




# ***Circuits séquentiels***

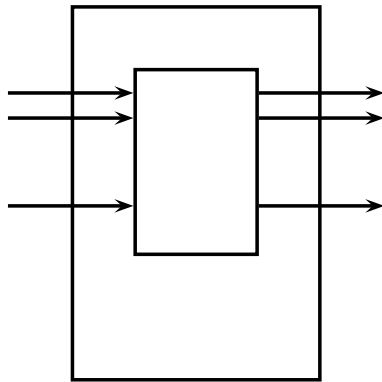
Vincent Risch, septembre 2006, révision mai 2014

I.U.T., Aix-Marseille Université

- 
- Circuits combinatoires → *dispositifs de calcul*
    - △ Algèbres de Boole : rappels
    - △ Décodeurs, multiplexeurs, démultiplexeurs, additionneur...
  - Circuits séquentiels → *dispositifs de mémorisation*
    - △ bascules asynchrones et synchrones
    - △ registres
    - △ séquenceur

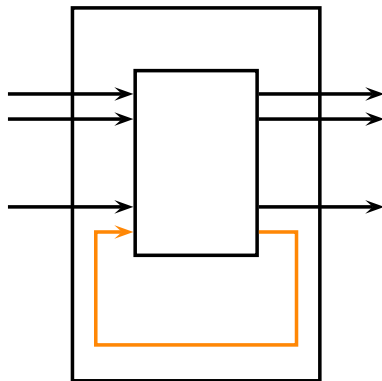
# Circuits séquentiels

Un circuit **séquentiel** est un circuit combinatoire dont les valeurs des variables de sortie dépendent des variables d'entrée et de variables internes.



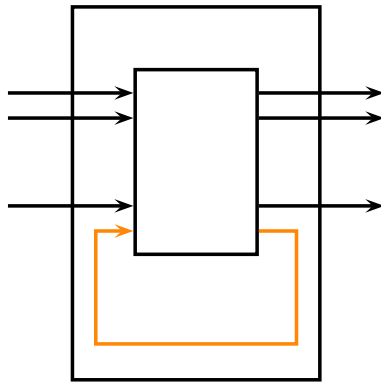
# Circuits séquentiels

Un circuit **séquentiel** est un circuit combinatoire dont les valeurs des variables de sortie dépendent des variables d'entrée et de **variables internes**.



# Circuits séquentiels

Un circuit **séquentiel** est un circuit combinatoire dont les valeurs des variables de sortie dépendent des variables d'entrée et de **variables internes**.



$\Delta t$  : temps de propagation rétroactif

# ***Circuits asynchrones vs. synchrones***



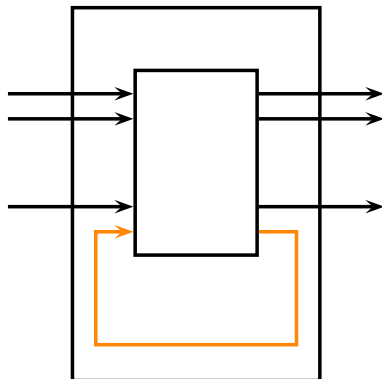
Les circuits séquentiels se divisent en deux catégories :

- asynchrones : les variables du système évoluent librement au cours du temps ;

# *Circuits asynchrones vs. synchrones*

Les circuits séquentiels se divisent en deux catégories :

- asynchrones : les variables du système évoluent librement au cours du temps ;
- synchrones : l'évolution des variables dépend d'une impulsion d'horloge comme un des signaux d'entrée.

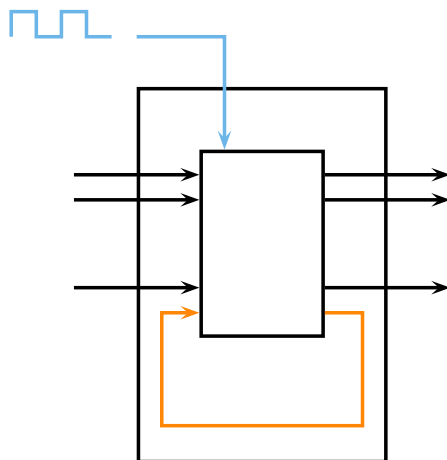


$\Delta t$  : temps de propagation rétroactif

# Circuits asynchrones vs. synchrones

Les circuits séquentiels se divisent en deux catégories :

- asynchrones : les variables du système évoluent librement au cours du temps ;
- synchrones : l'évolution des variables dépend d'une **impulsion d'horloge** comme un des signaux d'entrée.



$\Delta t$  : temps de propagation rétroactif



# *Élément de mémoire*



- L'*élément de mémoire* est défini comme un circuit séquentiel à deux états, 0 et 1, utilisé pour stocker le contenu d'un bit ;

# *Élément de mémoire*



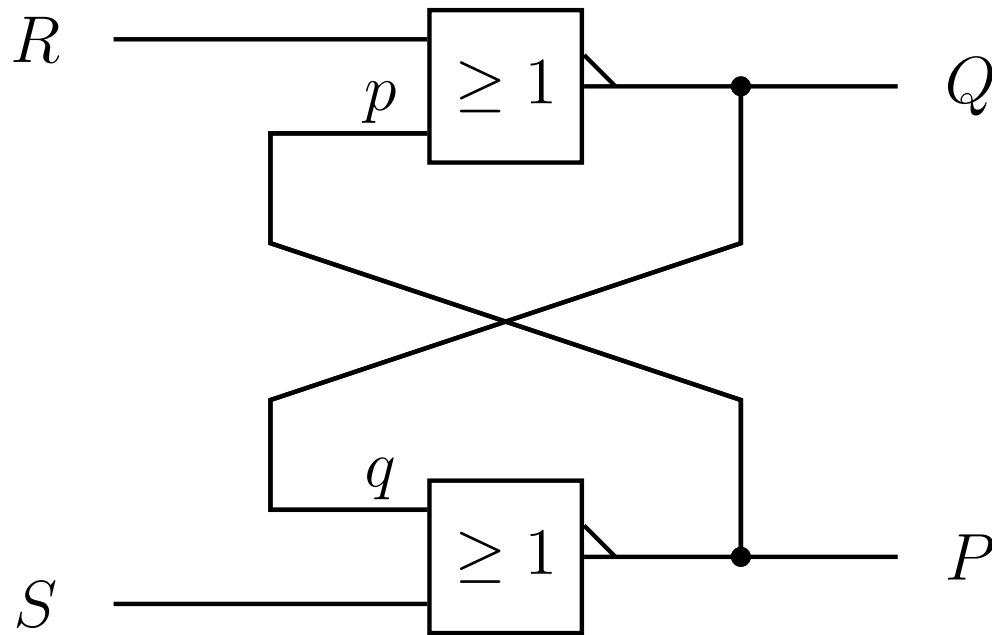
- L'*élément de mémoire* est défini comme un circuit séquentiel à deux états, 0 et 1, utilisé pour stocker le contenu d'un bit ;
- l'état du circuit est modifié par des signaux de commutation aux entrées ;

# *Élément de mémoire*

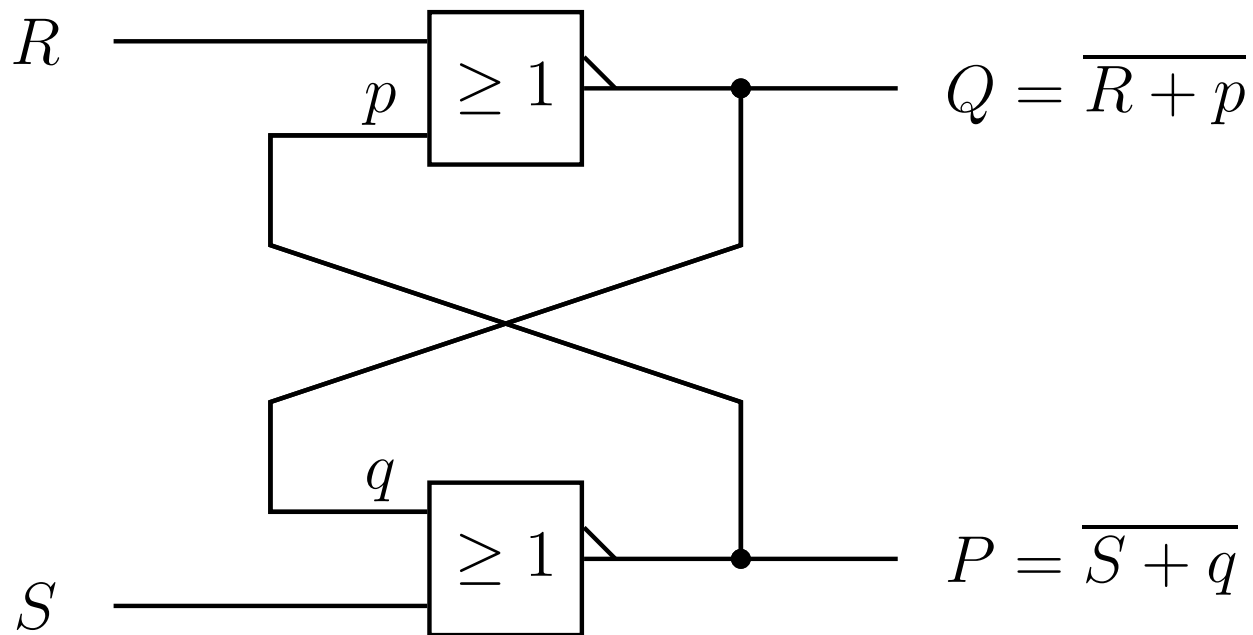


- L'*élément de mémoire* est défini comme un circuit séquentiel à deux états, 0 et 1, utilisé pour stocker le contenu d'un bit ;
- l'état du circuit est modifié par des signaux de commutation aux entrées ;
- il possède deux sorties dont l'une est le complément de l'autre.

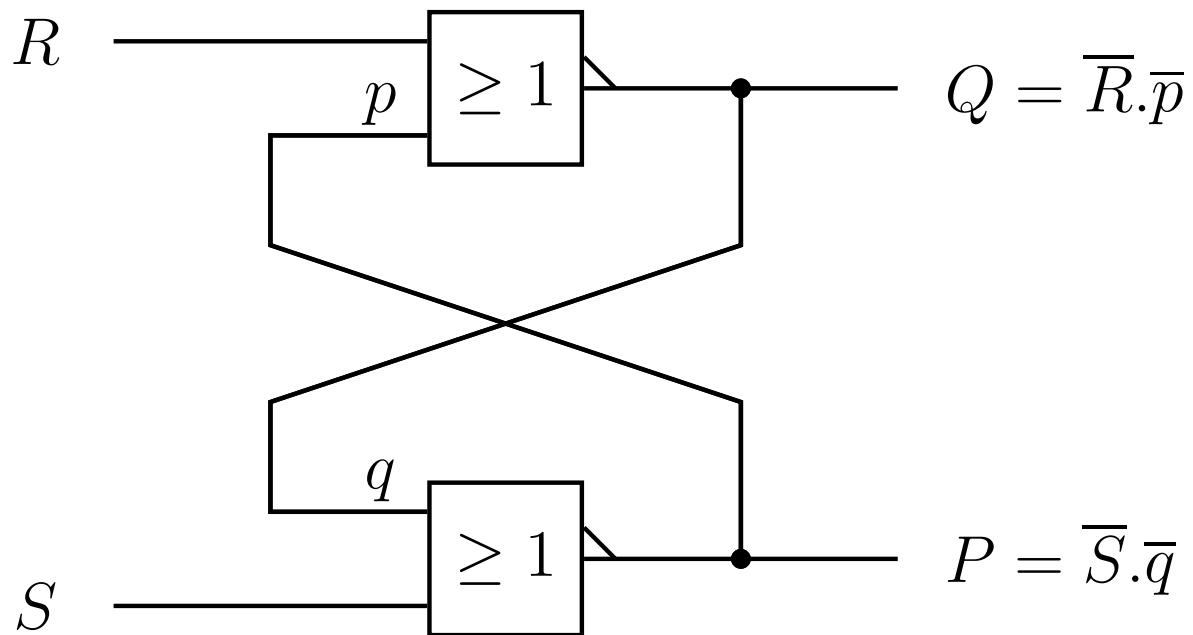
# La bascule RS



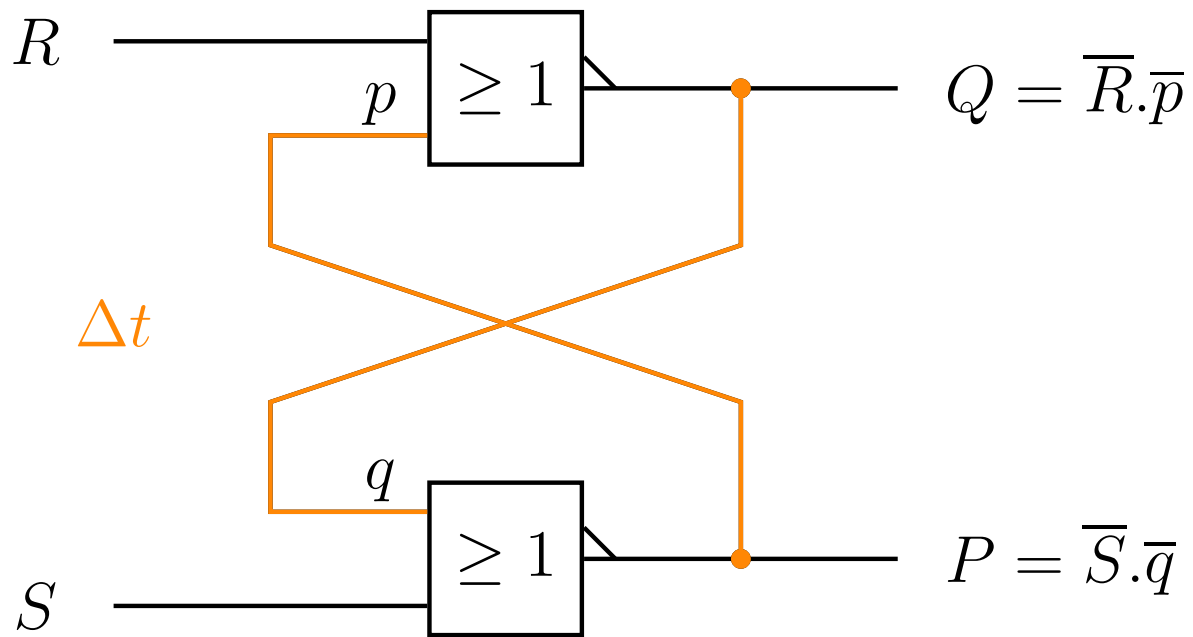
# La bascule RS



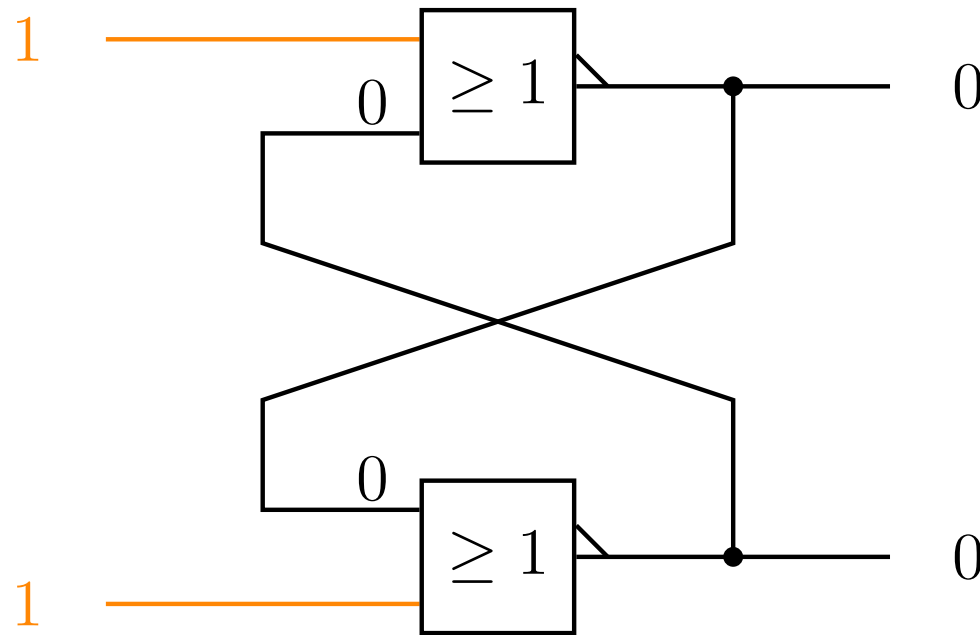
# La bascule RS



# La bascule RS

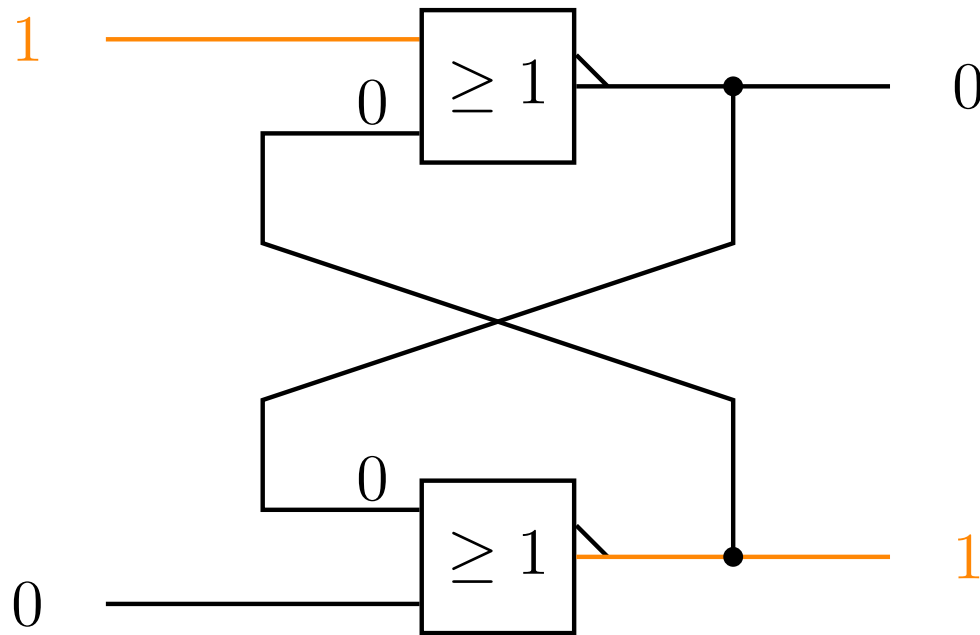


# Exemple 1 : $R=1, S=1 \rightarrow R=1, S=0$

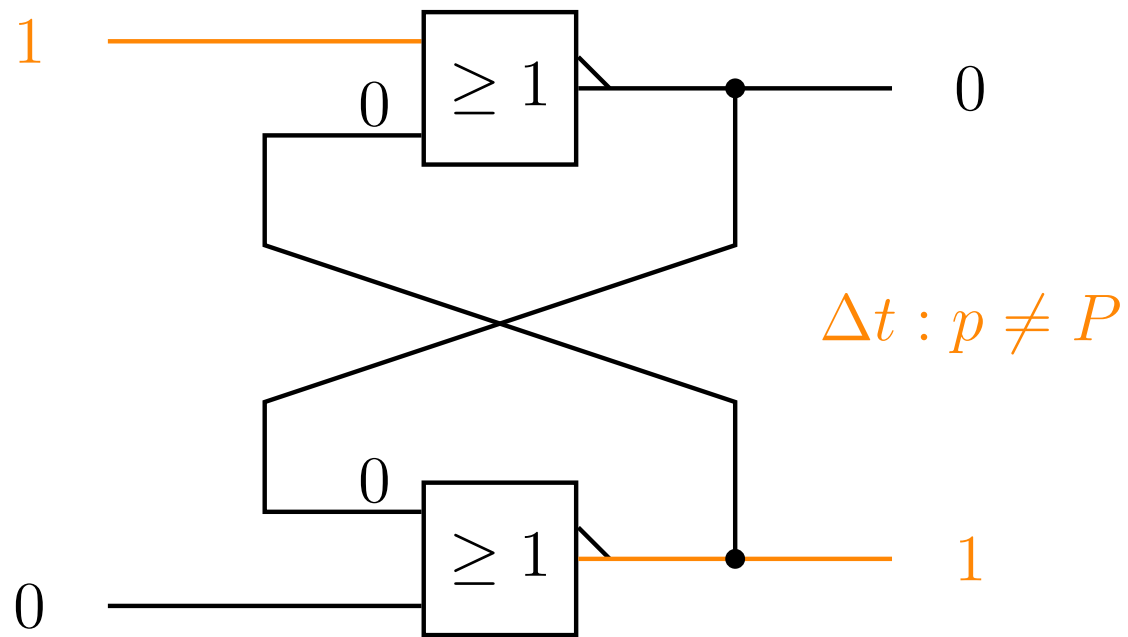




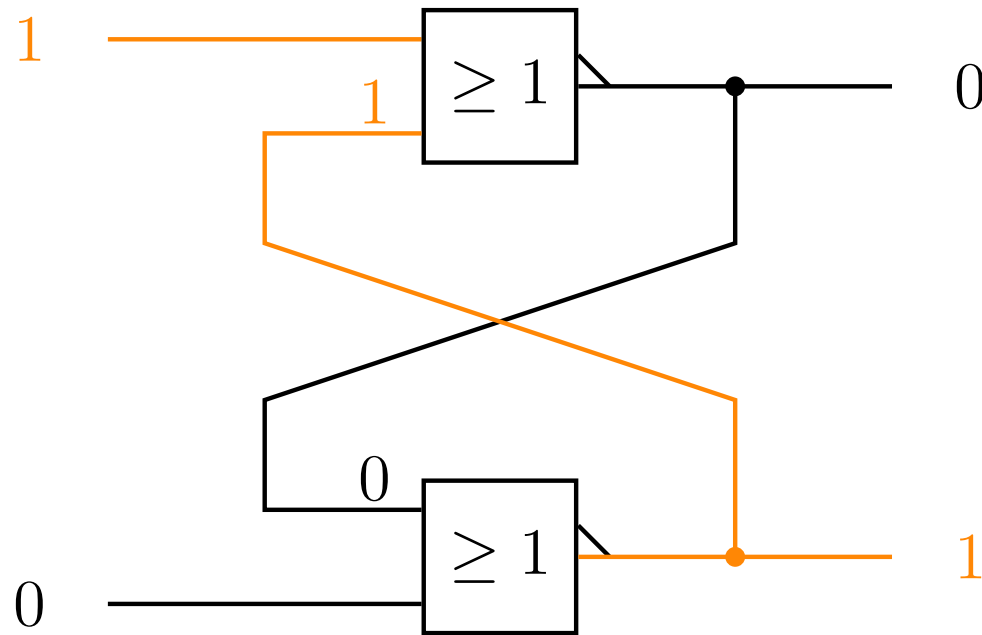
# Exemple 1 : $R=1, S=1 \rightarrow R=1, S=0$



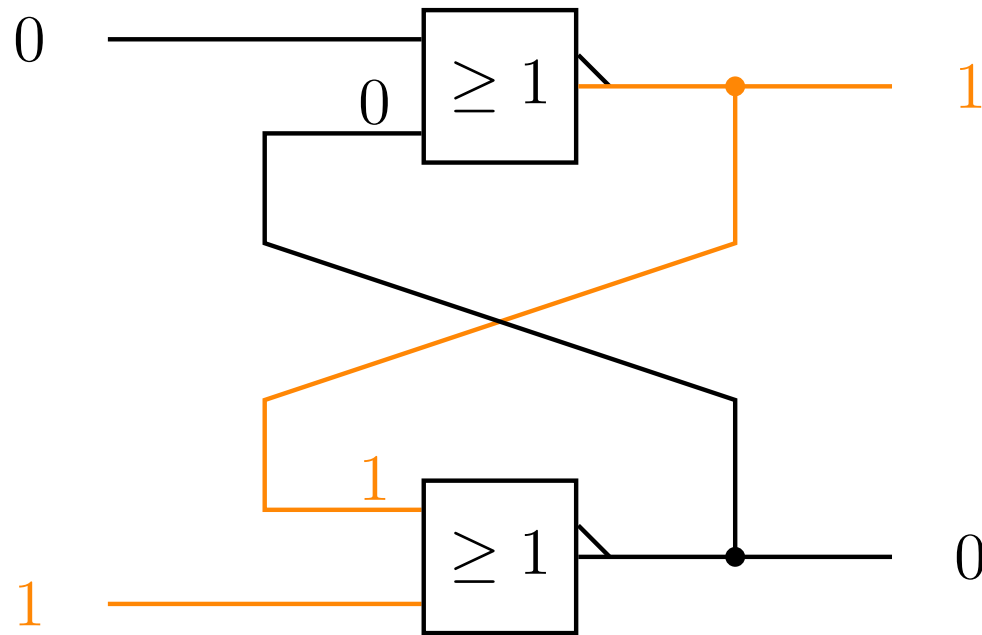
# Exemple 1 : $R=1, S=1 \rightarrow R=1, S=0$



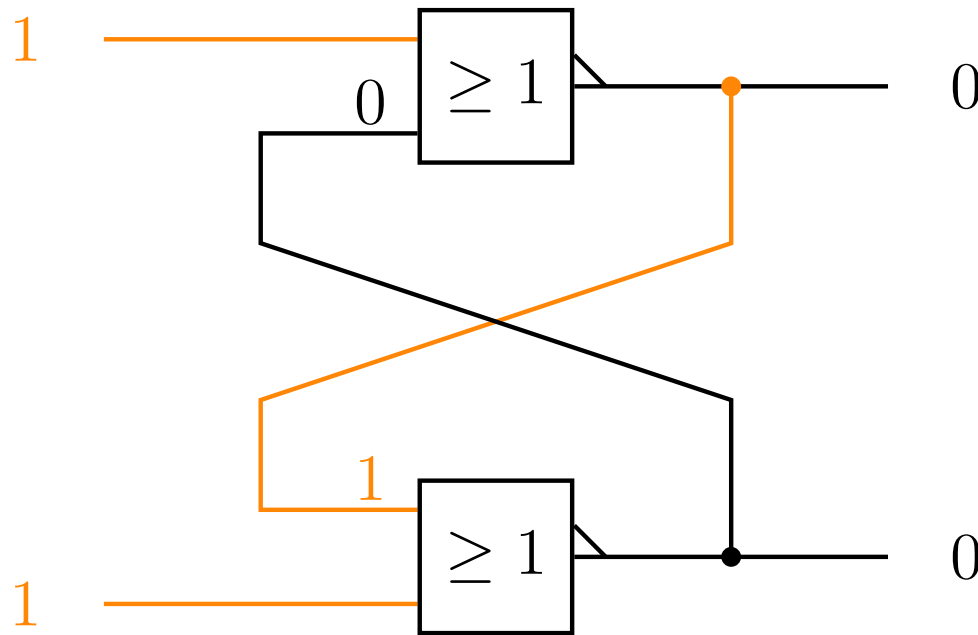
# Exemple 1 : $R=1, S=1 \rightarrow R=1, S=0$



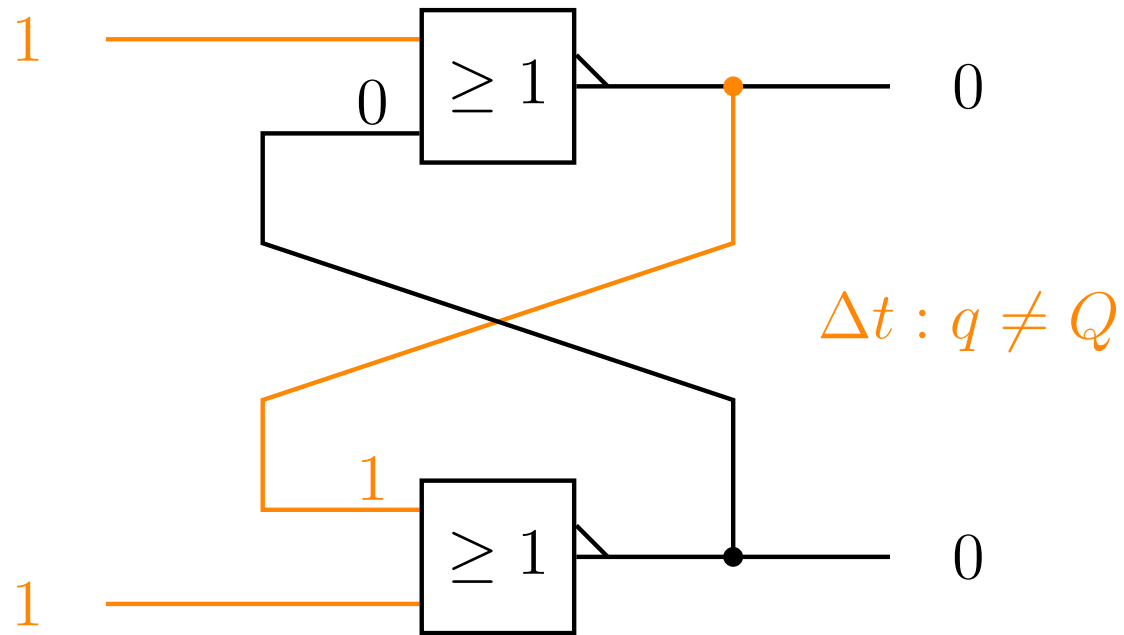
# Exemple 2 : $R=0, S=1 \rightarrow R=1, S=1$



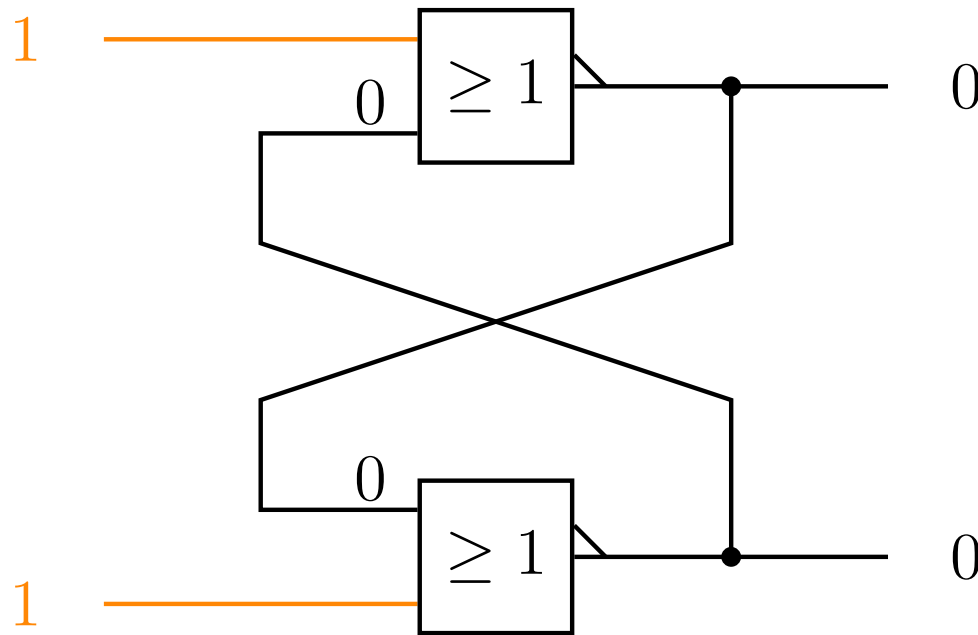
# Exemple 2 : $R=0, S=1 \rightarrow R=1, S=1$



# Exemple 2 : $R=0, S=1 \rightarrow R=1, S=1$



# Exemple 2 : $R=0, S=1 \rightarrow R=1, S=1$



# Remarques

On constate que :

- Avec  $S = 1$  (*Set*), la bascule mémorise la valeur 1.
- Avec  $R = 1$  (*Reset*), la bascule mémorise la valeur 0.
- En raison du temps de propagation rétroactif  $\Delta t$ , la bascule passe par une phase d'instabilité ( $q \neq Q$  ou  $p \neq P$ ) avant de se stabiliser dans un état déterminé.
- La bascule RS ne vérifie pas la dernière contrainte caractéristique d'un élément de mémoire (Exemple 2 : les sorties  $Q$  et  $P$  ne sont pas complémentaires l'une de l'autre).

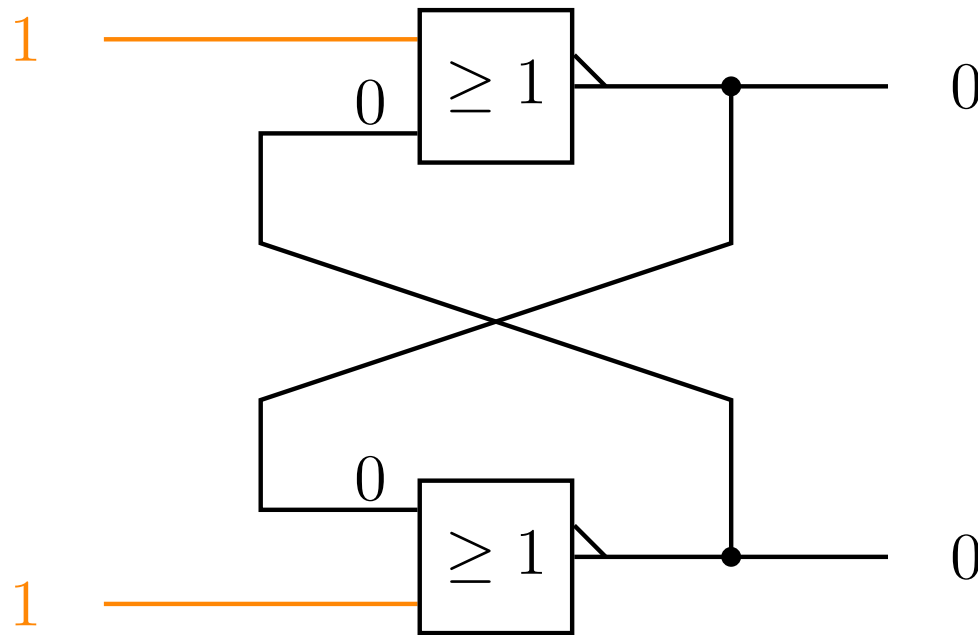


# Remarques

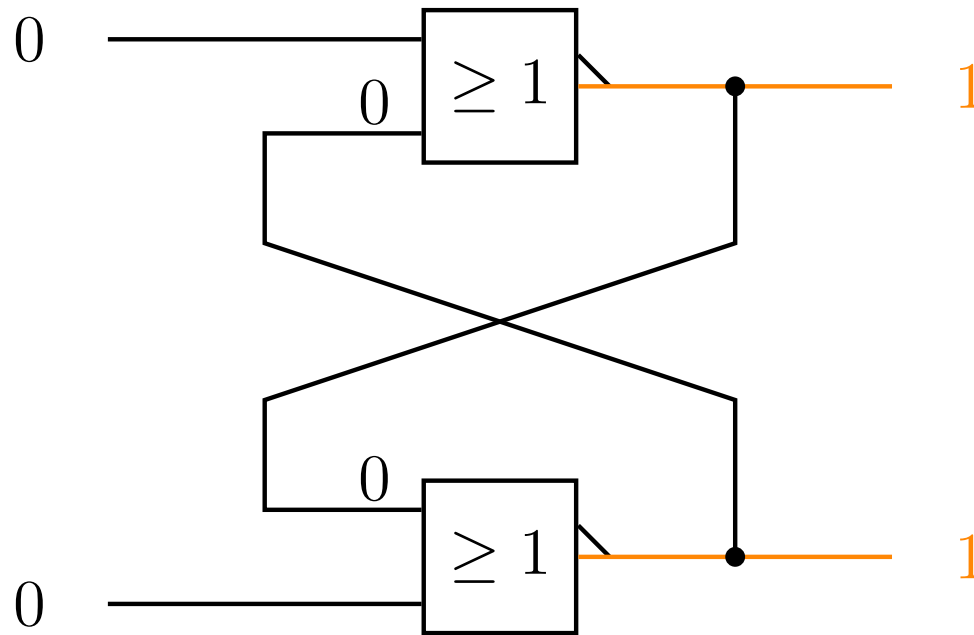
On constate que :

- Avec  $S = 1$  (*Set*), la bascule mémorise la valeur 1.
  - Avec  $R = 1$  (*Reset*), la bascule mémorise la valeur 0.
  - En raison du temps de propagation rétroactif  $\Delta t$ , la bascule passe par une phase d'instabilité ( $q \neq Q$  ou  $p \neq P$ ) avant de se stabiliser dans un état déterminé.
  - La bascule RS ne vérifie pas la dernière contrainte caractéristique d'un élément de mémoire (Exemple 2 : les sorties  $Q$  et  $P$  ne sont pas complémentaires l'une de l'autre).
- **Plus grave** : l'instabilité de la bascule peut devenir chronique...

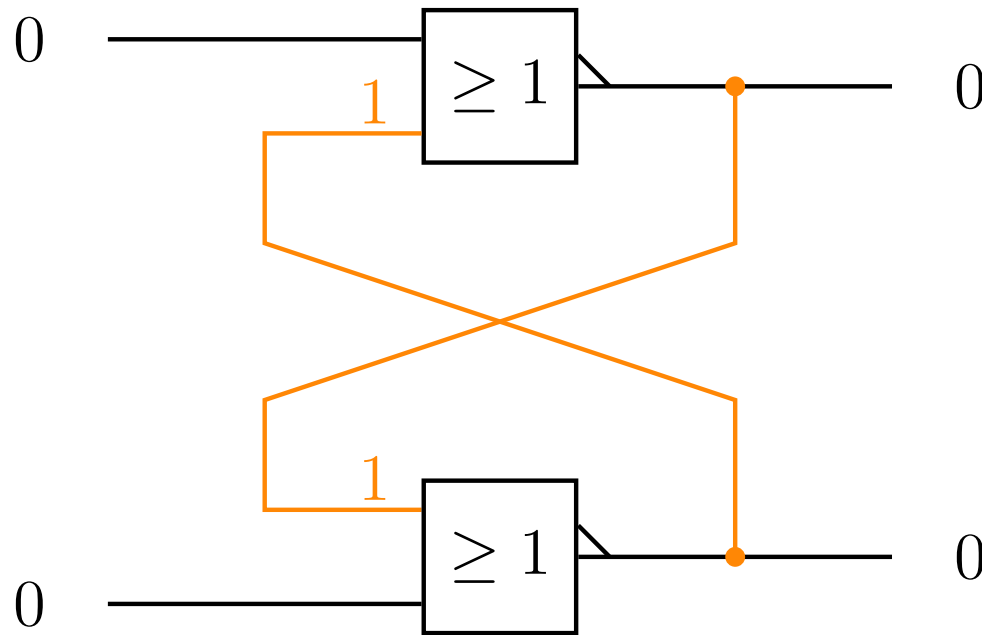
# Exemple 3 : $R=1, S=1 \rightarrow R=0, S=0$



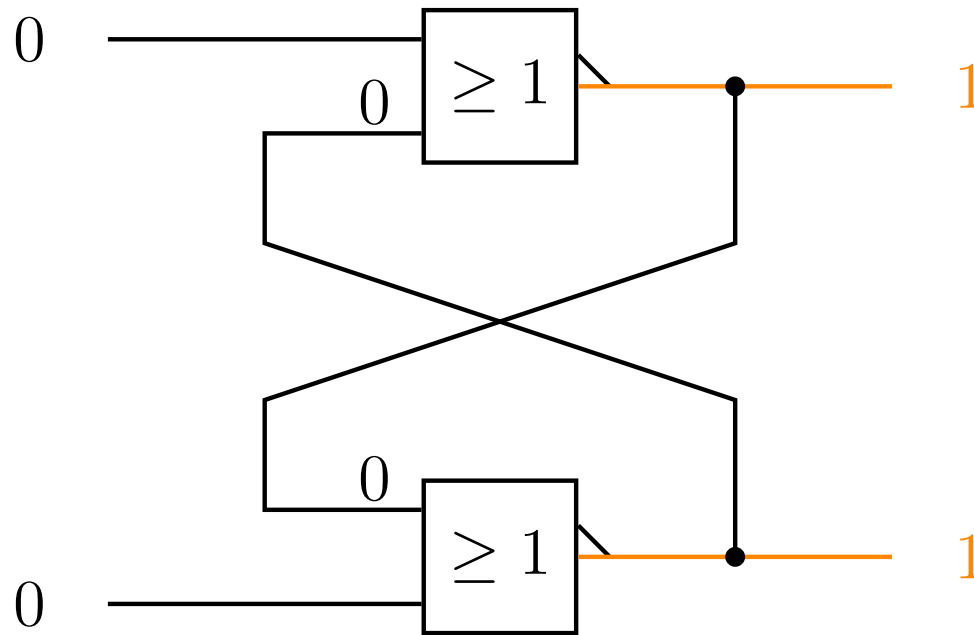
# ***Exemple 3 : $R=1, S=1 \rightarrow R=0, S=0$***



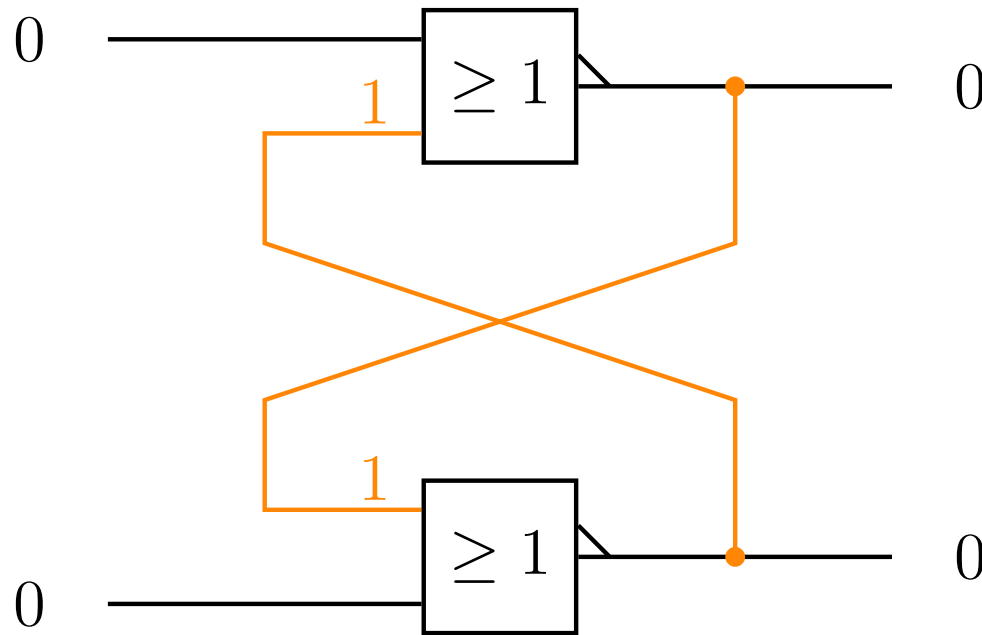
# Exemple 3 : $R=1, S=1 \rightarrow R=0, S=0$



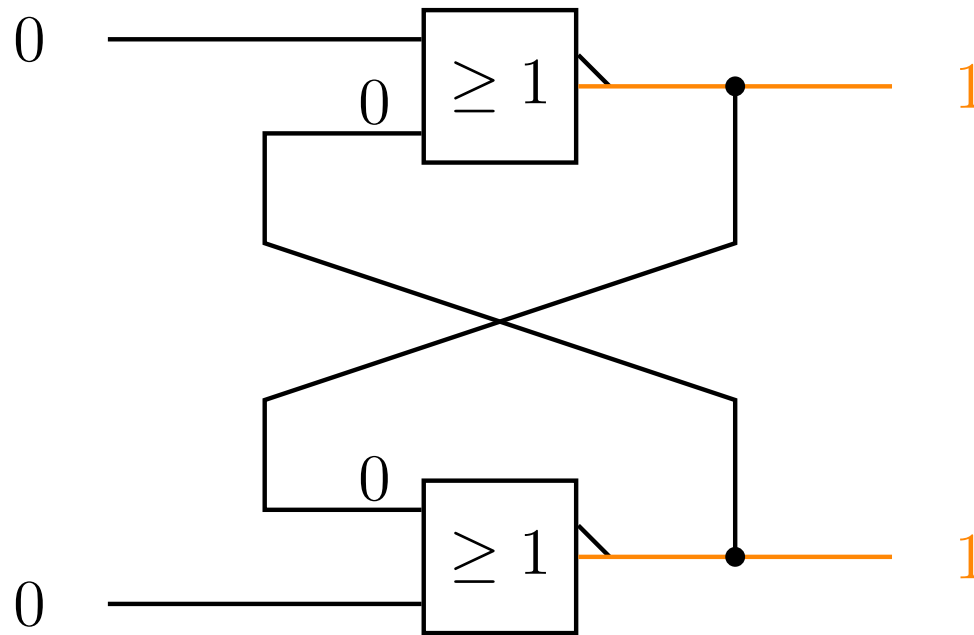
# Exemple 3 : $R=1, S=1 \rightarrow R=0, S=0$



# Exemple 3 : $R=1, S=1 \rightarrow R=0, S=0$



# Exemple 3 : $R=1, S=1 \rightarrow R=0, S=0$



# Conclusion



L'état  $R = S = 1$  est indésirable, car :

- il ne vérifie pas la dernière contrainte d'un élément de mémoire (sorties complémentées) ;
- il déclenche une instabilité chronique de la bascule.



# Conclusion



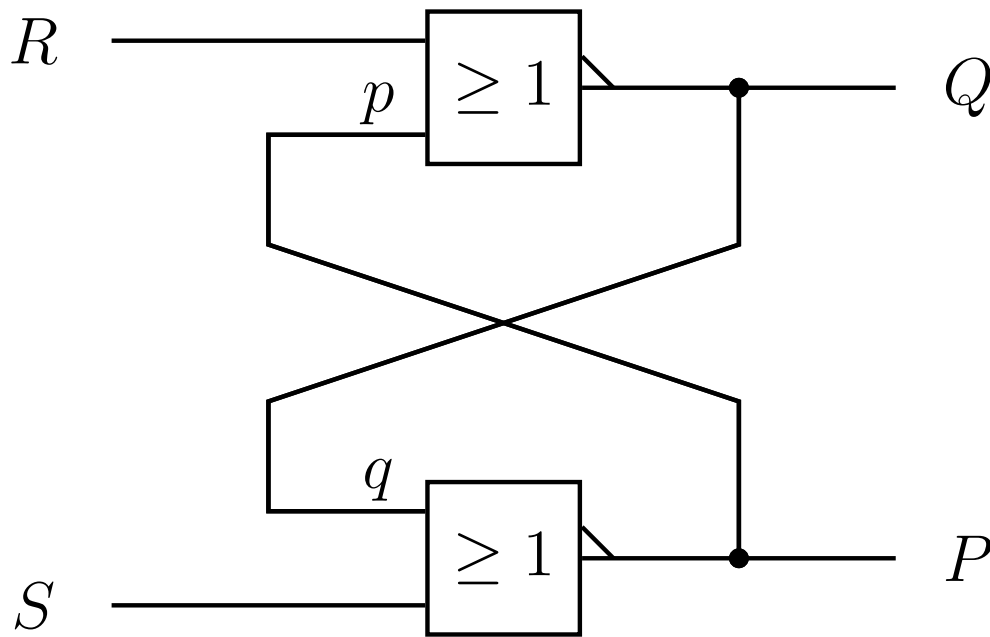
L'état  $R = S = 1$  est indésirable, car :

- il ne vérifie pas la dernière contrainte d'un élément de mémoire (sorties complémentées) ;
- il déclenche une instabilité chronique de la bascule.

→ Une étude systématique passerait par une modélisation en termes d'**automates cellulaires**, ce dont on se passera ici...

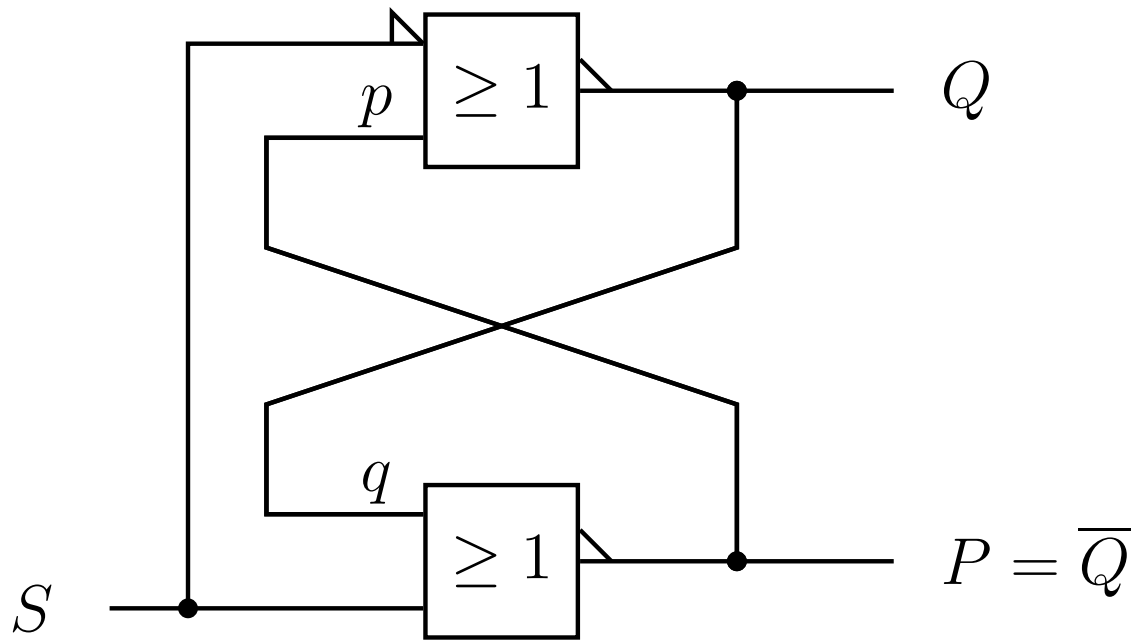
# Une solution...

Coupler  $R$  à  $S$  via une inversion logique



# Une solution...

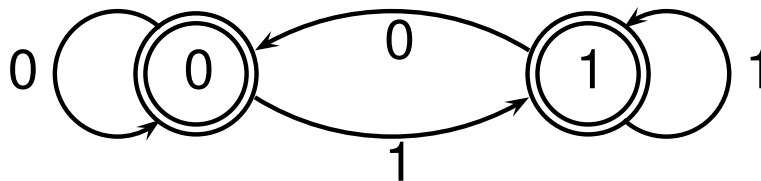
Coupler  $R$  à  $S$  via une inversion logique



# Bascule *D* asynchrone

La bascule obtenue en couplant  $R$  et  $S$  via une inversion logique est appelée *bascule D*. Il s'agit d'une bascule asynchrone qui fonctionne bien comme un élément de mémoire car

- les sorties sont complémentées ;
- la bascule mémorise la dernière valeur de l'entrée.



# Remarques

- La bascule fonctionne de façon *transparente* c'est-à-dire que la sortie  $Q$  recopie l'état de l'entrée ;

## *Remarques*

- La bascule fonctionne de façon *transparente* c'est-à-dire que la sortie  $Q$  recopie l'état de l'entrée ;
- étant asynchrone, elle présente peu d'intérêt car elle n'opère que comme un retardateur de signal.

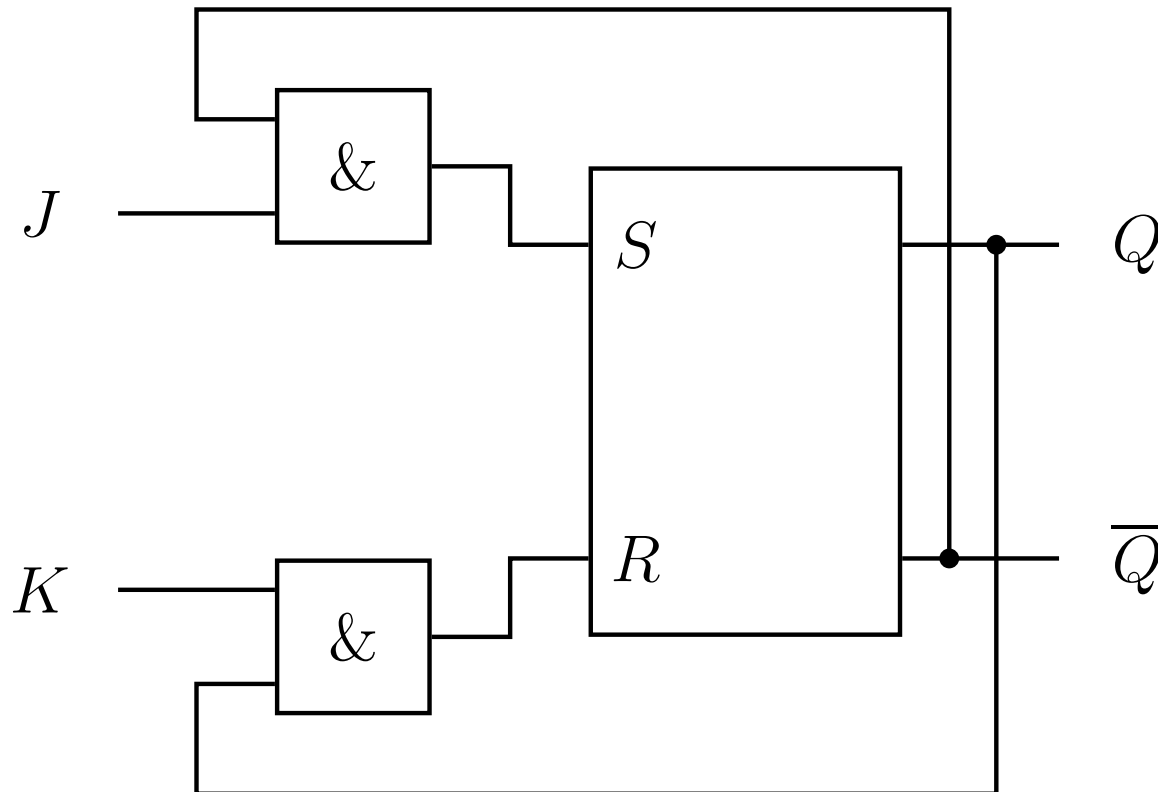
# Bascule JK asynchrone

- si  $J = 0$  et  $K = 0$  : la sortie  $Q$  ne change pas d'état ;
- si  $J = 0$  et  $K = 1$  : la sortie  $Q$  passe à 0 ;
- si  $J = 1$  et  $K = 0$  : la sortie  $Q$  passe à 1 ;
- si  $J = 1$  et  $K = 1$  : la sortie  $Q$  passe à  $\overline{Q}$ .

Table de fonctionnement :

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

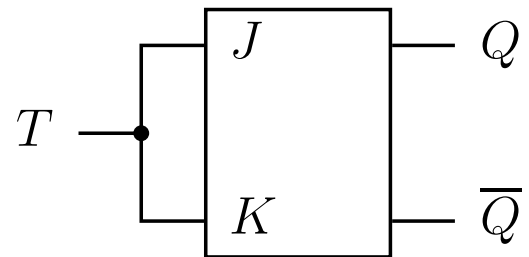
# *A partir d'une bascule RS*



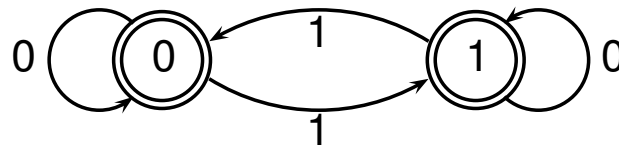


# Bascule *T* asynchrone

Bascule *trigger* obtenue en reliant *J* et *K*



$T$	$Q_{t+1}$	$Q_t$
0	$Q_t$	*
1	$\overline{Q_t}$	*



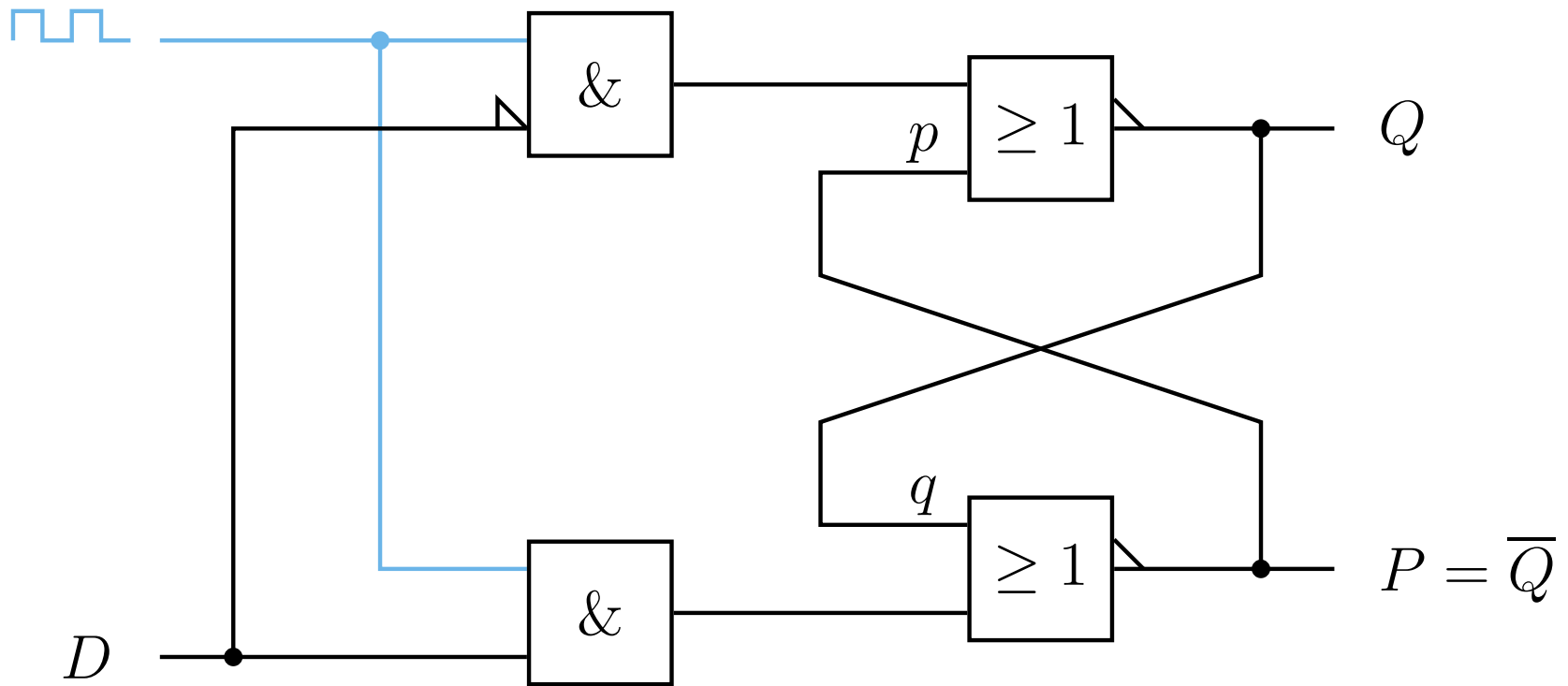
# ***Bascules synchrones***



→ Dépendent d'un signal d'horloge

- Les bascules *verrou* : actives pendant toute la durée du signal haut.
- Les bascules à *commutation sur front* : actives au seul moment de la transition sur front (antérieur ou postérieur) du signal d'horloge.

# Bascule D verrou



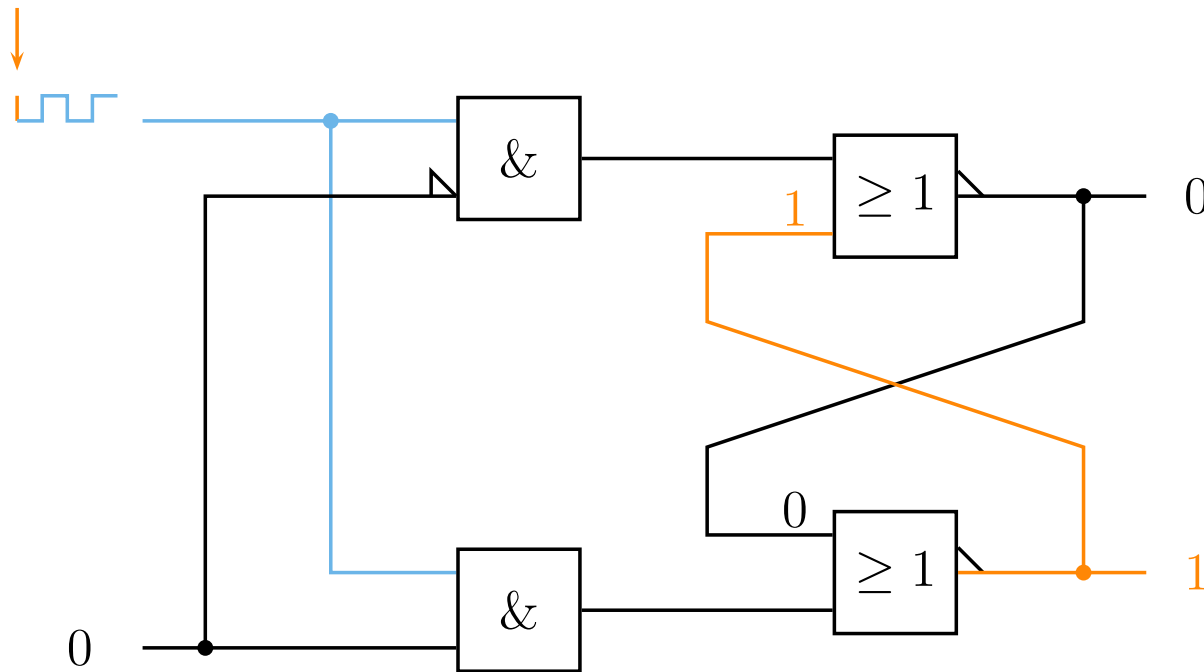
# ***Fonctionnement***



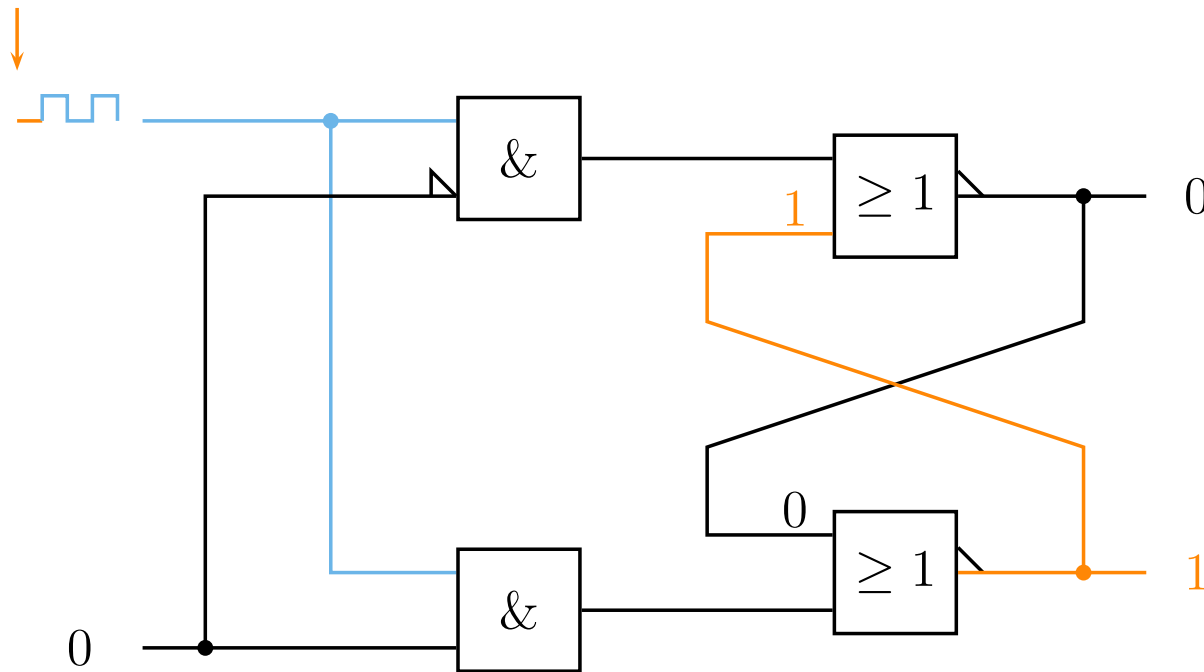
- La propagation du signal d'entrée se fait sur la valeur haute du signal d'horloge qui fait alors office d'autorisation en écriture ;
- la durée du cycle doit être supérieure ou égale à la durée de propagation dans la bascule :

$$\text{Cycle d'horloge} \geq \Delta t.$$

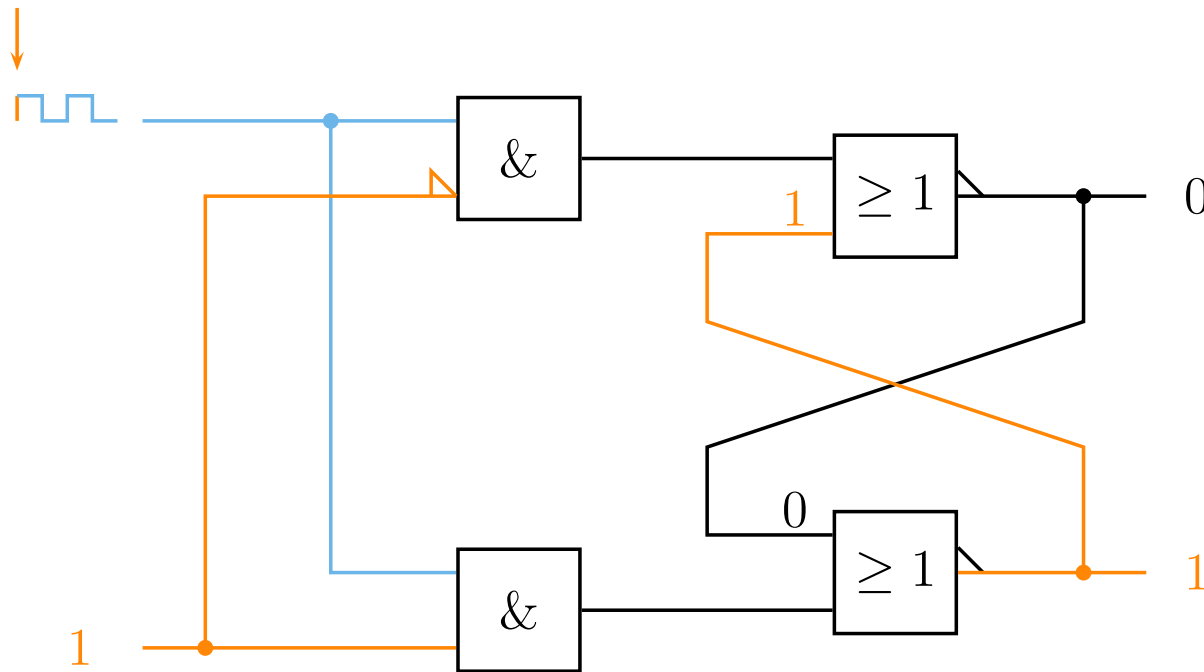
# Exemple 4 : $D = 0 \rightarrow D = 1$



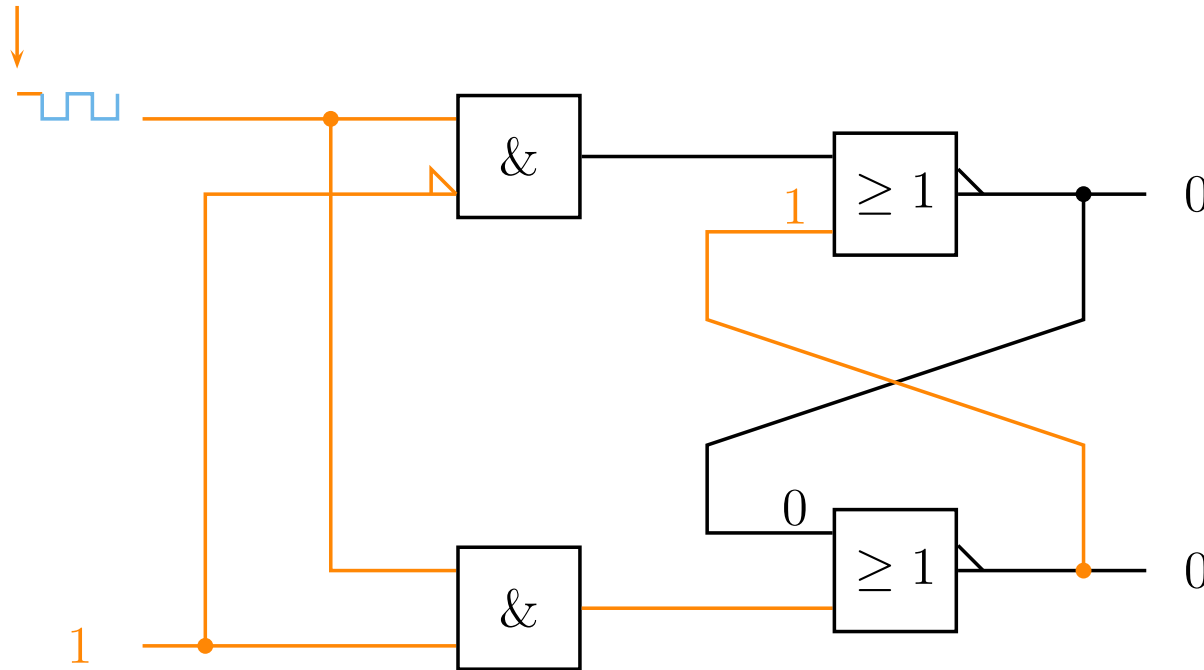
# Exemple 4 : $D = 0 \rightarrow D = 1$



# Exemple 4 : $D = 0 \rightarrow D = 1$

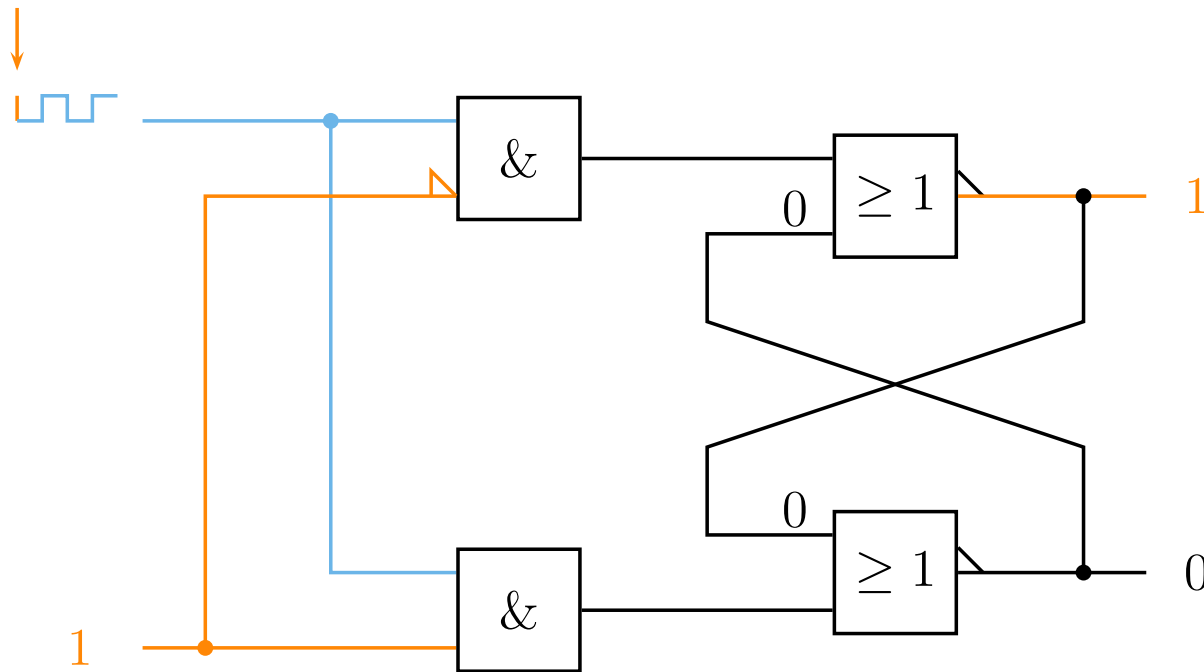


# Exemple 4 : $D = 0 \rightarrow D = 1$

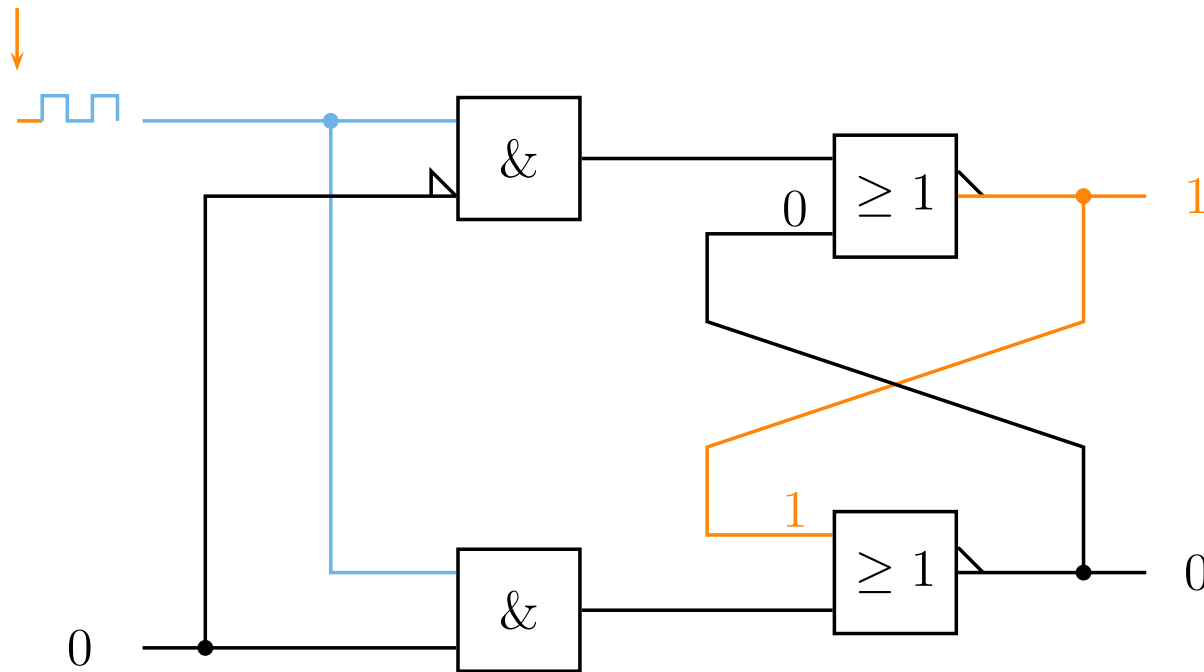




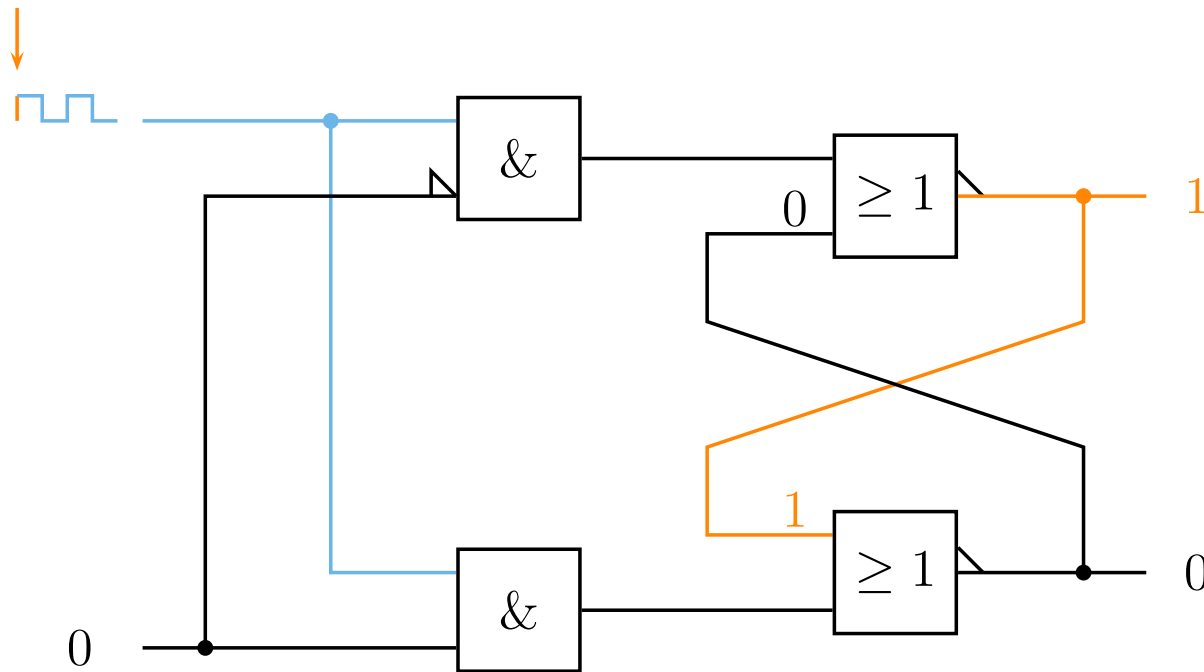
# Exemple 4 : $D = 0 \rightarrow D = 1$



# Exemple 4 : $D = 0 \rightarrow D = 1$



# Exemple 4 : $D = 0 \rightarrow D = 1$



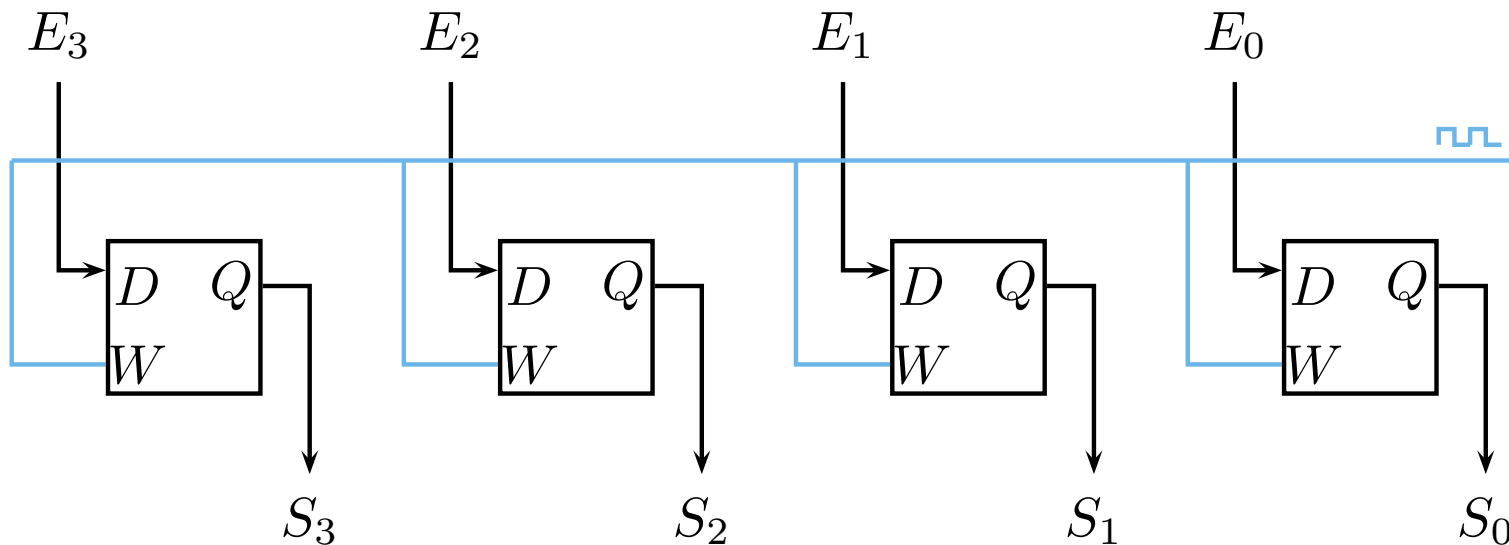
# *Quelques circuits séquentiels*



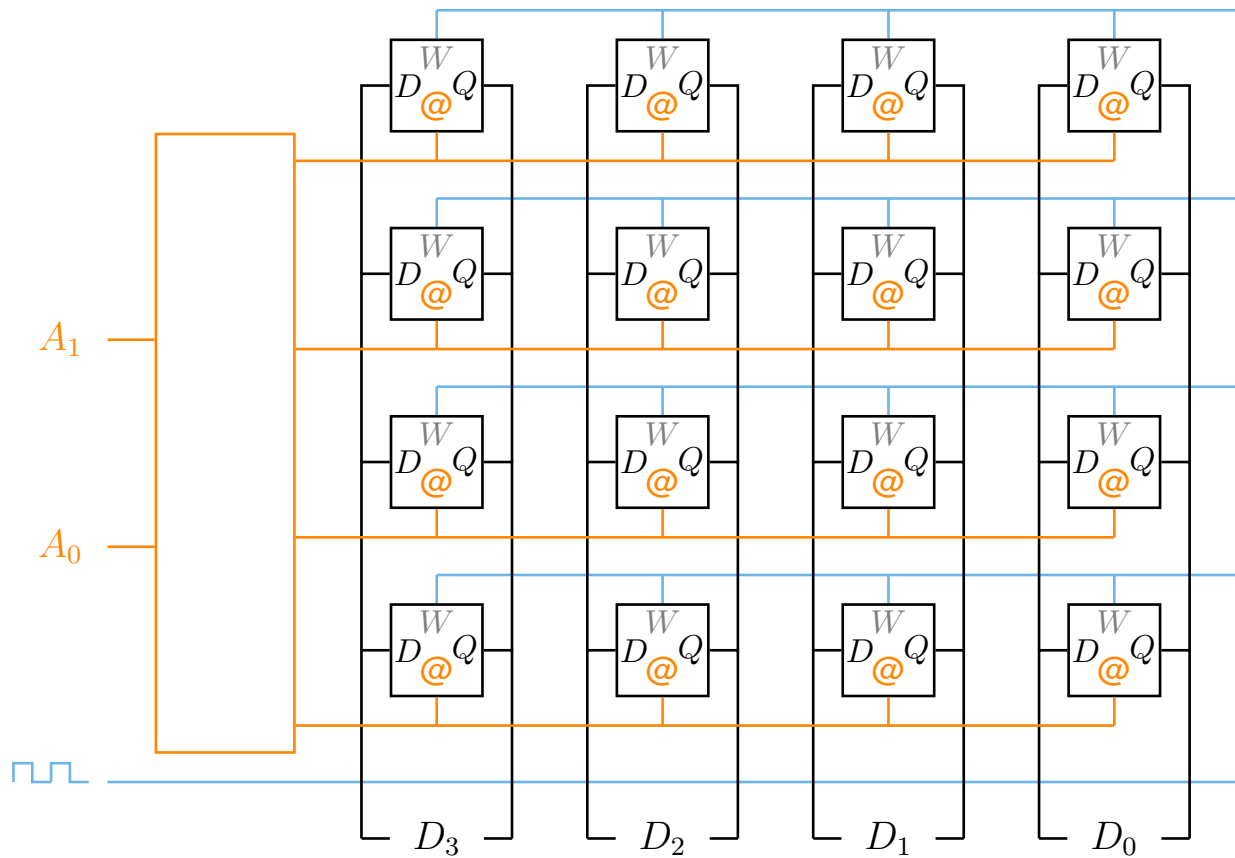
1. Registres
2. Bancs de registres
3. Compteurs
4. Séquenceurs

# Registre

La juxtaposition de  $n$  bascules  $D$  permet la constitution d'un registre de  $n$  bits.



# Banc de registres 4 bits

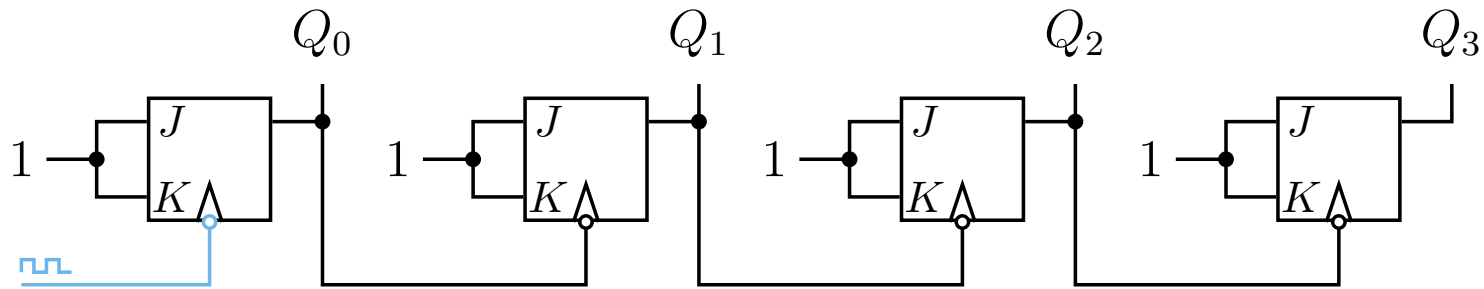


# Compteurs

- Un *compteur* est un registre dont le contenu est incrémenté à chaque impulsion d'une horloge ;
- il est construit à partir de  $n$  bascules interconnectées avec un circuit combinatoire affecté au mode de changement d'état ;
- il passe par  $2^n$  états distincts.

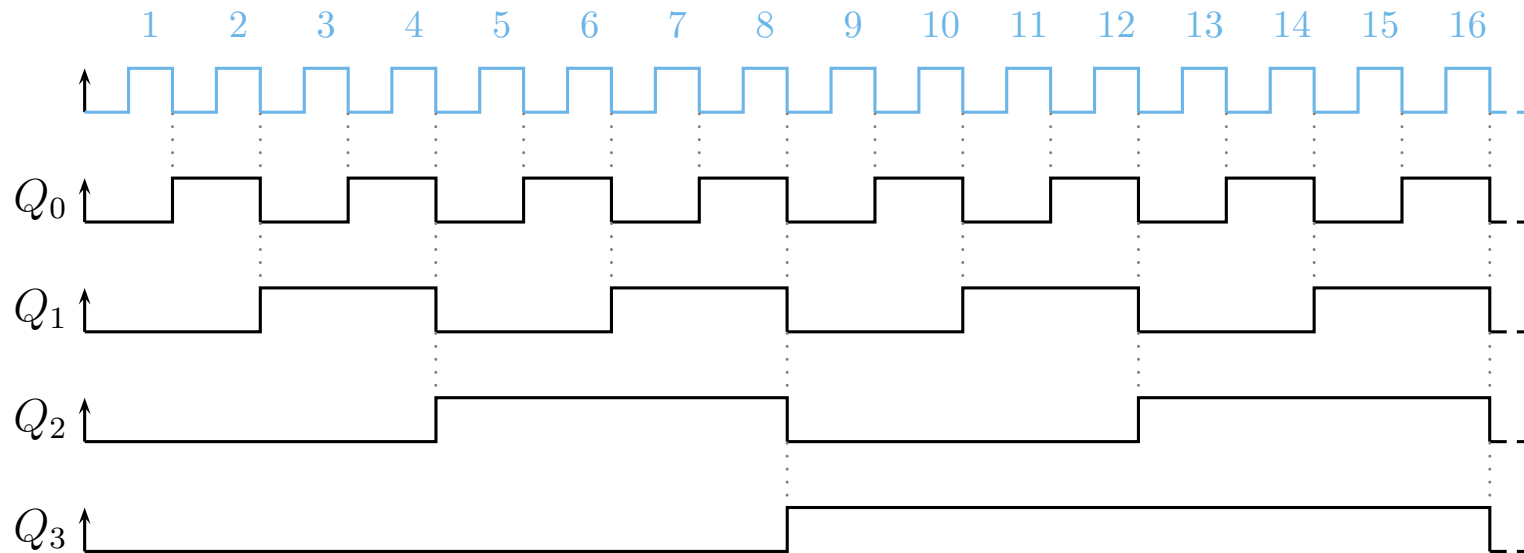
→ **Compteur binaire modulo 16 asynchrone** : 4 bascules  $JK$  à commutation sur front postérieur, utilisées en mode  $T$ .

# Compteur modulo 16 asynchrone

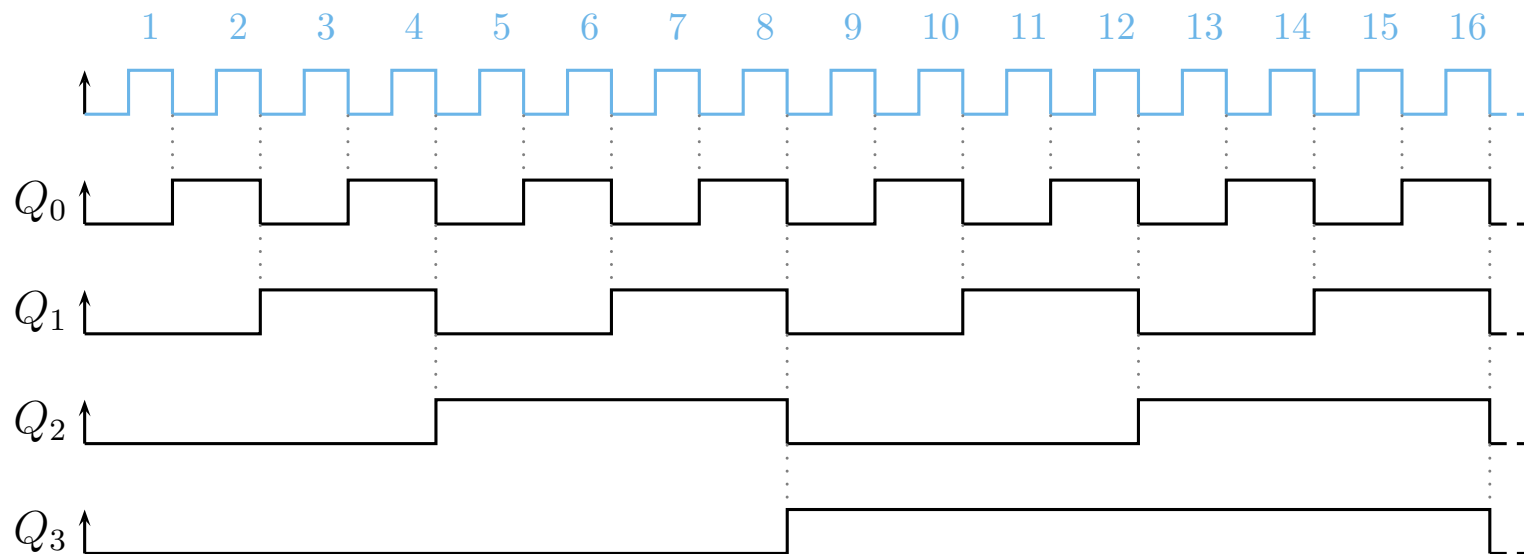




# Chronogramme

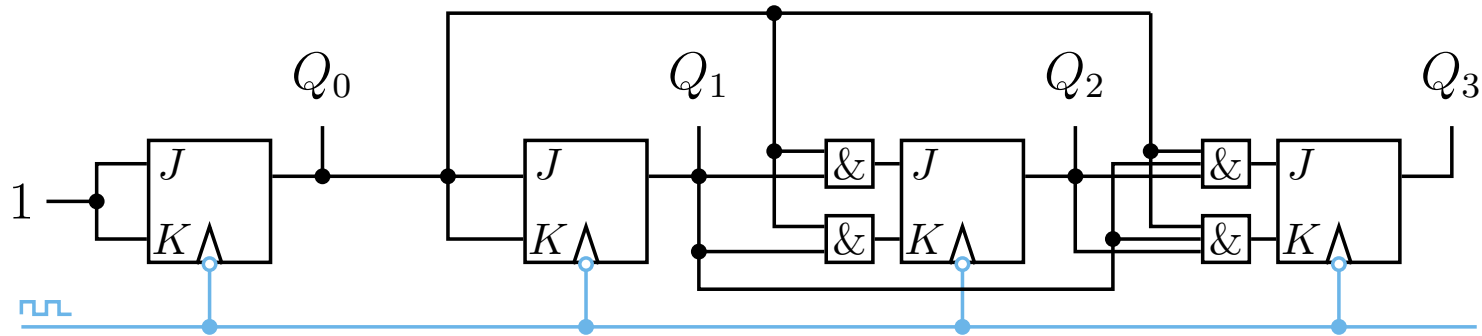


# Chronogramme



→ Chaque sortie divise la fréquence du signal d'entrée correspondant par deux

# Compteur modulo 16 synchrone

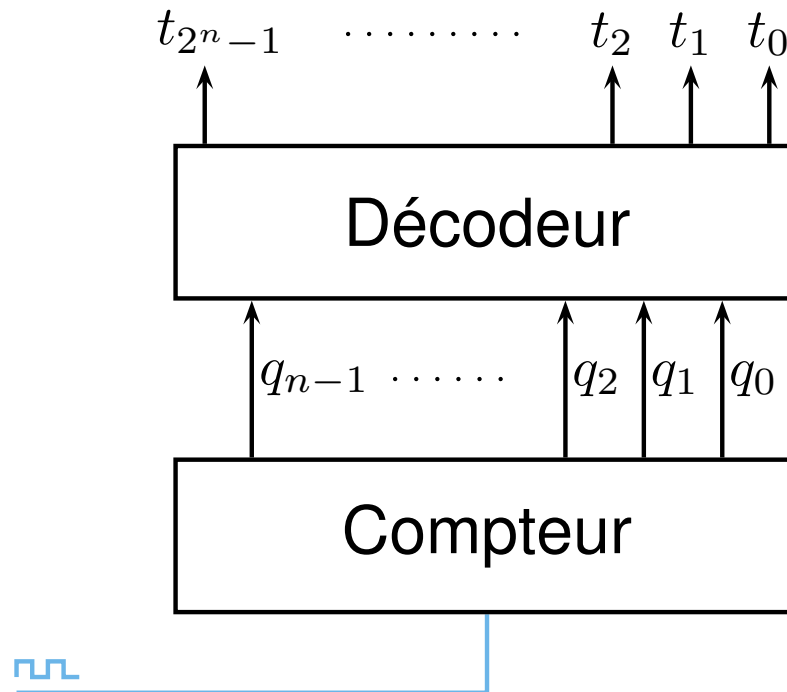


# Séquenceur



- ❑ Association d'un circuit combinatoire et d'un circuit séquentiel permettant de créer des signaux de commande décalés dans le temps ;
  - ❑ constitue le cœur de l'unité de contrôle d'un ordinateur.
- Le séquenceur ayant la structure la plus simple résulte de l'association d'un compteur et d'un décodeur

# Séquenceur



→ distribue successivement  $2^n$  signaux de commandes  $t_0, t_1, \dots, t_{2^n-1}$  correspondant à autant de cycles de l'architecture

# *Epilogue*



- On dispose des éléments permettant de construire une machine qui *calcule* et *mémoire* ;
- étape suivante : construire un calculateur commandé par un *programme*...