Solution : SERIE T.D N°04
Sequential Circuits

Answers to course questions

Q1: In Sequential Logic Circuits, we have the presence of memory elements: A sequential logic circuit can be defined as a combinational circuit incorporating memory elements. It differs from combinational logic circuits in that its outputs depend not only on the inputs (combinational) but also on the state of its outputs.

Q2 : Here are examples of flip-flops (check the correct answers) :

☑ RS ☑ D □ CLC □ CLS ☑ JK □ JSK □ RAM ☑ T

Q3 : Two types of sequential logic circuits are distinguished : □ Asynchronous □ synchronous

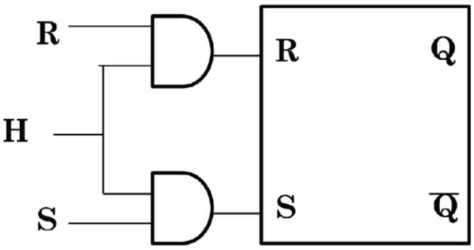
Q4 : In an asynchronous logic circuit, a clock signal is used to coincide with the moments of writing information into the flip-flops: □ True ☑ False

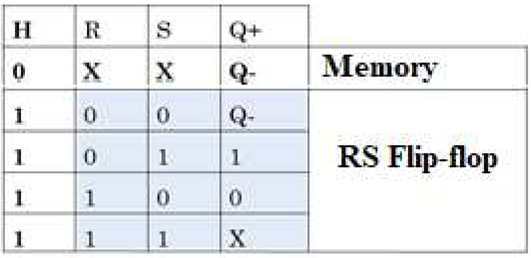
Q5 : In a synchronous logic circuit, a clock signal is used to synchronize the moments of state change of the flip-flops : ☑ True □ False

<u>Exercise 01</u> (solution is found at the end)

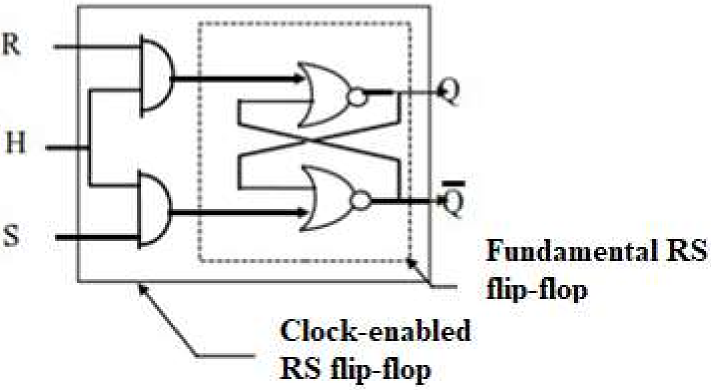
<u>Exercise 02 : (Solution)</u>

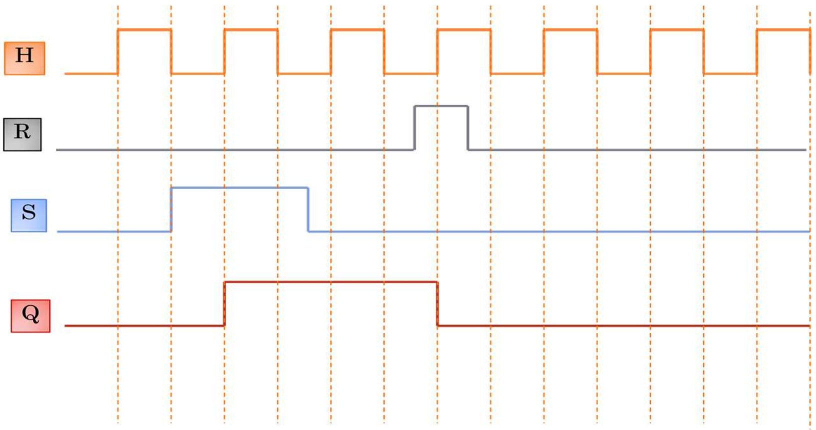
1- Simply perform a logical AND between its inputs and the clock signal, **<u>to ensure that the consideration</u>** of its inputs is synchronized with the clock signal.





H	R	S	Q+	
0	X	X	Q-	Memory
1	0	0	Q-	RS Flip-flop
1	0	1	1	
1	1	0	0	
1	1	1	X	

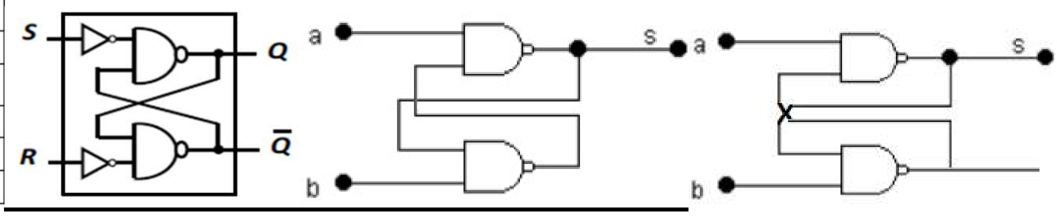




Exercise 03 : (Solution)

1- The truth table:

a	b	S	S^+
0	0	0	x
0	0	1	x
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



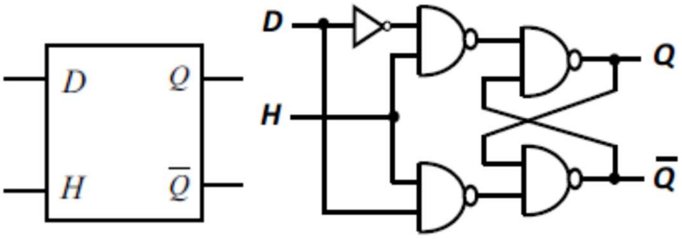
2- It is the Flip-flop SR with $R = a$ *and* $S = b$.

Exercise 04 : (Solution)

1- The D flip-flop is an RSH flip-flop where the R and S inputs are complemented to eliminate the forbidden state (R=S=1) and minimize the number of memory states.

- When H is active and D=1, the output Q will be set to 1.
- When H is active and D=0, the output Q will be reset to 0.
- When H is not active, the outputs remain in the previous state (memory).

D	Q_{n+1}	
	H active	H Inactive
0	0	Q_n
1	1	Q_n



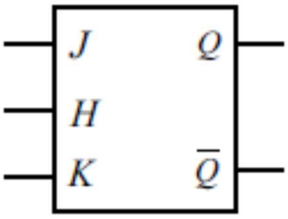
H	D	Q+
0	0	Q-
0	1	Q-
1	0	0
1	1	1

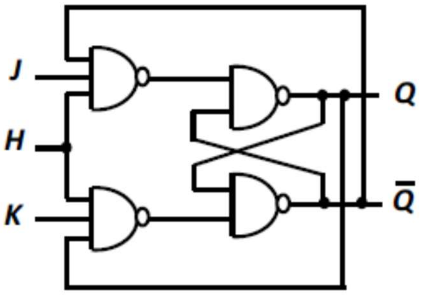
H	R	S	Q+
0	**X**	**X**	**Q-**
1	0	0	Q-
1	0	1	1
1	1	0	0
1	1	1	X

$$H_D = H_{RSH},\ R = \overline{D}\ ;\ S = D$$

1- The JK flip-flop is an RSH flip-flop where the forbidden state (R=S=1) is eliminated by feedback looping the outputs to the inputs.
2- When H is active and J=1 and K=0, the output Q will be set to 1.
3- When H is active and J=0 and K=1, the output Q will be reset to 0.
4- When H is active and J=1 and K=1, the output Q will be toggled.
5- When H is not active or J=0 and K=0, the outputs remain in the previous state (memory).

J	K	Q_{n+1}	
		H active	**H Inactive**
0	0	Q_n	Q_n
0	1	0	Q_n
1	0	1	Q_n
1	1	$\overline{Q_n}$	Q_n



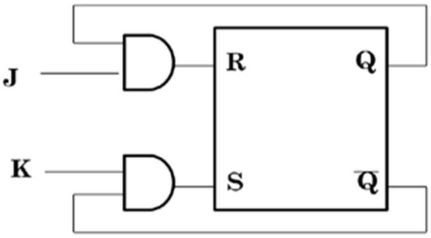


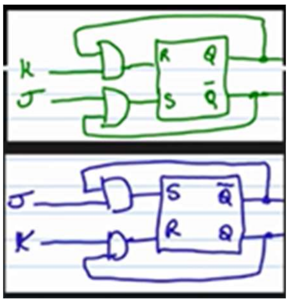
The truth table				
H	**J**	**K**	**Q**	**Comment**
↑	0	0	q	**Memory**
↑	0	1	0	**Resetting the output Q**
↑	1	0	1	**Setting the output Q to 1**
↑	1	1	$\bar{q}$	**Toggling the output state**

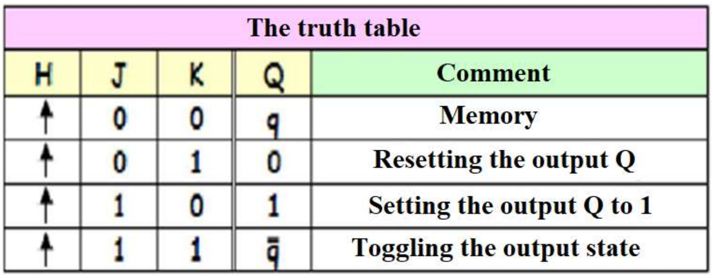
1- The asynchronous JK flip-flop using an RS flip-flop.

R	S	Q-	Q+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	**1**	**0**	**X**
1	**1**	**1**	**X**

J	K	Q-	Q+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**0**







The truth table				
H	J	K	Q	Comment
↑	0	0	q	Memory
↑	0	1	0	Resetting the output Q
↑	1	0	1	Setting the output Q to 1
↑	1	1	$\bar{q}$	Toggling the output state

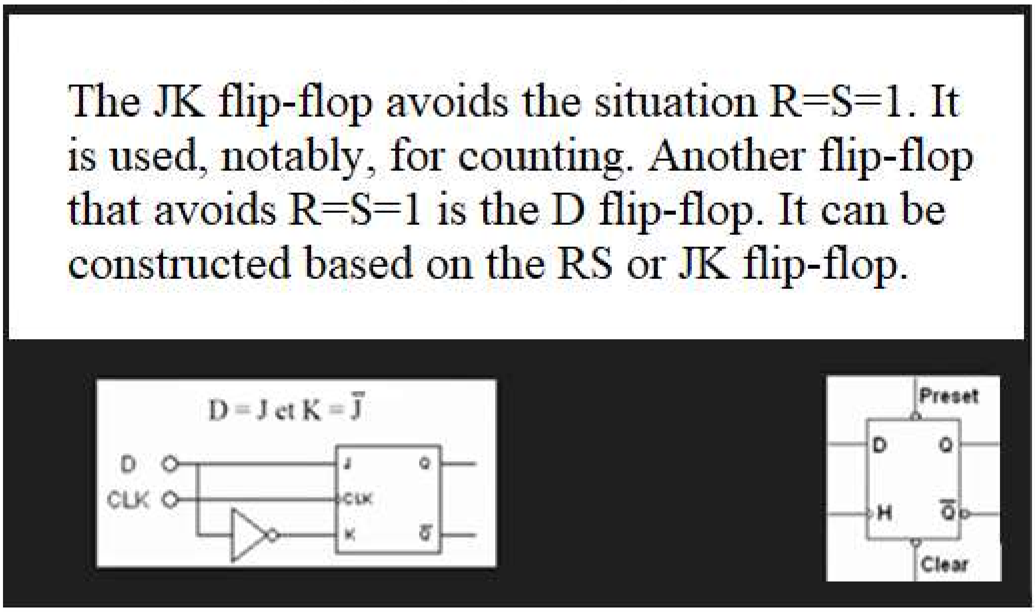
Exercise 05 :(Solution)

1-

H	D	Q+
0/1	0	Q-
0/1	1	Q-
↑	0	0
↑	1	1

H	J	K	Q+
0/1	X	X	Q-
↓	0	0	Q-
↓	0	1	0
↓	1	0	1
↓	1	1	$\overline{Q}$-

$$H_{JK} = \overline{H}_D,\ J = D\ ;\ K = \overline{D}$$



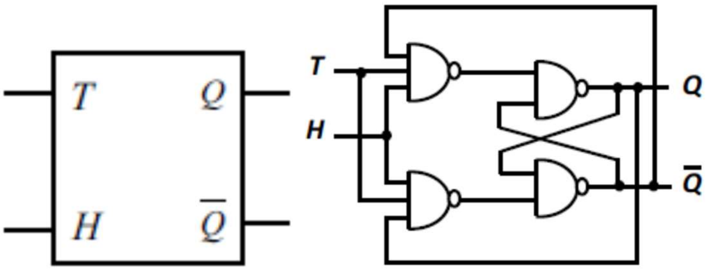
The JK flip-flop avoids the situation R=S=1. It is used, notably, for counting. Another flip-flop that avoids R=S=1 is the D flip-flop. It can be constructed based on the RS or JK flip-flop.

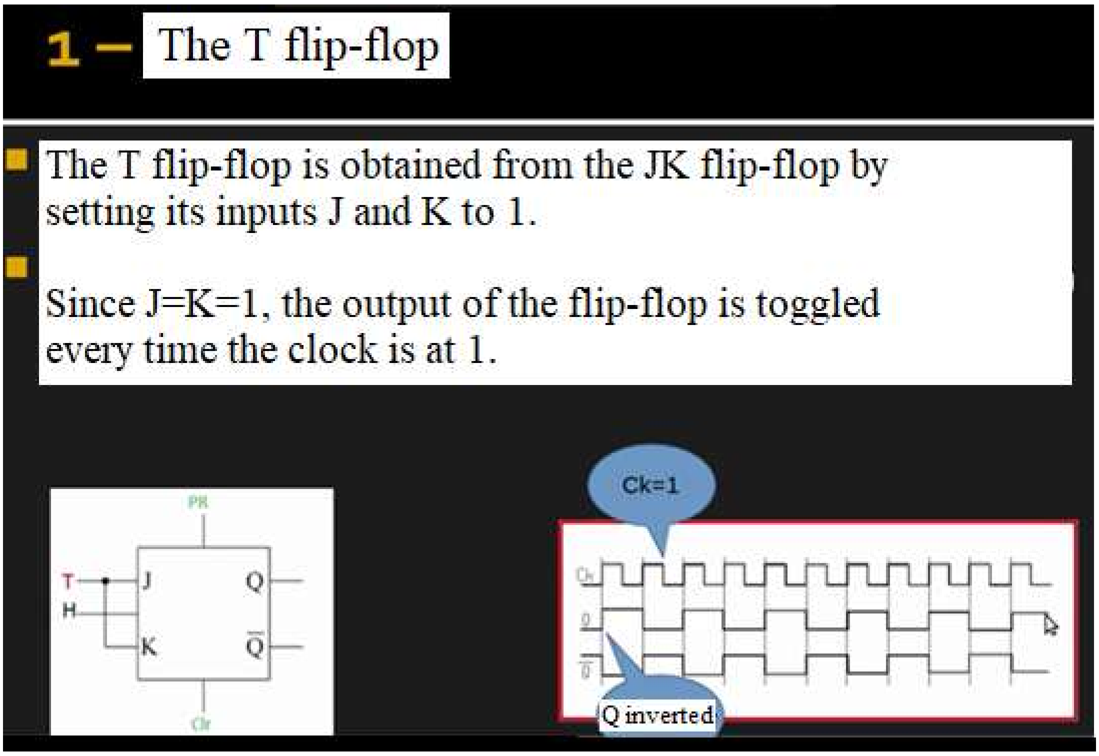
D = J et K = J̄
D
CLK
J
CLK
K
Q
Q̄
Preset
D
Q
H
Q̄
Clear

2- The T flip-flop is a flip-flop with a single input T. It is essentially a JK flip-flop with both inputs connected together

- When H is active and T=1, the output Q will be toggled.
- When H is not active or T=0, the outputs remain in the previous state (memory)

T	Q_{n+1}	
	H active	**H non active**
0	Q_n	Q_n
1	$\overline{Q_n}$	Q_n





1 – The T flip-flop

- The T flip-flop is obtained from the JK flip-flop by setting its inputs J and K to 1.
- Since J=K=1, the output of the flip-flop is toggled every time the clock is at 1.

T	H	Q+
0	**X**	**Q-**
1	0/1, ↑	Q-
1	↓	$\overline{\textbf{Q-}}$

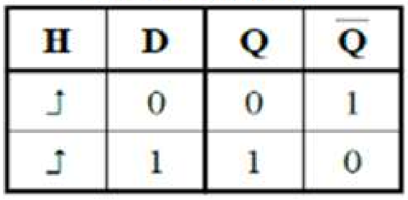
H	J	K	Q+
0/1, ↑	**X**	**X**	**Q-**
↓	0	0	Q-
↓	0	1	0
↓	1	0	1
↓	**1**	**1**	$\overline{\textbf{Q-}}$

$$H_T = H_{JK},\ J = K = T$$

Additional Exercises (Solution)

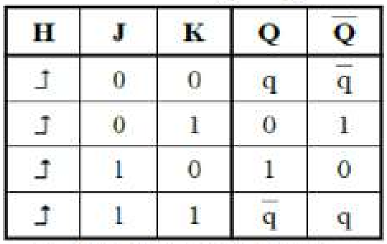
Exercise 06 : (Solution)

The D Flip-flop



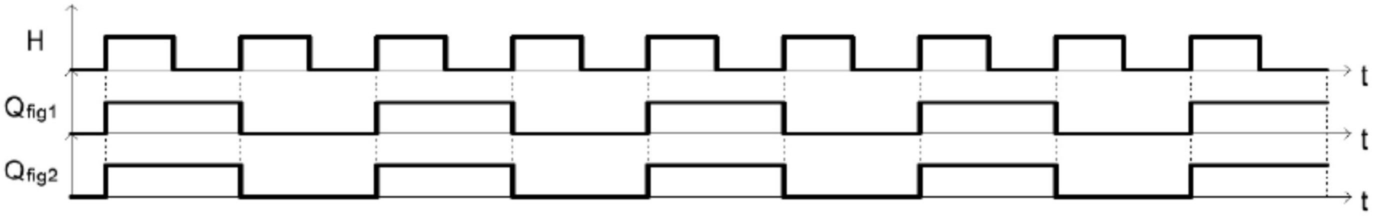
H	D	Q	$\overline{Q}$
↑	0	0	1
↑	1	1	0

The JK Flip-flop



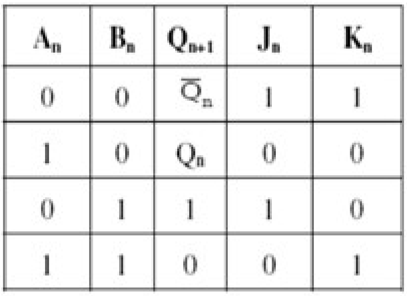
H	J	K	Q	$\overline{Q}$
↑	0	0	q	$\overline{q}$
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{q}$	q

q = the value of Q just before
the rising edge of the clock input.

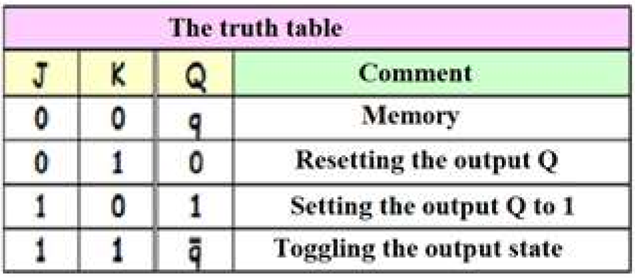


Exercise 07 (Solution) :

1- From the truth table of the AB flip-flop, we can define the values that the inputs J and K must have to provide the expected result: (Reminder of the truth table of the JK flip-flop).



A_n	B_n	Q_{n+1}	J_n	K_n
0	0	$\overline{Q}_n$	1	1
1	0	Q_n	0	0
0	1	1	1	0
1	1	0	0	1



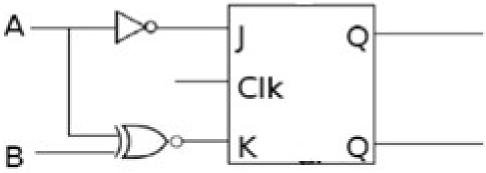
The truth table			
J	K	Q	Comment
0	0	q	Memory
0	1	0	Resetting the output Q
1	0	1	Setting the output Q to 1
1	1	$\overline{q}$	Toggling the output state

The expressions of J and K are deduced based on A and B:

$$\boxed{J = \bar{A}\bar{B} + \bar{A}B = \bar{A}}$$

$$\boxed{Q = \bar{A}\bar{B} + AB = \overline{A \oplus B}}$$

Hence, the logic diagram :



2- AB flip-flop based on a 4-to-1 MUX:

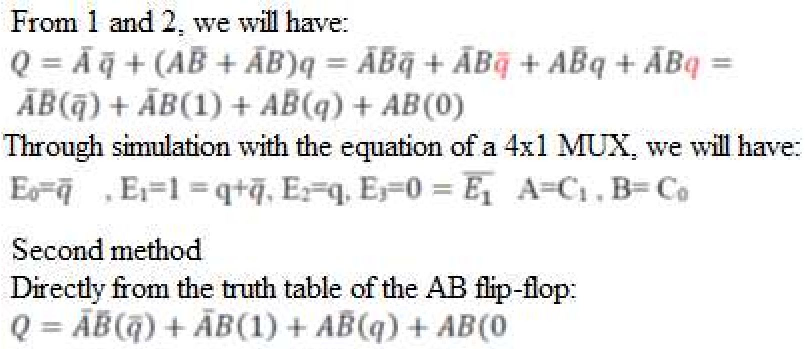
First method :
From the question 1:

$$\boxed{J = \bar A\bar B + \bar A B = \bar A}$$

$$\boxed{Q = \bar A\bar B + AB = \overline{A \oplus B}}$$ ……………….….(1)

And according to the characteristic equation of the JK flip-flop:

$Q=J\bar q+\bar K q$…………………… (2)



From 1 and 2, we will have:

$Q = \bar A\,\bar q + (A\bar B + \bar A B)q = \bar A\bar B\bar q + \bar A B\bar q + A\bar B q + \bar A B q =$
$\bar A\bar B(\bar q) + \bar A B(1) + A\bar B(q) + AB(0)$

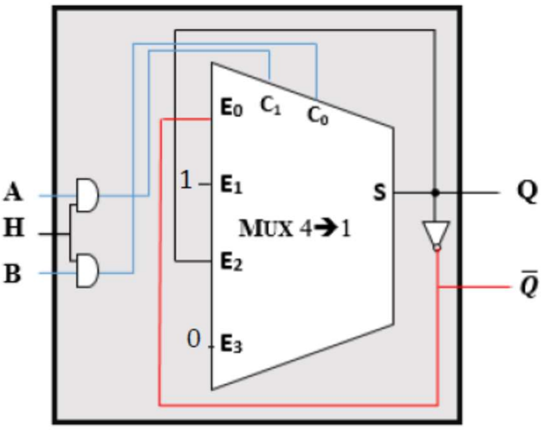
Through simulation with the equation of a 4x1 MUX, we will have:

$E_0=\bar q$, $E_1=1 = q+\bar q$, $E_2=q$, $E_3=0 = \overline{E_1}$ $A=C_1$, $B= C_0$

Second method

Directly from the truth table of the AB flip-flop:

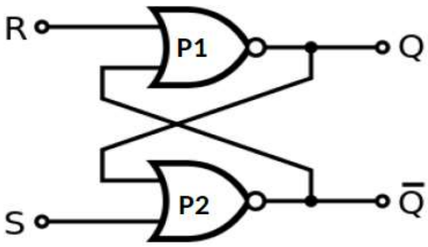
$Q = \bar A\bar B(\bar q) + \bar A B(1) + A\bar B(q) + AB(0$

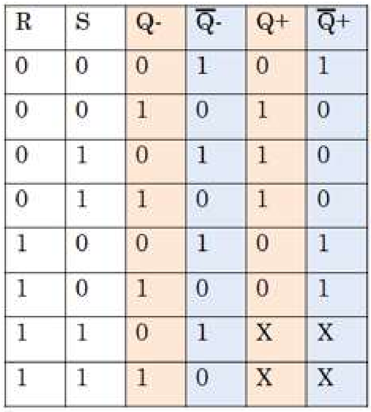


Exercise 01 (solution) :

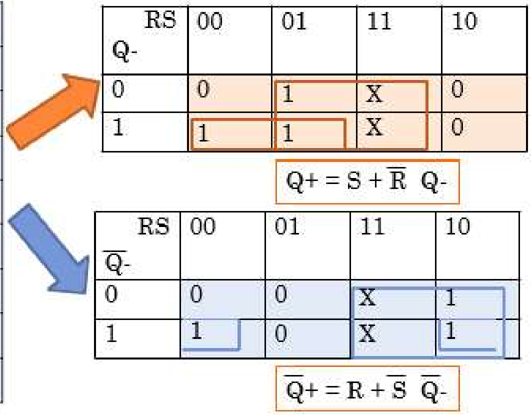
RS flip-flop (RS latch): The RS bistable (flip-flop) is a logic circuit capable, under certain circumstances, of maintaining the values of its outputs despite changes in input values, that is, of memorizing its state 'memory.' It is the storage element of sequential logic. It is controlled by two inputs, S and R, and has two outputs, Q and $\overline{Q}$. The S input (Set) allows the output Q to be set to 1 (set), while R (Reset) allows it to be reset to 0 (clear). The most commonly encountered latches are implemented with two NOR or NAND gates.

RS with only NOR gates





R	S	Q-	$\overline{Q}$-	Q+	$\overline{Q}$+
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	X	X
1	1	1	0	X	X



RS Q-	00	01	11	10
0	0	1	X	0
1	1	1	X	0

$Q+ = S + \overline{R}\ Q-$

RS $\overline{Q}$-	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$\overline{Q}+ = R + \overline{S}\ \overline{Q}-$

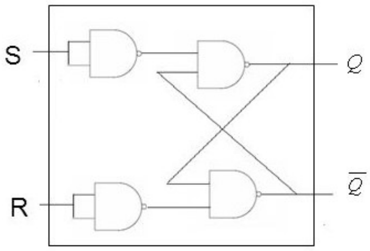
RS only with NAND gates

$$Q^{+} = S + \overline{R}.Q$$

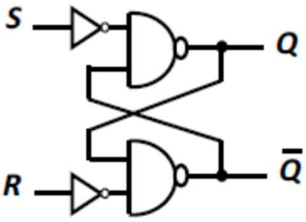
$$\overline{Q^{+}} = R + \overline{S}.Q$$

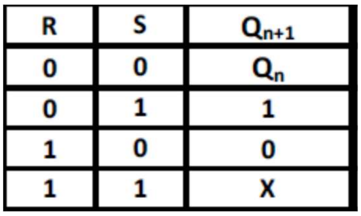
$$Q^{+} = S + \overline{R}.Q = \overline{\overline{S + \overline{R}.Q}} = \overline{S} \uparrow (\overline{R} \uparrow Q) = (S \uparrow S) \uparrow ((R \uparrow R) \uparrow Q)$$

$$\overline{Q^{+}} = R + \overline{S}.Q = \overline{\overline{R + \overline{S}.Q}} = \overline{R} \uparrow (\overline{S} \uparrow Q) = (R \uparrow R) \uparrow ((S \uparrow S) \uparrow Q)$$



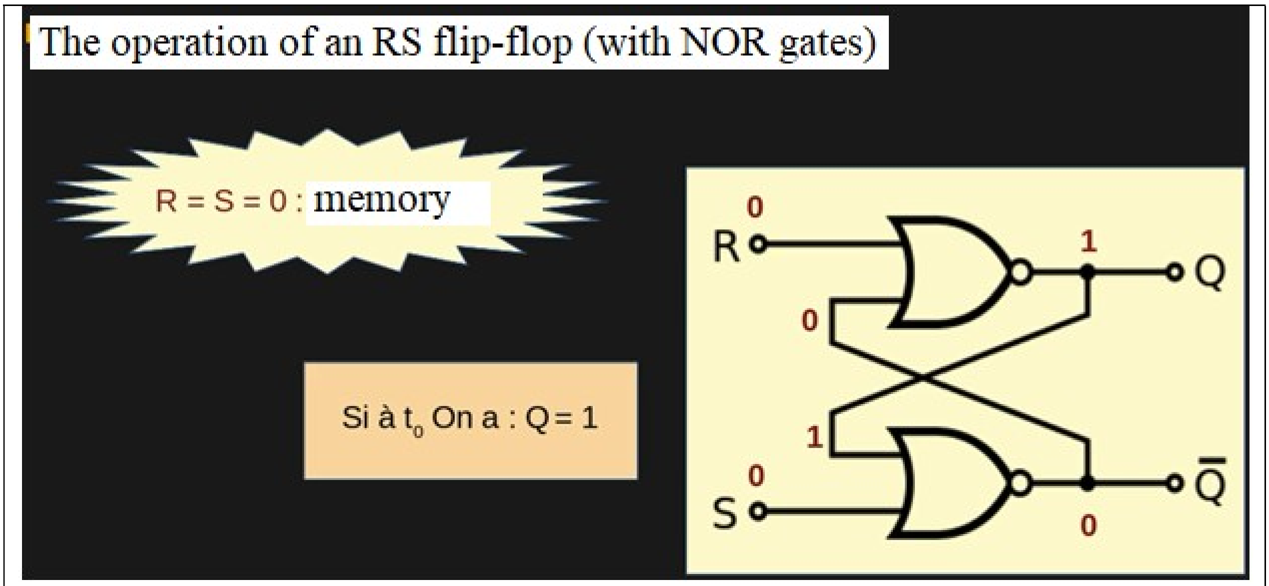
With $\overline{R}, \overline{S}$ and NOR gates





R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	X

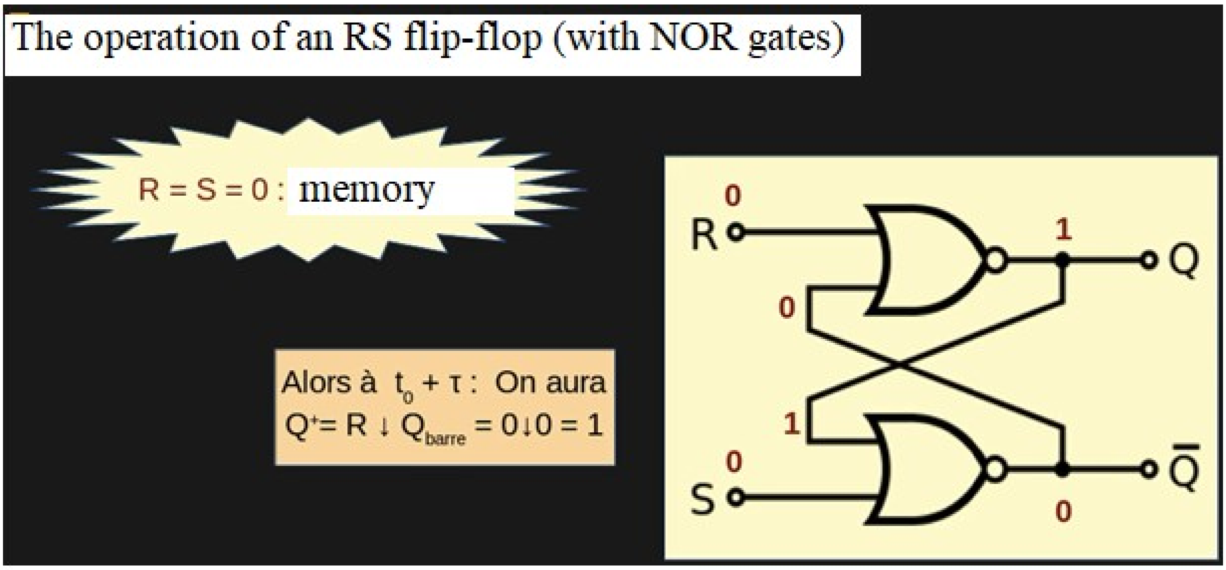
Analysis of RS Flip-flop Behavior using only NOR gates :



The operation of an RS flip-flop (with NOR gates)

R = S = 0 : memory

Si à t_0 On a : Q = 1

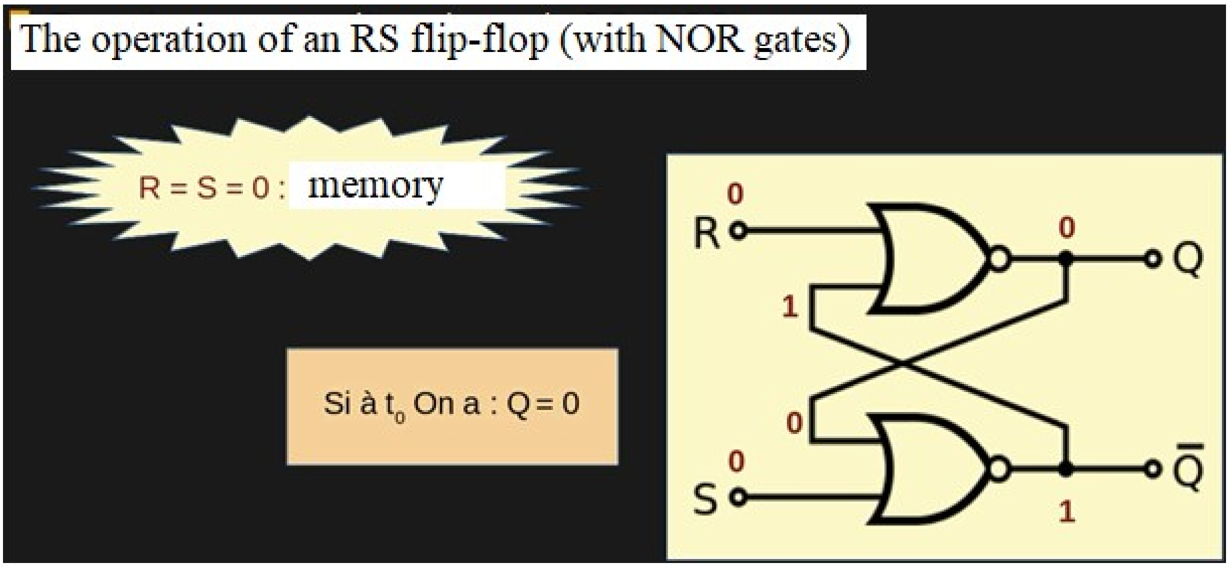


The operation of an RS flip-flop (with NOR gates)

R = S = 0 : memory

Alors à $t_0 + \tau$: On aura

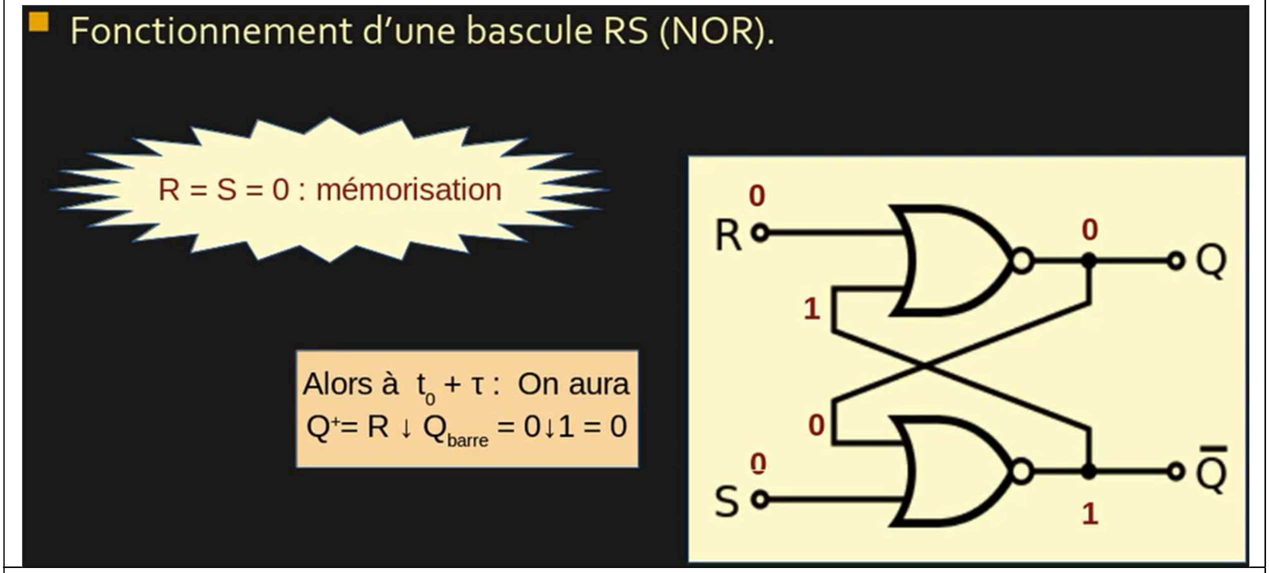
$Q^+ = R \downarrow Q_{barre} = 0 \downarrow 0 = 1$



The operation of an RS flip-flop (with NOR gates)

R = S = 0 : memory

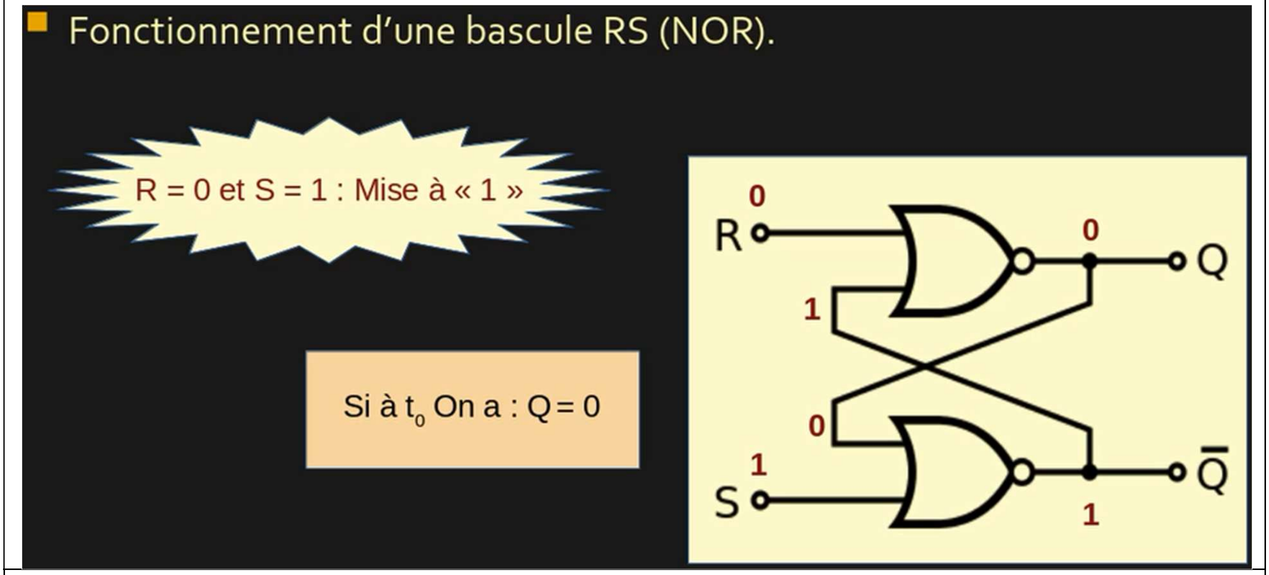
Si à t_0 On a : Q = 0



Fonctionnement d’une bascule RS (NOR).

R = S = 0 : mémorisation

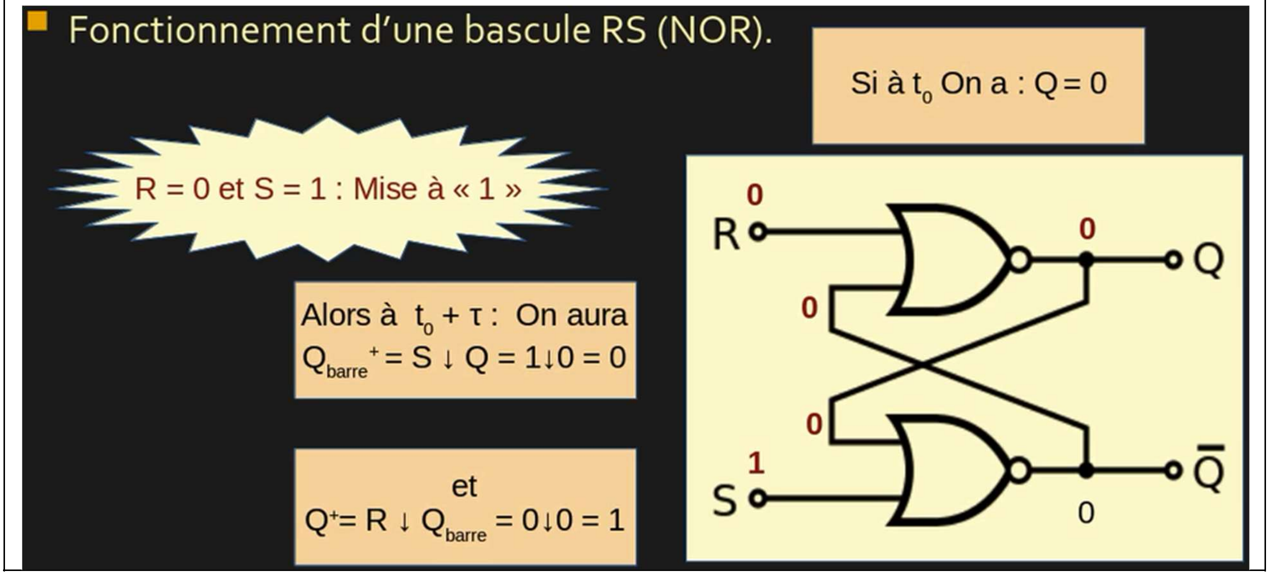
Alors à $t_0 + \tau$: On aura $Q^+ = R \downarrow Q_{barre} = 0 \downarrow 1 = 0$



Fonctionnement d’une bascule RS (NOR).

R = 0 et S = 1 : Mise à « 1 »

Si à t_0 On a : Q = 0



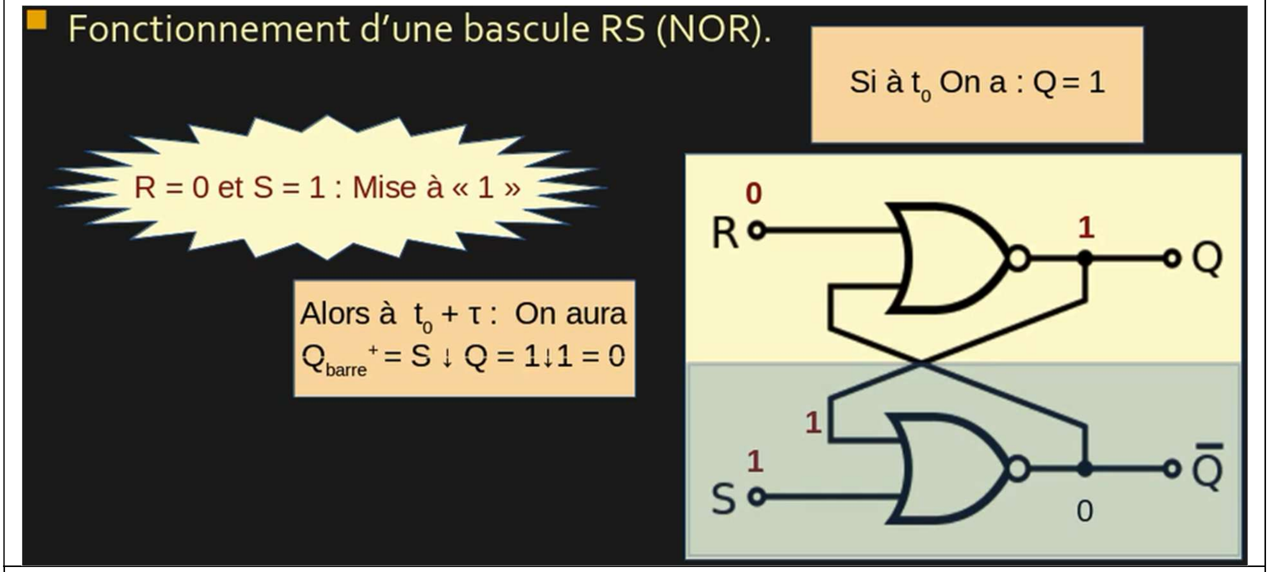
Fonctionnement d’une bascule RS (NOR).

Si à t_0 On a : Q = 0

R = 0 et S = 1 : Mise à « 1 »

Alors à $t_0 + \tau$: On aura $Q_{barre}^+ = S \downarrow Q = 1 \downarrow 0 = 0$

et $Q^+ = R \downarrow Q_{barre} = 0 \downarrow 0 = 1$

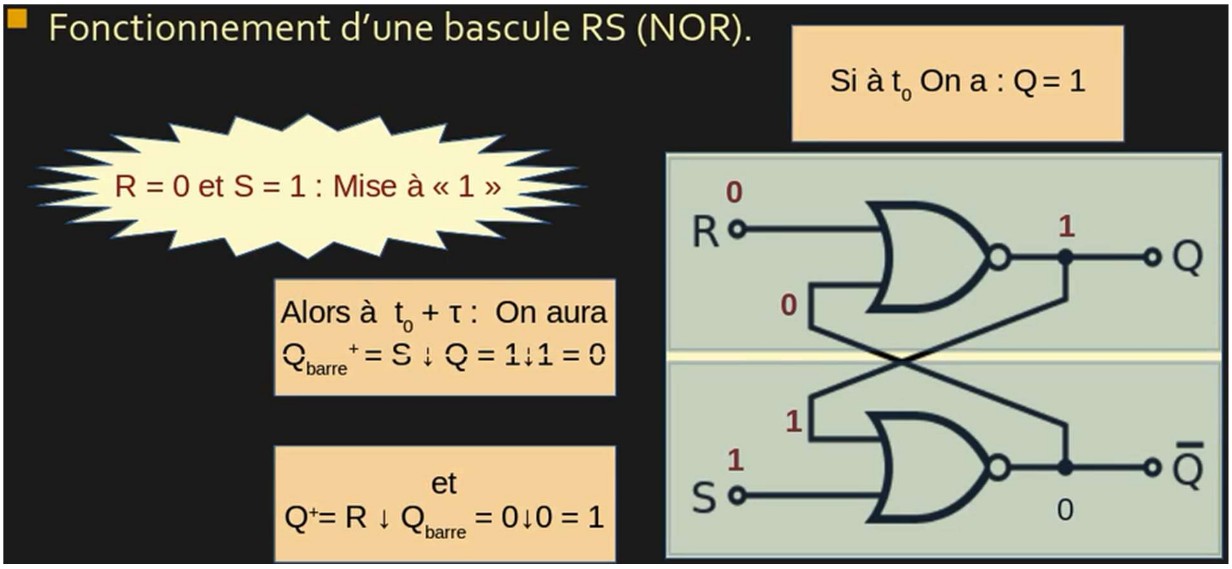


Fonctionnement d'une bascule RS (NOR).

Si à t_0 On a : $Q = 1$

R = 0 et S = 1 : Mise à « 1 »

Alors à $t_0 + \tau$: On aura $Q_{barre}{}^{+} = S \downarrow Q = 1 \downarrow 1 = 0$



Fonctionnement d'une bascule RS (NOR).

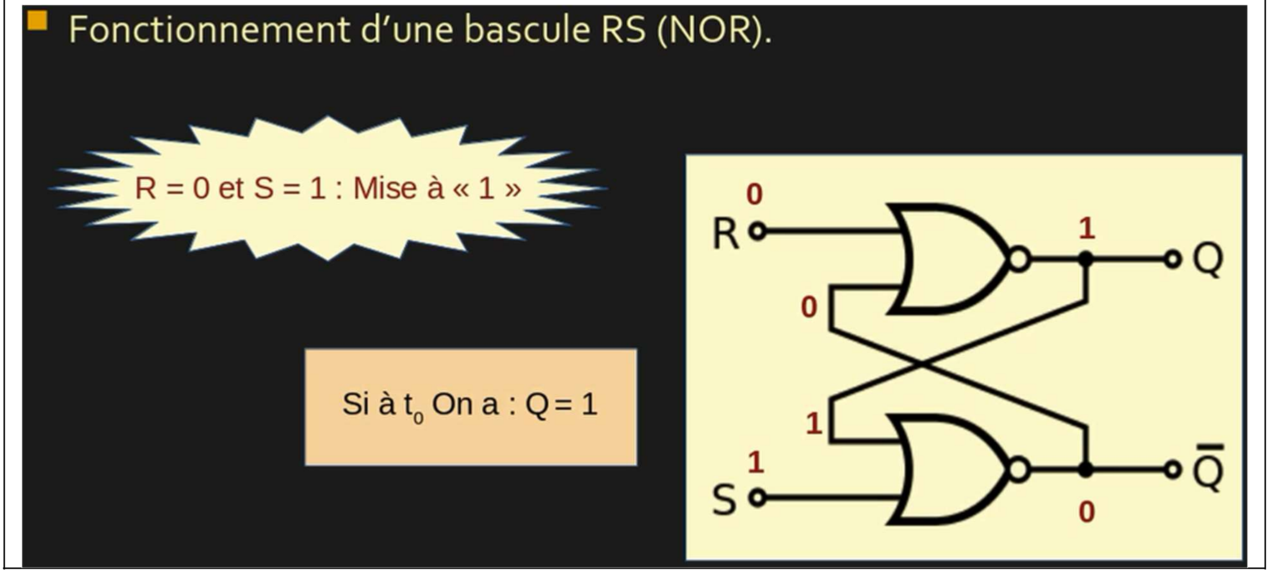
Si à t_0 On a : $Q = 1$

R = 0 et S = 1 : Mise à « 1 »

Alors à $t_0 + \tau$: On aura $Q_{barre}{}^{+} = S \downarrow Q = 1 \downarrow 1 = 0$

et

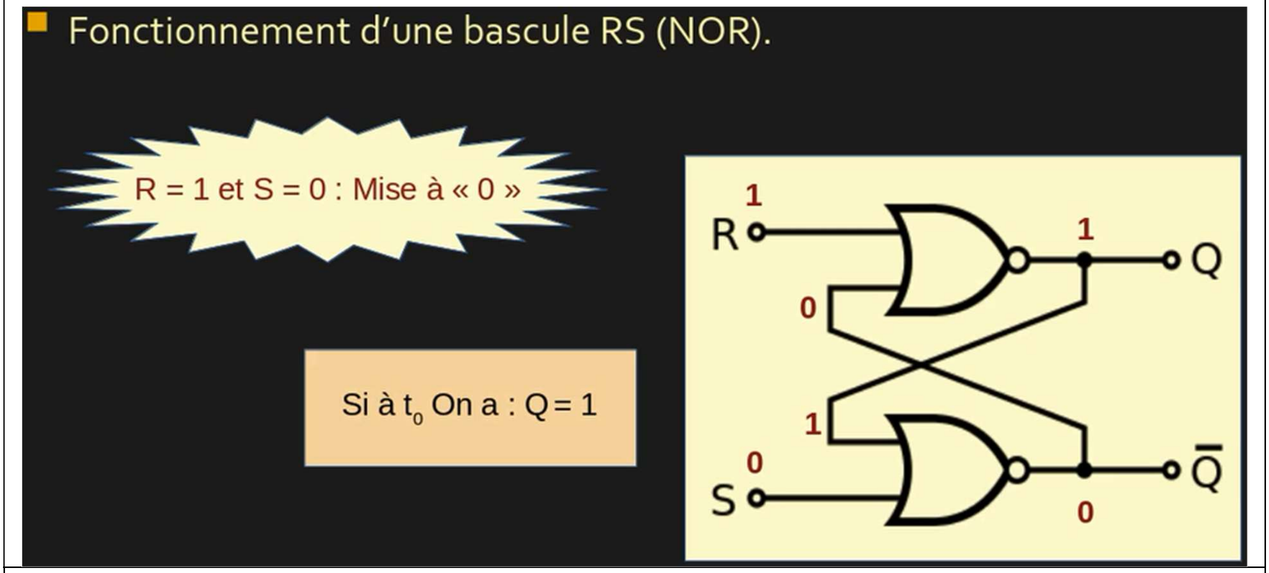
$Q^{+} = R \downarrow Q_{barre} = 0 \downarrow 0 = 1$



Fonctionnement d'une bascule RS (NOR).

R = 0 et S = 1 : Mise à « 1 »

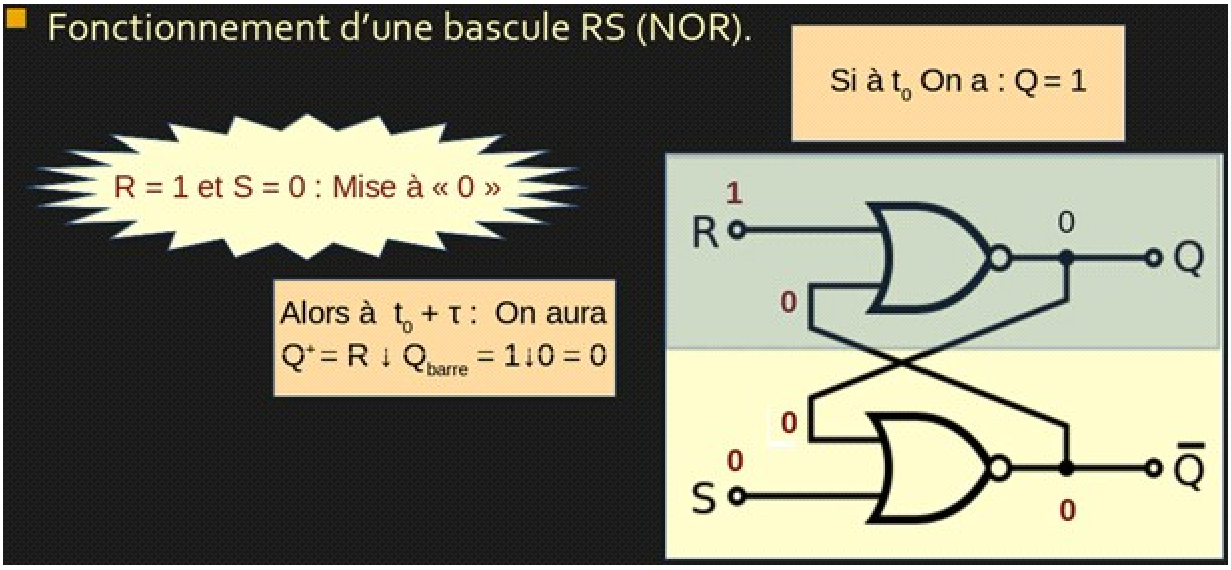
Si à t_0 On a : $Q = 1$



Fonctionnement d’une bascule RS (NOR).

R = 1 et S = 0 : Mise à « 0 »

Si à t_0 On a : Q = 1

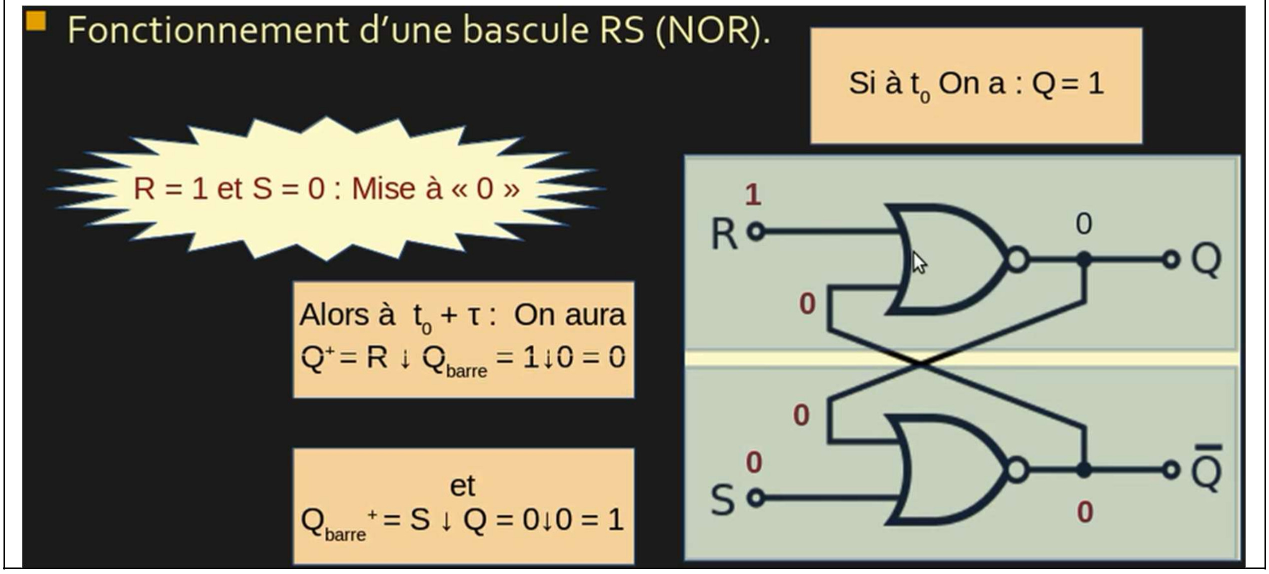


Fonctionnement d’une bascule RS (NOR).

Si à t_0 On a : Q = 1

R = 1 et S = 0 : Mise à « 0 »

Alors à $t_0 + \tau$: On aura
$Q^+ = R \downarrow Q_{barre} = 1 \downarrow 0 = 0$



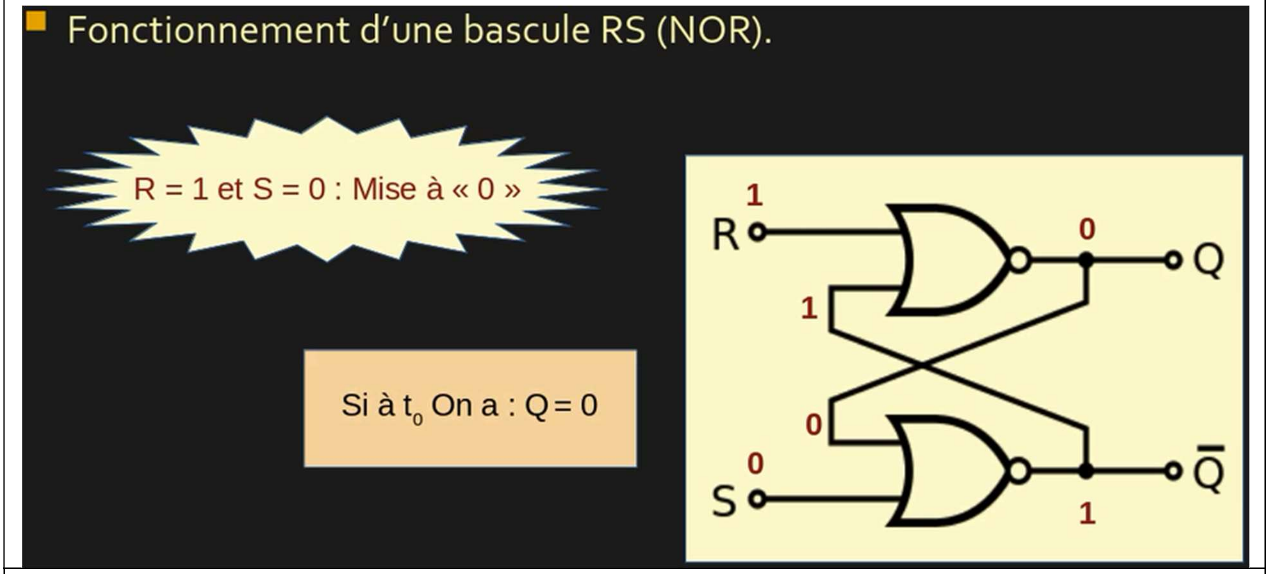
Fonctionnement d’une bascule RS (NOR).

Si à t_0 On a : Q = 1

R = 1 et S = 0 : Mise à « 0 »

Alors à $t_0 + \tau$: On aura
$Q^+ = R \downarrow Q_{barre} = 1 \downarrow 0 = 0$

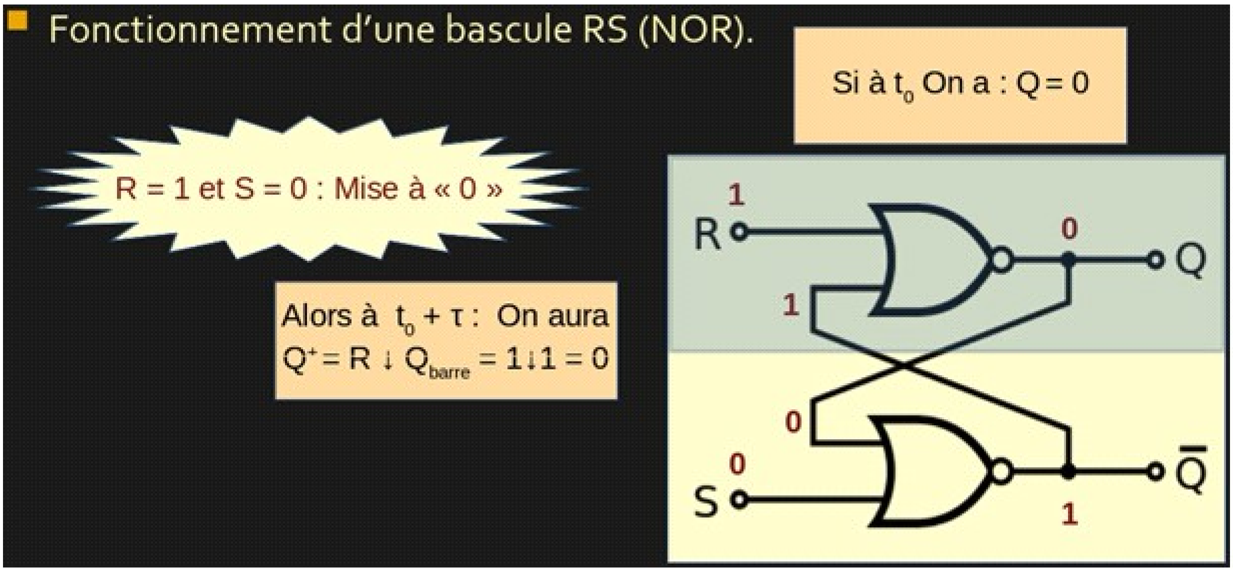
et
$Q_{barre}{}^+ = S \downarrow Q = 0 \downarrow 0 = 1$



Fonctionnement d'une bascule RS (NOR).

R = 1 et S = 0 : Mise à « 0 »

Si à t_0 On a : Q = 0

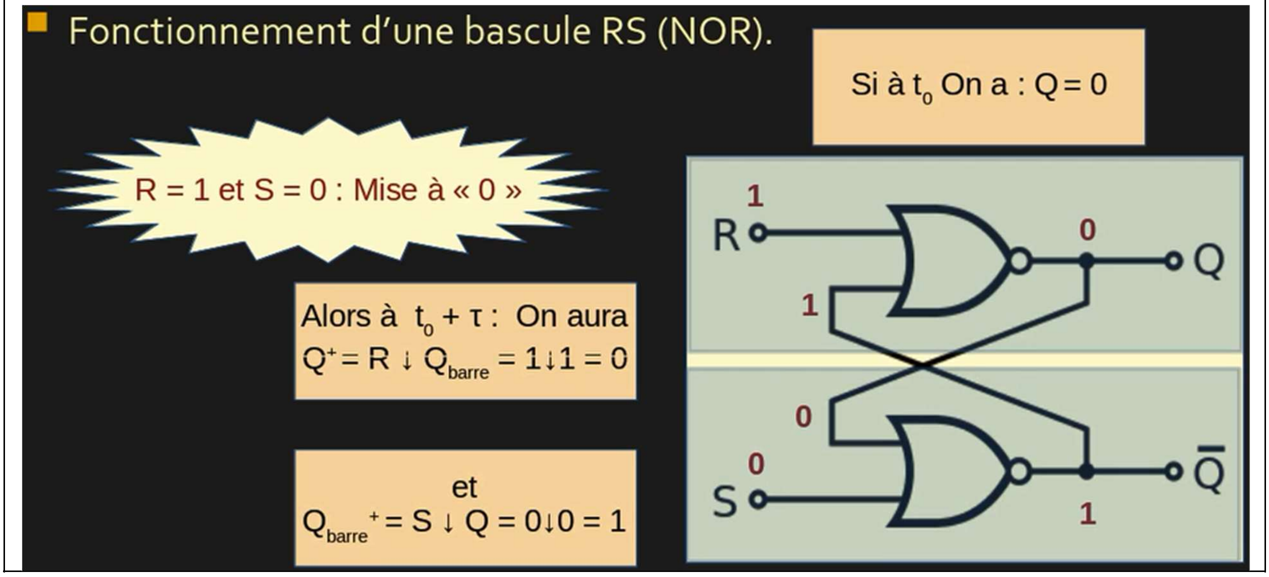


Fonctionnement d'une bascule RS (NOR).

Si à t_0 On a : Q = 0

R = 1 et S = 0 : Mise à « 0 »

Alors à $t_0 + \tau$: On aura $Q^+ = R \downarrow Q_{barre} = 1 \downarrow 1 = 0$



Fonctionnement d'une bascule RS (NOR).

Si à t_0 On a : Q = 0

R = 1 et S = 0 : Mise à « 0 »

Alors à $t_0 + \tau$: On aura $Q^+ = R \downarrow Q_{barre} = 1 \downarrow 1 = 0$

et $Q_{barre}{}^+ = S \downarrow Q = 0 \downarrow 0 = 1$