

Archi CH2: Circuits Séquentiels

1] Introduction:

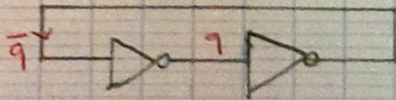
- * Circuit Séquentielle a pour but de mémoriser d'information
- * Élément séquentielle:
 - un élément séquentielle conserve un état, ainsi un circuit séquentielle n'est pas une fonction (\neq le circuit combinatoire)
 - \Rightarrow Impossible à représenter sous forme de table de vérité
 - n entrées, m éléments séquentiels $\Rightarrow 2^{n+m}$ cas
 - Éléments séquentiels contiennent des variables.
 - La séquentialité est importante:

$$a \Leftarrow 1 \text{ seq } b \Leftarrow a \neq b \Leftarrow a \text{ seq } a \Leftarrow 1$$
 - Représentation: On utilise des diagrammes de temps ou chronogrammes décrivent graphiquement une séquence particulière du fonctionnement

2) Elemente Sequentials: Versus D, Brücke D:

i) Inverse couples:

- C'est l'élément le plus simple qui peut mémoriser l'information :

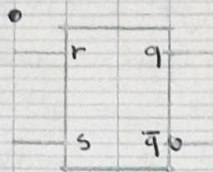


- a si le nombre d' inverses est impair \Rightarrow Comportement oscillatoire
 " " " " " pair \Rightarrow " stable.

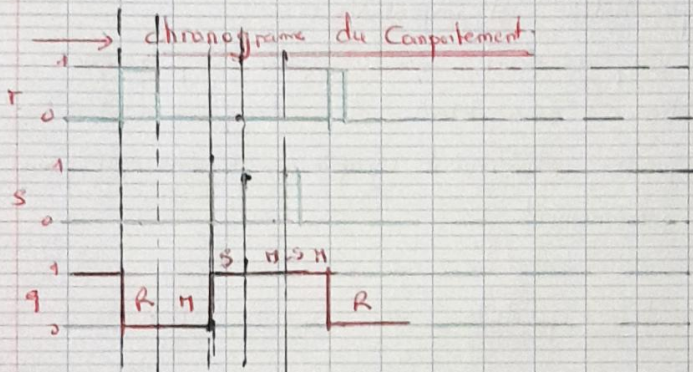
ii) Verrou Reset/Set: (RS)

→ Fonctionnement:

- Permet de forcer une valeur et de la conserver
- $\times r \leftarrow 1$: "mise à 0 du Verrou" (reset)
- $\times s \leftarrow 1$: "mise à 1 du Verrou" (set)
- r et s jamais à 1 simultanément.

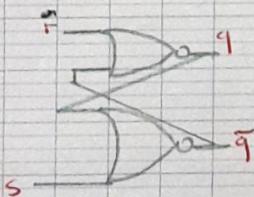


r	s	q	q̄	
0	0	q'	q̄'	⇒ M = (memory)
0	1	1	0	⇒ S = (set)
1	0	0	1	⇒ R = (reset)



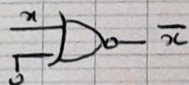
→ Implantation:

- Verrou Reset/set = Porte nor complée



Résumé:

- Porte nor:



En effet:

a \ b	0	1	nor
0	1	0	1 = 0
1	0	0	0 = 1

⇒ Si $r = s = 0 \rightarrow$ On a bien une mémorisation

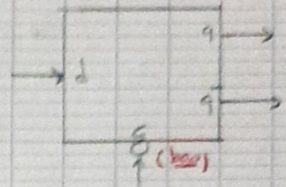
iii) Verrou D: (D latch)

→ fonctionnement:

- Échantillonnage entrée de donnée (d) sur Niveau de Commande (C)
 - * d : entrée de donnée
 - * q : Sortie de donnée
 - * C : entrée de commande (ou e, ou parfois g)

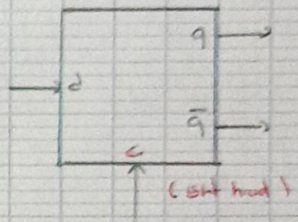
• i) si C active à l'état bas:

$$q = \begin{cases} d & \text{si } C = 0 \\ q' & \text{si } C = 1 \text{ (Mémorisation)} \\ & q' = q \end{cases}$$



ii) si C active à l'état haut:

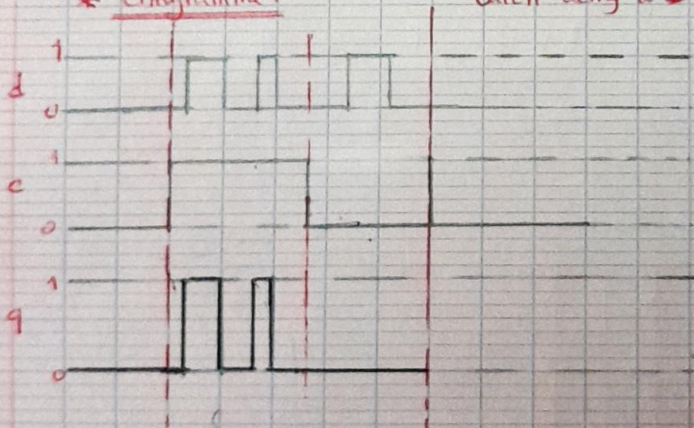
$$q = \begin{cases} d & \text{si } C = 1 \\ q' & \text{si } C = 0 \text{ (mémorisation)} \\ & q' = q \end{cases}$$



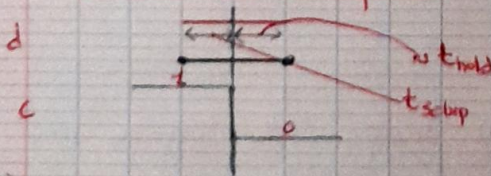
→ Comportement temporel:

* chronogramme:

Latch actif à 1



* Conditions à respecter:



* d doit être stable lorsque C s'active.

* les chronogrammes montrent que q change en même temps.

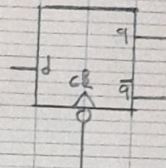
iv) Bascule D (D flip-flop):

→ Fonctionnement:

- Échantillonne entrée de donnée sur événement de commande
 - * d: entrée de donnée
 - * q: sortie de donnée
 - * ck: entrée d'horloge (aussi appelé clk)

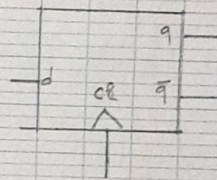
- i) si ck active sur front descendant:

$$q = \begin{cases} d & \text{si ck subit } 1 \rightarrow 0 \\ q' & \text{si non} \end{cases}$$



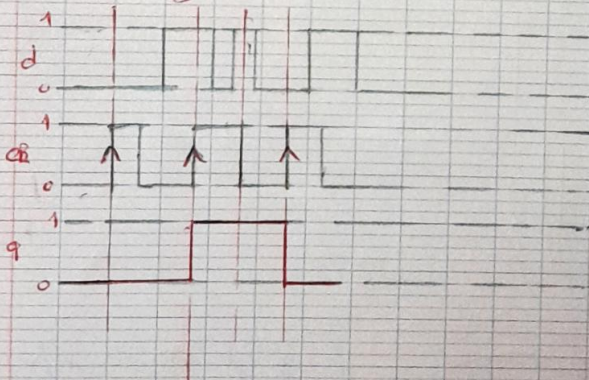
- ii) si ck active sur front montant:

$$q = \begin{cases} d & \text{si ck subit } 0 \rightarrow 1 \\ q' & \text{si non} \end{cases}$$



→ Comportement temporel:

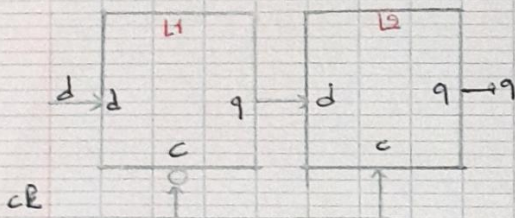
* chronogramme:



* Conditions à respecter:

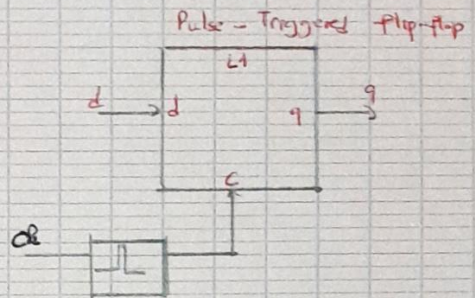
- d doit être stable lorsque ck affecte 0/1
- ck affecte 0/1 \Rightarrow q prend la valeur que d possède avant de finir
- chronogramme montrant ck et q changeant en même temps
- $t_p(ck, q) > t_{hold}$: pour échantillonnage correct

→ principes d'implantation :



Master-slave flip-flop

(Re : Attention à Q_1 et Q_2)

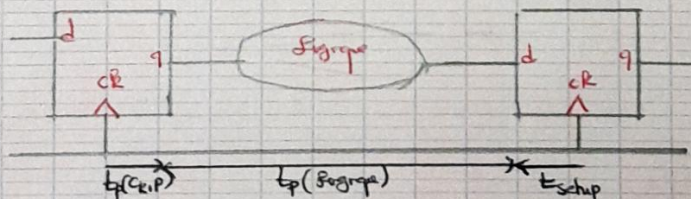


Re :

- Verrous et blocs D peuvent avoir :
 - * une entrée de reset (ou clear)
 - * une entrée de set (ou preset)

3] Circuits Séquentiels :

- * Circuits séquentiels = Intègrent des éléments mémorisants et des portes logiques.



- * Période minimale de l'horloge :

$$t_{CKmin} = t_p(CKIP) + t_{logique} + t_{setup} + t_{skew}$$

avec

t_{skew} : décalage temporel entre les 2 entrées CK

(Ex : CPU à 4 GHz $\Rightarrow t_{CKmin} = 250$ ps)

Re :

Il faut que : $t_p(CKIP) + t_{logique} - t_{skew} > t_{hold}$.