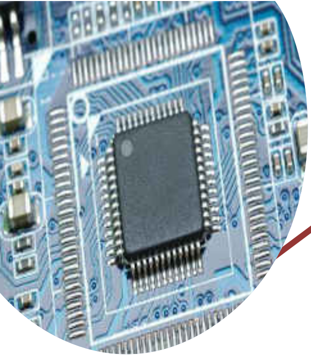
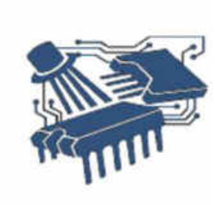


Ministre de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Seddik BENYAHIA - Jijel
Faculté des Sciences & de la Technologie
Département d'Electronique

POLYCOPIÉ PÉDAGOGIQUE

SYSTEMES LOGIQUES CABLES
PARTIE 1 : SYSTEMES SEQUENTIELS
- Cours & Exercices -

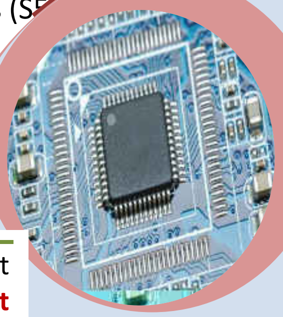


Elaboré par :

Dr-HDR. Ammar SOUKKOU

{ Soukkou.amr@gmail.com ;
soukkou.ammar@univ-jijel.dz }

- Manuscrit élaboré selon le programme officiellement agréé et confirmé par le **CPNDST.**
- **Public ciblé :**
  - ✓ Deuxième année **Socle Commun** (S4).
  - ✓ Troisième année **Licence** Electronique (S5).
  - ✓ Troisième année **Licence** Systèmes de Communications (S
  - ✓ **Master I:** Electronique des Systèmes Embarqués.



Contenu

a. **Objectifs**
b. **Connaissances préalables recommandées**
c. **Organisation**
d. **Références bibliographiques**

a. Objectifs

Connaître les circuits séquentiels usuels. Savoir représenter quelques applications des circuits séquentiels en utilisant les outils standards à savoir les tables de vérité, tables de Karnaugh, table des états,... etc.

b. Connaissances préalables recommandées

/

c. Sommaire

CHAPITRE 1 : INITIATION AUX SYSTEMES LOGIQUES

CHAPITRE 2 : SYSTEMES DE NUMERATION - CODAGE

CHAPITRE 3 : LES BASCULES

CHAPITRE 4 : LES REGISTRES

CHAPITRE 5 : LES COMPTEURS

CHAPITRE 1

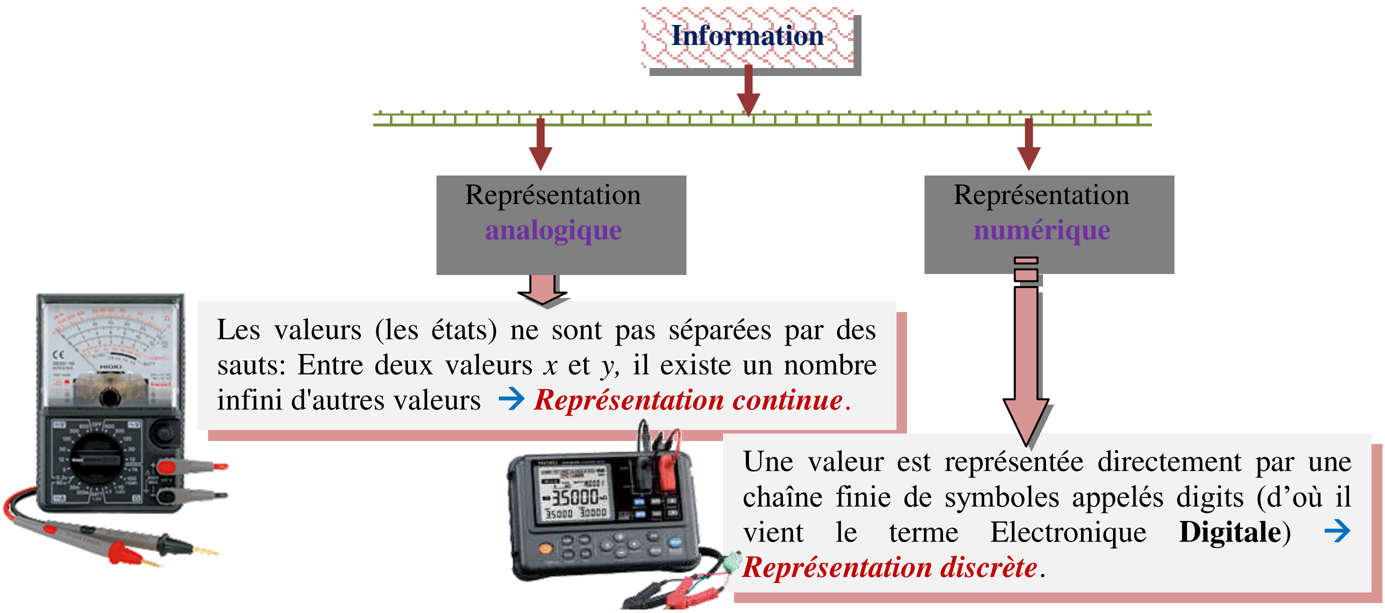
INITIATION AUX SYSTEMES LOGIQUES

Objectifs

- *L'objectif* est de
  - *Donner un apperçu général sur quelques notions élémentaires nécessaires à la compréhension des systèmes logiques.*
  - *D'appréhender les aspects conceptuels liés à la conception des systèmes logiques*

1.1. Représentation de l'information

La notion d'information correspond à la connaissance d'un état donné d'un phénomène physique qui peut être représenté par un signal electrique qui décrit sont evolution vis à vis du temps) parmi plusieurs états possibles à un instant donné. Une information (ou une quantité) peut être représentée de deux manières illustrée par la figure 1.1.



Les valeurs (les états) ne sont pas séparées par des sauts: Entre deux valeurs x et y, il existe un nombre infini d'autres valeurs → ***Représentation continue***.

Une valeur est représentée directement par une chaîne finie de symboles appelés digits (d'où il vient le terme Electronique **Digitale**) → ***Représentation discrète***.

Figure 1.1 : *Modes de représentation de l'information.*

☒ Déviation de l'aiguille est proportionnelle à la vitesse du moteur.

☒ Un exemple typique d'une information analogique/numérique est le mode d'ffichage des montres analogiques et numériques comme montre la figure 1.2.

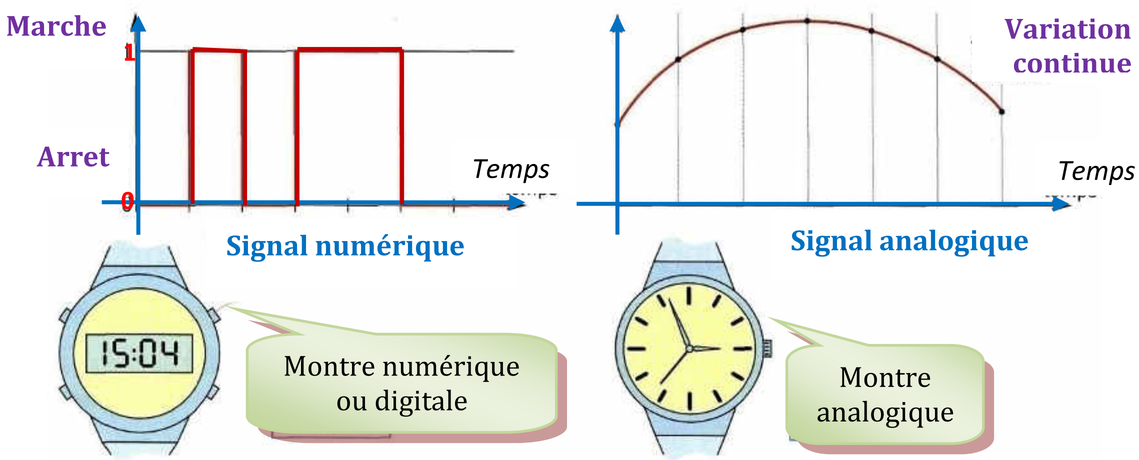
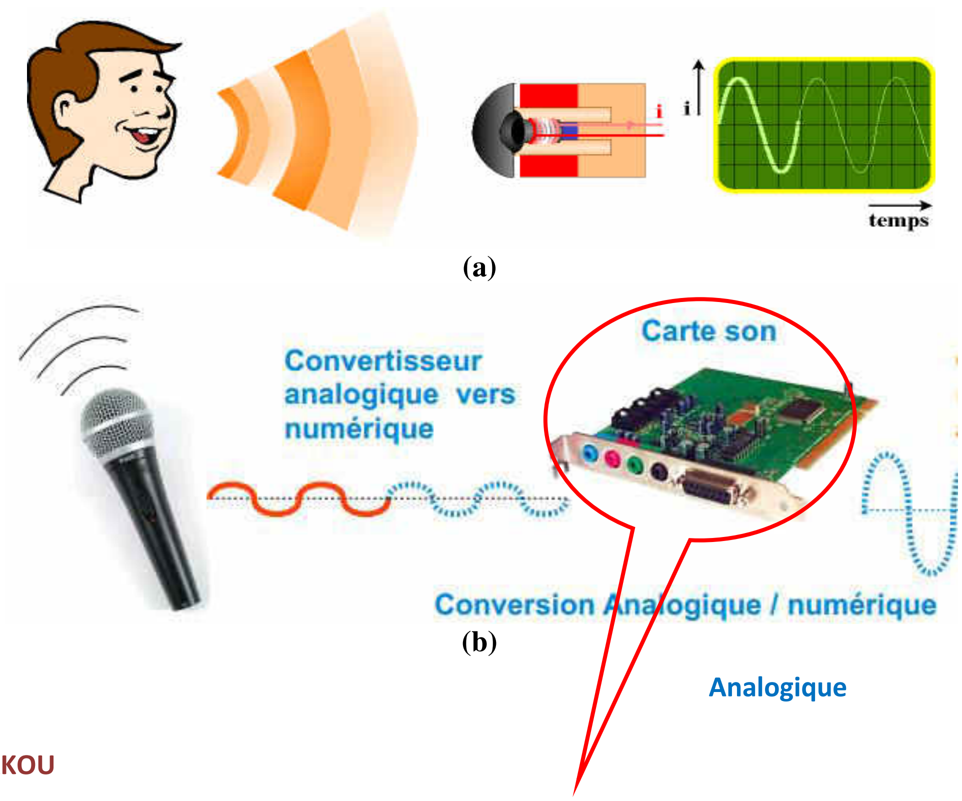


Figure 1.2 : *Modes d'affichage analogique / numérique.*

☒ La tension de sortie du microphone est proportionnelle à l'amplitude de l'onde sonore de la sortie du micrphone. La figure 1.3 est un exemple type de traitement analogique – numérique d'un signal sonore



(a)

(b)

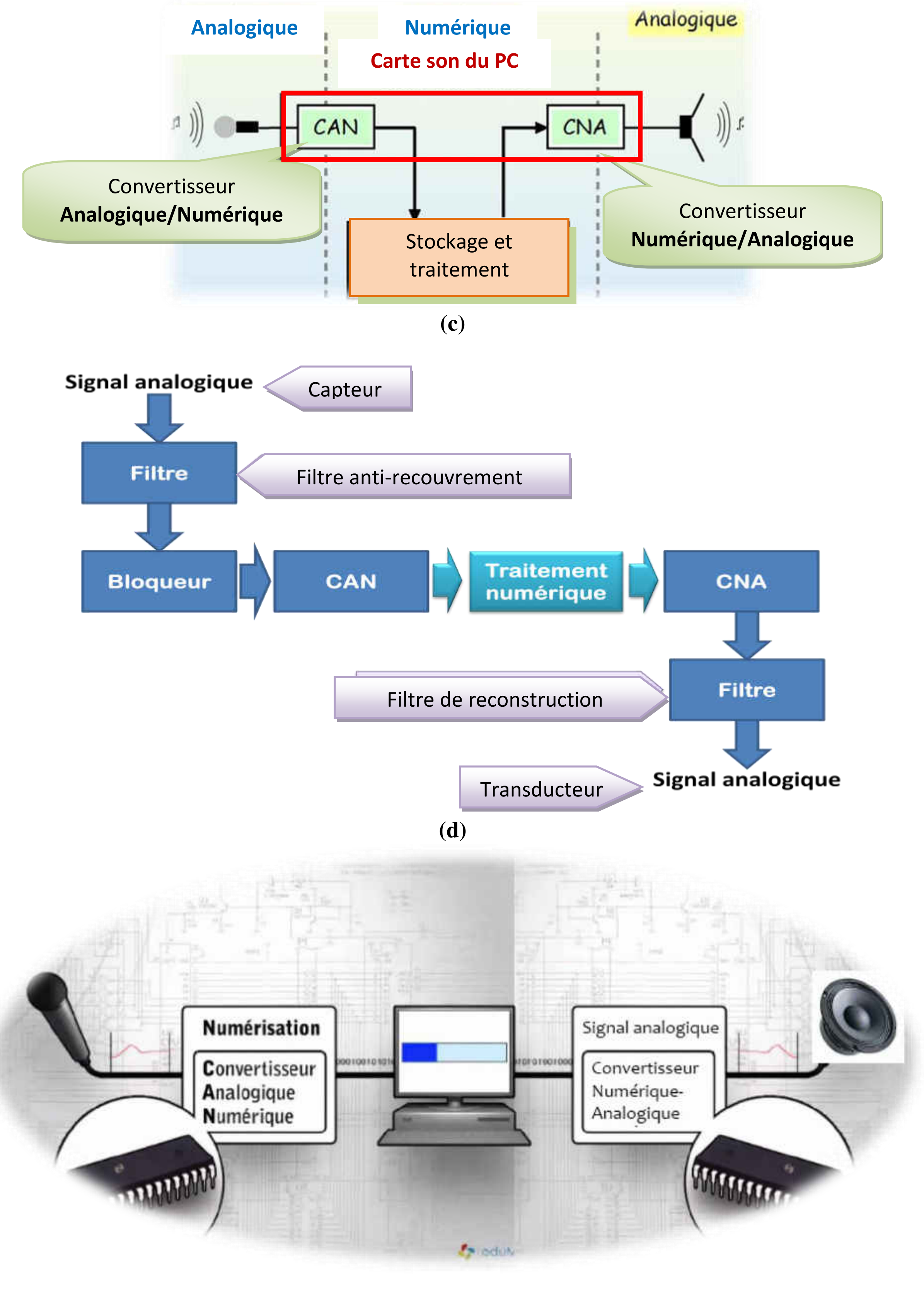


Figure 1.3 : Chaine de conversion et de traitement d'un signal sonore (analogique – numérique).

⌖ Une information (signal) analogique possède des valeurs continues. Alors qu'une information numérique renferme une série de valeurs discrètes comme indique la figure 1.4.

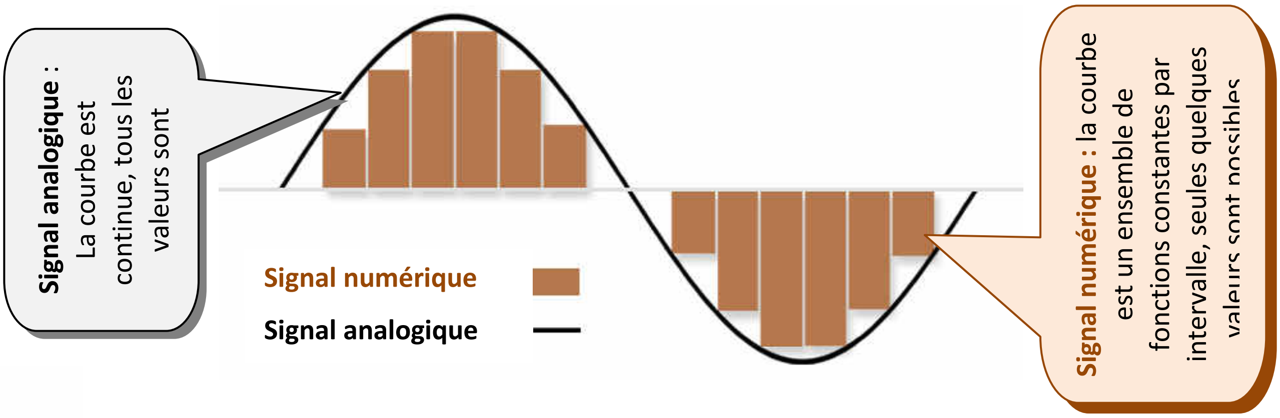


Figure 1.4 : *Exemple de signaux analogiques et numériques.*

⌖ Toute information dans un système informatique est représentée sous la forme d'un paquet de ***bits*** (Abréviation de ***Binary digit***). Un exemple typique de materialisation d'un bit est donné par le schéma de la figure 1.5.

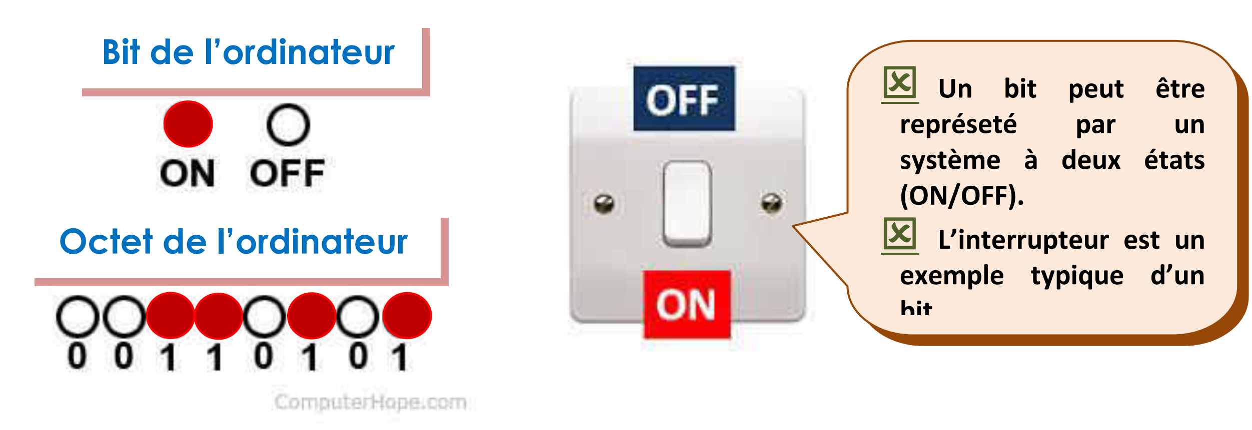
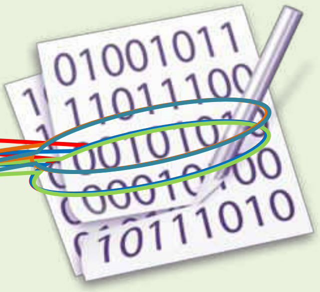


Figure 1.5 : *Exemple de materialisation d'un bit.*

⌖ La différence entre un type d'information et un autre est donnée seulement par le contexte: ***La même séquence de bits peut représenter***

- ***Un nombre entier,***
- ***Un nombre réel,***
- ***Un caractère,***
- ***Une instruction,***
- ***Un son,***
- ***... etc.***



Obtention d’un signal numérique à partir d’un enregistrement du signal sonore (signal analogique) ci-dessous.

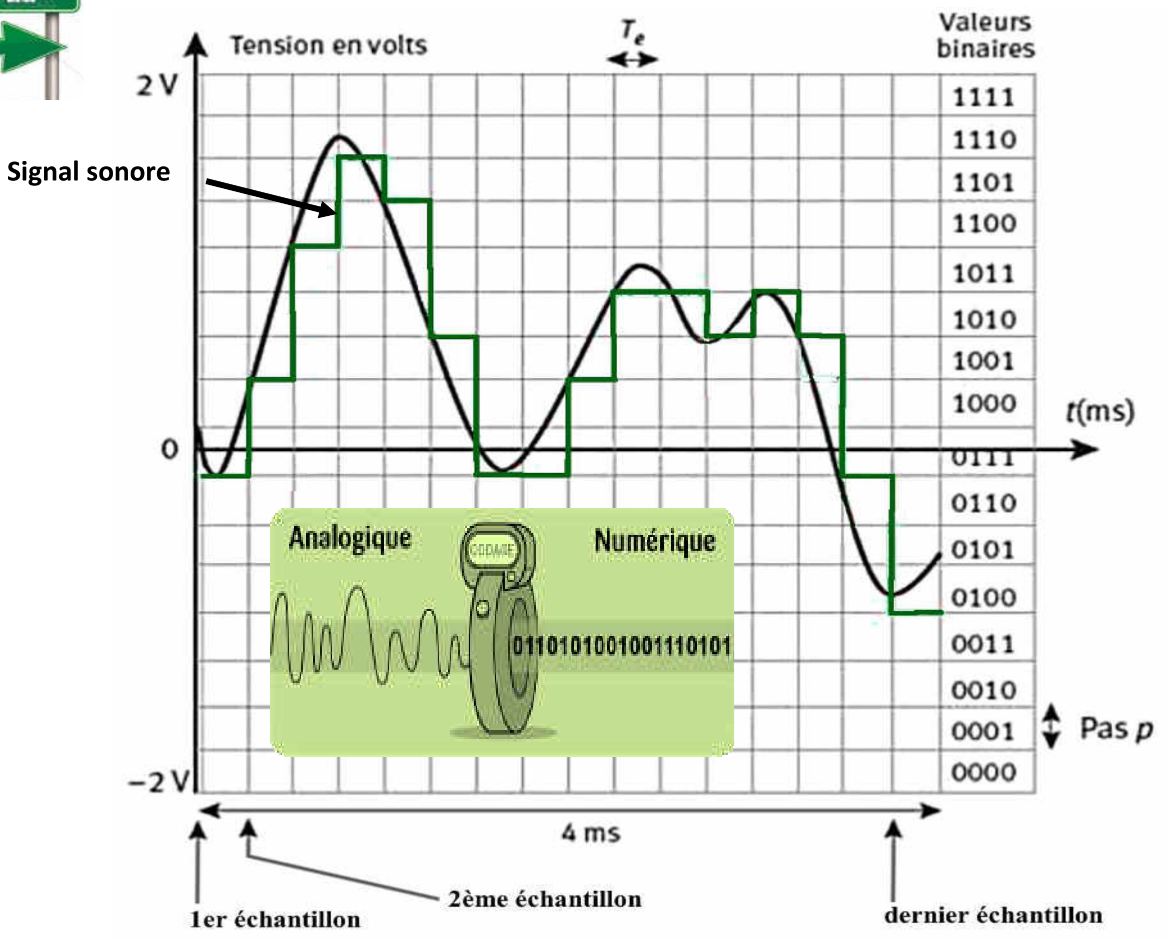


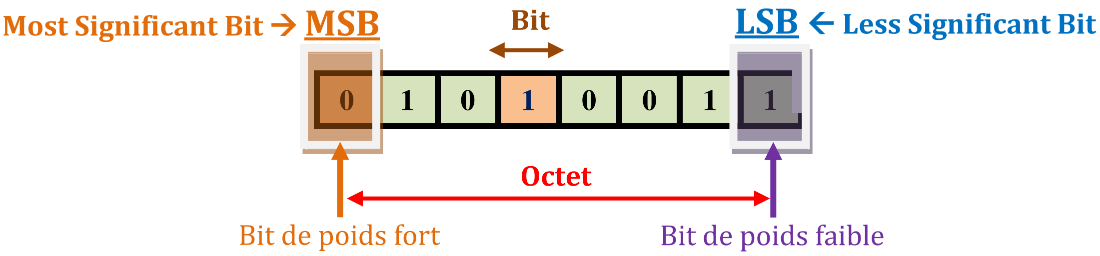
Figure 1.6 : *Exemple de numérisation d’un signal analogique.*

En savoir +

☒ **L’unité de mesure** des informations s’effectue par l’octet

01 Octet = 01 Byte = 08 bits

01 Quartet = 04 bits



☒ Avant 1998, les unités ci-dessous sont utilisées pour mesurer les différentes valeurs de l’octet.

Tableau 1.1 : *Unités de mesure des informations binaires.*

Unité	Abréviation	Valeur
1 Octet	1o	8 bits
1 Kilooctet	1 Ko	2^{10} octets = 1024 octets
1 Mégaoctet	1 Mo	2^{20} octets = 1024 Ko = 1 048 576 octets
1 Gigaoctet	1 Go	2^{30} octets = 1024 Mo = 1 073 741 824 octets
1 Téraoctet	1 To	2^{40} octets = 1024 Go = 1 099 511 627 776 octets

☒ Cette notation est encore très répandue, notamment dans certains logiciels et mêmes certains systèmes d'exploitation. Cependant, **en 1998, l'IEC (*International Electrotechnical Commission*) a statué sur cette notation et a décidé d'un standard comme indique le tableau ci-dessous.**

Tableau 1.2 : *Unités de mesure des informations binaires selon la norme IEC.*

Unité	Abréviation	Valeur
1 Octet	1o	8 bits
1 Kilooctet	1 Ko	1000 octets
1 Mégaoctet	1 Mo	1000 Ko = 1 000 000 octets
1 Gigaoctet	1 Go	1000 Mo = 1 000 000 000 octets
1 Téraoctet	1 To	1000 Go = 1 000 000 000 000 octets

- Un ordinateur manipule principalement deux types d'information :
  - *Les données.*
  - *Les instructions.*
- Les données représentent la matière première, les nombres avec lesquels l'ordinateur travaille, sur lesquels l'ordinateur effectue les calculs arithmétiques et logiques.
- L'ordinateur utilise ensuite ces données pour représenter l'information (mots, images, sons, etc), compréhensible par l'homme.
- Les modes de représenation des informations manipulées par les ordinateurs sont illustrées par le diagramme de la figure ci-dessous.

Toutes ces informations sont représentées par une succession de bits '0' et '1' selon un contexte bien déterminé.

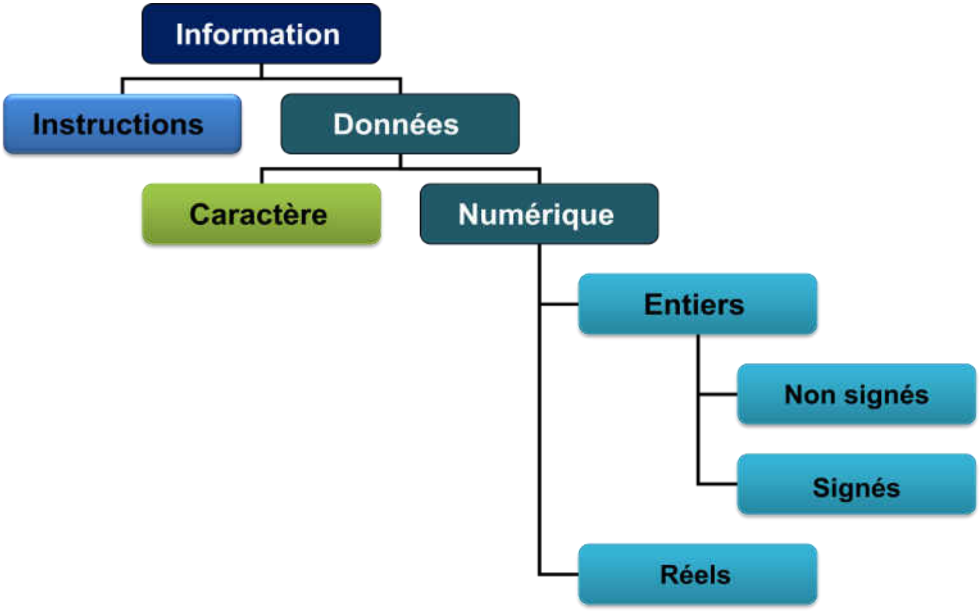


Figure 1.7 : Modes de représentation des informations dans un ordinateur.

En savoir +

- ✓ Dans les années 1930, Claude Shannon démontra que l'on peut effectuer des opérations logiques en utilisant uniquement des contacteurs (interrupteurs), appareils ne pouvant alors prendre que deux états :
    - ☒ Fermé (vrai) pour la valeur '**1**' et
    - ☒ Ouvert (faux) pour la valeur '**0**'.
- ✓ Ceci explique donc pourquoi, les ordinateurs manipulent des nombres binaires (base 2).
- ✓ Les processeurs ou unités de calcul des ordinateurs actuels sont, eux, composés de compsants semi-conducteurs : Transistors (**MOS** et **TTL**), ne générant que deux états logiques ('**0'** (0v (masse)) et '**1'** (5v (Vcc)) schématisés par la figure 1.8.

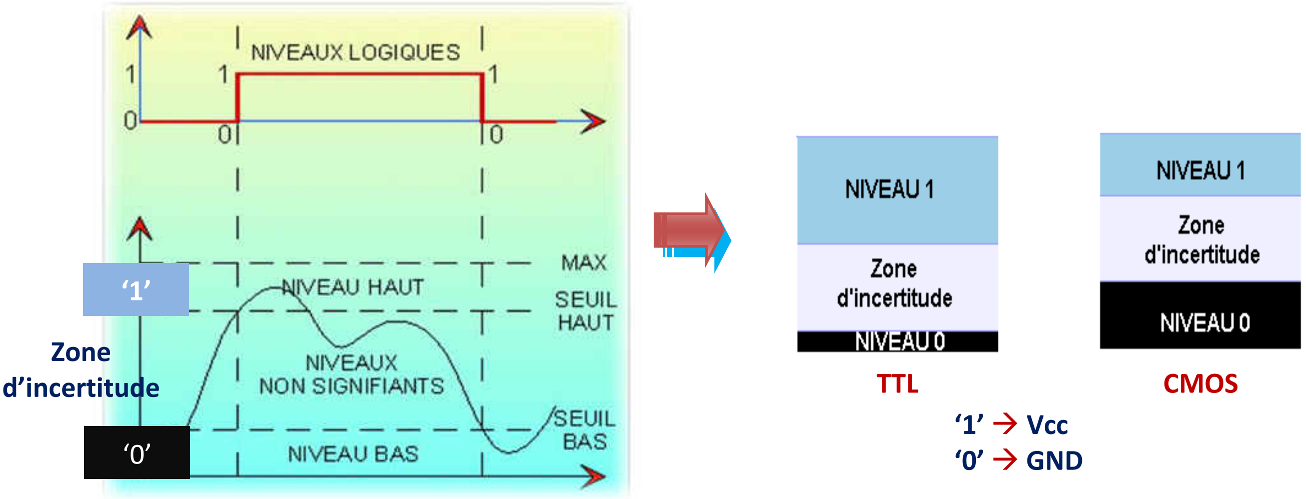


Figure 1.8 : Valeurs électriques des niveaux logiques.

⌖ La raison pour laquelle les ordinateurs manipulent des données binaires est liée au fonctionnement de leurs composants physiques. Les transistors et les condensateurs, qui sont les éléments de base d'un ordinateur, possèdent **deux états stables** : Activé/désactivé ou Saturé/Passant (**ON**/**OFF**).

- Ainsi, un transistor dans l'état activé va-t-il stocker l'information '1' (ou '0' s'il est dans l'état désactivé).

1.2. Conversion analogiques et numériques

Dans une chaine d'acquisition et de traitement de données, quatre blocs fonctionnels sont responsables de différentes actions de conversions analogiques et numériques, illustrés par l'exemple de la figure 1.9. La figure 1.10 illustre d'une manière détaillée le processus d'acquisition des données via les capteurs, traitement de ces données pour donner les actions convenables pour qur le robot manipulateur doit suivre un comportement désiré.

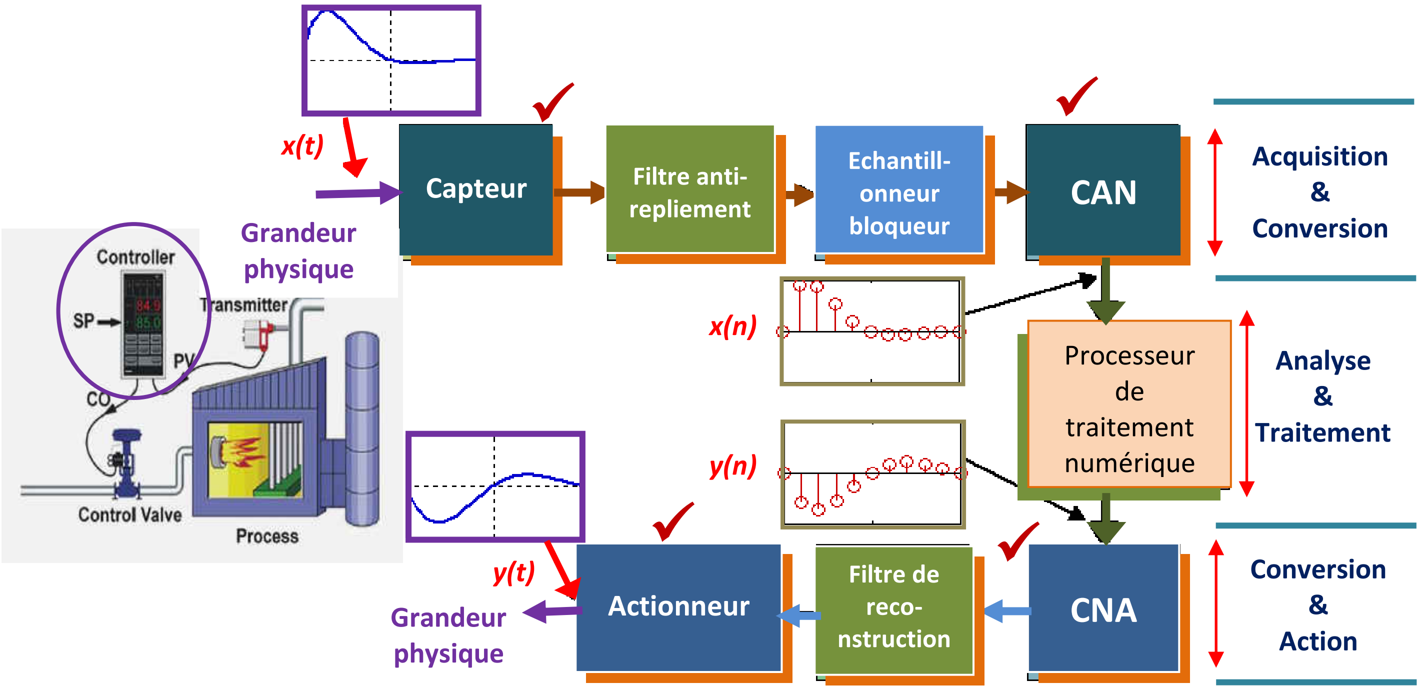


Figure 1.9 : Chaine d'acquisition et de traitement de données.

- ☒ **Capteur** : Compsant electronique qui permett la conversion d'un signal physique (analogique) en un signal électrique (analogique).
- ☒ **CAN** : Circuit qui permet la conversion d'un signal analogique en un signal numérique.
- ☒ **CNA** : Circuit qui permet la conversion d'un signal numérique en un signal analogique.
- ☒ **Transducteur** (Actionneur) : Compsant electromécanique qui assure la conversion d'un signal numérique en un signal électrique ou mécanique (analogique) bien adapté à la nature du système physique.

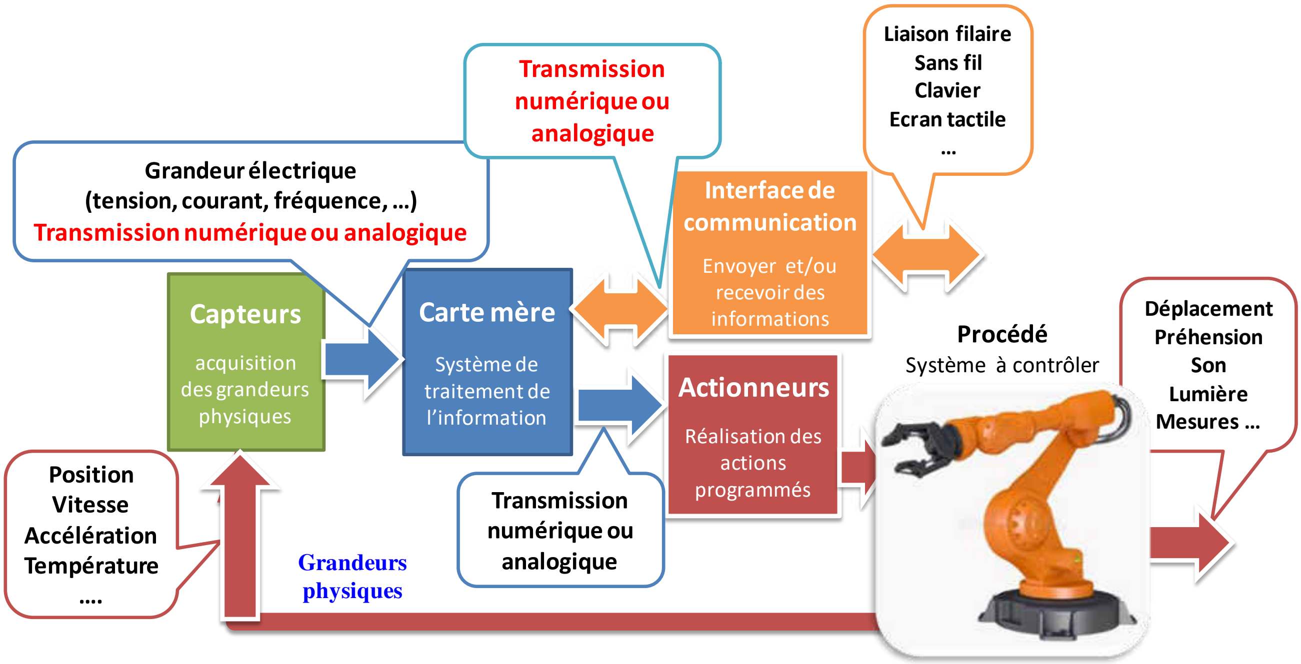


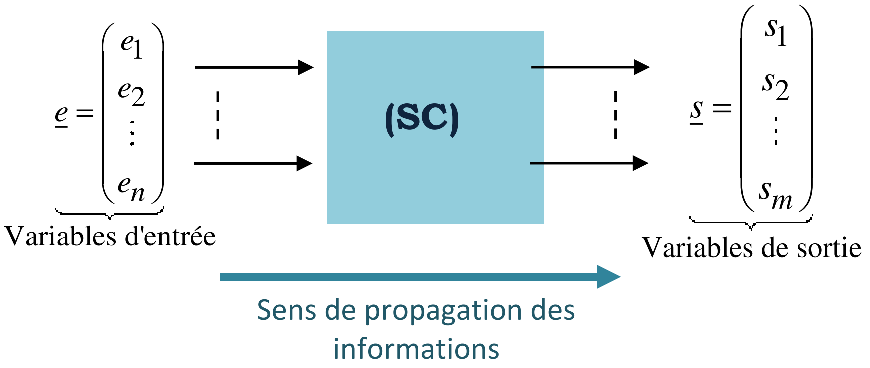
Figure 1.10 : Exemple d'une chaine de contrôle industriel d'un robot.

1.3. Systèmes logiques

1.3.1. Systèmes logiques combinatoires

Dans un système combinatoire (SC), les valeurs des variables de sortie sont complètement déterminées (uniquement) par les valeurs des variables d'entrée.

Soit le système combinatoire (**SC**) suivant :



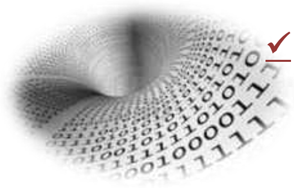
Entrée	Sortie	Type de système
Valeur	Valeur	SISO (Single Input – Single Output)
Vecteur	Valeur	MISO (Multiple Input – Single Output)
Valeur	Vecteur	SIMO (Single Input – Multiple Output)
Vecteur	Vecteur	MIMO (Multiple Input – Multiple Output)

Figure 1.11 : Structure et types des systèmes combinatoires.

à noter!

$\underline{s} = f(\underline{e})$ est traduite par la table des combinaisons (table de vérité). *La valeur actuelle de la variable de sortie dépend seulement de la valeur actuelle des variables d'entrées, c'est- à-dire, il n'ya de dépendance de l'état vis-à-vis des états précédents :*

$$\underline{s}(t) = f\left(e_1(t),\ e_2(t),\ \ldots,\ e_n(t)\right)$$



- Une variable logique est une variable qui permettera de caractériser l'état d'un ou de plusieurs éléments logiques dont le comportement est décrit par des états binaires (0, 1).
- L'analyse et la synthèse des systèmes combinatoires s'effectuent par application de *l'algèbre de Boole* comme outil principale.
- Dans un SC, la propagation des informations (données) est unidirectionnelle de l'entrée(s) vers la(es) sortie(s).
- Si l'entrée $\underline{e}$ est fixée → Sortie $\underline{s}$ connue.
- $\underline{s} = f\left(\underline{e}\right)$ est traduite par la table des combinaisons ou tout simplement la table de vérité.
- Une autre classe des systèmes logiques : ***Les systèmes séquentiels***, pour lesquels un même vecteur d'entrée $\underline{e} = (e_1 \quad e_2 \quad \ldots \quad e_n)^T$ ne donnera pas toujours le même vecteur de sortie.

1.3.2. Systèmes logiques séquentiels

Contrairement aux systèmes combinatoires, la valeur actuelle de la variable de sortie ne dépend pas seulement de la valeur actuelle des variables d'entrées, mais aussi de la valeur précédente (historique ou l'état passé), c'est- à-dire qu'il ya une dépendance de l'état vis-à-vis des états précédents :

$$\underline{s}(t) = f\left(e_1(t), e_2(t), \ldots, e_n(t), \underline{s}(t-\tau)\right)$$

Prenons **l'exemple** suivant : On considère un système à une entrée $e(t)$ et une sortie $s(t)$. La sortie $s(t)$ du système doit changer la valeur à chaque front montant de l'entrée $e(t)$. Ce cahier des charges peut être représenté par le chronogramme suivant :

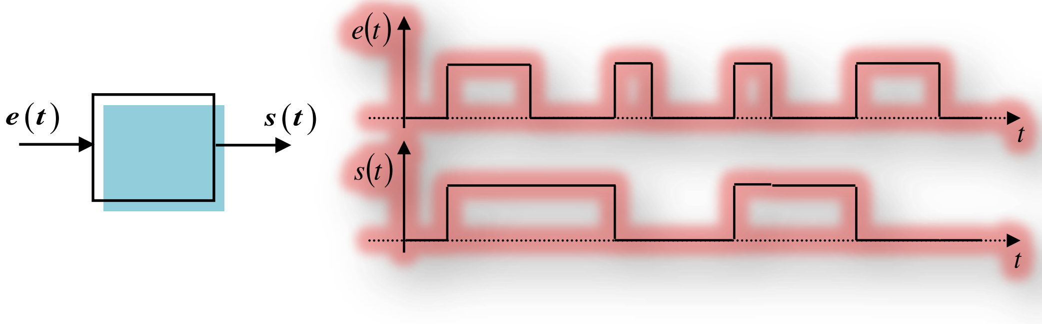


Figure 1.12 : Formes des signaux d'entrée/sortie.

- Pour une même valeur de $e(t)$, $s(t)$ peut prendre deux valeurs '0' ou '1'. Ce système n'est pas combinatoire. Donc, on ne peut pas définir $s(t) = f(e(t))$.

- ✓ Par contre la valeur de $s(t)$ peut être déterminée en prendre en considération les états précédents. Le système a en **mémoire** la valeur de $s(t)$ avant le changement.

En savoir+

- ✓ Un **système séquentiel** est un système dont les sorties à l'instant t dépendent à la fois des entrées à cet instant, mais aussi de ce qui s'est passé auparavant : l'histoire du système. Cette histoire sera représentée par une succession d'***états*** que prend le système au cours du temps.
- ✓ Le changement d'état sera provoqué par une variation des entrées. Les sorties sont fonction de l'état du système.
- ✓ *L'élément de base d'un système séquentiel est la **bascule** (bistable).*
- ✓ Quand le nouvel état pourra être déterminé uniquement à partir de l'état immédiatement précédent et des entrées, le système sera dit markovien.
- ✓ Les variables qui permettent de définir l'état du système à un instant donné *t*, dites : Variables internes ou secondaires. Elles apportent une information sur l'état du système à un instant *t*. Cette information sur l'état du système doit ramenée ou rebouclée vers l'entrée. Il ya, donc, des boucles de réaction dans le système.
- ✓ Les variables du système séquentiel (Variables primaires -Variables internes) permettant de caractériser l'état du système.

Definition

Un système séquentiel pourra être représenté par le schéma suivant:

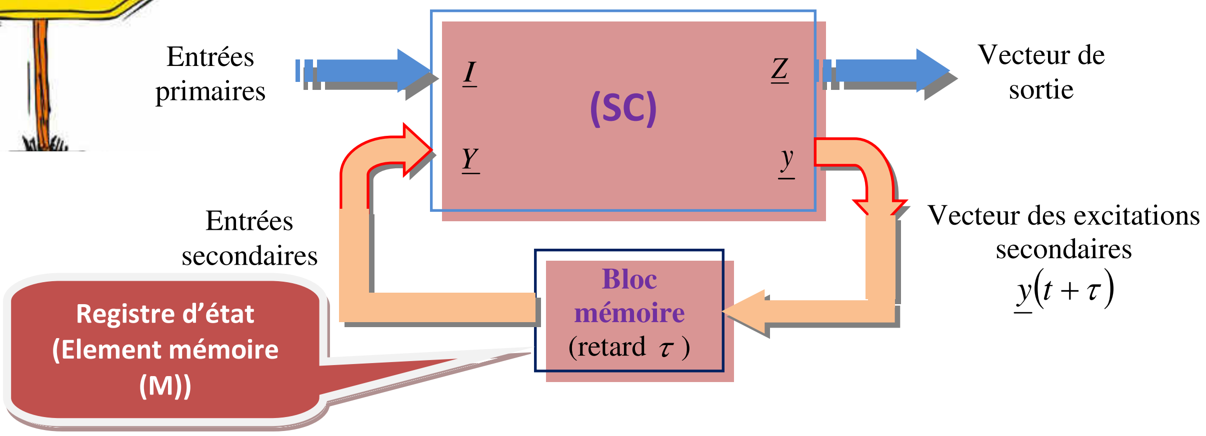
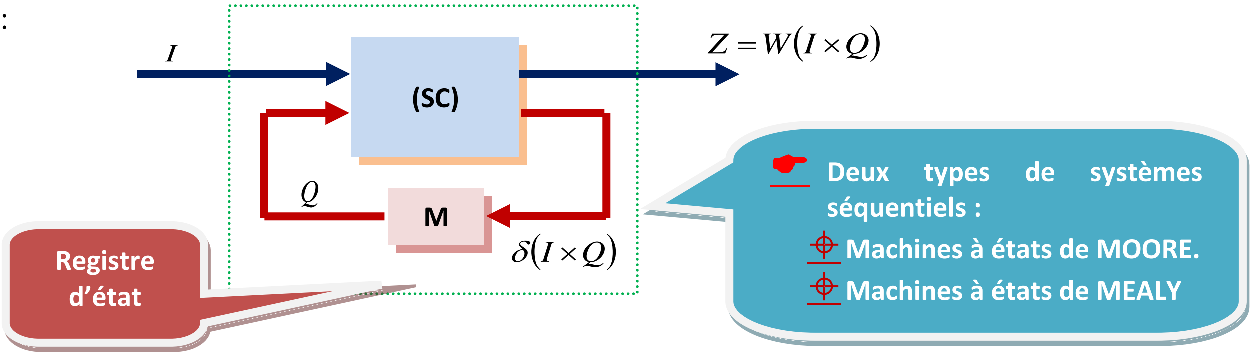


Figure 1.13 : *Structure d'un système séquentiel.*

D'une manière simplifiée, un système (machine) séquentiel est schématisé par la figure suivante:



Deux types de systèmes séquentiels :

- Machines à états de MOORE.
- Machines à états de MEALY

Figure 1.14 : Structure simplifiée d'un système séquentiel.

Deux types de systèmes séquentiels sont rencontrés, comme illustrent les figure 1.5 et 1.6 respectivement, à savoir :

- ✓ Les machines séquentielles de MOORE.
- ✓ Les machines séquentielles de MEALY.

Figure 1.15 : Machine à états de Moore.

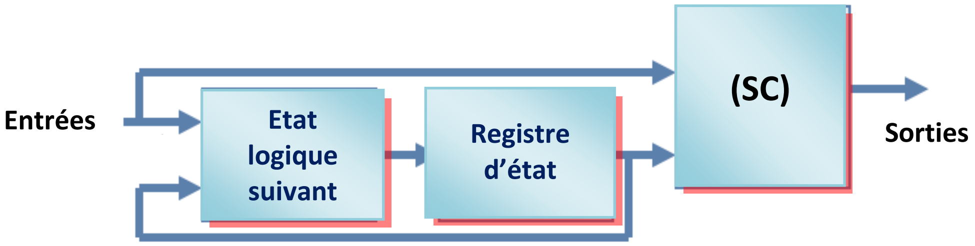
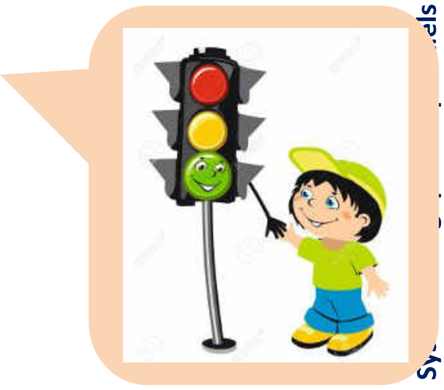


Figure 1.16 : Machine à états de Mealy.

D'une manière générale, une machine séquentielle (MS) est définie par le quintuplet :

$$\mathrm{MS} = \left(\underbrace{I,\ Q,\ Z}_{\text{Variable d'entrée/sortie}} \quad , \quad \overset{\text{Fonctions}}{\delta,\ W} \right)$$



avec,

$I = (I_1, I_2, \dots, I_n)^T$: Vecteur d'entrée.

Q : Etat du système.

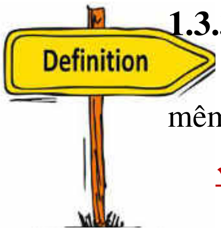
$Z = (Z_1, Z_2, \dots, Z_m)^T$: Vecteur de sortie.

$\delta : I \times Q \rightarrow Q$: Fonction d'état futur (suivant).

$W : I \times Q \rightarrow Z$: Fonction de sortie.

- Le bloc **(SC)** dans la figure précédente est un circuit combinatoire. A partir d'un vecteur d'entrée $\{\in \boldsymbol{I}\}$ et de l'état présent $\{\in \boldsymbol{Q}\}$ élabore d'une part, le vecteur de sortie $\{\in \boldsymbol{Z},\ (\boldsymbol{Z} = \boldsymbol{W}(\boldsymbol{I},\boldsymbol{Q}))\}$ et d'autre part l'état futur $\{\in \boldsymbol{Q},\ (\delta(\boldsymbol{I},\boldsymbol{Q}))\}$.
- Le bloc **M** est un **registre d'état** (bloc de mémorisation). Il permet de ramener vers l'entrée une information sur l'état interne du système.

1.3.3. Système Synchrone / Asynchrone



Un système est dit synchrone lorsque son évolution est contrôlée par les entrées elles-mêmes ou par un signal particulier : **Horloge** (**Clk**).

- ✓ L'évolution d'un système asynchrone n'est contrôlée par aucune entrée particulière de telle sorte que pour un vecteur d'entrée appliqué, le système évolue jusqu'à ce que l'état présent et l'état suivant soient identiques.

1.4. Modélisation des systèmes séquentiels

Le cahier des charges est constitué d'une suite de phrases décrivant le fonctionnement désiré du système. C'est la première étape de la conception d'un système.

- ✓ Afin d'analyser et de valider le cahier des charges, on le traduira en un formalisme qui ne permet aucune erreur d'interprétation. On parlera de la **modélisation**.
- ✓ Les modèles obtenus pourront être utilisés aussi pour la synthèse (élaboration matérielle de la commande) :

  - ✓ Chronogramme (diagramme des temps).
  - ✓ Graphe de fluence (transitions).
  - ✓ Tableaux d'état.
  - ✓ Graphe d'état.
  - ✓ Graphe d'événement.
  - ✓ GRAFCET.
  - ✓ Réseaux de Petri.
  - ✓ … etc.

Les étapes de développement 'résolution' des systèmes logiques (combinatoires ou séquentiels) sont schématisés par cet organigramme.

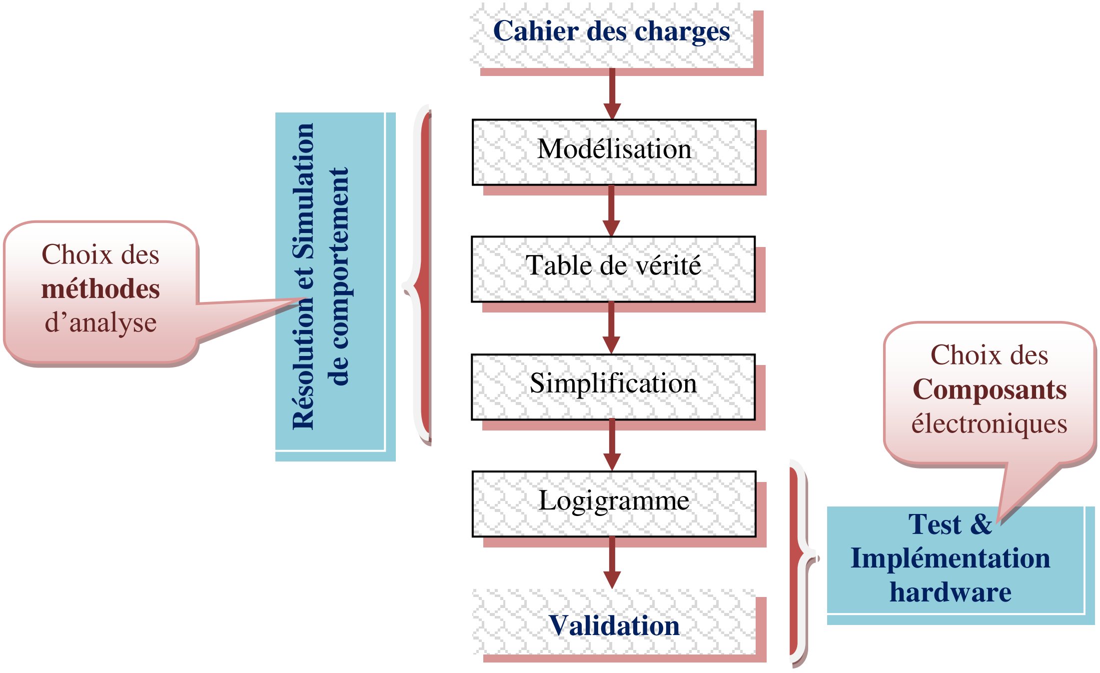


Figure 1.17 : *Etapes d'analyse et de validation des systèmes logiques.*

1.5. Synthèse des systèmes logiques

La synthèse des circuits logiques a pour but la réalisation d'une fonction logique qui remplit un **cahier des charges** et qui satisfait également à d'autres critères tels que coût et l'encombrement minimum, la consommation réduite par exemple.

Le nombre de circuits à produire, le matériel à disposition, le délai de réalisation, etc. sont d'autres paramètres dont il faut tenir compte lors de la synthèse.

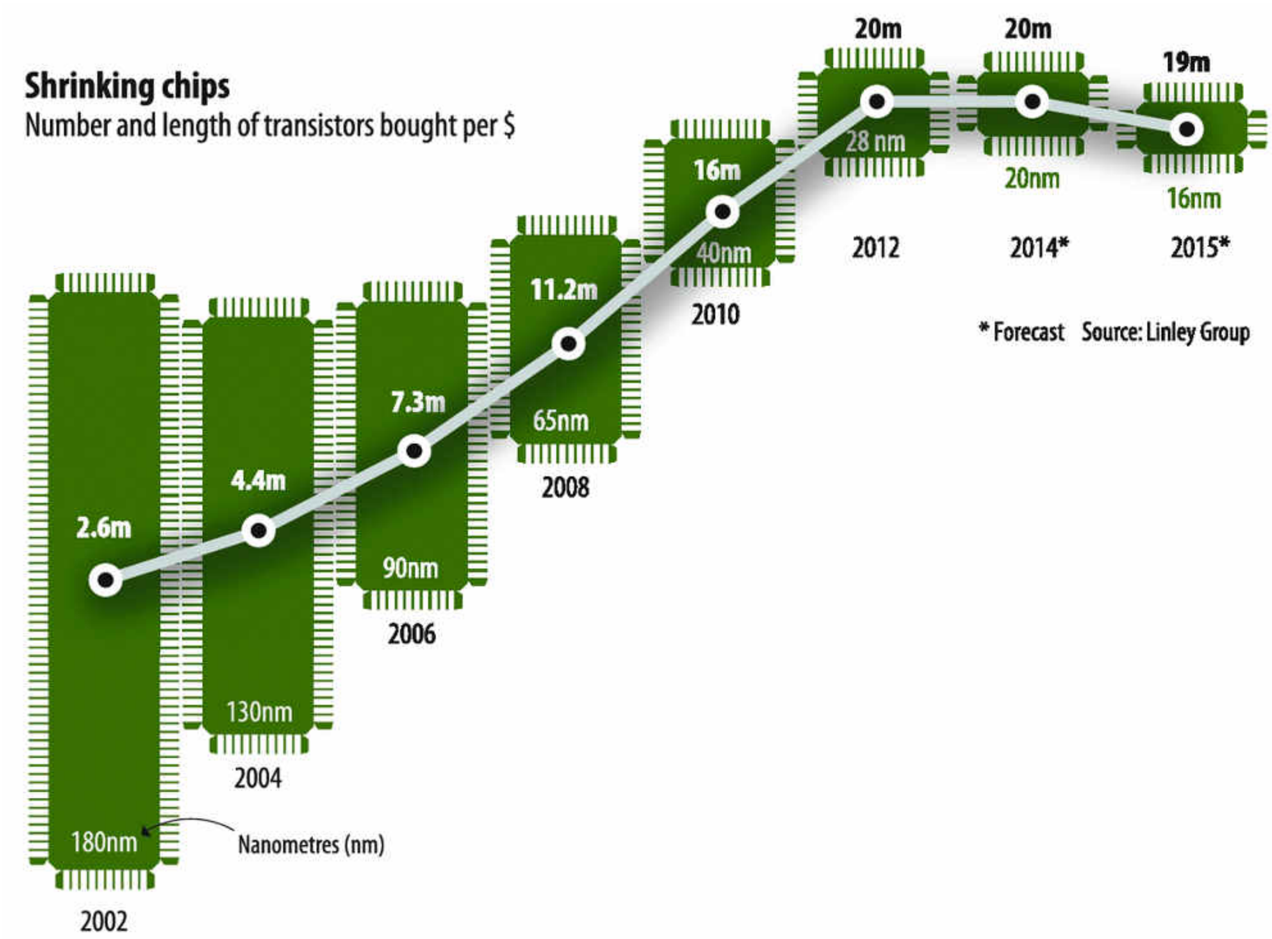
Donc, réaliser un montage ou un schéma électrique correspond à la description d'un problème technique défini par cahier des charges selon les circuits disponibles. Pour la recherche d'une solution *optimale* (meilleure) pour la conception (*synthèse*) des systèmes logiques, il faut prendre en consédération :

- *Les éléments de connexion universels* (ECU).
- *Les circuits intégrés* (CI) : En plus de la technologie utilisée pour la fabrication des CI (RTL, DTL, TTL, ECL, CMOS), l'échelle d'intégration mesuré par le nombre des transistors par puce est considéré comme une caractéristique fondamentale dans la classification des CI. Le tableau ci-dessous illustre le classement par echelle d'intégration.

Table 1.1 : *Familles des circuits intégrés.*

Famille	Description	Echelle d'intégration	Type de système
SSI	Small Scal Integration	**< 12** gates/chip	Portes logiques
MSI	Most Scal Integration	**12- 99** gates/chip	**Circuits combintoires**
LSI	Large Scal Integration	**1K** gates/chip	Systèmes à Micrprocesseurs
VLSI	Very Large Scal Integration	**10K** gates/chip	Systèmes à Microprocesseurs
ULSI	Ultra Large Scal Integration	**100K** gates/chip	Micrprocesseurs / Microcontrôleurs / FPGA…
GSI	Giga Scale Integration	**1Meg** gates/chip	

- Un circuit intégré contenant les portes logiques intégrées par divers procédés technologiques sur une petite plaquette de semi-conducteurs (Silicium, Germanium, …etc).
- Les technologues (Echelles d'intégrations) doivent jongler avec des procédés de fabrication de plus en plus complexes pour continuer à suivre la "loi de Moore".



- Un circuit combinatoire a des caractéristiques temporelles qui dépendent de la technologie employée

- La puce est enfermée dans un boîtier Sur les côtés sont disposées des broches (ou pattes).

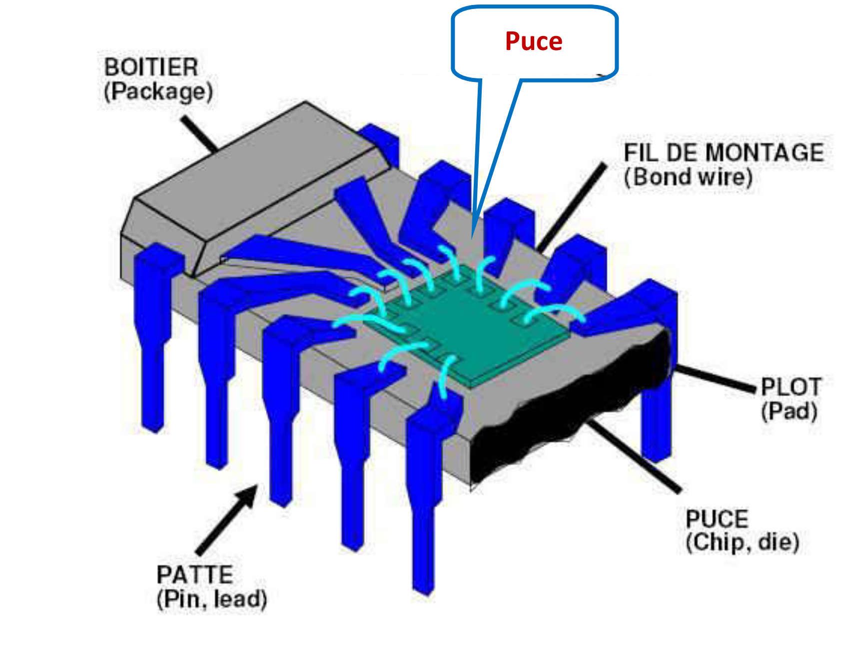


Figure 1.18 : *Structure externe d'un circuit intégré.*

1.6. Etapes de synthèse des systèmes logiques

Afin d'avoir résoudre un problème logique, le concepteur doit suivre les étapes suivantes :

1. Identifier les entrées et les sorties (IN / OUT) du circuit extrait du cahier de charge décrivant le fonctionnement du système logique.
2. Construire la table (les tables) de vérité.
3. Identifier chaque fonction à partir de la table de vérité.
4. Simplifier chaque fonction par application des méthodes de simplications algébriques et graphiques.
5. Tracer le schéma du circuit (logigramme) par l'emploi des portes logiques disponibles.

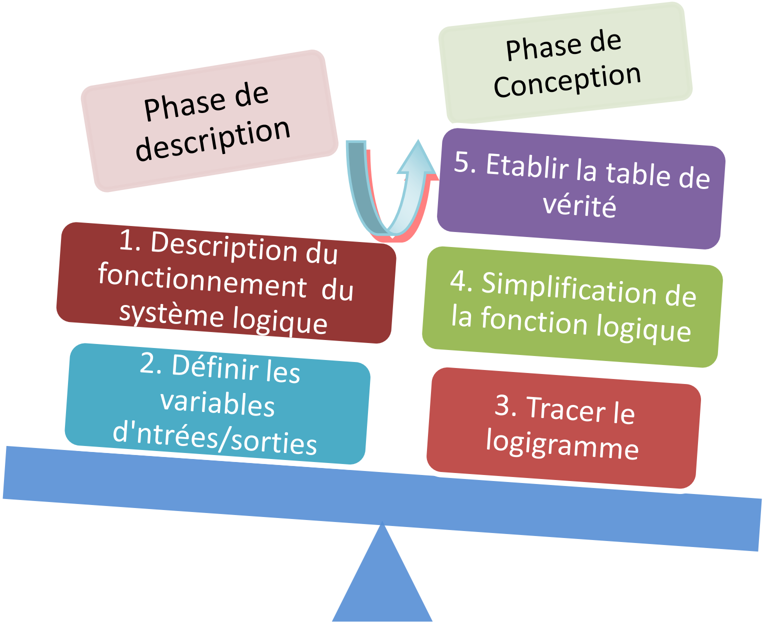


Figure 1.19 : *Etapes d'analyse et de synthèse des systèmes logiques.*

En savoir +

Une autre alternative de conception consiste à identifier le rôle ou le fonctionnement d'un circuit (ou d'un montage à base des circuits logiques), on se basant sur l'opération inverse de la procédure précédente, comme illustre le schéma de la figure ci-dessous.

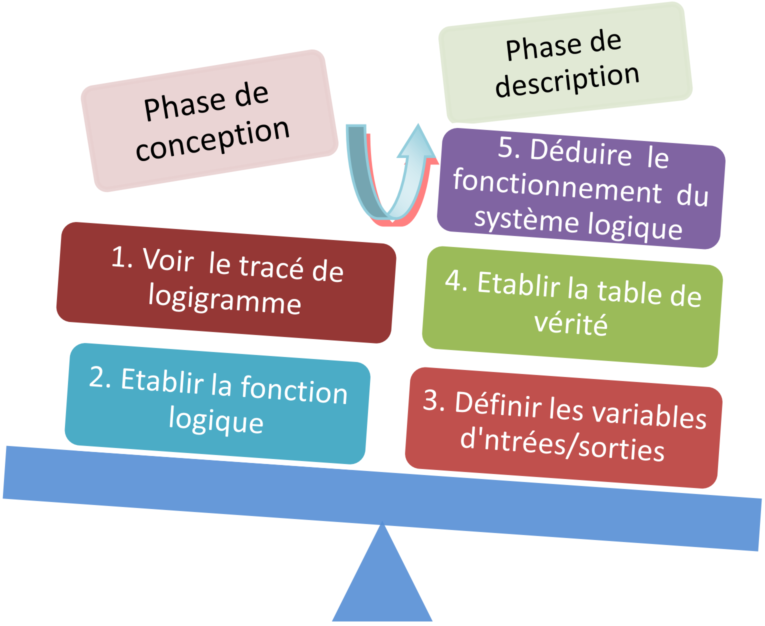


Figure 1.20 : *Etapes de déduction de fonctionnement des systèmes logiques.*

1.7. Exercices

EXERCICE 01 :

1. Définir un signal. Donner les caractéristiques d'un signal.
2. Qu'est-ce qu'un signal analogique ?
3. Qu'est-ce qu'un signal numérique ?
4. Comment passe-t-on de l'analogique au numérique ?
    - Qu'est-ce que le codage binaire ?
    - Qu'est-ce que la résolution du convertisseur ?
    - Comment fait-on l'échantillonnage ?
5. Comment fait-on la quantification ?
6. Comment transmet-on un signal numérique?

EXERCICE 02 :

Classer dans le tableau ci-contre les différents signaux :

✓ Amplitude de la voix humaine.

✓ Vitesse de rotation d'un moteur électrique.

✓ Signal « morse ».

✓ Signal d'une fin de course.

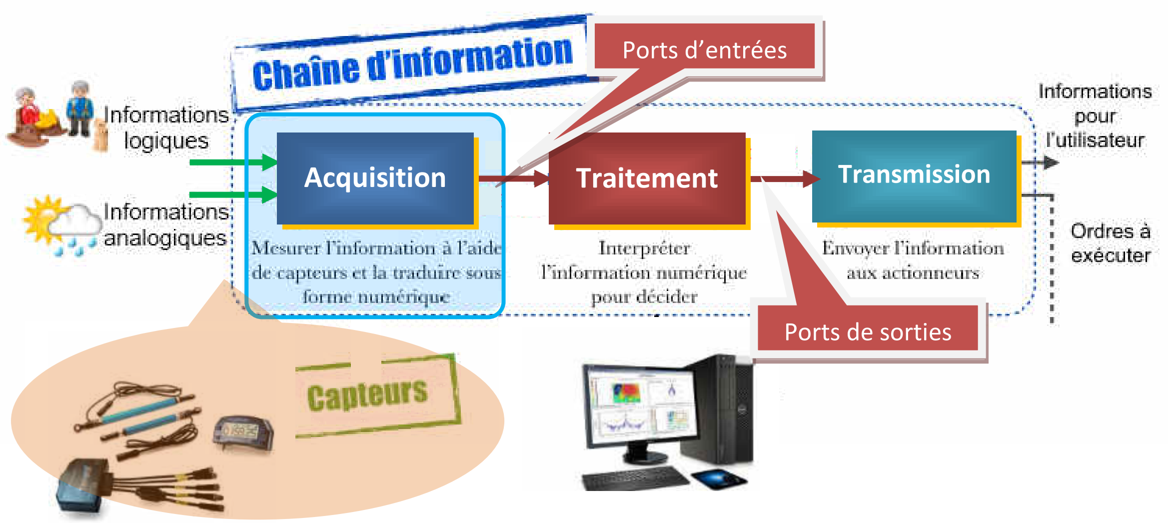
✓ Informations issue d'un clavier d'ordinateur.

✓ Pression d'air dans un compresseur.

Analogique	Numérique

EXERCICE 03 :

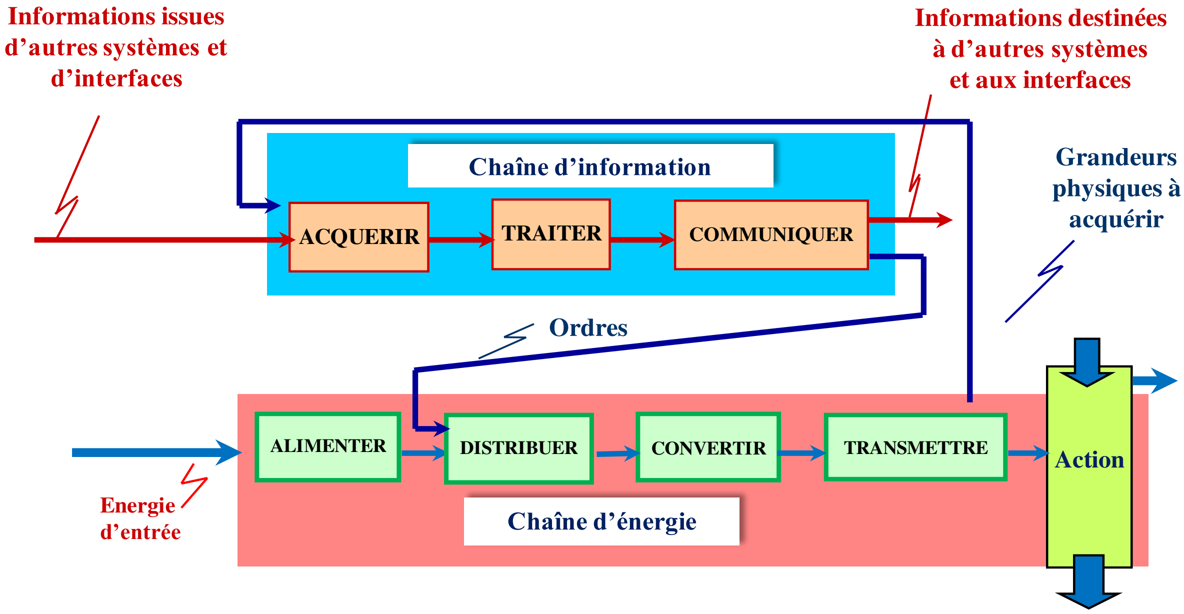
Soit le schéma de la figure de la chaîne d'information suivante :



- ✓ Expliquer le rôle de chaque bloc du schéma précédent.
- ✓ Représenter les différents types de données transmises dans la chaîne.
- ✓ Citer les différents types de ports d'entrées / sorties. Discuter…
- ✓ Donner un exemple concret d'une chaine d'acquisition et de traitement d'un phénomène physique quelconque.

EXERCICE 04 :

- ✓ Expliquer le rôle de chaque bloc du schéma suivant :



- ✓ Donner la forme des signaux dans l'entrée et la sortie de chaque bloc. Discuter…

EXERCICE 05 :

- ✓ Existe-t-il une représentation universelle de l'information ?
- ✓ Par quels moyens peut-on représenter des symboles et des nombres ?
- ✓ Est-il possible de construire une représentation exacte du monde réel ?
- ✓ Quand et comment envoyer des données de façon à pouvoir les recevoir à distance ?

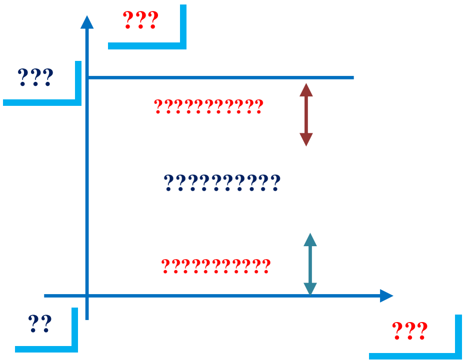
EXERCICE 06 :

Une disquette 3 Pouce ½ à une taille de stockage de 1.44 Mo.

- ✓ Citez les différentes unités de mesure de l'information.
- ✓ Quel est le nombre de caractères qui peut contenir un fichier dont la taille est égale à la taille de la disquette ?
- ✓ Combien de disquettes peut contenir un CD-ROM dont la taille est 650 Mo ?
- ✓ Combien de disquettes peut contenir un disque dur de 80 Go ?

EXERCICE 07 :

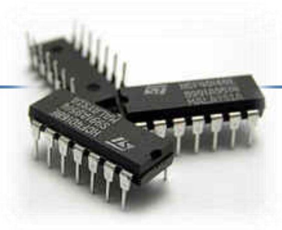
- ✓ Completer le schéma de la figure suivante :



- ✓ Qelles sont les informations tirées de cette figure ? Discuter…

EXERCICE 08 :

- ✓ Donner un classement adéquat des systèmes logiques ?
- ✓ D'où elle vient l'appelation système combinatoire ?
- ✓ D'où elle vient l'appelation système séquentiel ? Donner la structure générale d'un système séquentiel. Expliquer la fontion de chaque bloc élementaire ? Discuter…
- ✓ Donner des exemples d'application de ces systèmes.



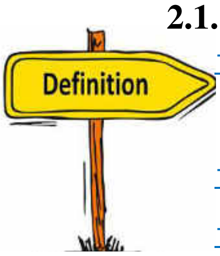
CHAPITRE 2

SYSTEMES DE NUMERATION - CODAGE DE L'INFORMATION

Objectifs

- *Traiter en détails les différents systèmes de numération : systèmes décimal, binaire, octal et hexadécimal ainsi que les méthodes de conversion entre les systèmes de numération..*
- *Etudier les codes numériques tels que les codes BCD, GRAY et ASCII.*
- *Traiter les opérations arithmétiques sur les nombres.*

2.1. Introduction



- Le système de numération est la représentation d'une grandeur numérique par des symboles.
- Manière d'énoncer ou d'écrire les nombres.
- Le nombre de symboles utilisés caractérise le numéro de la ***base.***
- Dans la vie quotidienne, nous utilisons la **base 10**, les ordinateurs utilisent la **base 2**.
- Le système digital ou numérique) est développée autour de la simple notion : *Exprimer toutes informations par une succession de **bit**s (**Bi**nary Digi**t**).*

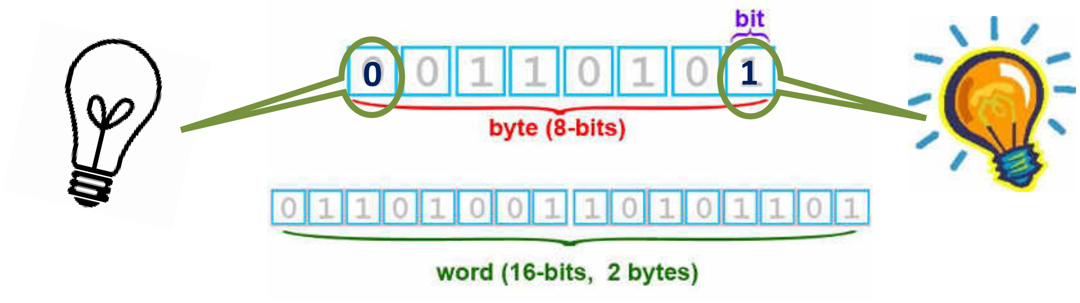


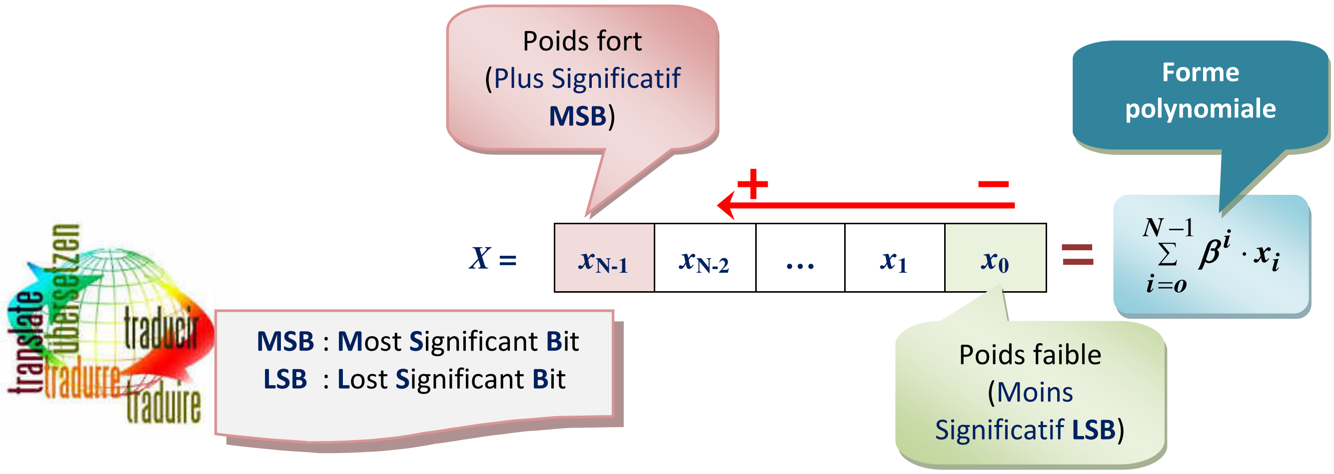
Figure 2.1 : *Représentation des informations en séquences de bits.*

- Le système numérique habituel (usuel) ou tout simplement le système décimal contient 10 digits représenté par les symboles **0,1, 2, 3, 4, 5, 6, 7, 8 et 9**.
- L'avantage du système binaire est la correspondance directe entre les symboles binaires {0, 1}et les deux états stables des dispositfs electroniques d'un ordinateur.

Figure 2.2 : *Sens de propagation des informations.*

2.2. Représentaion polynomiale d'un nombre

- Dans le mode de représentation des informations, en base ***n*** on utilise ***n*** symboles (chiffres) différents. Mais la valeur du chiffre change selon sa position.
- Si un naturel X s'écrit en base β sur N chiffres.



$$X = \boxed{x_{N-1}}\boxed{x_{N-2}}\boxed{\dots}\boxed{x_1}\boxed{x_0} = \sum_{i=o}^{N-1} \beta^i \cdot x_i$$

MSB : **M**ost **S**ignificant **B**it
LSB : **L**ost **S**ignificant **B**it

- ☒ On peut décomposer tout nobre N en fonction des puissances entières de la base de son système de numération.
- ☒ On note $\beta^{i=0}=1$.
- ☒ La forme polynomiale n'est que **la conversion décimale** du nombre en cours d'étude.
- ☒ Le rang d'un chiffre d'un nombre de base β quelconque est égal à l'exposant de la base (*i*) associé à ce chiffre dans la représentation polynomiale du nombre considéré.
- ☒ Au llieu de *rang*, on peut utiliser le terme *poids*.

☒

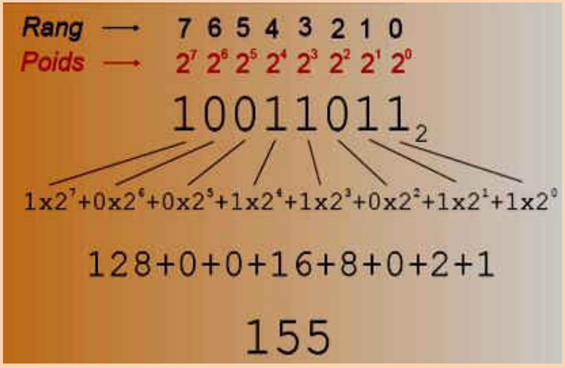


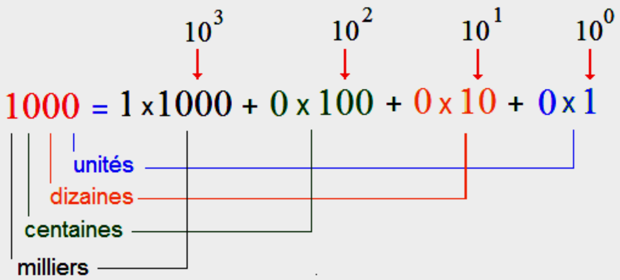
Table 2.1 : Exemples des bases.

Base	Symboles	Type de système
2	{0, 1}	Binaire
6	{0,1, …, 5}	
8	{0,1, …, 6, 7}	Octal
10	{0, 1, …, 8, 9}	Décimal
12	{0, 1, …, 8, 9, A, B, C}	
16	{0, 1, …, 8, 9, A, B, C, D, E, F}	Hexadécimal

Exemples :

⌖ Ecrire sous forme polynomiale les nombres suivants :
$(1000)_{10}$, $(95325)_{10}$, $(95325.\ 95325)_{10}$, $(6734)_8$, $(101101)_2$, $(A732)_{16}$

✓



$$(95325)_{10} = 5\times10^0 + 2\times10^1 + 3\times10^2 + 5\times10^3 + 9\times10^4$$

$$(95325.95325)_{10} = \underbrace{5\times10^0 + 2\times10^1 + 3\times10^2 + 5\times10^3 + 9\times10^4}_{\text{Partie entière}} + \underbrace{5\times10^{-5} + 2\times10^{-4} + 3\times10^{-3} + 5\times10^{-2} + 9\times10^{-1}}_{\text{Partie fractionnaire}}$$

$$(6734)_8 = 4\times8^0 + 3\times8^1 + 7\times8^2 + 6\times8^3$$

✓ $$= (3458)_{10}$$

$$(101101)_2 = 1\times2^0 + 0\times2^1 + 1\times2^2 + 1\times2^3 + 0\times2^4 + 1\times2^5$$

$$= (45)_{10}$$

$$(A732)_{16} = 2\times16^0 + 3\times16^1 + 7\times16^2 + 10\times16^3 + 9\times16^4$$

$$= (42802)_{10}$$

⊕ Soit $N = (23456)_{10}$, déterminer le rang du chiffre 5 et 3.

✓ Rang 5 = 1.

✓ Rang 3 = 3.

2.3. Chagement de base (Conversion)

Le passage d'une base à l'autre s'effectue, généralement par divisions successives du nombre de la base source sur la base de destination. Un point à signaler est que la décomposition sous forme polynomiale donne toujours la valeur décimale du nombre exprimé en base β. Il existe trois types de conversions :

- ☒ Conversion du système décimal en un autre système non décimal: cette opération s'appelle le **codage.**
- ☒ Conversion d'un système non décimal au système décimal: cette opération s'appelle le **décodage.**
- ☒ Conversion entre deux systèmes non décimaux: cette opération s'appelle le **transcodage.**

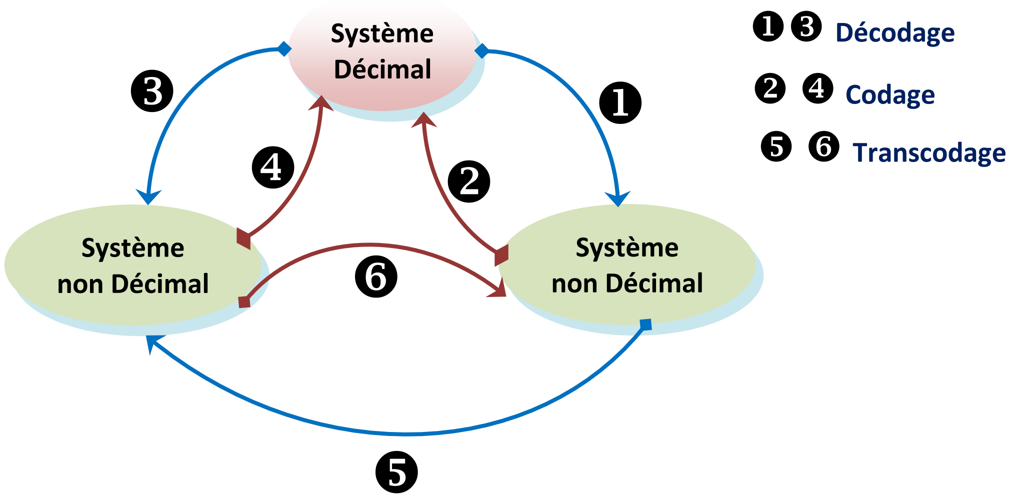


Figure 2.3 : Conversion des systèmes de numération.

Le tableau ci-dessous illustre les différents types de conversion avec des exemples concrets.

Table 2.2 : *Différents types de conversions.*

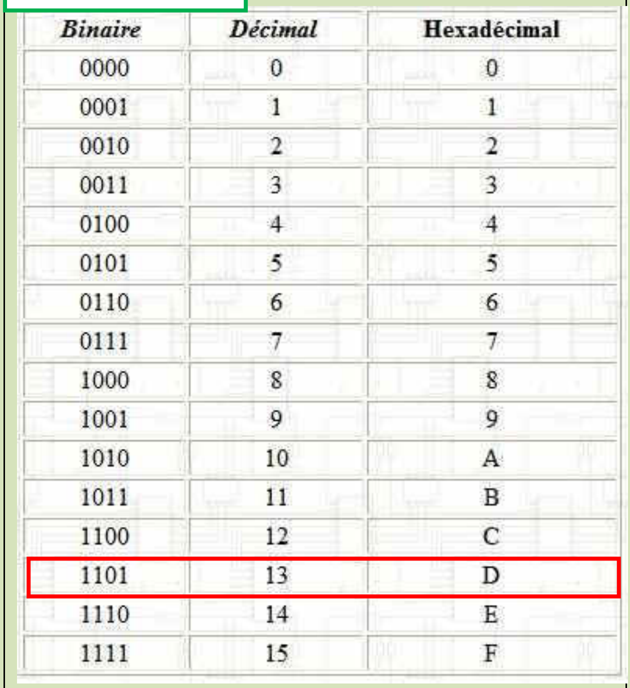
<table>
<tr><th>Type de Conversion</th><th>Description par des exemples</th><th>Remarques</th></tr>
<tr><td>Binaire - Décimal</td><td>

$$(11011)_2 = 1\times 2^0 + 1\times 2^1 + 0\times 2^2 + 1\times 2^3 + 1\times 2^4$$
$$= (27)_{10}$$
$$\left(\overline{101}.\overline{101}\right)_2 = 1\times 2^0 + 0\times 2^1 + 1\times 2^2$$
$$+ 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}$$
$$= (5.625)_{10}$$

</td><td>Application de la forme polynomiale</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Décimal – Binaire</td><td>

1. Table des équivalences

... 16 8 4 2 1



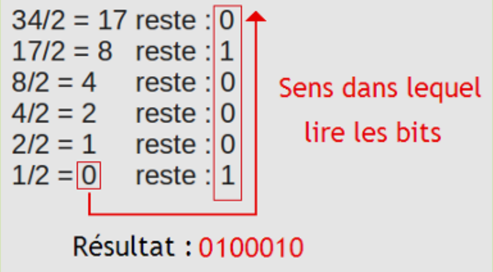
Binaire	*Décimal*	**Hexadécimal**
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Exemples :

$$(13)_{10} = 8+4+3 = (1101)_2$$
$$(25)_{10} = 16+8+1 = (11001)_2$$

2. méthodes de division

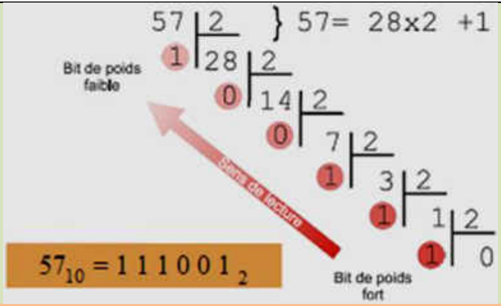
$(34)_{10}$ = (?)$_2$



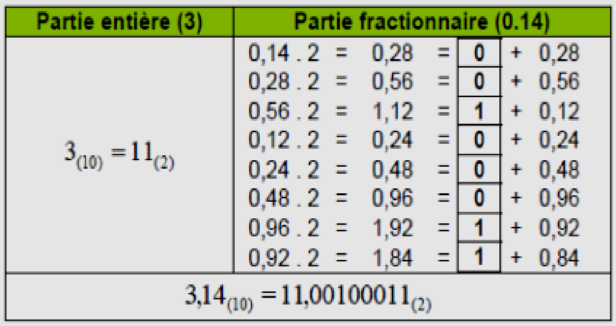
34/2 = 17 reste : 0
17/2 = 8 reste : 1
8/2 = 4 reste : 0
4/2 = 2 reste : 0
2/2 = 1 reste : 0
1/2 = 0 reste : 1
Résultat : 0100010

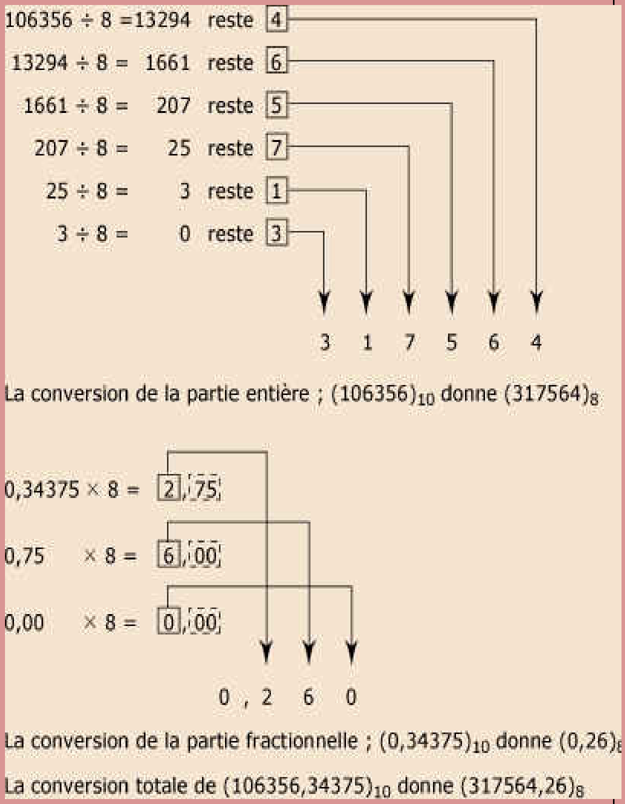
$(57)_{10}$ = (?)$_2$

</td><td>Soit on utilise le tableau des équivalences, soit on utilise les divisions successives sur la base 2.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>


$(3.14)_{10}$ = (?)$_2$


<table>
<tr><th>Partie entière (3)</th><th>Partie fractionnaire (0.14)</th></tr>
<tr><td rowspan="8">$3_{(10)} = 11_{(2)}$</td><td>0,14 . 2 = 0,28 = 0 + 0,28</td></tr>
<tr><td>0,28 . 2 = 0,56 = 0 + 0,56</td></tr>
<tr><td>0,56 . 2 = 1,12 = 1 + 0,12</td></tr>
<tr><td>0,12 . 2 = 0,24 = 0 + 0,24</td></tr>
<tr><td>0,24 . 2 = 0,48 = 0 + 0,48</td></tr>
<tr><td>0,48 . 2 = 0,96 = 0 + 0,96</td></tr>
<tr><td>0,96 . 2 = 1,92 = 1 + 0,92</td></tr>
<tr><td>0,92 . 2 = 1,84 = 1 + 0,84</td></tr>
<tr><td colspan="2">$3{,}14_{(10)} = 11{,}00100011_{(2)}$</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Décimal - Octal</td>
<td>
$(106356)_{10}$ = (?)$_8$



106356 ÷ 8 =13294 reste 4
13294 ÷ 8 = 1661 reste 6
1661 ÷ 8 = 207 reste 5
207 ÷ 8 = 25 reste 7
25 ÷ 8 = 3 reste 1
3 ÷ 8 = 0 reste 3

3 1 7 5 6 4

La conversion de la partie entière ; $(106356)_{10}$ donne $(317564)_8$

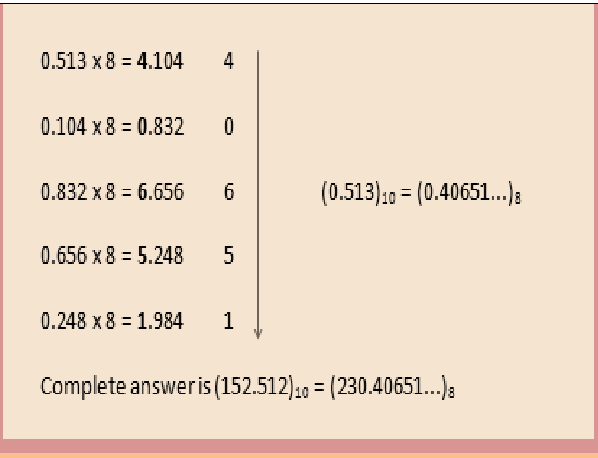
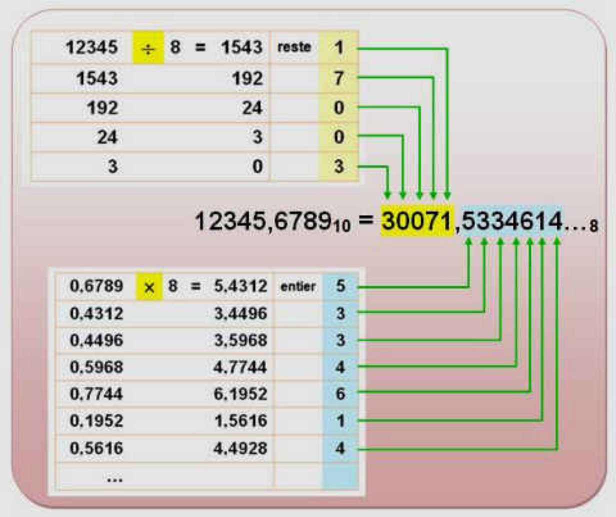
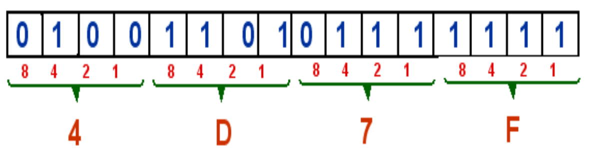
0,34375 × 8 = 2,75
0,75 × 8 = 6,00
0,00 × 8 = 0,00

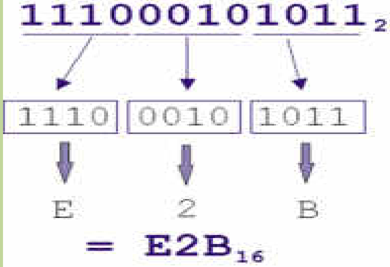
0 , 2 6 0

La conversion de la partie fractionnelle ; $(0{,}34375)_{10}$ donne $(0{,}26)_8$

La conversion totale de $(106356{,}34375)_{10}$ donne $(317564{,}26)_8$

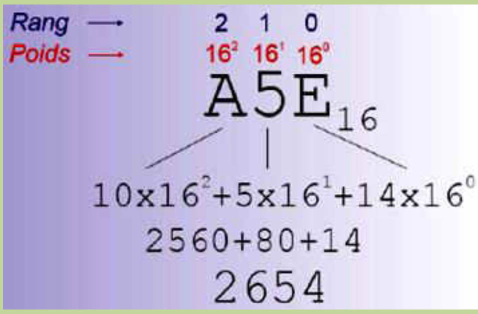
$(0.513)_{10}$ = (?)$_8$
</td>
<td>On utilise les divisions successives sur la base 8.</td>
</tr>
</table>

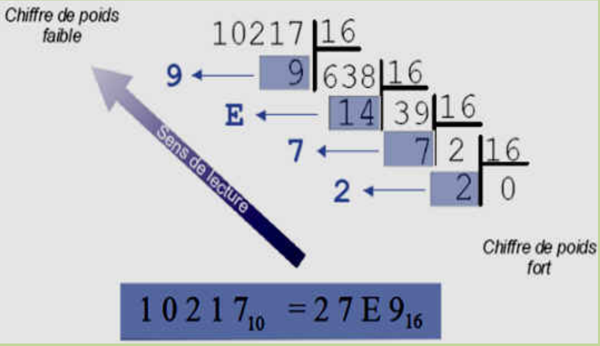
	 0.513 x 8 = 4.104 4 0.104 x 8 = 0.832 0 0.832 x 8 = 6.656 6 $(0.513)_{10} = (0.40651...)_8$ 0.656 x 8 = 5.248 5 0.248 x 8 = 1.984 1 Complete answer is $(152.512)_{10} = (230.40651...)_8$ **$(12345.6789)_{10} = (\ ?)_8$**  12345 ÷ 8 = 1543 reste 1 1543 192 7 192 24 0 24 3 0 3 0 3 $12345{,}6789_{10} = 30071{,}5334614\ldots_8$ 0,6789 × 8 = 5,4312 entier 5 0,4312 3,4496 3 0,4496 3,5968 3 0,5968 4,7744 4 0,7744 6,1952 6 0,1952 1,5616 1 0,5616 4,4928 4 ...	
Binaire - Octal	**$8 = 2^3$** $(110\ 111\ 011.001\ 101)_2 = (\ ?)_8$ $N = (\underline{110}\ \underline{111}\ \underline{011}.\underline{001}\ \underline{101})_2$ $(\mathbf{6}\quad \mathbf{7}\quad \mathbf{3}\ .\ \mathbf{1}\quad \mathbf{5})_8$ ⌖ Pour la conversion Octal → Binaire, c'est l'opération onverse. $N = (2\quad 4\quad 7\quad 6\ .\ 6\quad 7\quad 0\quad 1)_8$ $(010\ 100\ 111\ 110.110\ 111\ 000\ 001)_2$	A partir de la virgule, grouper les bits par bloc de trois en allant vers la guache pour la partie entière et vers la droite pour la partie fractionnaire → Convertir ensuite ces bloc en octal.
	Compter en octal ? ✓ **64, 65, 66, 67, ~~68~~, 70, 71, …** ✓ **275, 276, 277, ~~278~~, 300, 301, …**	On remarque la simplicité et la cohérence de la conversion Binaire → Octal
Binaire - Héxadécimal	**$16 = 2^4$** $(100110101111111)_2 = (\ ?)_{16}$  $(111000101011)_2 = (\ ?)_{16}$	On donne les mêmes propriétés qoue pour le système octal, mais cette fois on groupe les bits par blocs de quatre bits.

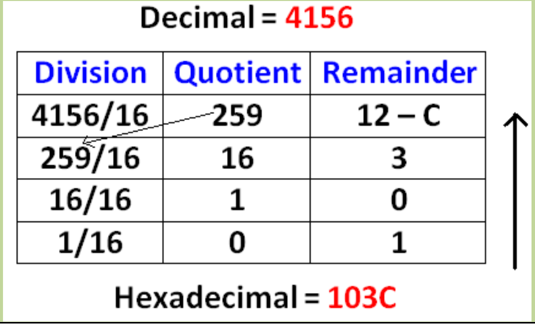
<table>
<tr>
<td></td>
<td>


(10011100101111010.0111)$_2$ = (?)$_8$

N = (<u>1001</u> <u>1100</u> <u>1011</u> <u>1010</u>.<u>0111</u>)$_2$
(9 C B A . 7)$_{16}$
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>
Compter en Héxadécimal ?
✓ 6B8, 6B9, 6BA, 6BB, 6BC, 6BD, 6BE, 6BF, 6C0, …
</td>
<td>On remarque la simplicité et la cohérence de la conversion Binaire → Héxadécimal</td>
</tr>
<tr>
<td>
Héxadécimal – Décimal


</td>
<td>
(A5E)$_{16}$ = (?)$_{10}$


</td>
<td>Application de la forme polynomiale</td>
</tr>
<tr>
<td>Décimal - Héxadécimal</td>
<td>
(10217)$_{10}$ = (?)$_{16}$


(4156)$_{10}$ = (?)$_{16}$


Decimal = 4156
<table>
<tr><th>Division</th><th>Quotient</th><th>Remainder</th></tr>
<tr><td>4156/16</td><td>259</td><td>12 – C</td></tr>
<tr><td>259/16</td><td>16</td><td>3</td></tr>
<tr><td>16/16</td><td>1</td><td>0</td></tr>
<tr><td>1/16</td><td>0</td><td>1</td></tr>
</table>
Hexadecimal = 103C
</td>
<td>On utilise les divisions successives sur la base 16.</td>
</tr>
</table>

Conversion rapide

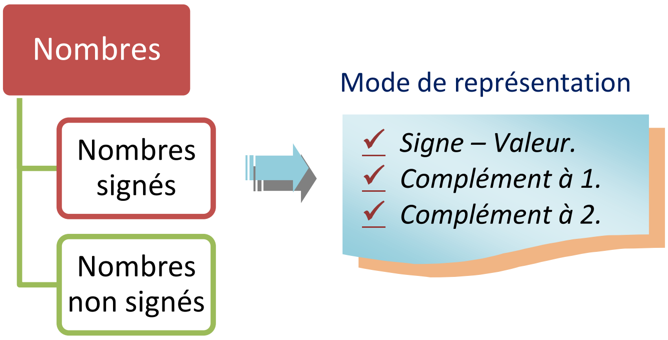
Pour la conversion **Décimal → Binaire**, deux cas qui se présentent :

1. Conversion Décimal → Octal, puis regrouper par blocs de trois bits.
2. Conversion Décimal → Héxadécimal, puis regrouper par blocs de quatre bits.

$$N = (11432)_{10} = (2CA8)_{16} = (0010\ 1100\ 1010\ 1000)_2$$
$$N = (175)_{10} = (AF)_{16} = (1010\ \ 1111)_2$$

2.4. Opérations arithmétiques en binaire

La plupart des calculateurs actuels utilisent le système binaire (base 2) pour représenter les informations. Les opérations entre nombres sont donc basées sur *l'arithmétique binaire*.

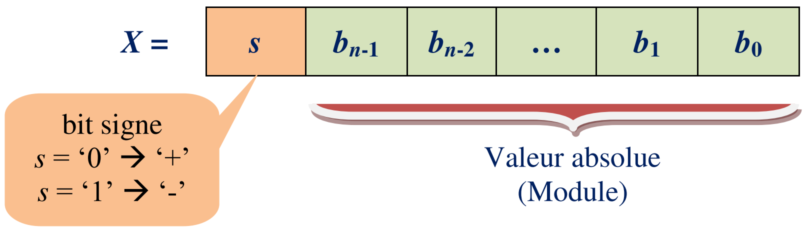


Le problème de la représentation des nombres signés (négatifs) pose de problèmes l'on veut manipuler des mots formés par une succession de bits à cause de l'absence des symboles **+/--**.

- *Comment indiquer à la machine qu'un nombre est négatif ou positif ?*
  - ✓ Il existe 3 méthodes pour représenter les nombres négatifs :
    - Signe - Valeur absolue.
    - Complément à 1 (complément restreint).
    - Complément à 2 (complément à vrai).
- La représentation des nombres entiers négatifs en complément à deux est la plus pratique et la plus utilisée dans le fonctionnement des microproceseurs.
- Pour les nombre de très grandes et/ou très petites valeurs (fractionnaires), la *représentation en virgule flottante* est adoptée.
- Dans les ordinateurs, l'information est structurée en mots (ensemble de bits). Lorqu'on manipule des données numériques, chaque mot comprend un digit (bit) de signe et un ensemble de bits dont le nombre est lié à l'architecture du microprocesseur (architecture sur 8 bits, 16 32, …etc).

2.4.1. Représentation des nombres négatifs

La représentation des nombres négatifs en complément à deux consiste à utiliser un bit supplémentaire (noté s) pour le signe pour un total de *n*+1 bits à gauche de la virgule et *m* bits à droite :



Le bit signe des nombres négatifs dans la représentation binaire signée prend la valeur 1. Par contre, les nombres positifs sont représentés en binaire signé en posant le bit de signe à 0 et en gardant la représentation non signée.

- La represénation des nombres dépend de la taille du registre de travail.
- Si on travail sur ***n*** bits, alors le bit du poids fort est utilisé pour indiquer le signe :
  - ☒ 1 : Signe négatif.
  - ☒ 0 : Signe positif.

Les autres bits (***n* -1**) désignent la valeur absolue (module) du nombre.

☒ Comment écrit-on +15 en binaire signé sur 06 bits (5 bits + 1 bit de signe) ?

$$15_{10} = (\mathbf{0}\ \mathbf{01111})_2$$

☒ • Comment écrit-on +35 en binaire signé (6 bits + 1 bit de signe) ?

$$35_{10} = (\mathbf{0}\ \mathbf{100011})_2$$

☒ Certaines valeurs sont impossible à représenter en binaire signé, selon la taille de registe où l'information est stockée. Par exemple :

- +15 en binaire signé avec 3 bits + 1 bit de signe → Impossible.
- •–15 en binaire signé avec 3 bits + 1 bit de signe → Impossible.
- +35 en binaire signé avec 4 bits + 1 bit de signe → Impossible.
- •–35 en binaire signé avec 4 bits + 1 bit de signe → Impossible.

Si on travail sur ***n*** bits , l'intervalle des valeurs qu'on peut représenter en Signe/Valeur absolue :

$$-(2^{(n-1)} -1) \leq N \leq +(2^{(n-1)} -1\ )$$

- Pour les opérations arithmétiques il nous faut deux circuits : L'un pour l'addition (Additionneur) et le deuxième pour la soustraction (Soustracteur).

⌖ L'idéal est d'utiliser un seul circuit pour faire les deux opérations, puisque

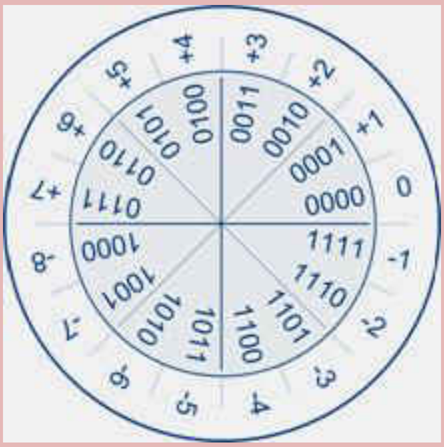
$$A - B = A + (-B)$$

Table 2.1 : *Opérations de complément à 1 et à 2.*

Opération	Description	Remarques
Complément à 1 (C1)	✓ Le complément à un (**N'**) d'un nombre (**N**) est donné par la formule : $\mathbf{N+N'=2^{n}-1}$ n : est le nombre de bits de la représentation du nombre (**N**). **Exemple :** Soit N = $(\mathbf{1010})_2$ sur 4 bits. Le complément à un de N est : N'= $(2^4 - 1)$-N N'= (16-1) - $(1010)_2$ = (15) - $(1010)_2$ = $(1111)_2 - (1010)_2 = (\mathbf{0101})_2$ ✓ En binaire, on soustrayant de 1 chaque bit du nombre, on obtient le C1 de ce nombre.	✓ Pour obtenir le C1 d'un nombre binaire, il suffit de compléter chaque bit, c-à-d, inverser tous les bits de ce nombre (0→1 et 1→0). ✓ La somme d'un nombre binaire et son complément est un nombre binaire composé uniquement de 1. ✓ Dans C1, le bit du poids fort nous indique le signe (0: positif, 1: négatif). ✓ Le C1 du C1 d'un nombre est égale au nombre lui même. **C1(C1(N)) = N**
Complément à 2 (C2)	✓ Si on prend deux nombres entiers **A** et **B** sur n bits, on remarque que la soustraction peut être ramener à une addition : **A - B = A + (-B)** ✓ Pour cela il suffit de trouver une valeur équivalente à **-B** **?** La valeur CA1(B)+1 s'appelle le complément à deux de B : **C1(B)+1 = C2(B)** **Exemple :** Soit N = $(01000101)_2$, Trouver le C2 de N sur 8 bits ? CA2(N) = CA1(N)+ 1 = (10111010) + 1 = (10111011)	✓ Soit $S = A - B$ $= A + 2^n - B$ $= A + (2^n - 1) - B + 1$ On a $B + C1(B) = 2^n - 1$ donc $C1(B) = (2^n - 1) - B$ Si on remplace dans la première équation on obtient : **A – B = A + C1(B) + 1** → **Transformation la soustraction en une addition.** ✓ Le C2 du C2 d'un nombre est égal au nombre lui-même : **C2(C2(N)) = N**
	☒ Si la taille du resgistre de travail est de ***n*** bits, l'intervalle des valeurs qu'on peut représenter en C2 est : $\mathbf{-(2^{(n-1)}) \leq N \leq +(2^{(n-1)} - 1)}$ ☒ Pour trouver le C2 d'un nombre, *Il faut parcourir les bits de ce nombre à partir du poids faible et* ***garder*** *tous les bits avant le premier 1 et* ***inverser*** *les autres bits qui viennent après.*	

N =	1	0	1	0	1	1	0	0
C2(N) =	**0**	**1**	**0**	**1**	**0**	1	0	0

☒ Il faut noter que l'écriture en C2 donne un **dénombrement cyclique**.



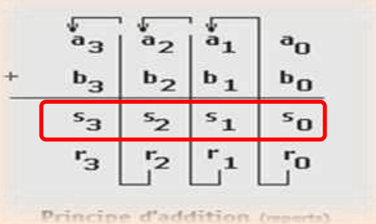
☒ Si la taille du resgistre de travail est de ***n*** bits, l'intervalle des valeurs qu'on peut représenter en C2 est :

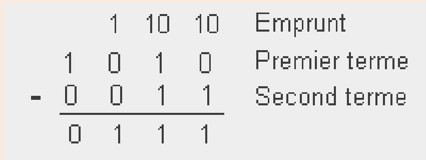
$$-(2^{(n-1)}) \leq N \leq +(2^{(n-1)} -1)$$

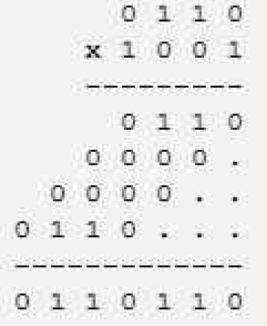
2.4.2. Opérations usuelles

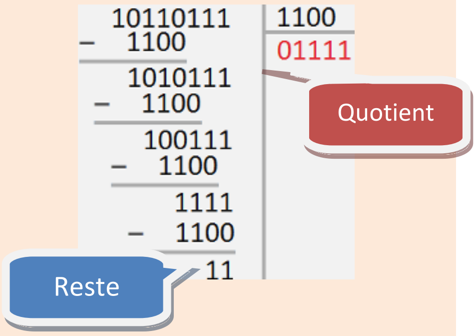
Les diverses opérations arithmétiques qui interviennent dans les machines numériques (ordinateurs et les calculatrices) portent sur des nombres exprimés en notation binaire. Le tableau ci-dessous illustre les opérations de base, à savoir l'addition, la soustraction, la multiplication et la division.

Table 2.2 : *Opérations arithmétiques de base.*

<table>
<tr><th>Opération</th><th>Description</th><th>Exemples/Remarques</th></tr>
<tr><td>Addition</td><td>
<table>
<tr><th><i>a</i></th><th><i>b</i></th><th><i>S=a+b</i></th><th><i>Report</i></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>
</td><td>


</td></tr>
<tr><td>Soustraction</td><td>
<table>
<tr><th><i>a</i></th><th><i>b</i></th><th><i>S=a-b</i></th><th><i>Retenue</i></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td></tr>
</table>
</td><td>

<table>
<tr><td></td><td></td><td>1</td><td>10</td><td>10</td><td>Emprunt</td></tr>
<tr><td></td><td>1</td><td>0</td><td>1</td><td>0</td><td>Premier terme</td></tr>
<tr><td>-</td><td>0</td><td>0</td><td>1</td><td>1</td><td>Second terme</td></tr>
<tr><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td>Multiplication</td>
<td>
<table>
<tr><th><i>a</i></th><th><i>b</i></th><th><i>S=a*b</i></th><th><i>Report</i></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
<td>


```
      0 1 1 0
    x 1 0 0 1
    ---------
      0 1 1 0
    0 0 0 0 .
  0 0 0 0 . .
0 1 1 0 . . .
-------------
0 1 1 0 1 1 0
```
</td>
</tr>
<tr>
<td>Division</td>
<td>
✓ La division d’un nombre binaire (le devidende) par un autre (le diviseur) est identique à la division de deux nombres décimaux.
✓ La division binaire s'effectue à l'aide de soustractions et de décalages, comme la division décimale, sauf que les digits du quotient ne peuvent être que 1 ou 0.
✓ Le bit du quotient est 1 si on peut soustraire le diviseur, sinon il est égal à 0.
</td>
<td>


</td>
</tr>
<tr>
<td></td>
<td colspan="2">✓ <i>L’addition est la base de toutes les opérations arithmétiques. Les opérations de soustraction, de multiplication et de division effectuées par les machines calculatrices ne sont que des variantes de l’opération d’addition.</i></td>
</tr>
</table>

2.4.3. Opérations arithmétiques en complément à 1 et à 2

La C1 et le C2 sont deux opérations importantes puiqu’elles permettent la représentation des nombres binaires négatifs. Les ordinateurs exploitent couramment la C2 dans le traitement des nombres négatifs.

Effectuer les opérations suivantes sur 8 bits (bit signe + 07 bits pou le module), en utilisant la représentation en C1 et en C2. En déduire l’algorithme de calcul.

+	1	6
+	1	3

-	1	6
+	1	3

+	1	6
-	1	3

-	1	6
-	1	3

☒ **Représentation en C1 :** On considère les différentes combinaisons de signes :

+16	**0**	0	0	1	0	0	0	0
+13	**0**	0	0	0	1	1	0	1
+29	**0**	0	0	1	1	1	0	1

Ajouter le report 1 au résultat pour obtenir la valeur correcte

Report	1	1	1	1					
+16		0	0	0	1	0	0	0	0
-13		1	1	1	1	0	0	1	0	C1(13)
		0	0	0	0	0	0	1	0
									1
+3		0	0	0	0	0	0	1	1

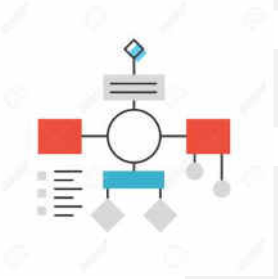
Report				1	1	1	1		
-16	1	1	1	0	1	1	1	1	C1(16)
+13	0	0	0	0	1	1	0	1	
	1	1	1	1	1	1	0	0	C1(module)
-3	1	0	0	0	0	0	1	1	

Bit signe = 1, on fait le C1 du résultat

Ajouter le report 1 au résultat pour obtenir la valeur correcte

Report	1	1	1	1	1	1	1			
-16		1	1	1	0	1	1	1	1	C1(16)
-13		1	1	1	1	0	0	1	0	C1(13)
		1	1	1	0	0	0	0	1	
									1	
-29		1	1	1	0	0	0	1	0	

A travers les exemples précédents, on peut déduire l'algorithme de l'opération de soustraction par utilisation de C1.



```
Debut
    LIRE (A) ;                    /* Lecture du premier opérande en C1 is < 0 */
    LIRE (B) ;                    /* Lecture du deuxième opérande en C1 si < 0 */
        R ← A + B ;                   /* Faire l'addition des deux opérandes */
    SI (Report le plus élevé = 1) Alors
        R ← R + 1 ;
        SINON
            SI (bit signe du résultat = 1) Alors
                R ← C1(|R|) ;         /* |R| est le module (valeur absolue) de R */
                SINON
                    R ← R ;
            FinSI ;
    FinSI ;
Fin.
```

☒ **Représentation en C2 :** On considère les différentes combinaisons de signes :

+16	0	0	0	1	0	0	0	0
+13	0	0	0	0	1	1	0	1
+29	0	0	0	1	1	1	0	1

Le report le plus élevé = 1 et bit signe = 1 → Débordement à éliminer et le résultat est correcte

Report	~~1~~	1	1	1						
+16		0	0	0	1	0	0	0	0	
-13		1	1	1	1	0	0	1	1	C2(13)
		0	0	0	0	0	0	1	1	
+3		0	0	0	0	0	0	1	1	

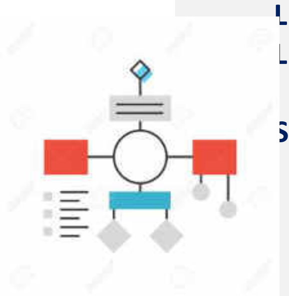
Report									
-16	1	1	1	1	0	0	0	0	C2(16)
+13	0	0	0	0	1	1	0	1	
	1	1	1	1	1	1	0	1	C2(module)
-3	1	0	0	0	0	0	1	1	

Bit signe = 1, on fait le C2 du résultat

Ajouter le report 1 au résultat pour obtenir la valeur correcte

Report	~~1~~	1	1	1						
-16		1	1	1	1	0	0	0	0	C2(16)
-13		1	1	1	1	0	0	1	1	C2(13)
		1	1	1	0	0	0	1	1	C2(module)
-29		1	0	0	1	1	1	0	1	

✍ A travers les exemples précédents, on peut déduire l'algorithme de l'opération de soustraction par utilisation de C2.



```
Debut
        LIRE (A) ;                    /* Lecture du premier opérande en C2 si < 0 */
        LIRE (B) ;                    /* Lecture du deuxième opérande en C2 sin <0 */
                R ← A + B ;                   /* Faire l'addition des deux opérandes */
        SI (Report le plus élevé = 1) Alors
                R ← R ;                       /* débordement à annuler */
                SINON
                        SI (bit signe du résultat = 1) Alors
                                R ← C2(|R|) ;         /* |R| est le module (valeur absolue) de R */
                                SINON
                                        R ← R ;
                        FinSI ;
        FinSI ;
Fin.
```

☒ On dit qu'il y a une **retenue** si une opération arithmétique génère un report.

☒ On dit qu'il y a un **débordement** (Over Flow) ou dépassement de capacité: si le résultat de l'opération sur *n* bits et faux .

- ✓ Le nombre de bits utilisés est insuffisant pour contenir le résultat.
- ✓ Autrement dit, le résultat dépasse l'intervalle des valeurs sur les *n* bits utilisés.

☒ La représentation en C2 est la représentation la plus utilisée pour la représentation des nombres négatifs dans les machines calculatrices, à cause de sa simplicité comparablement à la représentation en C1. Cette simplicité est mesurée par le nombre des opérations élémentaires à effetuées.

2.5. Représentation des nombres réels

Un nombre réel est constitué de deux parties entière et fractionnaire séparées par une virgule.

Signe	Partie réelle	Virgule	Partie fractionnaire
±	325	.	1258

Problème : Comment indiquer à la machine la position de la virgule ?

☒ Il existe deux méthodes pour représenter les nombres réels :

- ✓ **Virgule fixe** ou la position de la virgule est fixe.
- ✓ **Virgule flottante** ou la position de la virgule dynamique ou variable.

Table 2.1 : *Représentation des nombres réels.*

<table>
<tr><th>Position</th><th>Description</th></tr>
<tr><td>Virgule fixe</td><td>
✓ Dans cette représentation la partie entière est représentée sur <i>n</i> bits et la partie fractionnelle sur <i>p</i> bits, en plus un bit signe.
<u>Exemple</u> : si <i>n</i> = 3 et <i>p</i> = 2 on peut avoir les valeurs indiquer dans le tableau suivant :
<table>
<tr><th><i>Signe</i></th><th><i>Partie entière</i></th><th><i>Partie fractionnaire</i></th><th><i>Valeur</i></th></tr>
<tr><td>0</td><td>000</td><td>00</td><td>+ 0.0</td></tr>
<tr><td>0</td><td>000</td><td>01</td><td>+ 0.25</td></tr>
<tr><td>0</td><td>000</td><td>10</td><td>+ 0.5</td></tr>
<tr><td>0</td><td>000</td><td>11</td><td>+ 0.75</td></tr>
<tr><td>0</td><td>001</td><td>00</td><td>+ 1.0</td></tr>
<tr><td>0</td><td>001</td><td>01</td><td>+ 1.25</td></tr>
<tr><td>0</td><td>001</td><td>10</td><td>+ 1.5</td></tr>
<tr><td>0</td><td>001</td><td>11</td><td>+ 1.75</td></tr>
<tr><td>…</td><td></td><td>…</td><td>…</td></tr>
<tr><td>0</td><td>111</td><td>00</td><td>+ 7.0</td></tr>
<tr><td>0</td><td>111</td><td>01</td><td>+ 7.25</td></tr>
<tr><td>0</td><td>111</td><td>10</td><td>+ 7.5</td></tr>
<tr><td>0</td><td>111</td><td>11</td><td>+ 7.75</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>000</td><td>00</td><td>- 1.0</td></tr>
<tr><td>1</td><td>000</td><td>01</td><td>- 1.25</td></tr>
<tr><td>1</td><td>000</td><td>10</td><td>- 1.5</td></tr>
<tr><td>1</td><td>000</td><td>11</td><td>- 1.75</td></tr>
<tr><td>…</td><td></td><td>…</td><td>…</td></tr>
<tr><td>1</td><td>111</td><td>00</td><td>- 7.0</td></tr>
<tr><td>1</td><td>111</td><td>01</td><td>- 7.25</td></tr>
<tr><td>1</td><td>111</td><td>10</td><td>- 7.5</td></tr>
<tr><td>1</td><td>111</td><td>11</td><td>- 7.75</td></tr>
</table>
✓ <i>Dans cette représentation les valeurs sont limitées et leurs précisions s réduites (représentation moins précise).</i> ⚠
</td></tr>
</table>

✓ Dans cette représentation, chaque nombre réel peut être formulé de la manière suivante :

$$N = \pm M * b^{exp}$$

- ✓ **M** : La mantisse ,
- ✓ **b** : La base ,
- ✓ **exp** : L'exposant

Signe Mantisse	Exposant	Mantisse normalisée
1 bit	*m* bits	*p* bits

☒ La mantisse est sous la forme signe/valeur absolue (module)

- ✓ 1 bit pour le signe et
- ✓ *p* bits pour la valeur absolue.

☒ L'exposant (positif ou négatif) est représenté sur *m* bits.

Exemple : Soit

- $21{,}5 = 0{,}215 \ * 10^{+2}$
- $(110{,}101)_2 = -(0{,}110101)_2 * 2^{+3}$
- $(0{,}00101)_2 = (0{,}101)_2 * 2^{-2}$

On dit que la mantisse est normalisée si le premier chiffre après la est différent de 0 et le premier chiffre avant la virgule est égale à 0.

✓ Pour la représentation de l'exposant on utilise :

- ✓ Le complément à deux (C2),
- ✓ L'Exposant décalé (biaisé).

Représentation de l'exposant en complément à deux

☒ **Exemple** : On veut représenter les nombres $(0.015)_8$ et $-(15.01)_8$ en virgule flottante sur une machine ayant le format suivant :

Signe Mantisse	Exposant en C2	Mantisse normalisée
1 bit	4 bits	8 bits

- $(0.015)_8 = (0.000001101)_2 = 0.1101 * 2^{-5}$.
- Signe mantisse : Positif (0).
- Mantisse normalisé : 0.1101
- Exposant = -5 → Utiliser le complément à deux pour représenter l'exposant -5.

- 5	1	0	1	1	C2(0101)

0	1011	11010000
1 bit	4 bits	8 bits

$-(15{,}01)_8 = -(001101{,}000001)_2$

$= -\ 0{,}1101000001 * 2^4$

- Signe mantisse : négatif (1)
- Mantisse normalisée : 0,1101000001
- Exposant = 4, en C2 il garde la même valeur (0100)

On remarque que la mantisse est sur 10 bits (1101 0000 **01**), et sur la machine seulement 8 bits sont utilisés pour la mantisse.

→ Dans ce cas on va prendre les 8 premiers bits de la mantisse.

1	0100	11010000
1 bit	4 bits	8 bits

2.5.1. Opérations arithmétiques en virgule flottante

Soit deux nombres réels N1 et N2 tel que

$$N1=M1*b^{exp1} \quad \text{et} \quad N2=M2*b^{exp2}$$

☒ On veut calculer N1+N2 ?

Deux cas se présentent :

- ✓ Si exp1 = exp2 = exp, alors N3= (M1+M2) b^{exp}
- ✓ Si exp1 < exp2 ou exp1 > exp2, alors élever au plus grand exposant et faire l'addition des mantisses et, par la suite, normalisér la mantisse du résultat.

Exemple : Effectuer l'opération suivante : $(0,15)_8+(1,5)_8=(?)$:

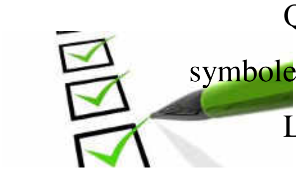
- $(0,15)_8 = (0,001101) = 0,1101 \ *2^{-2}$
- $(1,5)_8 = (001, 1 \ 01) = 0,1101 \ *2^{1}$
- $(0,15)_8+(1,5)_8 = 0,1101 \ *2^{-2} + 0,1101 \ *2^{1}$

  $= 0,0001101 \ *2^{1} + 0,1101 \ *2^{1}$

  $= 0,\ 1110101 * 2^{1}$

0	0001	111010
1 bit	4 bits	6 bits

2.6. Les codes non pondérés



Quand les informations (nombres, lettres, mots, …etc) sont représentées par un groupe de symboles, cette opération s'appelle Codage (**Encoder**). Le gropue de symboles s'appel **Code**.

La création des codes répond à plusieurs impératifs techniques ayant relation directe à la sécurité et la fiabilité des informations à taiter par les machines calaculteurs. Parmi ces impératifs, on peut citer :

- ☒ Facilité de traitement des informations.
- ☒ Fiabilité de materiels électroniques de traitement de ces informations dans certaines situations.
- ☒ Protection des informations contre les aléas (états parasites) lors de la transmission.

Les codes peuvent être classés en deux catégories qui sont :

- ✓ Les codes pondérés.
- ✓ Les codes non pondérés.

Table 2.1 : *Classement des codes.*

Codes		Remarques
Codes pondérés		
Code binaire	{0,1}	✓ Un nombre est constitué de plusieurs chiffres. ✓ Chaque chiffre est affecté d'un poids.
Code décimal	{0,1, …, 9}	
Code héxadécimal	{0,1, …, 9, A, …, F}	
Code BCD (DCB)	Binary Coded Decimal (Décimal Codé Binaire	
Codes non pondérés		
Code Gray	Binaire réflichi	✓ Les chiffres d'un nombre exprimé dans un code non pondéré ne peuvent pas être affectés d'un poids. ✓ Chaque code non pondéré est établi pour une application. ✓ Les codes non pondérés, ne sont pas adaptés pour le calcul numérique comparablement aux codes pondérés.
Code Excess 3 (XS 3)	Code STIBITZ	
Code Hamming		
Code AIKEN		
Code ISO		
Code EBCDIC		
Code ASCII		
Code AIKEN		
Codes 2 parmi 5		

2.6.1. Code Gray

Tout comme le binaire naturel, le binaire réfléchi peut coder n'importe quel nombre entier naturel.

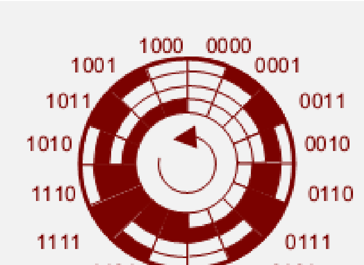
- ✓ La propriété principale de ce code est que le passage d'une combinaison vers une autre n'engendre le changement d'état que d'une seule variable.
- ✓ Les transitions s'effectuent sans ambiguïté, éliminant les risques d'aléas.
- ✓ Cyclique pour un nombre de mot-code égal à une puissance de 2.

» En savoir +

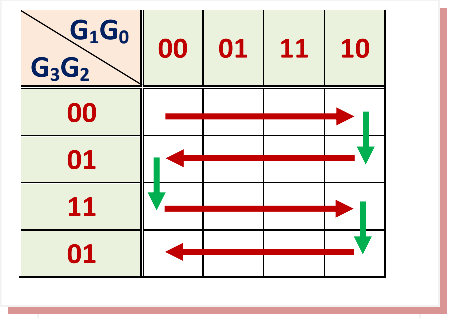
- ⌖ Le code Gray est le code utilisé dans les tableaux de Karnaugh, afin de coder la valeur des différentes entrées de telle sorte qu'un chagement à la fois aura lieur lors de passage d'un état à un autre.
- ⌖ Ce code n'est pas utilisé pour des calculs numériques, mais seulement pour les transferts des informations.

Table 2.4 : *Equivalences décimal – Binaire – Gray.*

Code décimal	Binaire pur $b_3b_2b_1b_0$	Code Gray $G_3G_2G_1G_0$
0	0000	0000
1	0001	0001
2	0010	0011



3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1101
9	1001	1111
10	1010	1110
11	1011	1010
12	1100	1011
13	1101	1001
14	1110	1100
15	1111	1000



Tableaux de Karnaugh

2.6.2. Code BCD

- ✓ Le code BCD est un code qui permet d'exprimer les dix chiffres du système décimal à l'aide du code binaire naturel.
- ✓ Il conserve les avantages du code décimal et du code binaire.
- ✓ Ce code est utilisé pour pouvoir réaliser une visualisation de résultats dans le système décimal, alors que la machine travaille dans le système binaire.

Table 2.4 : *Code BCD.*

Code décimal	Binaire BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

10	
11	
…	

Tétrades (**0, 1, 2, …, 8, 9**)
Pseudo-tétrades (**A, B, C, D, E, F**)

En savoir +

- On peut remarquer que les six dernières combinaisons du code binaire (A, B, …, F) n'apparaissent pas dans le tableau.
- Ces combinaisons (Pseudo-tétrades) n'existent pas pour le code BCD.
- Pour retrouver un chiffre décimal à partir de son mot-code en BCD il suffit d'effectuer une conversion binaire → décimal pour chacune des tétrades (0, 1, 2, …, 8, 9) composant le code BCD.

Exemple de codage en BCD

$(137)_{10} = (\ ?)_2 = (\ ?)_{BCD}$

1				3				7				
0	0	0	1	0	0	1	1	0	1	1	1	BCD
0	0	0	1	1	0	0	0	1	0	0	1	Binaire

2.6.3. Code EXCESS 3

- ✓ Le code EXCESS 3 ou code plus 3 ou excédent 3 est formé de la même manière que le code 8421 tout en ajoute systèmatiquement 3 à chaque chiffre.
- ✓ La table de conversion du code EXCESS3 ne concerne que les chiffres 0 à 9.
- ✓ Comme en BCD, pour coder un nombre en code EXCESS 3, il faut concaténer une succession de tétrades, traduisant chacune un chiffre du nombre à coder.
- ✓ Le code EXCESS 3 a été créé pour permettre la réalisation simple des opérations de soustraction.
- ✓ Le complément à 1 d'un nombre représente le complément à 9 dans l'ensemble source. Donc, les codes possédant cette propriété sont appelés des codes ***auto-complémentaires***.

Table 2.4 : *Code EXCESS 3.*

Code décimal	Binaire EXCESS 3
0	0011
1	0100
2	0101
3	0110

4	0111
5	1000
6	1001
7	1010
8	1011
9	1100
10	
11	
…	

Le complément à 1 d'un nombre en EXCESS 3 correspond au complément à 9 du chiffre en décimal.

2.6.4. Code AIKEN

- ✓ Dans le code AIKEN, les poids des éléments binaires sont **2 4 2 1** (**8 4 2 1** en BCD).
- ✓ AIKEN est un code auto-compléméntaire comme les codes EXCESS 3 et BCD.

Table 2.4 : *Code AIKEN.*

Code décimal	Binaire AIKEN
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111
10	
11	
…	

Le complément à 1 d'un nombre en code AIKEN correspond au complément à 9 du chiffre en décimal.

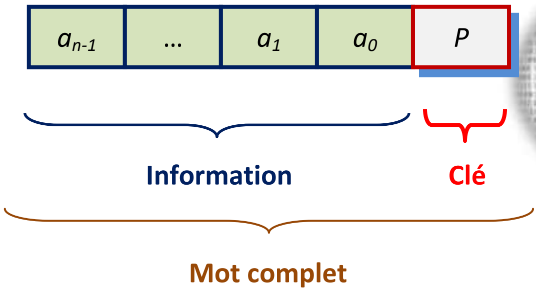
<u>Exemple</u> de codage en code AIKEN

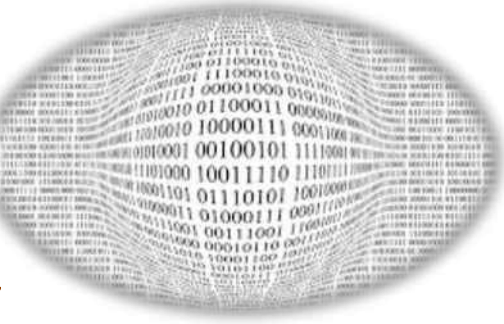
$(198)_{10} = (\ ?)_2 = (\ ?)_{BCD}$

7				8				9				
1	1	0	1	1	1	1	0	1	1	1	1	AIKEN
0	1	1	1	1	0	0	0	1	0	0	1	Binaire

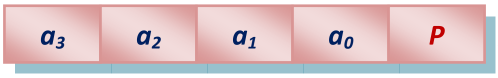
2.6.5. Code binaire avec bit de parité

- ✓ L’apparition des erreurs lors de la transmission ou de traitement des données numériques est pratiquement inévitable. Les causes de ces erreurs sont :
  - Transmission (changement 1→0 ou 0→1 du aux perturbations momentannées liés aux parasites).
  - Traitement.
  - Pannes d’éléments internes ou externes d’un ordinateur.
- ✓ Le principe consiste à ajouter un bit à chaque combinaison du code de telle sorte que le nombre de ‘**1**’ contenu dans chaque combinaison soit **paire** ou **impaire** suivant une convention entre émetteur et récepteur (parité paire ou impaire).
- ✓ Ce mode de coge est largement utilisé dans la transmission des données numériques.
- ✓ Les bits de parité ajoutés aux codes binaires sont destinés à accroître la fiabilité des systèmes. Ce bit permet la détection des erreurs lors de la transmission.
- ✓ Le récepteur recevant une combinaison binaire vérifie que le nombre de 1 est conforme à la convention.
- ✓ Lorsque la vérification est faite, le récepteur élimine le bit de parité dont l'usage est limité à la transmission de l'information et sa vérification.
- ✓ Cette technique de codage permet de detecter des erreurs (code **auto-vérificateur**) et de corriger leurs effets (code **auto-correcteurs**).





Exemple :

Soit le mot  , où P est le bit de parité impaire, telle que :

$$\begin{cases} P = 0 : \text{Parité paire} \\ P = 1 : \text{Parité impaire} \end{cases}$$

Table 2.4 : *Code binaire avec bit de parité.*

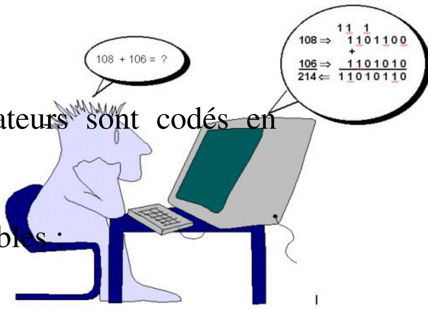
Binaire naturel				
a_3	a_2	a_1	a_0	P
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**1**
0	0	1	1	**0**
0	1	0	0	**1**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**0**

Bit de parité
(Parité impaire)

Il est possible d'ajouter un bit de parité à tous codes et de faciliter la vérification des erreurs.

2.6.6. Code ASCII

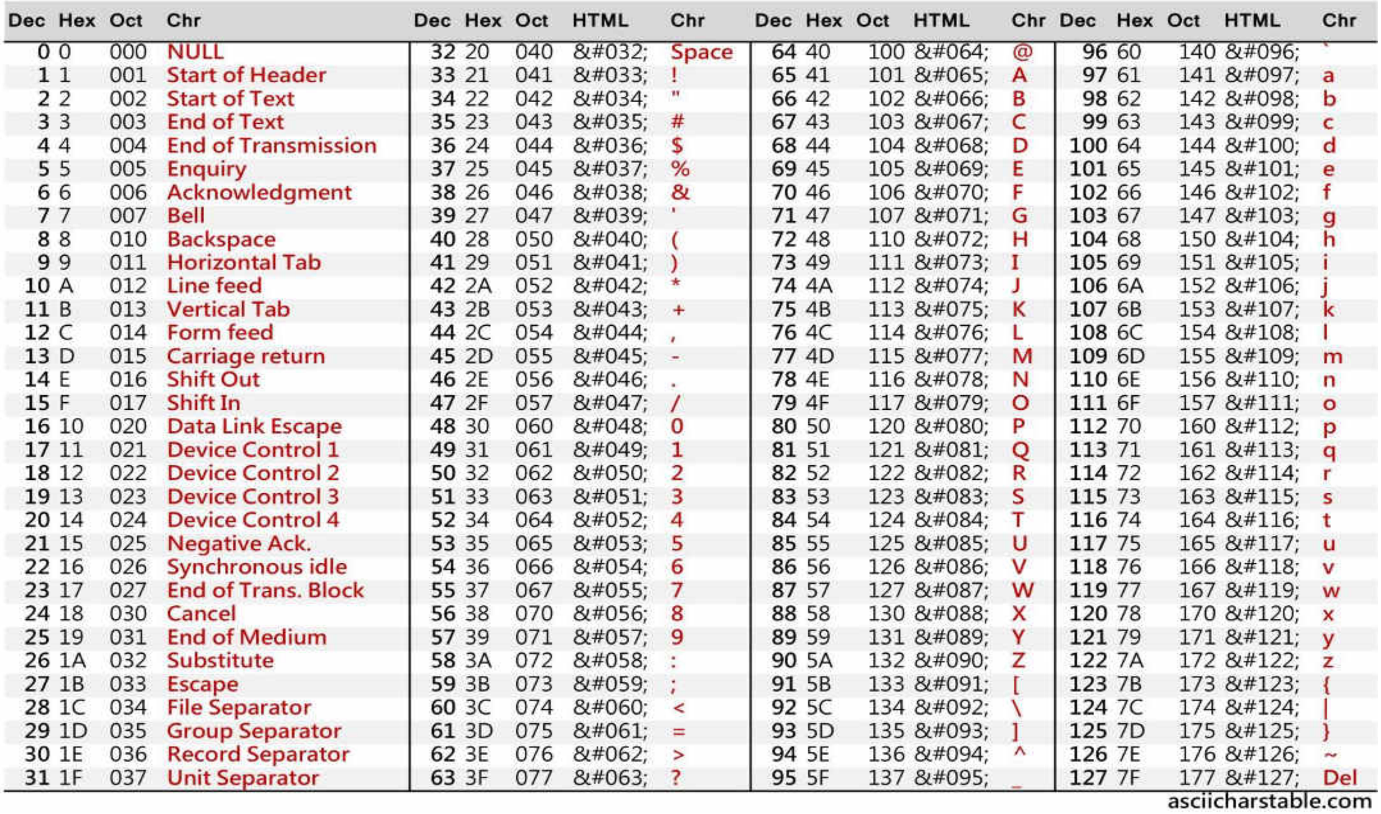
- ✓ ASCII abréviation de l'anglais '*American Standard Code for Information Interchange*'.
- ✓ Ce code est une norme universelle dans l'échange de données informatique :
  - Caractères alphabétiques et numériques (alphanumérique),
  - Ponctuation,
  - Codes de contrôles divers.
- ✓ Toutes les touches des claviers ***ALPHANUMERIQUES*** des ordinateurs sont codés en ASCII.
- ✓ ASCII définit **128** codes à **7** bits, comprenant **95** caractères imprimables :
  - Les chiffres arabes de 0 à 9,
  - Les lettres minuscules et capitales de A à Z, et
  - Des symboles mathématiques et de ponctuation.



- ✓ ASCII suffit pour représenter les textes en **anglais**, mais il est trop limité pour les autres langues, dont le **français** et ses lettres accentuées.
- ✓ Différentes variantes du code ASCII sont disponibles pour différentes langues. Il existe même une version étendu 'Extended' du code ASCII où le 8$^{\text{ème}}$ bit de données est utilisé,

ce qui permet de distinguer 2 fois plus de caractères (notamment les caractères accentués pour la langue française).

Table 2.4 : *Code ASCII.*



Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

asciicharstable.com

2.6.7. Codes particuliers

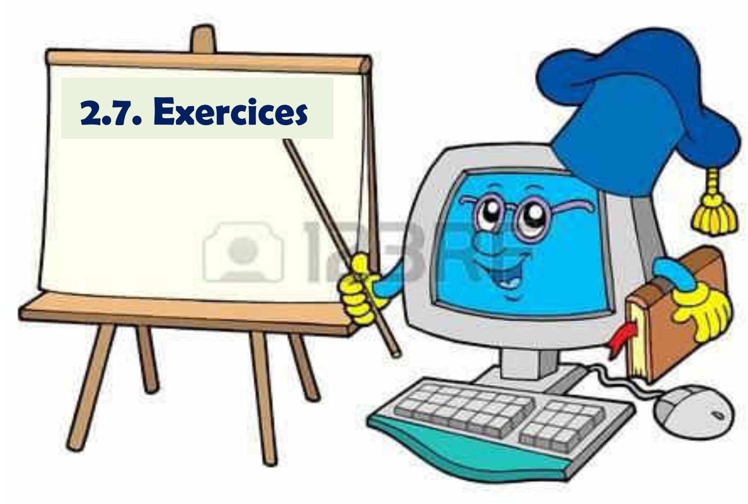
Différents codes sont utilisés dans des domaines un peu particulier à savoir la transmission des données numériques, l'identification des livres, compression des données, ..etc. Le tableau ci-dessous illustre quelques codes.

Table 2.4 : *Codes divers.*

Code	Description
Code Hamming  Richard **Hamming** (1915-1998)	✓ Ce code utilise les tests de parité. Ces test permettanat l'auto-correction. ✓ Ce code nécessite plusieurs bits de parité supplémentaires et is est utilisé, sourtout, pour la détection et la correction des erreurs lors de la transmission des données numériques.
Codage de Huffman	✓ Ce codage est une méthode de compression statistique de données permettant de reduire la longueur du codage d'un alphabet.

 David A. **Huffman** (1925-1999)	
Code ISBN 	✓ L'ISBN (*International Standard Book Number*) est un numéro international qui permet d'identifier, de manière unique, chaque livre publié. ✓ Il est destiné a simplifier la gestion informatique des livres dans les bibliothèques, librairies, etc.
Les codes-barres 	✓ Un code-barres est la représentation d'une donnée numérique ou alphanumérique sous forme d'un symbole constitue de barres et d'espaces dont l'epaisseur varie en fonction des données ainsi codées. ✓ Lorsque ces barres sont remplacées par des petits carrés ou des points, on parle de code en deux dimensions. ✓ Il existe des milliers de différents codes-barres : ceux-ci sont destines a une lecture automatisée par un capteur electronique.
QR Codes	✓ Le code QR (ou ***QR code*** en anglais) est un code-barres en deux dimensions (ou code a matrice) constitue de modules noirs disposés dans un carré a fond blanc. ✓ Le nom *QR* est l'acronyme de l'anglais *Quick Response*, car son contenu de données peut etre décodé rapidement.

Différents formats de code: *Code QR, Datamatrix, Shotcode, Colorzip, Maxicode.*



2.7. Exercices

EXERCICE 01 :

- ✓ Donner la forme polynomiale des nombres suivants :

$(EA735)_{16}$; $(B9107A)_{12}$; $(654321)_7$; $(11100011)_2$; $(3476)_8$; $(254)_6$

- ✓ Déterminer le chiffre de poids faible et le chiffre de poids fort des nombres suivants :

7432 ; 3890 ; 170003 ;

- ✓ Soit le nombre 49736421, déterminer le rang de 9 ; 2 ; 7 ; 1 et 4.

EXERCICE 02 :

Donnez la valeur décimale, hexadécimale et octale des nombres binaires suivants:

a) 00000000 01101101 **b)** 00000000 11110010 **c)** 00000000 00011101

d) 10010010 10010101 **e)** 11010111 01101011

EXERCICE 03 :

Donnez la valeur décimale, binaire et octale des nombres hexadécimaux suivants:

a) 0011 **b)** 00B7 **c)** 01FE **d)** 1234

EXERCICE 04 :

Donnez la valeur décimale, binaire et hexadécimale des nombres octaux suivants:

a) 000255 **b)**000047 **c)** 000506 **d)** 001234

EXERCICE 05 :

Convertissez en décimal les nombres binaires codés en complément à deux.

a) 0000 0000 0111 0111 **b)** 0000 0000 0110 1110 **c)** 1111 1111 1011 1101

d) 1111 1111 1111 1111 **e)** 1111 0110 0101 1001 **f)** 1111 1111 0000 0110

EXERCICE 06 :

Effectuer les opérations suivantes :

11101 + 10110 ; 110111 + 110101 ; 11101011101 + 101101011 ; 111101010 + 1110110

110 – 011 ; 100 – 010 ; 1101 – 1001 ; 1111 – 1011 ; 1101 – 0011

101001 * 1101 ; 11011 * 1001 ; 1000 * 101 ; 10 0101 * 10 ; 11010011 * 110

1001 ÷ 11 ; 1100 ÷ 10 ; 1111 ÷ 10

EXERCICE 07 :

Effectuez les opérations en hexadécimal (compl-2, 16 bits) et dites s'il y a débordement dans les cas suivants:

a) 50A3	**b)** 3826	**c)** 38A3	**d)** C839
+ 6A38	- 7000	+ A330	- 7000
-------	------	------	-------

EXERCICE 08 :

Effectuez les additions arithmétiques (compl-2, 16 bits).

a) 1F2E (base 16)	**b)** 100 (base 10)
-3FCA (base 16)	-177777 (base 8)
------	------
(base 16)	(base 16)

c) FF35 (base 16)	**d)** 1234 (base 16)	**e)** 7000 (base 16)
+1038 (base 16)	- 954 (base 10)	-93BC (base 16)
-001716 (base 8)	+004701(base 8)	-2000 (base 16)
------	------	----
(base 10)	(base 2)	(base 8)

EXERCICE 09 :

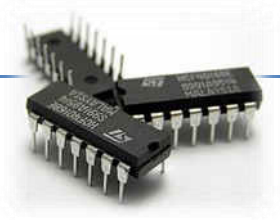
Parmi les nombres décimaux signes suivants, quels sont ceux représentables sur 8 bits ? sur 16 bits ? et sur 32 bits ?

a) 64	**b)** -56	**c)** 150	**d)** -5	**e)** -132
f) 1000	**g)** -4132	**h)**-16401	**i)**-42750	**j)** 59680

EXERCICE 10 :

Quelle chaîne de caractères correspond à la séquence hexadécimale suivante de codes ASCII (8 bits) ?

494E46323137302047726F757065203230



CHAPITRE 3

LES BASCULES

3.1. Les bascules

La bascule (Flip-Flop : FF) est un circuit bistable pouvant prendre deux états logiques ‘**0**’ ou ‘**1**’.

- ✓ L'état de la bascule peut être modifié en agissant sur une ou plusieurs entrées.
- ✓ Le nouvel état de la bascule dépend de l'état précédent, c'est l'élément de base des circuits séquentiels.
- ✓ La bascule peut conserver son état pendant une durée quelconque, elle peut donc être utilisée comme élément *mémoire*.

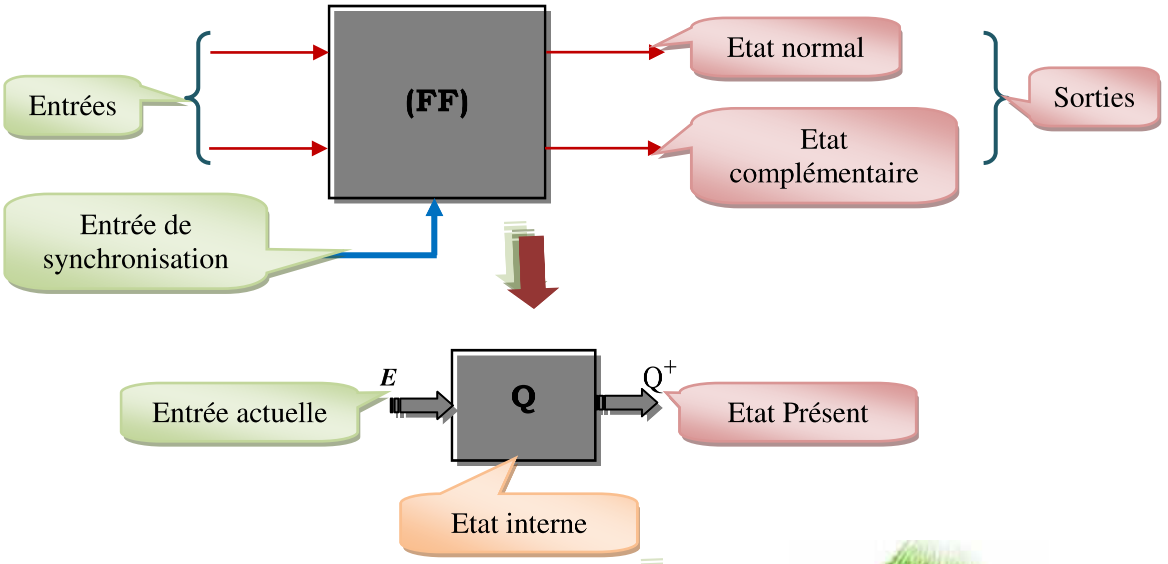


Figure 3.1 : *Structure externe d’une bascule.*

- ✓ 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 $\overline{\mathbf{Q}}$.
- ✓ La mémorisation fait appel à un verrou (latch) ou système de blocage, dont le principe de rétro-action peut être représenté par la suite.

En savoir +

- **R : RESET.**
- **S : SET.**
- **D : DATA.**
- **J : JOKER.**
- **K : KING.**
- **T : TOGGLE.**

Il existe deux types de bascules selon le nombre des entrées:

- Bascules à une entrée (**D** et **T**).
- Des bascules à deux entrées (**RS** et **JK**).

3.1.1. Bascules RS et $\overline{R}\overline{S}$

- ✓ La bascule RS est une bascule asynchrone (sans entrée d'horloge).
- ✓ C'est la bascule élémentaire, qui constitue la base de tous les autres types de bascules.
- ✓ La bascule RS peut être réalisée avec des portes OU-NON ou avec des portes ET-NON.

S : Set = Mise à **un.**
R : Reset = Remise à **zéro.**

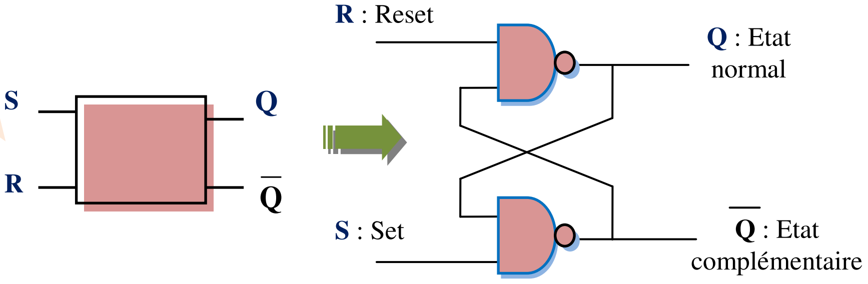


Figure 3.2 : *Bascule RS.*

Le tableau 3.1 résume les différents états possibles de la bascule RS.

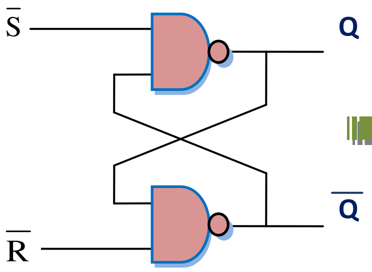
Table 3.1 : *Table des états et de transitions de la bascule RS.*

S	*R*	*Q*	Q^+	Signification
0	0	0	0	Pas de changement d'état
0	0	1	1	
0	1	0	0	Remise à '0'
0	1	1	0	
1	0	0	1	Mise à '1'
1	0	1	1	
1	1	0	x	**Ambiguïté**
1	1	1	x	

Table de transitions

Q	Q^+	*R*	*S*
0	0	x	0
0	1	0	1
1	0	1	0
1	1	0	x

Une autre version de la bascule RS et sa forme complémentaire : Bascule $\overline{R}\overline{S}$ dont son schéma et sa table de vérité sont données par la figure ci-dessous.



$\overline{S}$	$\overline{R}$	Etat de la bascule
0	0	Etat indéterminé
0	1	Mise à ‘1’ (Q = 1)
1	0	Remise à ‘0’ (Q = 0)
1	1	Pas de changement d’état

Figure 3.3 : *Bascule* $\overline{S}\overline{R}$.

3.1.1.1. Synchronisation de la bascule RS

- ✓ C'est une bascule RS dont la prise en compte de l'état des entrées est synchronisée par un *signal d'horloge* (**CLK**). Ceci permet d'éviter l'arrivée accidentelle de "zéro" sur R ou sur S.
- ✓ Lorsque **CLK = 0**, il y a mémorisation de l’état précédent.

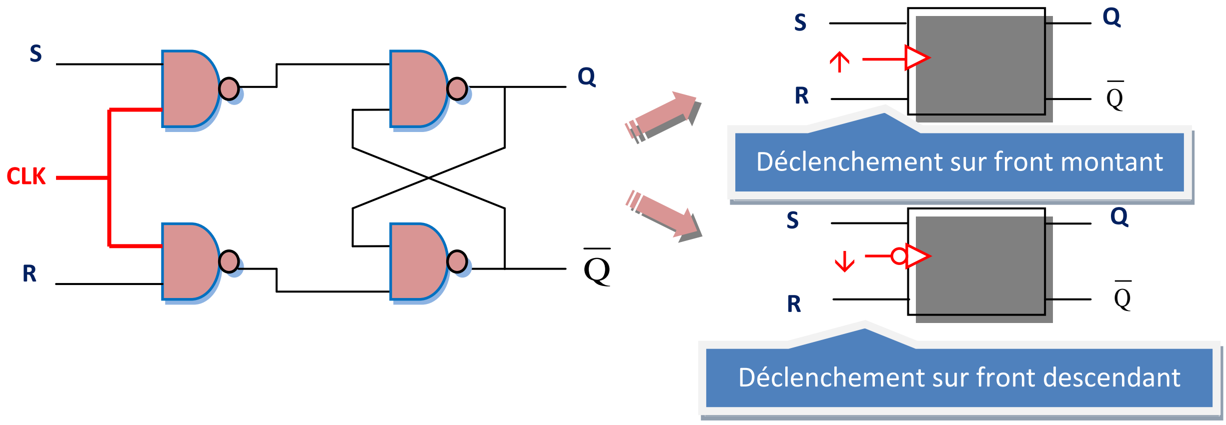


Figure 3.4 : *Bascule SR synchronisée par front d’horloge.*

Une bascule synchronisée peut être déclenchée sur le front montant ↑ ou sur le front descendant ↓ de l'impulsion d'horloge.

Exemple : La réponse temporelle d’une bascuke RS est illustrée par le diagramme temporel de la figure 3.5.

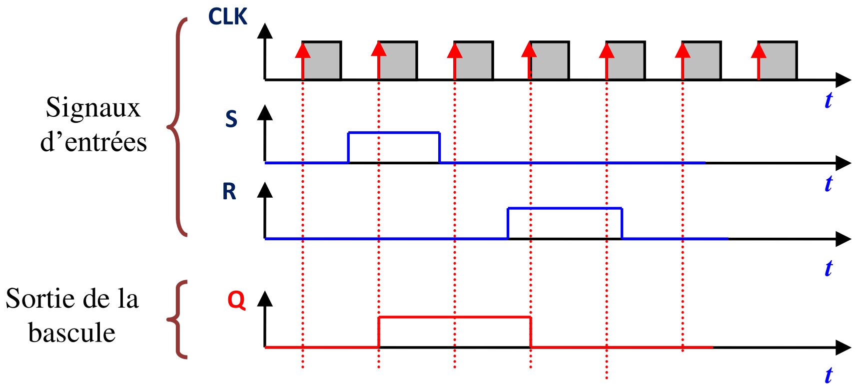


Figure 3.5 : *Diagramme temporel de la Bascule SR synchronisée par front montant.*

Références techniques : On peut retrouver la bascule RS dans les circuits intégrés ci-dessous :

- 74118, 74119, 74279
- 4044, …..

Exemple : Le circuit intégré **74279** comprenant **04** bascules du type $\bar{S}\bar{R}$.

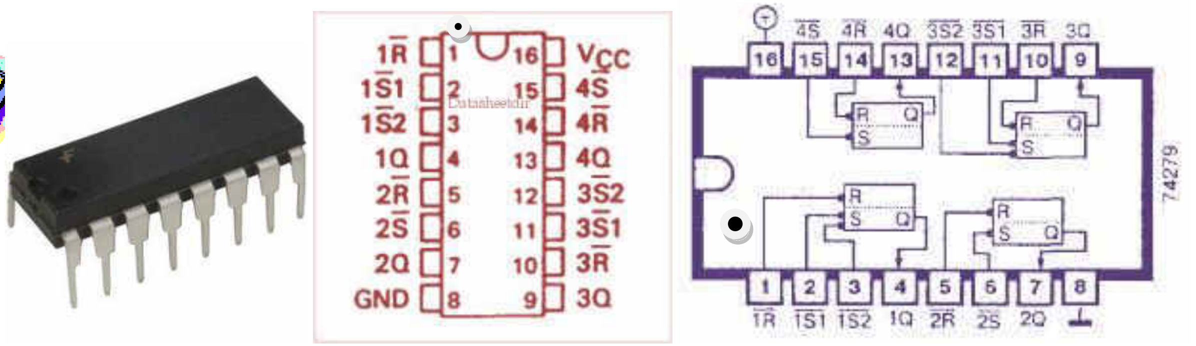


Figure 3.6 : *Brochage du circuit intégré **74279**.*

3.1.2. Bascule JK synchrone

- ✓ La bascule JK synchrone est obtenue à partir d'une bascule RS-CLK dont les sorties sont rebouclées sur les entrées.
- ✓ Ceci permet d'éliminer l'état d'indétermination **SR = 11**.

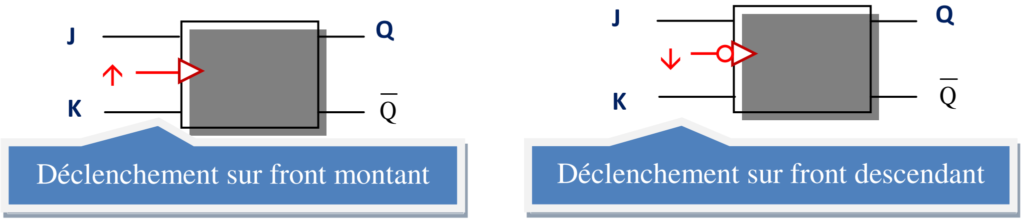


Table de vérité :

Figure 3.7 : *Bascule JK.*

Table 3.2 : *Table des états et de transitions de la bascule JK.*

J	*K*	*Q*	*Q^+*	*Signification*
0	0	0	0	Pas de changement d'état
0	0	1	1	
0	1	0	0	Remise à '0'
0	1	1	0	
1	0	0	1	Mise à '1'
1	0	1	1	
1	1	0	1	Basculement
1	1	1	0	

Table de transitions

Q	*Q^+*	*J*	*K*
0	**0**	**0**	**x**
0	**1**	**1**	**x**
1	**0**	**x**	**1**
1	**1**	**x**	**0**

- **Remarque:** Pour **J** = **K** = **1**, on dit que l'on est dans le mode basculement et l'on définit la bascule « **T** »(Toggle). Cette bascule passe à l'état opposé à chaque signal d'horloge.

- **Simplification par table de Karnaugh**

Q^+ JK / Q	00	01	11	10
0	0	0	1	1
1	1	0	0	1

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

$$= \overline{\overline{\overline{J\overline{Q}}\cdot\overline{\overline{K}Q}}} \quad \text{et}$$

$$\overline{\overline{K}Q} = \overline{\overline{K}Q + Q\overline{Q}} = \overline{Q\cdot\left(\overline{K}+\overline{Q}\right)} = \overline{Q\overline{KQ}},$$

$$\Rightarrow Q^+ = \overline{\overline{J\overline{Q}}\cdot\overline{Q\overline{KQ}}}$$

Logigramme

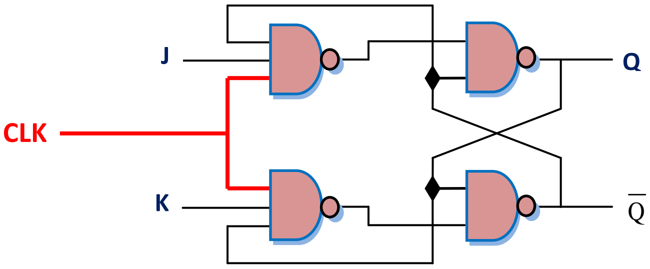


Figure 3.8 : Structure interne de la bascule JK.

- La réponse temporelle d'une bascuke JK est illustrée par le diagramme temporel de la figure 3.9.

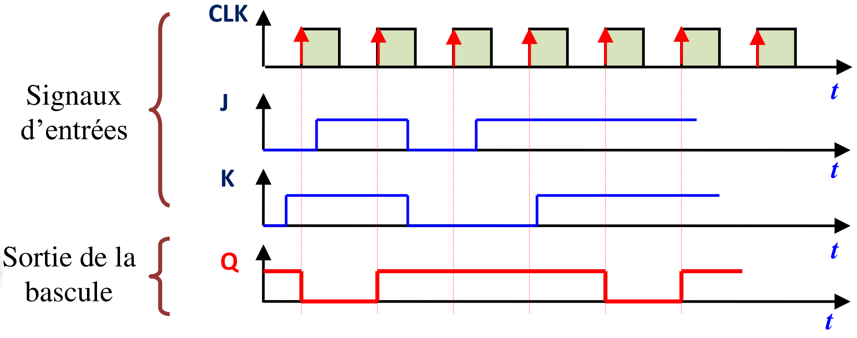


Figure 3.9 : Diagramme temporel de la Bascule JK synchronisée par front montant.

- Si CLK n'est pas sur un front actif, les sorties ne changent pas d'état.
- Si on applique un front montant sur CLK alors que J ≠ K, la sortie Q prend de la valeur de l'entrée J :
  - Si J = 0 → Remise à 0.
  - Si J = 1 → Mise à 1.
- Si on applique un front montant sur CLK alors que J = K = 0, les sorties de la bascule ne changent pas d'état, c'est la **mémorisation**.

Si on applique un front montant sur CLK alors que J = K =1, les sorties changent systématiquement d'état, on parle de **basculement.**

Références techniques : on peut retrouver la bascule JK dans les circuits intégrés ci-dessous :

- 7472, 7476, 74109, 74110, 74111
- 74LS112, …etc.

Exemple : Le circuit intégré **7476** comprenant **02** bascules du type JK .

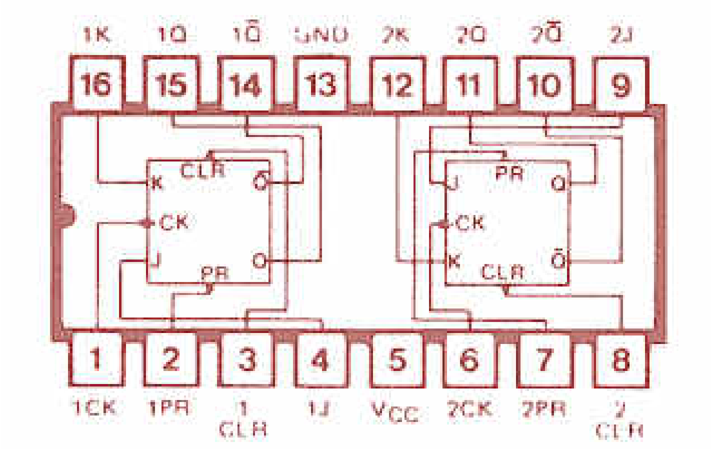


Figure 3.10 : Brochage du circuit intégré **7476**.

3.1.3. Bascule D

Cette bascule diffère des bascules JK et RS du faite quelle possède uniquement une seule entrée D. Il en existe deux types :

- ✓ La bascule D FLIP-FLOP.
- ✓ La bascule D LATCH.

3.1.3.1. Bascule D élémentaire (positive edge triggered)

Un transfert de l'entrée vers la sortie s'effectue à chaque Top d'horloge (circuit de retard : **Delay**).

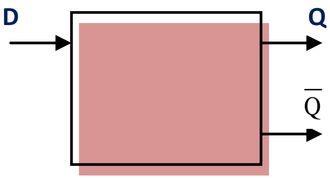


Table de vérité

D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1

D : Entrée.
Q : Etat passé.
Q^+ : Etat présent.

$Q^+ = D$ → La sortie suit l'entrée mais après un délai.

Figure 3.11 : Bascule D-Latch.

Bascule D à partir de SR

On peut obtenir une bascule D à partie d'une SR par application de la table de vérité (ou table des états)

Q	D	Q^+	R	S
0	0	0	x	0
0	1	1	0	1
1	0	0	1	0
1	1	1	1	x

$$\begin{cases} R = \overline{D} \\ S = D \end{cases}$$

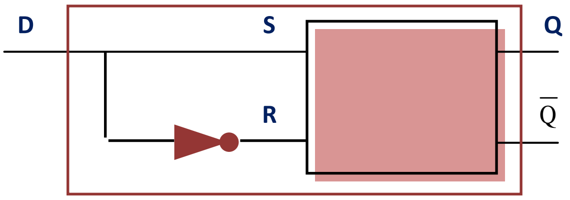


Figure 3.12 : *D à partir de SR.*

Bascule D synchrone

Une bascule D est réalisée à partir d'une bascule RS ou JK dont les entrées sont reliées par un inverseur. Ceci impose donc que les entrées prennent des états complémentaires.

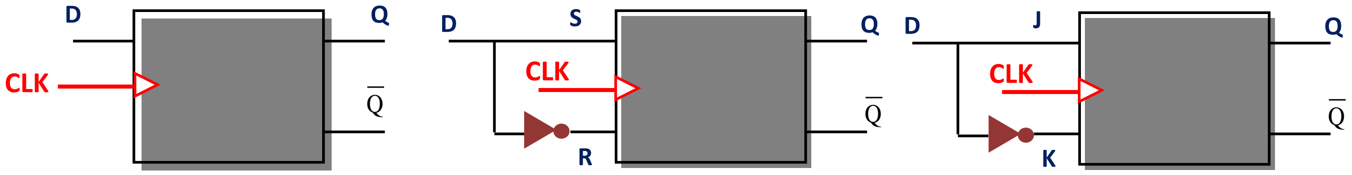


Figure 3.13 : *D à partir de SR et JK*

.

Chronogramme

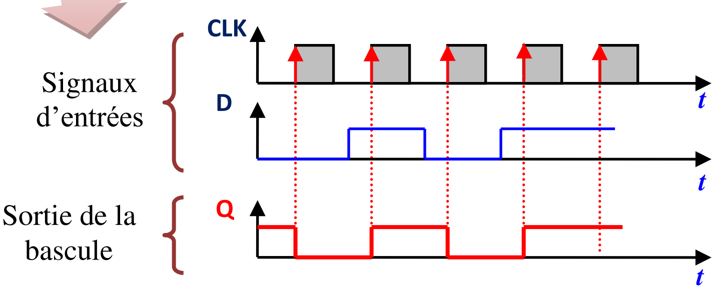
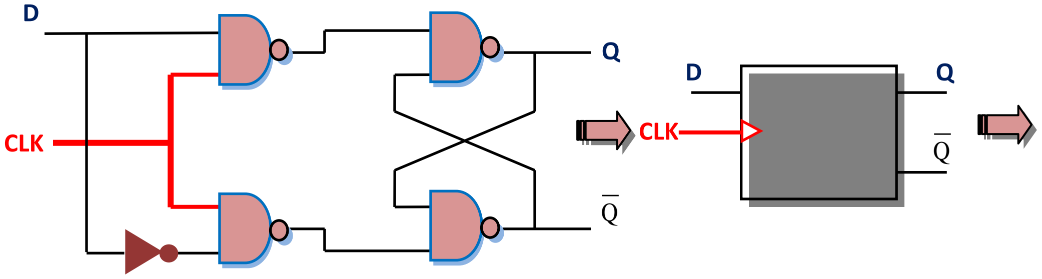


Figure 3.14 : *Diagramme temporel de la Bascule D synchronisée par front montant.*

La sortie prend l'état de l'entrée D après l'impulsion d'horloge. Ceci permet par exemple de synchroniser le transfert de données en parallèle.

3.1.3.2. Bascule D à verrouillage (Latch)

Cette bascule ne possède pas de circuit détecteur de front et la sortie Q prend donc l'état de l'entrée D tant que l'horloge est à l'état haut.



D	CLK	Q
x	0	Q
0	1	0
1	1	1

Figure 3.15 : *Bascule D à verrouillage.*

En savoir +

La table de vérité montre que :

- L'état de Q est invariable tant que le signal d'horloge est au niveau bas.
- Sur le front actif du signal d'horloge, la sortie Q prend l'état de D.
- Q recopie tous les états de D tant que le signal d'horloge est au niveau haut.
- *On ne parle plus dans cette bascule de l'entrée d'horloge mais plutôt de l'entrée de validation.*

Chronogramme

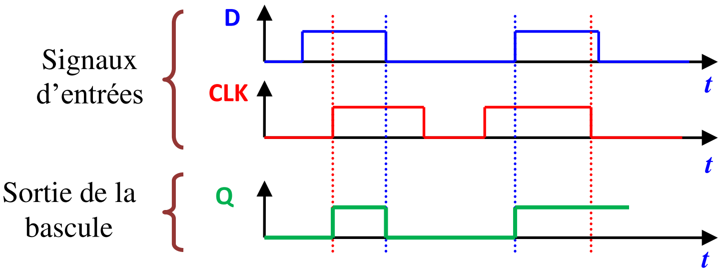


Figure 3.16 : Diagramme temporel de la Bascule D à verrouillage.

⌖ ***Références techniques :*** on peut retrouver la bascule D dans les circuits intégrés ci-dessous :

- **7474, 74175, 74273**
- **74LS377, 74LS379**
- **74HC574**

Exemple : Le circuit intégré **7474** comprenant **02** bascules du type D .

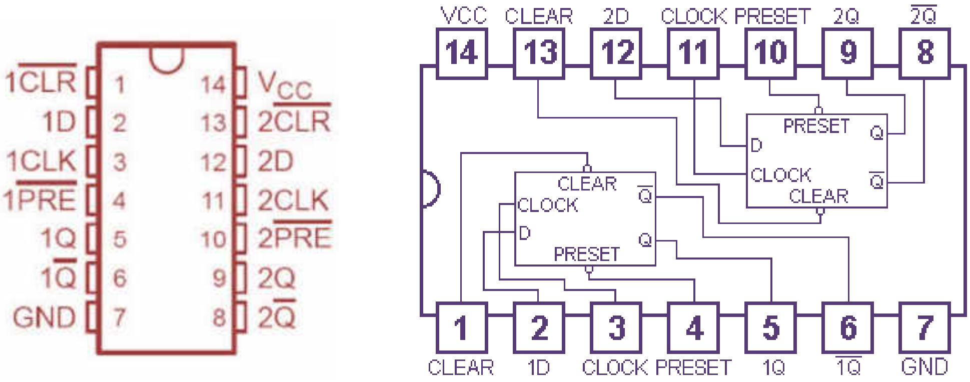


Figure 3.17 : Brochage du circuit intégré **7474**.

Exemple d'application : Transfert des données parallèle avec retard :

- Adressage des mémoires.
- Réalisation des registres à décalage.
- Réalisation des compteurs.

- …etc.

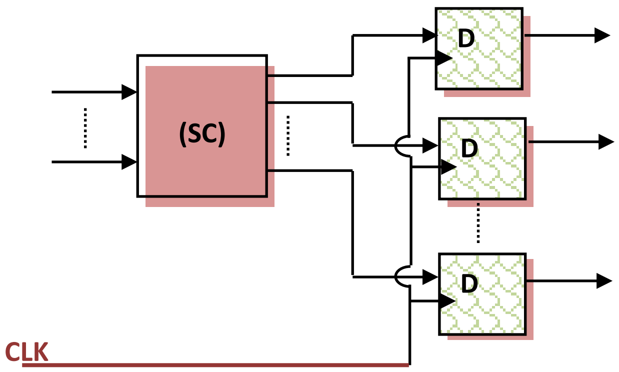


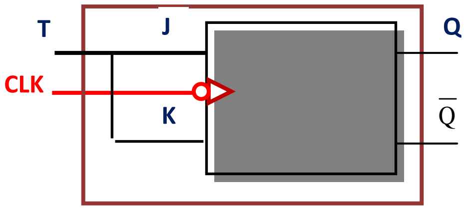
Figure 3.18 : *Transfert des données parallèles.*

La bascule JK est la bascule la plus complète, offrant tous les modes de fonctionnement que l'on peut demander à une bascule.

3.1.4. Bascule T (Symétrique : Toggle)

La principale fonction d'une bascule T est de réaliser un changement d'état du signal de sortie «Q» quand le signal de commande (signal d'horloge) est actif:

- ✓ Si la sortie «Q» est à un niveau logique «bas», alors elle passe à un niveau logique «haut».
- ✓ Si la sortie «Q» est à un niveau logique «haut», alors elle passe à un niveau logique «bas».
- ✓ Le basculement du signal de sortie «Q» intervient soit sur un front montant, soit sur un front descendant.
- ✓ Cette bascule est obtenue tout en connectant les entrées JK à une seule entrée T.



T	Q	Q^+	J	K	D
0	0	0	0	x	0
0	1	1	x	0	1
1	0	1	1	x	1
1	1	0	x	1	0

$$\begin{cases} Q^+ = T \oplus Q \\ J = K = T \end{cases}$$

$$\{ D = T \oplus Q$$

Figure 3.19 : *Bascule T.*

⌖ ***Références techniques :*** On peut trouver la bascule T dans les circuits intégrés ci-dessous : **SN74AS303**, etc.

Remarque : L'utilisation d'une bascule T permet de diviser la fréquence d'un signal par 2 (ou de doubler sa période).

- ✓ L'utilisation de plusieurs bascules T permet de réaliser des compteurs.

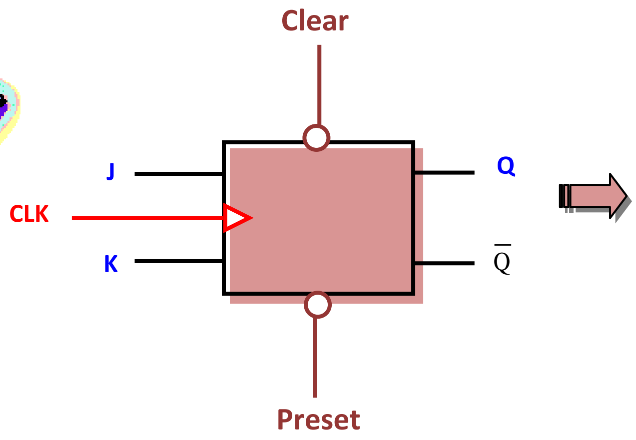
3.2. Entrées synchrones & asynchrones des bascules

Les entrées JK, SR, D, … sont des entrées synchrones parceque leurs effets sont conditionnées par le signal d'horloge **CLK** (**CLK** : Signal de synchronisation). Dans certaines situations, on aura besoin des entrées supplémentaires. Ces entrées sont des entrées asynchrones :

- Remise à '0' forcément (**Clear**).
- Mise à '1' forcément (**Preset**).

Ces deux entrées asynchrones sont désignées entrées d'initialisation ou de forçage.

Exemple pour la bascule JK :



Prest	Clear	Réponse de FF
0	0	Ambigüité
0	1	Q = 1 Forcément
1	0	Q = 0 Forcément
1	1	Pas d'effet

Figure 3.20 : *Entrées synchrones & asynchrones.*

En savoir +

- ✓ La négation logique sur les deux entrées asynchrones **Preset** et **Clear** indique qu'elles sont actives sur le niveau bas du signal qui leur est appliqué.
- ✓ Son champ d'utilisation, est particulièrement, dans la réalisation **des compteurs et des registres à décalage.**

3.3. Bascule Maitre-Esclave

Problème: Les bascules synchrones nécessitent des états stables sur leurs entrées au moment de la transition du signal d'horloge CLK, cela n'est pas toujours possible lorsque plusieurs bascules sont câblées entre elles (ex: en comptage) et l'on a des aléas de fonctionnement.

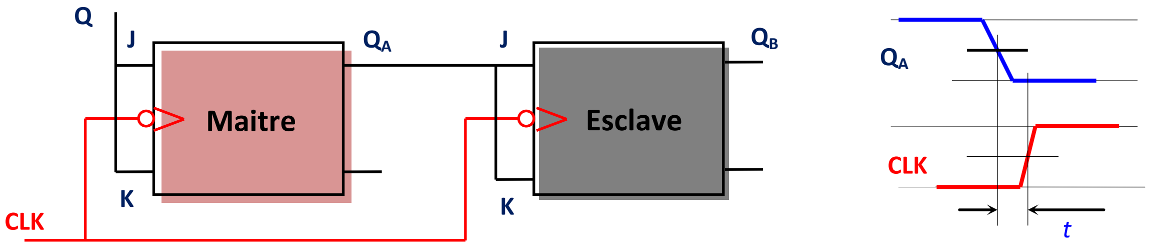


***Figure 3.21 :** Structure maître-esclave.*

⌖ **Solution:** Il existe des bascules à 2 étages qui évoluent en 2 temps.

1^er^ temps: ↑ Verrouillage du 2ème étage.
Prise en compte des entrées par le 1er étage.

⌖ Le premier étage (Maitre) ayant pour fonction d'enregistrer les informations d'entrées.

2ème temps : ↓ Verrouillage du 1er étage.
Prise en compte des données par le 2ème étage.

⌖ Le deuxième étage (Esclave) ayant pour fonction d'afficher l'état résultant de la bascule.

- ✓ La mise en cascade de plusieurs bascules est très utilisée pour différentes applications comme la réalisation des registres à décalage et les compteurs.
- ✓ Pour de telle application, il est nécessaire que les bascules ne changent pas d'état simultanément lors de l'application du signal d'horloge.

a) Bascule RS Maitre-Esclave

Une bascule de type Maître-Esclave est constituée de deux cellules élémentaires en cascade. Le chargement du signal d'entrée passe en premier dans la cellule Maître, puis est transféré dans la cellule Esclave.

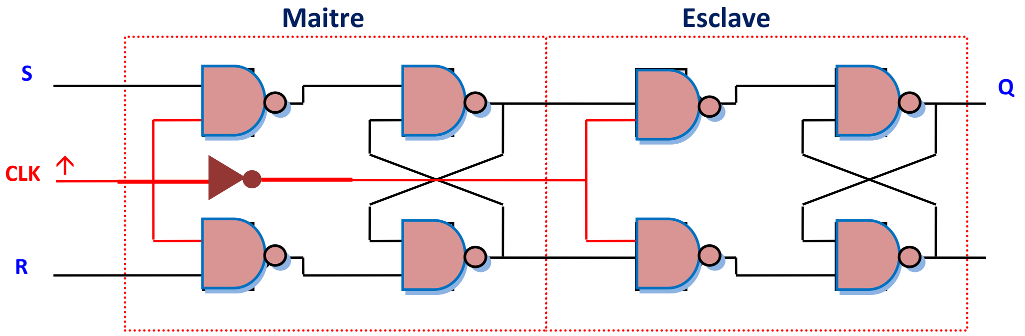


***Figure 3.22 :** Bascule RS maître-esclave (RS-MS).*

✓ Avec cette structure Maître-Esclave, les sorties de la bascule peuvent commuter seulement quand CLK passe de 1 à 0.

- ✓ De ce fait, la sortie Q de la bascule RS-MS change au moment des fronts descendants de CLK selon la table de vérité ci-dessous.

Table 3.3 : *Table des états de la bascule RS-MS.*

CLK	S	R	Q_t
⎍	0	0	Q_{t-1} maintien de l'état précédent
⎍	0	1	0 mise à 0 (reset)
⎍	1	0	1 mise à 1 (set)
X	1	1	à interdire

b) Bascule JK Maitre-Esclave

- ✓ Les entrées R et S, dans le cas de la bascule RS-MS ne doivent pas être simultanément à 1 pour ne pas donner lieu à un état d'indetermination (ambiguité).
- ✓ Cette ambiguïté est levée dans la bascule JK puisque l'état indéterminé est remplacé par un état complémenté, on dit qu'il y a basculement.

Comme le montre la figure ci-dessous, la bascule JK Maître-Esclave est obtenue à partir d'une bascule RS Maître-Esclave en introduisant une rétroaction.

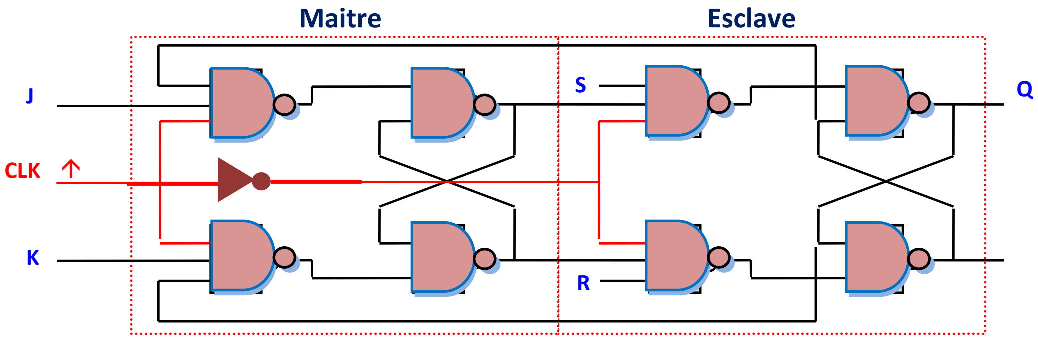
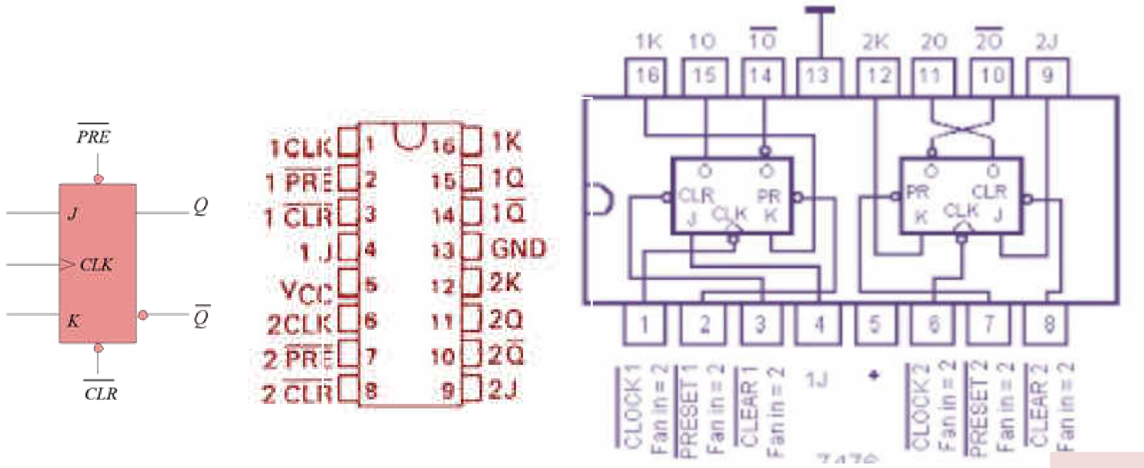


Figure 3.23 : *Bascule JK maître-esclave (JK-MS).*

Table 3.4 : *Table des états de la bascule JK-MS.*

CLK	J	K	Q_t
⎍	0	0	Q_{t-1} mémorisation
⎍	0	1	0 mise à 0
⎍	1	0	1 mise à 1
X	1	1	$\overline{Q}_{t-1}$ basculement

- ***Références techniques :*** On peut retrouver la bascule JK maitre-esclave dans les circuits intégrés ci-dessous : **74LS76**, etc.



Chronogramme

***Figure 3.24 :** Brochage du circuit intégré **74LS76**.*

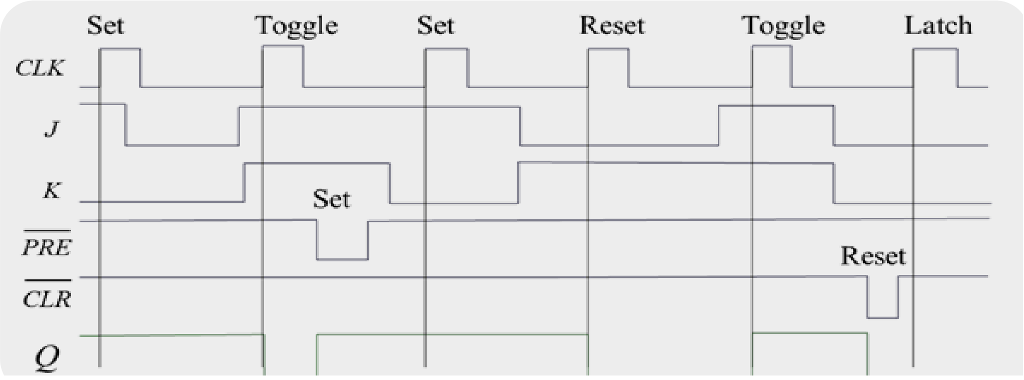


***Figure 3.25 :** Diagramme temporel de la Bascule JK maitre-esclave.*

3.4. Application des bascules

Les bascules (ou tout simplement les éléments mémoires de base) peuvent interviennent à l'implémentation de plusieurs système logiques. On peut citer :

- ✓ Circuits de mémorisation des données numériques.
- ✓ Générateurs des impulsions (diviseurs de fréquences).
- ✓ Transfert des données (transfert synchrone et transfert asynchrones).
- ✓ Elimination des effets des rebondissements des interrupteurs.
- ✓ Compteurs et registres à décalage.
- ✓ … etc.

Exemple : Aplication de la bascule RS → interrupteur anti-rebonds

Une application très utile de la bascule RS est l'interrupteur sans rebonds. Il est pratiquement impossible de réaliser un interrupteur mécanique dans lequel il n'y aurait qu'une seule transition entre deux tensions, à cause du phénomène de rebondissement des contacts. Ce phénomène est illustré par la figure suivante.

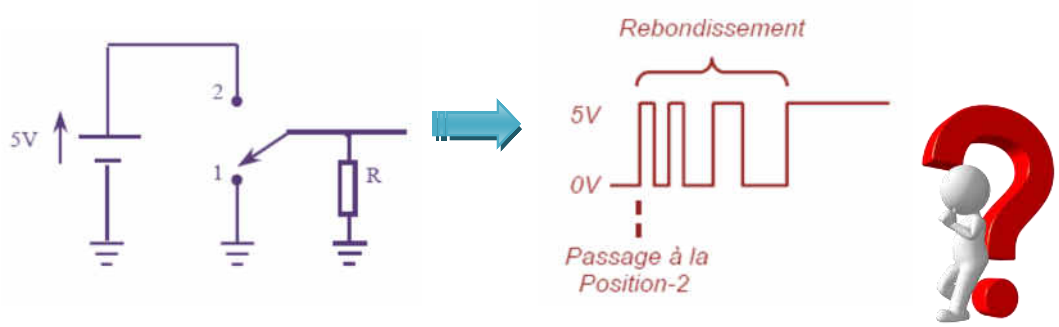


Figure 3.26 : Interrupteur avec rebondissement.

Les effets de rebondissement qui se produisent lors de la fermeture des contacts d'un interrupteur, ou d'un relais, peuvent être éliminés en utilisant une bascule RS, comme le montre la figure ci-dessous.

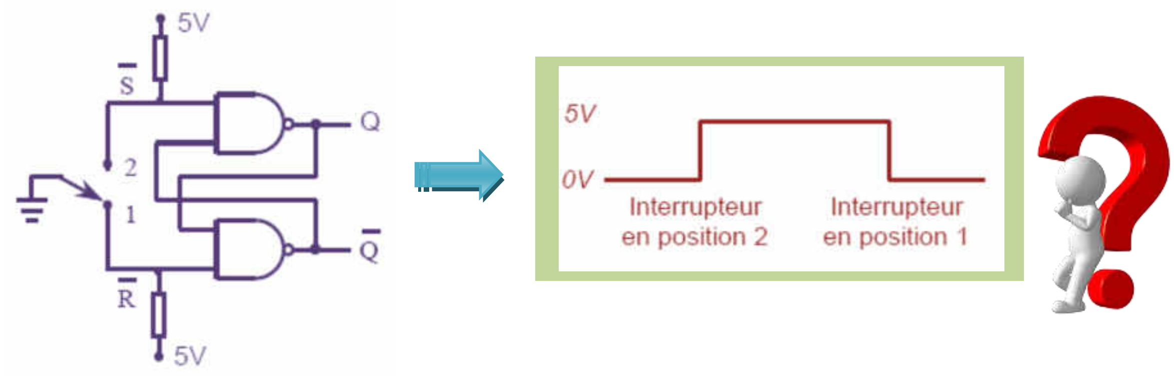


Figure 3.27 : Interrupteur sans rebondissement.

Fonctionnement du montage :

1. Supposons que l'interrupteur soit au début en position 1, de sorte que R = 0, donc Q = 0.
2. Quand l'interrupteur est amené en position 2, R passe à 1 et S à 0, cela a pour effet de placer la sortie Q à 1.
3. Maintenant, si l'interrupteur rebondit, S passe au niveau Haut, ce qui n'affecte en rien la valeur de Q (car S = R = 1).
4. On voit bien que Q reste inchangé malgré les rebonds de la lame sur le contact 2.
5. De même, quand l'interrupteur passe de la position 2 à la position 1, S passe à 1 et R à 0, donc Q passe à 0 et conserve cet état même si la lame rebondit.
6. Donc, l'insertion de cette bascule fait de sorte que Q effectue une seule transition quand l'interrupteur change de position.

3.5. Caractéristiques des bascules

✓ Le constructeur définit un certain nombre de paramètres dynamiques que l'on doit respecter pour obtenir un fonctionnement correct du circuit utilisé.

- ✓ Le bon fonctionnement d’une bascule exige deux conditions entre le changement d’état de l’entrée de commande et le front actif du signal d’horloge.

- ⌖ **Un temps minimal de stabilisation Tset-up** (ou de prépositionnement) qui est l’intervalle de temps qui précède le front actif de l’horloge et pendant lequel l’entrée synchrone doit être maintenue au niveau approprié. En d’autres termes, **Tset-up** est le temps qu’il faut garder les entrées de contrôles JK, SR, etc. stable avant synchronisation (**Tset-up** est de l’ordre de 5 à 50ns).

- ⌖ **Un temps minimal de maintien Thold** (Hold Time) qui est l’intervalle de temps qui suit immédiatement le front actif du signal d’horloge et pendant lequel l’entrée synchrone doit être gardée au niveau approprié. En d’autres termes, **Thold** est le temps qu’il faut maintenir les entrées de contrôle JK, SR, etc. stable après synchronisation (**Thold** est de l’ordre de 0 à 10ns).

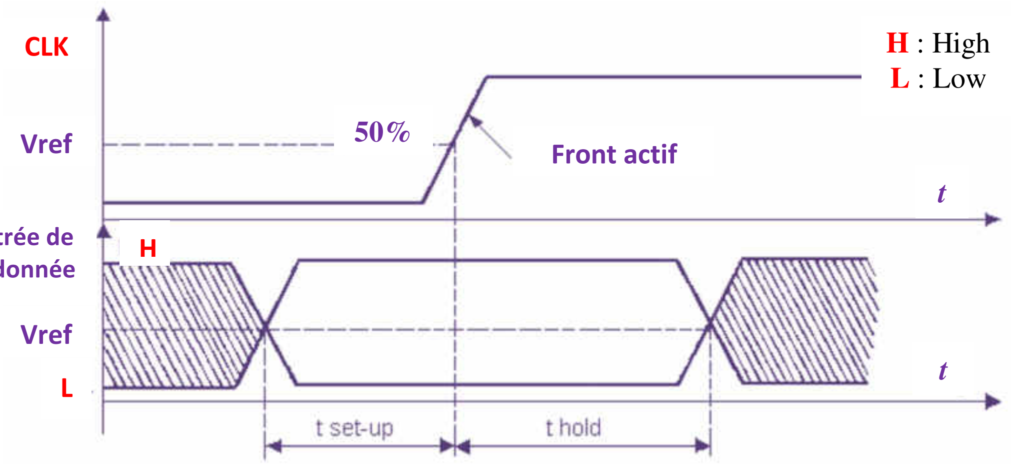
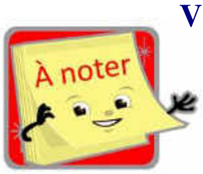


Figure 3.28 : Temps de prépositionnement et de maintien.



Vref correspond à la tension de basculement des portes du circuit **:**

- ⌖ ► **Vref = 1,5 V** en technologie **TTL standard.**
- ⌖ ► **Vref = 1,3 V** en technologie **TTL - LS.**
- ⌖ ► **Vref = Vdd / 2** en technologie **CMOS, Vdd** est la tension d'alimentation du circuit en technologie CMOS.

D’autres paramètres sont à prendre en compte pour le bon fonctionnement des bascules, à savoir :

- ✓ Le retard de propagation.
- ✓ La fréquence de synchronisation maximale (fmax).

⌖ **Le retard de propagation :** Chaque fois qu'un signal doit changer l'état d'une bascule, on observe un retard entre le moment où le signal est appliqué et le moment où le changement apparaît à la sortie.

- ✓ Le temps de propagation **t_{PLH}** est le temps qui s'écoule entre l'instant où l'entrée de commande devient active et l'instant où la sortie passe du niveau **L** au niveau **H**.
- ✓ L'entrée de commande peut être l'entrée **d'horloge**, l'entrée **CLEAR** ou l'entrée **PRESET.** Ce temps noté **t_{PLH}** est spécifié pour une entrée donnée (**CLOCK, CLEAR** ou **PRESET**) et une sortie donnée (**Q** ou $\overline{Q}$).
- ✓ En pratique, ce temps correspond au retard apporté par les portes internes du circuit.
- ✓ Le temps de propagation **t_{PHL}** est le temps qui s'écoule entre l'instant où l'entrée de commande devient active et l'instant où la sortie passe du niveau **H** au niveau **L.**

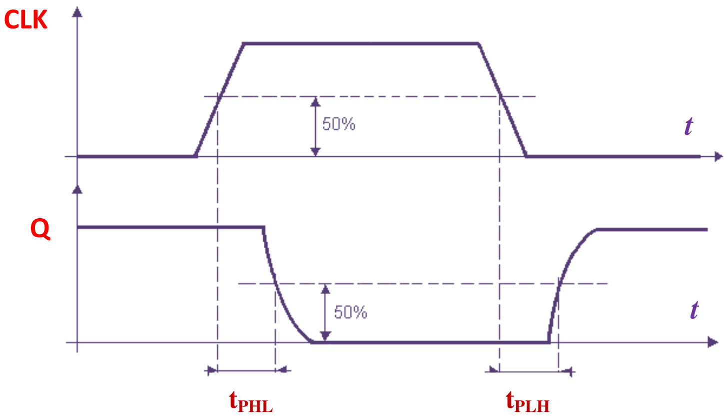


Figure 3.29 : *Temps de propagation.*

- **La fréquence de synchronisation maximale (fmax) :** il s'agit de la fréquence la plus haute que peut avoir le signal d'horloge et qui assure encore un déclenchement fiable (un bon fonctionnement) de la bascule.
  - ✓ Cette fréquence limite de fonctionnement **fmax** est due au retard apporté par les portes du circuit. Elle correspond à une période minimale **1 / fmax** du signal d'horloge comme l'indique la figure ci-dessous.

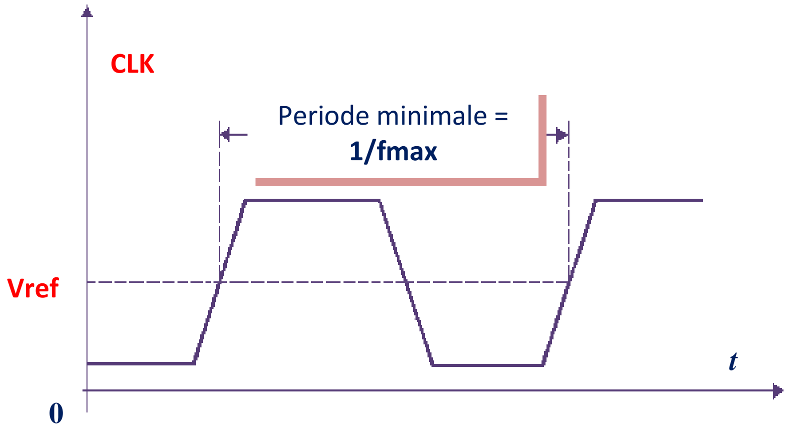


Figure 3.30 : *Fréquence de synchronisation maximale.*

3.6. Exercices

EXERCICE 01 :

- Etablir les équations caractéristiques des bascules RS, JK, T et D.
- Réaliser une bascule T à partir des flip-flops :
    1. RS,
    2. JK,
    3. D.
- Réaliser une bascule D à partir des bascules T, RS et JK.
- Etudier le problème de synchronisation des différentes bascules D, T, JK, RS.

EXERCICE 02 :

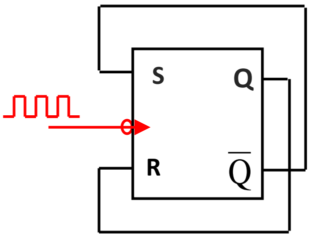
Soit la bascule $\overline{S}\overline{R}$:

4. Etablir sa table de transition.
5. Donner la table de vérité de la JK et montrer peut–on trouver son logigramme à partir de la ($\overline{S},\overline{R}$).
6. Définir la bascule T et trouver son circuit a partir de la JK.

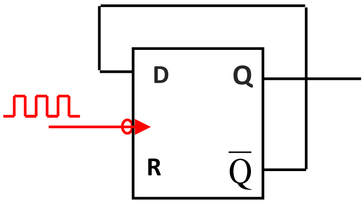
7. Définir la bascule D et trouver son circuit a partir de la SR.

EXERCICE 03 :

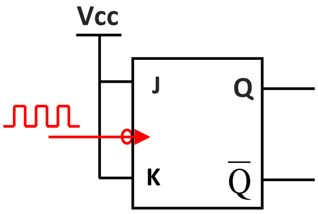
Donnez le chronogramme des sorties **Q** de chacune des bascules câblées ci-dessous en fonction d'une entrée d'horloge **CLK**.



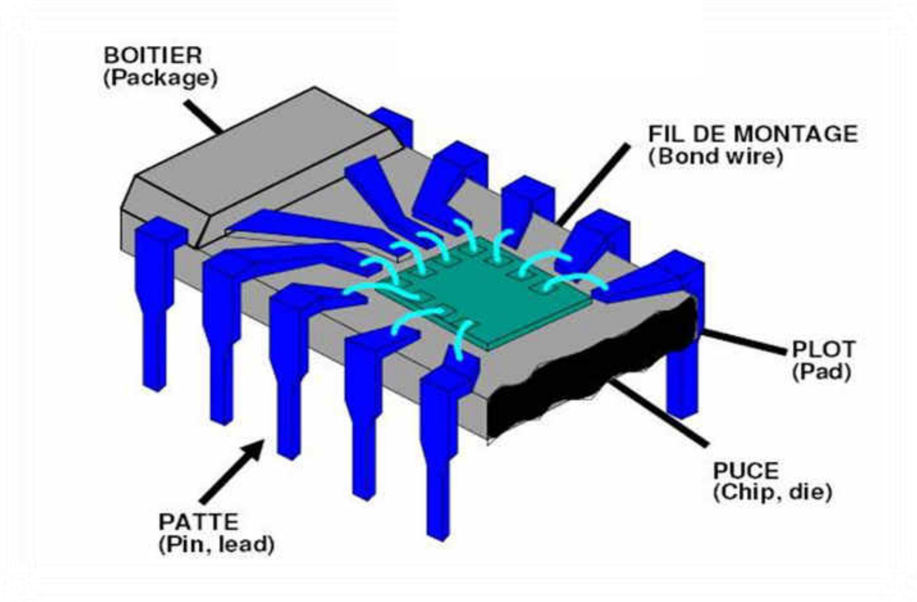
(a)



(b)



(c)



CHAPITRE 4

LES REGISTRES

4.1. Définition

Un registre est un système séquentiel, permettant la mémorisation d'un ensemble **d'informations** (de **bits**). Il est donc constitué de **n** bascules, mémorisant chacune un bit, connectées à la même horloge. La figure 4.1 donne un exemple de registre à **n** bits.

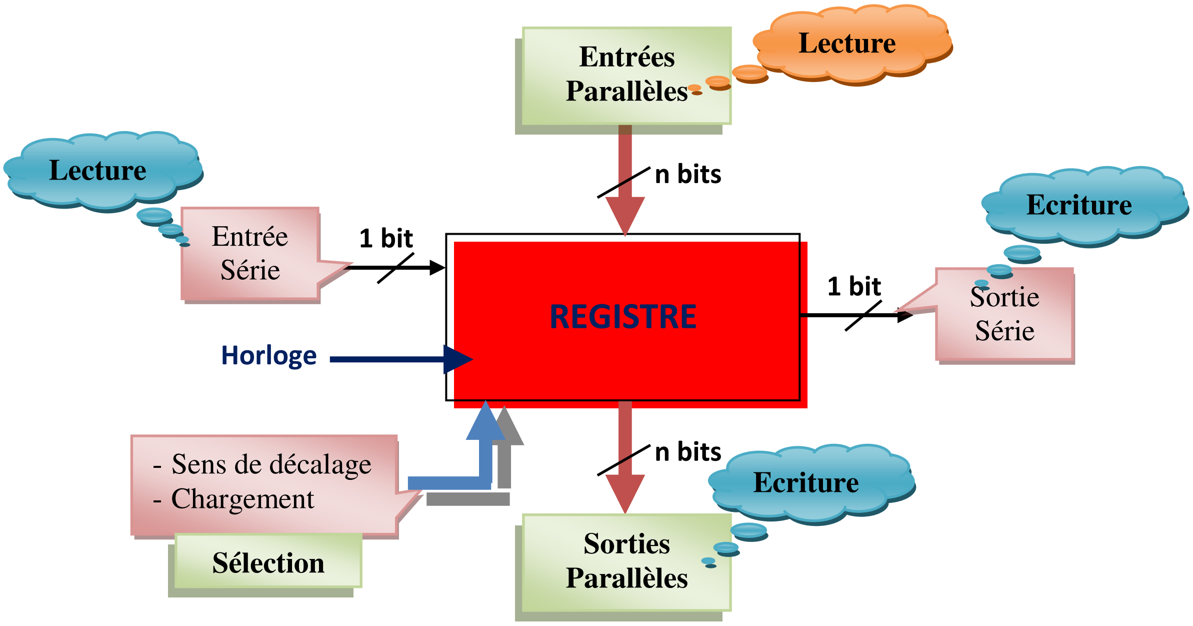


Figure 4.1 : *Structure générale d'un registre.*

4.2. Classement des registres

Il existe plusieurs types de registres dont chacun est bien adapté à un certain type d'application. Donc, la fonction du registre dépend de l'interconnexion entre les entrées et les sorties, comme indique la figure 4.2 avec un registre à 4 bits.

Table 4.1 : *Types des registres.*

Type	Structure	Description	Mode de fonctionnement		
			Ecriture	**Lecture**	**fonction**
PIPO	Parallèle - Parallèle	Parallel Input - Parallel Output	Parallèle	Parallèle	***Mémorisation***
SISO	Série - Série	Serial Input - Serial Output	Série	Série	***Décalage***
SIPO	Série - Parallèle	Serial Input - Parallel Output	Série	Parallèle	***Décalage***
PISO	Parallèle - Série	Parallel Input - Serial Output	Parallèle	Série	***Décalage***

Exemple d'un registre à 04 bits :

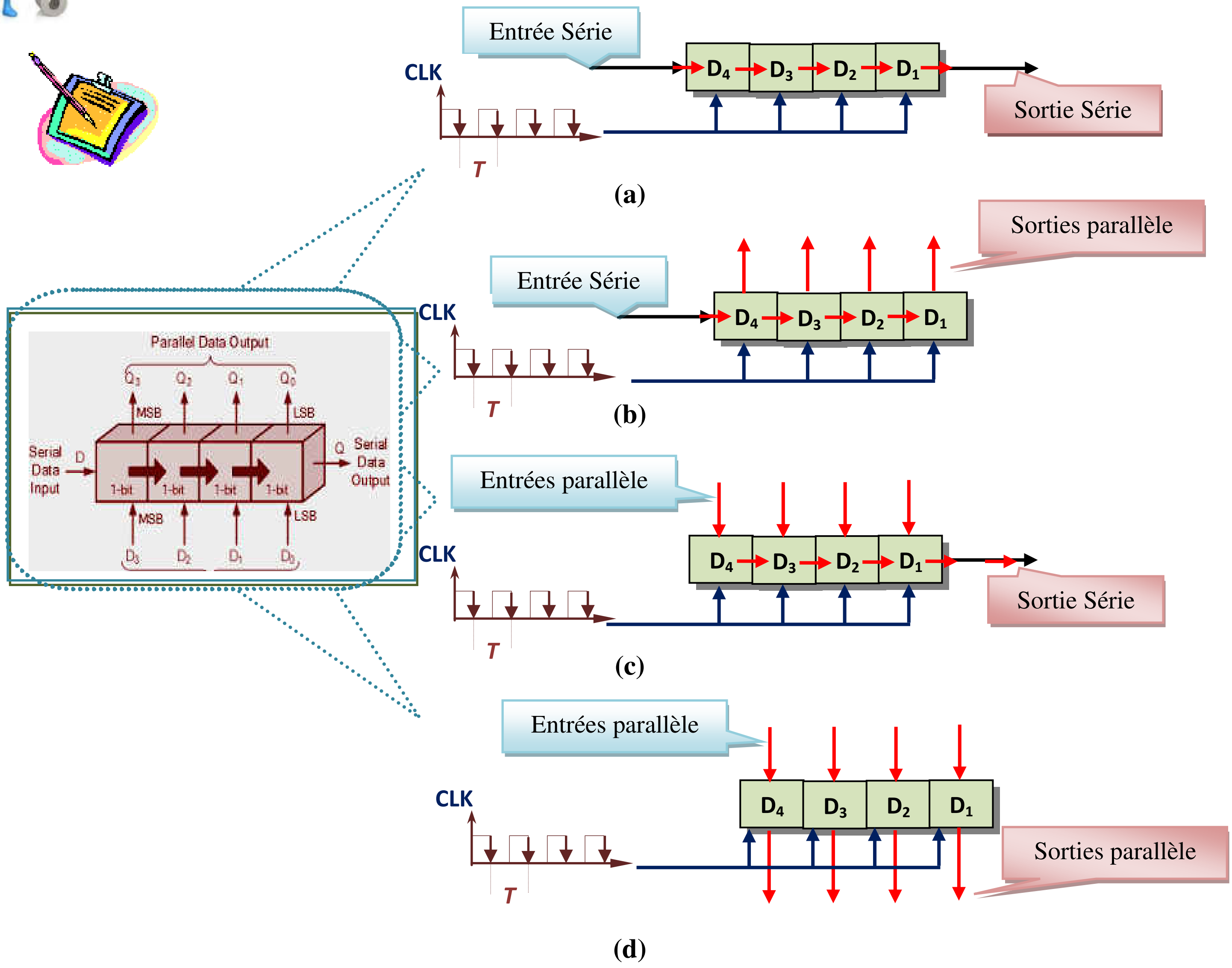
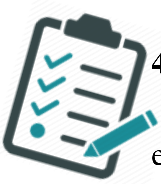


Figure 4.2 : *Différents types des registres.*

En savoir +

- ✓ Les bascules dans un registre sont reliées (interconnectées) entre elles soit directement, soit par des opérateurs combinatoires.
- ✓ Les opérations sur les bascules sont conditionnées par un signal externe : Signal de synchronisation (signal d'Horloge : **CLK**).

- Un autre type de registres dits ***registres universels*** (registres à entrées séries ou parallèles et sorties séries ou parallèles).



4.2.1. Applications des registres

Par l'emploi des registres, on peut réaliser un certains nombres d'applications particulière en électronique numérique, à savoir :

- Stockage (mémorisation) des informations sous forme binaire.
- Transfert (transmission) des données.
- Conversion des données (parallèle-série, Série- parallèle).
- Opérations arithmétiques et logiques (test, comptage, comparaison, division, …etc.).
- Synchronisation des signaux avec une horloge.
- …

Suite à ces applications offertes par les registres, on peut classer les registres en quatre grandes familles :

- Registres à décalage.
- Registres de mémorisation.
- Compteurs.
- Registres de synchronisation.

Donc, le classement des registres s'effectue à travers deux facteurs principaux :

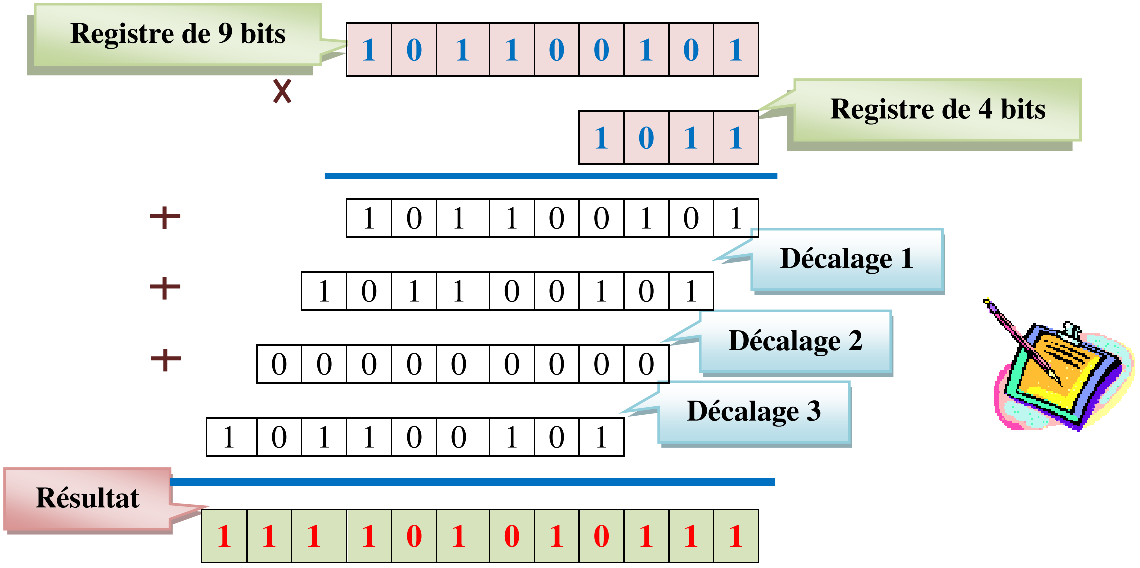
- ***Structurel***, mesuré par la taille du registre, c-à-d, le nombre de bascules.
- ***Fonctionnel***, mesuré par sa fonction (décalage à droite, …), par le type de transfert des données, …etc.

4.3. Registres à décalage

L'opération de décalage consiste à transférer des données entre deux cellules binaires consécutives, selon un sens (de transfert) :

- Bidirectionnel.
- Unidirectionnel (de droite vers la gauche, de gauche vers la droite).

Exemple : Dans la multiplication binaire, il est nécessaire de décaler des bits à gauche ou à droite.



4.3.1. Structure d'un registre universel

La structure d'un registre dépend de sa fonction, c-à-d, le type de chargement :

- ✓ Série (introduction des bits d'information l'un après l'autre) ou
- ✓ Parallèle (introduction simultanée des bits d'informations).

La structure générale d'un registre universel ainsi que son mode de fonctionnement sont schématisées par la figure 4.3.

Variable	Rôle
X	Entrée de chargement parallèle sur n bits
XR	Entrée série (décalage série à droite)
XL	Entrée série (décalage série à gauche)
CTRL	2 lignes de contrôle (chargement parallèle ou décalage série)
CLR	Entrée d'initialisation (a)synchrone
Z	Sortie sur n bits

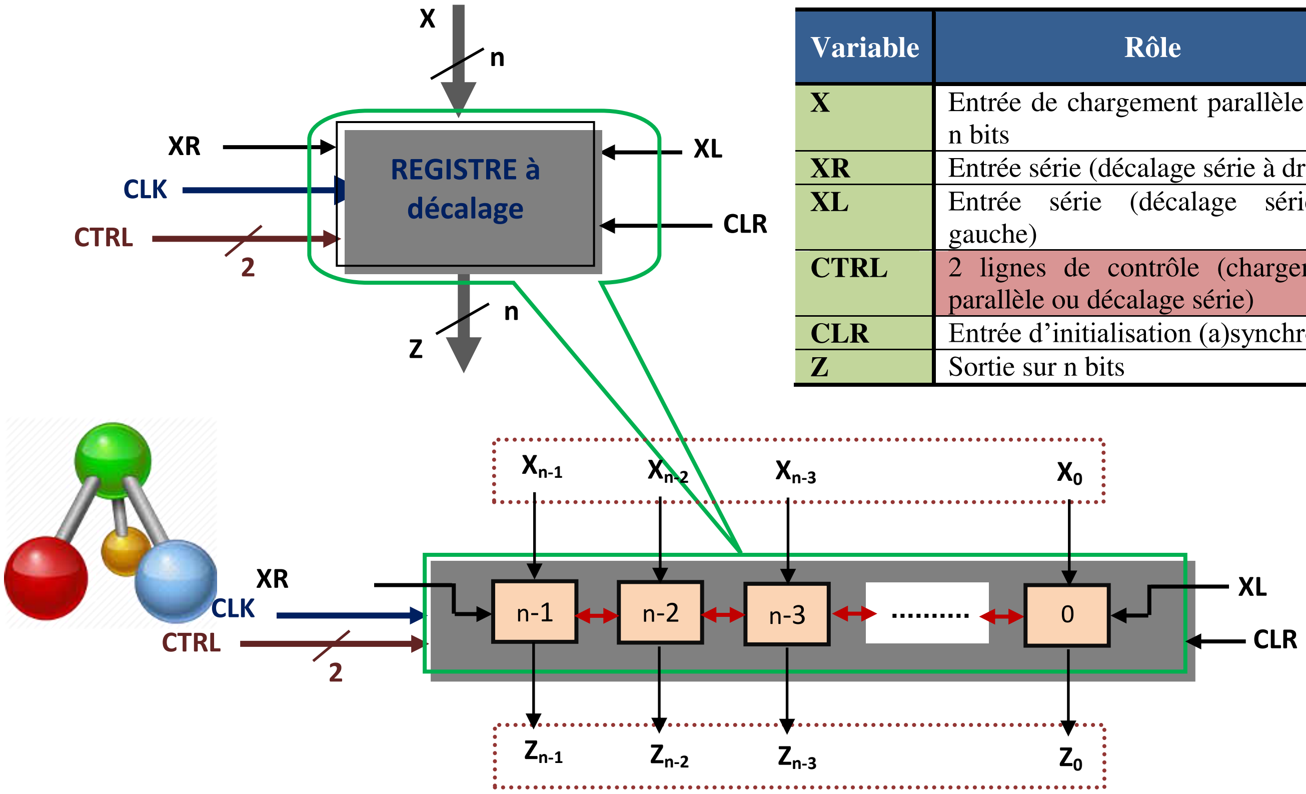
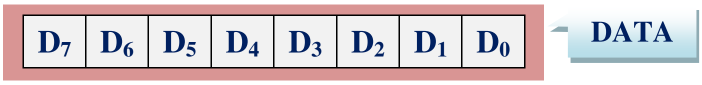


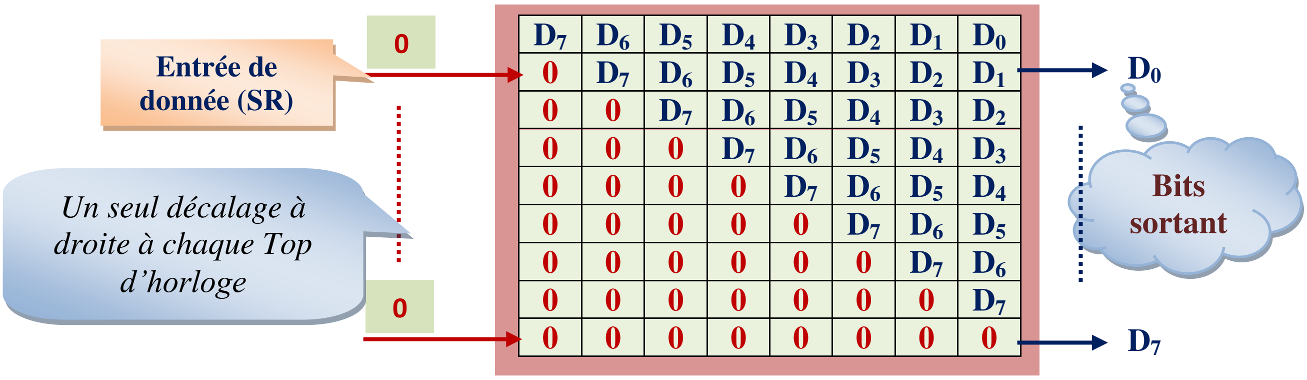
Figure 4.3 : *Schéma bloc et mode de fonctionnement d'un registre universel.*

Un registre dit ***universel*** est un registre qui effectue le décalage à droite ou à gauche et un chargement série ou parallèle. Ce type de registre dispose des commandes qui définissent le sens de décalage et le type de chargement.

Exemple : Soit un mot de 08 bits (taille de registre = 8), on peut lui faire subir plusieurs types de décalage, à savoir :

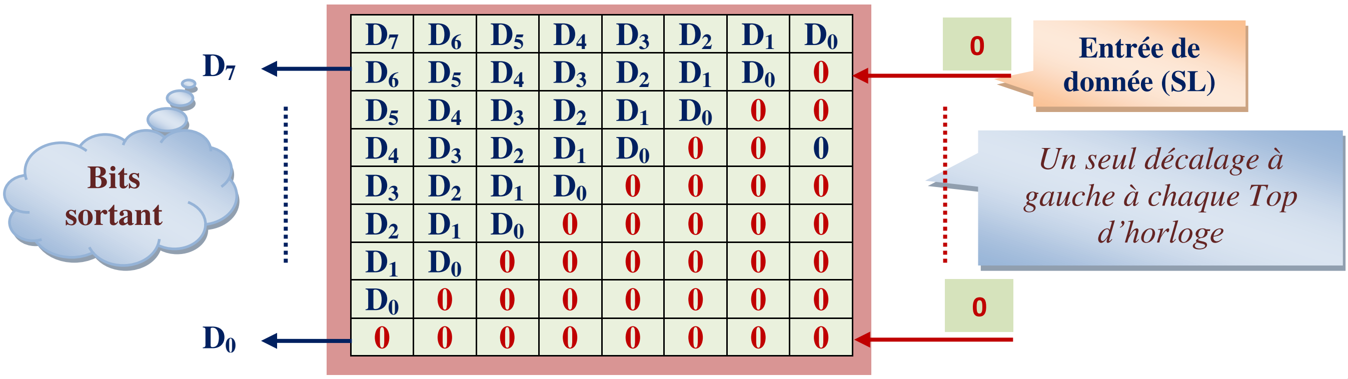
- Décalage à droite (SR : Shift Right).
- Décalage à gauche (SL : Shift Left).
- Décalage circulaire (avec rotation)
  - Rotation à gauche.
  - Rotation à droite.





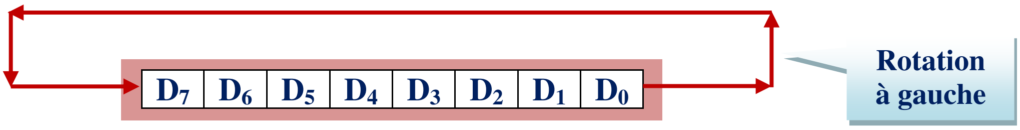
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	D_7	D_6	D_5	D_4	D_3	D_2	D_1
0	0	D_7	D_6	D_5	D_4	D_3	D_2
0	0	0	D_7	D_6	D_5	D_4	D_3
0	0	0	0	D_7	D_6	D_5	D_4
0	0	0	0	0	D_7	D_6	D_5
0	0	0	0	0	0	D_7	D_6
0	0	0	0	0	0	0	D_7
0	0	0	0	0	0	0	0

(a)

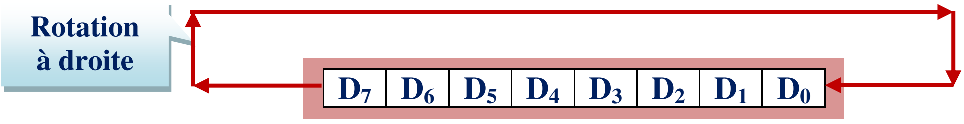


D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
D_6	D_5	D_4	D_3	D_2	D_1	D_0	0
D_5	D_4	D_3	D_2	D_1	D_0	0	0
D_4	D_3	D_2	D_1	D_0	0	0	0
D_3	D_2	D_1	D_0	0	0	0	0
D_2	D_1	D_0	0	0	0	0	0
D_1	D_0	0	0	0	0	0	0
D_0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)



(c)



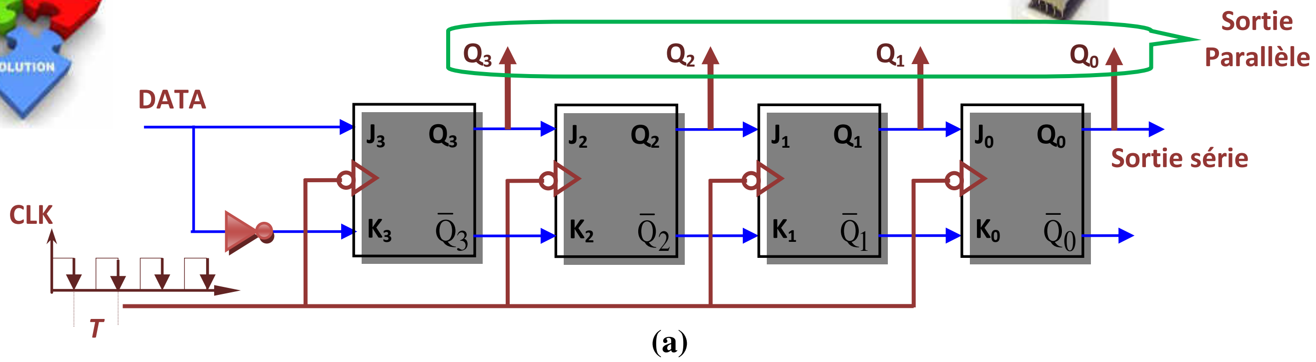
(d)

Figure 4.4 : *Différents type de décalage sur un registre de 8 bits.*

- **Problème :** Le problème consiste à trouver un circuit qui permet de transférer le contenu d'une bascule à son voisin (amont ou aval) à chaque top d'horloge ?

Exemples :

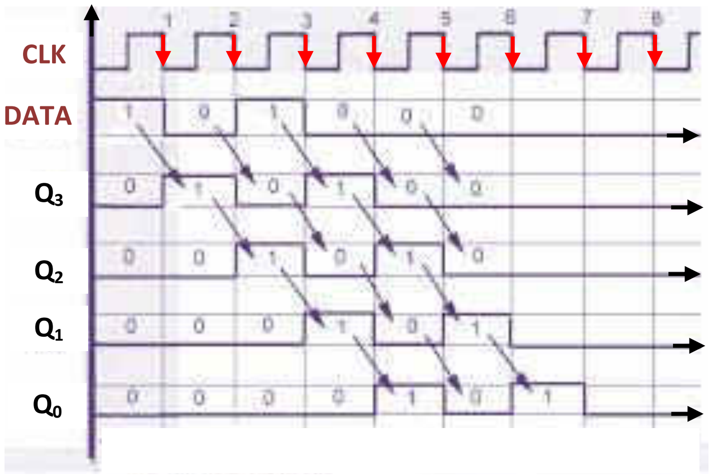
- Registre à **décalage à droite** à **04** bits à base des bascules **JK**.



(a)

DATA	$Q_3Q_2Q_1Q_0$	CLK
	0 0 0 0	Etat initial
1	1 0 0 0	1er Top d'horloge
0	0 1 0 0	2eme Top d'horloge
1	1 0 1 0	3eme Top d'horloge
0	0 1 0 1	4eme Top d'horloge
	0 1 0 1	Etat final

(b)



(c)

Figure 4.5 : *Décalage à droite synchrone: (a) Structure du registre. (b) Table des états. (c) Diagramme temporel.*

A chaque Top d'horloge, la sortie d'une bascule recopie l'information présente à ses entrées, c-à-d, la sortie de la bascule qui la précède ; $J_i = Q_{i-1}$, $K_i = \bar{Q}_{i-1}$. On parle d'un décalage de l'information (principe du maitre-esclave).

- Registre de transfert **Parallèle- Parallèle** à 04 bits à base des bascules D.

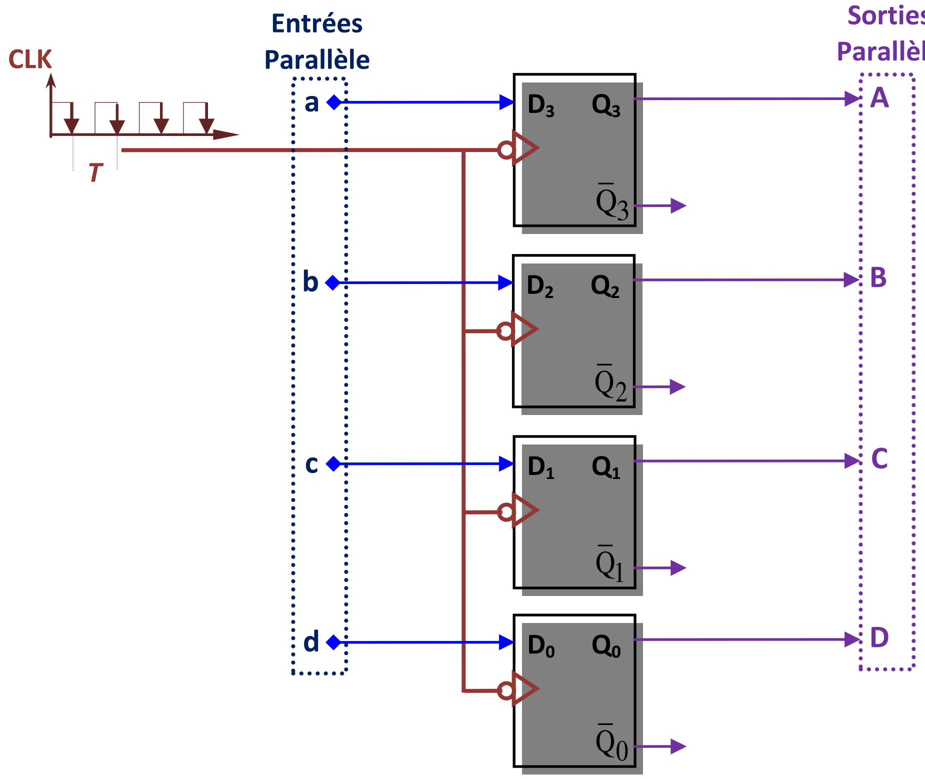


Figure 4.6 : *Transfert parallèle-parallèle à base des bascules D (synchrones).*

- ***Références techniques :*** On peut trouver des registres à décalage dans les circuits intégrés ci-dessous :
  - **74194** : Registre à décalage à 04 bits à entrée/sortie parallèle ou entrée série à décalage à droite ou à gauche.
  - **7496** : Registre à décalage à 05 bits à entrée/sortie parallèle ou entrée série.
  - **74198** : Registre à décalage à 08 bits à chargement ou série, à décalage à droite ou à gauche et sortie.
  - **74178** : Registre à décalage à chargement parallèle synchrone.
  - …

Exemple : Le circuit intégré **7496** comprenant **5** bascules.

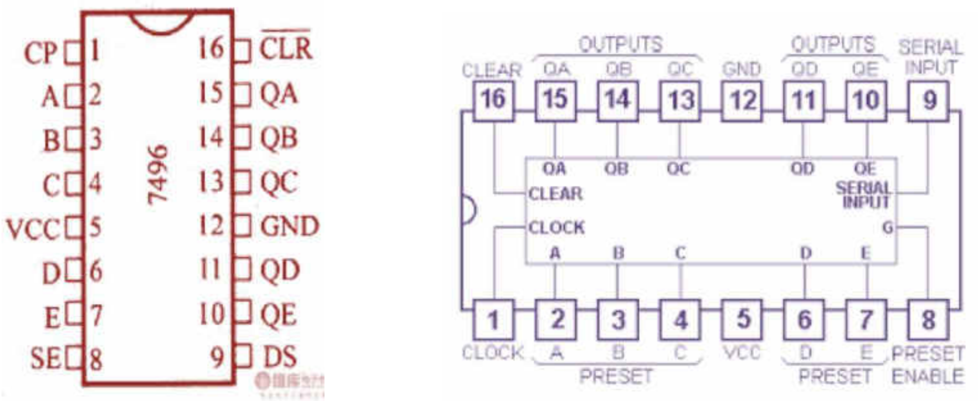


*Figure 4.7 : Brochage du circuit intégré **7496**.*

Remarque : La manipulation la plus fréquente qu'on fait subir aux données conservées dans des bascules ou des registres est le transfert (échange de données d'un registre à un autre).

- Dans les **transferts synchrones** (les plus courants), on utilisera l'**horloge**.
- Dans les **transferts asynchrones**, on utilisera les **entrées de remise à 0 ou 1 asynchrones**.

En savoir +

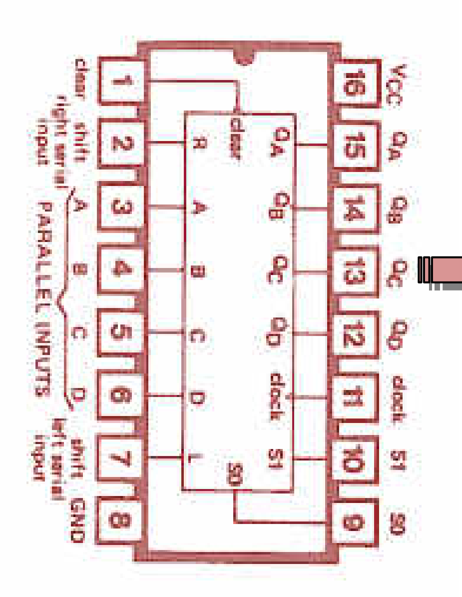
- Parmi les caractéristiques des registres, on peut citer :
  - ✓ Taille (nombre de bits qui constituent le registre).
  - ✓ Types de décalage (à droite, à gauche, les deux directions par une commande de mode (registres bidirectionnels).
  - ✓ Deux types de chargement des registres (parallèle, série).

4.4. Registre universel 74LS194A

Parmi les registres universels, on trouve le 74194 qui est un registre à décalage universel bidirectionnel à 4 bits synchrone. Il possède une entrée de remise à zéro asynchrone. Son évolution est conditionnée par les fronts de CLK. Il présente 4 modes selon les commandes S1 et S0, comme suit :

- **Mode 0 (S1S0 = 00)**: Aucune opération (Inhibition de l'horloge).
- **Mode 1 (S1S0 = 01)**: Décalage à droite avec entrée série gauche sur SR.
- **Mode 2 (S1S0 = 10)**: Décalage à gauche avec entrée série droite SL.
- **Mode 3 (S1S0 = 11)**: Chargement parallèle des entrées A, B, C et D.

La description et le schéma de brochage sont donnés par la figure ci-dessous.

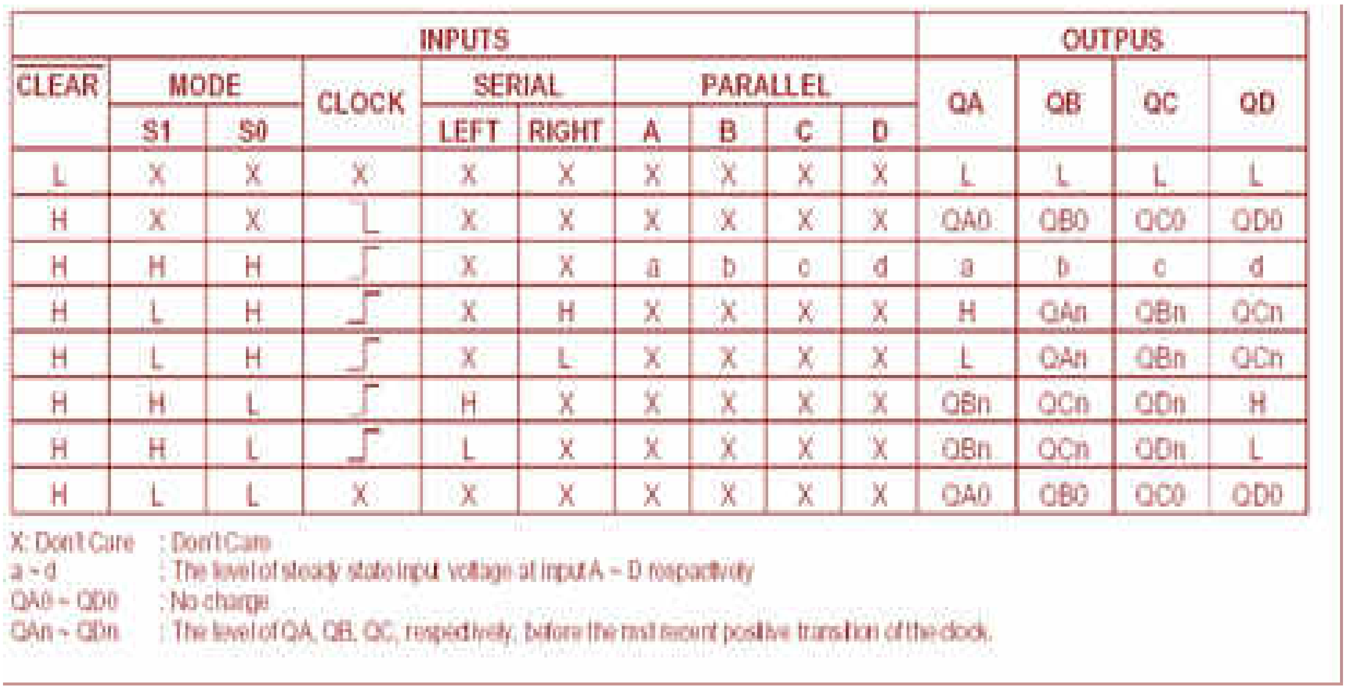


<table>
<tr><th>Broche</th><th colspan="2">Fonction</th></tr>
<tr><td>DCBA</td><td colspan="2">Entrées parallèles</td></tr>
<tr><td>SL</td><td colspan="2">Entrées séries gauche</td></tr>
<tr><td>SR</td><td colspan="2">Entrées séries droite</td></tr>
<tr><td rowspan="5">S1S0</td><td colspan="2">Sélection du mode de fonctionnement</td></tr>
<tr><td>00</td><td>Blocage</td></tr>
<tr><td>01</td><td>Décalage à droite</td></tr>
<tr><td>10</td><td>Décalage à gauche</td></tr>
<tr><td>11</td><td>Chargement parallèle</td></tr>
<tr><td>CLR</td><td colspan="2">Remise à zéro asynchrone des sorties</td></tr>
<tr><td>CLK</td><td colspan="2">Horloge de synchronisation</td></tr>
<tr><td>$Q_DQ_CQ_BQ_A$</td><td colspan="2">Sorties</td></tr>
</table>

Figure 4.8 : *Brochage du circuit intégré* 74LS194A.

Le tableau 4.2 illustre le fonctionnement détaillé du **registre universel 74LS194A**. Un échantillon du chronogramme ullustre graphiquement la table de vérité de ce registre.

Table 4.2 : *Table de vérité du registre universel **74LS194A**.*



INPUTS										OUTPUS			
CLEAR	MODE		CLOCK	SERIAL		PARALLEL				QA	QB	QC	QD
	S1	S0		LEFT	RIGHT	A	B	C	D				
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	L	X	X	X	X	X	X	QA0	QB0	QC0	QD0
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	H	↑	X	H	X	X	X	X	H	QAn	QBn	QCn
H	L	H	↑	X	L	X	X	X	X	L	QAn	QBn	QCn
H	H	L	↑	H	X	X	X	X	X	QBn	QCn	QDn	H
H	H	L	↑	L	X	X	X	X	X	QBn	QCn	QDn	L
H	L	L	X	X	X	X	X	X	X	QA0	QB0	QC0	QD0

X: Don't Care : Don't Care
a ~ d : The level of steady state input voltage at input A ~ D respectively
QA0 ~ QD0 : No charge
QAn ~ QDn : The level of QA, QB, QC, respectively, before the most recent positive transition of the clock.

Chronogramme

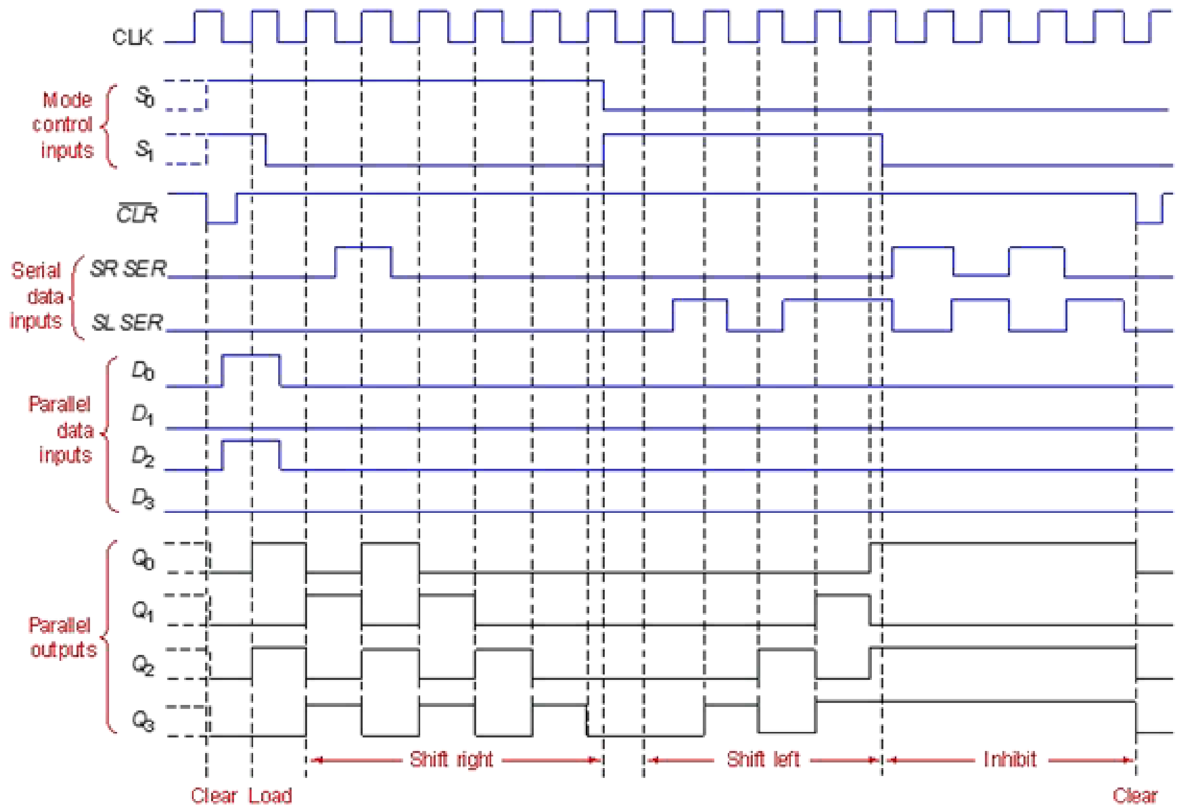


Figure 4.9 : *Echantillon de chronogramme du registre universel 74LS194A.*

Exemple d'application des registres : Transfert de données numériques par liaison série

Le transfert de l'information au sein des systèmes numériques est réalisé sur des mots binaires parallèles. Or lors du transport de l'information sur de longues distances, la transmission série est utilisée. Les informations sont envoyés sous forme série. Il faut donc avant l'émission, une conversion parallèle / série et à la réception, la conversion série / parallèle pour récupérer l'information. La figure 4.10 illustre le principe de transmission série des données numériques.

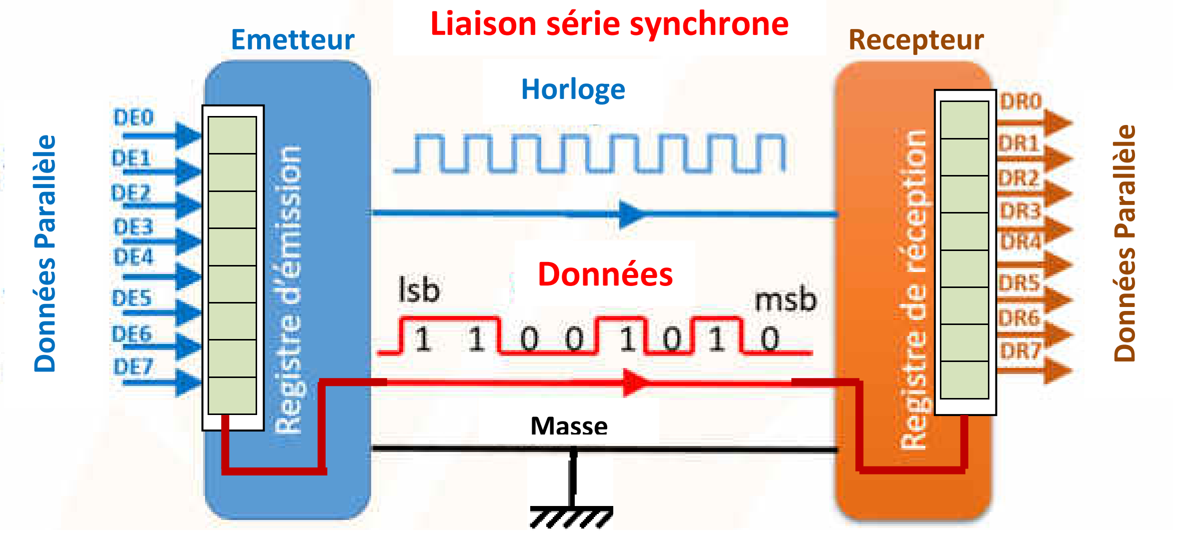


Figure 4.10 : *Principe de transmission série des données numériques.*

En savoir +

La synchronisation peut être obtenue de 2 manières :

- Soit en envoyant le signal d’horloge par un fil séparé ;
- Soit au niveau du récepteur, en reconstituant le signal d’horloge à partir des changements d’états du signal reçu.

4.5. Exercices

EXERCICE 01: QCM (avec explications)

- **Q1:** Chaque étage d'un registre à décalage représente une capacité de stockage de
    - ○ **01 bit,**
    - ○ **02 bits,**
    - ○ **04 bits, 08 bits.**
- **Q2:** Dans quel type de registre à décalage la sortie (Q ou $\overline{Q}$) d'un étage n'est pas connectée à l'entrée de l'étage voisine?
    - ○ Entrée parallèle / Sortie série (PISO),
    - ○ Entrée série / Sortie parallèle (SIPO),
    - ○ Entrée série / Sortie série (SISO),
    - ○ Entrée parallèle / Sortie parallèle (PIPO).
- **Q3:** Soit un registre à décalage de 08 bits Entrée série / Sortie parallèle (SIPO). Si ce registre ne dispose pas d'une entrée remise à zéro (RAZ), combien de cycles d'horloges sont nécessaires pour qu'un zéro à son entrée garantissent une telle initialisation ?
    - ○ **1,**
    - ○ **4,**
    - ○ **7,**
    - ○ **8,**

- ○ **9.**

✓ **<u>Q4:</u>** Soit un registre à décalage de 08 bits Entrée parallèle / Sortie série (PISO). Si ce mot (de 04 bits) à été chargé dans ce registre, combien de cycles d'horloges sont nécessaires avant que le premier bit du mot chargé apparaisse sur la sortie ?

- ○ **0,**
- ○ **1,**
- ○ **2,**
- ○ **3.**

✓ **<u>Q5:</u>** Avec une fréquence d'horloge de 200Khz, huit bits peuvent être chargés séquentiellement dans un registre à décalage en :

- ○ **4 μs,**
- ○ **40 μs,**
- ○ **400 μs,**
- ○ **40 ms.**

✓ **<u>Q6:</u>** Un registre à décalage de 08 bits Entrée série / Sortie série (SISO) est utilisé avec une fréquence d'horloge de 2 MHZ pour réaliser un retard de temps de :

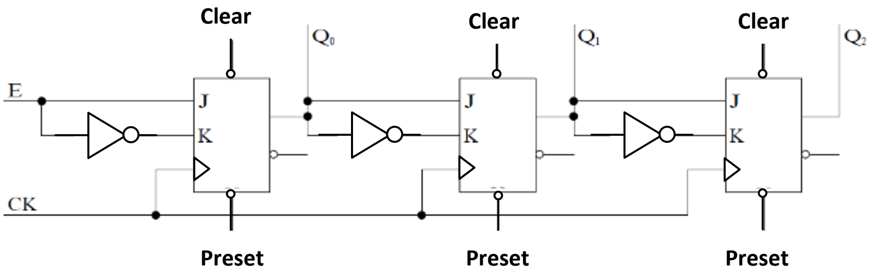
- ○ **16 μs,**
- ○ **8 μs,**
- ○ **4 μs,**
- ○ **2 ms.**

✓ **<u>Q7:</u>** Quand un registre à décalage de 08 bits Entrée série / Sortie série (SISO) est utilisé pour réaliser un retard de temps de 20 s, la fréquence de d'horloge doit être réglée à :

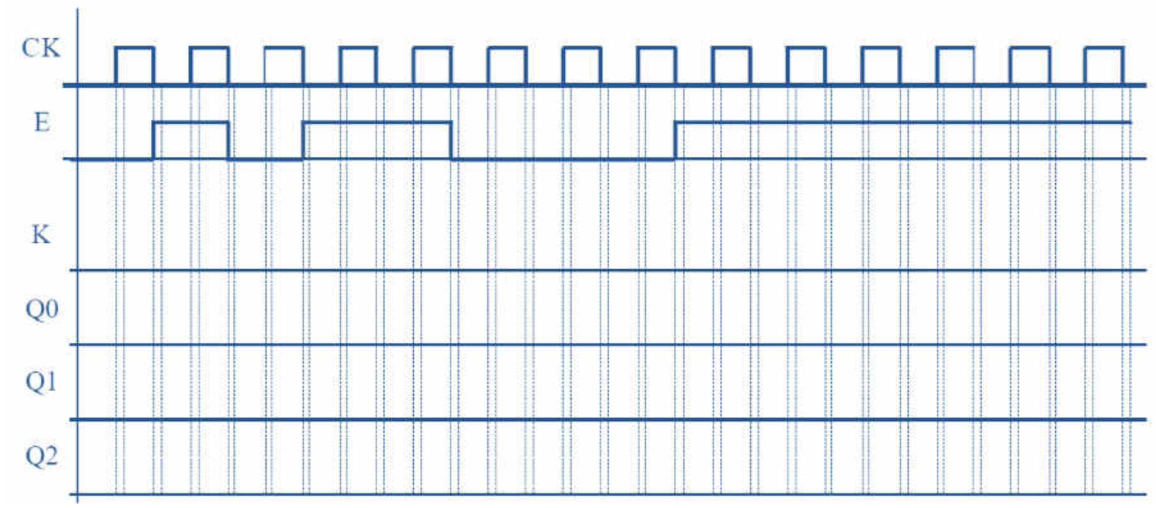
- ○ **40 KHZ,**
- ○ **50 KHZ,**
- ○ **400 KHZ,**
- ○ **500 KHZ.**

EXERCICE 02:

Soit le montage ci-dessous comprend 3 bascules JK du circuit intégré 4027.

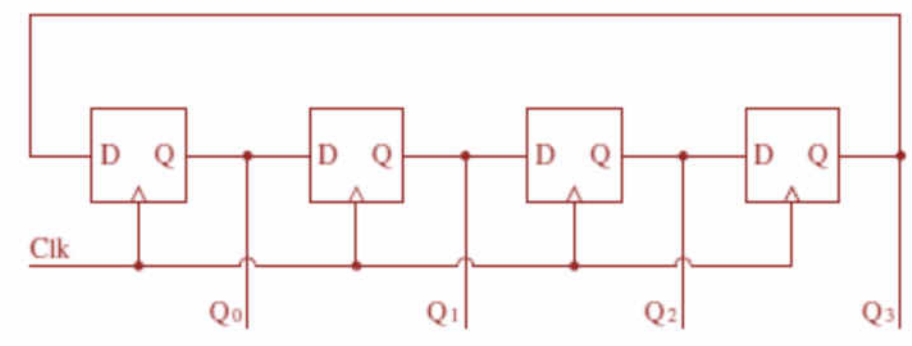


- ✓ Compléter le chronogramme ci-dessous.



EXERCICE 03:

On considère le circuit de la figure suivante.



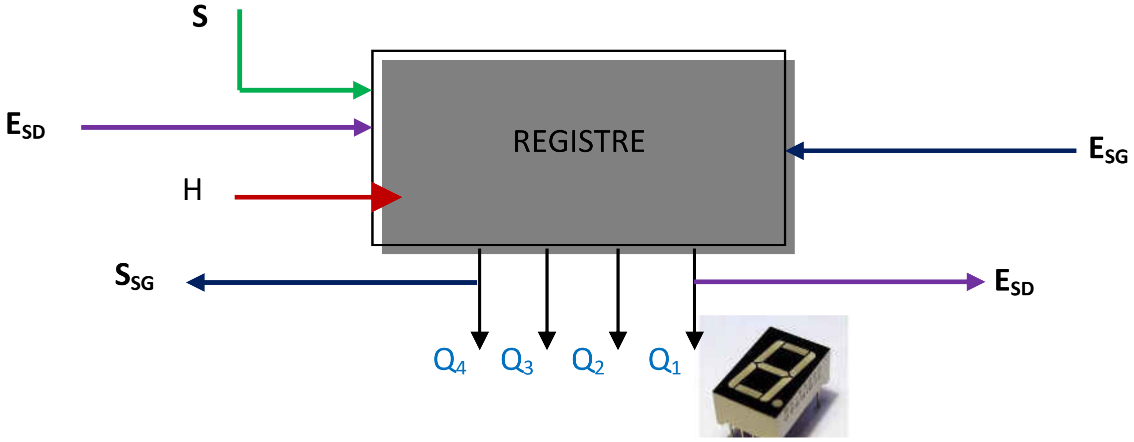
Supposez que, à l'état initial, les bascules ont été chargées avec des valeurs aléatoires, lors de la stabilisation des boucles de portes.

1. Quelles sont les valeurs des bascules après un top d'horloge ? Après deux tops d'horloge ?
2. Quelle est la fonction de ce circuit ?

EXERCICE 04 :

On veut réaliser un registre à décalage bidirectionnel à entrées série $\mathbf{E_{SD}}$ ou $\mathbf{E_{SG}}$ selon le sens du décalage. Les sorties séries sont $\mathbf{S_{SD}}$ ou $\mathbf{S_{SG}}$, les sorties parallèles sont $\mathbf{Q_1}$, $\mathbf{Q_2}$, $\mathbf{Q_3}$ et $\mathbf{Q_4}$. On utilise quatre bascules **D** à front montants.

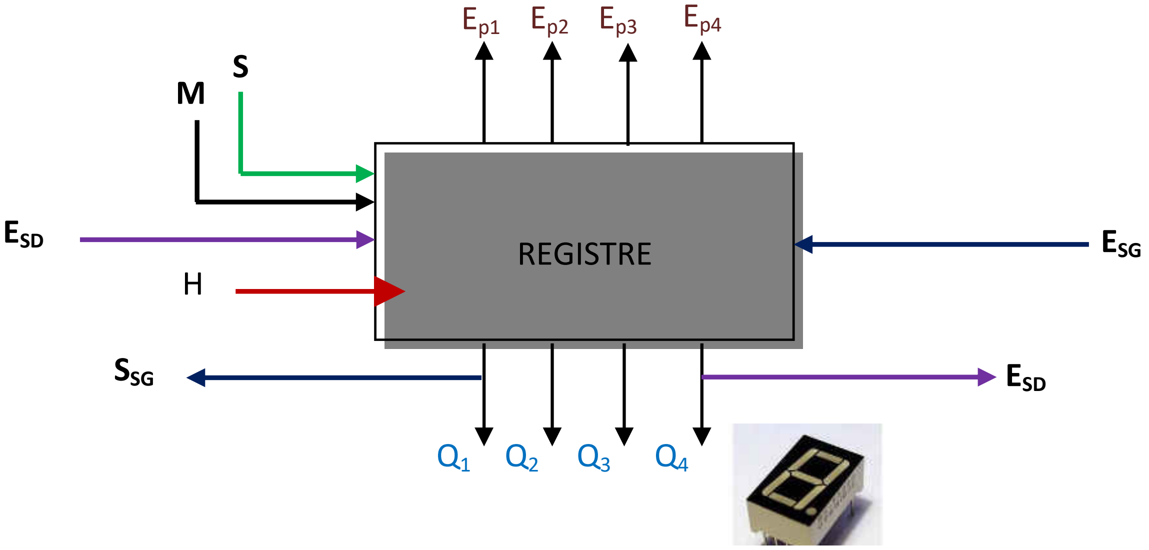
Le sens du décalage est commandé par le signal **S** (1 pour décaler à droite, 0 pour décaler à gauche).



- ✓ Donner les équations des signaux d'entrée des bascules D_i (i = 1, 2, 3, 4).
- ✓ Donner le schéma du circuit en utilisant des portes NAND en plus des bascules.

EXERCICE 05 :

On veut réaliser un registre dit '***universel***' comme indique cette figure



- Les entrées parallèles sont notées **E_i** (i = 1, 2, 3, 4),
- Entrées série **E_{SD}** et **E_{SG},**
- Les sorties parallèles sont **Q_1, Q_2 , Q_3** et **Q_4**.

On sélectionne le mode parallèle par **M = 1**, et le mode série par **M = 0**. Le sens de décalage dépend de **M** et du signal de commande **S** (1 pour décaler à droite, 0 pour décaler à gauche).

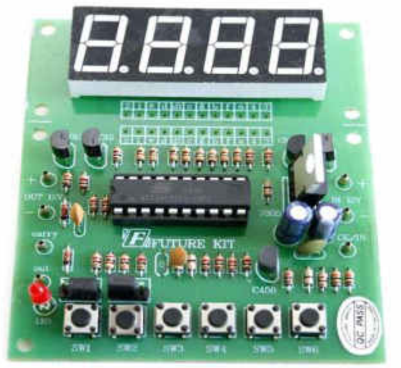
On réalisera ce registre à l'aide de quatre bascules **D** à front montants et des portes NAND.

EXERCICE 06 :

On désire utiliser le registre à décalage universel 74LS194 pour un chargement parallèle des données et une lecture série de ces données avec décalage vers la droite:

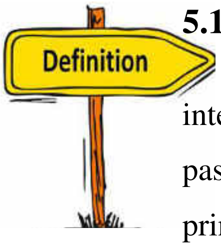
- ✓ Remplir le tableau de fonctionnement ci-dessous pour permettre un chargement parallèle du mot binaire 1011 dans le registre 74LS194 ?
- ✓ En supposant que le chargement du mot binaire a été effectué, donnez le câblage du 74LS194 pour réaliser un décalage rotatif de la gauche vers la droite?

Entrées								**Sorties**			
Clear	Mode		**CLK**	Parallèle				Q_D	Q_C	Q_B	Q_A
	S1	**S0**		**D**	**C**	**B**	**A**				
?	?	?	?	?	?	?	?	?	?	?	?



CHAPITRE
5

LES COMPTEURS



5.1. Les compteurs

Par définition, un compteur est un registre constitué d'un ensemble de bascules interconnectées de façon particulière dont les impulsions (données) d'entrée se propagent pas à pas suivant **une loi propre** au type de compteur. La figure 5.1 illustre les cinq caractéristiques principales des compteurs.

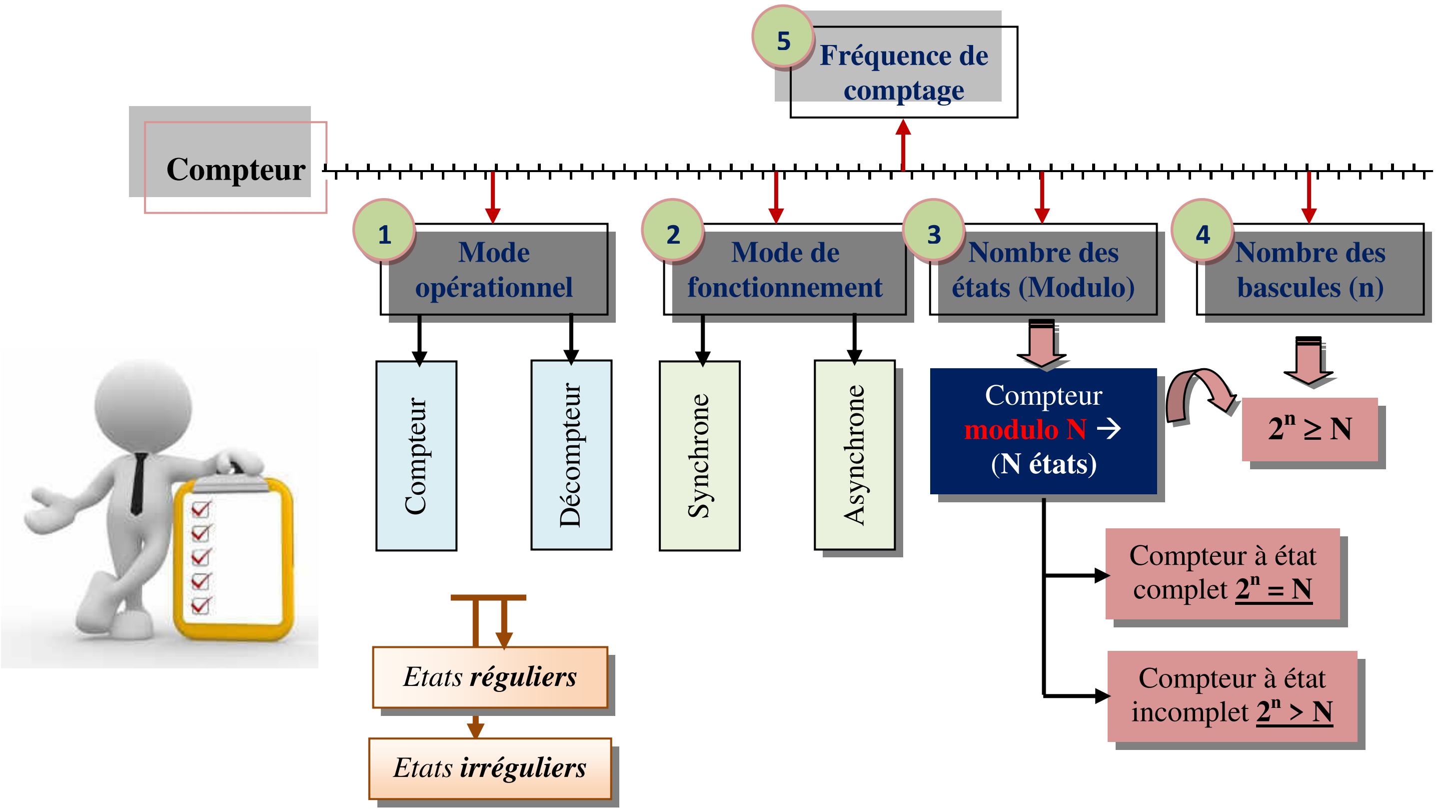
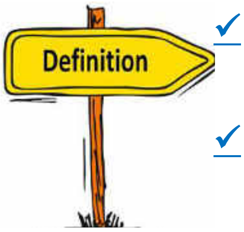


Figure 5.1 : Caractéristiques principales des compteurs.



- ✓ Le **modulo** d'un compteur est le **nombre des différents états logiques** que les sorties peuvent occuper.
- ✓ Un compteur modulo **2^n** est constitué de **n** bascules et peut compter de **0** jusqu'à **2^n-1**.

Exemples:

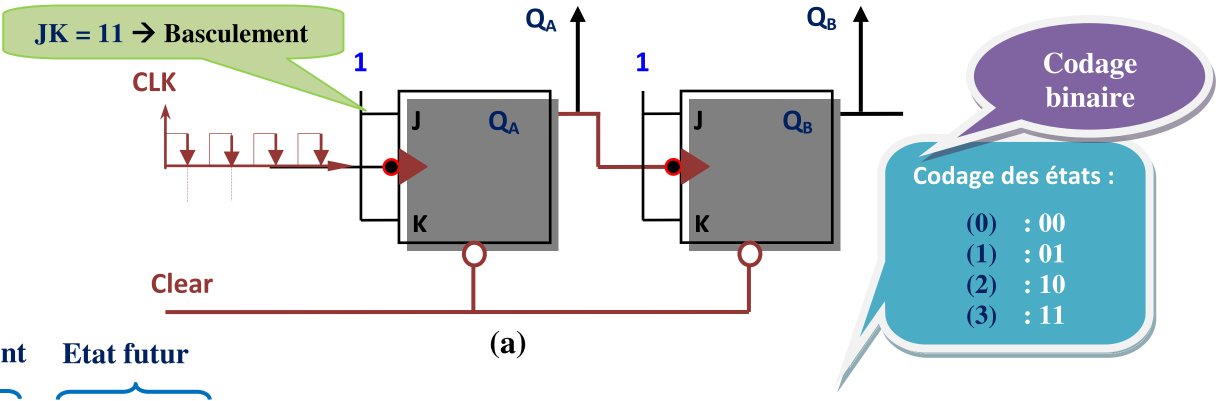
- Compteur modulo 16 (N = 16) ⇒ **$2^n \geq 16$ ⇒ n = 4 bascules.**
- Compteur modulo 10 (N = 10) ⇒ **$2^n \geq 10$ ⇒ n = 4 bascules.**
- Compteur modulo 6 (N = 6) ⇒ **$2^n \geq 6$ ⇒ n = 3 bascules.**
- Compteur modulo 5 (N = 5) ⇒ **$2^n \geq 5$ ⇒ n = 3 bascules.**

5.1.1. Compteurs asynchrones à cycle complet

La réalisation d'un compteur asynchrone **progressif** consiste à

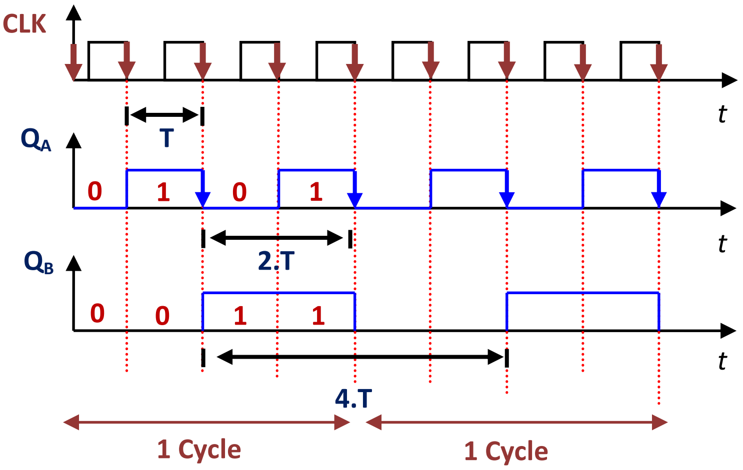
- ✓ Mettre en cascade des bascules (assurant la fonction diviseur par 2),
- ✓ Détecter la combinaison de remise à zéro, puis à
- ✓ L'appliquer à l'entrée de forçage à zéro de chaque bascule (**Clear**).

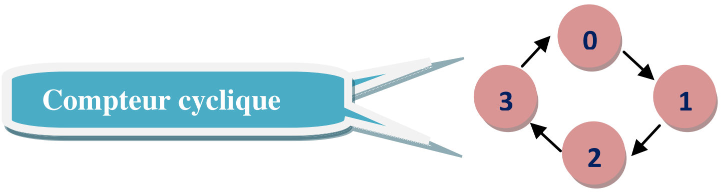
✓ **Compteur asynchrone modulo 4 :** ((**N = 4**) ⇒ **$2^n \geq 4$** ⇒ **n = 2** bascules) → Compteur binaire asynchrone de 2 bits → Compteur à état complet.



(a)

	Etat présent		Etat futur	
N	Q_B	Q_A	Q^+_B	Q^+_A
0	0	0	0	1
1	0	1	1	0
2	1	0	1	1
3	1	1	0	0





(b)

Figure 5.2 : *Structure, table des états et chronogramme d'un compteur asynchrone modulo 4.*

✓ Les 4 états représentés par les nombres **0**, **1**, **2** et **3** en binaires. Ce compteur part de l'état initial (**0 : 00**), compte en binaire pur jusqu'à l'état final (**3 : (11)**) en croissant, revient o (0 : 00) et recommence. On parle d'un **compteur progressif**.

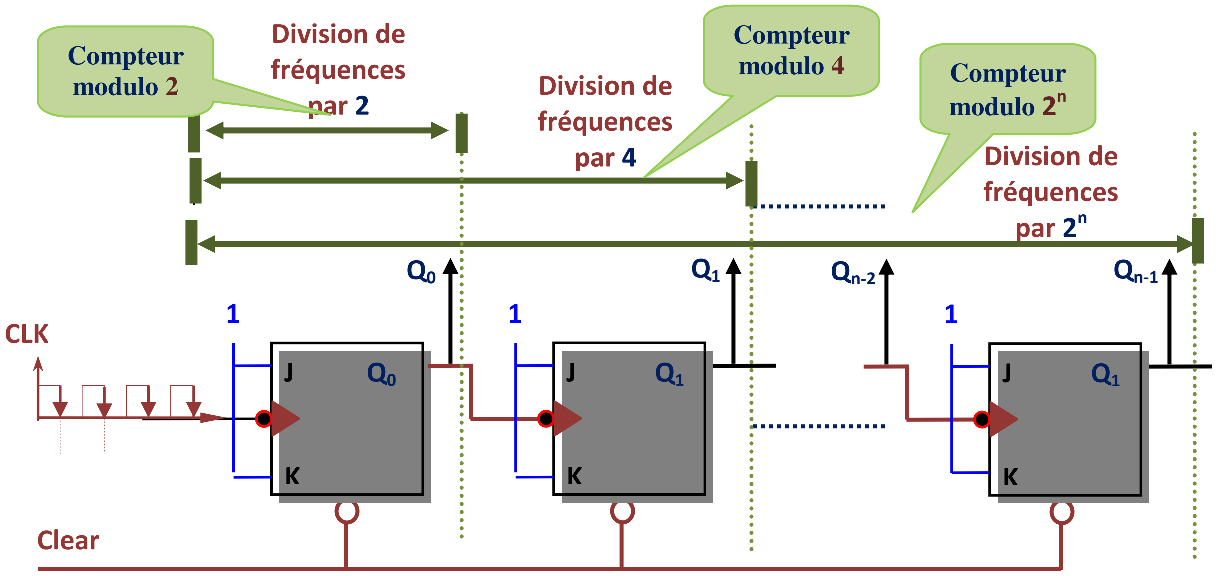
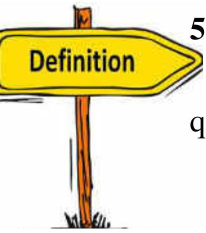


Figure 5.3 : *Diviseur de fréquences.*

Le tableau 5.1 illustre les différentes possibilités offertes par le montage de la figure précédente où les bascules sont interconnectées l'une après l'autres.

Table 5.1 : *Compteurs progressifs.*

Compteur ***progressif***	Nombre des états (N)	Nombre de bascules (n)	Etats décimaux	Type de diviseur
Compteur modulo 2	2	1	(0,1)	Diviseur de fréquences par 2
Compteur modulo 4	4	2	(0,1,2,3)	Diviseur de fréquences par 4
Compteur modulo 8	8	3	(0,1,2,3,…,7)	Diviseur de fréquences par 8
Compteur modulo 16	16	4	(0,1,2,…, 15)	Diviseur de fréquences par 16
Compteur modulo 32	32	5	(0,1,2,…,31)	Diviseur de fréquences par 32
Compteur modulo 64	64	6	(0,1,2,…,63)	Diviseur de fréquences par 64
---	---	---	---	---
Compteur modulo N	N	2^n = N	(0,1,2,…,N-1)	**Diviseur de fréquences par N**



5.1.2. Compteurs asynchrones à états (cycle) incomplets

Par définition, un compteur à cycle incomplet est un compteur qui n'exploite que quelques états stables parmi les 2^n états offerts par les **n** bascules que constitue ce compteur.

- ✓ Pour obtenir un compteur asynchrone à états incomplets (**M # 2^n**), où **N** est le nombre des états et **n** est le nombre des bascules.
- ✓ On prend la même structure que le compteur modulo **2^n**, sauf l'application du signal **Clear** (pour toutes les bascules du montage) lorsque le compteur arrive à l'état **M**.

⌖ Compteurs asynchrones modulo 11 :

Pour obtenir un compteur asynchrone modulo 11 (**11** états : 0,1,2, …, 10), on prend la même structure que le montage du compteur modulo **16** (**16 = 2^4**), puis on applique le signal externe **Clear** lorsque le compteur arrive à l'état final (**10** = (**1010**)) pour remise à l'état initial ((**0**) = **0000**) toutes les bascules, comme indique le tableau ci-dessous.

Table 5.2 : Table des états d'un compteur asynchrone modulo 11.

Compteurs asynchrones modulo **16**

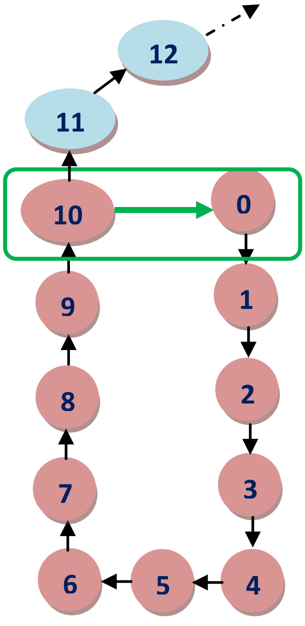
Compteurs asynchrones modulo **10**

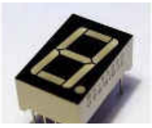
	Etat présent	**Etat futur**
Etat décimal	$Q_DQ_CQ_BQ_A$	$Q^+_D\ Q^+_C\ Q^+_B\ Q^+_A$
0	0000	0001
1	0001	0010
2	0010	0011
3	0011	0100
4	0100	0101
5	0101	0110
6	0110	0111
7	0111	1000
8	1000	1001
9	1001	1010
10	1010	0000
11	1011	1100
12	1100	1101
13	1101	1110
14	1110	1111
15	1111	000

Etat Initial

Etat final

$$\text{Clear} = Q_D\bar{Q}_CQ_B\bar{Q}_A$$





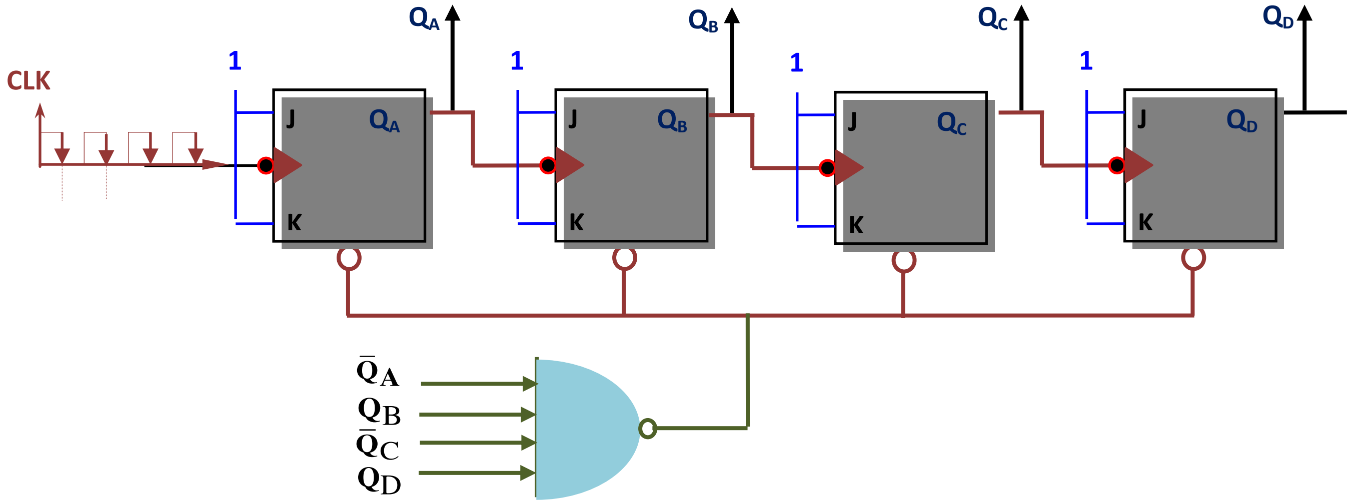


Figure 5.4 : Compteur asynchrone modulo 11 (Compteur self-stopping).

En savoir+

- ✓ Pour la réalisation des compteurs *self-stopping*, deux méthodes sont possibles :
  - Utiliser des bascules possédant les entrées asynchrones (**Clear**, **Preset**) pour bloquer les états indésirables (voir l'exemple précédent) ou
  - Exploiter les tables de transitions (**compteurs synchrones**).
- ✓ Pour la réalisation de Décompteurs asynchrones (**Compteur régressifs**), on garde la même structure de compteurs asynchrones, sauf la synchronisation s'effectue par **les sorties complémentaires** des bascules, comme indique la figure ci-dessous.

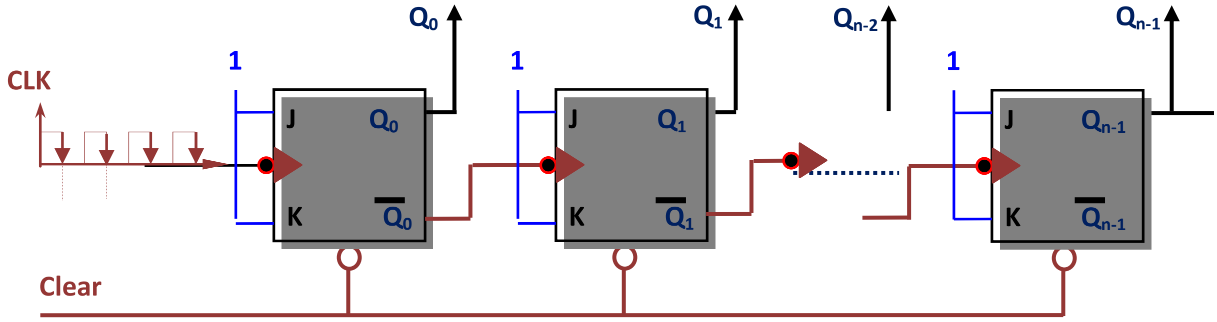


Figure 5.5 : Décompteur asynchrone (Compteur régressifs).

5.1.3. Compteurs synchrones

Dans ce type de compteurs, le changement des états des bascules s'effectue simultanément à chaque Top d'horloge.

5.1.3.1. Compteurs synchrones à cycle complet : N = 2^n

LEAD BY EXAMPLE

Prenons l'exemple d'un compteur modulo 8 synchrone pour mieux comprendre le fonctionnement de ce type des compteurs.

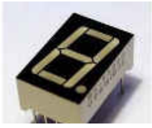
✍ **N = 8 états ⇒ n = 3** bascules (**$Q_CQ_BQ_A$**) sont nécessaires pour réaliser ce compteur. Le pricipe de fonctionnement de ce compteur est illustré par la table des états ci-dessous.

Table 5.3 : *Table des états d'un compteur synchrone modulo 8.*

	Etat présent	Etat futur	Entrées des bascules JK							Entrées des bascules D		
Etat décimal	$Q_CQ_BQ_A$	$Q^+_C\ Q^+_B\ Q^+_A$	C		B		A			BCA		
			J_C	K_C	J_B	J_A	D_C	D_B		D_C	D_B	D_A
0	000	001	0	x	0	1	1	x		0	0	1
1	001	010	0	x	1	x	x	1		0	1	0
2	010	011	0	x	x	1	1	x		0	1	1
3	011	100	1	x	x	x	x	1		1	0	0
4	100	101	x	0	0	1	1	x		1	0	1
5	101	110	x	0	1	x	x	1		1	1	0
6	110	111	x	0	x	1	1	x		1	1	1
7	111	000	x	1	x	x	x	1		0	0	0

Les entrées J et K des bascules sont obtenues par utilisation de la table de Karnaugh, avec :

$$\begin{cases} \mathbf{J_A = K_A = 1} \\ \mathbf{J_B = K_B = Q_A} \\ \mathbf{J_C = K_C = Q_A \cdot Q_B} \end{cases}$$



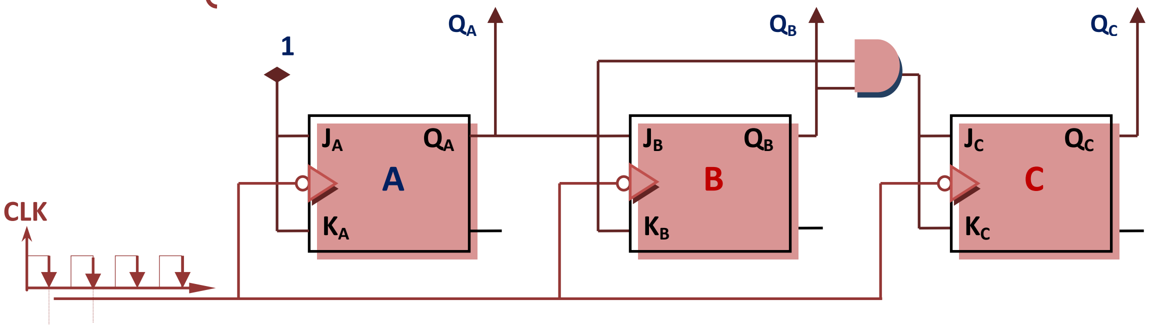


Figure 5.6 : *Logigramme d'un Compteur synchrone modulo 8 à base des bascules JK.*

De la même manière, on peut concevoir ce compteur à base des bascules D, où Les entrées D des bascules sont obtenues par utilisation de la table de Karnaugh, avec :

$$\begin{cases} D_A = \bar{Q}_A \\ D_B = Q_A \oplus Q_B \\ D_C = Q_C \oplus (Q_B \cdot Q_A) \end{cases}$$

5.1.3.2. Compteurs synchrones à cycle incomplet : $2^n \geq N$

L'exemple suivant illustre le fonctionnement d'un compteur synchrone à cycle incomplet où **$2^n \geq N$**

Exemple : Compteur synchrone modulo 6

✍ **N = 6 ⇒ n = 3** Bascules (**$Q_CQ_BQ_A$**) sont nécessaires pour réaliser ce compteur.

Table 5.4 : *Table des états d'un compteur synchrone modulo 6.*

	Etat présent	Etat futur	Entrées des bascules JK							Entrées des bascules D		
Etat décimal	$Q_CQ_BQ_A$	$Q^+_C\ Q^+_B\ Q^+_A$	C		B		A			BCA		
			J_C	K_C	J_B	J_A	D_C	D_B		D_C	D_B	D_A
0	000	001	0	x	0	1	1	x		0	0	1
1	001	010	0	x	1	x	x	1		0	1	0
2	010	011	0	x	x	1	1	x		0	1	1
3	011	100	1	x	x	x	x	1		1	0	0
4	100	101	x	0	0	1	1	x		1	0	1
5	101	000	x	0	1	x	x	1		1	1	0
6	110	xxx	x	x	x	x	x	x		x	x	x
7	111	xxx	x	x	x	x	x	x		x	x	x

Les entrées D des bascules sont obtenues par utilisation de la table de Karnaugh, avec :

$$\begin{cases} D_A = \bar{Q}_A \\ D_B = \bar{Q}_A \cdot Q_B + Q_A \cdot \bar{Q}_B \cdot Q_C \\ D_C = Q_A \cdot Q_B + \bar{Q}_A \cdot Q_C \end{cases}$$

Références techniques : on peut retrouver les compteurs dans les circuits intégrés comme indique le tableau ci-dessous.

Table 5.5 : *Compteurs en circuits intégrés.*

Circuit intégré	Fonction
7490	Compteur binaire asynchrone modulo 10
7492	Compteur binaire asynchrone modulo 12
7493	Compteur binaire asynchrone modulo 16
74163	Compteur binaire synchrone modulo 16
74160	Compteur binaire synchrone modulo 10

Exemple : Le circuit intégré **74163** comprenant **04** bascules du type JK .

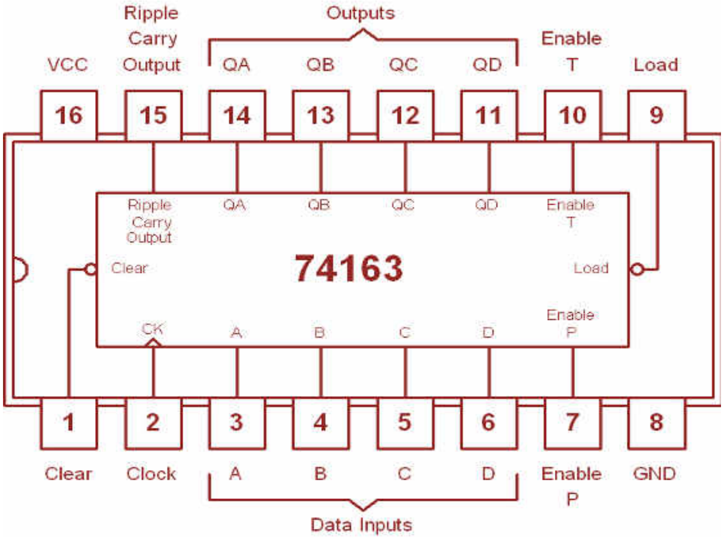
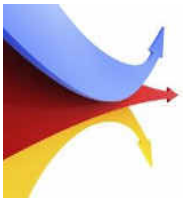


Figure 5.7 : *Brochage du circuit intégré* ***74163***.

5.2. Compteurs spécifiques : Les compteurs à registres à décalage

Un compteur à registre à décalage est un registre dont la sortie série est réacheminée à son entrée série pour produire des séquences spéciales. Les trois types les plus rencontrés sont :

- ✓ Compteurs à séquences irrégulières.
- ✓ Compteur Johnson.
- ✓ Compteur en anneau.



5.2.1. Compteurs à séquences irrégulières

Ce type de compteurs permet de produire une séquence binaire irrégulière illustrée par un diagramme d'état quelconque.

Exemple : Créer un compteur permettant de produire la séquence binaire irrégulière illustrée par le diagramme d'état ci-dessous, en utilisant les bascules JK.

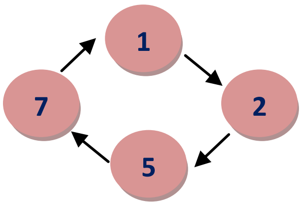
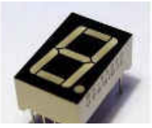


Table 5.6 : *Table des états d'un compteur synchrone à séquences irrégulières.*

	Etat présent	Etat futur	Entrées des bascules JK					
Etat décimal	$Q_CQ_BQ_A$	$Q^+_C\ Q^+_B\ Q^+_A$	C		B		A	
			J_C	K_C	J_B	J_A	D_C	D_B
0	000	xxx	x	x	x	x	x	x
1	001	010	0	x	1	x	x	1
2	010	101	0	x	x	1	1	x
3	011	xxx	x	x	x	x	x	x
4	100	xxx	x	x	x	x	x	x
5	101	111	x	0	1	x	x	1
6	110	xxx	x	x	x	x	x	x
7	111	001	x	0	1	x	x	1

Les entrées J et K des bascules sont obtenues par utilisation de la tables de Karnaugh, avec :



$$\begin{cases} J_A = 1 \\ K_A = \overline{Q}_C \\ J_B = K_B = 1 \\ J_C = K_C = Q_B \end{cases}$$

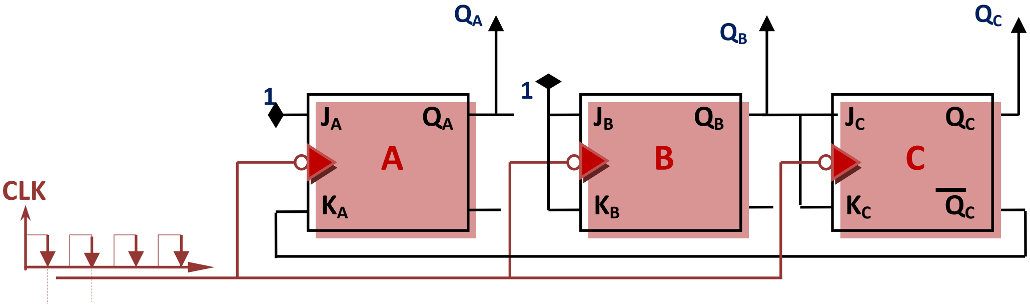


Figure 5.8 : *Compteur synchrone à séquences irrégulières à base des bascules JK.*

5.2.2. Compteur Johnson

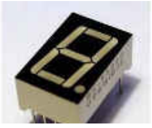
Compteur Johnson est un compteur modulo **N = 2*n**, où **n** est le nombre des bascules, dont le complément de la sortie de la dernière bascule sera retro-couplée vers l'entrée de la première bascule.

Exemple : Compteur Johnson à 04 bits → 2*4 = 8 états, comme indique la table ci-dessous.

Table 5.7 : Table des états d'un compteur Johnson à 04 bits.

Etats

Top d'horloge	$Q_AQ_BQ_CQ_D$
0	0000
1	1000
2	1100
3	1110
4	1111
5	0111
6	0011
7	0001



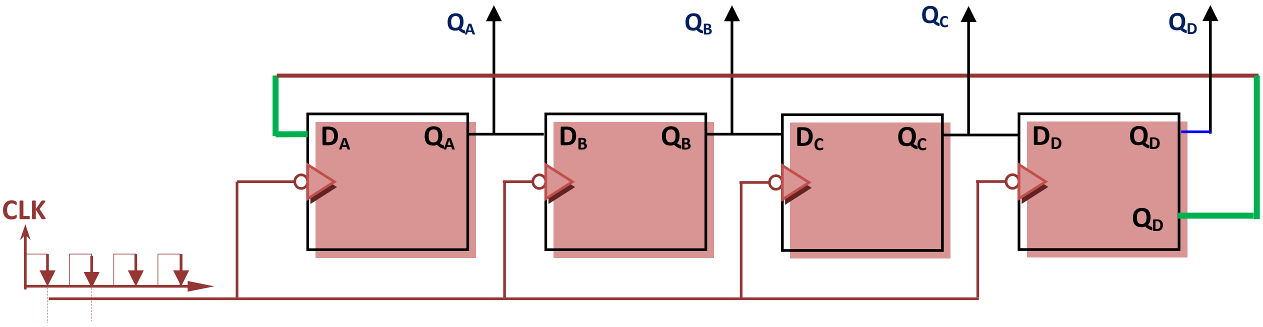


Figure 5.9 : Compteur Johnson à 04 bits à base des bascules D.

5.2.3. Compteur en anneau

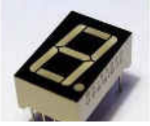
Compteur en anneau est un compteur, où chaque état est représenté par une bascule, dont la sortie de la dernière bascule est retro-couplée vers l'entrée de la première bascule.

Exemple : Compteur en anneau à **04 bits → 4 états**, comme indique-la table ci-dessous.

Table 5.8 : Table des états d'un compteur en anneau à 04 états.

Etats

Top d'horloge	$Q_AQ_BQ_CQ_D$
0	**1000**
1	0100
2	0010
3	0001



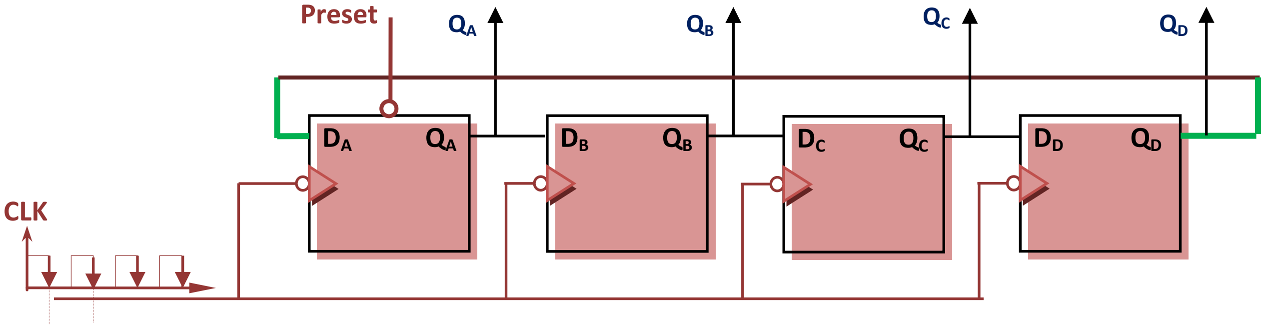


Figure 5.10 : *Compteur en anneau à 04 états à base des bascules D.*

Comment obtenir un compteur ayant un modulo plus élevé ?

La ***mise en cascade*** des compteurs permet d'obtenir un modulo plis élevé. Le modulo total des compteurs montés en cascade est égal au produit des modulos individuels.

- **Compteur modulo 32**
- **Compteur modulo 1536**
- **… etc.**

Table 5.9 : *Exemples de compteurs modulo élevés.*

Type	Division
Compteur Modulo 32	**4 x 8 = 32**
Compteur modulo 1536	**4 x 12 x 16 = 1536**
Division en valeurs de 2, 4, 8 et 16 afin d'avoir le modulo désiré.	

Principe

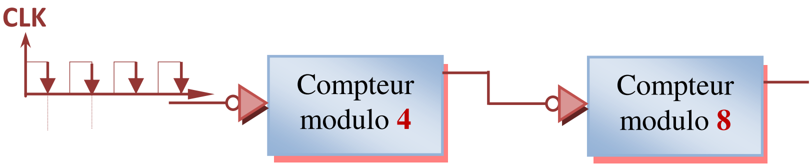


Figure 5.11 : *Compteur modulo 32.*

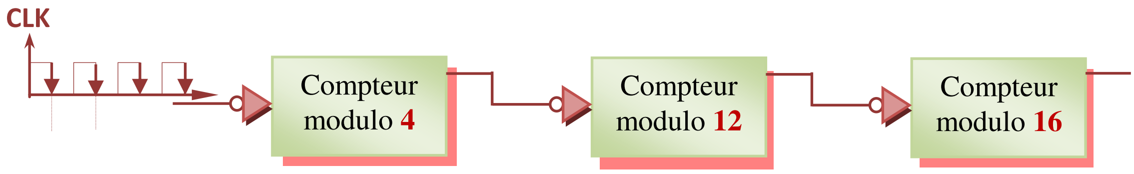


***Figure 5.11 :** Compteur modulo **1536**.*

5.4. Exercices

EXERCICE 01 :

- Effectuer la synthèse d’un compteur synchrone modulo 6 à l’aide de bascules JK fonctionnant sur fronts descendants.
- Réaliser de la même manière un compteur asynchrone modulo 5.
- Réaliser un compteur asynchrone avec des bascules JK fonctionnant sur fronts montants réalisant le cycle suivant :

0, 1, 2, 3, 7, 8, 9, 10, 15, 0, …

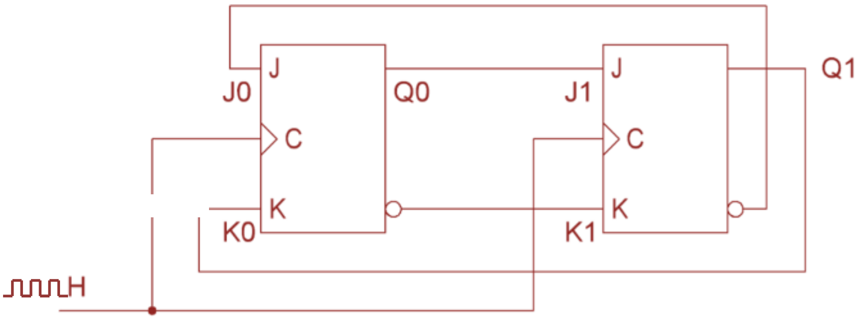
EXERCICE 02 :

Un compteur pair module 16 compte de 0, 2, 4, 6, 8, 10, 12, 14, 0, 2....

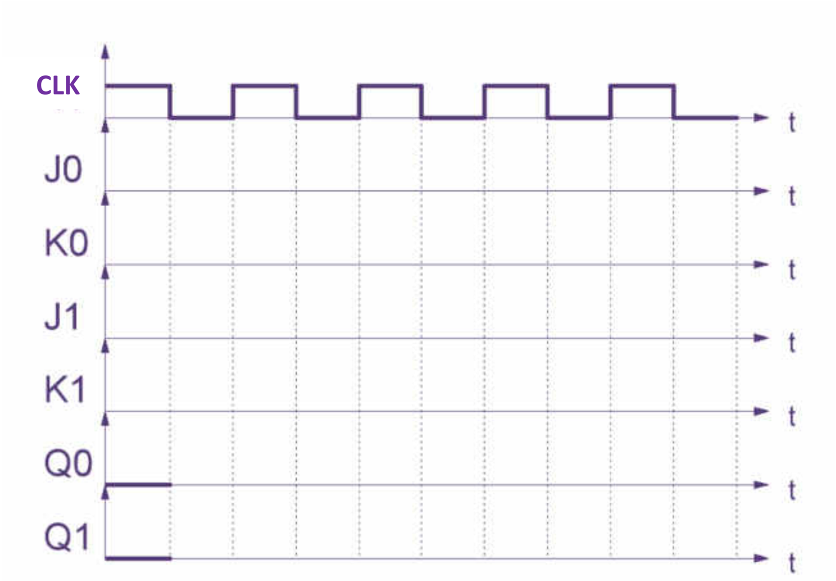
- Donner la table d'états du compteur.
- Que remarquer vous ?
- Réaliser le schéma à l'aide des bascules JK.

EXERCICE 03 :

- Donner les équations de J0, K0, J1, K1.



⌖ Remplissez le chronogramme en fonction du montage suivant :



EXERCICE 04 :

On veut réaliser un compteur des heures modulo 24.

- ✓ Combien de bascules JK, on doit utiliser?
- ✓ Donner l'équation de CL pour remettre le compteur à zéro.
- ✓ Réaliser un compteur modulo 24

EXERCICE 05 :

Un compteur déformé compte de 0, 1, 2, 3, 8, 9, 10, 11, 0, 1, 2

- ✓ Donner la table d'états du compteur.
- ✓ Réaliser le schéma à l'aide des bascules JK.

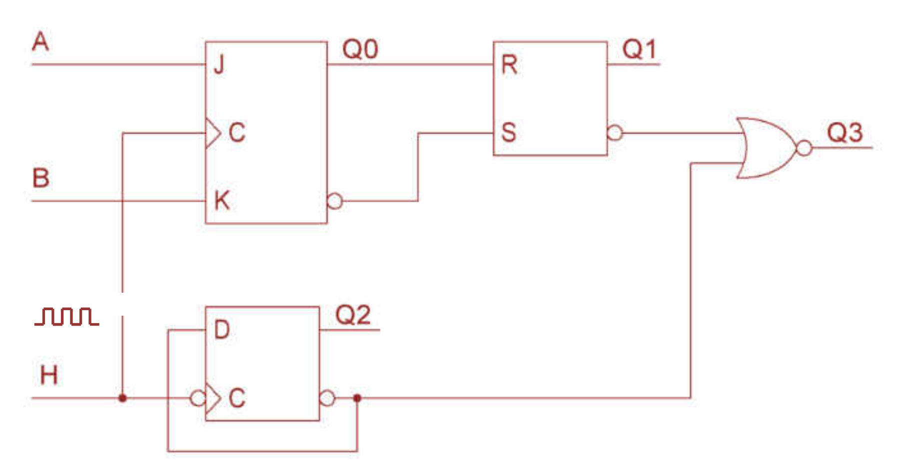
EXERCICE 06 :

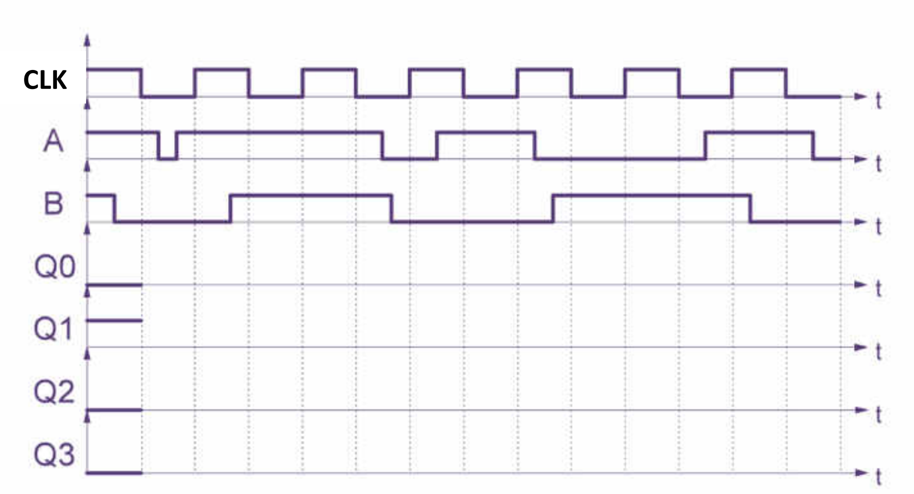
Un compteur déformé compte de 0, 1, 4, 5, 2, 3, 6, 7, 8, 9, 12, 13, 10, 11, 14, 15, 0, 1, ...

- ✓ Donner la table d'états du compteur.
- ✓ Que remarquer vous?
- ✓ Réaliser le schéma à l'aide des bascules JK.

EXERCICE 07 :

- Donner les équations de D, R,S, Q3
- Remplissez le chronogramme en fonction du montage suivant :



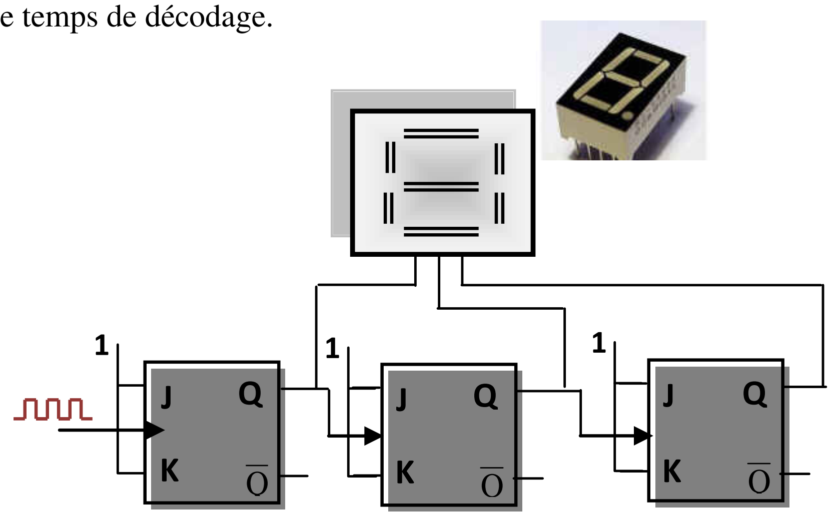


EXERCICE 08 :

Pour afficher l'état de tels compteurs, prenons l'affichage à sept segments à décodeur incorporé. Quelle est la fréquence maximale de l'utilisation.

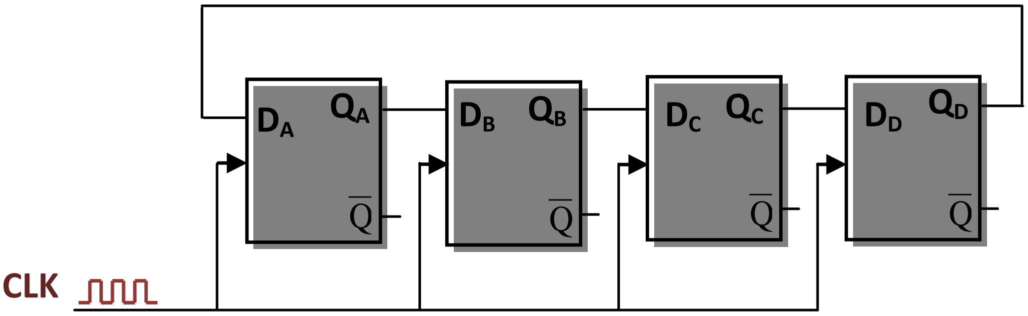
T_P = 50 ns, Le temps de propagation dans une bascule.

T_d = 100 ns, Le temps de décodage.



EXERCICE 09 :

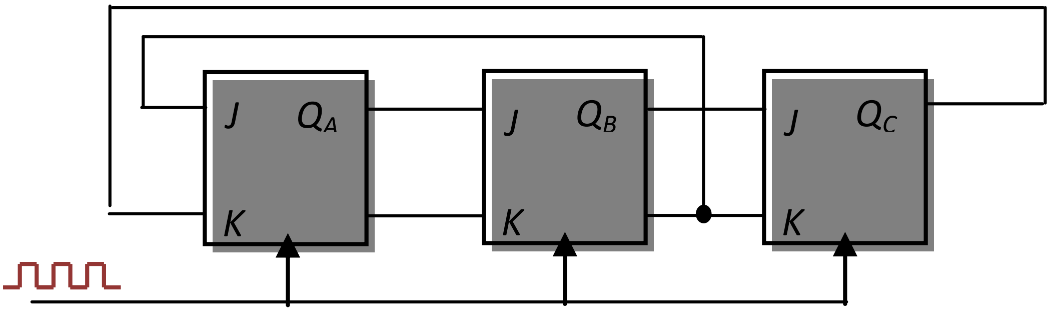
- Etant donné le compteur en anneau de la figure ci dessous. Etablir sa table de transitions et montrer qu'il possède quatre états distincts.



- Représenter l'allure des signaux Q_A, Q_B, Q_C et Q_D en fonction du temps (CLK : signal d'horloge de forme carré et de période *T*). En supposant que l'état initial $(Q_AQ_BQ_CQ_D)$= (1000).
- Sachant que les bascules D utilisées possèdent des entrées de mise à 1 (Preset) et de remise à 0 (Clear). Comment peut-on démarrer ce compteur à partir de l'état $(Q_AQ_BQ_CQ_D)$= (0010).
- Pour un compteur en Anneau modulo N, donner le nombre de bascules nécessaires. Comparer ce nombre avec celui obtenu pour un compteur binaire normal.
- Que devient ce compteur lorsque la sortie $\overline{Q_D}$ est reliée à l'entrée D de la première bascule. Représenter l'allure des signaux Q_A, Q_B, Q_C et Q_D et CLK à partir de l'état initial $(Q_AQ_BQ_CQ_D)$= (0000).

EXERCICE 10 :

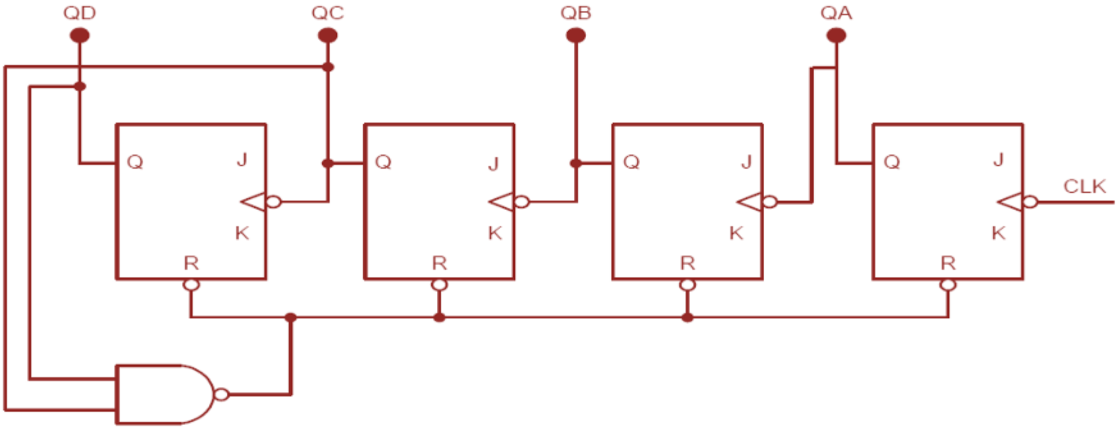
Soit le compteur suivant :

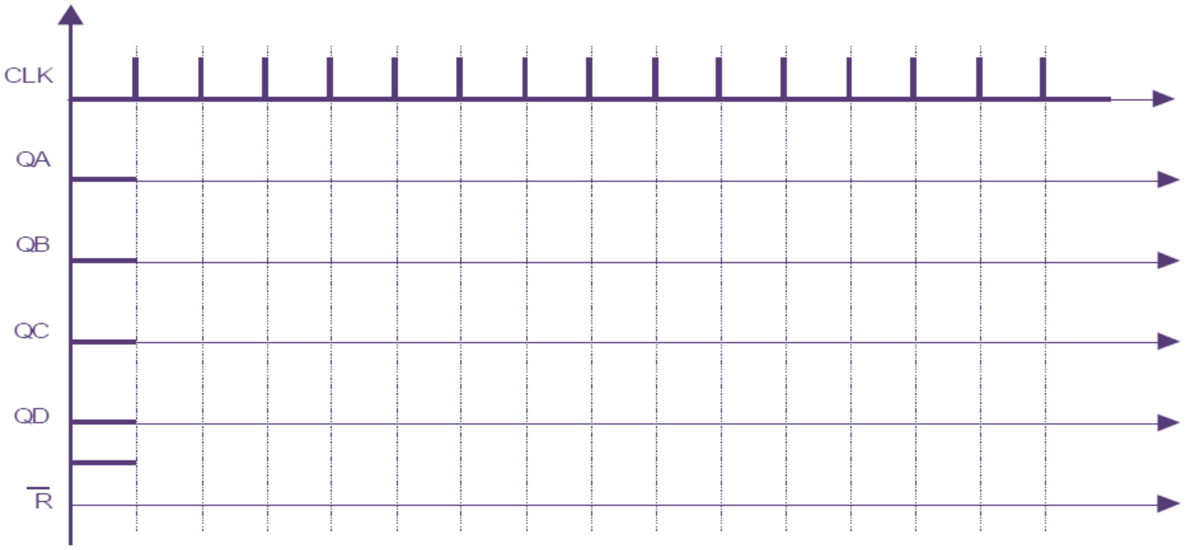


- ✓ Donner la séquence de ces états à partir de l'état 101.
- ✓ Même question à partir de l'état 000.

EXERCICE 11 :

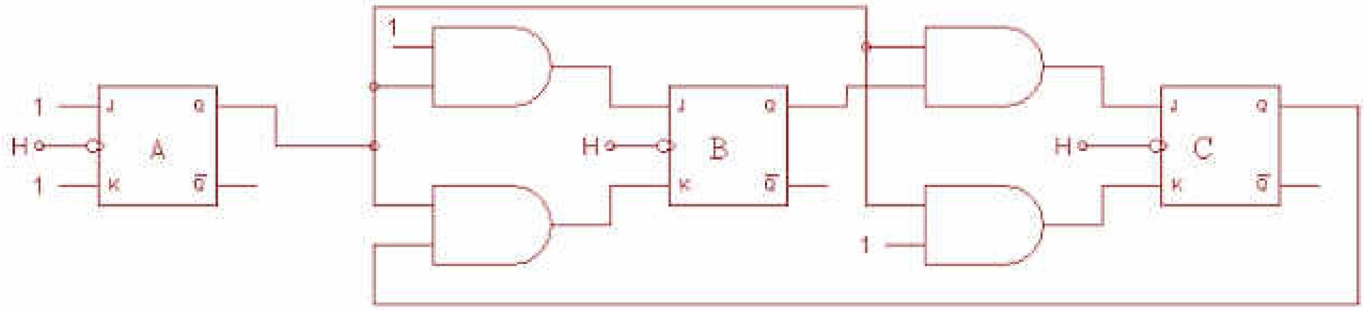
Dessinez les formes d'onde demandées suite à l'analyse de la figure suivante :





EXERCICE 12 :

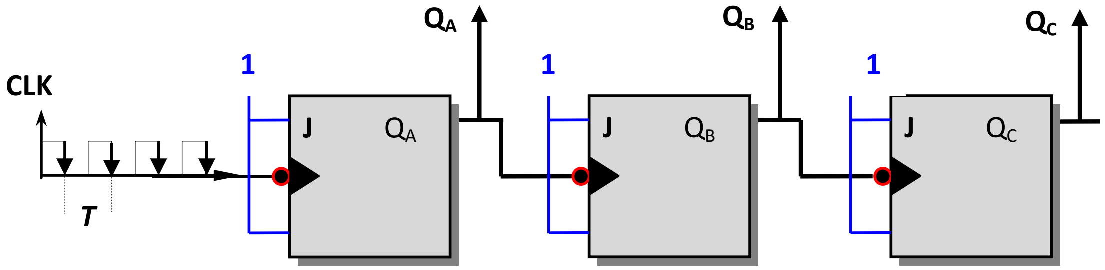
Soit le montage suivant :



- Donnez les équations des entrées J et K des 3 bascules.
- On suppose que le compteur par de l'état $Q_CQ_BQ_A$ = 000. Tracez les chronogrammes de l'horloge H et des sorties Q_A, Q_B et Q_C.
- Déterminez le modulo de ce compteur, la fréquence fI (I = A, B ou C) et le rapport cyclique aI pour Q_A, Q_B et Q_C.

EXERCICE 13 :

- A quoi servent les termes suivants : Johnson, Anneau. Discuter…
- Tracer la forme d'onde dans les sorties des bascules Q_A, Q_B et Q_C. Quelle est la fonction de ce montage ?



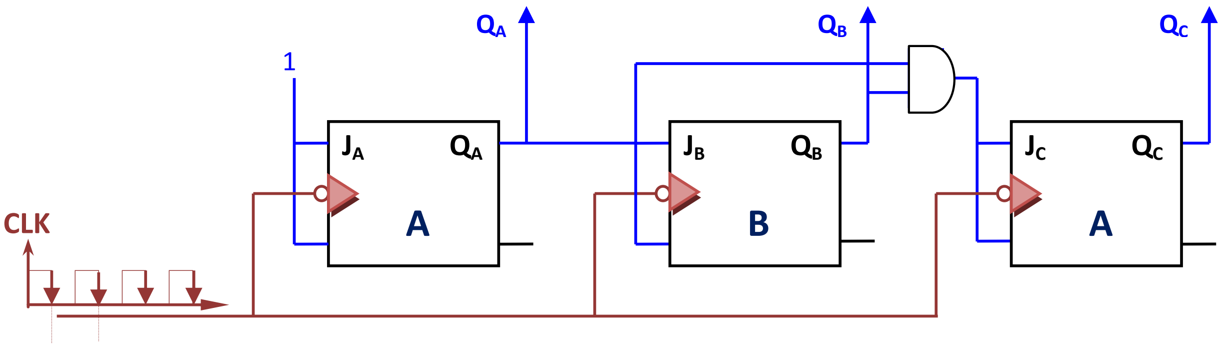
- Quel est le rôle du signal horloge (CLK) ?

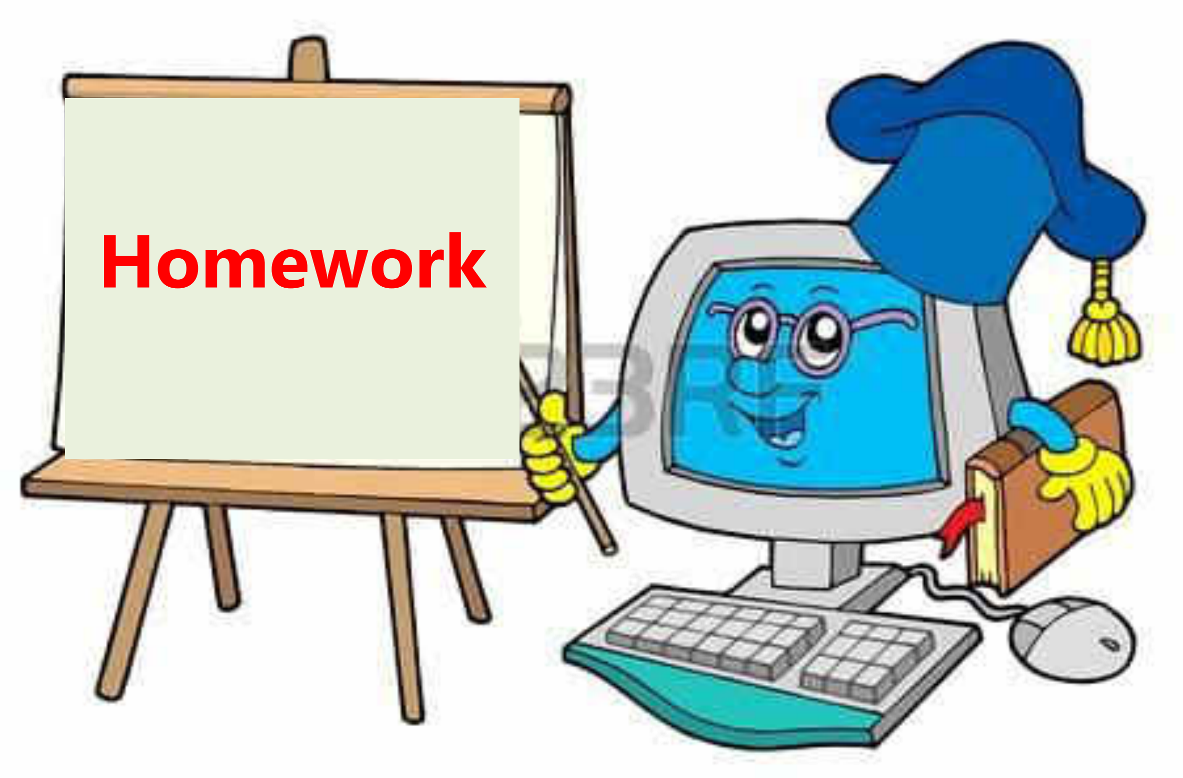
EXERCICE 14 :

- Que signifient les abréviations suivantes : RS, D, JK, T, FF, Clear, PRESET, CLK ? Discuter…
- Réaliser une bascule D à partir des bascules RS et JK.
- Donner les différents types des compteurs. Expliquer la démarche de conception d'un compteur quelconque.
- Etudier le processus de division de fréquences par 2.
- A quoi sert un circuit à trois états ? Discuter…

EXERCICE 15 :

- Expliquer la démarche de synthèse pour connaitre le fonctionnement du circuit ci-dessous.



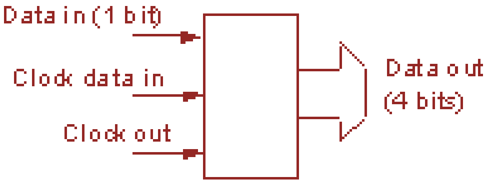


Source : http://ressource.electron.free.fr/cours/Exercice_de_logique_sequentielle.pdf

Homework

HW 1 :

Concevez un registre 4 bits rudimentaire, à entrée série et sortie parallèle, en utilisant des portes et des bascules D.



"Clock data in" doit valider le bit série, "Clock out" doit valider le mot de 4 bits.

✓ Comparez avec la spécification du HCT 164.

HW 2 :

Une mémoire travaille sur un bus de données de 8 bits, et un bus d'adresses de 13 bits.

✓ Quelle est sa capacité en Kilo-octets ? Etudiez la fiche technique du circuit mémoire FB 61C65.

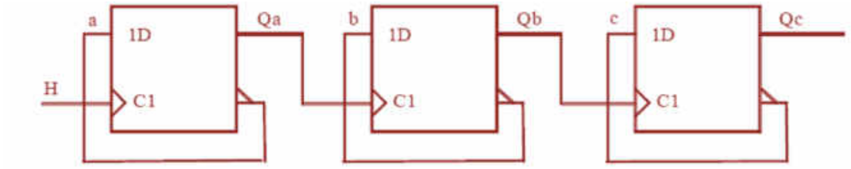
Concevez les deux mémoires suivantes:

a). 16 bits de données, 12 bits d'adresse

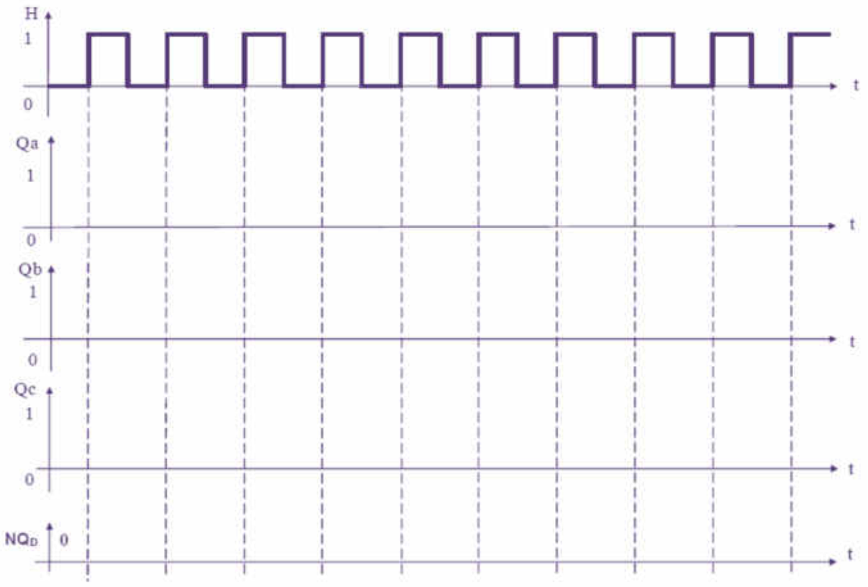
b). 8 bits donnée, 14 bits adresse (indication : utilisez le MSB de l'adressage pour sélectionner la mémoire désirée)

HW 3 : Fonction Decompteur Asynchrone A Bascule D

Le fonctionnement de ces bascules est-il synchrone ou asynchrone ? Argumenter votre réponse.

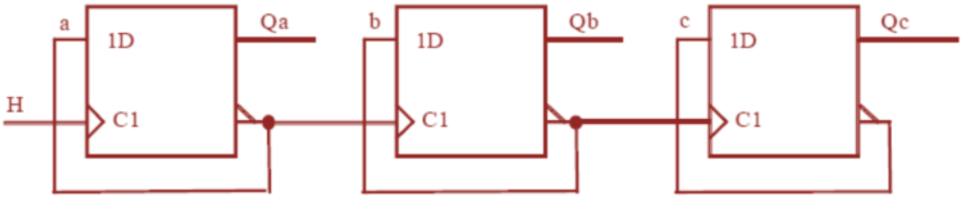


- ✓ Tracer les chronogrammes des sorties Qa, Qb et Qc (à l'état initial, Qa=Qb=Qc= "0").
- ✓ Convertir en décimal les trois bits binaires Qc, Qb et Qa en prenant Qa pour bit de poids faible.
- ✓ Quelle est la fonction réalisée ?
- ✓ Donner le modulo du compteur

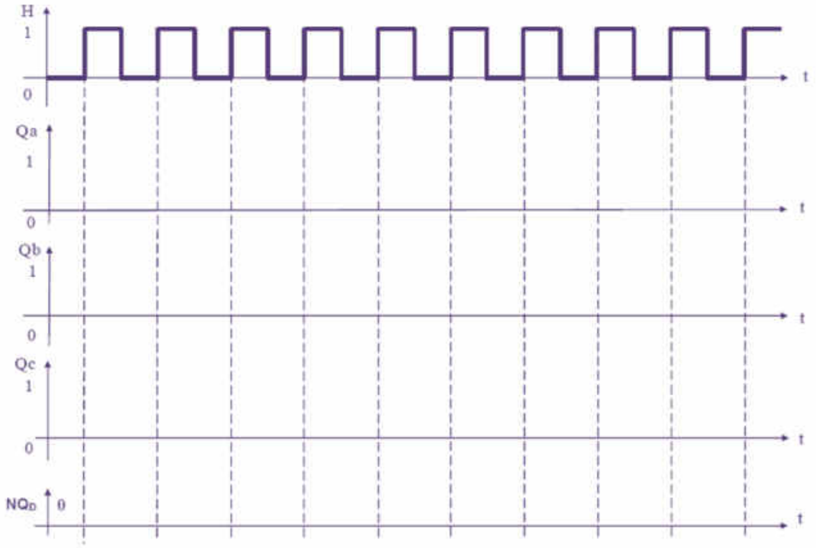


HW 4 : Fonction Compteur Asynchrone A Bascule D

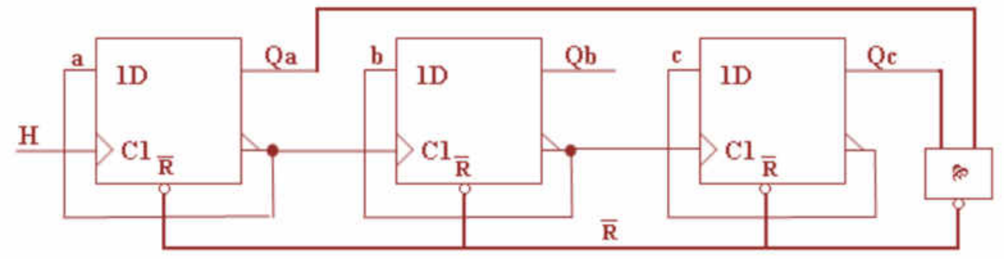
1. Le fonctionnement de ces bascules est-il synchrone ou asynchrone ? Argumenter votre réponse.



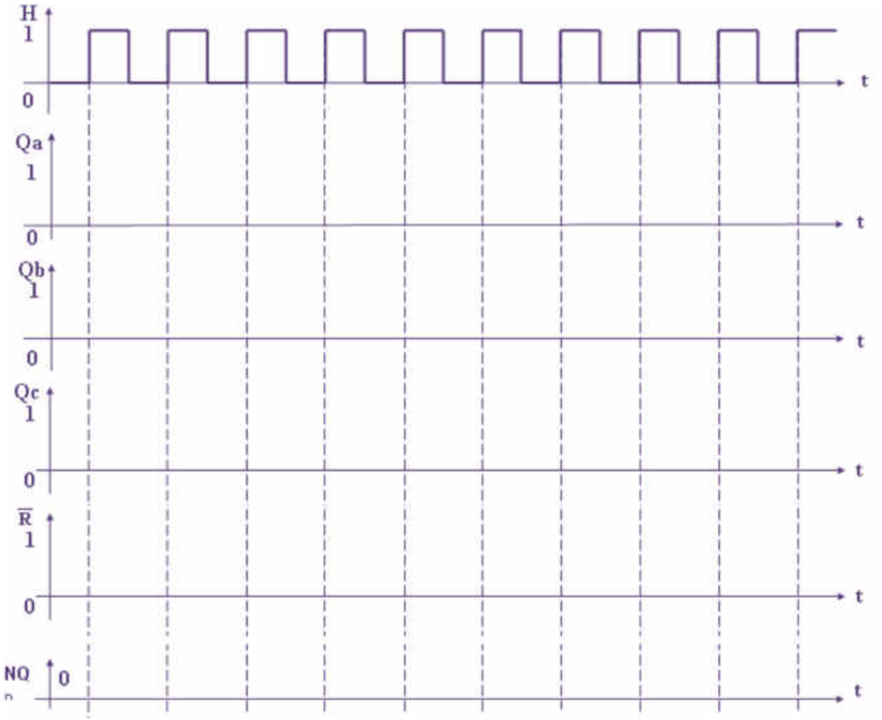
2. Tracer les chronogrammes des sorties Qa, Qb et Qc (à l'état initial, Qa=Qb=Qc= "0").
3. Convertir en décimal les trois bits binaires Qc, Qb et Qa en prenant Qa pour bit de poids faible.
4. Quelle est la fonction réalisée ? Comparer ce schéma structurel avec celui de l'exercice précédent et conclure sur l'incidence de la fonction réalisée.
5. Donner le modulo du compteur.



HW 5 : Fonction Compteur Asynchrone Modulo 5 A Bascule D

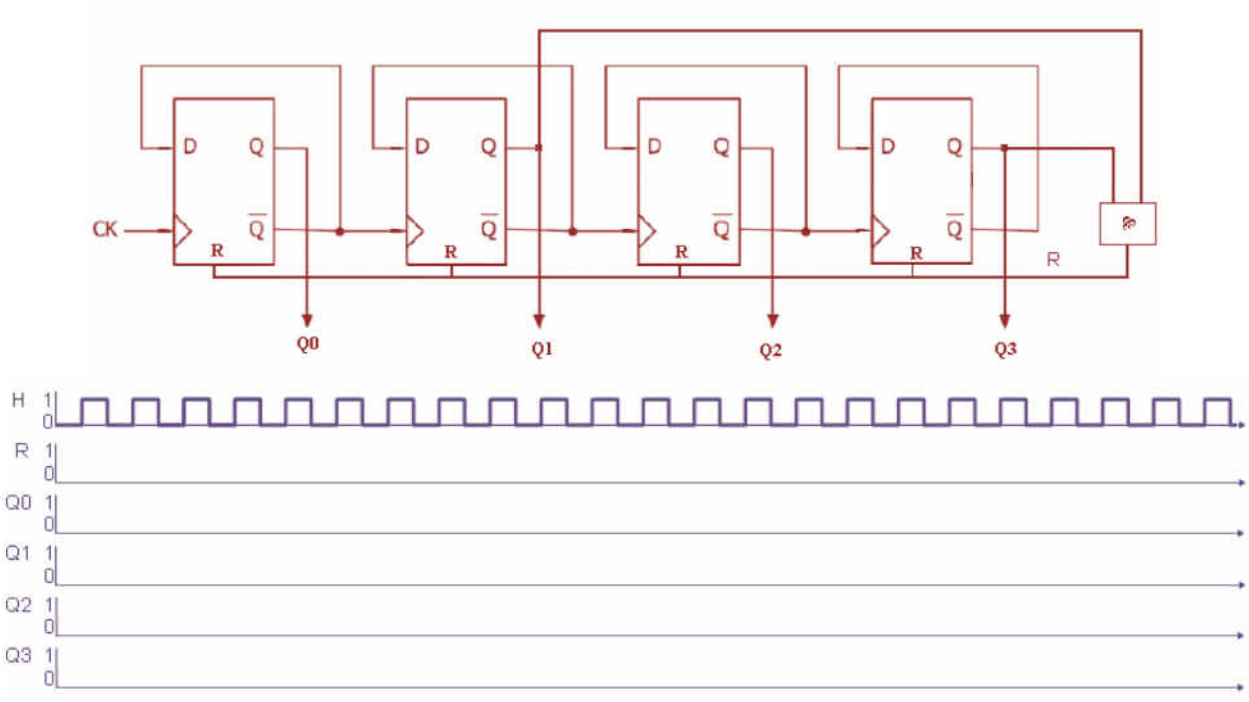


1. Donner la table de vérité de l’opérateur logique (/R = f (Qa, Qc))
2. Quel est le role de l’entrée /R ?A quel niveau est elle active ? Cette entrée est dite prioritaire, qu’entendez vous par là ?
3. Tracer les chronogrammes des sorties Qa, Qb, Qc et /R (à l’état initial, Qa=Qb=Qc= "0").
4. Convertir en décimal les trois bits binaires Qc, Qb et Qa en prenant Qa pour bit de poids faible.
5. Quelle est la fonction réalisée ?



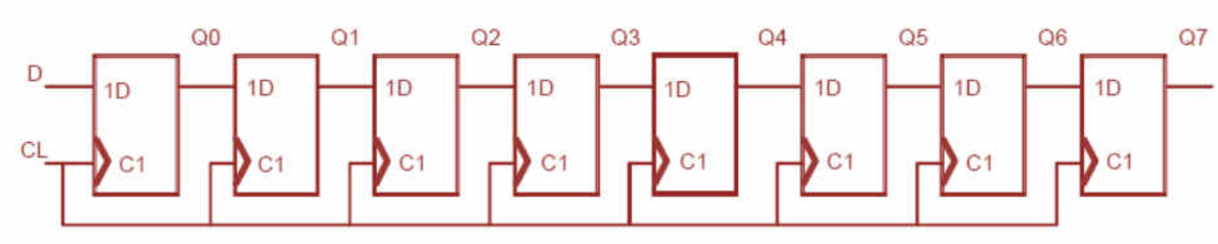
HW 6 : Fonction Compteur Asynchrone Modulo 10 A Bascule D

- Donner la table de vérité de l’opérateur logique (R = f (Q1, Q3))
- Quel est le rôle de l’entrée R ? A quel niveau est elle active ?
- Tracer les chronogrammes des sorties Q0, Q1, Q2, Q3 et R (à l’état initial, Q0=Q1=Q2=Q3= "0").
- Convertir en décimal les trois bits binaires Q0, Q1, Q2 et Q3 en prenant Q0 pour bit de poids faible.
- Quelle est la fonction réalisée ?

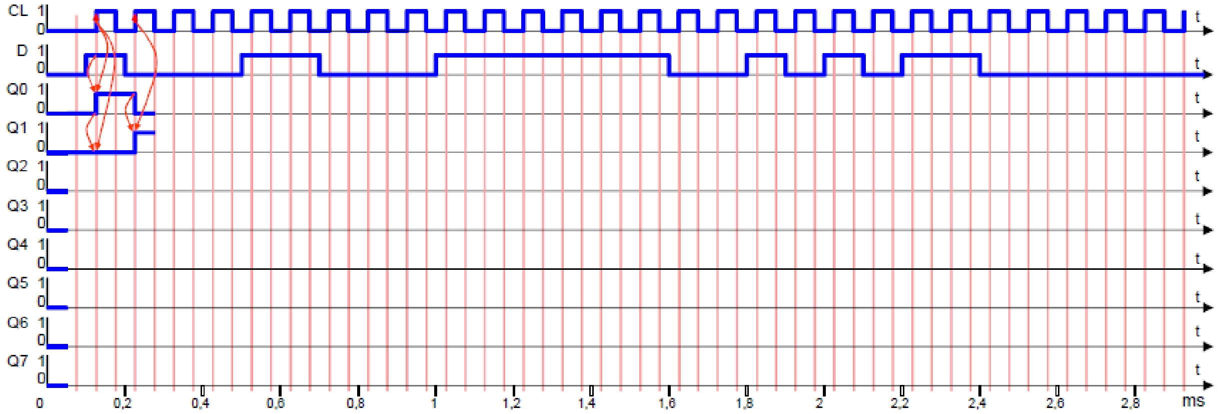


HW 7 : Fonction "Registre A Decalage"

Le schéma structurel pourrait être réalisé à partir du circuit logique CD4013A ou d'un 74LS374

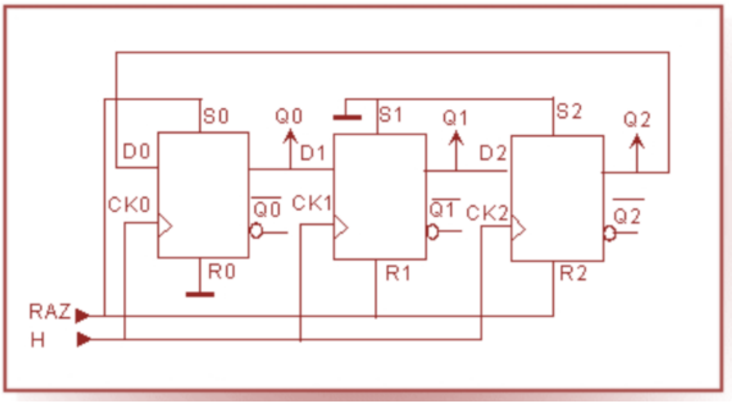


- ✓ Construire le chronogramme de cette structure demande d'avoir à l'esprit que tout opérateur introduit un temps de latenceentre le moment de la commande et celui où le résultat aboutit en sortie. Ce temps est appelé temps de propagation. Or ici les entrées de commandes sont actionnées simultanément. Lors d'un front montant de CL un opérateur voit donc l'état de l'opérateur qui le précède avant que celui-ci n'ait eu le temps de changer d'état. Ce principe étant admis vous pouvez construire successivement les chronogrammes de Q0, Q1,…, Q6 et Q7.

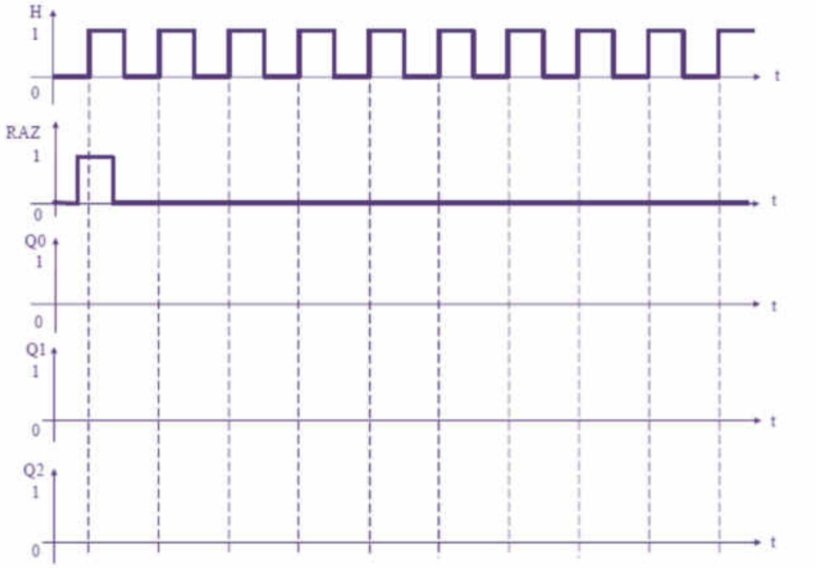


HW 8 : Etude Du "Compteur A Anneau"

Soit le circuit suivant :

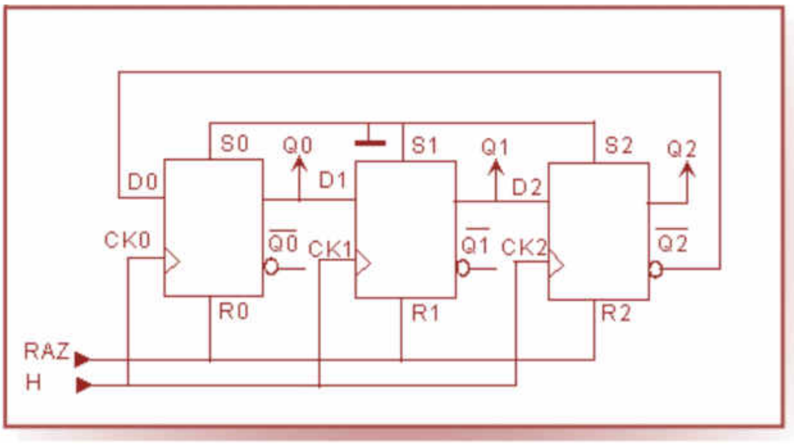


1. Tracer les chronogrammes de Q0, Q1 et Q2 (s'aider du mémotech pour la documentation du CD4013).
2. Exprimer la fréquence F_{Q0} en fonction de F_H
3. Au vu des chronogrammes, indiquer le modulo de ce compteur.



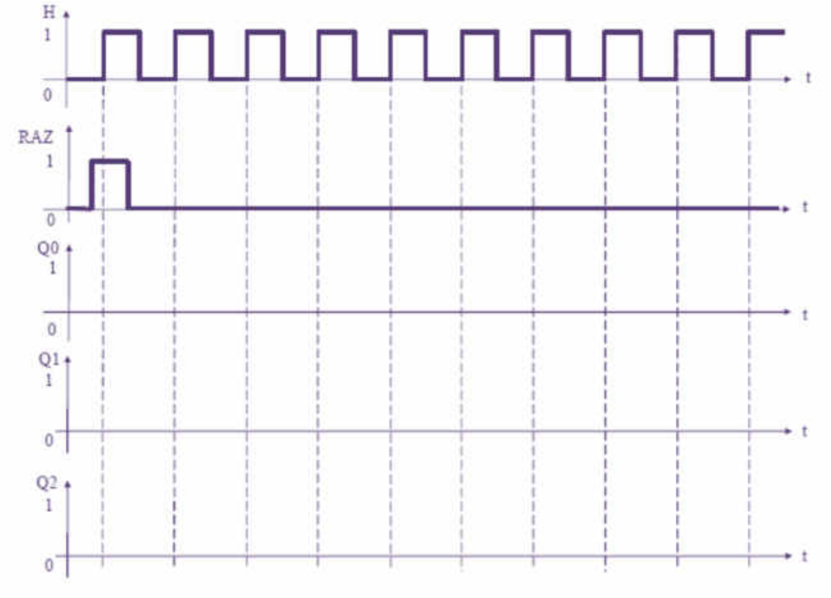
HW 9 : Etude Du "Compteur De Johnson"

Soit le circuit suivant :



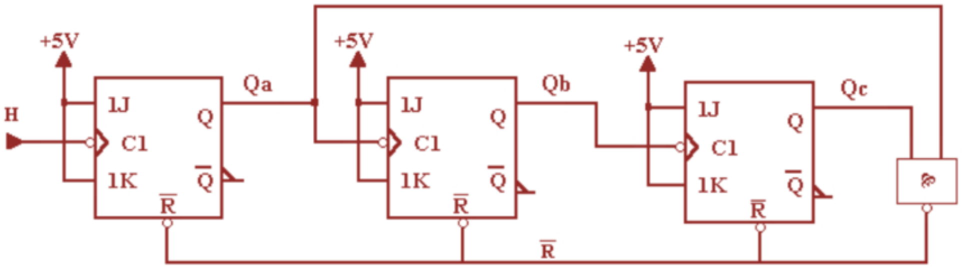
1. Faire le même travail que précédemment (compteur en anneau) sur ce nouveau schéma.

2. 2. Quelle différence existe-t-il entre ce schéma et le précédent ? Quelle en est la conséquence sur le modulo et la fréquence des signaux de sortie?

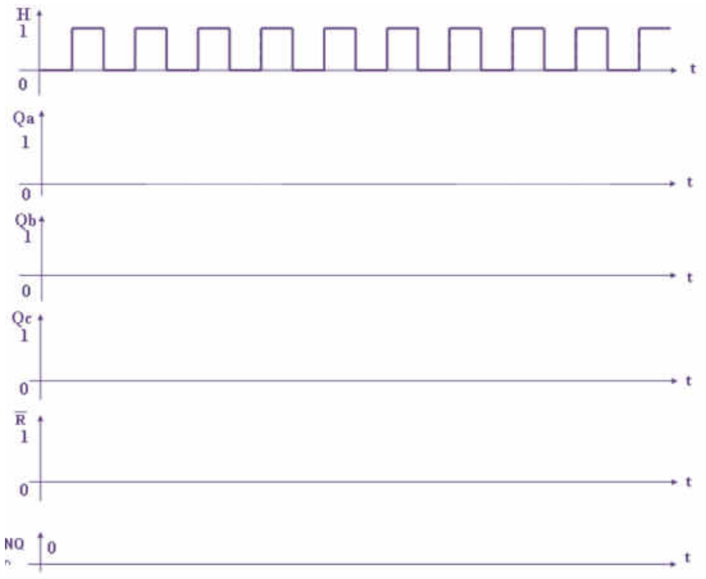


HW 10 : Fonction Compteur Asynchrone Modulo X à Bascules JK"

Soit le circuit suivant :



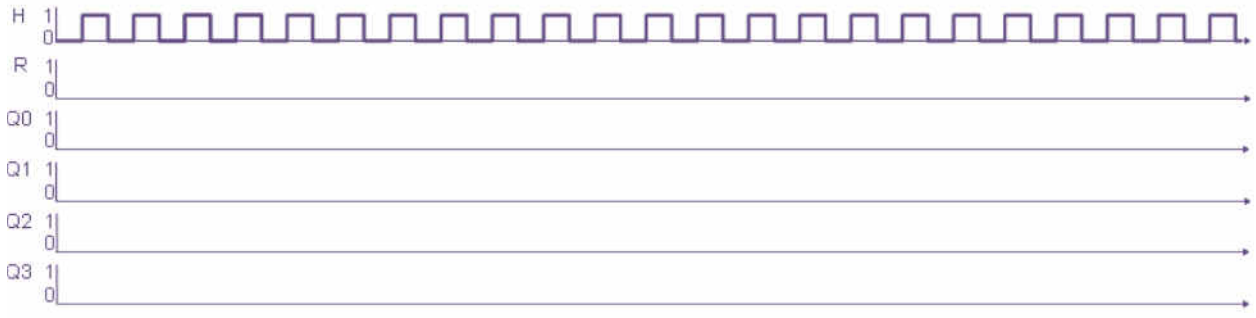
1. Sur quel front fonctionnent les bascules ?
2. A quel niveau logique les entrées /R sont elles activent ?
3. Compléter les chronogrammes de Qa, Qb, Qc et de /R (à l'état initial, Qa=Qb=Qc= "0").
4. Donner un nom à cette structure (modulo) ?



HW 11 : Fonction Compteur Asynchrone Modulo 10 àBascule JK

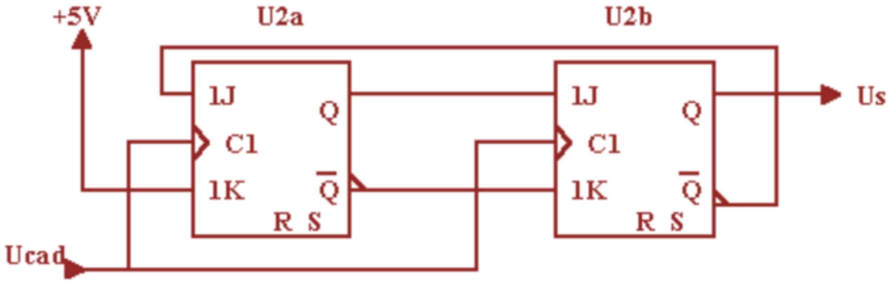
On désire réaliser un compteur asynchrone modulo 10 à l'aide de bascules JK activent sur front montant.

1. Réaliser le schéma permettant de réaliser ce compteur
2. Tracer les chronogrammes des sorties Q0, Q1, Q2, Q3 et Raz (à l'état initial, Q0=Q1=Q2=Q3= "0").
3. Convertir en décimal les quatre bits binaires Q3, Q2, Q1 et Q0 en prenant Q0 pour bit de poids faible.



HW 12 : Fonction "Division de Frequence"

Le schéma structurel est réalisé à partir du circuit logique HEF4027B.



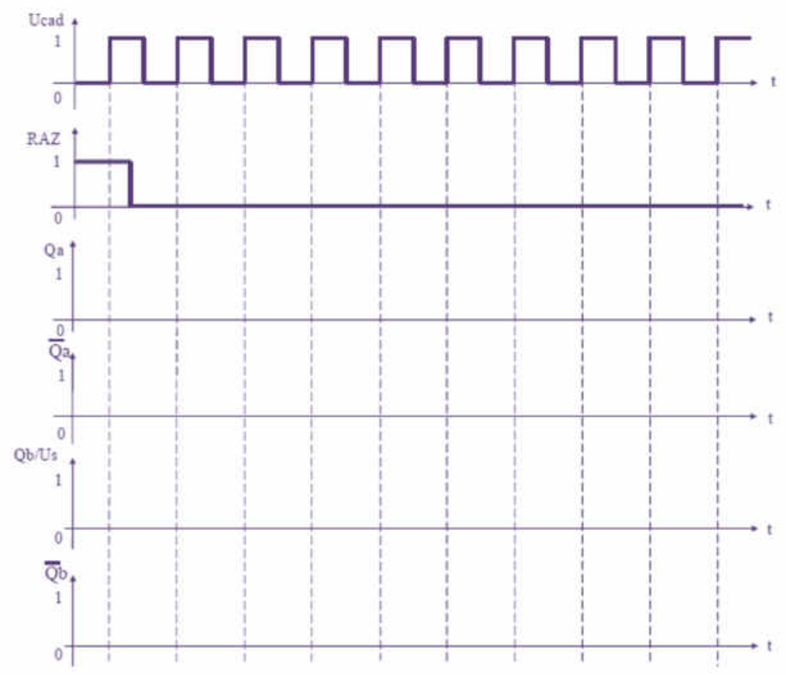
- Le circuit U2 est alimenté sous 0/5V.

1. Il vous appartient de câbler les broches repérées S et R de façon à inhiber la "mise à un" et à effectuer une "remise à zéro" de la sortie Us dès la mise sous tension du circuit. On utilisera le signal RAZ (cf chronogrammes).

Conditions initiales:

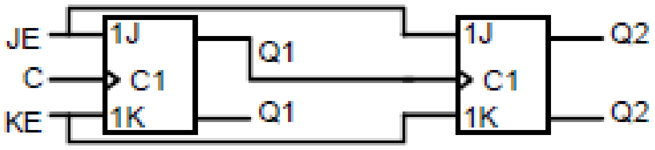
- La sortie Q de U2a est au niveau bas,
- La sortie Q de U2b est au niveau bas.

2. Sachant que Ucad est une ddp logique 0/5V de fréquence F=10KHz, représenter les chronogrammes des grandeurs J, K, Q et Q (chronogrammes en page suivante) pour les deux bascules JK, mettant en évidence le fonctionnement de la structure. Et ceci pour 9 périodes de Ucad.
    - Déterminer la fréquence du signal de sortie, et préciser la division effectuée.

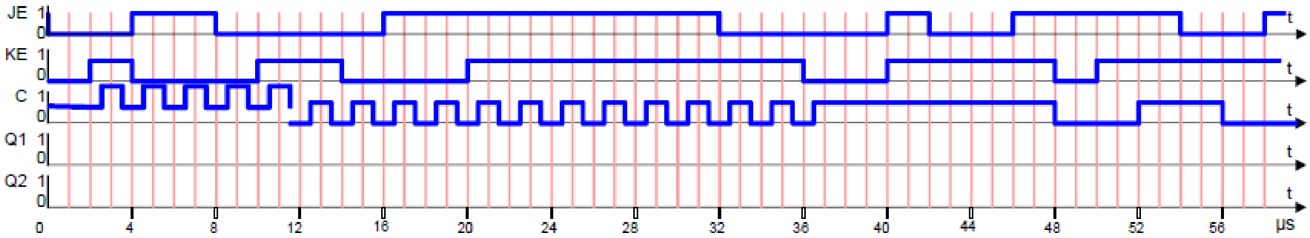


HW 13 : Fonction Asynchrone à Bascule JK

Le schéma structurel est réalisé à partir du circuit logique HEF4027B

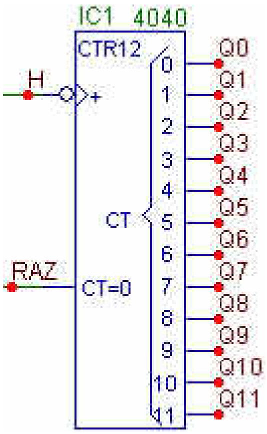


- Tracer les chronogrammes des sorties Q1 et Q2.



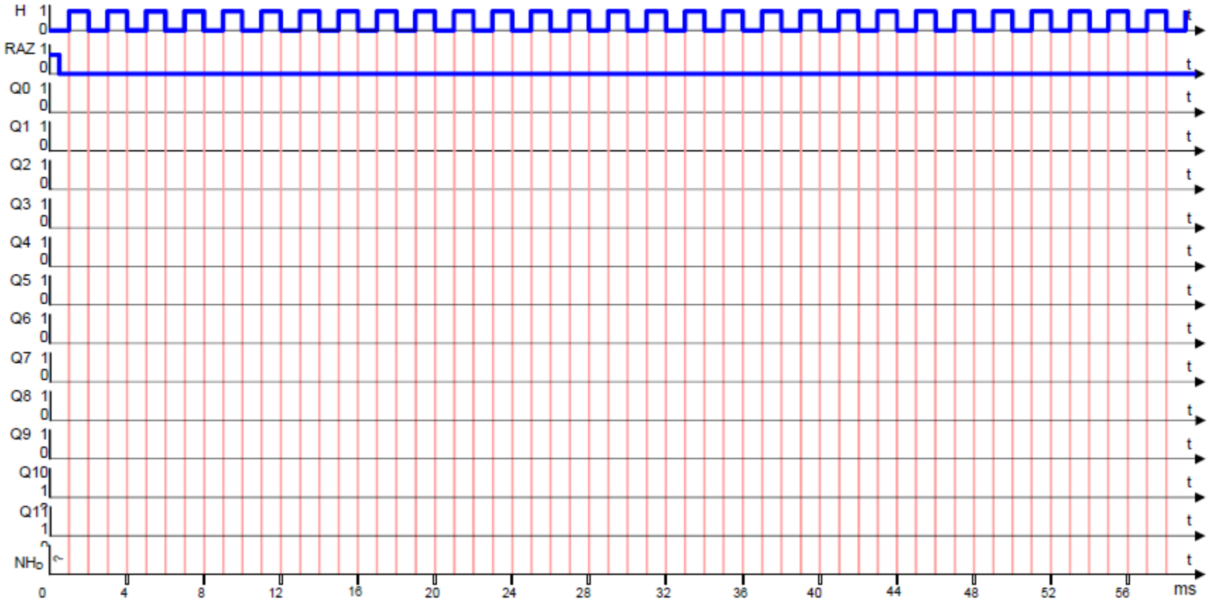
HW 14 : Etude D'un Compteur Binaire

Soit le symbole :



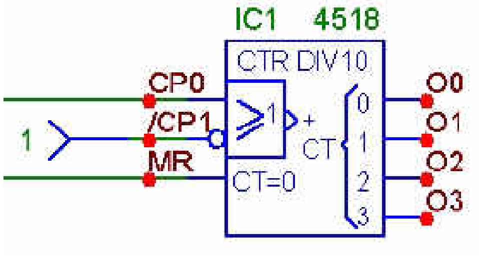
1. En exploitant sa table de vérité ou sa représentation déterminer :
   - ✓ L'entrée et l'événement provoquant sa mise à zéro ;
   - ✓ L'entrée et l'événement provoquant le comptage ;
   - ✓ Le modulo du comptage ;
   - ✓ Le nombre mini possible en sortie ;
   - ✓ Le nombre maxi possible en sortie.

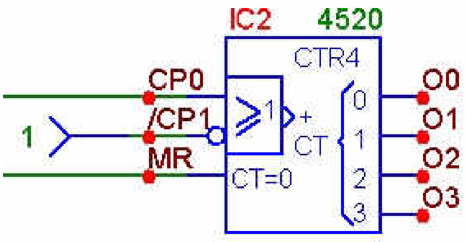
2. Compléter les chronogrammes suivants :



HW 14 : Etude de Compteurs

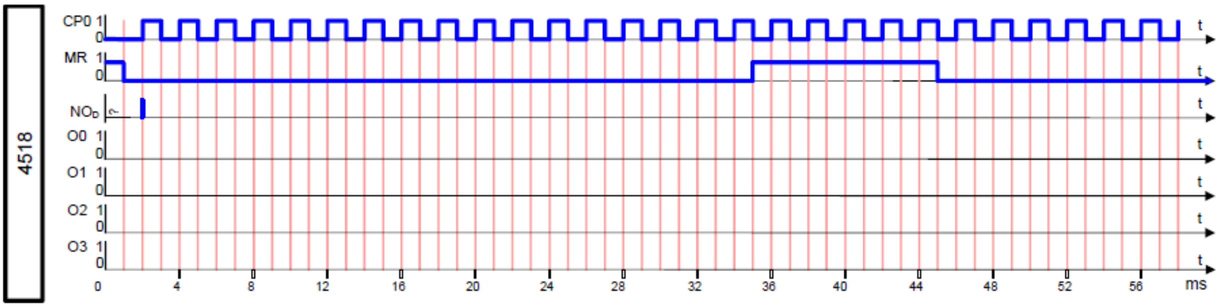
Soit les symboles :

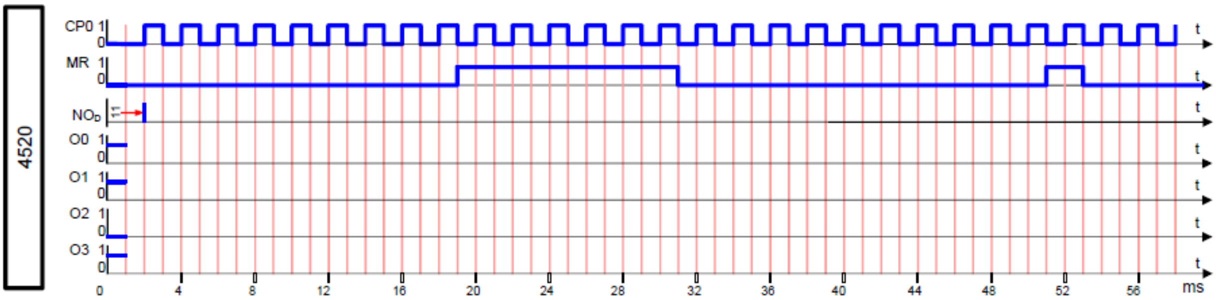




1. Pour chacun des composants représentés ci-dessous, déterminer :
   - ✓ L'entrée et l'événement (0, 1 □ ou □) provoquant leur mise à zéro ;
   - ✓ L'entrée et l'événement provoquant le comptage ;
   - ✓ et le modulo de comptage.

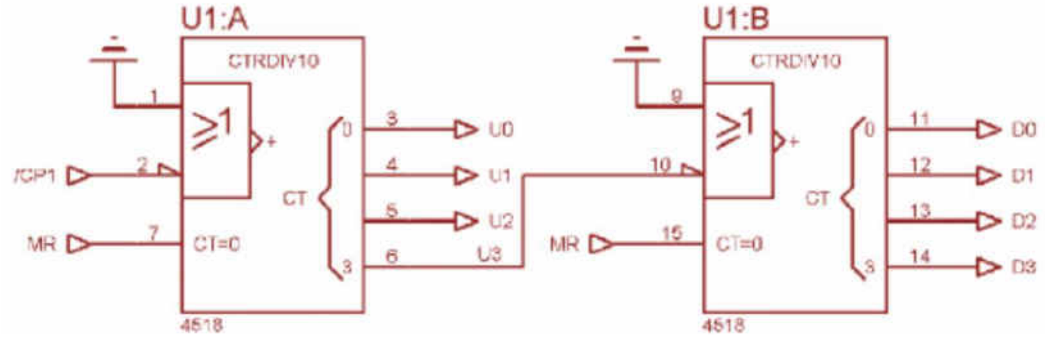
2. Compléter les chronogrammes suivants.
   - ✓ En faisant attention aux événements de mise à zéro (RAZ) compléter le chronogramme du nombre de sortie NO.
   - ✓ En déduire l'état de chaque ligne de sortie du compteur



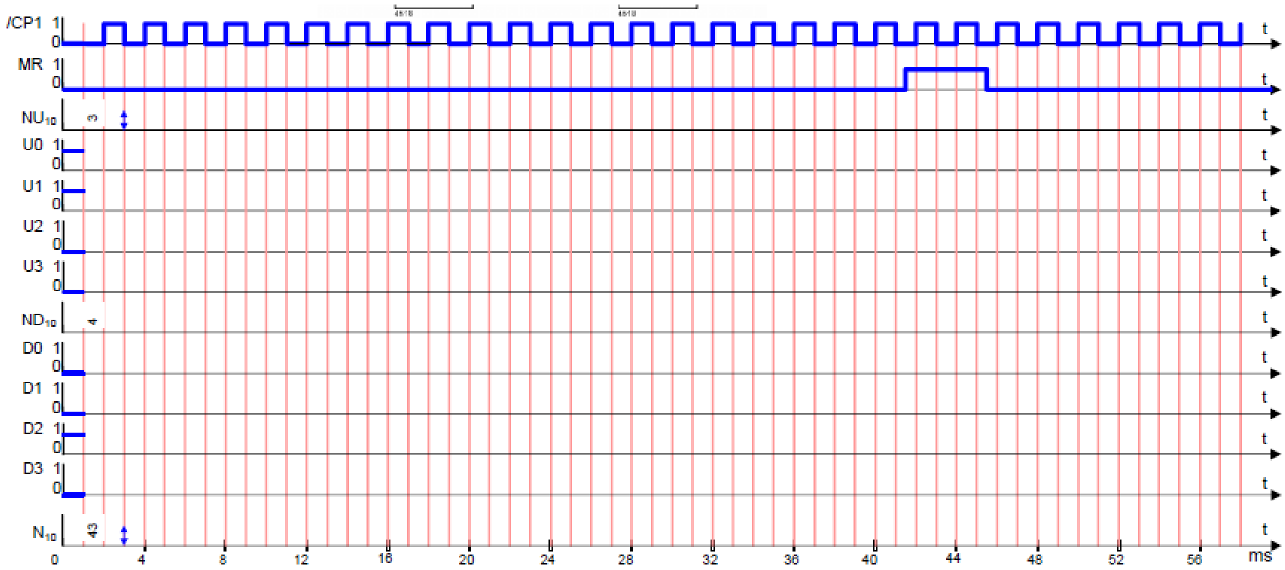


HW 15 : Associations de Compteurs Modulo 10

Soit le schéma structurel suivant :

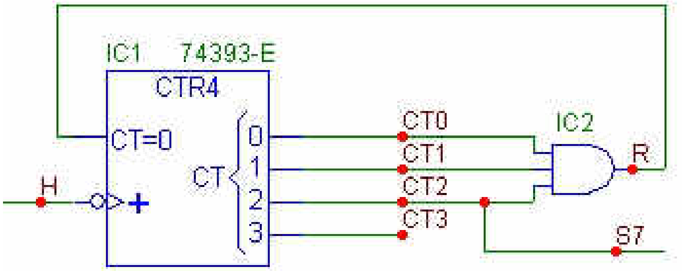


Tracer les chronogrammes ci-dessous



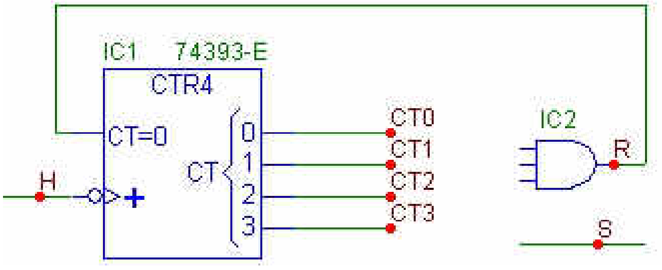
HW 16 : Fonction "Compteur à Modulo défini par Cablage"

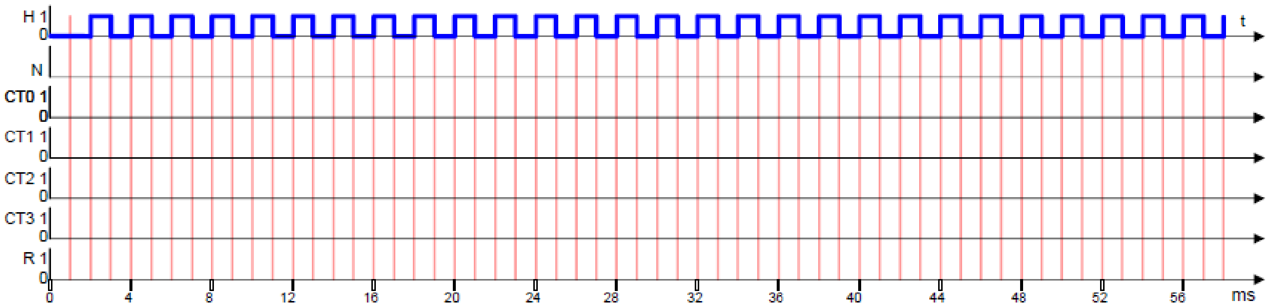
Soit le schéma structurel suivant :



Le schéma ci-dessus est celui d'un compteur dont le modulo est déterminé par câblage.

1. Pour IC1, déterminer :
   - L'entrée et l'événement provoquant le comptage ;
   - L'entrée et l'événement provoquant la mise à zéro de son contenu.
2. Déterminer la première valeur du contenu remettant à zéro le compteur.
3. Enoncer la succession de nombres stables produits par ce compteur.
4. Compléter les chronogrammes sur la page suivante.
5. Proposer un nouveau câblage afin d'obtenir une fonction comptage modulo 12. Enoncer la suite des nombres stables produits par cette structure.

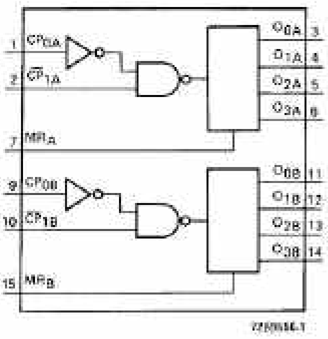




HEF4518B Dual BCD counter

FUNCTION TABLE

CP_0	CP_1	MR	MODE
↑	H	L	counter advances
L	↓	L	counter advances
↓	X	L	no change
X	↑	L	no change
↑	L	L	no change
H	↓	L	no change
X	X	H	O_0 to O_3 = LOW



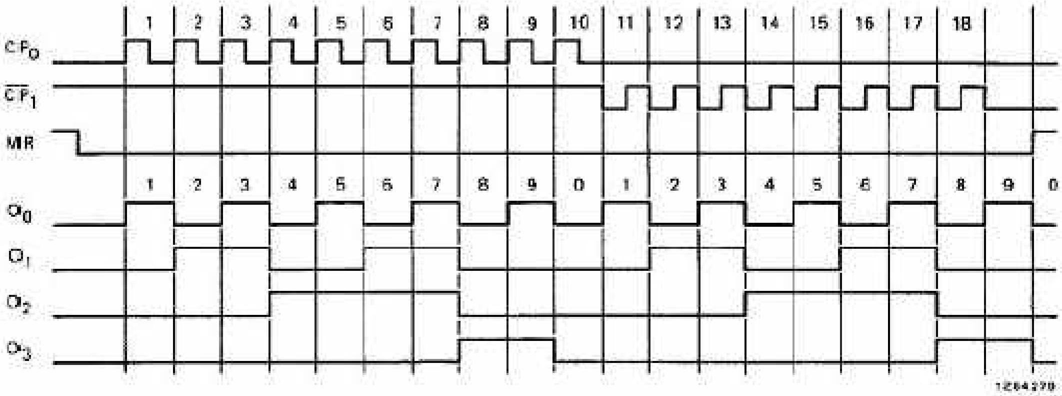


Fig.6 Timing diagram.

74HC/HCT393 Dual 4-bit binary ripple counter

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 13	1CP, 2CP	clock inputs (HIGH-to-LOW, edge-triggered)
2, 12	1MR, 2MR	asynchronous master reset inputs (active HIGH)
3, 4, 5, 6, 11, 10, 9, 8	$1Q_0$ to $1Q_3$, $2Q_0$ to $2Q_3$	flip-flop outputs
7	GND	ground (0 V)
14	V_{CC}	positive supply voltage

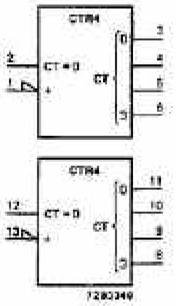


Fig.3 IEC logic symbol.

REFERENCES BIBLIO-WEB GRAPHIQUES

Références Bibliographiques

[1]. Letocha, Introduction aux circuits logiques, Mc-Graw Hill.

[2]. C. Lafont; Cours et problèmes d'électronique numérique, 124 exercices avec solutions; Edition Ellipses.

[1]. J.M. Bernard, J. Hugon ; De la logique câblée aux microprocesseurs, Tomes 1 à 4 ; Eyrolles.

[2]. R. Delsol ; Electronique numérique, Tomes 1 et 2 ; Edition Berti.

[3]. P. Cabanis ; Electronique digitale ; Edition Dunod.

[4]. M. Gindre ; Logique séquentielle ; Edition Ediscience.

[5]. J. P. Vabre et J. C. Lafont ; Cours et problèmes d'électronique numérique ; Ellipses, 1998.

[6]. H. Chemmali, Notes de cours, TEC 480, Département d'Electronique, Université de Sétif, 1993-1994.

[7]. Thomas L. Floyd ; Systèmes numériques : Concepts et applications ; Les éditions Renaold Goulet Inc, 2000.

[8]. Jean-Claude Lafond, Jean Paul Vabre ; Cours et problèmes d'électronique numérique : 124 exercices avec solutions ; Edition Marketing, 1986.

[9]. C. Brie, Logique combinatoire et séquentielle, Ellipses, 2002.

Références Webgraphiques

[10]. http://www.info.univ-angers.fr/~richer/ensl3i_crs4.php

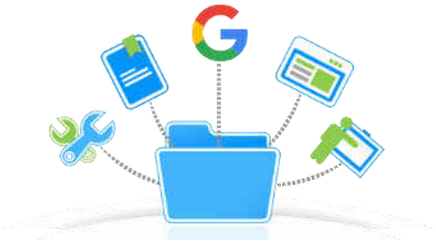
[11]. http://ambroise.brou1.free.fr/en_007.htm .

[12]. http://www.courstechinfo.be/Hard/Memoire.html

[13]. http://electronique-et-informatique.fr/Digit/Digit_5TS.html

[14]. http://sti.ac-orleans-tours.fr/spip2/IMG/pdf/Memoires_complet.pdf

[15]. http://www.les-electroniciens.com/sites/default/files/cours/sam1a_coursv11.pdf



Feuille des remarques – Cours -

ENSEIGNANT : ______________________________ **SALLE :** ________

Remarques	Page
	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………
………	………

Feuille des remarques – TD -

ETUDIANT : ________________________ **SALLE :** __________

Remarques	Page
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………
……	………