

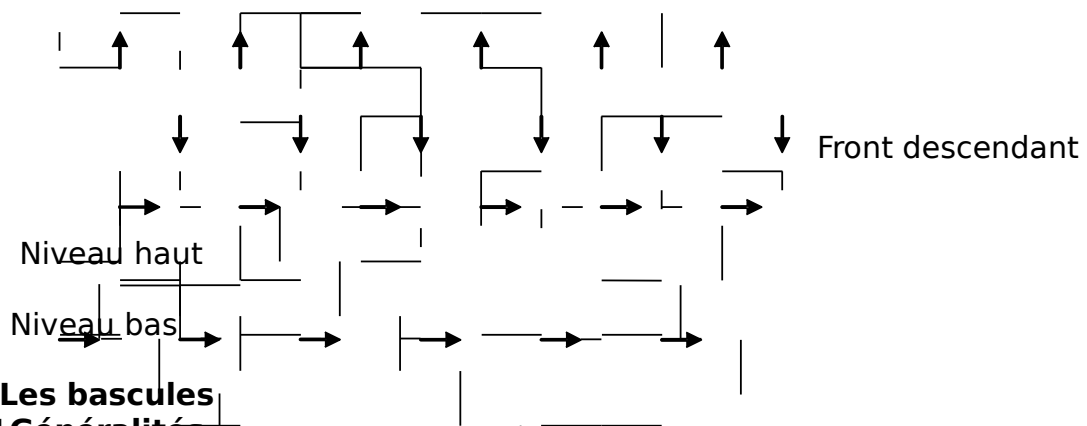
Logique séquentielle

1. Introduction

Dans les circuits logiques combinatoires, les sorties dépendent uniquement des entrées. Pour mettre au point des circuits logiques plus complexes (Unité de mémoire de contrôle d'un ordinateur), on a besoin de circuits dont les sorties dépendent non seulement des entrées mais aussi de leurs séquences. Ces circuits logiques qui ont une mémoire du passé sont appelés circuits séquentiels.

L'étude des circuits séquentiels repose sur la théorie des automates finis. Il existe deux types de circuits séquentiels : les circuits asynchrones où l'évolution du système dépend des signaux d'entrée et les circuits synchrones où le système évolue en fonction d'un signal d'horloge.

La synchronisation de l'horloge peut se faire soit sur un front (montant ou descendant) soit sur un niveau (haut ou bas).



2. Les bascules

2.1 Généralités

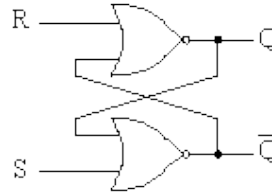
Une bascule a pour rôle de mémoriser une information élémentaire. C'est une mémoire à 1 bit. Une bascule possède deux sorties complémentaires Q et \bar{Q} .

Un système séquentiel est un circuit logique qui doit se rappeler de la succession des entrées jusqu'à l'instant t pour pouvoir déterminer les valeurs des sorties à l'instant $t+1$. Il a donc une mémoire du passé. Un tel circuit comportera nécessairement des éléments de mémoire des éléments de mémoire capable de conserver l'état qui sera pris en compte à l'instant $t+1$.

2.2 La bascule RS

La bascule RS est l'élément de mémoire le plus simple. Elle peut être réalisée avec deux portes NOR ou deux portes NAND. Les sorties de cette bascule sont complémentées et sont notées Q et \bar{Q} .

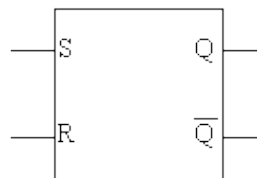
On considère dans un premier temps le circuit suivant :



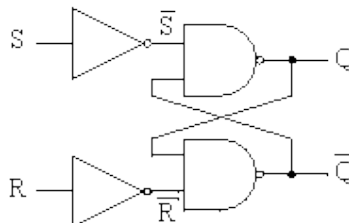
La table suivante donne la table de vérité correspondante :

R	S	Q_{t+1}	\bar{Q}_{t+1}	Commentaires
0	0	Q_t	\bar{Q}_t	Sorties inchangées: Mémoire
0	1	1	0	Set: Remise à Un ou Initialisation
1	0	0	1	Reset: Remise à Zéro ou Réinitialisation
1	1	0	0	Indétermination de l'état des sorties

Le schéma fonctionnel d'une bascule RS est donné d'après la figure suivante :



Considérons maintenant la bascule réalisée avec des portes NAND.



La table suivante donne la table de vérité correspondante :

R	S	\bar{R}	\bar{S}	Q_{t+1}	\bar{Q}_{t+1}	Commentaires
0	0	1	1	Q_t	\bar{Q}_t	Sorties inchangées: Mémoire
0	1	1	0	1	0	Set: Remise à Un ou Initialisation
1	0	0	1	0	1	Reset: Remise à Zéro ou Réinitialisation
1	1	0	0	0	0	Indétermination de l'état des sorties

La table d'états ou de transition :

R	S	Q _t	Q _{t+1}	Commentaires
0	0	0	0	Mémoire
0	0	1	1	
0	1	0	1	Mise à 1
0	1	1	1	
1	0	0	0	Mise à zéro
1	0	1	0	
1	1	0	X	Interdit
1	1	1	X	

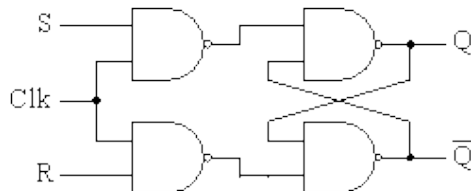
Q _t \ RS	00	01	11	10
0	0	1	X	0
1	1	1	X	1

Equation caractéristique:

$$Q_{t+1} = S + RQ$$

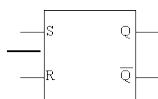
La bascule RS étudiée est une bascule asynchrone car la commutation à un nouvel état se fait immédiatement après application d'un signal actif aux entrées. Pour la synchroniser a besoin d'un signal d'horloge qui va déterminer l'instant de passage à l'état futur.

- Bascule RS synchrone ou Bascule RST ou RS Clock
C'est une bascule dont les entrées ne sont prises en compte qu'en coïncidence avec un signal de commande. Ce signal peut être fourni par une horloge, nous avons alors une bascule synchrone. Ce circuit peut être réalisé de la façon suivante :

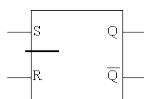


H	R	S	Q _{t+1}
0	X	X	Q _{t+1}
1	0	0	Q _t
1	0	1	1
1	1	0	0
1	1	1	X

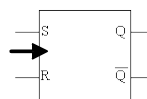
Symboles utilisés pour la synchronisation :



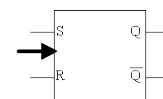
Niveau haut



Niveau bas



Front montant



Front descendant

2.3 La bascule D

La sortie recopie, avec un certain retard l'entrée D.

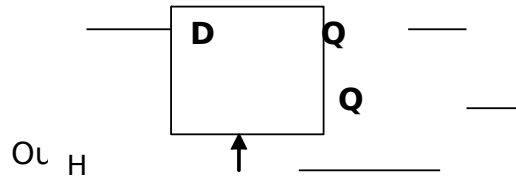
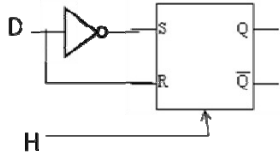


Table d'états :

D	Q _t	Q _{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

Equation caractéristique:

$$Q_{t+1} = D$$

2.4 La bascule T

La bascule T passe à l'état complétementé si au moment de l'impulsion d'horloge l'entrée T se trouve au niveau logique 1.

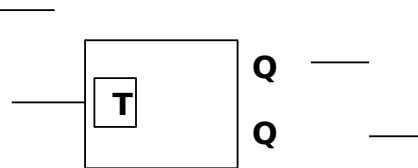
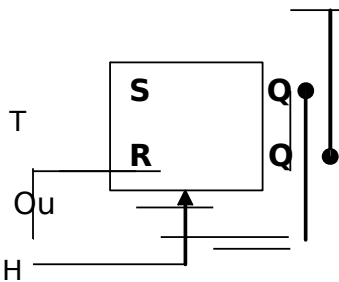


Table d'états

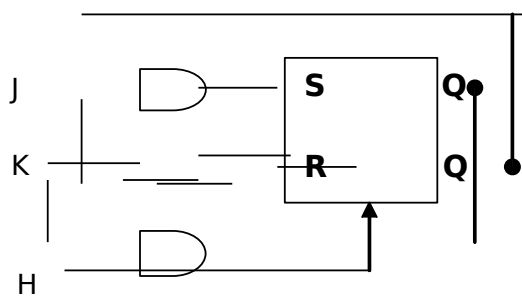
T	Q _t	Q _{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

Equation caractéristique:

$$Q_{t+1} = TQ + \bar{T}\bar{Q} = T \oplus Q_t$$

2.5 La bascule JK

La bascule JK permet de lever l'ambigüité qui existe dans la table d'état de la bascule RS. Ceci peut être obtenu en asservissant les entrées R et S aux sorties et Q selon le schéma logique indiqué comme suit :



Nous avons alors les signaux R et S

$$S = JQ$$

$$R = KQ$$

Ce qui nous permet de construire la table de vérité de la bascule JK.

J_t	K_t	Q_t	\bar{Q}_t	S	R	Q_{t+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

Nous constatons qu'on ne rencontre jamais la combinaison $S=R=1$. Cette table peut se résumer sous la forme suivante :

J_t	K_t	Q_{t+1}	
0	0	Q_t	Mémorisation
0	1	0	Mise à 0
1	0	1	Mise à 1
1	1	\bar{Q}_t	Complémentation

$Q_t \backslash J_t K_t$	00	01	11	10
0			1	1
1	1			1

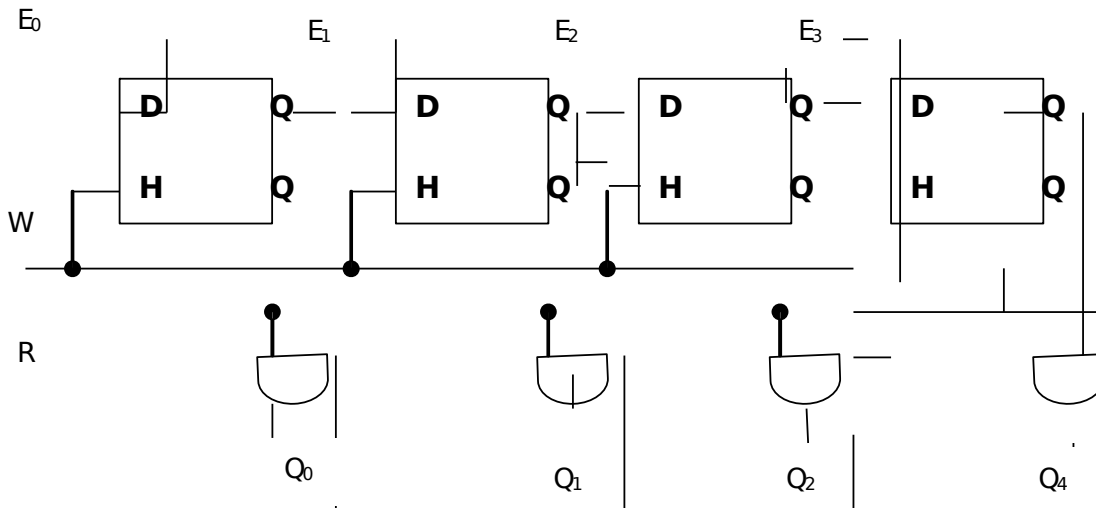
Equation caractéristique:

$$Q_{t+1} = JQ_t + K\bar{Q}_t$$

3. Les registres

Les bascules ne permettent de mémoriser qu'un seul bit. Pour pouvoir stocker plusieurs bits à la fois il est nécessaire de regrouper des bascules, on forme ainsi ce qu'on appelle un registre.

La figure suivante donne un exemple de registre à 4 bits réalisé avec 4 bascules D.



En synchronisation avec le signal d'horloge W le registre mémorise les données présentes sur les entrées E_1, E_2, E_3 . Elles sont conservées jusqu'au prochain signal de commande W.

Dans cet exemple, les états mémorisés peuvent être lus avec les sorties Q et Q en coïncidence avec un signal de commande R.

3.1 La fonction mémoire d'un registre

La mémorisation d'une information par un registre dépend de la nature de la bascule utilisée. Pour une bascule RS ($R=S=0$), pour une bascule JK ($J=K=0$), pour une bascule D ($D=Q$)

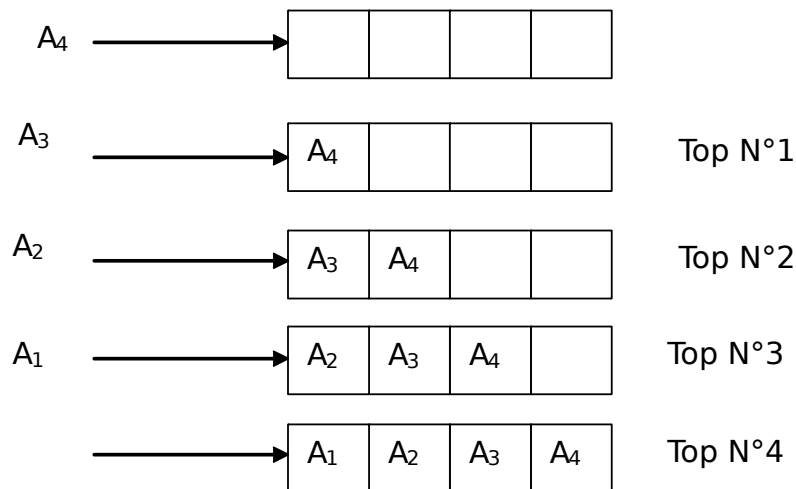
3.2 La fonction décalage d'un registre

Le principe du décalage consiste à faire recopier le bit de la bascule n dans la bascule n+1 ou n-1 selon le type de décalage.

L'information binaire peut être chargée ou lue en série ou en parallèle et une seule impulsion d'horloge suffit pour la circulation des bits.

- Entrée parallèle : C'est le cas d'un registre de mémorisation
- Entrée série : L'information est présentée séquentiellement bit après bit à l'entrée de la première bascule. A chaque signal d'horloge un nouveau bit est introduit pendant que ceux déjà mémorisés sont décalés d'un niveau dans le registre.

La figure suivante schématise le chargement d'un registre 4 bits en quatre coups d'horloge.



4. Les compteurs

Les compteurs sont des systèmes séquentiels logiques qui permettent d'enregistrer et d'indiquer un nombre d'événements se succédant dans le temps. Donc c'est un ensemble de n bascules interconnectées par des portes logiques. Ils peuvent donc mémoriser des mots de n bits. Au rythme d'une horloge, ils peuvent décrire une séquence déterminée c'est-à-dire occuper une suite d'états binaires. Il ne peut y avoir au maximum que 2 combinaisons. Ces états restent stables et accessibles entre les impulsions d'horloge. Le nombre total N de combinaisons successives est appelé le modulo du compteur.

On a $N \leq 2^n$ Si $N < 2^n$ un certain nombre d'états ne sont jamais utilisés.
 L'élément de base d'un compteur est une bascule (de type D ou JK).
 Les compteurs binaires peuvent être classés en deux catégories :

- Les compteurs asynchrones
- Les compteurs synchrones

4.1 Compteurs asynchrones

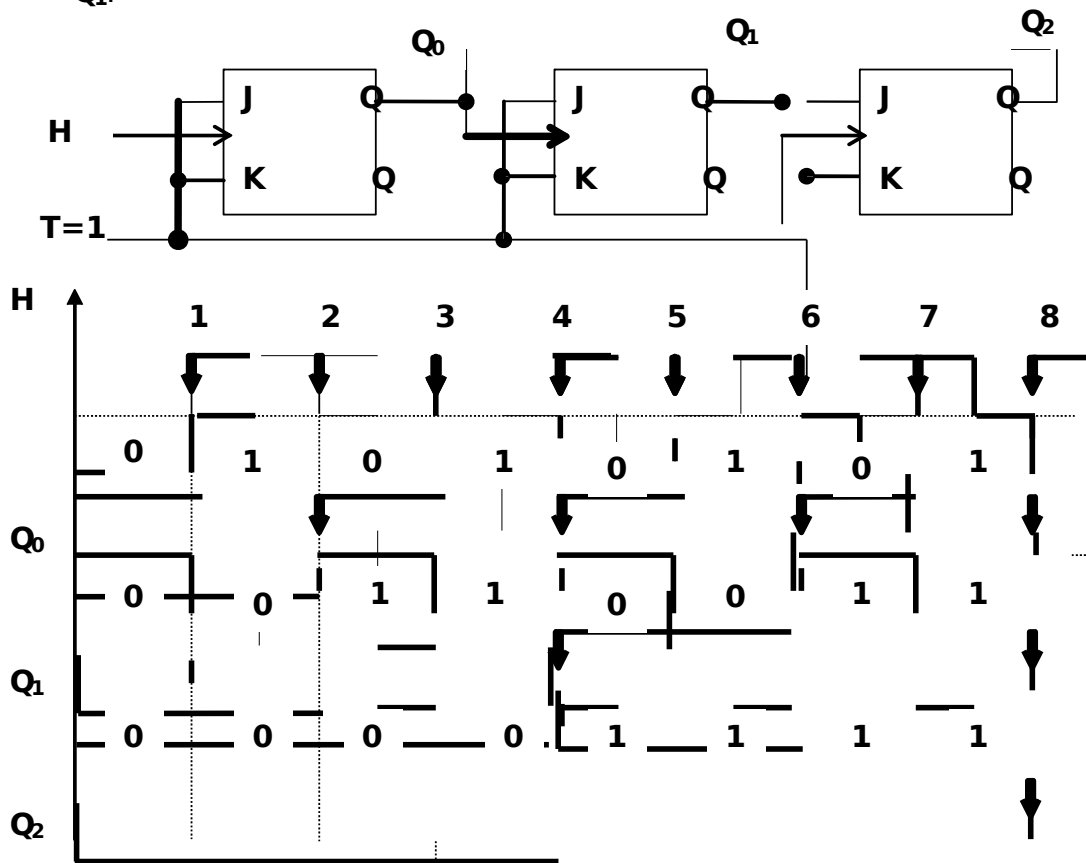
Ce sont des compteurs où l'ordre de changement d'état des bascules se fait en cascade. Le signal d'horloge est placé à l'entrée de la première bascule et les entrées d'horloge des autres bascules reçoivent aussi la sortie des bascules qui les précèdent.

Un compteur modulo N permet de compter les nombres binaires compris entre 0 et N-1.

Considérons par exemple un compteur modulo 8 suivant le code binaire pur constitué de trois bascules JK fonctionnant en mode T.

Supposons les trois bascules à zéro à l'instant $t=0$. L'évolution temporelle des trois sorties Q_0 , Q_1 et Q_2 par rapport aux impulsions d'horloge est représentée sur la figure ci-dessous.

La sortie Q_0 bascule sur chaque front descendant du signal d'horloge. La sortie Q_1 change d'état à chaque transition $1 \rightarrow 0$ de la sortie Q_0 . De même le basculement de la sortie Q_2 est déclenché par une transition $1 \rightarrow 0$ de la sortie Q_1 .



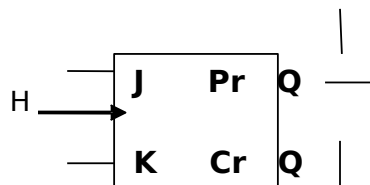
A partir de ce chronogramme nous pouvons écrire la liste des états successifs des trois états.

Impulsion	Q ₂	Q ₁	Q ₀
Etat 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

La séquence normale d'un compteur binaire peut être modifiée pour éliminer les états indésirables et pour réaliser un tel compteur, on dispose de deux méthodes : le rebouclage asynchrone et le conditionnement des entrées des bascules.

- **Cas du rebouclage asynchrone :**

La figure suivante donne la représentation symbolique d'une bascule JK avec les entrées Preset et Clear.



Les entrées asynchrones (car à utiliser en absence de signal d'horloge) **Pr** (Preset) et **Cr** (Clear) permettent d'assigner l'état initial de la bascule. Ex : Compteur modulo 6

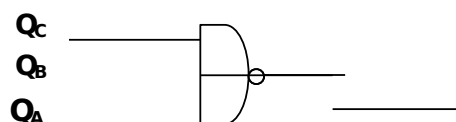
Q _C	Q _B	Q _A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
0	0	0

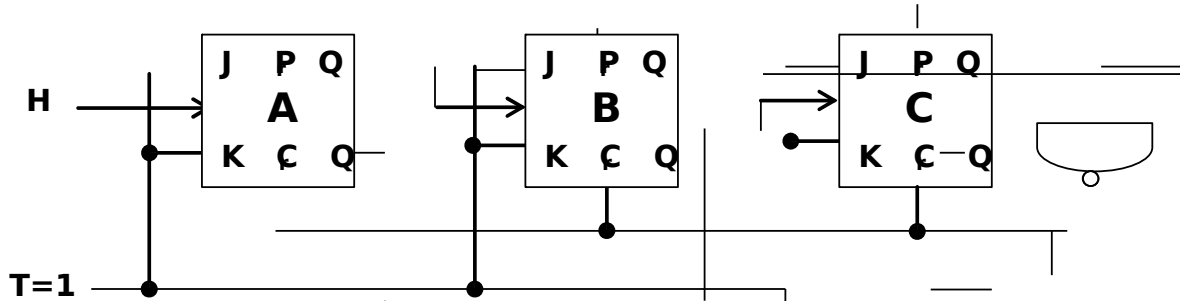
Au lieu de 110

Le compteur est mis à 000 au lieu d'afficher 110.

La remise à zéro est réalisée par un circuit logique combinatoire qui agit sur les entrées de forçage dès l'apparition du 6 binaire.

Le circuit de forçage est généralement une porte **NAND** dont les entrées sont la combinaison des sorties de l'état indésirable.





• Cas du conditionnement des entrées :

Q_C	Q_B	Q_A	$Q_C Q_A$	$\overline{Q_C} \overline{Q_A}$	$Q_C Q_A + \overline{Q_C} \overline{Q_A}$	$Q_A + \overline{Q_C} \overline{Q_A}$
0	0	0	0	1	0	0
0	0	1	0	1	0	1
0	1	0	0	1	1	1
0	1	1	0	1	1	1
1	0	0	0	1	0	0
1	0	1	1	0	1	1
0	0	0	0	1	0	0

Evolution de Q_A

N_{10}	Q_A
0	0
1	1
2	0
3	1
4	0
5	1
0	0

Q_A change au rythme de l'horloge donc il n'y a pas de conditionnement pour la bascule **A**

Evaluation de Q_B

N_{10}	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0
5	0	1
0	0	0

Q_B est contrôlé par Q_A . Q_B change d'état à chaque front descendant de Q_A pour la transition **5 → 0**. Durant cette transition Q_B mémorise son état antérieur : il faut conditionner les entrées de la bascule B pour la mémorisation. Q_C convient pour ce conditionnement car Q_C est égal à 0 pour la transition **5 → 0** et reste égal à 1 pour les autres transitions.

Evaluation de Q_c

N_{10}	Q_c	Q_b
0	0	0
1	0	0
2	0	1
3	0	1
4	1	0
5	1	0
0	0	0

Q_c est contrôlé par le front descendant de Q sauf à la transition $5 \rightarrow 0$.

Il faut conditionner ses entrées en agissant sur la commande d'horloge. Deux fronts descendants sont nécessaires $3 \rightarrow 4$ et $5 \rightarrow 0$.

$Q_c = Q_a + Q$ et $Q = Q + Q$ conviennent

4.2 Compteurs synchrones

Dans ce type de compteur toutes les bascules changent d'état en même temps parce qu'elles sont synchronisées par la même horloge. Il est donc possible de faire une synthèse de ces compteurs en établissant leur table d'évolution.

4.2.1 Synthèse des compteurs synchrones par la méthode de Marcus

(Voir TD)

4.2.2 Synthèse des compteurs synchrones par la méthode de la fonction de commutation (Voir TD)

Projet de Codages et Logiques

Fiche N°1

Synthèses des compteurs synchrones par la méthode de Marcus

On désire réaliser un compteur synchrone à 3 bits avec 3 bascules **JK** A, B, C. La méthode de Marcus permet de calculer chaque entrée des bascules afin de conditionner leurs évolutions lors de la prochaine impulsion d'horloge.

1. A partir de la table de fonctionnement de la bascule **JK**, remplir la table suivante appelée table d'évolution de la bascule **JK** :

Q_t	Q_{t+1}	J_t	K_t
0	0		
0	1		
1	0		
1	1		

2. Remplir la table d'évolution du compteur en considérant la table d'évolution de la bascule JK.

N	Q_C	Q_B	Q_A	J_C	K_C	J_B	K_B	J_A	K_A
0	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						
0	0	0	0						

3. Déterminer l'équation simplifiée de chacune des entrées des bascules
4. Donner la représentation schématique du compteur

Projet de Codages et Logiques

Fiche N°2

Synthèses des compteurs synchrones par la méthode de la fonction de commutation

La méthode de la fonction de commutation permet de réduire le nombre de tableau de Karnaugh utilisé dans la procédure de Marcus.

Soit **X** une fonction binaire (appelée fonction de commutation) définie par :

X=1 pour un changement d'état

X=0 pour une mémorisation

1. En considérant les états (futur et présent) d'une bascule considérée. Exprimer la fonction **X** en fonction des variables de sortie **Q** et **Q** de la bascule.
2. Donner l'expression de l'état futur d'une bascule **JK** en fonction des entrées de la bascule et de l'état présent.
3. Montrer que $X = J\bar{Q} + KQ$
4. Remplir la table suivante appelée table d'évolution du compteur :

N	Q_C	Q_B	Q_A	X_C	X_B	X_A
0	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			
0	0	0	0			

5. Déterminer l'équation simplifiée des entrées des différentes bascules
6. Donner la représentation schématique du compteur.