

Sciences du numérique 1

Logique séquentielle

Pierre Héroux

Pierre.Heroux@univ-rouen.fr
Université de Rouen

L1 Informatique – EEEA

Plan

- 1 Définitions
- 2 Les bascules
 - Bascule RS
 - Bascule RST – RSClock
 - Bascule JK
 - Bascule D
 - Bascule T
- 3 Les registres
 - Registre de mémorisation
 - Registre à décalage
- 4 Les séquenceurs
 - Séquenceurs asynchrones
 - Séquenceurs synchrones

Plan

1 Définitions

2 Les bascules

- Bascule RS
- Bascule RST – RSClock
- Bascule JK
- Bascule D
- Bascule T

3 Les registres

- Registre de mémorisation
- Registre à décalage

4 Les séquenceurs

- Séquenceurs asynchrones
- Séquenceurs synchrones

Logique séquentielle

- En logique combinatoire, les états de sortie ne dépendent que des états d'entrée.
- En logique séquentielle, les sorties dépendent des entrées mais également de l'état du système.
- Soit un système séquentiel d'entrée X , de sortie Y et d'état Q

$$\begin{cases} Q = f(X, Q) \\ Y = g(X, Q) \end{cases}$$

- La logique séquentielle permet de réaliser des circuits dont le comportement est variable avec le temps.
- L'état du système constitue une mémoire du passé.

Synchrone vs. asynchrone

- Dans un système *asynchrone*, les changements d'état des divers composants dépendent
 - des instants où se produisent les changements d'état des entrées
 - des temps de réponse des composants
 - des temps de propagation des signauxIls peuvent subir quelques aléas
- Pour éviter cela, on peut décider de synchroniser les changements d'états des divers composants sur un signal périodique provenant d'une horloge
Le système est alors dit *synchrone*

Définitions	Bascule RS
Les bascules	Bascule RST – RS _{Clock}
Les registres	Bascule JK
Les séquenceurs	Bascule D
	Bascule T

Plan

1 Définitions

2 Les bascules

- Bascule RS
- Bascule RST – RS_{Clock}
- Bascule JK
- Bascule D
- Bascule T

3 Les registres

- Registre de mémorisation
- Registre à décalage

4 Les séquenceurs

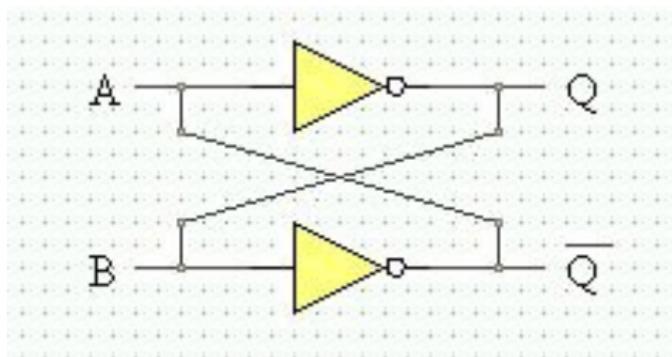
- Séquenceurs asynchrones
- Séquenceurs synchrones

Les bascules

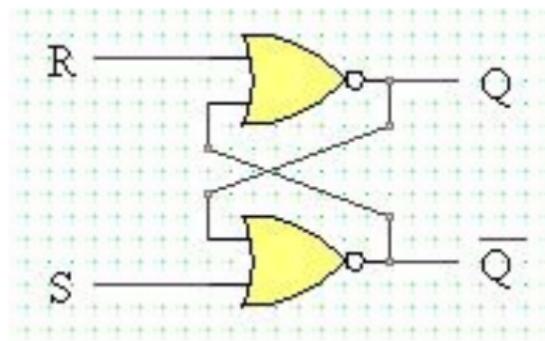
Une bascule :

- mémorise une information élémentaire
- possède deux sorties complémentaires Q et \bar{Q}

Le fonctionnement d'une bascule est basé sur le principe du verrou (latch).

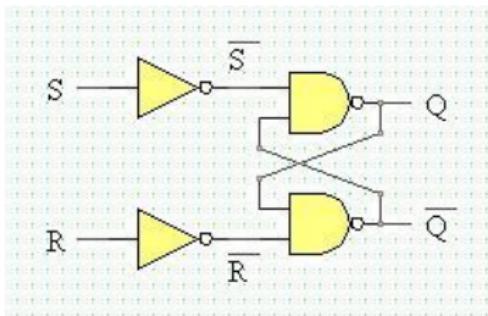


Bascule RS



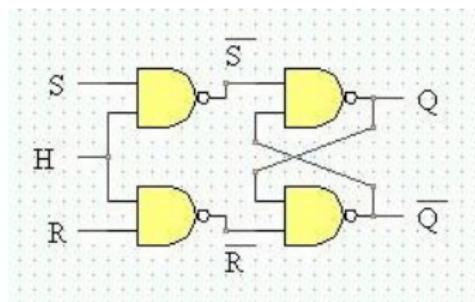
R	S	Q	\bar{Q}	
0	0	Q	\bar{Q}	état mémoire
0	1	1	0	mise à 1
1	0	0	1	mise à 0
1	1	0	0	état à proscrire

Bascule RS



R	S	\bar{R}	\bar{S}	Q	\bar{Q}	
0	0	1	1	Q	\bar{Q}	état mémoire
0	1	1	0	1	0	mise à 1
1	0	0	1	0	1	mise à 0
1	1	0	0	1	1	état à proscrire

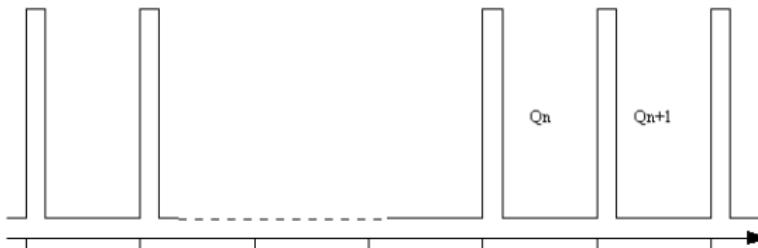
Bascule RST - RSClock



- Lorsque H vaut 1, le fonctionnement est identique à une bascule RS
- Lorsque H vaut 0, les sorties Q et \bar{Q} ne sont pas affectées par les changements de R et S . La bascule est bloquée. Les valeurs de Q et \bar{Q} sont celles qui avaient cours lorsque H est passé de 1 à 0.
- La bascule est synchronisée par H .

Signal d'horloge

- H peut être un signal d'horloge, c'est à dire une succession d'impulsions périodiques de largeur t_p petite par rapport à la période T .
- Les sorties Q et \bar{Q} ne peuvent évoluer que lors de ces instants très courts où H vaut 1.
- Q et \bar{Q} restent bloquées le reste du temps
- On note Q_n la valeur de Q précédent la n^{e} impulsion
- On note Q_{n+1} la valeur de Q suivant la n^{e} impulsion



Bascule RST – RSClock

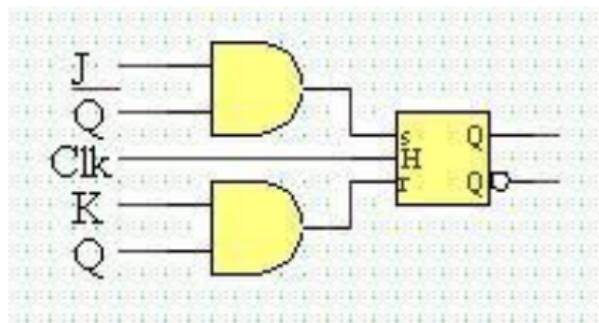
R	S	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	?
1	1	1	?

R	S	Q_{n+1}
		Q_n
0	0	0
0	1	1
1	0	0
1	1	?

\Rightarrow

Bascule JK

La bascule JK permet de résoudre le problème de la combinaison interdite de la bascule RS



$$\begin{cases} S = J\bar{Q} \\ R = KQ \end{cases}$$

Bascule JK

J	K	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	0

\Rightarrow

J	K	Q_{n+1}
		Q_n
0	0	0
0	1	0
1	0	1
1	1	$\overline{Q_n}$

On ne rencontre pas la combinaison $S = 1$ et $R = 1$.

Bascule JK

On obtient la table de transition donnant les valeurs des entrées J et K en fonction des transitions de Q_n à Q_{n+1} .

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

\Rightarrow

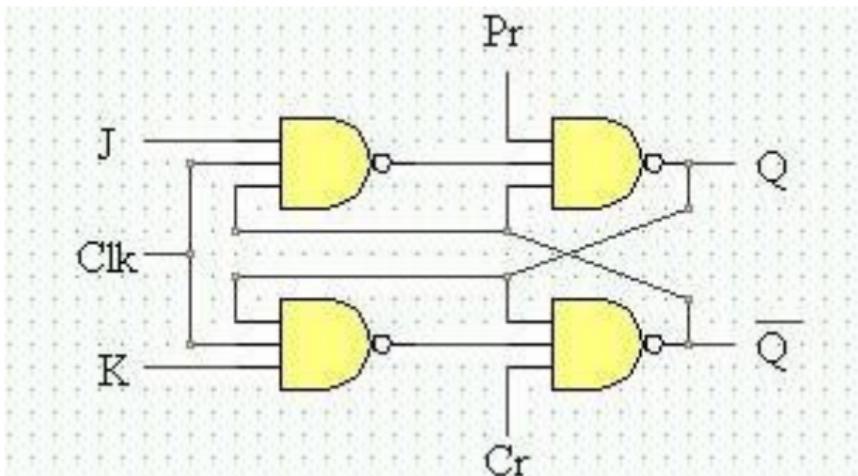
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

On peut également déduire l'expression de Q_{n+1}

$$Q_{n+1} = J\overline{Q_n} + \overline{K}Q_n$$

Entrées asynchrones

- La bascule JK dispose d'entrées *Cr* et *Pr* dites asynchrones
- Elle permettent de positionner l'état de la bascule indépendamment de l'horloge



Entrées asynchrones

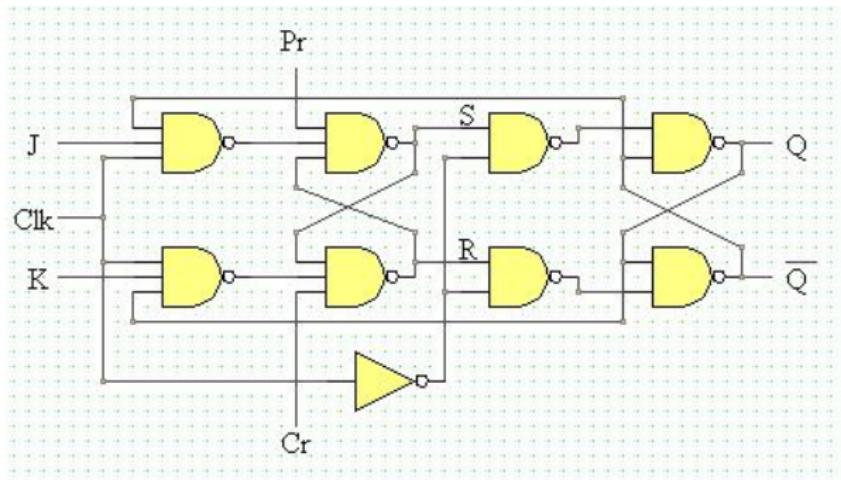
- Ces entrées sont dites actives à l'état bas, c'est-à-dire qu'elles agissent lorsqu'elles valent 0
- Lorsque ces entrées sont toutes les deux à 1 la bascule fonctionne normalement
- Lorsque Pr (Preset) vaut 0, la sortie Q est positionnée à 1
- Lorsque Cr (Clear) vaut 0, la sortie est positionnée à 0

Cr	Pr	Q
1	1	Q
1	0	1
0	1	0

Bascule JK Maître-Esclave

- La combinaison $J = 1$ et $K = 1$ a pour vocation de faire commuter (inverser) la sortie de la bascule.
- Il est possible que la sortie commute plusieurs fois durant la largeur de l'impulsion de H , alors qu'une seule commutation est désirée.
- Il n'est alors pas possible de prévoir la valeur de la sortie.
- Pour palier ce problème, on propose le schéma de la bascule JK Maître-Esclave

Bascule JK Maître-Esclave



Une bascule JK Maître-Esclave est constituée de 2 bascules dont le signal d'horloge est inversé.

Bascule JK Maître-Esclave

- Lorsque l'une est bloquée, l'autre est libérée
- Les entrées de la première bascule (Maître) sont asservies par les sorties de la seconde (Esclave).
- Pendant la n^{e} impulsion, le signal d'horloge est haut pour la bascule Maître et bas pour la bascule esclave
- Lorsque le signal d'horloge passe à 0, la bascule Maître est bloquée et la bascule esclave est libérée.

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} Q_M = 1 \\ \overline{Q_M} = 0 \end{array} \right\} \\ \left\{ \begin{array}{l} Q_M = 0 \\ \overline{Q_M} = 1 \end{array} \right\} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} S = 1 \\ R = 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \overline{Q_{n+1}} = 1 \\ \overline{Q_{n+1}} = 0 \end{array} \right\}$$

$$\Rightarrow \left\{ \begin{array}{l} S = 0 \\ R = 1 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \overline{Q_{n+1}} = 0 \\ \overline{Q_{n+1}} = 1 \end{array} \right\}$$

- L'état de la bascule Maître est transmis à la bascule Esclave lors du front descendant du signal d'horloge.

Bascule D

- Une bascule D (delay) est construite à partir d'une bascule JK
- $J = D$ et $K = \overline{D}$
- La bascule D est le plus souvent synchronisée sur front du signal d'horloge.

D	Q_n	J	K	Q_{n+1}		D	Q_{n+1}
0	0	0	1	0		0	0
0	1	0	1	0		1	1
1	0	1	0	1			
1	1	1	0	1			

Bascule D

- L'entrée est répercutée sur la sortie au moment du front.
- Entre deux fronts, quelle que soit l'évolution de l'entrée D , la sortie reste inchangée.
- La bascule a mémorisé la valeur lors du dernier front

Bascule T

- La bascule T (trigger) est construite à partir d'une bascule JK
- $J = T$ et $K = T$
- La bascule T fonctionne le plus souvent sur front du signal d'horloge

T	Q_n	J	K	Q_{n+1}		T	Q_{n+1}
0	0	0	0	0		0	Q_n
0	1	0	0	1		1	\overline{Q}_n
1	0	1	1	1			
1	1	1	1	0			

\Rightarrow

T	Q_{n+1}
0	Q_n
1	\overline{Q}_n

Plan

1 Définitions

2 Les bascules

- Bascule RS
- Bascule RST – RSClock
- Bascule JK
- Bascule D
- Bascule T

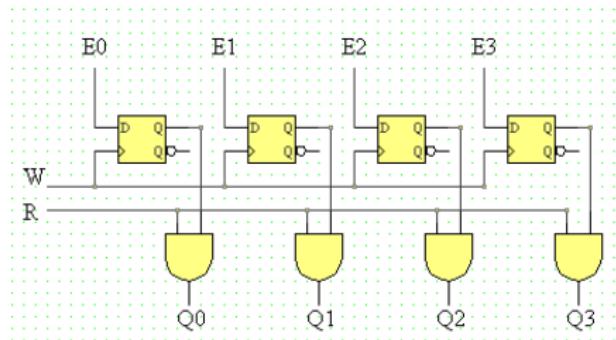
3 Les registres

- Registre de mémorisation
- Registre à décalage

4 Les séquenceurs

- Séquenceurs asynchrones
- Séquenceurs synchrones

Registre de mémorisation



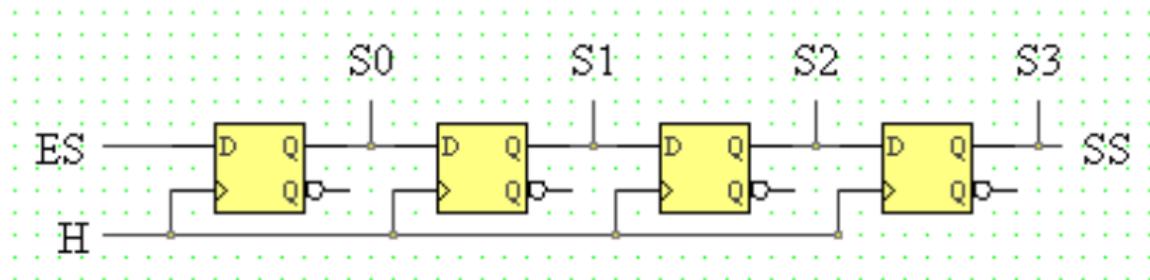
- Un registre permet la mémorisation de n bits
- Il est constitué d'autant de bascules
- Un front montant sur W permet à chaque bascule de mémoriser la valeur qui lui est présentée en entrée
- Un niveau haut sur R permet de répercuter les valeurs mémorisées sur les bascules sur les sorties du registre

Registre à décalage

Dans un registre à décalage, les bascules sont connectées de telle sorte que :

- La sortie de la i^{e} bascule soit l'entrée de la $(i + 1)^{\text{e}}$ bascule.
On obtient un registre à décalage à gauche (multiplication par 2)
- La sortie de la i^{e} bascule soit l'entrée de la $(i - 1)^{\text{e}}$ bascule.
On obtient un registre à décalage à droite (division par 2)
- Les registres à décalage peuvent être utilisés pour transformer
 - une entrée série en sortie parallèle
 - une entrée parallèle en sortie série

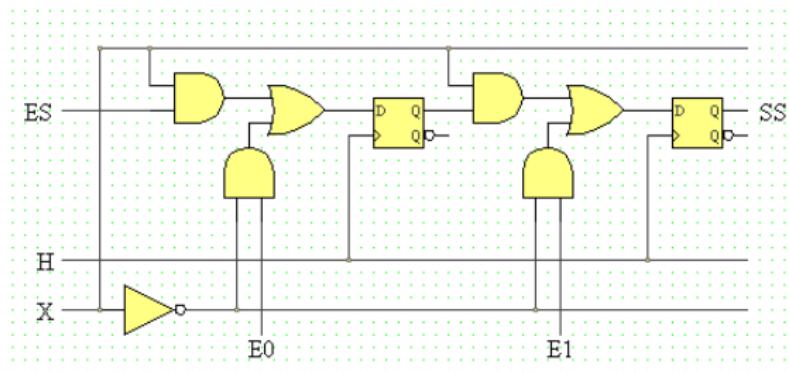
Entrée série – sortie parallèle



À chaque front montant de l'horloge H ,

- l'entrée série ES est répercuté sur la sortie S_0
- S_0 sur S_1
- S_1 sur S_2
- S_2 sur S_3

Sortie série

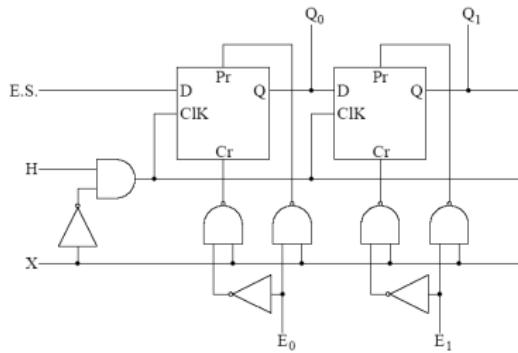


Registre à 2 bascules.

On sélectionne le mode d'entrée par le signal de commande X

- Si $X = 0$, l'entrée active est l'entrée parallèle
- Si $X = 1$, l'entrée active est l'entrée série

Entrée parallèle – Sortie parallèle



- Si $X = 0$, alors $Pr = Cr = 1$ ce qui implique un fonctionnement normal des bascules
- Si $X = 1$,

$$\left\{ \begin{array}{l} E_i = 0 \Rightarrow (Pr = 1, Cr = 0) \Rightarrow Q_i = 0 \\ E_i = 1 \Rightarrow (Pr = 0, Cr = 1) \Rightarrow Q_i = 1 \end{array} \right\} \Rightarrow Q_i = E_i$$

Plan

1 Définitions

2 Les bascules

- Bascule RS
- Bascule RST – RSClock
- Bascule JK
- Bascule D
- Bascule T

3 Les registres

- Registre de mémorisation
- Registre à décalage

4 Les séquenceurs

- Séquenceurs asynchrones
- Séquenceurs synchrones

Les séquenceurs : Définition

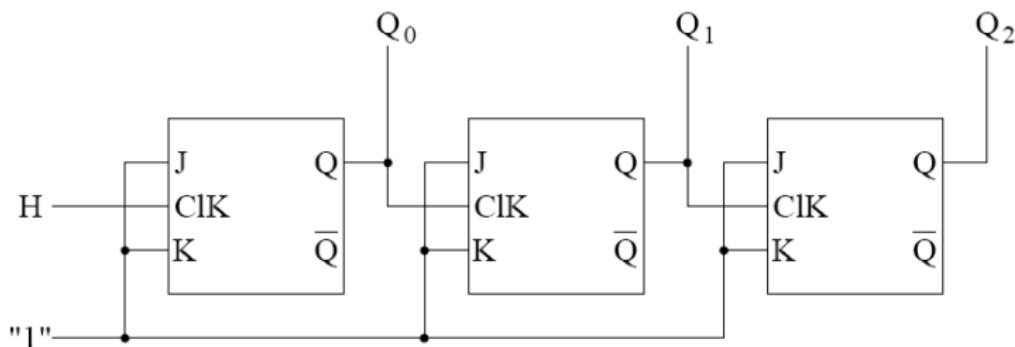
- Un séquenceur est un ensemble de n bascules interconnectées par des portes logiques permettant de mémoriser des mots de n bits.
- Au rythme d'un signal d'horloge, le séquenceur décrit une séquence d'états binaires
- La séquence peut comporter au maximum 2^n états (combinaisons). Ces états restent stables et accessibles entre deux impulsions
- Le nombre d'états est appelé *modulo* du séquenceur

$$N \leq 2^n$$

- On distingue les séquenceurs asynchrones et les séquenceurs synchrones

Séquenceurs asynchrones

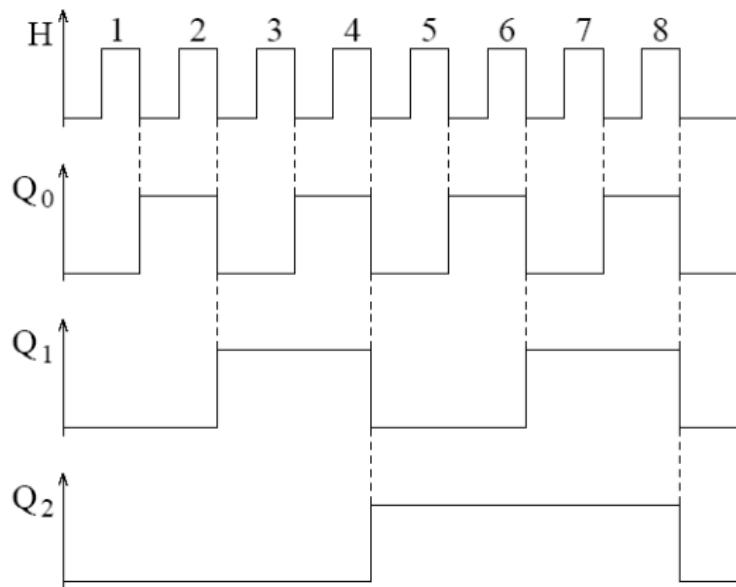
- Un séquenceur asynchrone est composé de n bascules fonctionnant en mode trigger
- Le signal d'horloge est reçu par la première bascule (LSB : Least Significant Bit)
- Le signal d'horloge des autres bascules est fourni par la sortie de la bascule précédente



Séquenceur asynchrone

- Les bascules fonctionnant en mode trigger, chaque sortie Q_i bascule sur le front actif de la bascule de rang i .
- Supposons que les bascules fonctionnent sur front descendant et que l'état initial de $Q_2 Q_1 Q_0$ soit 000 :
 - Q_0 commute sur un front descendant de H
 - Q_1 commute sur un front descendant de Q_0
 - Q_2 commute sur un front descendant de Q_1

Séquenceur asynchrone

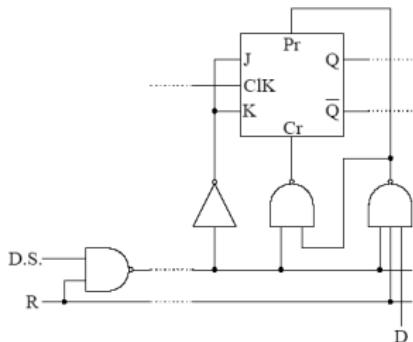


Séquenceur asynchrone

impulsion	Q_2	Q_1	Q_0
état initial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

- Il s'agit donc d'un compteur modulo 8 comptant de 0 à 7
- On obtiendrait un décompteur de 7 à 0 en lisant les sorties \bar{Q}

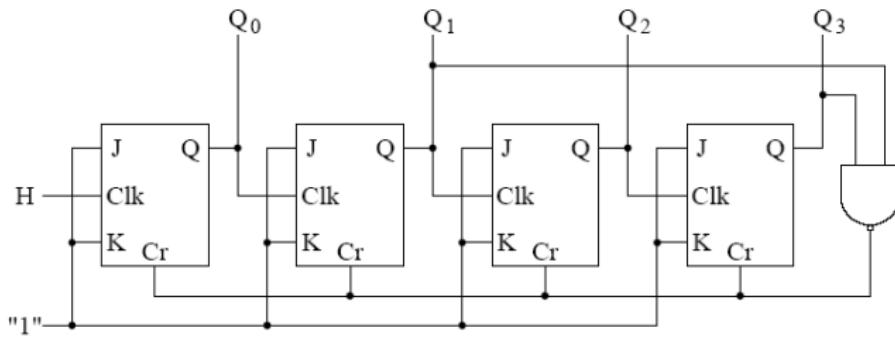
Remize à zéro – Chargement



- Pour une mise à zéro, on positionne R à 0 ce qui implique $J = K = 0$ interdisant le basculement et $Pr = 1$ et $Cr = 0$
- Pour charger une valeur, on positionne $R = 1$ et $DS = 0$ et la valeur à charger sur D . Ainsi, $J = K = 0$ interdisant le basculement
 - Si $D = 0$, alors $Pr = 1$ et $Cr = 0$ et donc $Q = 0$
 - Si $D = 1$, alors $Pr = 0$ et $Cr = 1$ et donc $Q = 1$

Cycle incomplet

- Si le modulo N n'est pas une puissance de 2, on utilise n bascules avec n tel que $2^{n-1} < N < 2^n$
- On asservit les entrées asynchrones pour réinitialiser le cycle à la première combinaison interdite

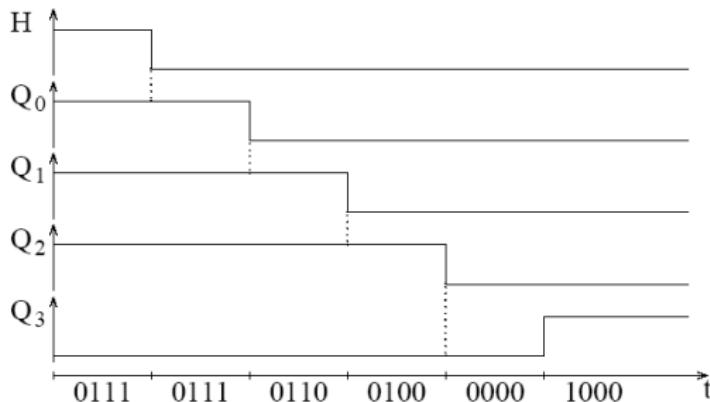


Inconvénient des séquenceurs asynchrones

- Chaque bascule a un temps de transition non nul
- Le signal d'horloge ne parvient pas à toutes les bascules au même instant
- Il existe des états transitoires indésirables entre l'intant où le signal d'horloge parvient à la première bascule et celui où il parvient à la seconde

Inconvénient des séquenceurs asynchrones

Par exemple lorsque qu'un séquenceur asynchrone modulo 15 passe de 0111 à 1000, il passe par les états transitoires 0110, 0100, 0000 et enfin 1000



Séquenceur synchrone

- Dans un séquenceur synchrone, toutes les bascules reçoivent le même signal d'horloge au même instant.
- Pour que le séquenceur décrive la séquence souhaitée, il faut définir les fonctions logiques des entrées de chaque bascule en fonction de l'état précédent du séquenceur
- On utilise la table de transition des bascules utilisées pour définir la combinatoire nécessaire pour positionner les entrées

Séquenceur synchrone : Exemple

- On souhaite réaliser un séquenceur synchrone modulo 8, comptant de 0 à 7 à l'aide de bascules T
- Ce séquenceur nécessite 3 bascules dont les entrées sont T_2 , T_1 et T_0 et dont les sorties sont Q_2 , Q_1 et Q_0

états	Instant n			Instant $n + 1$			Entrées		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	0	0	1	1	1
4	1	0	0	1	0	1	0	0	1
5	1	0	1	1	1	0	0	1	1
6	1	1	0	1	1	1	0	0	1
7	1	1	1	0	0	0	1	1	1

Séquenceur synchrone : Exemple

On en déduit les équations des entrées :

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_1 \cdot Q_0$$