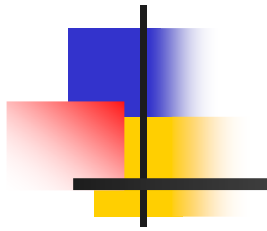
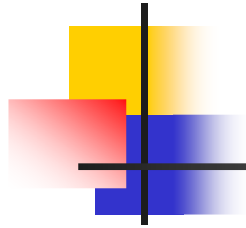


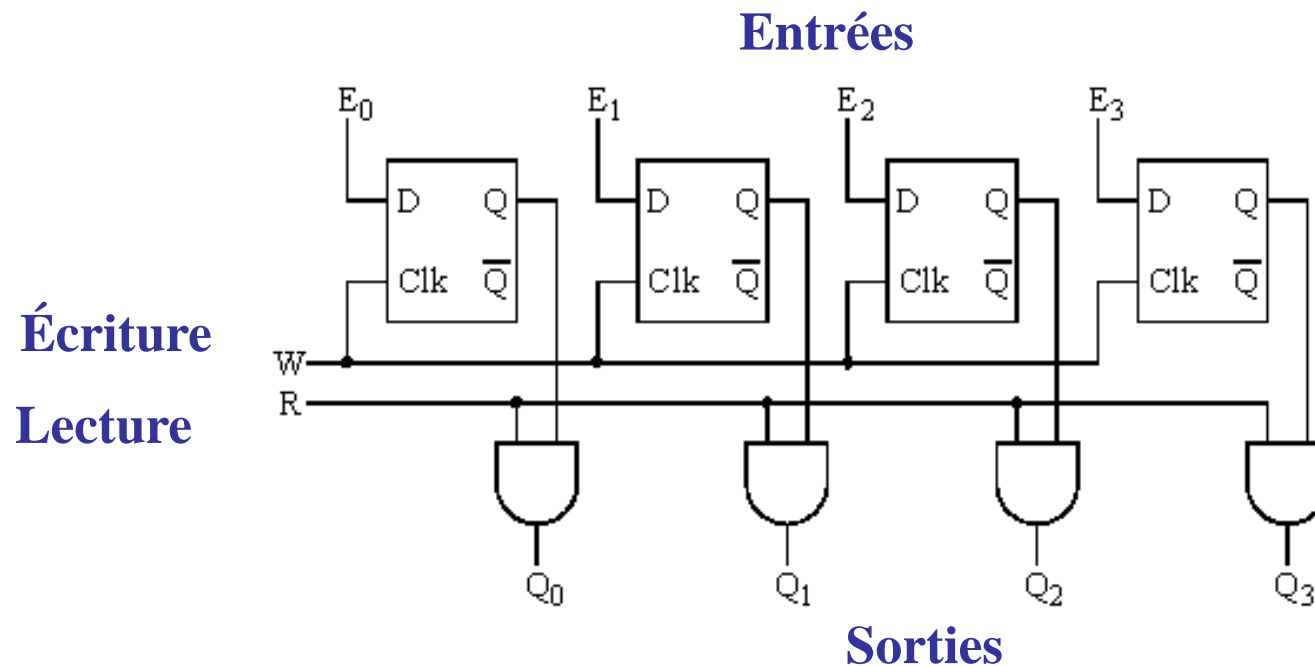
# Les circuits logiques séquentiels





# Les registres de mémorisation

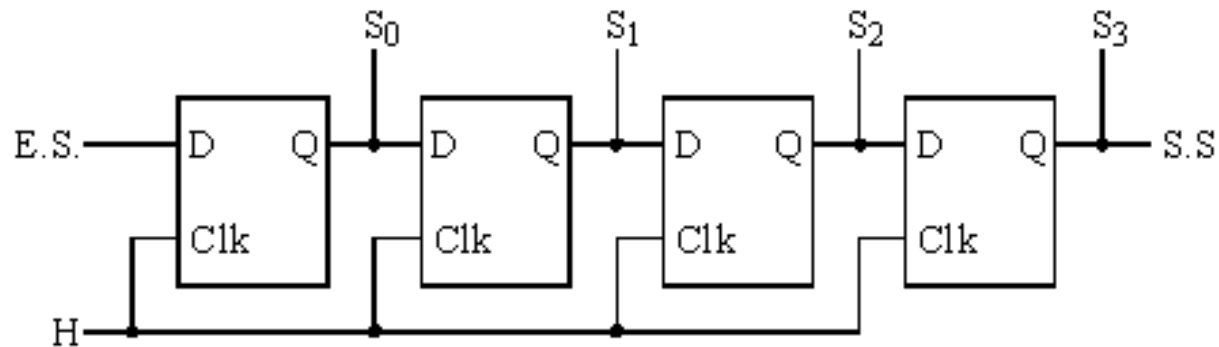
- Un registre permet la mémorisation de  $n$  bits. Il est donc constitué de  $n$  bascules, mémorisant chacune un bit. L'information est emmagasinée sur un signal de commande et ensuite conservée et disponible en lecture.



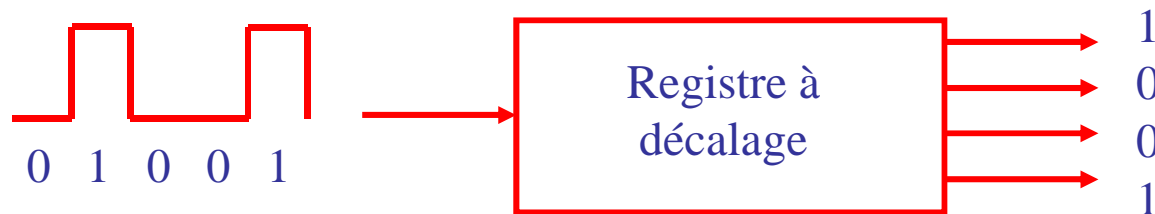
Registre de 4 bits commandé par les entrées W et R

# Les registres à décalage

**Entrée série - Sortie parallèle et série :**

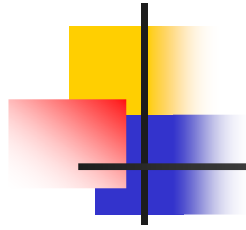


Ce type de registre permet de transformer un codage temporel en un codage spatial.



La sortie série peut également être utilisée. L'intérêt d'utilisation d'un registre à décalage en chargement et lecture série réside dans la possibilité d'avoir des fréquences d'horloge différentes au chargement et à la lecture. Le registre constitue alors un tampon.





# Les registres à décalage

Les diverses possibilités sont sélectionnées par les lignes commande  $S_0$  et  $S_1$ .  
Considérons la ligne transportant le signal d'horloge aux bascules, elle est gouvernée par l'expression logique :

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

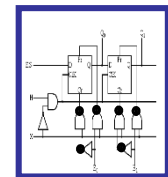
Le signal d'horloge sera donc inhibé si  $S_0 + S_1 = 0$

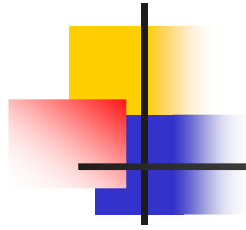
Pour sélectionner le chargement parallèle (entrées A, B, C et D) il faut :

$$\overline{\overline{S_0 + S_1}} = \overline{\overline{S_0} \cdot \overline{S_1}} = 1$$



$S_0 = S_1 = 1$ . Le chargement parallèle se fait sur un front actif d'horloge.



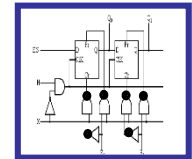


# Les registres à décalage

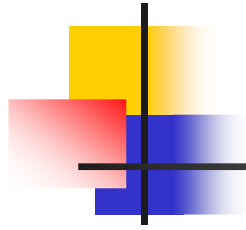
Pour sélectionner le décalage à droite (entrée  $E_1$ , sortie  $Q_D$ ) :  $S_0 = 1$  et  $S_1 = 0$

Pour le décalage à gauche (entrée  $E_0$ , sortie  $Q_A$ ) :  $S_0 = 0$  et  $S_1 = 1$

$S_0$	$S_1$	Fonction
0	0	Registre bloqué
0	1	Décalage à gauche
1	0	Décalage à droite
1	1	Chargement parallèle



Rq : un registre à décalage à droite et à gauche permet d'effectuer des multiplications et des divisions entières par des puissances de 2.



# Les compteurs

---

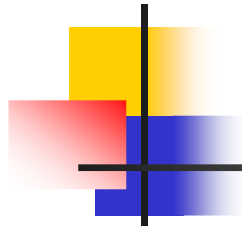
Un compteur est un ensemble de  $n$  bascules interconnectées par des portes logiques. Ils peuvent donc mémoriser des mots de  $n$  bits. Au rythme d'une horloge, ils peuvent décrire une séquence déterminée c'est-à-dire occuper une suite d'états binaires et peut représenter au maximum  $2^n$  combinaisons. Ces états restent stables et accessibles entre les impulsions d'horloge. Le nombre total  $N$  des combinaisons successives est appelé le modulo du compteur.

On a  $N \leq 2^n$ . Si  $N < 2^n$  un certain nombre d'états sont hors du cycle nominal de fonctionnement.

Les compteurs binaires peuvent être classés en deux catégories :

- les compteurs asynchrones;
- les compteurs synchrones.

De plus on distingue les compteurs réversibles ou compteurs-décompteurs.

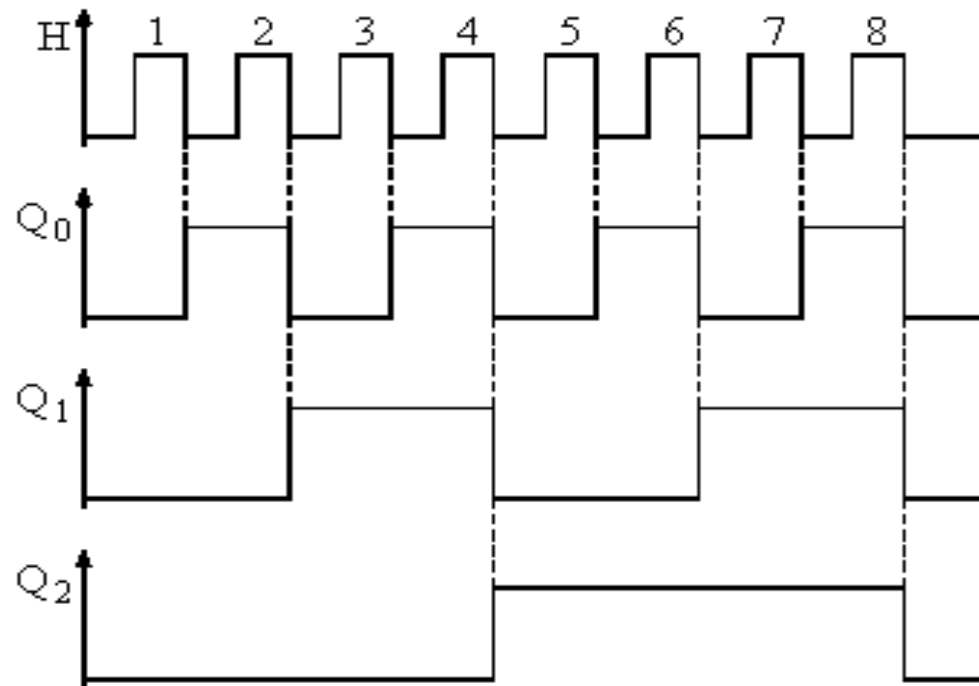


# Les compteurs synchrones

Soit un compteur modulo 8 comptant en binaire naturel, nous choisissons un codage permettant de réduire au maximum le CCS :

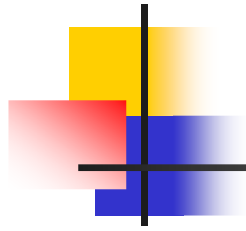


Impulsion	$S_2$	$S_1$	$S_0$
état initial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0



les sorties  $Q_0$ ,  $Q_1$  et  $Q_2$  fournissent des signaux périodiques de fréquences respectivement 2, 4 et 8 plus faibles que celle de H. La division de fréquence est une des applications des compteurs.





# Les compteurs synchrones

En utilisant des bascules JK comme des bascules T (J = K)

# Top	$Q_2$	$Q_1$	$Q_0$	$K_2=J_2$	$K_1=J_1$	$K_0=J_0$
0	0	0	0	0	0	1
1	0	0	1	0	1	1
2	0	1	0	0	0	1
3	0	1	1	1	1	1
4	1	0	0	0	0	1
5	1	0	1	0	1	1
6	1	1	0	0	0	1
7	1	1	1	1	1	1
8 = 0	0	0	0	-	-	-

$$K_0=J_0= 1$$

$$K_1=J_1= Q_0$$

$$K_2=J_2= Q_1Q_0$$

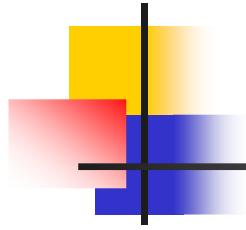


En généralisant :

$$K_i = J_i = Q_{i-1} \dots Q_0$$

Ou

$$K_i = J_i = J_{i-1} Q_{i-1} = K_{i-1} Q_{i-1}$$



# Les décompteurs synchrones

En utilisant des bascules JK comme des bascules T

# Top	$Q_2$	$Q_1$	$Q_0$	$K_2=J_2$	$K_1=J_1$	$K_0=J_0$
0	1	1	1	0	0	1
1	1	1	0	0	1	1
2	1	0	1	0	0	1
3	1	0	0	1	1	1
4	0	1	1	0	0	1
5	0	1	0	0	1	1
6	0	0	1	0	0	1
7	0	0	0	1	1	1
8 = 0	1	1	1	-	-	-

$$K_0=J_0= 1$$

$$K_1=J_1= \overline{Q_0}$$

$$K_2=J_2= \overline{Q_1}\overline{Q_0}$$



En généralisant :

$$K_i = J_i = \overline{Q_{i-1}} \dots \overline{Q_0}$$

$$\text{Ou} \\ K_i = J_i = J_{i-1} \overline{Q_{i-1}} = K_{i-1} \overline{Q_{i-1}}$$



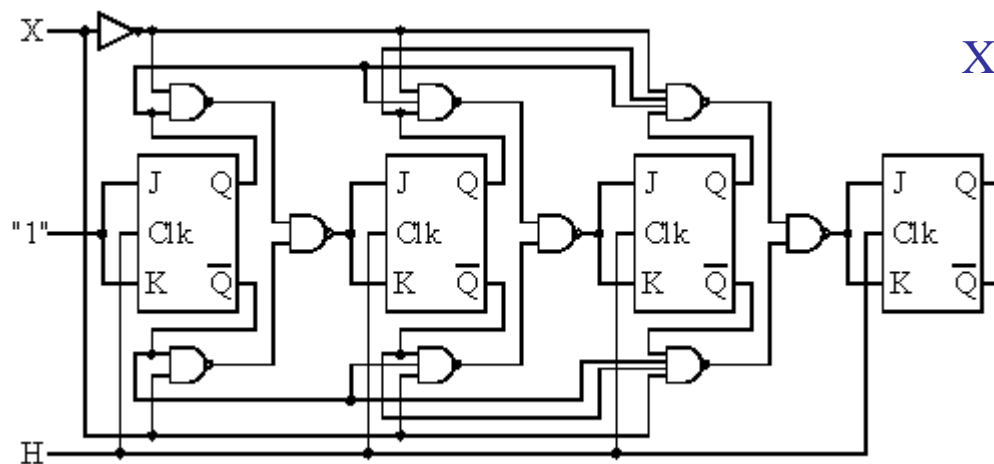
# Les compteurs synchrones

Aux deux manières d'exprimer les relations de récurrence des équations de commutation correspondent deux types de circuits.

- $K_i = J_i = Q_{i-1} \dots Q_0$  est dit à report parallèle
- $K_i = J_i = J_{i-1} Q_{i-1}$  est dit à report série



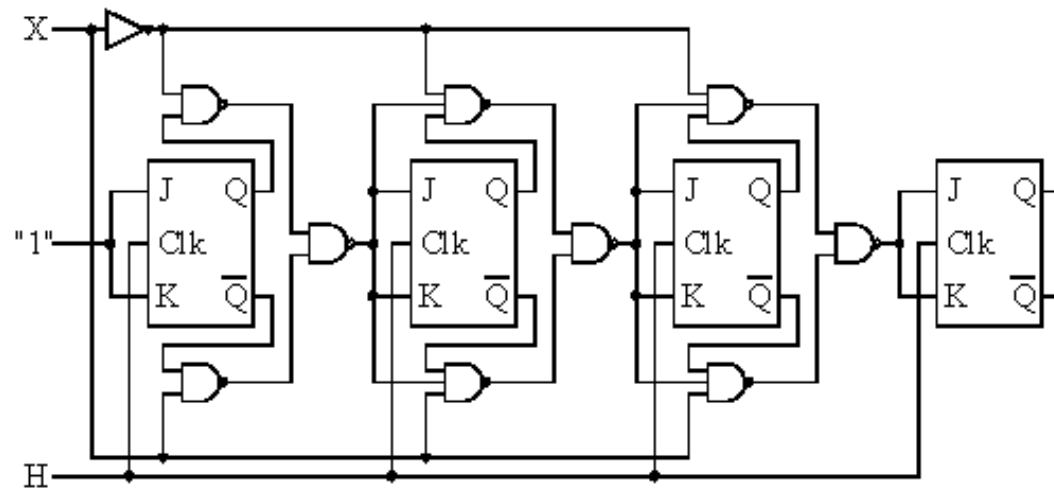
évite ainsi des portes à multiples entrées  
mais limite la fréquence de l'horloge !!!!



$X = 0 \Rightarrow$  Compteur,  $X = 1 \Rightarrow$  Décompteur

Compteur Décompteur  
à report parallèle

# Les compteurs synchrones



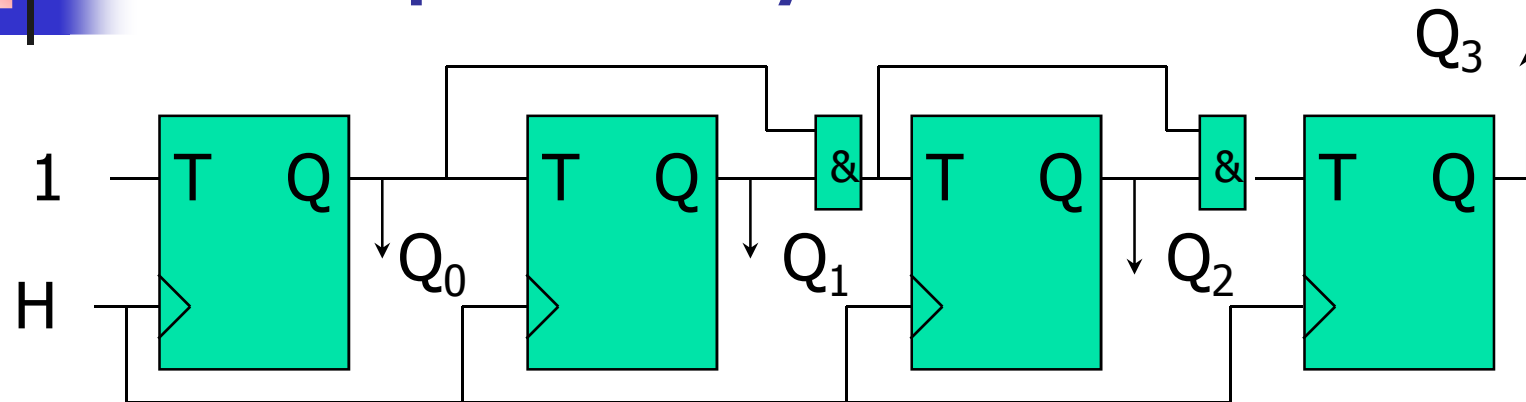
Compteur Décompteur  
à report série

Si le temps de propagation d'une porte NAND :  $\text{MAX}(T_{\text{pHL}}, T_{\text{pLH}}) = \delta$  s (15 ns pour TTL)

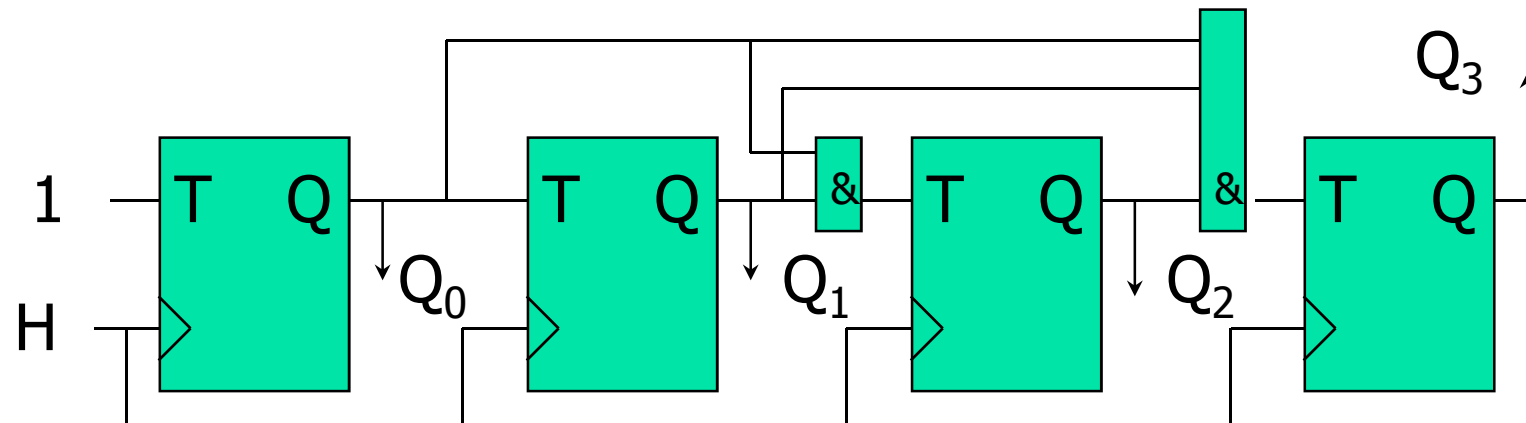


$$T_H \gg 6\delta$$

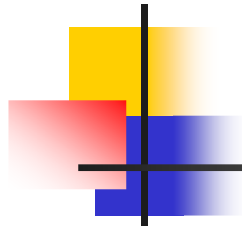
# Compteur synchrone



Compteur synchrone à report série :  $T_{\min}(H) > T_p(\text{bascule}) + (n-2) * T_p(\text{porte ET})$



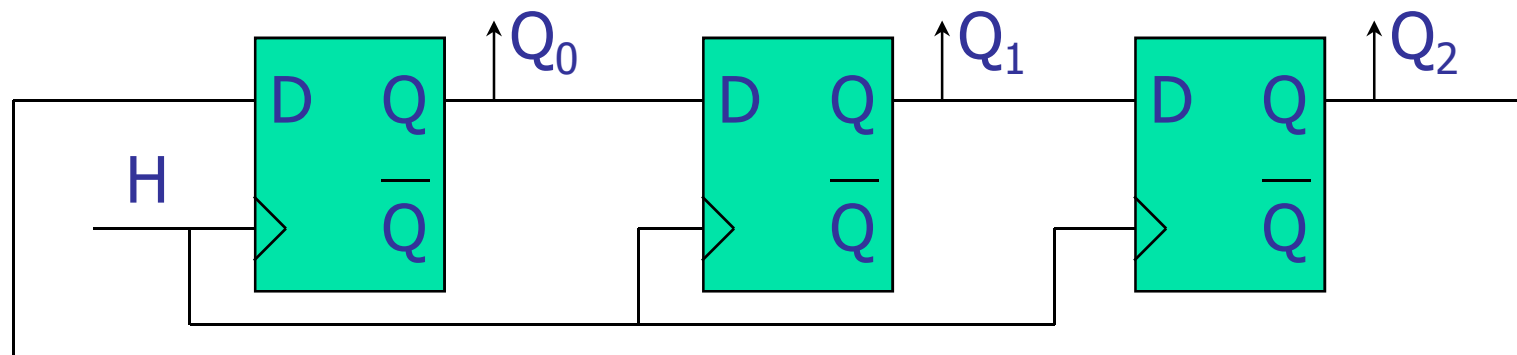
Compteur synchrone à report parallèle :  $T_{\min}(H) > T_p(\text{bascule}) + T_p(\text{porte ET})$



# Compteur synchrone

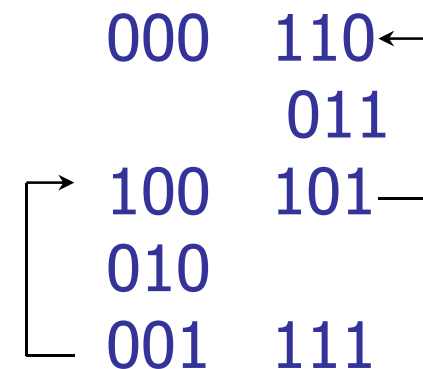
Compteur en anneau : rapide (pas de porte)

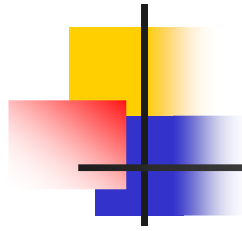
Registre à décalage bouclé



N bascules : plusieurs cycles de comptage

Beaucoup de cycles piège : obligation d'ajouter une logique pour les casser (cf synthèse synchrone) : moins rapide

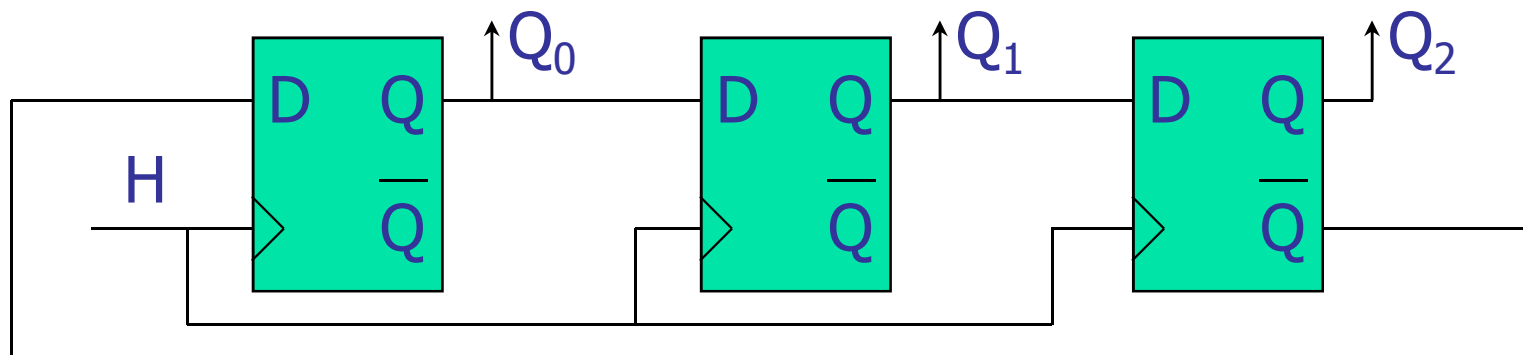




# Compteur synchrone

Compteur de Johnson : Le plus rapide (pas de porte)

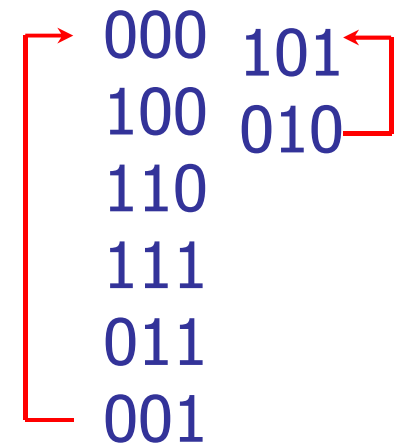
Registres à décalage bouclés par inversion

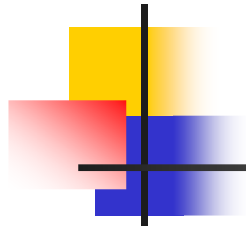


N bascules : 2 cycles de comptage

Un nominal de longueur  $2N-2$  états

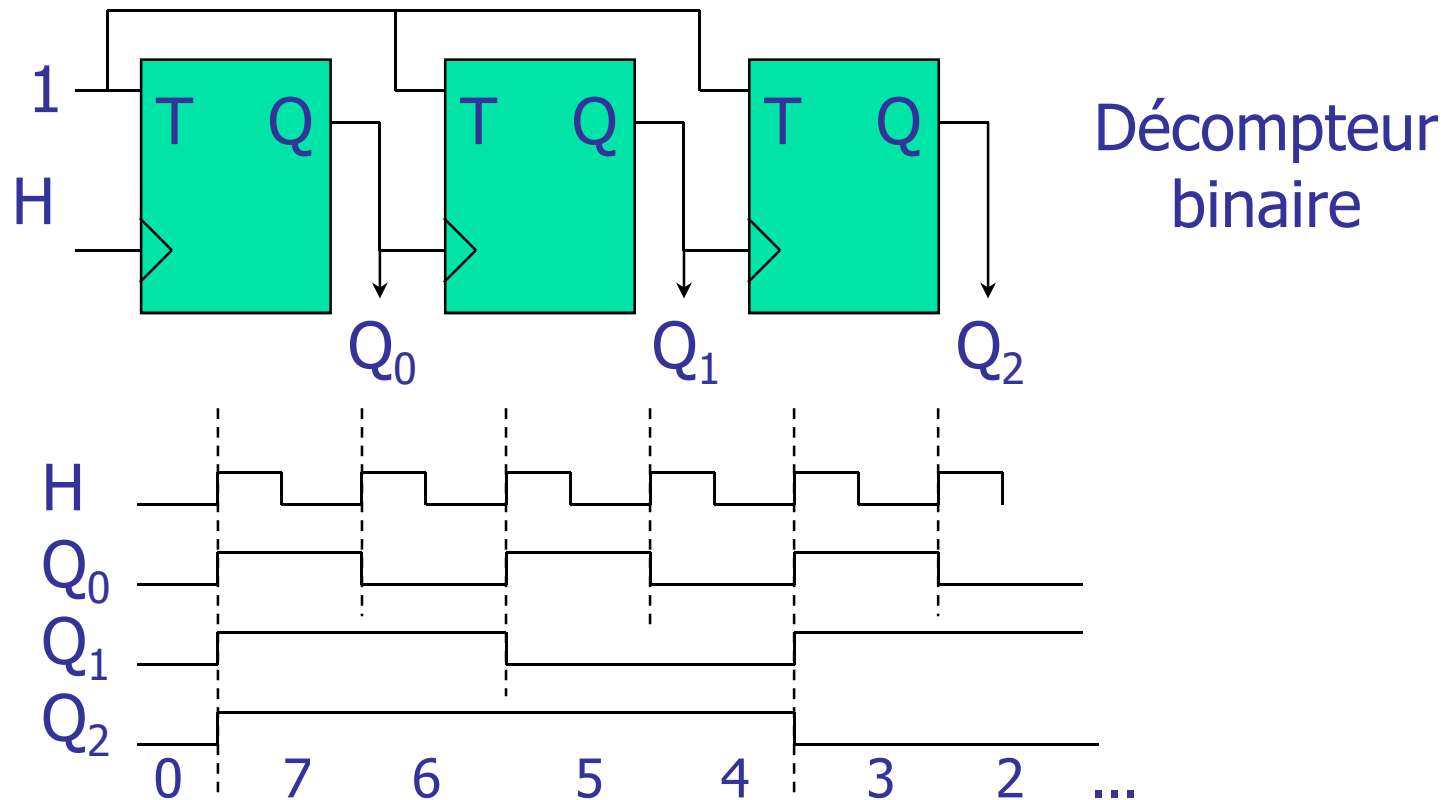
Un cycle piège de 2 états





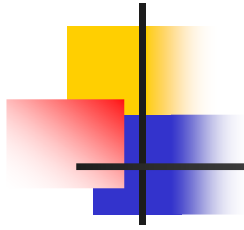
# Compteur chaîné asynchrone (principe)

Cascade de diviseurs de fréquence par deux sur fronts montants. Les bascules n'ont plus la même horloge



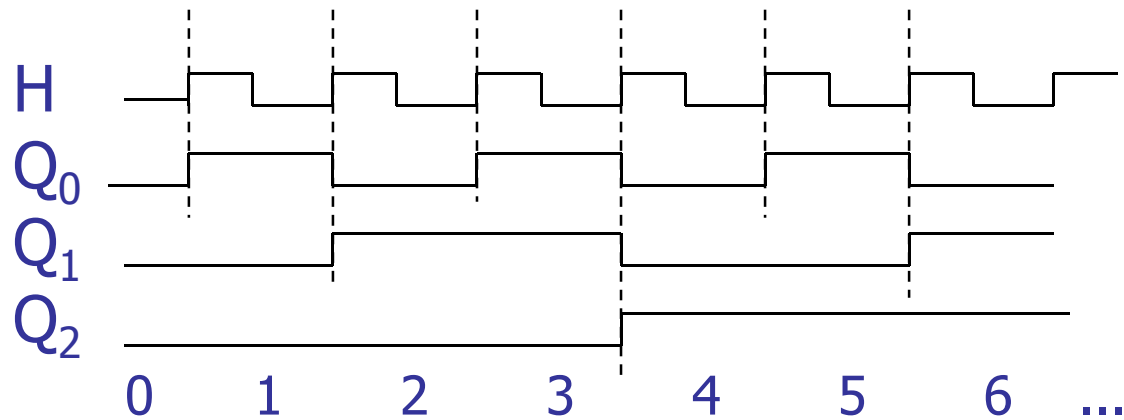
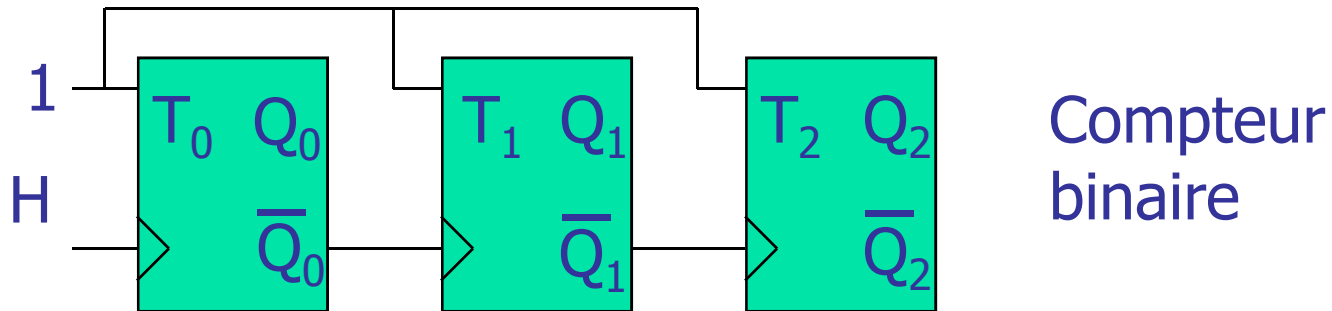
Structure à éviter car asynchrone



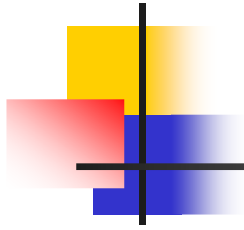


# Compteur chaîné asynchrone (principe)

Cascade de diviseurs de fréquence sur fronts descendants



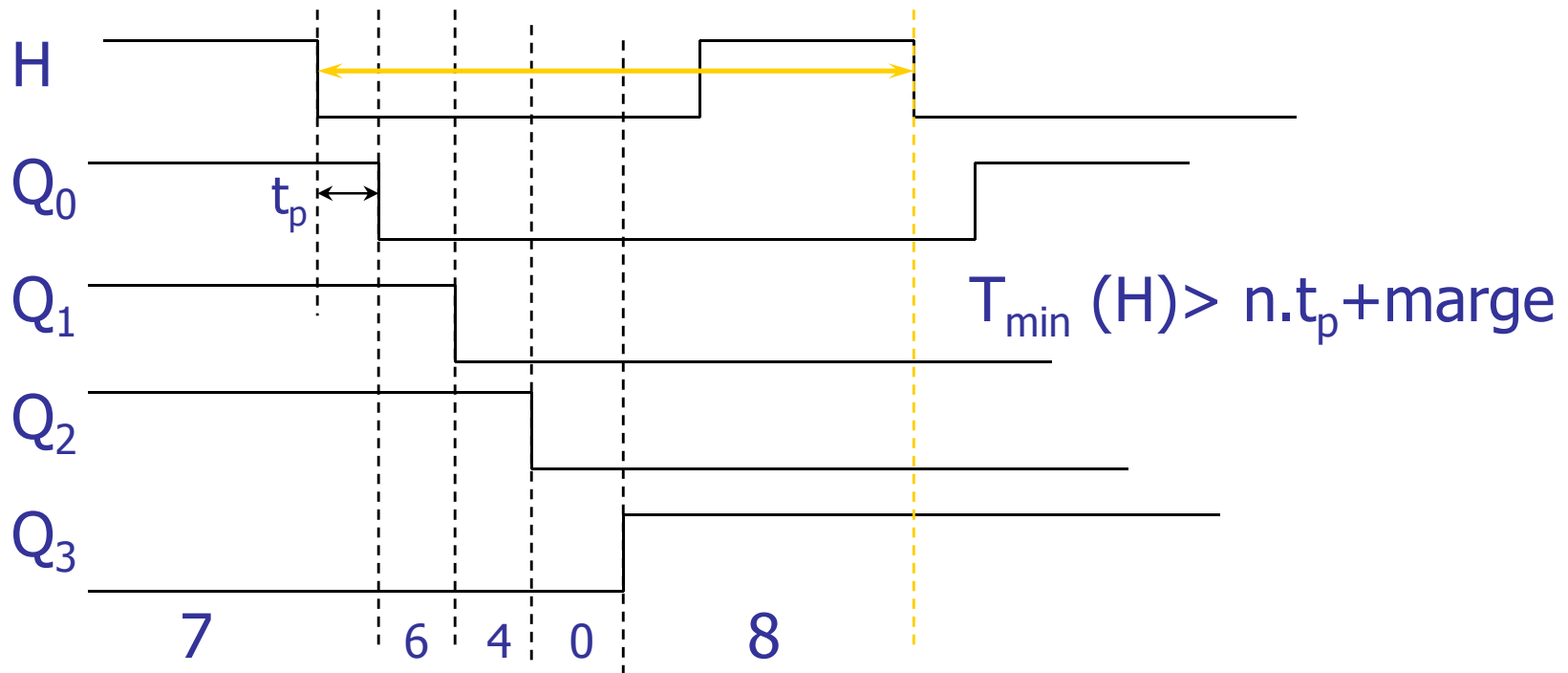
Structure à éviter car asynchrone



# Compteur chaîné asynchrone (principe)

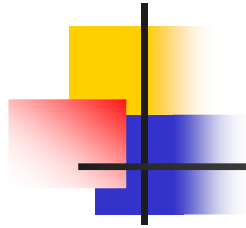
Accumulation des temps de propagation : baisse de la fréquence

Exemple : passage de 7 à 8



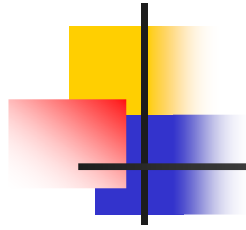
Si  $n$  bascules changent d'état :  $n-1$  états transitoires, par contre les excitations des bascules sont synchronisées





## Les compteurs synchrones (modulo N)

	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
Bouclage	0	0	0	0



# Les compteurs synchrones (modulo N)

Remise à l'état initial (0000) pour tous les états hors cycle par une action sur le reset des bascules : **Synthèse risque minimal**

$$(10)_{(10)} = (1010)_{(2)}$$

$$(11)_{(10)} = (1011)_{(2)}$$

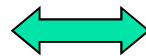
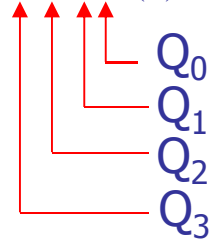
$$(12)_{(10)} = (1100)_{(2)}$$

$$(13)_{(10)} = (1101)_{(2)}$$

$$(14)_{(10)} = (1110)_{(2)}$$

$$(15)_{(10)} = (1111)_{(2)}$$

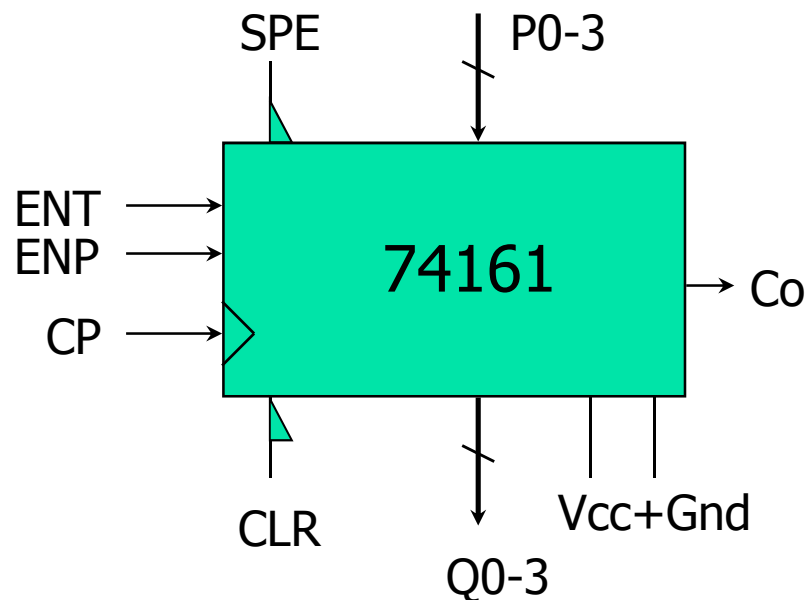
Etats hors cycle



$$\text{RESET} = Q_3(Q_2 + Q_1)$$

# Compteur synchrone

74161 compteur intégré synchrone 4 bits binaire  
programmable cascadable de la famille TTL (16 broches (pins))



CP : horloge (front montant)

$Q_{0-3}$  : sorties du compteur

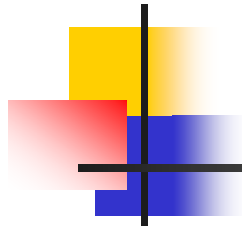
$P_{0-3}$  : entrées parallèles (de prog)

CLR : RAZ asynchrone active à 0

SPE : charg<sup>t</sup> parallèle synchrone  
actif au niveau bas  
prioritaire sur le comptage

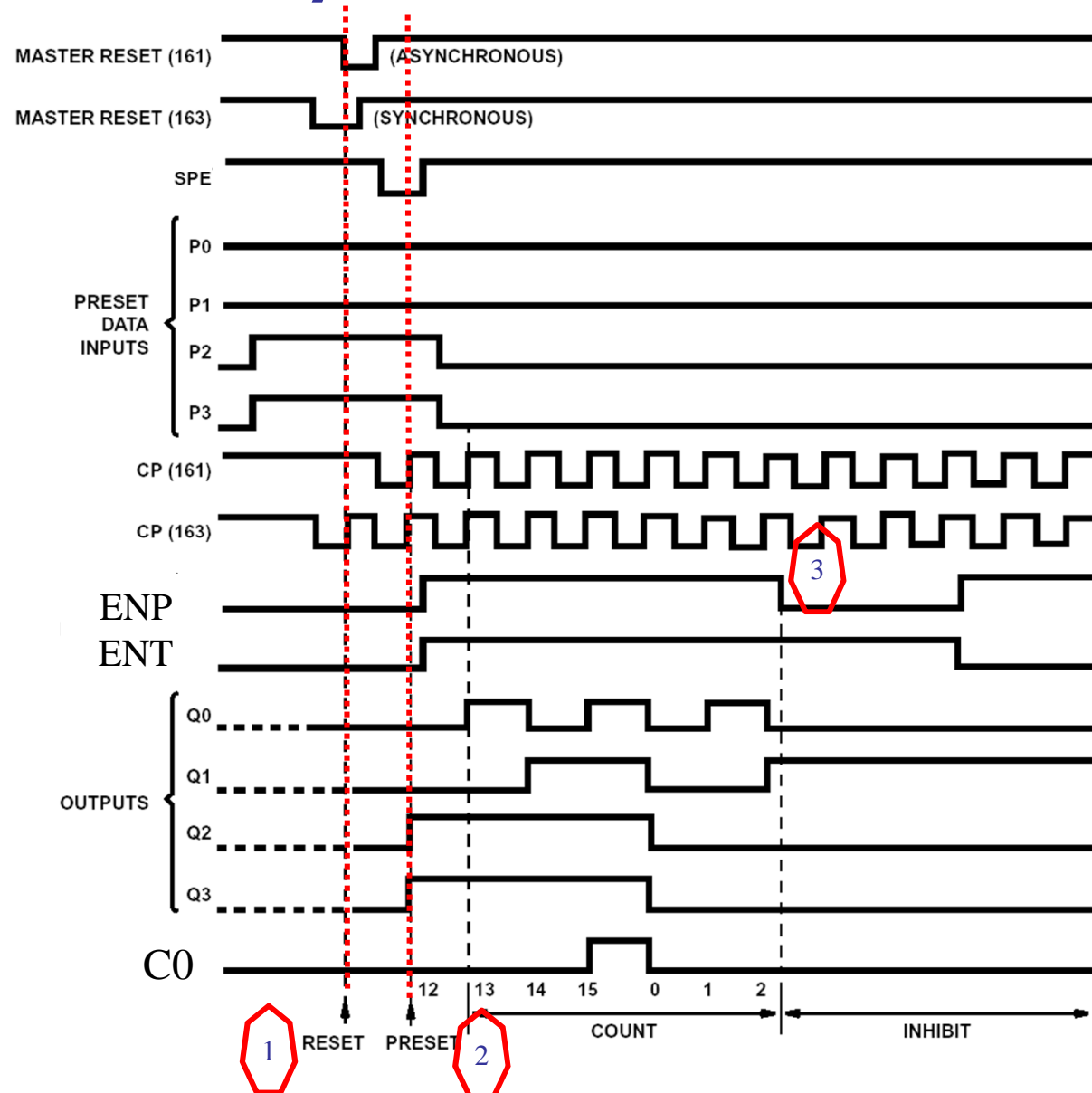
ENP/ENT : autorisation de comptage  
comptage si  $ENP \cdot ENT = 1$

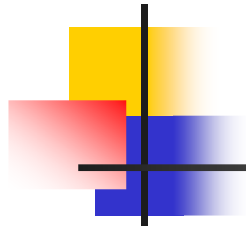
Co : retenue = 1 si  $Q_3 Q_2 Q_1 Q_0 \cdot ENT = 1$   
(ENT agit sur Co en asynchrone)



# Compteur synchrone

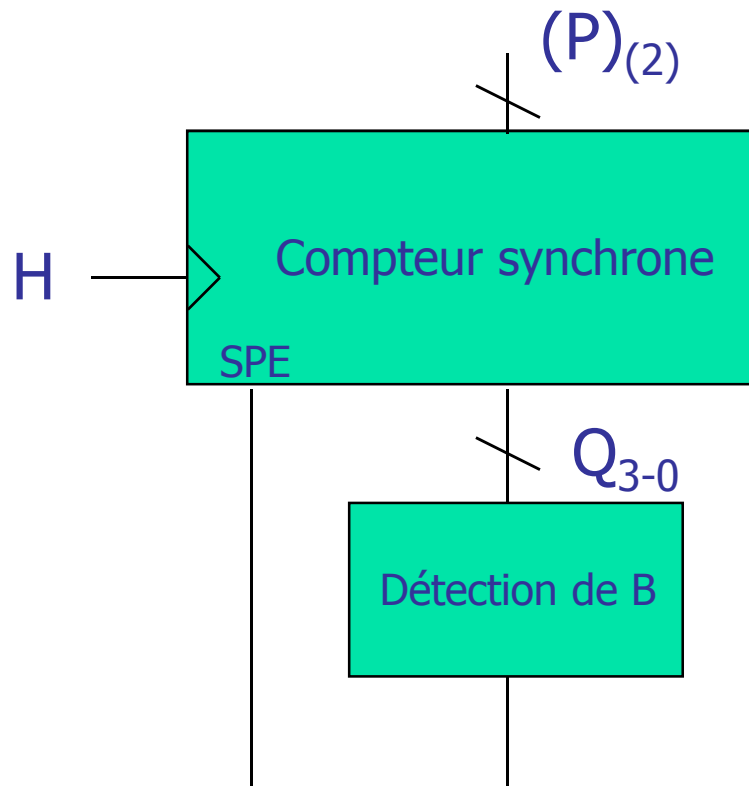
Présentation du  
74161 ou 74163  
Chronogramme





# Compteur synchrone de $(A)_{(2)}$ à $(B)_{(2)}$

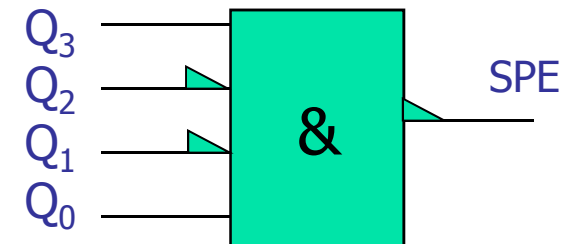
Utilisation d'un compteur avec chargement parallèle : lorsque le modulo est détecté, on recharge le compteur (entrée prioritaire)



$(A)_{(2)}$  : Première valeur du cycle nominal

$(B)_{(2)}$  : Dernière valeur du cycle nominal

## Détecteur



Si  $B = 1001 \Rightarrow \text{SPE}=0$

$Q_{3-0} = (A)_{(2)}$





# Compteurs synchrones cascades

On dispose de compteurs 4 bits, on veut compter sur N bits ( $N > 4$ )

Exemple ( $N=12$ ) :

0000 0000 0000  
0000 0000 0001

Il faut 3 compteurs 4 bits

Le premier compte tout le temps.

...

0000 0000 1111  
0000 0001 0000  
0000 0001 0001

Le deuxième ne compte que quand le premier est à 15.

...

0000 1111 1111  
0001 0000 0000

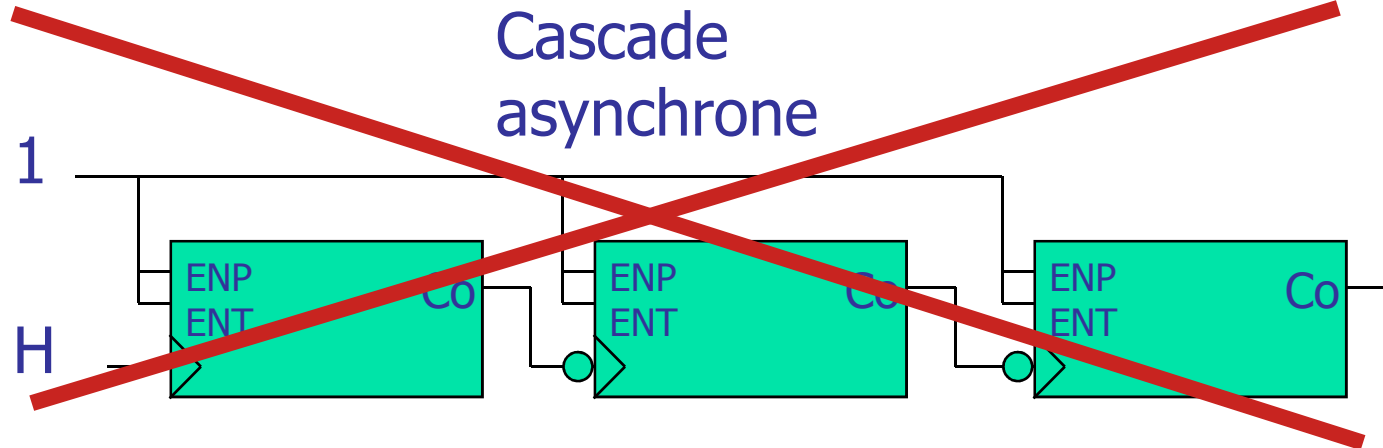
(le troisième compte quand premier=15 et deuxième=15)

Compteur LSB (de 0 à 15)

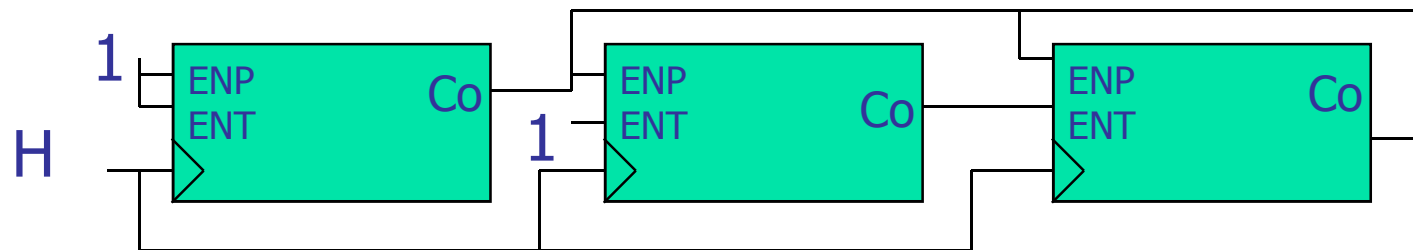
Compteur MSB (de 0 à 15)

**Cascadage des compteurs**

# Compteurs synchrones cascadés

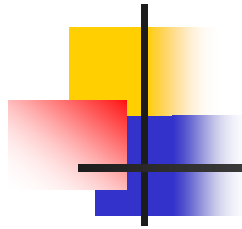


Cascade synchrone parallèle :



Période de H :

$$T_{\min} > T_p(\text{Clk/Co}) + T_p(\text{Ent/Co})$$



# Le pipeline

## Optimisation de machine séquentielle complexe :

L'architecture pipeline : Extrait du cours du MIT I09-1

Entrée :



Sortie :



Device : machine à laver  
Temps :  $\text{Washer}_{pd} = 30 \text{ mn}$   
Fonction : laver, essorer



Device : sèche linge  
Temps :  $\text{Drier}_{pd} = 60 \text{ mn}$   
Fonction : sécher

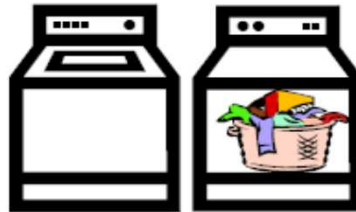
# Le pipeline

Approche Traditionnelle (one shot) :

Step 1:



Step 2:



$$\begin{aligned} \text{Total} &= \text{Washer}_{PD} + \text{Dryer}_{PD} \\ &= \underline{90} \text{ mins} \end{aligned}$$

# Le pipeline

## Approche Traditionnelle :



Step 1:



Step 2:



Step 3:



Step 4:



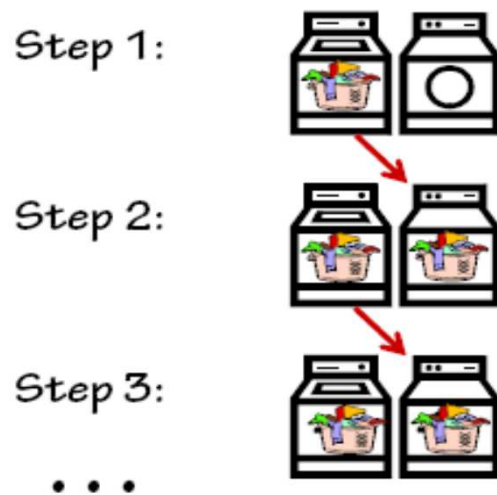
...

$$\text{Total} = N * (\text{Washer}_{PD} + \text{Dryer}_{PD})$$

$$= \frac{N * 90}{\quad} \text{ mins}$$

# Le pipeline

Approche optimisée (pipeline) :



En négligeant les 30 premières minutes

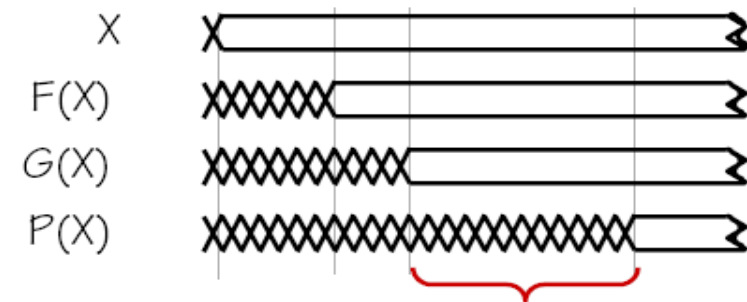
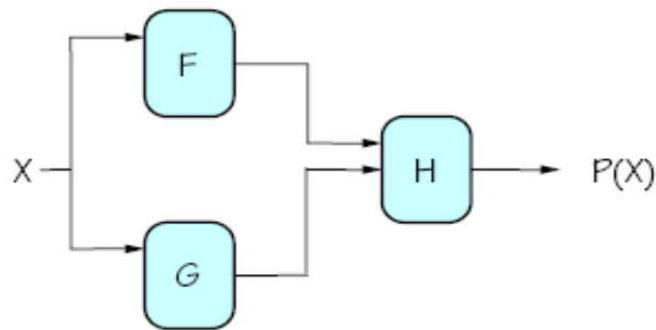
$$\begin{aligned} \text{Total} &= N * \text{Max}(\text{Washer}_{PD}, \text{Dryer}_{PD}) \\ &= \frac{N * 60}{1} \text{ mins} \end{aligned}$$

	Temps de propagation	Fréquence d'obtention d'un bac de linge
Approche traditionnelle :	90 mn	1/90mn
Approche optimisée :	120 mn*	1/60mn

\* : les changements ont lieu à chaque fin de lavage

# Le pipeline

Application aux systèmes logiques séquentielles :



$T_p$  : temps de propagation du circuit,  
 $1/T_p$  : fréquence.

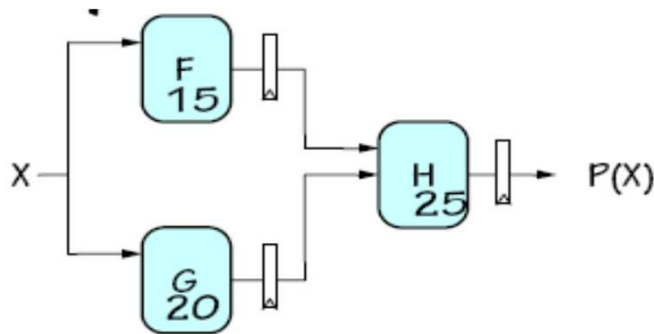
En posant :  $T_p(F)=15\text{ns}$ ,  
 $T_p(G)=20\text{ns}$ ,  
 $T_p(H)=25\text{ns}$

F et G sont « en attente »  
(idle) lorsque H établit son  
niveau

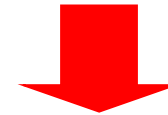
Le temps de propagation de ce circuit est de 45ns

# Le pipeline

Positionnons des bascules synchrones à la sortie de chaque partie :



F et G peuvent traiter  $X_{i+1}$  alors que H traite les sorties  $F(X_i)$  et  $G(X_i)$  contenu dans les bascules.



Pipeline à 2 étages

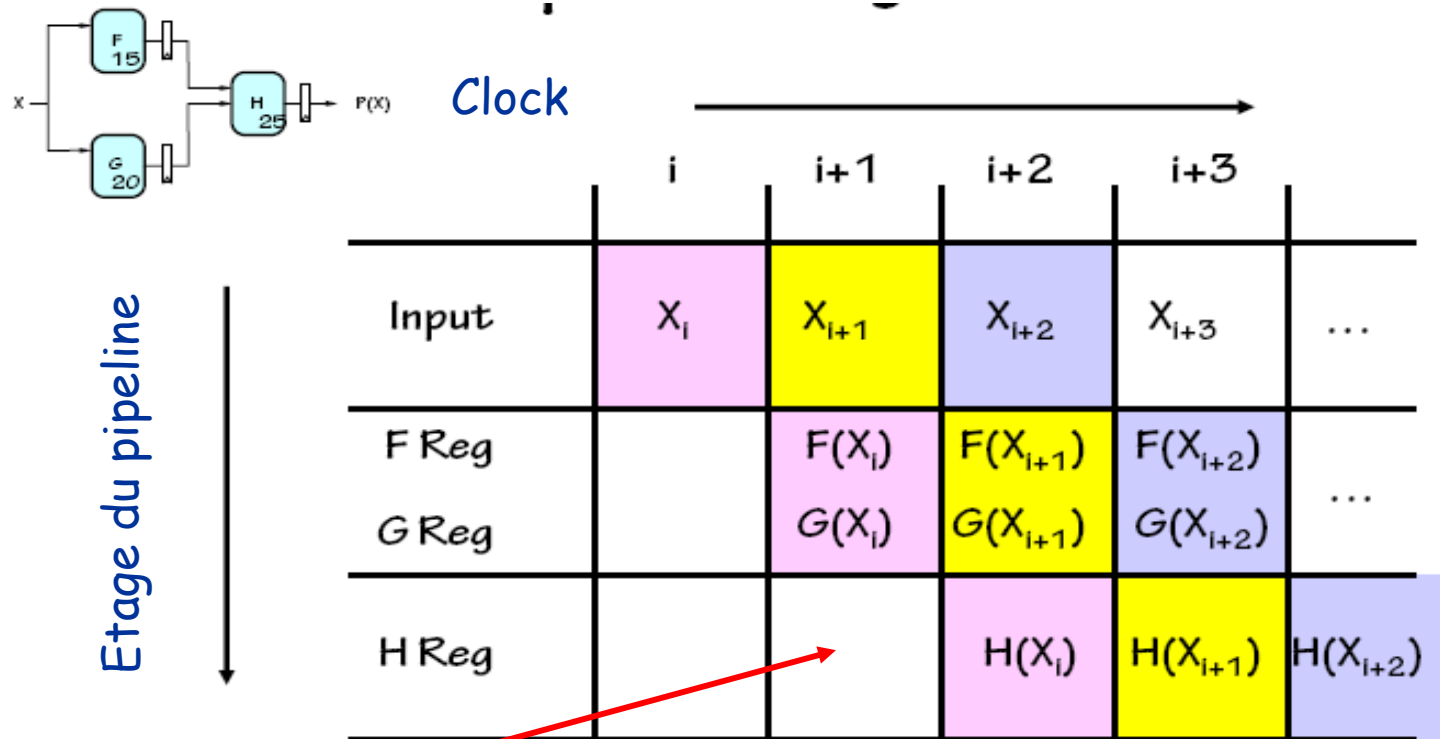
	Temps de propagation	Fréquence
Approche traditionnelle :	45 ns	1/45 ns
Approche optimisée :	50 ns*	1/25 ns

\* : l'élément le plus lent fixe la fréquence de l'horloge



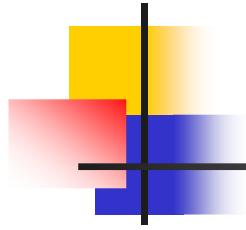
# Le pipeline

Diagramme d'un pipeline :



Temps de  
chargement du  
pipeline

Propagation en diagonale des signaux dans le diagramme



# Le pipeline

---

Application aux traitement des instructions :

L'exécution schématique d'une instruction est la suivante :

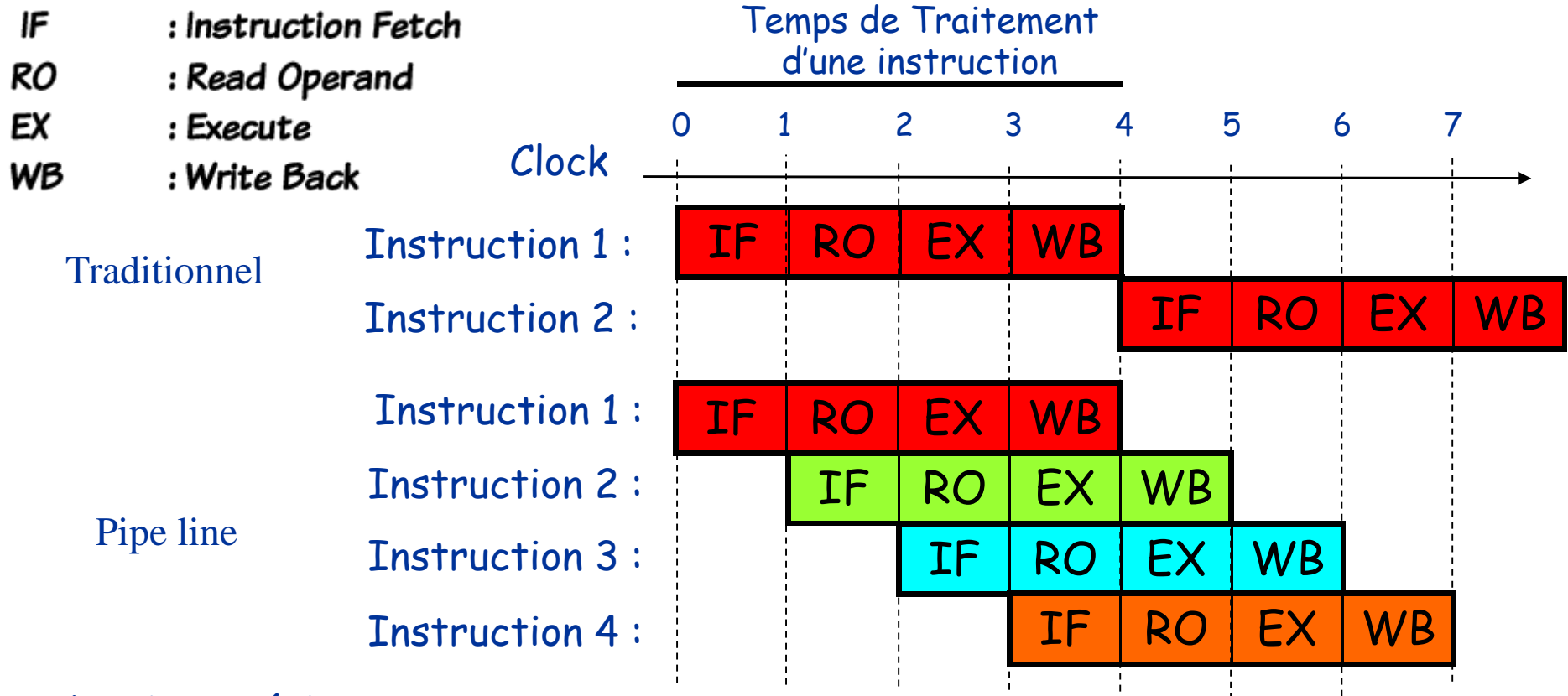
**Fetch, Decode, Execute, Save**

1. Chargement d'une instruction de la mémoire (fetch : chercher),
2. Analyse l'instruction (decode : ...),
3. Exécution de l'instruction (execute : ...)
4. Sauvegarde du résultat (save : ...)

Le nombre d'étape peut varier en fonction de l'architecture de processeur

# Le pipeline

Pipeline sur une machine de Harvard :



## Les inconvénients :

Programmation plus complexe de la structure,  
 Gestion des interruptions (charge et décharge du pipeline),  
 Erreur d'exécution (pipeline hazard, boucle, )...