

• Systèmes séquentiels

Profs. Peña & Perez-Uribe & Mosqueron

Basé sur le cours du Prof. E. Sanchez

Classification des systèmes logiques

- Systèmes combinatoires :
 - Etat des sorties dépend uniquement des entrées

système univoque

- Systèmes séquentiels :
 - Etat des sorties dépend des entrées et de l'historique (événements passés)

implique une mémorisation

- Exemples simples: registres, compteurs, générateurs de séquence, ...

Pourquoi des systèmes séquentiels?

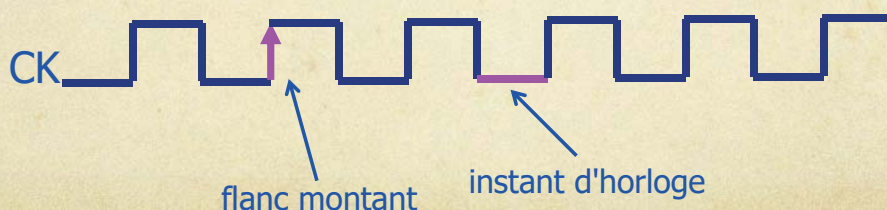
- L'évolution de certains systèmes dépend de l'historique des événements
- Pour une même combinaison des entrées l'état des sorties peut être différent
⇒ Ce n'est plus un système univoque

Applications:

- Résoudre des problèmes de nature séquentiel
- Décomposer des systèmes combinatoires en traitement séquentiel pour des raisons de coût et de complexité
- Dans la pratique: les systèmes sont généralement séquentiels

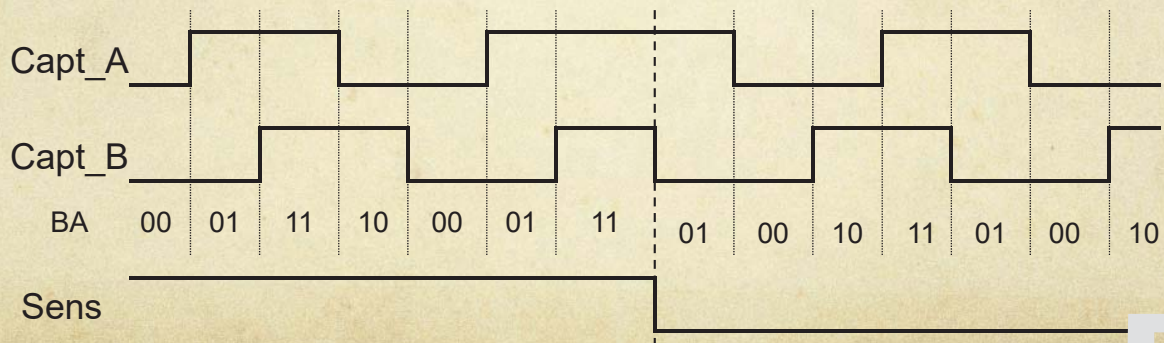
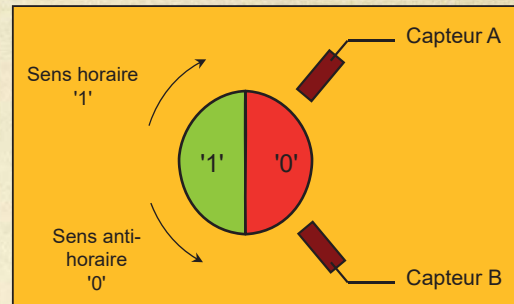
Systèmes séquentiels

- La valeur des sorties dans un système séquentiel ne dépend pas uniquement des valeurs des entrées à l'instant présent, mais également de la valeur des entrées dans les instants précédents
- Les systèmes séquentiels introduisent donc la notion de **mémoire** (pour stocker l'état du système) et d'**horloge** (pour déterminer les instants de mesure)



Exemple de système séquentiel

- Détecteur de sens de rotation
(voir animation)



ARO1 - APE & CPN & RMQ

5

Analyse du détecteur de sens de rotation

- La combinaison des entrées Capt_A et Capt_B ne permet pas de déterminer le sens de rotation
- La sortie Sens dépend :
 - Succession des combinaisons Capt_A/Capt_B
 - Nécessaire de mémoriser l'état précédent du système

ARO1 - APE & CPN & RMQ

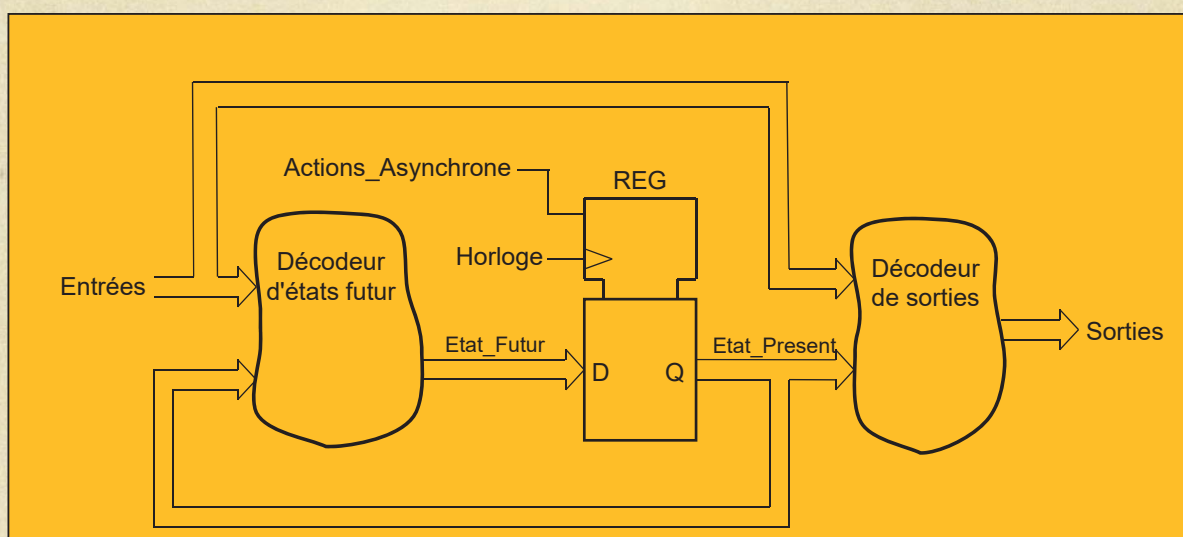
6

Conception de systèmes séquentiels

- Combiner des cellules séquentielles (mémoires) avec de la logique combinatoire
- Ramener les systèmes séquentiels à :
 - des systèmes combinatoires + des variables internes (mémoires)
 - puis établir une table des états similaire à une TDV



Schéma bloc système séquentiel



- Décodeurs d'états futur et de sortie sont des systèmes combinatoires
- REG : système séquentiel (registre) = états internes

Types d'éléments mémoires

- Systèmes séquentiels synchrones :

Systèmes séquentiels sont basés sur des flip-flops

- flip-flop : bascule sensible au flanc
- type principalement utilisé : flip-flop D (DFF)
- autres types : flip-flop T et JK

Qu'appelle-t-on une bascule ?

- Bascule = circuit ayant :
 - un comportement séquentiel (dépendant du temps, de l'historique des événements)
 - une seule sortie à 2 états
 - pas plus de 2 états stables
- Bascules classées selon le nombre d'états stables :
 - astable (oscillateur) : aucun état stable
 - monostable : 1 état stable
 - **bistable** : 2 états stables

Bascules bistables

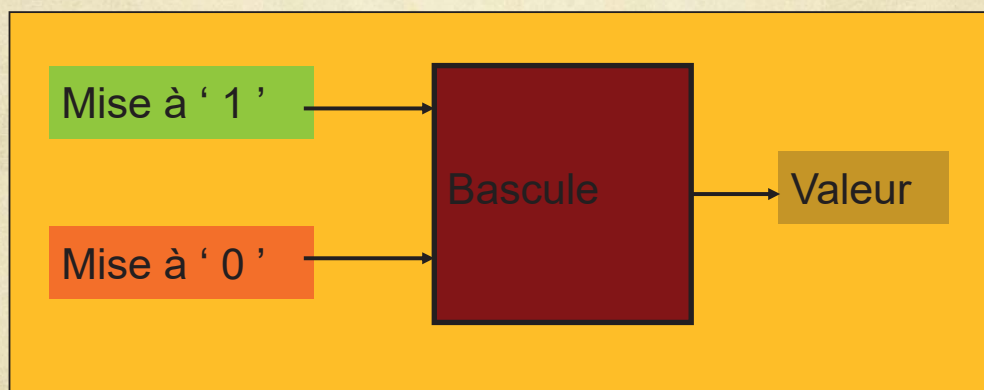
Définition d'une bascule bistable :

- 2 états stables
- Des actions sur les entrées permettent de forcer un état ou un changement d'état
- En l'absence de nouvelles actions, l'état est maintenu
 - Effet de mémorisation
- L'état ne dépend pas uniquement de l'état actuel des entrées, mais aussi de l'historique des évènements

11

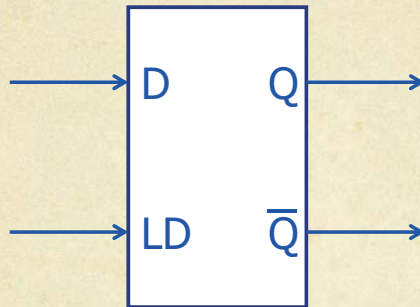
Bascule élémentaire

- Module logique permettant d'initialiser et de stocker une valeur logique



12

Elément de mémoire D (*latch*)



D: entrée d'excitation
LD: entrée de contrôle
Q: sortie ou état du *latch*

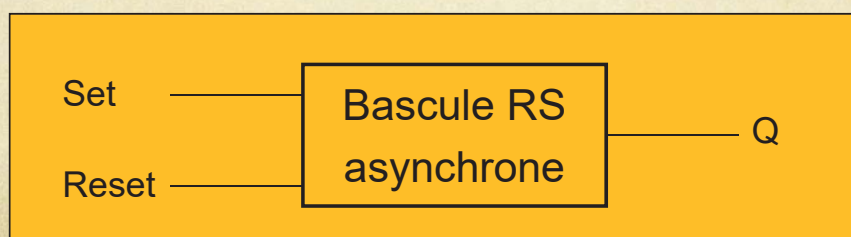
ARO1 - APE & CPN & RMQ

13

Bascule RS asynchrone

Comportement :

- Set = '1' mise à '1' de la sortie
- Reset = '1' mise à '0' de la sortie
- Set = Reset = '0' maintien, la sortie conserve sa valeur
- Set = Reset = '1' interdit



E. Messerli (HES-SO / HEIG-VD / REDS), 2016

p

14

Bascule RS asynchrone

Conception d'une bascule RS :

- l'évolution dépend des entrées **et** de l'état présent (historique), nous rajoutons Q comme entrée dans la TDV

	R	S	Q	Q+	
Q : Etat présent de la bascule RS	0	0	0	.0.	} Maintien
	0	0	1	.1.	
Q+ : Etat futur de la bascule RS	0	1	0	.1.	} Set
	0	1	1	.1.	
	1	0	0	.0.	} Reset
	1	0	1	.0.	
	1	1	0	'..	interdit
	1	1	1	'..	interdit

E. Messerli (HES-SO / HEIG-VD / REDS), 2016

Bascule RS asynchrone

Conception de la bascule RS:

- Etablir l'équation simplifiée de Q^+ à l'aide d'une table de Karnaugh
- Dessiner le schéma logique de Q^+
- Connecter le signal Q sur Q^+
- Transformer le schéma logique en utilisant des portes NOR et des inverseurs NOT

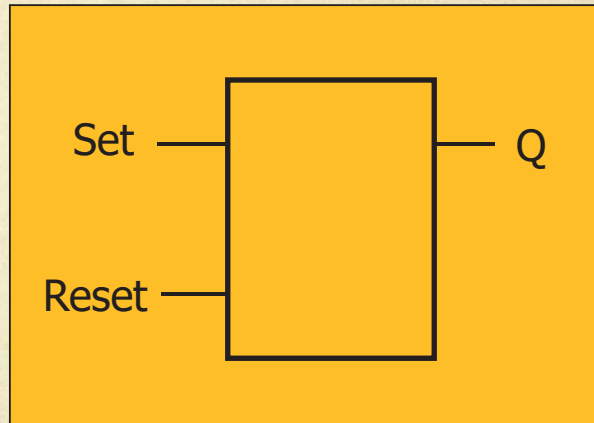
E. Messerli (HES-SO / HEIG-VD / REDS), 2016

Bascule RS asynchrone

○ Table des transitions

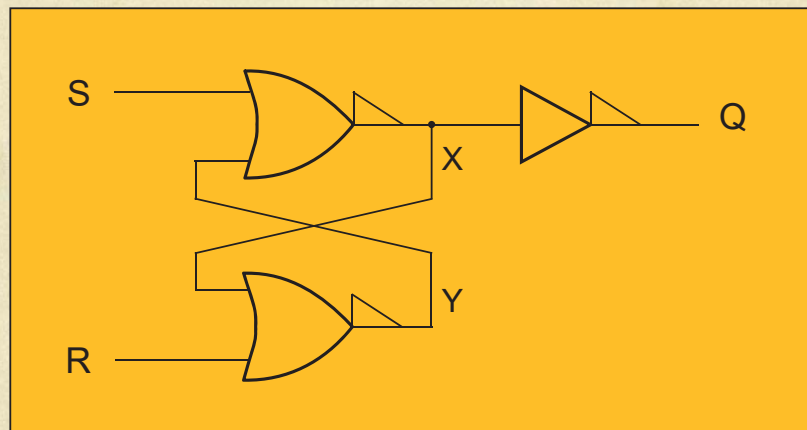
R	S	Q+
0	0	Q
0	1	1
1	0	0
1	1	interdit

○ Symbole CEI



Bascule RS asynchrone

○ Schéma logique :



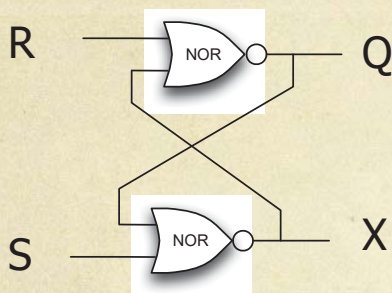
A quoi sert la porte inverseur sur la sortie Q ?

Bascule RS asynchrone: **inconvenients**

- La bascule RS asynchrone est sensible à toute impulsion sur ses entrées
 - Si durée supérieure au temps de propagation de 2 portes => changement d'état définitif de la bascule RS
 - Si durée inférieure au temps de propagation de 2 portes => état final de la bascule RS indéterminé !
- Les commutations des circuits produisent des parasites
- Nos circuits génèrent des aléas
- Ces parasites et ces aléas peuvent faire changer d'état une bascule RS asynchrone

E. Messerli (HES-SO / HEIG-VD / REDS), 2016

Latch R-S



1) Si $R = 1, S = 0,$
 $Q = (1 + X)' = 0$
 $X = (0 + Q)' = Q' = 1$

2) Si $R = 0, S = 1,$
 $X = (1 + Q)' = 0$
 $Q = (0 + X)' = X' = 1$

3) Si $R = 0, S = 0,$
 $Q = (0 + X)' = X'$
 $X = (0 + Q)' = Q'$

4) On interdit $R = 1, S = 1$

si $R = 1 \rightarrow$ reset ($Q = 0$)
si $S = 1 \rightarrow$ set ($Q = 1$)
Si $S=0, R=0 \rightarrow$ "hold"
 $X = Q'$

Bascule à verrouillage (latch)

- Solution pour éviter les basculements indésirables :
 - verrouiller la bascule lorsque des parasites et/ou des aléas peuvent apparaître sur les entrées
 - verrouiller = rendre les entrées inactives
- Ajout d'une entrée de commande, nommée G, soit:
 - $G = '1'$ → active les actions des entrées S et R
 - $G = '0'$ → désactive les actions des entrées S et R

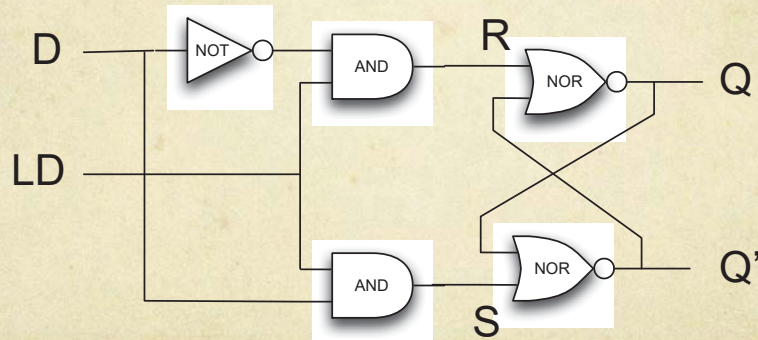
Latch D ...

But : mémoriser une information d'un bit, au moment souhaité

- Entrées :
 - D : donnée
 - G : commande de verrouillage (gate)
- Comportement :
 - $G = '1'$ => la sortie prend la valeur de l'entrée D
:
 - **cette bascule est transparente** (verrou ouvert)
 - $G = '0'$ => la sortie conserve sa valeur

Latch D

Latch D: si LD = 1: S = D, R = D' -> Q = D
 si LD = 0 -> R=0, S=0 -> "hold"



ARO1 - APE & CPN & RMQ

23

Latch D

D	LD	Q
0	0	Q
0	1	0
1	0	Q
1	1	1

- Si l'entrée de contrôle est inactive (LD=0), la sortie conserve sa valeur: fonction de mémorisation (HOLD).
- Si l'entrée de contrôle est active (LD=1), la sortie prend la valeur de son entrée d'excitation (LOAD).
- La table d'opérations du *latch* est donc:

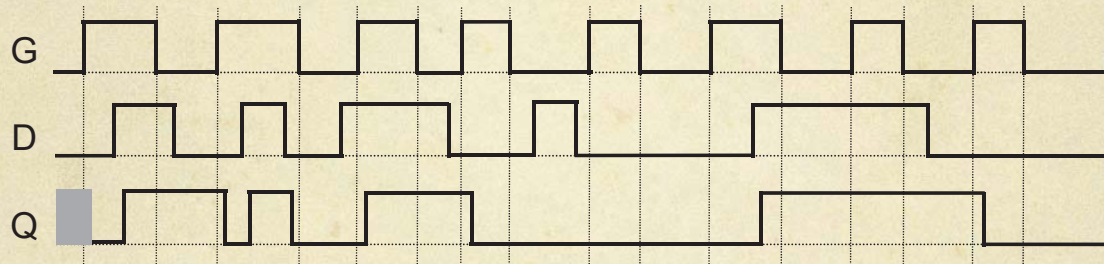
opération	description	LD
HOLD	$Q \leftarrow Q$	0
LOAD	$Q \leftarrow D$	1

ARO1 - APE & CPN & RMQ

24

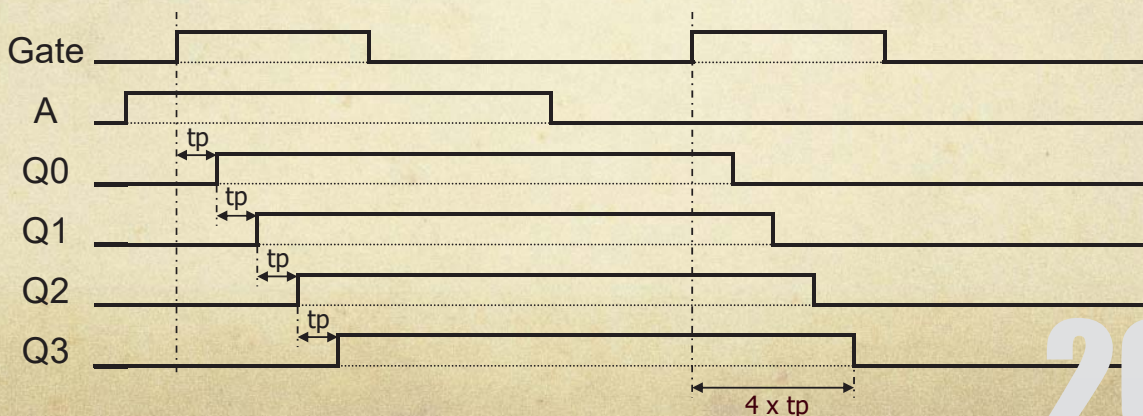
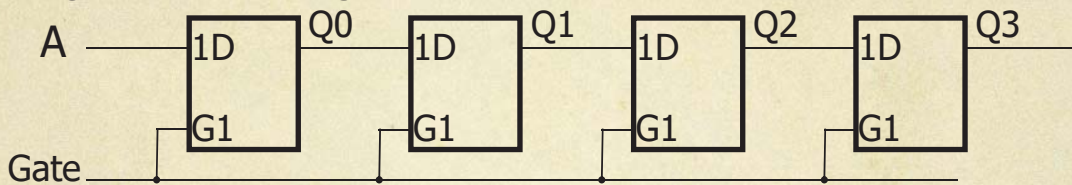
Latch D

○ Compléter le chronogramme suivant:



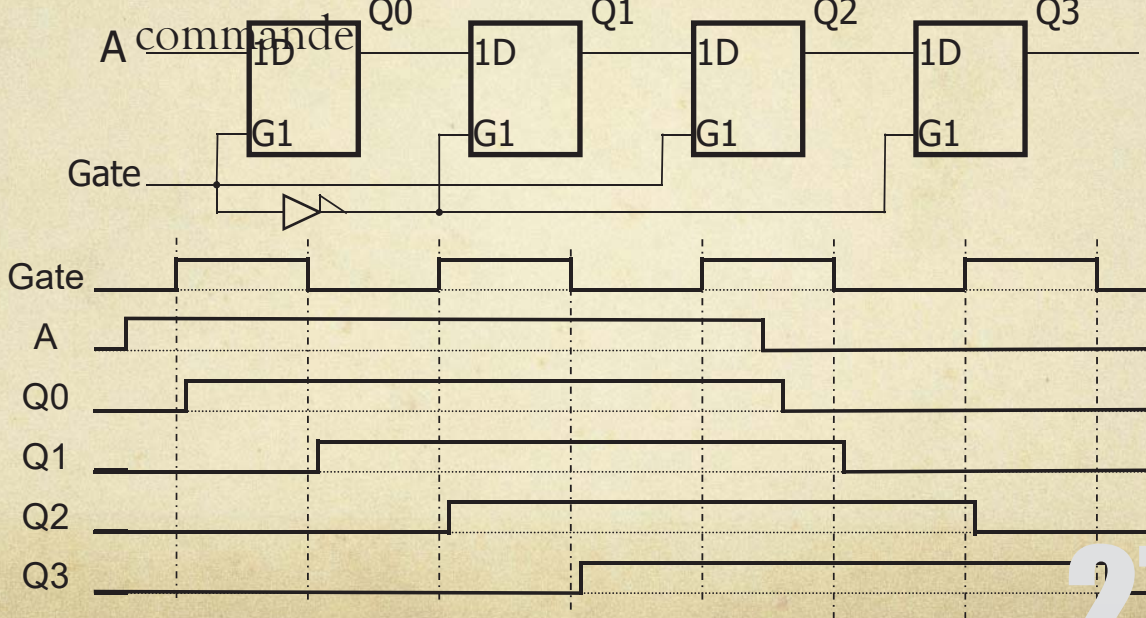
Décalage avec des Latches

Registre à décalage avec des Latches



Décalage avec des Latches "bis"

Registre à décalage avec des latches et 2 signaux de

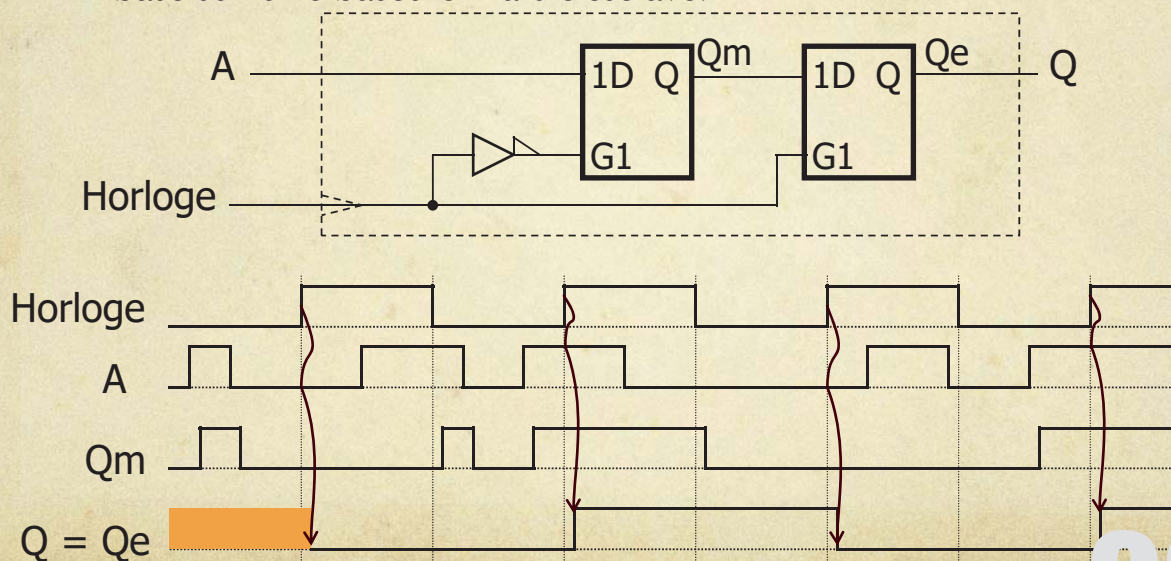


ARO1 - APE & CPN & RMQ

27

Principe de bascule "maître-esclave"

Soit le schéma de principe d'une bascule **sensible au flanc** basé sur une bascule maître-esclave:

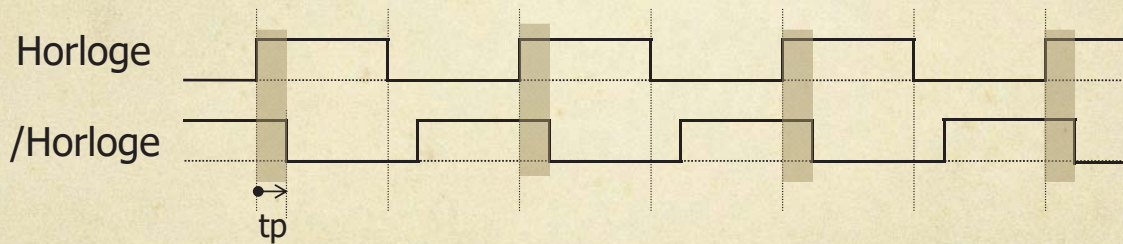
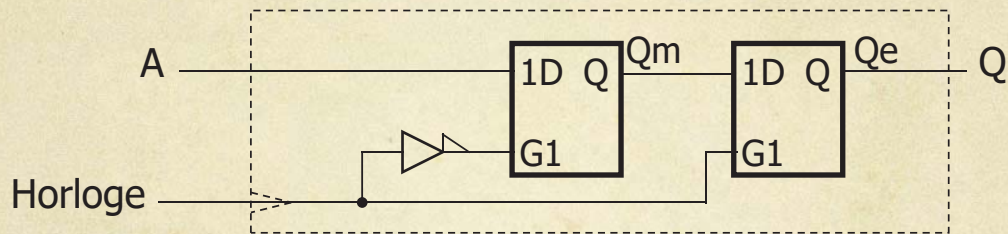


ARO1 - APE & CPN & RMQ

28

Principe de bascule "maître-esclave"

Problème dans le fonctionnement de ce schéma:



Zone où les 2 latches sont ouverts. Risque que l'état de l'entrée A passe à travers les 2 latches simultanément !

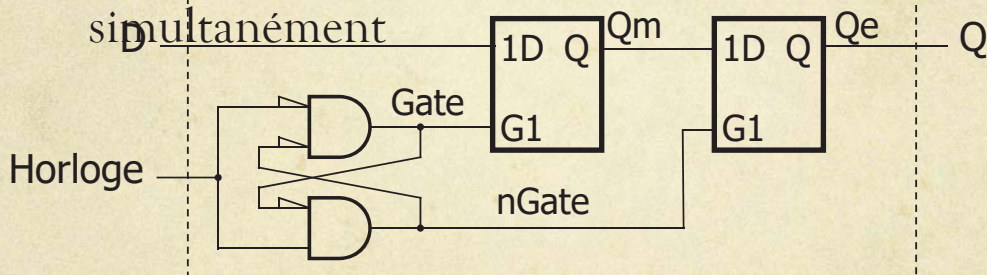
29

Bascule "maître-esclave"



Bascule constituée de 2 Latches avec 2 signaux gates inversés

ces deux signaux ne doivent pas être actifs simultanément



30

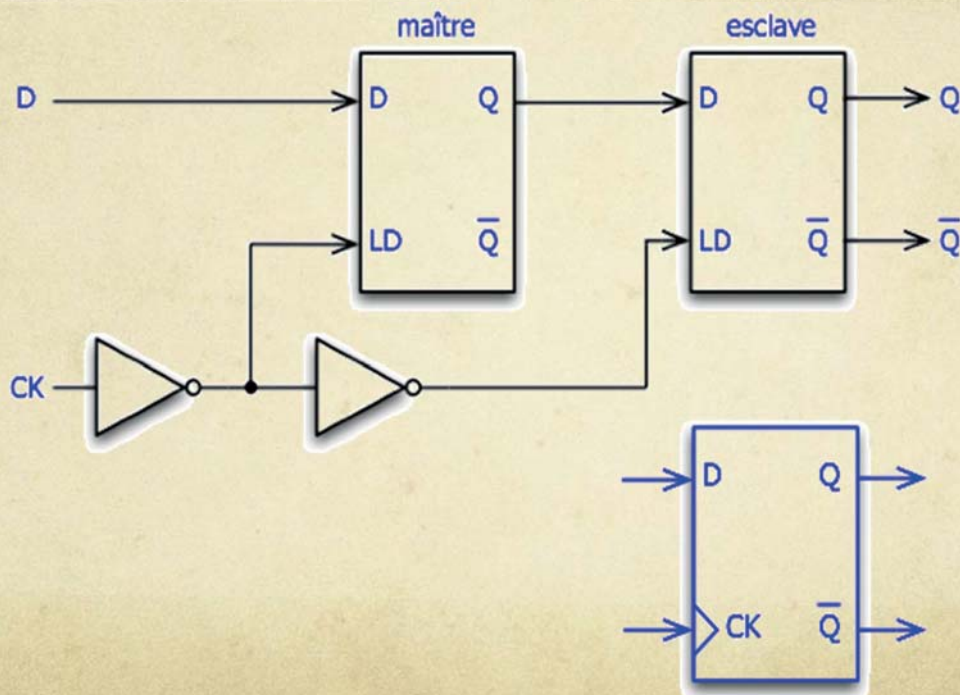
Bascule "maître-esclave"

Analyse du fonctionnement interne



Le comportement correspond à une **bascule sensible au flanc**

Bascule bistable D (*flip-flop*)



Bascule bistable D (*flip-flop*)

- Si CK=0, l'élément maître laisse passer l'entrée D à sa sortie, mais la sortie de l'élément esclave reste inchangée. Lorsque CK=1, la sortie de l'élément maître est conservée, et passe à la sortie de l'élément esclave
- La sortie de la bascule peut changer seulement **pendant la montée du signal d'horloge CK**
- La sortie de la bascule prend la valeur présente à l'entrée D au moment de la montée du signal d'horloge

NB: Le signal d'horloge peut être : CK ou CLK

33

Bascule bistable D (*flip-flop*)

- La **table d'états** de la bascule est donc:

D	Q	Q ⁺
0	0	0
0	1	0
1	0	1
1	1	1

Cette table montre la valeur de la sortie future en fonction de l'entrée et de l'état présent

- Le comportement de la bascule peut être décrit également par son équation caractéristique:

$$Q^+ = D$$

$$Q \leftarrow D$$

(la sortie Q prend la valeur de l'entrée D au prochain flanc montant de l'horloge)

34

Flip-flop D (DFF)

Table des transitions

Horloge	D	Q+
⌋	0	0
⌋	1	1

⇒

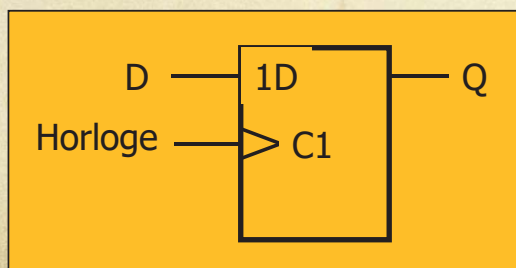
Table des

D	Q+
0	0
1	1

S:

Signal d'horloge implicite

Symbole CEI



Polarité flanc

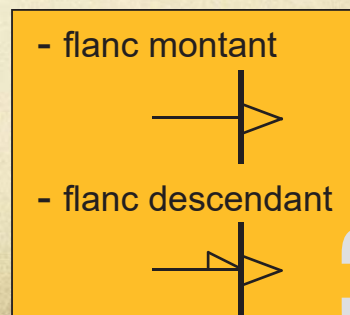
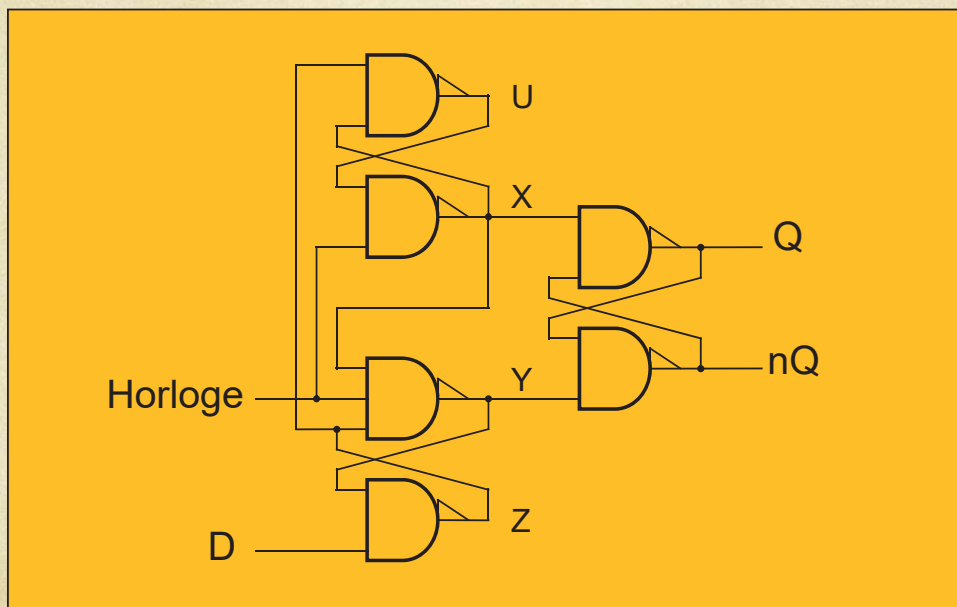
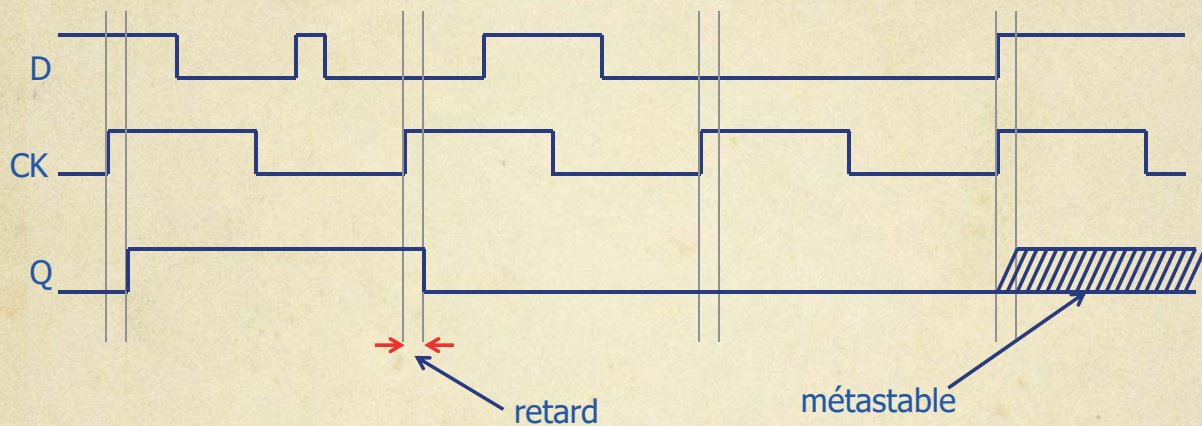


Schéma interne flip-flop D (DFF)



Bascule bistable D (*flip-flop*)



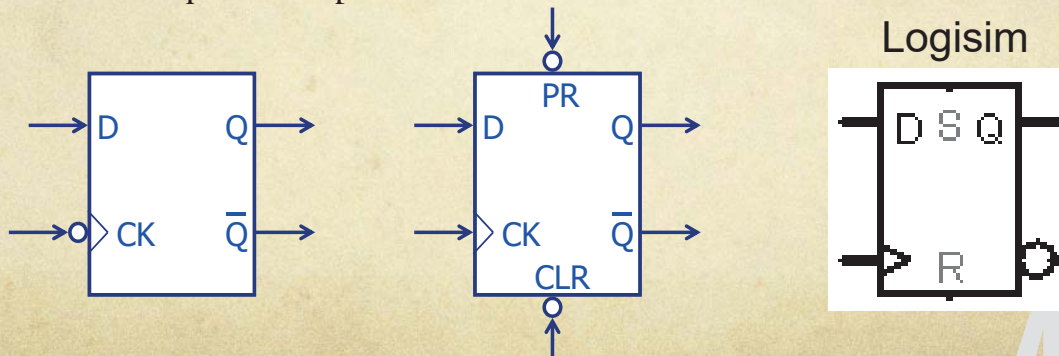
ARO1 - APE & CPN & RMQ

39

Bascule bistable D (*flip-flop*)

- Il existe également des bascules D sensibles au flanc descendant de l'horloge
- La bascule D peut avoir des entrées asynchrones, qui forcent la sortie à 1 ou 0, indépendamment des entrées D et CK:
 - PR ou S (*preset ou set*) met la sortie à 1
 - CLR ou R (*clear ou reset*) met la sortie à 0

Dans la mesure du possible, ces entrées asynchrones sont utilisées uniquement pour l'initialisation ou le test des bascules

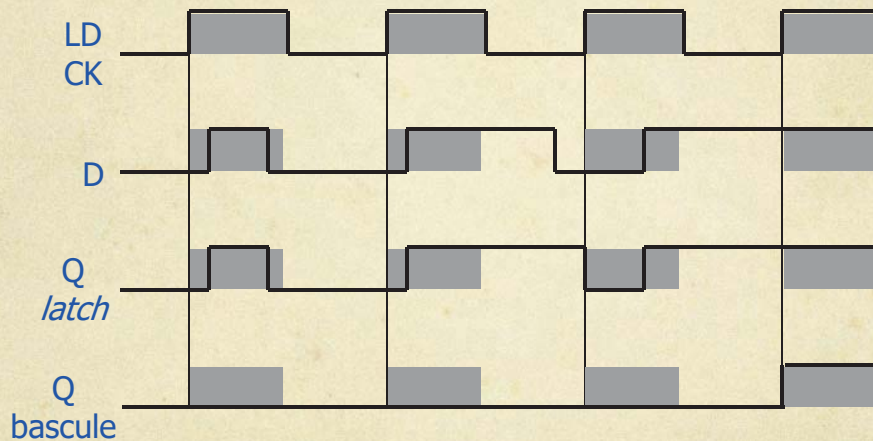


ARO1 - APE & CPN & RMQ

40

Bascule bistable D (*flip-flop*)

Différence de comportement entre le latch et la bascule:



ARO1 - APE & CPN & RMQ

41

Flip-flop T (toggle)

- Bascule sensible au flanc :
 - Si $T = 0$ alors la sortie est maintenue
 - Si $T = 1$ alors la sortie Q est inversée

Table des transitions synchrones

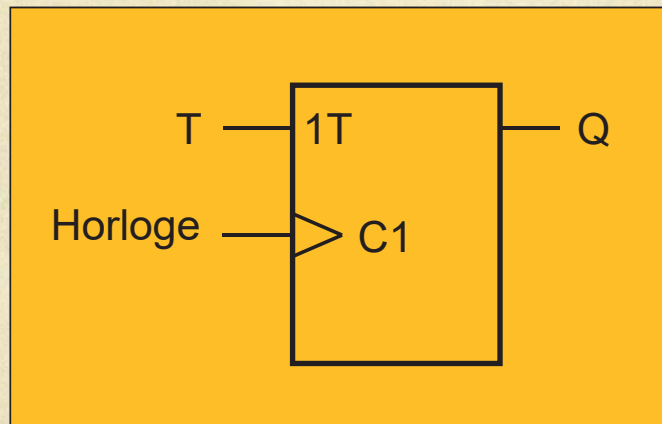
T	Q+
0	Q
1	not Q

ARO1 - APE & CPN & RMQ

42

Flip-flop T (toggle)

- Symbole CEI :



Flip-flop JK

- Bascule sensible au flanc :
 - si $J = K = 0$ la sortie conserve sa valeur (hold)
 - si $J = 1$ et $K = 0$ la sortie prend la valeur 1 (set)
 - si $K = 1$ et $J = 0$ la sortie prend la valeur 0 (reset)
 - si $J = K = 1$ la sortie prend l'état inverse (toggle)

Table des transitions synchrones

J	K	Q+
0	0	Q
0	1	0
1	0	1
1	1	not Q

... flip-flop JK ...

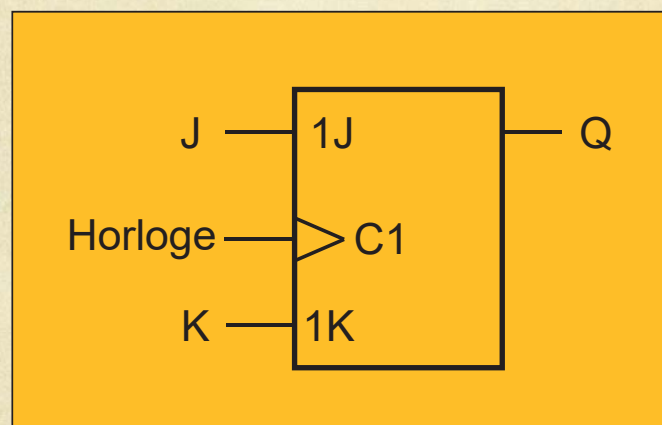
- Autre table des transitions synchrones pour flip-flop JK

Etat présent	Etat suivant	J	K
'0'	'0'	'0'	'1'
'0'	'1'	'1'	'1'
'1'	'0'	'1'	'0'
'1'	'1'	'0'	'0'

45

Flip-flop JK

- Symbole CEI :



46

Types de bascules bistables / entrées

- Asynchrone :
 - type SR (Set, Reset)
- Latch (sensible au niveau):
 - type SR
 - type D
- Flip-flop (sensible au flanc):
 - type D (delay)
 - type DFFE (delay with enable)
 - type JK (préfér  au SR)
 - type T (toggle)