

Notion de système séquentiel

- Un circuit séquentiel est un circuit qui possède une mémoire interne
- Éléments principaux d'un circuit séquentiel:
 - *état interne* : Le circuit peut changer d'état ; il a donc une faculté de mémorisation.
 - *un événement* est une action qui permet le changement de l'état interne d'un système. Un événement a une valeur.
 - *un événement pur* : c'est un événement sans action ; pour nous, il correspondra au top d'horloge. Il permet de prendre en considération la séquentialité des événements.

Notion de système séquentiel

■ Exemple :

- une personne a trois états (pauvre, modeste, riche)
- deux événements (argent, facture).

En fonction de l'événement, la personne va changer d'état.



Modélisation d'un système séquentiel

■ définitions :

- variables en entrée (événements) : $v \in V$
- entrée de l'horloge (événement pur) : H
- état interne : $e \in E$ (ensemble des états du système)
- sortie : $s \in S$ (ensemble des sorties)
- fonction de transition : $ft : E \times V \rightarrow E$
permet le calcul du nouvel état en fonction des entrées et de l'état interne du système.
- fonction de sortie : $fs : E \times V \rightarrow S$
permet le calcul de l'ensemble des sorties.

Modélisation d'un système séquentiel

■ Remarques

- La sortie du système dépend de ce qu'on met en entrée et de l'état interne ; c'est ce qui définit la faculté de mémorisation des circuits séquentiels, par opposition aux circuits combinatoires.
- La valeur de l'horloge en elle-même ne rentre pas dans les calculs.
- On peut trouver une certaine analogie avec les machines à états finis, ou automates finis. On pourra représenter le système par un graphe.

Modélisation d'un système séquentiel

■ notations :

- $e_{t+1} = (\text{état au temps } t+1) = f_T(e_t, v_t).$

Le nouvel état à l'instant $(t+1)$ se calcule avec l'état au temps t , et la valeur de l'entrée au temps t .

- Le (temps de) cycle est le temps qui s'écoule entre deux tops d'horloge, soit entre le top t et le top $(t+1)$.

- En sortie, on a une fonction de l'état interne, et de l'entrée.
 $s_t = f_s(e_t, v_t)$

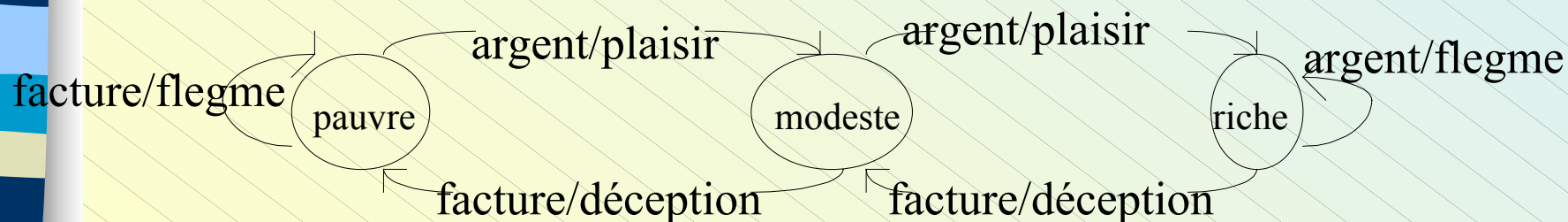
Modélisation d'un système séquentiel

■ Représentations

– Table des états

V	E	$f_T(e_t, v_t).$	$f_s(e_t, v_t)$
facture	pauvre	pauvre	flegme
facture	modeste	pauvre	déception
facture	riche	modeste	déception
argent	pauvre	modeste	plaisir
argent	modeste	riche	plaisir
argent	riche	riche	flegme

– Diagrammes des états

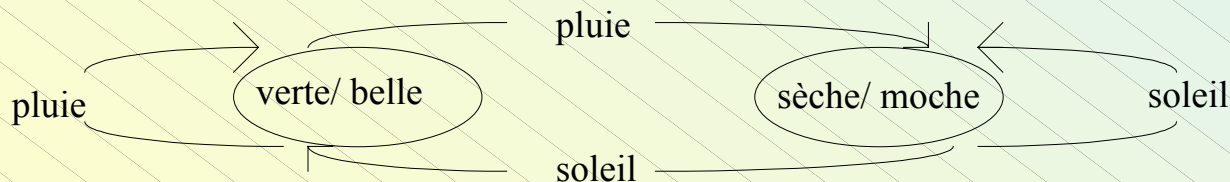


Modélisation d'un système séquentiel

- Il y a deux types de systèmes séquentiels :
 - **la machine de Mealy** : la sortie est fonction de l'état interne et des entrées (exemple précédent avec la personne).
 - **la machine de Moore** : la sortie est fonction uniquement de l'état interne mais pas des entrées. On a donc $s=f_s(e)$.

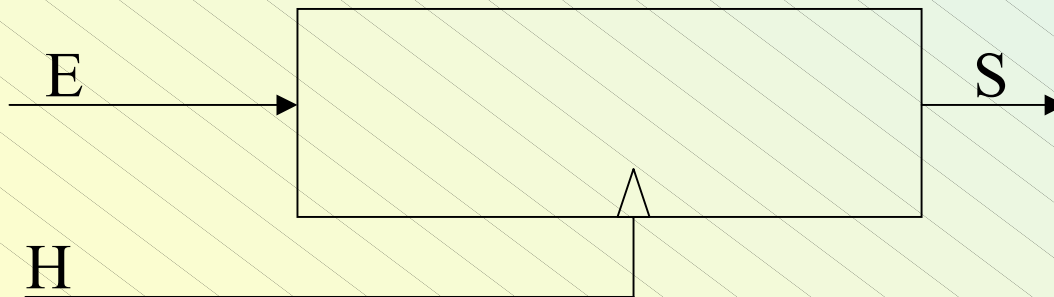
Exemple : la pelouse

elle a deux états (verte et sèche) et deux événements (pluie, soleil)



Circuits synchrones

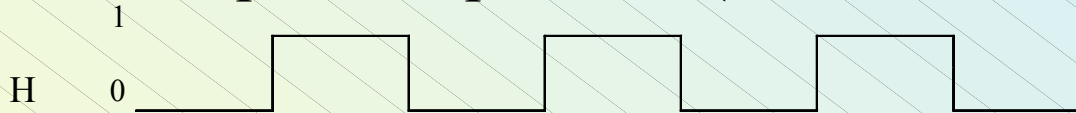
- Un circuit synchrone est un circuit séquentiel aligné sur une horloge qui provoque les changements d'états du circuit.
- Plusieurs circuits séquentiels peuvent être alignés sur la même horloge dans le cas d'un assemblage.



Circuits synchrones

■ 1- Conventions d'usage

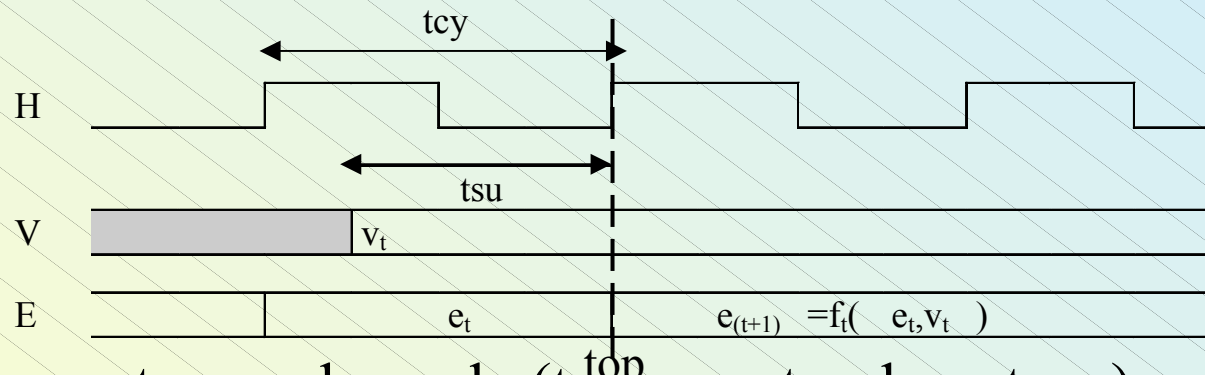
- Une horloge est un signal qui peut prendre la valeur 1 ou 0. On la représente par H ou (Ck comme clock).



- La prise en compte du top se fait pendant les transitions de l'horloge. Deux transitions possibles:
 - front montant : $0 \rightarrow 1$
 - front descendant : $1 \rightarrow 0$
- Une horloge est un signal périodique.
 - T : la période est exprimée en secondes
 - $f = 1/T$ est la fréquence exprimée en hertz (Hz).

Circuits synchrones

■ 2- Contraintes temporelles



- t_{cy} = temps de cycle (temps entre deux tops).
- Il faut présenter les valeurs avant un certain temps pour qu'elles soient prises en compte = temps de prépositionnement = t_{su} .
- Le changement d'état se produit au moment du top

Exemples de circuits séquentiels

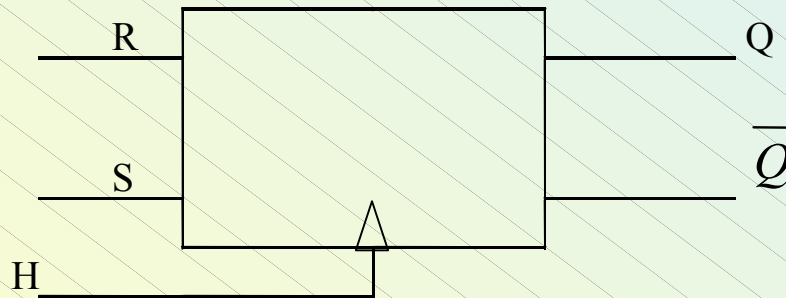
■ 1- *Bascules (ou bistables ou flip flop)*

- Ce sont des circuits qui ont deux états codés 0 et 1.
- L'état du circuit est directement visible en sortie.
- L'état du circuit est mémorisé pendant un temps de cycle.
- Ils ont une sortie notée Q et éventuellement une sortie inverse de la sortie normale notée \overline{Q} .
- Le nombre d'états donne la capacité de mémorisation (1 bit).

Exemples de circuits séquentiels

■ a- la bascule R-S

- elle a deux entrées : R et S
- les entrées sont mutuellement exclusives (jamais à 1 en même temps)



- R : Reset (mise à 0)
- S : Set (mise à 1)

Exemples de circuits séquentiels

■ la bascule R-S

Table de vérité

R	S	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0

Table de vérité résumée

R	S	Q_{t+1}
1	0	0
0	1	1
0	0	Q_t

Exemples de circuits séquentiels

■ b- la bascule D

- Elle a une seule entrée notée : D
- La valeur en entrée est visible en sortie

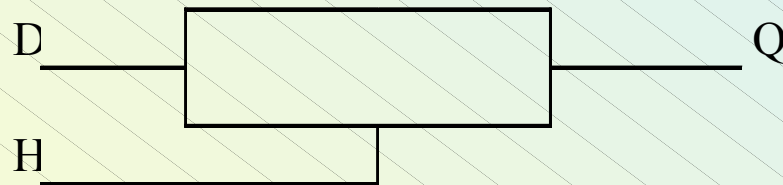


Table de vérité

D	Q
0	0
1	1

Exemples de circuits séquentiels

■ la bascule J-K

- Même principe que la bascule R-S mais les entrées J et K ne sont plus exclusives

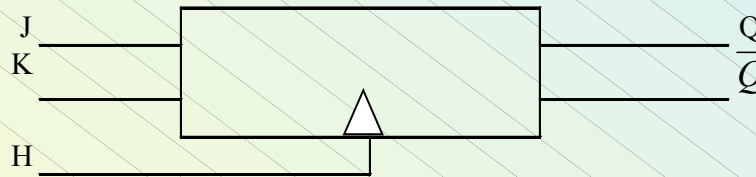


Table de vérité

J	K	Q_{t+1}
0	0	Q_t
1	0	1
0	1	0
1	1	$\overline{Q_t}$

Exemples de circuits séquentiels

■ 2- Les compteurs

- Un compteur est un circuit dont l'état est interprété comme un nombre.
- Il peut être :
 - incrémenté,
 - décrémenté,
 - chargé (i.e. on peut lui imposer une valeur au départ). Le chargement normal est l'initialisation à 0.

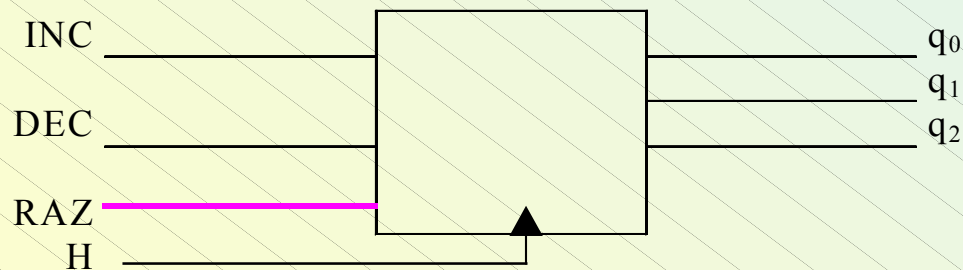
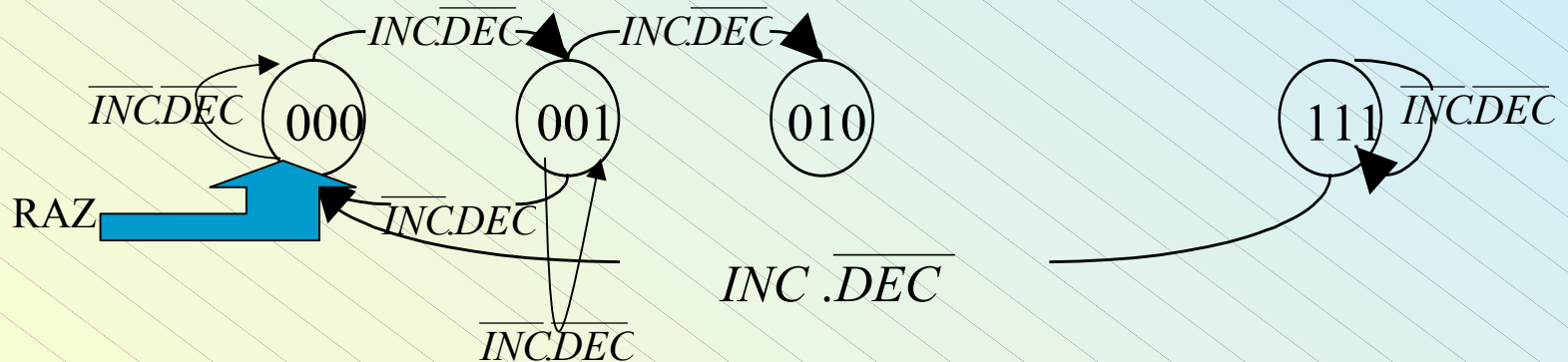
Exemples de circuits séquentiels

- Les différents types de compteurs :
 - **les compteurs permanents** : ce sont des compteurs sans entrée, toujours incrémentés modulo une certaine valeur, à chaque top d'horloge.
 - **les compteurs-décompteurs** : ils possèdent au minimum deux entrées (et éventuellement une troisième entrée s'ils sont chargeables) :
 - Inc : pour incrémenter,
 - Dec : pour décrémenter.
 - Raz : pour la remise à zéro.

Exemples de circuits séquentiels

Exemple :

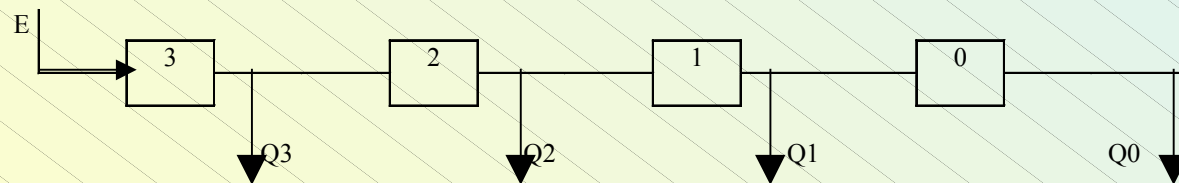
- un compteur-décompteur chargeable à 8 états.
- Les états sont codés sur 3 bits.



Exemples de circuits séquentiels

■ 3- les registres

- Un registre est un regroupement de n bascules qui permet la mémorisation de n bits.
- les différents types de registres :
 - registre à décalage sortie série
 - registre à décalage sorties parallèles (voir dessin)
 - registre entrées parallèles et sorties parallèles

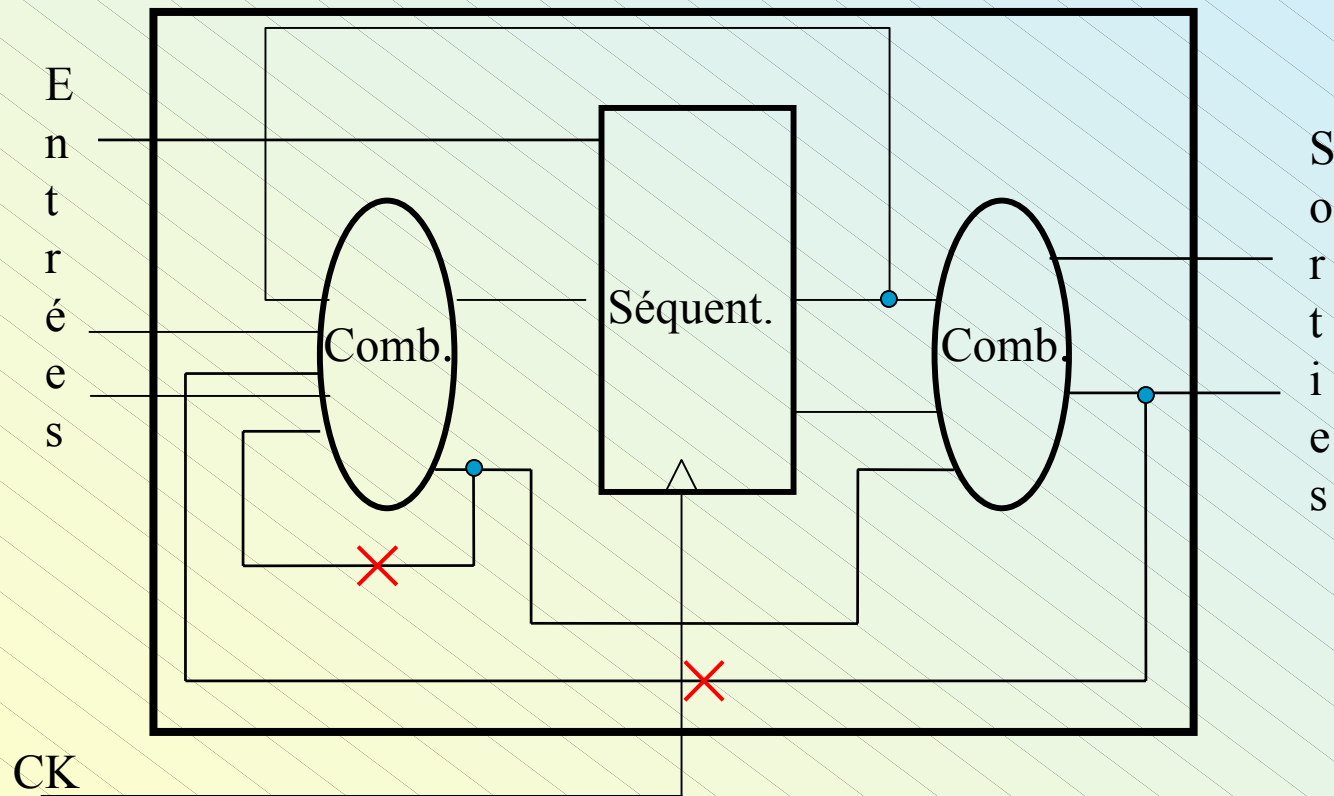


Assemblage de circuits synchrones

- Dans un assemblage de circuits synchrones, on a plusieurs circuits séquentiels qui sont synchronisés sur la même horloge et éventuellement des circuits combinatoires.
- Quand on fait un assemblage de circuits synchrones, il faut absolument éviter les boucles combinatoires : on ne peut pas prendre une sortie d'un circuit combinatoire pour en faire une de ses entrées, ni une entrée d'un autre circuit combinatoire ;
- Cependant, il est tout à fait licite de faire la même chose à partir d'un circuit séquentiel.

Assemblage de circuits synchrones

■ Exemple:



Quelques méthodes de réalisation de circuits synchrones simples

■ *Principes généraux*

- Un circuit est simple s'il a un nombre d'états inférieur à 15.
- **Processus :**
 - 1- définir les entrées
 - 2- définir les états et choisir le codage de ces états.
 - pour n bits, le nombre d'états est compris entre 2^{n-1} et 2^n
 - 3- définir la fonction de transition ft (diagramme des états).
 - 4- définir la fonction de sortie $fs(e, v)$ (diagramme des états).
 - 5- transformation des fonctions de transition et de sortie
 - » $ft(e, v) \rightarrow Ft(q_1, q_2, \dots, q_n, v)$
 - » $fs(e, v) \rightarrow Fs(q_1, q_2, \dots, q_n, v)$
 - » Définition des équations logiques des q_i et des S_i .
 - 6-Réalisation du circuit.

Quelques méthodes de réalisation de circuits synchrones simples

1. Utilisation des bascules D

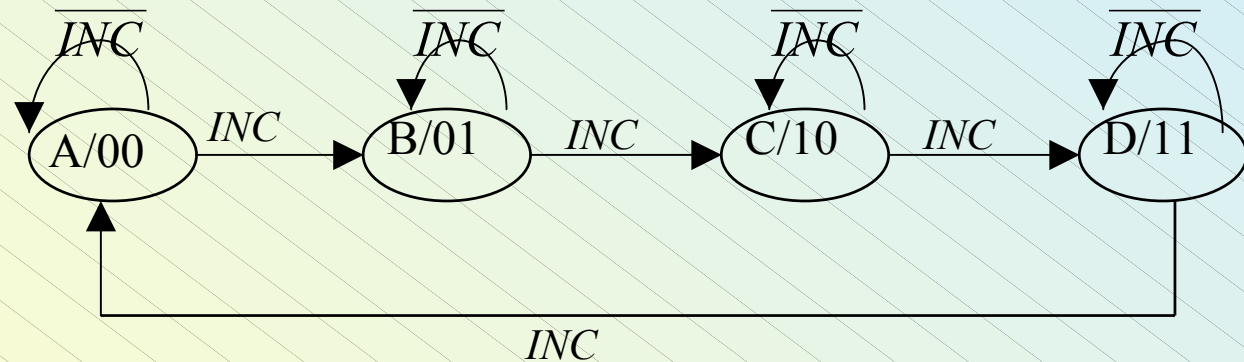
- Si le codage des états est sur n bits alors chaque bit représente une bascule.
- $q_1, \dots, q_n = Ft(q_1, q_2, \dots, q_n, v)$
- La valeur en entrée est celle voulue en sortie

Exemple : compteur modulo 4 avec une commande Inc

- 1- Définition des entrées: Une seule entrée Inc
- 2- Définition des états: Le système a 4 états notés A, B, C, D donc le codage est sur 2 bits.
- 3- Définition de la fonction de transition, on fait le diagramme des états

Réalisation avec les bascules D

- compteur modulo 4



Quelques méthodes de réalisation de circuits synchrones simples

(1b) Utilisation des bascules D

4- codage des états:

A : 00

B : 01

C : 11

D : 10

5- fonction de transition : Ft

Inc \ q1q0	0	1
	0 0 0 1 1 1 1 0	0 1 1 1 1 0 0 0

$$q1 = q1\overline{INC} + q0INC$$

$$q0 = q0\overline{INC} + \overline{q1}INC$$

Quelques méthodes de réalisation de circuits synchrones simples

(1c) Utilisation des bascules D

5b- Fonction de sortie

q1	q0	S1	S0
0	0	0	0
0	1	0	1
1	1	1	0
1	0	1	1

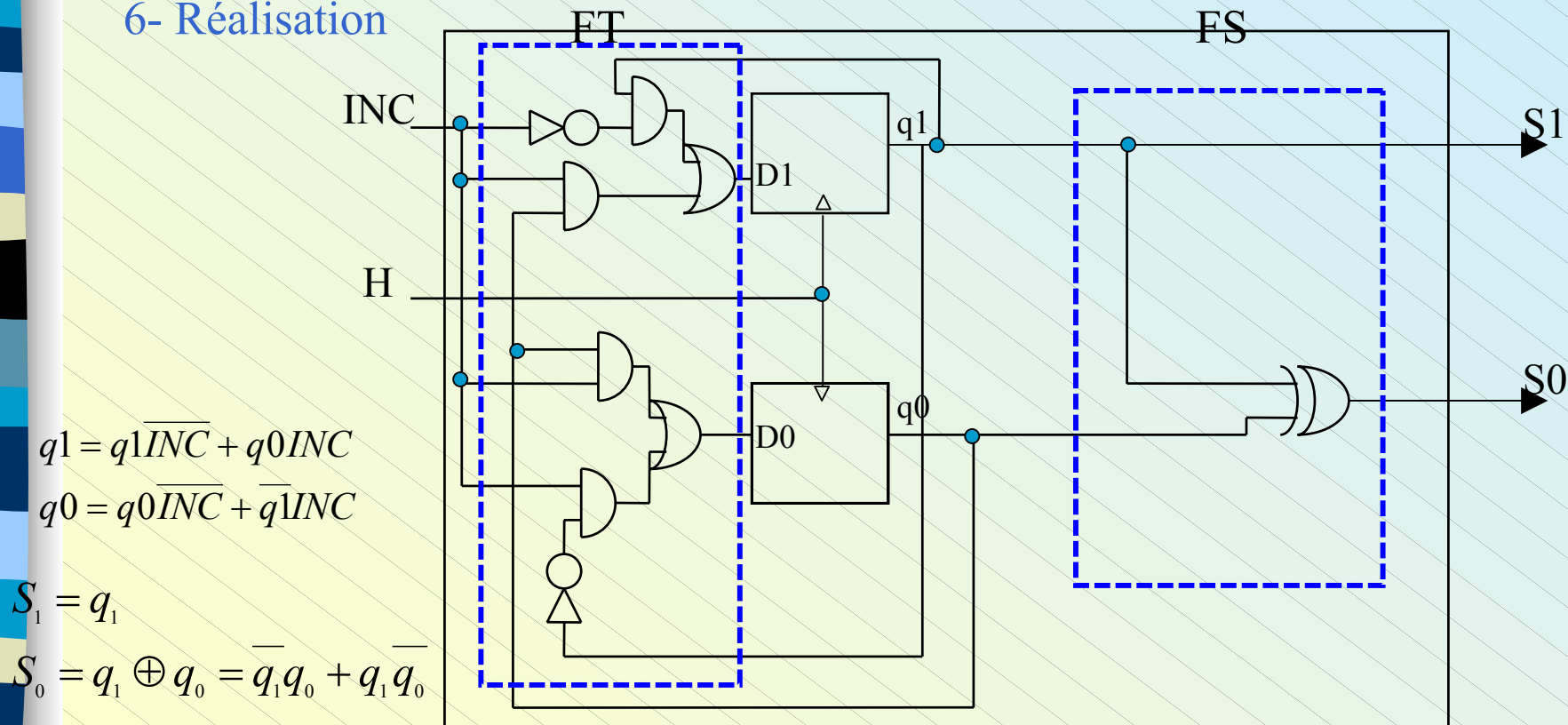
$$S_1 = q_1$$

$$S_0 = q_1 \oplus q_0 = \overline{q_1}q_0 + q_1\overline{q_0}$$

Quelques méthodes de réalisation de circuits synchrones simples

(1c) Utilisation des bascules D

6- Réalisation



Quelques méthodes de réalisation de circuits synchrones simples

2- Utilisation des bascules J-K

- Le processus est le même que celui utilisé avec les bascules D.

(les étapes 1 à 3 sont exactement les mêmes que pour les bascules D)

- La différence est sur la définition de la fonction de transition car les bascules ont deux entrées : il faut définir les équations logiques des entrées (J_i et K_i) en considérant les différentes transitions possibles:

on a 4 transitions possibles.

q_i	q_{i+1}	J_i	K_i
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

X vaut
soit 0
soit 1
(valeur quelconque)

Quelques méthodes de réalisation de circuits synchrones simples

(2b) Utilisation des bascules J-K

Exemple du compteur modulo 4

(étapes 1 à 3 idem bascules D)

On a deux bascules . Il faut donc définir les équations logiques des entrées:

- J1 et K1 pour la première bascule
- J0 et K0 pour la deuxième bascule

q_i	q_{i+1}	J_i	K_i
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

4. équation logique de J1

q_1q_0	Inc	0	1
00		0	0
01		0	1
11		X	X
10		X	X

$$J_1 = q_0 INC$$

Quelques méthodes de réalisation de circuits synchrones simples

(2c) Utilisation des bascules J-K

4b. équation logique de K1

INC		0	1
q_1q_0			
00		X	X
01		X	X
11		0	0
10		0	1

q_i	q_{i+1}	J_i	K_i
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$$K_1 = \overline{q_0} INC$$

- On trouve ensuite les équation logique de J2 et K2
- puis on détermine la fonction de sortie (étape 5 idem bascules D)
- puis on fait la réalisation à partir des équations logiques (étape 6)

Quelques méthodes de réalisation de circuits synchrones simples

3- Machines à jetons

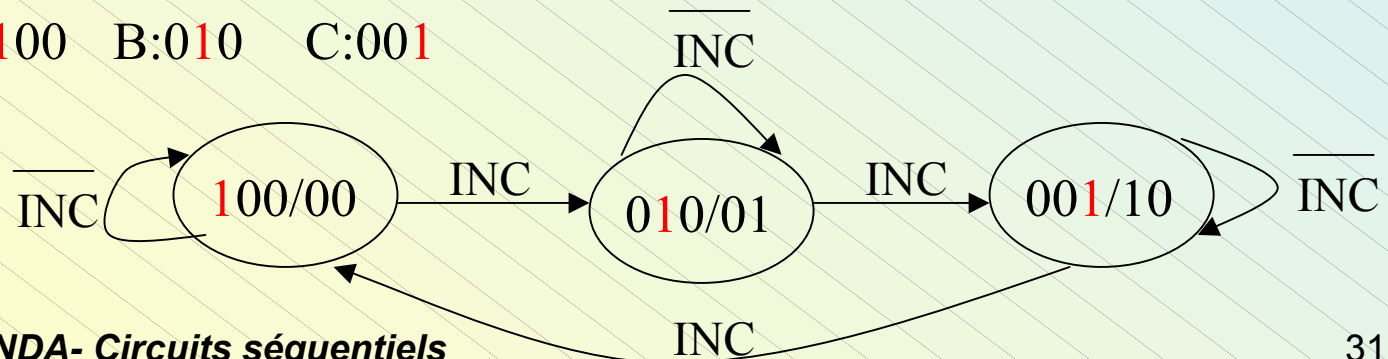
– Principe:

- le nombre de bits pour le codage des états est égal au nombre d'états
- à chaque état on fait correspondre un bit « baladeur »

– Exemple: Compteur modulo 3

3 états donc le codage est avec 3 bits:

A: 100 B: 010 C: 001

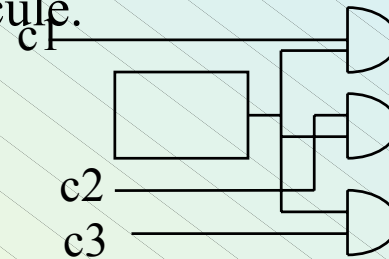
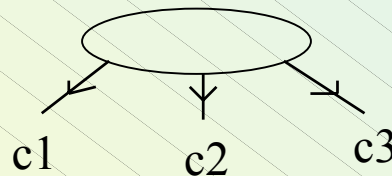


Quelques méthodes de réalisation de circuits synchrones simples

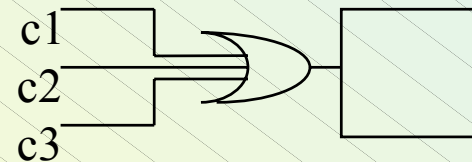
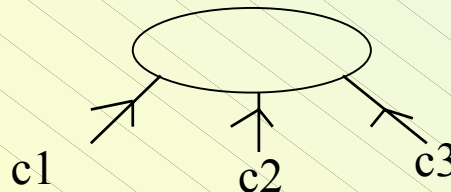
■ (3b) Machines à jetons

– Règles de réalisation:

- 1- A chaque flèche sortant d'un état, on va faire correspondre un "Et" avec la sortie de la bascule.



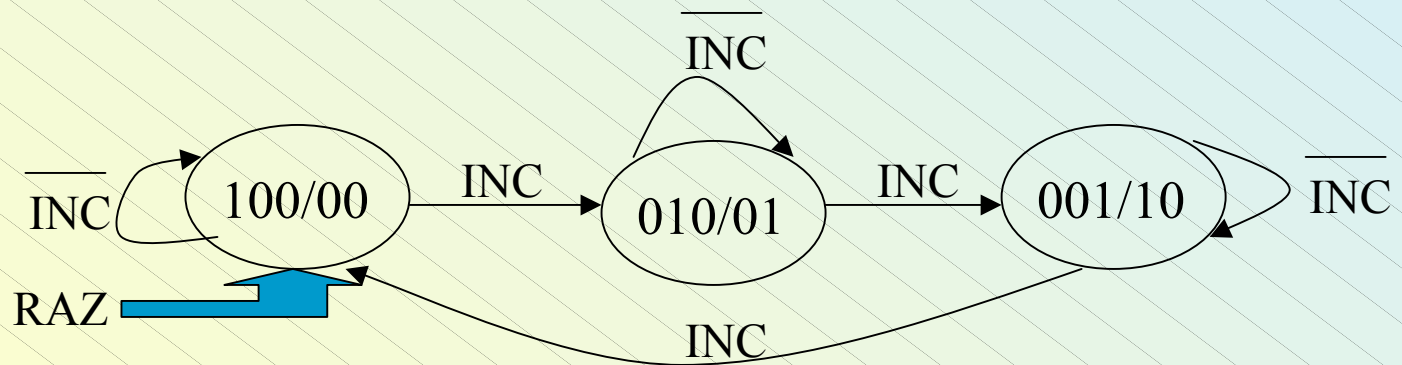
- 2- Aux flèches "entrant" dans un état, correspond un "Ou" avec l'entrée de la bascule.



Quelques méthodes de réalisation de circuits synchrones simples

(3c) Machines à jetons

- 3- Si on a une initialisation, deux conditions sont à respecter :
 - La commande est envoyée sur le “OU” de la bascule de l’état initial
 - Le Non de la commande est envoyé sur le “ET” à l’entrée des autres bascules.

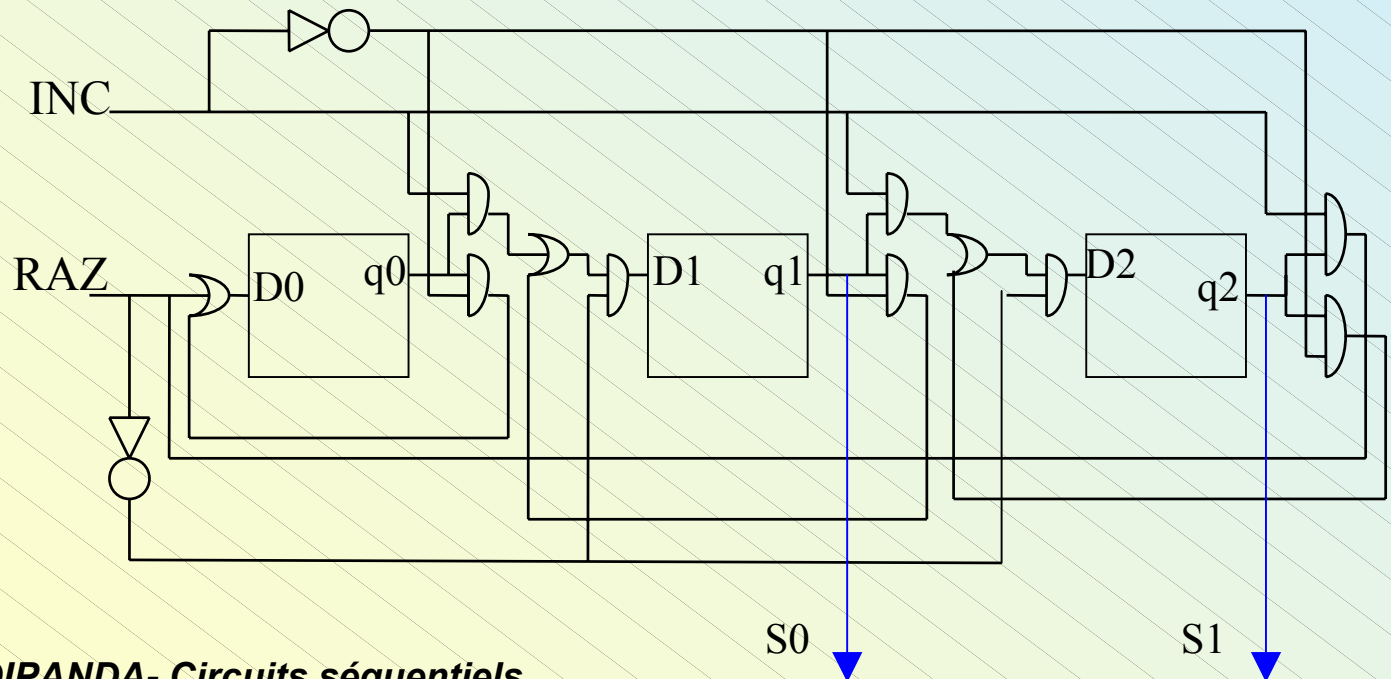
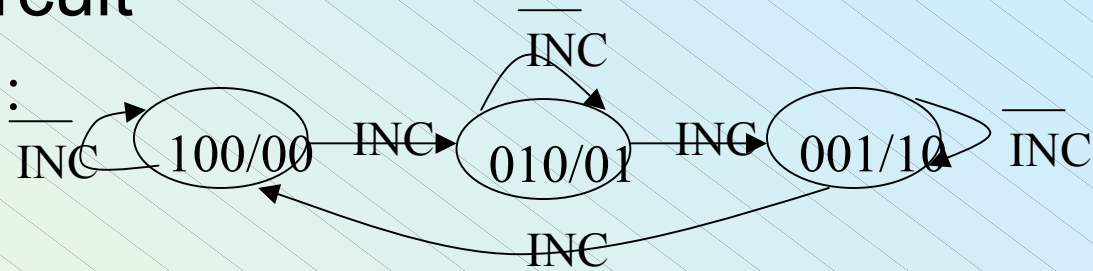


Quelques méthodes de réalisation de circuits synchrones simples

■ Réalisation du circuit

– Fonction de sortie :

- $S0=q1$
- $S1=q2$



Changements d'états asynchrones

Les changements d'états ne sont donc plus synchronisés sur une horloge.

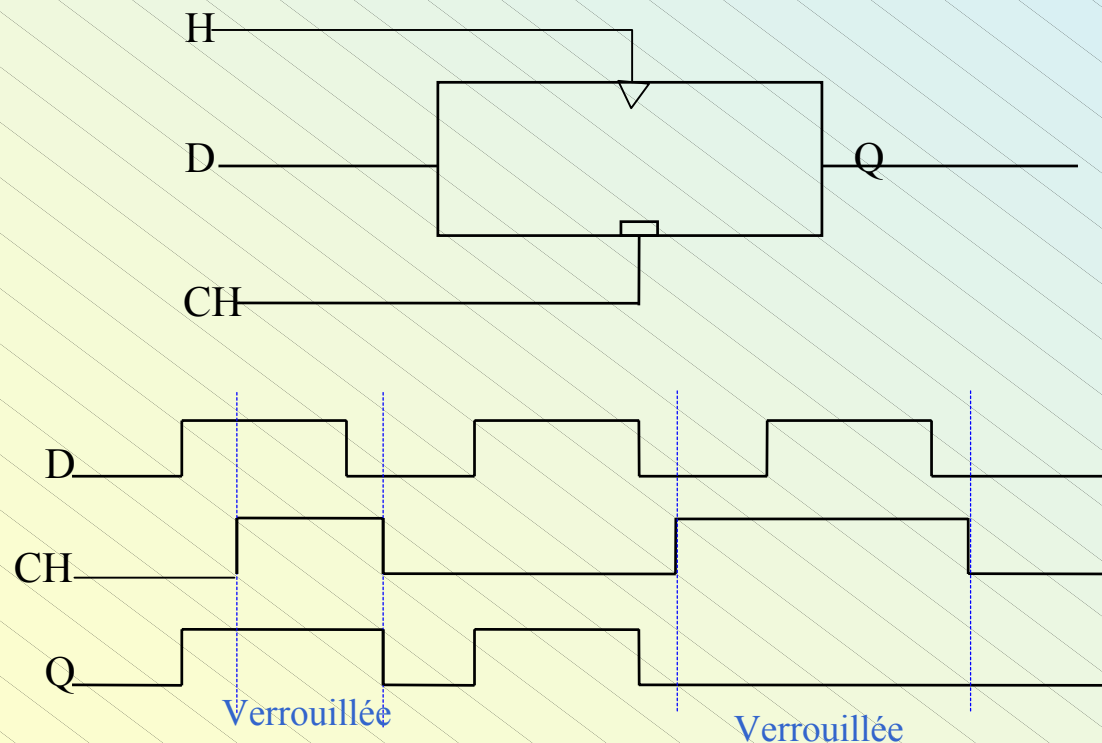
1- Les verrous

- Un verrou a un comportement analogue à une bascule. Il possède de même un bit de mémorisation.
- Les changements d'états sont commandés par une commande de verrouillage (ou de déverrouillage).

Changements d'états asynchrones

■ Exemple de bascule D avec commande de verrouillage.

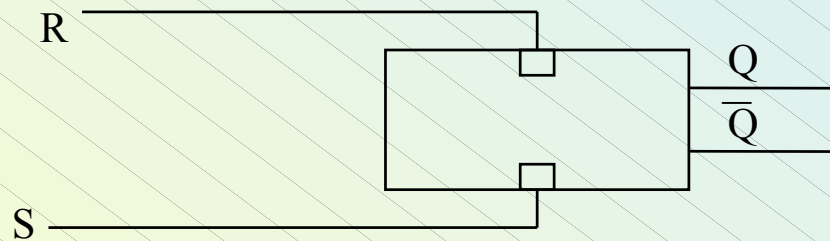
Principe: Quand la commande de verrouillage est à 1, la sortie Q est verrouillée (convention à imposer).



Exemples de verrous

■ a- le verrou RS

Il a deux entrées R et S et une sortie Q



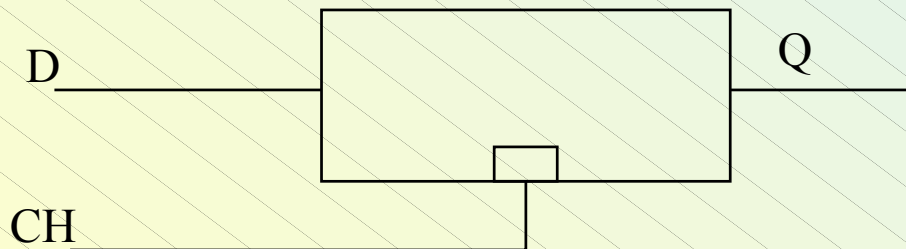
- Quand S est actif, (i.e. vaut 1), l'état est à 1
- Quand R est actif, (i.e. vaut 1), l'état est à 0
- Il y a verrouillage lorsque R et S sont à 0 (Q garde sa valeur précédente).
- Il n'est pas permis d'avoir simultanément R et S à 1.

Exemples de verrous

■ b- le verrou D

Il est équivalent à une bascule D, mais au lieu d'avoir une horloge, on a une commande de verrouillage.

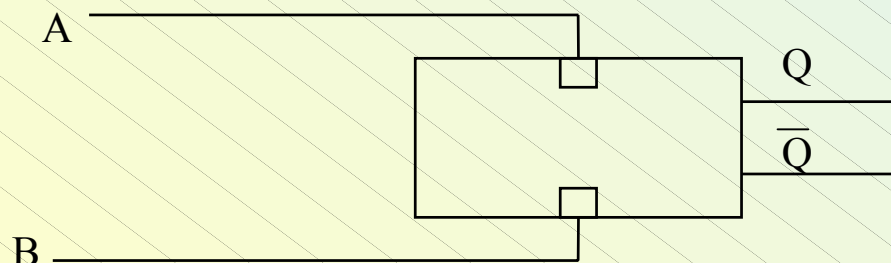
- Déverrouiller $\rightarrow Q=D$
- Quand Ch vaut 1 (ou 0 selon la convention choisie), Q reste inchangé, c'est-à-dire que le système est verrouillé.



Exemples de verrous

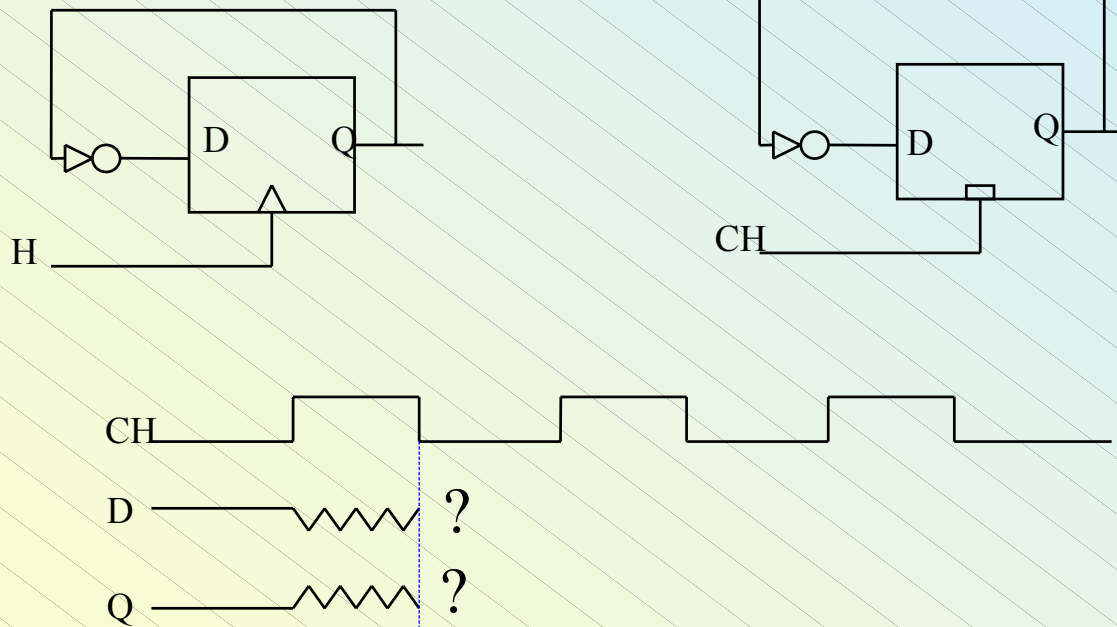
■ c- verrou de Muller ou verrou de rendez-vous

- comprend deux entrées A et B et une sortie Q.
- Comportement :
 - Q est à 1 quand A et B sont actives
 - Q est à 0 lorsque A et B sont à 0.
 - quand A et B sont à 1 alors Q est à 1 et Q revient à 0 lorsque A et B passent à 0; entre temps, on peut très bien avoir A=1 , B=0 et Q=1 ou A=0, B=1 et Q=1.
 - Notion de rendez-vous :
 - Q=1 quand tout le monde est présent
 - Q=0 quand tout le monde est parti.



Synchronisation des systèmes asynchrones

■ Exemple: Inverseur

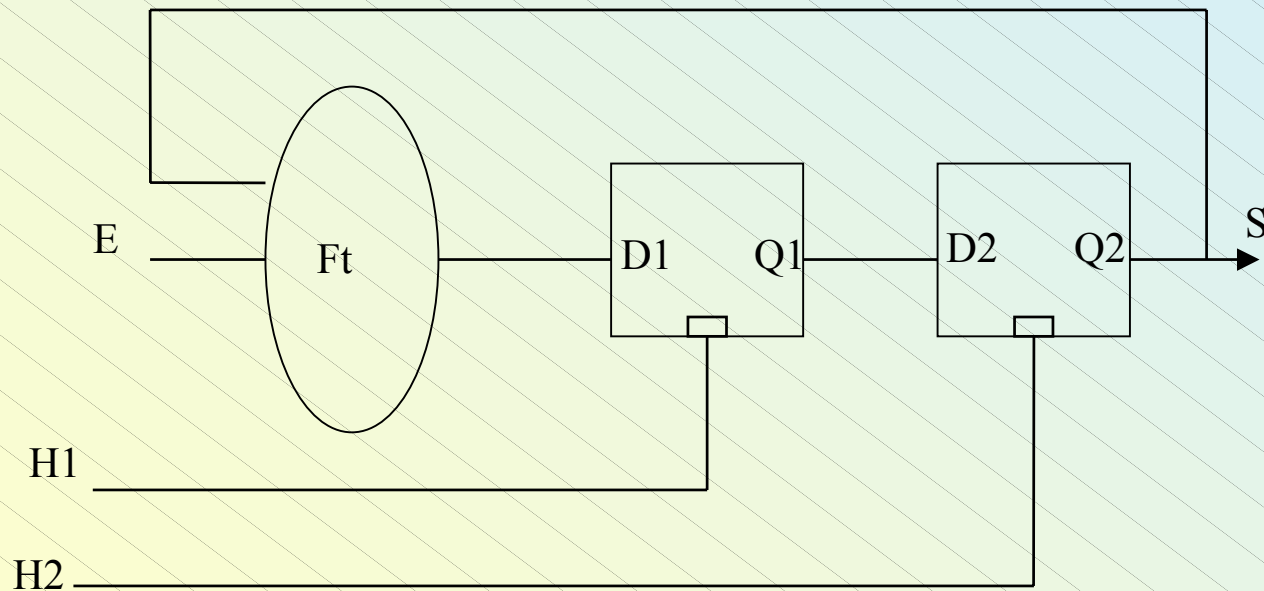


Verrouillage si CH=0

Synchronisation des systèmes asynchrones

1- *Enregistrement maître esclave*

- On utilise deux verrous et deux horloges H1 et H2 mutuellement exclusives.

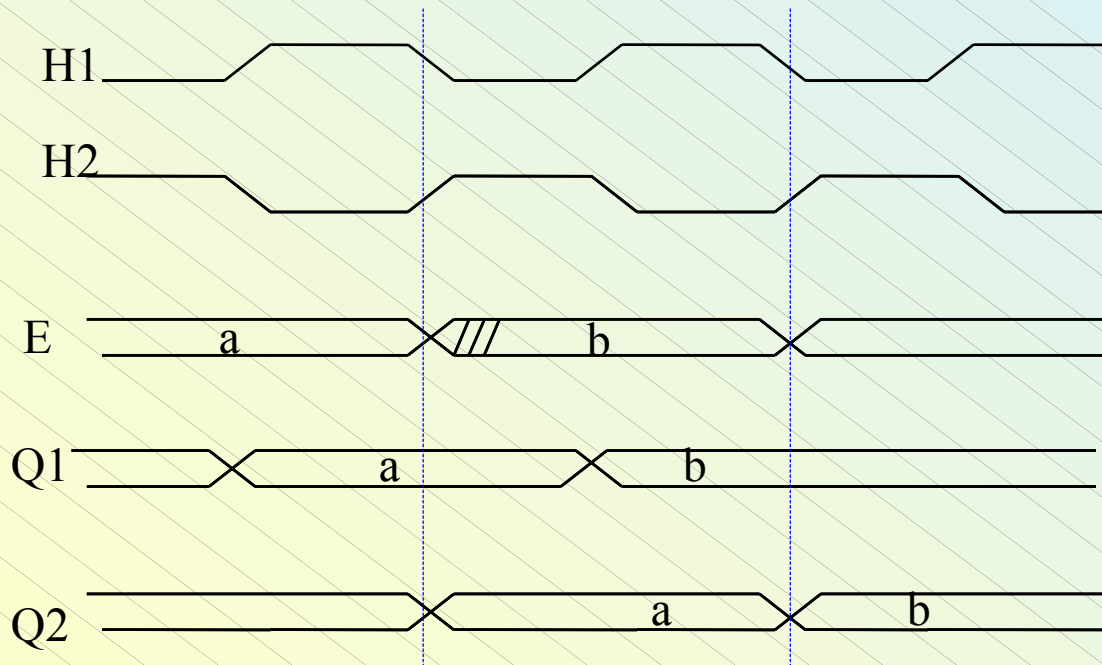


Synchronisation des systèmes asynchrones

(1b) *Enregistrement maître esclave*

Exemple de simulation du fonctionnement d'une bascule

Dessin précédent avec Ft vide



Synchronisation des systèmes asynchrones

2- Synchronisation des entrées asynchrones

- **Problème:** Comment prendre en compte d'une séquence de valeurs fournies de manière asynchrone.
- **Solution:** Utilisation du protocole d'interaction séquentielle (**Hand-shake**).
 - l'entrée *valid* (request) indique l'arrivée d'une nouvelle valeur.
 - la sortie d'acquiescement *Ack*, indique la prise en compte de la valeur.
- Un dialogue entre l'utilisateur et le système va s'instaurer.

Synchronisation des systèmes asynchrones

Description du protocole d'interaction séquentielle

- Au départ, attente d'une valeur : **Ack** et **Valid** sont à 0.
 - 1- Arrivée d'une nouvelle valeur, l'utilisateur le signale au système: **valid** passe à 1
 - 2- Attente de l'acquittement de la valeur par le système ;**Ack** passe à 1
 - 3- confirmation de l'acquittement par l'utilisateur : **valid** passe à 0
 - 4- **Ack** retombe à 0 : le système fait savoir à l'utilisateur qu'il est prêt à recevoir une nouvelle valeur.
- Le dialogue recommence jusqu'à épuisement de la séquence des valeurs de l'utilisateur.

Synchronisation des systèmes asynchrones

- Description du protocole d'interaction séquentielle

