

Logique séquentielle

I. Introduction

□ Contrainte temporelle des circuits

Un circuit est formé de plusieurs portes:

- ✓ Chemin critique : chemin le « plus long » pour la propagation des signaux à travers le circuit
- ✓ Détermine le temps total de propagation des signaux à travers tout le circuit
- ✓ Temps minimal à attendre pour avoir une sortie valide
- ✓ Intuitivement : chemin passant par le plus grand nombre de portes
- ✓ Mais dépend aussi du temps de propagation de chaque type de porte

Logique séquentielle

I. Introduction

□ Horloge

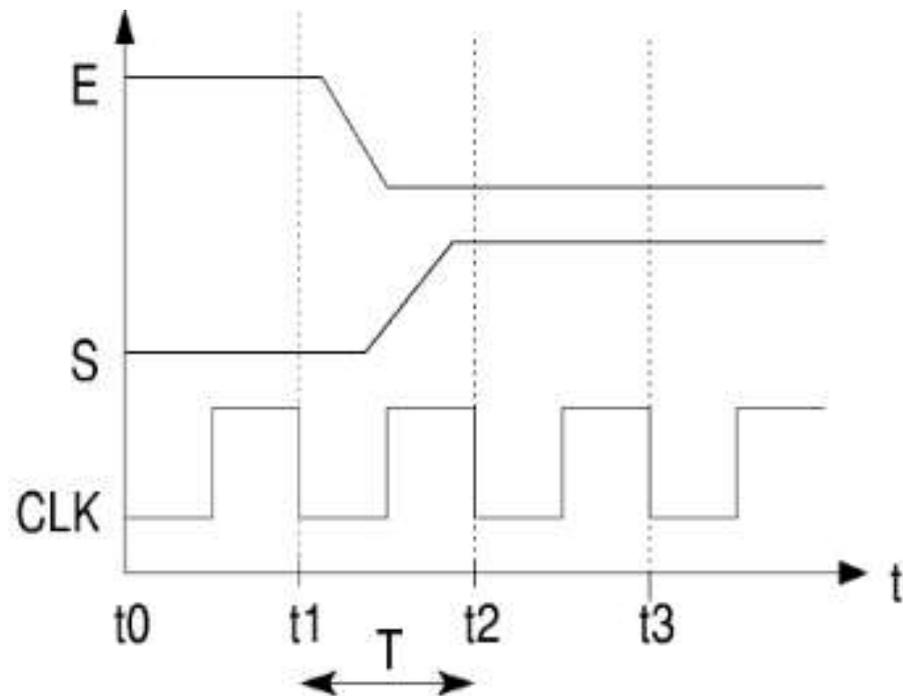
- À cause de tous les délais (montée, descente, propagation) un signal n'est pas dans un état valide en permanence
- Idée : on ne lit ses valeurs qu'à des instants précis et à des intervalles réguliers
 - ✓ Instants donnés par une horloge
- Horloge
 - ✓ Système logique qui émet régulièrement une suite d'impulsions calibrées
 - ✓ L'intervalle de temps entre 2 impulsions représente le temps de cycle ou la période de l'horloge

Logique séquentielle

I. Introduction

□ Horloge

- Signal périodique
 - un demi période à 0, l'autre à 1
- Début d'une nouvelle période : instant t_i
- Exemple
 - Instant $t1$: $E = 1, S = 0$
 - Instant $t2$: $E = 0, S = 1$
 - CLK = Clock = signal d'horloge



Logique séquentielle

I. Introduction

□ Circuit logique a Mémoire

Les fonctions de sortie dépendent non seulement de l'état des variables d'entrée mais également de l'état antérieur de certaines variables de sortie (propriétés de mémorisation)



Logique séquentielle

I. Introduction

□ Circuits séquentiels

- Circuits combinatoires

- ✓ Les sorties ne dépendent que des valeurs des entrées

- Circuits séquentiels

- ✓ Ajout des notions d'état et de mémoire
- ✓ Ajout de la notion de temps (horloge)

Logique séquentielle

I. Introduction

□ Circuits séquentiels

- Les valeurs de sorties du circuit dépendent
 - ✓ Des valeurs en entrée
 - ✓ De valeurs calculées précédemment
 - ✓ De l'état dans lequel on se trouve

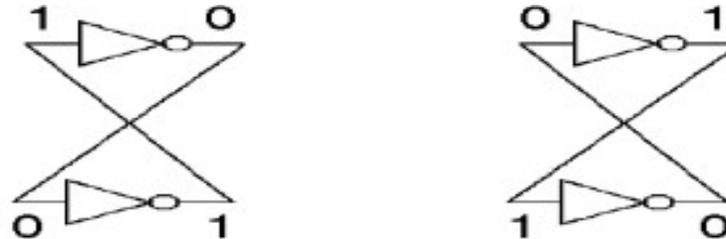
- Théories utilisées pour étudier/spécifier les différents types de circuits
 - ✓ Circuits combinatoires : algèbre de Boole
 - ✓ Circuits séquentiels : théorie des automates finis

Logique séquentielle

II. Les Bascules

□ **Bistable** : 2 états stables dans le temps

- ✓ Principe général d'une bistable : 2 portes NON (inverseurs) en opposition



□ **Bascule** : (flip-flop) : système permettant de mémoriser une information élémentaire , c'est composant qui met en œuvre une bistable

- ✓ Possibilité de passer d'un état à l'autre, de changer l'état mémorisé
- ✓ mémoire à 1 bit ayant 2 états : Q et /Q
- ✓ utilise un mécanisme de verrou (latch)
- ✓ Plusieurs façons de gérer et changer l'état
- ✓ Plusieurs types de bascules : RS, D, JK ...

II. Les Bascules

3 types de bascules

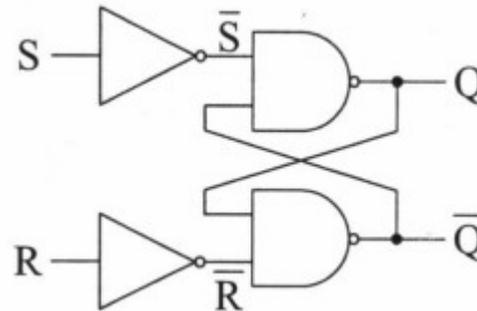
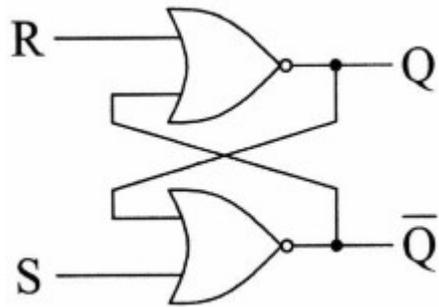
- ❑ **Asynchrone**: Quand les entrées changent et la sortie est recalculée
- ❑ **Synchrone sur niveau**: Quand le niveau (1 en général, mais 0 aussi) est atteint
- ❑ **Synchrone sur front**: Au moment du passage de 0 à 1 ou de 1 à 0 selon le type de front utilise par la bascule

Logique séquentielle

II. Les Bascules

□ Bascule RS

- réalisées avec des portes NOT ou NAND
- portes NOR

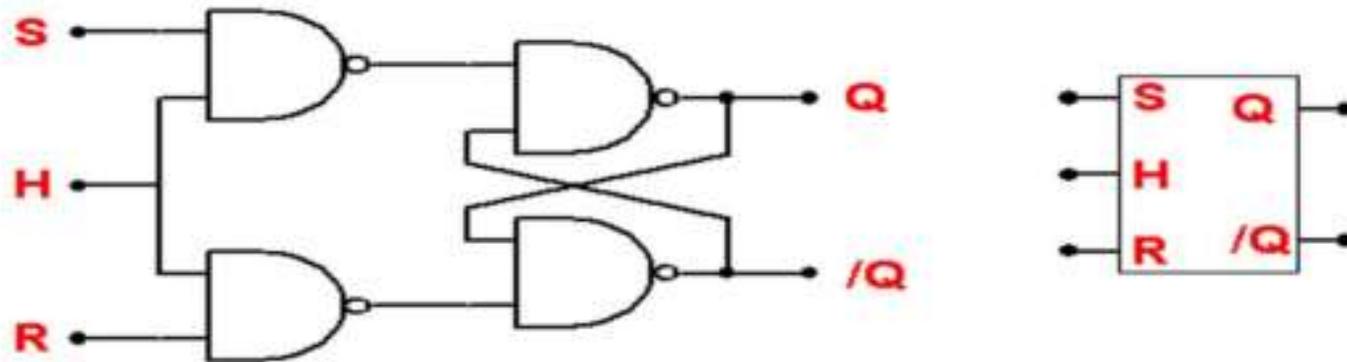


S	R	Q	/Q	
0	0	Q	/Q	Etat Mémoire
0	1	0	1	Remise a 0
1	0	1	0	Remise a 1
1	1			Etat Interdit

II. Les Bascules

□ Bascule RSH/RST

- bascule RS dans laquelle les entrées R et S ne sont prises en comptes que si elles sont en coïncidence avec un signal de commande
- bascule bloquée quand le signal de commande est à 0
- si le signal de commande est fourni par une horloge : bascule synchrone



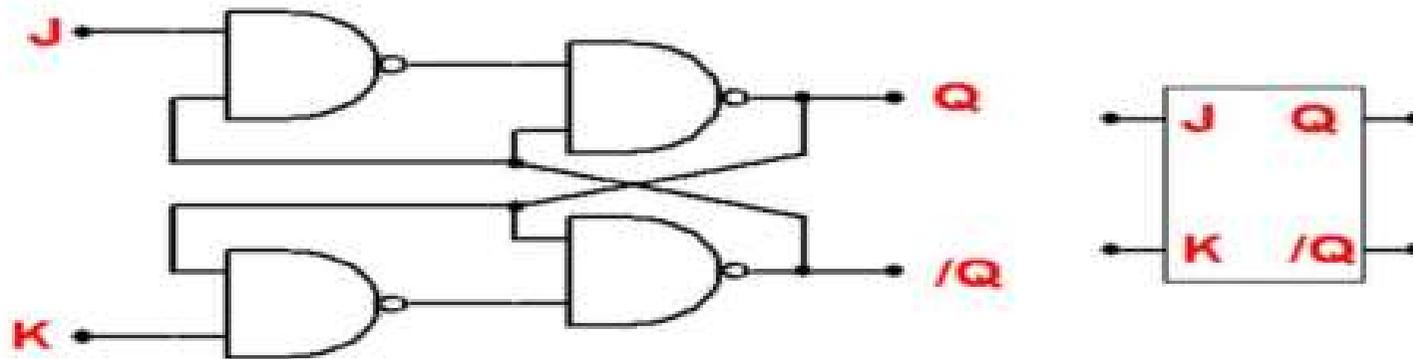
■ H=1 → Table de vérité

■ H=0 → mémoire

Logique séquentielle

II. Les Bascules

□ Bascule JK

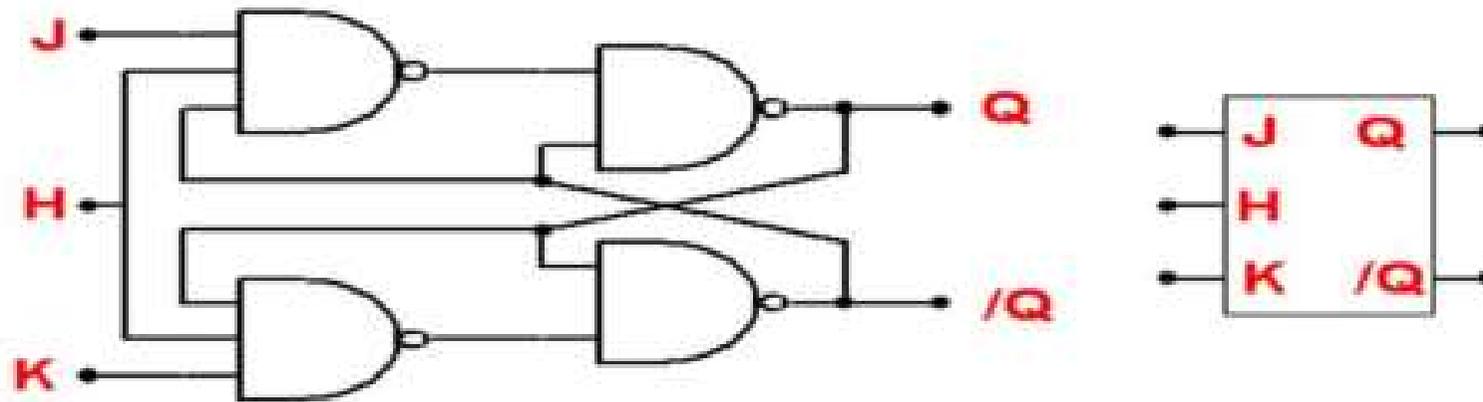


J	K	Q	/Q	
0	0	Q	/Q	Etat Mémoire
0	1	0	1	0
1	0	1	0	1
1	1			Basculement

Logique séquentielle

II. Les Bascules

□ Bascule JKH



- H=1 → Table de vérité

- H=0 → sorties inchangées

II. Les Bascules

□ **Bascules réagissant sur front d'horloge**

- ❖ Bascule en permanence dans l'état Mémoire sauf pendant les fronts de l'horloge

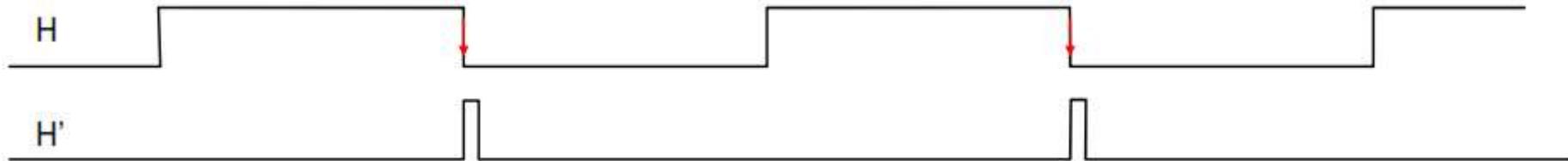
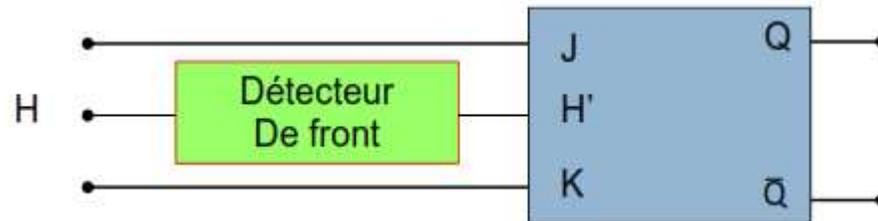
Deux techniques :

- Utiliser un détecteur de front sur l'entrée horloge
- Utiliser une structure Maître Esclave

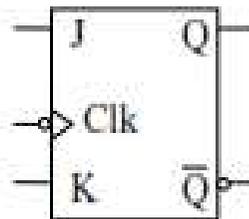
Logique séquentielle

II. Les Bascules

□ Bascule JK avec détecteur de front descendant



H	J	K	Q
↓	0	0	mémoire
↓	0	1	0
↓	1	0	1
↓	1	1	basculement

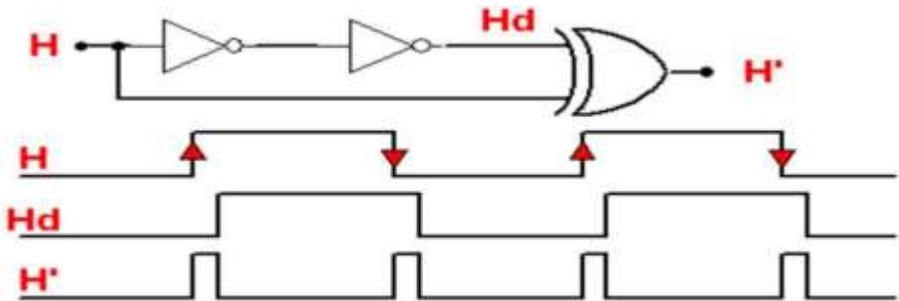
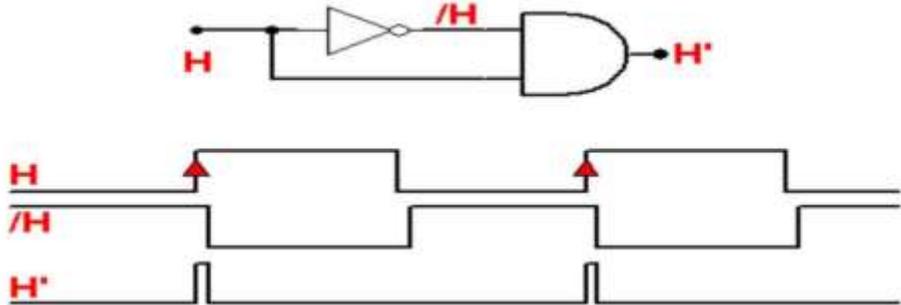
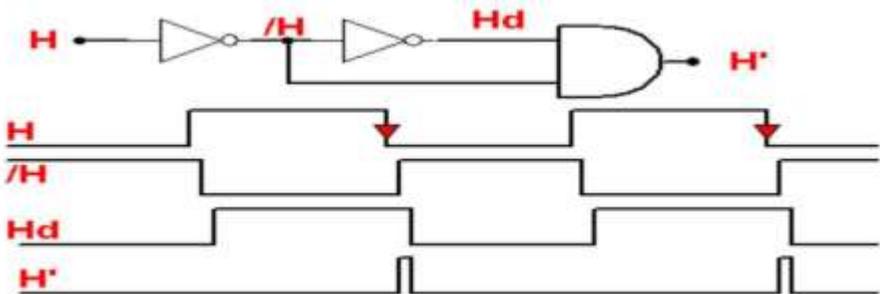


H	J	K	Q	Observation
↓	0	0	Q _p	Mémoire
↓	0	1	0	Sortie suit J
↓	1	0	1	
↓	1	1	\bar{Q}_p	Alternance

Logique séquentielle

II. Les Bascules

□ Exemple de détecteur de front

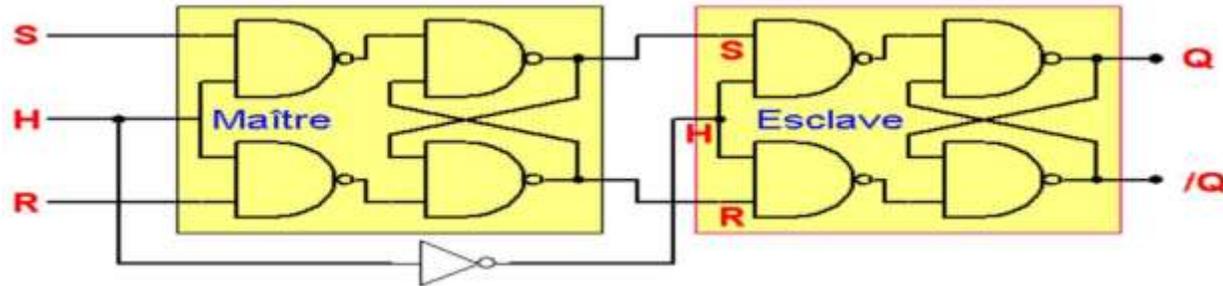


On exploite le retard élémentaire des portes logiques

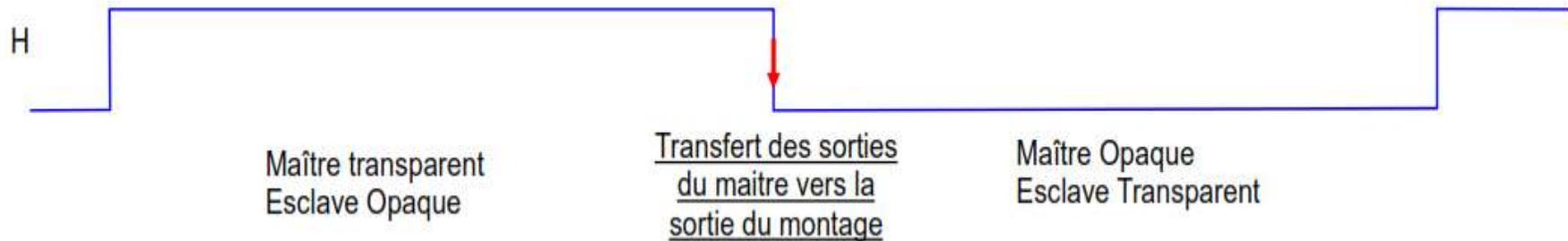
Logique séquentielle

II. Les Bascules

□ Bascule RS maître esclave



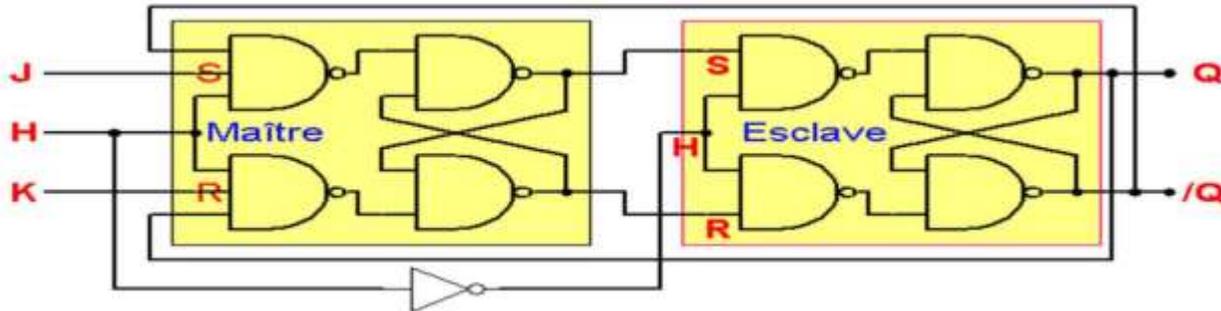
H	S	R	Q
↓	0	0	mémoire
↓	0	1	0
↓	1	0	1
↓	1	1	indéterminé



Logique séquentielle

II. Les Bascules

□ Bascule JK maître esclave



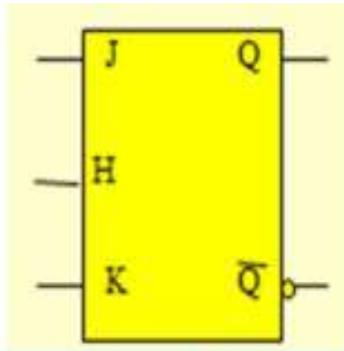
H	S	R	Q
↓	0	0	mémoire
↓	0	1	0
↓	1	0	1
↓	1	1	basculement



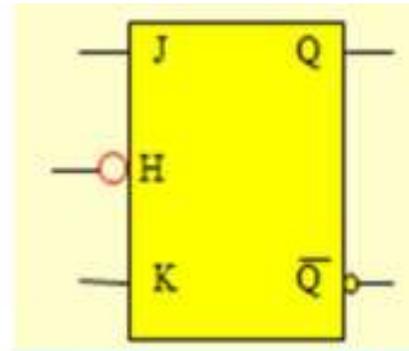
Logique séquentielle

II. Les Bascules

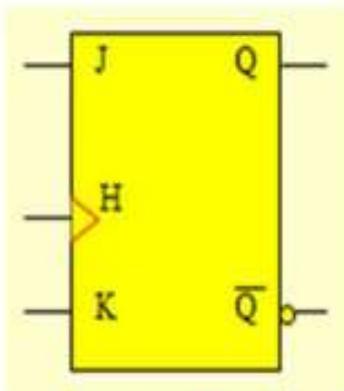
□ Convention de dessin



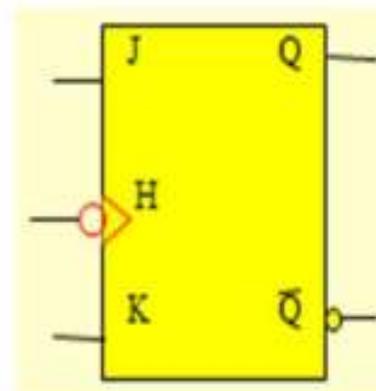
Bascule réagissant sur **niveau haut** de H (latch)



Bascule réagissant sur **niveau bas** de H (latch)



Bascule réagissant sur **front montant** de H



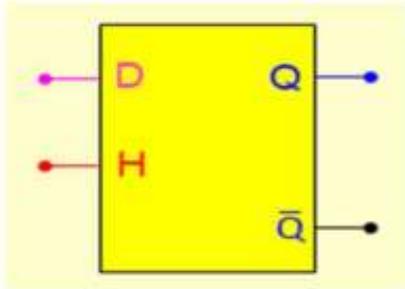
Bascule réagissant sur **front descendant** de H

Logique séquentielle

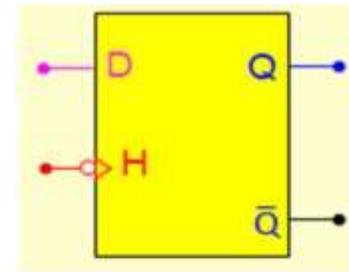
II. Les Bascules

□ Bascule D

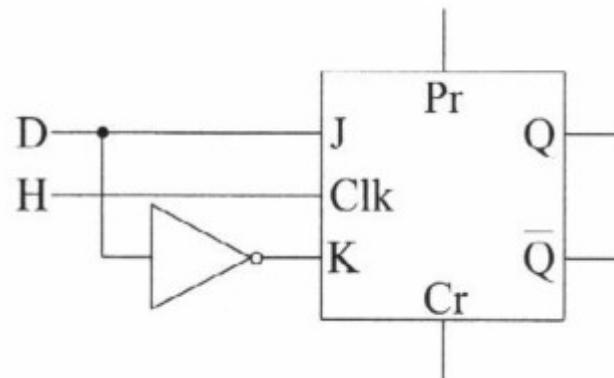
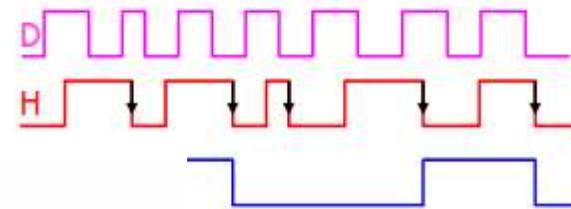
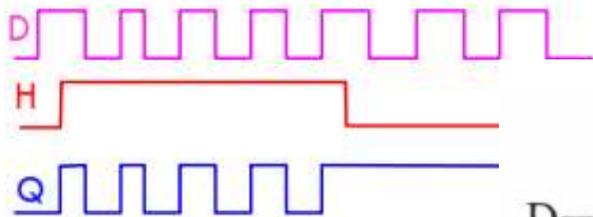
permettent de générer un "retard" (delay) ou de stocker de l'information (latch)



- $H = 1$: Q suit D
- $H = 0$: Q reste bloqué : état mémoire



Pendant le **front descendant** de H, Q prend la valeur de D et la garde jusqu'au prochain front

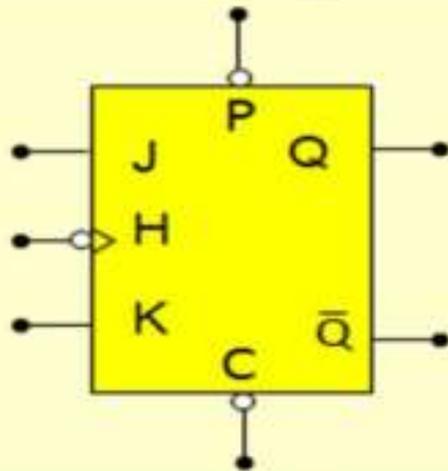


Logique séquentielle

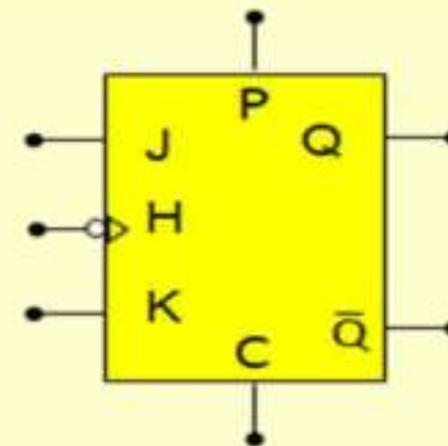
II. Les Bascules

□ Entrée de forçage Clear et Preset

- L'entrée **C** force la sortie à **0** $\forall J, K, H$
- L'entrée **P** force la sortie à **1** $\forall J, K, H$
- Il ne faut pas activer C et P simultanément



C et P actifs au
niveau bas



C et P actifs au
niveau haut

III. Les Compteurs

- ❑ **compteur** : ensemble de n bascules connectées par des portes logiques
- ❑ décrivent une **séquence** déterminée (c'est-à-dire occuper une suite d'états binaires) au rythme d'une horloge
- ❑ 2^n combinaisons possibles
- ❑ les états sont stables et accessibles entre deux impulsions de l'horloge
- ❑ N nombre total de combinaisons successives utilisées $N \leq 2^n$
: **modulo** du compteur
- ❑ compteur asynchrones ou synchrones
 - réversibles ou compteurs-décompteurs

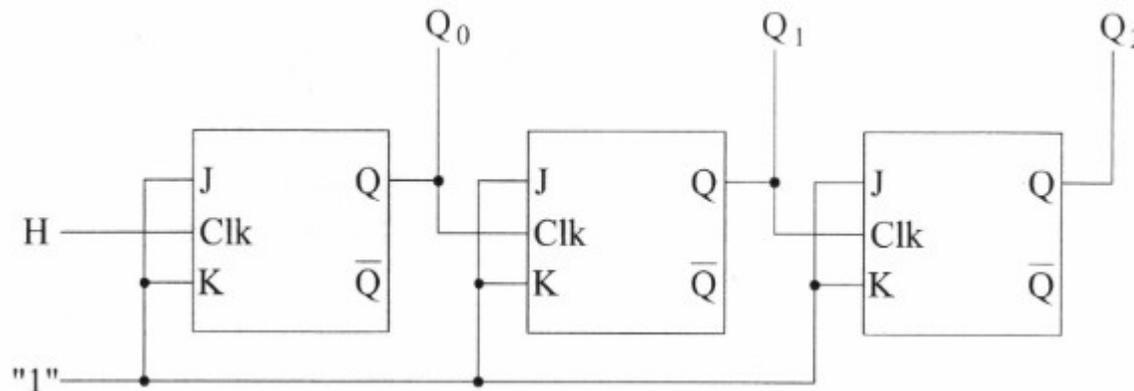
Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone

Ce type de compteur est constitué de n bascules JK fonctionnant en mode T (Toggle) : $J=K=1$. Ces bascules sont montées en cascade c'est-à-dire le signal d'horloge commande uniquement la première bascule tandis que pour chacune des autres bascules le signal d'horloge est fourni par la sortie de la bascule de rang immédiatement inférieur.

- Pour bien comprendre le principe, réalisons un compteur modulo 8 permettant de compter de 0 à 7 :

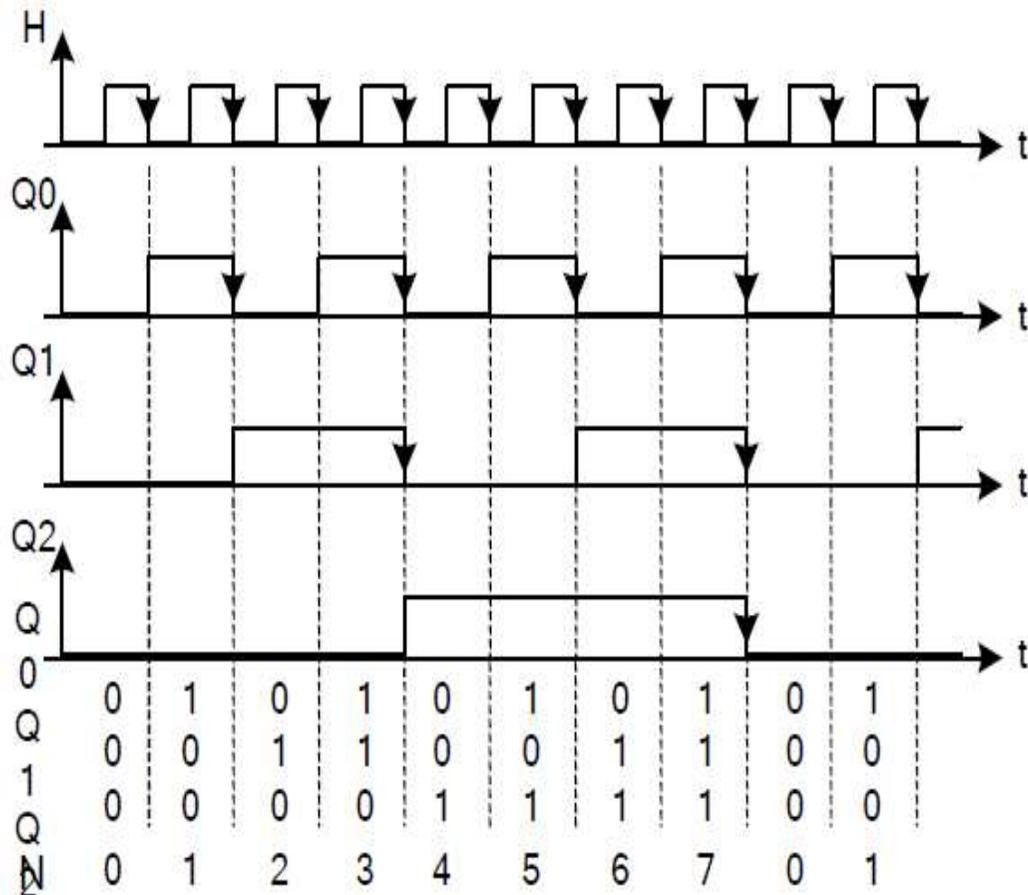


Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone modulo $N=2^n$

Les chronogrammes et la table de vérité d'un tel compteur sont

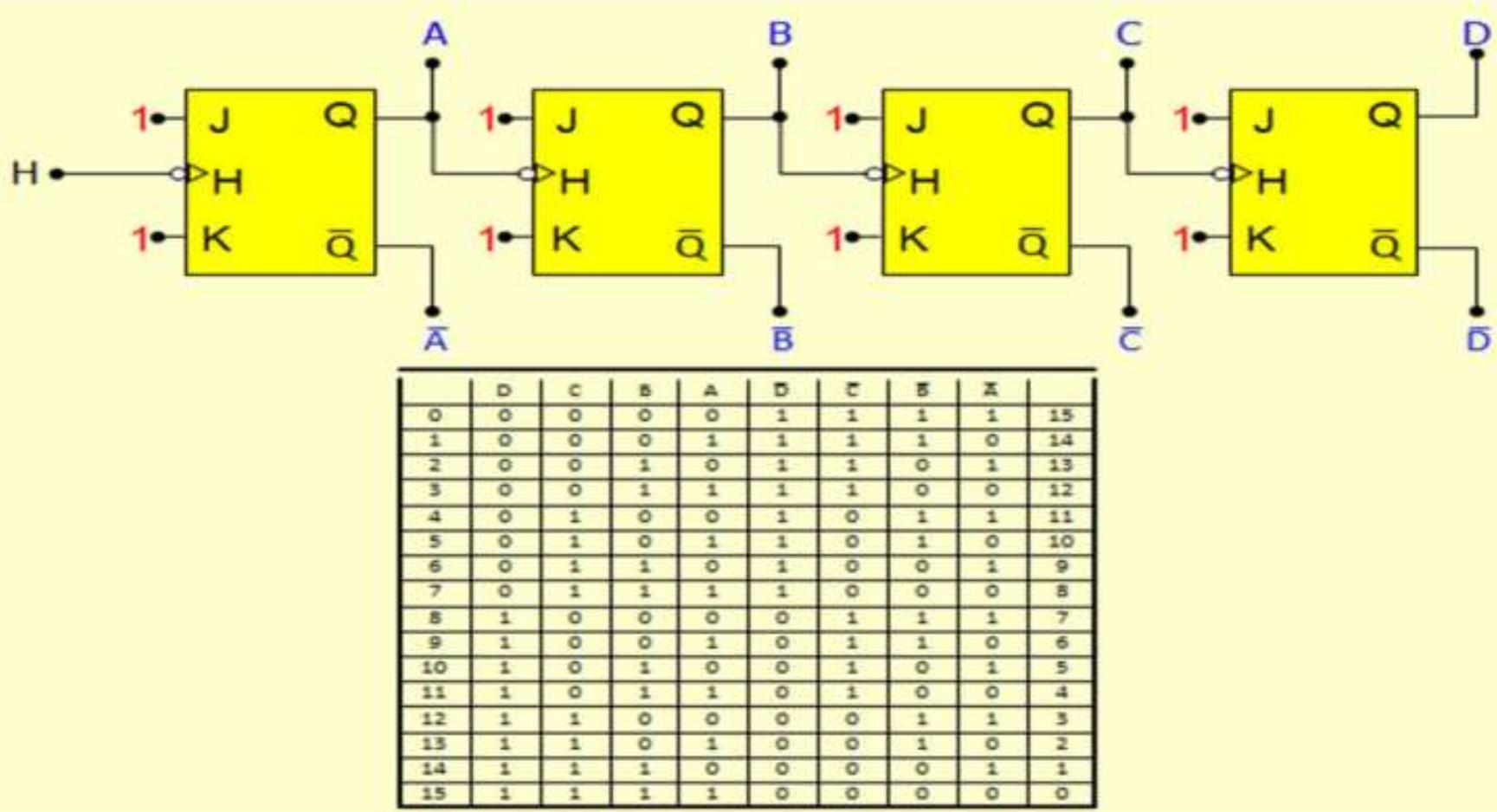


H	Q3(MSB)	Q2	Q1(LSB)
...	0	0	0
↓	0	0	1
↓	0	1	0
↓	0	1	1
↓	1	0	0
↓	1	0	1
↓	1	1	0
↓	1	1	1
↓	0	0	0

Logique séquentielle

III. Les Compteurs

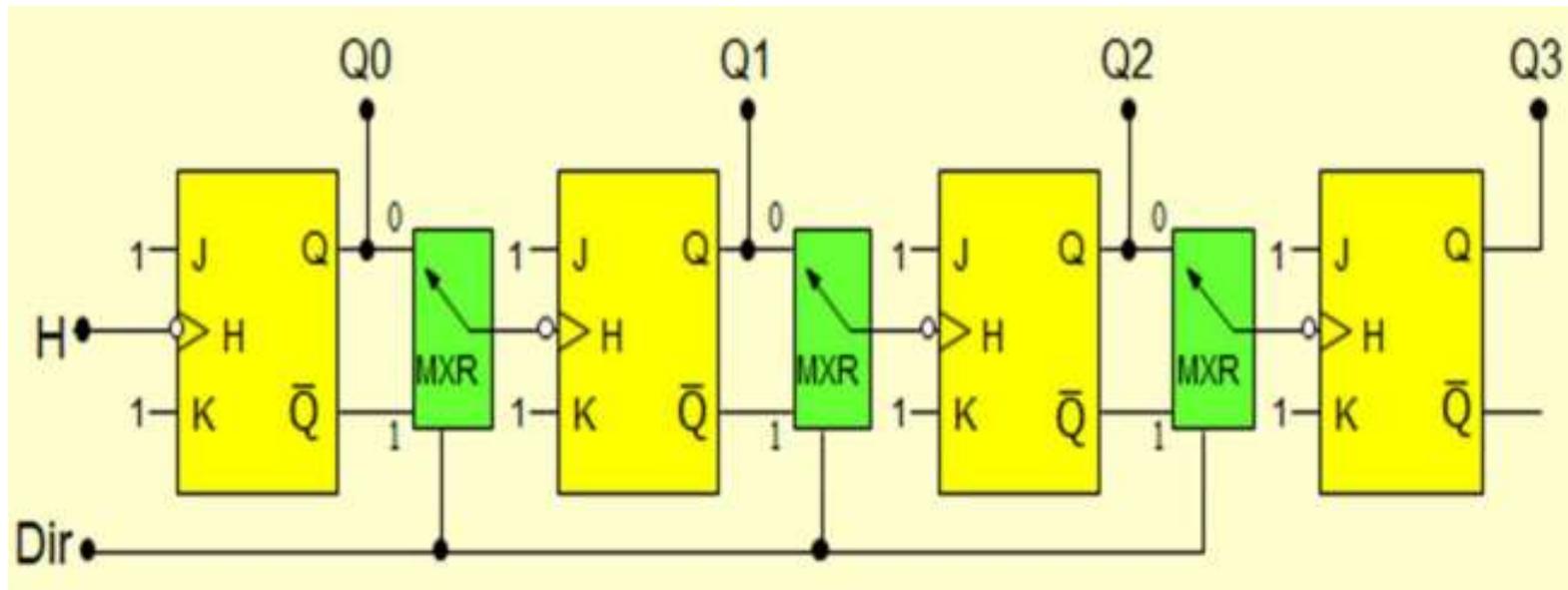
1. Décompteur asynchrone (autre méthode)



Logique séquentielle

III. Les Compteurs

1. Compteur-Décompteur asynchrone

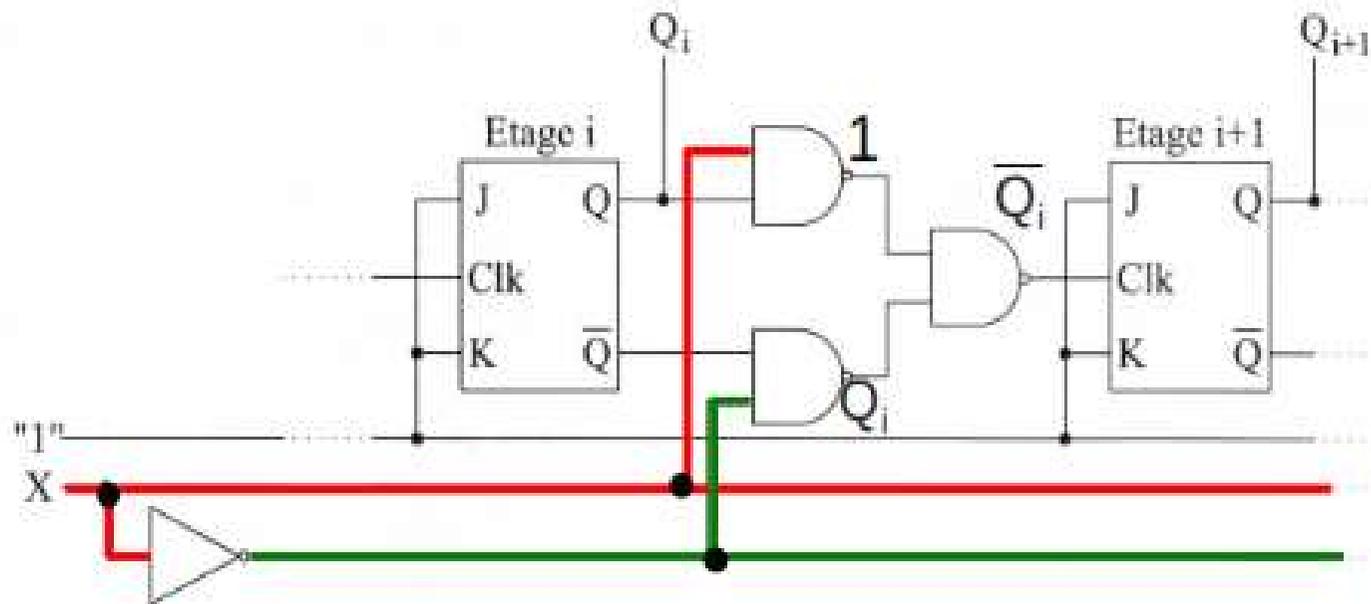


Au changement de direction, si $Q=1$, le passage à \bar{Q} génère un front descendant provoquant un changement d'état de la bascule suivante ce qui peut être indésirable.

Logique séquentielle

III. Les Compteurs

1. Compteur-Décompteur asynchrone



- ✓ la ligne de commande X permet de sélectionner le mode :
 - ✗ $X=0 \Rightarrow$ décompteur

Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone modulo $N \neq 2^n$

Compteur n bits

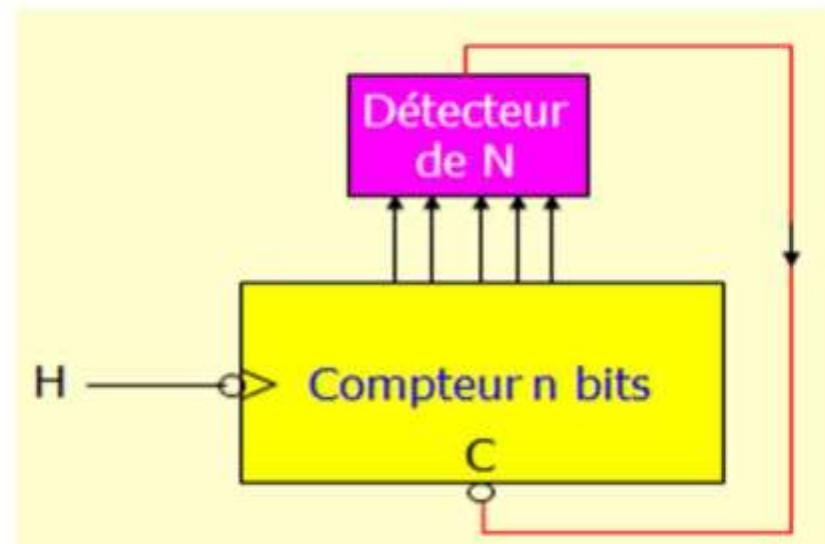


Compteur modulo 2^n

Compteur modulo $N < 2^n$



Détecter N et razer le Compteur



III. Les Compteurs

1. Compteurs asynchrone modulo $N \neq 2^n$

Pour ce type général de compteur qui compte de 0 à $N-1$, on va étudier l'exemple d'un compteur asynchrone modulo 10 (0 à 9). Pour le réaliser, il y a deux étapes :

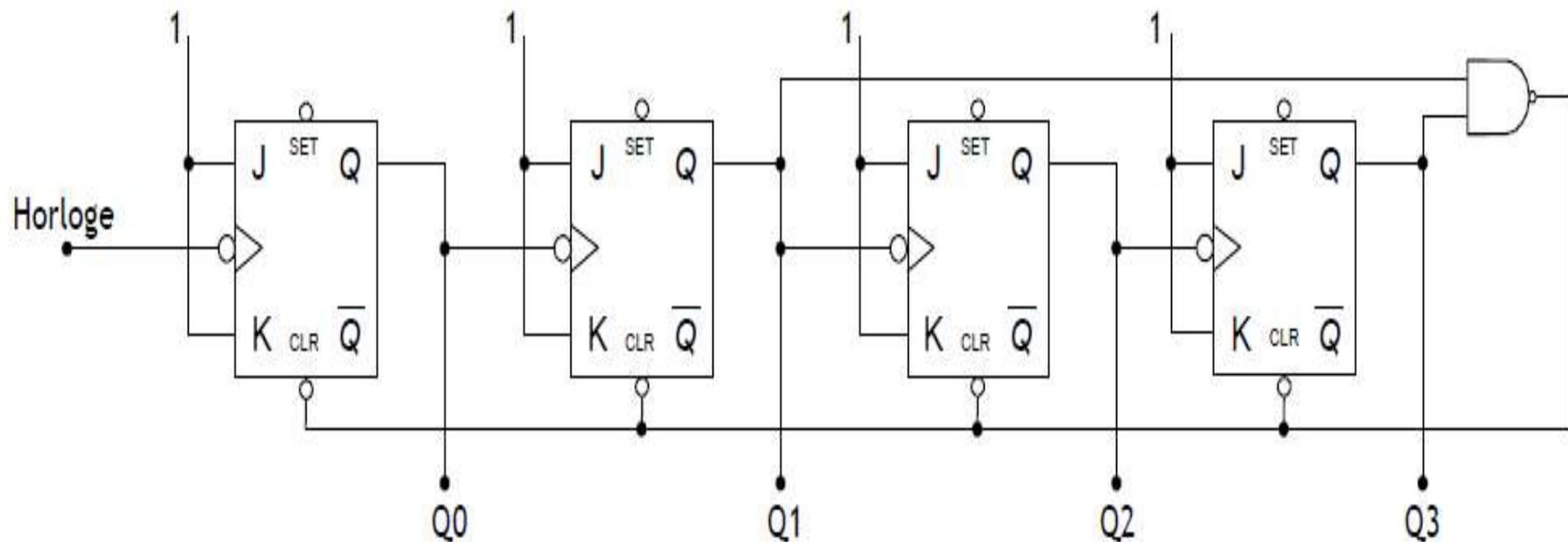
❖ On cherche d'abord la puissance de 2 immédiatement supérieure à N et qui est pour notre compteur $2^4 = 16$. L'exposant de cette puissance de 2 donne le nombre de bascules JK à monter en cascade, 4 pour notre exemple ;

❖ On détecte ensuite l'état N qui remettra le compteur à 0 et qui est pour notre compteur $10 = (1010)_2$. On relie les sorties $Q = 1$ (Q_1 et Q_3) pour N aux entrées d'une porte NAND dont la sortie commandera l'entrée CLR de chaque bascule .

Logique séquentielle

III. Les Compteurs

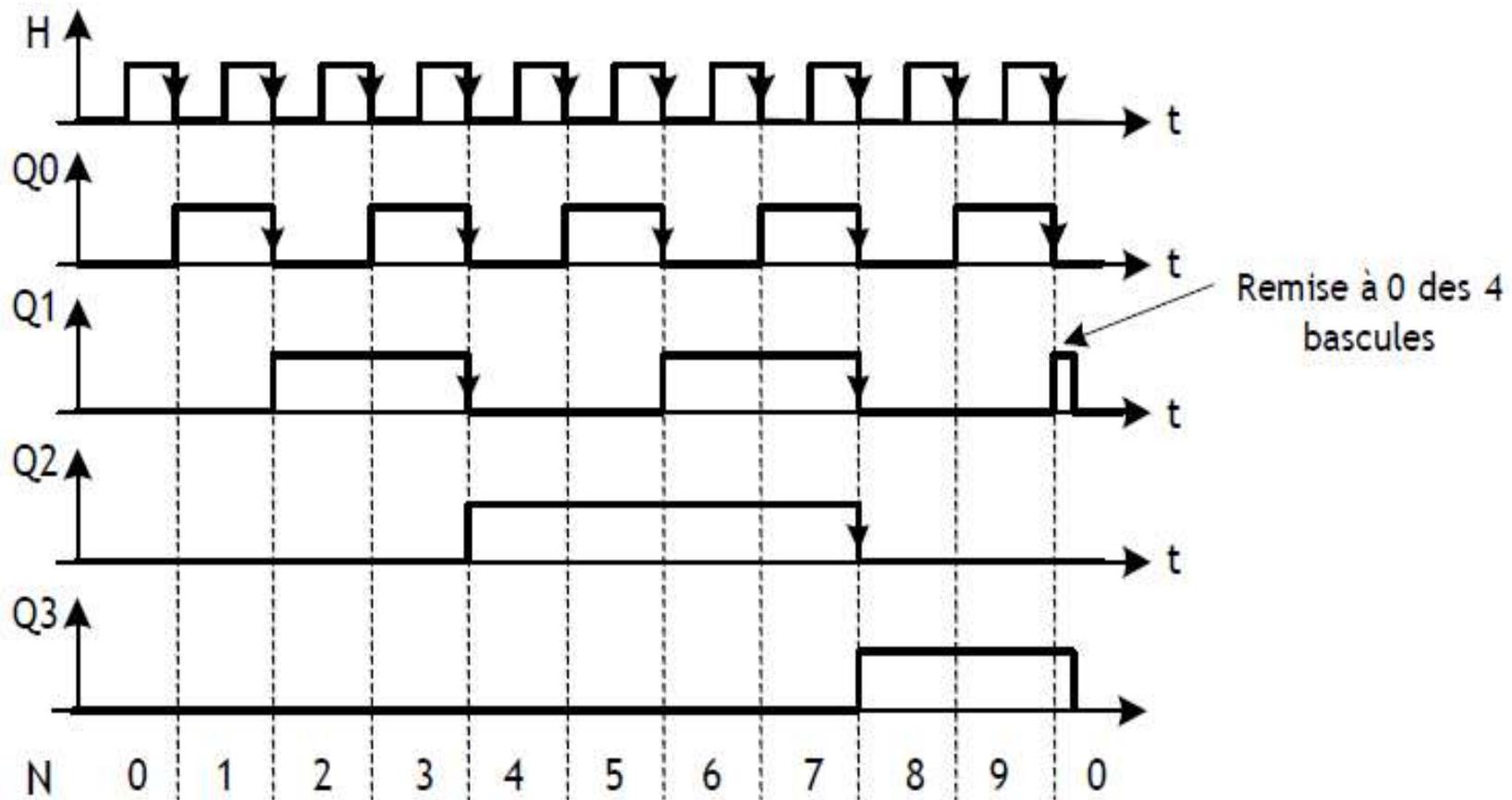
1. Compteurs asynchrone modulo $N \neq 2^n$



Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone modulo $N \neq 2^n$

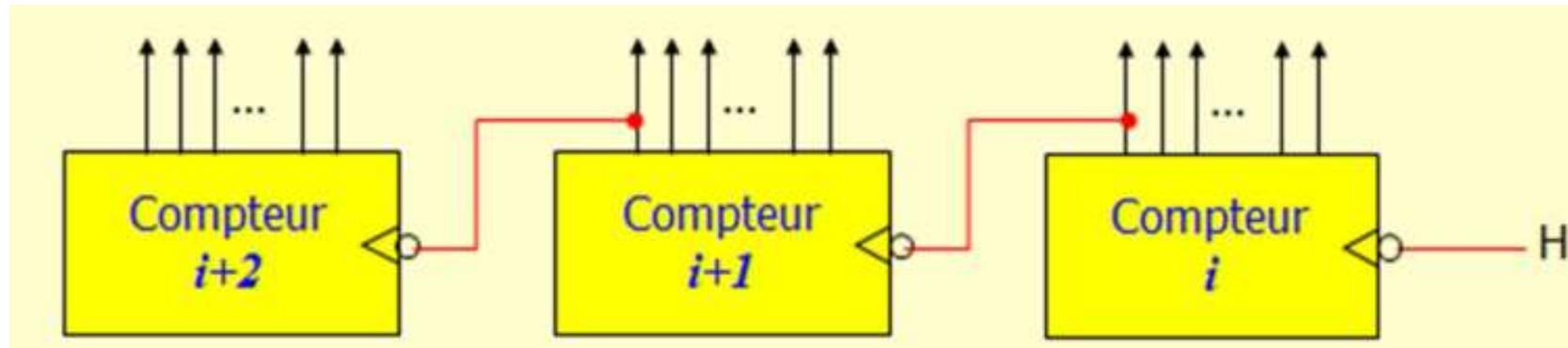


Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone

Mise en cascade des compteurs asynchrone



- Le MSB du compteur i est appliqué à l'horloge du compteur $i+1$ (suivant)
- Chaque fois qu'un compteur repasse à 0, le compteur suivant est incrémenté

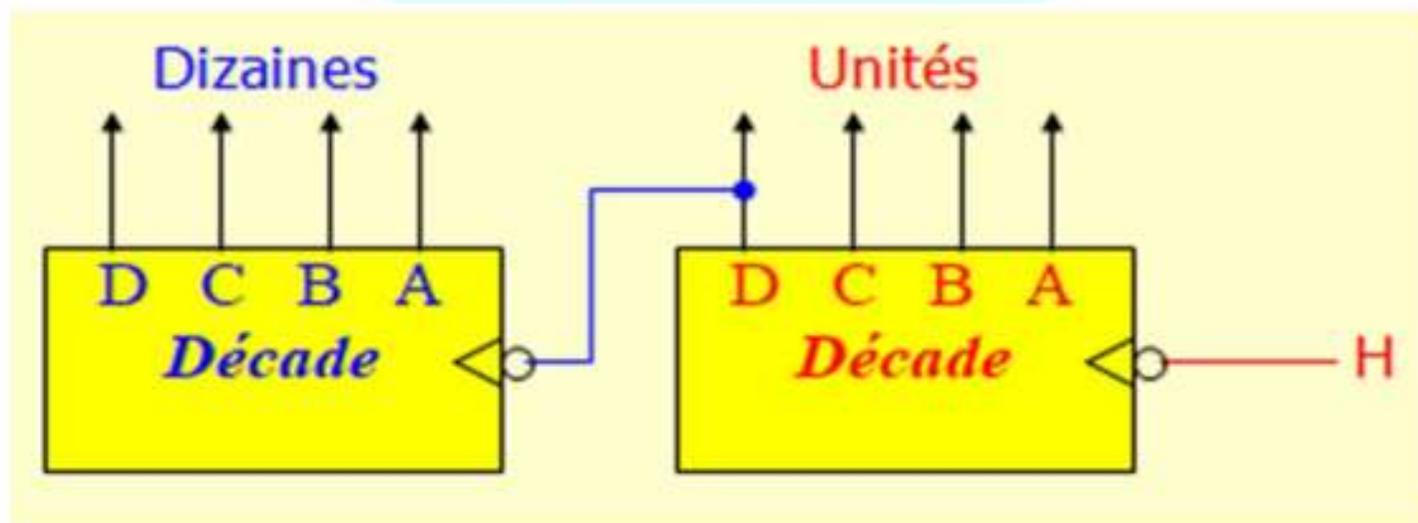
Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone

compteurs asynchrone BCD

On obtient un compteur BCD en cascadeant des compteurs modulo 10

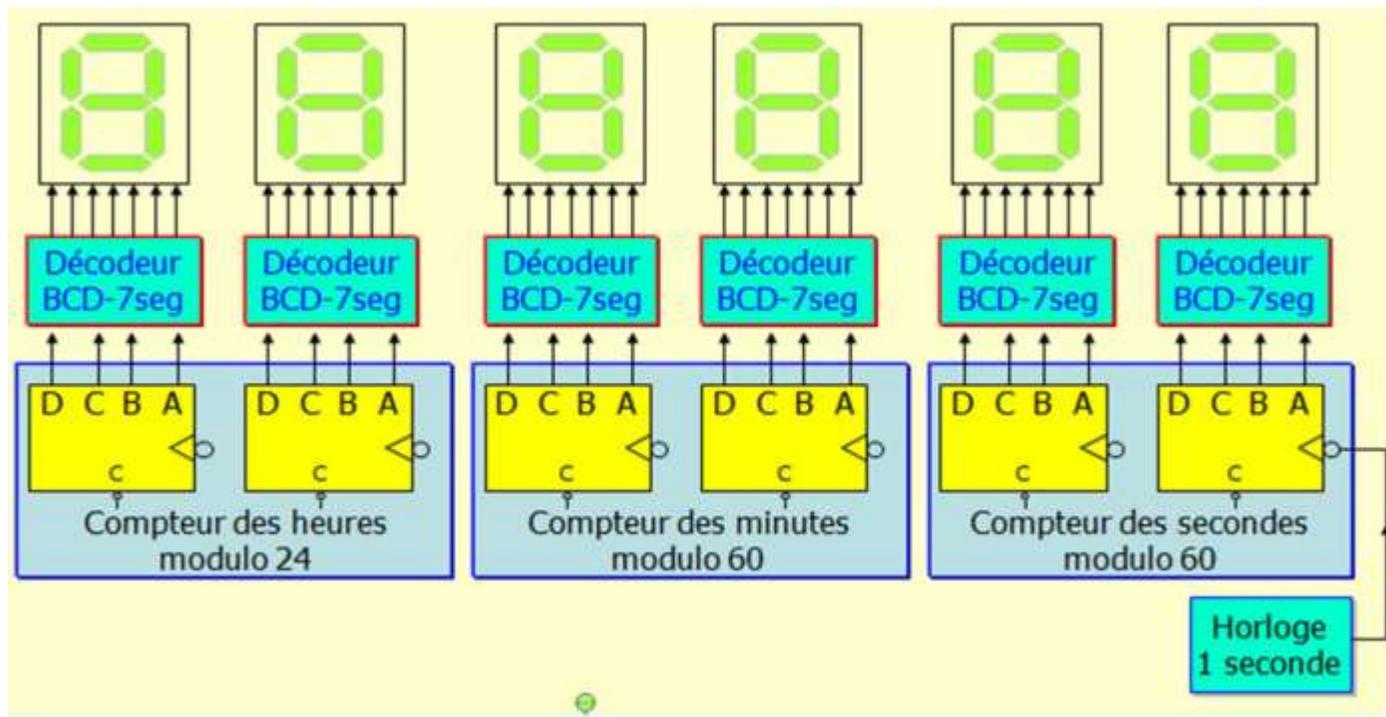


Logique séquentielle

III. Les Compteurs

1. Compteurs asynchrone

Horloge



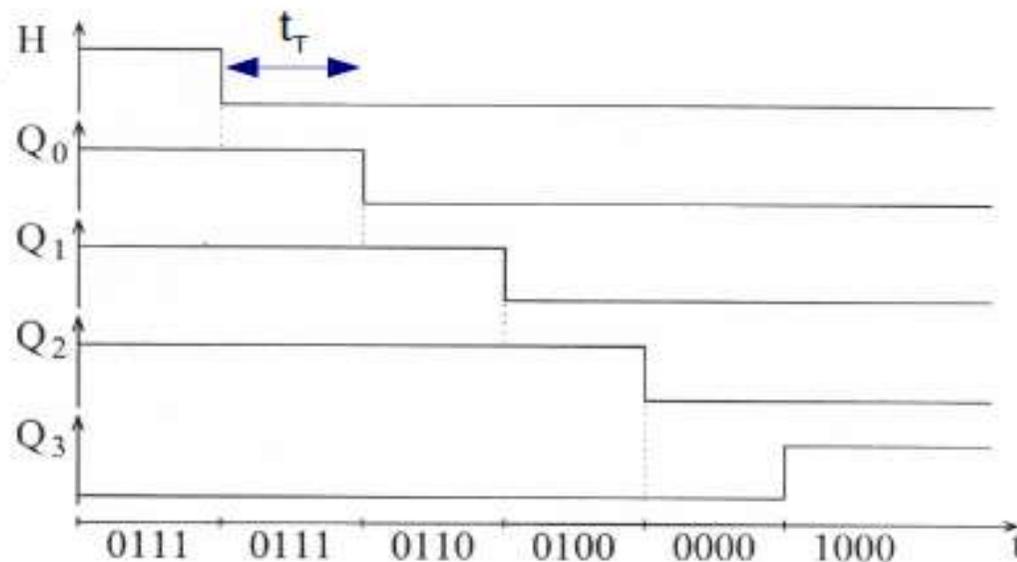
Tous les compteurs sont des 4 bits (binaire naturel)

Donner les interconnexions nécessaires pour obtenir un compteur secondes, minutes, heures

III. Les Compteurs

1. Compteurs asynchrone

- ❑ inconvénient des compteurs asynchrones
- ✓ à cause du temps de réponse non nul des bascules, le signal d'horloge ne parvient pas simultanément à toutes les bascules
- ✓ exemple : passage de 0111 à 1000 sur un compteur modulo 16, avec un temps de réponse t_T pour toutes les bascules



on passe par les états transitoires 0110, 0100, et 0000 qui sont faux

III. Les Compteurs

3. Compteurs synchrone

- toutes les bascules reçoivent en parallèle le même signal d'horloge
- pour faire décrire au compteur une séquence déterminée, il faut définir les entrées J et K de chaque bascule à chaque top d'horloge, en utilisant la table de transition des bascules J-K

Q_n	Q_{n+1}	J_n	K_n
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Synthèse d'un compteur synchrone 4 bits

	D	C	B	A	J _A	K _A	J _B	K _B	J _C	K _C	J _D	K _D
0	0	0	0	0	1	x	0	x	0	x	0	x
1	0	0	0	1	x	1	1	x	0	x	0	x
2	0	0	1	0	1	x	x	0	0	x	0	x
3	0	0	1	1	x	1	x	1	1	x	0	x
4	0	1	0	0	1	x	0	x	x	0	0	x
5	0	1	0	1	x	1	1	x	x	0	0	x
6	0	1	1	0	1	x	x	0	x	0	0	x
7	0	1	1	1	x	1	x	1	x	1	1	x
8	1	0	0	0	1	x	0	x	0	x	x	0
9	1	0	0	1	x	1	1	x	0	x	x	0
10	1	0	1	0	1	x	x	0	0	x	x	0
11	1	0	1	1	x	1	x	1	1	x	x	0
12	1	1	0	0	1	x	0	x	x	0	x	0
13	1	1	0	1	x	1	1	x	x	0	x	0
14	1	1	1	0	1	x	x	0	x	0	x	0
15	1	1	1	1	x	1	x	1	x	1	x	1

Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Synthèse d'un compteur synchrone 4 bits

En utilisant les tables de Karnaugh, on obtient :

$$J_A = 1$$

$$J_B = A$$

$$J_C = AB$$

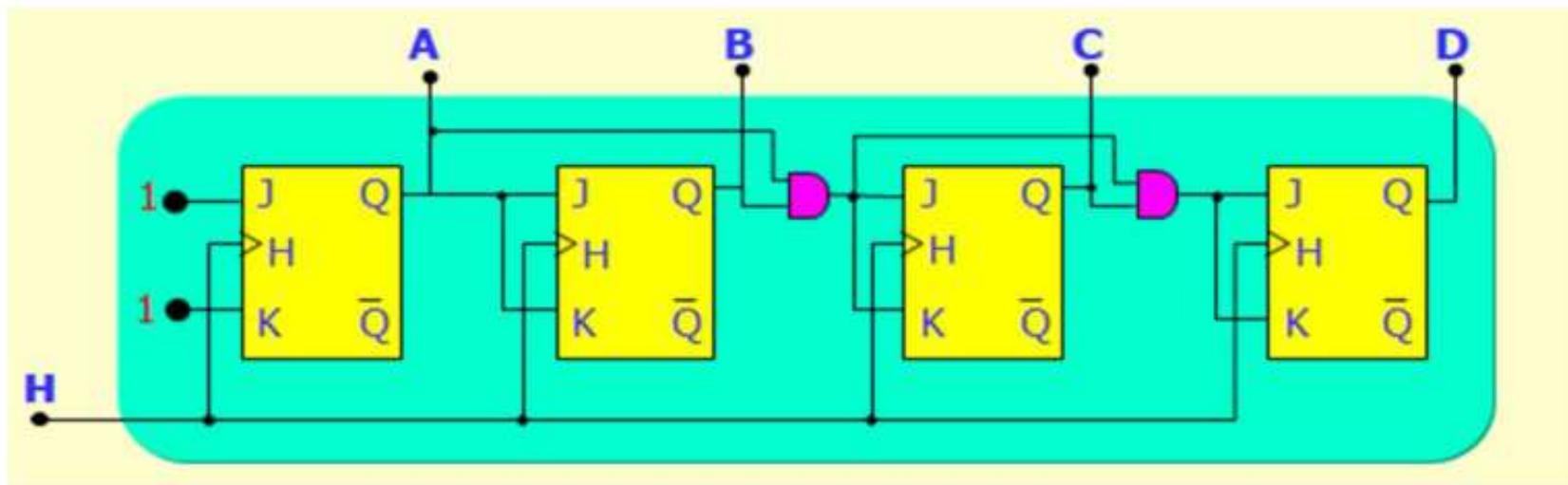
$$J_D = ABC$$

$$K_A = 1$$

$$K_B = A$$

$$K_C = AB$$

$$K_D = ABC$$



Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Synthèse d'un **décompteur** synchrone 4 bits

Avec une étude similaire, on obtient :

$$J_A = 1$$

$$J_B = A$$

$$J_C = A B$$

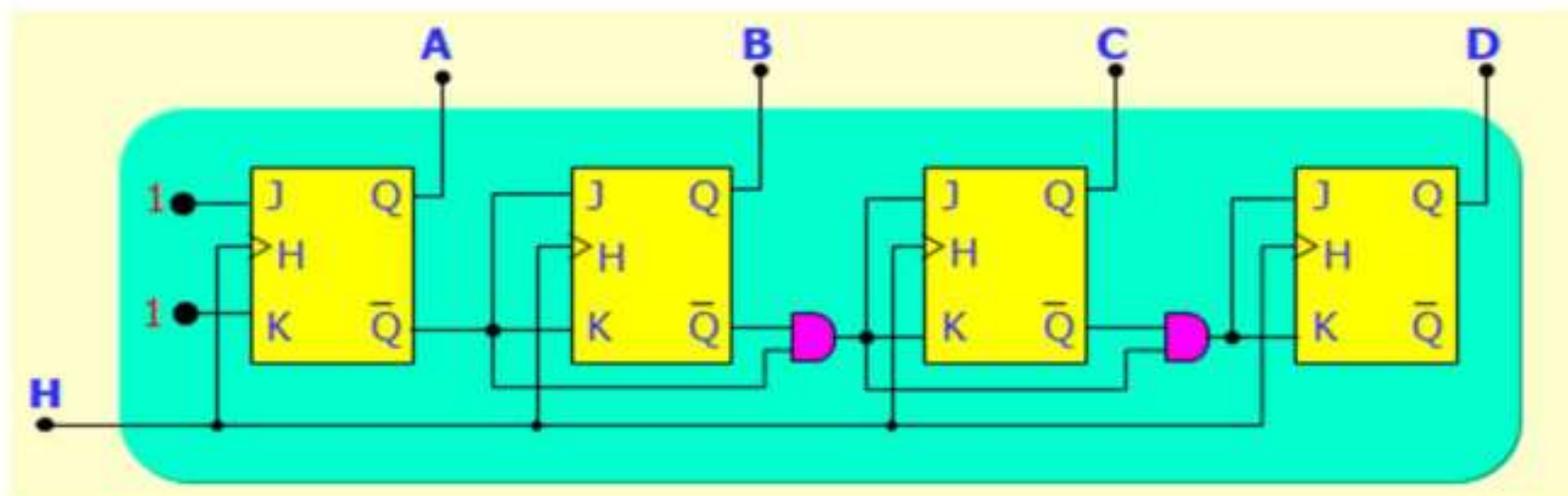
$$J_D = A B C$$

$$K_A = 1$$

$$K_B = A$$

$$K_C = A B$$

$$K_D = A B C$$

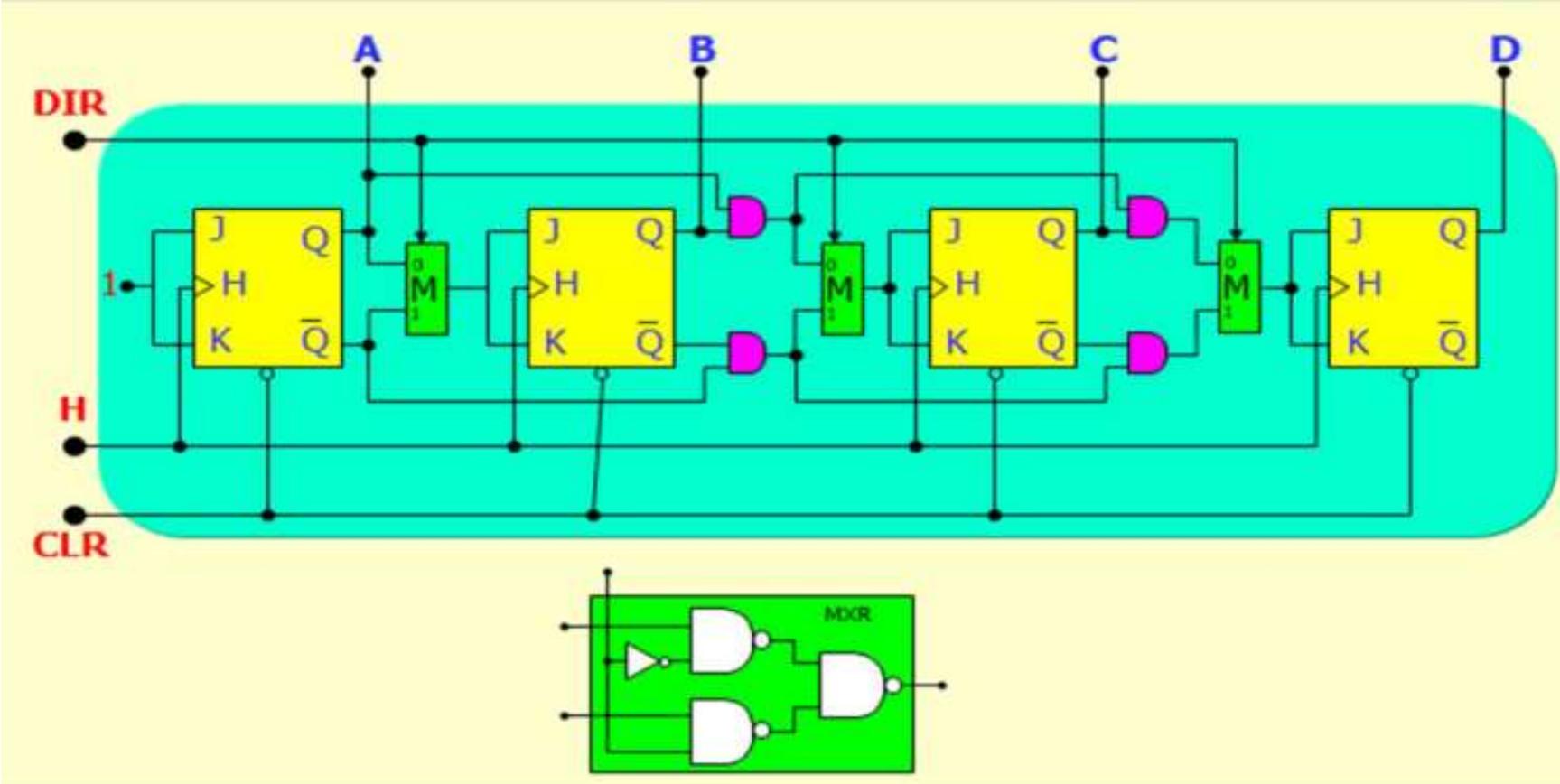


Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Compteur- décompteur synchrone

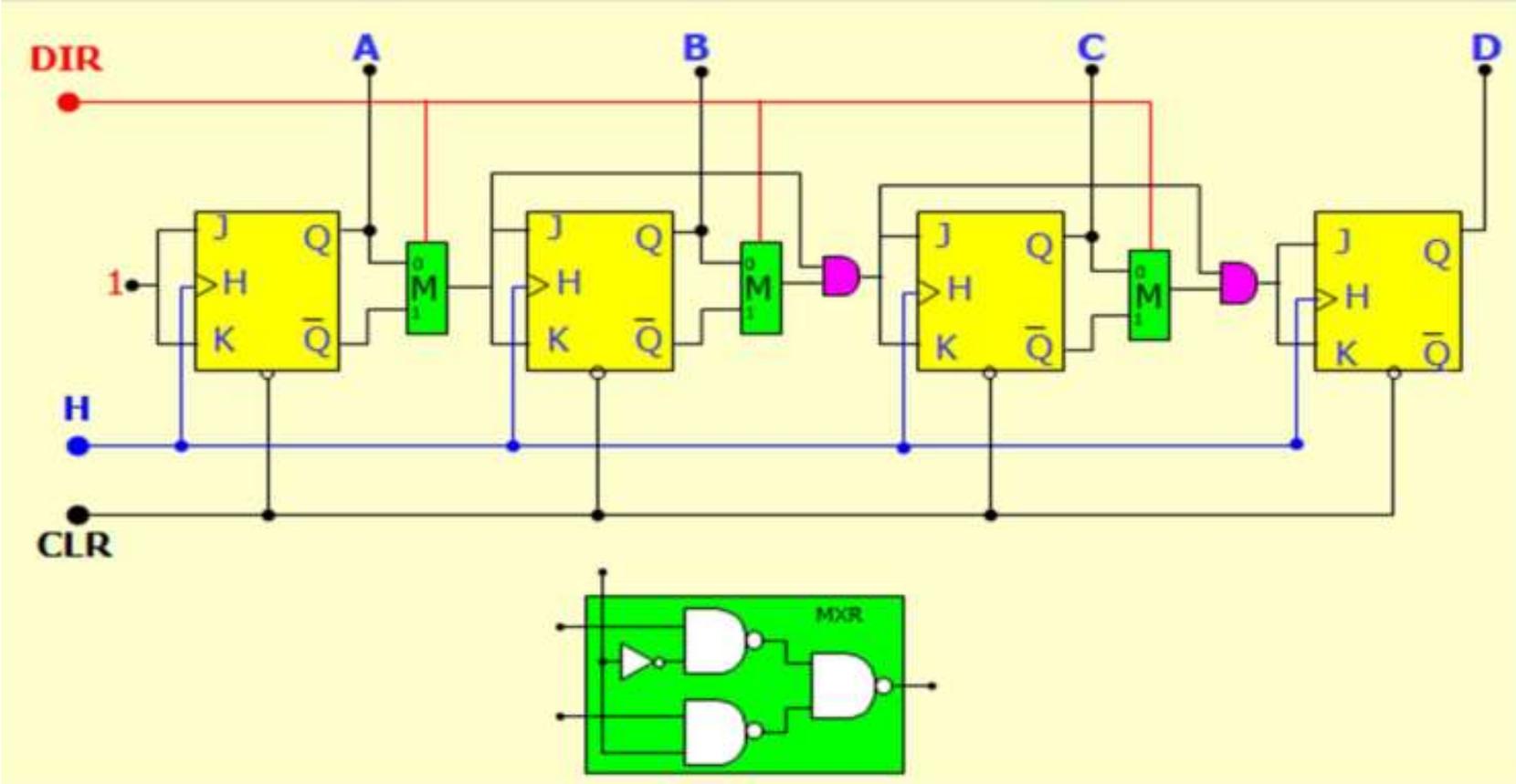


Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Compteur- décompteur synchrone avec RAZ

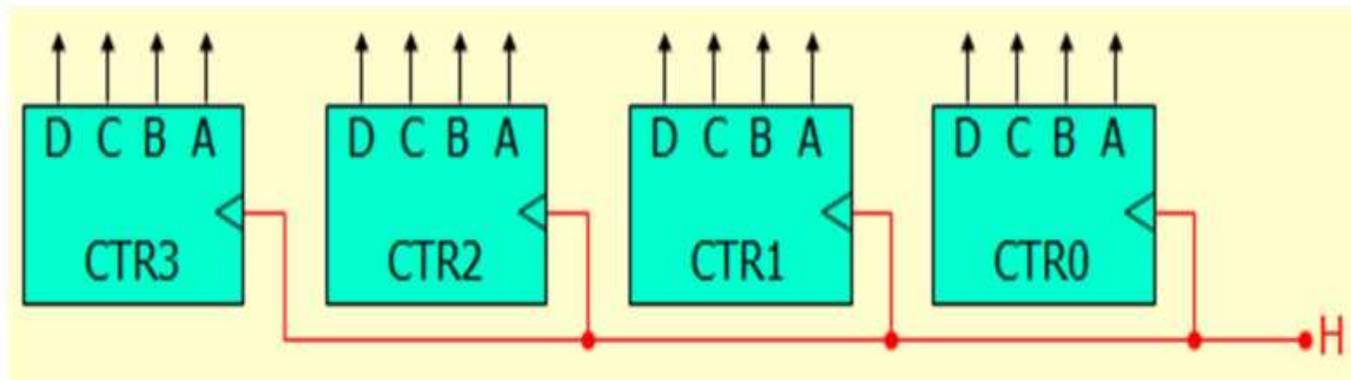


Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Mise en cascade des compteurs synchrones



La mise en cascade doit être **synchrone**. Tous les compteurs doivent recevoir la **même Horloge**

Dans ces conditions, tous les compteurs fonctionneront simultanément et on n'aura pas le comptage désiré.

Il faut qu'un compteur ne s'incrémente que pendant le débordement du compteur précédent.

Logique séquentielle

III. Les Compteurs

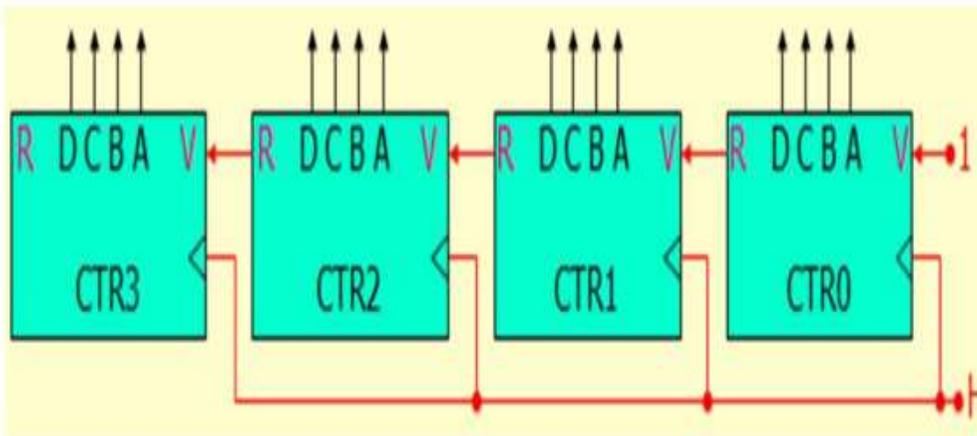
3. Compteurs synchrones

Mise en cascade des compteurs synchrones

On va rajouter à chaque compteur une entrée de **validation V** et une sortie de **retenue R**

L'**entrée** de validation V permettra de le contrôler : $V=1 \rightarrow$ Comptage, $V=0 \rightarrow$ arrêt

La **sortie** de retenue R passe à 1 pour indiquer au compteur suivant qu'il peut s'incrémenter



Un compteur ne doit s'incrémenter que si tous les compteurs qui sont à sa droite sont à 9. Autrement dit un compteur ne doit valider le compteur suivant ($R=1$) que si lui même contient 9 et il reçoit $V=1$ (le précédent = 9)

$R = 1$ quand $(DCBA=9)$ ET $V=1$

Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

compteur synchrone a base de bascule D

Faire l'étude d'un compteur Synchrone 4 bits à base de bascules D

D	C	B	A	D _D	D _C	D _B	D _A
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

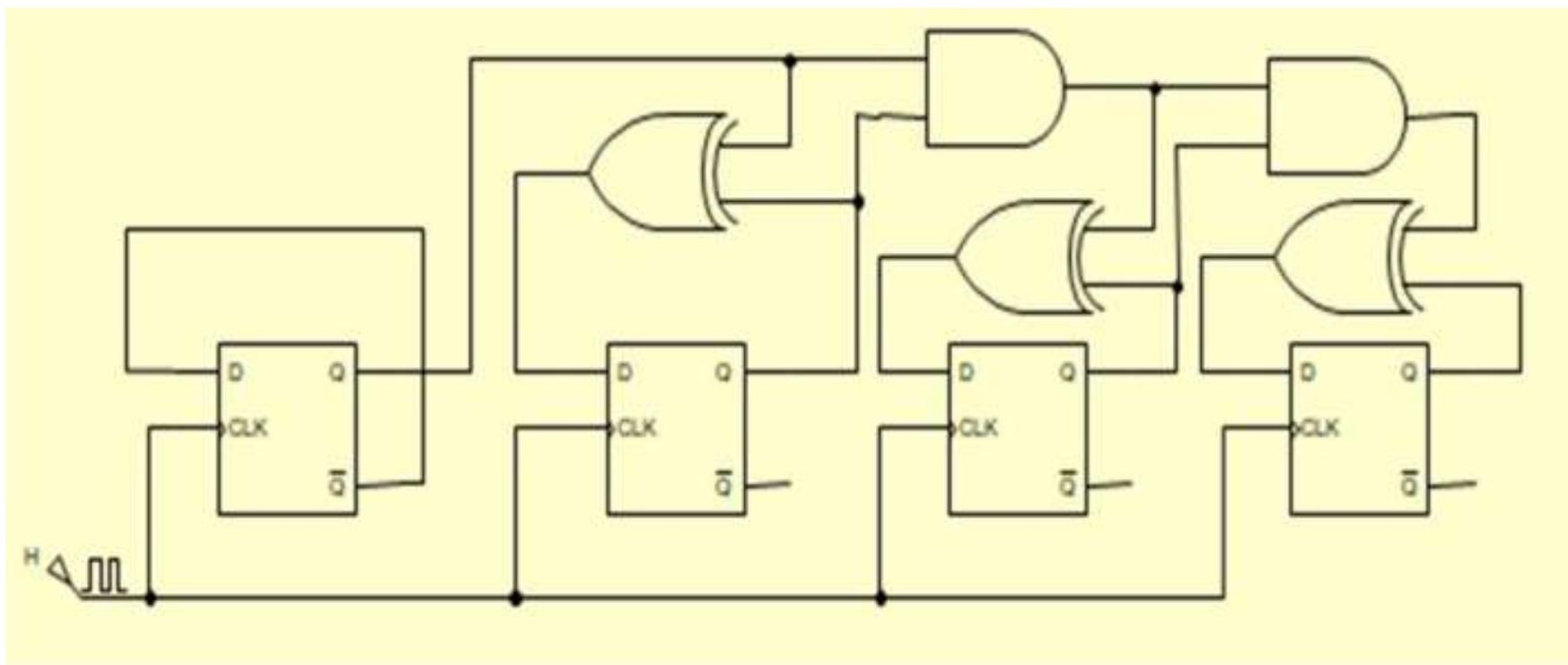
- $D_A = \bar{A}$
- $D_B = B \oplus A$
- $D_C = C \oplus (AB)$
- $D_D = D \oplus (ABC)$

Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

compteur synchrone a base de bascule D



Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

Décompteur synchrone a base de bascule D

- Faire l'étude d'un décompteur Synchrone 4 bits à base de bascules D

D	C	B	A	D _D	D _C	D _B	D _A
0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	0	1	1	0
1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	1
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	1	1	0	0
1	1	1	0	1	1	0	1
1	1	1	1	1	1	1	0

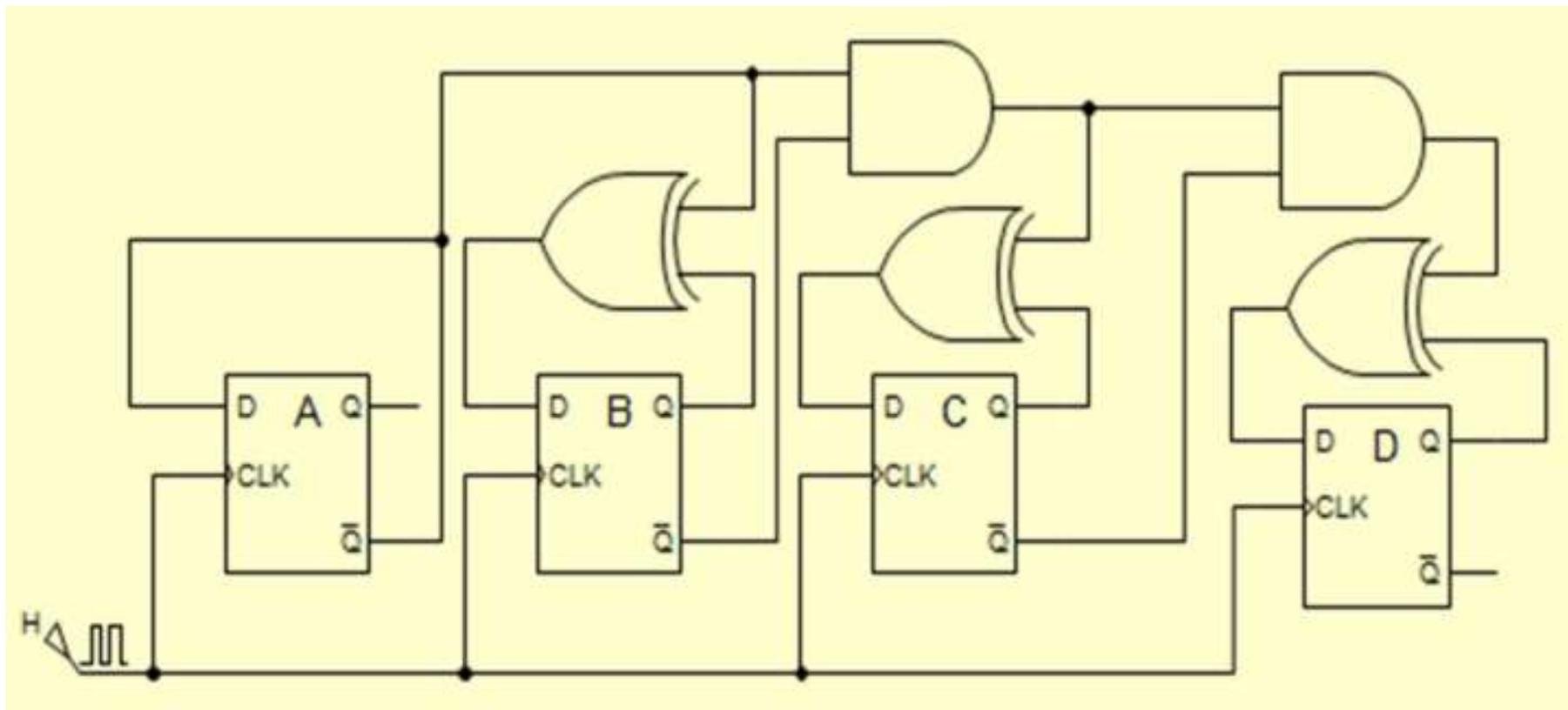
- $D_A = \bar{A}$
- $D_B = B \oplus \bar{A}$
- $D_C = C \oplus (\bar{A}\bar{B})$
- $D_D = D \oplus (\bar{A}\bar{B}\bar{C})$

Logique séquentielle

III. Les Compteurs

3. Compteurs synchrone

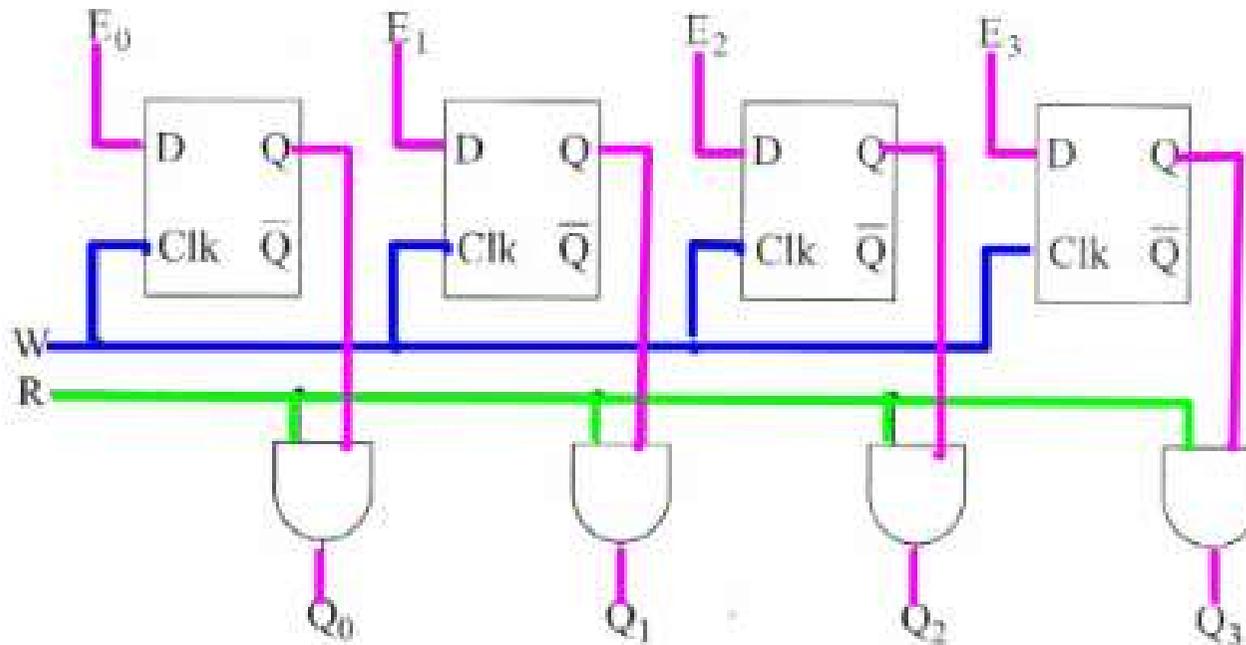
Décompteur synchrone a base de bascule D



Logique séquentielle

IV. Les Registres

- association de n bascules D pour mémoriser n bits
- Les entrées présentes sur E3, E2, E1 et E0 sont mémorisées en synchronisation avec les signal W
- Elles peuvent être lues sur les sortie Q3, Q2, Q1 et Q0 en coïncidence avec le signal de validation R

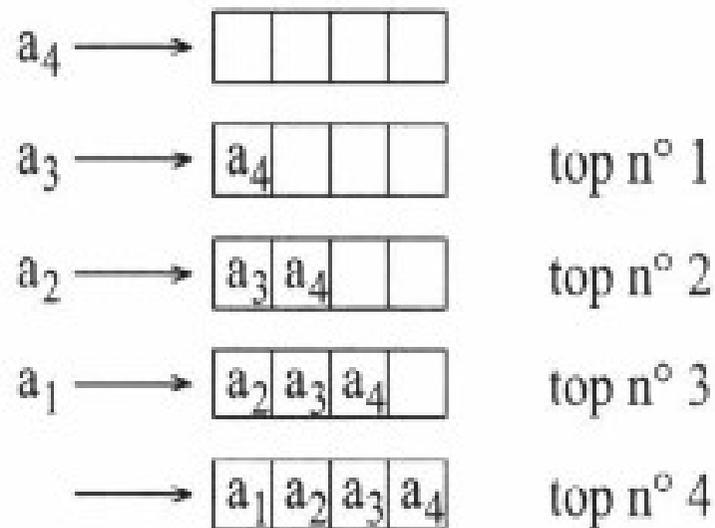


Logique séquentielle

IV. Les Registres

□ Registres a décalage

- bascules D interconnectées de façon à ce que l'état de la bascule de rang i soit transmis à la bascule de rang $i+1$ (ou $i-1$) quand un signal d'horloge est appliqué à l'ensemble des bascules
- 2 types d'entrées
 - ✓ parallèle, comme dans un registre de mémorisation
 - ✓ série
 - l'information est présentée séquentiellement bit après bit à la 1ère bascule
 - à chaque coup d'horloge, un nouveau bit est présenté et ceux déjà chargés sont décalés d'un rang

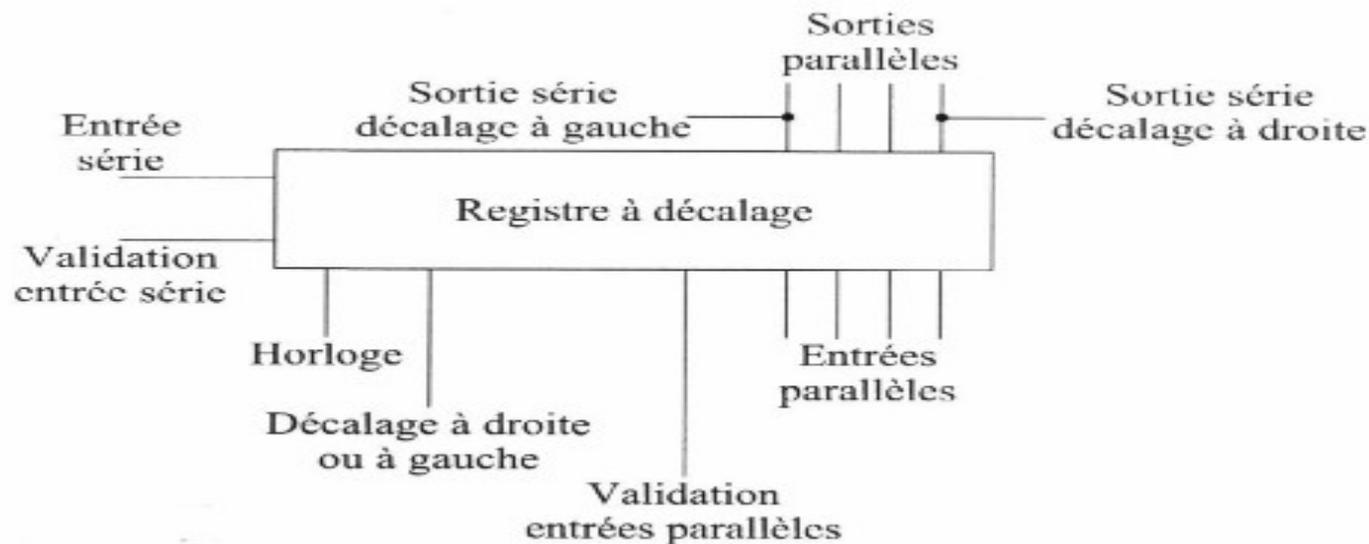


Logique séquentielle

IV. Les Registres

□ Registres a décalage

- l'information stockée dans un registre à décalage peut être lue de la même manière, en série ou en parallèle
- certains registres sont capables de décaler à droite ou à gauche (registres à décalage universels)
- généralement réalisés avec des bascules du type maître esclave D ou R-S



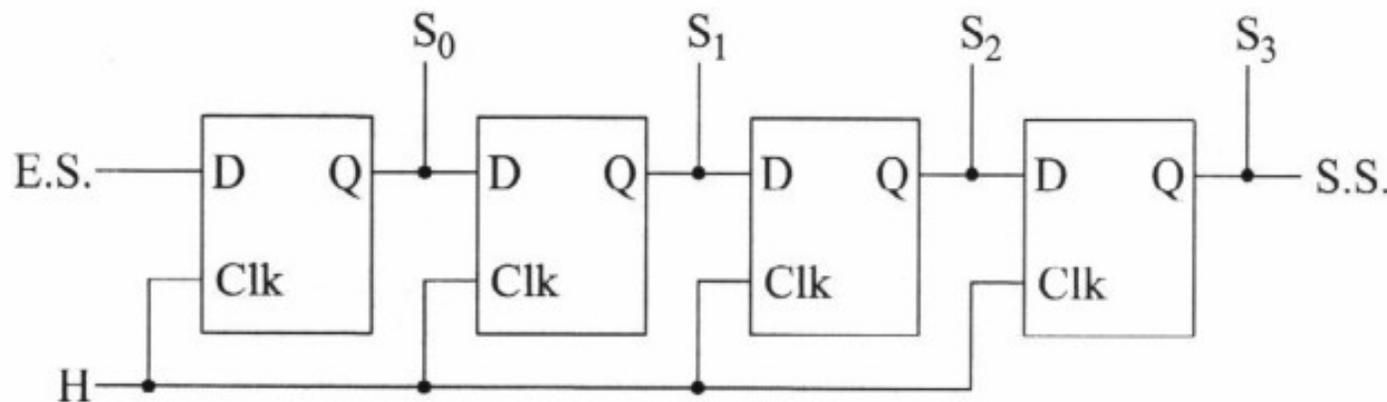
Logique séquentielle

IV. Les Registres

□ Registres a décalage

entrée série – sortie parallèle

- pour transformer un codage temporel (succession des bits dans le temps en codage spatial (information stockée dans une mémoire statique
- on peut utiliser aussi la sortie série, avec éventuellement des fréquences d'horloge différentes en entrée et en sortie
 - le registre sert alors de mémoire tampon (buffer)

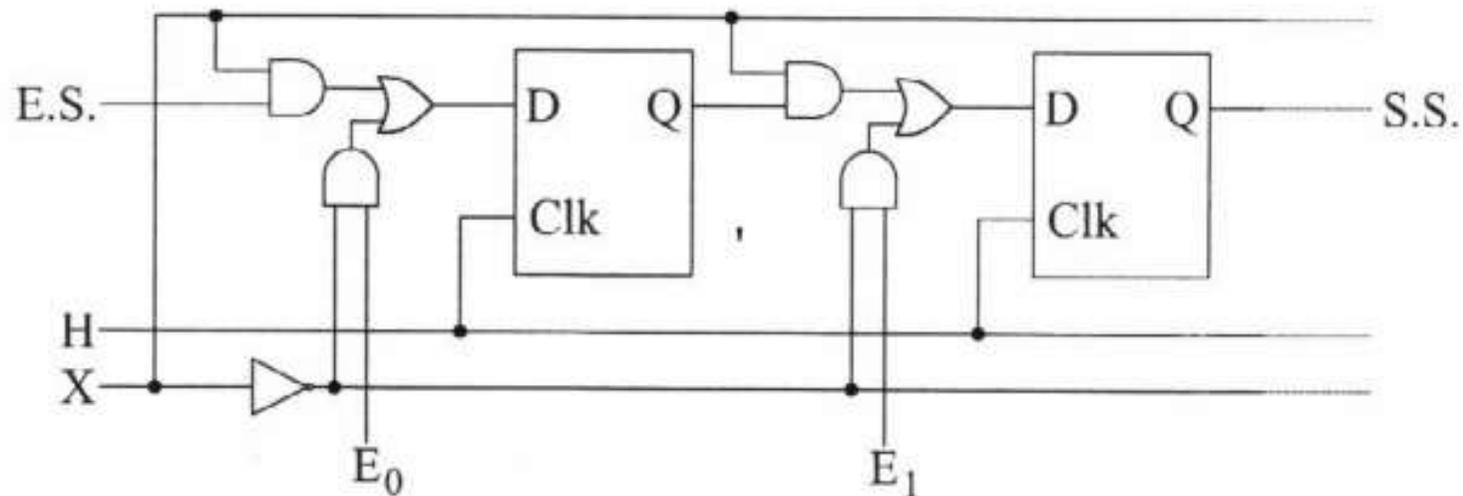


Logique séquentielle

IV. Les Registres

- Registres a décalage
entrée parallèle – sortie série

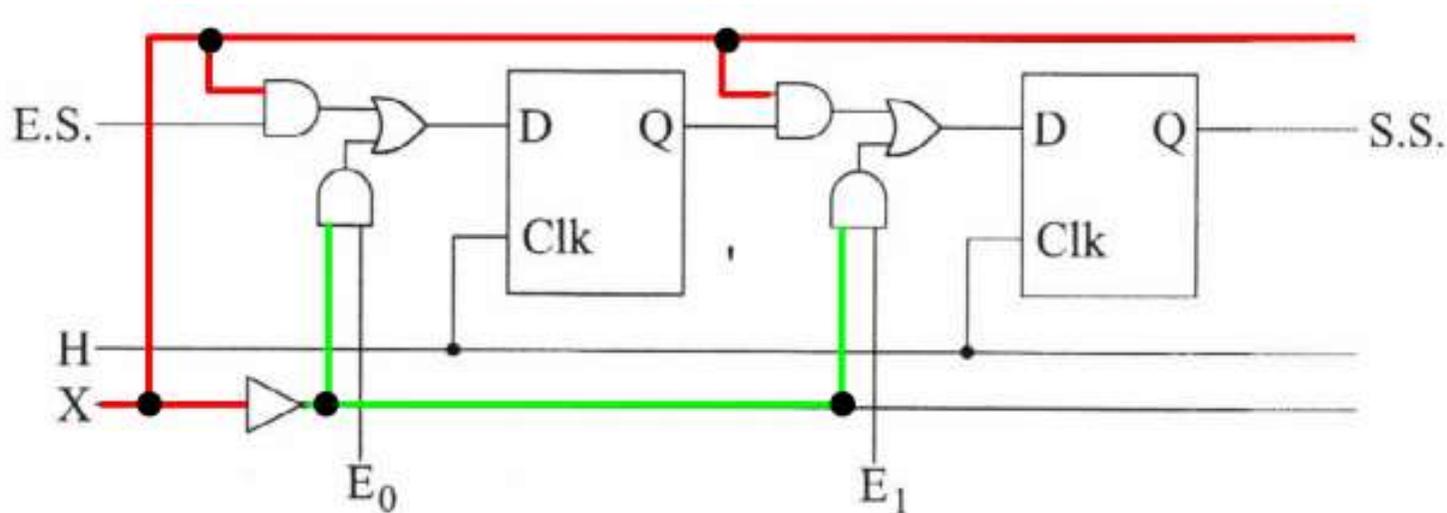
Etudiant le registre suivant



Logique séquentielle

IV. Les Registres

- Registres a décalage
entrée parallèle – sortie série



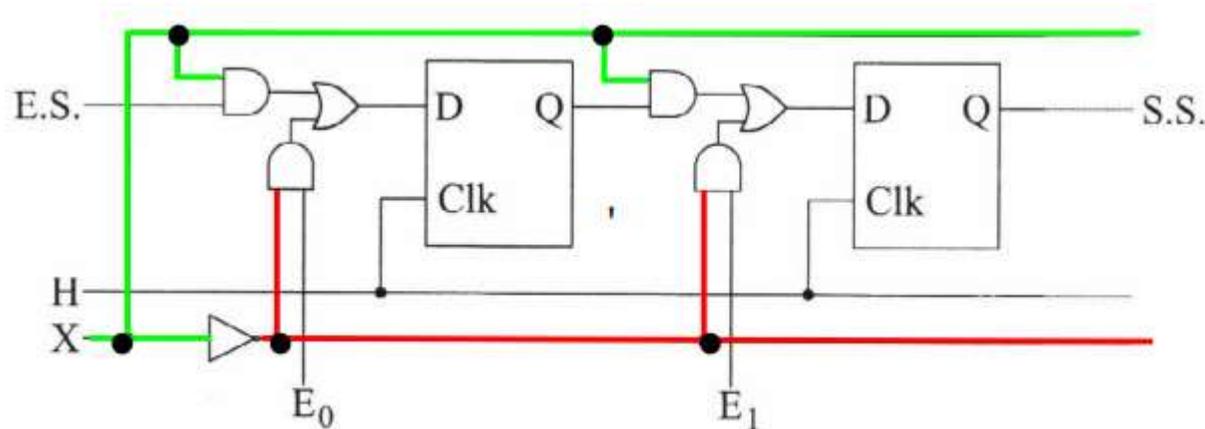
exemple d'une entrée parallèle+série et sortie série

- si $X=0$, l'entrée série est inhibée et l'entrée parallèle est validée

Logique séquentielle

IV. Les Registres

- Registres a décalage
entrée parallèle – sortie série



exemple d'une entrée parallèle+série et sortie série

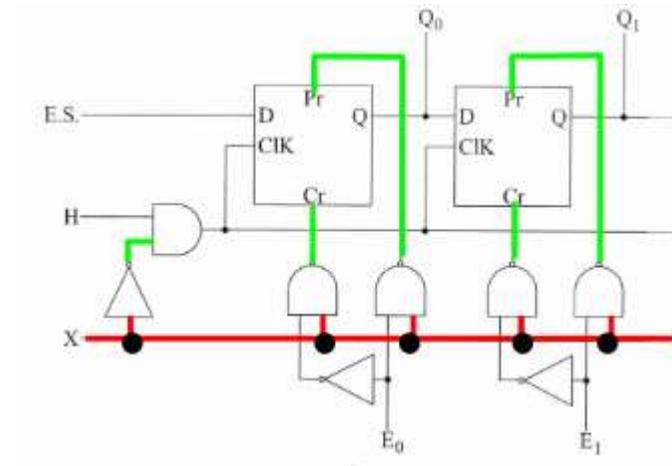
- si $X=1$, l'entrée parallèle est inhibée et l'entrée série est validée
- permet de transformer un codage spatial en codage temporel

Logique séquentielle

IV. Les Registres

- Registres a décalage
entrée parallèle – sortie parallèle

la commande X permet
de sélectionner le mode
de chargement



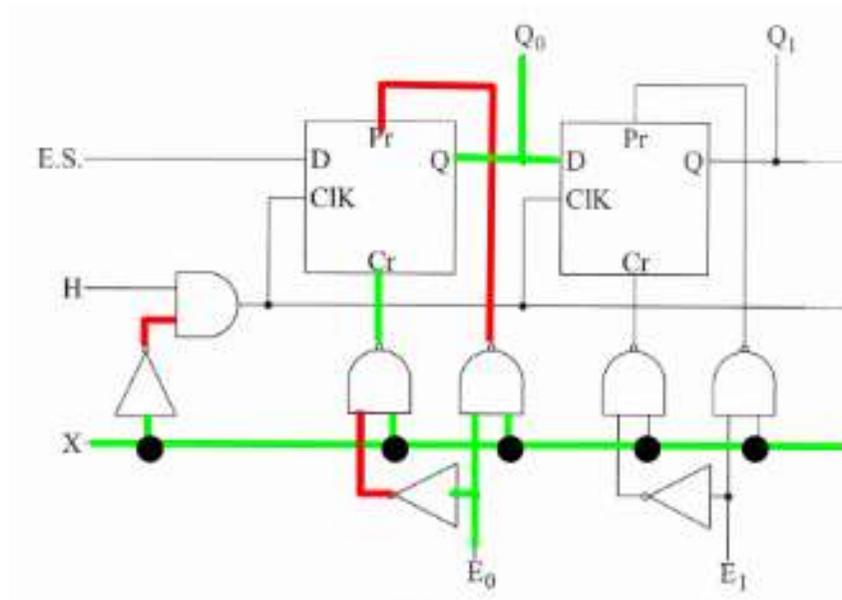
- $X=0$ $Pr=Cr=1$ fonctionnement normal des bascules \Rightarrow inhibition des entrées parallèles

Logique séquentielle

IV. Les Registres

- Registres a décalage
entrée parallèle – sortie parallèle

la commande X permet
de sélectionner le mode
de chargement



- $X=0$ $Pr=Cr=1$ fonctionnement normal des bascules \Rightarrow inhibition des entrées parallèles
- $X=1$: $E_i=1 \Rightarrow (Pr=0, Cr=1) \Rightarrow Q_i=1$

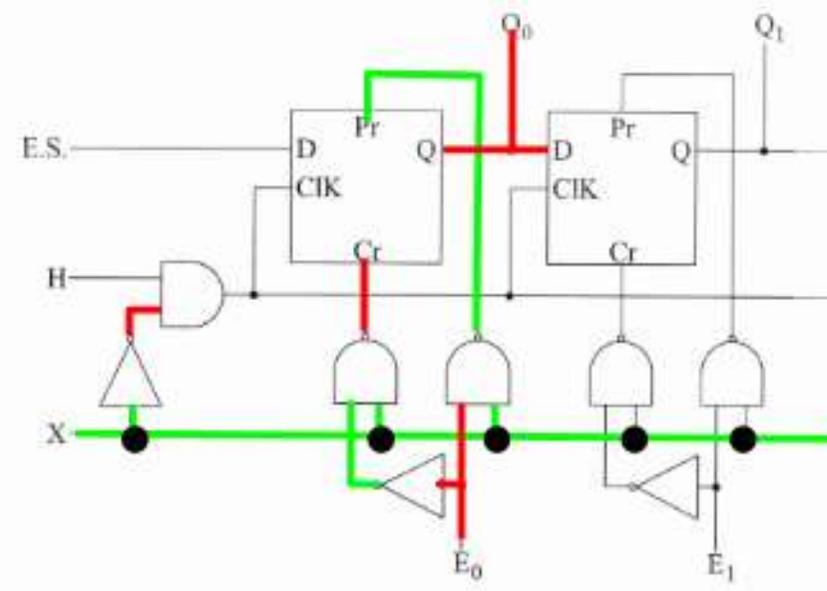
Logique séquentielle

IV. Les Registres

□ Registres a décalage

entrée parallèle – sortie parallèle

la commande X permet de sélectionner le mode de chargement



➤ $X=0$ $Pr=Cr=1$ fonctionnement normal des bascules \Rightarrow inhibition des entrées parallèles

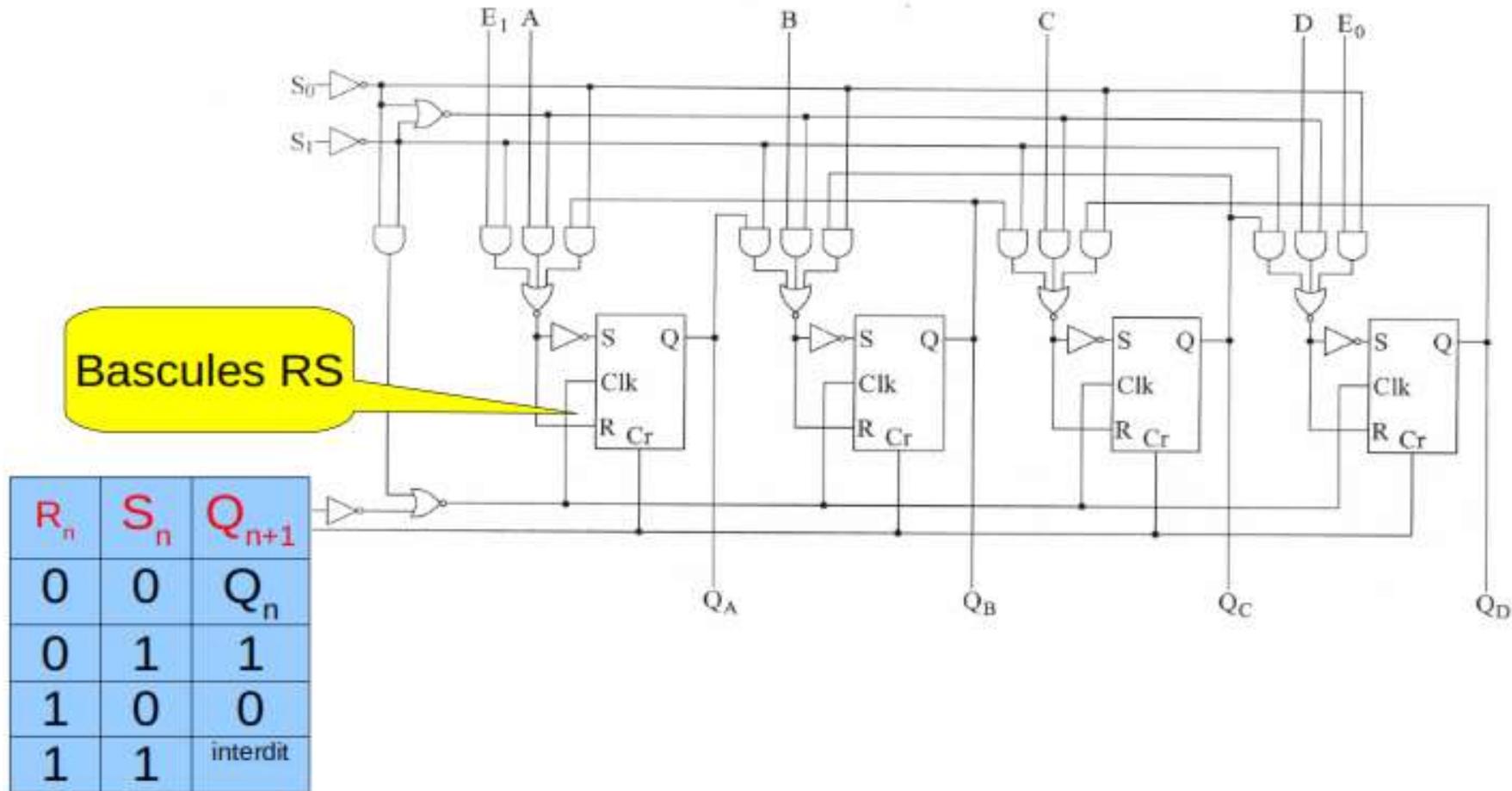
➤ $X=1$:

$\times E_i=1 \Rightarrow (Pr=0, Cr=1) \Rightarrow Q_i=1$	} $Q_i = E_i$
$\times E_i=0 \Rightarrow (Pr=1, Cr=0) \Rightarrow Q_i=0$	

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche

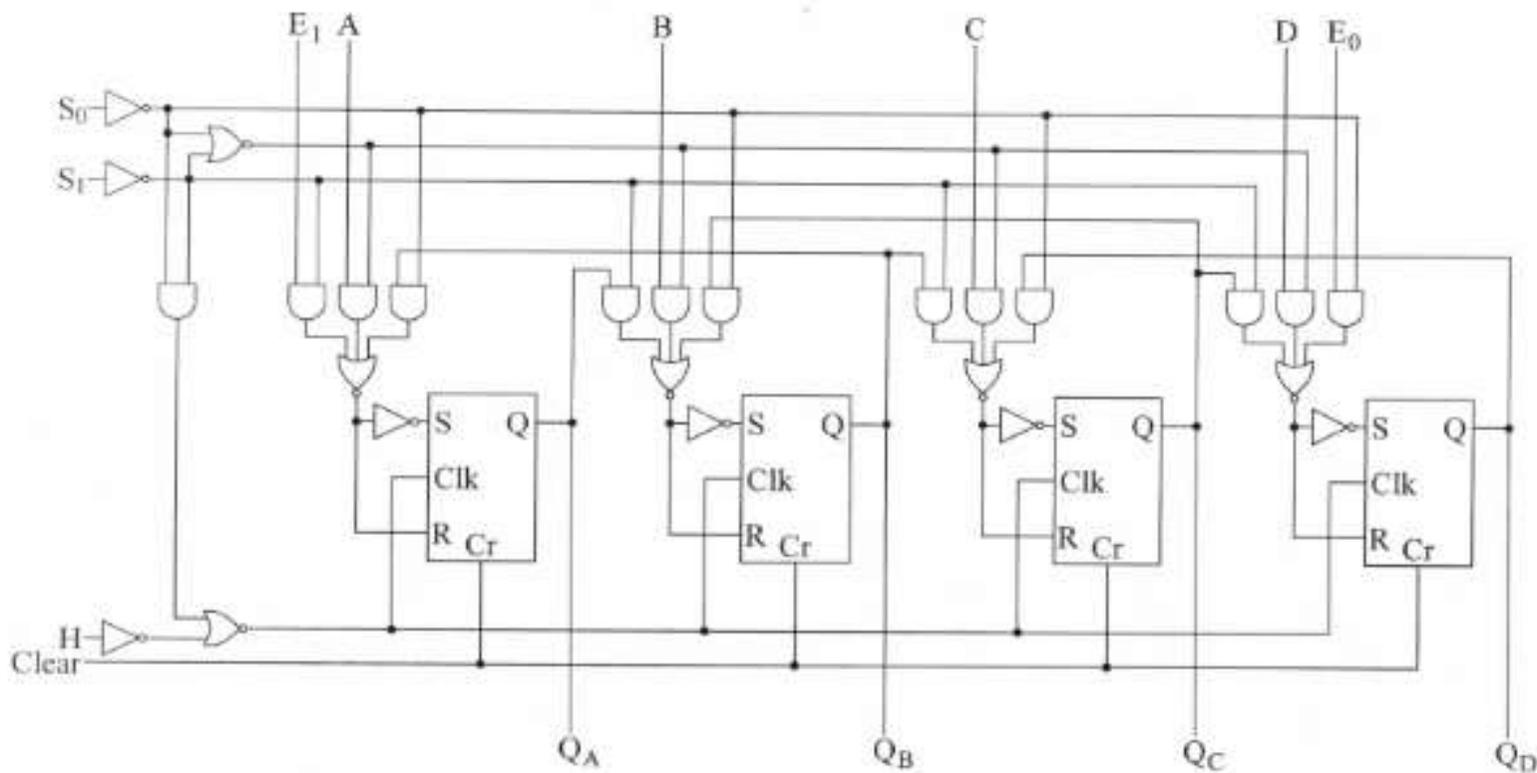


➤ mode de fonctionnement commandé par les entrées S_0 et S_1

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche

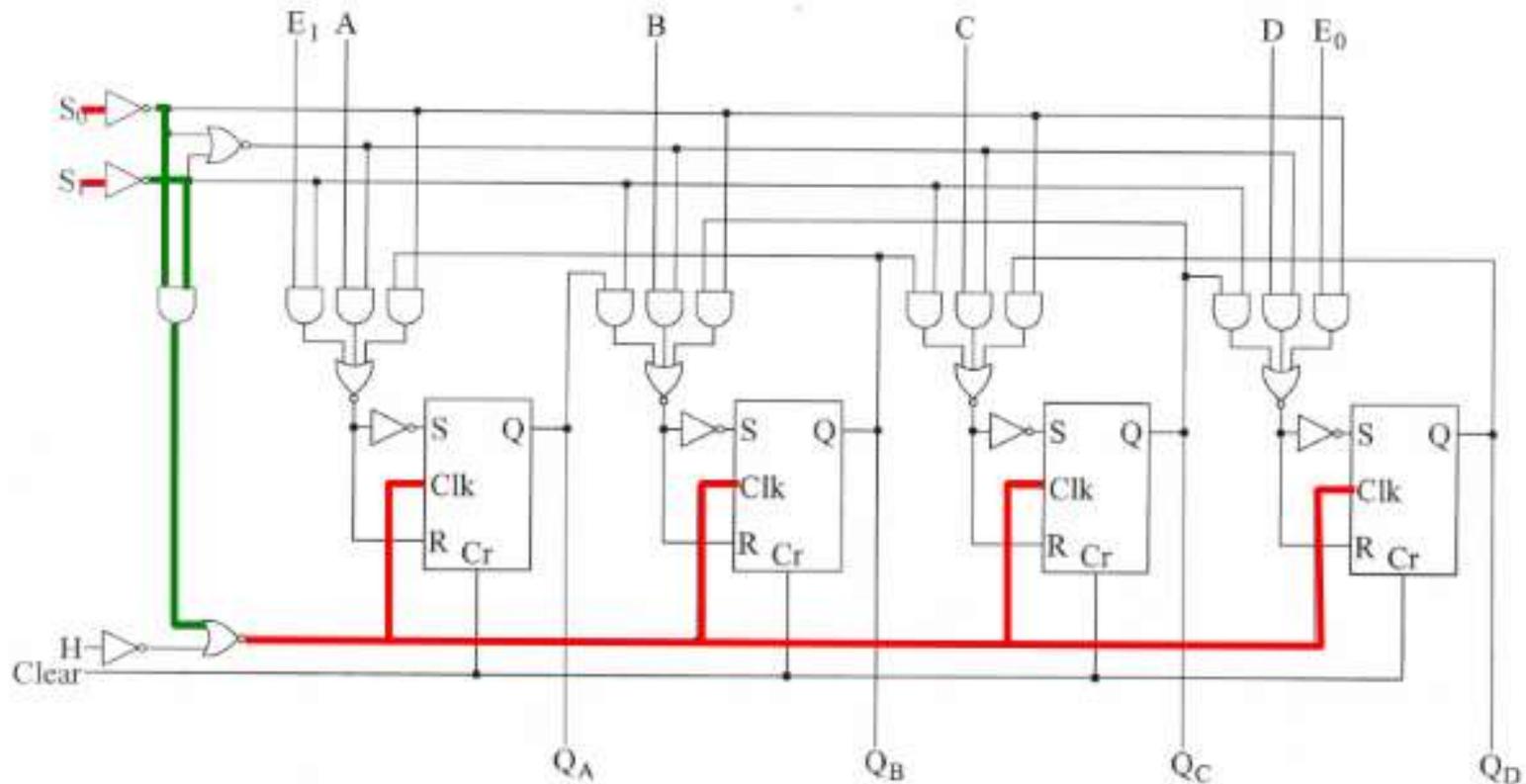


$$\text{Clk} = \overline{H + \overline{S_0} \cdot \overline{S_1}} = H \cdot (S_0 + S_1)$$

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche

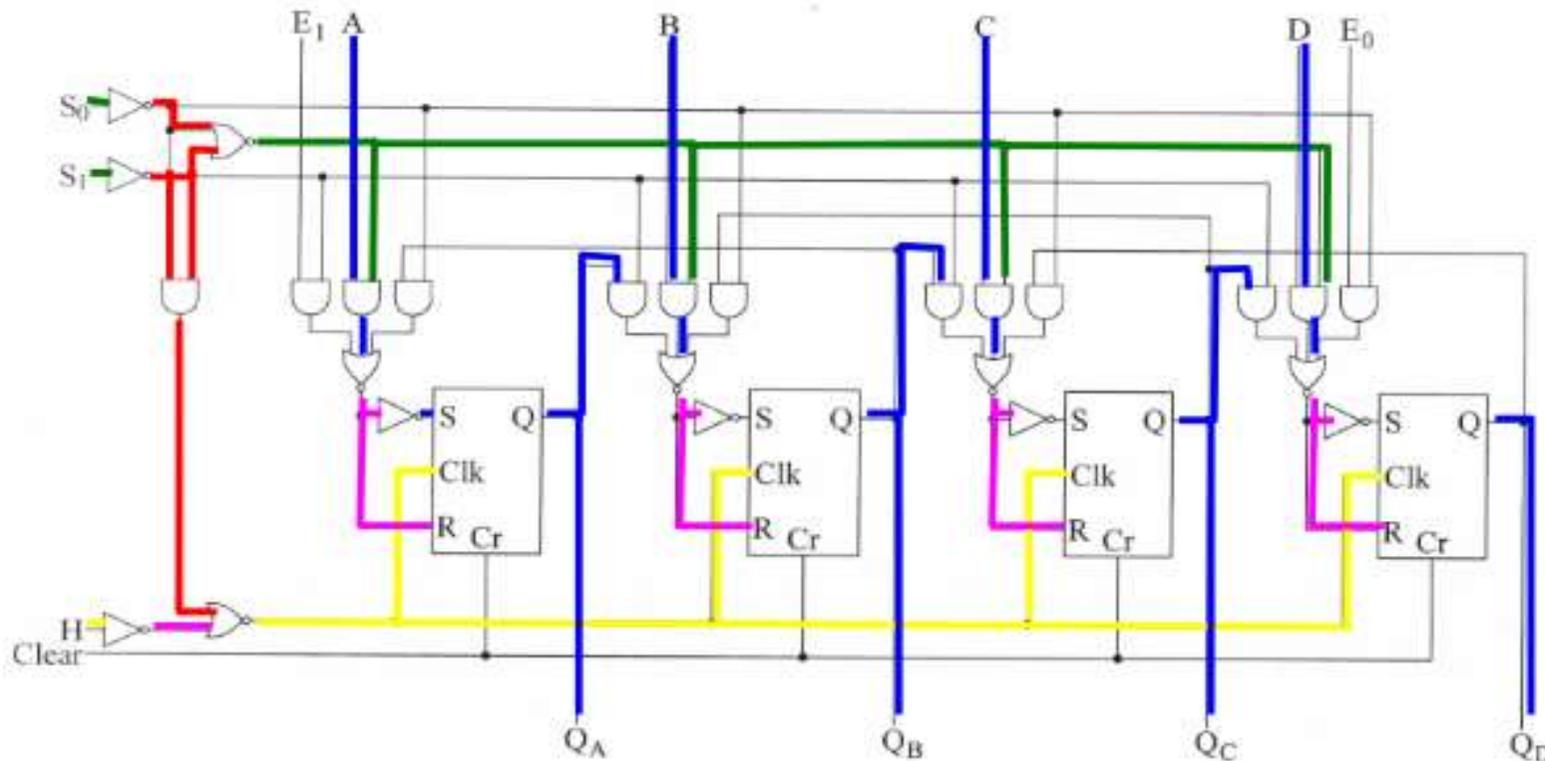


- $Clk = \overline{H + S_0 \cdot S_1} = H \cdot (S_0 + S_1)$
 - ✓ signal d'horloge inhibé si $S_0 = S_1 = 0$

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche

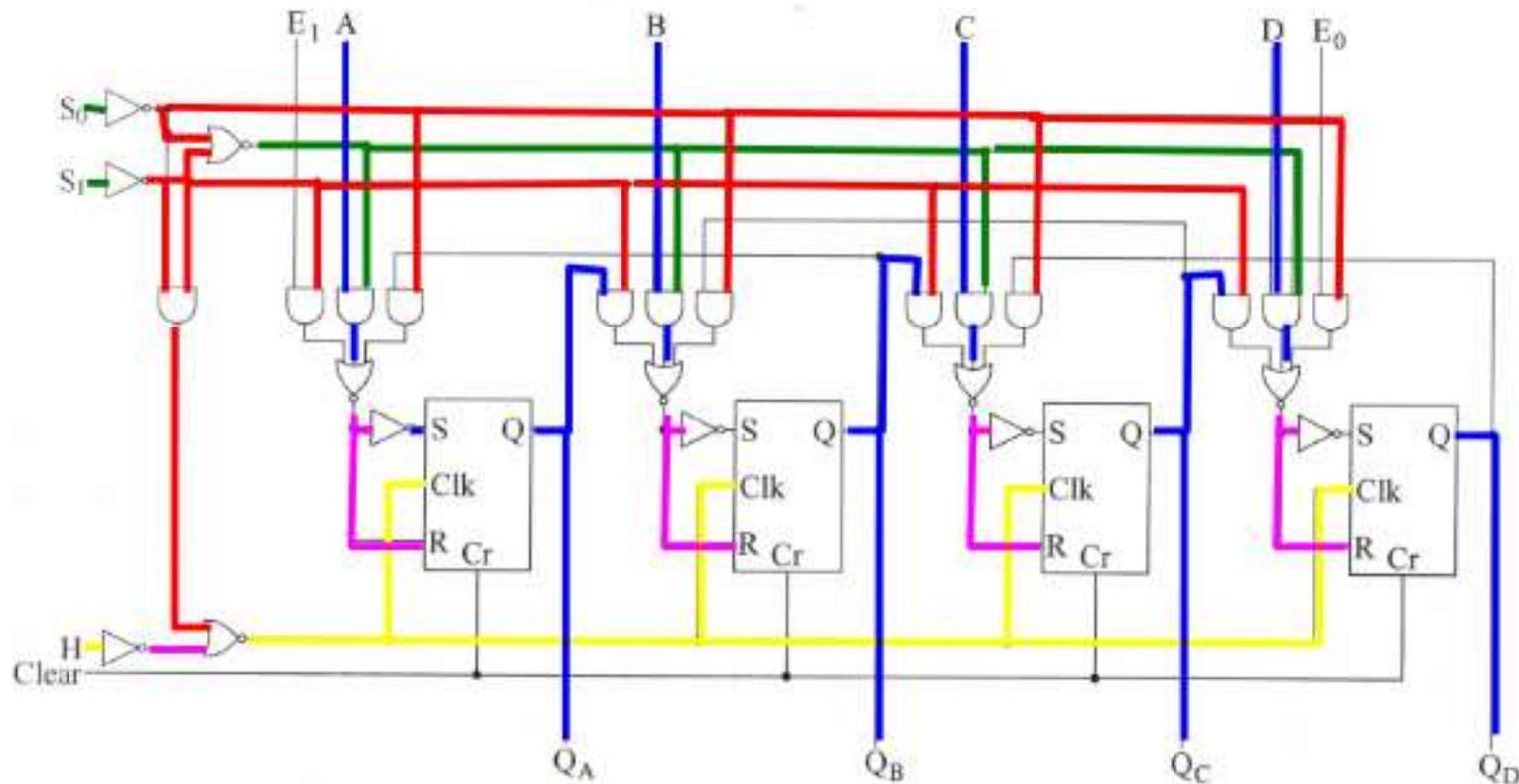


- sélection du chargement parallèle si
 - ✓ $\overline{S_0 + S_1} = S_0 \cdot S_1 \Rightarrow S_0 = S_1 = 1$
lignes d'entrées A, B, C, D validées

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche



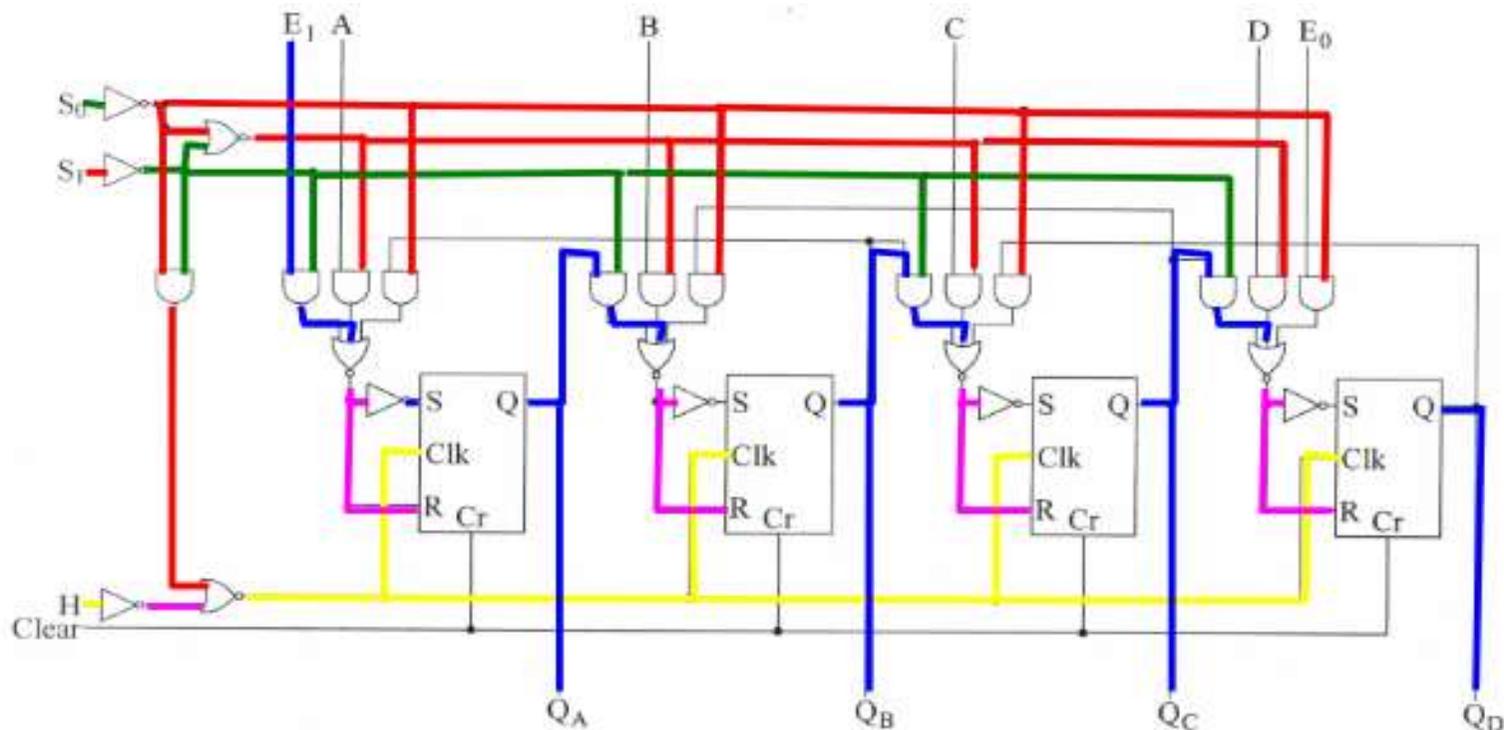
➤ sélection du chargement parallèle si

- ✓ $\overline{S_0} + \overline{S_1} = S_0 \cdot S_1 \Rightarrow S_0 = S_1 = 1$
entrées E_0 et E_1 bloquées, liens Q-RS bloqués

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche

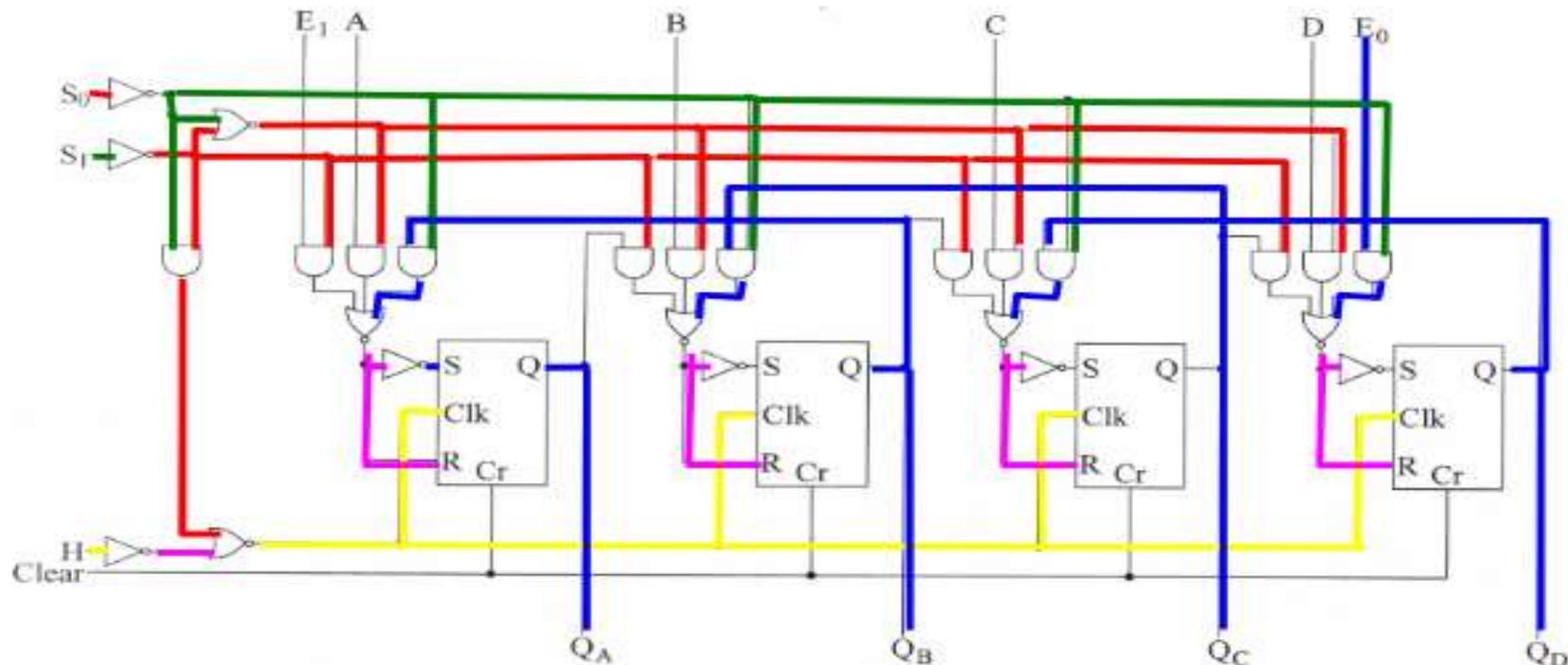


- sélection du décalage à droite (entrée E_1 , sortie Q_D) si
 - ✓ $S_0 = 1$ et $S_1 = 0$
entrée E_0 bloquée, lien $Q - RS$ vers la droite

Logique séquentielle

IV. Les Registres

□ Registres a décalage a droite et a gauche



- sélection du décalage à gauche si
 - ✓ $S_0 = 0$ et $S_1 = 1$
 - entrée E_1 bloquée, lien Q – RS vers la gauche

IV. Les Mémoires

1. Définition et schéma

Une mémoire est un dispositif capable d'emmagasiner puis de restituer une information. L'unité d'information (bit, octet, etc.) s'appelle «point mémoire » ou «cellule»

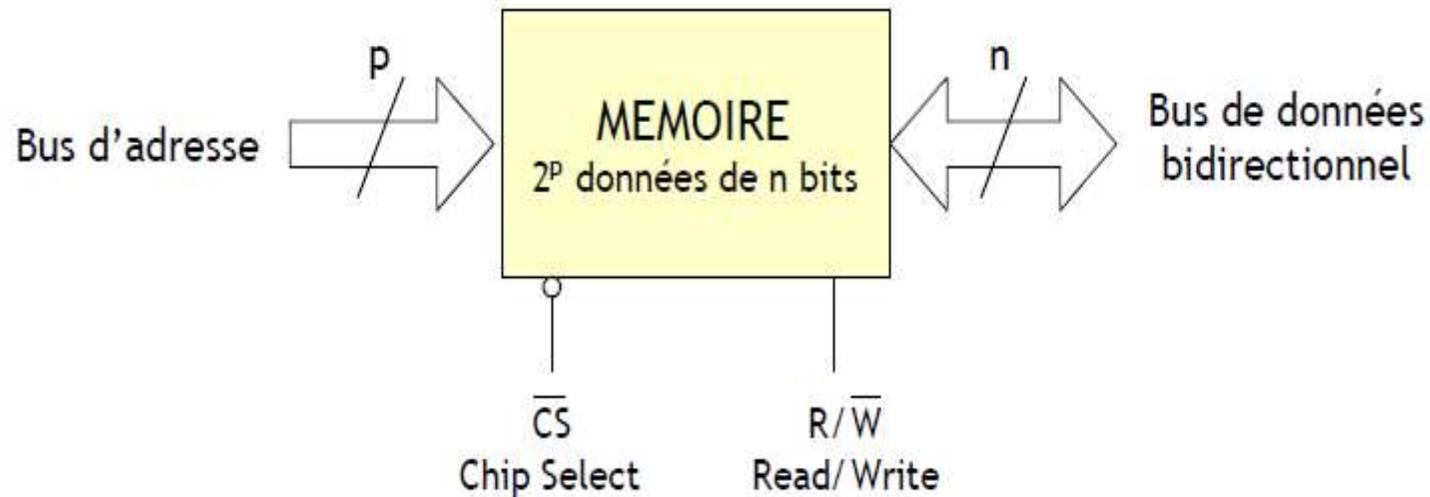


Schéma fonctionnel d'une mémoire

IV. Les Mémoires

- Le Bus d'adresse fournit l'adresse sur p bits qui permet de localiser la donnée
- La donnée de n bits entre (écriture) et sort (lecture) par le bus de données qui est bidirectionnel : deux sens possibles, en liaison avec le signal R/W
- La mémoire peut stocker 2^p données de n bits chacune
- Le signal CS permet la sélection du circuit ou le mettre en haute impédance ; cette possibilité permet, comme on le verra, l'extension de la capacité mémoire d'un système.

IV. Les Mémoires

2. Utilisation des mémoires

Une mémoire peut être utilisée de 2 façons, soit en :

Lecture :

- Appliquer le mot adresse sur le bus d'adresse ;
- Sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne CS ;
- Sélectionner le mode lecture en appliquant un niveau logique haut sur la ligne R/W ;

Écriture :

- Appliquer le mot d'adresse sur le bus d'adresse ;
- Le mot de donnée sur le bus de données ;
- Sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne CS ;
- Sélectionner le mode écriture en appliquant un niveau logique bas sur la ligne R/W ;

IV. Les Mémoires

3. Caractéristiques des mémoires

La capacité : c'est la quantité d'information qui peut être stockée dans la mémoire. Elle s'exprime en bits ou en mots de n bits. Par exemple : 64b, 4Kb, 8Ko (o : octet ou byte) avec 1o = 8b ; 1K = $2^{10} = 1024$; 1M = $2^{20} = 1048576$

L'organisation : elle définit le nombre de mots et la longueur de chaque mot. Par exemple :

- Une mémoire de 64Kx 1 est constituée de 65536 mots de 1 bit. Sa capacité est donc de 64Kb (8Ko) ;
- Une mémoire de 8Kx8 contient 8192 mots de 8 bits. Sa capacité est de 64Kb (8Ko) ;

Le temps d'accès : c'est le temps qui s'écoule entre une demande d'information et le moment où elle est effectivement disponible

IV. Les Mémoires

4. les différents type de mémoires

Il y a deux types de famille de mémoires

- Mémoires mortes (**ROM pour Read Only Memory**) : mémoire à lecture seule;
- Mémoires vives (**RAM pour Random Access Memory**) : mémoire à lecture et écriture.

IV. Les Mémoires

4. les différents type de mémoires

a. Les mémoires mortes

Une mémoire morte (i.e. **ROM**) est une mémoire dont le contenu a été défini et réalisé une bonne fois pour toutes au moment de la fabrication. Une coupure de l'alimentation électrique n'altère pas le contenu.

❖ Les **PROMs** sont des mémoires non volatiles, dont le contenu comme dans le cas des ROMs, est défini une fois pour toutes. Toutefois, contrairement aux ROMs, elles sont programmables (1 seule fois) par l'utilisateur.

IV. Les Mémoires

4. les différents type de mémoires

b. Les mémoires Vives

Dès qu'un système doit conserver temporairement des informations, la RAM trouve sa place. En informatique, elles sont largement mises en œuvre en quantités importantes.

c. Les mémoires programmables et effaçables par l'utilisateur

Les mémoires programmables sont intermédiaires entre les RAM et les ROM. Leur contenu peut être défini par l'utilisateur

- Les **PROM (Programmable ROM)** : sont composées de fusibles que l'on peut **détruire une** seule fois ;
- Les **EPROM (Erasable PROM)** : ce sont des mémoires effaçables par ultraviolet et programmables électriquement ;
- Les **EEPROM (Electrical Erasable PROM)** : ce sont des mémoires effaçables et programmables électriquement .

IV. Les Mémoires

5. Les extensions des mémoires

Il est parfois utile de grouper plusieurs circuits pour augmenter la capacité, on utilise le même principe avec les mémoires pour augmenter leur capacité (nombre des mots et/ou longueur des mots).

Par exemple, à l'aide de 4 boîtiers mémoires de 1Kx4bits, on peut réaliser les mémoires suivantes : 1Kx16bits, 4Kx4bits, 2Kx8bits.

Exercice: réaliser une mémoire de 1Kx16bits à partir d'une mémoire élémentaire de 1Kx4bits

Logique séquentielle

IV. Les Mémoires

5. Les extensions des mémoires

Solution: on aura besoin de:

- Nécessité de 4 boîtiers ;
- Nécessité de 10 bits d'adresses A0 à A9 ; (nombre de mots = 2nombre de bits d'adresse)
- Nécessité de 16 bits de données D0 à D15 ;

