Part II

COMBINATIONAL AND SEQUENTIAL LOGIC

Chapter 6:

SEQUENTIAL LOGIC

Reminder

- **Combinatorial circuits:**
  - Outputs depend on input values only

- **Sequential circuits:**
  - Addition of the state and the memory notions
  - Addition of the concept of time (clock)

1. Introduction

Several circuits used in everyday life need memory. Thus, a sequential system is a logical system whose state of the output variables depends of the state of the input variables + the previous state of the output variables.

The system remembers the past by recording the previous states of its outputs, using internal variables, or memories, to do this.

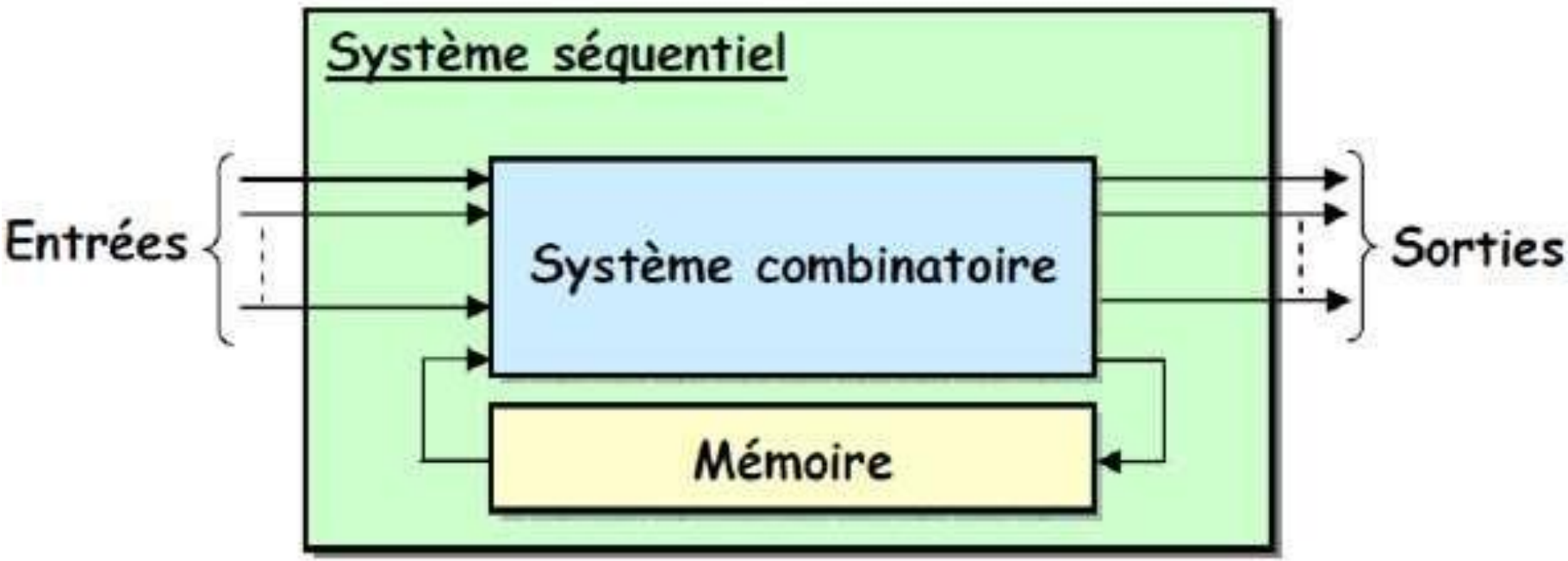
Introduction

- The circuit output values depend on:
  - Input values
  - Previously calculated values
  - the actual state

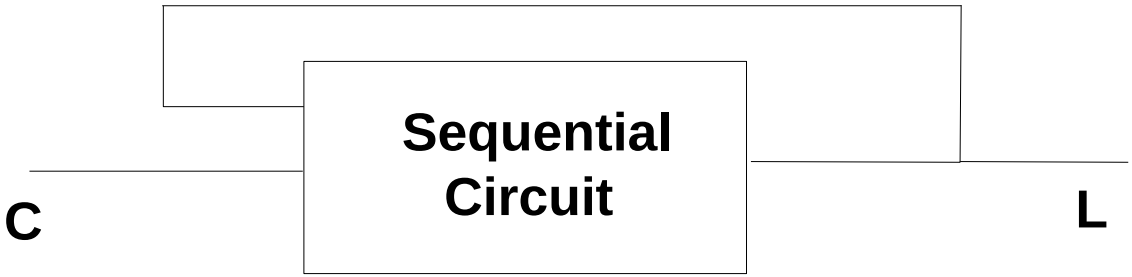
- Theories used to study/specify different types of circuits
  - **Combinatorial circuits**: Boolean algebra
  - **Sequential circuits**: finite state automata theory

Introduction

A sequential circuit is a digital (logic) circuit whose state at time t+1 is a function of the inputs at the same time t+1 and the previous state of the system (time t).



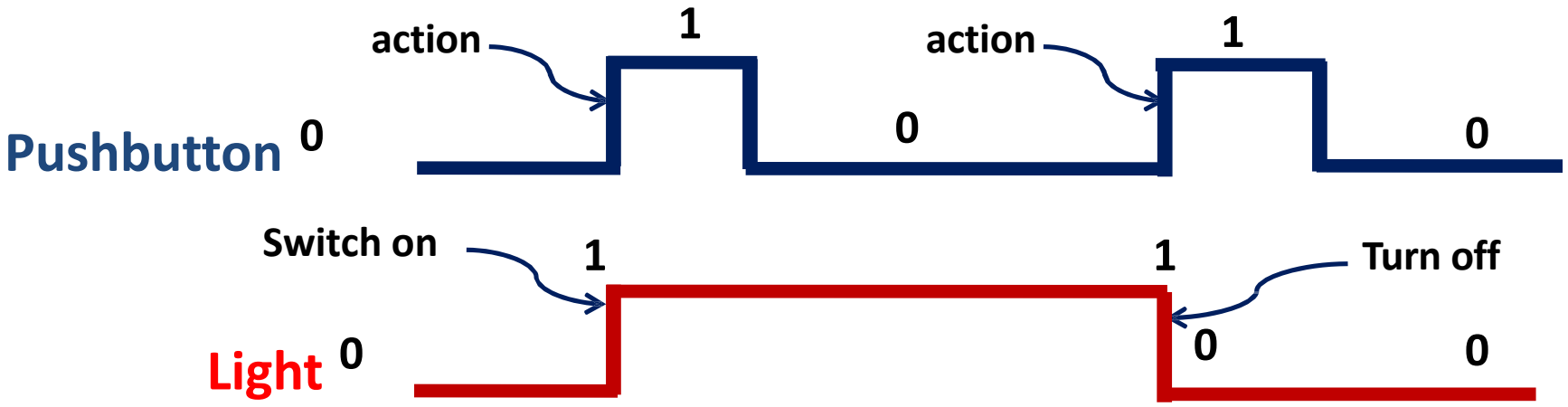
Example of a sequential circuit



C	L		L+	
1	X		L	Memory
0	0		1	Switch
0	1		0	Switch

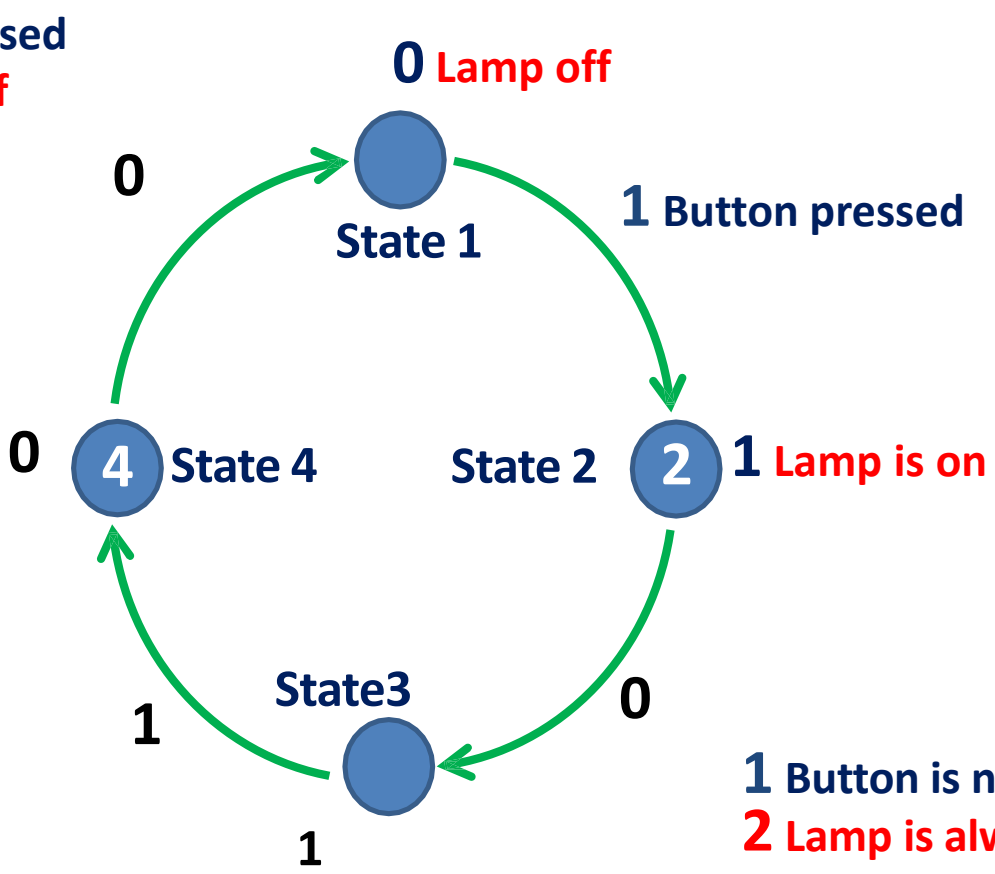
2. State Concept

By way of illustration, let's take the example of an electric lamp controlled by a pushbutton in such a way that pressing the button turns the lamp on, and a successive action turns it off, according to the example in the following chronogram :



State concept

Phase graph



3. Types of sequential circuits

There are two types of sequential circuits:

1. **Synchronous**, where the behavior is determined by signals at discrete instants (measured, moderate) of time (clocked to the rhythm of a signal called clock or CLK or H or CK).

2. **Asynchronous**, where the behavior is determined by the signals at any given time, and the order with which the inputs vary. Thus, in asynchronous circuits, the output is modified as soon as there is a change in the state of the inputs. In synchronous circuits, the output only changes after a clock signal.

Synchronous System (Notion of the clock)

Synchronization is achieved using a clock, which provides a periodic signal.

This periodic signal is distributed throughout the system so that memorisation elements are only affected by the arrival of a clock pulse.

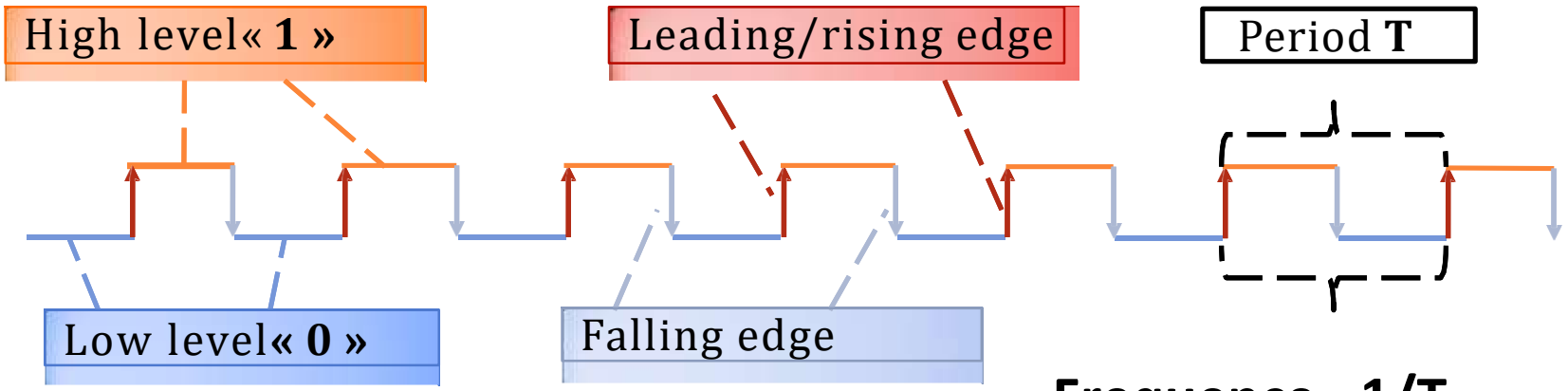
The clock determines when there is an activity in the circuit, and the other signals determine **what** (what changes) in the circuit. Memorisation elements that are controlled by the clock transition are called **flip-flops**.

4. Flip-flops & Clock

The flip-flop is the basic element allowing the construction of sequential circuits (Memory, Register, Counter, etc.).

A clock is a logical variable that periodically passes successively from 0 to 1 and from 1 to 0.

This variable is often used as an input to sequential circuits (the clock is denoted by h or ck (clock)).



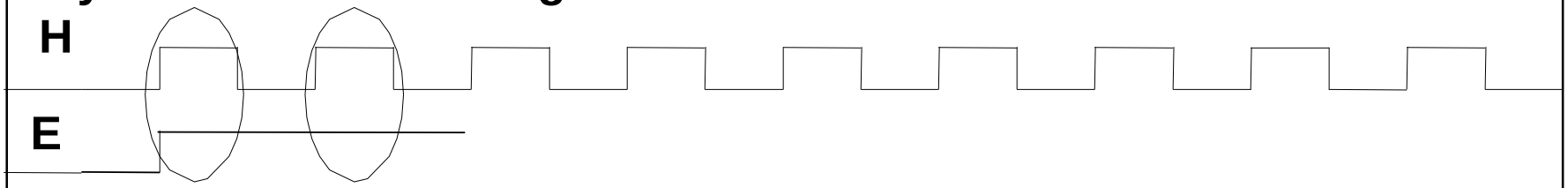
Frequence= 1/T

Clock

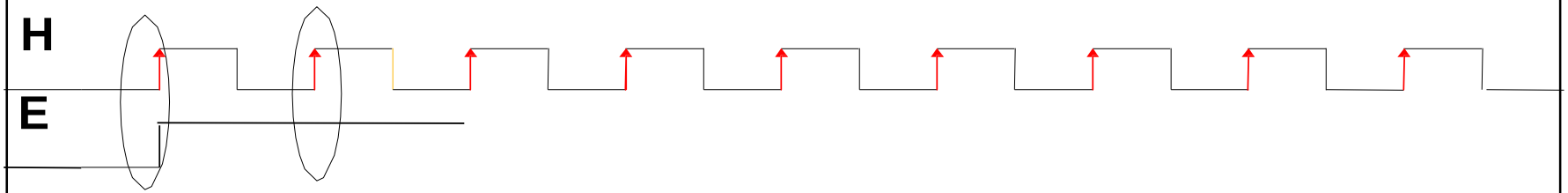
Frequence = 1/T = number of changes per second in hertz(Hz)

Clock	Period
1 Hz	1 second
1 Mega Hz	1 milliseconds
1 Giga Hz	1 nanoseconds

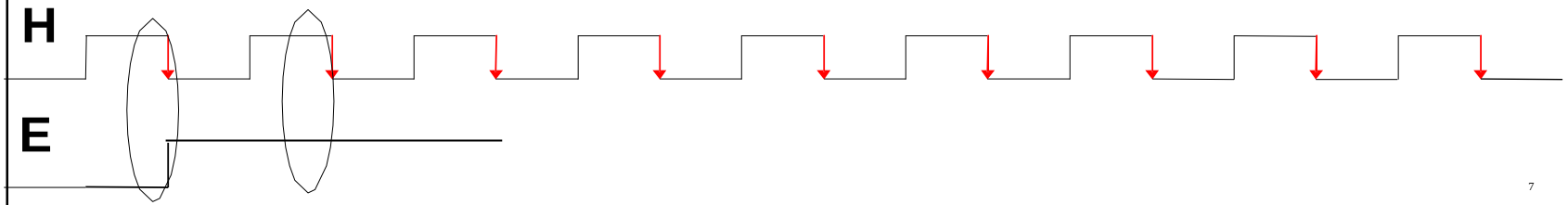
Synchronization on high level



Synchronization on leading/rising edge

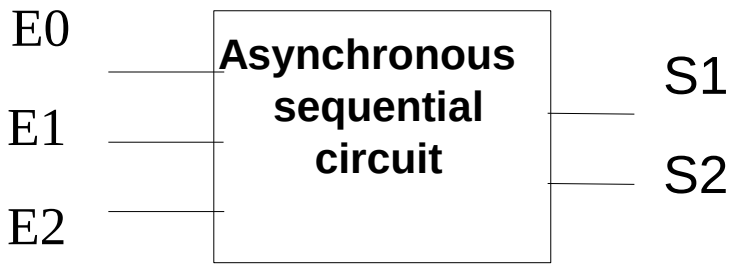


Synchronization on falling edge



Asynchronous sequential circuits

When a sequential circuit does not have a clock as an input variable or if the circuit operates independently of this clock then **this circuit is asynchronous**.

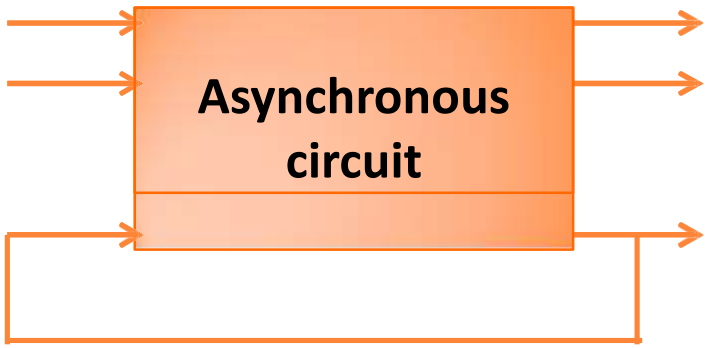


8

Asynchronous/synchronous sequential circuits

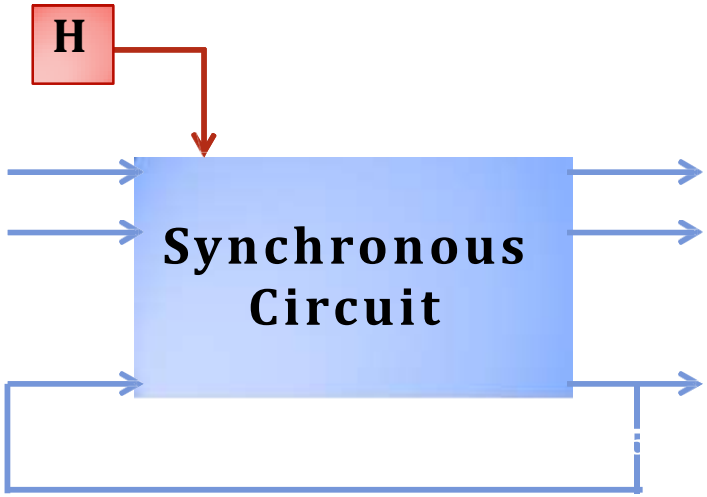
Asynchronous Circuits

- System variables evolve freely over time.



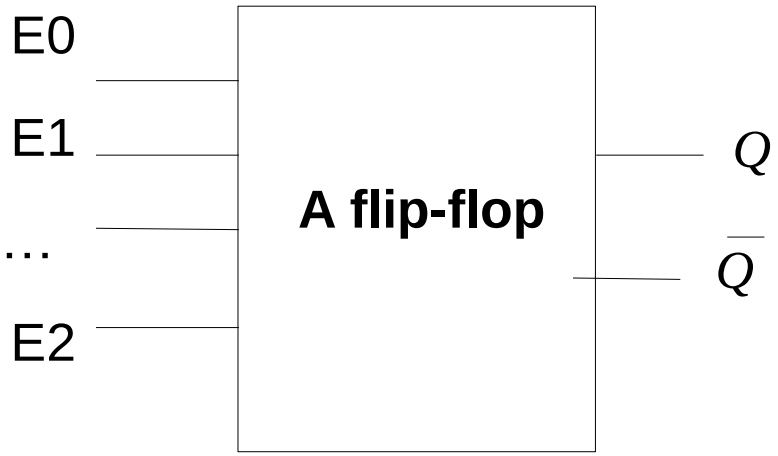
Synchronous Circuits

- The evolution of the variables depends on a clock pulse as one of the input signals.



5. Flip-flops/Latch

- ✓Flip-flops are the basic circuits of the sequential logic.
- ✓A flip-flop can have a clock (synchronous) or not (asynchronous).
- ✓Each flip-flop has inputs and two outputs Q and $\overline{Q}$.
- ✓A flip-flop has a memory and a switching function.



$$Q^{+} = F(Ei, Q)$$

There are several types of flip flops and latches : **SR, D, T, and JK**

5.1. Definition of a SR latch

In the name of this latch, we can notice 2 letters: “S” and “R”.
These are the inputs to this latch:
✓“S” for set or setting to a “1”, and
✓“R” for reset or zero “0”

Obviously, this latch will have an output that we will call Q.

In reality, we will always have 2 outputs: Q and $\overline{Q}$.

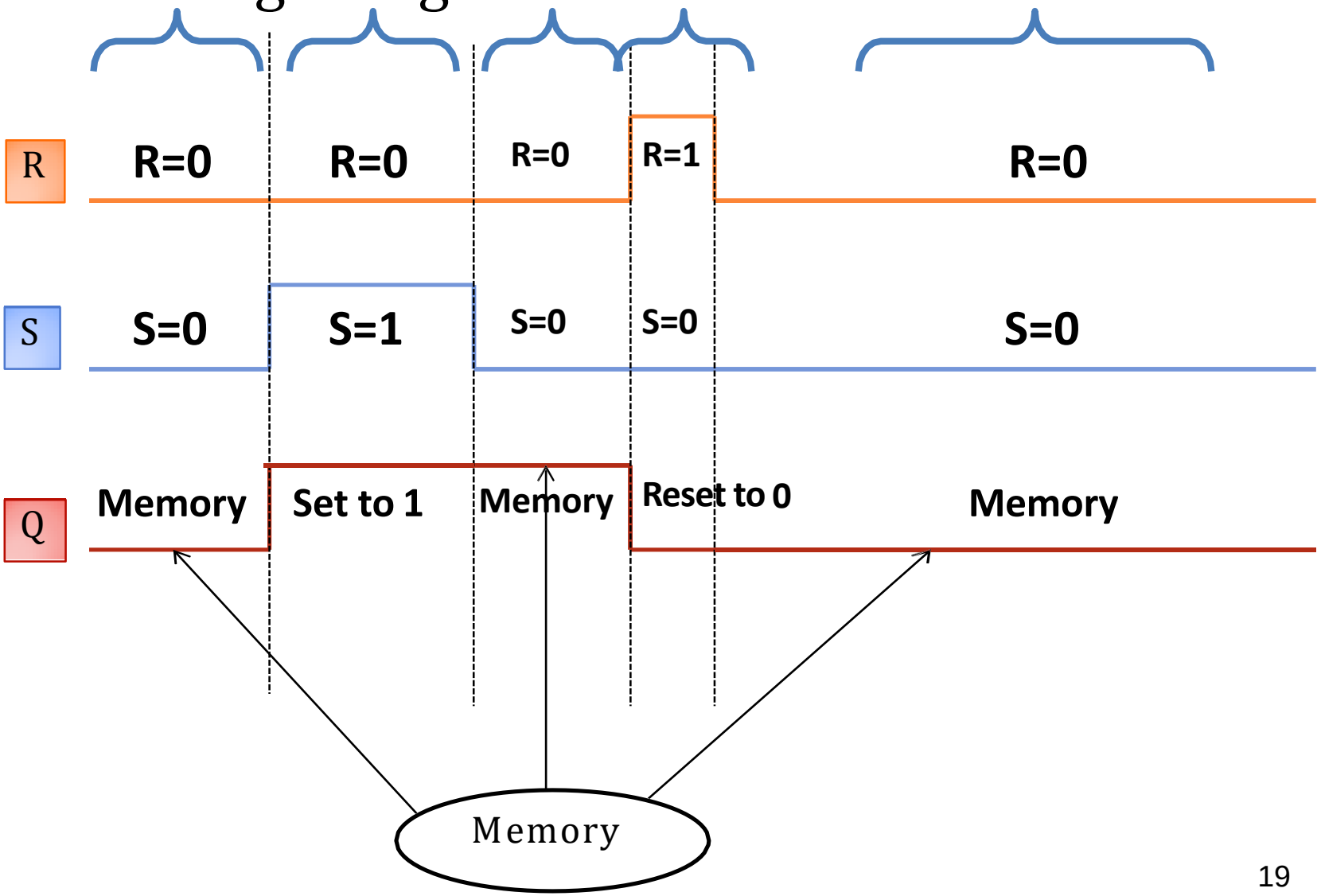
5.1 SR Latch (Set, Reset)

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

R	S	Q-		Q+	
0	0	0		0	Memorisation
0	0	1		1	
0	1	0		1	Set to 1
0	1	1		1	
1	0	0		0	Reset to 0
1	0	1		0	
1	1	0		X	Forbidden
1	1	1		X	

SR Latch

Timing diagram: CHRONOGRAM



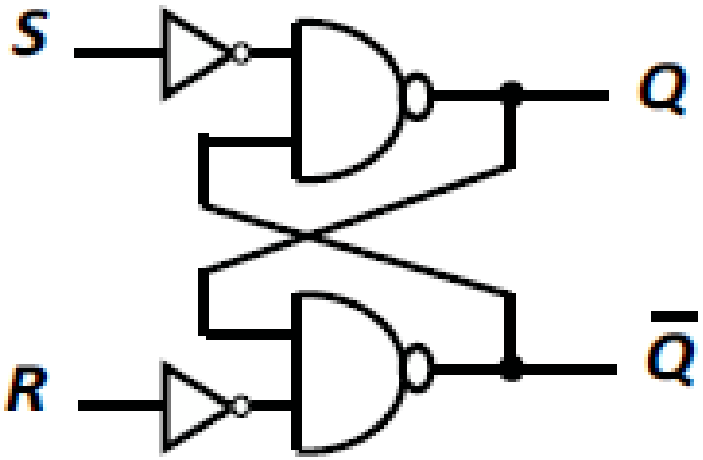
Internal structure of an SR Latch

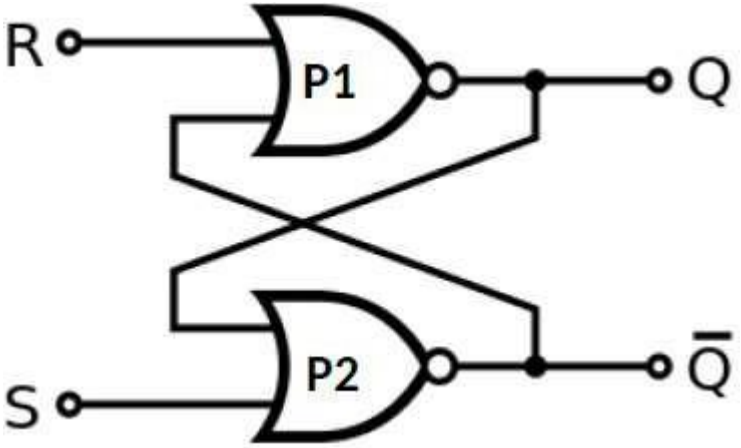
$Q^+ = S + \overline{R}.Q$ **(using the Karnaugh map)**

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

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

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



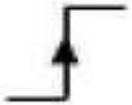
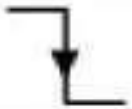


Synchronous Flip-Flop

The change of state of an output of a synchronous sequential system depends on the state of the control inputs and that of the active synchronization signal, the clock signal.

- **Clock signal action modes:**

There are four clock action modes known by the following symbols:

Symbol	Designation
	High level
	Low level
	Leading edge
	Falling edge

SYNCHRONOUS FLIP-FLIPS

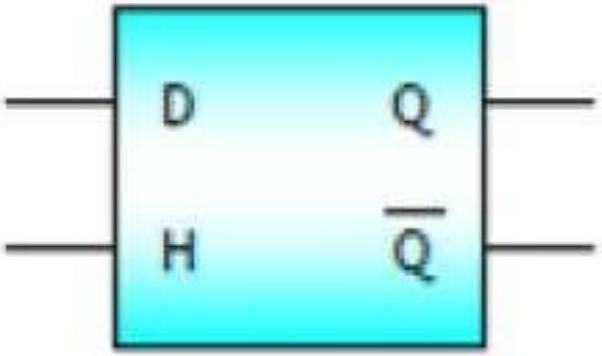
Depending on the mode of the clock action, there are two families of flip-flops:

- Flip-flops controlled by the clock level (high level or low level):
  The flip-flop is said to be statically controlled (active on high level or active on low level).

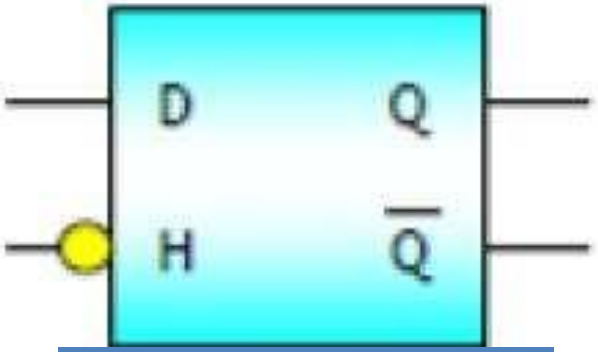
- Flip-flops controlled by the clock edge (rising or falling edge):
  The flip-flop is said to be dynamically controlled (active on leading/rising edge or active on falling edge).

SYNCHRONOUS FLIP-FLIPS

Symbole :

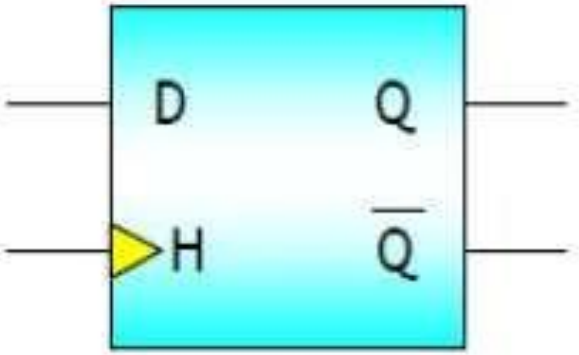


Active on High level

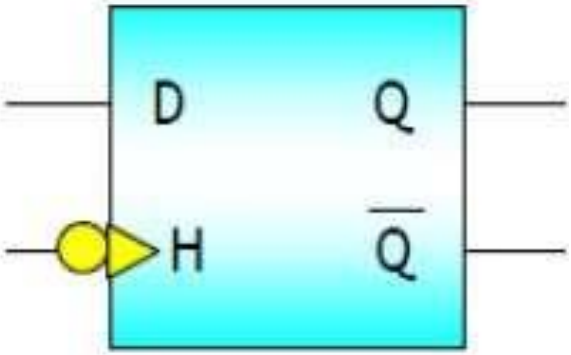


Active on Low level

Symbole :



Active on leading edge



Active on falling edge

5.2. SR synchronous flip-flop

SR synchronous flip-flop (SR Latch with Clock level activation)

The RS latch can easily be modified to make it synchronous, i.e. to allow changes only when the clock signal is 1 (or 0). The SR flip-flop is a flip-flop for which the S and R inputs are only taken into account in coincidence with a control signal.

This signal can be supplied by a clock, in which case we have a synchronous flip-flop.

In this case, the SR flip-flop is an RS latch whose commands Set and Reset commands change the state of output Q only after a clock signal H (Clock CK) has been enabled.

SR Flip-Flop: SR Latch with an adge activation clock signal

The SR flip-flop is: an RS Latch synchronized with a clock signal H or Klc.

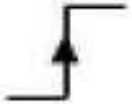
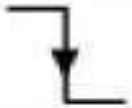
H	R	S	Q+	
0	X	X	Q-	Memorisation
1	0	0	Q-	SR Latch
1	0	1	1	
1	1	0	0	
1	1	1	X	

Reminder: Types of synchronizations

There are several ways to take into account changes in flip-flop status:

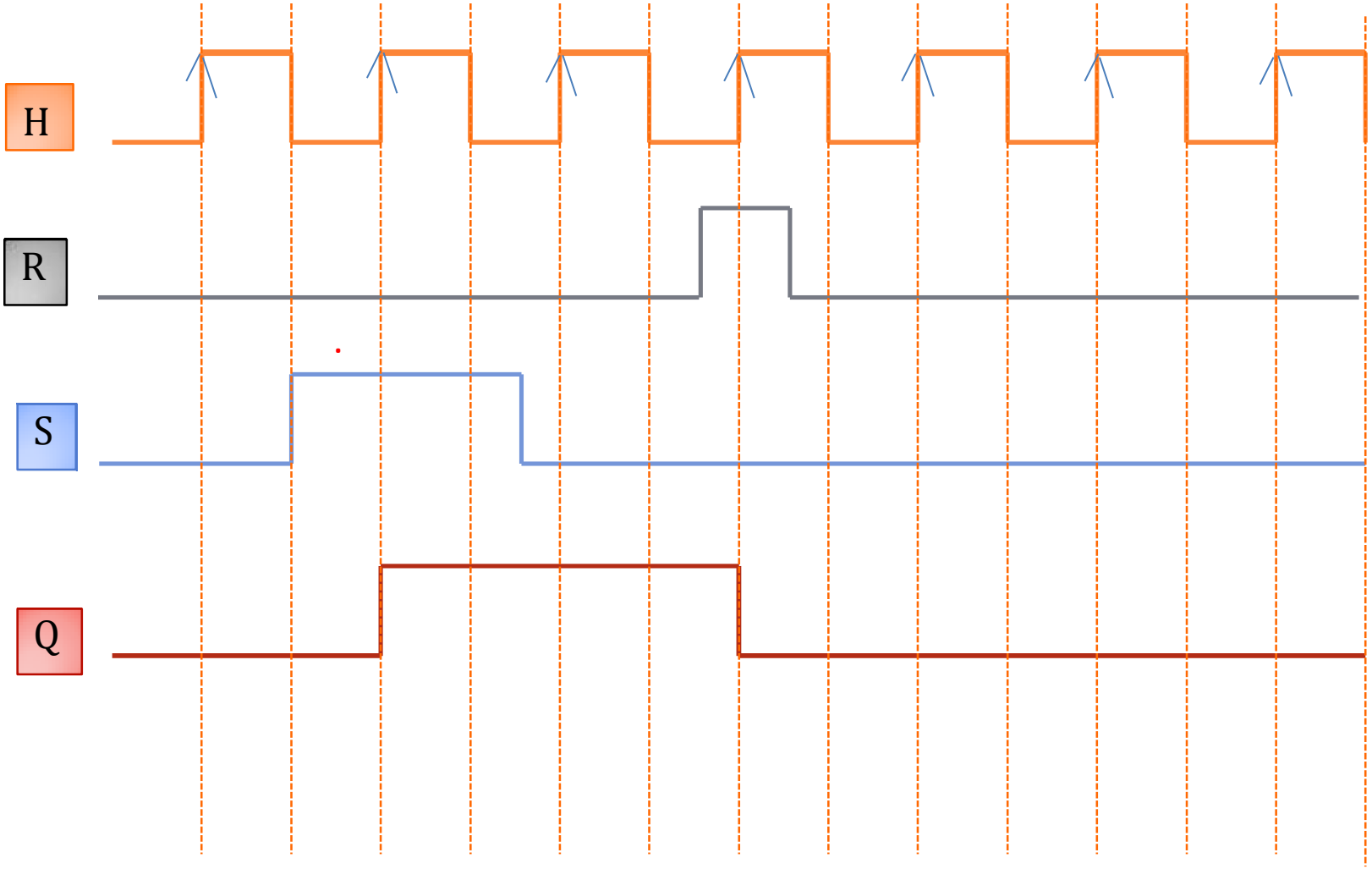
1. High clock level: flip-flop inputs are taken into account when the clock level is high (at "1").
2. Low clock level: toggle inputs are taken into account when the clock level is low (at "0").

3. Leading clock edge.
4. Falling clock edge.

Symbol	Designation
	High level
	Low level
	Leading edge
	Falling edge

SR flip-flop (clock active on leading edge)

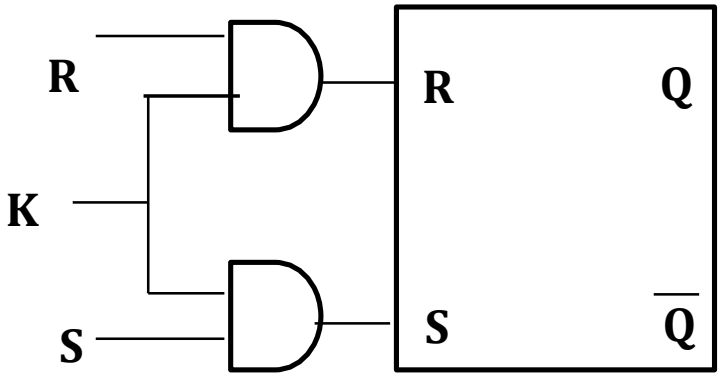
Timing diagram: CHRONOGRAMME



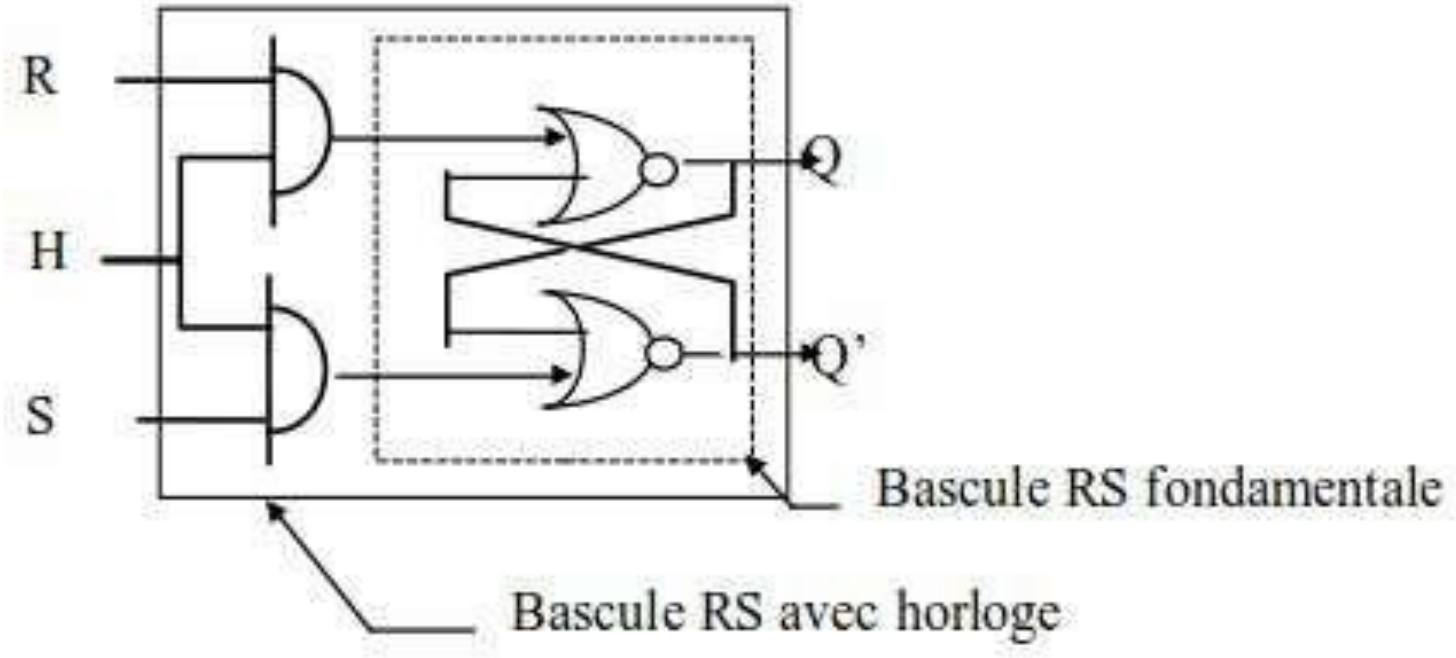
SR Flip-Flop

Question: Give the circuit of this SR flip-flop using the RS Latch.

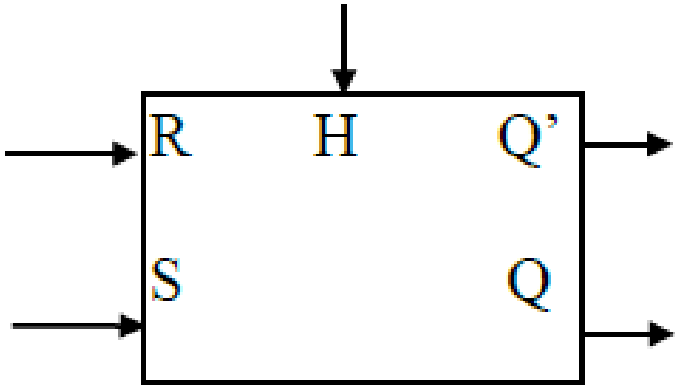
Solution: Simply apply the logic gate AND on its inputs with the clock signal to force the clock signal to take its inputs into account.



SR Flip-Flop

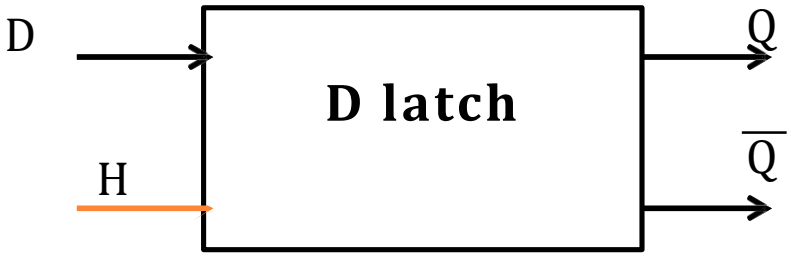


Schematically, the SR flip-flop is represented as follows:



5.3.D-Latch

It is a synchronous latch on high level or low level



On high level« 1 »

If H = 1 then Q+ = D

On low level« 0 »

If H = 0 then Q+ = D

H/$\overline{H}$	Q+
0	Q-
1	D

H/$\overline{H}$	D	Q+
0	0	Q-
0	1	Q-
1	0	0
1	1	1

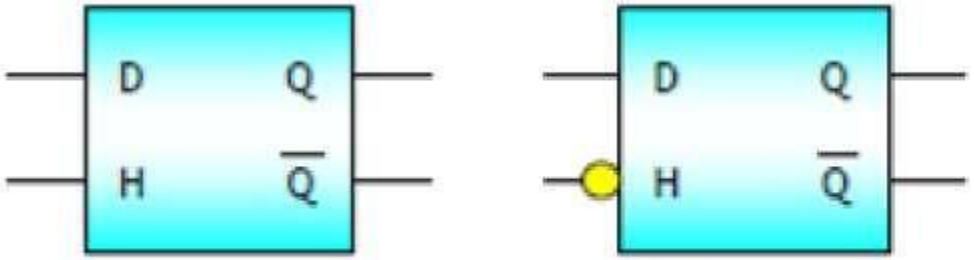
D-Latch principle

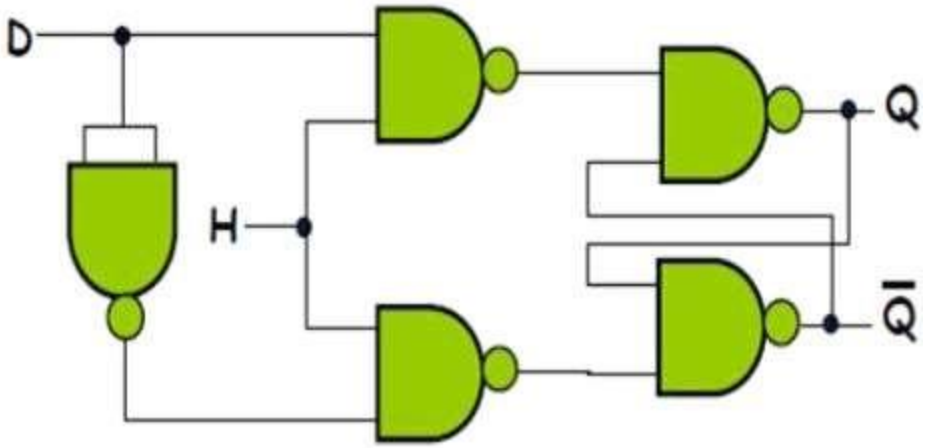
It is a synchronous static latch on the level clock whose operation is as follows:
The flip-flop is transparent as long as the clock signal is high (or low).The Q output follows all variations of the D input. The lock is said to be transparent.

The state of the Q output is latched (memory) as long as the clock signal is low (or high). The Q output maintains its logical state. The lock is said to be blocked.

D-Latch principle

Symbole :





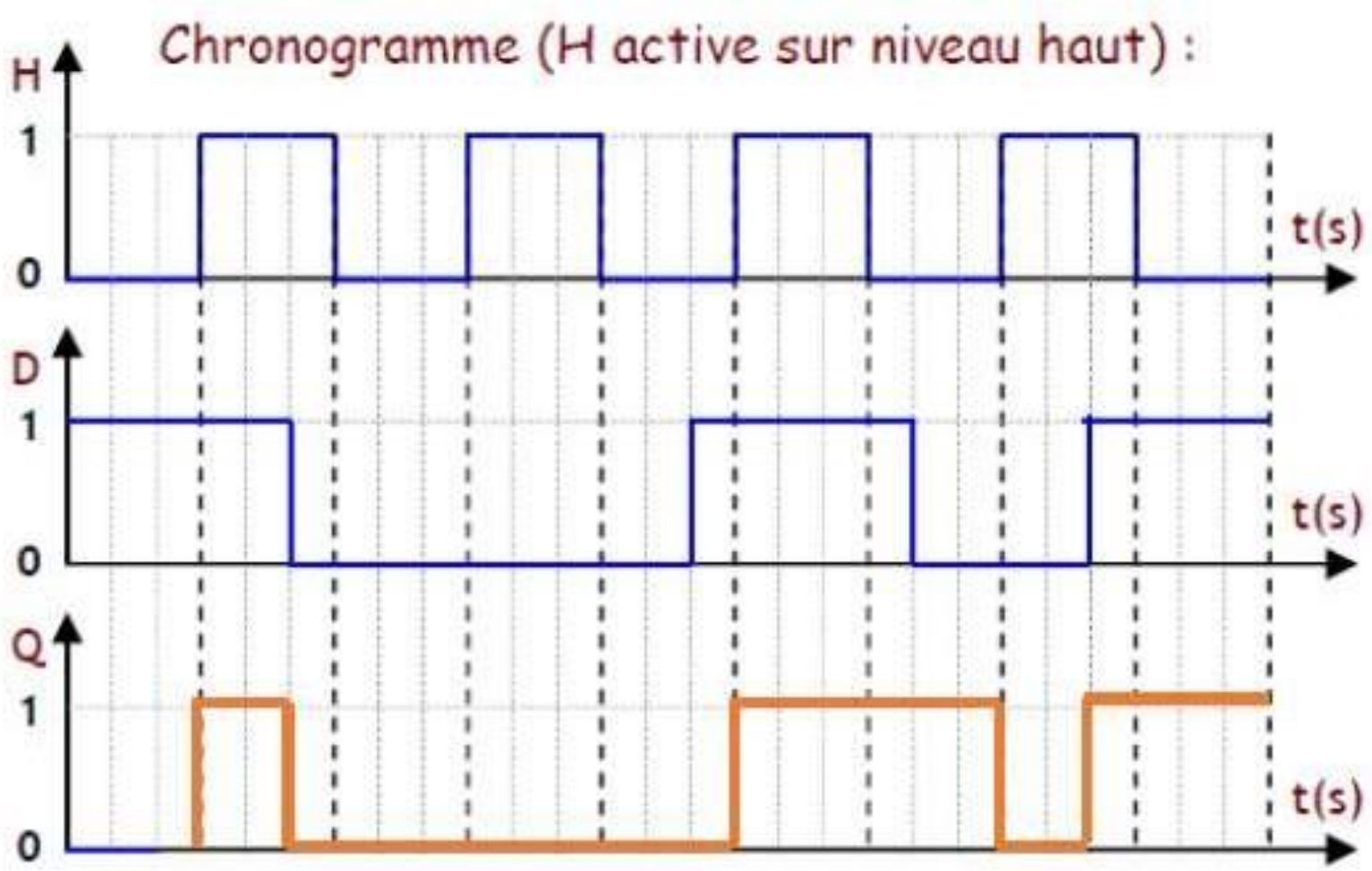
D-Latch principle

- When H = 1, the output Q is the same as D (Q=D); the latch is said to be transparent.
- When H = 0, the output Q remains at the last value of D it had before H went to level 0
- In other words, the output is latched to D and does not change as long as CLK remains low, even if D changes value.
- The operation of the D "latch" flip-flop is summarized in the following truth table.

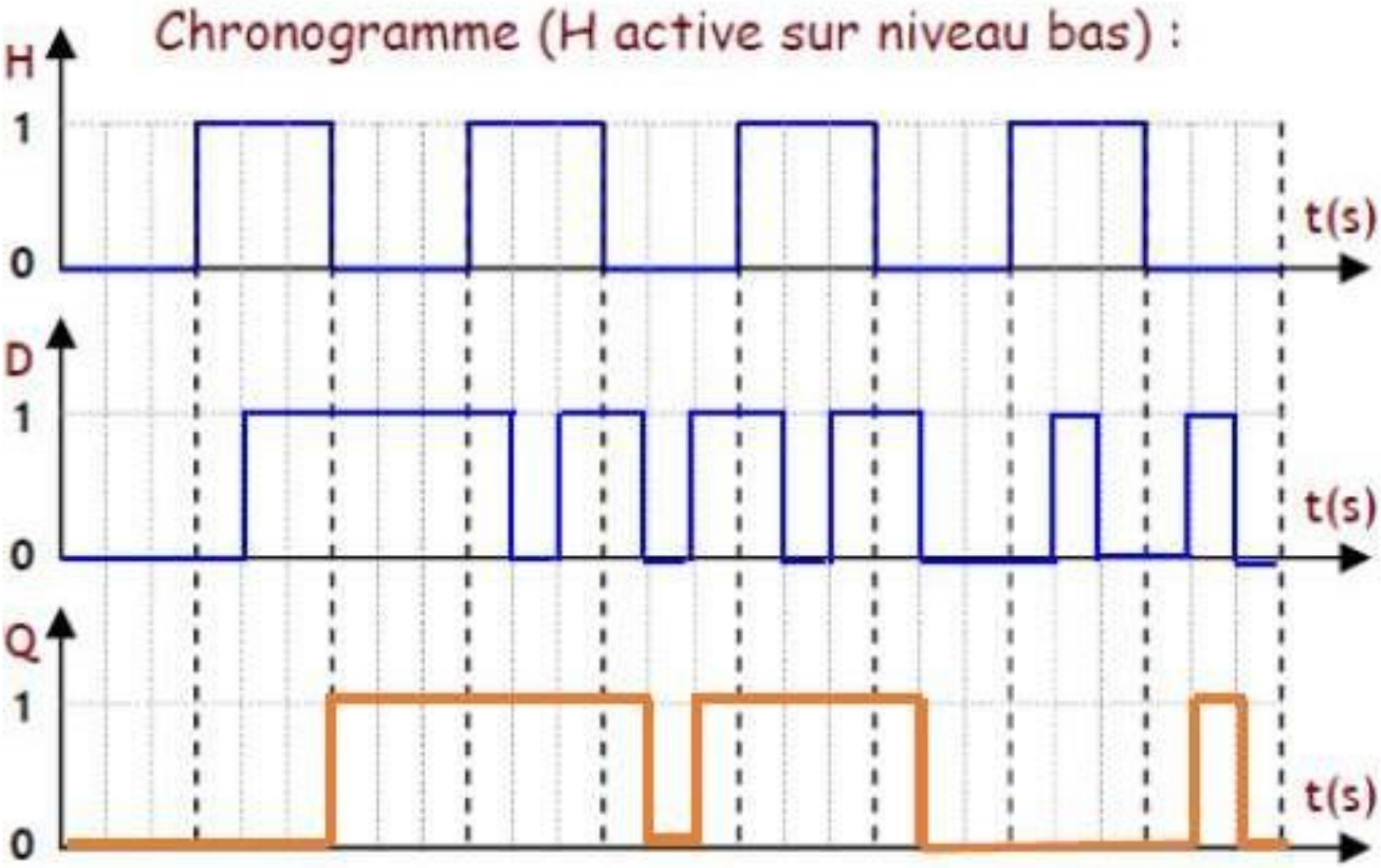
D-Latch principle

Table de vérité			
H	D	Q	Commentaire
0	x	q	Mémorisation
1	0	0	La bascule recopie la valeur de D sur Q
1	1	1	

D-Latch Timing diagram: Chronogramme (Klock is Active on high level)

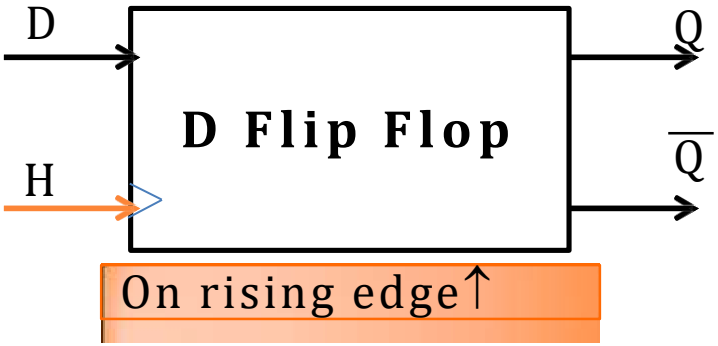


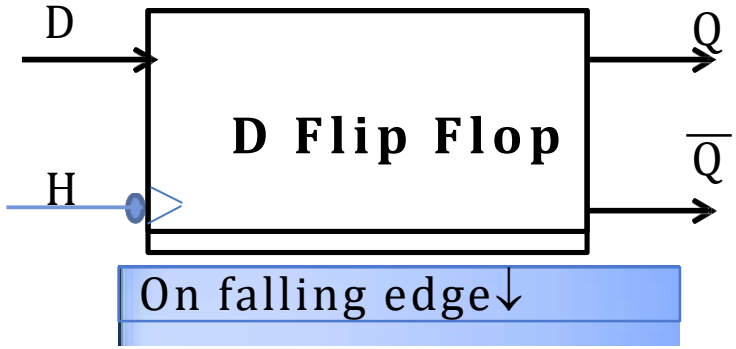
D-Latch Timing diagram: Chronogramme (Klock is Active on low level)



5.4. D Flip-Flop

It is a flip-flop synchronized on rising or falling edge



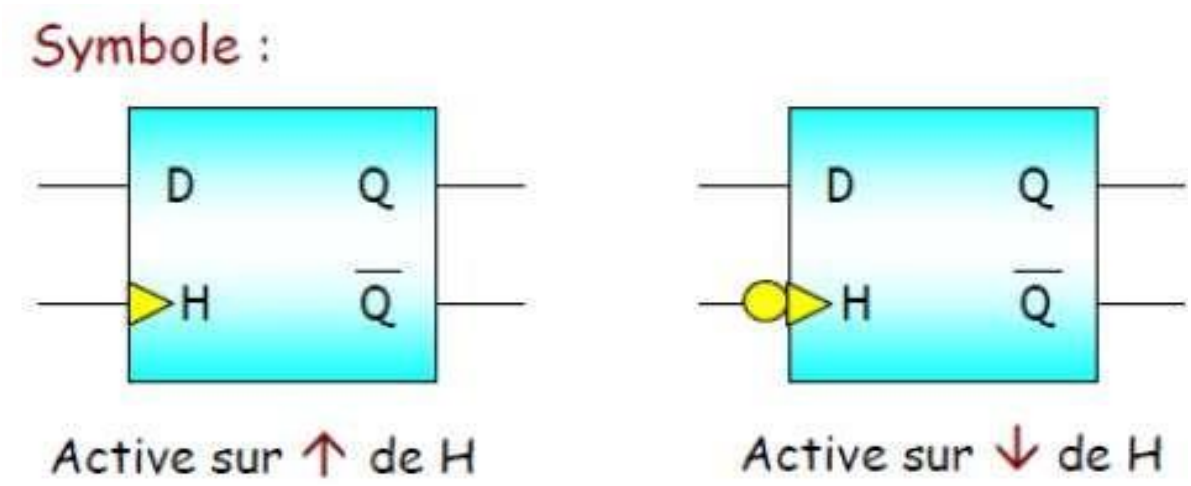


H	Q+
0/1/↓	Q-
↑	D

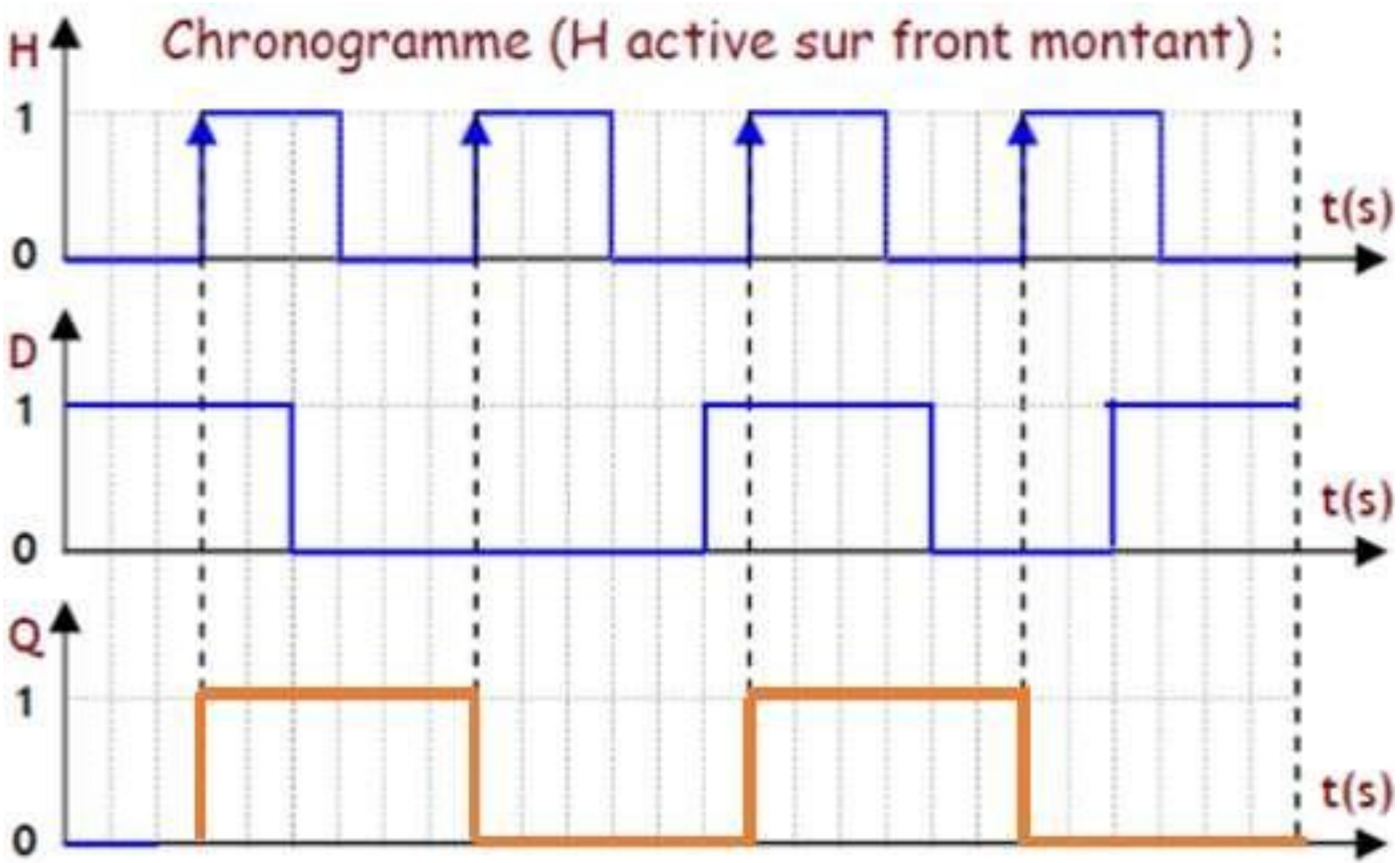
H	D	Q+
0/1/↓	0	Q-
0/1/↓	1	Q-
↑	0	0
↑	1	1

A flip-flop or edge-controlled D flip-flop

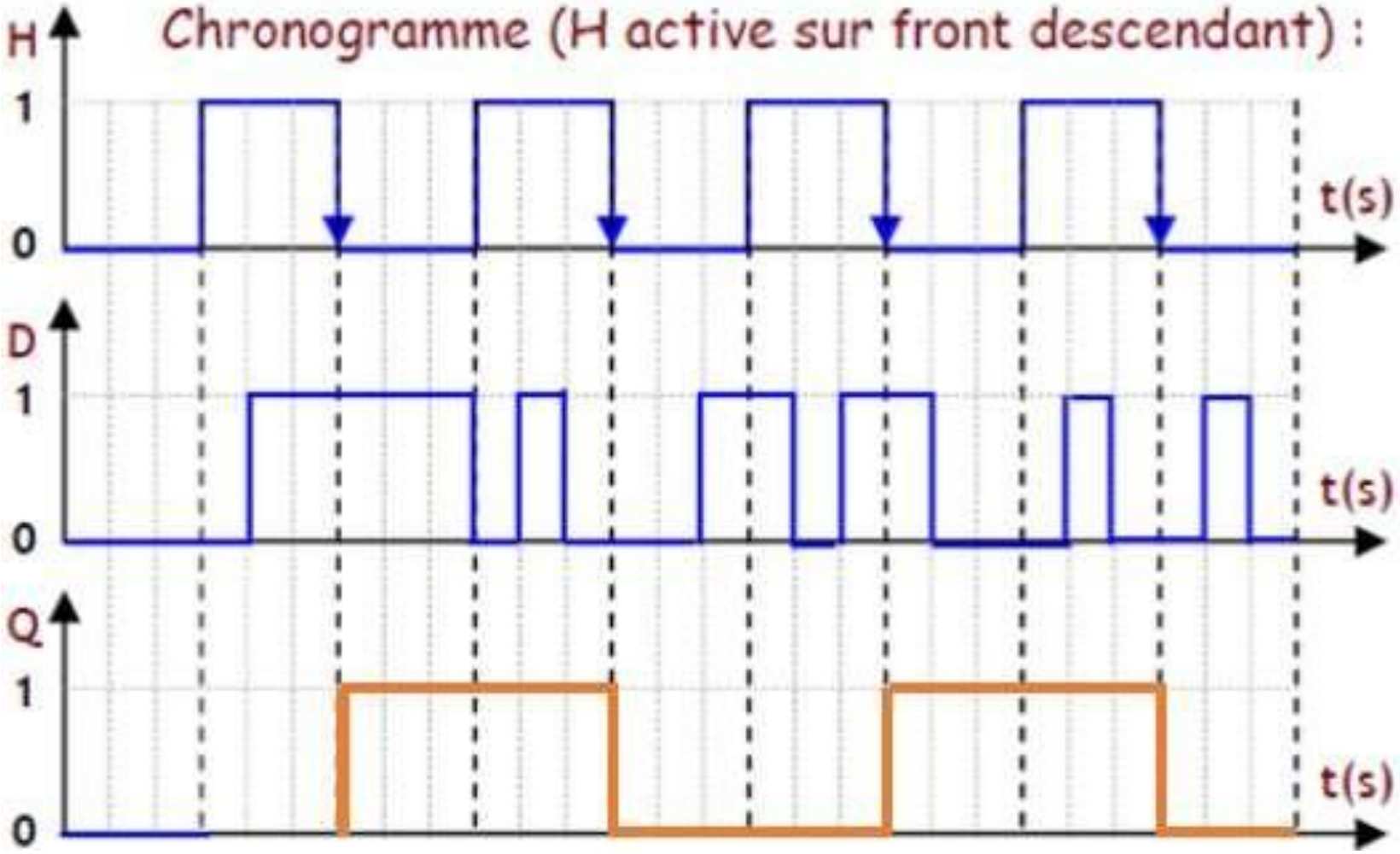
- It is a dynamic synchronous flip-flop on the edge of the clock, whose operation is as follows:
- ✓In the presence of the active edge of the clock, the flip-flop copies the logic state of input D onto output Q.
- ✓In the absence of the active edge of the clock, the flip-flop memorizes its logical state of the output Q.



A flip-flop or rising edge controlled D flip-flop

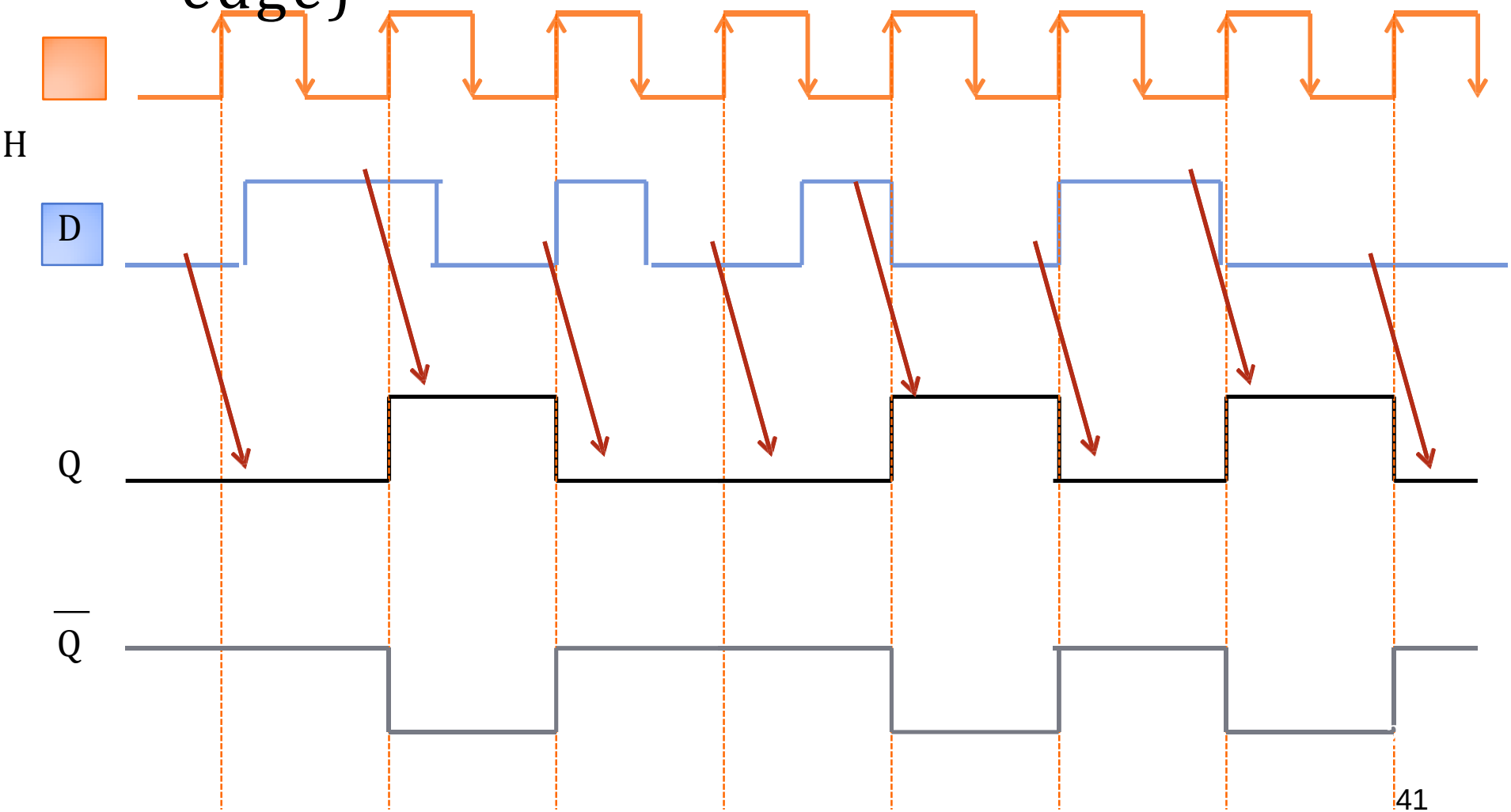


A flip-flop or falling edge controlled D flip-flop



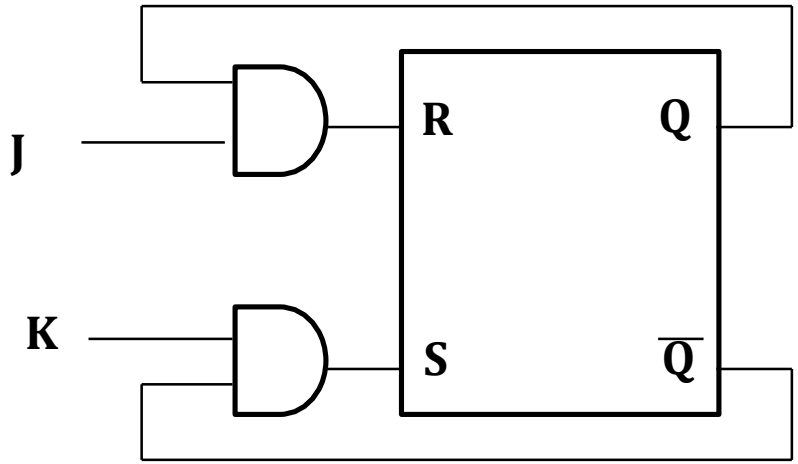
D Flip-Flop

Timing diagram: CHRONOGRAMME (Rising edge)



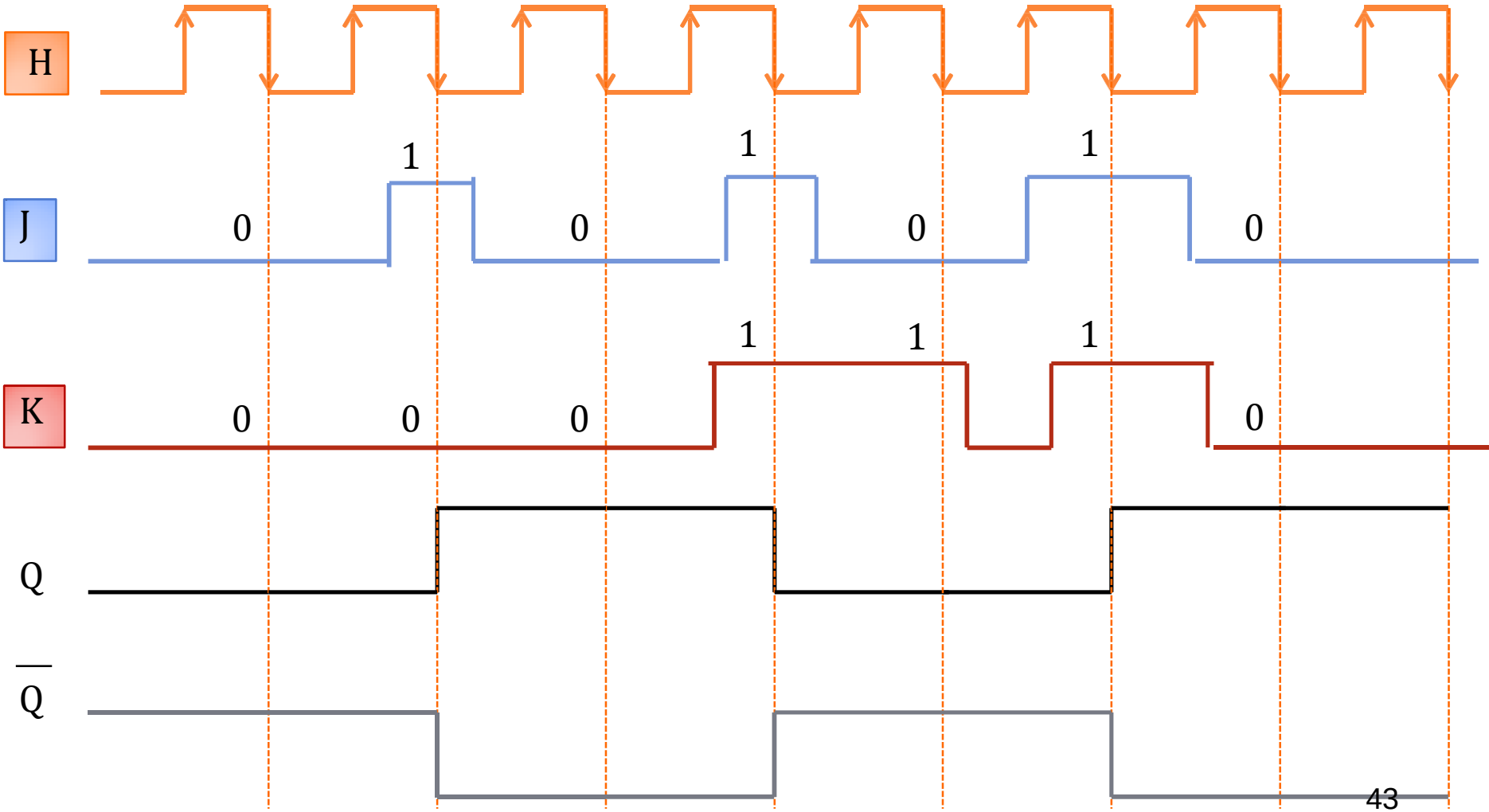
5.5. JK Flip-Flop

The JK flip-flop is a development of the SR flip-flop. The indeterminate case of the latter (S and R equal to 1) is defined in the JK flip-flop, by an inversion of the states of the outputs by enslaving (attaching) the inputs R and S to the outputs $S=J\bar{q}$ et $R=Kq$



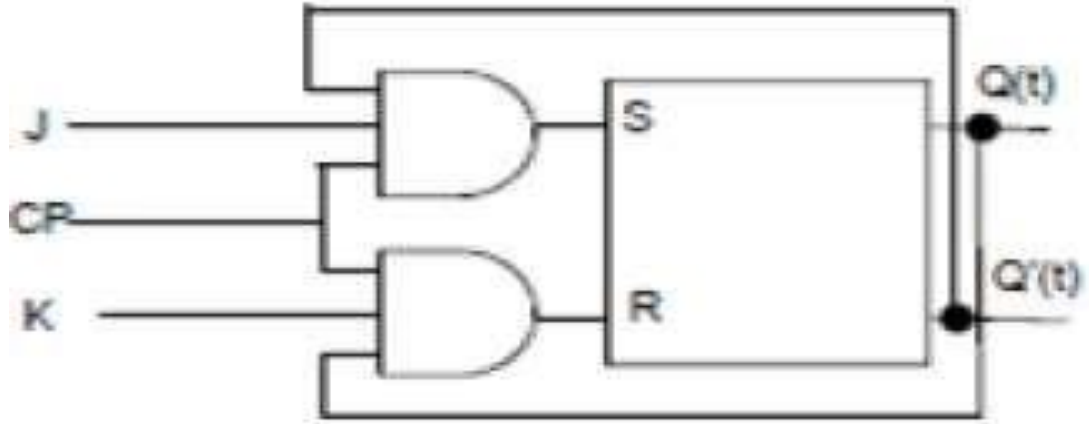
JK Flip-Flop

Timing diagram: CHRONOGRAMME (Falling edge)



JK Flip-Flop

The JK flip-flop therefore makes it possible to remove the ambiguity that exists for the S=R=1 combination of the SR flip-flop.



Equation des sorties

Q \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$$Q(t+1) = J\overline{Q} + \overline{K}Q$$

JK Flip-Flop

The JK flip-flop therefore makes it possible to remove the ambiguity that exists for the S=R=1 combination of the SR flip-flop.

J	K	Q(t)	S	R	Q(t+1)	Comment	Commentaire
0	0	0	0	0	Q(t)	Q(t)	Pas de changement
0	0	1	0	0	Q(t)		
0	1	0	0	0	Q(t)=0	0	Mise à zéro
0	1	1	0	1	0		
1	0	0	1	0	1	1	Mise à 1
1	0	1	0	0	Q(t)=1		
1	1	0	1	0	1	Q'(t)	État inversé
1	1	1	0	1	0		

JK Flip-Flop

The JK flip-flop is a dynamic synchronous flip-flop with two inputs controlling the state of the flip-flop, and a synchronization input H or Ck:

1. The input of the J engagement (Jump, jump to one): plays the role of the S input of the SR flip-flop.
2. Trigger input K (Kill, set to 0): plays the role of input R of the SR flip-flop.
3. In the absence of the clock signal, the flip-flop retains the previous state of the Q output (memorization).
4. For the combination J=K=0, the flip-flop memorizes the state of the output Q at each active clock edge (rising or falling edge).

JK Flip-Flop

- ✓ When JK =00, the output Q copies the state of the input at each active clock edge (memorisation).
- ✓ For the combination JK=10, the Q output is set to 1 at each active clock edge.
- ✓ For the combination JK=01, the output Q is set to 0 at each active clock edge.
- ✓ When acting simultaneously on J and K (J=K=1), the flip-flop changes state at each active clock edge, this is the toggle mode.

Synchronous JK flip-flop with the clock activated on rising edge

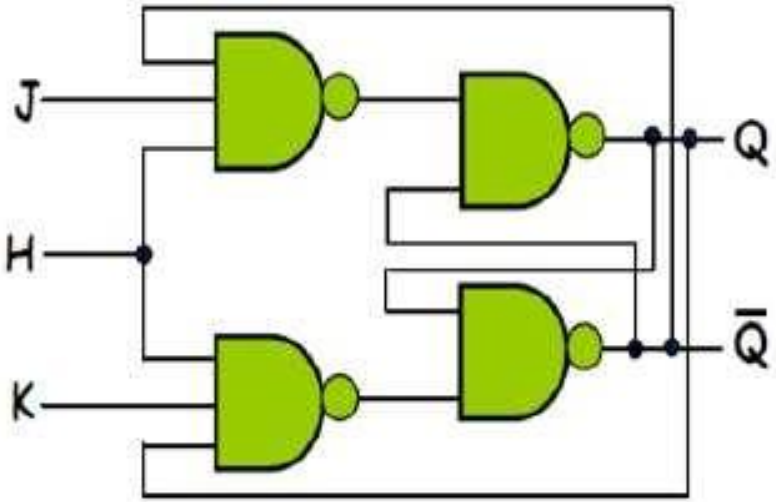


Table de vérité				
H	J	K	Q	Commentaire
↑	0	0	q	Mémorisation
↑	0	1	0	Mise à 0 de la sortie Q
↑	1	0	1	Mise à 1 de la sortie Q
↑	1	1	$\bar{q}$	Basculement de l'état de sortie

JK Flip-Flop

ASYNCHRONOUS

It is a flip-flop variant of SR where we take into account the case where R=S=1

J	K	Q+	
0	0	Q-	Memory state
0	1	**0**	Reset to 0
1	0	**1**	Reset to 1
1	1	$\overline{Q}$-	Toggle (Basculement)

JK Flip-Flop

SYNCHRONOUS

It is a flip-flop with two inputs J and K and a clock (rising or falling edge)

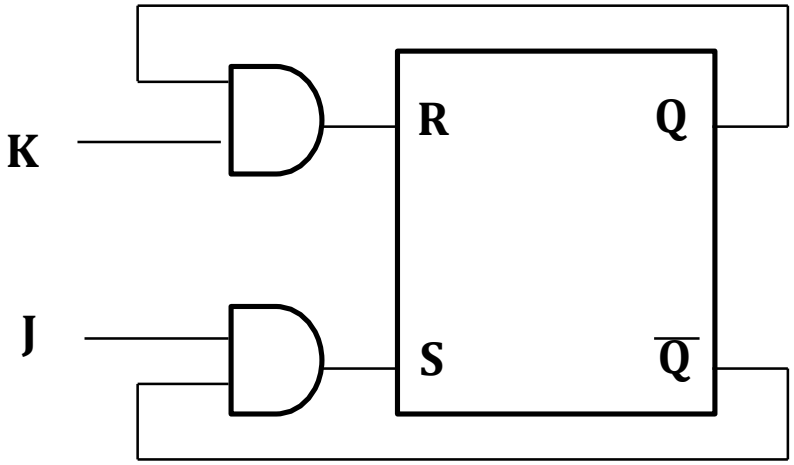
On rising edge↑

On falling edge↓

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

JK Flip-Flop

Exercise: Realize an asynchronous JK flip-flop using an SR flip-flop.



JK Flip-Flop

Exercise : Realize an asynchronous JK flip-flop using an SR 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**

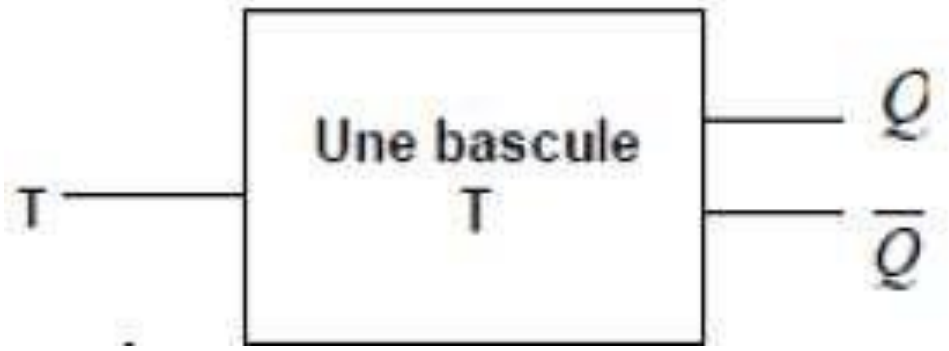
5.6. T Flip-Flop

The T flip-flop (toggle, Trigger) operates on a clock edge.

It allows to keep the value of the previous output or to invert it (complement or not the current state).

This type of this flip-flop is particularly interesting for making counters.

The T flip-flop can be made from a D flip-flop.



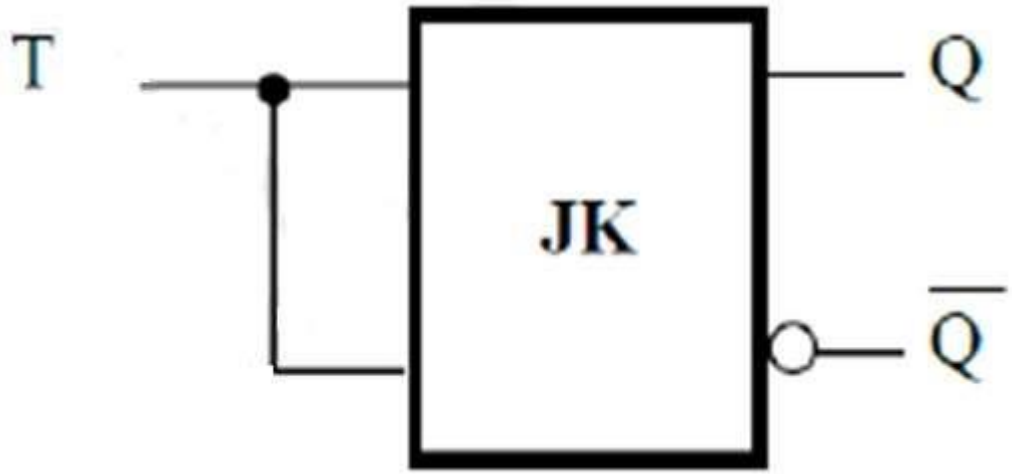
T		Q+
0		q
1		$\bar{q}$

T Flip-Flop

For a J-K flip-flop, we see that if J=K=1, the state of the output is inverted every clock cycle.

Thus, the flip-flop T has a single input called “Trigger”, which can be obtained by connecting inputs J and K to the same source.

This is why it is sometimes called the complement toggle.



T Flip-Flop

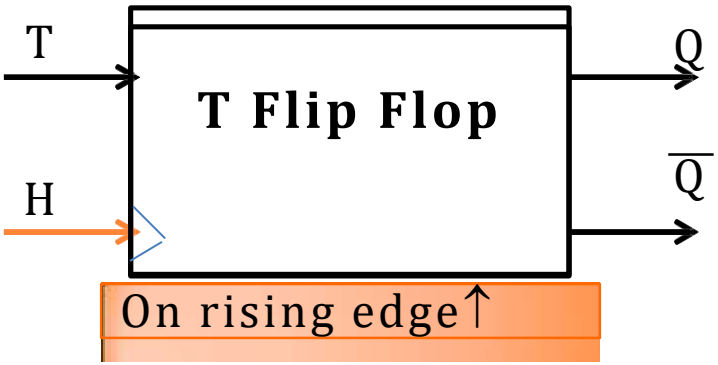
The synchronous flip-flop T is triggered by the clock signal H.

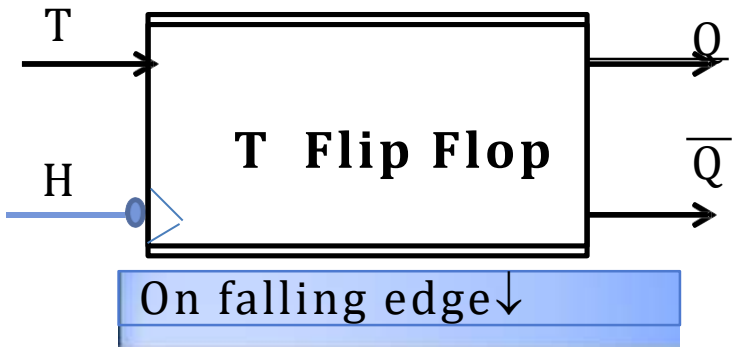
The single T (Trigger) input controls the state of the flip-flop.

The Q output changes state each time the T input goes to logic state 1 and maintains its state the rest of the time.

T Flip-Flop

The T flip-flop (Toggle) switches on each clock pulse (rising or falling edge) when its T input is active.

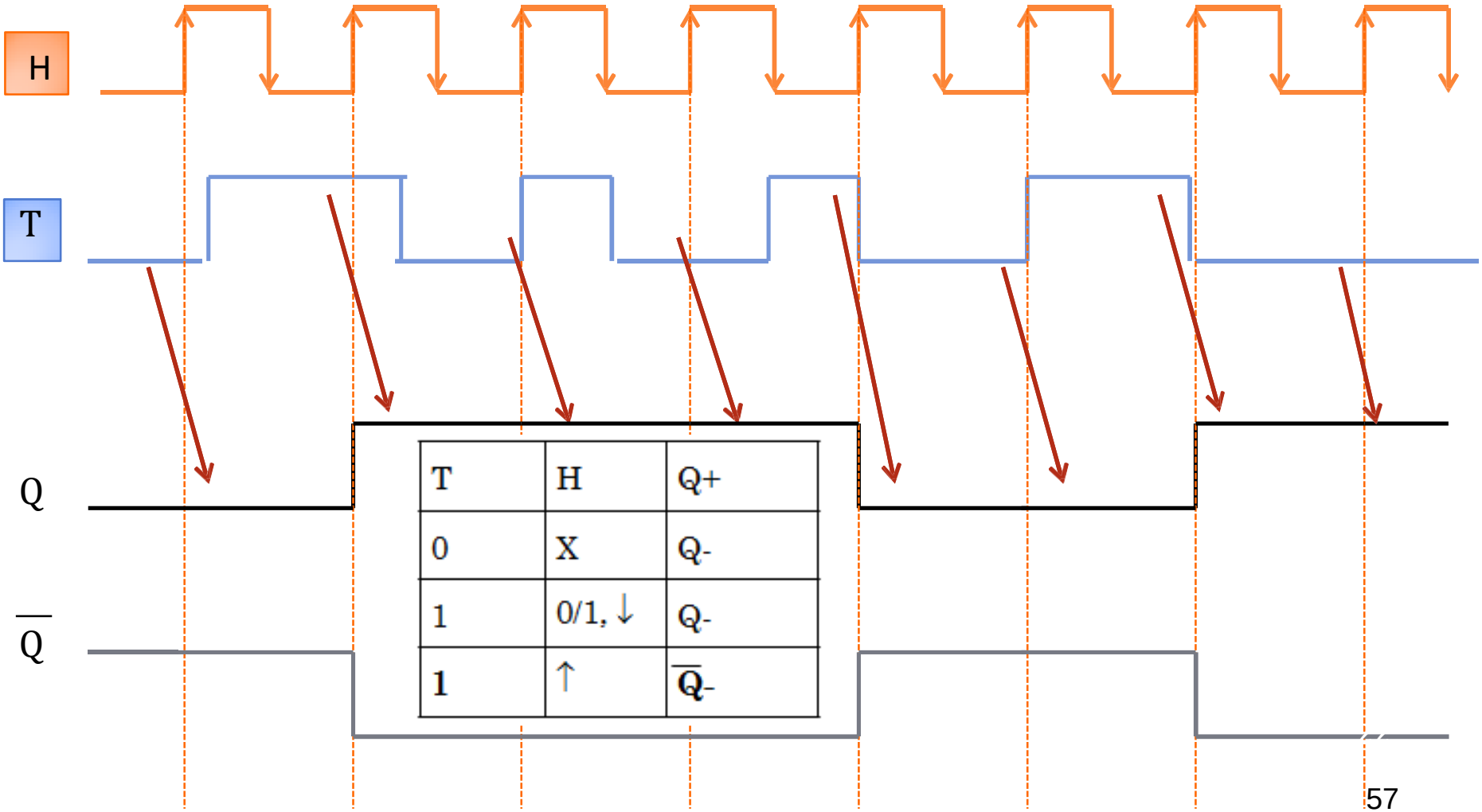




T	H	Q+
0	X	Q-
1	0/1, ↓	Q-
1	↑	$\overline{\mathbf{Q}}$-

T Flip-Flop

Timing diagram: CHRONOGRAMME (Rising edge)



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

Flip-flop excitation table

The flip-flop characteristic tables specify the next state when the inputs and the current state are known. However, during the design of a process (logic circuit), the transition states (present and next state) are often known and these are the inputs which must be determined to ensure the desired transition.

For this reason, we need a table that gives the necessary entries for a given state transition.

This table is called the excitation table of the flip-flop (or of the combinational circuit).

Flip-flop excitation table

It therefore consists of two columns Qt and Qt+1 and one column for each entry.

There are four transitions (4 lines) to go from Qt to Qt+1.

For example, for the J-K flip-flop, to obtain the transition 0 to 1, input J must be in state 1, whatever the state of input K.

J = K = 1 which inverts the state of the flip-flop or

J = 1 and K = 0 which loads 1 into the flip-flop.

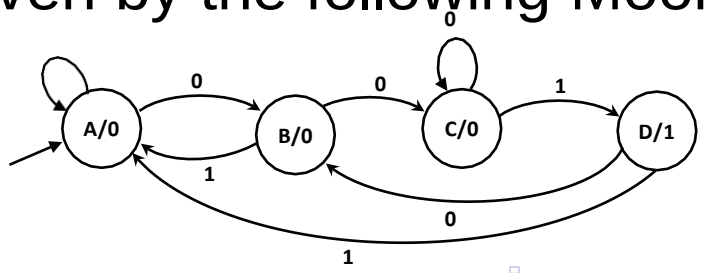
A cross (X) indicates that the state of the input considered is indifferent: 0 or 1.

Synthesis of sequential circuits

The synthesis of a sequential circuit is the operation which allows to move from specifications describing the operation of the system to the corresponding logic circuit.

The operation described in this specification can be represented by a graph called a state graph.

Example: Synthesis of a sequential circuit which detects the sequence 001 using D flip-flops. The state graph of this circuit will be given by the following Moore graph:



The registers

A register is a sequential circuit made up of n flip-flops placed in series in order to store binary information on n bits.