



## Intitulé du polycopié

**Polycopié de Cours : Structure machine 2**

**Destiné aux étudiants**

**Niveau : Première année Licence**

**Spécialité : Mathématique et informatique**

**Auteur**

**MAHROUG RABIAA**

Experts du polycopié	Grade	Établissement d'affiliation
Bendoumia Redha	MCA	Université de Blida 1
Hachichi Hiba	MCA	Université de D.Bounaama khemis miliana

**Date de validation du polycopié par l'instance scientifique habilitée**

**CSD et/ou CSF**

**CSD 26-06-2023**

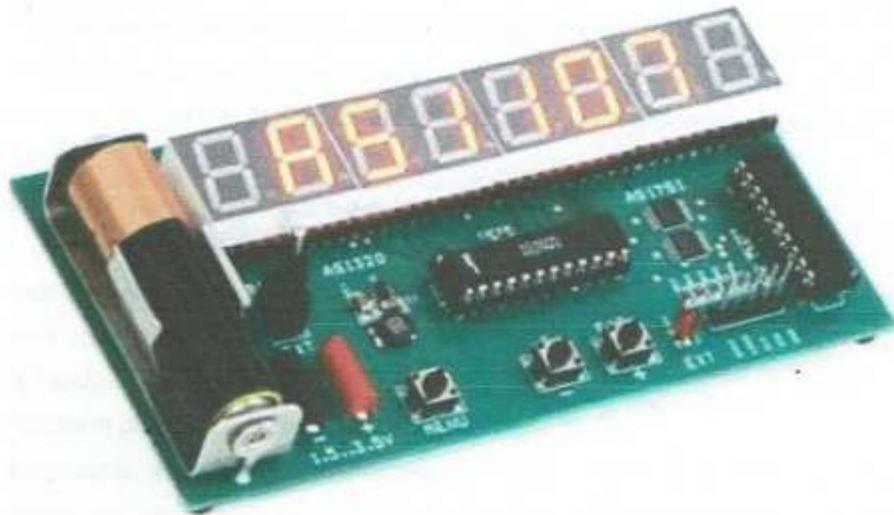
**CSF 24-10-2023**

**Année universitaire : 2022/2023**

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique  
Université Djilali Bounaâma de Khemis Miliana  
Faculté des sciences et techniques  
Département de maths et informatique  
Niveau : 1<sup>ère</sup> année MI - Licence



# Polycopié de Cours Structure machine 2



Rédigé par : Dr. MAHROUG RABIAA  
E-mail : r.mahroug@univ-dbk.m.dz

URL du fichier: <https://>

Année universitaire 2022-2023

## Introduction

Ce polycopié intitulé « Polycopié de Cours : Structure machine 2 » complète le polycopié « Polycopié de Cours : Structure machine 1 », le but de ce polycopié est de familiariser la conception des circuits combinatoires, la conception des circuits séquentiels et les circuits intégrés. Il constitue un support de cours et des séries d'exercices avec solution de matière structure machine 2. Structure machine 2 se concentre sur les fonctions et liaisons qui doivent être établis pour obtenir de meilleures performances de la machine. En d'autres termes, la structure de machine 2 fait référence à la disposition des organes d'une machine électronique et les relations entre ces organes.

Ce polycopié se compose d'un guide de cours de structure machine 2 et quelques exercices résolus pour chaque chapitre. Il explique d'une manière simple et facile l'architecture et le fonctionnement d'un ordinateur en commençant par des notions de base. Ce polycopié s'adresse d'abord aux étudiants LMD (1<sup>ère</sup> année licence) socle commun Mathématique et Informatique, et, ceux qui s'intéressent aux connaissances de base des circuits numériques, de l'architecture des ordinateurs et l'ingénierie informatique.

A l'issue de ce polycopié, l'étudiant(e) sera capable de :

- Comprendre les notions de circuits numériques.
- Apprendre la conception des fonctions logiques et l'algèbre de Boole.
- Découvrir les diverses fonctions intégrées de la logique combinatoire (Codage, décodage, transcodage, circuits arithmétiques et circuits d'aiguillage).
- Savoir les circuits logiques élémentaires (portes logiques, bascules).
- Comprendre le fonctionnement des bascules de base : D, T, JK et RS.

Apprendre les différentes méthodes de synthèse des systèmes de logique combinatoire et de séquentiel.

Distinguer les circuits logiques combinatoires et les circuits séquentiels.

- Etablir le chronogramme d'un système séquentiel.

Comprendre les systèmes logiques combinatoires et séquentiels liés à son domaine de spécialité.

- apprenant acquerra les divers outils permettant de concevoir et de réaliser des applications à base des circuits logiques combinatoires et séquentiels.
- Comprendre la notion de circuits intégrés.

Dans ce polycopié, nous aborderons quelques concepts de base qui mènent à la conception des circuits numérique (combinatoires, séquentiels et intégrés). Nous traiterons, dans la première partie de ce polycopié les circuits logiques combinatoire, et nous allons synthétiser certains de ces circuits comme les circuits arithmétiques (l'additionneur, le soustracteur et le comparateur), codeur, décodeur, transcodage et circuits d'aiguillage (multiplexeur et démultiplexeur). Puis, nous présenterons certains circuits séquentiels très importants comme les compteurs, décompteurs, les mémoires et les registres dans lesquels la notion de chronologie des événements joue un rôle central. Ensuite, nous terminerons notre étude des systèmes séquentiel en présentant une méthode de synthèse des compteurs et des décompteurs. La dernière partie porte sur la conception d'un circuit intégré, nous étudierons de manière générale les deux grandes familles TTL et CMOS en donnant leurs principales caractéristiques. Dans ce qui suit, on détaille le programme de la matière.

## Table des matières

Introduction.....	i
Table des matières.....	ii
Abréviation.....	v
<b>Chapitre I : Introduction Générale</b> .....	<b>1</b>
1.1. C'est quoi le système électronique ? .....	1
1.2. C'est quoi un circuit électronique ? .....	1
1.3. Types de composants de base .....	2
1.4. Supports de stockage.....	3
1.4.1. Définition des supports de stockage .....	3
1.4.2. Les unités de mesure en informatique.....	3
1.4.3. Type des supports de stockage .....	4
1.4.3.1. Les supports optiques .....	4
1.4.3.2. Les supports miniaturisés.....	4
<b>Chapitre II : Conception des Circuits Combinatoires</b> .....	<b>7</b>
2.1 Définition.....	7
2.2. Synthèse d'une fonction combinatoire .....	7
2.3. Classification .....	8
2.4. Circuits combinatoires usuelles.....	9
2.4.1. Additionneur .....	9
2.4.1.1. Demi-Additionneur.....	9
2.4.1.2. Additionneur complet.....	10
2.4.1.3. Additionneur à 4 bits.....	11
2.4.2. Soustracteur .....	12
2.4.2.1. Demi-soustracteur .....	12
2.4.2.2. Soustracteur complet.....	13
2.4.2.3. Soustracteur à 4 bits.....	14
2.4.3. Additionneur/Soustracteur.....	15
2.4.4. Comparateur.....	15
2.4.4.1. Comparateur à 1 bit.....	15
2.4.4.2. Comparateur à 2 bits .....	16
2.4.4.3. Comparateur 2 bits avec des comparateurs 1 bit .....	17
2.4.5. Multiplexeur .....	18
2.4.5.1. Multiplexeur 2x1.....	18
2.4.5.2. Multiplexeur 4X1 .....	19



2.4.5.3. Multiplexeur 8X1 .....	20
2.4.6. Démultiplexeurs.....	22
2.4.6.1. Démultiplexeurs 1X4 .....	23
2.4.6.2. Démultiplexeurs 1X8 .....	24
2.4.7. Décodeur binaire .....	25
2.4.7.1. Décodeur binaire 2x4 .....	25
2.4.7.2. Décodeur binaire 3X8.....	26
2.4.8. Encodeur binaire (codeur).....	27
2.4.8.1. Encodeur binaire 4x2.....	28
2.4.8.2. Encodeur binaire 8x3.....	28
2.4.9. Transcodeur .....	30
2.4.9.1. Transcodeur 7-segments.....	30
2.4.9.2. Transcodeur BCD/EXCESS3.....	33
Série d'exercices N°1 (Circuits combinatoires).....	36
<b>Chapitre III : conception des Circuits Séquentiels .....</b>	<b>41</b>
3.1. Introduction.....	41
3.2. Système séquentiels synchrone et asynchrones.....	41
3.3. Les bascule .....	43
3.3.1. Définition d'une bascule .....	43
3.3.2. Les types des bascules .....	43
3.3.2.1. Bascule RS :( Reset_set) .....	43
3.3.2.2. Bascules RST .....	44
3.3.2.3. Bascule JK.....	46
3.3.2.4. Bascule T (Trigger flip-flop) .....	47
3.3.2.5. Bascule D (Delay).....	48
3.4. Utilisation des bascules.....	49
3.4.1. Utilisation des bascules pour réaliser un registre .....	49
3.4.1.1. Définition d'un registre.....	49
3.4.1.2. Fonctionnement d'un registre.....	50
3.4.1.3. Type des registres .....	50
3.4.2. Utilisation des bascules pour la mémoire centrale .....	54
3.4.2.1. Définition d'une mémoire.....	54
3.4.2.2. Différents types de la mémoire.....	54
3.4.2.3. Caractéristiques d'une mémoire.....	57
3.4.3. Utilisation des bascules pour réaliser des compteurs.....	58
3.4.3.1. C'est quoi un compteur ?.....	58
3.4.3.2. Types de compteur.....	59



3.4.4. Utilisation des bascules pour réaliser Les décompteurs.....	71
3.4.5. Utilisation des bascules pour réaliser Les compteurs/décompteurs.....	72
3.5. Synthèse de circuits séquentiels.....	75
3.5.1. Définition d'une machine à états finis.....	75
3.5.1.1. Tables de transitions .....	75
3.5.1.2. Diagramme d'état .....	76
3.5.2. Classes de MSA.....	78
3.5.2.1. Machine de Moore.....	78
3.5.2.2. Machine de Mealy .....	78
3.5.3. Analyse d'un circuit séquentiel.....	78
3.5.4. Synthèse d'un circuit séquentiel.....	79
Série d'exercices N° 2(Circuits Séquentiels) .....	80
<b>Chapitre IV : Conception des Circuits Intégrés.....</b>	<b>85</b>
4.1. Définition d'un circuit intégré.....	85
4.2. Caractéristiques des circuits intégrées .....	85
4.4. Présentation des CI.....	88
4.5. Identification des CI.....	89
4.6. Classement des circuits intégrés .....	89
4.7. Technologies de fabrication de CI .....	90
4.7.1. Circuits logiques TTL.....	90
4.7.2. Circuits logiques CMOS .....	93
Série d'exercices N°3(Circuits Intégrés) .....	95
<b>Solution des séries d'exercices.....</b>	<b>97</b>
1. Solution de la série d'exercices N°1 (Circuits combinatoires) .....	97
2. Solution de la série d'exercices N°2 (Circuits séquentiels) .....	116
3. Solution de la série d'exercices N°3 (Circuits intégrés).....	133
<b>Bibliographies .....</b>	<b>137</b>



## Abréviation

DA	: Demi-Additionneur
ADD	: Additionneur complet
MUX	: Multiplexeur
DEMUX	: Démultiplexeur
CLK / H ou h	: Clock / Horloge
RAM	: Random Access Memory (mémoires vives)
ROM	: Read Only Memory (mémoires mortes)
FSM	: Finit State Machine
MEF	: Machine à Etats Finis
CI	: Circuit Intégré
CMOS	: Complementary Metal Oxide Semiconductor
TTL	: Transistor Transistor Logic



# Chapitre I : Introduction Générale

## 1.1. C'est quoi le système électronique ?

L'électronique est une science de l'ingénieur, l'une des branches les plus importantes de la physique appliquée, qui permet l'étude et la conception de structures qui effectuent le traitement du signal électrique, tels que les courants ou les tensions électriques, les porteurs d'informations...etc. C'est à dire visant à traiter des signaux électroniques à l'aide de des composants matériels [1].

Un signal est une grandeur ou un ensemble de grandeurs produit par un capteur ou un ensemble de capteurs, considéré comme représentant de manière suffisamment satisfaisante une grandeur physique donnée et qui porte l'information à traiter. Il s'agit généralement d'une tension ou d'un courant électrique, mais il peut également s'agir d'un champ magnétique ou électrique.

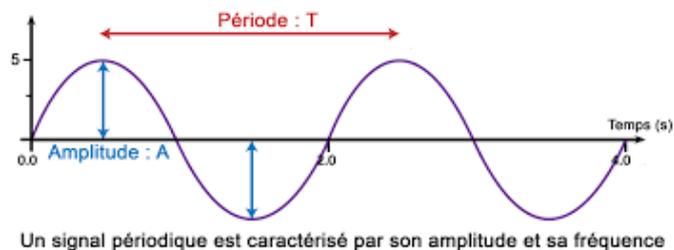


Figure 1.1 : Représentation du signal.

La fréquence est considérée en physique comme le nombre de fois qu'un phénomène périodique se reproduit par unité de mesure du temps. Elle est exprimée en hertz d'un circuit électronique représente le nombre de changements d'état possibles d'une valeur en une seconde.

- Hertz (Hz) est une unité dérivée de la fréquence du système international SI. Ce qui équivalent à un événement par seconde ( $s^{-1}$  ou  $1/s$ ).
- la fréquence est liée à la vitesse et à la longueur d'onde par la relation suivante:  $f = v / \lambda$
- La période est l'inverse de la fréquence:  $t = 1/f$ .

## 1.2. C'est quoi un circuit électronique ?

Les circuits électroniques sont les principaux sujets de recherche de la science de l'électronique. Un circuit électronique est un système composé de plusieurs composants électroniques liés. Le mot circuit vient du fait que le traitement s'effectue grâce aux courants électriques circulant dans les composants associés. Les systèmes électroniques sont généralement conçus en deux parties, la partie opérationnelle qui gère les signaux de puissance porteurs d'énergie (courants forts) et la partie informationnelle qui gère les signaux porteurs d'information (courants faibles). Les circuits électroniques modernes se composent de centaines de millions de d'éléments élémentaires.

Les applications électroniques se divisent en deux groupes distincts : le groupe de traitement de l'information, comprend des domaines de l'informatique, les télécommunications et les mesures, et le groupe de commande qui s'occupe de la gestion de l'information (ordres), tels que les microprocesseurs, les PIC ou les moteurs pas à pas.

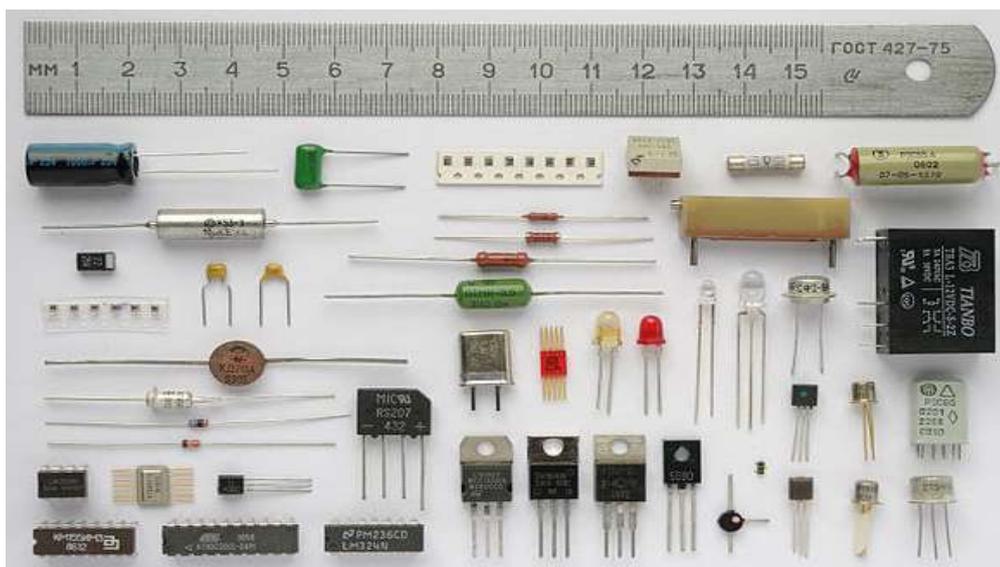
### 1.3. Types de composants de base

Il est difficile de faire une liste complète de tous les composants électroniques pouvant être présents, mais il est possible de lister ceux qui sont très courants dans les montages électriques. Ils peuvent être classés de différentes façons, en fonction de leur fonctionnement, sa fonction qu'il exerce ou même les boîtiers qui les composent. Par conséquent, les classifications suivantes peuvent être définies comme suit :

**Composants actifs** : définis les composants qui augmentent la capacité de courant et/ou de tension. L'une des caractéristiques est que ces composants ont besoin d'une alimentation pour fonctionner. Les composants actifs incluent notamment les diodes et les transistors.

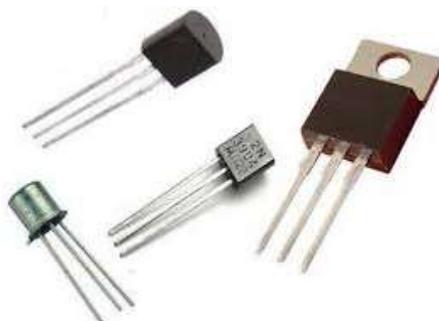
**Composants passifs** : définis les composants qui n'augmentent pas la puissance d'un signal électrique. L'une des caractéristiques est que ces composants n'ont pas besoin d'une alimentation pour fonctionner. Les composants passifs incluent notamment les condensateurs, les résistances fixes et les inductances.

**Composants discrets** : définis les composants électroniques de base dont le rôle est d'assurer une fonction élémentaire. Les composants passifs incluent notamment les résistances ou les condensateurs.



Les composants électroniques peuvent également être classés comme suit :

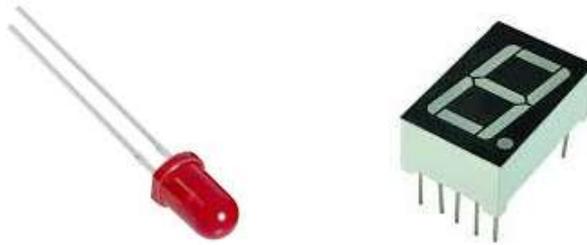
- ✓ Composants d'électroniques analogiques comprennent des résistances, des condensateurs, des diodes, des transistors, ..., etc.



- ✓ Composants d'électroniques numériques comprennent des microprocesseurs, des microcontrôleurs, des mémoires informatique, .... etc.



- ✓ Composants qui agissent comme une interface humaine tels que des afficheurs, des haut-parleurs, des buzzers, des commutateurs, ..., etc.



- ✓ Composants d'électroniques de puissance comprennent des ferrites, des thyristors, des fusibles, des relais, ...etc.



## 1.4. Supports de stockage

### 1.4.1. Définition des supports de stockage

Les supports de stockage permettent de stocker les informations d'une façon permanente. De nos jours, le stockage des informations est garanti par un support physique, c'est-à-dire matériel.

### 1.4.2. Les unités de mesure en informatique

La capacité de la mémoire, ou en d'autres termes la taille de la mémoire, est la quantité d'informations pouvant être stockées dans la mémoire (unité de stockage). La capacité peut être exprimée en bits, qui est l'élément de base pour représenter l'information.

On mesure des capacités importantes en :

Octet	kilo-octet (KO )	Méga-octet ( MO)	Giga-octet ( GO)	Téra-octet (To)
1 Octet = 8 bits	1 KO= 1024 octets = 2 <sup>10</sup> octets	1 MO=1024 KO = 2 <sup>20</sup> octets	1 GO=1024 MO = 2 <sup>30</sup> octets	1 TO= 1024 Go =2 <sup>40</sup> octets

### 1.4.3. Type des supports de stockage

On peut classer les supports de stockage en quatre catégories :

#### 1.4.3.1. Les supports optiques

Les supports optiques tels que le disque compact CD, DVD ou le Blu-ray. Ils permettent le stockage des informations sous forme de reliefs détectés par un rayon laser (creux=0, bosse=1).

1. **CD-Rom** : Support de stockage très répandu, qui permet le stockage des textes, des images, des sons, des programmes et des vidéos de qualité moyenne. C'est un support multifonctions. Il permet un enregistrement permanent c'est le cas d'un CD-R enregistrable, il peut également être CD-RW réinscriptible. Leur capacité est d'environ de : 650 Mo (74 minutes) à 800 Mo (90 minutes).



2. **DVD-ROM** : il s'agit d'un disque digital polyvalent (Digital Versatile Disk), est une amélioration en tout point du CD, sauf la surface du support qui est restée la même. La capacité de stockage varie entre 4,7 Go et plus de 15 Go aujourd'hui. Il existe à l'intérieur une protection anti-copie, afin d'empêcher les utilisateurs d'avoir les permissions en écriture pour des logiciels ou applications propriétaires.

- Enregistrable (DVD-R)
- Réinscriptible (DVD-RW)
- DVD-vidéos de capacité aller jusqu'à 17 Go.

#### 3. Disque Blu-ray :

Le disque Blu-ray est une amélioration du DVD, pouvant stocker de 25 Go à 128 Go de données, tout en étant légèrement plus petit que celui-ci selon le modèle. Le terme Blu-ray est un terme anglais, Blu désigne la couleur bleue et ray désigne rayon ; parce que le laser Blu-ray est bleu. Il s'agit d'un support de stockage optique qui offre une grande capacité de stockage tout en ayant la même surface que les CD et les DVD. Utilisé pour stocker des vidéos haute définition, telles que des vidéos HD ou Full HD.



#### 1.4.3.2. Les supports miniaturisés

Les supports miniaturisés tel qu'une clé USB, une carte SD, une Carte microSD et un SSD utilisent des semi-conducteurs pour stocker les 1 et les 0. tel que, un 1 est stocké si le transistor bloque le courant électrique, sinon un 0 est stocké lorsque le courant électrique circule.

### 1. Clé USB :

Grosse comme une clé de voiture, Elle a toutes les caractéristiques de la disquette sans les inconvénients, elle se branche sur le port USB de l'ordinateur. Les opération d'enregistrement et de suppression à volonté, leur capacité varie entre 32 Mo pour les plus anciennes et 4Go pour les plus récentes.



### 2. Carte mémoire :

Les cartes mémoire sont couramment utilisées pour stocker des images numériques dans des appareils photo, pour enregistrer des jeux sur des consoles de jeux vidéo, mais aussi dans des lecteurs MP3 ou des appareils électroniques professionnels tels que des stations totales.



### 3. SSD :

Un lecteur à l'état solide (SSD ou Solid-state drive), qui est une unité électronique utilisée pour stocker des données, est matériellement plus solide et plus rapide qu'un disque dur , et ses plateaux sont généralement en acier et en verre. Cette caractéristique le rend plus résistant aux chocs et aux vibrations par que les disques mécaniques. De capacités vont de 32Go à 2 To.



#### 1.4.3.3. Les supports de stockage magnétique

Dans ce type de support de stockage, les nombres 1 et 0 sont stockés sous forme de champs magnétiques.

##### 1. Disque dur :

Il s'agit d'un disque magnétique situé dans l'ordinateur, qui permet de stocker des données.

C'est un support de grande capacité qui supporte à la fois le système d'exploitation, les logiciels et les données (textes, images, sons et vidéos). Le disque dur peut aussi être externe, alors connecté à l'unité centrale par une prise USB ou Firewire.

Parmi les caractéristiques du disque dur nous présentons les suivantes :

- ✓ La capacité de stockage exprimée en gigaoctet ou en téraoctet.
- ✓ Le temps d'accès exprimé en millisecondes.
- ✓ La densité exprimée en gigaoctet par plateau.
- ✓ La vitesse de rotation exprimée en tours par minute.
- ✓ L'interface de connexion telles que SCSI, IDE ou SATA.
- ✓ Le taux de transfert moyen exprimé en mégaoctet par seconde.



## 2. Disque dur externe (ou amovible) :

Le disque dur amovible est un disque dur que l'on peut déplacer aujourd'hui aisément, sans avoir besoin de le connecter directement à la carte mère. En général, les disques durs externes



peuvent être connectés directement via USB. Sa capacité de stockage varie de 250 Go à plus de 1 To. L'inconvénient de ce support est qu'il est fragile, là où un disque dur interne est à l'abri dans la machine, le disque dur externe risque à tout moment de tomber ou de subir de trop grandes vibrations, ce qui peut entraîner un dysfonctionnement irréversible. La

réécriture est possible sur ce support.

### 1.4.3.4. Les supports de stockage réseaux,

Comme le Cloud et les serveurs NAS.

#### 1. Cloud :

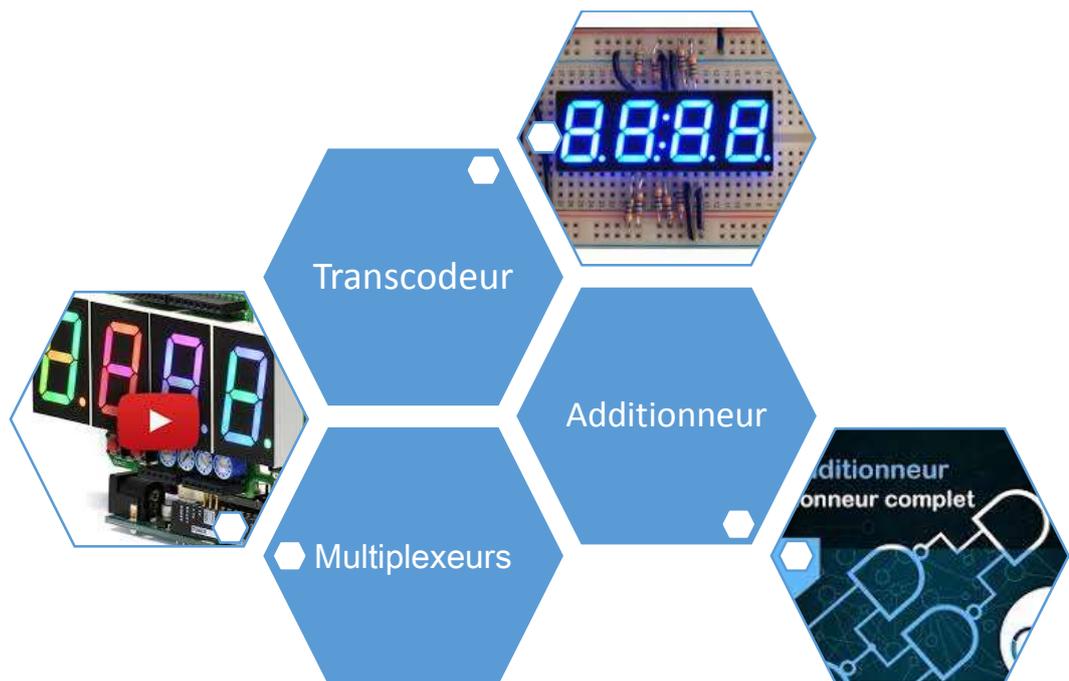
Le cloud est un terme anglais désignant le nuage, qui est l'exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants via un réseau, généralement internet. C'est un serveur virtuel où l'on peut stocker ses données et les récupérer de n'importe où. Permet un travail synchronisé avec toutes les personnes ayant été autorisées au partage.



**2. Serveur NAS (ou disque dur réseaux)** Le serveur NAS fonctionne indépendamment de son propre système d'exploitation. Par conséquent, ce type de support de stockage offre un haut niveau de sécurité pour les données qui y sont stockées.

C'est très pratique car certains disques durs sont généralement embarqués, avec une capacité de stockage pouvant dépasser plusieurs téraoctets dans la plupart des cas.

# Chapitre II : Conception des Circuits Combinatoires



## Chapitre II : Conception des Circuits Combinatoires

### 2.1 Définition

Un circuit combinatoire est un circuit numérique défini par une ou plusieurs fonctions logiques dont les sorties ne dépendent que de l'entrées.

- $S_i = F(E_i)$
- $S_i = F(E_1, E_2, \dots, E_n)$



Les circuits combinés peuvent être utilisés pour créer d'autres circuits plus complexes. Voici quelques exemples de circuits combinatoires : Demi Additionneur, Additionneur complet, Comparateur, Multiplexeur, Démultiplexeur, Encodeur et Décodeur...etc.

### 2.2. Synthèse d'une fonction combinatoire

Pour étudier et réaliser un circuit combinatoire il faut suivre les étapes suivantes :

- 1- Comprendre le fonctionnement du système.
- 2- Définir les variables d'entrée.
- 3- Déterminer les variables de sortie.
- 4- Etablir la table de vérité.
- 5- Extraire les expressions algébriques des sorties.
- 6- Effectuer les opérations de simplifications (algébrique ou par Karnaugh).
- 7- Faire le logigramme avec un minimum de portes logiques.

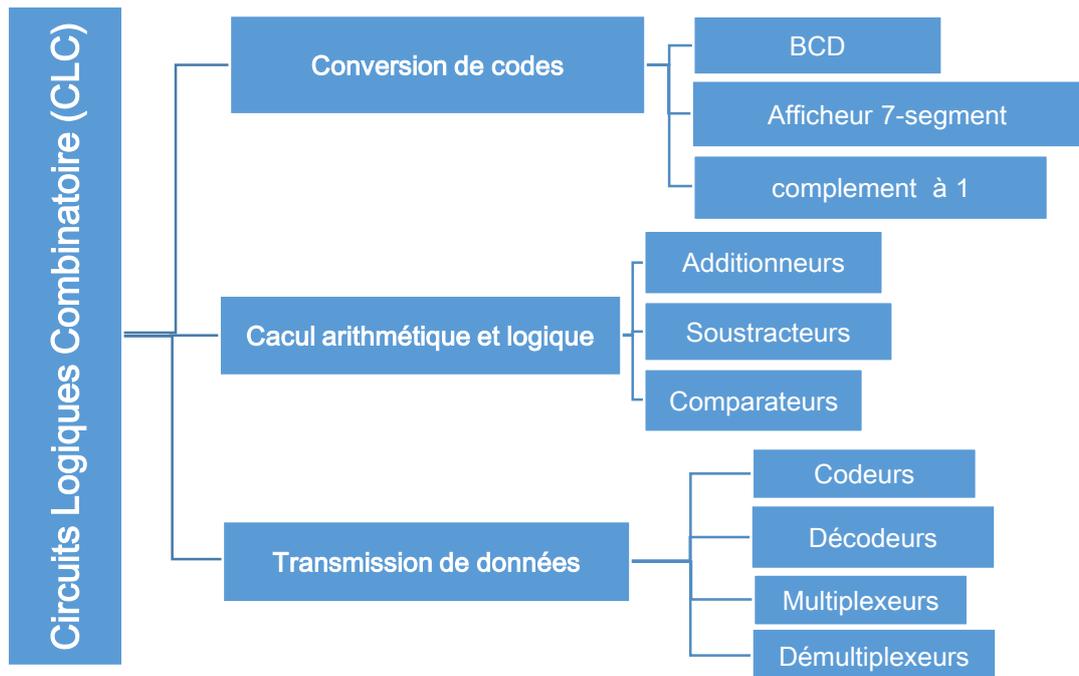
Synthétiser un circuit combinatoire revient à déterminer son organigramme simplifié à partir de sa fonction logique. En général, le processus de synthèse est le suivant :

- a- Schéma symbolique : construire le schéma symbolique en définissant les entrées et les sorties de la fonction logique.
- b- Table de vérité : construire la table de vérité.
- c- Equations de sorties : extraire les équations de sorties via la table de vérité, simplifier ces fonctions algébriquement à l'aide des théorèmes de l'algèbre de Boole ou graphiquement via les tables de Karnaugh.
- d- Schéma logique : Dessiner le schéma du circuit logique avec des ports logiques (NOT, AND, OR, NAND, NOR, ...etc)

### 2.3. Classification

Les circuits logiques combinatoires sont classés en trois catégories [2]:

- Les circuits combinatoire de conversion de codes tels que les convertisseurs de code comme l'afficheur à sept segments;
- Les circuits combinatoire de calcul arithmétiques et logiques tels que additionneurs, les soustracteurs et les comparateurs;
- Les circuits combinatoire d'aiguillage et de transmission de données tels que les codeurs, les décodeurs, les multiplexeurs et les démultiplexeurs.



➤ **Les circuits combinatoires de conversion de codes** : comme son nom l'indique, est un circuit qui transforme une information codée sur  $n$  bits en un autre code codé sur  $m$  bits. Bien que tous les ordinateurs numériques traitent des nombres binaires, mais il existe des situations où la représentation binaire naturelle des nombres n'est pas pratique ou est inefficace ce qui nécessite des codes plus appropriés. Dans ce cas, les circuits de conversion d'un code vers un autre code sont utilisés pour représenter la même information [2].

➤ **Les circuits combinatoires de calcul arithmétiques et logiques** : Ce sont généralement des circuits combinatoires qui permettent d'effectuer des calculs arithmétiques et logiques sur des entiers ou des nombres à virgule flottante, pour les opérations arithmétiques, nous avons des additionneurs, des soustracteurs et des multiplicateurs, pour des opérations logiques nous avons des négations, des ET, des OU ou des OU-Exclusifs. Ces circuits se trouvent couramment dans les processeurs des ordinateurs communément appelées unité arithmétique et logique (UAL).

➤ **Les circuits combinatoires de transmission de données** : Ce sont des circuits qui permettent d'aiguiller les informations binaires via des lignes électriques appelées BUS à partir d'une petite mémoire appelée des registres ou des interrupteurs vers une destination telle qu'un registre ou un afficheur. Comme exemples de ce type de circuits nous avons les décodeurs et les multiplexeurs [2].

## 2.4. Circuits combinatoires usuelles

### 2.4.1. Additionneur

L'opération d'addition est l'une des opérations très courantes dans les microprocesseurs et dans l'unité arithmétique, où elle est utilisée pour l'incrémement du compteur de programme et pour les calculs d'adresses. Il existe nombreuses façons de construire des additionneurs en termes de temps et de nombre de portes logiques utilisées [3].

#### 2.4.1.1. Demi-Additionneur

Le demi-additionneur est un circuit combinatoire qui permet d'effectuer la somme de deux nombres binaires A et B sur un seul bit. En sortie nous avons la somme S et la retenue R.

a-Schéma symbolique :



Dans le système binaire, l'addition sur un bit est effectuée comme suit :

$$\left\{ \begin{array}{l} 0 + 0 = 0 \text{ retenue} = 0 \\ 0 + 1 = 1 \text{ retenue} = 0 \\ 1 + 0 = 1 \text{ retenue} = 0 \\ 1 + 1 = 0 \text{ retenue} = 1 \end{array} \right.$$

b-Table de vérité :

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

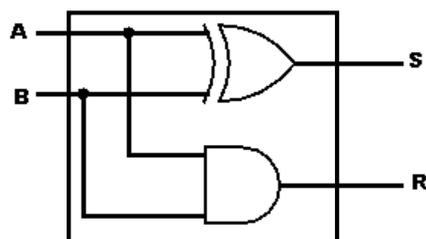
c- Equations de sortie :

À partir de la table de vérité, on trouve les équations de sortie suivantes :

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$R = AB$$

d-Schéma logique



Demi-additionneur à 1 bit

### 2.4.1.2. Additionneur complet

En système binaire, lorsque vous faites une opération d'addition, vous devez prendre en compte la retenue entrante.

$$\begin{array}{rcccccc}
 R_4 & R_3 & R_2 & R_1 & R_0 = 0 & & R_{i-1} \\
 & A_4 & A_3 & A_2 & A_1 & & A_i \\
 + & B_4 & B_3 & B_2 & B_1 & & + B_i \\
 \hline
 R_4 & S_4 & S_3 & S_2 & S_1 & & R_i & S_i
 \end{array}$$

Pour un additionneur complet à 1 bit, nous avons trois entrées (le premier et le deuxième nombre et la retenue entrante) et deux sorties (la somme et la retenue sortante) :

Les entrées	Les sorties
$A_i$ : le premier nombre sur 1 bit.	$S_i$ : la somme sur 1 bit.
$B_i$ : le deuxième nombre sur 1 bit.	$R_i$ : la retenue sortante sur 1 bit.
$R_{i-1}$ : la retenue entrante sur 1 bit.	

#### a-Schéma symbolique



#### b-Table de vérité :

$A_i$	$B_i$	$R_{i-1}$	$S_i$	$R_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

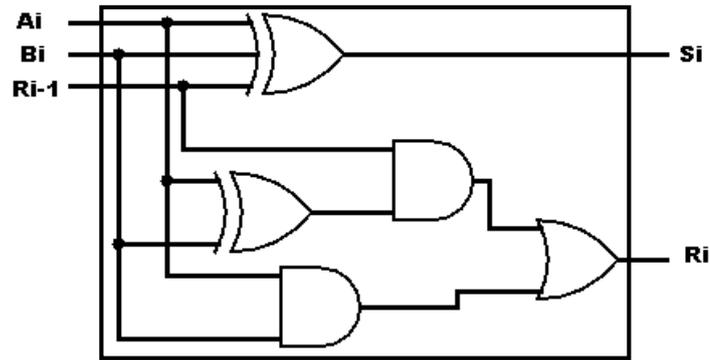
#### c- Equations de sortie :

De la table de vérité on trouve :

$$\begin{aligned}
 S_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= \bar{A}_i (\bar{B}_i R_{i-1} + B_i \bar{R}_{i-1}) + A_i (\bar{B}_i \bar{R}_{i-1} + B_i R_{i-1}) \\
 &= \bar{A}_i (B_i \oplus \bar{R}_{i-1}) + A_i (\bar{B}_i \oplus R_{i-1}) \\
 &= A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

$$\begin{aligned}
 R_i &= \bar{A}_i B_i R_{i-1} + A_i \bar{B}_i R_{i-1} + A_i B_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= R_{i-1} (\bar{A}_i B_i + A_i \bar{B}_i) + A_i B_i (\bar{R}_{i-1} + R_{i-1}) \\
 &= R_{i-1} (A_i \oplus B_i) + A_i B_i
 \end{aligned}$$

d-Schéma logique :



Additionneur complet à 1 bit

### 2.4.1.3. Additionneur à 4 bits

Un additionneur complet de 4 bits est un circuit qui additionne deux nombres  $A(A_4A_3A_2A_1)$  et  $B(B_4B_3B_2B_1)$  de 4 bits chacun. De plus, il doit tenir compte de la retenue entrante. En sortie, nous aurons deux éléments, la somme sur 4 bits ainsi que la retenue sortante sur 1 bit (le cinquième bit en sortie). Au total, ce circuit a donc 9 bits d'entrée et 5 bits de sortie. C'est à dire qu'avec 9 entrées nous avons  $2^9 = 512$  combinaisons. Par conséquent, Il faut trouver une solution plus facile et plus efficace pour concevoir ce circuit.

Lorsque l'addition se fait dans le système binaire, on additionne bit par bit en commençant à partir du poids faible et à chaque fois on propage la retenue sortante au bit du rang supérieur jusqu'à le bit du poids fort. Par conséquent, un additionneur complet de 4 bit peut être réalisé via l'association en série de quatre circuits additionneurs complets à 1 bit.

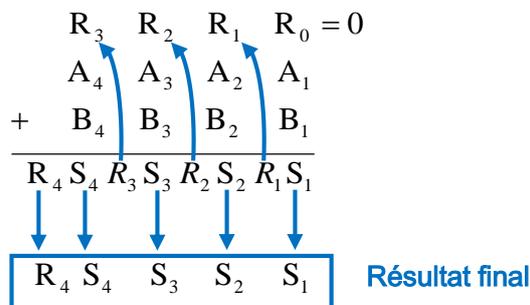
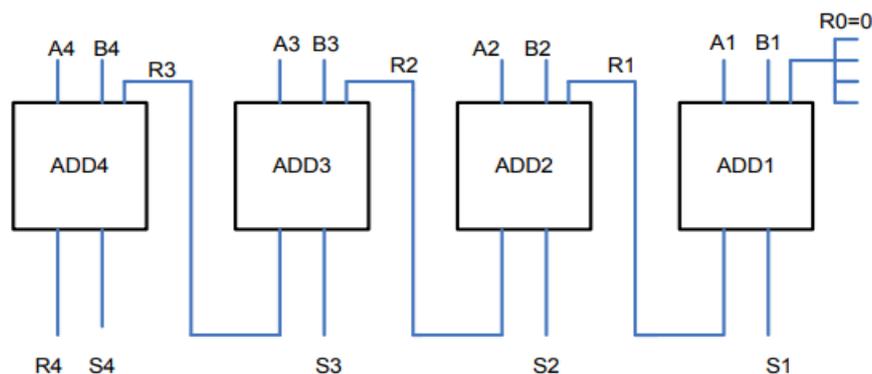


Schéma logique d'un additionneur complet à 4 bits



En général, pour un **additionneur de n bits** : Pour additionner deux nombres de n bits, il suffit d'enchaîner entre n additionneurs complets de 1 bit. Ainsi, la retenue se propage d'un additionneur à l'autre. Cet additionneur de n bits est appelé un additionneur série. Bien que tous les chiffres de deux nombres de n bits soient disponibles simultanément au début du calcul, le temps de calcul est déterminé par la propagation de la retenue à travers les n additionneurs de 1 bit [4].

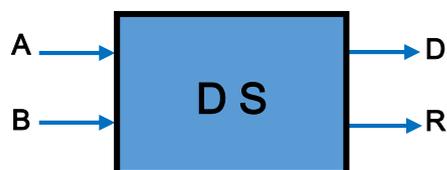
## 2.4.2. Soustracteur

Pour soustraire les deux nombres A et B, la même procédure que pour l'addition peut être suivie. En pratique, il y a un problème de représentation des nombres signés dans le cas où le nombre B est supérieur au nombre A. Pour résoudre ce problème, on accepte de représenter les nombres négatifs, puis la soustraction ramenée à l'addition. En général, la représentation utilisée est la représentation en complément vrai (Cà2).

### 2.4.2.1. Demi-soustracteur

Le demi soustracteur est un circuit combinatoire qui permet d'effectuer la différence arithmétique de deux nombres A et B sur un seul bit. En sortie nous avons la différence D et la retenue R.

a-Schéma symbolique :



Dans le système binaire, la soustraction sur un bit est effectuée comme suit :

$$\begin{cases} 0 - 0 = 0 & \text{retenue} = 0 \\ 0 - 1 = 1 & \text{retenue} = 1 \\ 1 - 0 = 1 & \text{retenue} = 0 \\ 1 - 1 = 0 & \text{retenue} = 0 \end{cases}$$

b-Table de vérité :

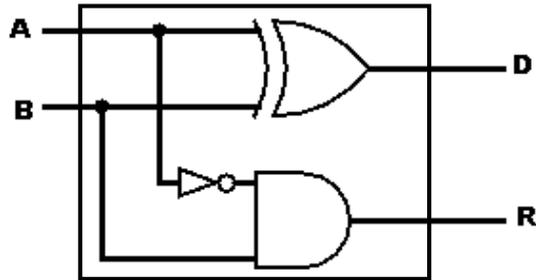
A	B	D	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

c- Equations de sortie :

À partir de la table de vérité, on trouve les équations de sortie suivantes :

$$\begin{cases} D = \bar{A}B + A\bar{B} = A \oplus B \\ R = \bar{A}B \end{cases}$$

d-schéma logique :



Demi-soustracteur à 1 bit

### 2.4.2.2. Soustracteur complet

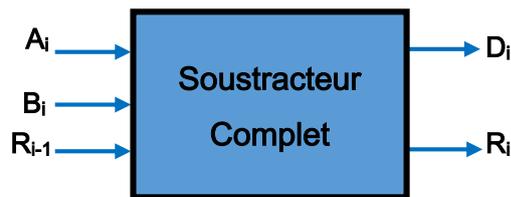
En système binaire, lorsque vous faites une opération de soustraction, vous devez prendre en compte la retenue entrante ( $R_{i-1}$ ).

$$\begin{array}{r}
 R_4 \quad R_3 \quad R_2 \quad R_1 \quad R_0 = 0 \\
 A_4 \quad A_3 \quad A_2 \quad A_1 \\
 - \quad B_4 \quad B_3 \quad B_2 \quad B_1 \\
 \hline
 D_4 \quad D_3 \quad D_2 \quad D_1
 \end{array}
 \qquad
 \begin{array}{r}
 R_{i-1} \\
 A_i \\
 - \quad B_i \\
 \hline
 R_i \quad D_i
 \end{array}$$

Pour un soustracteur complet à 1 bit, nous avons trois entrées (le premier et le deuxième nombre et la retenue entrante) et deux sorties (la différence et la retenue sortante) :

Les entrées	Les sorties
$A_i$ : le premier nombre sur 1 bit.	$D_i$ : la différence sur 1 bit.
$B_i$ : le deuxième nombre sur 1 bit.	$R_i$ : la retenue sortante sur 1 bit.
$R_{i-1}$ : la retenue entrante sur 1 bit.	

a-Schéma symbolique



b-Table de vérité :

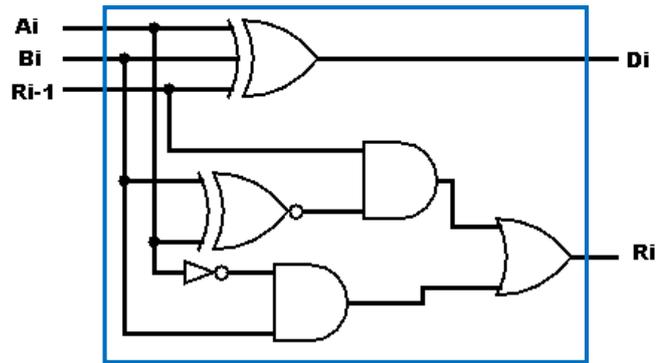
$A_i$	$B_i$	$R_{i-1}$	$D_i$	$R_i$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

**c- Equations de sortie :**

$$\begin{aligned}
 D_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= \bar{A}_i (\bar{B}_i R_{i-1} + B_i \bar{R}_{i-1}) + A_i (\bar{B}_i \bar{R}_{i-1} + B_i R_{i-1}) \\
 &= \bar{A}_i (B_i \oplus \bar{R}_{i-1}) + A_i (\overline{B_i \oplus R_{i-1}}) \\
 &= A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

$$\begin{aligned}
 R_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + \bar{A}_i B_i R_{i-1} + A_i B_i R_{i-1} \\
 &= R_{i-1} (\bar{A}_i \bar{B}_i + A_i B_i) + \bar{A}_i B_i (\bar{R}_{i-1} + R_{i-1}) \\
 &= R_{i-1} (\overline{A_i \oplus B_i}) + \bar{A}_i B_i
 \end{aligned}$$

**d-Schéma logique :**

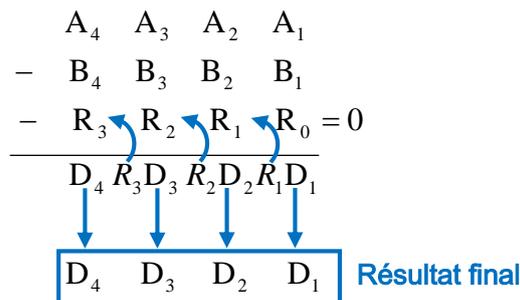


**Soustracteur complet à 1 bit**

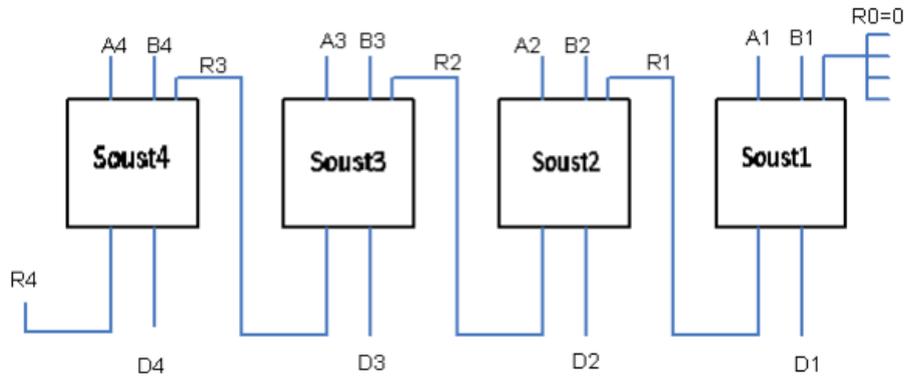
**2.4.2.3. Soustracteur à 4 bits**

Un soustracteur complet de 4 bits est un circuit qui fait la différence deux nombres A(A<sub>4</sub>A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>) et B(B<sub>4</sub>B<sub>3</sub>B<sub>2</sub>B<sub>1</sub>) de 4 bits chacun. De plus, il doit tenir compte de la retenue entrante. En sortie, nous aurons deux éléments, la différence sur 4 bits ainsi que la retenue sortante sur 1 bit (le cinquième bit en sortie). Au total, ce circuit a donc 9 bits d'entrée et 5 bits de sortie. C'est à dire qu'avec 9 entrées nous avons 2<sup>9</sup> = 512 combinaisons. Par conséquent, Il faut trouver une solution plus simple et plus efficace pour concevoir ce circuit de soustraction.

Lorsque la soustraction se fait dans le système binaire, on soustrait bit par bit en commençant à partir du poids faible et à chaque fois on propage la retenue sortante au bit du rang supérieur jusqu'à le bit du poids fort. Par conséquent, un soustracteur complet de 4 bit peut être réalisé via l'association en série de quatre circuits soustracteur complets à 1 bit.



## Schéma logique d'un soustracteur complet à 4 bits



### 2.4.3. Additionneur/Soustracteur

Le circuit d'addition/soustraction effectue à la fois l'addition et la soustraction. Plus de deux entrées, ce circuit possède une troisième entrée pour la commande Cmd, la sélection de l'opération doit être faite en fonction de la valeur de la commande  $E_C$ . Le circuit calcule la somme de deux nombres A et B si  $E_C$  vaut 1 et la différence entre A et B si  $E_C$  vaut 0.

### 2.4.4. Comparateur

#### 2.4.4.1. Comparateur à 1 bit

Un circuit comparateur à 1 bit est un circuit combinatoire qui permet de faire la comparaison entre deux nombres binaires A et B. Pour un comparateur à 1 bit, nous avons deux entrées (le premier et le deuxième nombre) et trois sorties (égalité, inférieur et supérieur) :

Les entrées	Les sorties
<b>A</b> : le premier nombre sur 1 bit.	<b>fe</b> : égalité ( $A=B$ ) sur 1 bit.
<b>B</b> : le deuxième nombre sur 1 bit.	<b>fi</b> : inférieur ( $A < B$ ) sur 1 bit.
	<b>fs</b> : supérieur ( $A > B$ ) sur 1 bit.

a-Schéma symbolique :



b-Table de vérité :

A	B	fs	fe	fi
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

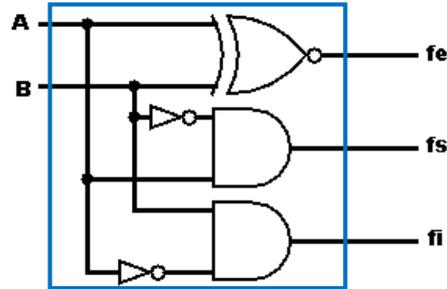
c- Equations de sortie :

$$f_s = A\bar{B}$$

$$f_i = \bar{A}B$$

$$f_e = \bar{A}\bar{B} + AB = \overline{A \oplus B}$$

d-Schéma logique :



Comparateur à 1 bit

#### 2.4.4.2. Comparateur à 2 bits

Un comparateur à deux bits est un circuit combinatoire qui permet de comparer deux nombres binaires A ( $A_2A_1$ ) et B ( $B_2B_1$ ) chacun sur deux bits.

a-Schéma symbolique :



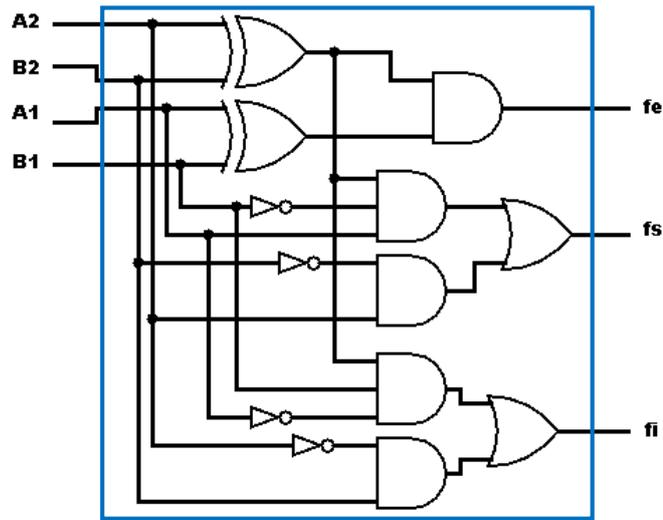
b-Table de vérité :

A <sub>2</sub>	A <sub>1</sub>	B <sub>2</sub>	B <sub>1</sub>	f <sub>s</sub>	f <sub>e</sub>	f <sub>i</sub>
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

**c- Equations de sortie :**

$$\begin{aligned}
 f_s &= \bar{A}_2 A_1 \bar{B}_2 \bar{B}_1 + A_2 \bar{A}_1 \bar{B}_2 \bar{B}_1 + A_2 \bar{A}_1 \bar{B}_2 B_1 + A_2 A_1 \bar{B}_2 \bar{B}_1 + A_2 A_1 \bar{B}_2 B_1 + A_2 A_1 B_2 \bar{B}_1 \\
 &= A_2 \bar{B}_2 + (A_2 \oplus B_2) A_1 \bar{B}_1 \\
 f_i &= \bar{A}_2 \bar{A}_1 \bar{B}_2 B_1 + \bar{A}_2 \bar{A}_1 B_2 \bar{B}_1 + \bar{A}_2 \bar{A}_1 B_2 B_1 + \bar{A}_2 A_1 B_2 \bar{B}_1 + \bar{A}_2 A_1 B_2 B_1 + A_2 \bar{A}_1 B_2 B_1 \\
 &= \bar{A}_2 B_2 + (\bar{A}_2 \oplus B_2) \bar{A}_1 B_1 \\
 f_e &= \bar{A}_2 \bar{A}_1 \bar{B}_2 \bar{B}_1 + \bar{A}_2 A_1 \bar{B}_2 \bar{B}_1 + A_2 \bar{A}_1 B_2 \bar{B}_1 + A_2 A_1 B_2 B_1 \\
 &= (A_2 \oplus B_2) (A_1 \oplus B_1)
 \end{aligned}$$

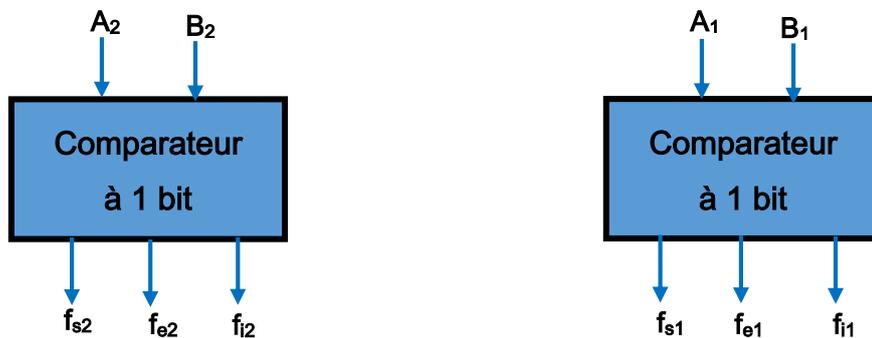
**d-Schéma logique :**



Comparateur à 2 bit

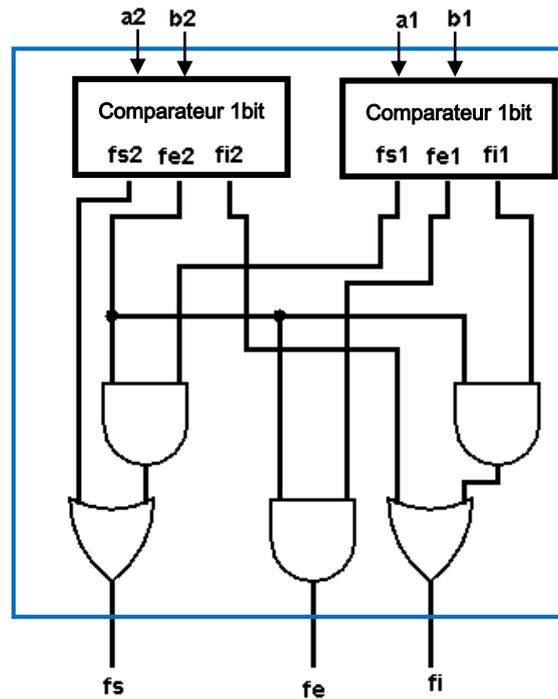
**2.4.4.3. Comparateur 2 bits avec des comparateurs 1 bit**

Un comparateur à 2 bits peut être réalisé en utilisant des comparateurs 1 bit et des portes logiques. Le premier comparateur à 1 bit doit être utilisé pour comparer les bits du poids faible et le second comparateur à 1 bit pour comparer les bits du poids fort. Il est nécessaire de combiner les sorties des deux comparateurs utilisés pour produire la sortie du comparateur final.



$$\begin{aligned}
 A = B \text{ si } A_2 = B_2 \text{ et } A_1 = B_1 &\longrightarrow f_e = (\overline{A_2 \oplus B_2}) (\overline{A_1 \oplus B_1}) = f_{e2} f_{e1} \\
 A > B \text{ si } A_2 > B_2 \text{ ou } (A_2 = B_2 \text{ et } A_1 > B_1) &\longrightarrow f_s = A_2 \bar{B}_2 + (\overline{A_2 \oplus B_2}) A_1 \bar{B}_1 = f_{s2} + f_{e2} f_{s1} \\
 A < B \text{ si } A_2 < B_2 \text{ ou } (A_2 = B_2 \text{ et } A_1 < B_1) &\longrightarrow f_i = \bar{A}_2 B_2 + (\overline{A_2 \oplus B_2}) \bar{A}_1 B_1 = f_{i2} + f_{e2} f_{i1}
 \end{aligned}$$

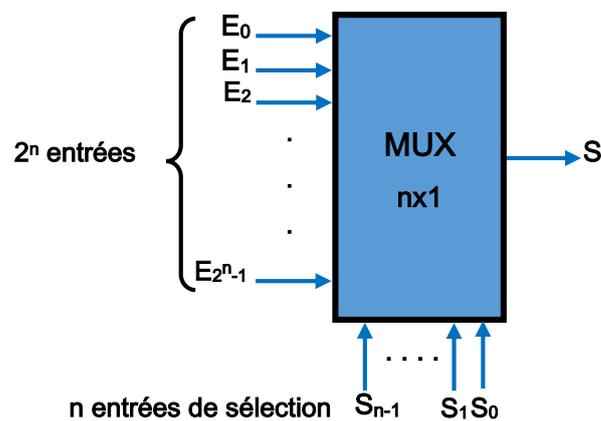
### Schéma logique d'un comparateur à deux entrées à deux bits



#### 2.4.5. Multiplexeur

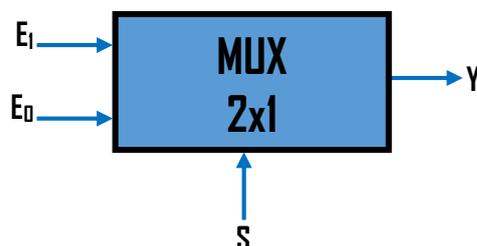
Un multiplexeur est un circuit combinatoire qui agit comme un sélecteur de données en dirigeant une donnée parmi ces données présentes sur ses entrées vers sa sortie [5]. Il possède :

- $2^n$  entrées.
- Une sortie.
- $n$  entrées de commande (sélection).



##### 2.4.5.1. Multiplexeur 2x1

a-Schéma symbolique :



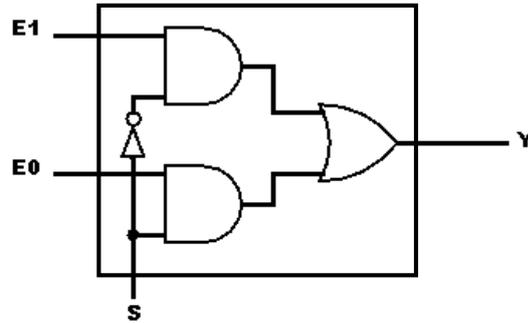
b-Table de vérité :

S	Y
0	E <sub>0</sub>
1	E <sub>1</sub>

c- Equations de sortie :

$$Y = \bar{S}E_0 + S\bar{E}_1$$

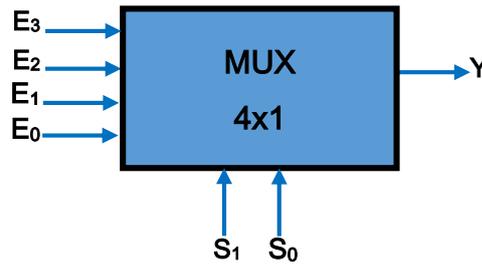
d-Schéma logique :



Multiplexeur 2x1

#### 2.4.5.2. Multiplexeur 4X1

a-Schéma symbolique :



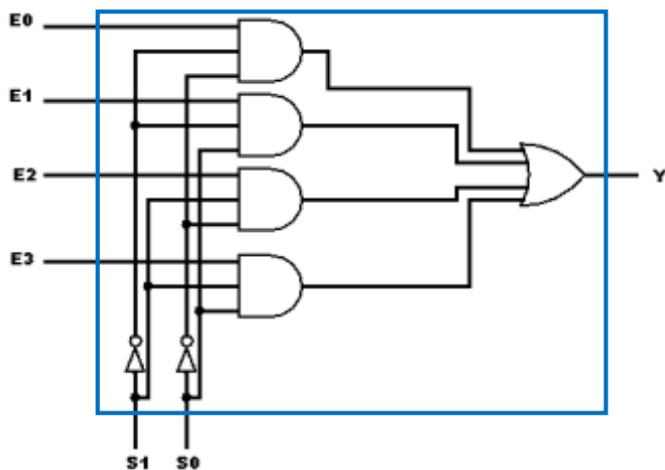
b-Table de vérité :

S <sub>1</sub>	S <sub>0</sub>	Y
0	0	E <sub>0</sub>
0	1	E <sub>1</sub>
1	0	E <sub>2</sub>
1	1	E <sub>3</sub>

c- Equations de sortie :

$$Y = \bar{S}_1\bar{S}_0E_0 + \bar{S}_1S_0E_1 + S_1\bar{S}_0E_2 + S_1S_0E_3$$

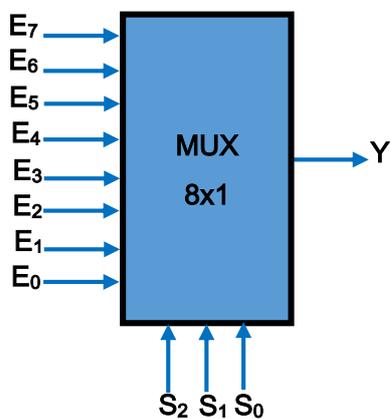
d-Schéma logique :



Multiplexeur 4x1

### 2.4.5.3. Multiplexeur 8X1

a-schéma symbolique :



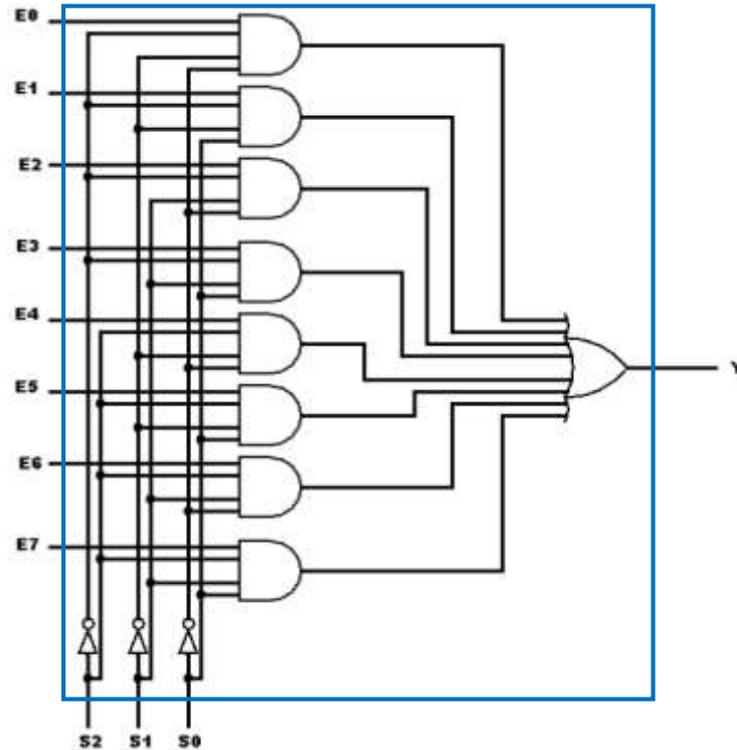
b-Table de vérité :

S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Y
0	0	0	E <sub>0</sub>
0	0	1	E <sub>1</sub>
0	1	0	E <sub>2</sub>
0	1	1	E <sub>3</sub>
1	0	0	E <sub>4</sub>
1	0	1	E <sub>5</sub>
1	1	0	E <sub>6</sub>
1	1	1	E <sub>7</sub>

c- équations de sortie :

$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 E_0 + \bar{S}_2 \bar{S}_1 S_0 E_1 + \bar{S}_2 S_1 \bar{S}_0 E_2 + \bar{S}_2 S_1 S_0 E_3 + S_2 \bar{S}_1 \bar{S}_0 E_4 + S_2 \bar{S}_1 S_0 E_5 + S_2 S_1 \bar{S}_0 E_6 + S_2 S_1 S_0 E_7$$

d-schéma logique :



Multiplexeur 8X1

Pour réaliser une fonction logique à l'aide des multiplexeurs, il faut suivre les étapes suivantes [6] :

- Écrire l'équation de cette fonction logique sous la première forme canonique.
- Écrire l'équation du multiplexeur caractérisé par le nombre d'entrée d'adresse de sorte que le nombre d'entrée de sélection soit égal au nombre d'entrées de la fonction.
- Identifier les deux équations.

Exemple : réalisation d'un additionneur complet à 1 bit à l'aide des multiplexeurs 8X1.

Pour réaliser un additionneur complet à 1 bit à l'aide des multiplexeurs 8X1, nous devons utiliser d'utiliser deux multiplexeurs 8x1, le premier multiplexeur pour donner la fonction de la somme et le second pour donner la retenue.

b-Table de vérité :

$A_i$	$B_i$	$R_{i-1}$	$S_i$	$R_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

### c- Equations de sortie :

-La fonction de la somme

$$S_i = \bar{A}_i \bar{B}_i \bar{R}_{i-1}(0) + \bar{A}_i \bar{B}_i R_{i-1}(1) + \bar{A}_i B_i \bar{R}_{i-1}(1) + \bar{A}_i B_i R_{i-1}(0) + A_i \bar{B}_i \bar{R}_{i-1}(1) + A_i \bar{B}_i R_{i-1}(0) + A_i B_i \bar{R}_{i-1}(0) + A_i B_i R_{i-1}(1)$$

$$S_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(E_0) + \bar{S}_2 \bar{S}_1 S_0(E_1) + \bar{S}_2 S_1 \bar{S}_0(E_2) + \bar{S}_2 S_1 S_0(E_3) + S_2 \bar{S}_1 \bar{S}_0(E_4) + S_2 \bar{S}_1 S_0(E_5) + S_2 S_1 \bar{S}_0(E_6) + S_2 S_1 S_0(E_7)$$

$$S_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(0) + \bar{S}_2 \bar{S}_1 S_0(1) + \bar{S}_2 S_1 \bar{S}_0(1) + \bar{S}_2 S_1 S_0(0) + S_2 \bar{S}_1 \bar{S}_0(1) + S_2 \bar{S}_1 S_0(0) + S_2 S_1 \bar{S}_0(0) + S_2 S_1 S_0(1)$$

Tel que, on pose

$$S_2 = A_i$$

$$S_1 = B_i$$

$$S_0 = R_{i-1}$$

$$E_0 = 0, E_1 = 1, E_2 = 1, E_3 = 0, E_4 = 1, E_5 = 0, E_6 = 0, E_7 = 1$$

-La fonction de la retenue

$$R_i = \bar{A}_i \bar{B}_i \bar{R}_{i-1}(0) + \bar{A}_i \bar{B}_i R_{i-1}(0) + \bar{A}_i B_i \bar{R}_{i-1}(0) + \bar{A}_i B_i R_{i-1}(1) + A_i \bar{B}_i \bar{R}_{i-1}(0) + A_i \bar{B}_i R_{i-1}(1) + A_i B_i \bar{R}_{i-1}(1) + A_i B_i R_{i-1}(1)$$

$$R_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(E_0) + \bar{S}_2 \bar{S}_1 S_0(E_1) + \bar{S}_2 S_1 \bar{S}_0(E_2) + \bar{S}_2 S_1 S_0(E_3) + S_2 \bar{S}_1 \bar{S}_0(E_4) + S_2 \bar{S}_1 S_0(E_5) + S_2 S_1 \bar{S}_0(E_6) + S_2 S_1 S_0(E_7)$$

$$R_i = \bar{S}_2 \bar{S}_1 \bar{S}_0(0) + \bar{S}_2 \bar{S}_1 S_0(0) + \bar{S}_2 S_1 \bar{S}_0(0) + \bar{S}_2 S_1 S_0(1) + S_2 \bar{S}_1 \bar{S}_0(0) + S_2 \bar{S}_1 S_0(1) + S_2 S_1 \bar{S}_0(1) + S_2 S_1 S_0(1)$$

Tel que, on pose

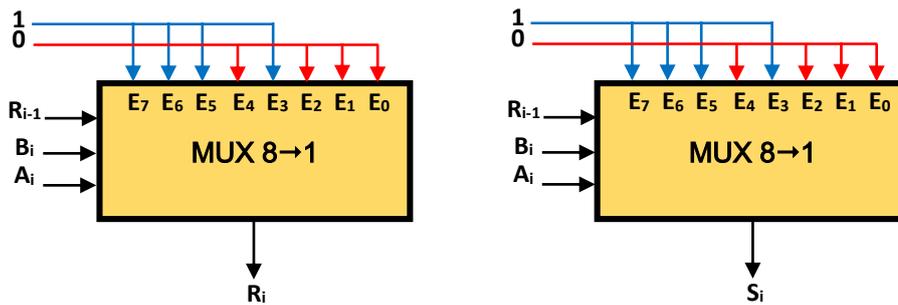
$$S_2 = A_i$$

$$S_1 = B_i$$

$$S_0 = R_{i-1}$$

$$E_0 = 0, E_1 = 1, E_2 = 1, E_3 = 0, E_4 = 1, E_5 = 0, E_6 = 0, E_7 = 1$$

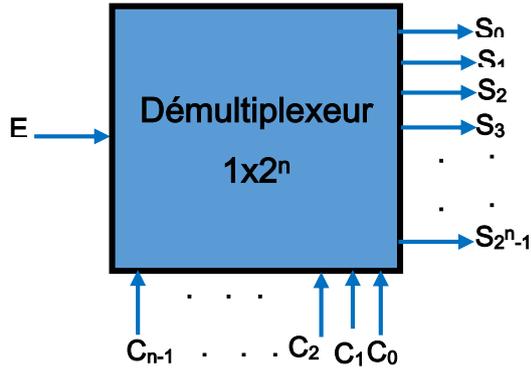
### d-Schéma logique :



### 2.4.6. Démultiplexeurs

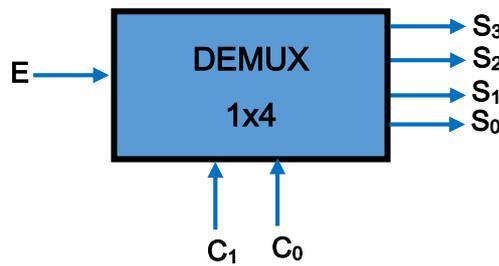
Un démultiplexeur est un circuit logique combinatoire qui joue le rôle inverse d'un multiplexeur, permettant aux données d'entrée de passer sur l'une des sorties en fonction des valeurs des entrées de sélection. Il possède :

- Une entrée.
- $2^n$  sorties.
- $n$  entrées de commande (sélection).



2.4.6.1. Démultiplexeurs 1X4

a-schéma symbolique :



b-Table de vérité :

C <sub>1</sub>	C <sub>0</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

c- équations de sortie :

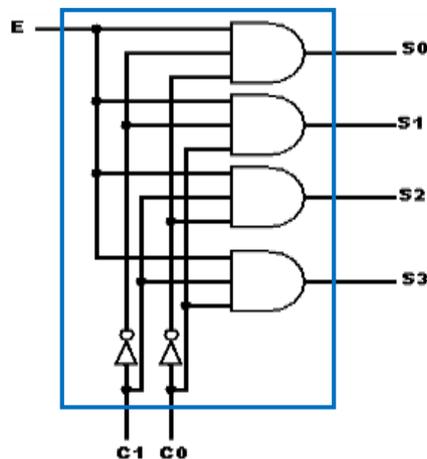
$$S_0 = \bar{C}_1 \bar{C}_0 E$$

$$S_1 = \bar{C}_1 C_0 E$$

$$S_2 = C_1 \bar{C}_0 E$$

$$S_3 = C_1 C_0 E$$

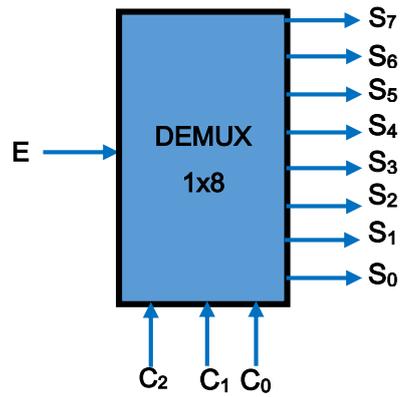
d-Schéma logique :



Démultiplexeurs 1X4

## 2.4.6.2. Démultiplexeurs 1X8

a-Schéma symbolique :



b-Table de vérité :

C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	S <sub>7</sub>	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	E
0	0	1	0	0	0	0	0	0	E	0
0	1	0	0	0	0	0	0	E	0	0
0	1	1	0	0	0	0	E	0	0	0
1	0	0	0	0	0	E	0	0	0	0
1	0	1	0	0	E	0	0	0	0	0
1	1	0	0	E	0	0	0	0	0	0
1	1	1	E	0	0	0	0	0	0	0

c- Equations de sortie :

$$S_0 = \overline{C_2} \overline{C_1} \overline{C_0} E$$

$$S_1 = \overline{C_2} \overline{C_1} C_0 E$$

$$S_2 = \overline{C_2} C_1 \overline{C_0} E$$

$$S_3 = \overline{C_2} C_1 C_0 E$$

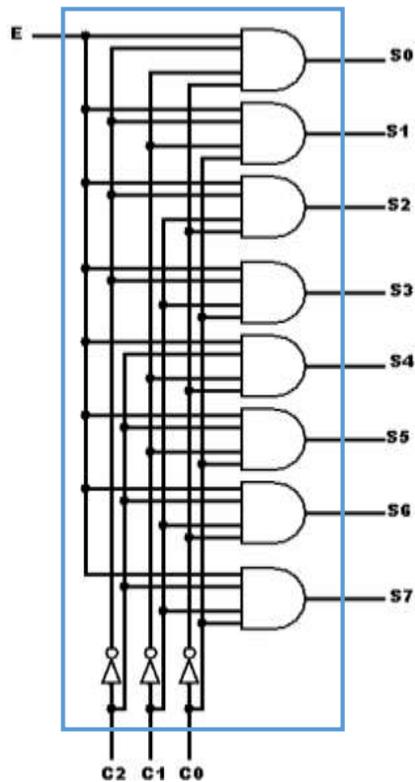
$$S_4 = C_2 \overline{C_1} \overline{C_0} E$$

$$S_5 = C_2 \overline{C_1} C_0 E$$

$$S_6 = C_2 C_1 \overline{C_0} E$$

$$S_7 = C_2 C_1 C_0 E$$

d-Schéma logique :

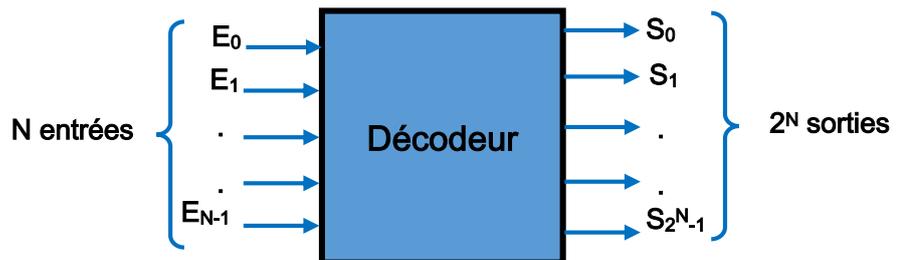


Démultiplexeurs 1X8

### 2.4.7. Décodeur binaire

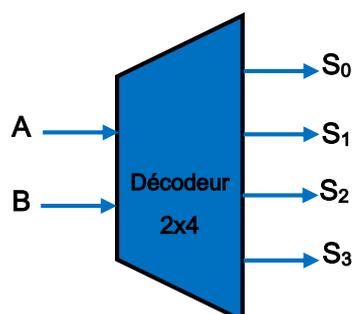
Un décodeur binaire est un circuit logique combinatoire, dans lequel pour chaque combinaison d'entrées, une seule sortie soit active à la fois, ce circuit est se compose de :

- N entrées.
- $2^N$  sorties.



#### 2.4.7.1. Décodeur binaire 2x4

a-schéma symbolique :



b-Table de vérité :

A	B	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

c- équations de sortie :

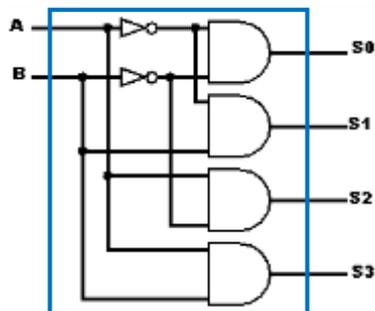
$$S_0 = \bar{A}\bar{B}$$

$$S_1 = \bar{A}B$$

$$S_2 = A\bar{B}$$

$$S_3 = AB$$

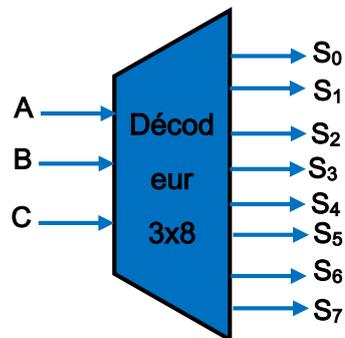
d-schéma logique :



Décodeur binaire 2x4

#### 2.4.7.2. Décodeur binaire 3X8

a-Schéma symbolique :



b-Table de vérité :

A	B	C	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

c- Equations de sortie :

$$S_0 = \overline{A}\overline{B}\overline{C}$$

$$S_1 = \overline{A}\overline{B}C$$

$$S_2 = \overline{A}B\overline{C}$$

$$S_3 = \overline{A}BC$$

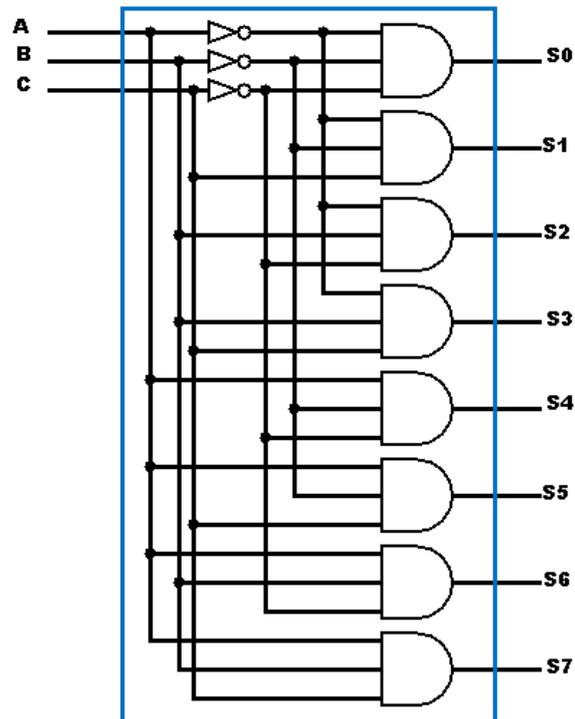
$$S_4 = A\overline{B}\overline{C}$$

$$S_5 = A\overline{B}C$$

$$S_6 = ABC\overline{C}$$

$$S_7 = ABC$$

d-Schéma logique :

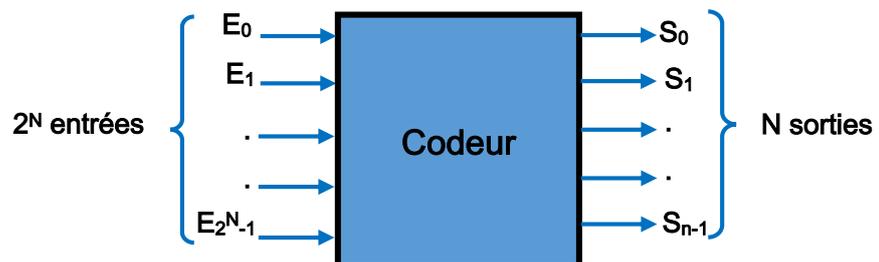


Décodeur binaire 3X8

#### 2.4.8. Encodeur binaire (codeur)

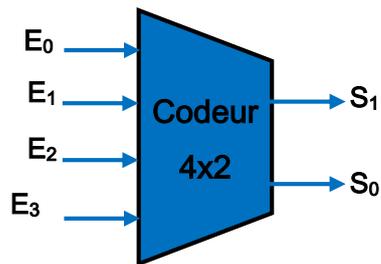
Un encodeur binaire, autrement dit codeur, est un circuit logique combinatoire, dans lequel pour chaque combinaison d'entrées on va avoir son numéro en binaire à la sortie, Il joue le rôle inverse d'un décodeur. Ce circuit est se compose de :

- $2^N$  entrées
- N sorties



### 2.4.8.1. Encodeur binaire 4x2

a-Schéma symbolique :



b-Table de vérité :

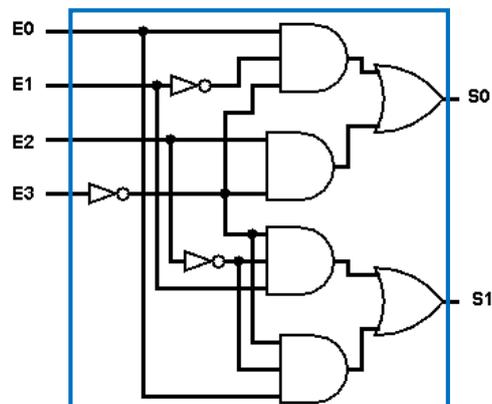
E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	S <sub>1</sub>	S <sub>0</sub>
1	X	X	X	0	0
0	1	X	X	0	1
0	0	1	X	1	0
0	0	0	1	1	1

c- équations de sortie :

$$S_0 = \bar{E}_3 E_2 + \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 = \bar{E}_3 (E_2 + \bar{E}_2 \bar{E}_1 E_0) = \bar{E}_3 (E_2 + \bar{E}_1 E_0) = \bar{E}_3 E_2 + \bar{E}_3 \bar{E}_1 E_0$$

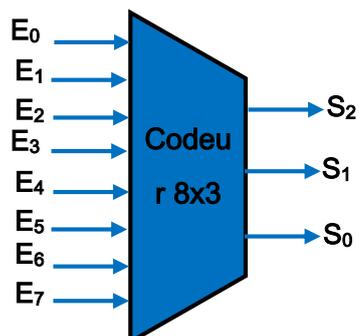
$$S_1 = \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 = \bar{E}_3 \bar{E}_2 (E_1 + \bar{E}_1 E_0) = \bar{E}_3 \bar{E}_2 (E_1 + E_0) = \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_3 \bar{E}_2 E_0$$

d-Schéma logique :



### 2.4.8.2. Encodeur binaire 8x3

a-Schéma symbolique :



**b-Table de vérité :**

E <sub>7</sub>	E <sub>6</sub>	E <sub>5</sub>	E <sub>4</sub>	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	X	X	0	0	1
0	0	1	X	X	X	X	X	0	1	0
0	0	0	1	X	X	X	X	0	1	1
0	0	0	0	1	X	X	X	1	0	0
0	0	0	0	0	1	X	X	1	0	1
0	0	0	0	0	0	1	X	1	1	0
0	0	0	0	0	0	0	1	1	1	1

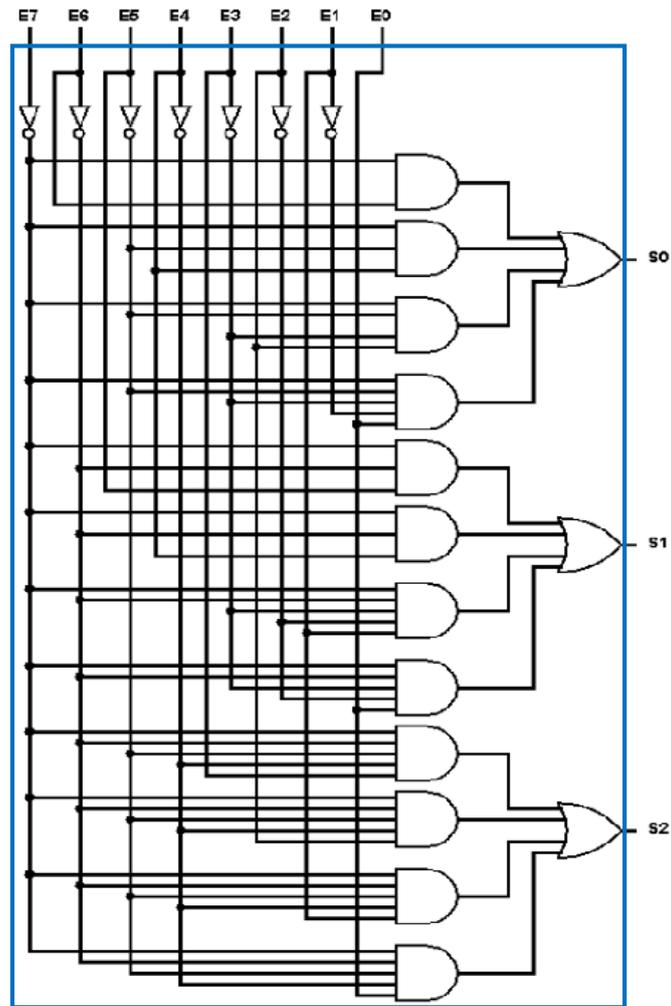
**c- Equations de sortie :**

$$\begin{aligned}
 S_0 &= \bar{E}_7 E_6 + \bar{E}_7 \bar{E}_6 \bar{E}_5 E_4 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 E_2 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 \\
 &= \bar{E}_7 E_6 + \bar{E}_7 \bar{E}_5 E_4 + \bar{E}_7 \bar{E}_5 \bar{E}_3 E_2 + \bar{E}_7 \bar{E}_5 \bar{E}_3 \bar{E}_1 E_0
 \end{aligned}$$

$$\begin{aligned}
 S_1 &= \bar{E}_7 \bar{E}_6 E_5 + \bar{E}_7 \bar{E}_6 \bar{E}_5 E_4 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 \\
 &= \bar{E}_7 \bar{E}_6 E_5 + \bar{E}_7 \bar{E}_6 E_4 + \bar{E}_7 \bar{E}_6 \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_3 \bar{E}_2 E_0
 \end{aligned}$$

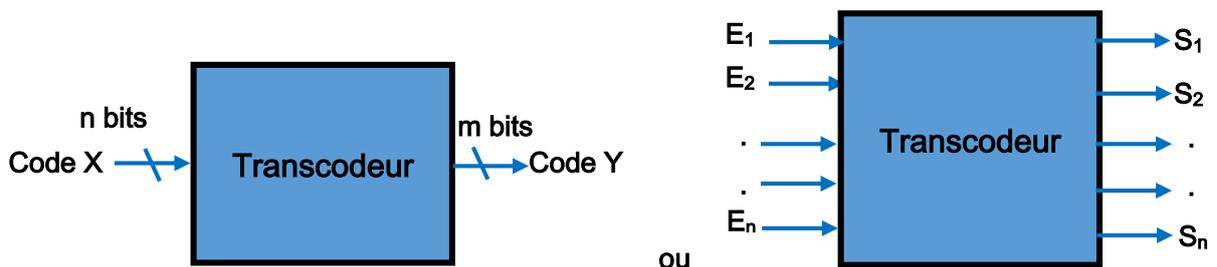
$$\begin{aligned}
 S_2 &= \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_3 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 E_2 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 \bar{E}_3 \bar{E}_2 \bar{E}_1 E_0 \\
 &= \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_3 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_2 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_1 + \bar{E}_7 \bar{E}_6 \bar{E}_5 \bar{E}_4 E_0
 \end{aligned}$$

**d-Schéma logique :**



### 2.4.9. Transcodeur

Un transcodeur est un circuit combinatoire qui transforme un code X sur n bits en entrée en un autre code Y sur m bits en sortie. C'est-à-dire le passage d'un code X vers un autre code Y. Voici quelques exemples de transcodeurs : Binaire  $\Rightarrow$  Binaire réfléchi, BCD  $\Rightarrow$  7-segments, BCD  $\Rightarrow$  XS3, ..., etc.



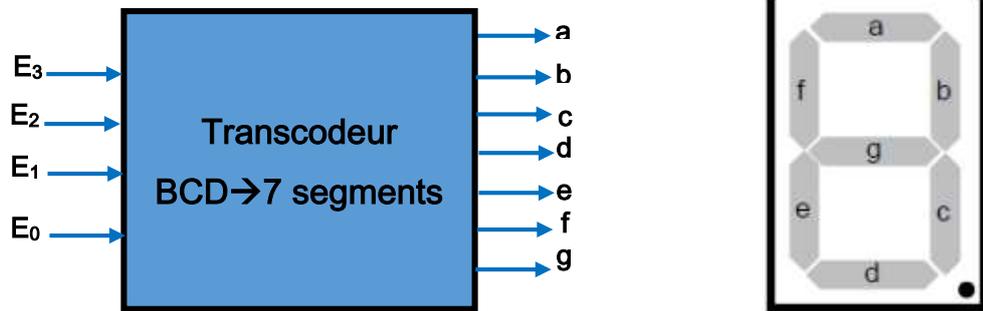
#### 2.4.9.1. Transcodeur 7-segments

##### BCD $\Rightarrow$ code affichage chiffre (afficheur 7-segments)

Le transcodeur 7 segments permet de convertir une entrée de 4 bits ( $E_3E_2E_1E_0$ ) représentée en BDC vers son équivalent décimal via un afficheur numérique, le déclenchement de la sortie

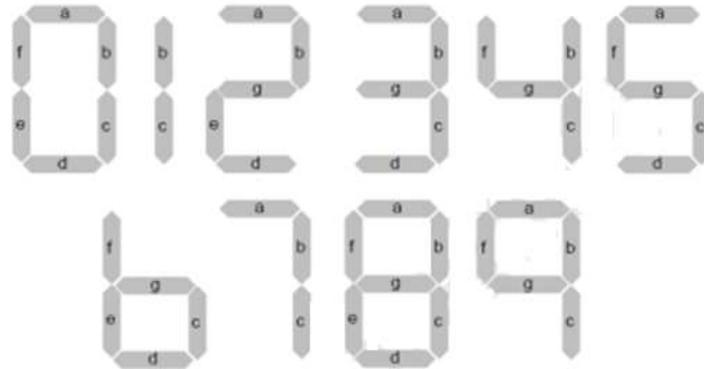
permettra au courant de passer à travers les segments d'un afficheur numérique pour former les chiffres décimaux de 0 jusqu'à 9.

→ a-schéma symbolique



→ b-Table de vérité

Il y'a 6 combinaisons intitulés 10, 11, 12, 13, 14, 15. Les autres chiffres sont affichés comme suit :



Affichage	Table de vérité										
	Entrées				Sorties						
	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

→ c- Equations de sortie

Segment a

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	0	X	X

Segment b

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	1	X	1
01	1	0	X	1
11	1	1	X	X
10	1	0	X	X

$$a(E_3, E_2, E_1, E_0) = E_3 + E_2E_0 + \bar{E}_2E_1 + \bar{E}_2\bar{E}_0$$

$$b(E_3, E_2, E_1, E_0) = \bar{E}_2 + \bar{E}_1\bar{E}_0 + E_1E_0$$

Segment c

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	1	X	1
01	1	1	X	1
11	1	1	X	X
10	0	1	X	X

Segment d

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	0	X	1
01	0	1	X	0
11	1	0	X	X
10	1	1	X	X

$$c(E_3, E_2, E_1, E_0) = E_2 + \bar{E}_1 + E_0$$

$$d(E_3, E_2, E_1, E_0) = E_1\bar{E}_0 + \bar{E}_2E_1 + \bar{E}_2\bar{E}_0 + E_2\bar{E}_1E_0$$

Segment e

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	0	X	1
01	0	0	X	1
11	0	0	X	X
10	1	1	X	X

Segment f

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	1	1	X	1
01	0	1	X	1
11	0	0	X	X
10	0	1	X	X

$$e(E_3, E_2, E_1, E_0) = E_1\bar{E}_0 + E_3E_0 + \bar{E}_2\bar{E}_0$$

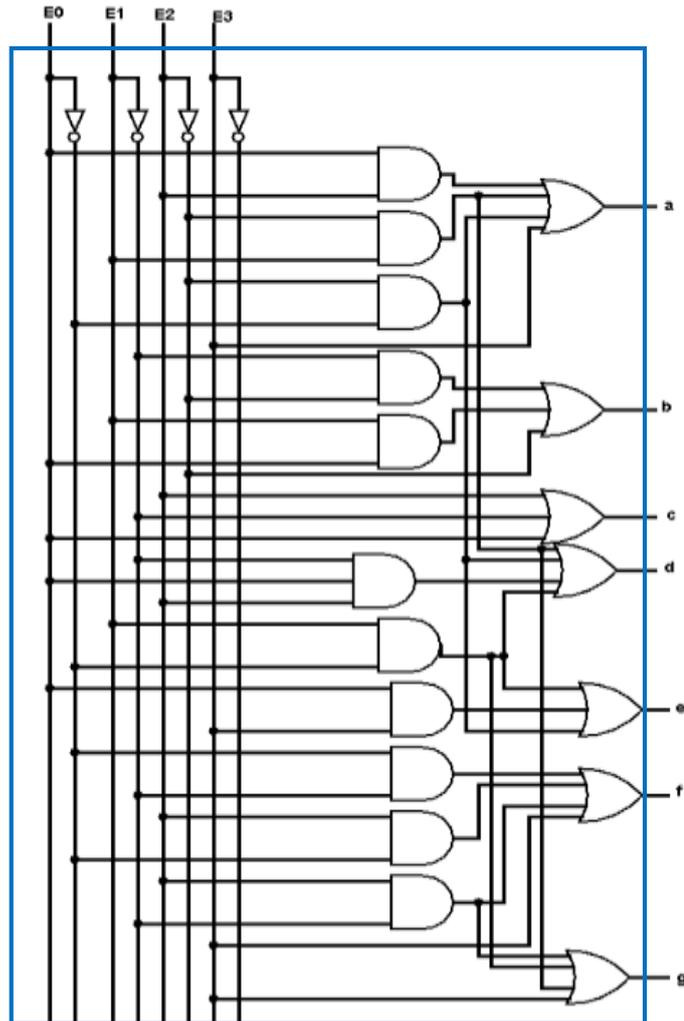
$$f(E_3, E_2, E_1, E_0) = E_3 + \bar{E}_1\bar{E}_0 + E_2\bar{E}_1 + E_2\bar{E}_0$$

Segment g

$E_3E_2$ \ $E_1E_0$	00	01	11	10
00	0	1	X	1
01	0	1	X	1
11	1	0	X	X
10	1	1	X	X

$$g(E_3, E_2, E_1, E_0) = E_3 + E_2\bar{E}_1 + E_1\bar{E}_0 + \bar{E}_2E_1$$

→ d- Schéma logique



Transcodeur BCD → 7 segments

### 2.4.9.2. Transcodeur BCD/EXESS3

Le transcodeur BCD/EXESS3 permet de convertir un nombre représenté en BDC sur 4 bits vers son équivalent en EXESS3 sur 4 bits.

→ a-schéma symbolique



→ b-Table de vérité :

Entrées				Sorties			
E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

→ c- Equations de sortie :

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	0	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	0	1	X	0
01	1	0	X	1
11	1	0	X	X
10	1	0	X	X

$$S_3(E_3, E_2, E_1, E_0) = E_3 + E_2E_1 + E_2E_0$$

$$S_2(E_3, E_2, E_1, E_0) = E_2\bar{E}_1\bar{E}_0 + \bar{E}_2E_0 + \bar{E}_2E_1$$

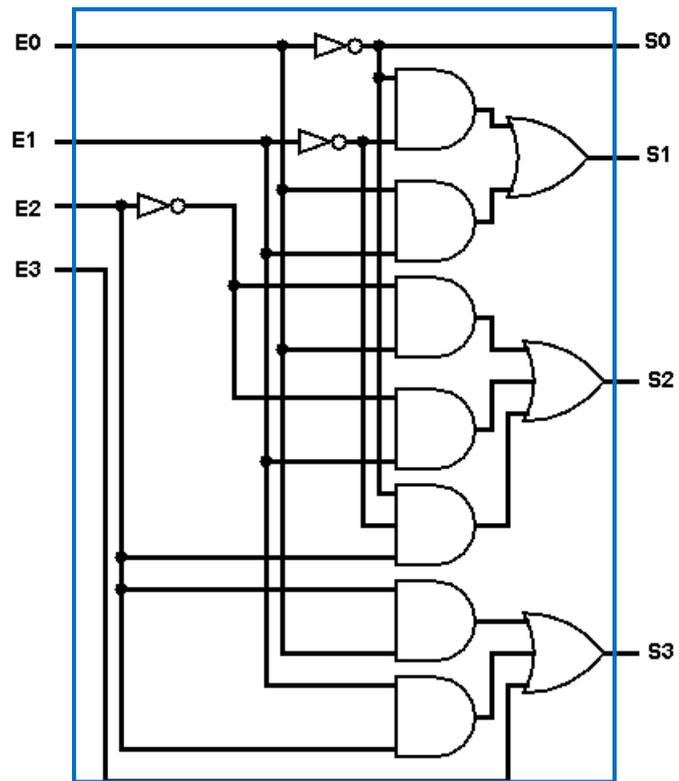
E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	1	1	X	X
10	0	0	X	X

E <sub>3</sub> E <sub>2</sub> \ E <sub>1</sub> E <sub>0</sub>	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	0	0	X	X
10	1	1	X	X

$$S_1(E_3, E_2, E_1, E_0) = \bar{E}_1\bar{E}_0 + E_1E_0$$

$$S_0(E_3, E_2, E_1, E_0) = \bar{E}_0$$

→ d- Schéma logique



Transcodeur BCD/EXCESS3

## Série d'exercices N°1 (Circuits combinatoires)

### Exercice n° 1 :

On veut réaliser un circuit combinatoire permettant d'effectuer une addition ou une soustraction sur 1 bit avec retenue entrante et sortante.

1-établir la table de vérité d'un demi-additionneur qui calcule la somme de deux nombres binaires ( $S_i = A_i + B_i$ ) et la retenue sortante. Puis, établir le schéma logique de ce demi-additionneur.



2-établir la table de vérité d'un additionneur complet à 1bit qui calcule la somme de deux nombres binaires ( $S_i = A_i + B_i + R_{i-1}$ ) et la retenue sortante. Puis, établir le schéma logique de ce demi-additionneur.

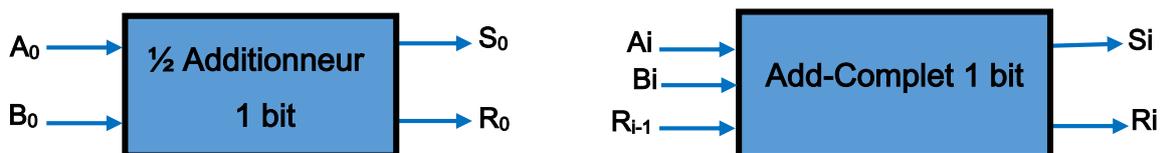


3- Répéter les mêmes questions 1 et 2 pour le demi soustracteur et le soustracteur complet.

4- réaliser un circuit logique qui permet à l'aide d'une commande externe de réaliser soit l'addition, soit la soustraction sur 1 bit avec retenue entrante et sortante.

### Exercice n° 2 :

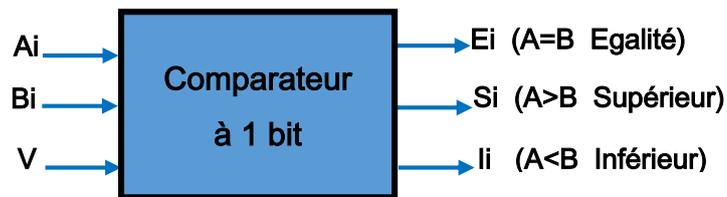
Donner un schéma logique d'un additionneur 3 bits via une cascade à base des circuits demi-additionneurs et additionneurs complets à 1 bit ci-dessous.



### Exercice n° 3 :

La figure ci-dessous montre un comparateur de deux nombres binaires  $A_i$  et  $B_i$  à 1 bit, muni d'une entrée de validation  $V$  permettant d'effectuer la comparaison. Si  $V = 0$ , alors toutes les sorties sont nulles, sinon le processus est le suivant :

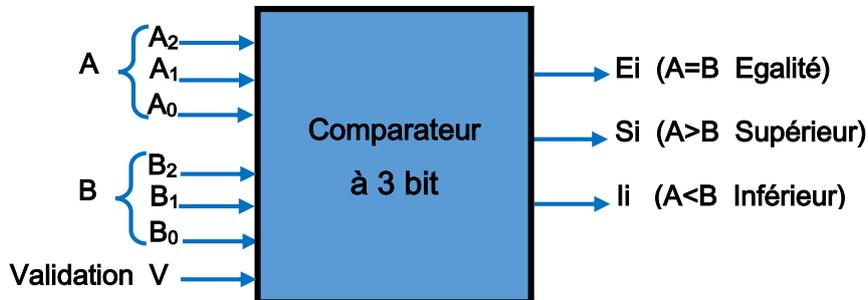
$$\begin{cases} E_i = 1 & \text{si } A_i = B_i \\ S_i = 1 & \text{si } A_i > B_i \\ I_i = 1 & \text{si } A_i < B_i \end{cases}$$



- a- Donner la table de vérité du comparateur,
- b- Donner les équations logiques des sorties du comparateur,
- c- En déduire le schéma logique du comparateur.

**Exercice n° 4 :**

On veut maintenant réaliser un comparateur de deux nombres binaires à trois bits  $A = (A_2A_1A_0)$  et  $B = (B_2B_1B_0)$ , dont le schéma symbolique est donné par la figure ci-dessous à l'aide des trois comparateurs à 1 bits (de l'exercice 3) et des portes logiques 'OU' et 'ET'. On note que  $A_0$  et  $B_0$  sont les bits de poids les plus faibles.



- a- Déterminer les équations simplifiées des sorties E, S et I en fonction des sorties des comparateurs à 1 bit ( $E_i, S_i, I_i$ ) ; avec  $i = 0, 1, 2$ .
- b- En déduire le schéma logique du comparateur à 3 bits.

**Exercice n°5:**

On cherche à afficher les sorties du comparateur (S, I, E) de l'exercice 4 sur un afficheur 7 segments via un transcodeur, comme représenté sur la figure 1, et ceci pour obtenir l'affichage illustré à la figure 2.

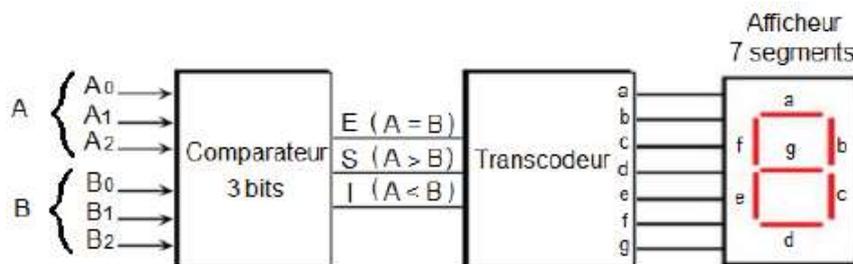


Figure 1

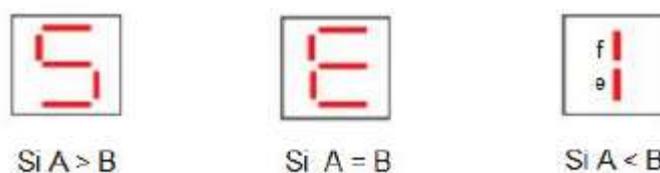


Figure 2

- a- Etablir la table de vérité du transcodeur qui permet de passer du code S, I, E au code 7 segments.
- b- Déterminer les expressions simplifiées des sorties en utilisant le tableau de Karnaugh.
- c- En déduire le schéma interne du transcodeur.

**Exercice n°6:**

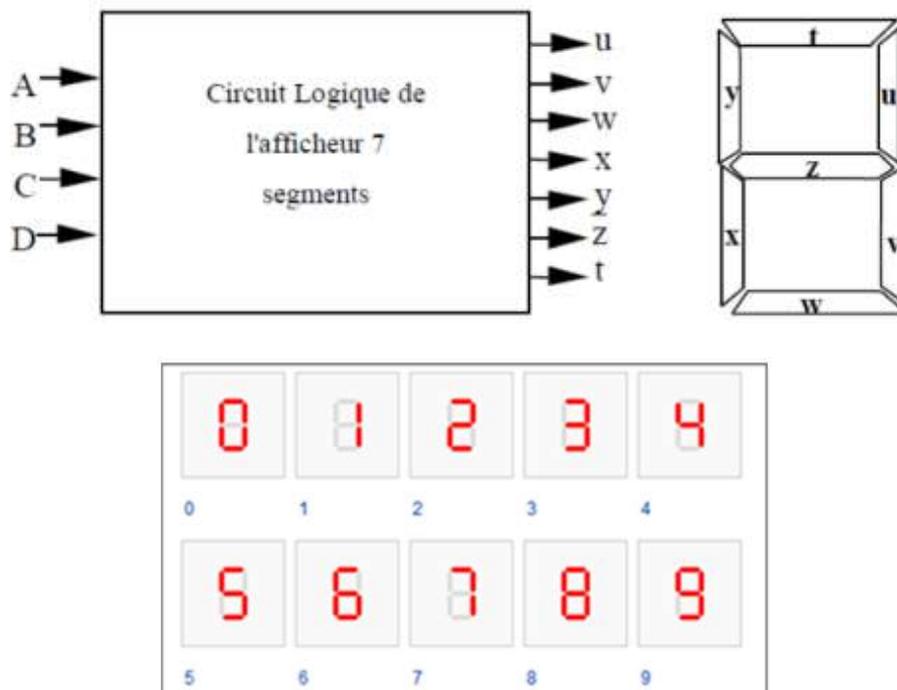
On veut réaliser un multiplieur de deux entrées binaires de deux bits selon les spécifications suivantes :

- deux entrées :  $X = X_1X_0$  et  $Y = Y_1Y_0$
- une seule sortie :  $Z = Z_3Z_2Z_1Z_0$  où  $Z = X*Y$

1. Etablir la table de vérité du multiplieur.
2. Déterminer les équations simplifiées des sorties  $Z_0$ ,  $Z_1$ ,  $Z_2$  et  $Z_3$
3. En déduire le schéma du multiplieur.

**Exercice n°7:**

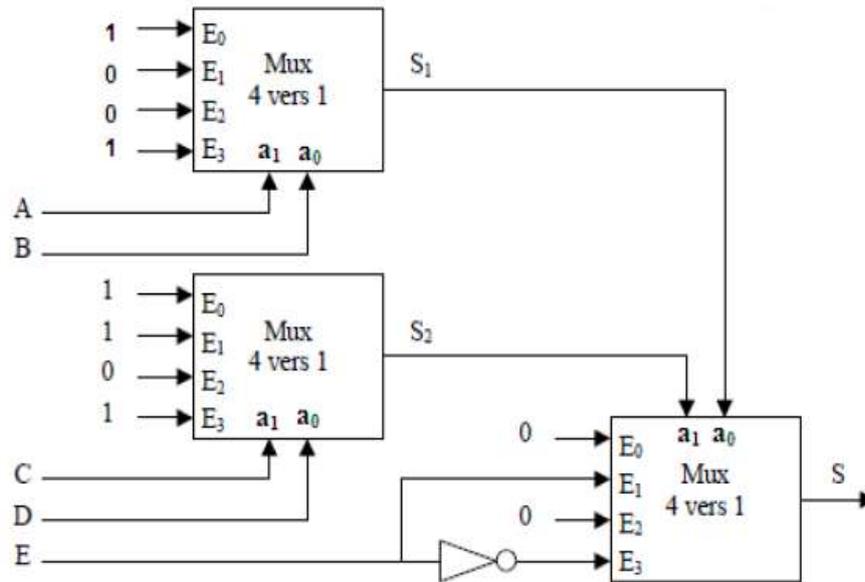
L'afficheur 7 segments permet de représenter les nombres décimaux à partir de leur code BCD. Chaque chiffre décimal est représenté par son équivalent binaire sur quatre bits notés A, B, C, D comme illustré par la figure ci-dessous.



1. Donner la table de vérité de l'afficheur 7 segments.
2. Donner les expressions logiques des trois sorties u, v et t, puis simplifier les soit par la méthode graphique (Karnaugh), soit par la méthode algébrique.
3. Donner le logigramme de ces trois sorties u, v et t de l'afficheur 7 segments.

**Exercice n°8:**

Donner l'équation logique  $S=f(A,B,C,D,E)$  du circuit suivant :



**Exercice n°9:**

Soit  $f$  une fonction logique à trois entrées ( $a$ ,  $b$  et  $c$ ), qui possède le fonctionnement suivant :

$$f(a, b, c) = \begin{cases} 1 & \text{si } (abc)_2 \text{ comporte un nombre pair de "1"} \\ 0 & \text{si } (abc)_2 \text{ comporte un nombre impair de "1"} \end{cases}$$

1. Etablir la table de vérité de cette fonction logique.
2. Réaliser cette fonction via un multiplexeur puis un décodeur.

**Exercice n°10:**

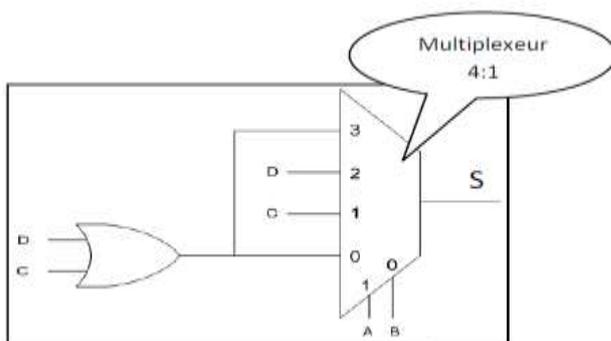
Réaliser un circuit logique qui détermine si un nombre binaire sur 5 bits n'est pas multiple de 3, à l'aide d'un multiplexeur (Etablir table de vérité puis le schéma interne à l'aide d'un multiplexeur).

**Exercice n°11:**

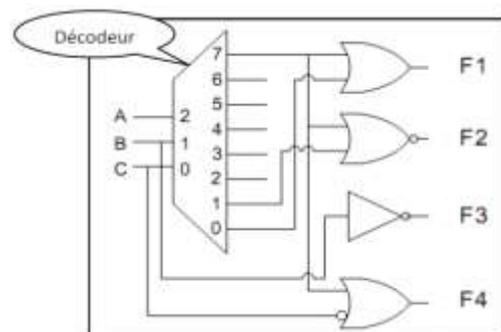
Analyser les circuits combinatoires suivants, en répondant aux questions suivantes :

- 1- Trouver la (ou les) équation (s) reliant ses sorties à ses entrées.
- 2- Simplifier les équations trouvées des deux circuits.

Circuit1 :



Circuit2 :



**Exercice n°12:**

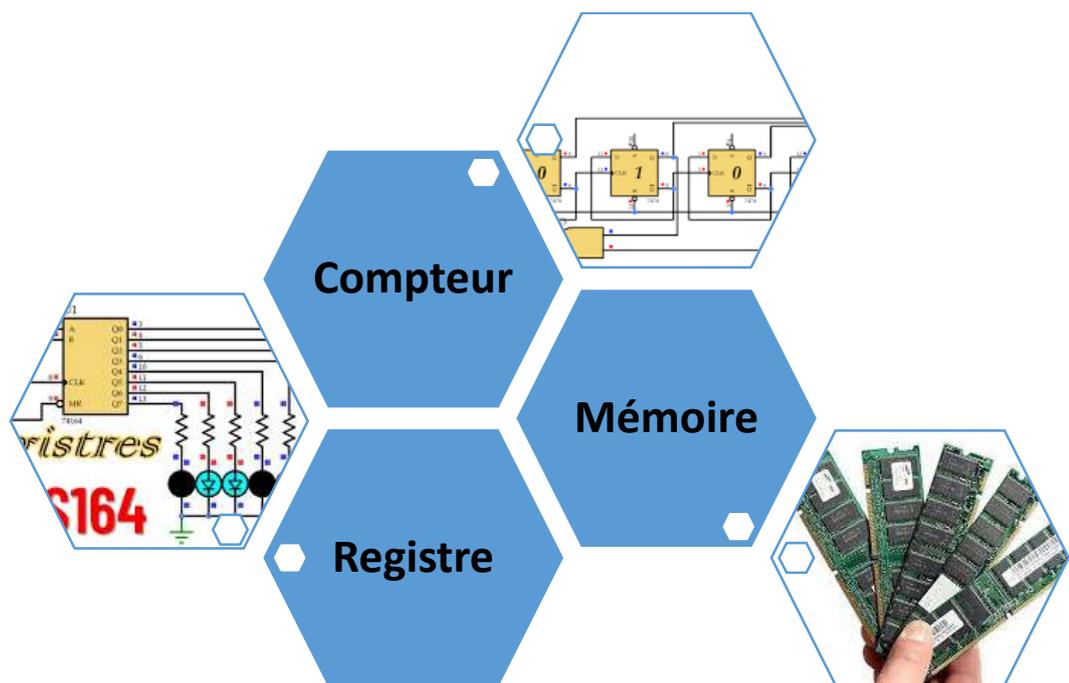
Soit  $f$  une fonction logique à quatre entrées ( $a$ ,  $b$ ,  $c$  et  $d$ ), qui possède le fonctionnement suivant :

$$f(a, b, c) = \begin{cases} 1 & \text{si } (abcd)_2 \text{ comporte un nombre pair de "0"} \\ 0 & \text{si } (abcd)_2 \text{ comporte un nombre impair de "0"} \end{cases}$$

Réaliser cette fonction à l'aide :

- a. / d'un multiplexeur 16 x 1,
- b. / d'un multiplexeur 8 x 1 et d'un minimum de portes,
- c. / de multiplexeurs 4 x 1 et sans aucune porte logique.

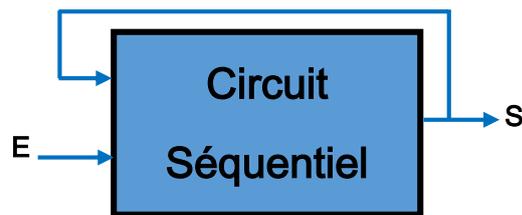
# Chapitre III : Conception des Circuits Séquentiels



## Chapitre III : conception des Circuits Séquentiels

### 3.1. Introduction

En logique combinatoire, les signaux de sortie ne dépendent que des états des variables d'entrée. Mais pour les circuits logiques séquentiels, l'état du système doit être pris en compte. Les circuits logiques séquentiels sont des systèmes dont les sorties dépendent d'une part des valeurs d'entrées et d'autre part de l'état du système, de plus, ils ont une caractéristique de mémoire. L'élément de base de la logique séquentielle est une bascule, contrairement à la logique combinatoire qui est une porte logique.



$$S_{t+1} = f(E, S_t) \text{ ou } S^+ = f(E, S)$$

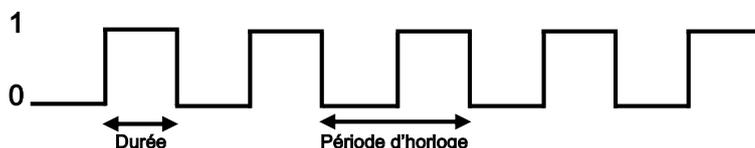
### 3.2. Système séquentiels synchrone et asynchrones

Les systèmes séquentiels se divisent en deux catégories principales :

**a-Système séquentiel asynchrone** : il s'agit d'un système séquentiel qui peut évoluer seul sans commande extérieure. Les sorties sont définies seulement par les entrées et par le fonctionnement du circuit.

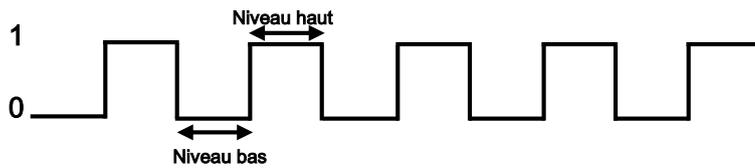
**b-Système séquentiel synchrone** : il s'agit d'un système séquentiel qui ne peut évoluer que lorsqu'il y a une commande extérieure (en absence de commande le système reste figé dans l'état où il se trouve). Cette entrée de commande est appelée horloge dont sa fréquence est l'inverse de sa période (ou temps de cycle) [7].

La figure ci-dessous montre un signal d'horloge, qui est un signal de synchronisation périodique logique qui passe périodiquement de l'état 0 à l'état 1 et de l'état 1 à l'état 0 en fonction du temps.



## Forme des signaux de commandes :

### a. Signal à niveau :



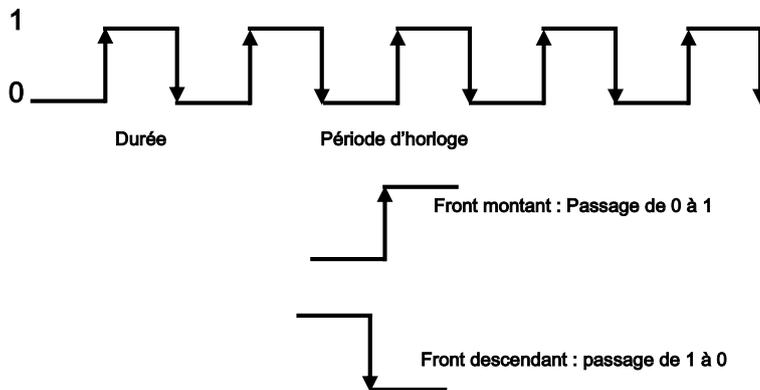
#### ⇒ Synchronisation sur niveau haut

- Si H=0 : quelles que soient les valeurs des entrées, la sortie reste dans son état.
- Si H=1 : la bascule fonctionne en mode normal, la sortie obéit à l'entrée. C'est à dire la bascule ne fonctionne que si H=1(Niveau haut).

#### ⇒ Synchronisation sur niveau bas

- Si H=1 : quelles que soient les valeurs des entrées, la sortie reste dans son état.
- Si H=0 : la bascule fonctionne en mode normal.

### b. Signal impulsionnel



#### ⇒ Synchronisation sur front

Les variables logiques ont deux niveaux : le niveau logique bas « 0 » et le niveau logique haut « 1 ».

- Le passage du niveau bas vers le niveau haut appelé front montant.
- Le passage du niveau haut vers le niveau bas appelé front descendant.

Symbole	Fonctionnement	Forme du signal
	Horloge fonctionnant sur niveau haut (H)	
	Horloge fonctionnant sur niveau bas (B)	
	Horloge fonctionnant sur passage du niveau bas au niveau haut	
	Horloge fonctionnant sur passage du niveau haut au niveau bas	

Table 2.1. Types d'horloges et leurs symboles.

Les entrées ne sont validées que au moment où les impulsions d'horloge sont produites, certain sont sensible à des fronts montants ou descendants.

### 3.3. Les bascule

#### 3.3.1. Définition d'une bascule

Une bascule (flip-flop) représente la base de la logique séquentielle dont le rôle est de mémoriser et stocker une information élémentaire. En d'autres termes, une bascule est une mémoire à 1 bit. La bascule est un circuit séquentiel constituée par une ou plusieurs entrées et deux sorties complémentaires  $Q$  et  $\bar{Q}$ .

#### 3.3.2. Les types des bascules

Les bistables les plus couramment utilisées sont : RS, JK, D et T.

##### 3.3.2.1. Bascule RS :( Reset\_set)

La bascule RS est constituée de deux entrées S (mise à 1 ou Set) et R (mise à 0 ou Rset) et de deux sorties complémentaires  $Q$  et  $\bar{Q}$  [8] dont le nouvel état de la bascule  $Q^+$  dépend de l'état des entrées R et S et de son état antérieur  $Q$ .

$$Q^+ = f(Q, R, S)$$



Fonctionnement statique :

$$\left\{ \begin{array}{ll} R = S = 0 & \text{conservation d'état} \\ R = 1, S = 0 & \text{mise à 0} \\ R = 0, S = 1 & \text{mise à 1} \\ R = S = 1 & \text{indéterminé} \end{array} \right.$$

Table de vérité :

R	S	$Q$	$Q^+$
0	0	0	0 mémorisation de l'information
0	0	1	1 mémorisation de l'information
0	1	0	1 mise à 1 de la sortie
0	1	1	1 mise à 1 de la sortie
1	0	0	0 mise à 0 de la sortie
1	0	1	0 mise à 0 de la sortie
1	1	0	X état indéterminé
1	1	1	X état indéterminé

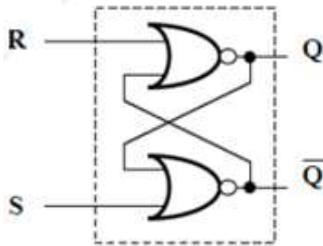
**La table de vérité condensée**

R	S	$Q^+$
0	0	$Q$ mémorisation (conservation de l'état interne)
0	1	1 mise à 1 ( $\forall Q$ )
1	0	0 mise à 0 ( $\forall Q$ )
1	1	X état indéterminé

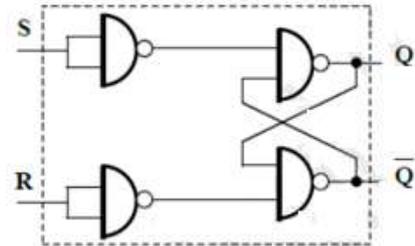
**Les équations d'état**

$$\begin{aligned}
 Q^+ &= \overline{R}\overline{S}Q + \overline{R}SQ + \overline{R}S\overline{Q} \\
 &= \overline{R}\overline{S}Q + \overline{R}S(Q + \overline{Q}) \\
 &= \overline{R}\overline{S}Q + \overline{R}S \\
 &= \overline{R}(\overline{S}Q + S) \\
 &= \overline{R}((\overline{S} + S)(Q + S)) \\
 &= \overline{R}(Q + S) \\
 Q^+ = \overline{\overline{Q}^+} &= \overline{\overline{\overline{R}(Q + S)}} = \overline{\overline{R + (Q + S)}}
 \end{aligned}$$

**Représentation à l'aide de portes NOR**



**Représentation à l'aide de portes NAND**

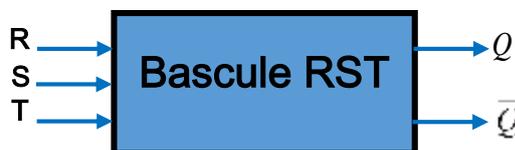


**Table de transition d'une bascule RS**

$Q$	$Q^+$	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

**3.3.2.2. Bascules RST**

La bascule RST est constituée de trois entrées S, R et T et de deux sorties complémentaires  $Q$  et  $\overline{Q}$  dont les entrées R et S sont prises en compte que si elles sont en coïncidence avec un signal de commande T (horloge).



**Fonctionnement :**

- Si  $T=0$  la bascule conserve son état.
- Si  $T=1$ 

$$\left\{ \begin{array}{ll} R = S = 0 & \text{conservation d'état} \\ R = 1, S = 0 & \text{mise à 0} \\ R = 0, S = 1 & \text{mise à 1} \\ R = S = 1 & \text{indéterminé} \end{array} \right.$$

**Table de vérité :**

T	R	S	$Q$	$Q^+$
0	0	0	0	0 conservation d'état
0	0	0	1	1 conservation d'état
0	0	1	0	0 conservation d'état
0	0	1	1	1 conservation d'état
0	1	0	0	0 conservation d'état
0	1	0	1	1 conservation d'état
0	1	1	0	0 conservation d'état
0	1	1	1	1 conservation d'état
1	0	0	0	0 conservation d'état
1	0	0	1	1 conservation d'état
1	0	1	0	1 mise à 1 de la sortie
1	0	1	1	1 mise à 1 de la sortie
1	1	0	0	0 mise à 0 de la sortie
1	1	0	1	0 mise à 0 de la sortie
1	1	1	0	X état indéterminé
1	1	1	1	X état indéterminé

**La table de vérité condensée**

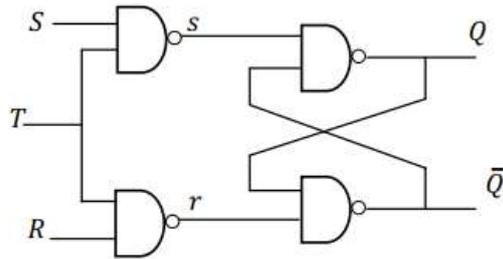
T	R	S	$Q^+$
0	X	X	$Q$ conservation de l'état (mémorisation de l'information)
1	0	0	$Q$ conservation de l'état (mémorisation de l'information)
1	0	1	1 mise à 1 ( $\forall Q$ )
1	1	0	0 mise à 0 ( $\forall Q$ )
1	1	1	X état indéterminé

**Les équations d'état**

$$Q^+ = S\bar{R}T + \bar{T}Q + \bar{S}R\bar{T}Q$$

$$\left\{ \begin{array}{l} s = \bar{T}\bar{S} = \bar{T} + \bar{S} \\ r = \bar{T}R = \bar{T} + \bar{R} \end{array} \right.$$

### Représentation par les portes NAND



### 3.3.2.3. Bascule JK

La bascule JK est identique à la bascule RS (l'entrée J correspond à S et l'entrée K correspond à R) sauf pour la condition  $J=K=1$  cette bascule fonctionnent en mode Toggle à savoir que leur sortie  $Q$  passe à l'état opposé à chaque impulsion d'horloge. Par rapport à la bascule RS, la bascule JK permet d'utiliser toutes les combinaisons des entrées.



Fonctionnement statique :

- $J = K = 0$  conservation d'état
- $J = 0, K = 1$  mise à 0
- $J = 1, K = 0$  mise à 1
- $J = K = 1$  inverse d'état

Table de vérité :

J	K	$Q$	$Q^+$
0	0	0	0 conservation d'état (mémorisation de l'information)
0	0	1	1 conservation d'état (mémorisation de l'information)
0	1	0	0 mise à 0 de la sortie
0	1	1	0 mise à 0 de la sortie
1	0	0	1 mise à 1 de la sortie
1	0	1	1 mise à 1 de la sortie
1	1	0	1 inverse d'état (basculement)
1	1	1	0 inverse d'état (basculement)

La table de vérité condensée

J	K	$Q^+$
0	0	$Q$ conservation de l'état interne
0	1	0 mise à 0 ( $\forall Q$ )
1	0	1 mise à 1 ( $\forall Q$ )
1	1	$\bar{Q}$ inverse d'état (basculement)

### Les équations d'état

$$\begin{aligned}
 Q^+ &= \bar{J} \bar{K} Q + J \bar{K} \bar{Q} + J \bar{K} Q + J K \bar{Q} \\
 &= \bar{K} Q (\bar{J} + J) + J \bar{Q} (K + \bar{K}) \\
 &= \bar{K} Q + J \bar{Q}
 \end{aligned}$$

### Table de transition d'une bascule JK

La table des transitions d'une bascule JK est remplie via sa table de vérité.

- ligne 1  $\Rightarrow$  Q passe de 0 à 0  $\Rightarrow$   $\begin{cases} \text{état mémoire (J = 0, K = 0)} \\ \text{mise à 0 (J = 0, K = 1)} \end{cases} \Rightarrow (J = 0, K = X)$
- ligne 2  $\Rightarrow$  Q passe de 0 à 1  $\Rightarrow$   $\begin{cases} \text{état inverseur (J = 1, K = 1)} \\ \text{mise à 1 (J = 1, K = 0)} \end{cases} \Rightarrow (J = 1, K = X)$
- ligne 3  $\Rightarrow$  Q passe de 1 à 0  $\Rightarrow$   $\begin{cases} \text{état inverseur (J = 1, K = 1)} \\ \text{mise à 0 (J = 0, K = 1)} \end{cases} \Rightarrow (J = X, K = 1)$
- ligne 4  $\Rightarrow$  Q passe de 1 à 1  $\Rightarrow$   $\begin{cases} \text{état mémoire (J = 0, K = 0)} \\ \text{mise à 1 (J = 1, K = 0)} \end{cases} \Rightarrow (J = X, K = 0)$

Q	Q <sup>+</sup>	J	K	
0	0	0	X	mise à 0 ou état mémoire
0	1	1	X	mise à 1 ou basculement
1	0	X	1	mise à 0 ou basculement
1	1	X	0	mise à 1 ou état mémoire

### 3.3.2.4. Bascule T (Trigger flip-flop)

La bascule T est constituée d'une seule entrée T et de deux sorties complémentaires Q et  $\bar{Q}$ . C'est une bascule à déclenchement.



#### Fonctionnement :

- si T=1, inversion d'état ;
- si T=0, conservation de l'état (état mémoire), c'est à dire Q ne change pas.

#### Table de vérité :

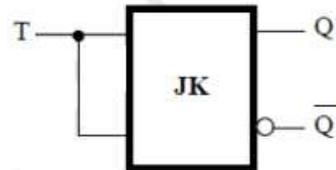
T	Q	Q <sup>+</sup>	
0	0	0	conservation de l'état (état mémoire)
0	1	1	conservation de l'état (état mémoire)
1	0	1	inverse d'état (basculement)
1	1	0	inverse d'état (basculement)

### Les équations d'état

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

$$= T \oplus Q$$

### Réalisation



Dans l'équation de sortie de la bascule JK, si on remplace J et K par T, on obtient :

$$Q^+ = TQ + \bar{T}\bar{Q}$$

$$= T \oplus \bar{Q}$$

### 3.3.2.5. Bascule D (Delay)

La bascule D est constituée d'une seule entrée D et de deux sorties complémentaires Q et  $\bar{Q}$  dont le signal d'entrée D est recopié sur sa sortie Q.

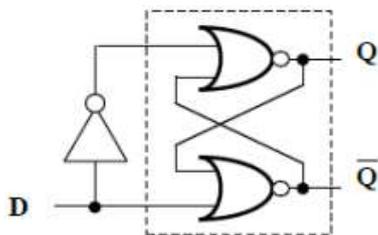
#### Fonctionnement :

- si D=1, mise à 1 de la sortie Q.
- si D=0, mise à 0 de la sortie Q.

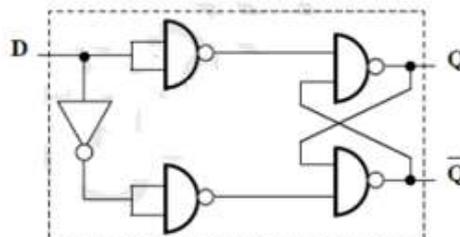
#### Table de vérité :

D	Q	Q <sup>+</sup>
0	0	0 mise à 0
0	1	0 mise à 0
1	0	1 mise à 1
1	1	1 mise à 1

#### Réalisation à l'aide de portes NOR



#### Réalisation à l'aide de portes NAND

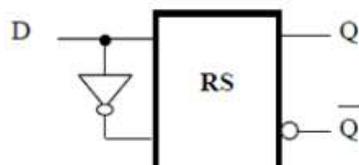


**Remarque :** dans l'équation de la bascule RS, si on pose S=D et R=  $\bar{D}$ , on obtient :

$$Q^+ = D\bar{D}Q + \bar{D}D$$

$$= D$$

Ainsi, si on rajout un inverseur entre S et R, on obtient une bascule D.



### Table de transition d'une bascule D

$Q$	$Q^+$	D	
0	0	0	mise à 0
0	1	1	mise à 1
1	0	0	mise à 0
1	1	1	mise à 1

### 3.4. Utilisation des bascules

#### 3.4.1. Utilisation des bascules pour réaliser un registre

##### 3.4.1.1. Définition d'un registre

Un registre est avant tout un ensemble de cellules ou de cases mémoire capables de stocker une information. Un registre est constitué d'un groupe de bascules de même type, pilotées par la même impulsion d'horloge. Le nombre de bascules qui composent un registre détermine sa capacité [9]. Donc, un registre est une collection ordonnée de bascules.

- Une case mémoire est définie via une bascule
- L'élément de base de la logique séquentielle est la bascule.
- Une bascule permet la mémorisation d'un seul bit.
- Un registre de taille n bits est une collection ordonnée de n bascules.
- Un registre de capacité n bits permet la mémorisation d'une information de taille n bits.

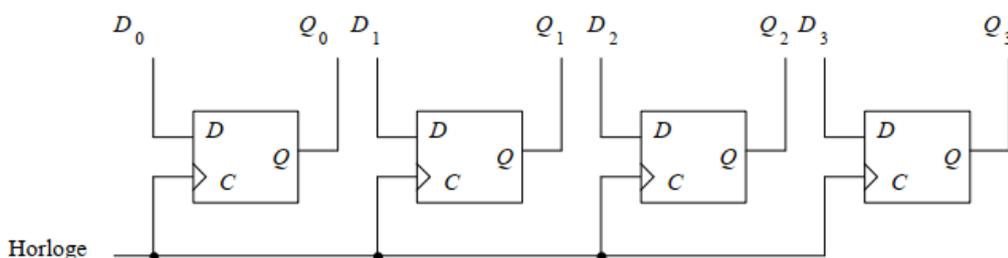


Figure 3. 1. Exemple d'un registre à 4 bits à base de bascules D.

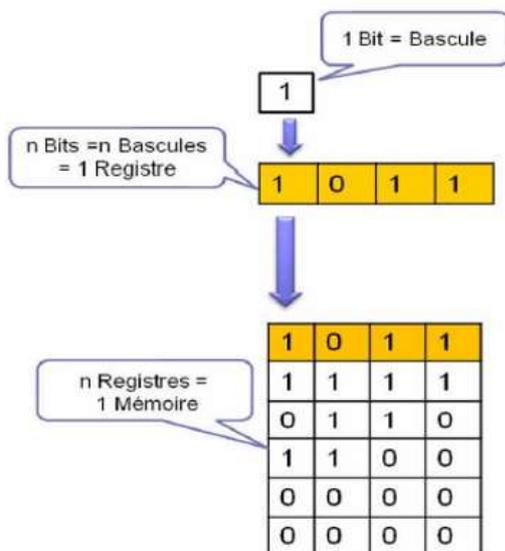


Figure 3. 2. Illustration de la relation bit, registre et mémoire.

### 3.4.1.2. Fonctionnement d'un registre

Le registre est utilisé pour stocker une information (un mot ou un nombre binaire). Les principales fonctions d'un registre sont :

➔ La mémorisation

Mémoriser l'information telle qu'elle reste sans aucun changement

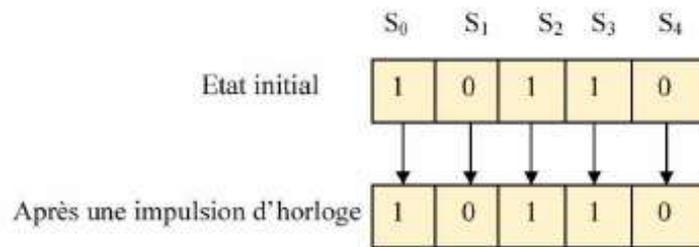


Figure 3. 3. Illustration de la mémorisation.

➔ Le décalage à droite ou à gauche

Décaler l'information soit de la gauche vers la droite ou de la droite vers la gauche

- Décalage à droite : pour chaque impulsion d'horloge la bascule de rang  $i+1$  prend le contenu de la bascule de rang  $i$ .

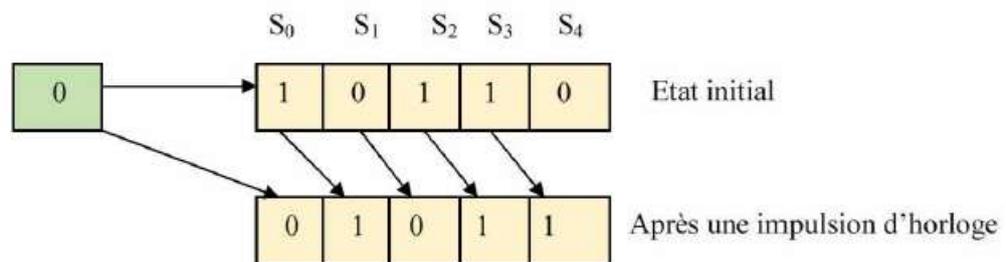


Figure 3. 4. Illustration du décalage à droite.

- Décalage à gauche : pour chaque impulsion d'horloge la bascule de rang  $i-1$  prend le contenu de la bascule de rang  $i$ .

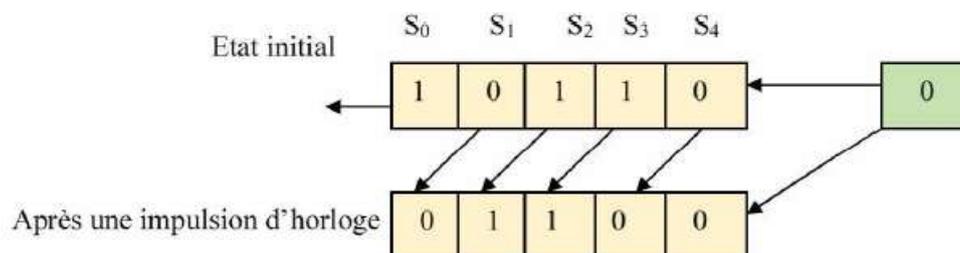
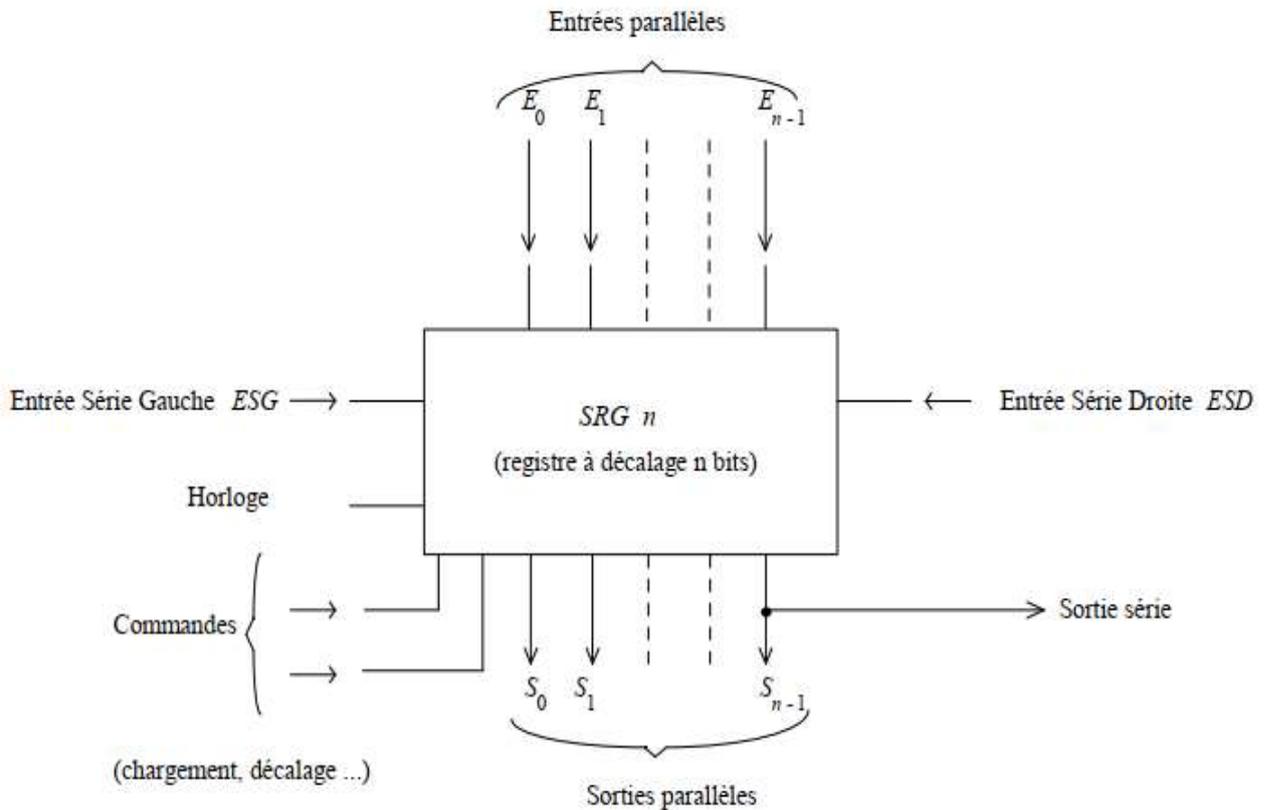


Figure 3. 5. Illustration du décalage à gauche.

### 3.4.1.3. Type des registres

En général, un registre peut être représenté par le schéma ci-dessous [10]:



Il existe plusieurs types de registres :

- Registre de mémorisation
- Registre à chargement parallèle (Registre à entrées parallèles et sorties parallèles).
- Registre à chargement série (Registre à entrée série et sortie série).
- Registre à entrée parallèle et sortie série.
- Registre à entrée série et sortie parallèle.
- Registre à décalage à droite.
- Registre à décalage à gauche.
- Registre à décalage circulaire.

### 3.4.1.3.1. Registre de mémorisation

Registre de mémorisation peut se réaliser sous la forme illustrée dans l'exemple au-dessous qui comprend la désactivation du fonctionnement de l'horloge en insérant une porte logique ET en série.

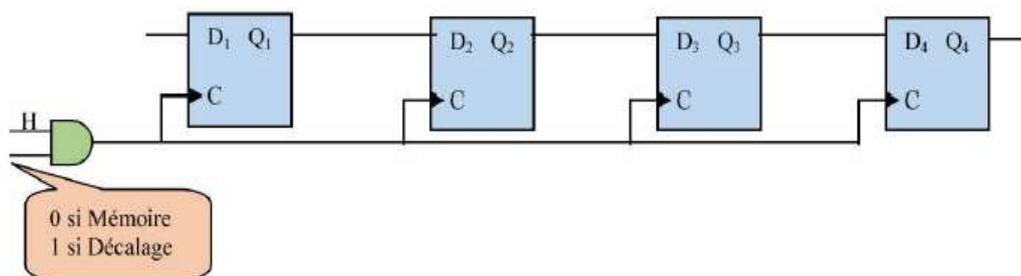


Figure 3. 6. Exemple de registre de mémorisation à base de bascules D.

### 3.4.1.3.2. Registre à chargement parallèle

Ce registre permet le chargement des états de  $n$  bascules en même temps dont chaque bascule de rang  $i$  prenant la valeur de l'information  $i$ . Donc, Il peut charger une information de  $N$  bits en même temps. Il Contient une entrée de chargement  $chg$  ( $chg=0$  état mémoire,  $chg=1$  chargement ).

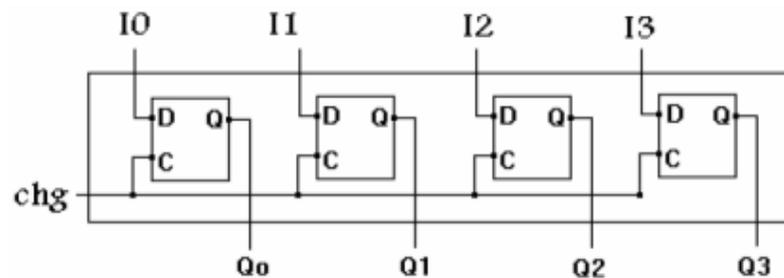


Figure 3. 7. Exemple de registre à chargement parallèle à base de bascules D.

### 3.4.1.3.3. Registre à entrée série et sortie série

Au niveau de ce registre, l'information est introduite en série (bit par bit) de sorte que la première bascule reçoit la nouvelle entrée  $E_s$  et l'ensemble du registre est décalé d'une bascule de rang  $i$  vers la bascule de rang  $i+1$ .

• Un tel registre est appelé soit:

➔ Registre à entrée série à gauche et à sortie série à droite.

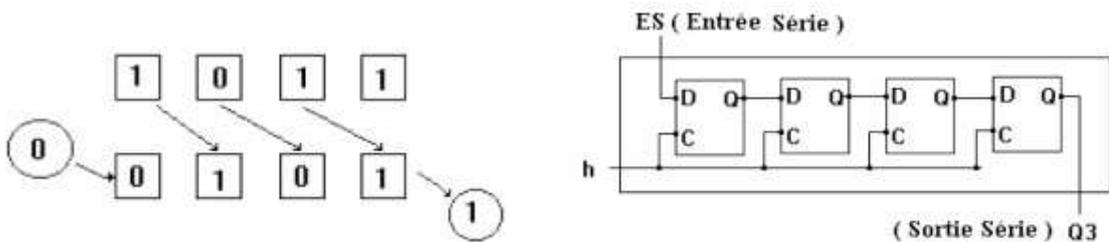


Figure 3. 8. Exemple de registre à entrée série à gauche et à sortie série à droite à base de bascules D.

➔ Registre à entrée série à droite et à sortie série à gauche

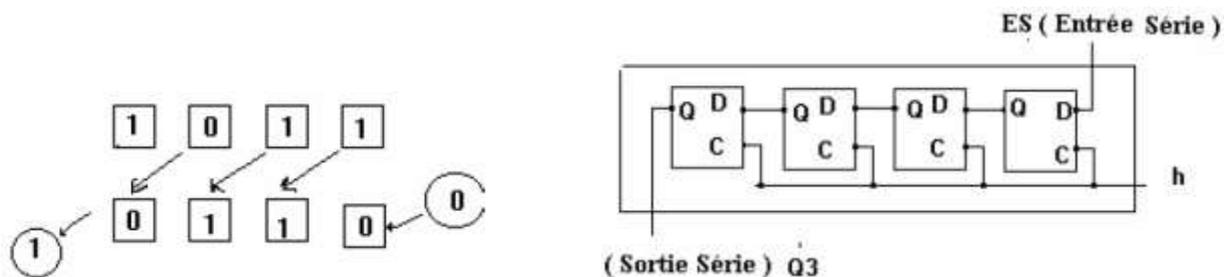


Figure 3. 9. Exemple de registre à entrée série à droite et à sortie série à gauche à base de bascules D.

#### 3.4.1.3.4. Registre à entrée série et sortie parallèle

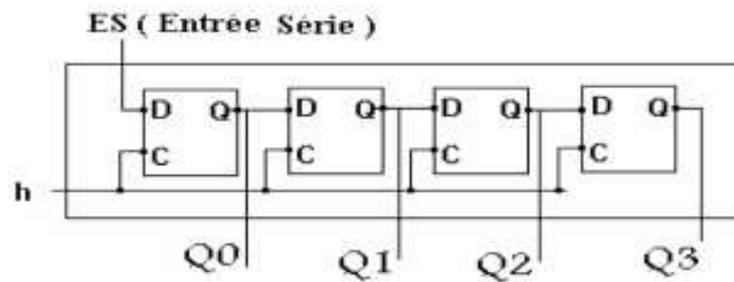


Figure 3. 10. Exemple de registre à entrée série et à sortie parallèle à base de bascules D.

#### 3.4.1.3.5. Registre à entrée parallèle et sortie série

Ce registre convertit un codage spatial en codage temporel. Dans l'exemple ci-dessous, si  $X=1$  le chargement par l'entrée parallèle est bloqué et par l'entrée série est autorisé. Si  $X = 0$  le chargement par l'entrée série est bloqué par contre par l'entrée parallèle est autorisé.

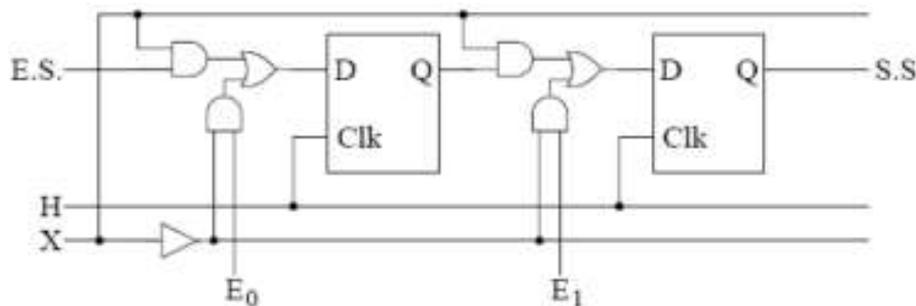


Figure 3. 11. Exemple de registre à entrée parallèle et à sortie série à base de bascules D.

#### 3.4.1.3.6. Registre à décalage à droite

Ce registre permet d'effectuer un décalage vers la droite tel que l'entrée de la première bascule reçoit une nouvelle entrée  $E_s$ .

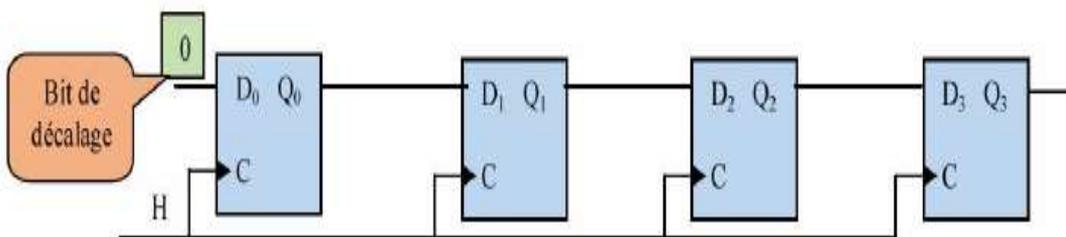


Figure 3. 12. Exemple de registre à décalage à droite à base de bascules D.

#### 3.4.1.3.7. Registre à décalage à gauche

Ce registre permet d'effectuer un décalage vers la gauche tel que l'entrée de la première bascule reçoit une nouvelle entrée  $E_s$ .

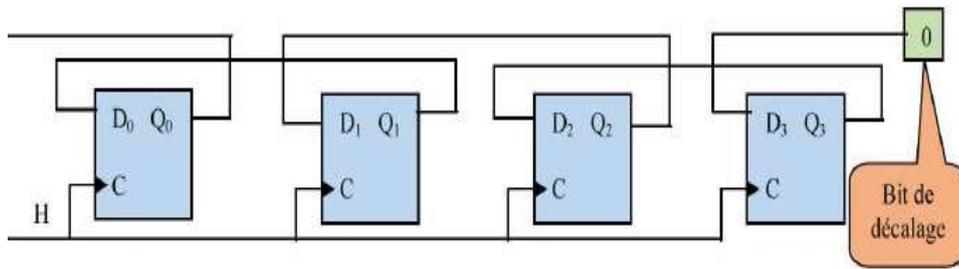


Figure 3. 13. Exemple de registre à décalage à gauche à base de bascules D.

### 3.4.1.3.6. Registre à décalage circulaire

Ce registre permet d'effectuer un décalage vers la gauche en faisant passer la sortie de la dernière bascule à l'entrée de la première bascule. Le décalage peut être vers la droite (circulaire droite) ou vers la gauche (circulaire gauche).

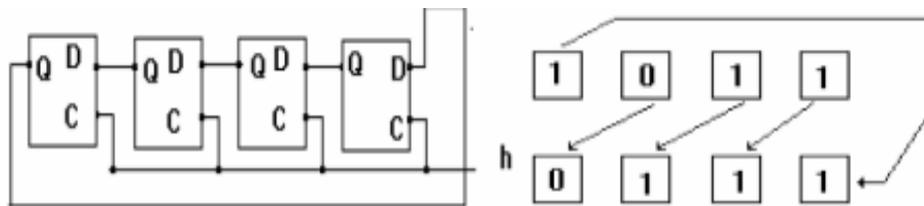


Figure 3. 14. Exemple de registre à décalage circulaire à base de bascules D.

## 3.4.2. Utilisation des bascules pour la mémoire centrale

### 3.4.2.1. Définition d'une mémoire

Elle s'agit d'un dispositif capable d'enregistrer, de conserver et de restituer les informations. Les éléments de mémoire d'un ordinateur sont divisés en plusieurs niveaux caractérisés par leur capacité et leur temps d'accès.

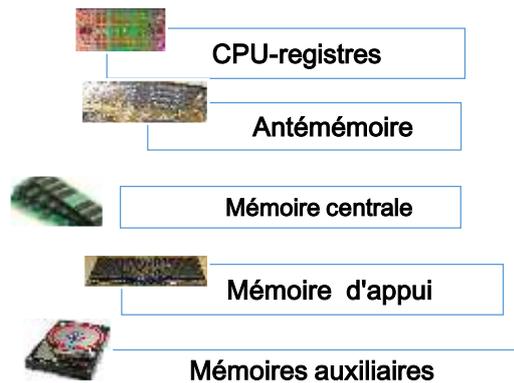


Figure 3. 15. Principaux niveaux de mémoire d'un ordinateur.

### 3.4.2.2. Différents types de la mémoire

Lorsqu'on s'éloigne du CPU vers les mémoires auxiliaires, le temps d'accès et la capacité de la mémoire augmentent, mais le coût par bit diminue.

1. **les registre** : est l'élément de mémoire situé dans l'unité centrale de traitement (CPU), qui permet de stocker les opérandes et les résultats intermédiaires.

2. **L'antémémoire** ou **mémoire cache** : L'antémémoire également appelée mémoire cache, ou plus simplement cache, est une mémoire rapide de faible capacité qui agit comme un tampon entre le processeur et la mémoire [11]. Cette mémoire permet au CPU d'avoir moins accès à la mémoire centrale, ce qui permet de gagner du temps.

3. **La mémoire centrale** : est le principal dispositif de stockage des informations utilisé par le CPU pour enregistrer, stocker et restituer les informations [12]. On distingue deux types :

### 3.1. Mémoires vives

- Les RAM ou mémoires vives (signifie en anglais : Random Access Memory) sont des mémoires à lecture et écriture qui utilisées pour enregistrer, stocker et restituer les informations.



Figure 3. 16. RAM.

- La mémoire centrale est composée d'un ensemble ordonné de  $2^m$  cellules (point mémoire), chaque cellule contenant un mot de  $n$  bits qui sont traités (écrits ou lus) simultanément. La cellule mémoire est la plus petite subdivision (entité atomique) de la mémoire dans laquelle il est possible de lire ou d'écrire une information élémentaire [13].

- Une mémoire peut être représentée comme une armoire de rangement composée de différents tiroirs. Ensuite chaque tiroir représente une case mémoire qui peut contenir un seul élément et être identifiable par un numéro. Ce numéro est appelé adresse afin que chaque donnée devienne alors accessible grâce à son adresse [14].

Donc la taille du bloc mémoire est de  $2^m$ , le premier mot est à l'adresse 0 et le dernier mot est à l'adresse  $2^m-1$ . Afin de pouvoir reconnaître chaque mot individuellement,  $m$  lignes d'adresse sont utilisées.

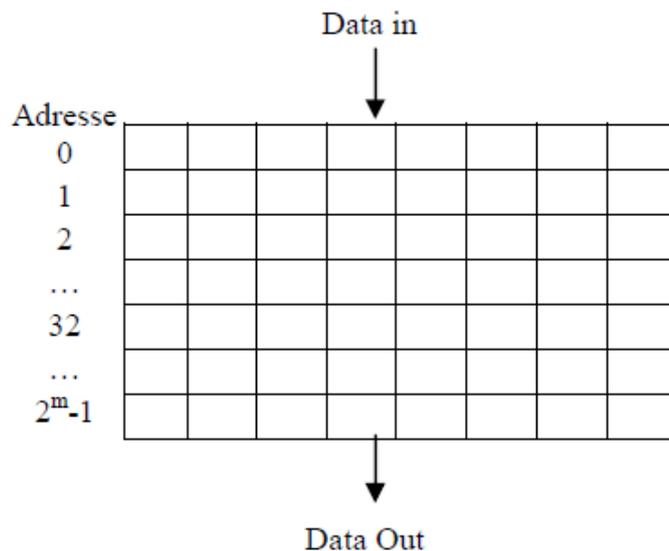


Figure 3. 17. Exemple d'une cellule d'un octet.

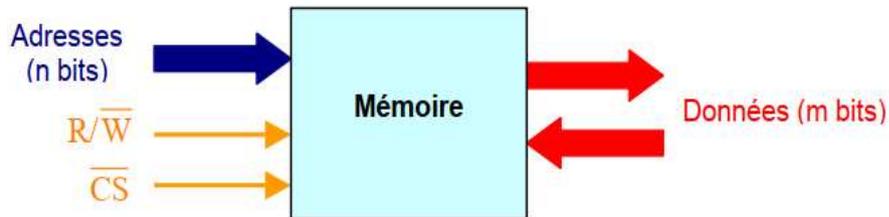


Figure 3. 18. Présentation d'un circuit mémoire.

La figure (3.18) montre un circuit mémoire composé des éléments suivants:

- Les entrées d'adresses sur n bits.
- Les entrées de données sur m bits.
- Les sorties de données
- deux entrées de commandes qui sont :
  - Une entrée de sélection de lecture ou d'écriture ( $R/\overline{W}$ ).
  - Une entrée de sélection du circuit  $\overline{CS}$ .

Les opérations de lecture ou d'écriture en mémoire suivent toujours le cycle suivant:

1. Sélection de l'adresse de la donnée.
2. Choix de l'opération à effectuer sur la donnée ( $R/\overline{W}$ ).
3. Sélection de la mémoire ( $\overline{CS} = 0$ )
4. Écriture ou lecture de la donnée.

- La RAM se divise en deux grandes familles, la RAM statique (SRAM) ou la RAM dynamique (DRAM), la mémoire DRAM est largement utilisée car peu coûteuse [15].
- La bascule est l'élément de mémorisation dans les RAM statique. Tandis que, le condensateur (capacité) commandée par un transistor est l'élément de mémorisation dans les RAM dynamiques (DRAM).

### 3.2. Mémoires mortes

Appelée ROM (Read Only Memory) est une mémoire en lecture seule), la ROM est utilisée pour stocker les données nécessaires au démarrage de l'ordinateur, ne s'efface pas quand le système est éteint, Contient les informations nécessaires au démarrage de l'ordinateur (BIOS, instructions de démarrage, microcode).



Figure 3. 19. ROM.

4. **La mémoire d'appui** : est la mémoire intermédiaire entre la mémoire centrale et les mémoires auxiliaires.

5. **Les mémoires auxiliaires** : en d'autres termes, la mémoire périphérique ou de masse ou mémoire secondaire est une mémoire qui a une capacité élevée avec un coût relativement faible ; permet de stocker les informations plus longtemps que dans la mémoire principale avec une capacité plus limitée. Il s'agit, par exemple, des disques et disquettes, ou des bandes magnétiques.

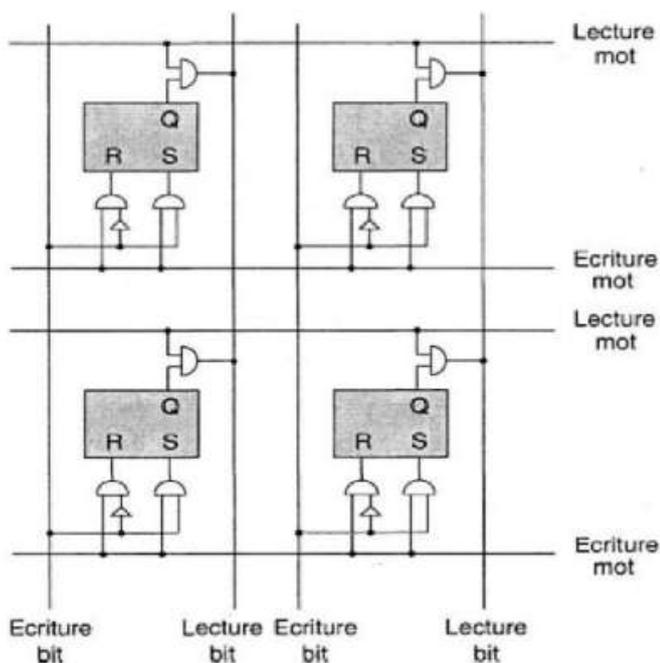


Figure 3. 20. Présentation d'une mémoire.

#### 3.4.2.3. Caractéristiques d'une mémoire

➔ **La capacité** : représente le nombre total de bits que contient la mémoire. Elle est également exprimée en octet. Si  $m$  est le nombre de bits d'adresse et  $n$  est le nombre

de bits par mot, la capacité de la mémoire est donnée par : Capacité =  $2^m$  mots =  $2^m \times n$  bits

- ➔ **Le format des données** : le format correspond à la longueur des mots. C'est à dire le nombre de bits que l'on peut mémoriser par case mémoire.
- ➔ **Le temps d'accès** : c'est le temps écoulé entre le moment où l'opération de lecture/écriture est initiée en mémoire et le moment où la première information devient disponible sur le bus de données.
- ➔ **Le temps de cycle** : est l'intervalle de temps minimum qui doit séparer deux requêtes successives de lecture ou d'écriture.
- ➔ **Le débit** : représente le nombre maximum d'informations lues ou écrites par seconde.
- ➔ **Volatilité** : elle caractérise la permanence des informations en mémoire. Volatile permet à un ordinateur de stocker temporairement des données, de sorte que toutes les données en mémoire sont irrémédiablement effacées chaque fois que l'ordinateur est éteint. Et non volatile, ce qui permet de conserver les informations qu'il contient même lorsque la mémoire n'est pas alimentée électriquement.

### 3.4.3. Utilisation des bascules pour réaliser des compteurs

#### 3.4.3.1. C'est quoi un compteur ?

Un compteur est un circuit séquentiel composé de  $n$  bascules et possède  $N$  états ( $E_0, E_1, \dots, E_{n-1}$ ). Il passe de l'état  $E_i$  à l'état  $E_{i+1}$  à chaque impulsion d'horloge. Il revient toujours à l'état initial  $E_0$ . Un compteur a un cycle (une séquence d'états). Le nombre d'états du compteur est toujours inférieur ou égale à  $2^n$ . Chaque compteur est caractérisé par [16]:

- Commande d'horloge (synchrone ou asynchrone)
- Capacité de comptage
- Code de comptage
- Vitesse de comptage
- Comptage/décomptage
- Possibilités de présélection

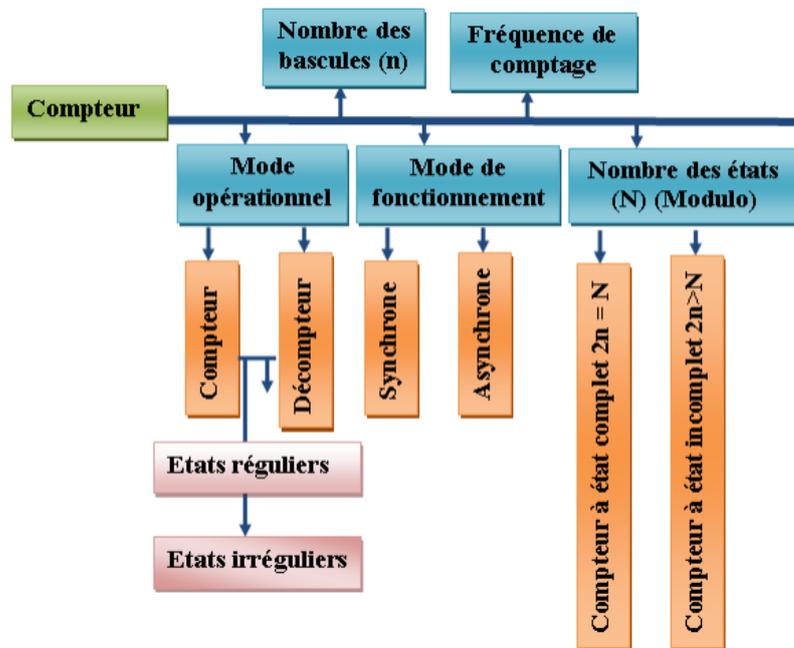


Figure 3. 21. Caractéristiques principales des compteurs [17].

### 3.4.3.2. Types de compteur

Il existe deux types de compteurs selon le mode de fonctionnement :

- Les compteurs parallèles (ou synchrones).
- Les compteurs séries (ou asynchrones).

#### 3.4.3.2.1. Compteurs synchrones

Dans un compteur synchrone, toutes les bascules reçoivent le même signal d'horloge en parallèle. Cela signifie que le signal d'horloge synchronise toutes les bascules simultanément.

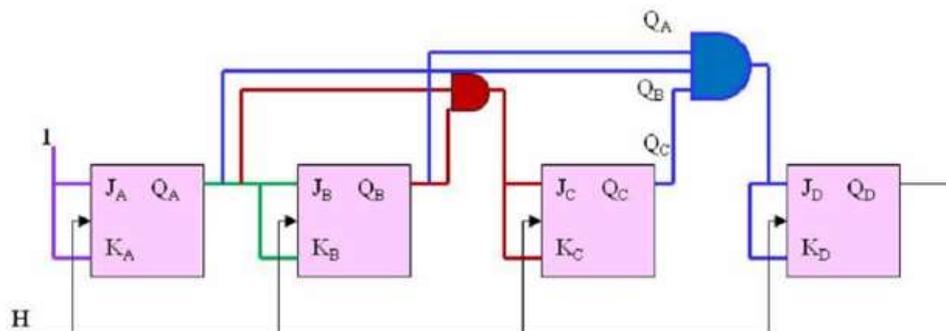


Figure 3. 22. Exemple d'un compteur synchrone à base de bascules JK.

Il existe trois types de compteurs selon le cycle réalisé :

#### a. Les compteurs synchrones modulo $2^n$ (cycle complet)

- modulo 4, signifie que  $n=2$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ .
- modulo 8, signifie que  $n=3$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ .
- modulo 16, signifie que  $n=4$ , de cycle suivant :  
 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$ .

**b. Les compteurs synchrones modulo N (cycle incomplet)**

- modulo 5, signifie que  $N=5$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ .
- modulo 10, signifie que  $N=10$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$ .

**c. Les compteurs synchrones à cycle quelconque**

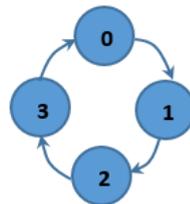
- Exemple :  $0 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 10$

**3.4.3.2.1.1. Étude des compteurs synchrones modulo  $2^n$**

**Exemple1** : réalisation d'un compteur synchrone modulo  $2^2 = 4$  à l'aide des bascules JK

Ce compteur est de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$

**Diagramme d'état**



**Table de vérité**

On utilise la table de transition de la bascule JK pour définir les entrées synchrones J et K à chaque impulsion d'horloge.

$Q_1$	$Q_0$	$Q_1^+$	$Q_0^+$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

**Les équations des entrées des bascules**

$Q_1 \backslash Q_0$	0	1
0	0	X
1	1	X

$J_1 = Q_0$

$Q_1 \backslash Q_0$	0	1
0	X	0
1	X	1

$K_1 = Q_0$

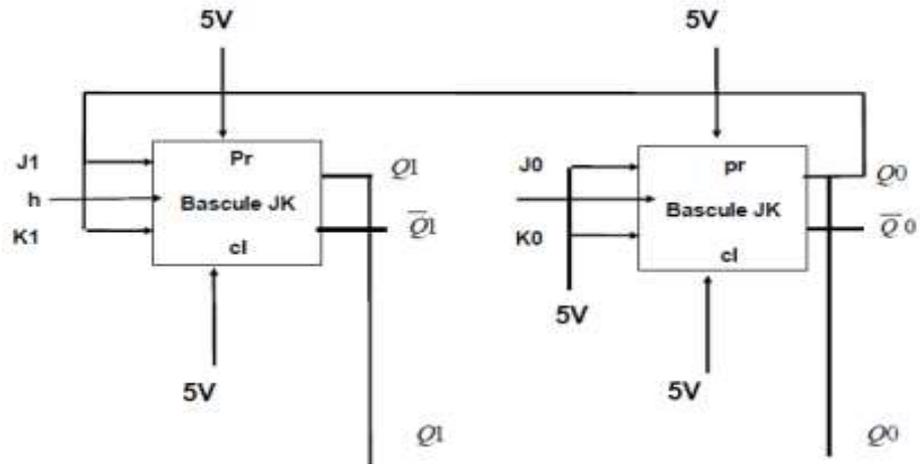
$Q_1 \backslash Q_0$	0	1
0	1	1
1	X	X

$J_0 = 1$

$Q_1 \backslash Q_0$	0	1
0	1	1
1	X	X

$K_0 = 1$

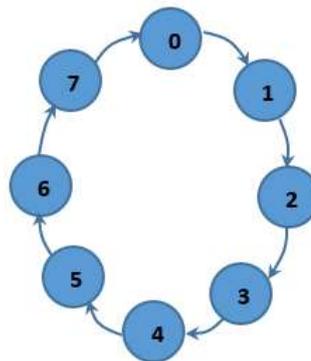
### Schéma d'un compteur synchrone modulo 4



**Exemple2** : réalisation d'un compteur synchrone modulo  $2^3 = 8$  à l'aide des bascules JK

Le cycle de ce compteur est  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ , et toutes les bascules ont la même horloge.

### Diagramme d'état



### Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

### Les équations des entrées des bascules

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	0	0	X	X
$Q_0 \backslash 1$	0	1	X	X

$$J_2 = Q_1 Q_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	X	X	0	0
$Q_0 \backslash 1$	X	X	0	1

$$K_2 = \bar{Q}_1 Q_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	0	X	X	0
$Q_0 \backslash 1$	1	X	X	1

$$J_1 = Q_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	X	0	0	X
$Q_0 \backslash 1$	X	1	1	X

$$K_1 = Q_0$$

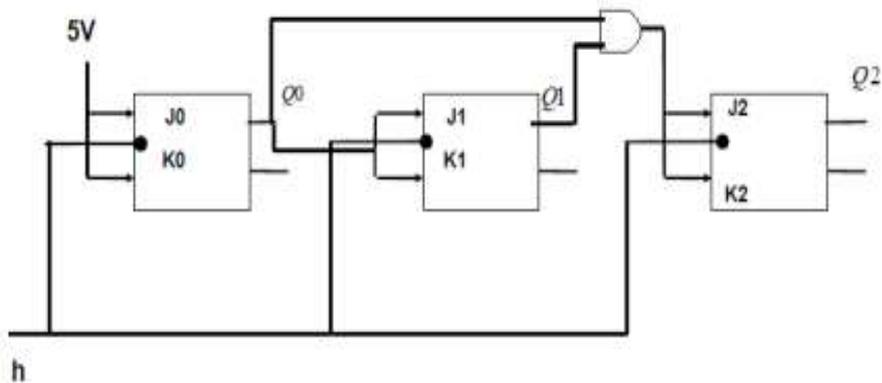
$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	1	1	1	1
$Q_0 \backslash 1$	X	X	X	X

$$J_0 = 1$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0 \backslash 0$	X	X	X	X
$Q_0 \backslash 1$	1	1	1	1

$$K_0 = 1$$

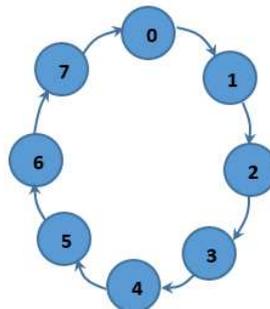
### Schéma d'un compteur modulo 8 synchrone



### Exemple 3 : réalisation d'un compteur synchrone modulo $2^3 = 8$ à l'aide des bascules T

Ce compteur possède le cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$

### Diagramme d'état



**Table de vérité**

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

**Les équations des entrées des bascules**

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	0	0	0
1	0	1	1	0

$$T_2 = Q_1 Q_0$$

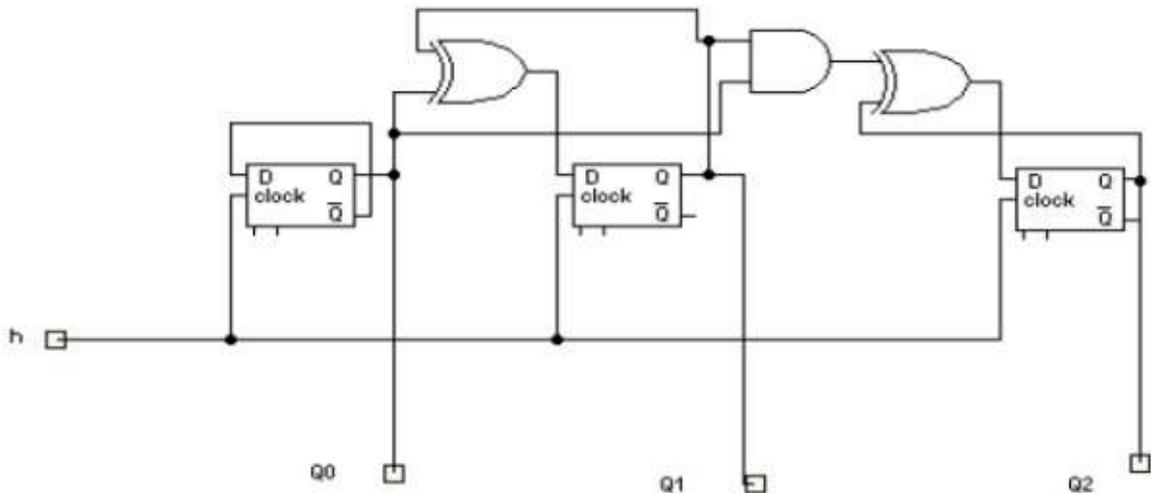
$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	0	0	0
1	1	1	1	1

$$T_1 = Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	1	1	1	1
1	1	1	1	1

$$T_0 = 1$$

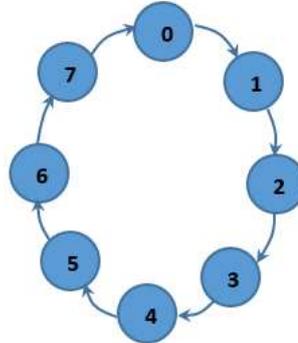
**Schéma d'un compteur modulo 8 synchrone avec des bascules T**



**Exemple 4** : réalisation d'un compteur synchrone modulo  $2^3 = 8$  à l'aide des bascules D

Ce compteur possède le cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ .

**Diagramme d'état**



**Table de vérité**

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

**Les équations des entrées des bascules**

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	0	1	1
1	0	1	0	1

$$\begin{aligned}
 D_2 &= \bar{Q}_2 Q_1 Q_0 + Q_2 \bar{Q}_1 + Q_2 \bar{Q}_0 \\
 &= \bar{Q}_2 Q_1 Q_0 + Q_2 (\bar{Q}_1 + \bar{Q}_0) \\
 &= \bar{Q}_2 Q_1 Q_0 + Q_2 (\overline{Q_1 Q_0}) \\
 &= Q_2 \oplus (Q_1 Q_0)
 \end{aligned}$$

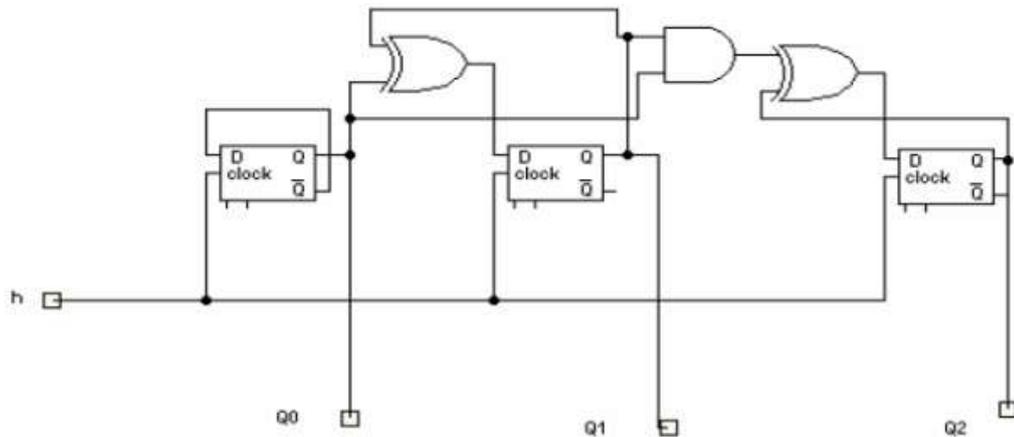
$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	1	1	0
1	1	0	0	1

$$\begin{aligned}
 D_1 &= Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0 \\
 &= Q_1 \oplus Q_0
 \end{aligned}$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	1	1	1	1
1	0	0	0	0

$$D_0 = \bar{Q}_0$$

### Schéma d'un compteur modulo 8 synchrone avec des bascules D



### 3.4.3.2.1.2. Étude des Compteurs synchrones modulo N

**Exemple 1** : réalisation d'un compteur Modulo 6 à l'aide des bascules D

Ce compteur possède le cycle suivant : 0 → 1 → 2 → 3 → 4 → 5

Diagramme d'état

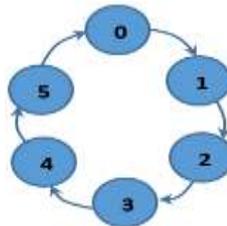


Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	0	0	0	0	0	0

Les équations des entrées des bascules

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	0	X	1
1	0	1	X	0

$$D_2 = Q_2 \bar{Q}_0 + Q_1 Q_0$$

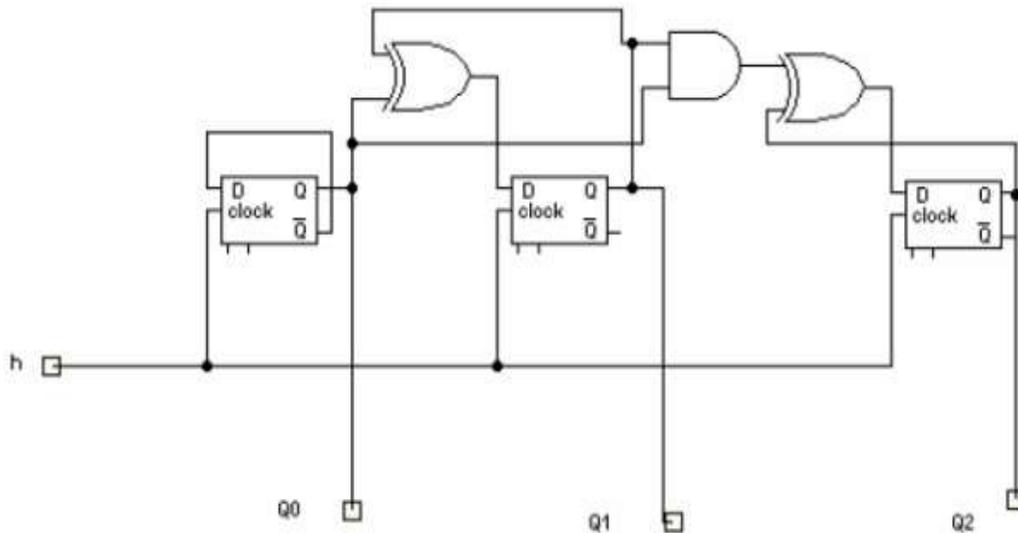
$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	0	1	X	0
1	1	0	X	0

$$D_1 = \bar{Q}_2 \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$$

$Q_2 \backslash Q_1$	00	01	11	10
$Q_0$				
0	1	1	X	1
1	0	0	X	0

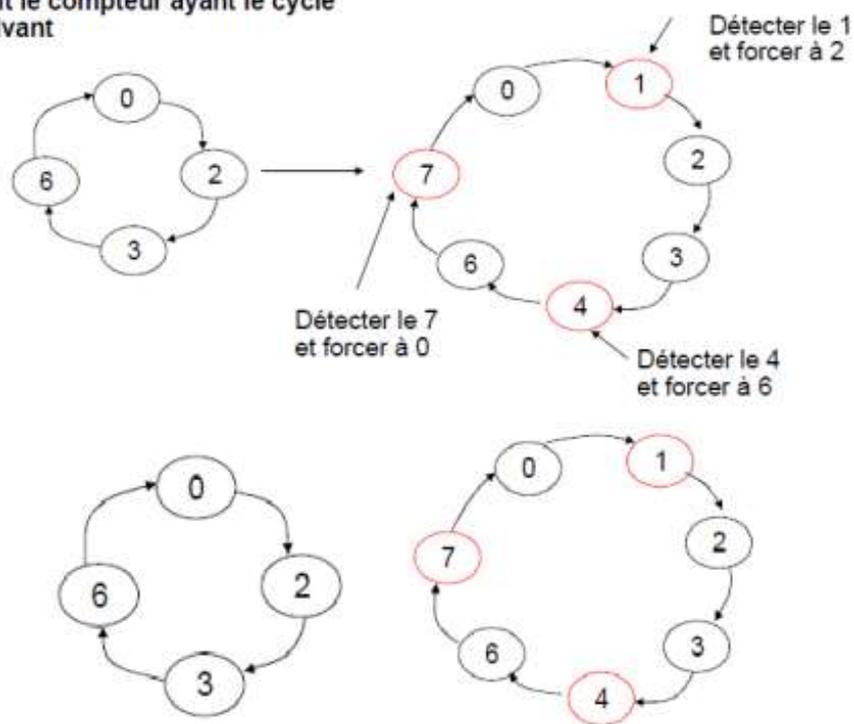
$$D_0 = \bar{Q}_0$$

### Schéma d'un compteur modulo 6 synchrone avec des bascules D



### 3.4.3.2.1.3. Étude des compteurs synchrones à cycle quelconque

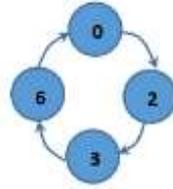
Soit le compteur ayant le cycle suivant



- Pour faire passer le compteur d'un état à un autre, il faut agir sur l'entrée des bascules.
- Pour les états qui n'appartiennent pas au cycle du compteur, ils doivent être considérés comme des états indéterminés

**Exemple :** réalisation d'un compteur synchrone à cycle quelconque à l'aide des bascules JK  
 Dans cet exemple le cycle est le suivant : 0→2→3→6.

## Diagramme d'état



## Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	1	0	0	X	0	X	1	X
0	1	0	0	1	1	0	X	1	X	X	1
0	1	1	1	1	0	0	X	X	0	1	X
1	1	0	0	0	0	1	X	X	1	X	1

## Les équations des entrées des bascules

$Q_2 \ Q_1$	00	01	11	10
$Q_0$	0	0	X	X
1	0	1	X	X

$$J_2 = Q_1 Q_0$$

$Q_2 \ Q_1$	00	01	11	10
$Q_0$	0	X	X	0
1	X	X	0	1

$$K_2 = \overline{Q_1} Q_0$$

$Q_2 \ Q_1$	00	01	11	10
$Q_0$	0	X	X	0
1	1	X	X	1

$$J_1 = Q_0$$

$Q_2 \ Q_1$	00	01	11	10
$Q_0$	0	X	0	0
1	X	1	1	X

$$K_1 = Q_0$$

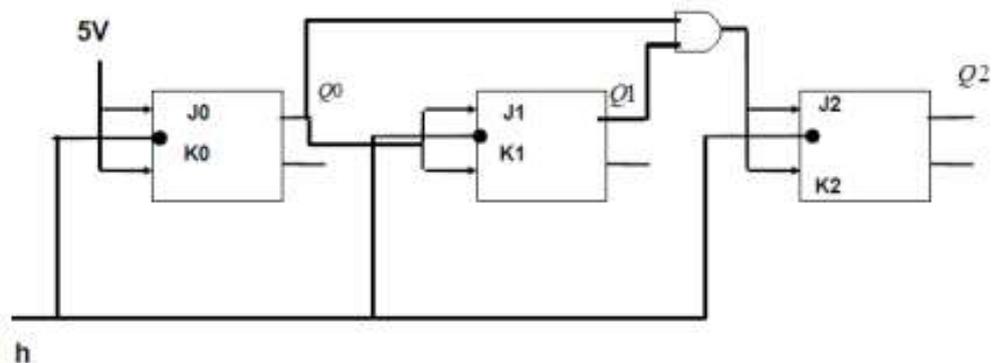
$Q_2 \ Q_1$	00	01	11	10
$Q_0$	0	1	1	1
1	X	X	X	X

$$J_0 = 1$$

$Q_2 \ Q_1$	00	01	11	10
$Q_0$	0	X	X	X
1	1	1	1	1

$$K_0 = 1$$

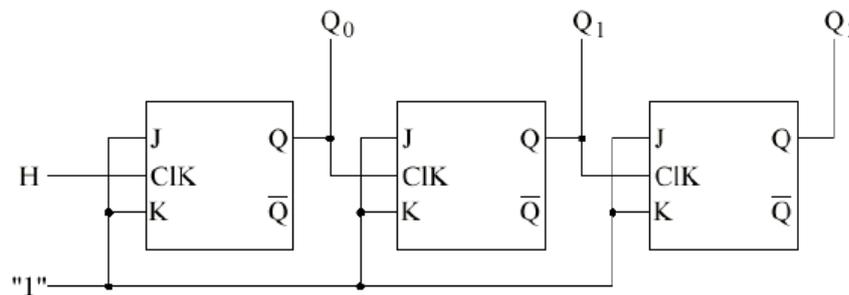
## Schéma d'un compteur synchrone modulo 8



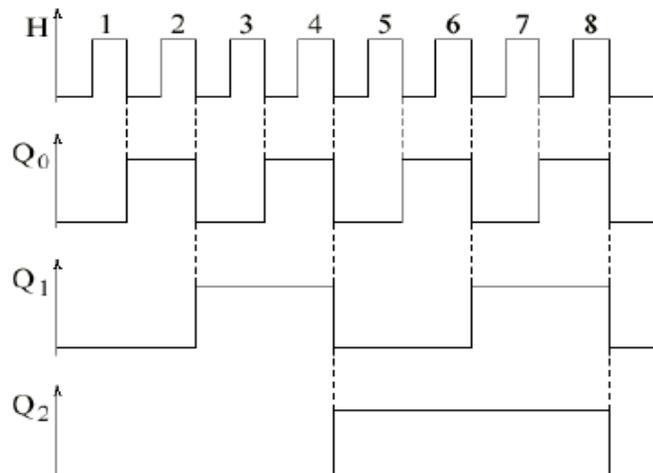
### 3.4.3.2.2. Compteurs asynchrones ou séries

Un compteur asynchrone est un système logique constitué de bascules où les impulsions appliquées à l'entrée doivent passer par la première bascule avant de pouvoir commander la seconde et ainsi de suite jusqu'à la dernière bascule [18]. Cela signifie que le signal d'horloge n'est reçu que par la première bascule LSB (Least Significant Bit). Pour chacune des autres bascules, le signal d'horloge est fourni par la sortie de la bascule de rang immédiatement inférieur.

La figure ci-dessous montre un compteur asynchrone à base de bascules JK. La sortie la première bascule  $Q_0$  change d'état sur chaque front descendant du signal d'horloge. La sortie de la deuxième bascule  $Q_1$  change d'état à chaque transition 1 vers 0 de la sortie  $Q_0$ . La sortie de la troisième bascule  $Q_2$  change d'état à chaque transition 1 vers 0 de la sortie  $Q_1$ .



La figure suivante montre les chronogrammes de ce compteur.



Il existe trois types de compteurs asynchrones selon le cycle réalisé :

#### d. Les compteurs asynchrones modulo $2^n$ (cycle complet)

- modulo 4, signifie que  $n=2$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ .
- modulo 8, signifie que  $n=3$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ .
- modulo 16, signifie que  $n=4$ , de cycle suivant :  
 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$ .

#### e. Les compteurs asynchrones modulo N (cycle incomplet)

- modulo 5, signifie que  $N=5$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ .
- modulo 10, signifie que  $N=10$ , de cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$ .

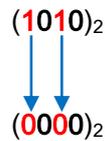
### f. Les compteurs asynchrones à cycle quelconque

- Exemple :  $0 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 10$ .

**Exemple 1** : réalisation d'un compteur asynchrone modulo 10 avec des bascules D.

	$Q_3$	$Q_2$	$Q_1$	$Q_0$
	0	0	0	0
	0	0	0	1
	0	0	1	0
	0	0	1	1
	0	1	0	0
	0	1	0	1
	0	1	1	0
	0	1	1	1
	1	0	0	0
	1	0	0	1
<b>Remise à zéro</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>

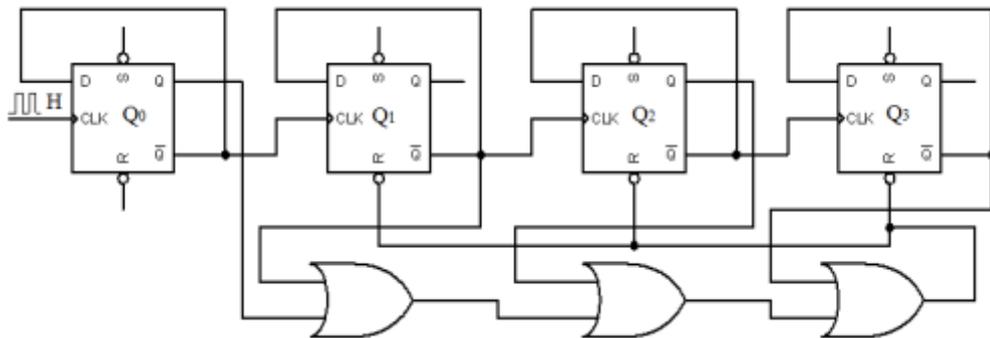
On désire que le compteur passe à  $(0000)_2 = (0)_{10}$  quand il passe de  $(1001)_2$  à  $(1010)_2$ .



On désire que le compteur passe à 0 quand il atteint :  $1010 = (1010)_2$ . Pour cela on peut écrire l'expression logique :

$$R = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} = \overline{Q_3} + Q_2 + \overline{Q_1} + Q_0$$

Ce n'est pas suffisant de remettre à zéro  $Q_3$  et  $Q_1$  ( $(1010)_2$  c'est-à-dire  $Q_3 = 1$ ,  $Q_2 = 0$ ,  $Q_1 = 1$  et  $Q_0 = 0$ ) ; il faut mettre  $Q_2$  à zéro. Sinon, quand  $Q_1$  passe de 1 à 0,  $Q_2$  passe de 0 à 1 ( $Q_1$  horloge de la bascule  $Q_2$ ).

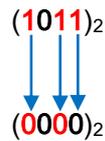


**Exemple 2** : réalisation d'un compteur asynchrone modulo 11 avec des bascules D.

$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1

Remise à zéro

On désire que le compteur passe à  $(0000)_2 = (0)_{10}$  (l'entrée R soit à 0) quand il passe de  $(1010)_2$  à  $(1011)_2$ .



On désire que le compteur passe à 0 quand il atteint  $(11)_{10} = (1011)_2$ . Pour cela on peut écrire l'expression logique :

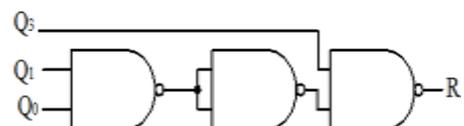
$$R = \overline{Q_3 Q_2 Q_1 Q_0} = \overline{Q_3} + \overline{Q_2} + \overline{Q_1} + \overline{Q_0}$$

Ce n'est pas suffisant de remettre à zéro  $Q_3$ ,  $Q_1$  et  $Q_0$  ( $(1011)_2$  c'est-à-dire  $Q_3 = 1$ ,  $Q_2 = 0$ ,  $Q_1 = 1$  et  $Q_0 = 1$ ) ; On peut simplifier cette expression en ne tenant compte que des sorties à 1. On peut écrire :

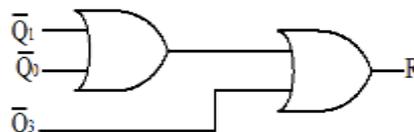
$$R = \overline{Q_3 Q_1 Q_0} = \overline{Q_3} + \overline{Q_1} + \overline{Q_0}$$

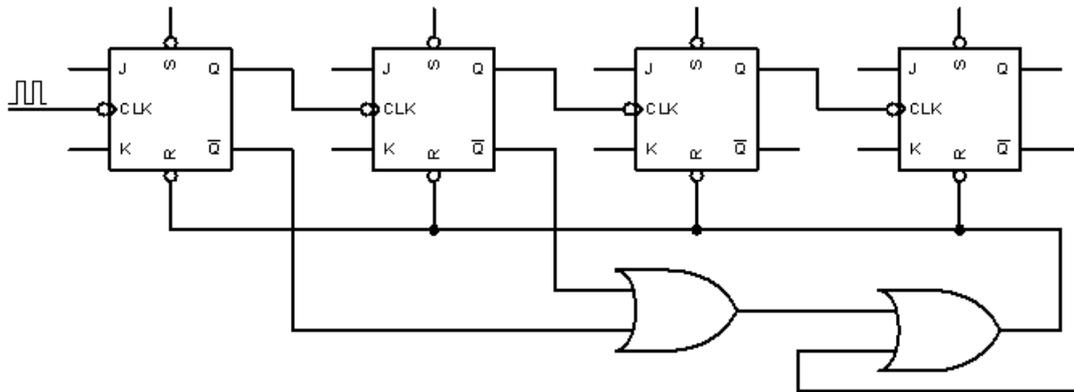


Ou



Ou





Lorsque les bascules D sont en cascade, la première impulsion d'horloge change l'état de la première bascule, la seconde la fait changer à nouveau d'état et cela provoque le changement de la deuxième bascule, ...etc [19].

### 3.4.4. Utilisation des bascules pour réaliser Les décompteurs

**Exemple 1 :** réalisation d'un décompteur asynchrone modulo 10 à l'aide des bascules JK

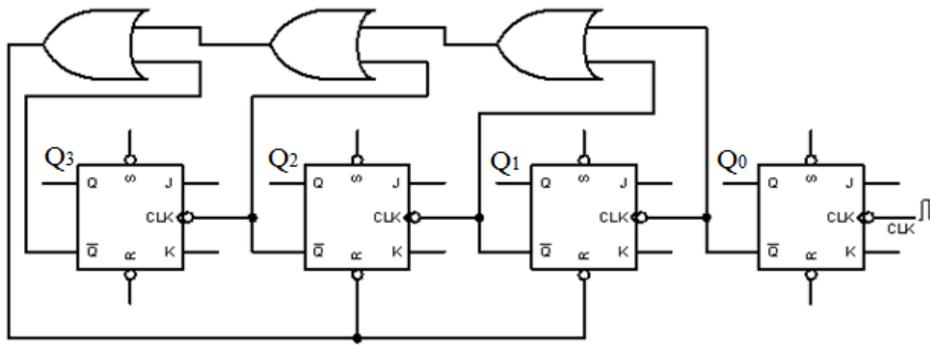
	$Q_3$	$Q_2$	$Q_1$	$Q_0$
	1	0	0	1
	1	0	0	0
	0	1	1	1
	0	1	1	0
	0	1	0	1
	0	1	0	0
	0	0	1	1
	0	0	1	0
	0	0	0	1
	0	0	0	0
<b>Remise à neuf</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>

On désire que le décompteur passe à  $(1001)_2 = (9)_{10}$  quand il passe de  $(0000)_2$  à  $(1111)_2$

$$\begin{array}{c}
 (1111)_2 \\
 \Downarrow \\
 (1001)_2
 \end{array}$$

Donc, il suffit de remettre  $Q_2$  et  $Q_1$  à zéro. On veut que le décompteur passe à 9 quand il atteint  $(15)_{10} = (1111)_2$ . Alors, on peut écrire l'expression suivante :

$$R = \overline{Q_3 Q_2 Q_1 Q_0} = \overline{Q_3} + \overline{Q_2} + \overline{Q_1} + \overline{Q_0}$$



**Exemple 2 :** réalisation d'un décompteur asynchrone de cycle suivant : 9→8→7→6→5→4 à l'aide des bascules JK.

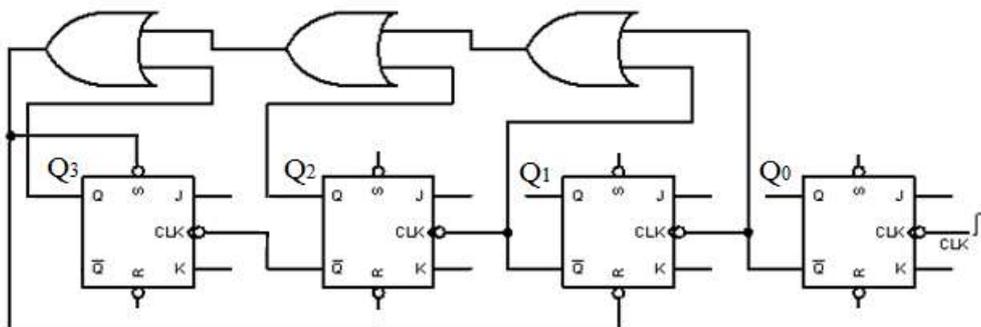
	$Q_3$	$Q_2$	$Q_1$	$Q_0$
	1	0	0	1
	1	0	0	0
	0	1	1	1
	0	1	1	0
	0	1	0	1
	0	1	0	0
<b>Remise à neuf</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>

On désire que le décompteur passe à  $(1001)_2 = (9)_{10}$  quand il passe de  $(0100)_2$  à  $(0011)_2$

$(0011)_2$   
 $\downarrow$   
 $\downarrow$   
 $(1001)_2$

Donc, il suffit de remettre  $Q_1$  à zéro ( $R_1$ ) et  $Q_3$  à 1 ( $S_3$ ).

$$R_1 = S_3 = \overline{Q_3} \overline{Q_2} Q_1 Q_0 = Q_3 + Q_2 + \overline{Q_1} + \overline{Q_0}$$



### 3.4.5. Utilisation des bascules pour réaliser Les compteurs/décompteurs

Le compteur/décompteur est un circuit séquentiel qui peut réaliser les deux opérations de comptage et décomptage, il possède une entrée de commande  $C_m$ .

$$\begin{cases} C_m = 0 \Rightarrow \text{comptage} \\ C_m = 1 \Rightarrow \text{décomptage} \end{cases}$$

**Exemple 1** : un compteur/décompteur synchrone modulo 8 à l'aide de bascules T

**Table de vérité**

C	$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	0	1	1	0	1	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	0	1	0	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1

**Les équations des entrées des bascules**

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	0	0	0	0

$$T_2 = \bar{C}Q_1Q_0 + C\bar{Q}_1\bar{Q}_0$$

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$T_1 = \bar{C}Q_0 + C\bar{Q}_0$$

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$T_0 = 1$$

**Exemple 2** : un compteur/décompteur asynchrone modulo 8 à l'aide de bascules T

**Table de vérité**

C	$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	0	1	1	0	1	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	0	1	0	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1

**Les équations des entrées des bascules**

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	0	0	0	0

$$T_2 = \bar{C}Q_1Q_0 + C\bar{Q}_1\bar{Q}_0$$

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$T_1 = \bar{C}Q_0 + C\bar{Q}_0$$

$C Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$T_0 = 1$$

### 3.5. Synthèse de circuits séquentiels

La théorie des automates finis est un modèle théorique très utile pour synthétiser des circuits séquentiels trop complexes.

#### 3.5.1. Définition d'une machine à états finis

Un automate fini ou machine à états finis (MEF ou en anglais FSM : *Finite State Machine*) est un système séquentiel qui peut se trouver couramment dans les mémoires. Une machine d'états finis comporte une partie combinatoire et les bascules d'états. La partie combinatoire contient la logique de calcul des sorties et des états suivants [20]. On peut les utiliser pour mémoriser des informations. Autres appellations : (Machine à états, Machine à états finis, Séquenceur, Contrôleur, Automate, Machine séquentielle ou Machine séquentielle algorithmique).

Un automate prend un nombre fini d'état (ou nœuds). Il est caractérisé par :

- Sa sortie  $s$ .
- Sa entrée  $e$ .
- Son état  $q$ .

D'où, formellement une FSM est un sextuplet  $M = (Q, U, Y, Init, R, S)$ , où

$Q$  : un ensemble fini d'états de la machine.

$U$  : un ensemble fini de signaux d'entrée.

$Y$  : un ensemble fini de signaux de sortie.

$Init \subseteq Q$  : Un ensemble d'états initiaux.

$R : Q \times U \rightarrow Q$ , une fonction appelée fonction de transition.

$S : Q \times U \rightarrow Y$ , une fonction appelée fonction de sortie.

**Remarque** : On peut avoir une FSM où tout état peut être un état initial donc, inutile de spécifier l'ensemble des états initiaux ou sans sortie donc, la fonction  $Y$  n'a aucun sens, ...etc. On peut représenter une machine par :

- Un chronogramme.
- Une équation logique.
- Une table de transition.
- Un diagramme d'état.
- Une structure logique.

Si les concepts de chronogramme, équation logique et structure logique sont connus les notions de tables de transitions et diagramme d'état restent à définir.

##### 3.5.1.1. Tables de transitions

Les diagrammes de transitions ou tables de transitions ou matrices de transitions décrivent sous forme matricielle les fonctions  $R$  et  $S$ . Elles doivent contenir les états  $q_i$  à l'instant  $t$ , les entrées  $e_i$ , les états futurs  $q_i^+$  et les sorties  $s_i$ .

### Exemple

Soit un automate  $M = (Q, U, Y, R, S)$  définie par :  $Q = U = Y = \{0,1\}$  et dont les fonctions de transition et de sortie sont définies par les tables suivantes :

R	0	1
0	0	1
1	0	1

S	0	1
0	0	0
1	1	1

-Catégories de MSA

➤ Synchronne

Un automate est dit synchronne lorsque le passage d'un état présent au état futur se produit lors de la transition d'un signal horloge commun à toutes les bascules de l'automate [21].

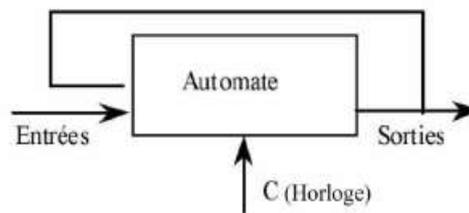


Figure 3. 23. Automate synchronne

➤ Principalement synchronne

➤ Asynchronne

#### 3.5.1.2. Diagramme d'état

On peut représenter un automate d'une autre manière, par exemple en utilisant un graphe de transition, c'est-à-dire une description équivalente des tables de transitions.

- Les états sont représentés par des cercles.
- Les transitions sont représentées par des arcs orientés, allant de l'état initial à l'état final.
- Les états initiaux par deux cercles.



Figure 3. 24. Exemples de diagramme d'états

#### Remarques

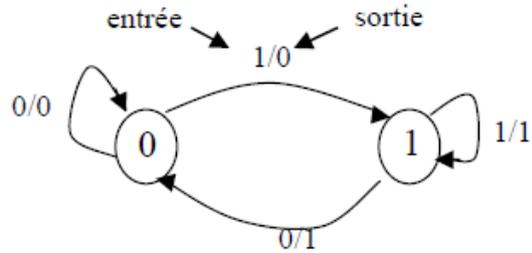
- Une transition est inconditionnelle, c'est-à-dire que si le système est dans l'état source considéré, la transition se produit au prochain front d'horloge actif.
- Une transition est conditionnelle, c'est-à-dire que lorsque le système est à l'état source, la transition se produit au prochain front d'horloge actif si une condition sur les entrées est vérifiée.

- Un état peut être conservé pour certaines valeurs d'entrée. Dans ce cas, l'arc orienté représentant la transition est fermé sur le même cercle.

### Exemples

1. Diagramme d'état de l'exemple précédent.

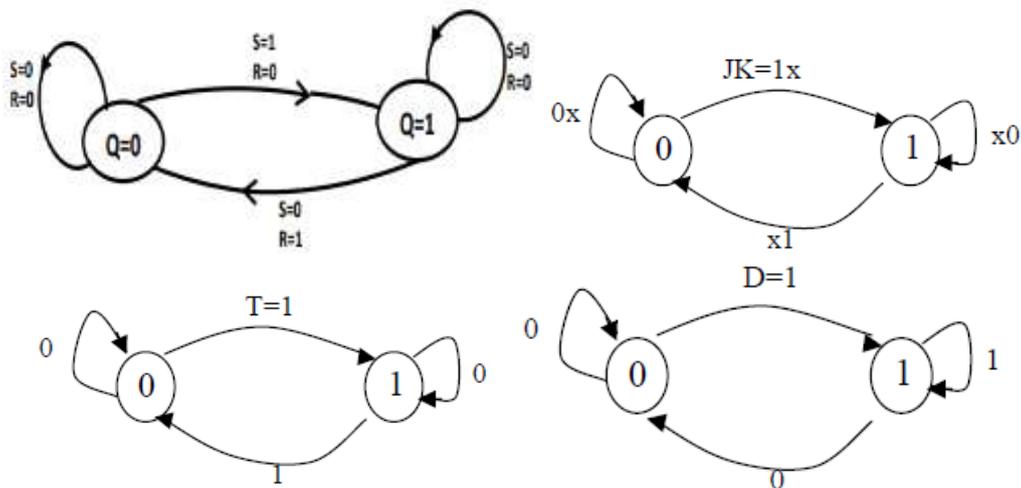
Il suffit de traduire les données des tables de transitions en diagrammes d'états.



2. Diagrammes d'état des bascules RS, JK, T et D

Rappelons les tables caractéristiques réduites des bascules JK, T et D

$Q$	$Q^+$	$J$	$K$	$T$	$D$
0	0	0	X	0	0
0	1	1	X	1	1
1	0	X	1	1	0
1	1	X	0	0	1



### Structure des machines à états finis

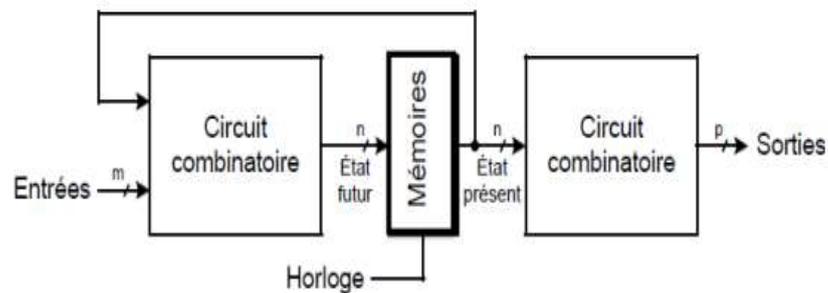


Figure 3. 25. Exemple d'une machine à états finis.

### 3.5.2. Classes de MSA

Souvent, on distingue entre deux modèles de machines d'états finis à savoir le modèle de Moore et modèle de Mealy [20]. Ils diffèrent seulement par la manière dont la sortie est générée.

#### 3.5.2.1. Machine de Moore

Dans une machine de Moore, l'état des sorties n'est qu'une fonction de l'état actuel de la machine, résultant de l'état actuel des entrées et de l'état antérieur de la machine. C'est le vecteur d'état, qui permet de prendre en compte l'état antérieur de la machine.

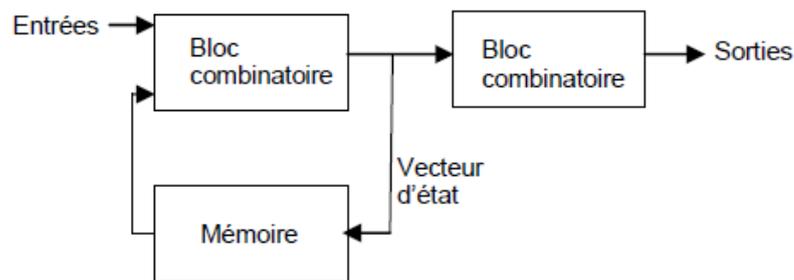


Figure 3. 26. Structure d'une machine de Moore.

#### 3.5.2.2. Machine de Mealy

Dans une machine de Mealy, l'état des sorties est déterminé à partir de l'état actuel des entrées et également à partir de l'état actuel de la machine, matérialisé par le vecteur d'état.

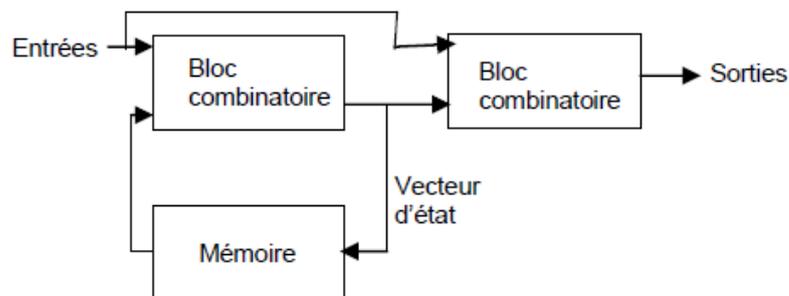


Figure 3. 27. Structure d'une machine de Mealy.

### 3.5.3. Analyse d'un circuit séquentiel

Analyser un circuit séquentiel c'est déterminer son rôle. Pour analyser un circuit séquentiel, on peut suivre les étapes suivantes :

1. Déterminer les fonctions des variables d'entrée.
2. Dresser la table caractéristique du circuit. De cette table, déduire les variables de sortie en se basant sur les expressions logiques des variables d'entrée. Elle a la forme suivante :

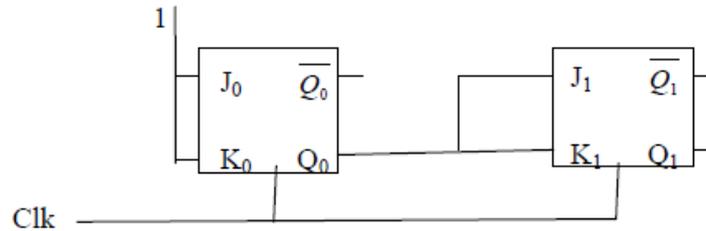
Variables d'entrée	$Q$	$Q^+$
connue	connue	A déterminer

Pour déterminer les états de sortie  $Q^+$ , il faut utiliser la table caractéristique dont il est question dans le circuit à analyser.

3. Déduire le rôle du circuit analysé.

### Exemple

Analysons le circuit suivant :



Détermination des fonctions d'entrée de chaque bascule.

$$\text{bascule 0} \begin{cases} J_0 = 1 \\ K_0 = Q_0 \end{cases} \qquad \text{bascule 1} \begin{cases} J_1 = Q_0 \\ K_1 = Q_1 \end{cases}$$

Table caractéristique.

$Q_1$	$Q_0$	$J_1$	$K_1$	$J_0$	$K_0$	$Q_1^+$	$Q_0^+$
0	0	0	0	1	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0

Conclusion : C'est un compteur binaire comptant de 0 jusqu'à 3.

### 3.5.4. Synthèse d'un circuit séquentiel

La synthèse d'un circuit séquentiel consiste à élaborer le circuit logique à partir du cahier de charge. Pour faire la synthèse d'un circuit séquentiel, on suit les étapes suivantes :

1. Etablir la table d'excitation correspondant au circuit à réaliser. Elle est de la forme :

$Q$	$Q^+$	Variables d'entrée
connue	connue	A déterminer

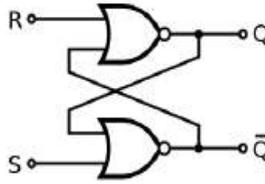
2. Déduire l'expression de chaque variable d'entrée.

3. Réaliser le circuit en utilisant les bascules et les portes logiques requises.

## Série d'exercices N° 2(Circuits Séquentiels)

### Exercice N°1 :

A. Cette bascule est-elle synchrone ou asynchrone ?



B. Remplir sa table de vérité suivante.

R	S	Q	Q <sup>+</sup>	Observation
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

### Exercice N°2:

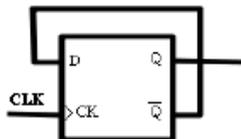
1 • Dresser la table de vérité de la bascule SR.

2 • Compléter le tableau suivant, en indiquant l'état de la sortie Q de la bascule RS pour chaque instant T.

T	1	2	3	4	5	6	7	8	9	10
S	0	0	0	1	1	0	0	1	1	0
R	0	0	1	0	0	0	1	1	0	0
Q	0									

### Exercice N°3:

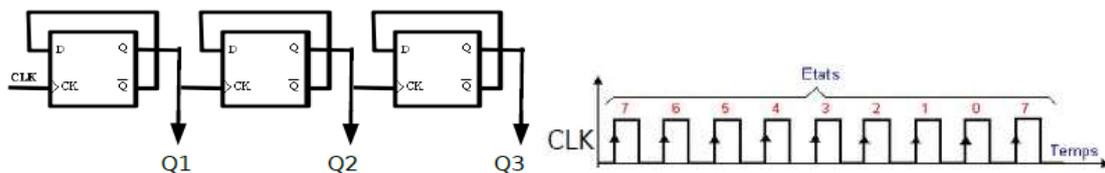
Soit le montage suivant



A. Compléter le chronogramme de la sortie de bascule D.



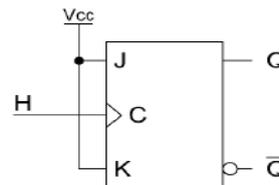
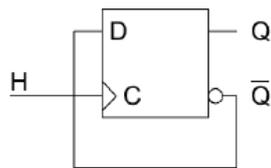
B. A partir du chronogramme précédent de la bascule D, déduire le chronogramme du circuit suivant (Q<sub>1</sub>, Q<sub>2</sub> et Q<sub>3</sub>):



C. Ce circuit est-il synchrone ?

**Exercice N°4:**

- 1 • Dresser les tables de vérité des bascule D et JK synchronisées sur front montant.
- 2 • Donner les chronogrammes des sorties de chacune des bascules de la figure ci-dessous en fonction d'une entrée d'horloge H.



**Exercice N°5 :**

En utilisant les bascules JK synchronisées sur front montant. Réaliser un compteur synchrone modulo 8. Les étapes à suivre :

1. Donner la table des transitions d'une bascule JK.
2. Donner le diagramme d'état.
3. Dresser la table de vérité via la table des transitions.
4. Donner les expressions des entrées des bascules  $J_0, K_0, J_1, K_1, J_2$  et  $K_2$ .
5. En déduire le schéma logique du compteur.

**Exercice N°6 :**

1. étudier et donner les schémas de câblage des compteurs suivants à l'aide des bascules JK synchronisées sur front montant :

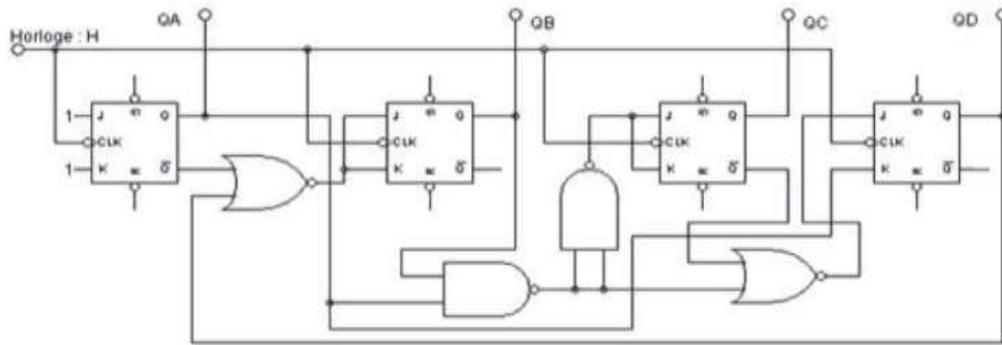
- a. Compteur synchrone modulo 10.
- b. Compteur synchrone de cycle suivant :  $0 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 10 \rightarrow 12$ .

2. étudier et donner les schémas de câblage des décompteurs suivants à l'aide des bascules D synchronisées sur front montant :

- a. Décompteur synchrone modulo 8
- b. Décompteur synchrone de cycle suivant :  $0 \rightarrow 3 \rightarrow 4 \rightarrow 7$ .

**Exercice N°7:**

Soit le circuit séquentiel suivant :



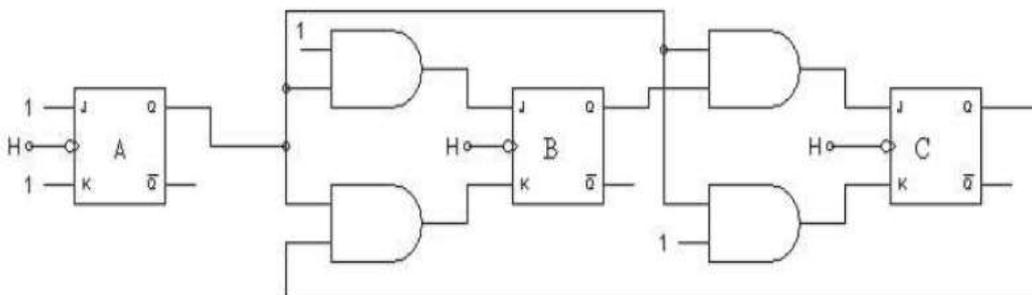
1. Donner les expressions des entrées des bascules  $J_A, K_A, J_B, K_B, J_C, K_C, J_D$  et  $K_D$ .
2. Compléter le tableau suivant, Supposons que le compteur commence à partir de l'état  $Q_D Q_C Q_B Q_A = 0000$ .

$Q_D$	$Q_C$	$Q_B$	$Q_A$		$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0									

3. Tracer les chronogrammes des sorties des bascules  $Q_A, Q_B, Q_C$  et  $Q_D$  en fonction d'une entrée d'horloge H.
4. Déduire le cycle réalisé par ce compteur.

**Exercice N°8 :**

Analyser le fonctionnement du circuit séquentiel ci-dessous, en commence d'un état initial  $Q_C Q_B Q_A = 000$ .



1. Déterminer les expressions des entrées des trois bascules.
2. Compléter le tableau suivant :

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

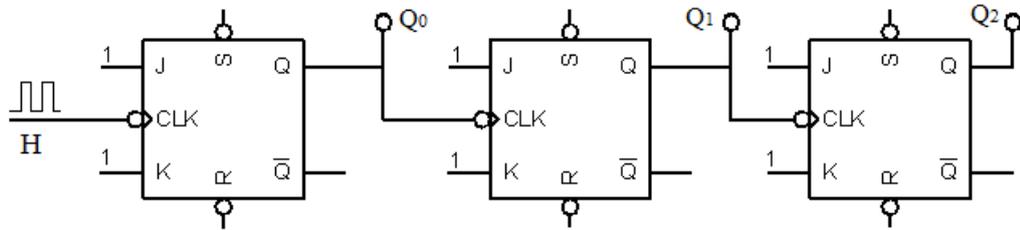
3. Tracer les chronogrammes des sorties des bascules  $Q_A, Q_B$  et  $Q_C$  en fonction d'une entrée d'horloge H.

4. Dédurre le cycle réalisé par ce circuit séquentiel.

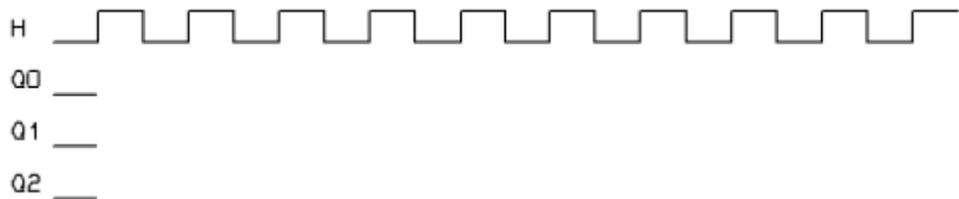
5. calculer les fréquences  $f_A$ ,  $f_B$  et  $f_C$  des sorties des bascules  $Q_A$ ,  $Q_B$  et  $Q_C$  en fonction de la fréquence de l'horloge  $f_H$ .

**Exercice N°9 :**

Soit le circuit séquentiel suivant :

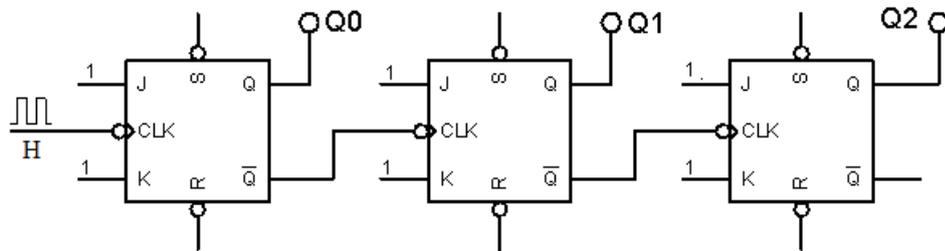


1. Tracer les chronogrammes des sorties des bascules  $Q_0$ ,  $Q_1$  et  $Q_2$  en fonction de l'horloge H.

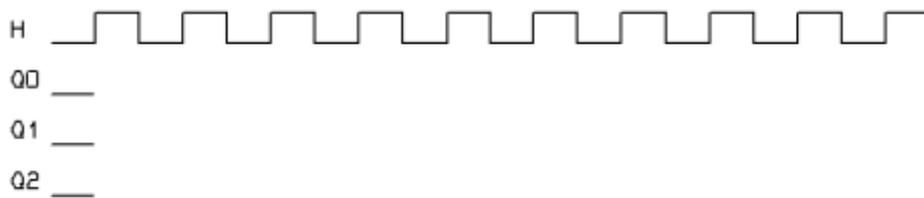


2. Dédurre la séquence réalisée.
3. Dédurre le modulo de ce circuit ?

Soit le circuit séquentiel suivant :



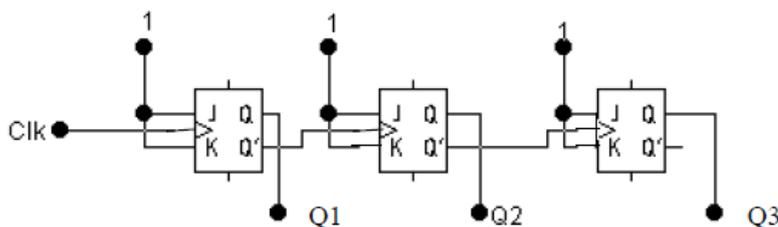
4. Tracer les chronogrammes des sorties des bascules  $Q_0$ ,  $Q_1$  et  $Q_2$  en fonction de l'horloge H.



5. Dédurre la séquence réalisée.
6. Dédurre le modulo de ce circuit ?
7. Conclure pour le compteur et le décompteur.
8. A partir des schémas de deux circuits précédents concevoir un circuit compteur/décompteur.

**Exercice N°10 :**

Soit le circuit ci-dessous à base des bascules JK synchronisées sur front montant.



1. Décrire le comportement des trois bascules de ce circuit.
2. Ce circuit est-il synchrone ou asynchrone ? Justifier ?
3. Tracer les chronogrammes de fonctionnement des sorties de ce circuit en fonction de l'horloge H.
4. donner la table des transitions de ce circuit.
5. En déduire le travail effectué par ce circuit.

**Exercice N°11 :**

Réaliser un registre à décalage de droite vers la gauche avec une entrée série droite et une sortie série gauche à base de bascules D synchrone au front montant.

**Exercice N°12 :**

En vous basant sur la bascule D. Réaliser un registre à décalage à droite et à gauche 4 bits, selon une entrée de sélection S tel que :

$$S = \begin{cases} 0 \Rightarrow \text{décalage à droite} \\ 1 \Rightarrow \text{décalage à gauche} \end{cases}$$

Ce registre permet de d'effectuer un décalage à gauche ou à droite à chaque front montant de l'horloge.

**Exercice N°13 :**

Soit M un automate, tel que  $M = (Q, U, Y, Init, R, S)$

où  $Q = \{p, q, r\}$  et  $U = Y = \{0,1\}$  ; définie par les diagrammes de transition et de sortie suivants :

R	0	1
p	q	p
q	r	q
r	p	r

S	0	1
p	0	1
q	1	0
r	1	1

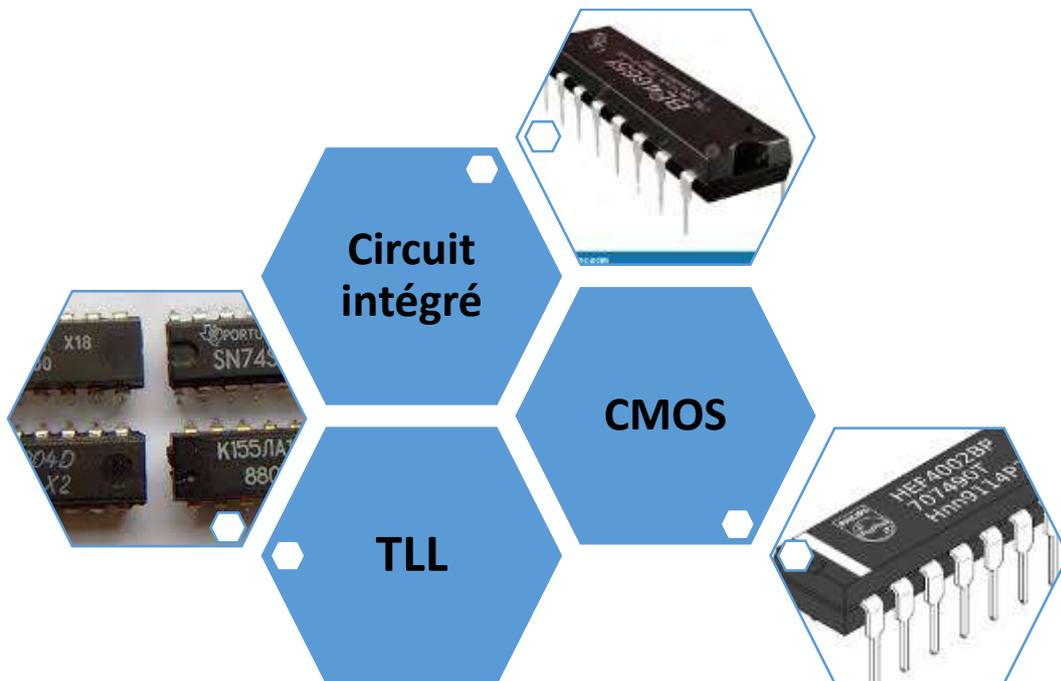
Tracer le graphe d'état de cet automate.

**Exercice N°14 :**

On considère une mémoire centrale de 2 MO, où chaque octet peut être adressé séparément.

1. Calculer l'adresse du sixième élément d'un tableau en octal, tel que l'adresse du premier élément est  $(77)_8$  et tous les éléments sont constitués de 16 bits.
2. Calculer la taille de cette mémoire en mots de 16 bits et en mots de 32 bits.

# Chapitre IV : Conception des Circuits Intégrés



## Chapitre IV : Conception des Circuits Intégrés

### 4.1. Définition d'un circuit intégré

Un circuit intégré est un bloc composé d'un cristal de silicium (puce) d'une surface de plusieurs millimètres carrés à l'intérieur duquel est inscrit un nombre variable de composants électroniques de base (transistors, diodes, résistances, condensateurs, ... etc.).

Les circuits intégrés logiques sont classés selon leur technologie de fabrication en plusieurs familles (bipolaire TTL, bipolaire ECL, MOS, ...etc.). Chaque technologie offre des performances différentes sur le plan électrique (tensions, courants, puissances) et temporel (rapidité).

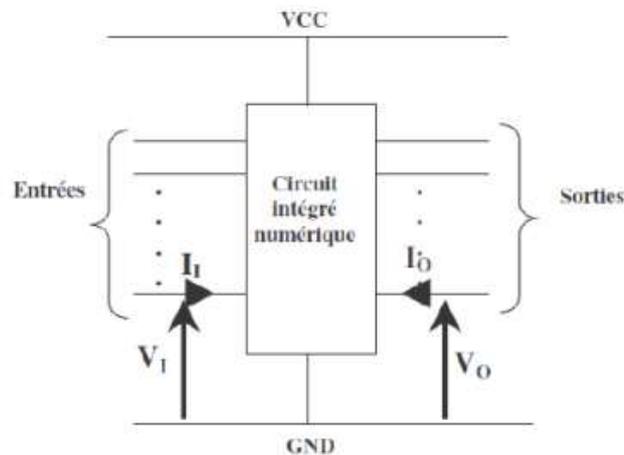


Figure 4.1. Représentation d'un circuit intégré.

### 4.2. Caractéristiques des circuits intégrés

Les circuits intégrés logique sont caractérisés par les paramètres électriques suivants :

#### a. Tension d'alimentation

Tension d'alimentation  $V_{CC}$  : représente la tension nominale nécessaire pour alimenter les circuits intégrés.

Famille	Tension
TTL(série 74)	$V_{CC}=5V+5\%$
TTL(série 54)	$V_{CC}=5V+10\%$
CMOS4000	$V_{DD}=3 \text{ à } 15V$

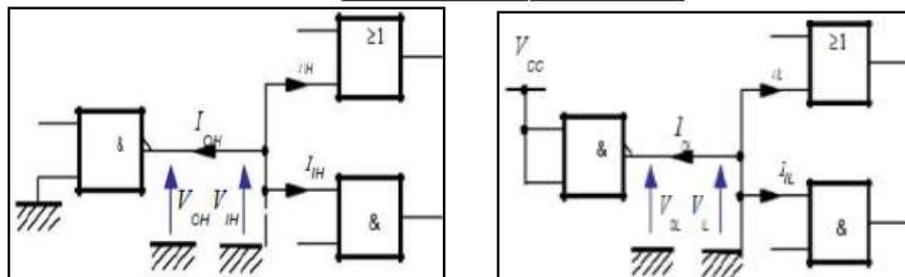


Figure 4.2. Les deux niveaux logiques haut (high level) et bas (low level).

**b. La plage des tensions et des courants associés à chaque niveau logique, en entrée ou en sortie :**

➤ Les tensions d'entrée et de sortie pour chaque niveau logique :

- Tension d'entrée pour le niveau haut ( $V_{IH}$ ) : représente le niveau de tension nécessaire pour avoir l'état logique 1 en entrée d'un circuit intégré.
- Tension d'entrée pour le niveau bas ( $V_{IL}$ ) : représente le niveau de tension nécessaire pour avoir l'état logique 0 en entrée d'un circuit intégré.
- Tension de sortie pour le niveau haut ( $V_{OH}$ ) : représente le niveau de tension nécessaire pour avoir l'état logique 1 en sortie d'un circuit intégré.
- Tension de sortie pour le niveau bas ( $V_{OL}$ ) : représente le niveau de tension nécessaire pour avoir l'état logique 0 en sortie d'un circuit intégré.

➤ Les courants d'entrée et de sortie pour chaque niveau logique :

- Courant d'entrée pour le niveau haut ( $I_{IH}$ ) : représente le courant qui traverse une borne d'entrée d'un circuit intégré quand une tension niveau haut est appliquée à cette entrée.
- Courant d'entrée pour le niveau bas ( $I_{IL}$ ) : représente le courant qui traverse une borne d'entrée d'un circuit intégré quand une tension niveau bas est appliquée à cette entrée.
- Courant d'entrée pour le niveau haut ( $I_{OH}$ ) : représente le courant qui traverse une borne de sortie d'un circuit intégré quand une tension niveau bas est appliquée à cette sortie.
- Courant d'entrée pour le niveau haut ( $I_{IH}$ ) : représente le courant qui traverse une borne de sortie d'un circuit intégré quand une tension niveau haut est appliquée à cette sortie.

**c. Classes d'intégrations**

Selon l'ordre chronologique, il existe quatre classes d'intégration, comme le montre le tableau ci-dessous :

La classe	Abréviation	Nombre de transistors par $\text{cm}^2$
Les microcircuits SSI	Single Size Intégration	$\approx 100$ transistors par $\text{cm}^2$ .
Les circuits intégrés MSI	Médium Size Intégration	$\approx 1000$ transistors par $\text{cm}^2$ .
Les circuits LSI	Large Size Intégration	$\approx 10000$ à $100000$ transistors par $\text{cm}^2$ .
Les circuits VLSI	Very Large Size Intégration	$\approx 0.1$ à $1$ million transistors par $\text{cm}^2$ .

**d. Gabarit de tension**

Chaque niveau logique possède une plage de tension, par exemple la plage de tension pour le niveau logique 1 est comprise entre  $V_{cc}$  et la limite inférieure à  $V_{cc}$ , mais la plage de tension pour le niveau logique 0 est comprise entre 0 V et la limite supérieure, comme le montre sur la figure (4.3).

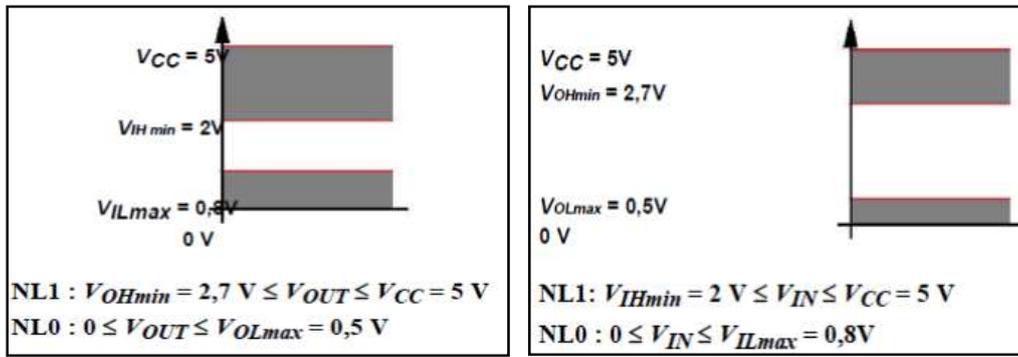


Figure 4.3. Gabarit des tensions : a) d'entrée, b) de sortie

### e. Gabarit de transfert

Le gabarit de transfert représente la combinaison de deux graphes des deux niveaux logiques (figure (4.3)) en un seul pour traduire la fonction logique entre ces tensions.

Une porte satisfait le gabarit si sa courbe de transfert est dans la partie non grisée. La tension de basculement  $V_T$  correspond à la tension d'entrée pour laquelle la sortie change d'état.

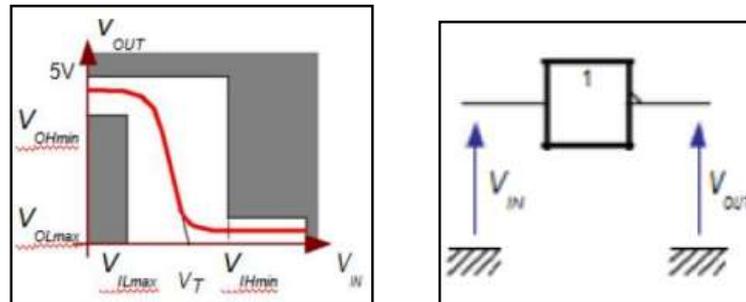


Figure 4.4. Caractéristique de transfert d'une porte inverseuse.

### f. Compatibilité des niveaux logiques

Pour une compatibilité au niveau logique haut,  $V_{OHmin}$  doit être supérieur à  $V_{IHmin}$ , et pour une compatibilité au niveau logique bas,  $V_{ILmax}$  doit être supérieur à  $V_{OLmax}$ .

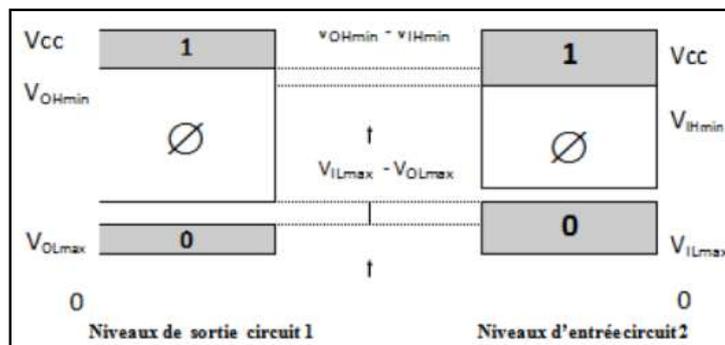
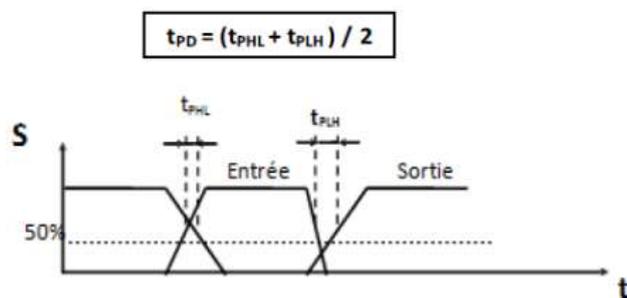


Figure 4.5. Compatibilité des niveaux logiques.

### i. Temps moyen de propagation

Lorsqu'un niveau logique est appliqué à l'entrée d'un circuit, il y a un certain délai pour que la sortie réagisse. Ce délai est le temps moyen de propagation  $t_{PD}$ .



- $t_{PHL}$ : Temps de propagation du niveau haut au niveau bas, autrement dit le temps de montée des signaux en sortie d'une porte logique.
- $t_{PLH}$ : Temps de propagation du niveau bas au niveau haut, autrement dit le temps de descente des signaux en sortie d'une porte logique.

### g. Facteur de charge : Sortance N

Ce paramètre caractérise le nombre maximum N d'entrées de porte logique pouvant être commandées par la sortie d'un autre opérateur logique de la même famille.

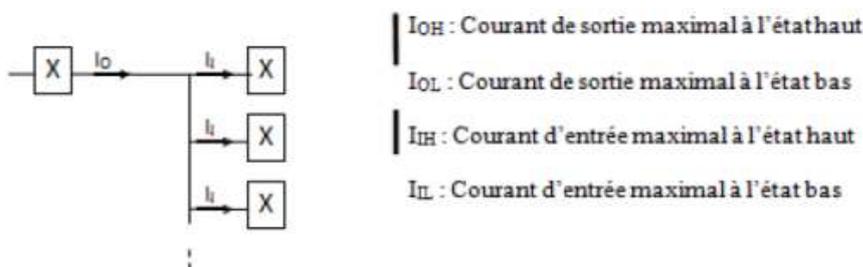


Figure 4.7. Facteur de charge.

- $I_{OH} = \text{sortance } N$  (à l'état logique haut).
- $I_{OL}/I_{IL} = \text{sortance } N$  (à l'état logique bas).

## 4.4. Présentation des CI

Un circuit intégré contient :

- Plusieurs portes logiques, dont leurs entrées et sorties sont accessibles sur les différentes bornes du circuit intégré.
- L'ergo est un petit creux qui permet d'orienter correctement le circuit intégré pour repérer ses différentes bornes. L'ergo est situé en dessous dans les circuits intégrés.
- Les branches (pin) : sont numérotées, et pour déterminer les positions de ces pins, il faut repérer une encoche sur le circuit tel que en regardant le circuit intégré avec l'ergo

vers le haut, la borne n°1 est la borne située en haut à gauche, les autres bornes sont numérotées en tournant dans le sens inverse des aiguilles d'une montre.

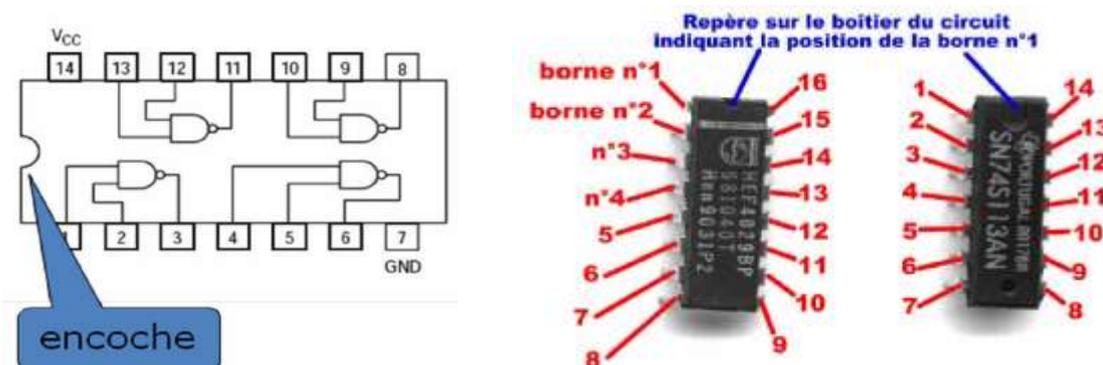


Figure 4.8. Présentation des circuits intégrés.

#### 4.5. Identification des CI

Tout circuit intégré a une référence imprimée sur le dessus de son boîtier. Cette référence est composée de quatre à sept caractères numériques et/ou alphabétiques.

#### 4.6. Classement des circuits intégrés

Selon la densité d'intégration (nombre de portes par circuit ou nombre de transistors par circuit), on peut classer les circuits intégrés en six familles :

famille	Année	Nombre de portes logique	Exemple
<b>SSI : Small Scale Integration</b>	1961-1966	1-10	Portes logiques : AND, OR, NOT,....etc.
<b>MSI : Medium Scale Integration</b>	1967-1971	10-100	Bascules, compteurs, multiplexeurs, décodeurs,etc
<b>LSI :Large Scale Integration</b>	1972-1980	100-10000	Mémoire de petite capacité, circuit logique programmable.
<b>VLSI : Very Large Scale Integration</b>	1981-1990	10000-100000	Mémoire de capacité importante, microprocesseur.
<b>ULSI : Ultra Large Scale Integration</b>	1990-2000	100000-1000000	Microprocesseur graphique.
<b>GLSI :Giga Large Scale Integration</b>	2000- aujourd'hui	>1000000	Pentium Dual Core microprocesseur.

Figure 4.9. Classement des circuits intégrés.

## 4.7. Technologies de fabrication de CI

Le Die d'un circuit intégré est constitué de formes miniaturisées principalement de transistors, de diodes, de résistances, de condensateurs, et rarement d'inductances, car elles sont plus difficiles à miniaturiser.

Selon la technologie de fabrication d'un circuit intégré, on peut classer les circuits intégrés en quatre familles :

- **DTL (Diode Transistor Logic)**, Il a été rapidement abandonné, Ancienne technologie.
- **TTL (Transistor Transistor Logic)**, ( $V_{CC}=5V$ ,  $H=5V$ ,  $L=0V$ ) meilleur compromis vitesse/consommation. Dont la référence commence par 74.
- **ECL (Emitter Coupled Logic)** consommation importante, très rapide (communications rapides Gbit/s).
- **CMOS (Complementary Metal Oxide Semiconductor)** ( $V_{CC}=0.8V$  à  $18V$ ,  $H=V_{CC}$ ,  $L=0V$ ) très basse consommation, technologie actuellement dominante. Dont la référence est de la forme 4000.

Notons que :

- $V_{CC}$  : tension d'alimentation du circuit ;
- $H$  : niveau de tension haut équivalent à 1 logique ;
- $L$  : niveau de tension bas équivalent à 0 logique.

### 4.7.1. Circuits logiques TTL

#### Présentation

TTL (Transistor-Transistor Logic) a été inventé dans les années 1960, est une famille de circuits logiques, réalisés en utilisant la technologie des transistors bipolaires, cette famille de circuit tend à disparaître en raison de la forte consommation d'énergie par rapport aux circuits CMOS.

- La famille TTL présente les avantages suivants :
  - L'entrée restante en 'air' a un état logique de 1 par défaut.
  - Une bonne immunité au bruit.
  - Faible temps de propagation.
- La famille TTL présente les inconvénients suivants :
  - L'alimentation doit être exactement de  $5V \pm 5\%$  sinon il y a un risque d'endommager le circuit.
  - Il consomme beaucoup de courant par rapport à la famille CMOS, car il est constitué de transistors bipolaires.

## Caractéristiques

Référence de boîtier	Caractéristiques de fonctionnement																
<ul style="list-style-type: none"> <li>• TTL standard (n'est plus utilisée) : 74 XX</li> <li>• TTL Low Power: 74 L XX</li> <li>• TTL Schottky (Rapide): 74 SXX</li> <li>• TTL Low Power Schottky: 74 LSXX</li> <li>• TTL Advanced Schottky: 74 ASXX TTL</li> <li>• Advanced Low Power Schottky: 74 ALSXX</li> </ul>	<p>Gamme d'alimentation : 5 V +/- 5%.</p> <ul style="list-style-type: none"> <li>• Gamme de température : de 0 °C à + 70 °C.</li> <li>• Puissance dissipée : environ 2 mW par porte (série LS).</li> </ul> <p>Fréquence de fonctionnement : jusqu'à 3 MHz.</p> <ul style="list-style-type: none"> <li>• Sortance : jusqu'à 20 (série LS). (Nombre d'entrées que l'on peut relier à une sortie de porte)</li> </ul>																
<p>Niveaux Logiques d'une porte logique TTL (LS) en entrée</p>	<p>Sortie à collecteur ouvert (Open collector)</p>																
	<p>On sort directement sur le collecteur du transistor de sortie. Obligation de connecter une résistance R de tirage au +5 V.</p> <p>La sortie est équivalente à un interrupteur.</p>																
<p>Niveaux Logiques d'une porte logique TTL (LS) en sortie</p>	<p>Sortie 3 états (3-state)</p>																
	<table border="1"> <thead> <tr> <th>EN</th> <th>T1</th> <th>T2</th> <th>Etat</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Passant</td> <td>Bloqué</td> <td>Haut</td> </tr> <tr> <td>1</td> <td>Bloqué</td> <td>Passant</td> <td>Bas</td> </tr> <tr> <td>0</td> <td>Bloqué</td> <td>Bloqué</td> <td>Haute impédance (Sortie 'en l'air').</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>• Dans une porte classique, l'un des 2 transistors T1 ou T2 du totem pôle est conducteur.</li> <li>• Dans une porte 3 états, il est possible de bloquer simultanément les 2 transistors T1 et T2 par l'entrée de validation EN (EN = 0).</li> </ul>	EN	T1	T2	Etat	1	Passant	Bloqué	Haut	1	Bloqué	Passant	Bas	0	Bloqué	Bloqué	Haute impédance (Sortie 'en l'air').
EN	T1	T2	Etat														
1	Passant	Bloqué	Haut														
1	Bloqué	Passant	Bas														
0	Bloqué	Bloqué	Haute impédance (Sortie 'en l'air').														

**Exemple : circuit TTL nommé SN74LS00.**

SN : signifie que le constructeur est Texas Instruments.

74 : fait référence aux circuits intégrés grands publics qui résistent à des températures ambiantes de 0 à 70 degrés.

LS ou HCT : indiquent la sous famille du circuit TTL.

00 : les derniers chiffres représentant la fonction logique réalisée par le composant (Porte NAND=00, Porte NOR=02, Porte AND=08 .... etc.).

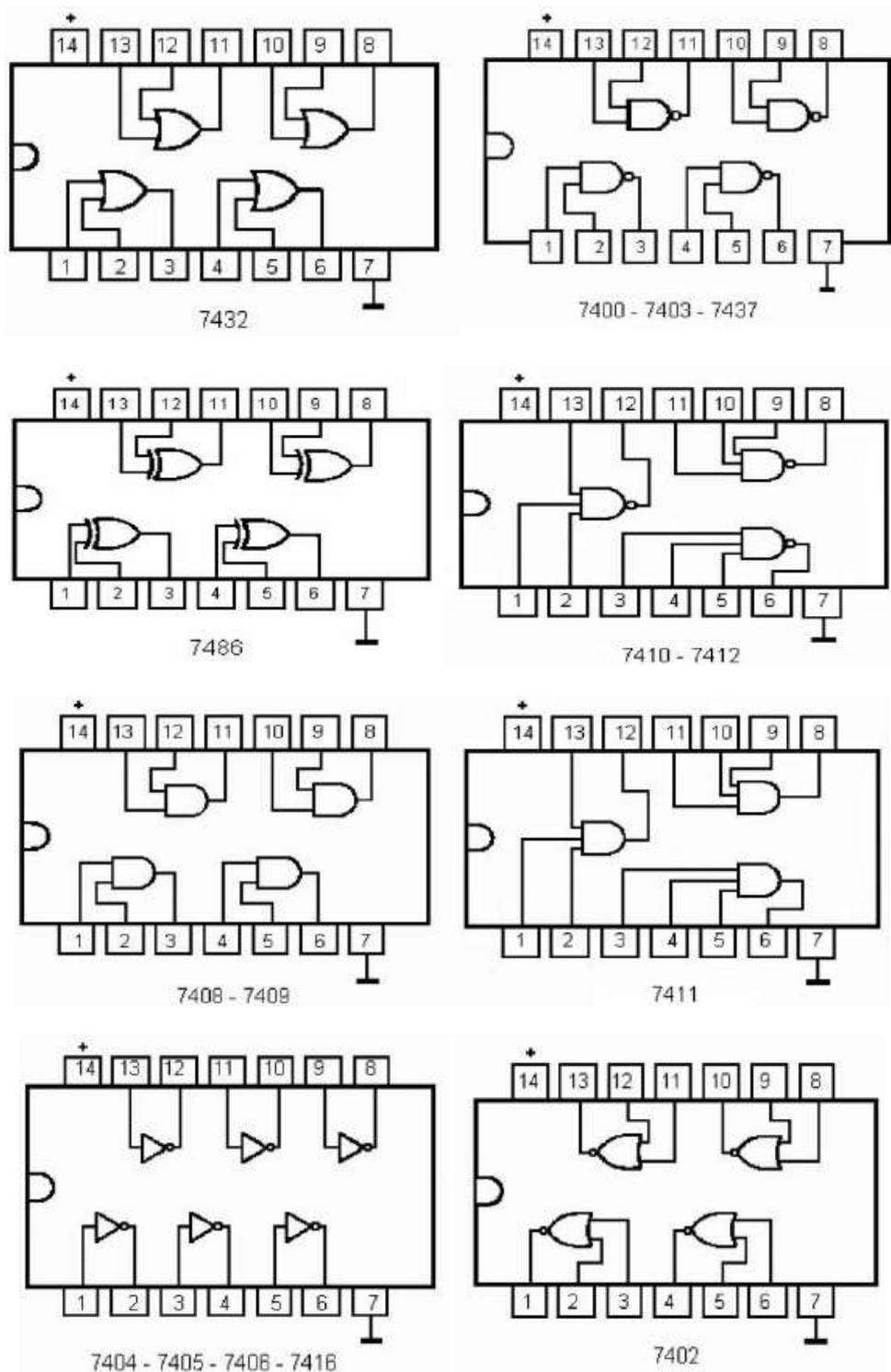


Figure 4.10. Exemples de circuits intégrés TTL.

## 4.7.2. Circuits logiques CMOS

### Présentation

Cette famille CMOS (Complementary Metal Oxide Semi-conductor) est constituée de transistors à effet de champ, comme pour les transistors classiques de type NPN et PNP, les transistors MOS sont disponibles en canal N et en canal P. Les circuits CMOS intègrent à la fois des canaux N et des canaux P [22]. Le premier dispositif MOS est apparu en 1960 et les premiers circuits CMOS sur substrat SOI ont été introduits sur le marché en 1999 [23].

- Cette famille CMOS présente les avantages suivants :
  - L'alimentation peut aller de 3V à 18V.
  - Le courant d'entrée est nul, car elle est réalisée avec des transistors à effet de champs qui sont commandés en tension.
  - Excellente immunité contre le bruit.
- Cette famille CMOS présente les inconvénients suivants :
  - La vitesse de commutation est inférieure à celle de la technologie TTL.

### Caractéristiques

Référence de boîtier	Caractéristiques de fonctionnement
<ul style="list-style-type: none"> <li>✓ Série 4000</li> <li>• 40 00 B (sorties bufférisées : amplifiées)</li> <li>• 40 00 UB (sorties non-bufférisées)</li> <li>✓ Série 74 :</li> <li>• 74 C 00 (identique à la série 4000)</li> <li>• 74 HC 00 (High-speed CMOS : CMOS rapides)</li> </ul>	<ul style="list-style-type: none"> <li>• Gamme d'alimentation : de 3 V à 15 V. Gamme de température : de - 40 °C à + 85 °C. Puissance dissipée : environ 10 nW par porte.</li> <li>• Fréquence de fonctionnement : jusqu'à 12 MHz.</li> <li>• Sortance : jusqu'à 50 (série 4000B).(Nombre d'entrées que l'on peut relier à une sortie de porte)</li> <li>• Excellente immunité aux bruits.</li> </ul>
Schéma des étages d'entrée et de sortie	Modélisation de l'entrée d'une porte logique CMOS
<p>Schéma des étages d'entrée et de sortie connectés</p>	

Exemple : circuit CMOS nommé CD4011BE.

CD : préfixe signifie que le constructeur est Texas Instruments.

4011 : indique le numéro du circuit. Il s'agit ici d'un quadruple porte NAND (NON-ET) à deux entrées chacune.

B : indique que la tension maximale est de 18V.

E : indique que le circuit est encapsulé dans un boîtier DIP.

Codes circuit = préfixe fabricant + numéro du circuit + suffixe + code boîtier

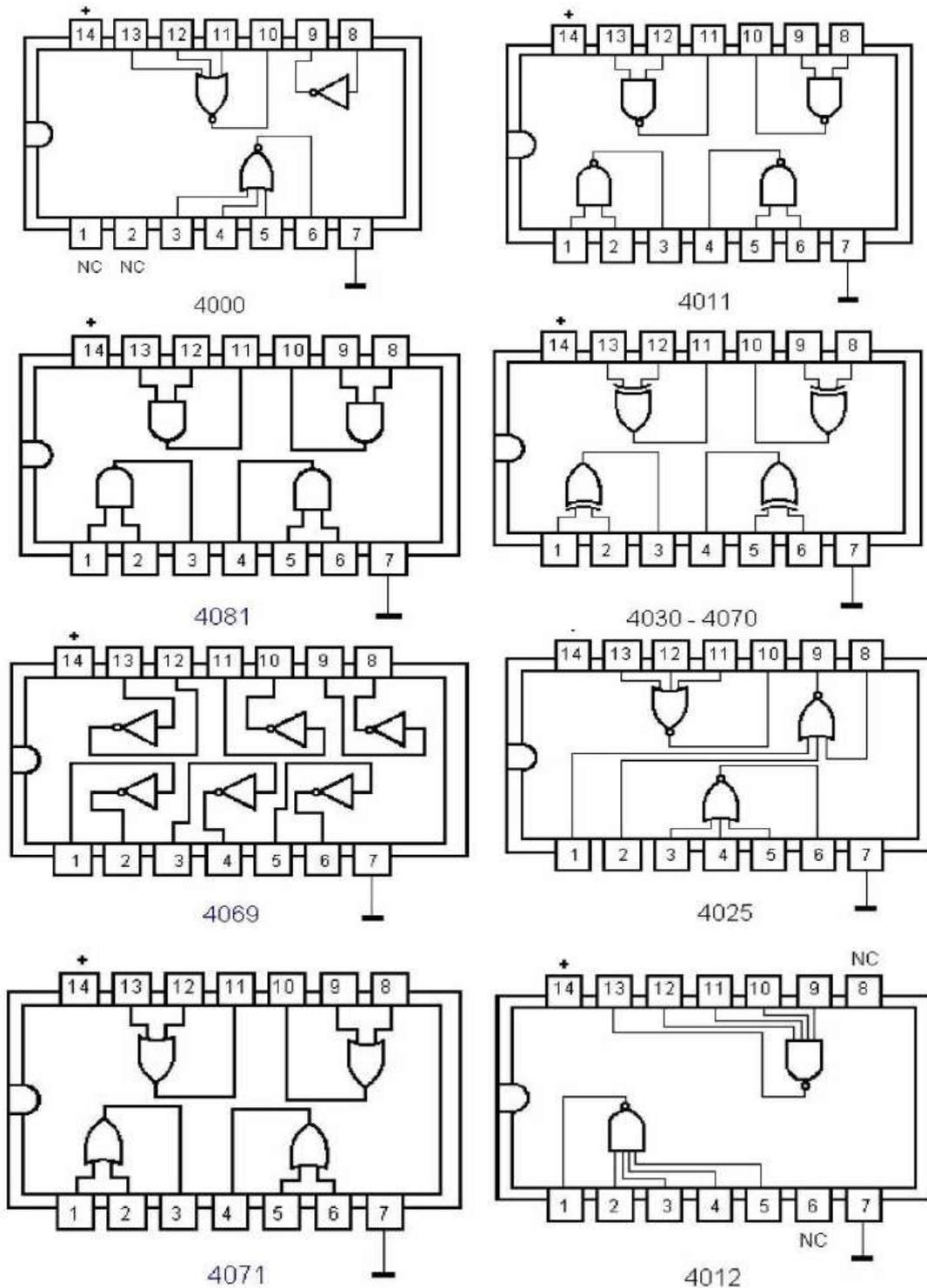


Figure 4.11. Exemples de circuits intégrés CMOS.

## Série d'exercices N°3(Circuits Intégrés)

### Exercice N°1 :

1. Expliquer la référence d'un : SN 74LS00 et CD4011.
2. Définir les tensions et courants suivants :  $V_{IL}$ ,  $V_{IH}$ ,  $V_{OL}$ ,  $V_{OH}$ ,  $I_{IL}$ ,  $I_{IH}$ ,  $I_{OL}$ ,  $I_{OH}$ . Ainsi qu'un ordre de grandeur et leurs signes pour la TTL standard.
3. Expliquer comment est défini un temps de montée, un temps de descente.
4. Définir : L'immunité aux bruits, L'entrance, La sortance, La charge TTL.
5. Donner la valeur de la sortance en TTL standard.

### Exercice N°2 :

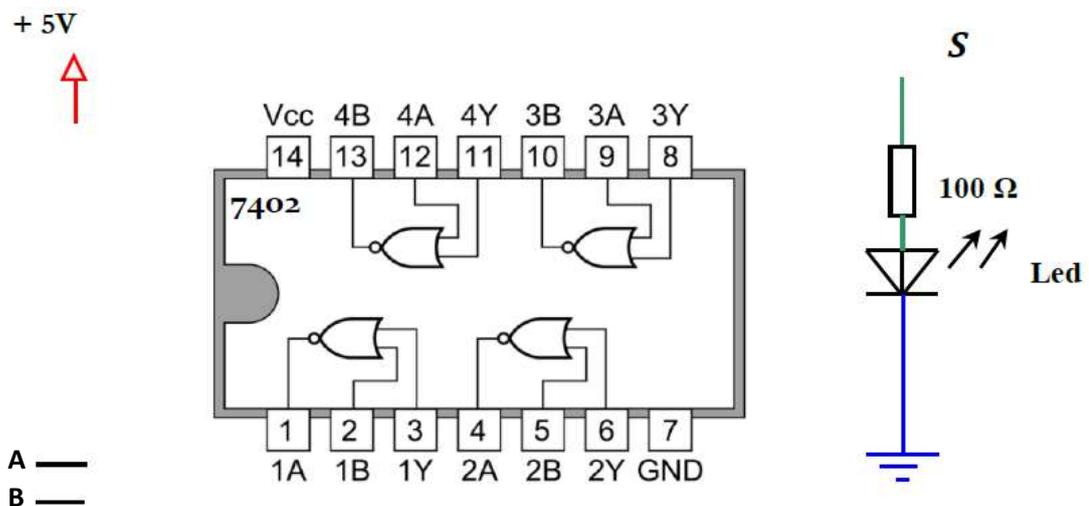
Soit la fonction :  $F(A, B, C) = \overline{\overline{ABC}}$

Tracer le logigramme de F et représenter le montage de la fonction à base du circuit 74LS00.

### Exercice N°3 :

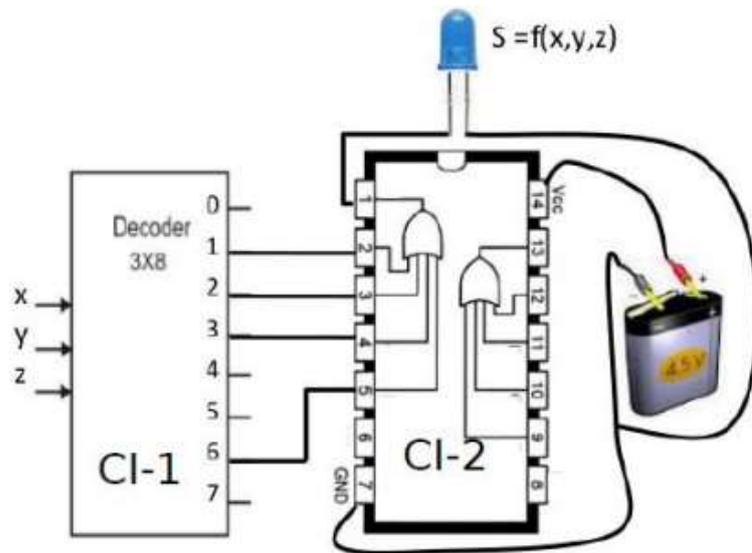
Soit la fonction Non-ou-exclusif (XNOR) de deux variables A et B.

1. Transformer l'expression de cette fonction en utilisant un minimum de portes NOR.
2. Compléter le câblage du montage suivant afin de réaliser cette fonction XNOR.



### Exercice N°4:

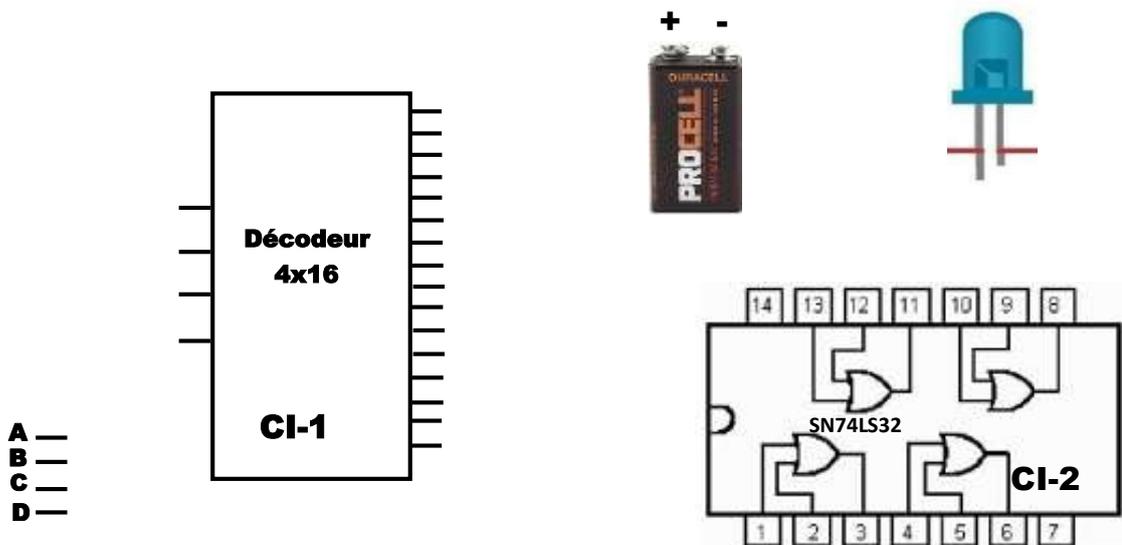
1. A quoi correspond la broche numéro 7 du circuit intégré CI-2 ?
2. A quoi correspond la broche numéro 14 du circuit intégré CI-2 ?
3. Indiquez l'expression algébrique de  $S=f(x,y,z)$  réalisée dans le montage suivant :



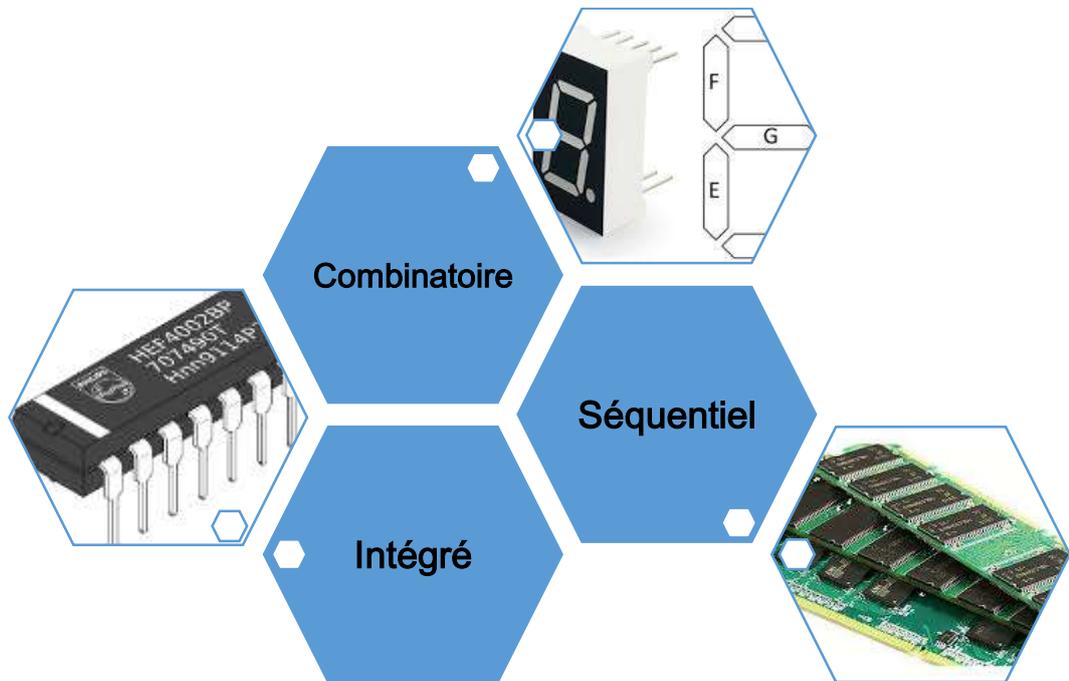
**Exercice 5 :**

Soit la fonction :  $F(A, B, C, D) = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D + ABCD$

1. Expliquez la référence suivante : SN74LS32.
2. Compléter le câblage du montage suivant afin de réaliser cette fonction  $F(A, B, C, D)$ .



# Solution des séries d'exercices



## Solution des séries d'exercices

### 1. Solution de la série d'exercices N°1 (Circuits combinatoires)

Exercice N°1 :

⇒ 1-demi-additionneur

a-Schéma symbolique



En système binaire, l'addition sur 1 bit se fait de la façon suivante :

$$\left\{ \begin{array}{l} 0+0=0 \text{ retenue}=0 \\ 0+1=1 \text{ retenue}=0 \\ 1+0=1 \text{ retenue}=0 \\ 1+1=0 \text{ retenue}=1 \end{array} \right.$$

b-Table de vérité

$A_i$	$B_i$	$S_i$	$R_i$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

c- Equations de **sortie**

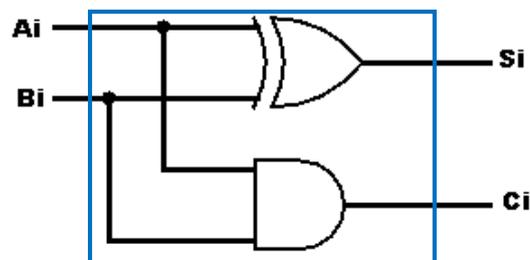
De la table de vérité on trouve :

$$S_i = \bar{A}_i B_i + A_i \bar{B}_i$$

$$= A_i \oplus B_i$$

$$R_i = A_i B_i$$

d-Schéma logique



## ⇒ 2- Additionneur complet

### a-Schéma symbolique



### b-Table de vérité

A <sub>i</sub>	B <sub>i</sub>	R <sub>i-1</sub>	S <sub>i</sub>	R <sub>i</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

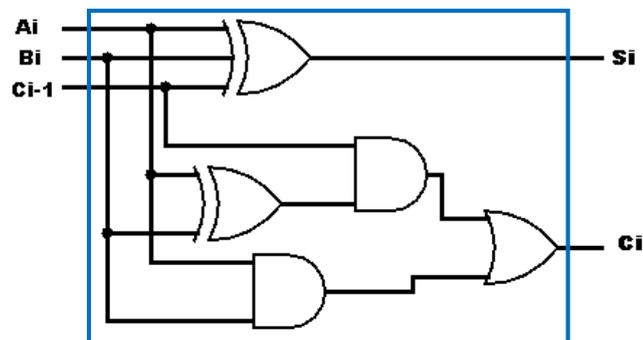
### c- Equations de sortie

De la table de vérité on trouve :

$$\begin{aligned}
 S_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= \bar{A}_i (\bar{B}_i R_{i-1} + B_i \bar{R}_{i-1}) + A_i (\bar{B}_i \bar{R}_{i-1} + B_i R_{i-1}) \\
 &= \bar{A}_i (B_i \oplus R_{i-1}) + A_i (\overline{B_i \oplus R_{i-1}}) \\
 &= A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

$$\begin{aligned}
 R_i &= \bar{A}_i B_i R_{i-1} + A_i \bar{B}_i R_{i-1} + A_i B_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= (\bar{A}_i B_i + A_i \bar{B}_i) R_{i-1} + A_i B_i (\bar{R}_{i-1} + R_{i-1}) \\
 &= (A_i \oplus B_i) R_{i-1} + A_i B_i
 \end{aligned}$$

### d-Schéma logique :



⇒ 3- pour l'opération de soustraction

⇒ 3.1. Demi-soustracteur

a-Schéma symbolique



En système binaire, la soustraction sur 1 bit se fait de la façon suivante :

$$\begin{cases} 0 - 0 = 0 & \text{retenue} = 0 \\ 0 - 1 = 1 & \text{retenue} = 1 \\ 1 - 0 = 1 & \text{retenue} = 0 \\ 1 - 1 = 0 & \text{retenue} = 0 \end{cases}$$

b-Table de vérité

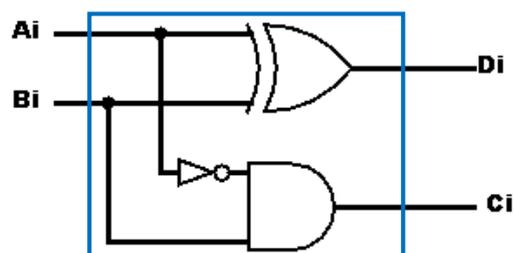
A <sub>i</sub>	B <sub>i</sub>	D <sub>i</sub>	R <sub>i</sub>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

c- Equations de sortie

$$D_i = \bar{A}_i B_i + A_i \bar{B}_i = A_i \oplus B_i$$

$$R_i = \bar{A}_i B_i$$

d-schéma logique



⇒ 3.2- Soustracteur complet

a-Schéma symbolique :



b-Table de vérité :

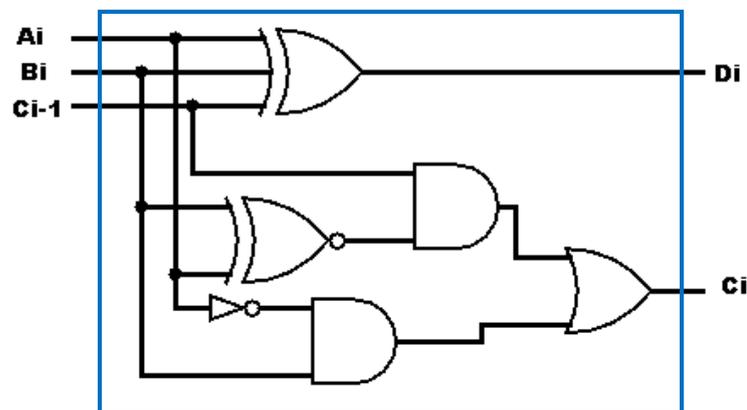
A <sub>i</sub>	B <sub>i</sub>	R <sub>i-1</sub>	D <sub>i</sub>	R <sub>i</sub>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

c- Equations de sortie :

$$\begin{aligned}
 D_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + A_i \bar{B}_i \bar{R}_{i-1} + A_i B_i R_{i-1} \\
 &= \bar{A}_i (\bar{B}_i R_{i-1} + B_i \bar{R}_{i-1}) + A_i (\bar{B}_i \bar{R}_{i-1} + B_i R_{i-1}) \\
 &= \bar{A}_i (B_i \oplus R_{i-1}) + A_i (\overline{B_i \oplus R_{i-1}}) \\
 &= A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

$$\begin{aligned}
 R_i &= \bar{A}_i \bar{B}_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} + \bar{A}_i B_i R_{i-1} + A_i B_i R_{i-1} \\
 &= (\bar{A}_i \bar{B}_i + A_i B_i) R_{i-1} + \bar{A}_i B_i (\bar{R}_{i-1} + R_{i-1}) \\
 &= (\overline{A_i \oplus B_i}) R_{i-1} + \bar{A}_i B_i
 \end{aligned}$$

d-Schéma logique :



⇒ 4- circuit Additionneur-soustracteur complet

a-Schéma symbolique



b-Table de vérité :

V	A <sub>i</sub>	B <sub>i</sub>	R <sub>i-1</sub>	S <sub>i</sub>	R <sub>i</sub>
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

c- Equations de sortie :

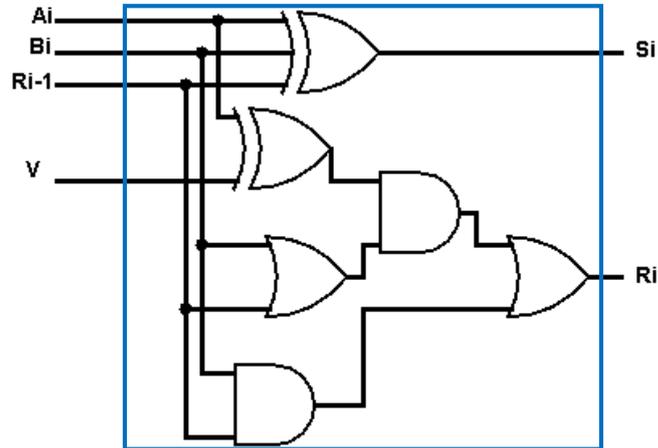
V A <sub>i</sub> \ B <sub>i</sub> R <sub>i-1</sub>	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	0	1	1	0
10	1	0	0	1

$$\begin{aligned}
 S_i(V, A_i, B_i, R_{i-1}) &= A_i \bar{B}_i \bar{R}_{i-1} + \bar{A}_i \bar{B}_i R_{i-1} + A_i B_i R_{i-1} + \bar{A}_i B_i \bar{R}_{i-1} \\
 &= (\bar{A}_i \bar{B}_i + A_i B_i) R_{i-1} + (A_i \bar{B}_i + \bar{A}_i B_i) \bar{R}_{i-1} \\
 &= (\overline{A_i \oplus B_i}) R_{i-1} + (A_i \oplus B_i) \bar{R}_{i-1} \\
 &= A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

V A <sub>i</sub> \ B <sub>i</sub> R <sub>i-1</sub>	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	1	1	1	1
10	0	1	0	1

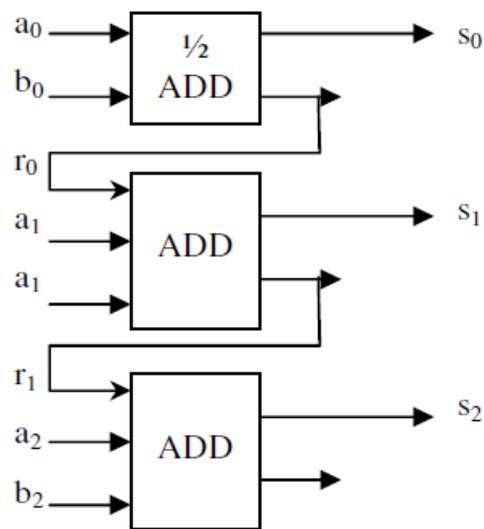
$$\begin{aligned}
 R_i(V, A_i, B_i, R_{i-1}) &= B_i R_{i-1} + \bar{V} A_i R_{i-1} + \bar{V} A_i B_i + V \bar{A}_i R_{i-1} + V \bar{A}_i B_i \\
 &= B_i R_{i-1} + (\bar{V} A_i + \bar{V} A_i) B_i + (V \bar{A}_i + V \bar{A}_i) R_{i-1} \\
 &= B_i R_{i-1} + (V \oplus A_i) B_i + (V \oplus A_i) R_{i-1} \\
 &= B_i R_{i-1} + (V \oplus A_i) (B_i + R_{i-1})
 \end{aligned}$$

d-Schéma logique :



Exercice N°2 :

Schéma logique d'un additionneur à 3 bits :



Exercice N°3 :

a-Schéma symbolique



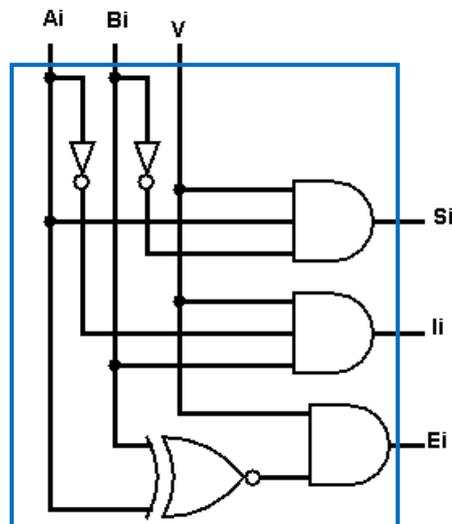
b-Table de vérité

V	A <sub>i</sub>	B <sub>i</sub>	E <sub>i</sub>	S <sub>i</sub>	I <sub>i</sub>
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	1	0	0

c- expressions logique des sorties :

$$\begin{aligned}
 E_i &= \overline{V} \overline{A_i} \overline{B_i} + V A_i B_i \\
 &= V (\overline{A_i} \overline{B_i} + A_i B_i) \\
 &= V (\overline{A_i \oplus B_i}) \\
 S_i &= V A_i \overline{B_i} \\
 I_i &= V \overline{A_i} B_i
 \end{aligned}$$

d-Schéma logique du comparateur :



Exercice N°4 :

a/ Les expressions logiques des sorties E, S et I :



- Si la validation  $V=0$  est égale à 0 on aura  $\Rightarrow E = S = I = 0$
- Si la validation  $V=1$ , on aura le fonctionnement suivant
  - $A=B$  si  $(A_2=B_2)$  et  $(A_1=B_1)$  et  $(A_0=B_0)$ 

$$\Rightarrow E = E_2 E_1 E_0$$

$$\Rightarrow E = (\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1})(\overline{A_0 \oplus B_0})$$
  - $A>B$  si  $(A_2>B_2)$  ou  $((A_2=B_2)$  et  $(A_1>B_1))$  ou  $((A_2=B_2)$  et  $(A_1=B_1)$  et  $(A_0>B_0))$ 

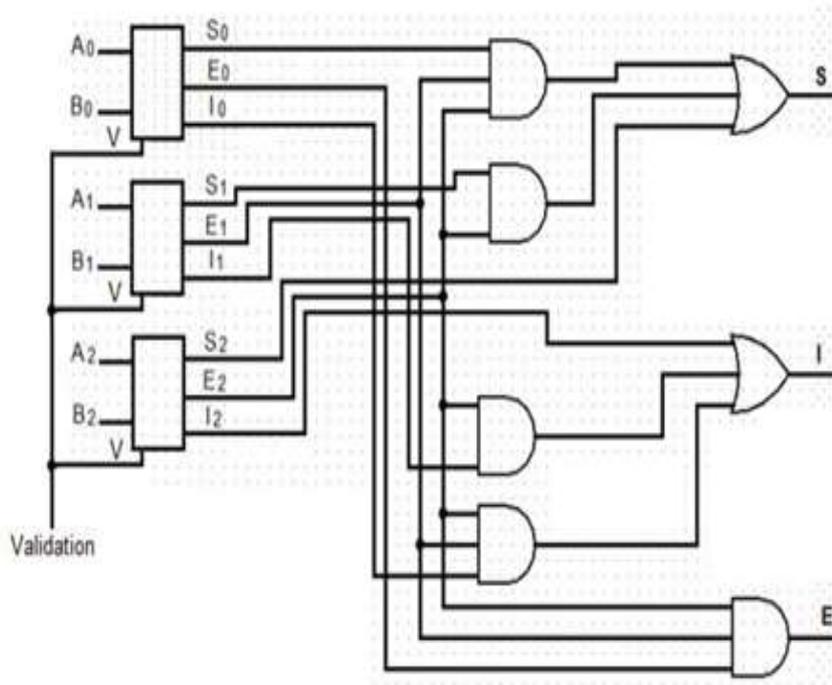
$$\Rightarrow S = S_2 + E_2 S_1 + E_2 E_1 S_0$$

$$\Rightarrow S = A_2 \overline{B_2} + (\overline{A_2 \oplus B_2}) A_1 \overline{B_1} + (\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1}) A_0 \overline{B_0}$$
  - $A<B$  si  $(A_2<B_2)$  ou  $((A_2=B_2)$  et  $(A_1<B_1))$  ou  $((A_2=B_2)$  et  $(A_1=B_1)$  et  $(A_0<B_0))$ 

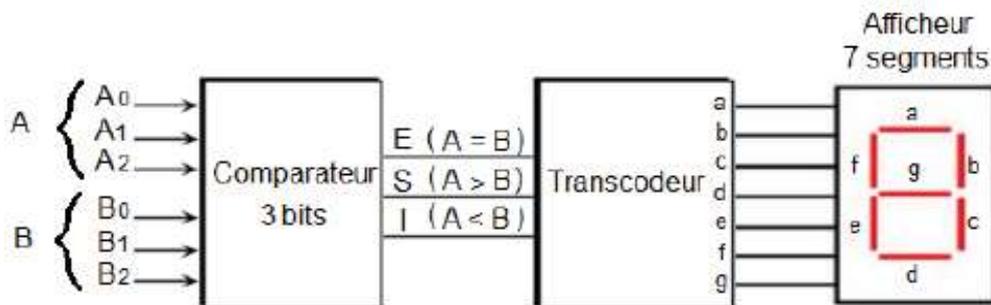
$$\Rightarrow I = I_2 + E_2 I_1 + E_2 E_1 I_0$$

$$\Rightarrow I = \overline{A_2} B_2 + (\overline{A_2 \oplus B_2}) \overline{A_1} B_1 + (\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1}) \overline{A_0} B_0$$

b/ schéma interne du comparateur à 3 bits



Exercice N°5 :

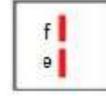




Si  $A > B$



Si  $A = B$



Si  $A < B$

a/ Table de vérité du transcodeur

Entrées			Sorties						
E	S	I	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	1	0
0	1	0	1	0	1	1	0	1	1
0	1	1	X	X	X	X	X	X	X
1	0	0	1	0	0	1	1	1	1
1	0	1	X	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X

b/ expressions simplifiées des sorties en utilisant le tableau de Karnaugh

	SI	00	01	11	10
E					
0		0	0	X	1
1		1	X	X	X

$$a = E + S$$

	SI	00	01	11	10
E					
0		0	0	X	0
1		0	X	X	X

$$b = 0$$

	SI	00	01	11	10
E					
0		0	0	X	1
1		0	X	X	X

$$c = S$$

	SI	00	01	11	10
E					
0		0	0	X	1
1		1	X	X	X

$$d = E + S = a$$

	SI	00	01	11	10
E					
0		0	1	X	0
1		1	X	X	X

$$e = E + I$$

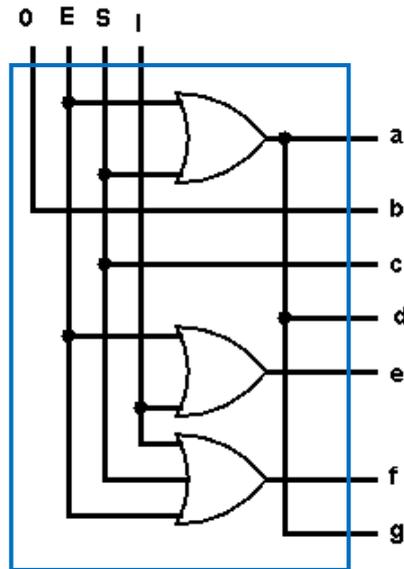
	SI	00	01	11	10
E					
0		0	1	X	1
1		1	X	X	X

$$f = E + S + I$$

$$g = E + S = a$$

	SI	00	01	11	10
E					
0		0	0	X	1
1		1	X	X	X

c/ schéma interne du transcodeur



Exercice N°6 :

a/ Table de vérité

X		Y		Z				X*Y=Z
X <sub>1</sub>	X <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
0	0	0	0	0	0	0	0	0*0=0
0	0	0	1	0	0	0	0	0*1=0
0	0	1	0	0	0	0	0	0*2=0
0	0	1	1	0	0	0	0	0*3=0
0	1	0	0	0	0	0	0	1*0=0
0	1	0	1	0	0	0	1	1*1=1
0	1	1	0	0	0	1	0	1*2=2
0	1	1	1	0	0	1	1	1*3=3
1	0	0	0	0	0	0	0	2*0=0
1	0	0	1	0	0	1	0	2*1=2
1	0	1	0	0	1	0	0	2*2=4
1	0	1	1	0	1	1	0	2*3=6
1	1	0	0	0	0	0	0	3*0=0
1	1	0	1	0	0	1	1	3*1=3
1	1	1	0	0	1	1	0	3*2=6
1	1	1	1	1	0	0	1	3*3=9

b/ Expressions simplifiées des sorties

$X_1X_0$	00	01	11	10
$Y_1Y_0$				
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

$X_1X_0$	00	01	11	10
$Y_1Y_0$				
00	0	0	0	0
01	0	0	0	0
11	0	0	0	1
10	0	0	1	1

$$Z_3 (X_1, X_0, Y_1, Y_0) = X_1 X_0 Y_1 Y_0$$

$X_1X_0$	00	01	11	10
$Y_1Y_0$				
00	0	0	0	0
01	0	0	1	1
11	0	1	0	1
10	0	1	1	0

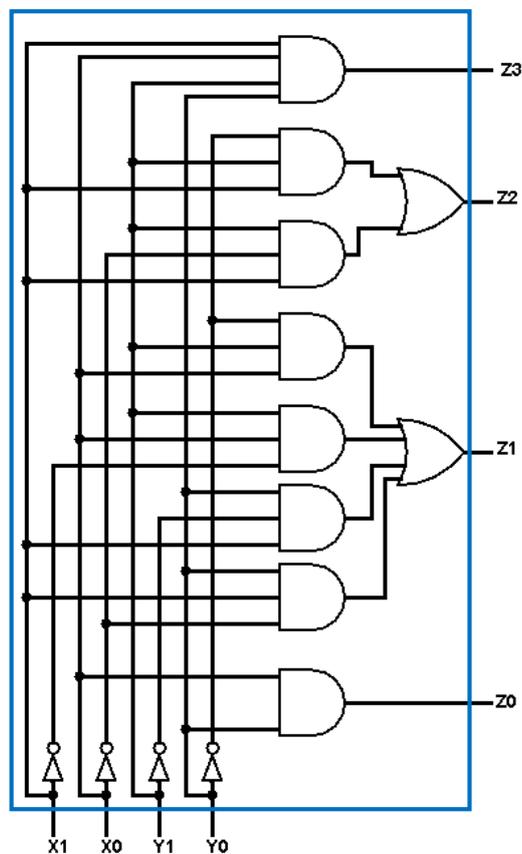
$$Z_2 (X_1, X_0, Y_1, Y_0) = X_1 \bar{X}_0 Y_1 + X_1 Y_1 \bar{Y}_0$$

$X_1X_0$	00	01	11	10
$Y_1Y_0$				
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$$Z_1 (X_1, X_0, Y_1, Y_0) = X_1 \bar{X}_0 Y_0 + X_1 \bar{Y}_1 Y_0 + \bar{X}_1 X_0 Y_1 + X_0 Y_1 \bar{Y}_0$$

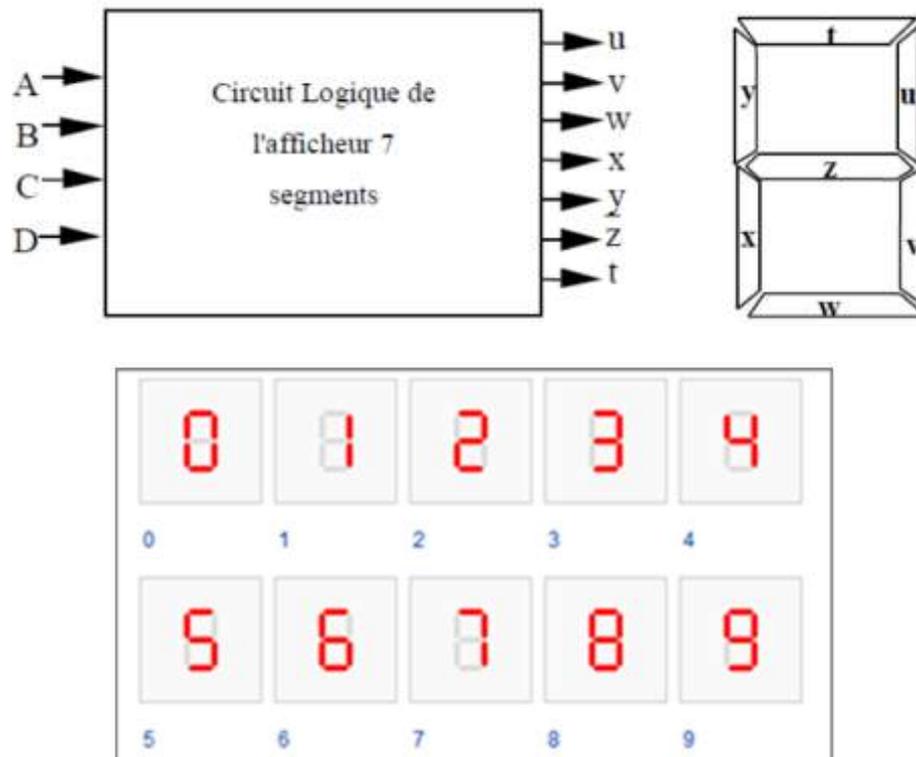
$$Z_0 (X_1, X_0, Y_1, Y_0) = X_0 Y_0$$

c/ Schéma logique d'un multiplieur à deux bits



## Exercice N°7 :

L'afficheur 7 segments



### 1. Table de vérité

A	B	C	D	u	v	w	x	y	z	t
0	0	0	0	1	1	1	1	1	0	1
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	1	0
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

2. Simplification graphique (Karnaugh) les expressions des trois sorties u, v et t.

**Segment t**

AB \ CD	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	1	X	X

$$t(A, B, C, D) = A + BD + C + \bar{B}\bar{D}$$

**Segment v**

AB \ CD	00	01	11	10
00	1	1	X	1
01	1	1	X	1
11	1	1	X	X
10	0	1	X	X

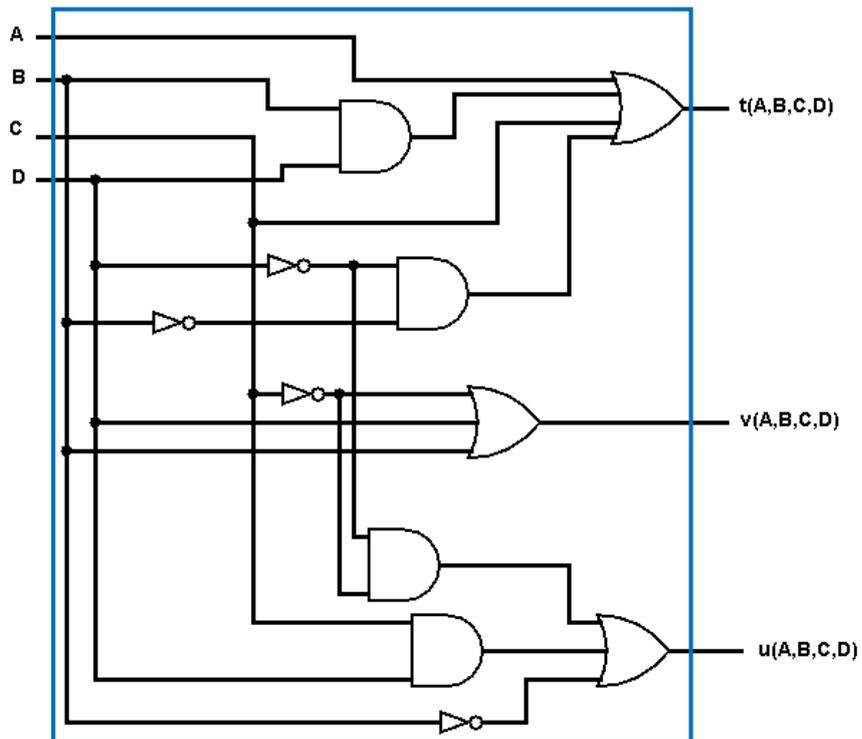
$$v(A, B, C, D) = B + \bar{C} + D$$

**Segment u**

AB \ CD	00	01	11	10
00	1	1	X	1
01	1	0	X	1
11	1	1	X	X
10	1	0	X	X

$$u(A, B, C, D) = \bar{C}\bar{D} + \bar{B} + CD$$

3. logigramme de ces trois sorties u, v et t de l'afficheur 7 segments.



**Exercice N°8 :**

- L'équation de premier multiplexeur de sortie  $S_1$

$$\begin{aligned} S_1 &= \overline{A}\overline{B}E_0 + \overline{A}BE_1 + A\overline{B}E_2 + ABE_3 \\ &= \overline{A}\overline{B}.1 + \overline{A}B.0 + A\overline{B}.0 + AB.1 \\ &= \overline{A}\overline{B} + AB \\ &= \overline{A \oplus B} \end{aligned}$$

- L'équation de deuxième multiplexeur de sortie  $S_2$

$$\begin{aligned} S_2 &= \overline{C}\overline{D}E_0 + \overline{C}DE_1 + C\overline{D}E_2 + CDE_3 \\ &= \overline{C}\overline{D}.1 + \overline{C}D.1 + C\overline{D}.0 + CD.1 \\ &= \overline{C}\overline{D} + \overline{C}D + CD \\ &= \overline{C}(\overline{D} + D) + CD \\ &= \overline{C} + CD \\ &= (\overline{C} + C)(\overline{C} + D) \\ &= \overline{C} + D \end{aligned}$$

- L'équation de troisième multiplexeur de sortie  $S$

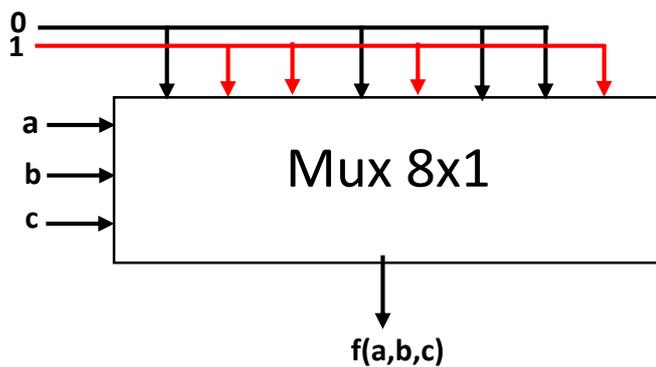
$$\begin{aligned} S &= \overline{S}_2\overline{S}_1E_0 + \overline{S}_2S_1E_1 + S_2\overline{S}_1E_2 + S_2S_1E_3 \\ &= \overline{S}_2\overline{S}_1.1 + \overline{S}_2S_1.E + S_2\overline{S}_1.0 + S_2S_1.\overline{E} \\ &= \overline{S}_2\overline{S}_1.E + S_2S_1.\overline{E} \\ &= \overline{S}_1(\overline{S}_2E + S_2\overline{E}) \\ &= \overline{S}_1(S_2 \oplus E) \\ &= \overline{A \oplus B}((\overline{C} + D) \oplus E) \end{aligned}$$

**Exercice N°9 :**

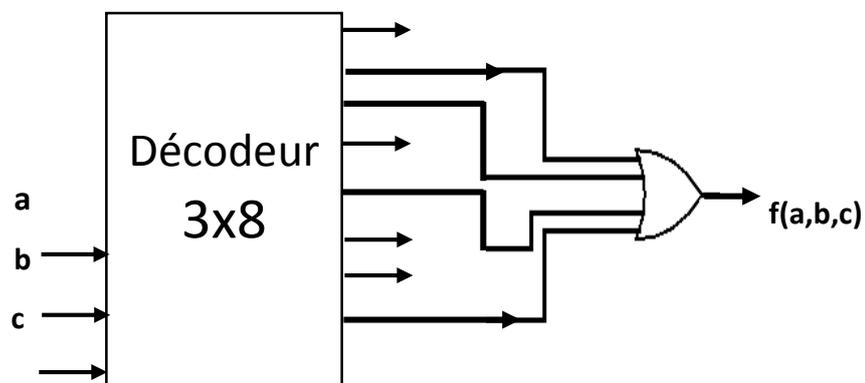
1. Table de vérité

A	b	c	f(a,b,c)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

2.1. Réalisation de la fonction à l'aide d'un multiplexeur



2.2. Réalisation de la fonction à l'aide d'un décodeur



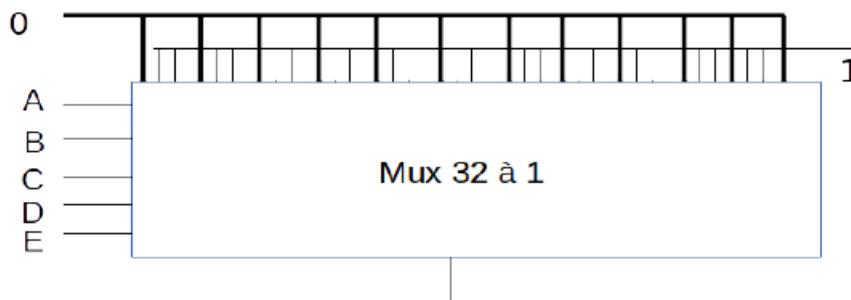
Exercice N°10 :

Table de vérité :

A	B	C	D	E	F
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0

1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	1

Schéma logique à l'aide d'un multiplexeur :



**Exercice N°11 :**

- L'analyse du circuit N°1 :

➤ La fonction du circuit :

$$S(A, B, C, D) = \bar{A}\bar{B}(C + D) + \bar{A}BD + A\bar{B}C + AB(C + D)$$

➤ La formule simplifiée de la fonction :

$$\begin{aligned}
 S(A, B, C, D) &= \bar{A}\bar{B}(C + D) + \bar{A}BD + A\bar{B}C + AB(C + D) \\
 &= \bar{A}\bar{B}C + \bar{A}\bar{B}D + \bar{A}BD + A\bar{B}C + ABC + ABD \\
 &= \bar{B}C(\bar{A} + A) + \bar{A}\bar{B}D + BD(\bar{A} + A) + ABC \\
 &= \bar{B}C + \bar{A}\bar{B}D + BD + ABC \\
 &= C(\bar{B} + AB) + D(\bar{A}\bar{B} + B) \\
 &= C((\bar{B} + A)(\bar{B} + B)) + D((\bar{A} + B)(\bar{B} + B)) \\
 &= C((\bar{B} + A)) + D((\bar{A} + B)) \\
 &= C((\bar{B} + A)(\bar{B} + B)) + D((\bar{A} + B)(\bar{B} + B)) \\
 &= \bar{B}C + AC + \bar{A}D + BD
 \end{aligned}$$

• L'analyse du circuit N°2 :

➤ La fonction du circuit :

$$\begin{cases} F_1(A, B, C) = \overline{A}\overline{B}\overline{C} + ABC \\ F_2(A, B, C) = \overline{\overline{A}\overline{B}C + ABC} \\ F_3(A, B, C) = \overline{B} \\ F_4(A, B, C) = ABC + C \end{cases}$$

➤ La formule simplifiée de la fonction :

$$\begin{cases} F_1(A, B, C) = \overline{A}\overline{B}\overline{C} + ABC \\ F_2(A, B, C) = \overline{\overline{A}\overline{B}C + ABC} = \overline{C(\overline{A}\overline{B} + AB)} = \overline{C(A \oplus B)} \\ F_3(A, B, C) = \overline{B} \\ F_4(A, B, C) = C(AB + 1) = C \end{cases}$$

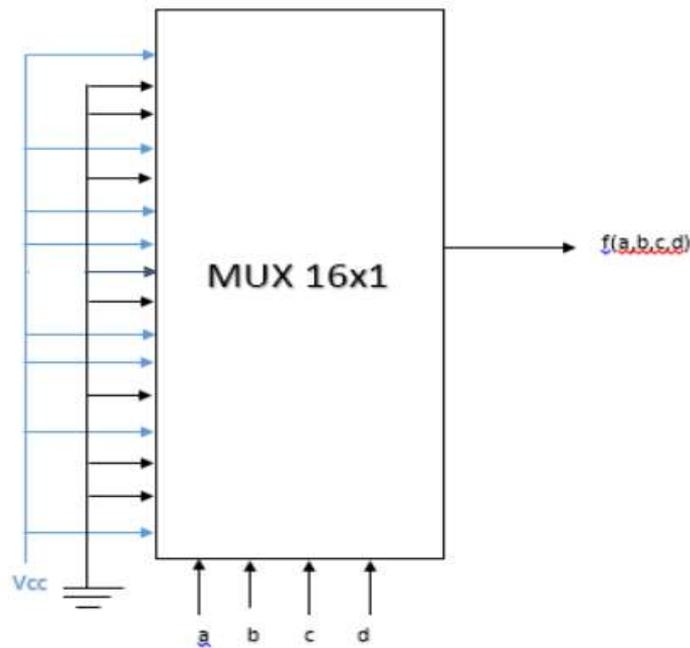
**Exercice N°12 :**

Table de vérité :

a	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

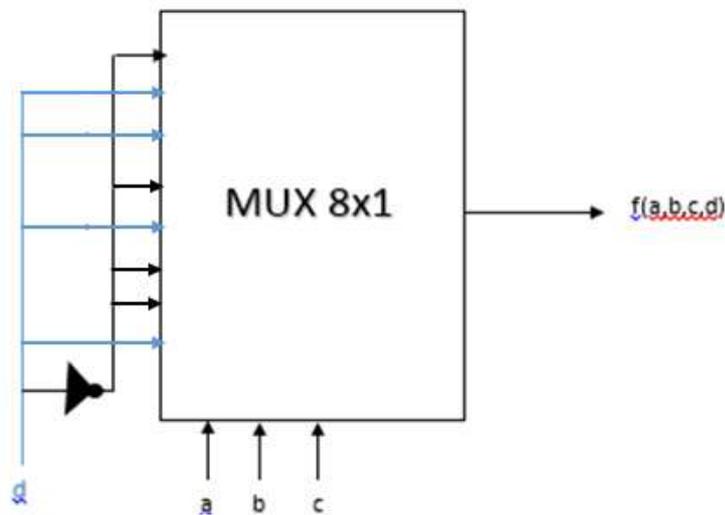
$$\begin{aligned} f(a, b, c, d) &= \sum(0, 3, 5, 6, 9, 10, 12, 15) \\ &= \overline{a}\overline{b}\overline{c}\overline{d} + \overline{a}\overline{b}c\overline{d} + \overline{a}b\overline{c}\overline{d} + \overline{a}b\overline{c}d + \overline{a}b\overline{c}\overline{d} + \overline{a}b\overline{c}d + \overline{a}b\overline{c}\overline{d} + \overline{a}b\overline{c}d \\ &= \overline{a}\overline{b}\overline{c}\overline{d}1 + \overline{a}\overline{b}\overline{c}d0 + \overline{a}\overline{b}c\overline{d}0 + \overline{a}\overline{b}cd1 + \overline{a}b\overline{c}\overline{d}0 + \overline{a}b\overline{c}d1 + \overline{a}b\overline{c}\overline{d}1 + \overline{a}bcd0 + \\ &\quad \overline{a}b\overline{c}\overline{d}0 + \overline{a}b\overline{c}d1 + \overline{a}b\overline{c}\overline{d}1 + \overline{a}b\overline{c}d0 + \overline{a}b\overline{c}\overline{d}1 + \overline{a}b\overline{c}d0 + \overline{a}bcd\overline{d}0 + \overline{a}bcd1 \end{aligned}$$

Réalisation de la fonction f à l'aide d'un multiplexeur 16x1



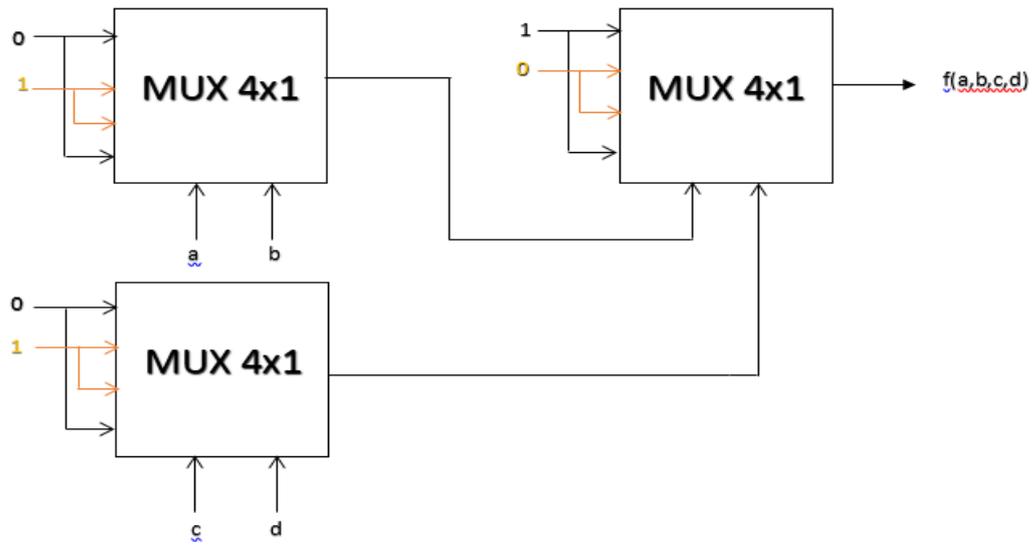
Réalisation de la fonction f à l'aide d'un multiplexeur 8x1 et minimum de portes :

$$\begin{aligned}
 f(a,b,c,d) &= \sum(0,3,5,6,9,10,12,15) \\
 &= \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + ab\bar{c}\bar{d} + abcd \\
 &= (\bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}b\bar{c} + \bar{a}b\bar{c})\bar{d} + (\bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}b\bar{c} + \bar{a}b\bar{c})d
 \end{aligned}$$



Réalisation de la fonction f à l'aide des multiplexeurs 4x1 et sans portes logiques :

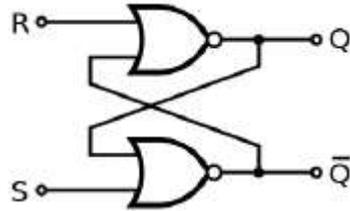
$$\begin{aligned}
 f(a,b,c,d) &= \sum(0,3,5,6,9,10,12,15) \\
 &= \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + ab\bar{c}\bar{d} + abcd \\
 &= \bar{a}\bar{b}(\bar{c}\bar{d} + cd) + ab(\bar{c}\bar{d} + cd) + \bar{a}b(\bar{c}\bar{d} + \bar{c}d) + \bar{a}b(\bar{c}\bar{d} + \bar{c}d) \\
 &= (\bar{a}\bar{b} + ab)(\bar{c}\bar{d} + cd) + (\bar{a}b + \bar{a}b)(\bar{c}\bar{d} + \bar{c}d) \\
 &= (\overline{a \oplus b})(\overline{c \oplus d}) + (a \oplus b)(c \oplus d) \\
 &= \overline{(a \oplus b) \oplus (c \oplus d)}
 \end{aligned}$$



## 2. Solution de la série d'exercices N°2 (Circuits séquentiels)

### Exercice N°1 :

1. Cette bascule n'est pas synchrone



2. Table de vérité

R	S	Q	Q <sup>+</sup>	Observation
0	0	0	0	mémorisation de l'information
0	0	1	1	
0	1	0	1	mise à 1 de la sortie
0	1	1	1	
1	0	0	0	0 mise à 0 de la sortie
1	0	1	0	
1	1	0	X	X état indéterminé
1	1	1	X	

### Exercice N°2 :

1. La table de vérité de la bascule RS

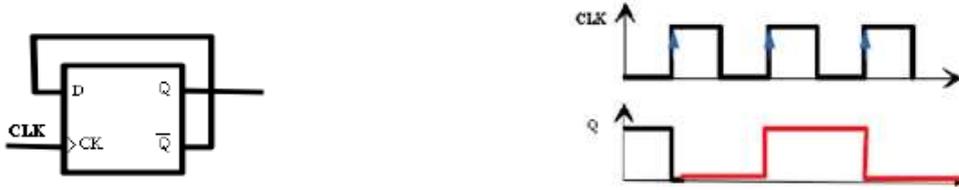
R	S	Q	Q <sup>+</sup>
0	0	0	0 mémorisation de l'information
0	0	1	1 mémorisation de l'information
0	1	0	1 mise à 1 de la sortie
0	1	1	1 mise à 1 de la sortie
1	0	0	0 mise à 0 de la sortie
1	0	1	0 mise à 0 de la sortie
1	1	0	X état indéterminé
1	1	1	X état indéterminé

2. Tableau

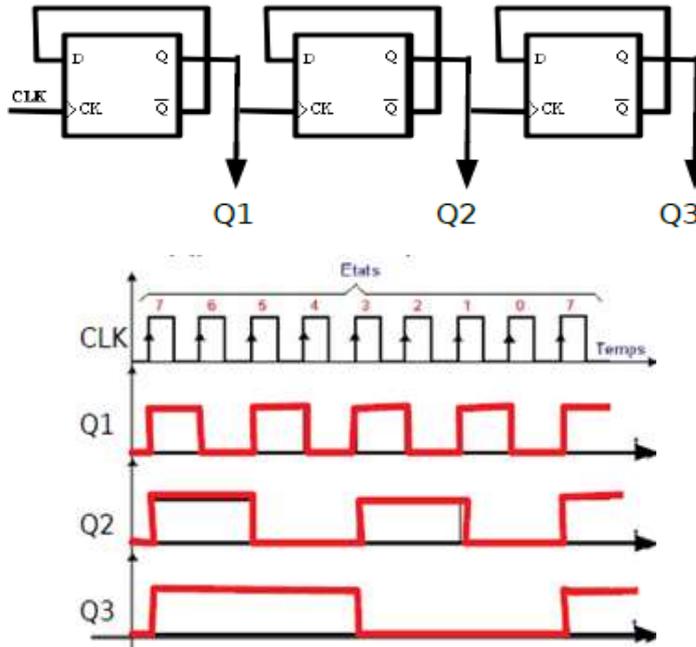
T	1	2	3	4	5	6	7	8	9	10
S	0	0	0	1	1	0	0	1	1	0
R	0	0	1	0	0	0	1	1	0	0
Q	0	0	0	1	1	1	0	X	1	1

**Exercice N°3 :**

**A. chronogramme de la bascule D (Q)**



**B. Les chronogrammes du circuit (Q<sub>1</sub>, Q<sub>2</sub> et Q<sub>3</sub>)**



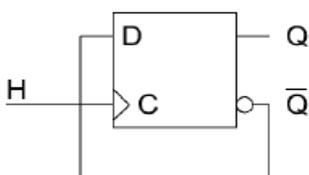
C. Ce circuit (compteur) n'est pas synchrone.

**Exercice N°4 :**

- Les tables de vérité des bascules JK et D sur front montant

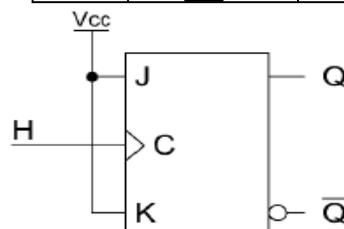
**Bascule JK**

H	J	K	$Q$	$\bar{Q}$
$\uparrow$	0	0	$q$	$\bar{q}$
$\uparrow$	0	1	0	1
$\uparrow$	1	0	1	0
$\uparrow$	1	1	$\bar{q}$	$q$

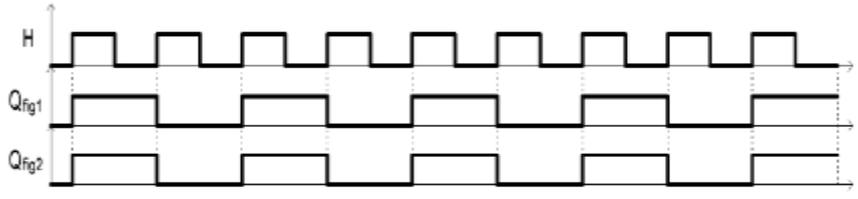


**Bascule D**

H	D	$Q$	$\bar{Q}$
$\uparrow$	0	0	1
$\uparrow$	1	1	0



- Les chronogramme des sorties de chacune des bascules



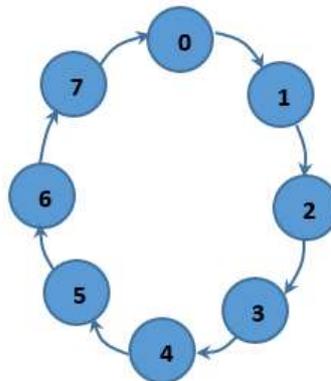
**Exercice N°5 :**

Réalisation d'un compteur synchrone modulo 8 à l'aide de bascules JK synchronisées sur front montant

1. La table des transitions d'une bascule JK.

$Q$	$Q^+$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

2. Diagramme d'état



3. Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

4. Les équations des entrées des bascules  $J_0, K_0, J_1, K_1, J_2$  et  $K_2$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	0	X	X
1	0	1	X	X

$$J_2 = Q_1 Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	X	X	0	0
1	X	X	1	0

$$K_2 = Q_1 Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	0	X	X	0
1	1	X	X	1

$$J_1 = Q_0$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	X	0	0	X
1	X	1	1	X

$$K_1 = Q_0$$

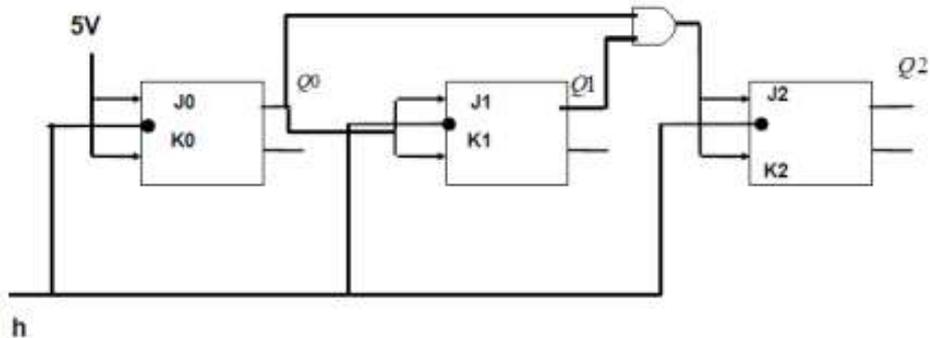
$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	1	1	1	1
1	X	X	X	X

$$J_0 = 1$$

$Q_2 Q_1$	00	01	11	10
$Q_0$				
0	X	X	X	X
1	1	1	1	1

$$K_0 = 1$$

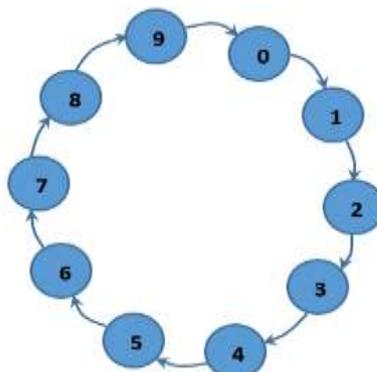
5. Le schéma de câblage de compteur modulo 8 synchrone



Exercice N°6 :

1. compteur synchrone modulo 10 à l'aide des bascules JK

a. Diagramme d'état



b. Table de vérité

$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3^+$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_3$	$K_3$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	X	1	0	X	0	X	X	1

c. Les équations des entrées des bascules  $J_0, K_0, J_1, K_1, J_2, K_2, J_3$  et  $K_3$ .

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	0	X	X
01	0	0	X	X
11	0	1	X	X
10	0	0	X	X

$$J_3 = Q_2 Q_1 Q_0$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	X	X	X	0
01	X	X	X	1
11	X	X	X	X
10	X	X	X	X

$$K_3 = Q_0$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	X	X	0
01	0	X	X	0
11	1	X	X	X
10	0	X	X	X

$$J_2 = Q_1 Q_0$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	X	0	X	X
01	X	0	X	X
11	X	1	X	X
10	X	0	X	X

$$K_2 = Q_1 Q_0$$

$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$				
00	0	0	X	0
01	1	1	X	0
11	X	X	X	X
10	X	X	X	X

$$J_1 = \overline{Q_3} Q_0$$

$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$				
00	X	X	X	X
01	X	X	X	X
11	1	1	X	X
10	0	0	X	X

$$K_1 = Q_0$$

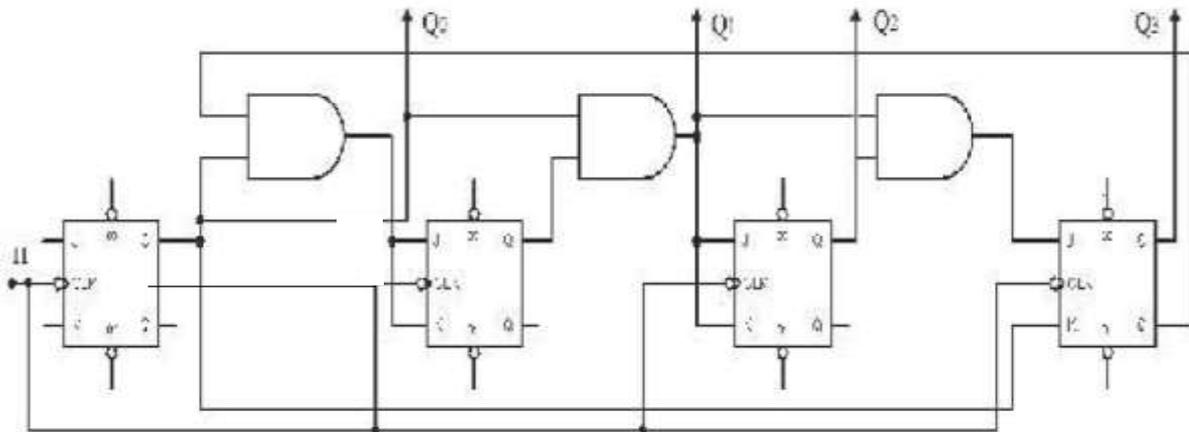
$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$				
00	1	1	X	1
01	X	X	X	X
11	X	X	X	X
10	1	1	X	X

$$J_0 = 1$$

$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$				
00	X	X	X	X
01	1	1	X	1
11	1	1	X	X
10	X	X	X	X

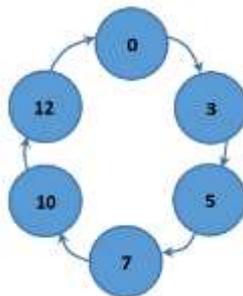
$$K_0 = 1$$

d. Schéma de compteur modulo 10 synchrone



2. compteur synchrone du cycle suivant : 0→3→5→7→10→12.

a. Diagramme d'état



b. Table de vérité

$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3^+$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_3$	$K_3$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	0	1	1	0	X	0	X	1	X	1	X
0	0	1	1	0	1	0	1	0	X	1	X	X	1	X	0
0	1	0	1	0	1	1	1	0	X	X	0	1	X	X	0
0	1	1	1	1	0	1	0	1	X	X	1	X	0	X	1
1	0	1	0	1	1	0	0	X	0	1	X	X	1	0	X
1	1	0	0	0	0	0	0	X	1	X	1	0	X	0	X

c. Les équations des entrées des bascules  $J_0, K_0, J_1, K_1, J_2, K_2, J_3$  et  $K_3$ .

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	X	X	X
01	X	0	X	X
11	0	1	X	X
10	X	X	X	X

$$J_3 = Q_2 Q_1$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	X	X	1	X
01	X	X	X	X
11	X	X	X	X
10	X	X	X	0

$$K_3 = Q_2$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	0	X	X	X
01	X	X	X	X
11	1	X	X	X
10	X	X	X	1

$$J_2 = Q_1$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	X	X	1	X
01	X	0	X	X
11	X	1	X	X
10	X	X	X	X

$$K_2 = Q_3 + Q_1$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	1	X	0	X
01	X	1	X	X
11	X	X	X	X
10	X	X	X	X

$$J_1 = \bar{Q}_3$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	1	0	X	X
10	X	X	X	1

$$K_1 = \bar{Q}_2$$

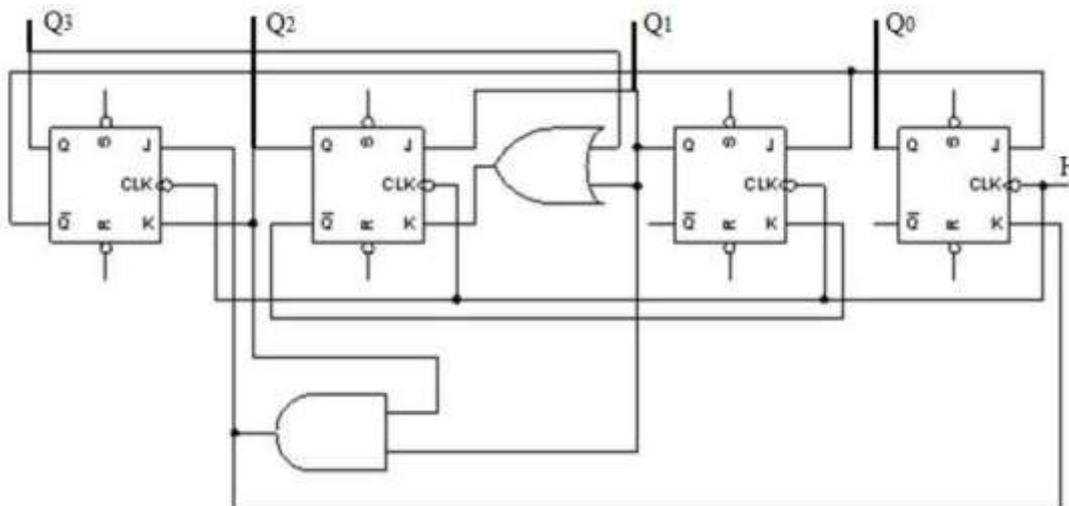
$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	1	X	0	X
01	X	X	X	X
11	X	X	X	X
10	X	X	X	0

$$J_0 = \bar{Q}_3$$

$Q_3 Q_2$ \ $Q_1 Q_0$	00	01	11	10
00	X	X	X	X
01	X	0	X	X
11	0	1	X	X
10	X	X	X	X

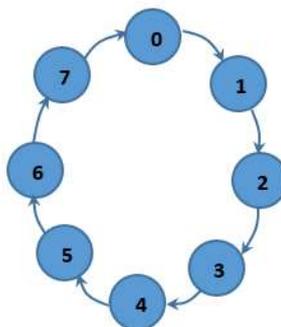
$$K_0 = Q_2 Q_1$$

d. Schéma de compteur modulo 10 synchrone



### 3. Compteur synchrone modulo 8 à l'aide des bascules D

a. Diagramme d'état



b. Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

c. Les équations des entrées  $D_0$ ,  $D_1$  et  $D_2$

$Q_2 \backslash Q_1$	00	01	11	10
0	0	0	1	1
1	0	1	0	1

$$\begin{aligned}
 D_2 &= \bar{Q}_2 Q_1 Q_0 + Q_2 \bar{Q}_1 + Q_2 \bar{Q}_0 \\
 &= \bar{Q}_2 Q_1 Q_0 + Q_2 (\bar{Q}_1 + \bar{Q}_0) \\
 &= \bar{Q}_2 Q_1 Q_0 + Q_2 (\overline{Q_1 Q_0}) \\
 &= Q_2 \oplus (Q_1 Q_0)
 \end{aligned}$$

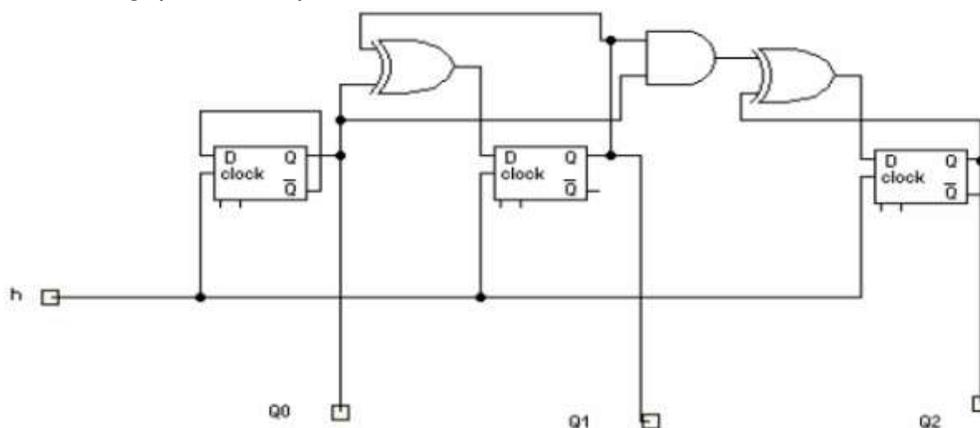
$Q_2 \backslash Q_1$	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$$\begin{aligned}
 D_1 &= Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0 \\
 &= Q_1 \oplus Q_0
 \end{aligned}$$

$Q_2 \backslash Q_1$	00	01	11	10
0	1	1	1	1
1	0	0	0	0

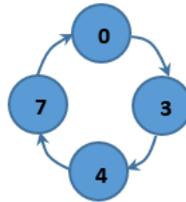
$$D_0 = \bar{Q}_0$$

d. Schéma logique de compteur



#### 4. Compteur synchrone de cycle 0→3→4→7 à base des bascules D

a. Diagramme d'état



b. Table de vérité

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$D_2$	$D_1$	$D_0$
0	0	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

c. Les équations des entrées  $D_0$ ,  $D_1$  et  $D_2$

$Q_2$ $Q_1$	00	01	11	10
0	0	X	X	1
1	X	1	0	X

$Q_2$ $Q_1$	00	01	11	10
0	1	X	X	1
1	X	0	0	X

$$D_2 = Q_2\bar{Q}_1 + \bar{Q}_2Q_1$$

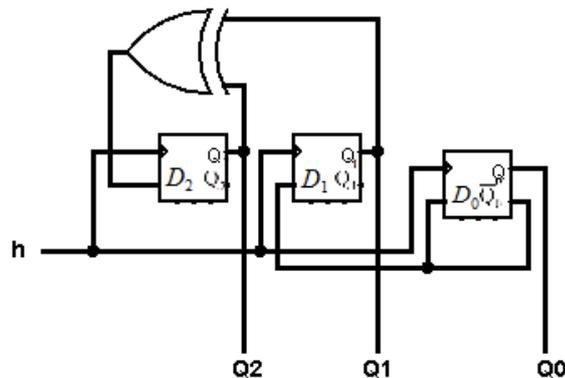
$$= Q_2 \oplus Q_1$$

$$D_1 = \bar{Q}_0$$

$Q_2$ $Q_1$	00	01	11	10
0	1	X	X	1
1	X	0	0	X

$$D_0 = \bar{Q}_0$$

d. Schéma de compteur modulo synchrone avec des bascules D



**Exercice N°7 :**

1. Les équations des entrées des bascules

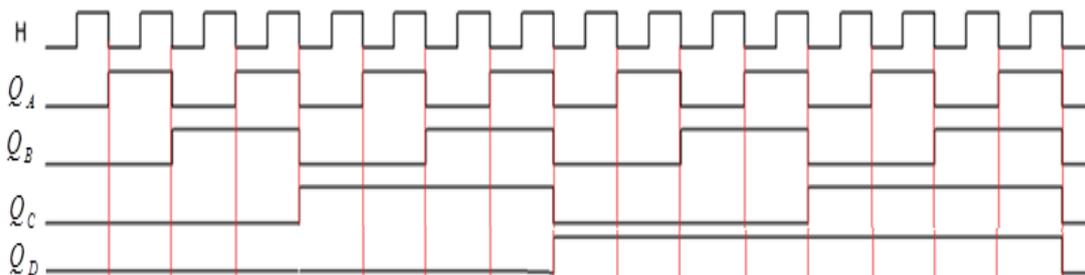
$$\begin{cases} J_A = 1 \\ K_A = 1 \end{cases} \quad \begin{cases} J_B = \overline{Q_A} + Q_D = Q_A \overline{Q_D} \\ K_B = \overline{Q_A} + Q_D = Q_A \overline{Q_D} \end{cases}$$

$$\begin{cases} J_C = \overline{Q_A Q_B} \overline{Q_A Q_B} = Q_A Q_B \\ K_C = \overline{Q_A Q_B} \overline{Q_A Q_B} = Q_A Q_B \end{cases} \quad \begin{cases} J_D = \overline{Q_A Q_B} + \overline{Q_C} = Q_A Q_B Q_C \\ K_D = Q_A \end{cases}$$

2. Table des transitions

état	$Q_D$	$Q_C$	$Q_B$	$Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0	1	1	1	1
2	0	0	1	0	0	0	0	0	0	0	1	1
3	0	0	1	1	0	1	1	1	1	1	1	1
4	0	1	0	0	0	0	0	0	0	0	1	1
5	0	1	0	1	0	1	0	0	1	1	1	1
6	0	1	1	0	0	0	0	0	0	0	1	1
7	0	1	1	1	1	1	1	1	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	1	1
9	1	0	0	1	0	1	0	0	0	0	1	1
0	0	0	0	0								

3. Les chronogrammes des sorties  $Q_A, Q_B, Q_C$  et  $Q_D$  en fonction de l'horloge H



4. Le tableau montre que ce circuit est un compteur modulo 10.

**Exercice N°8 :**

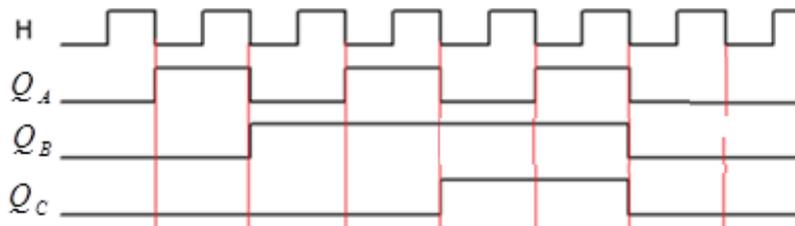
1. Les équations des entrées des bascules

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

## 2. Table des transitions

état	$Q_C$	$Q_B$	$Q_A$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	0	0	0	1	1
1	0	0	1	0	1	1	0	1	1
2	0	1	0	0	0	0	0	1	1
3	0	1	1	1	1	1	0	1	1
6	1	1	0	0	0	0	0	1	1
7	1	1	1	1	1	1	1	1	1
0	0	0	0						

### 1. Chronogrammes et des sorties $Q_A$ , $Q_B$ et $Q_C$ en fonction de l'horloge H



### 2. Ce compteur réalise la séquence suivante : 0→1→2→3→6→7→0

### 3. Détermination des fréquences $f_A$ , $f_B$ et $f_C$ des sorties $Q_A$ , $Q_B$ et $Q_C$ en fonction de la fréquence de l'horloge $f_H$

$$T_A = 2T, T_B = 6T \text{ et } T_C = 6T$$

$$f_H = \frac{1}{T}$$

$$f_A = \frac{1}{T_A} = \frac{1}{2T} = \frac{f_H}{2}$$

$$f_B = \frac{1}{T_B} = \frac{1}{6T} = \frac{f_H}{6}$$

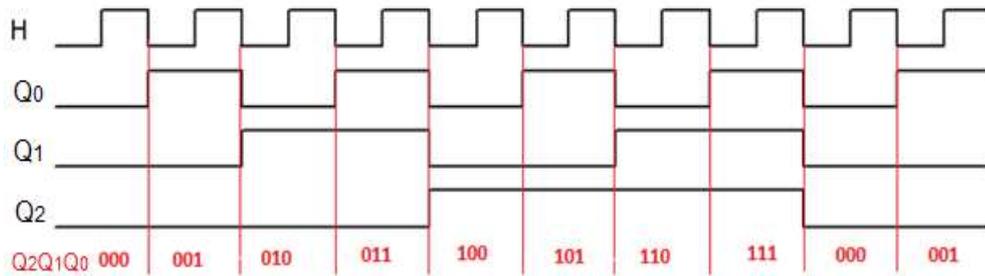
$$f_C = \frac{1}{T_C} = \frac{1}{6T} = \frac{f_H}{6}$$

## Exercice N°9 :

### 1. Comportement des bascules

- $Q_0$  a pour horloge H. Donc,  $Q_0$  bascule de 0 à 1 ou de 1 à 0 à chaque front descendant de H, c'est-à-dire change d'état.
- $Q_1$  a pour horloge  $Q_0$ . Donc,  $Q_1$  bascule de 0 à 1 ou de 1 à 0 à chaque front descendant de  $Q_0$ , c'est-à-dire change d'état.

- $Q_2$  a pour horloge  $Q_1$ . Donc,  $Q_2$  bascule de 0 à 1 ou de 1 à 0 à chaque front descendant de  $Q_1$ , c'est-à-dire change d'état.

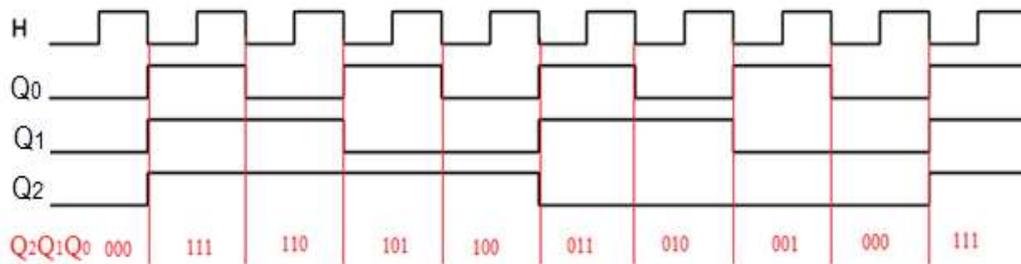


2. Ce premier circuit possède le cycle suivant :  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0$ .

3. Ce circuit est un compteur modulo 8.

4. Comportement des bascules

- $Q_0$  a pour horloge H,  $Q_0$  change d'état à chaque front descendant de H.
- $Q_1$  a pour horloge  $\overline{Q_0}$ ,  $Q_1$  change d'état à chaque front descendant de  $\overline{Q_0}$ , c'est à dire à chaque front montant de  $Q_0$ .
- $Q_2$  a pour horloge  $\overline{Q_1}$ ,  $Q_2$  change d'état à chaque front descendant de  $\overline{Q_1}$ , c'est-à-dire à chaque front montant de  $Q_1$ .



5. Ce deuxième circuit possède le cycle suivant :  $0 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$

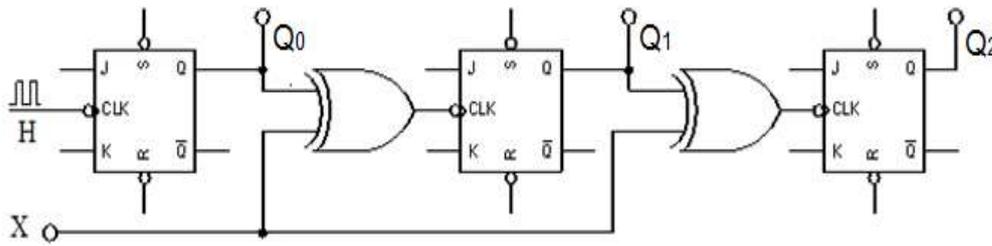
6. Ce circuit est un décompteur modulo 8.

7. circuit compteur/décompteur

- Pour la première bascule : son horloge dans les 2 cas est H.
- Pour les autres bascules : on utilise une variable X pour faire le choix entre le compteur et le décompteur, comme le montre le tableau suivant :

X	Mode	horloge
0	Compteur	$Q_{n-1}$
1	décompteur	$\overline{Q_{n-1}}$

$$\text{horloge} = Q_{n-1}\bar{X} + \bar{Q}_{n-1}X = Q_{n-1} \oplus X$$



### Exercice N°10 :

#### 1. Les équations

##### Bascule 1:

$$J_1 = K_1 = 1 \Rightarrow Q_1^+ = \bar{Q}_1$$

A chaque front montant de  $Clk_1$  on a  $Q_1^+ = \bar{Q}_1$

##### Bascule 2:

$$J_2 = K_2 = 1 \Rightarrow \begin{cases} Q_2^+ = \bar{Q}_2 \\ Clk_2 = \bar{Q}_1 \end{cases}$$

A chaque front descendant de  $Q_1$  on a  $Q_2^+ = \bar{Q}_2$

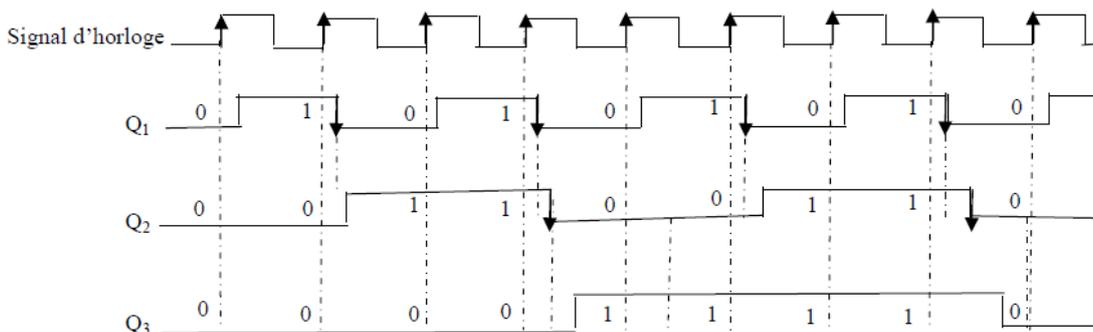
##### Bascule 3:

$$J_3 = K_3 = 1 \Rightarrow \begin{cases} Q_3^+ = \bar{Q}_3 \\ Clk_3 = \bar{Q}_2 \end{cases}$$

A chaque front descendant de  $Q_2$  on a  $Q_3^+ = \bar{Q}_3$

2. D'après le schéma du circuit, on a l'entrée d'horloge est appliquée seulement à la première bascule et l'état de chaque bascule est fonction des états des bascules précédentes, alors ce circuit est asynchrone.

#### 3. chronogramme du fonctionnement du circuit



#### 4. Table de transition

$Q_3$	$Q_2$	$Q_1$	$Q_3^+$	$Q_2^+$	$Q_1^+$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

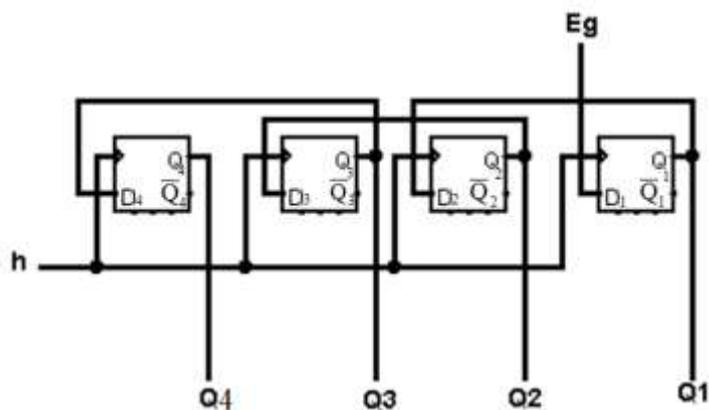
5. Ce circuit est un compteur asynchrone modulo 8.

#### Exercice N°11:

Réalisation d'un registre à décalage à droite vers la gauche avec une entrée série droite et une sortie série gauche à l'aide de quatre bascule D synchronisées sur le front montant de l'horloge.

- Les équations caractéristiques de ce registre sont :
 
$$\begin{cases} D_4 = Q_3 \\ D_3 = Q_2 \\ D_2 = Q_1 \\ D_1 = E_g \end{cases}$$

- Le schéma logique de ce registre :



#### Exercice N°12 :

- Les équations caractéristiques d'un registre à décalage à droite vers la gauche avec une entrée série droite et une sortie série gauche à l'aide de quatre bascule D synchronisées sur le front montant de l'horloge sont :



**Exercice N°14:**

- Calcul de l'adresse du sixième élément de tableau

Tant que Chaque élément est stocké sur 2 octets, donc le sixième élément se trouve à l'adresse de départ + 10 octet, ce qui donne  $(77)_8 + (12)_8 = (111)_8$  C'est-à-dire  $(63+10=73)$ .

- Calcul de la taille de la mémoire :

On a la taille de la mémoire est de 2 MO.

La taille de la mémoire en mots de 16 bits =  $\frac{2\text{MO}}{2} = 1\text{Mmots}$

car  $\Rightarrow 1\text{mot}(\text{de } 16\text{bits}) = 2\text{mots}(\text{de } 8\text{ bits})$

La taille de la mémoire en mots de 32 bits =  $\frac{2\text{MO}}{4} = 0.5\text{Mmots}$

$= \frac{2 * 1024 \text{KO}}{4} = 512 \text{Kmots de } 32\text{bits}$

car  $\Rightarrow 1\text{mot}(\text{de } 32\text{bits}) = 4\text{ mots}(\text{de } 8\text{ bits})$

### 3. Solution de la série d'exercices N°3 (Circuits intégrés)

#### Exercice N°1 :

##### 1. Explication des références suivantes :

###### ⇒ SN 74LS00 :

SN : signifie que le constructeur est Texas Instruments.

74 : fait référence aux circuits intégrés grand public qui résistent à des températures ambiantes de 0 à 70 degrés.

LS ou HCT : indiquent la sous famille du circuit TTL.

00 : les derniers chiffres indiquent la fonction logique réalisée par ce composant, par exemple 00 représente une porte NAND.

###### ⇒ CD4011 :

CD : signifie que le constructeur est Texas Instruments.

4011 : indique le numéro du circuit. Il s'agit d'un quadruple porte NAND (NON-ET) avec deux entrées chacune.

B : signifie que la tension maximale est de 18V.

E : cela signifie que le circuit est encapsulé dans un boîtier DIP.

##### 2. Définition des tensions et courants suivants :

- $V_{IH}$  (min) : tension d'entrée nécessaire pour avoir un 1 logique (niveau HAUT) en entrée.
- $V_{IL}$  (max) : tension d'entrée nécessaire pour avoir un 0 logique (niveau BAS) en entrée.
- $V_{OH}$  (min) : tension de sortie nécessaire pour avoir un 1 logique (niveau HAUT) en sortie.
- $V_{OL}$  (max) : tension de sortie nécessaire pour avoir un 0 logique (niveau BAS) en sortie.
- $I_{IH}$  : courant d'entrée qui traverse une borne d'entrée placée au niveau logique 1 est appliquée à cette entrée.
- $I_{IL}$  : courant d'entrée qui traverse une borne d'entrée placée au niveau logique 0 est appliquée à cette entrée.
- $I_{OH}$  : courant de sortie qui traverse une borne de sortie placée au niveau logique 1 est appliquée à cette sortie.
- $I_{OL}$  : courant de sortie qui traverse une borne de sortie placée au niveau logique 0 est appliquée à cette sortie.

(VCC = tension d'alimentation)

Technologie TTL : ( $V_{CC}=5V$ ) :

$$V_{IH\min} = 2V$$

$$V_{IL\max} = 0.8V$$

$$V_{OH\min} = 2.4V$$

$$V_{OL\max} = 0.4V$$

$$V_{IH \min} = 0.55 * V_{CC}$$

$$V_{IL \max} = 0.45 * V_{CC}$$

$$V_{OH \min} = 0.95 * V_{CC}$$

$$V_{OL \max} = 0.05 * V_{CC}$$

Technologie CMOS : (dépend de  $V_{CC}$ ) :

3.

⇒ **Temps de montée** : c'est le temps que met le courant IC pour passer de 10% de sa valeur maximale à 90% de sa valeur maximale. Il peut être noté  $t_m$  en français ou  $t_r$  (rise time) en anglais.

⇒ **Temps de descente** : c'est le temps que met le courant IC pour passer de 90% de sa valeur maximale à 10% de sa valeur maximale. Il peut être noté  $t_d$  en français ou  $t_f$  (fall time) en anglais.

4.

⇒ **L'immunité aux bruits** : représente la capacité d'un circuit intégré à tolérer les signaux parasites sans pour autant avoir un comportement incontrôlé ou erratique.

⇒ **L'entrance** : Représente le nombre maximum de sorties standard pouvant être connectées sur les entrées d'une porte logique.

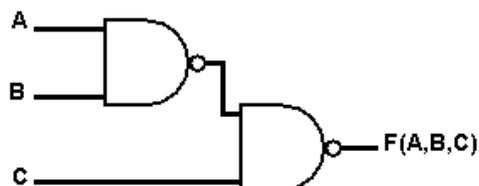
⇒ **La sortance** : Représente le nombre maximal d'entrées logiques peuvent être connectées simultanément à la sortie de la porte logique, et que cette sortie peut alimenter correctement.

5. la valeur de la sortance en TTL standard est la définition d'une charge unitaire est la suivante : 40  $\mu$ A à l'état HAUT et 1,6 mA à l'état BAS. Ces valeurs correspondent à la série TTL standard.

**Exercice 2 :**

Soit la fonction :  $F(A, B, C) = \overline{\overline{ABC}}$

Logigramme de  $F(A, B, C)$







## Bibliographies

- [1] : N. Agar et A. Agouni Kaci. « Réalisation d'un système de mesure de concentration à l'aide d'une carte arduino UNO ». Mémoire de master, Université Mouloud Mammeri, 2018.
- [2] : L. BOUZIDI, « Structure Machine 2, A la découverte des circuits logiques », Support de cours, 2020.
- [3] : O. Carton, « Circuits et architecture des ordinateurs », Univ. Paris Diderot, 2015.
- [4] : O. Temam, « Architecture des ordinateurs », Polycopié EP, 2008.
- [5] : L. Museur, « Electronique numérique Logique combinatoire et séquentielle ». Université Paris 13, Institut Galilée, 2015.
- [6] : A. Kachouri, « Logique combinatoire ». Support de cours, Université Virtuelle de Tunis, 2006.
- [7] : R. Strandh et I. Durand, « Architecture de l'ordinateur : Portes logiques, circuits combinatoires, arithmétique binaire, circuits séquentiels et mémoires ». Exe. Dunod, 2005.
- [8] : D. Gozim et K. Guesmi, « Logique combinatoire et séquentielle », support de cours, Université Ziane Achor de Djelfa, 2019.
- [9] : V. Risch, « Eléments d'Architecture des Ordinateurs », Dép. D'Informatique Inst. Univ. Technol. Univ. Méditerranée, 2015.
- [10] : E. G. Almouzni, « Architecture Des Ordinateurs », Cours, TD, TP. EISTI, 2013.
- [11] : P. M. F. Bouami, « Architecture des Ordinateurs », Support de Cours Filière SMI, Semestre 4, faculté pluridisciplinaire de Nador. 2020.
- [12] : A. Cazes et J. Delacroix, « Architecture des machines et des systèmes informatiques, Dunod, 2005. • T », Cormen C Leiserson R Rivest Introd. À Algorithmique Dunod, vol. 1, 2002.
- [13] : P. Darche, « Architecture des ordinateurs, mémoires à semi-conducteurs : principe de fonctionnement et organisation interne des mémoires vives ». Edition VUIBERT, paris, 2004.
- [14] : T. Dumartin, « Architecture des ordinateurs », Polycopié, vol. 4, no 4, p. 4.
- [15] : E. Viennet, « Architecture des ordinateurs ». Support de cours, GTR, 1999.

- [16] : N. Julien, « Logique séquentielle », Support de cours, Université de Bretagne Sud Lorient, 2003.
- [17] : F. Ardjani et M. Belkheir, « Logique combinatoire et séquentielle », Polycopié Pédagogique, Centre Universitaire Nour Bachir El Bayadh, 2020.
- [18] : A. Bouchemel, « Logique Combinatoire et S equentielle », 2019.
- [19] : J.M. Poitevin, « Aide-mémoire Electronique analogique et numérique », Dunod, 2008.
- [20] : M. Karam, « Génération de test de circuits intégrés fondée sur des modèles fonctionnels. (Test generation of integrated circuits based on functional models) ». Grenoble Institute of Technology, France, 1991.
- [21] : D. Etiemble, « Composants logiques et opérateurs matériels : exemples de réalisations », polycopié, Université Paris Sud, 2002.
- [22] : M. Bauge et J. Rivierre, « conception des circuits intégrés complexes », Laboratoire de développement de composants IBM France, H 695.
- [23] : C. Douillard et A. Thépaut, « Logique combinatoire et circuits MOS », Polycopié, Ecole Nationale Supérieure des Télécommunications de Bretagne, 2009.