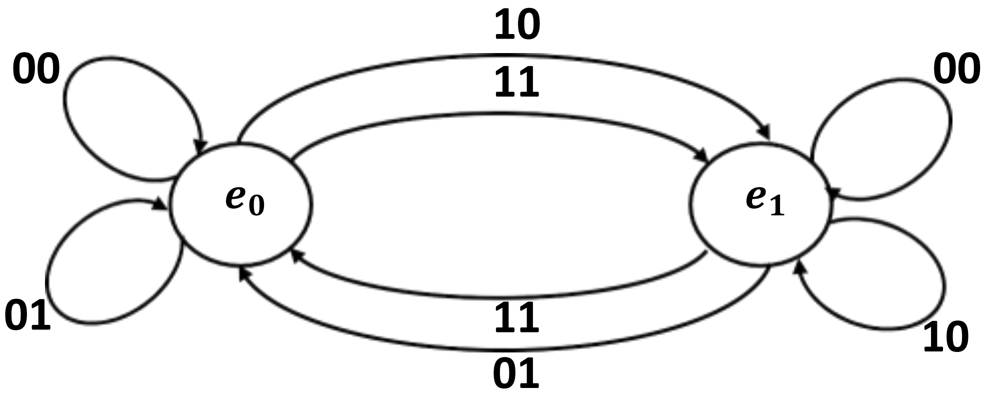
The circuit contains one bit, so the automaton will have a maximum of two states.

We call $\boldsymbol{e_0}$ the state of the automaton associated with the state **Q = 0** of the flip-flop and $\boldsymbol{e_1}$ the state of the automaton associated with the state **Q = 1** .

According to the truth tale :

- If the automaton is in state $\boldsymbol{e_0}$ and **JK=00** (input variable) then the future state is $\boldsymbol{e_0}$ (storage). In this case there are no output variables. The output is confused with the future state of the flip-flop.
- If the automaton is in state $\boldsymbol{e_0}$ and **JK=01**, then the future state is $\boldsymbol{e_0}$ (the contents of the flip-flop are erased, future state $Q^+ = \mathbf{0}$).
- If the automaton is in state $\boldsymbol{e_0}$ and **JK=10**, then the future state is $\boldsymbol{e_1}$ (set the flip-flop to **1**, future state $Q^+ = \mathbf{1}$).
- If the automaton is in state $\boldsymbol{e_0}$ and **JK=11** then the future state is$\boldsymbol{e_1}$. (invert the contents of the flip-flop, future state $Q^+ = \mathbf{1}$)
- If the automaton is in state $\boldsymbol{e_1}$and **JK=00** then the future state is$\boldsymbol{e_1}$.
- If the automaton is in state $\boldsymbol{e_1}$and **JK=01** then the future state is $\boldsymbol{e_0}$.(clearing the contents of the flip-flop, future state $Q^+ = \mathbf{0}$)

- If the automaton is in state $\boldsymbol{e_1}$and **JK=10**, then the future state is $\boldsymbol{e_1}$ (set the flip-flop to **1**, future state $\boldsymbol{Q^+ = 1}$).
- If the automaton is in state $\boldsymbol{e_1}$ and **JK=11** then the future state is $\boldsymbol{e_0}$.(invert the contents of the flip-flop, future state $\boldsymbol{Q^+ = 0}$)



Automaton describing the operation of the flip-flop **JK**

3-2 Synthesis of sequential circuits

Creating a sequential circuit to solve a problem involves applying a synthesis procedure as follows:

a. Understand the problem.

b. Synthesise any combinatory part of the circuit.

c. Create the finite state automaton that solves the problem.

d.Determine the number of flip-flops to be integrated into the sequential circuit.

e.Transform the automaton into a table of transitions and future states (a truth table), based on the characteristic tables of the flip-flops used in the circuit.

f.Simplify the functions relating to the triggering of the flip-flops and any functions relating to the outputs of the circuit.

g. Draw the sequential circuit.

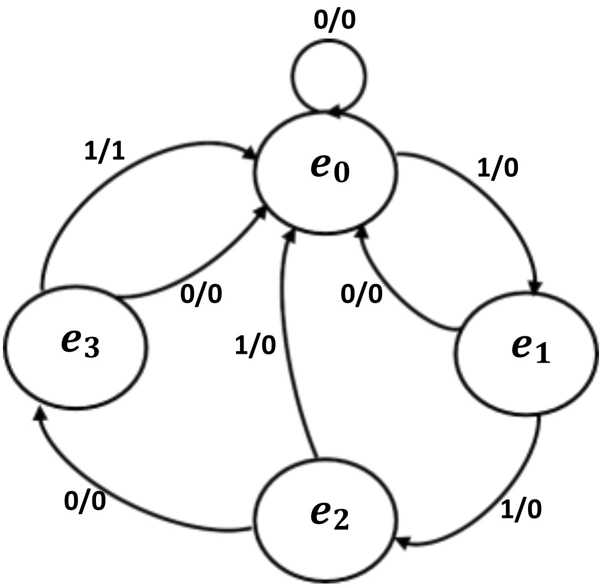
Example:

Make a sequential circuit that recognises the string **1011** with overlay (the string is read bit/bit from right to left), the output is equal to 1 if the string is recognised (use **JK** flip-flops).

1 In this example, the circuit does not include an independent combinatory part.

2 Creation of the finite state automaton to solve the problem:

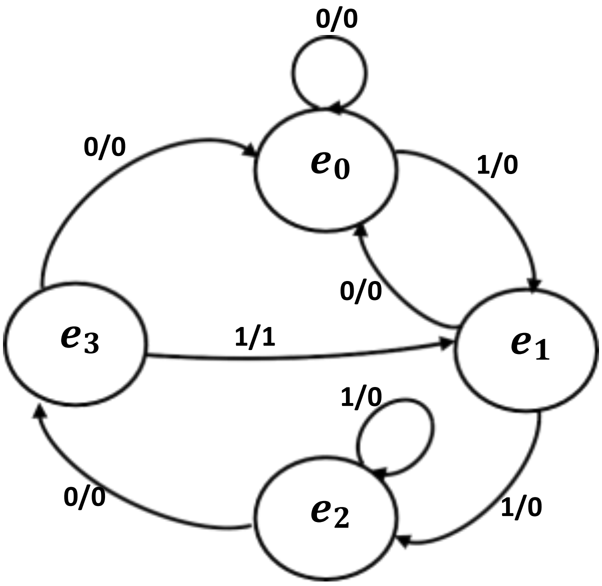
- Each bit in the chain will be represented by states in the automaton. The automaton will be modelled by four(**4**) states (number of states **n=4**).
- To draw the raw automaton :
- Each bit in the string read is associated with a state responsible for recognising that bit.
- The first bit on the right will be associated with state $\mathbf{e_0}$ (101**1**): state $\mathbf{e_0}$ will be responsible for recognising the reading of the first "**1**" in the string and will hand over to state $\mathbf{e_1}$ for recognising the second bit in the string (10**1**1).
- The transition from $\mathbf{e_0}$ to $\mathbf{e_1}$ will be labelled **I/O** (**I** for the bit read and **O** for the output), the output **S=1** only if the string is completely read.
- If en is in a state $\mathbf{e_i}$ of the automaton and the bit read does not correspond to the bit in the string, then en must restart from the initial state $\mathbf{e_0}$.
- Diagram of the crude automaton:



Note:

the crude automaton developed above does not meet the description of the problem posed. In fact, the crude automaton does not achieve the required overlap (a new correct string can start before the end of a string currently being recognised). In order to create the sequential circuit, the crude automaton must be corrected to solve the problem.

Diagram of the improved Automaton



3 Determining the number of flip-flops required to complete the circuit :

- The number of flip-flops **P** is given by the formula: : $\mathbf{2^{P-1} < n <= 2^{P}}$

Where: **n** represents the number of states in the automaton.

- In the example shown, $\mathbf{2^{P-1} < 4 <= 2^{P}}$ ⇒ **P=2**..
- The circuit then consists of two **JK** type flip-flops (the type of flip-flop is indicated in the problem statement).

4 Building the table of future states, transitions and outputs (truth table) :

To construct this table we will need to encode the states of the automaton using the states of the flip-flops in the circuit. We will also use the transition state tables of the asynchronous **JK flip-flop** to construct the truth table.

The circuit has **two JK** flip-flops:

$\boldsymbol{Q_1}$represents the state of the first flip-flop and $\boldsymbol{Q_2}$ represents the state of the second flip-flop.

Thus:

Automaton State	Q_1Q_2
$\mathbf{e_0}$	00
$\mathbf{e}_1$	01
$\mathbf{e}_2$	10
$\mathbf{e}_3$	11

- **JK flip-flop transition table:**

Q	Q^+	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

- **Table of future states, transitions and outputs :**

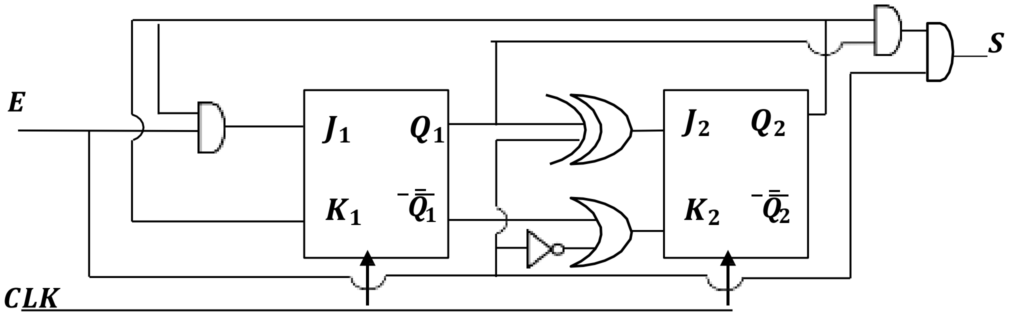
Entrée	***Etat***		***Etat futur***		***Entrée bascule 1***		***Entrée bascule2***		***Sortie***
E	Q_1	Q_2	Q_1^+	Q_2^+	J_1	K_1	J_2	K_2	S
0	0	0	0	0	0	d	0	d	0
0	0	1	0	0	0	d	d	1	0
0	1	0	1	1	d	0	1	d	0
0	1	1	0	0	d	1	d	1	0
1	0	0	0	1	0	d	1	d	0
1	0	1	1	0	1	d	d	1	0
1	1	0	1	0	d	0	0	d	0
1	1	1	0	1	d	1	d	0	1

The variables **J1, K1, J2, K2** and **S** will be considered as functions in the truth table and will be simplified using Karnaugh's method.

5- Simplification:

$$J_1 = EQ_2\ ;\quad K_1 = Q_2;\quad J_2 = E \oplus Q_1\ ;\quad K_2 = \overline{E} + \overline{Q_1}\ ;\ S = EQ_1Q_2$$

6- The sequential circuit


E
J_1
Q_1
K_1
$\bar{Q}_1$
J_2
Q_2
K_2
$\bar{Q}_2$
S
CLK