

---

# **Circuits numériques**

## **Logique séquentielle**

# Plan

---

- Spécificités de la logique séquentielle et des signaux synchrones / asynchrones
- Éléments de mémorisation :
  - circuit RS
  - point mémoire 6T et SRAM
  - D-latch, bascule D et registre
- Premier système séquentiel : le compteur
- Synthèse de circuits séquentiels
  - Machine de Moore
  - Machine de Mealy

# Logique combinatoire - rappel

---

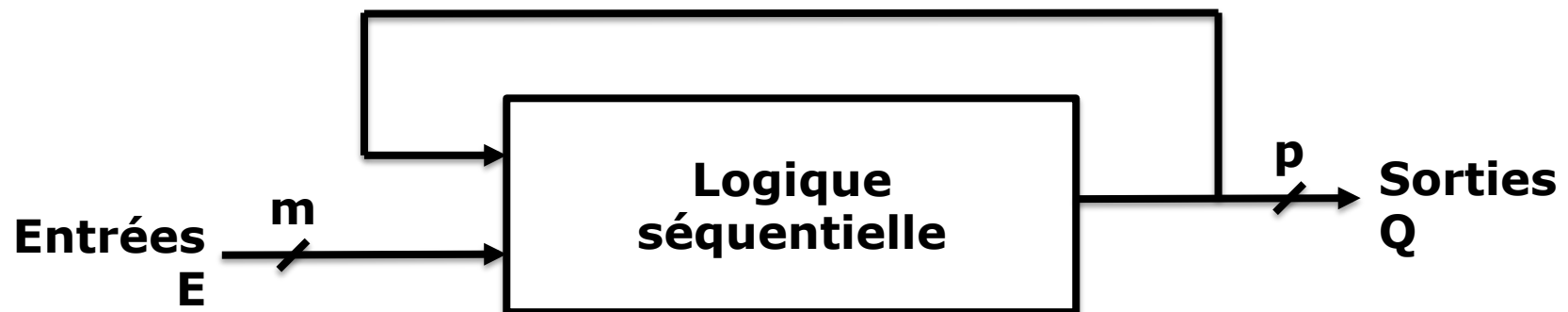
$$\text{Sorties} = f(\text{Entrées})$$

- $f$  représente un vecteur de fonctions combinatoires
- Ces fonctions sont déterministes et peuvent s'exprimer par des tableaux de vérité exhaustifs sur toutes les combinaisons des entrées
- Le calcul des sorties est effectué dès que la valeur de l'une des entrées se modifie
- Un état stable au niveau des entrées résulte en un état stable au niveau des sorties

# Principes de la logique séquentielle

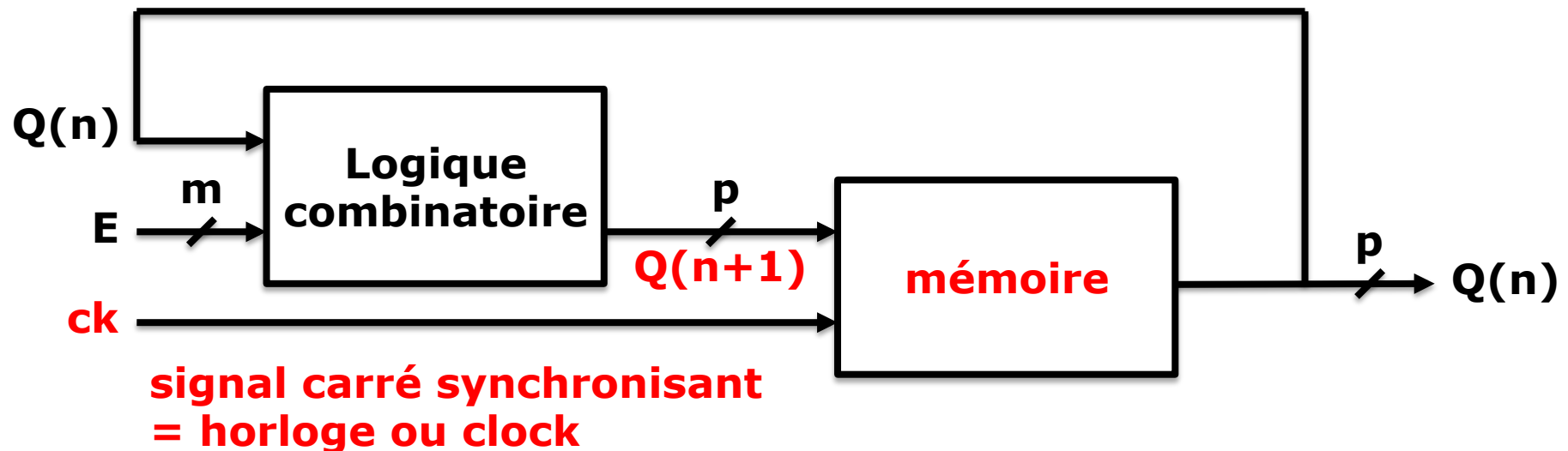
**Sorties = f(Entrées, Sorties précédentes)**

- Une fonction de logique séquentielle a la capacité de se « souvenir » des états antérieurs des sorties
- Une même combinaison des entrées pourra avoir des effets *différents* sur les sorties
- On introduit donc une notion de **mémorisation** de l'état actuel des sorties



# Logique séquentielle synchrone

- On utilisera le plus souvent des fonctions de logique séquentielle **synchrone** : tous les éléments de mémorisation reçoivent le même signal d'horloge



$$Q(n+1) = f(E, Q(n))$$

$n$  s'incrmente à chaque front actif de l'horloge ; autrement dit,  $n$  correspond au nombre entier de périodes de l'horloge depuis  $t=0$

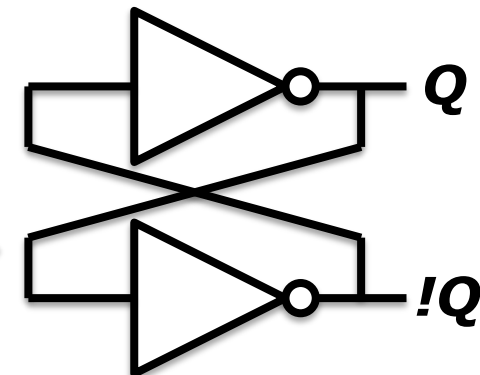
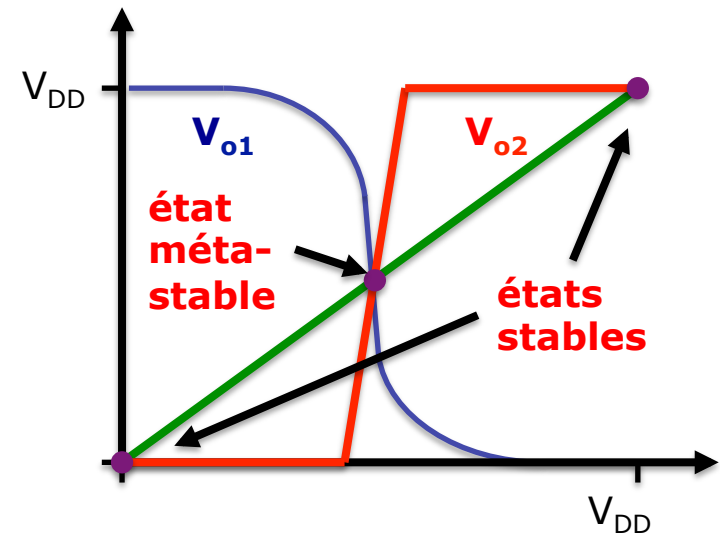
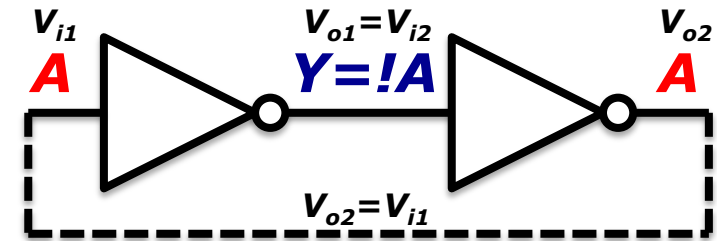
# Signaux synchrones / asynchrones

---

- Les signaux synchrones changent d'état à un instant corrélé au front actif de l'horloge
  - Les sorties Q sont nécessairement synchrones
  - Les entrées E sont synchrones si issues de logique séquentielle synchrone située en amont
- Les signaux asynchrones changent d'état à un instant qui n'est pas corrélé au front actif de l'horloge
  - Il s'agit en général de signaux de contrôle
  - Remise à zéro (RESET, ou RAZ)
  - Mise à 1 (SET)
  - Chargement d'une valeur (LOAD)

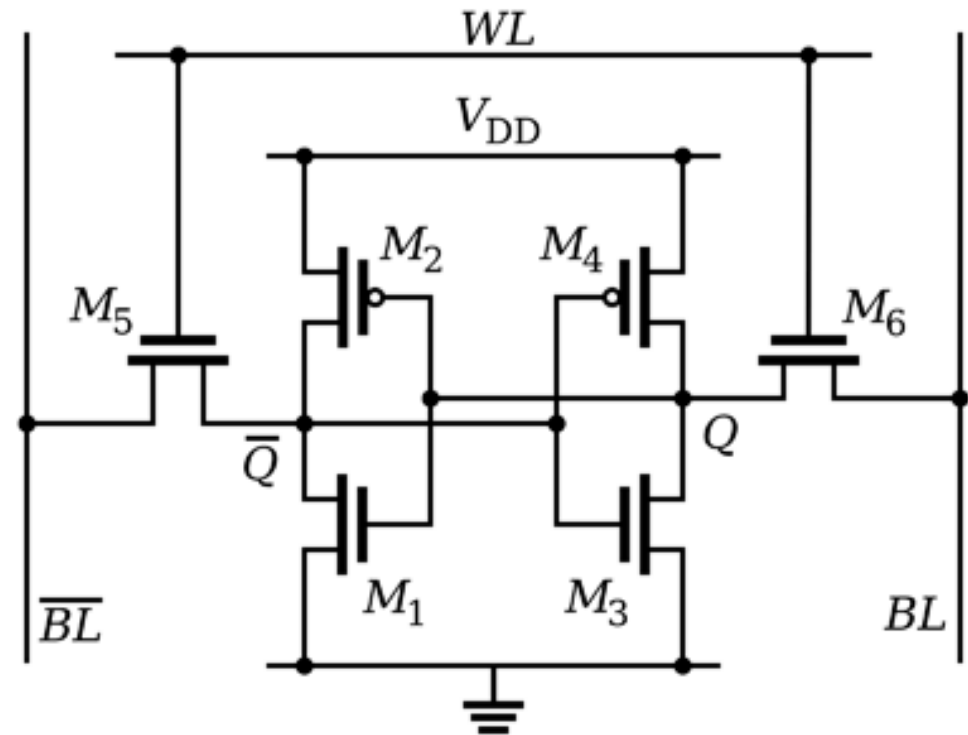
# Phénomène bistable

- Bistable = circuit avec 2 états stables (il y a aussi un état métastable)
- Les 2 états correspondent à la mémorisation des 2 états possibles d'un bit
- Mais il n'y a pas d'entrée : impossible de contrôler l'état
- Représentation usuelle



# Parenthèse : Point mémoire 6T (SRAM)

- $M_1/M_2$  et  $M_3/M_4$  constituent les deux inverseurs connectés en boucle
- BL (bit line) transmet (en signaux complémentaires) la valeur à mémoriser ou à lire

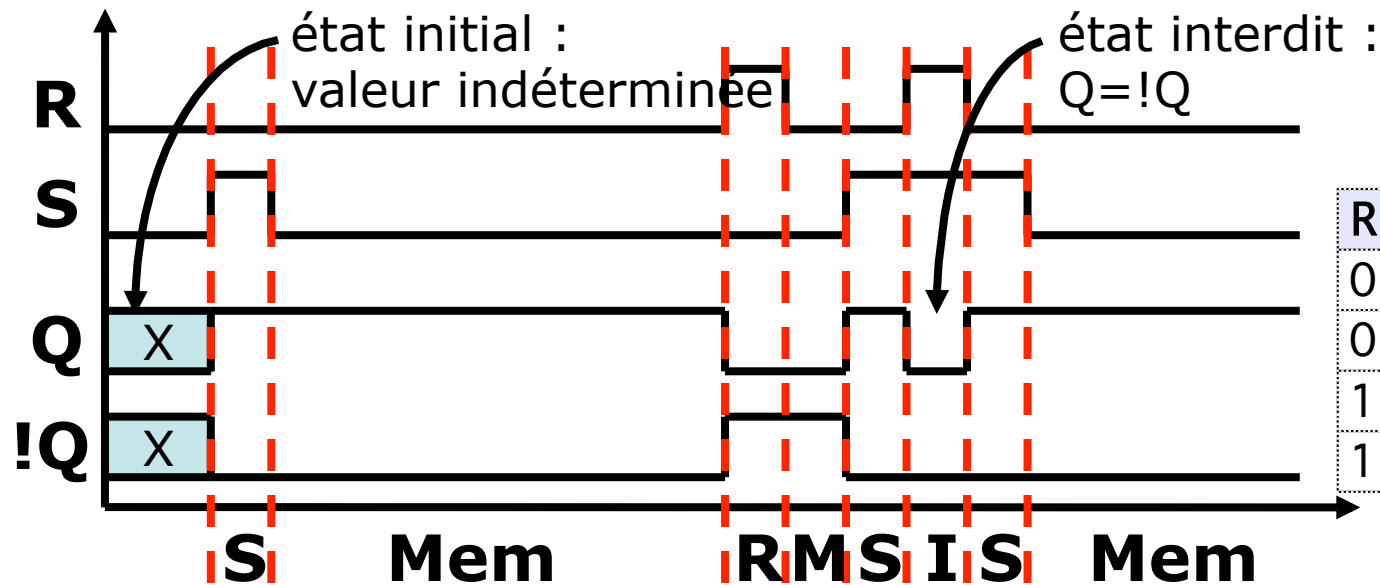
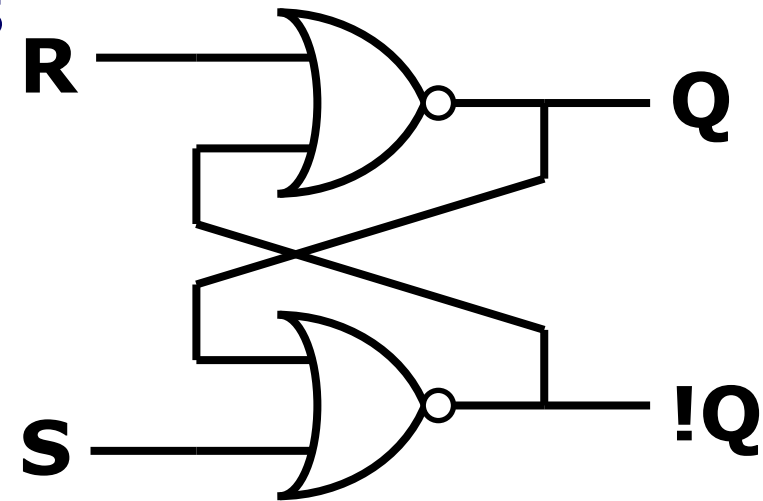


- WL (word line) commande les transistors  $M_5/M_6$  pour connecter BL à Q (ainsi que  $\overline{BL}$  à  $\overline{Q}$ )



# Bistable → bascule RS

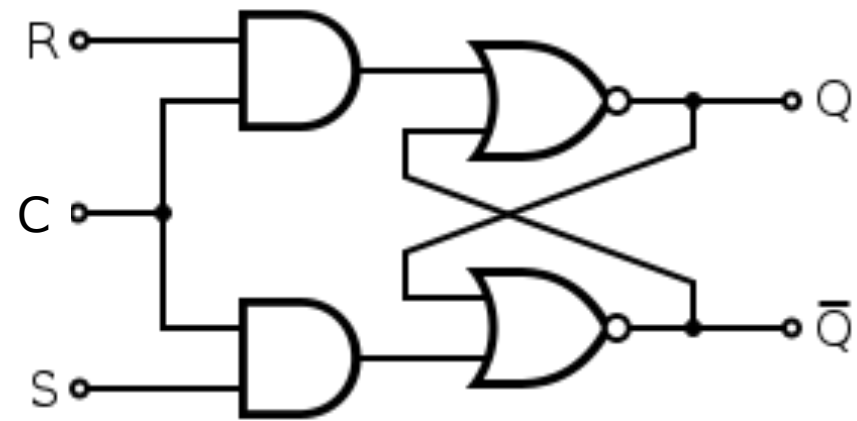
- Remplacer les inverseurs par des portes NOR
- Entrées R (RESET) et S (SET)
- Existe aussi en version NAND (R et S inversées)



R	S	$Q(n+1)$	$!Q(n+1)$
0	0	$Q(n)$	$!Q(n)$
0	1	1	0
1	0	0	1
1	1	0	0

# Bascule RS + horloge

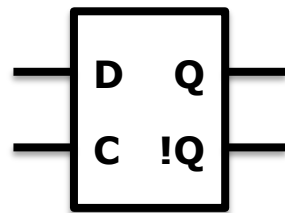
- Rajouter un étage d'entrée pour activer/désactiver la bascule
- $C=0$  :  $Q(n+1)=Q(n)$
- $C=1$  :  $Q(n+1)=S.\bar{R}$
- État interdit ( $Q=\bar{Q}$  lorsque  $S=R=1$ ) existe toujours



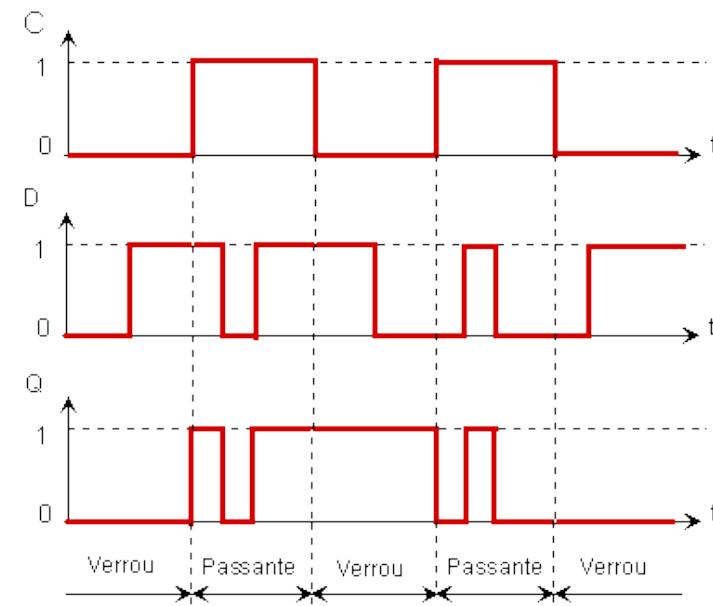
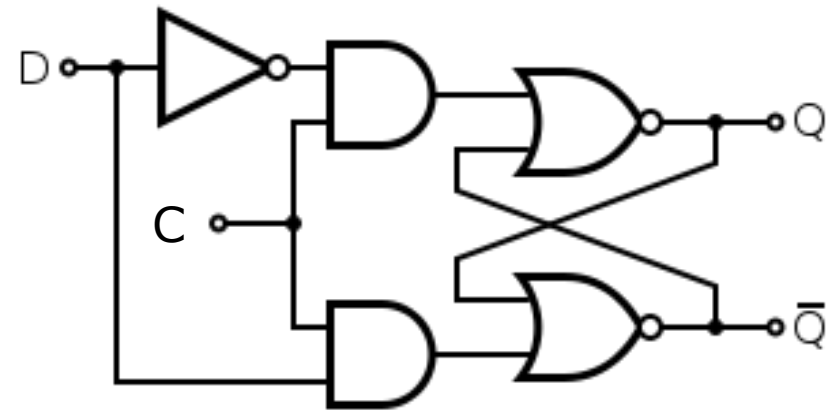
C	S	R	$Q(n+1)$	$\bar{Q}(n+1)$
0	X	X	$Q(n)$	$\bar{Q}(n)$
1	0	0	$Q(n)$	$\bar{Q}(n)$
1	0	1	0	1
1	1	0	1	0
1	1	1	Interdit	Interdit

# Le D-latch

- Rajouter un étage d'entrée pour contrôler R et S à partir d'un seul bit de donnée (D comme data)
- $C=0$  :  $Q(n+1)=Q(n)$
- $C=1$  :  $Q(n+1)=D$
- Plus d'état interdit

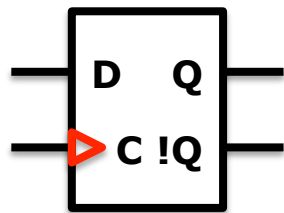
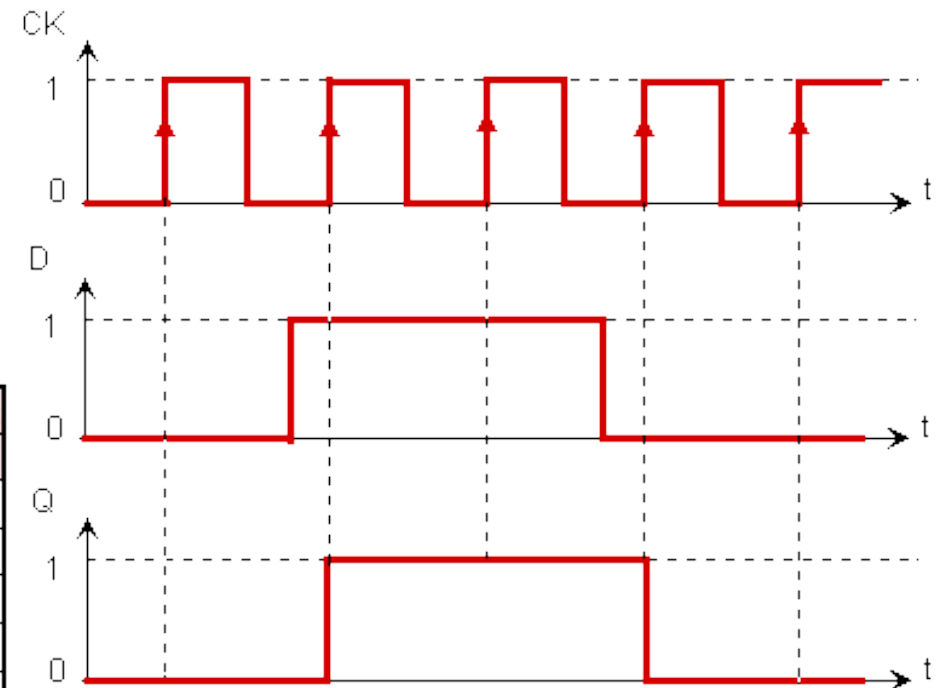
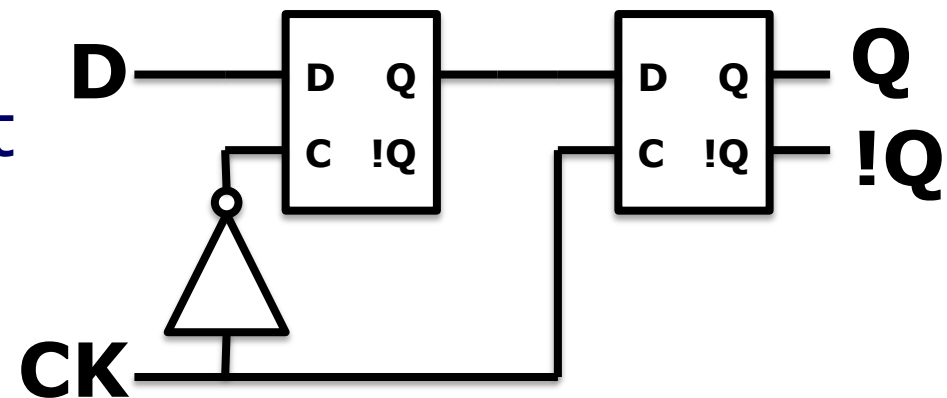


Entrées		Sorties	
C	D	$Q_{n+1}$	$\overline{Q}_{n+1}$
0	X	$Q_n$	$\overline{Q}_n$
1	0	0	1
1	1	1	0



# La bascule D

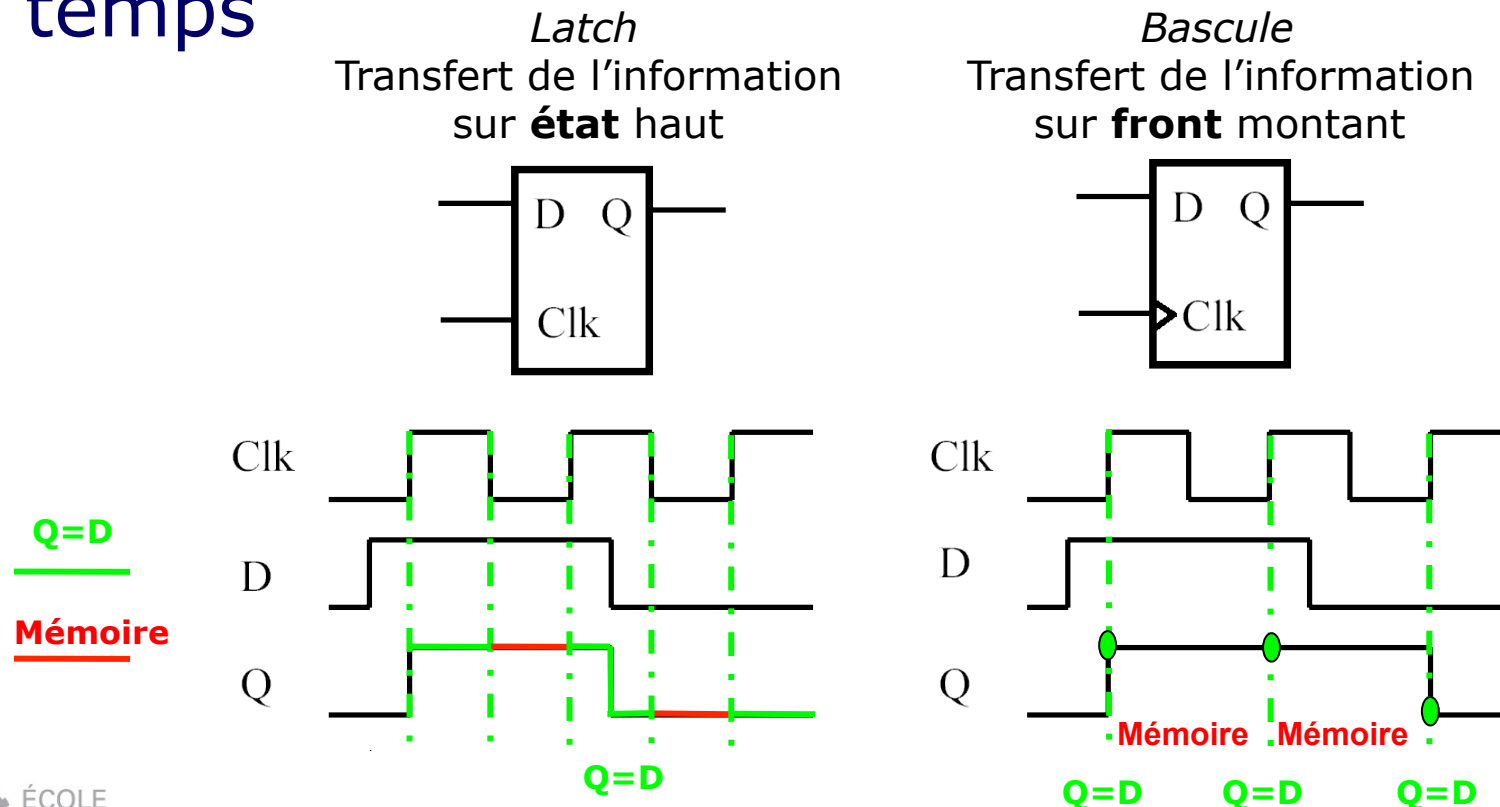
- 2 D-latch connectés en maître-esclave et fonctionnant en opposition de phase de l'horloge
- La synchronisation se fait sur le **front actif** (montant ici) de CK



Entrées		Sorties	
CK	D	$Q_{n+1}$	$\overline{Q}_{n+1}$
0	X	$Q_n$	$\overline{Q}_n$
1	X	$Q_n$	$\overline{Q}_n$
↓	X	$Q_n$	$\overline{Q}_n$
↑	0	0	1
↑	1	1	0

# D-latch vs. bascule D

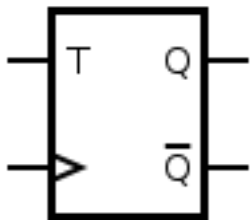
- La synchronisation sur front actif (montant ou descendant), cas de la bascule, permet de déclencher simultanément diverses fonctions : corrélation des données dans le temps



# Parenthèse : D'autres bascules

- Bascule T

- Inverse la sortie Q si  $T=1$
- Fréquence de Q =  $\frac{1}{2}$  la fréquence de l'horloge
- Utilisation dans les compteurs

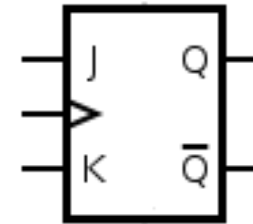


T	Q(n+1)
0	Q(n)
1	!Q(n)

$$Q(n+1) = T \oplus Q(n) = T\overline{Q(n)} + \overline{T}Q(n)$$

- Bascule JK

- $J=\overline{K}$  : bascule D
- $J=K$  ; bascule T
- $J=S$  et  $K=R$

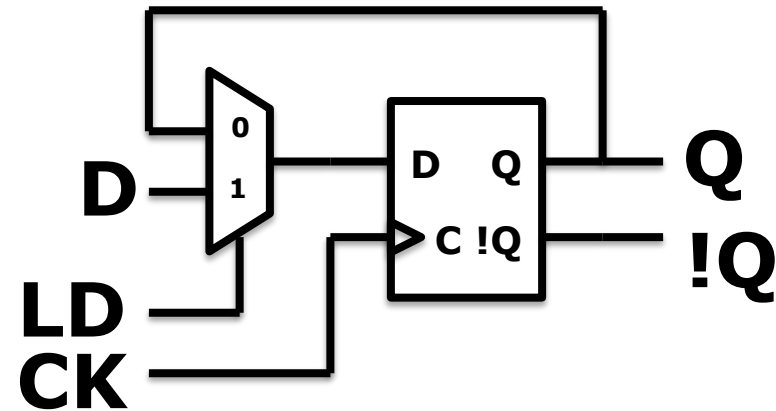


$$Q(n+1) = J\overline{Q(n)} + \overline{K}Q(n)$$

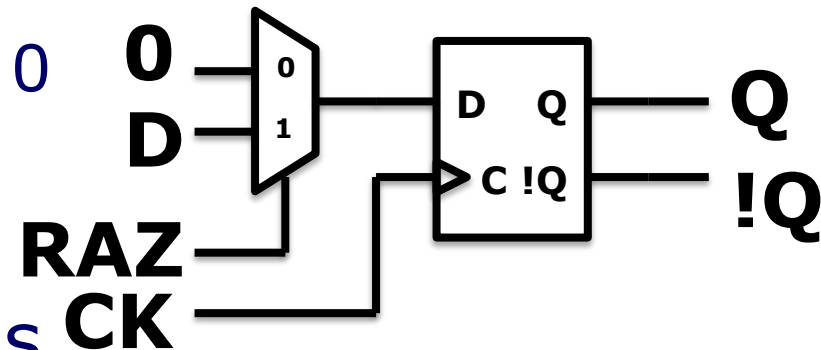
Entrées			Sorties	
CK	J	K	Q <sub>n+1</sub>	$\overline{Q}_{n+1}$
0	X	X	Q <sub>n</sub>	$\overline{Q}_n$
1	X	X	Q <sub>n</sub>	$\overline{Q}_n$
↓	X	X	Q <sub>n</sub>	$\overline{Q}_n$
↑	0	0	Q <sub>n</sub>	$\overline{Q}_n$
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\overline{Q}_n$	Q <sub>n</sub>

# Signaux de contrôle

- Exemples synchrones
- Chargement (LD)
  - LD=1 : chargement de la donnée D
  - LD=0 : maintien de la donnée en mémoire (isolation de l'entrée)



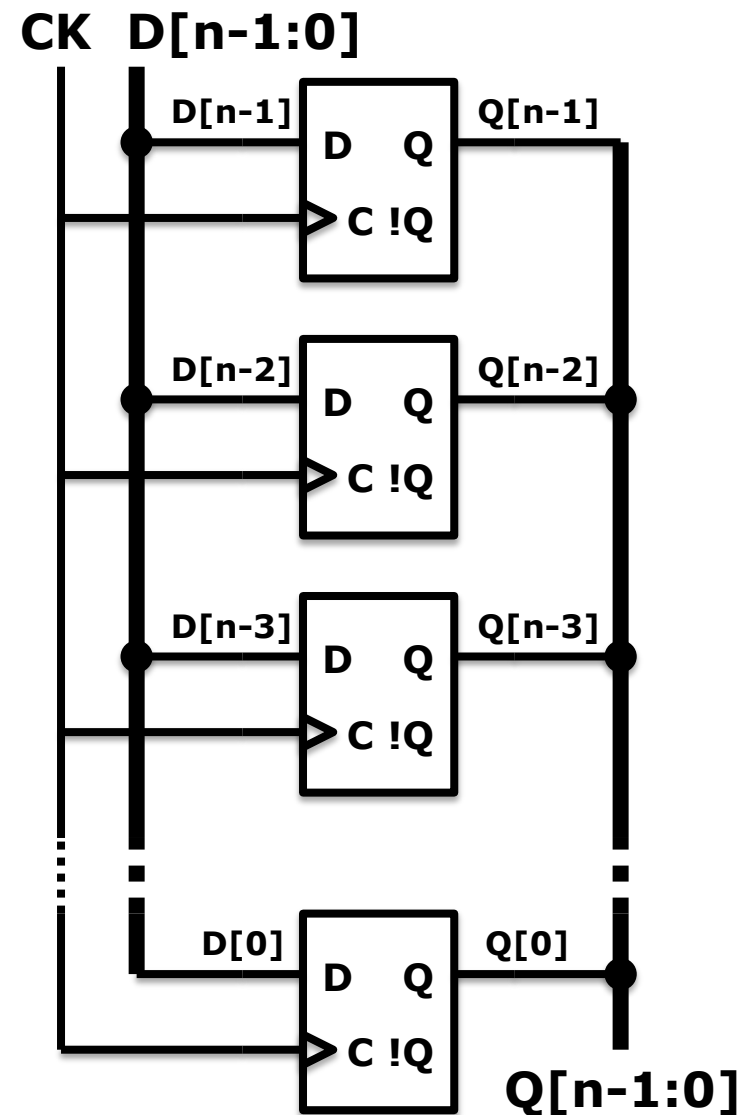
- Remise à zéro (RAZ)
  - RAZ=0 : chargement de 0
  - RAZ=1 : chargement de la donnée D



- Possible de combiner les signaux de contrôle

# Registres

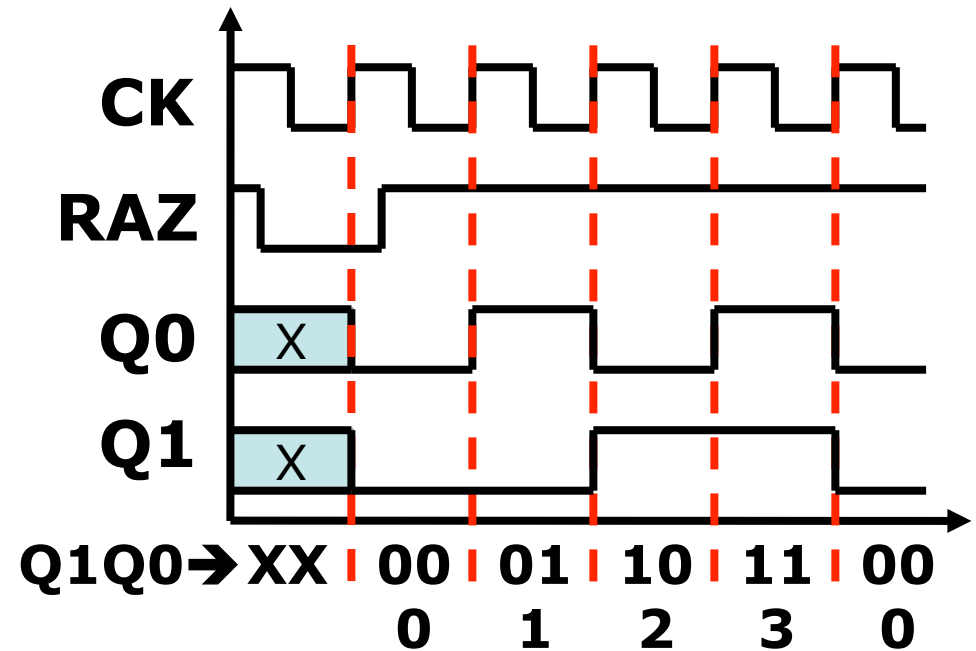
- Mise en parallèle de  $n$  bascules D pour mémoriser une donnée de  $n$  bits
- Le signal d'horloge est commun à l'ensemble des bascules
- Tout signal de contrôle (LD, RAZ ...) est aussi commun à l'ensemble des bascules





# Premier système séquentiel : le compteur

- Un compteur est un ensemble de bascules reliées de manière à compter des impulsions de l'horloge
- Il génère ainsi une séquence ordonnée, sans données en entrée
- Un compteur modulo N compte de 0 à N-1



- Compteur modulo 4
- Division de fréquence :
  - $f_{Q0} = f_{CK}/2$
  - $f_{Q1} = f_{Q0}/2 = f_{CK}/4$
  - ...

# Compteur modulo 4 : états

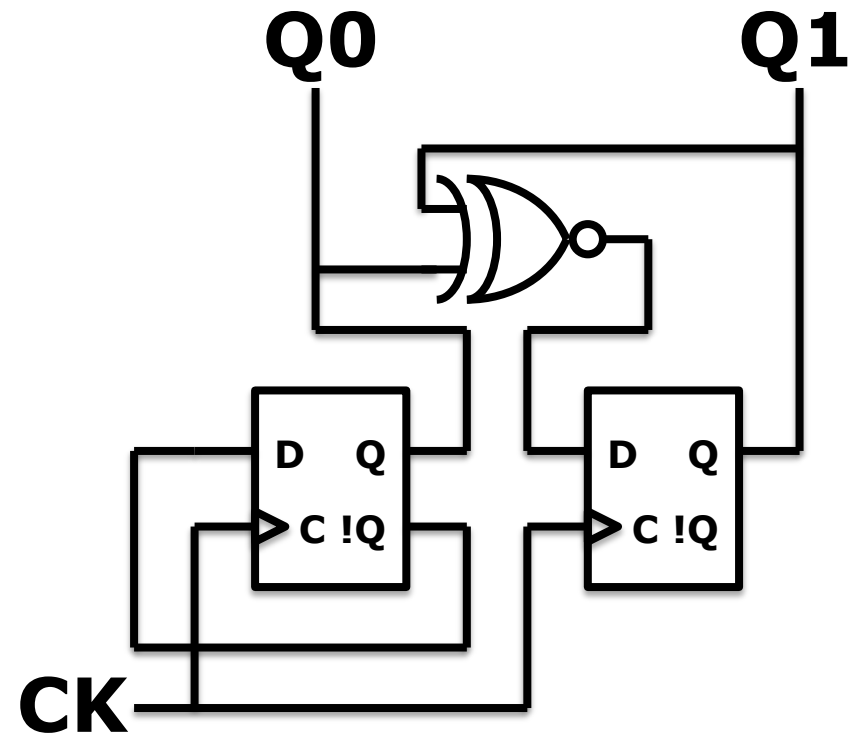
- 4 états nécessitent  $\log_2(4)=2$  bascules D
- Le réseau combinatoire pour générer les entrées des bascules en fonction de la séquence désirée s'établit avec la table d'états

Etat (n)	Q1 (n)	Q0 (n)	Etat (n+1)	Q1 (n+1) = D1(n)	Q0 (n+1) = D0(n)
0	0	0	1	0	1
1	0	1	2	1	0
2	1	0	3	1	1
3	1	1	0	0	0

$$D_0 = \overline{Q_0}$$

$$D_1 = Q_1 \oplus Q_0$$

# Compteur modulo 4 : structure



# Compteur modulo 10 : tableau d'états

Etat (n)	Q3 (n)	Q2 (n)	Q1 (n)	Q0 (n)	Etat (n+1)	Q3 (n+1) = D3(n)	Q2 (n+1) = D2(n)	Q1 (n+1) = D1(n)	Q0 (n+1) = D0(n)
0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	2	0	0	1	0
2	0	0	1	0	3	0	0	1	1
3	0	0	1	1	4	0	1	0	0
4	0	1	0	0	5	0	1	0	1
5	0	1	0	1	6	0	1	1	0
6	0	1	1	0	7	0	1	1	1
7	0	1	1	1	8	1	0	0	0
8	1	0	0	0	9	1	0	0	1
9	1	0	0	1	0	0	0	0	0

# Compteur modulo 10 : équations

**D0**

		Q1Q0			
		00	01	11	10
Q3Q2	00	1	0	0	1
	01	1	0	0	1
	11	X	X	X	X
	10	1	0	X	X

$$D_0 = \overline{Q_0}$$

**D1**

		Q1Q0			
		00	01	11	10
Q3Q2	00	0	1	0	1
	01	0	1	0	1
	11	X	X	X	X
	10	0	0	X	X

$$D_1 = \overline{Q_3} (Q_1 \oplus Q_0)$$

**D2**

		Q1Q0			
		00	01	11	10
Q3Q2	00	0	0	1	0
	01	1	1	0	1
	11	X	X	X	X
	10	0	0	X	X

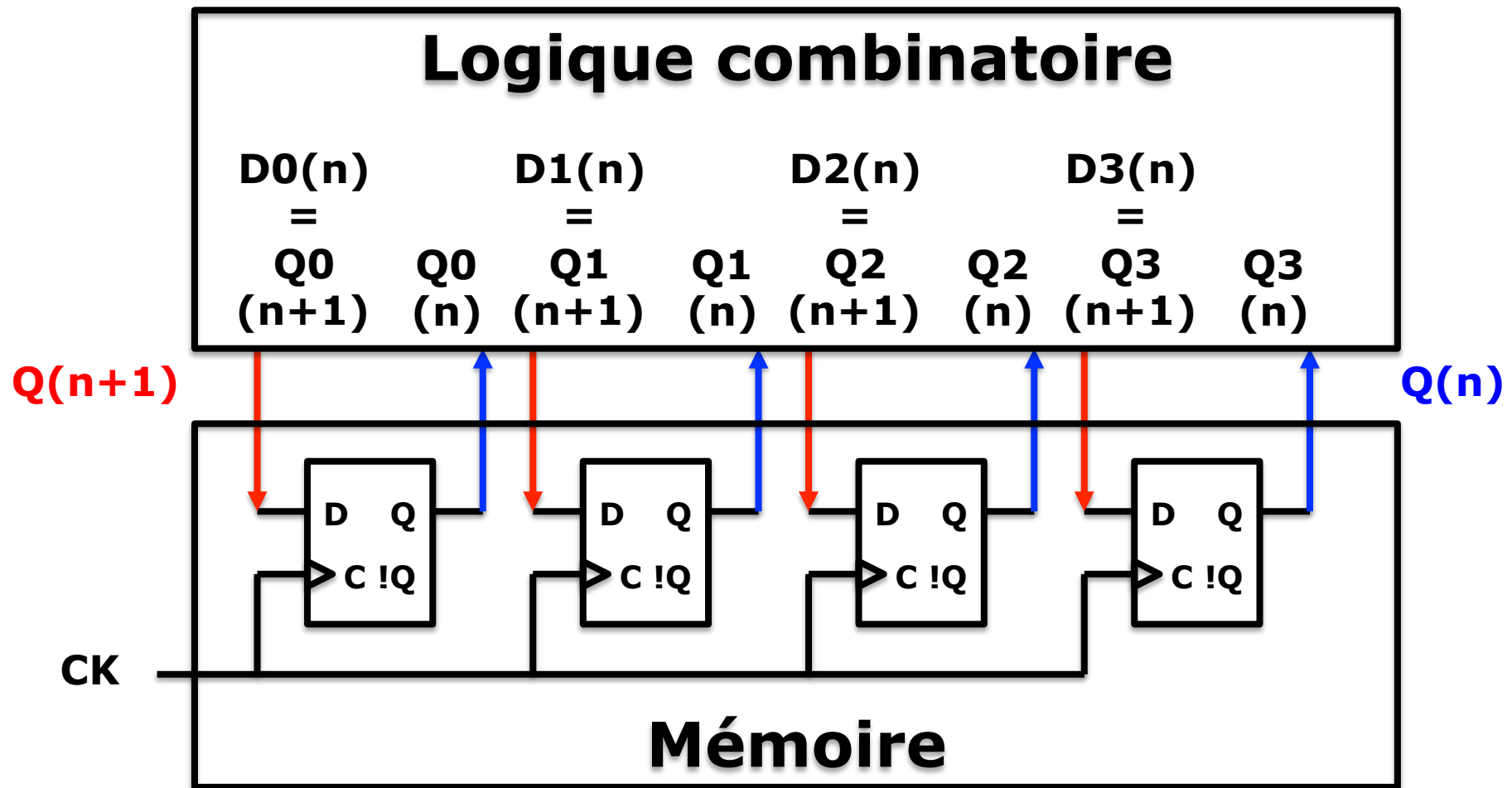
$$D_2 = Q_2 (\overline{Q_1} + \overline{Q_0}) + \overline{Q_2} Q_1 Q_0$$

**D3**

		Q1Q0			
		00	01	11	10
Q3Q2	00	0	0	0	0
	01	0	0	1	0
	11	X	X	X	X
	10	1	0	X	X

$$D_3 = \overline{Q_3} \overline{Q_1} \overline{Q_0} + Q_2 Q_1 Q_0$$

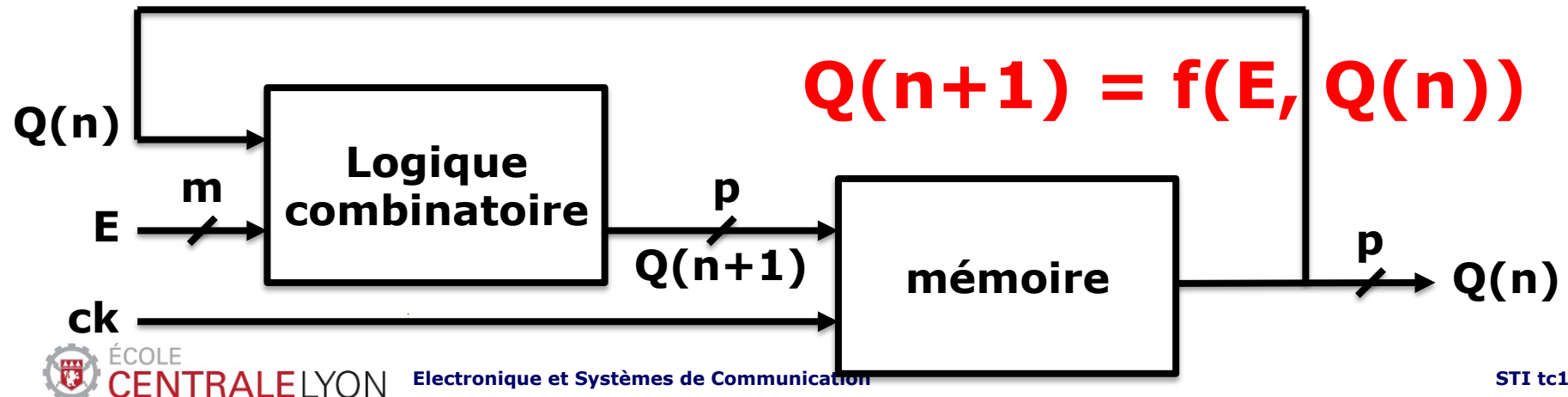
# Compteur modulo 10 : structure



**$Q(n+1) = f(Q(n))$  : pas d'entrées ici**

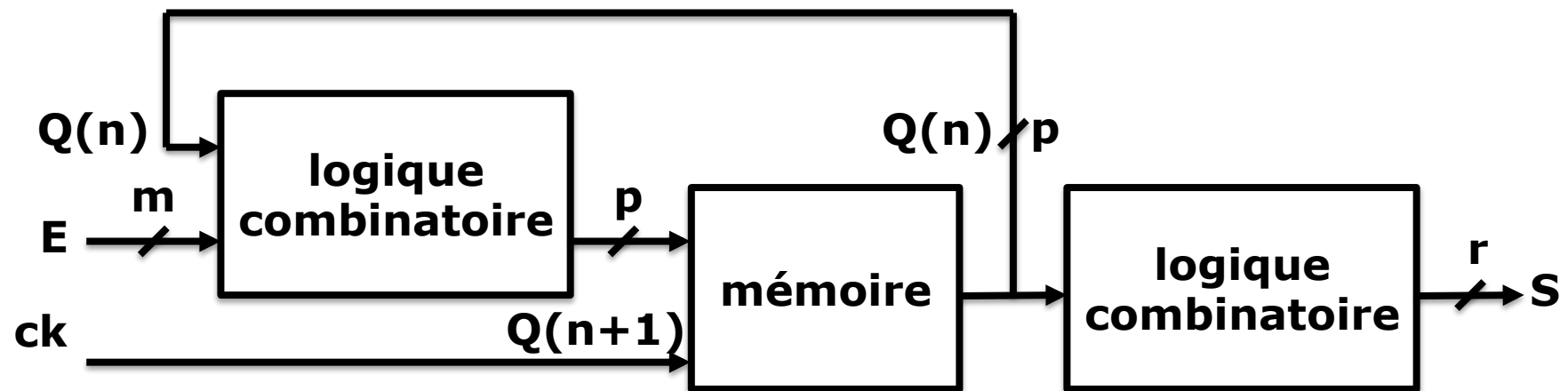
# Système séquentiel complet

- Les entrées E représentent les stimuli pour faire passer le système d'un état vers un autre : il s'agit plus généralement d'une machine d'états finis
  - Exemples simples : détecteurs de séquence / code, feux de circulation, ascenseurs, vendeurs automatiques
  - Exemples plus complexes : protocoles de communication, parseurs linguistiques ...



# Machine de Moore : structure générique

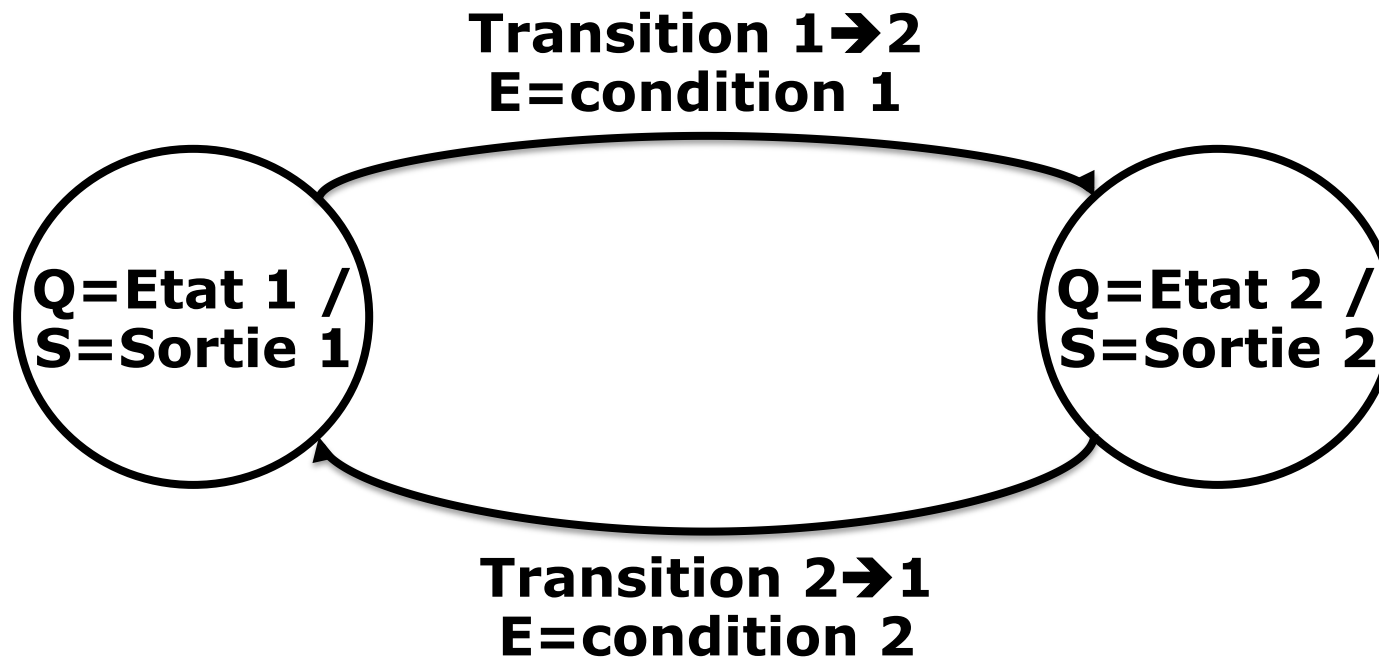
- Les sorties  $S$  dépendent de l'état présent  $Q(n)$  : les sorties changent de manière **synchrone** sur un front d'horloge et il n'y a pas de connexion directe entrée-sortie
- L'état futur  $Q(n+1)$  dépend des entrées  $E$  et de l'état présent  $Q(n)$



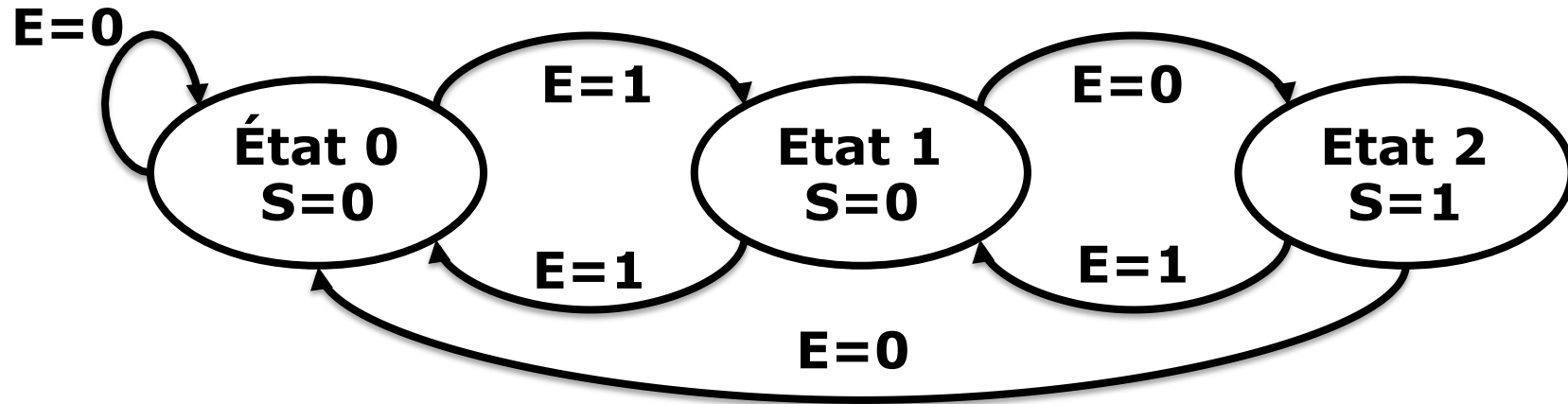


# Machine de Moore : diagramme d'états

- Les sorties dépendent seulement des états
- Les transitions entre les états (condition sur les entrées) sont prises en compte de manière synchrone



# Exemple : détection de séquence « 10 »



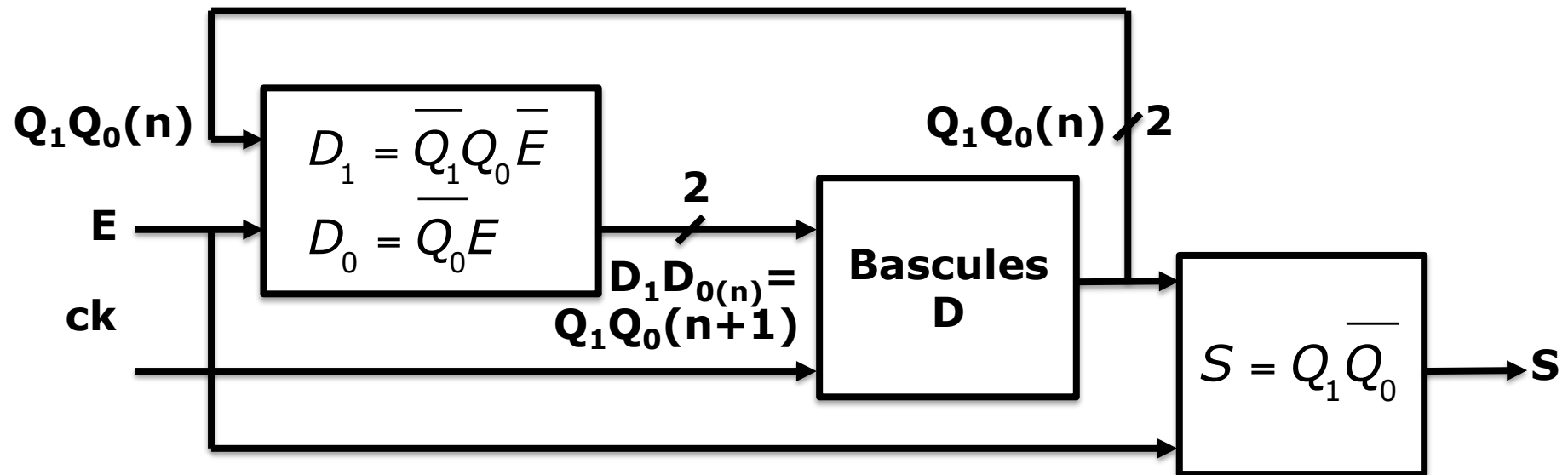
Etat (n)	Q1 (n)	Q0 (n)	E	Etat (n+1)	Q1(n+1) = D1(n)	Q0(n+1) = D0(n)	S
0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0
1	0	1	0	2	1	0	0
1	0	1	1	0	0	0	0
2	1	0	0	0	0	0	1
2	1	0	1	1	0	1	1

$$D_1 = \overline{Q_1} \overline{Q_0} \overline{E}$$

$$D_0 = \overline{Q_0} E$$

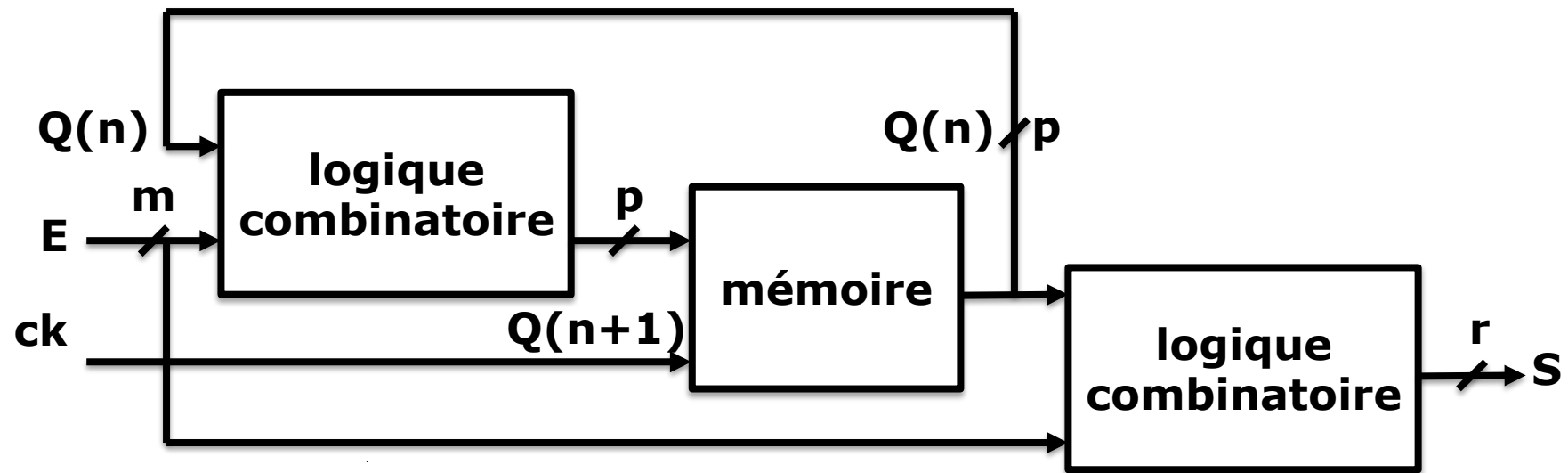
$$S = Q_1 Q_0$$

# Détection de séquence « 10 » : structure



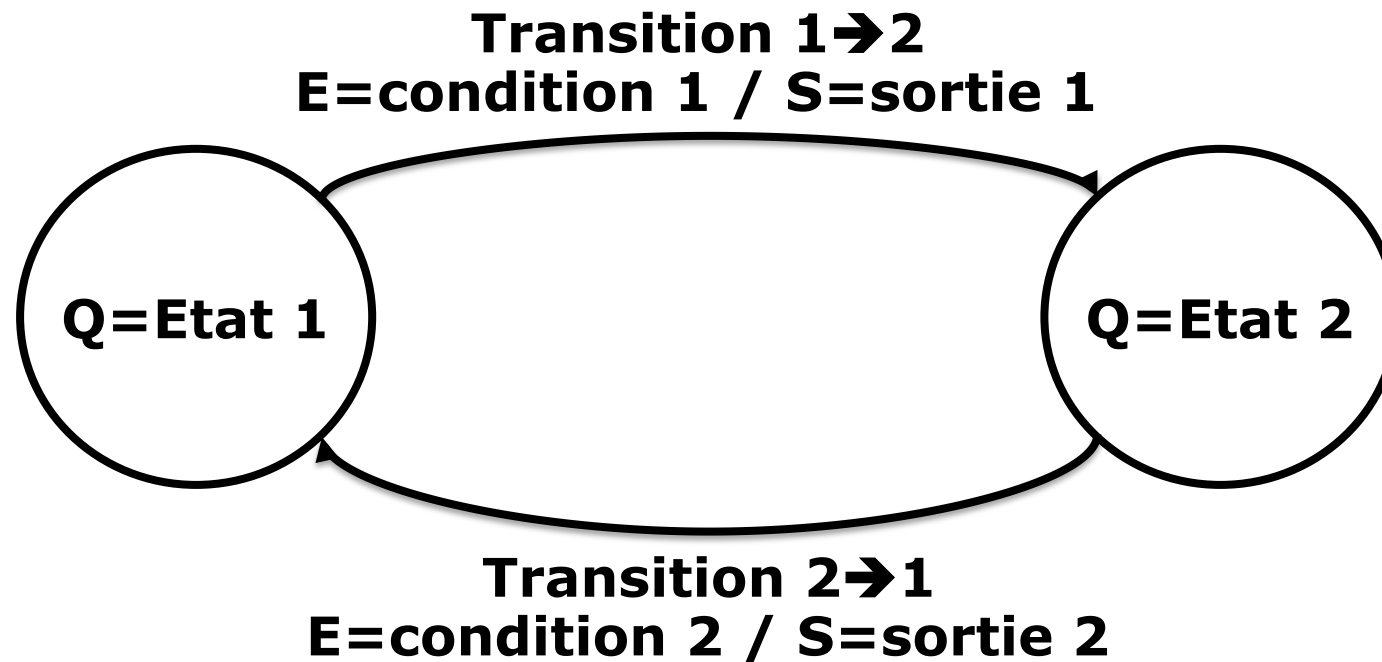
# Machine de Mealy : structure générique

- Les sorties  $S$  dépendent de l'état présent  $Q(n)$  et des entrées  $E$  : les sorties changent de manière **asynchrone** en raison de la connexion directe entrée-sortie
- L'état futur  $Q(n+1)$  dépend des entrées  $E$  et de l'état présent  $Q(n)$

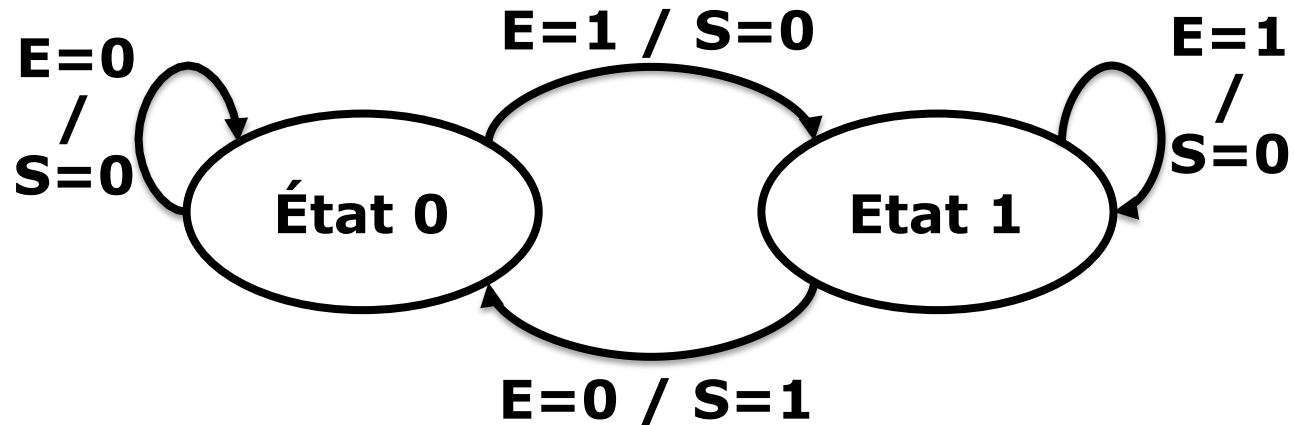


# Machine de Mealy: diagramme d'états

- Les sorties dépendent des états et des entrées : elles sont donc associées aux **transitions** (combinaison d'un état et d'une condition sur les entrées) et changent de manière asynchrone



# Exemple : détection de séquence « 10 »



Etat (n)	Q0 (n)	E	Etat (n+1)	Q0(n+1) = D0(n)	S
0	0	0	0	0	0
0	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	0

$$D_0 = E$$

$$S = Q_0 \overline{E}$$

# Moore vs Mealy

---

- Machines de Moore et de Mealy peuvent être fonctionnellement équivalentes
- Machine de Mealy permet une description plus complexe avec un nombre d'état réduit
  - Surface et consommation réduite
- Machine de Mealy : prise en compte immédiate d'un changement en entrée, temps de réponse réduit d'un cycle par rapport à son équivalent de Moore
- Une machine de Moore n'a pas de chemin combinatoire liant l'entrée et la sortie
  - Absence d'un chemin critique le plus court

# Aide-mémoire

---

**Sorties = f(Entrées, Sorties précédentes)**

- Bistable au cœur de la mémorisation
- Bascule D = 2 D-latches en maître-esclave, mémorise sur **front** d'horloge
- Registre = plusieurs bascules D en parallèle
- Compteur modulo N =  $\log_2(N)$  bascules avec logique combinatoire pour coder une séquence
- Machines d'états finis (Moore ou Mealy) = bascules D avec logique combinatoire pour coder les sorties et l'état futur en fonction des entrées et de l'état présent