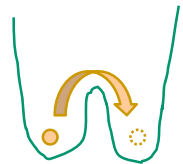
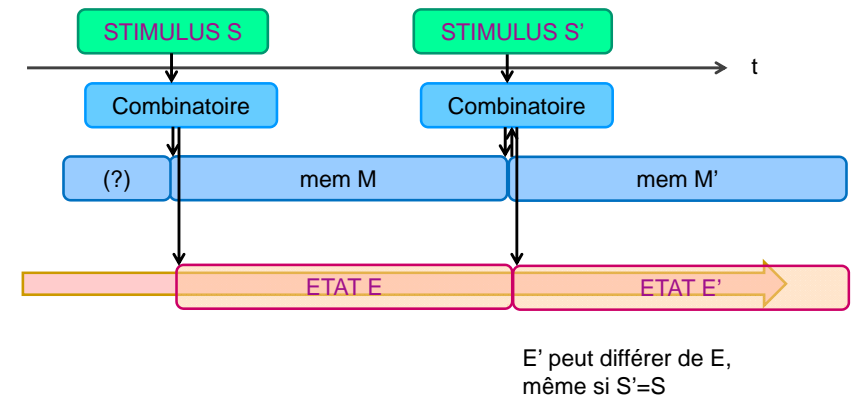
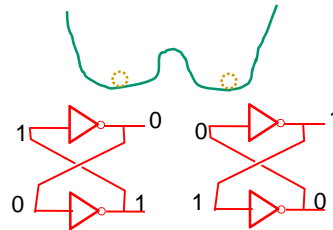
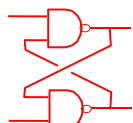
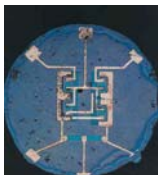


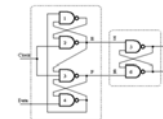
- **Besoin de mémoire**
 - Circuits ayant une « histoire »,
 - sortie dépendant des états antérieurs
- **Bascule à deux éléments**
- **Bascule réelle (bascule D)**
 - Modes de fonctionnement (preset, CLK, sync, async)
- **Bascules multiples**
 - Registre à décalage
 - Génération de séquences
 - Conversion série/parallèle
 - Mémoire, calculs itératifs (cryptage), multiplication binaire.
- « **Timers** », Circuits monostables et astables, retriggerables, ...
- *Un mot sur les mémoires dynamiques et statiques ?*
- *Un mot sur l'architecture de processeur de von Neumann (ALU) ?*
- *Un mot sur la récupération d'horloge ?*



→ **Physiquement : un système bistable qui a besoin d'énergie pour changer de puits de potentiel**



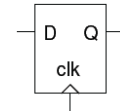
→ **Commande (~) par les deux pattes restantes (bascule dite « RS »)**



→ Zoologies des bascules ('RS', 'JK', 'D')

→ Celle qui est le plus proche d'une logique « signal » : la bascule **D** (et qui évite un cas d'incertitude...)

→ Entrée = **CLK, D**
→ Sortie = **Q**, complétée par \bar{Q}



Fonctionnement :

- A chaque front montant de l'horloge **clk** (passage de 0 → 1) l'entrée **D** est copiée (échantillonnée, mémorisée) sur la sortie **Q**.
- Entre deux fronts d'horloge, la sortie **Q** ne change pas.

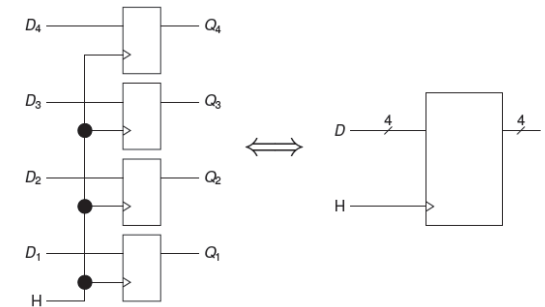
BASCULE D : table de vérité

D	clk	Q	
0	↑	0	copie de D sur Q
1	↑	1	copie de D sur Q
x	0	Q	Q conserve sa valeur
x	1	Q	Q conserve sa valeur
x	↓	Q	Q conserve sa valeur

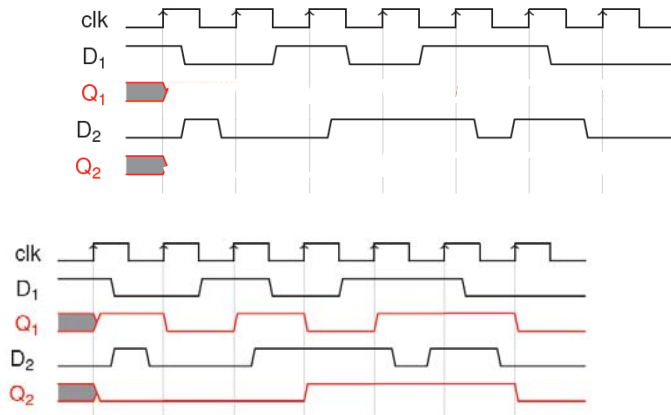
Registre = N bascules D

Un registre est un ensemble de bascules fonctionnant en parallèle.

- Exemple un registre 4bits

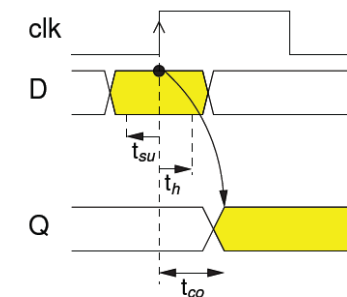


Exemple



- Q₁ recopie D₁ avec un cycle de retard.
- Q₂ recopie D₂ en filtrant les impulsions de durée inférieure à la période de l'horloge.

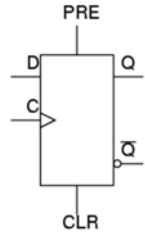
Contrainte de fonctionnement sur bascule D



- La donnée doit être stable au front d'horloge :
 - avoir atteint sa valeur t_{su} avant le front d'horloge (setup).
 - cette valeur doit être maintenue t_h après le front d'horloge (hold).
- La copie de l'entrée sur la sortie se fait avec un retard de t_{co} (clock to output).

Contrôle asynchrone de la bascule D

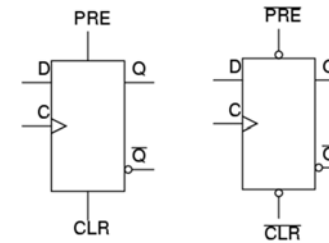
- Un signal extérieur est utilisé pour forcer l'état des bascules.
- On parle de "reset" (remise à zéro). ou de « CLEAR »
- Il peut s'agir d'une mise à "1", on parle alors de preset.



→ entrées
asynchrones :
Forcent Q
immédiatement à 0 ou à 1
sans attendre le front d'horloge

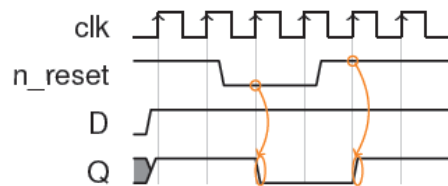
Contrôle asynchrone de la bascule D

Quelle différence ?



Contrôle « synchrone » de la bascule D ?

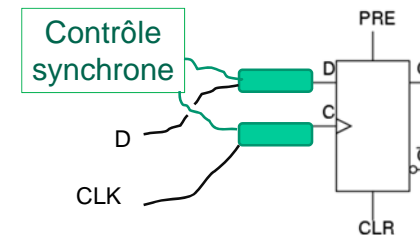
→ résultat souhaité:



- Vient de la logique qui précède les bascules.
- Son action n'est effective qu'au front de l'horloge.
- C'est une remise à zéro fonctionnelle.

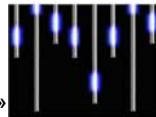
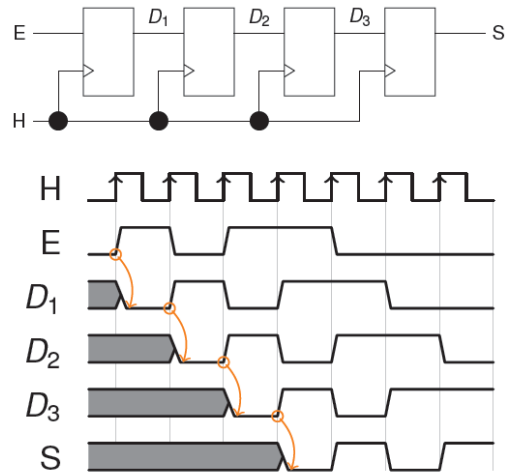
Contrôle « synchrone » de la bascule D ?

→ Juste pour une bascule ?
une logique élémentaire extérieure suffit:



→ Mais pour une combinaison
de bascules comme celles
qu'on va voir, cette idée de
«reset synchrone» est
pertinente

Application : registre à décalage (shift register)

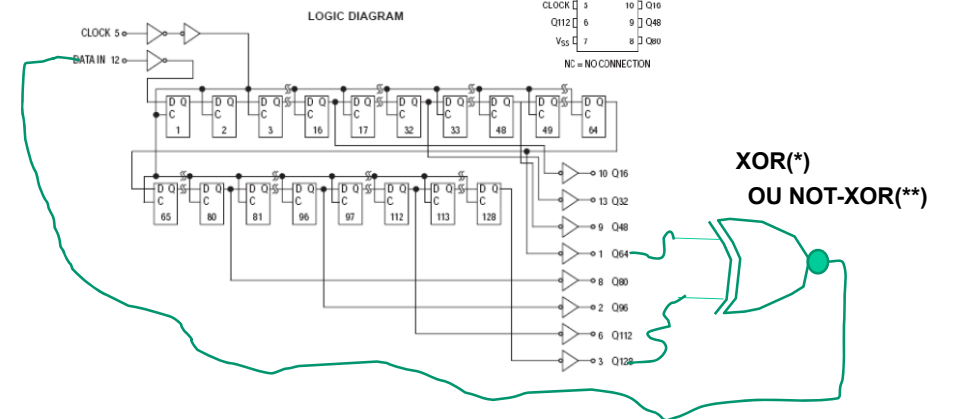


version « les LEDs en 2015 » boîte de nuit vintage : le « chenillard »

Registre à décalage / génération de séquence

Pourquoi peut-il être intéressant d'avoir certaines sorties d'un long registre ?

Exemple de registre : MC14562
128-bit shift register

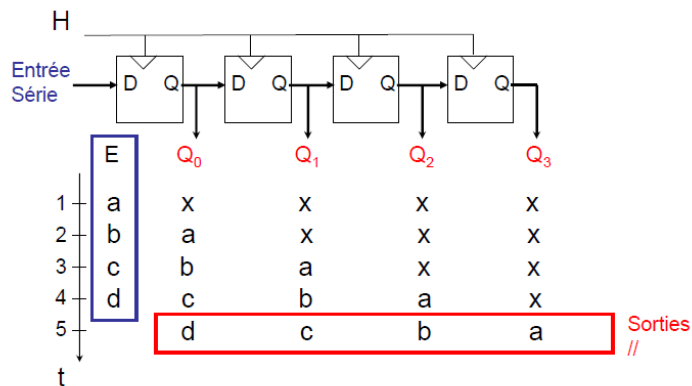


(*) Dans ce cas, l'état « 1 partout » est stable, le système décrit les $2^n - 1$ autres états

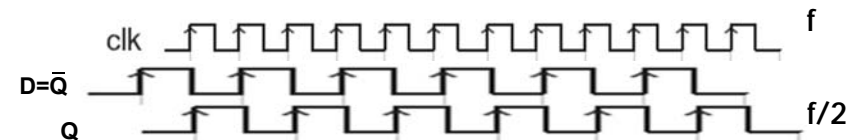
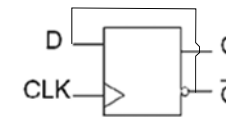
(**) Dans ce cas, l'état « 0 partout » est stable, le système décrit les $2^n - 1$ autres états

Conversion série → parallèle

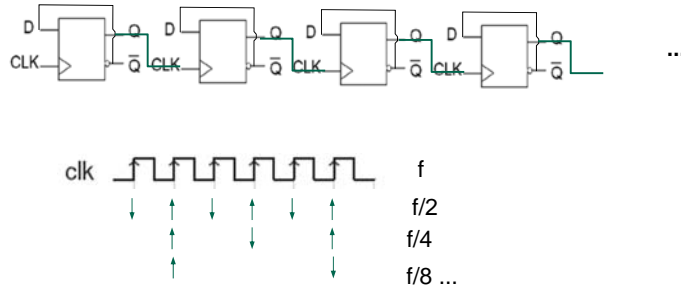
Registre à décalage à droite : conversion série //



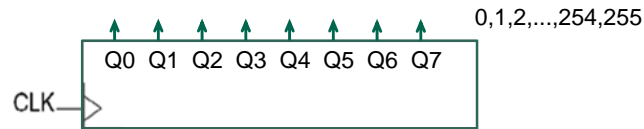
Division de fréquence par 2



Division de fréquence par 2^n



C'est le coeur d'un compteur en binaire !



Compteurs

• Basique = comptage binaire sur n bits

→ Plus des fonctions comme « reset »

→ Ou mise à 1 ;

ou mise de toute la sortie à une valeur de départ.

Ce qui se fait en adressant

en parallèle les CLR/SET des bascules D.

→ Ou: choix du sens : « up/down counter »

Ce qui se fait par exemple

en aiguillant (demux) la cascade par Q ou par le complémentaire

→ Ou: on ne compte pas en binaire

mais en « BCD »

→ A quoi ressemble un « pulse-mètre » qui

donne la durée d'un créneau (~pulse)

en multiple d'un temps de base ?

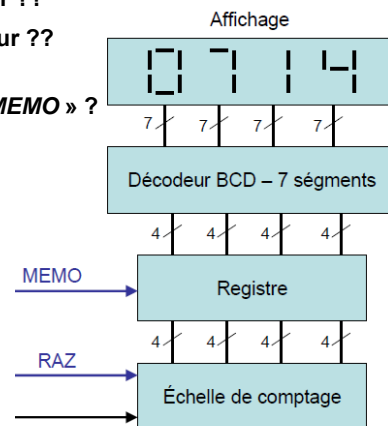
(base de beaucoup de mesures physiques autre que durée)

Pulse-mètre ?

Le pulse va sur ??

L'horloge va sur ??

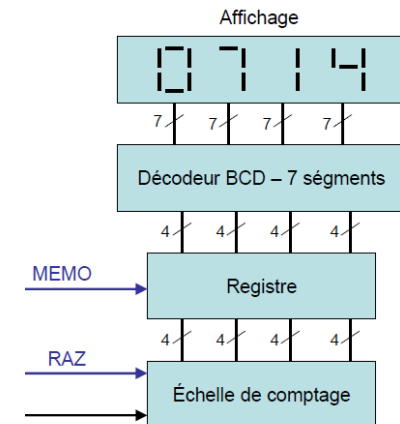
A quoi sert « MEMO » ?



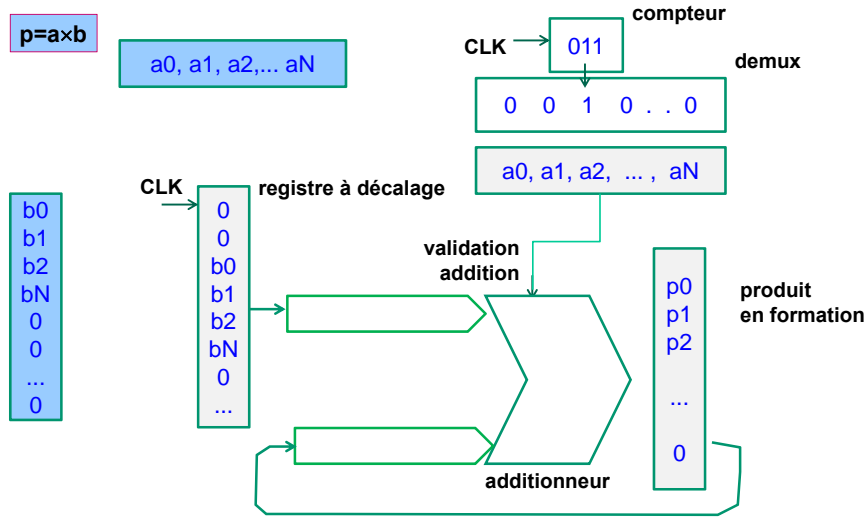
Périodemètre ?

Un pulse de largeur 1 s va sur ??

L'horloge va sur ??

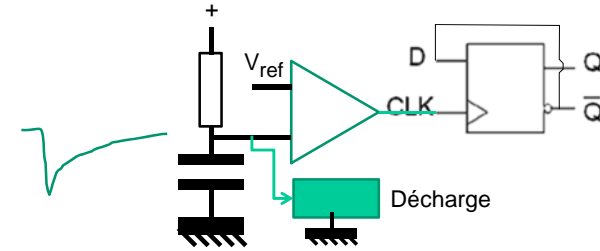


Multiplication binaire



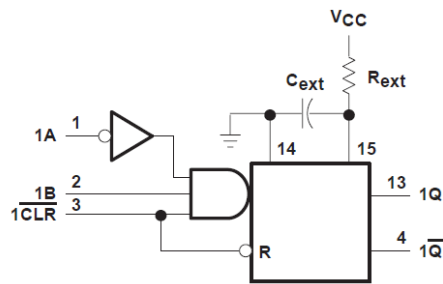
On peut rendre cette opération rebouclée bien plus générale (p ex : pour le cryptage).

Timers, « mutivibrateur astable et monostable »



- Circuits de base de la temporisation (TTL, CMOS, ou le vieil inclassable NE555)
- Incorporent une partie analogique, minime mais « précise ».
- Ont une importante logique de contrôle (décharge capa..)
Une question classique étant « c'est Retriggerable ou pas ? »
- Aussi capables de fournir un oscillateur (« astable »)

Exemple : un demi-74LS123



FUNCTION TABLE				
CLEAR	A INPUT	B INPUT	Q	Q̄
L	X	X	L	H
X	H	X	L [†]	H [†]
X	X	L	L [†]	H [†]
H	L	↑	LHL [‡]	HLH [§]
H	↓	H	LHL [‡]	HLH [§]
↑	L	H	LHL [‡]	HLH [§]

[†] These lines of the functional tables assume that the indicated steady-state conditions at the A and B inputs have been set up long enough to complete any pulse started before the setup.

[‡] This is a low-to-high-to-low pulse.

[§] This is a high-to-low-to-high pulse.

Exemple d'application basique

Delayed-Pulse Generator With Override

In Figure 11, the first one-shot (OS₁) determines the delay time by preselected values of R_{ext1} and C_{ext1}. The second one-shot (OS₂) determines the output pulse duration by preselected values of R_{ext2} and C_{ext2}. The output pulse can be terminated at any time by a positive rising pulse into the override input.

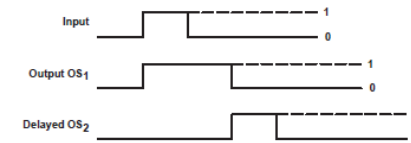
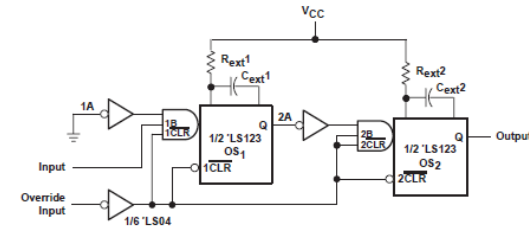


Figure 11. Delayed-Pulse Generator with Override