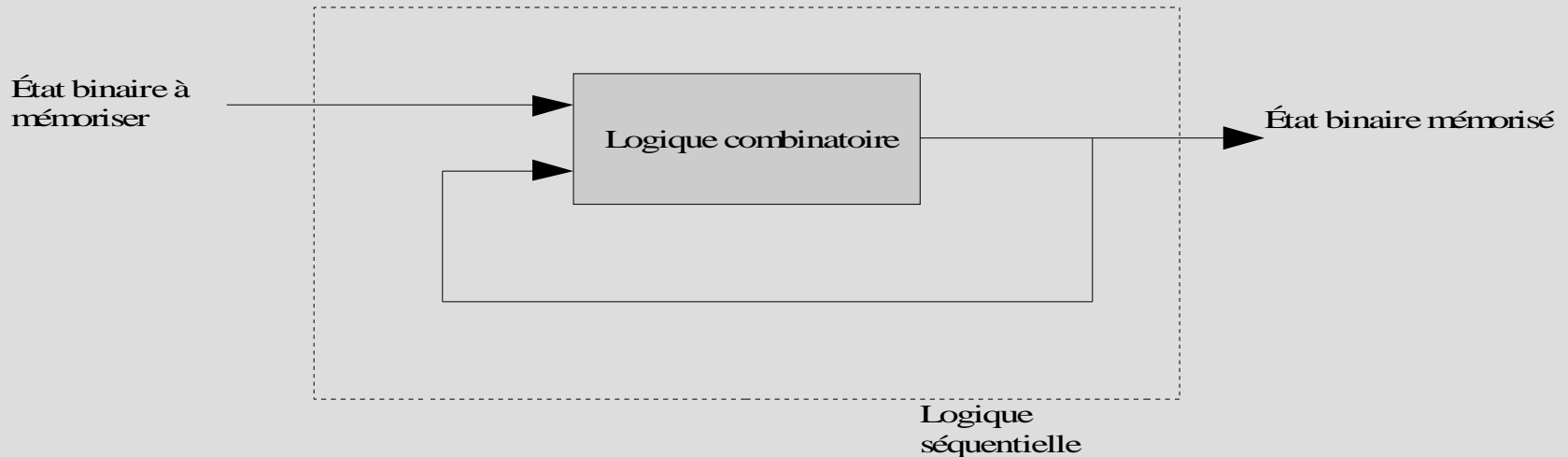


# **Circuits séquentiels**

# La Logique séquentielle

- **Contrairement à la logique combinatoire elle permet de mémoriser des états binaires.**

- Principe :

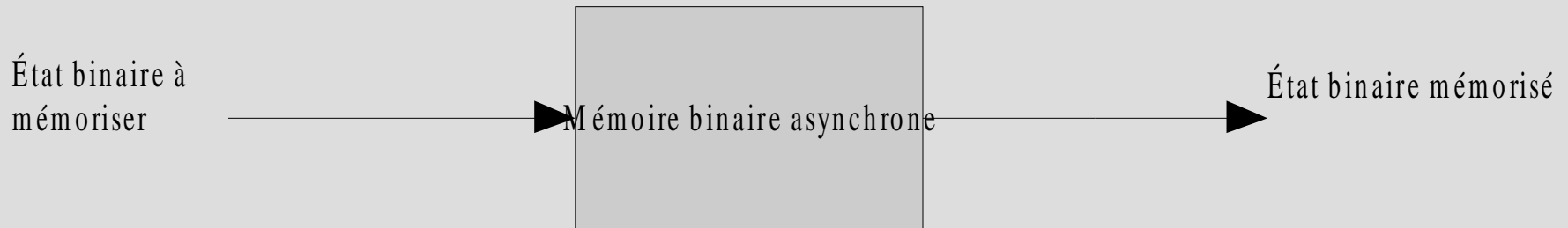


- **Pour déterminer l'état présent en sortie, il faut :**
  - **L'état sur l'entrée**
  - **L'état passé de la sortie**

# Deux types de logique séquentielle

## ➤ **Asynchrone :**

- **Le système mémorise à tout moment l'état présent sur son entrée :**



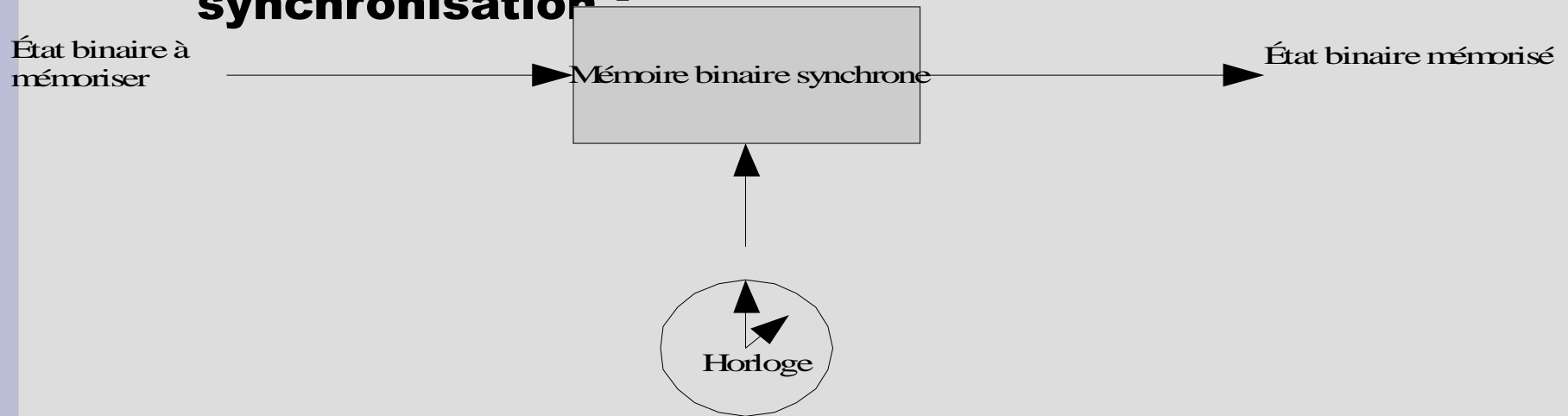
### • **Applications :**

- **Mémoriser un état binaire isolé (bouton poussoir)**
- **Utilisé dans les vieux automates câblés.**

# Deux types de logique séquentielle

## ➤ **Synchrone :**

- **Le système mémorise l'état présent sur son entrée si et seulement si une horloge fournit un signal de synchronisation**

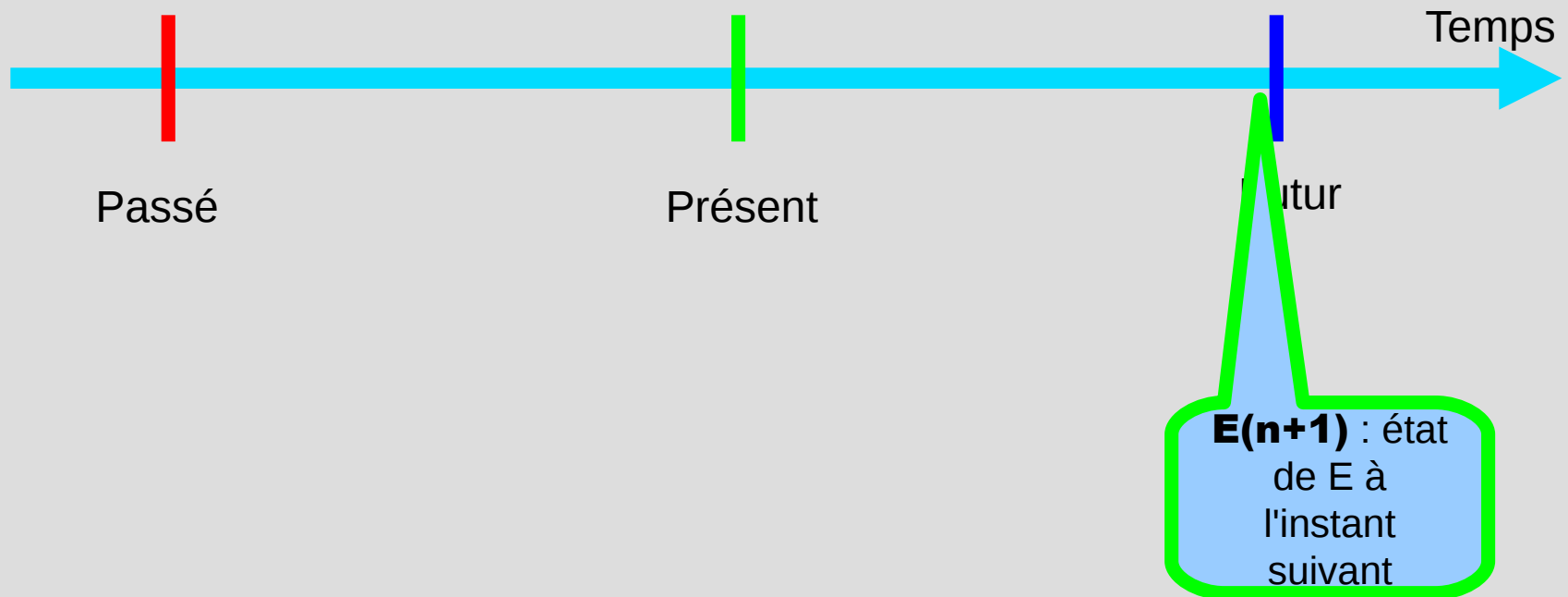


## • **Applications :**

- **Ordinateurs, consoles de jeux**
- **Montres électroniques.**

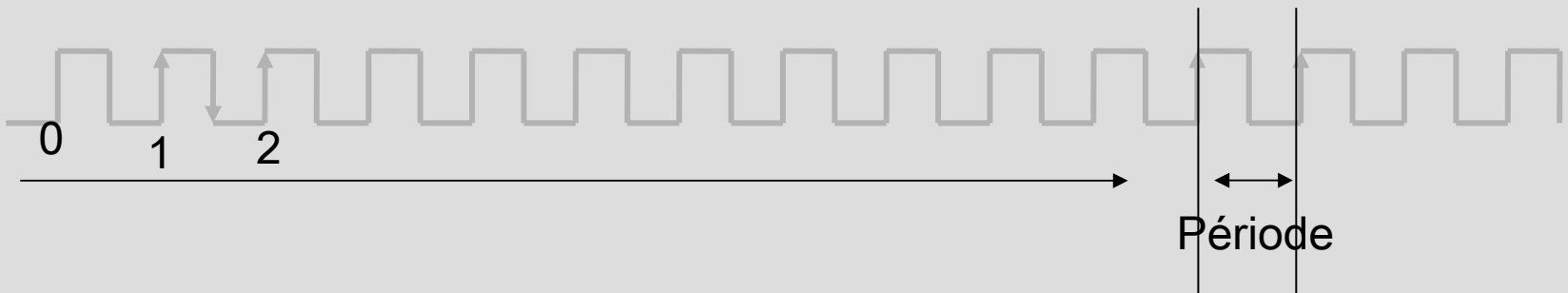
# Notation des états

- Comment noter un état « E » dans le passé, le présent ou le futur?
  - Notion de signal ( en VHDL)



# Horloge (Clock)

- **Horloge : composant passant indéfiniment et régulièrement d'un niveau haut à un niveau bas (succession de 1 et de 0), chaque transition s'appelle un *top*.**



**Fréquence = nombre de changement par seconde en hertz (Hz)**

**Fréquence =  $1/\text{période}$**

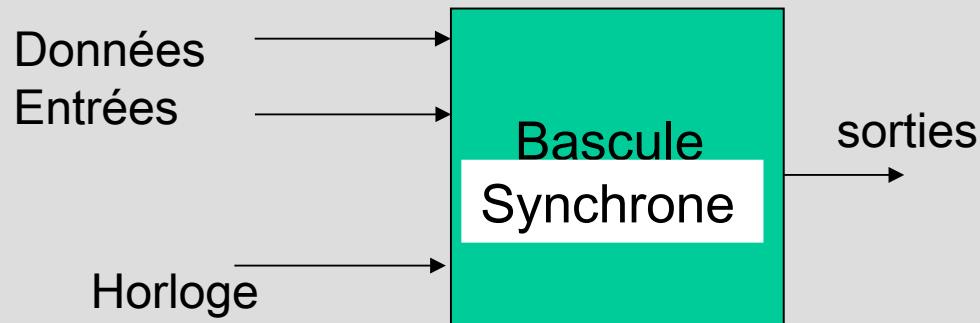
**Une horloge de 1 hertz a une période de 1 seconde**

.....1 megahertz.....1 milliseconde

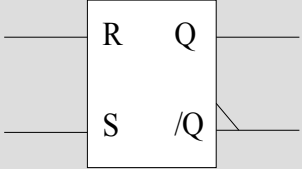
.....1 gigahertz.....1 nanoseconde

# Basculer Synchrones/Asynchrones

- Les bascules synchrones : asservies à des impulsions d'horloge et donc insensibles aux bruits entre deux tops
- Les bascules asynchrones, non asservies à une horloge et prenant en compte leurs entrées à tout moment.

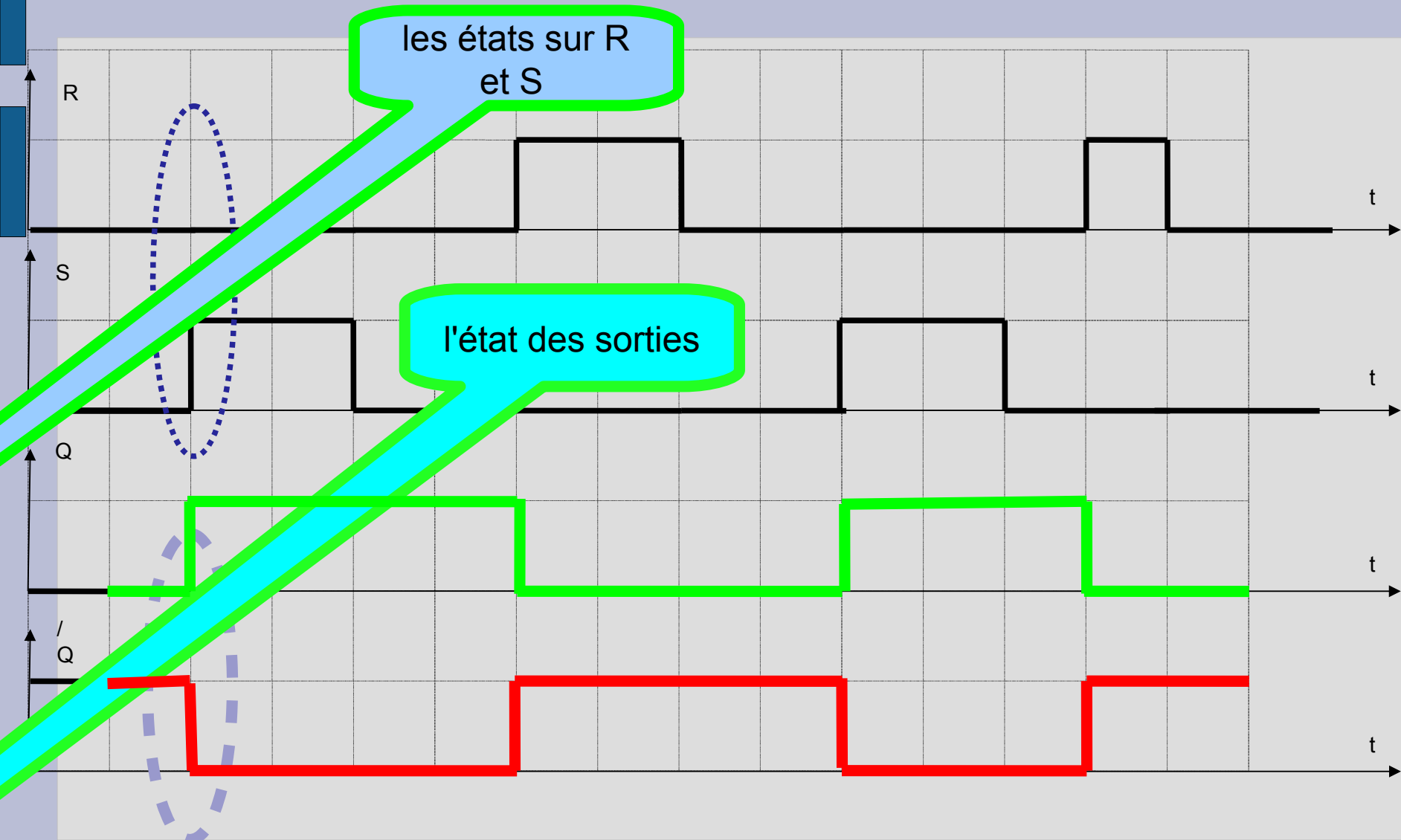


# Bascule asynchrone : bascule RS

<i>Symbole Européen</i>				<i>Explications</i>
				<ul style="list-style-type: none"> <li>Entrées : <ul style="list-style-type: none"> <li>R : remise à zéro (Reset)</li> <li>S : mise à un (Set)</li> </ul> </li> <li>Sorties : <ul style="list-style-type: none"> <li>Q : sortie principale</li> <li>/Q : sortie complémentée</li> </ul> </li> </ul>
<i>Table de vérité</i>				<i>Modes de fonctionnement</i>
R	S	Q(n)	/Q(n)	
0	0	Q(n-1)	/Q(n-1)	→ État mémoire
0	1	1	0	→ Mise à 1 (mémorisation)
1	0	0	1	→ Mise à 0 (mémorisation)
1	1	??	??	→ INTERDIT !

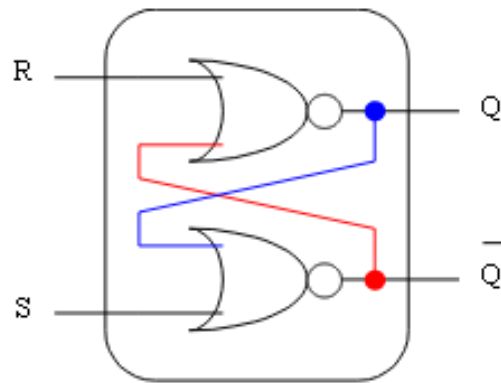


# Bascule RS : chronogramme



# Bascule RS avec des portes

Rappel sur le Nand  
 0 NAND X = 1  
 1 NAND X = Non X



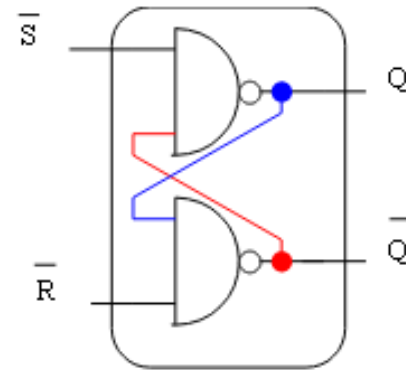
**Technologie NON-OU :**

Si  $R = S = 1$  alors  $Q = 0$  :

$$Q = \overline{R + \overline{Q}} \quad \& \quad \overline{Q} = \overline{S + Q}$$

$$Q(t) = \overline{R + S + Q(t-1)}$$

$$Q(t) = \overline{\overline{R} \cdot (S + Q(t-1))}$$



**Technologie NON-ET :**

Si  $S = R = 1$  alors  $Q = 1$  :

$$Q = \overline{\overline{S} \cdot \overline{Q}} \quad \& \quad \overline{Q} = \overline{\overline{R} \cdot Q}$$

$$Q(t) = \overline{\overline{S} \cdot \overline{R} \cdot Q(t-1)}$$

$$Q(t) = \overline{S + \overline{R} \cdot Q(t-1)}$$

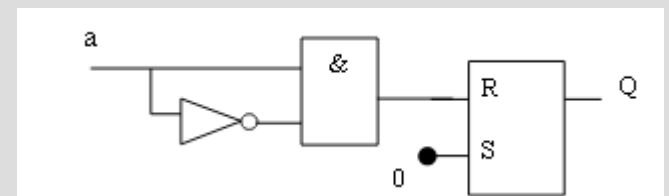
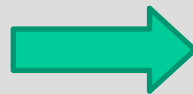
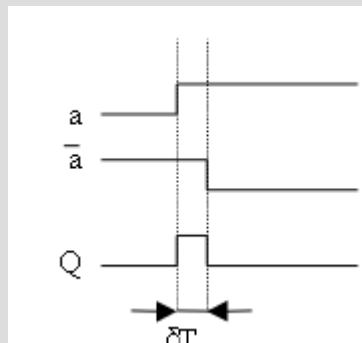
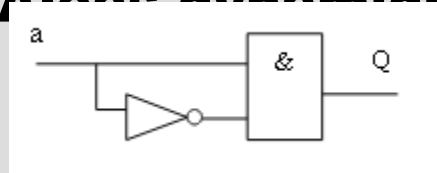
# Utilisation d'un RS

## ➤ Réduire les aléas des circuits

### ➤ Aléas statiques



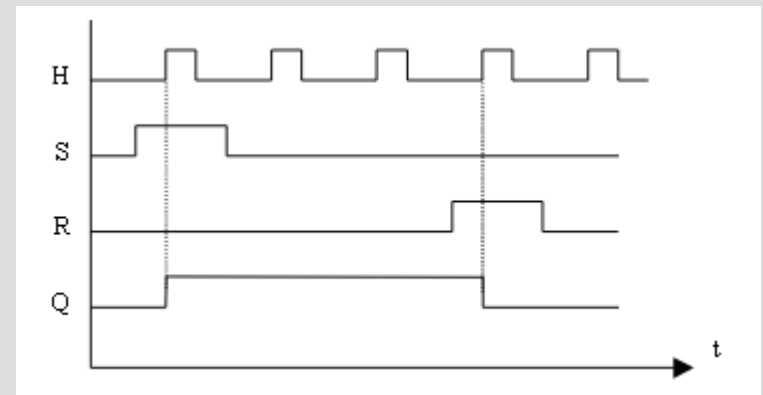
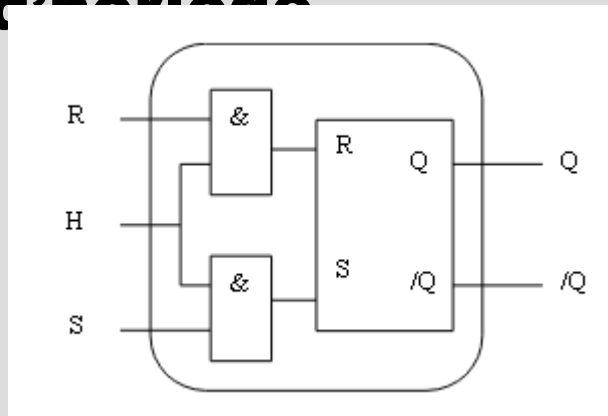
### ➤ Aléas dynamiques



La bascule RS mémorise l'état stable précédent pendant l'état transitoire

# Bascule RS synchrone

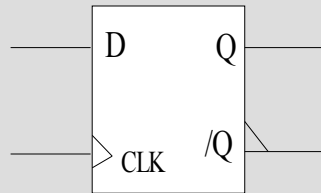
- **On synchronise les entrées avec les impulsions d'horloge**



# Bascule synchrone : bascule D

La base de la  
mémoire d'un  
ordinateur

Front  
montant



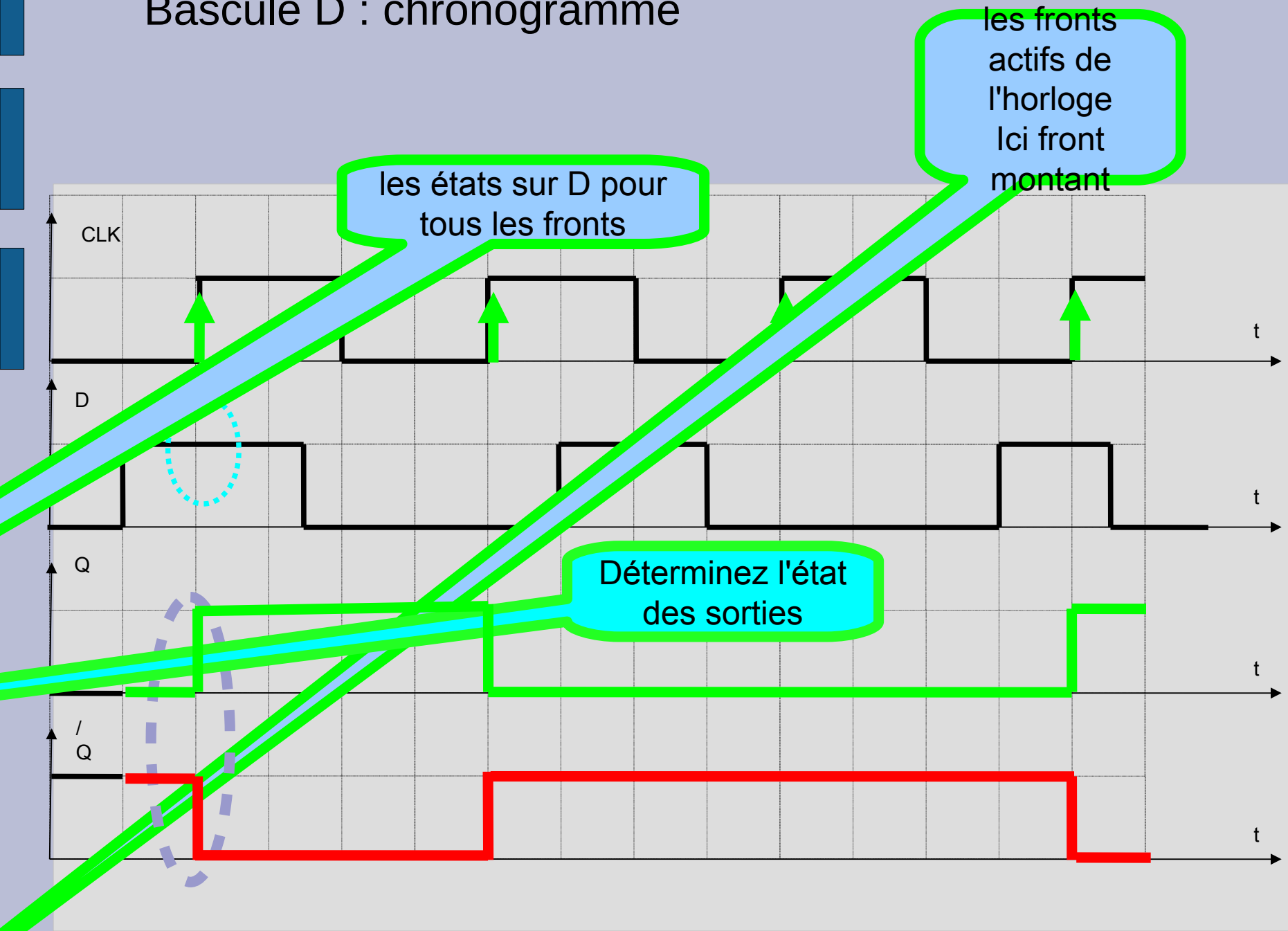
- Entrées :
- **D : état à mémoriser (Data)**
  - **CLK : horloge de synchronisation**
- Sorties :
- **Q : sortie principale**
  - **/Q : sortie complémentaire**

Table de vérité

Modes de fonctionnement

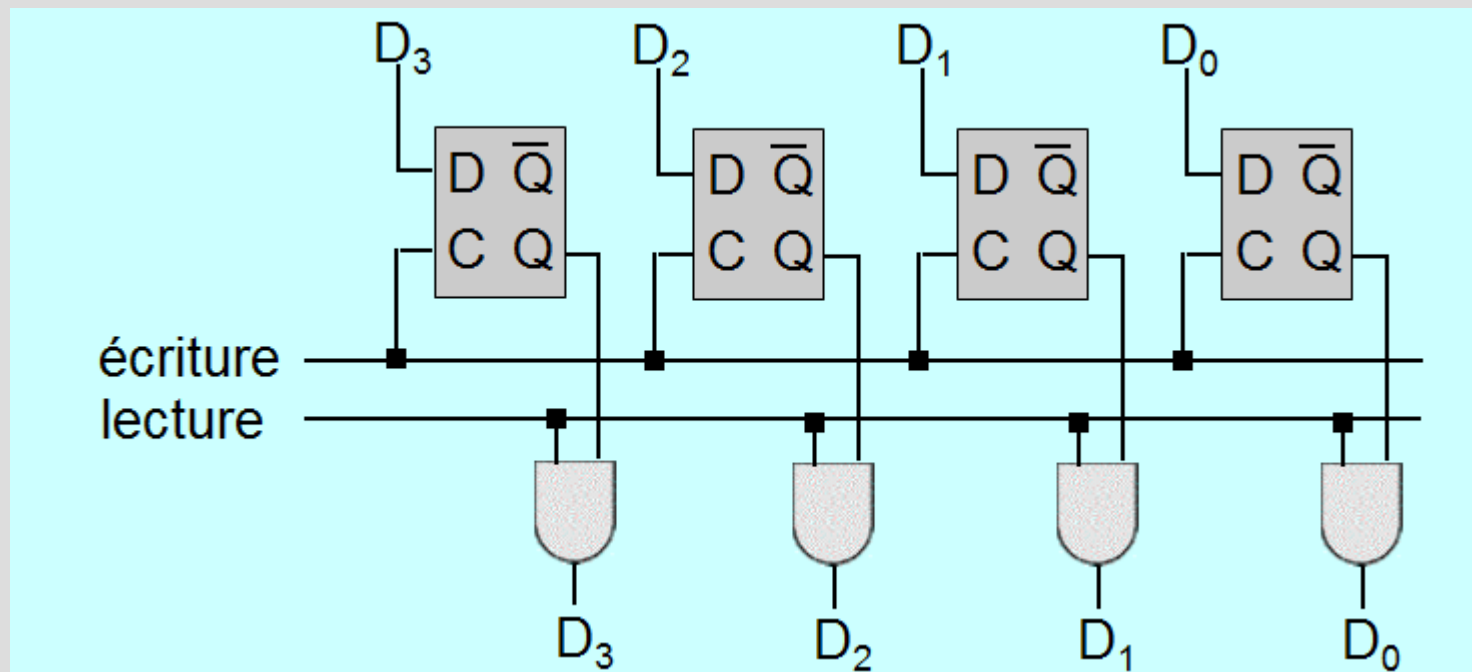
CLK	D	Q(n)	/Q(n)	
0	X	Q(n-1)	/Q(n-1)	➡ État mémoire
1	X	Q(n-1)	/Q(n-1)	➡ État mémoire
↑	0	0	1	➡ Mémorisation d'un état 0
↑	1	1	0	➡ Mémorisation d'un état 1

# Bascule D : chronogramme

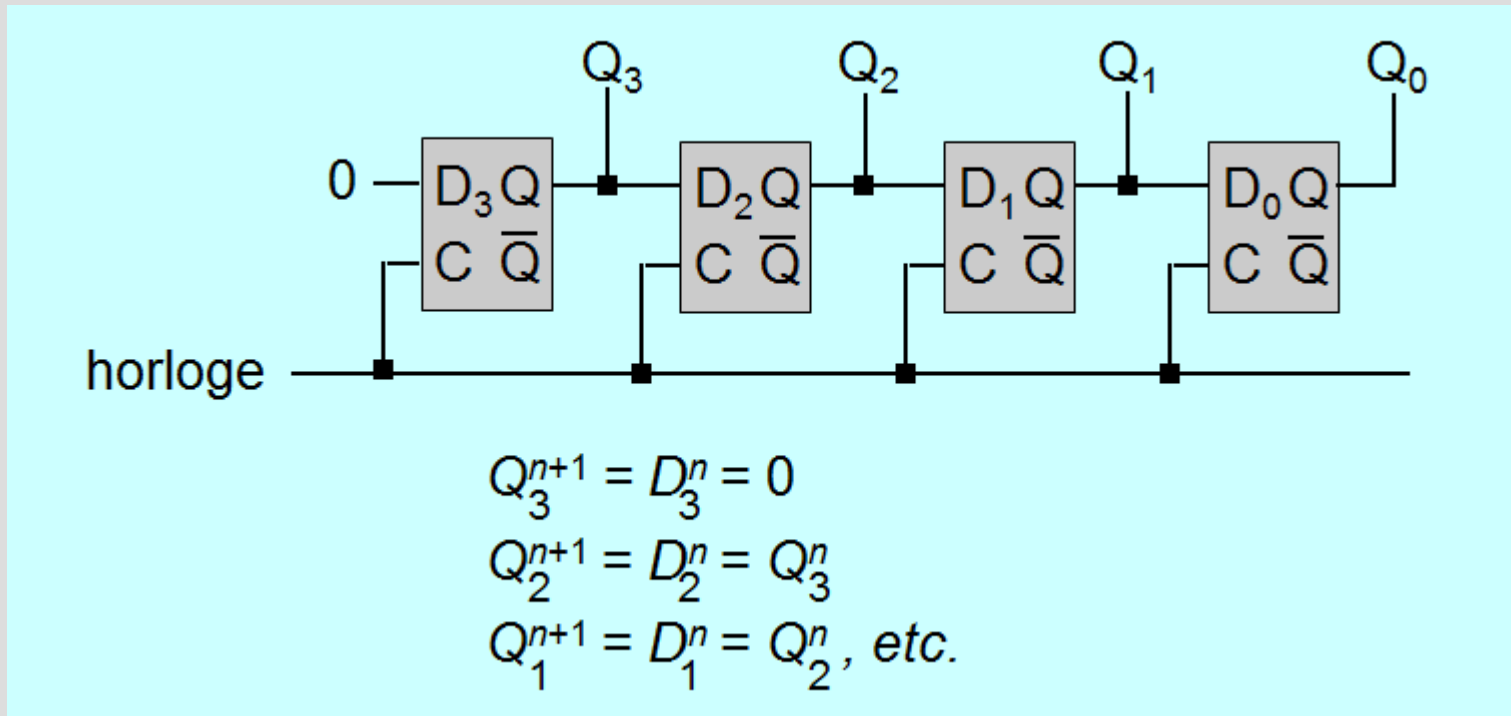


# Utilisation d'une bascule D

## ➤ **Registre à 4 bits**



## ➤ Décaleur à droite





# Compteur/diviseur

- Une bascule D qui change d'état à chaque top d'horloge

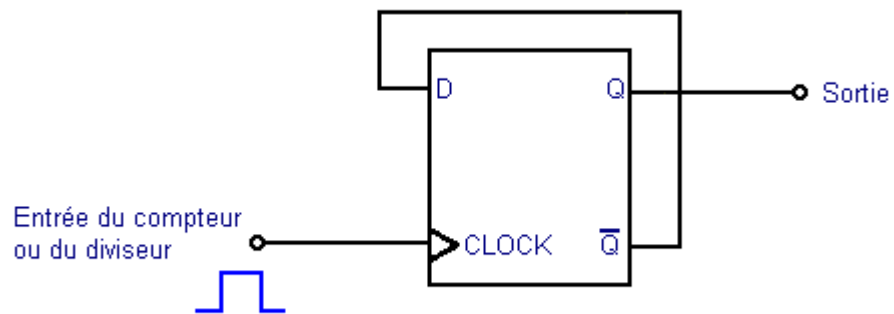


Fig. 5. - Bascule de type D raccordée en diviseur par deux.

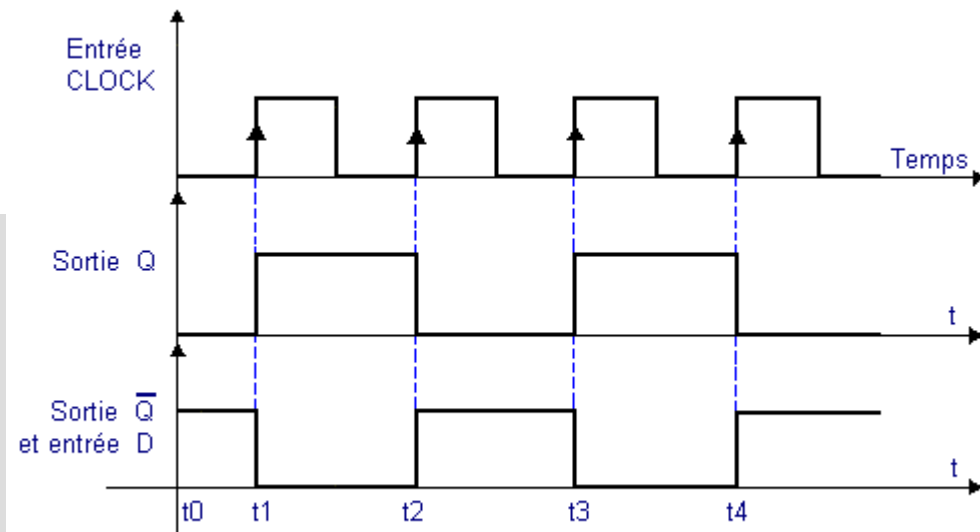


Fig. 6. - Chronogramme relatif au fonctionnement du diviseur par 2.

# Compteur modulo 4

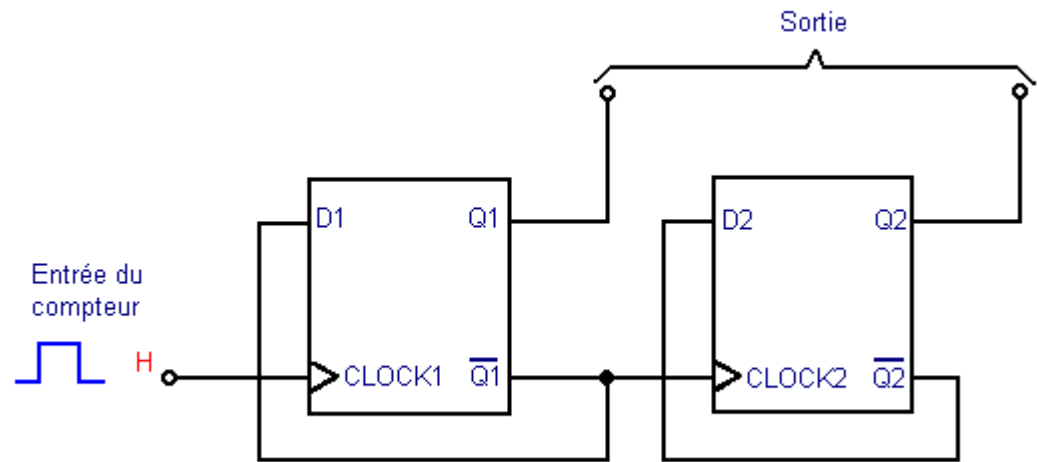


Fig. 9. - Compteur modulo 4.

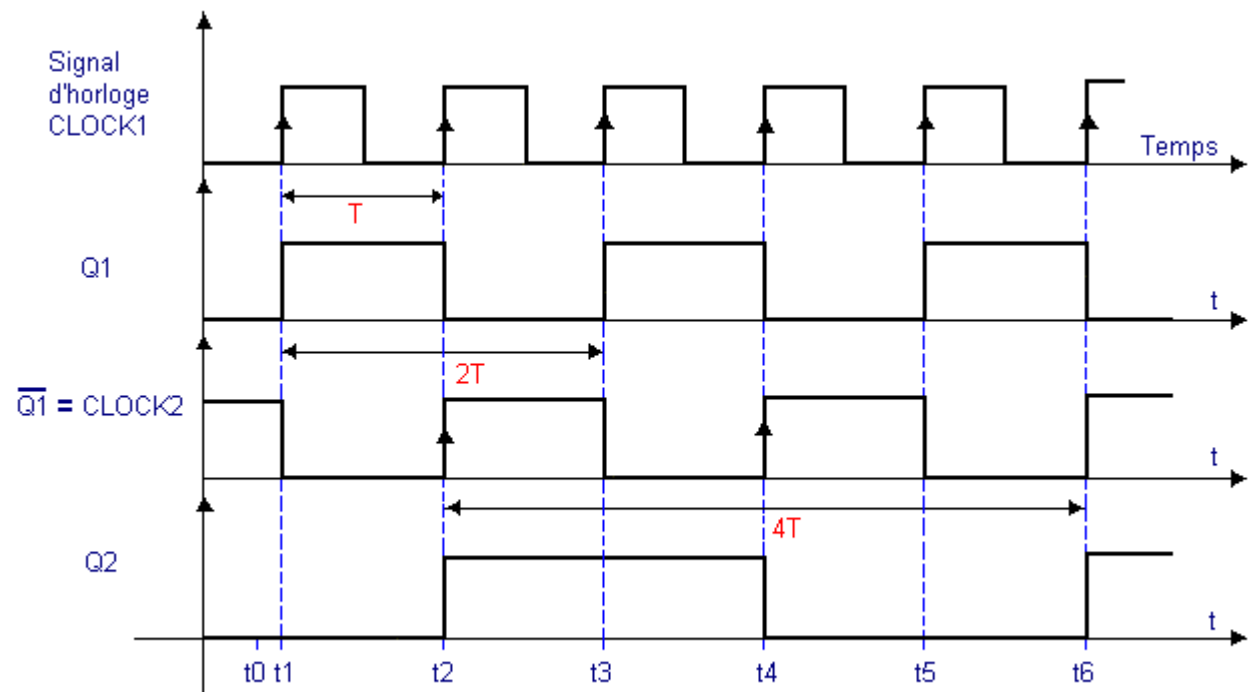


Fig. 10. - Chronogramme relatif au fonctionnement du diviseur par 4.

# Modulo 8

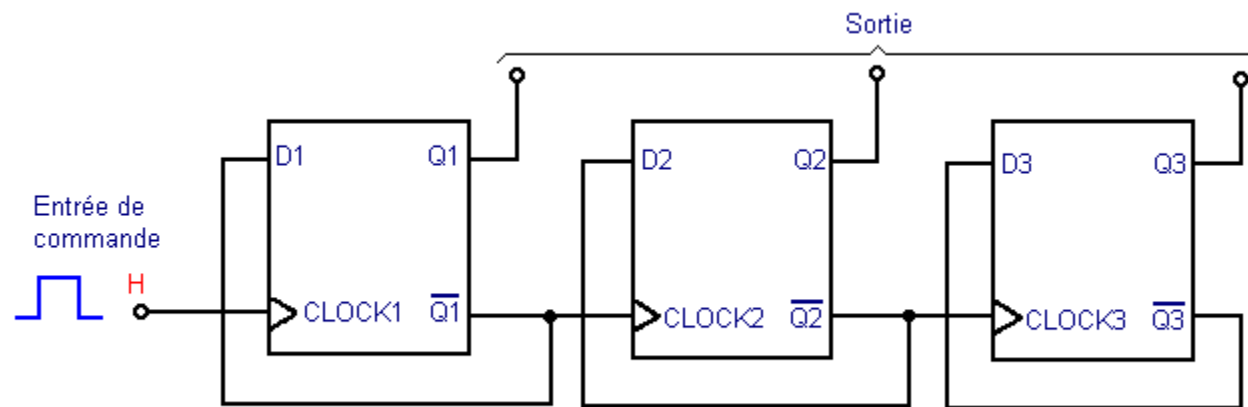
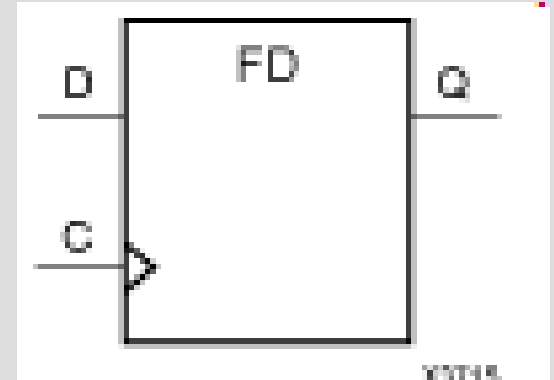


Fig. 12. - Compteur modulo 8.

# FD

## Primitive: D Flip-Flop

This design element is a D-type flip-flop with data input (D) and data output (Q). The data on the D inputs is loaded into the flip-flop during the Low-to-High clock (C) transition. This flip-flop is asynchronously cleared, outputs Low, when power is applied.



### Inputs

**D**

**C**

**0**



**1**



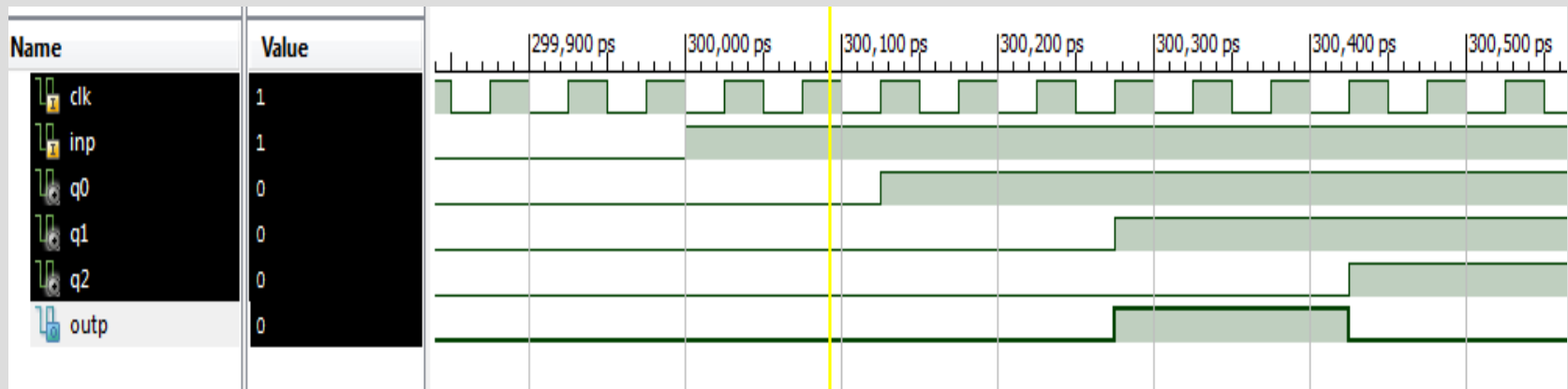
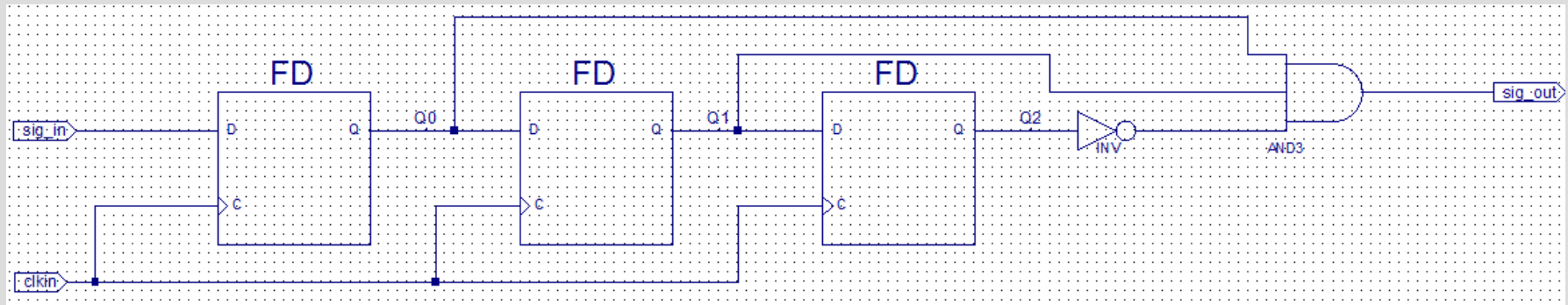
### Outputs

**Q**

**0**

**1**

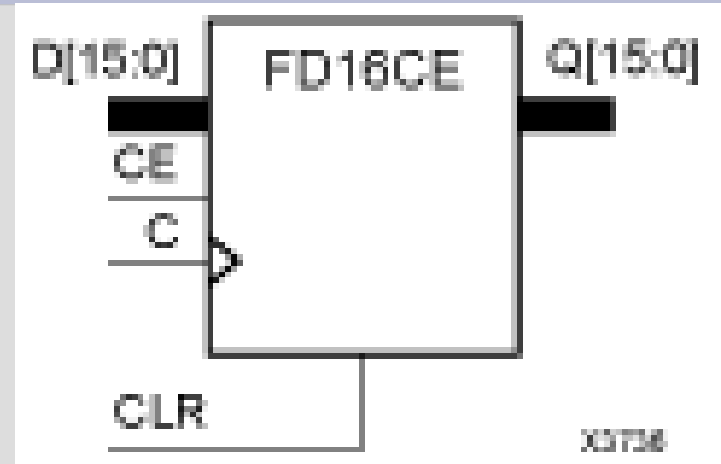
# Anti rebound



# FD16CE

## Macro: 16-Bit Data Register with Clock Enable and Asynchronous Clear

This design element is a 16-bit data register with clock enable and asynchronous clear. When clock enable (CE) is High and asynchronous clear (CLR) is Low, the data on the data inputs (D) is transferred to the corresponding data outputs (Q) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data outputs (Q) Low. When CE is Low, clock transitions are ignored. This register is asynchronously cleared, outputs Low, when power is applied.

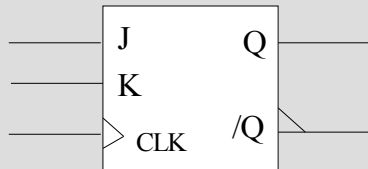


	Inputs				Outputs
	CLR	CE	Dz : D0	C	Qz : Q0
1	X		X	X	0
0	0		X	X	No Change
0	1		Dn	↑	Dn

z = bit-width - 1

# Bascule synchrone : bascule JK

*Symbole Européen*



*Explications*

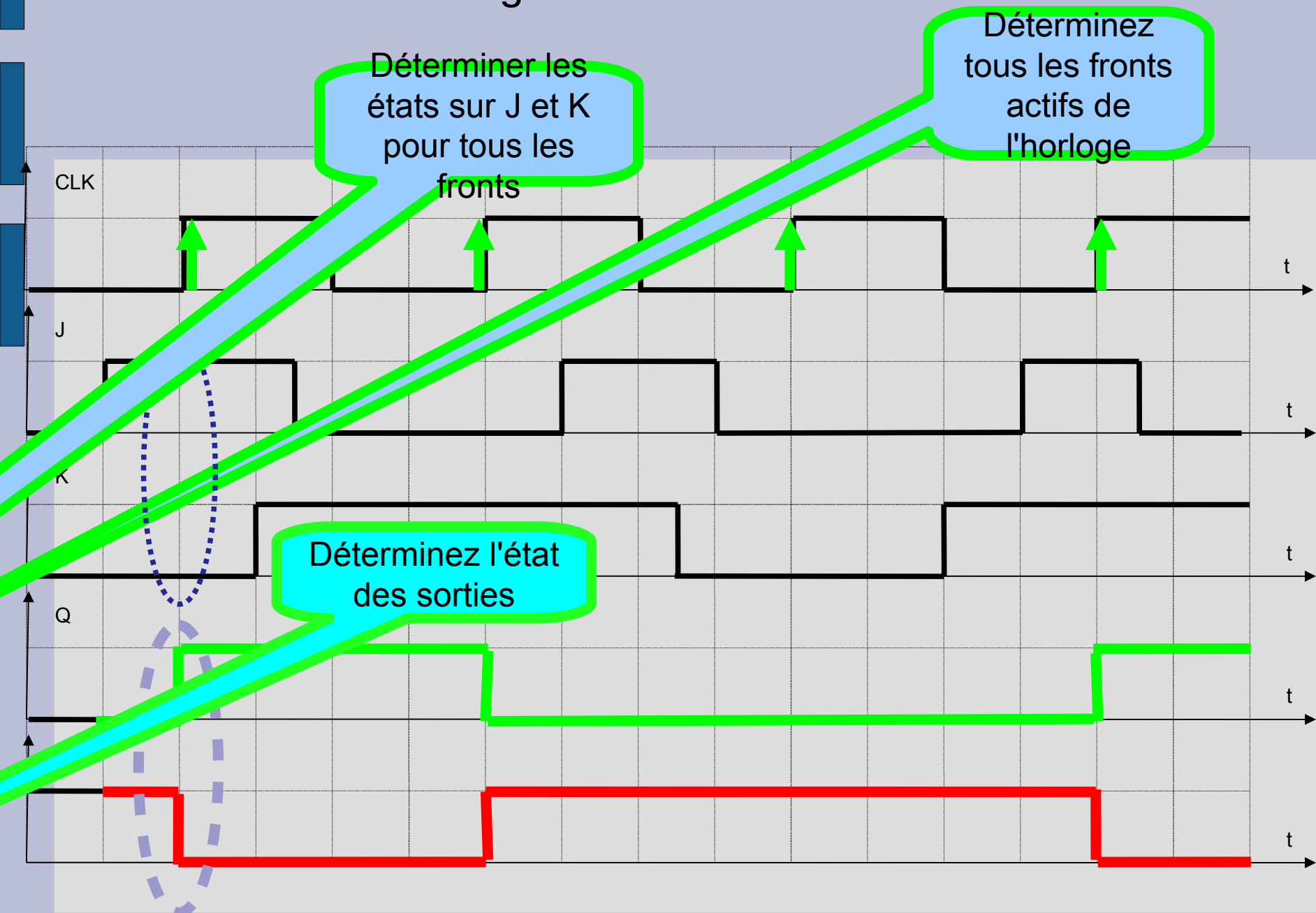
- Entrées :
  - J : entrée n°1
  - K : entrée n°2
  - CLK : horloge de synchronisation
- Sorties :
  - Q : sortie principale
  - /Q : sortie complémentée

*Table de vérité*

*Modes de fonctionnement*

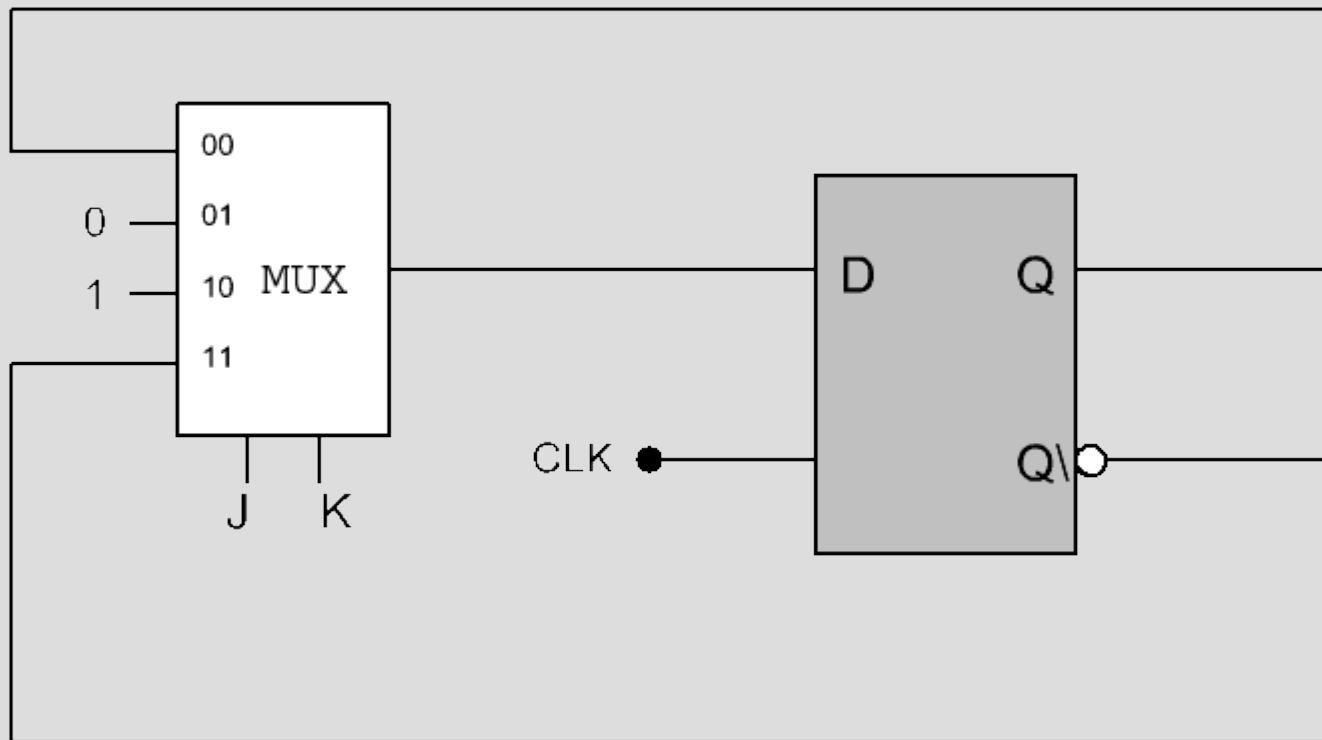
CLK	J	K	Q(n)	/Q(n)	
X	X	X	Q(n-1)	/Q(n-1)	➡ État mémoire
↑	0	0	Q(n-1)	/Q(n-1)	➡ État mémoire
↑	0	1	0	1	➡ Mémorisation de l'état 0
↑	1	0	1	0	➡ Mémorisation de l'état 1
↑	1	1	/Q(n-1)	Q(n-1)	➡ État mémoire complémenté

# Bascule JK : chronogramme

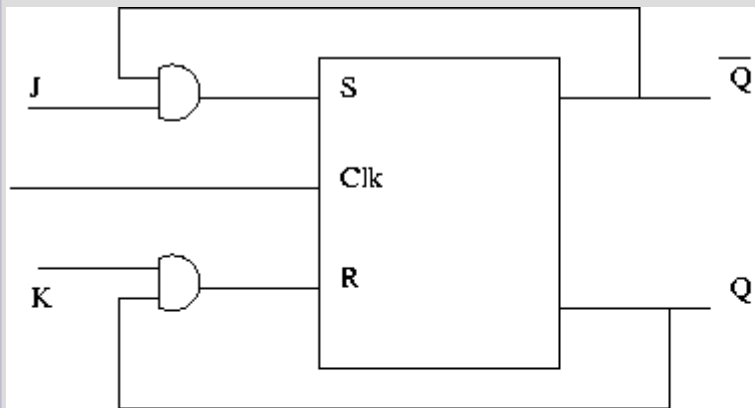




# Bascule JK avec une D

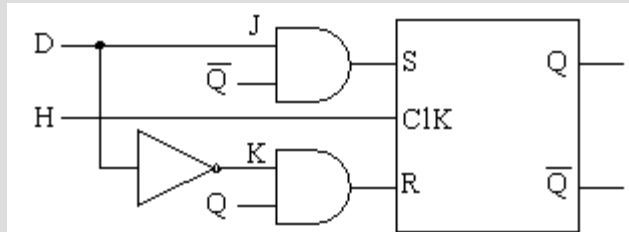


# Bascule JK et D avec RS



J	K	$Q^{n+1}$
0	0	$Q^n$
0	1	0
1	0	1
1	1	$\overline{Q^n}$

Basculer JK

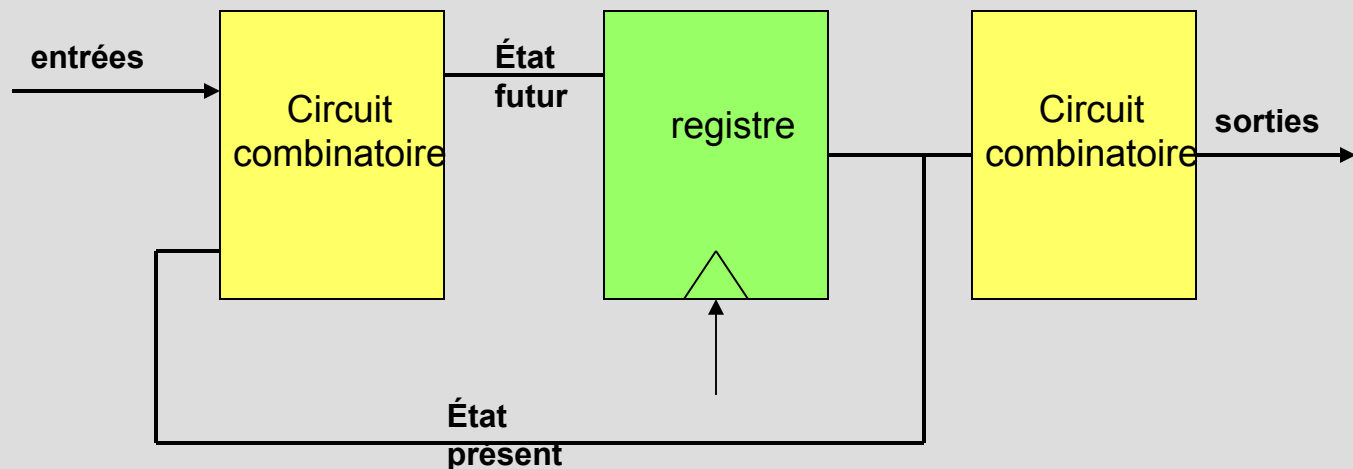


Basculer D

# Description de machines d'état

**Description de circuits séquentiels, où on raisonne en**  
***état présent*                      *état futur***

**avec des conditions de transitions→**



# liens

- [http://fr.wikipedia.org/wiki/Bascule\\_\(circuit\\_logique\)](http://fr.wikipedia.org/wiki/Bascule_(circuit_logique))