

Chapitre 4 : Systèmes logiques séquentiels

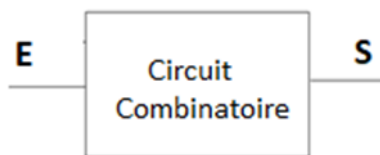
Plan

1. Introduction
2. Les bascules
 - 2.1 Bascule RS asynchrone
 - 2.2 Bascule RS Synchrone
 - 2.3 Bascule JK,
 - 2.4 bascule D
 - 2.5 Toggle,
3. Analyse d'un circuit Séquentiel
4. Synthèse d'un circuit séquentiel
5. Les compteurs
6. Les registres
7. Les mémoires

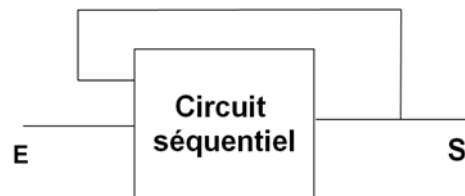
1. Introduction

1.1. Définition

1. Un circuit combinatoire est un circuit numérique dont les sorties dépendent uniquement des entrées :
2. Dans les circuits séquentiels, les signaux de sortie dépendent des entrées, mais aussi des états antérieurs : ils ont une mémoire du passé.



$$S = f(E)$$



$$S_{t+1} = f(E, S_t)$$

Exemple :

Soit une lampe L initialement éteinte (L=0) et 2 boutons poussoirs M et A.

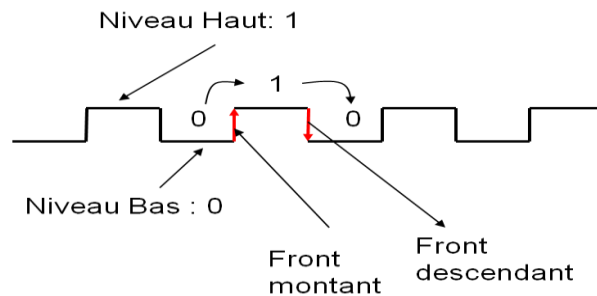
M	A	L	
0	0	0	Aucun bouton enfoncé: lampe éteinte
1	0	1	Appui sur « M »: lampe s'allume
0	0	1	Relâche « M »: lampe reste allumée
0	1	0	Appui sur « A »: lampe s'éteint
0	0	0	Relâche « A »: lampe reste éteinte

2 états de sortie (0 ; 1) différents pour un même vecteur d'entrée : (0 0) : Pas possible en combinatoire

1.2 Circuits asynchrones et synchrones

a. Horloge

- Une horloge est une variable logique qui passe successivement de 0 à 1 et de 1 à 0 d'une façon périodique.



- L'horloge est notée par **h**, **c** ou **ck** (clock).

b. Circuits asynchrones et synchrones

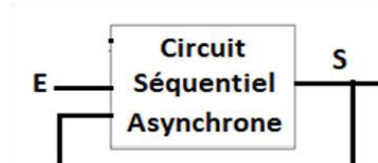
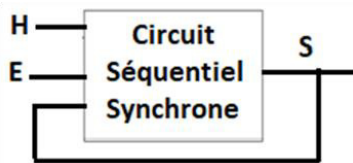
Il existe deux types de circuits séquentiels

✓ Systèmes asynchrones

Quand un circuit séquentiel n'a pas d'horloge comme variable d'entrée alors il est asynchrone. Dans ce cas, les sorties réagissent immédiatement aux variations des entrées, ce qui peut provoquer des états transitoires, des retards de durées différentes et des risques d'instabilité.

✓ Systèmes Synchrones:

Le système mémorise l'état présent sur son entrée si et seulement si une horloge fournit un signal de synchronisation. Les circuits synchrones sont plus simples à synthétiser et à analyser.



2. Bascules (Bistables)

Les bascules sont des circuits de base de la logique séquentielle. Une bascule est une mémoire unitaire (1 bit) ayant 2 états : **Q** et **\bar{Q}** .

Il existe plusieurs types de bascules: RS, RST, D, JK, T

2.1 La bascule RS asynchrone

✓ Entrée/sorties

- 2 entrées : **R** et **S**
- 2 sorties **Q** et **\bar{Q}** qui correspond à l'état stocké et son inverse

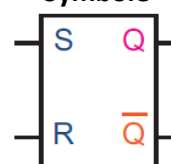
✓ Principe : la valeur de **Q** à **t + 1** dépend de **R** (reset) et **S** (set) de la valeur de **Q** à **t**

- **S = 1 et R = 0** : **Q** mis à 1
- **S = 0 et R = 1** : **Q** mis à 0
- L'état **R = S = 0** (mode mémoire) maintient l'état de la sortie.
- L'état **R = S = 1** (mode interdit) est interdit car il conduit à mettre simultanément la sortie à 1 et à 0.

Table de fonctionnement

R	S	Q ⁺	
0	0	Q ⁻	Mémoire
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	Φ	Interdit

Symbole



a) Réalisation à l'aide des portes Nand

Table de vérité

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	Φ
1	1	1	Φ

Mémoire

Mise à 1

Mise à 0

Interdit

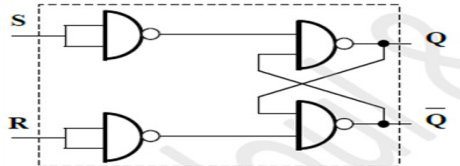
Table de Karnaugh

RS	00	01	11	10
Q ⁻ 0	0	1	Φ	0
1	1	1	Φ	0

Equation logique

$$\begin{aligned}
 Q_{n+1} &= S + Q_n \bar{R} \\
 &= \overline{\overline{S} + \overline{Q_n \bar{R}}} \\
 &= \overline{\overline{S} \cdot Q_n \bar{R}}
 \end{aligned}$$

Circuit logique



- Cette bascule RS est prioritaire au 1 car, pour la combinaison $R=S=1$, la sortie Q est mise à 1 (les Φ ayant été fixés à 1 pour la simplification de Q).
- Remarque : le logigramme fait apparaître une sortie supplémentaire égale au complément de la sortie Q uniquement si la combinaison $R=S=1$ n'apparaît pas

b) Réalisation à l'aide des portes Nor

Table de vérité

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	Φ
1	1	1	Φ

Mémoire

Mise à 1

Mise à 0

Interdit

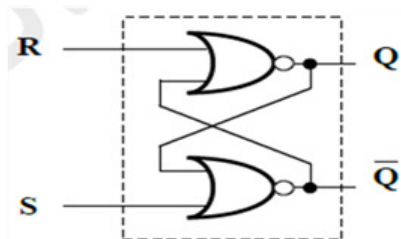
Table de Karnaugh

RS	00	01	11	10
Q ⁻ 0	0	1	Φ	0
1	1	1	Φ	0

Equation logique

$$\begin{aligned}
 Q_{n+1} &= \bar{R} \cdot (Q_n + S) \\
 &= \overline{\overline{\bar{R} \cdot (Q_n + S)}} \\
 &= \overline{\overline{\bar{R}} \cdot \overline{(Q_n + S)}} \\
 &= \overline{\overline{\bar{R}} \cdot (\bar{Q}_n \bar{S})} \\
 &= \overline{\bar{R} \cdot (Q_n + S)}
 \end{aligned}$$

Circuit logique



Cette bascule RS est prioritaire au 0 car, pour la combinaison $R=S=1$, la sortie Q est mise à 0 (les Φ ayant été fixés à 0 pour la simplification de Q).

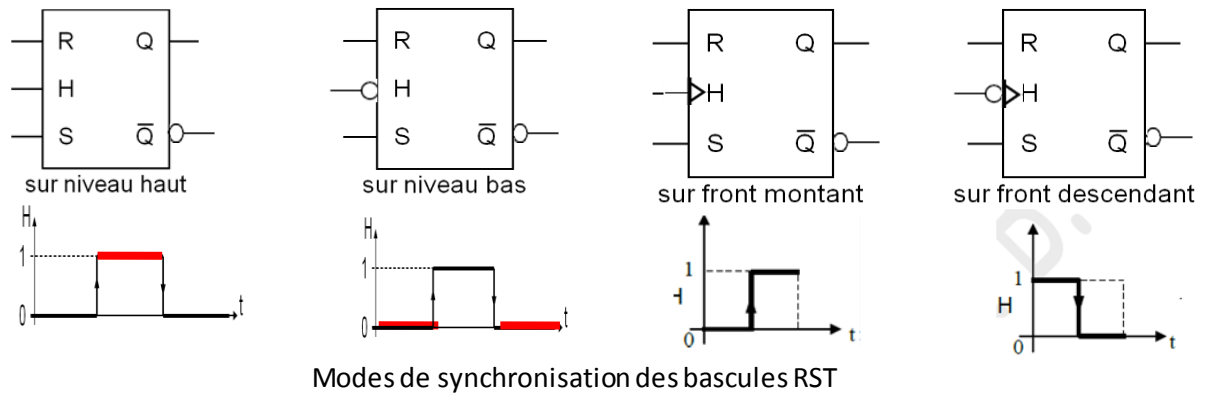
- ♦ Remarque : le logigramme fait apparaître une sortie supplémentaire égale au complément de la sortie Q uniquement si la combinaison $R=S=1$ n'apparaît pas.

2.2 La bascule RS synchrone

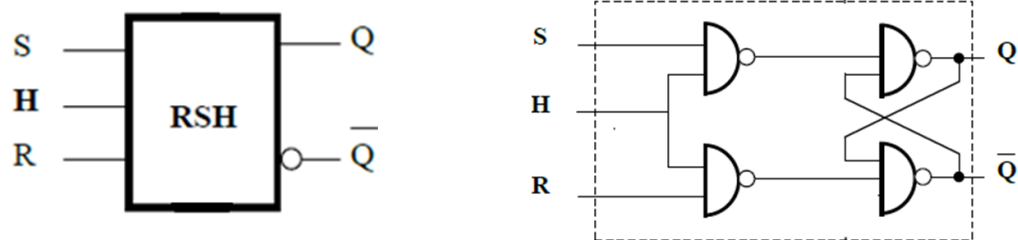
Pour une bascule RS synchrone l'enclenchement et le déclenchement ne sont donc autorisés qu'à la présence du signal d'horloge H.

La bascule RS synchrone fonctionne en fonction de l'horloge selon 4 Modes :

- niveau haut : la bascule n'est active que si l'horloge est à 1
- niveau bas : la bascule n'est active que si l'horloge est à 0
- sur le front montant de l'horloge
- Sur le front descendant de l'horloge

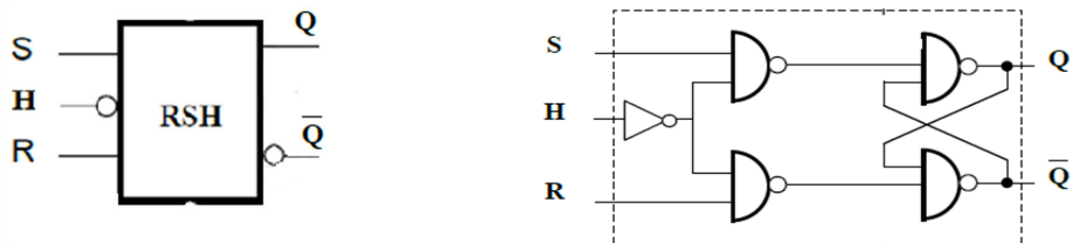


Exemple1 : La bascule RST synchronisée par le niveau haut de l'horloge :



- Si $H=0$: les sorties \bar{S} et \bar{R} sont bloquées à 1 quelques soient R et S , la sortie garde l'état précédent.
- Si $H=1$: la bascule **RS** fonctionne normalement les sorties obéissent aux entrées.
- Donc la bascule **RS** ne fonctionne normalement que si $H=1$ (Niveau Haut).

Exemple2: La bascule RST synchronisée par le niveau bas de l'horloge



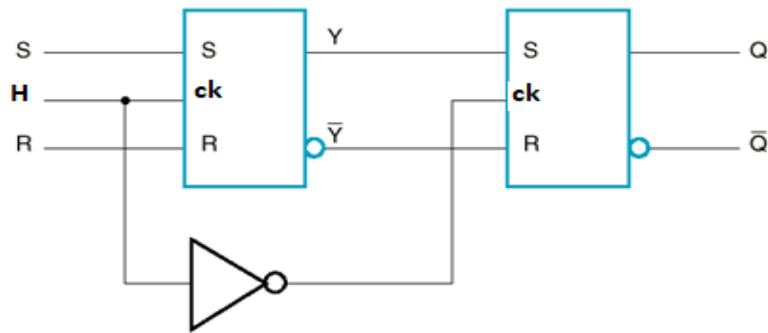
- Si $H=1$: les sorties \bar{S} et \bar{R} sont bloquées à 1 quelques soient R et S , la bascule est inactive la sortie garde l'état précédent (mémoire).
- Si $H=0$: la bascule **RS** fonctionne normalement les sorties obéissent aux entrées.
- Donc la bascule **RS** n'est active que si $H=0$ (Niveau bas).

Remarque

La synchronisation sur niveau a beaucoup d'inconvénients : la bascule est sensible aux entrées pendant toute la durée de l'état de l'horloge pour niveau haut (ou 0 pour le niveau bas). Si, pendant que $H=1$ (ou $H=0$), des parasites apparaissent sur les entrées S et R , ils peuvent entraîner des changements d'état imprévus sur la sortie Q .

Afin de minimiser au maximum la durée de cet état sensible, on s'arrange pour que la bascule reste dans son état mémoire sauf pendant un bref instant, juste au moment où l'entrée passe de 0 à 1 (ou de 1 à 0). La bascule est dite synchronisée sur front.

La bascule RS Maître- Esclave (active sur front)



Si $H=1$ la première bascule est active alors que la deuxième est bloquée, et quand $H=0$ la première bascule est bloquée et transmet la commande à la deuxième qui fonctionne normalement ; les deux bascules fonctionnent donc ensemble au moment de passage de H de 1 à 0 c'est-à-dire au moment du front descendant.

Donc toute la bascule maître esclave dont la maître travaille sur le niveau haut et l'esclave travaille sur le niveau bas est une bascule synchronisée sur front descendant.

De même une bascule maître esclave dont le maître travaille sur niveau bas de l'horloge, et l'esclave sur le niveau bas, est une bascule synchronisée sur front montant.

2.3 bascule J-K

- La bascule JK (variante de RS) est une bascule synchrone (le plus souvent sur front) qui possède une entrée J de mise à 1 (S), une entrée K de mise à 0 (R), une entrée d'horloge H, une sortie Q et une sortie complément de Q.
- La différence entre la bascule JK et la bascule RS réside dans le fait qu'il n'y a plus l'état indéfini. La combinaison $J=1$ et $K=1$ est utilisée pour faire basculer la sortie, dans ce cas elle passe à l'état complémentaire : $Q^+ = \overline{Q}$ (**basculement**).

J	K	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J	K	Q+	
0	0	Q	mémoire
0	1	0	mise à 0
1	0	1	mise à 1
1	1	\overline{Q}	basculement

J	K	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

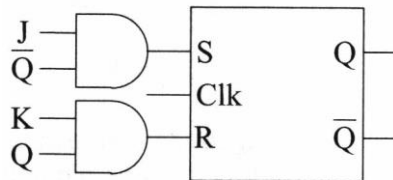
Q	J	K	Q+
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$Q^+ = J\overline{Q} + \overline{K}Q$$

$$Q_{n+1} = S + Q_n \overline{R}$$

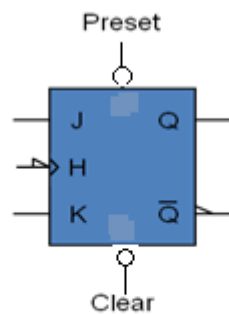
On obtient la bascule JK à partir d'une bascule SR en posant :

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



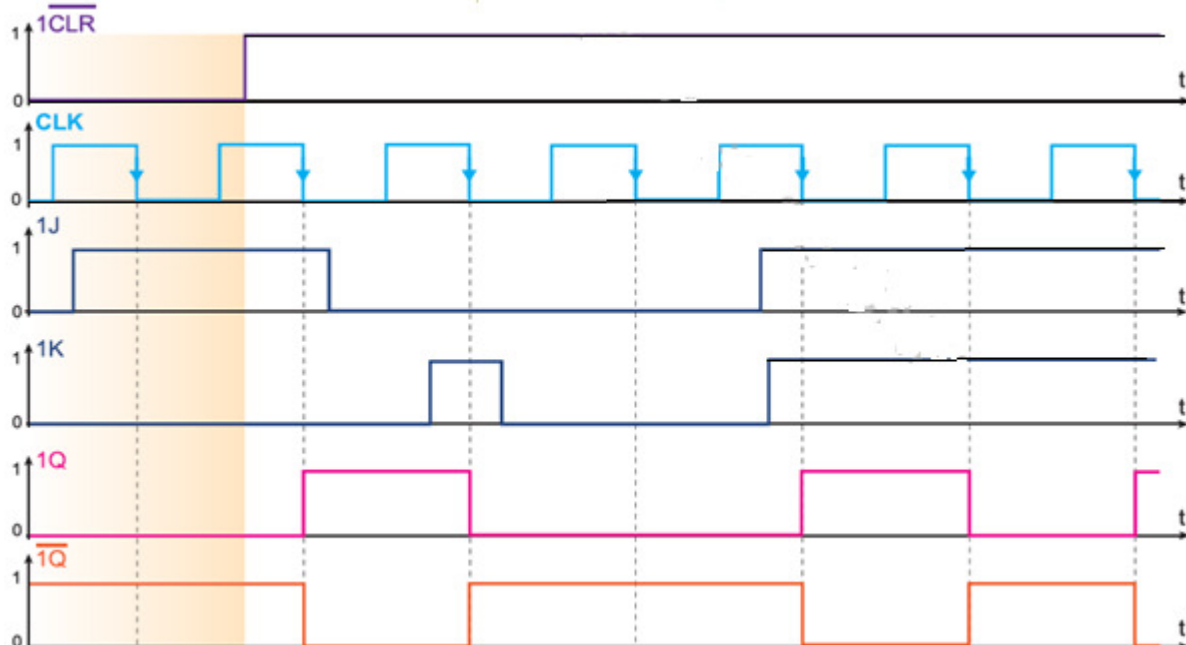
Les bascules J.K en mode asynchrone

- toutes les bascules synchrones commercialisées possèdent des entrées asynchrones de forçage **de mise à 0 (Cr ou Clear)**, et **de mise à 1 (Pr ou Preset)** prioritaires sur toutes autres entrées.
- Pr et Cr fonctionnent avec la logique negative, en fonctionnement normal, elles doivent donc être maintenues à 1



Bascule Jk active sur front descendant avec des entrées asynchrones

Ci-dessous un exemple d'un chronogramme d'une Bascule JK active sur front descendant avec l'entrée asynchrone Clear



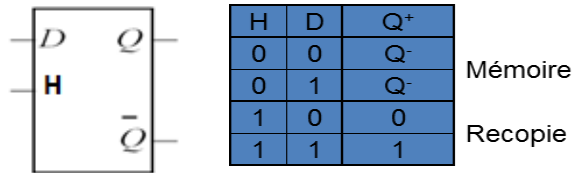
Applications des bascules JK :

- Diviseurs de fréquences
- Réalisation des compteurs

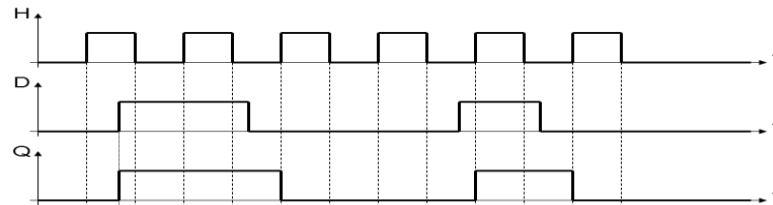
2.4 Bascule D

- La bascule D est une bascule synchrone qui possède une entrée de donnée D (Data), une entrée d'horloge H, une sortie Q et une sortie complément de Q.
- Le signal de synchronisation est actif :
 - soit sur un niveau (haut ou bas) de l'horloge (bascule D latch)
 - soit sur un front (montant ou descendant) de l'horloge (bascule D edge triggered)

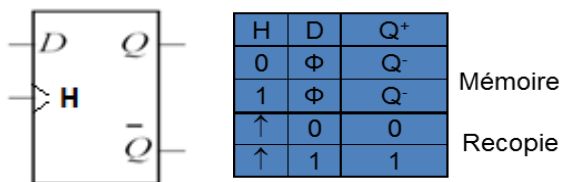
Exemple Bascule **D latch** synchronisée par le niveau haut :



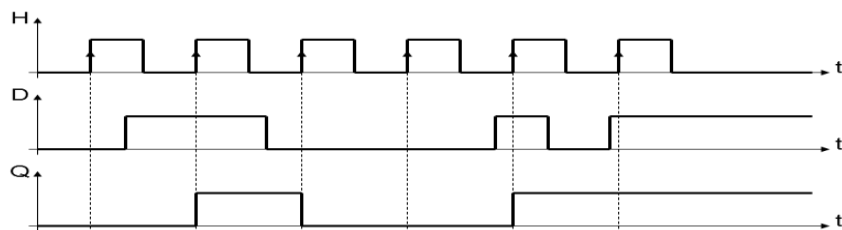
la sortie recopie l'entrée sur le niveau haut d'horloge $Q^+ = D$. Sur le niveau bas, la sortie est mémorisée



Exemple : Bascule **D edge triggered** synchronisée par le front montant :

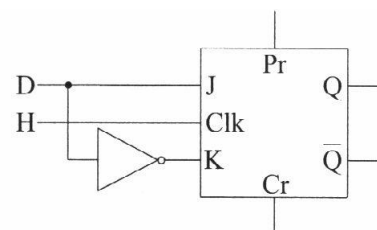


La sortie recopie l'entrée sur un front montant d'horloge sinon elle ne change pas d'état (maintien de l'état, mémorisation).



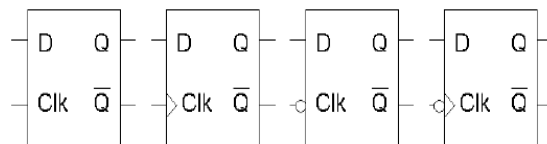
Réalisation

On peut réaliser une bascule D à partir d'une bascule **SR** ou (**JK**) En envoyant une donnée D sur l'entrée **S** ou (**J**) et son inverse sur l'entrée **R** ou (**K**)



Représentation des différents types de bascules **D** ; de gauche à droite :

- Bascule D latch sur niveau haut
- Bascule D flip-flop sur front montant
- Bascule D latch sur niveau bas
- Bascule D flip-flop sur front descendant

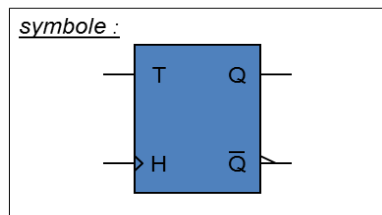


Applications

Réalisation de Registres : Mémorisation des mots binaires, décalage

2.5 La bascule T (Toggle)

- La bascule T est une bascule synchrone qui possède une entrée de donnée T, une entrée d'horloge H, une sortie Q et une sortie inverse \bar{Q} .

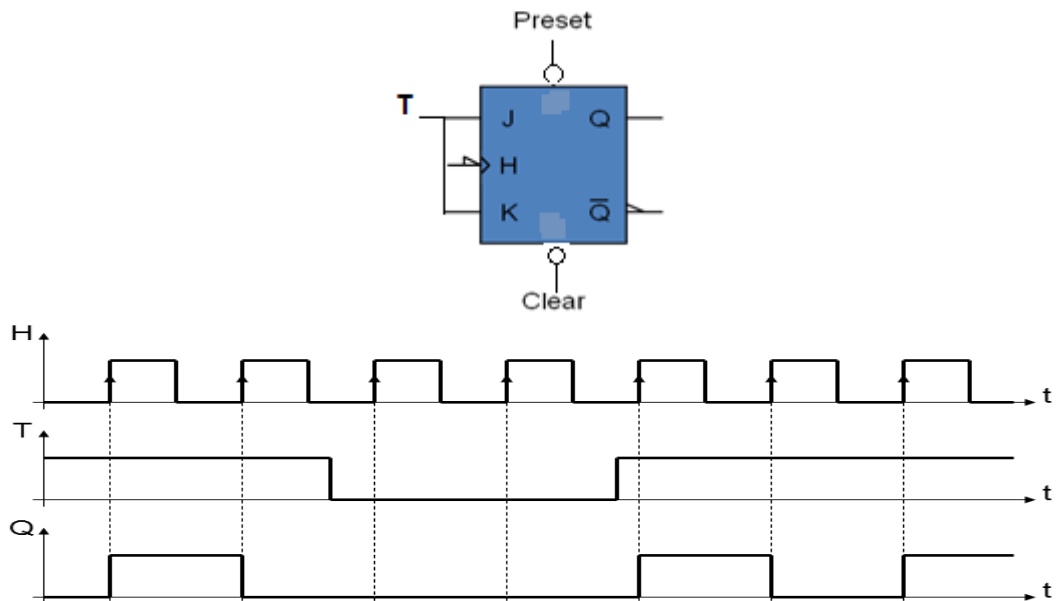


H	T	Q ⁺
0	Φ	Q ⁻
1	Φ	Q ⁻
\uparrow	0	Q ⁻
\uparrow	1	\bar{Q}

Mémoire

Inversion

- Son fonctionnement est un cas particulier de la bascule JK ou les entrées J et K sont connectées ensemble (0 ou mises à 1).



3. Analyse d'un circuit séquentiel

3.1 Concept d'automate fini

Un automate fini possède un nombre fini d'éléments et de mémoires.

Un automate fini ne peut prendre que 2^n états appelés états internes, où n est le nombre de bits de mémoire.

On peut caractériser un automate par :

- Sa fonction de transfert
- Sa table de transition
- Son diagramme d'états ou de transition

Exemple :

Diagramme d'état ou de transition

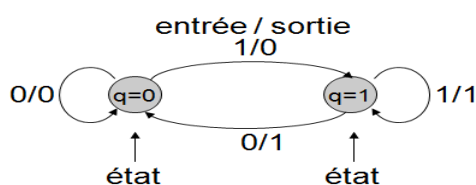


Table de transition

q(t)	e(t)	0	1
0		0	1
1		0	1

q(t)	e(t)	0	1
0		0	0
1		1	1

Fonction de transfert :

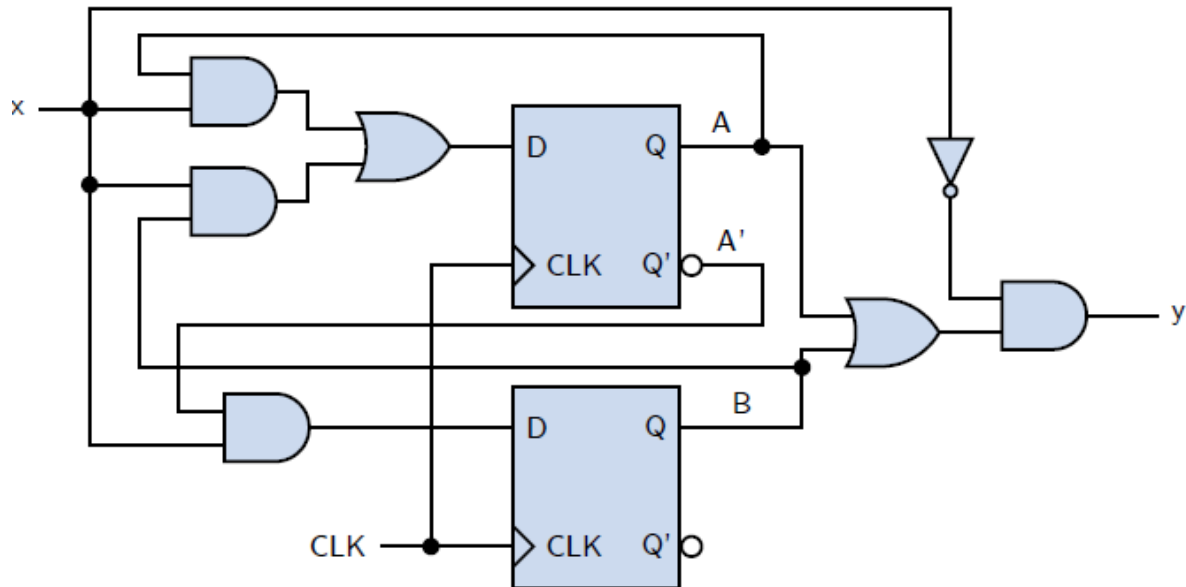
$$q(t+1) = e(t)$$

$$s(t) = q(t)$$

3.2 Analyse d'un circuit séquentiel

- L'analyse est un peu différente de l'analyse de circuits combinatoires
- Le comportement du circuit dépend des entrées, des sorties, et de l'état des bascules
- Il faut obtenir une table ou un diagramme pour la séquence d'entrées, de sorties et d'états internes

Exemple :



Équations pour le prochain état:

$$A(t+1) = A(t)x(t) + B(t)x(t)$$

$$B(t+1) = \bar{A}(t)x(t)$$

ou

$$A(t+1) = Ax + Bx$$

$$B(t+1) = \bar{A}x$$

Pour la sortie:

$$y(t) = (A + B)\bar{x}$$

Tableau d'état

- Tableau qui montre toutes les transitions selon la séquence d'entrées, de sorties et d'états
- Normalement quatre sections: état présent, entrée, prochain état et sortie
- Le prochain état est basé sur les équations d'état

État Présent			Prochain État		
A	B	Entrée x	A	B	Sortie y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Tableau d'état pour le circuit de l'exemple précédent

Tableau d'état, forme 2

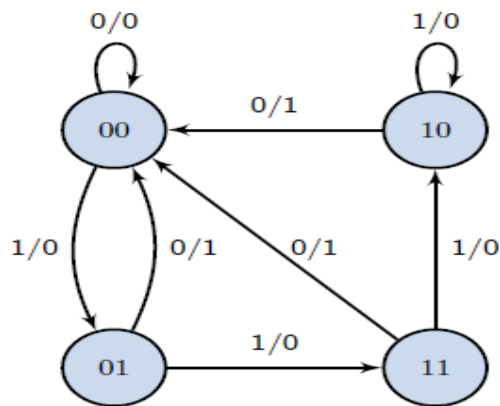
État Présent		Prochain État				Sortie	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Autre forme du tableau d'état pour le circuit de l'exemple précédent

Diagramme d'état

Autre forme graphique pour le tableau d'état

L'état est présenté comme un cercle, et les transitions (déclenchées par l'horloge) entre états sont représentés par des flèches qui se dirigent d'un cercle à l'autre



Exemple de diagramme d'état pour le circuit de l'exemple précédent

- Les chiffres binaires dans les cercles du diagramme représentent l'état actuel
- La transition est montrée par les flèches, selon l'entrée
- Les deux chiffres à côté de la flèche représentent l'entrée et la sortie
- Exemple: si le circuit est dans l'état 00, et que l'entrée est 1, la sortie sera 0 (le 1/0) et le prochain état est 01

Gabriel Cormier

4. Synthèse d'un circuit séquentiel

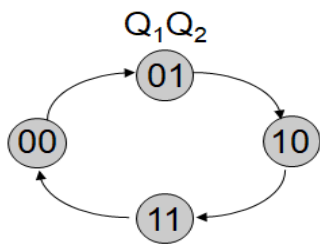
Pour faire la synthèse d'un circuit séquentiel,

- on établit d'abord son diagramme de transition.
- On construit ensuite sa table d'états.
- On réalise le circuit combinatoire associé à chaque bistable.
- On réalise le circuit combinatoire associé à chaque sortie.

Exemple 1 : compteur binaire synchrone modulo-4 sans entrée

Diagramme de transition

Table d'états



Réalisation au moyen de bistables D

Q_1^n	Q_2^n	Q_1^{n+1}	Q_2^{n+1}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Q_1^n	Q_2^n	Q_1^{n+1}	Q_2^{n+1}	D_1^n	D_2^n
0	0	0	1	0	1
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0

$$D_1^n = Q_1^{n+1} = Q_1^n \oplus Q_2^n$$

$$D_2^n = Q_2^{n+1} = \overline{Q_2^n}$$

