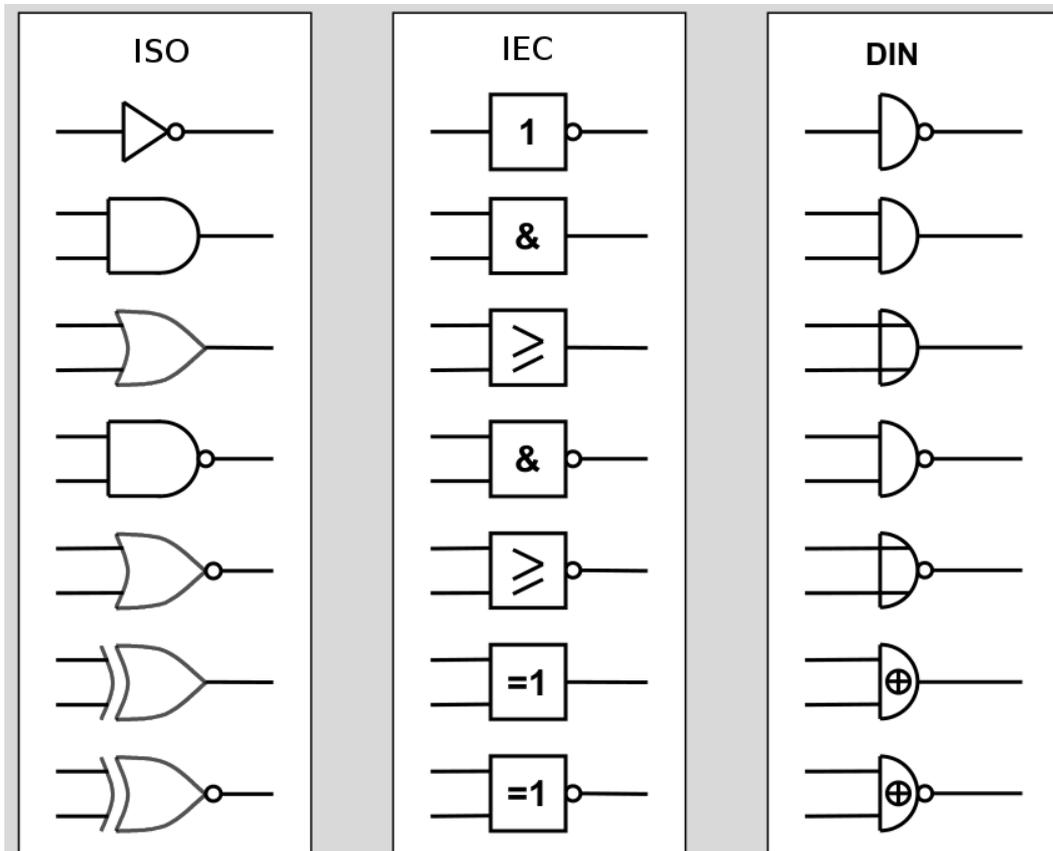
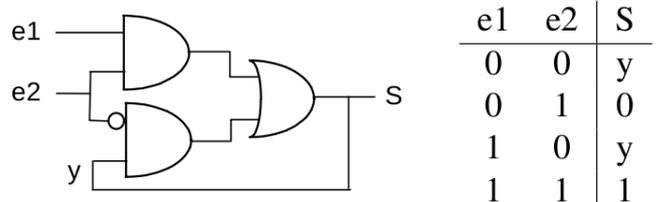


Annexe 2 : Symboles normalisés des Portes Logiques

1. Introduction aux systèmes séquentiels

Dans les **systèmes combinatoires**, les valeurs des sorties à un instant donné sont directement imposées par celles des entrées.

Mais que se passe-t-il si on reboucle l'une des sorties d'un tel système sur l'une de ses entrées ?



De manière générale, les systèmes numériques qui font apparaître des boucles de rétroaction permettent de mémoriser des informations relatives aux stimuli antérieurs appliqués sur le circuit. La sortie d'un tel circuit, en plus des variables d'entrées et de sorties, est aussi fonction de **variables internes** (y dans l'exemple précédent), appelées **variables d'état**.

Ces systèmes sont appelés des **systèmes séquentiels**. Ils sont caractérisés par le fait que pour chaque combinaison de variables d'entrée (chaque état d'entrée), les sorties peuvent prendre plusieurs valeurs possibles (plusieurs états de sortie). La valeur présente en sortie dépend également de l'**état précédent du système**. Il est alors nécessaire de pouvoir stocker cet état et donc d'introduire la fonction de **mémorisation** d'une grandeur binaire.

1.1. Notion d'état

Les valeurs des variables internes d'un système évoluent au fil du temps, en fonction des changements sur les entrées et des valeurs des sorties. A chaque combinaison de valeurs des variables du système correspond une configuration de celui-ci qu'on appelle l'**état du système**.

On identifie un état du système par un ensemble de variables de ce système appelées **variables d'état**. Elles correspondent au plus petit ensemble de variables indépendantes qui permettent de coder l'état du système. Lorsqu'il existe p variables d'état, il existe 2^p états possibles.

Remarque : Le nombre de variables d'état n'est pas unique : on peut toujours utiliser plus de variables d'état que le strict nécessaire. On peut par exemple faire correspondre à chaque état une variable d'état qui passe '1' lorsque l'état correspondant est atteint : codage *un parmi n*.

1.2. Systèmes synchrones ou asynchrones

Un système séquentiel peut être :

- **synchrone** : son évolution (et en particulier celle de ses sorties) est contrôlable de l'extérieur par un signal appelé **horloge** ;
- **asynchrone** : son évolution n'est pas contrôlable de l'extérieur.

Les systèmes séquentiels **asynchrones** sont plus rapides, mais plus difficiles à mettre au point dans les applications réelles, puisque dès lors qu'un changement intervient sur son entrée, il peut provoquer un changement sur ses sorties instantanément.

Les systèmes séquentiels **synchrones** sont moins efficaces, mais plus fiables et plus prédictibles, l'évolution de leur sortie étant directement contrôlée par une horloge.

Nous nous intéresserons dans la suite de ce cours uniquement à la synthèse des systèmes synchrones (sauf compteurs asynchrones).

2. Représentation des systèmes séquentiels

La succession des divers états pris par le système au cours de son fonctionnement constituent ce qu'on appelle une **séquence** (d'où le terme de systèmes séquentiels).

Chaque état d'une machine à état peut être représenté par l'ensemble des valeurs de ses variables d'état : à un ensemble de valeurs donné correspond **un état unique**.

Il est courant d'**identifier un état donné** par un symbole :

- une lettre ou un groupe de lettre ayant une signification dans le système considéré ;
- un nombre servant de numéro à l'état considéré.

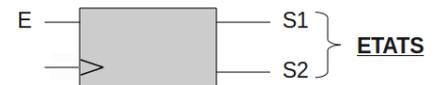
L'évolution d'une machine à état est guidé par les valeurs des variables d'entrée. La **transition** d'un état à l'autre dépend de ces valeurs.

Il existe plusieurs façons de représenter ces séquences et ces changements d'état. On peut utiliser :

- une **équation logique** ;
- un **chronogramme** ;
- une **table de transition** ;
- un **diagramme d'état** ;
- une **structure** logique.

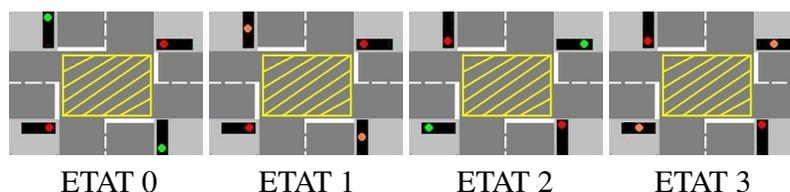
Il existe également d'autres représentations possibles de ces systèmes tels que les **GRAFCET** ou les **réseaux de Pétri**. Ces représentations sont souvent associées à un type particulier de machine qu'on appelle des **automates programmables industriels (API)**.

Pour pouvoir illustrer les différentes représentation possible d'un système séquentiel, nous allons prendre l'exemple de la gestion d'un *carrefour à 2 voies de circulation*.



Le système est commandé par une entrée E . La machine change d'état à chaque changement de valeur sur cette entrée E .

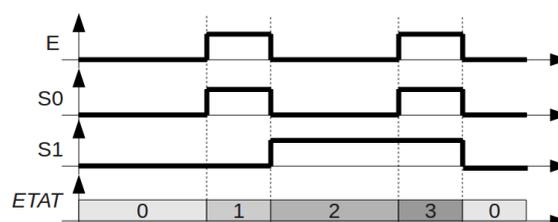
4 états sont nécessaires pour représenter ce système, il faudra au minimum deux variables d'état pour les coder. Ces variables d'état seront appelés $S1$ et $S2$



2.1. Chronogramme

Les **chronogrammes** sont très répandus dans le monde de l'électronique pour pouvoir représenter l'**évolution temporelle** d'un signal. C'est un modèle graphique qui représente l'évolution au cours du temps de toutes les entrées et sorties du système.

On ne représente pas toujours le signal d'horloge. Mais sur un système synchrone, un changement en sortie ne peut avoir lieu que lors d'un **front actif** sur l'entrée d'horloge.



Cette représentation permet de définir un certain nombre d'états du système. Dès que l'on augmente le nombre d'entrées/sorties, il existe un risque d'oublier certains de ces états. Ce mode de représentation n'est pas synthétique et doit être réservé à la représentation d'un **exemple concret de fonctionnement** du système et non à l'intégralité du fonctionnement.

2.2. Table de transition

Une **table de transition** permet de représenter dans un tableau la **transition** qu'il existe d'un état à l'autre en fonction des entrées du système. C'est une représentation équivalente à la table de vérité pour les systèmes combinatoires avec d'un côté les entrées et les états actuels du système et de l'autre les états suivants du système.

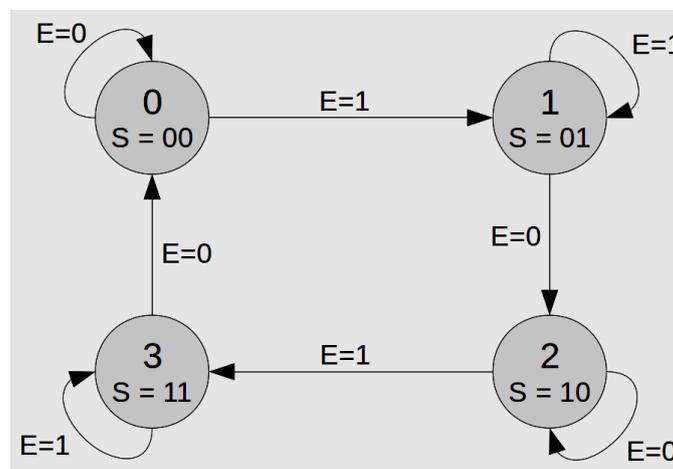
Etats actuels	E	Etats suivants
$S_1^- S_0^-$	E	$S_1^+ S_0^+$
0 0	0	0 0
0 0	1	0 1
0 1	0	1 0
0 1	1	0 1
1 0	0	1 0
1 0	1	1 1
1 1	0	0 0
1 1	1	1 1

C'est à partir de cette représentation qu'il est le plus simple de concevoir la structure d'un système numérique séquentiel. Cette étude, appelée **synthèse**, sera étudiée dans une des sections suivantes de ce cours.

2.3. Diagramme d'état

Un **diagramme d'état** permet de visualiser l'ensemble des états du système et des transitions existantes de manière graphique. Chaque état est représenté dans un **cercle** portant le nom de l'état. Dans ce cercle est également indiquée la **valeur des sorties**.

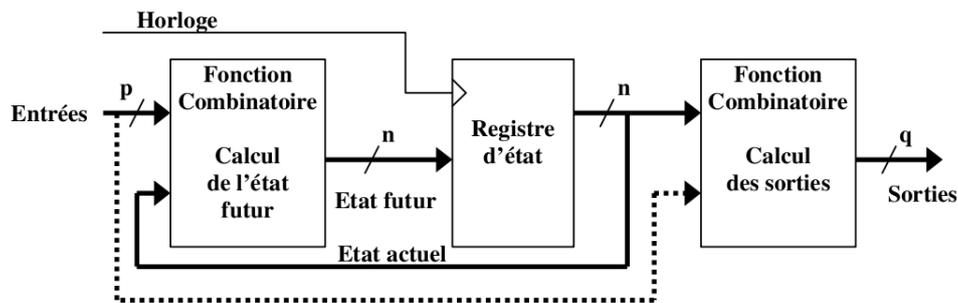
Entre différents états, il peut exister des **transitions**. Elles sont matérialisées par des **flèches**, qui indiquent le sens de la transition ainsi que la condition de franchissement de celle-ci. Ces franchissements ne peuvent avoir lieu, dans un système synchrone, que sur **front actif de l'horloge**.



2.4. Structure d'une machine à état

Enfin, il est possible de représenter un système séquentiel par sa **structure complète**, tel un logigramme. Un tel système peut être schématisé par :

- un **élément de mémorisation**, retenant la valeur des variables d'état pour l'état actuel ;
- un **élément de calcul**, pour déterminer la valeur des variables d'état pour l'état futur ;
- optionnellement, un élément de calcul supplémentaire qui élabore les valeurs de sortie à partir des variables d'état.



3. Bascules : composants élémentaires de la logique séquentielle

Dans la structure précédente, nous avons déjà vu comment réaliser des fonctions de calcul à l'aide de fonctions combinatoires. Nous allons à présent nous intéresser aux composants de base permettant la réalisation de la **fonction mémoire**. Ces composants s'appellent des **bascules**.

3.1. Bascule RS

La bascule la plus simple est la **bascule RS**. C'est une mémoire possédant deux entrées R et S et une sortie Q . Le S signifie **Set** (ou mise à un) et R signifie **Reset** (ou mise à zéro).

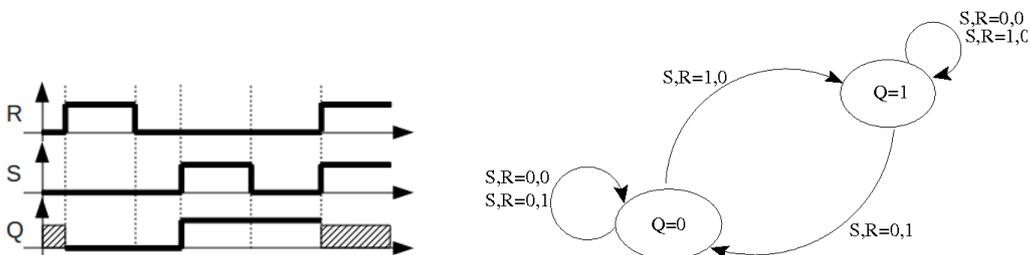
Lorsque l'entrée S vaut '1', la sortie Q passe à '1' (mise à un).

Lorsque l'entrée R vaut '1', la sortie Q passe à '0' (mise à zéro).

Lorsqu'aucune des entrées n'est à '1', alors la bascule conserve la valeur précédente. C'est l'**état mémoire**.

Son équation est la suivante :

$$Q^+ = S + R \cdot Q^-$$



Malgré le fait qu'elle soit **asynchrone** (pas de signal d'horloge) et qu'il existe un état interdit ($R=1$ et $S=1$), cette bascule est encore utilisée dans un certain nombre de procédés tels que les **systèmes anti-rebond**.

3.2. Bascule JK

La **bascule JK** est considérée comme la version synchrone de la bascules RS. C'est une mémoire possédant deux entrées J et K et une sortie Q . Etant synchrone, elle possède également une entrée spécifique d'horloge CLK . Le J signifie **Jump** (ou mise à un) et K signifie **Knock** (ou mise à zéro).

Son équation est la suivante :

$$Q^+ = Q^- \cdot \bar{K} + \bar{Q}^- \cdot J$$

Cette bascule permet 4 fonctions différentes, résumées dans la table de transition suivante :

J	K	Q^+	Fonction
0	0	Q^-	mémoire
0	1	0	reset (knock)
1	0	1	set (jump)
1	1	\bar{Q}^-	toggle / basculement

Remarque : le signal d'horloge n'intervient pas dans l'écriture de l'équation, ni même dans la table de transition, mais **la sortie ne peut évoluer que sur un front actif** (montant ou descendant, selon la bascule) de ce signal d'horloge.

3.3. Bascule T

Cette bascule élémentaire est essentiellement utilisée dans la réalisation de compteurs (voir section 4.1). Le T signifie *toggle* (ou basculement). Elle possède une entrée d'horloge CLK , une entrée T et une sortie Q . Elle est **synchrone**.

Elle change d'état à chaque front actif d'une horloge (entrée CLK) lorsque $T = 1$ et conserve son état lorsque $T = 0$.

Son équation est la suivante :

$$Q^+ = T \cdot \bar{Q}^- + \bar{T} \cdot Q^-$$

Son fonctionnement peut aussi être représenté par la table de transition suivante :

T	Q^+	Fonction
0	Q^-	mémoire
1	\bar{Q}^-	toggle / basculement

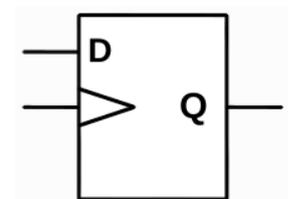
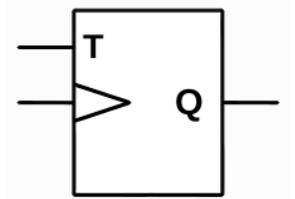
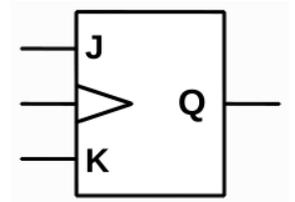
3.4. Bascule D

La **bascule D** est probablement la plus utilisée de toutes les bascules. On la retrouve dans un grand nombre de systèmes séquentiels, entre autre dans les circuits logiques complexes de type CPLD⁴ ou FPGA⁵. Elle possède une entrée d'horloge CLK , une entrée de donnée D et une sortie Q .

Cette bascule recopie l'entrée de donnée D sur sa sortie Q à chaque front actif de l'horloge.

Son équation est la suivante :

$$Q^+ = D$$



4. Complex Programmable Logic Device

5. Field Programmable Gate Array

4. Fonctions séquentielles standard

Il existe des fonctions séquentielles plus complexes que les bascules, qui sont régulièrement utilisés dans les systèmes numériques.

Nous allons étudier plus en détail par la suite :

- les compteurs / décompteurs
- les registres

4.1. Compteurs / Décompteurs / Diviseurs de fréquence

Un **compteur** est une association de n bascules permettant de décrire, au rythme d'une horloge, une séquence déterminée qui peut avoir au maximum 2^n combinaisons différentes. Les combinaisons apparaissent toujours dans le même ordre.

Lorsque la succession des états correspondra à un ordre croissant, on utilisera le terme de **compteur**, et dans le cas contraire, le terme de **décompteur**.

On peut également parler de **modulo** lorsqu'il est question de compteur. Le **modulo** est le **nombre d'états différents** que peut prendre un compteur.

Un compteur modulo N , par exemple, démarre de la valeur 0 et compte dans l'ordre binaire naturel jusqu'à $N - 1$.

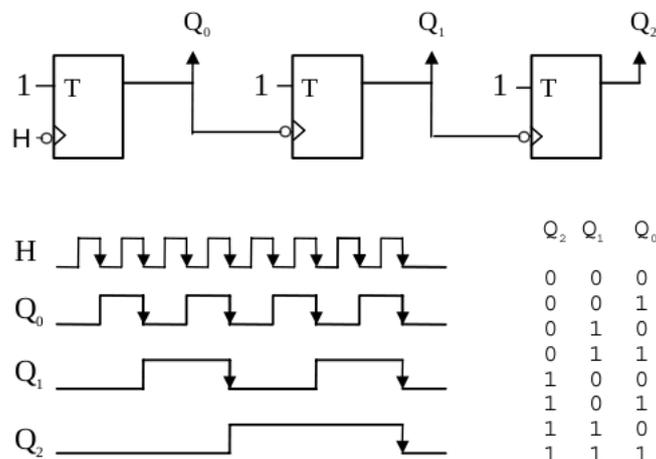
On appliquera souvent un qualificatif pour caractériser un compteur, selon le codage des états et le nombre de ceux-ci :

- compteur **binaire** si les états correspondent à un codage binaire naturel des variables d'état ;
- compteur **décimal** ou BCD dans le cas où il s'agit d'un codage BCD des variables d'état ;
- d'autres codages (Johnson, sexagésimal, bi-quinaire...).

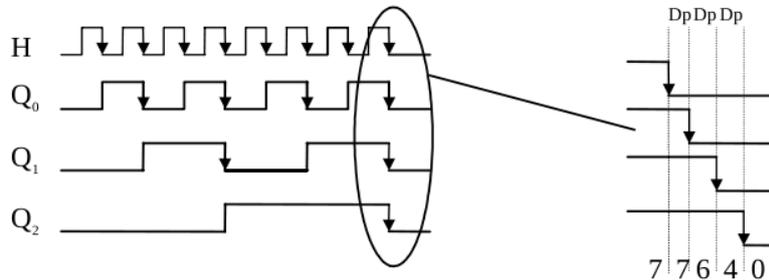
4.1.1 Compteurs asynchrones

La façon la plus simple de réaliser un compteur est de remarquer d'une bascule T, dont l'entrée est à '1' en permanence, se comporte comme un **diviseur de fréquence par 2** de l'horloge d'entrée.

Ainsi la mise en cascade de plusieurs bascules de ce type permet de propager l'état futur d'une bascule à l'autre.



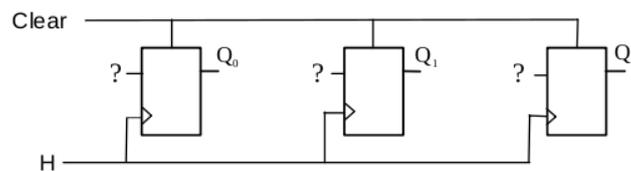
Ce type de structure n'est cependant pas recommandé car il peut être source de nombreux ennuis. En effet, le fait de mettre en cascade ces bascules entraîne également un **cumul des temps de réaction** (ou propagation) de chacune d'entre elles. Ainsi on peut voir apparaître à certains moments de la séquence des **états non désirés**.



4.1.2 Compteurs synchrones

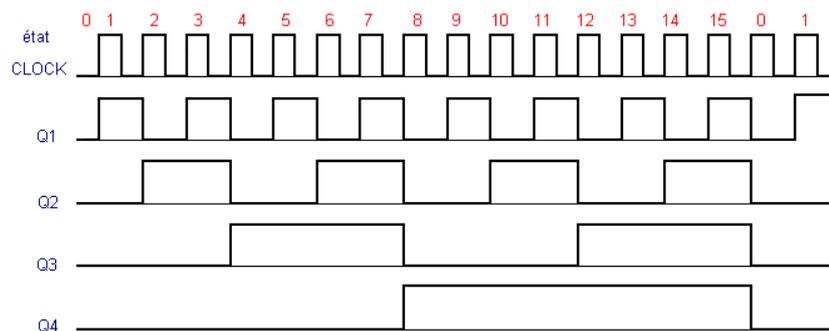
La solution pour pouvoir s'affranchir de ces états perturbateurs est de rendre le système **totalemt synchrone**. Pour cela, il est indispensable que toutes les bascules reçoivent le **même signal d'horloge**.

La synthèse d'un compteur synchrone consiste alors à trouver la commande nécessaire des entrées des bascules (T, D ou JK) pour obtenir la séquence déterminée sur les sorties.



Compteur synchrone modulo 2ⁿ

Par exemple, on souhaite réaliser un compteur modulo 16, tel que décrit par le chronogramme suivant :



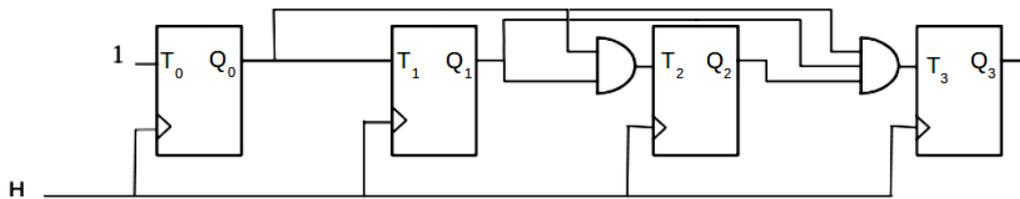
On peut remarquer que le bit de poids faible change à tous les coups d'horloge et qu'un bit quelconque change lorsque tous les bits de poids plus faible sont égaux à 1.

La réalisation à partir de bascules T ou JK est alors simple. Sachant que sur une bascule T (ou JK), il y a une inversion de la sortie pour $T = 1$ ($JK = 11$), on peut en déduire les entrées de chacune des bascules.

$$T_0 = 1 \quad T_1 = Q_0 \quad T_2 = Q_0 \cdot Q_1$$

$$T_n = Q_0 \cdot Q_1 \cdot \dots \cdot Q_{n-1}$$

On obtient alors la structure suivante :

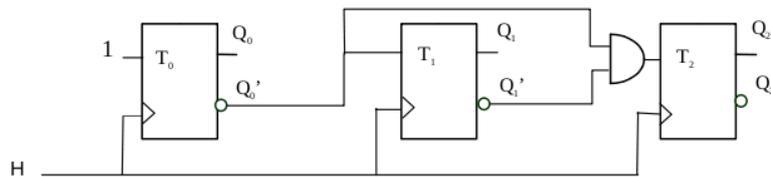


Le raisonnement fait précédemment avec des bascules T (ou JK) peut être mené à l'identique avec des bascules D.

De même, il est possible de synthétiser des compteurs ayant des **modulo différents d'une puissance de deux** (par exemple les compteurs décimaux ou modulo 10). Ce sont les équations des entrées des bascules qui vont changer.

4.1.3 Décompteurs synchrones

On peut montrer qu'en utilisant les sorties complémentées des bascules, il est possible d'obtenir un fonctionnement en décompteur. La structure d'un **décompteur modulo 8** est alors la suivante :



4.1.4 Diviseurs de fréquence

Les compteurs sont aussi utilisés pour **diviser la fréquence** d'une horloge. Cela permet de **ralentir la cadence** d'un système pour l'adapter à une fréquence voulue.

La sortie p d'un compteur a pour période :

$$T_p = 2^p \cdot T_{CLK}$$

où T_{CLK} est la période de l'horloge principale du système.

Ainsi, dans le domaine fréquentiel on obtient :

$$F_p = \frac{F_{CLK}}{2^p}$$

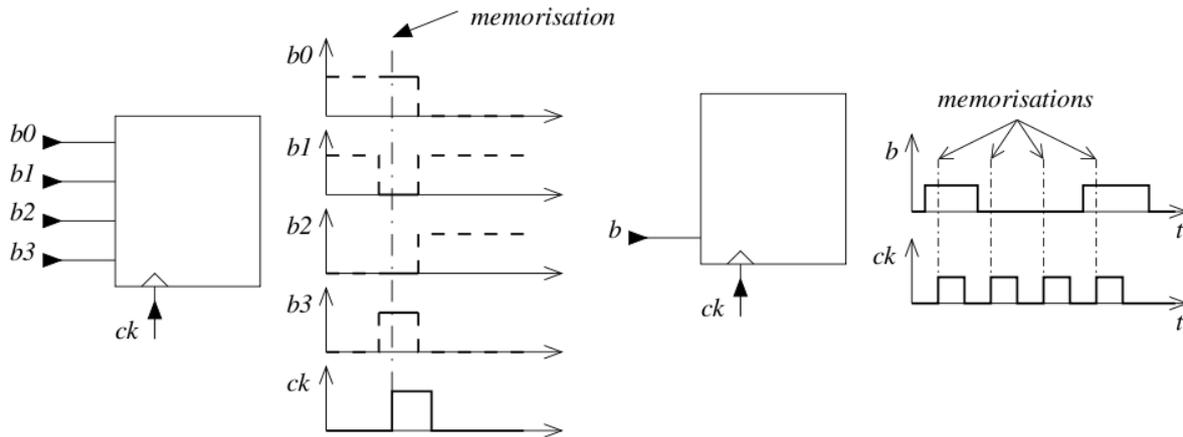
4.2. Registres

Un registre est une **association de n bascules** utilisées conjointement pour mémoriser les n bits d'un mot binaire.

Les n bascules ont une horloge commune. Les **basculés D** sont les plus utilisées dans les registres.

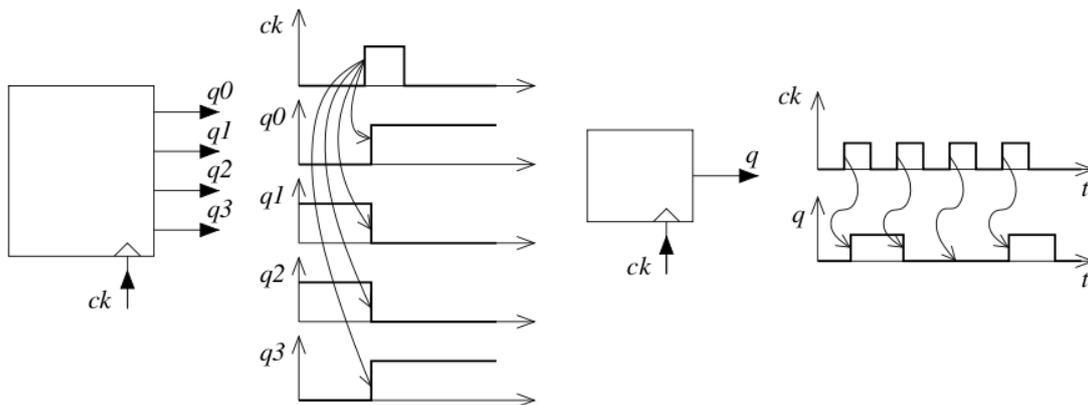
Il existe deux **modes de chargement en entrée** :

- en **parallèle** : un fil séparé est prévu pour chaque entrée D des bascules ;
- en **série** : un seul fil est prévu pour l'entrée D de la première bascule.



Il existe deux **modes de lecture en sortie** :

- en **parallèle** : un fil séparé est prévu pour chaque sortie Q des bascules ;
- en **série** : un seul fil est prévu pour la sortie Q de la dernière bascule.

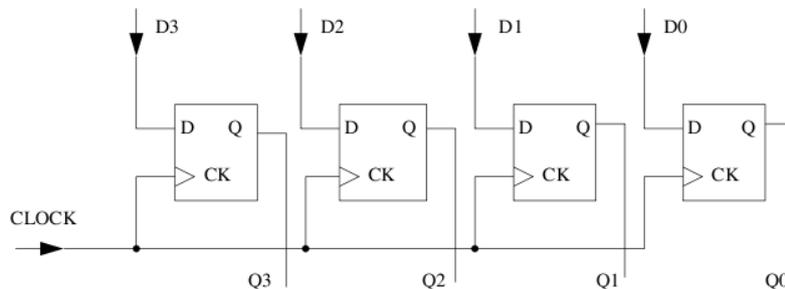


Il existe alors plusieurs combinaisons possibles d'entrée et de sortie. Selon ces combinaisons, les registres portent des noms différents :

- **registre à décalage** : série / série
- conversion **série/parallèle** de données
- conversion **parallèle/série** de données
- **registre d'état** / mémorisation : parallèle/parallèle

4.2.1 Registres de mémorisation

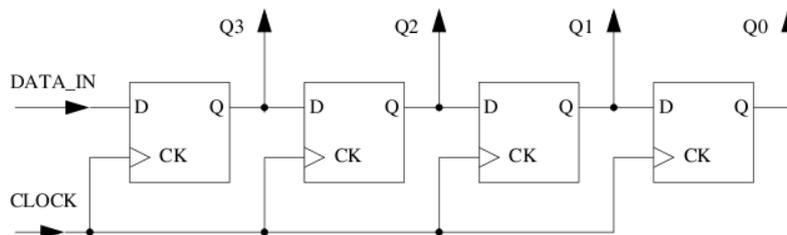
Le **registre de mémorisation** est le registre élémentaire. Il est constitué d'une juxtaposition de bascules permettant de mémoriser un mot binaire. Ce registre est également appelé registre à entrées parallèles.



4.2.2 Registres à décalage

Le registre à décalage est une association de bascules permettant de décaler un mot binaire, à droite ou à gauche.

Dans tous les cas, l'information est disponible intégralement en n coups d'horloge après le chargement pour un mot de n bits.



4.2.3 Compteur Johnson

C'est un registre à décalage dont la dernière sortie est rebouclée sur l'entrée via un inverseur.

Une séquence principale est initialisée dans le dispositif et tourne en permanence dans cette structure.

