

## Chap 2 Systèmes séquentiels

### 2.1 Combinatoire et séquentiel

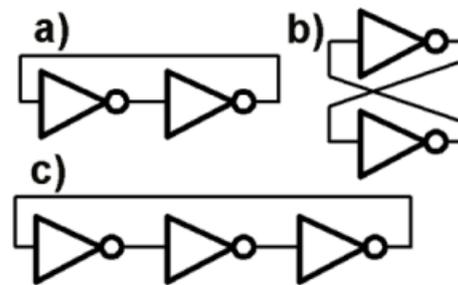
Les systèmes combinatoires ont des sorties qui ne dépendent que de l'état des entrées (les délais sont supposés nuls). Le temps est un nouveau paramètre dans les systèmes séquentiels; on considère un état futur qui dépend de l'état présent, avec une évolution synchrone (à un instant d'horloge) ou asynchrone (immédiatement, mais avec les délais induits par la technologie).

Que penser des ces schémas pathologiques?

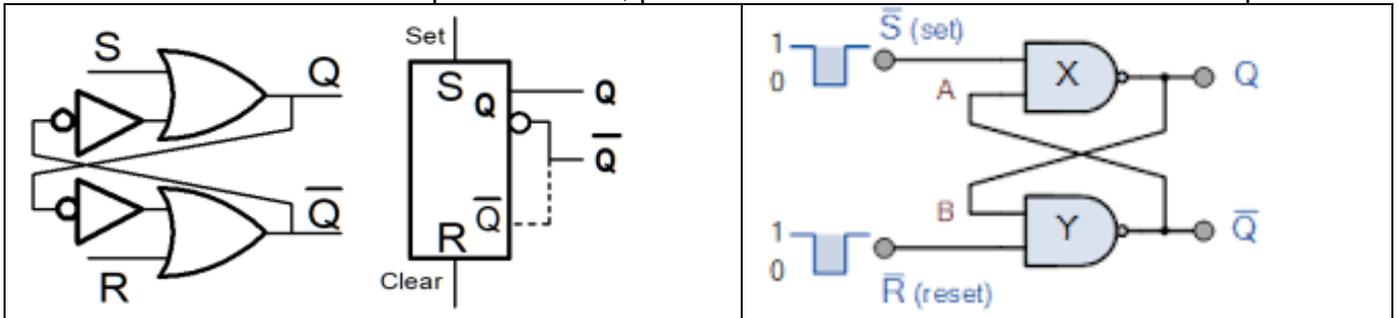
Dans le cas a), supposons que l'entrée à gauche est à "1". La sortie est à zéro et le signal de nouveau inversé maintient la première entrée à "1". On a construit un élément de mémoire, qui garde l'état pris à la mise sous tension.

En b), seul le dessin a changé.

Le cas c) n'est pas stable. Il oscille si le temps de basculement est suffisant. En augmentant le nombre impair d'inverseurs en série, on réduit la fréquence d'oscillations. Mais ce n'est pas le bon moyen pour faire un oscillateur!



Pour faire basculer l'élément mémoire que l'on vient de voir, on peut forcer du 5 volts sur l'une ou l'autre des entrées; le court-circuit dure le temps de basculement. Une porte OU fait cela plus proprement. L'inverseur peut être intégré dans la porte (porte NOR). Le schéma usuel d'une bascule RS utilise des portes NAND, plus efficaces dans les circuits TTL historiques..



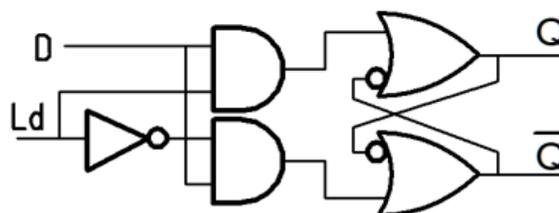
On comprend que S (set) et R (reset) font basculer. Mais S et R ne doivent pas être actifs en même temps, ou plus exactement ne doivent pas retourner à zéro simultanément; dans ce cas, Q va mettre un temps imprévisible pour décider dans quel sens basculer. C'est le même problème avec des bascules mécaniques, mais elles sont trop mal construites pour qu'on le voie.

### 2.2 Verrou statique

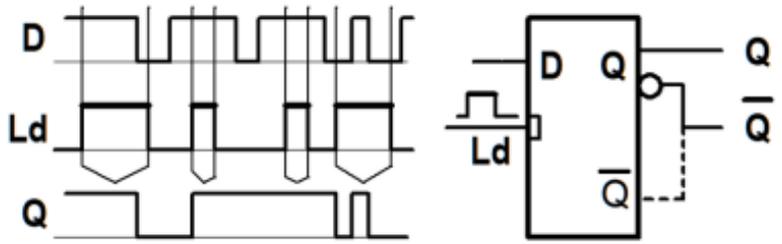
Pour mémoriser un bit (en général il y en a plusieurs à mémoriser simultanément), il faut une bascule avec une entrée D (data) et un signal unique qui dit quand mémoriser.

Ce signal peut être statique ou dynamique, comprenons bien la différence.

Avec un verrou (latch en anglais) un signal Ld (load) copie l'entrée sur la sortie quand il est actif, et bloque l'information si inactif. On reconnaît la bascule SR précédée d'un aiguillage qui dévie l'entrée D du côté S ou R.

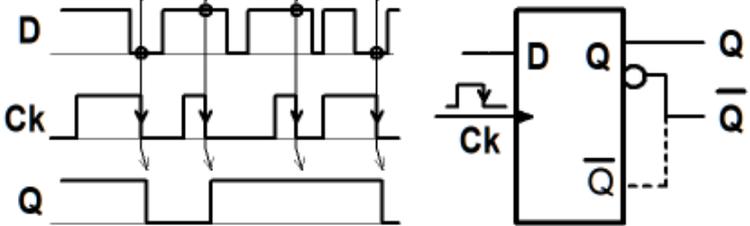


Le symbole du verrou montre un petit rectangle pour distinguer l'entrée Ld qui "charge" la bacle.  
 Le terme de "transparent latch" évoque bien la fonctionnalité: quand Ld=1, la sortie est la copie de l'entrée. Quand Ld=0, l'état est bloqué.



### 2.3 Bascule D dynamique

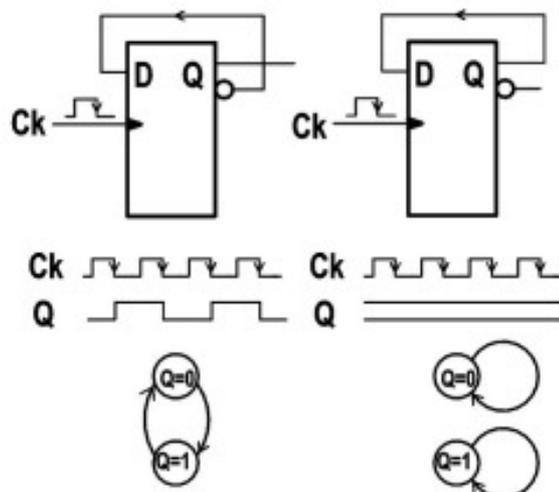
Avec une bascule D (D-flip-flop), c'est la transition du signal Ck (Clock), qui est active. Le schéma avec des portes est plus compliqué et ne nous apporte rien.  
 Très important, le triangle qui caractérise l'entrée dynamique CK. On peut dire que le front descendant de CK fait une "photo instantanée" de l'entrée D.



Le verrou par contre garde l'obturateur ouvert, et la dernière image est bloquée quand l'obturateur se ferme.

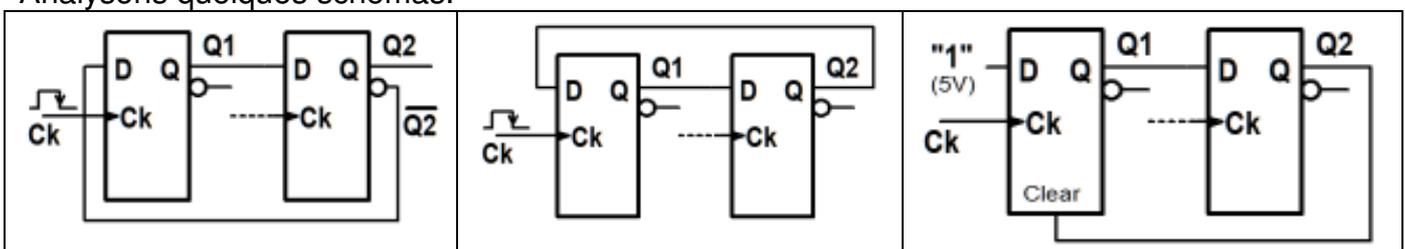
On remarque que les symboles logiques doivent porter le plus d'information possible sur leur nature. Un rectangle sur une entrée, c'est un Load statique; un triangle plein sur une entrée, c'est le Ck d'une bascule dynamique. La technologie des circuits "74" forçait à distinguer plus de symboles; notre triangle plein remplace le triangle vide avec rond d'inversion des circuits TTL.

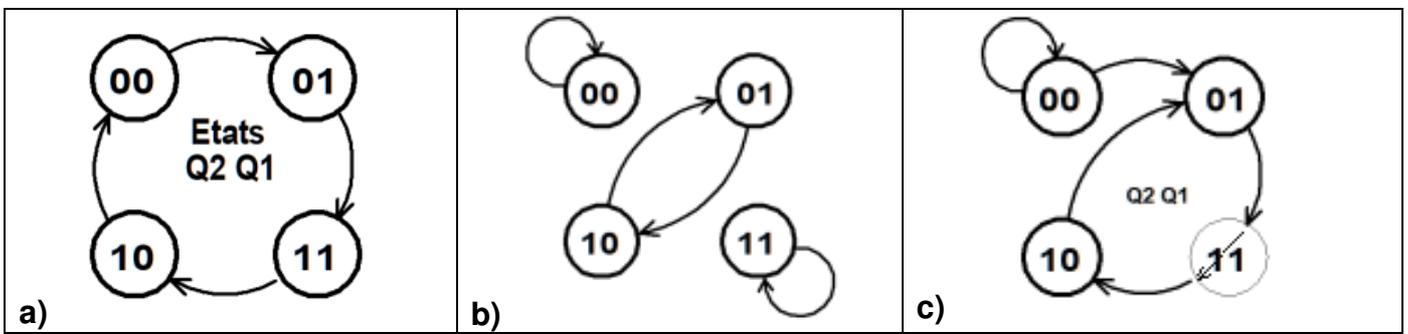
En reliant l'entrée à la sortie inversée, l'état change à chaque coup d'horloge. On a un diviseur par 2.  
 Si la sortie Q est reliée à l'entrée, le système n'évolue pas; il a deux états d'équilibre stables.  
 Que se passe-t-il si on relie Q ou /Q à une entrée Set ou Clr? Pour répondre il faut avoir le schéma et des spécifications électriques s'il y a risque d'oscillations.  
 Expérimenter en observant si une LED branchée sur la sortie Q est allumée ou éteinte n'a aucune valeur scientifique.



### 2.4 Diagrammes d'états

Avec 2 bascules, il y a 4 états. La décision concernant l'état futur se fait à l'instant d'horloge. Analysons quelques schémas.





On a en a) le compteur par 4 en code Gray. On remarque qu'un seul bit change à chaque transition, ce qui est très intéressant dans certaines applications.

On voit dans le schéma b), que des machines logiques peuvent avoir des "états pièges"; on peut s'y trouver suite à une perturbation et il faudra réinitialiser.

On voit dans les cas c) une machine semi-synchrone avec un état 11 qui agit statiquement pour passer à un autre état avec un décalage temporel supplémentaire.

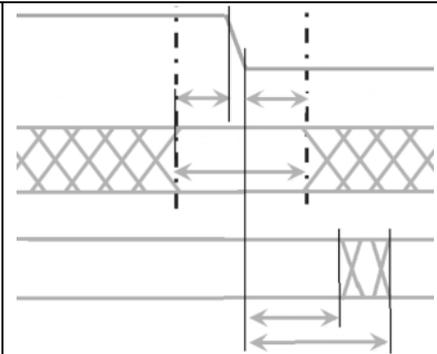
On voit aussi que nommer les bascules  $Q_i$   $Q_2$  dans cet ordre sur le schéma, et utiliser l'ordre inverse dans les diagrammes amène une difficulté supplémentaire qui est source d'erreurs.

## 2.5 Autres représentations

Le diagramme des temps et l'organigramme ne sont pas les seuls moyens de représenter graphiquement l'évolution d'un système dans le temps selon l'état des entrées. Un organigramme, un structogramme, un grafcet, du pseudo-langage facilitent l'analyse d'un comportement avant de le coder dans le langage choisi.

## 2.6 Note technologique

Une contrainte technologique veut que le front d'horloge soit très vertical (une valeur maximale en nanosecondes est spécifiée pour chaque technologie). L'information en entrée doit être stable un certain temps autour du front d'horloge pour que l'échantillonnage soit correct, et une fourchette de temps dit quand la sortie est à jour. On doit vérifier en cascasant les circuits que les fourchettes de temps sont compatibles dans les cas pires. C'est le "worst case design" grâce auquel le fonctionnement des ordinateurs est si fiable.



## 2.7 Programmer un verrou et une bascule D

Une bascule physique a deux entrées et une sortie (la sortie inversée est une commodité historique qui ne compte pas fonctionnellement). On peut la simuler par programmation et la tester dans un environnement avec des entrées sur une carte Arduino ou comme bouton dans Android. Il faut définir des variables ou fonctions  $Ld$ ,  $D$ ,  $Q$  qui ne prennent que deux valeurs booléennes 0,1.

Le corps d'un programme en C qui programme un verrou (latch) est alors le suivant:

```
void loop () {
  if ((Ld==1) & (D==1)) {Q=1;}
  if ((Ld==1) & (D==0)) {Q=0;}
}
```

Plus simplement

```
void loop () {
  if (Ld==1) {Q=D;}
}
```

Pour une bascule D, la différence fondamentale est que le "Clock" agit au front descendant. On parle d'évènement dans certains langages, de gestion particulière par interruption dans d'autres.

En programmation bloquante, c'est-à-dire que le processeur ne fait rien d'autre dans son programme principal, on attend quand le signal est à zéro, on attend quand il est à 1 et on agit quand il repasse à zéro.

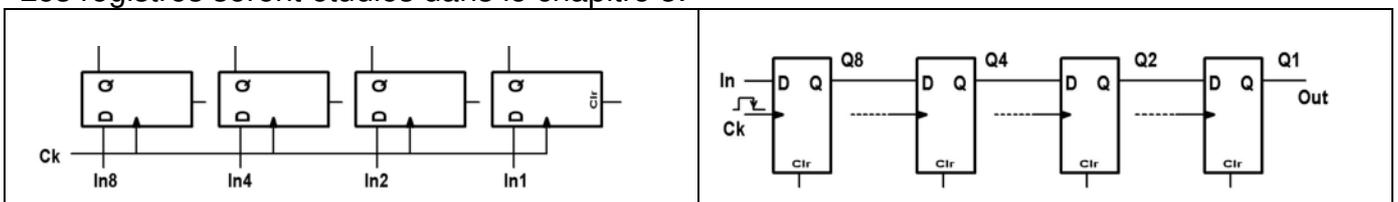
```
void loop () {
  while (Ld==0);
  while (Ld==1);
  { Q=D; }
}
```

Ce serait intéressant de montrer le programme non bloquant et comment on peut agir par interruption.

### 2.8 Registres parallèles et série

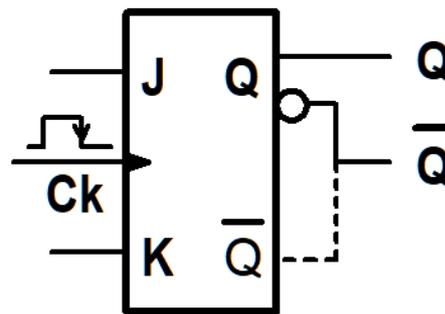
Si une bascule D mémorise un bit, n bascules en parallèle avec la même horloge vont mémoriser n bits simultanément. Le registre parallèle 8 bits, 16 bits, plus est un élément de base de toute machine informatique, et oblige à déclarer la taille des variables et des ports d'entrée-sortie dans la plupart des langages.

Pour transférer l'information à distance, on cascade les bascules pour former un registre série. Les registres seront étudiés dans le chapitre 3.



### 2.9 Bascule JK

La bascule JK a deux entrées *J* et *K* qui "aiguillent" le Ck. Si *J*=1 cela bascule du côté *J*, si *K*=1 du côté *K*, si les deux sont à 1, cela bascule alternativement d'un côté et de l'autre. Une telle description n'est évidemment pas utilisable, mais elle rend la bascule "intuitive". Des entrées statiques Set et Clr permettent sur la plupart des bascules de forcer l'état, pour avoir *Q*=1 ou *Q*=0.



La table de vérité des systèmes séquentiels doit tenir compte de l'état présent, avant l'instant d'horloge, ce qui peut s'exprimer de 2 façons. Le diagramme d'état apporte la même information.

J	K	Q <sub>t+1</sub>
0	0	Q <sub>t</sub>
0	1	0
1	0	1
1	1	$\overline{Q}_t$

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

The state transition diagram shows two states: Q=0 and Q=1. From Q=0, a transition to Q=1 occurs when J=1 and K=0. From Q=1, a transition to Q=0 occurs when J=0 and K=1. Self-loops are present on both states when J=K=0.

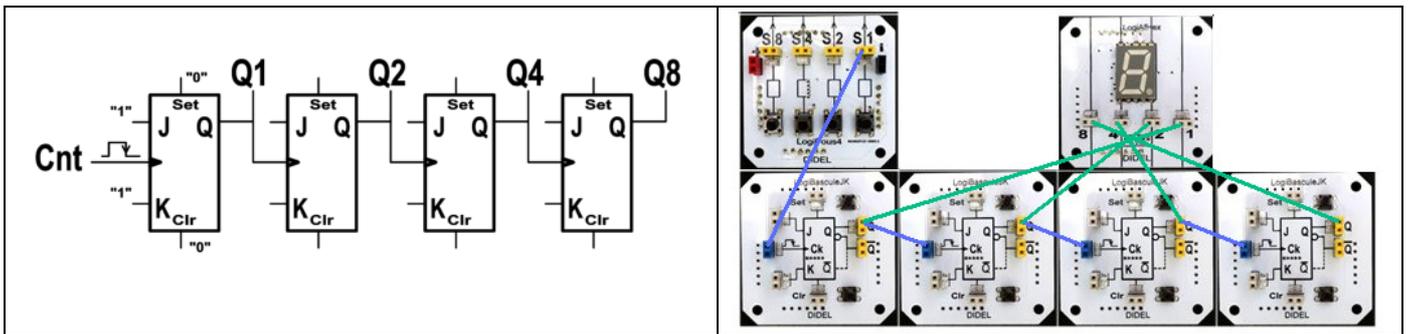
The photograph shows a breadboard circuit for a JK flip-flop using a 74107 IC. The IC is labeled 'LogiBasculeJK'. It has Set and Clr inputs, J and K data inputs, a Ck clock input, and Q and Q-bar outputs. The breadboard also contains a DIP switch and a LED labeled 'DIDEL'.

La bascule D est utilisée essentiellement pour mémoriser et transférer l'information. La bascule JK est plus flexible pour réaliser des séquenceurs et des compteurs.

### 2.10 Compteur binaire

Puisque la bascule JK bascule à chaque CK si *J*=*K*=1, il suffit de cascader 4 bascules pour avoir un compteur binaire par 2<sup>4</sup>=16, qui peut afficher son résultat en hexadécimal. Les entrées

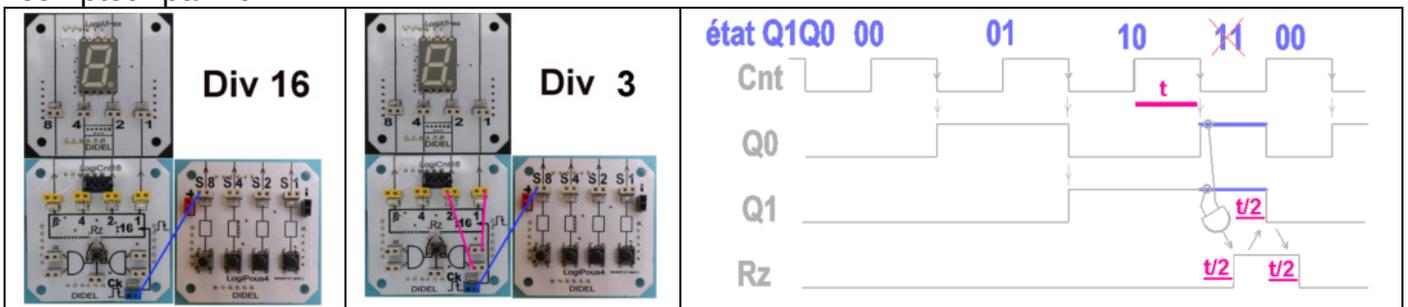
J et K sont par défaut à l'état "1", grâce à des résistances qui n'empêchent pas l'état 0. On peut donc se dispenser de les câbler dans la fonction diviseur par 2.



On remarque que le câblage en cascade crée des retards qui peuvent être gênants. Une logique plus complexes permet des compteurs synchrones, où toutes les bascules ont la même horloge.

### 2.11 Compteur par 16

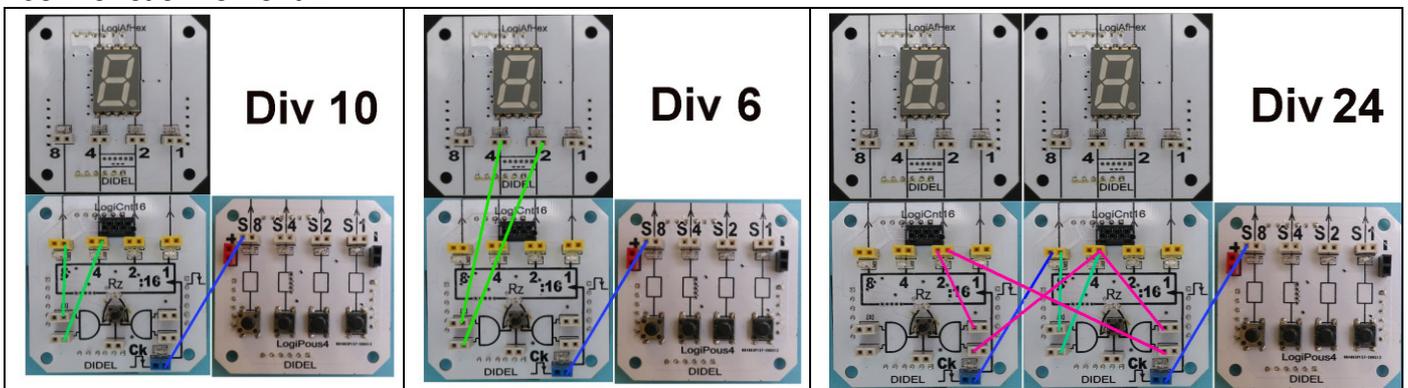
Les compteurs par 16 existent évidemment sous forme de circuit intégré. Le compteur Cnt16 des logidules a un câblage spécialement riche sur l'entrée Reset/cir pour éviter de câbler des portes extérieures dans des câblages fréquents. On voit qu'il y a une porte OU à 4 entrées sur l'entrée Reset. Comme les signaux sont par défaut à l'état 0, il n'y a rien à câbler pour avoir un compteur par 16.



Le logidule est piloté par un microprocesseur dont on utilise 4 sorties et 7 entrées. Une programmation astucieuse mesure le temps où l'horloge est à "1" crée des retards pour rendre visible la séquence des opérations internes: décodage de l'état - activation du reset - remise à zéro - désactivation du reset.

### 2.12 Actions sur les entrées statiques

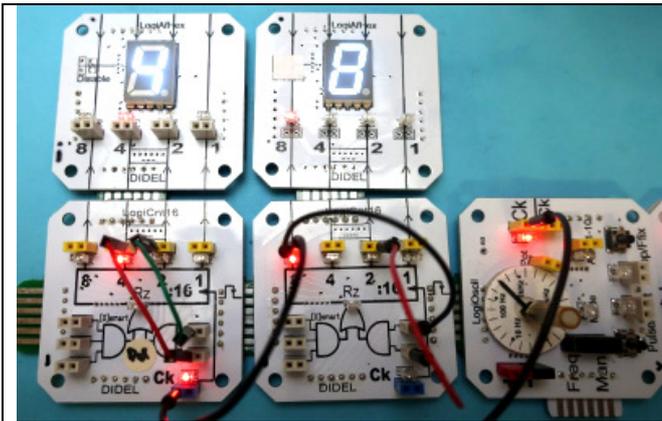
Les entrées Set et Clr sont prioritaires sur l'entrée Ck et facilitent le câblage de compteurs décimaux et autres. Pour un compteur par 10, on génère un Clr sur toutes les bascules lorsque Q2 et Q8 (état 0xA) sont actifs. Une porte ET est utilisée. Quelles sont alors les conditions de bon fonctionnement ?



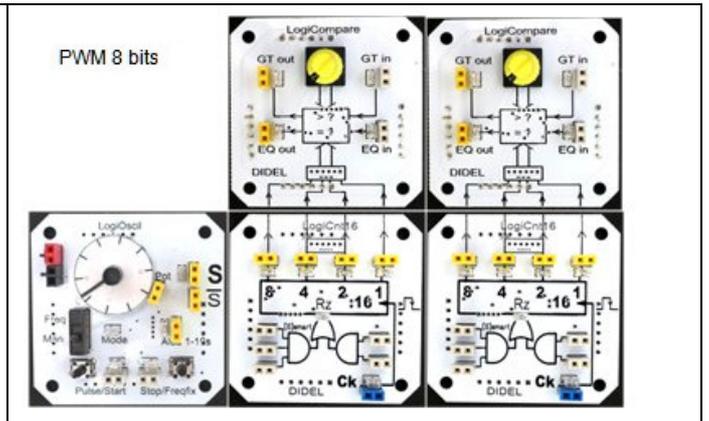
Les compteur par 10 est évident. Attention avec le compteur par 6, la bascule Q8 ne change pas, et c'est le sortie Q4 qui doit être utilisée pour commander l'étage suivant.

### 2.13 PWM

Générer des impulsions de longueur variable est utilisé pour commander la vitesse d'un moteur continu ou varier l'intensité lumineuse d'une LED. Utiliser la sortie d'un comparateur est un des schémas possible.



Compteur par 60



PWM 8 bits

### 2.14 Séquenceurs

Presque toutes les applications ont besoin d'un séquenceur. Un signal, un poussoir demande une action qui compte et envoie des impulsions de commande. Si c'est complexe, on décompose en plusieurs séquenceurs simples et l'analyse du diagramme des temps conduit à l'utilisation de quelques bascules et compteurs.

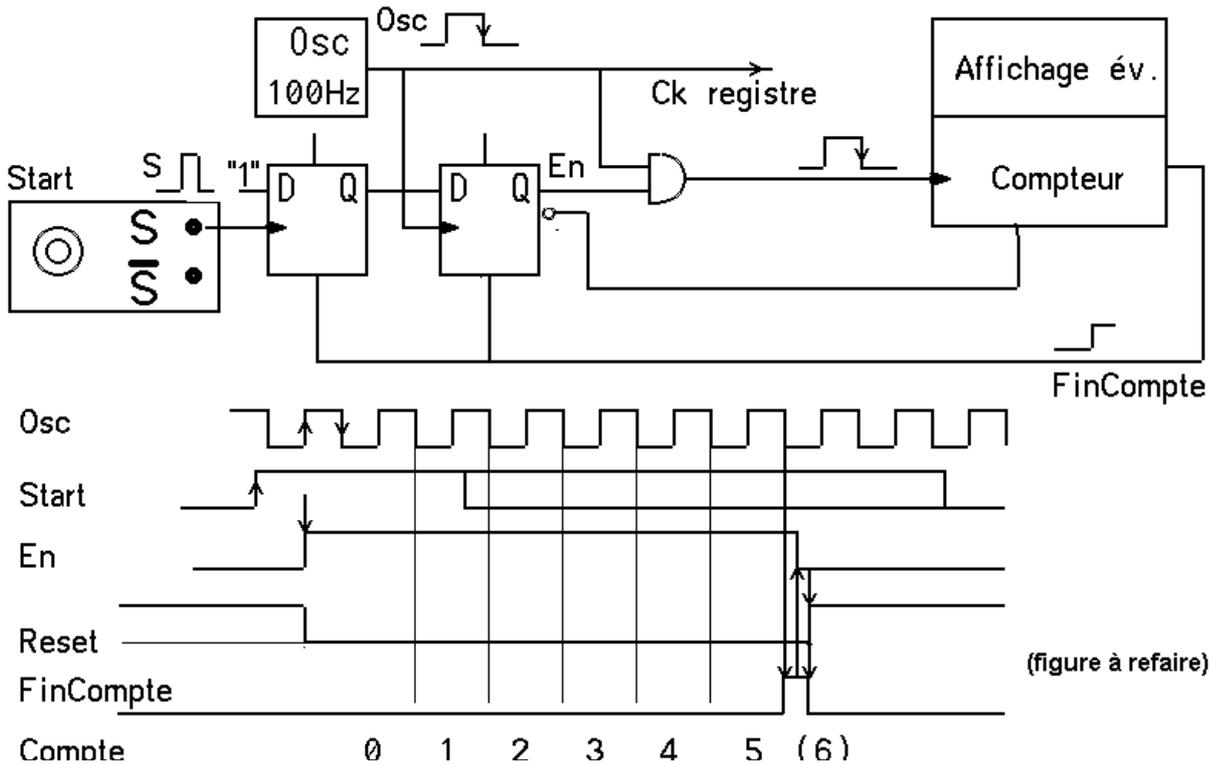
Un oscillateur est toujours utilisé pour rythmer les opérations. On a des instants d'horloge et l'évolution du système (passage d'un état au suivant) suit un front descendant de l'horloge.

Tout signal extérieur doit être synchronisé sur cette horloge si on veut garantir la fiabilité.

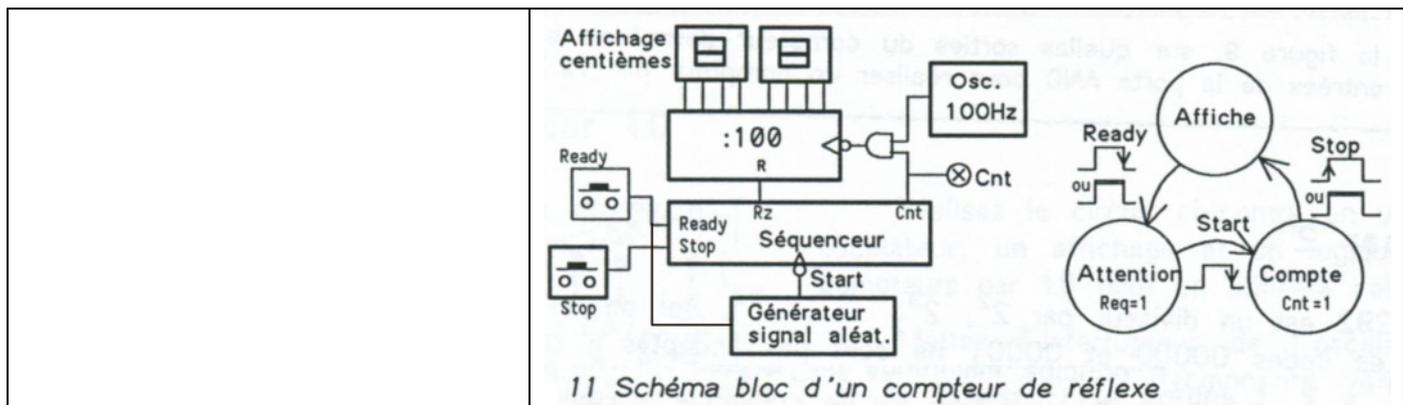
Une action fréquente est d'envoyer 8 ou 16 impulsions pour décaler un registre.

L'oscillateur génère des impulsions à une fréquence qui peut être élevée. L'utilisateur pèse sur un bouton pendant une durée quelconque, l'action ne peut pas être statique (elle se répéterait tant que l'on presse); la transition, en général quand on relâche la pression, doit être active.

La bascule Q1 autorise l'envoi des impulsions au compteur. L'état final est décodé et mets à zéro la bascule Q1. Le diagramme des temps doit envisager toutes les situations, comme ce serait le cas avec une description formelle.



## 2.15 Application: un mesureur de réflexes



Pour cette manip, l'oscillateur logidule a été complété par un générateur aléatoire, ce qui permet d'en parler..

Doc inachevée le 200813