



Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

وزارة التعليم العالي والبحث العلمي

Université TAHRI Mohammed Bechar

جامعة طاهري محمد بشار

Faculté de Technologie

كلية التكنولوجيا

V/Doyen de la Post Graduation, de la recherche scientifique  
et des relations extérieures

نيابة ما بعد التدرج و البحث العلمي و العلاقات الخارجية

## *Polycopié Pédagogique de Cours*

*Intitulé :*

# Système à microprocesseur

Code de la Matière : F511 Niveau : 3 Année Licence

Filière : Electronique

Spécialité : .Electronique

---

Etabli par l'enseignant(e) : Dr. REBHI MHAMED

Experts :

Pr. Bounaama Fatah

Dr. Mahi Mohammed Salah

## *Dédicace*

- إلى روح الفقيد والدي ربحي علي بن أحمد

الذي رحل عن الدنيا يوم 19 يناير 2022

داعياً المولى عز وجل أن يتغمده برحمته

الواسعة ويسكنه فسيح جناته.

- إلى الوالدة الكريمة داعياً المولى عز وجل

أن يرزقها دوام الصحة والعافية والسعادة

بین أبنائهما وزوجاتهما وأحفادها .

## *Présentation*

L'électronique a traversé plusieurs étapes d'invention, d'architecture et d'implantation, à partir de l'électronique analogique qui a permis de représenter, modéliser, convertir et mesurer les grandeurs physiques sous forme de signaux continus ou en fonction du temps jusqu'à l'électronique des systèmes embarqués et intelligents actuellement. En 1968, la fabrication du circuit intégré a établi une plateforme extraordinaire de l'architecture du système à microprocesseur permettant de traiter les informations sous forme binaire, ce brevet était le fondement d'accès à l'électronique digitale qui est devenue un outil de développement dans tous des domaines techniques contemporains.

Le microprocesseur étant un circuit intégré à base d'un matériau semi-conducteur, joue un rôle du cerveau de l'ordinateur en traitant les informations à l'aide d'un programme structuré. Ce manuscrit est destiné aux étudiants de premier cycle (licence) du génie électrique, a pour but de donner les connaissances de base d'un système à microprocesseur, il est adapté et organisé suivant le programme académique de la tutelle d'une manière plus facile et à la portée des étudiants obtenant des connaissances préalables de la logique combinatoire et séquentielle.

Le support pédagogique est subdivisé en huit chapitres :

- Le chapitre I permet de présenter des notions générales sur la logique séquentielle en concentrant sur les bascules de base.
- Le chapitre II présente les registres et la classification des compteurs binaires.
- Le chapitre III donne une description détaillée sur les types de mémoires à semi-conducteur, leurs architectures et principes de fonctions.
- Le chapitre IV poursuit l'histoire et l'évolution des ordinateurs et leurs architectures actuelles, en résumant la conception de base d'un système à microprocesseur.
- Le chapitre V permet d'expliquer l'histoire, le rôle, l'architecture interne et principe de fonctionnement du microprocesseur pour traiter les informations.
- Le chapitre VI est de présenter un exemple d'application d'un type du microprocesseur 8 bits et ses instructions de programmation.
- Le chapitre VII est consacré d'expliquer l'unité d'entrée-sortie ou des interfaces pour échanger les informations avec le monde extérieur.
- Le chapitre VIII donne un aperçu sur les interruptions et les critères de priorité.
- Une conclusion finale résume les points indispensables du système à microprocesseur.

## Table de Matière

<b>Titre</b>	<b>Page</b>
<b>Dédicace</b>	
Liste des figures	
Liste des tableaux	
Présentation	01
<b>Chapitre I : Généralités sur la logique séquentielle</b>	
I.1.Introduction	02
I.2.logique combinatoire	02
I.3.Logique séquentielle	02
I.3.1. Les bascules	03
I.3.1.1. Bascule RS	03
I.3.1.2. Bascule D	06
I.3.1.3. Bascule JK	07
I.3.1.4. Bascule T	08
I.4. Conclusion	08
<b>Chapitre II Les registres</b>	
II.1 Introduction	10
II.2. Registres classiques	10
II.2.a. Registre parallèle	10
II.2.b. Registre série	11
II.3. Registre universel.	15
II.4. Les compteurs	17
II.5. Applications des compteurs synchrones	18
II.6.Conclusion	18
<b>Chapitre III Les mémoires à semi-conducteurs</b>	
III.1. Introduction	28
III.3. Différentes technologies des mémoires	28
III.2. Notion d'une mémoire à semi conducteur	28
III.4. Différents types des mémoires à semi-conducteurs	29
III.5. Caractéristiques générales d'une mémoire	31
III.6. Organisation interne d'une mémoire	31
III.7. Critères du choix d'une mémoire	32

III.8. Extension de la mémoire.	33
III.9. Conclusion	
<b>Chapitre IV Historique et Evolution des ordinateurs</b>	
IV.1. Introduction	35
IV.2. Historique	35
IV.3. Organisation d'un ordinateur en blocs fonctionnels	37
IV.4. Les périphériques	40
IV.5. Horloge	41
IV.6. Vocabulaire de l'ordinateur	41
IV.7. Architecture d'un ordinateur	41
IV.7.1. Architecture Von Neumann	41
IV.7.2. Architecture Harvard	42
IV.8. Conclusion	42
<b>Chapitre V : Architecture et fonctionnement d'un microprocesseur</b>	
V.1.Historique	43
V.2.Définition	43
V.3.Caractéristiques générales d'un microprocesseur	44
V.4.Brochage et signaux	45
V.5.Architecture interne d'un microprocesseur :	46
a- Unité arithmétique et logique (UAL) :	46
b- Unité de commande et de contrôle :	47
c- Les registres de stockage temporaire :	47
d- Notion de pile :	48
V.6.Format d'une instruction,	48
V.7.Langage machine	48
V.8. Langage assembleur	49
V.9.Notions d'architecture RISC et CISC	51
V.10.Notions sur les interruptions	52
V.11.Conclusion	52
<b>Chapitre VI : Etude d'un microprocesseur 8 bits</b>	
VI.1. Introduction	54
VI.2. Les différentes familles de microprocesseurs 8 bits	54
VI.3. Compatibilité entre microprocesseurs	54
VI.3.2. Prédominance des microprocesseurs Intel et Motorola	55
VI.4. Etude détaillée d'un type de microprocesseur 8 bits, brochage et signaux externes	55

VI.5. Architecture interne	56
VI.6. Format d'une instruction	60
VI.7. Modes d'adressage	60
VI.8. Jeu d'instructions	61
VI.9. La gestion de la pile.	65
VI.10. Programme en assembleur	66
VII.11. Conclusion	66
<b>Chapitre VII : Les interfaces d'entrées /sorties</b>	
VII.1. Introduction	68
VII.2. Définition	68
VII.3. Communication entre le microprocesseur et interface	69
VII.4. Différents types d'interfaces	69
VII.4.1 .Interface parallèle 8255	69
VII.4.2. Interface série	74
VII.5. Application sur l'interface parallèle 8255 :	78
VII.6. Conclusion	78
<b>Chapitre VIII. Les interruptions</b>	
VIII.1. Introduction	80
VIII.2. Définition	80
VIII.3. Protocoles d'échanges de données.	80
VIII.4. La gestion des interruptions	81
VIII.5. La classification des interruptions	82
VIII.6. Interruptions masquables et interruptions non masquables,	83
VIII.7. Processus du traitement d'une interruption	84
VIII.8. Application : Programme de l'interruption Horloge.	85
VIII.9. Conclusion	
<b>IX. Conclusion générale</b>	86
<b>X. Références</b>	87

## Liste des figures

<b>Titre ou description</b>	<b>Page</b>
Figure I.1 : Système combinatoire.	02
Figure I.2 : Système séquentiel	02
Figure I.3 : chronogramme de fonctionnement de bascule RS.	04
Figure I.4 : Structure NAND de bascule RS	04
Figure I.5 : Structure NOR de bascule RS	05
Figure I.6 : chronogramme de bascule RSH	06
Figure I.7 : Schéma fonctionnel de bascule D	06
Figure I.8: Chronogramme de bascule D.	07
Figure I.10 : chronogramme de bascule JK.	08
Figure I.11 : Schéma fonctionnel de bascule T	08
Figure I.12 : Chronogramme de bascule T	09
Figure II.13. : Registre parallèle à 4 bits	10
Figure II.14. : Schéma fonctionnel du registre PIPO	11
Figure II.15 : schéma fonctionnel de registre SISO	11
Figure II.16. : registre de décalage à droite	12
Figure II.17: Registre de décalage à gauche.	12
Figure II.18: registre de décalage réversible.	13
Figure II.19. : Registre PISO.	13
Figure II.20. : Registre PISO réalisé par les bascules D	14
Figure II.21. : Registre SIPO	14
Figure II.22 : Registre SIPO réalisé par les bascules D.	14
Figure II.23: vue externe d'un registre universel	15
Figure II.24 : détail d'une cellule de registre universel	16
Figure II.25: bascule montée en diviseur par 2	17
Figure II.26 : Compteur asynchrone à cycle complet sur 3 bits	19
Figure II.27 : Schéma fonctionnel du compteur asynchrone	20
Figure II.28 : Chronogramme du compteur asynchrone	20
Figure II.28 : Schéma fonctionnel du compteur synchrone	23

Figure II.29 : chronogramme du compteur synchrone	24
Figure II.30 : Schéma et chronogramme du compteur synchrone incomplet	25
Figure II.31 : compteur programmable	26
Figure II.32 : cellule d'un compteur programmable	27
Figure III.33 : structure de la mémoire	29
Figure III.34 : Schéma fonctionnel d'une mémoire	29
Figure III.35 : Image d'une barrette de mémoire RAM.	30
Figure III.36 : image de mémoire ROM	30
Figure III.37 : organisation d'une mémoire	32
Figure III.38 : cycle de fonctionnement en mode écriture d'une mémoire RAM	32
Figure III.39 : Construction d'un espace mémoire	33
Figure III.40 : Extension d'une mémoire	34
Figure IV.41: organisation d'un système à microprocesseur	38
Figure IV.42 : schéma fonctionnel d'un microprocesseur	39
Figure IV.43 : Hiérarchie de mémoires	40
Figure IV.44: architecture Von Neumann	42
Figure IV.45 : architecture Harvard.	42
Figure V.46 : Le microprocesseur de Marcian Hoff et Federico Faggin	44
Figure V.47 : Brochage du microprocesseur	46
Figure V.48 : architecture interne standard d'un microprocesseur	46
Figure V.49 : Structure de l'unité arithmétique et logique	47
Figure VI.50 : brochage du microprocesseur 8085	55
Figure VI.51 : Démultiplexage du bus d'adresses/ données	56
Figure VI.52 : Structure interne du microprocesseur 2085	59
Figure VII.53 : microprocesseur et interfaces	68
Figure VII.54 : Brochage d'interface 8255	69
Figure VII.55 : Modes de programmation de l'interface 8255	71
Figure VII.56 : modes d'interface série	74
Figure VII.57 : Brochage de l'interface série 8250	75
Figure VII.58 : topologie d'un réseau USB	78
Figure VII.59 : Exemple d'application du 8255A en mode 0 en entrée et en sortie	78
Figure VIII.60 : Interconnexion microprocesseur-interface	80

Figure VIII.61 : la demande d'interruption	81
Figure VIII.62 : Brochage du contrôleur 8259A	87
Figure VIII.63 : architecture interne du contrôleur 8259A	88

## Liste des tableaux

<b>Tableau</b>	<b>Page</b>
Tableau I.1 : Table de transition de bascule RS	04
Tableau I.2 : Table de transition de bascule RSH	05
Tableau I.3 : table de transition bascule JK	08
Tableau I.4 : Table de transition de bascule T	09
Tableau VII.5 : les adresses de sélection d'un port	70
Tableau VII.6 : adresses détaillant la sélection des ports	71
Tableau VII.7 : mode 0 du 8250	72
Tableau VII.8 : Registre de contrôle	74
Tableau VII.9 : brochage d'interface 8250	78
Tableau VIII.10 : Interruptions logicielles du 8085.	84

**Chapitre I****Généralités sur la logique séquentielle****I.1. Introduction**

La logique combinatoire ne suffit pas à la manipulation des informations dans les systèmes numériques, il est incapable de conserver des données après les résultats, la logique séquentielle est la solution adéquate pour cette mission.

**I.2. Logique combinatoire**

Un système logique est dit combinatoire si l'état de ses sorties  $S_k$  ne dépend que de l'état de ses entrées  $E_k$ . Le système combinatoire ne doit donc pas présenter de réactions de la sortie sur l'entrée, de sorte à ce que l'état de la sortie ne dépend pas de l'histoire du système.

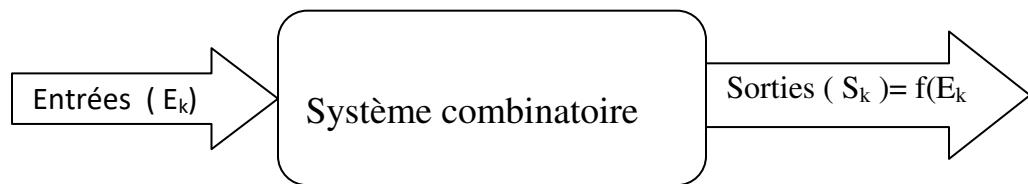


Figure I.1 : Système combinatoire.

**I.3. Logique séquentielle [01]**

Système dont l'état des sorties dépend non seulement des entrées mais également de l'état antérieur des sorties. Ces circuits doivent donc être capables de mémoriser

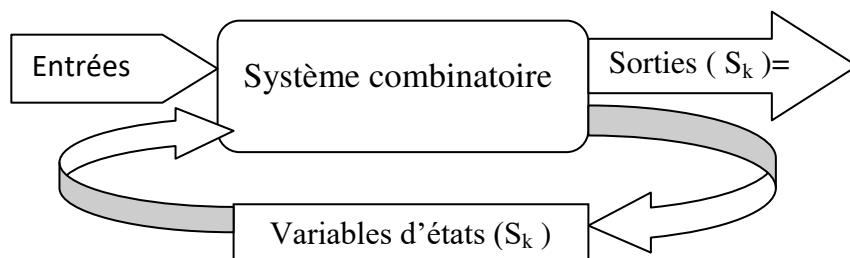


Figure I.2 : Système séquentiel

Circuit séquentiel : un circuit séquentiel est un système bouclé permettant la conservation d'un état dépendant de la valeur des variables d'entrée ainsi que de l'état antérieur du système. Les fonctions séquentielles de base sont : mémorisation, comptage, décalage.

Les circuits séquentiels de base sont : bascules (3 types), compteurs, registres, mémoire RAM

Ces circuits séquentiels peuvent travailler soit en mode synchrone, soit en mode asynchrone :

- **Mode asynchrone** : À tout moment, les signaux d'entrée peuvent provoquer le changement d'état des sorties (après un certain retard qu'on appelle « temps de réponse »). Ces systèmes sont difficiles à concevoir et à dépanner.

- **Mode synchrone** : Le moment exact où les sorties peuvent changer d'état est commandé par un signal d'horloge (train d'ondes carrées ou rectangulaires). Les changements d'état s'effectuent tous pendant une transition appelée « front » (montant ou descendant).

La majorité des systèmes numériques séquentiels sont synchrones même si certaines parties peuvent être asynchrones (ex. : reset).

Les avantages principaux du mode synchrone sont :

- préparer les entrées sans perturber les sorties ;
- protéger des parasites survenant en entrée.

### I.3.les bascules

La bascule constitue le système séquentiel de base et permet de mémoriser un élément d'information élémentaire appelé bit, il ya plusieurs types des bascules dont : RS , Jk, D , T

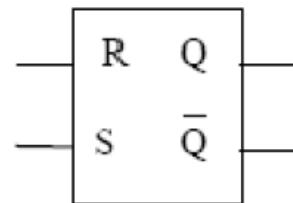
#### I.3.1 Bascule RS

La bascule RS est le circuit séquentiel le plus simple. C'est une bascule asynchrone, et toutes les autres bascules, synchrones ou asynchrones, reposent sur cette bascule. Son rôle consiste à mémoriser une information fugitive, selon le fonctionnement suivant : une apparition, même fugitive, de S entraîne un état stable  $Q=1$ , et une apparition, même fugitive, de R entraîne un état stable  $Q=0$ .

##### Bascule RS asynchrone

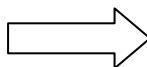
Asynchrone : l'état de la bascule peut changer à n'importe quel instant (dès que le changement des variables d'entrées influent sur le changement de l'état de la sortie). (**S-R = S:** Set: mise à 1

**R: Reset: mise à 0**), son symbole est donné comme suit:



La table de transition peut être donnée

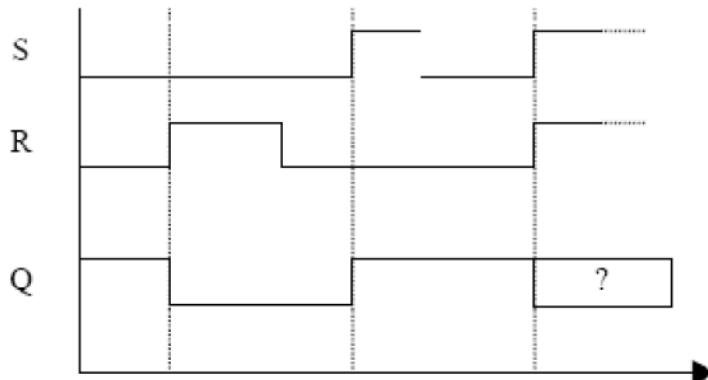
S	R	$Q_t$	$Q^+$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X



S	R	$Q^+$
0	0	Mémorisation
0	1	0
1	0	1
1	1	Interdit

Tableau I.1 : Table de transition bascule RS

Chronogramme



- 2 états stables :
- $S = 0, R = 0 Q = 0$
  - $S = 1, R = 0 Q = 1$
- 1 position mémoire
- $S = 0, R = 0 Q+ = Q-$
- 1 position indéterminée
- $S = 1, R = 0 Q = ?$

Figure I.3 : chronogramme de fonctionnement de bascule RS.

### Réalisation

Si  $X = 1$ ,  $Q = S + R \cdot Q$ . Les états indéterminés sont forcés à 1 : la bascule est dite à enclenchement prioritaire. La somme de produit implique une réalisation à l'aide de portes NAND

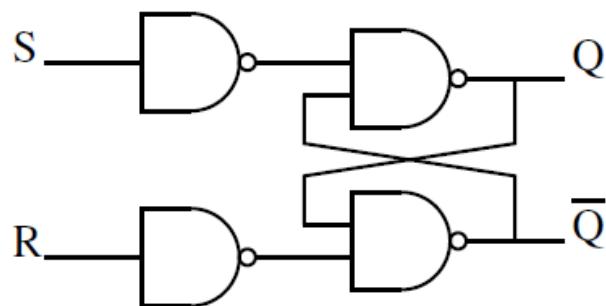


Figure I.4 : Structure NAND de bascule RS

Si  $X=0$   $Q = \overline{R}(S+Q)$ . Les états indéterminés sont forcés à 0 : la bascule est dite à déclenchement prioritaire. Le produit de sommes implique une réalisation à l'aide de portes NOR.

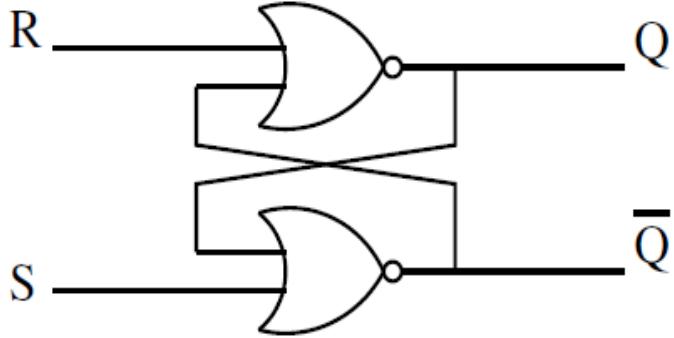
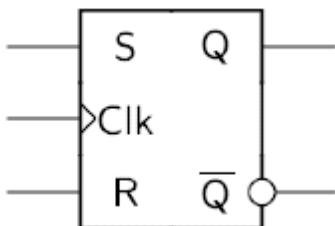


Figure I.5 : Structure NOR de bascule RS.

### Bascule RSH

La bascule RSH1 est un bascule RS synchronisée par un signal d'horloge H. Lorsque H est au niveau bas, la bascule fonctionne comme une mémoire, et lorsque H est au niveau haut, la bascule fonctionne comme une bascule RS classique, et conserve donc les états interdits pour  $R=S=1$ . Son symbole est donné par :

Table de transition



S	R	H	$Q_{N+1}$
X	X	0	$Q_N$
0	0	1	$Q_N$
0	1	0	1
1	0	1	1
1	1	X	X

Tableau I.2 : Table de transition de bascule RSH

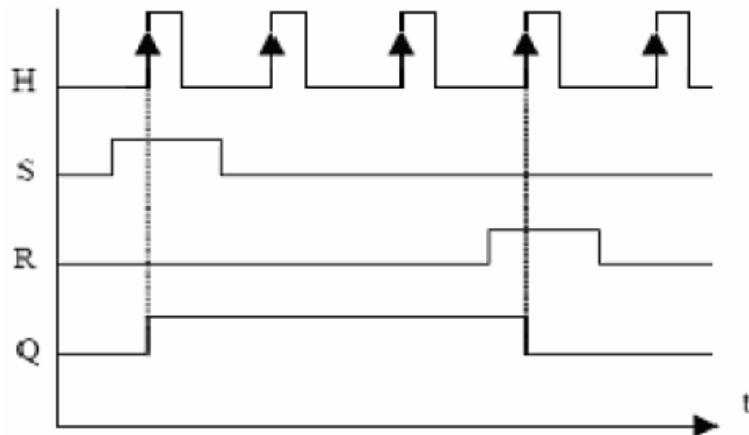


Figure I.6 : chronogramme de bascule RSH.

### I.3.2.Bascule D

A partir d'une bascule **R-S-H**, on commande les entrées par le même signal en utilisant Inverseur sur S.

- Une seule entrée d'excitations **D**
- 2 sorties :  $Q$  et  $\bar{Q}$ .
- Une bascule **D** synchrone fonctionne au rythme d'une horloge **H**

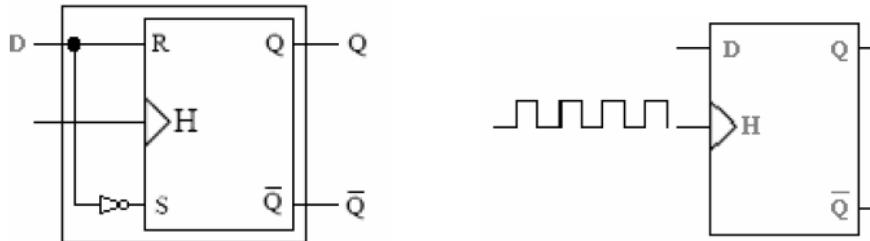


Figure I.7 : Schéma fonctionnel de bascule D.

$$D = S = R$$

$$\text{Si } S = 0 \Rightarrow R = 1, Q+ = 0 \Rightarrow D = 0$$

$$\text{Si } S = 1 \Rightarrow R = 0, Q+ = 1 \Rightarrow D = 1$$

$$\text{Si } D = 0 \Rightarrow Q+ = 0 \text{ Si } D = 1 \Rightarrow Q+ = 1$$

Équation caractéristique de la bascule D

$$Q+ = D$$

“ Fonctionnement

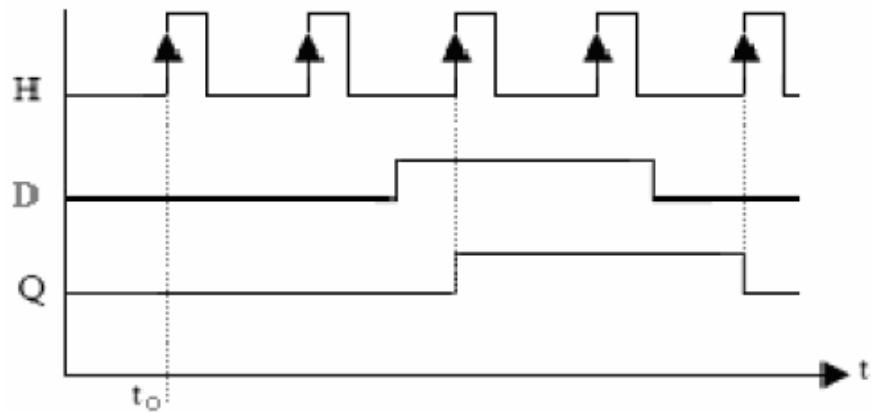


Figure I.8: Chronogramme de bascule D.

On considère une D (front montant)

### I.3.3. Bascule J-K

C'est une bascule RS excitée par trois entrées J,K,et  $Q^-$  que vérifié les

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

Les bascules JK sont des bascules maître-esclave fonctionnant seulement en mode synchrone.

Elles sont plus polyvalentes que les bascules RS, car elles n'ont pas d'état ambigu

$$Q_{N+1} = J \cdot \overline{Q_N} + \overline{K} \cdot Q_N$$

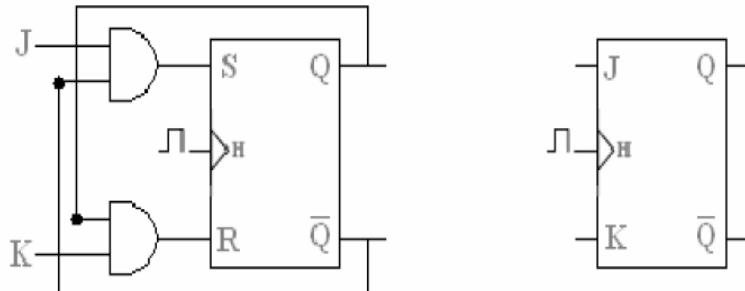


Figure I.9 : schéma fonctionnel de bascule JK.

$$Q_+ = S + \overline{R}Q = J\bar{Q} + (\overline{K}\bar{Q})Q \Rightarrow Q_+ = J\bar{Q} + \overline{K}Q$$

$$\overline{Q}_+ = KQ + \overline{J}\bar{Q}$$

Table de transition :

<b>J</b>	<b>K</b>	<b>Q<sub>t</sub></b>	<b>Remarques</b>
0	0	Q <sub>t-1</sub>	Aucun changement : la bascule reste dans l'état
0	1	0	Mise à 0 de la sortie Q
1	0	1	Mise à 1 de la sortie Q
1	1	Q̄ <sub>t-1</sub>	Complément de la sortie

Tableau I.3 : table de transition bascule JK.

Chronogramme :

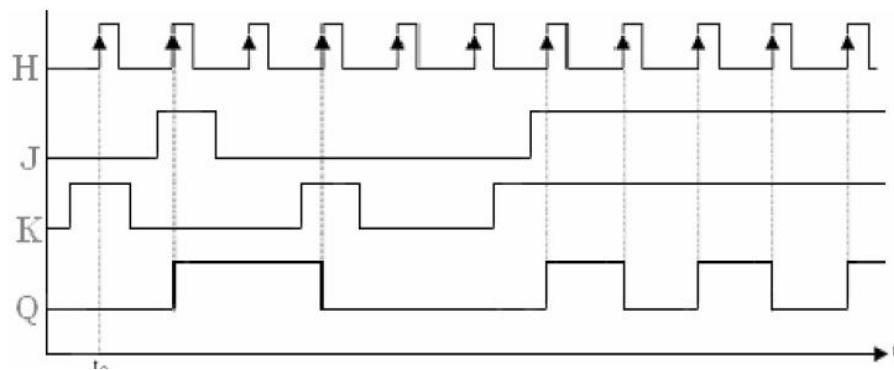


Figure I.10 : chronogramme de bascule JK.

### I.3.4. Bascule T

C'est un cas particulier de la bascule J-K, elle correspond à  $T = J = K$ .

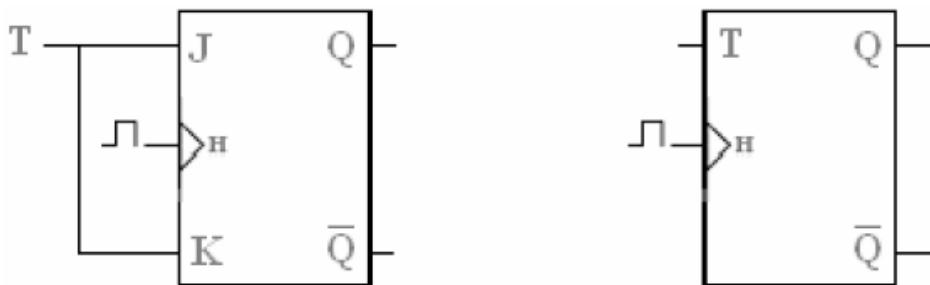


Figure I.11 : Schéma fonctionnel de bascule T

Équation caractéristique de la bascule T

$$Q_+ = T \oplus Q$$

T	Q	$Q_+$
0	0	0
0	1	1
1	0	1
1	1	0

Tableau I.4 : Table de transition de bascule T

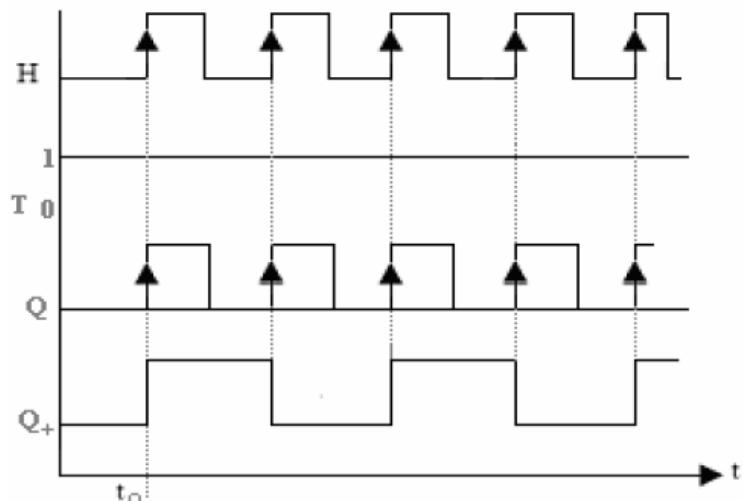


Figure I.12 : Chronogramme de bascule T

#### I.4. Conclusion

La bascule représente la cellule de base pour stocker une information, donc il nécessaire de savoir son type , sa structure et son fonctionnement, ces connaissances préalables permettent de traverser vers le système séquentiel pour circuler les informations suivant leurs tailles et capacités du stockage.

**Chapitre II :****les registres****II.1. Introduction**

Un registre est un ensemble des mémoires élémentaires dites bascules, une bascule permet de conserver un bit (0 ou 1), les registres sont classés suivant le nombre de bits et le mode de fonctionnement (unique ou multiple), on peut trouver les types suivants :

**II.1. les registres classiques [02]:****II.1.1. les registres à décalage :**

Les registres à décalage sont des registres permettant de déplacer un ou ensembles des bits sur la chaîne des bascules en sens parallèle ou en série.

**a. Registre parallèle :** c'est un registre de mémorisation ou un registre PIPO ( Parallel IN Parallel Out ) dans lequel les différentes étages (bascules) sont indépendants les uns des autres, cependant certains signaux agissent sur l'ensemble des étages tel que remise à 0 et remise à 1.

Dans ce registre, les bascules sont chargées en parallèle et lues en parallèle en synchrone avec le signal d'horloge H

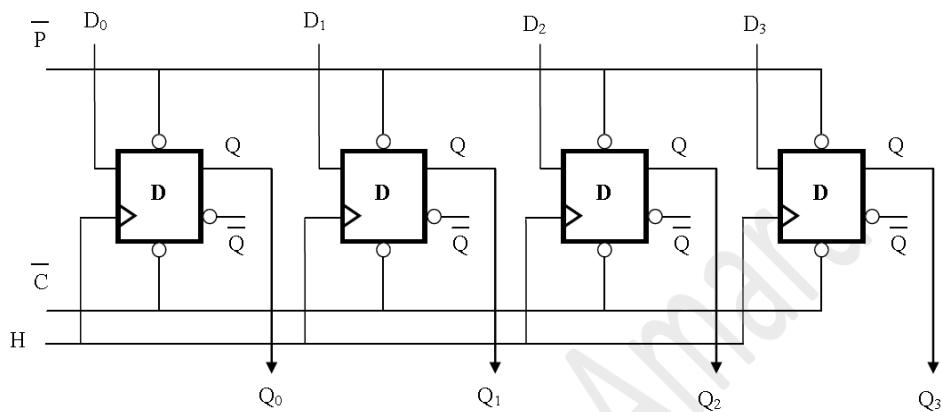


Figure II.13. : Registre parallèle à 4 bits

Le schéma fonctionnel de registre PIPO peut être représenté dans la figure suivante :

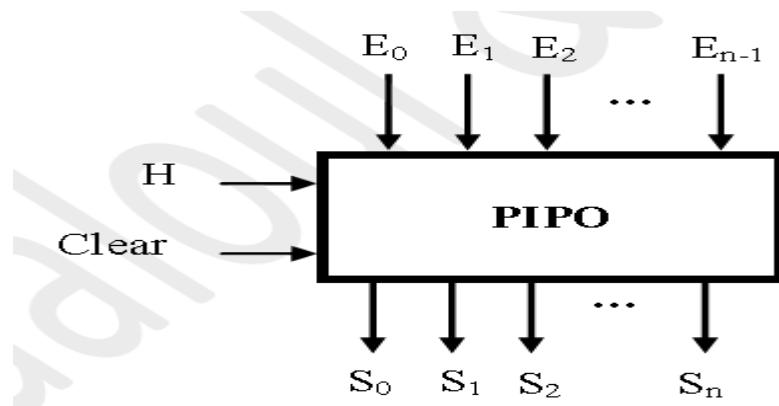


Figure II.14. : Schéma fonctionnel du registre PIPO

$E_0, E_1, E_2, \dots, E_{n-1}$  : les entrées de chargement.

$H$  : signal d'horloge ou de synchronisation.

Clear : signal de remise à 0 (effacer).

$S_0, S_1, S_2, \dots, S_{n-1}$  : les sorties de registre.

### b. Registre série :

Ce type de registre est principalement utilisé comme mémoire d'information dynamique ; la fonction de décalage consiste de faire glisser l'information de chaque cellule élémentaire dans une autre cellule élémentaire adjacente. Ce type de registre est appelé aussi registre **SISO**.

Le schéma fonctionnel de registre série est donné par :

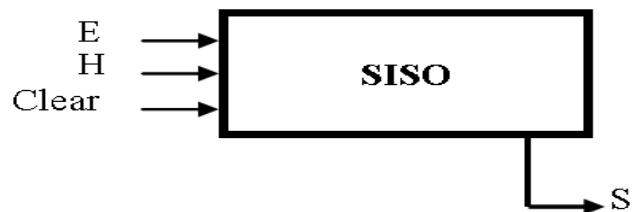


Figure II.15 : schéma fonctionnel de registre SISO

On trouve registre de décalage à droite et registre de décalage à gauche

#### Décalage à droite

La bascule du rang  $i$  doit recopier la sortie de la bascule du rang  $(i-1)$  ainsi son entrée doit être connectée à la sortie  $(i-1)$ .

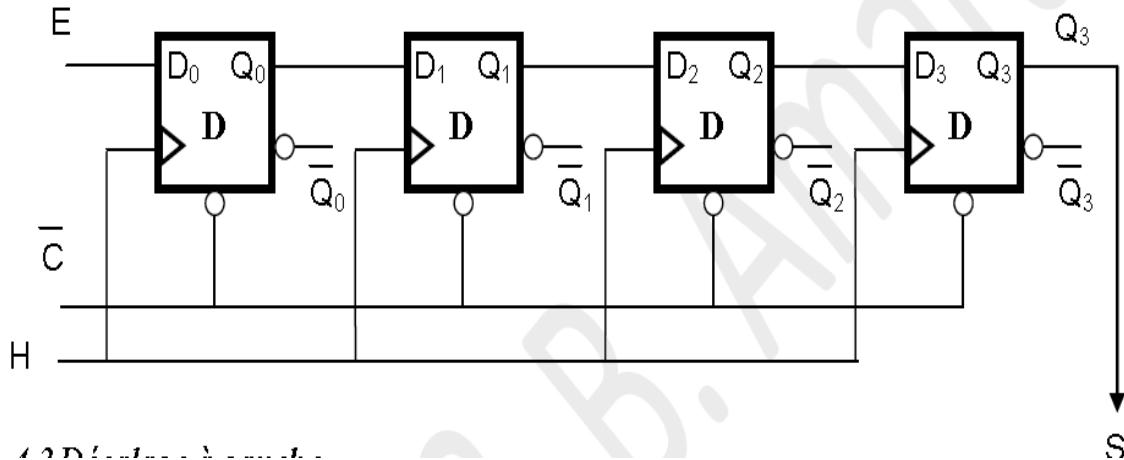


Figure II.16. : registre de décalage à droite

### Décalage à gauche

L'entrée de la bascule du rang  $i$  doit recopier la sortie de la bascule du rang  $(i+1)$ .

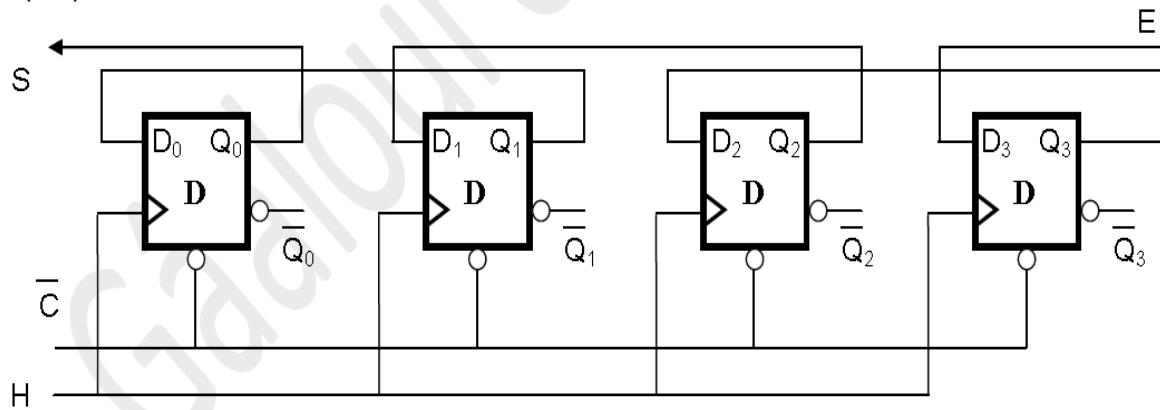


Figure II.17: Registre de décalage à gauche.

### Décalage réversible

Il existe des registres à décalage réversibles, c'est à dire des registres où le décalage s'effectue vers la droite et vers la gauche en fonction du niveau logique applique à l'entrée  $S$  : « sens de décalage »

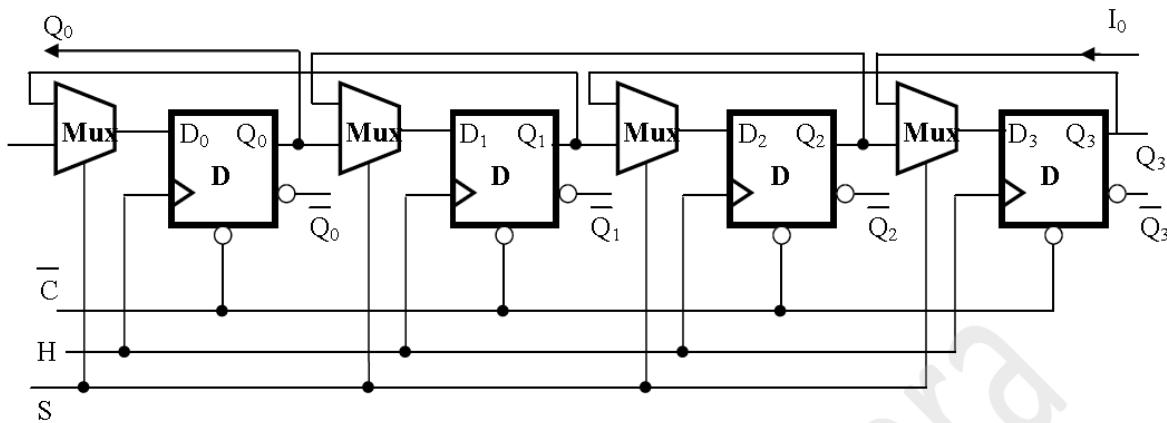


Figure II.18. : registre de décalage réversible.

En fonction de la valeur de l'entrée S de sélection, on a l'opération suivante :

Etat du signal S	Opération
0	Décalage à gauche
1	Décalage à droite

### c. Registre mixte

On peut trouver des registres mixtes, donc on peut écrire en parallèle et lire en série (**PISO**), ou vice versa (**SIP**O), ou qui offrent les deux possibilités « au choix ».

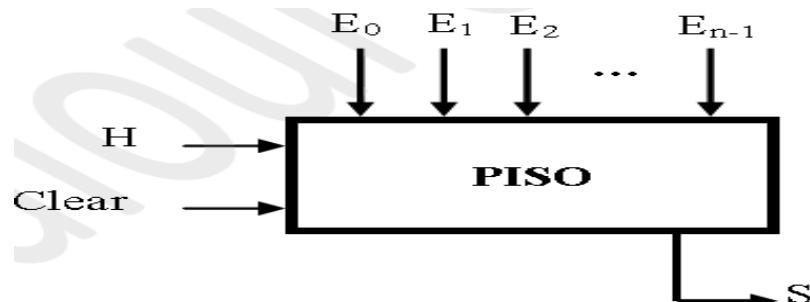


Figure II.19. : Registre PISO.

Le registre PISO peut être réalisé par les bascules D comme suivant :

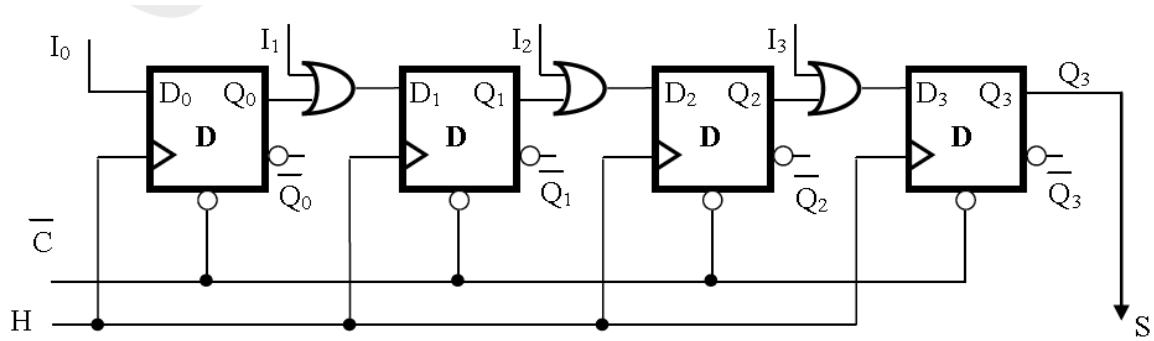


Figure II.20. : Registre PISO réalisé par les bascules D.

### Registre SIPO

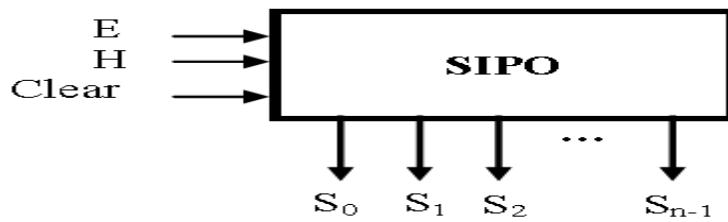


Figure II.21. : Registre SIPO

Le registre SIPO peut être réalisé par les bascules D comme suivant :

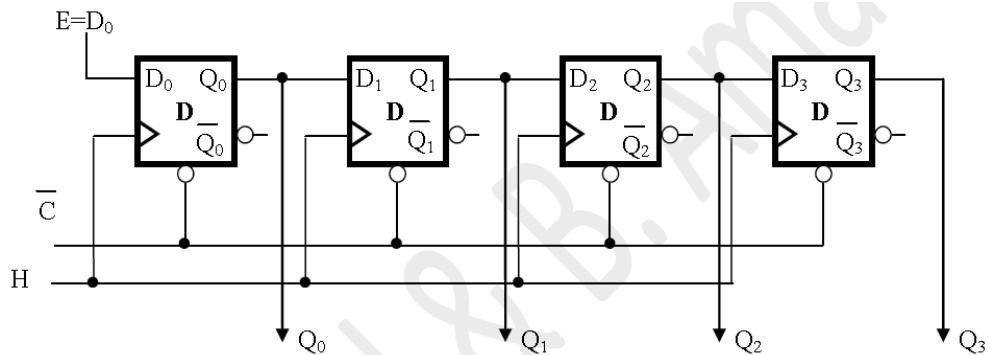


Figure II.22 : Registre SIPO réalisé par les bascules D.

Les deux types de registres (**PISO** et **SIPO**) sont utilisés dans les liaisons séries ; ils forment la base des modems. Par exemple, si on veut transmettre une information entre deux ordinateurs distants de quelques dizaines de mètres. Transmettre l'information en parallèle nécessite beaucoup de fils et très coûteux. La solution est alors d'utiliser un registre **PISO** pour envoyer

les bits sur une seule ligne. Au bout de laquelle, un registre **SIPO** reçoit ces bits et reconstitue des octets qui sont transmis à l'ordinateur de destination

### II.3. Registre universel

Les fabricants de circuits intégrés proposent dans leurs catalogues de circuits standards des registres dits « universels » regroupant les différentes fonctionnalités décrites dans les sections précédentes. Ces circuits permettent, en mode synchrone, de réaliser les fonctions suivantes :

Chargement parallèle ou série,

- | || lecture parallèle ou série,
- | || décalage à gauche ou à droite

La vue externe d'un tel registre est donnée par la figure

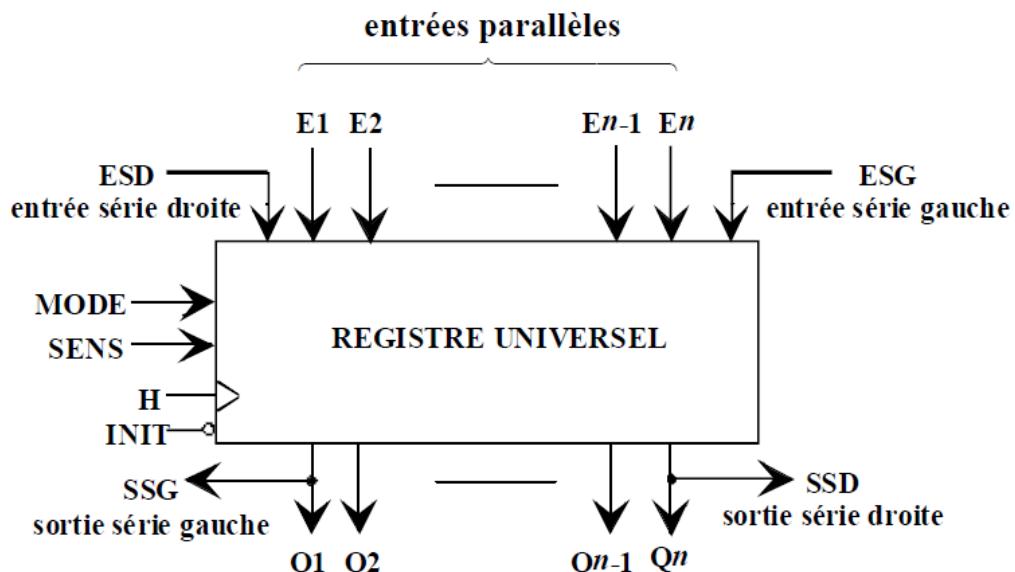


Figure II.23: vue externe d'un registre universel

Le mode de fonctionnement du registre est déterminé par deux **entrées de contrôle**, *MODE* et *SENS*. La valeur logique de l'entrée *MODE* indique le mode de fonctionnement du registre, décalage ou chargement parallèle. La valeur logique de *SENS* indique le sens du décalage, droite ou gauche. Les **entrées de données** prises en compte sont *ESD* pour un décalage à droite, *ESG* pour un décalage à gauche, *E1LEn* pour un chargement parallèle.

Le contenu du registre peut être lu en parallèle sur les sorties *Q1LQn* des bascules, ou en série sur *SSD* (*SSG*) dans le cas d'un décalage à droite (à gauche).

Toutes les entrées, excepté  $INIT$ , sont statiques, car leur état n'est pris en compte que sur le front actif de l'horloge  $H$ . L'entrée d'initialisation est dynamique car son activité est indépendante de celle de l'horloge (asynchrone). Elle est, par conséquent, prioritaire sur les autres entrées.

### Structure d'une cellule

La figure montre un exemple de réalisation d'une cellule du registre décrit ci-dessus, avec les Commandes suivantes :

- | ||  $MODE = 0$ , mode chargement parallèle (ou mode registre tampon),
- | ||  $MODE = 1$ , mode décalage,
- | ||  $SENS = 1$ , décalage à droite,
- | ||  $SENS = 0$ , décalage à gauche

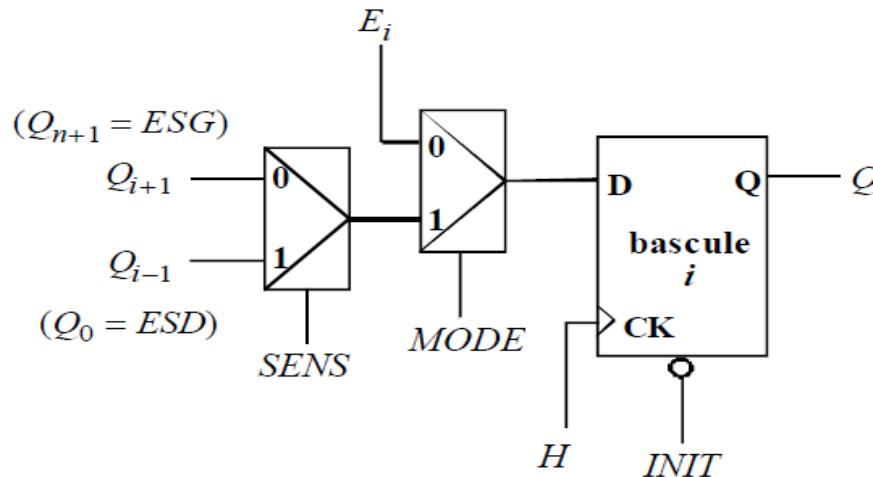


Figure II.24 : détail d'une cellule de registre universel

## II.3. Les compteurs [03]

### I.3.1. La fonction de comptage

Un compteur est un circuit séquentiel permettant d'établir une correspondance univoque entre le nombre d'impulsions appliquées sur son entrée d'horloge et l'état de sortie correspondant. Les opérateurs de base d'un compteur sont les bascules à déclenchement sur front. L'état du comptage à chaque instant est donné par la sortie des bascules. Un compteur constitué de  $n$  bascules peut délivrer au plus  $2^n$  combinaisons de sortie. Les compteurs les plus courants énumèrent le codage binaire naturel.

#### Le diviseur par 2

La fonction de comptage la plus élémentaire, comptage sur 1 bit, est obtenue en connectant la sortie complémentaire  $Q^*$  avec l'entrée d'une bascule D comme indiqué sur la figure 25. Le chronogramme présenté part de la condition initiale  $Q = 0$ , soit  $D = 1$ . Au premier front actif (montant, dans ce cas) de l'horloge  $H$ , la valeur de  $D = Q^* = Q$  est recopiée sur  $Q$  qui s'inverse donc. La sortie  $Q$  change ainsi de valeur à chaque front actif de l'horloge. Ce montage est plus couramment appelé diviseur par 2 car les sorties des bascules  $Q$  et  $Q^*$  sont des signaux de fréquence  $f / 2$ , si  $f$  est la fréquence de l'horloge  $H$ . Ces signaux sont, de plus, de **rapport cyclique** 1/2 car les durées des niveaux haut et bas sont égales

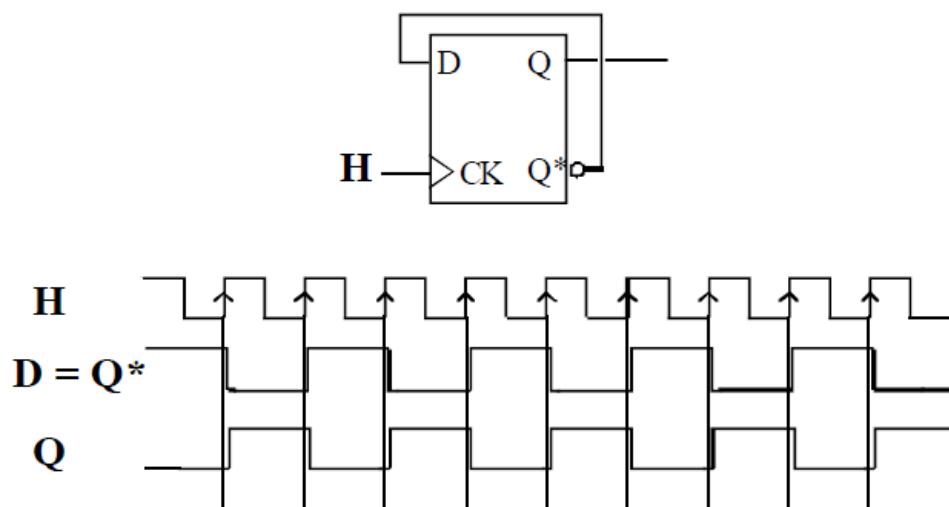


Figure II.25: bascule montée en diviseur par 2

#### . Notion de l'Etat :

La combinaison de sortie d'un compteur est appelé, exemple : 0000, 01010, 1010,1111.

### . Notion du modulo :

C'est le nombre d'états occupés par le compteur pendant un cycle complet, le modulo d'un compteur à N bits ( N bascules ) est  $2^N$ , Exemple le modulo d'un compteur à 4 bits = $2^4 = 16$ , on dit un compteur modulo16.

## II.3.2. Comptage synchrone / asynchrone

Les compteurs sont classés en deux catégories suivant leur mode de fonctionnement et leur structure. On distingue :

- Les compteurs **asynchrones**
- Les compteurs **synchrone**s

Un système séquentiel est **synchrone** lorsque tous les changements d'états du système sont liés à l'activité du même signal d'horloge (cf. § 2.4). Si cette condition n'est pas vérifiée, le système est dit **asynchrone**. Par exemple, les registres à décalage et le diviseur par 2 sont des circuits synchrones.

### II.3.2.1 Les compteurs asynchrones

Un compteur asynchrone possède un signal d'horloge, mais celui-ci ne sert qu'au déclenchement de la première bascule. Le signal d'horloge des bascules suivantes résulte d'une combinaison logique des sorties des autres bascules. La structure des compteurs asynchrones permet de propager en cascade l'ordre de changement d'état des bascules

#### II.3.2.1.1 Compteurs binaires asynchrones à cycles complets

Un compteur binaire à cycle complet sur  $n$  bits est constitué de  $n$  bascules et permet d'énumérer dans l'ordre les  $2^n$  valeurs du code binaire naturel. Les bascules sont montées en diviseur par 2 et cascadées comme suit : la première bascule est cadencée par l'horloge du circuit  $H$ , et chaque bascule suivante utilise comme horloge la sortie  $Q^*$  de la bascule précédente. A titre d'exemple, la figure présente un compteur asynchrone à cycle complet sur 3 bits. Les changements de valeur sur  $Q_2$  et  $Q_3$  sont déclenchés sur les fronts descendants respectifs de  $Q_1$  et  $Q_2$ . Le chronogramme de la figure a été tracé à partir de l'état initial 000. Le compteur énumère sur ses sorties le code binaire naturel sur 3 bits de manière cyclique.

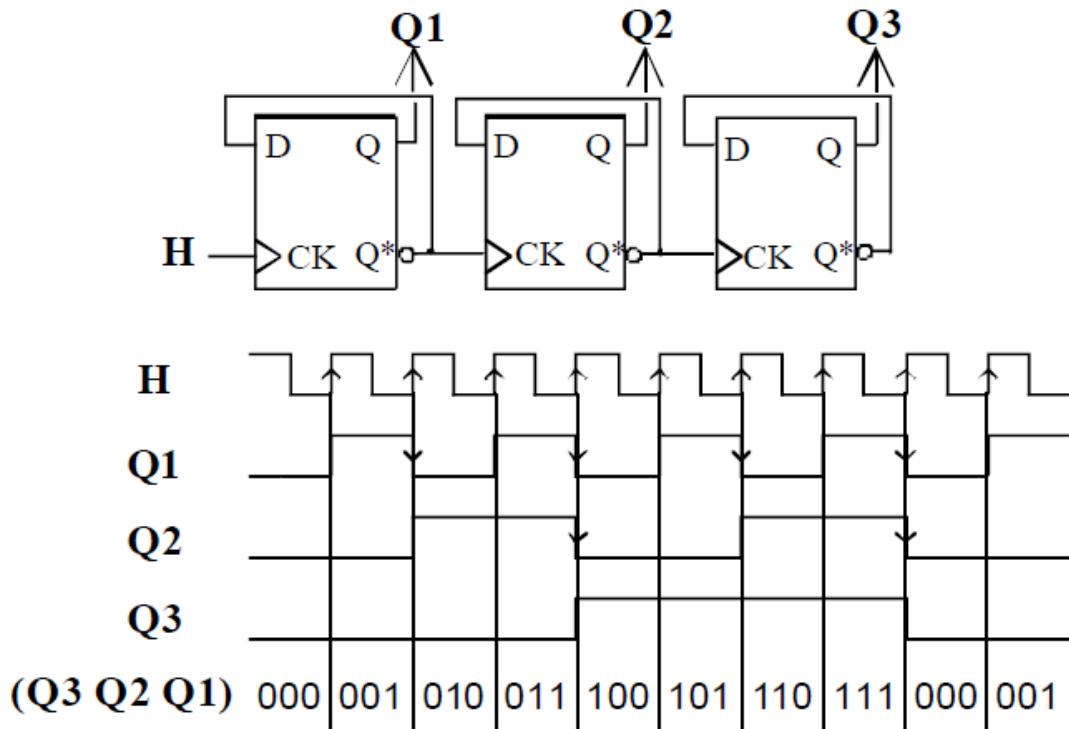


Figure II.26 : compteur asynchrone à cycle complet sur 3 bits

On notera que ce compteur réalise également les fonctions de division de fréquence par 2 (sorties  $Q_1$  et  $Q_1^*$ ), par 4 (sorties  $Q_2$  et  $Q_2^*$ ), et par 8 (sorties  $Q_3$  et  $Q_3^*$ ).

Le chronogramme de la figure 5.32 ne tient pas compte des temps de propagation dans les Opérateurs. En pratique, la structure cascadée du compteur induit un cumul des retards entre l'horloge  $H$  et les sorties. Les temps de propagation sur les 3 sorties sont donnés par :

$$\begin{aligned} t_p(H \rightarrow Q_1) &= t_p(CK \rightarrow Q)_{\text{bascule1}} \\ t_p(H \rightarrow Q_2) &= t_p(H \rightarrow Q_1^*) + t_p(Q_1^* \rightarrow Q_2) = t_p(CK \rightarrow Q^*)_{\text{bascule1}} + t_p(CK \rightarrow Q)_{\text{bascule2}} \\ t_p(H \rightarrow Q_3) &= t_p(H \rightarrow Q_1^*) + t_p(Q_1^* \rightarrow Q_2^*) + t_p(Q_2^* \rightarrow Q_3^*) \\ &= t_p(CK \rightarrow Q^*)_{\text{bascule1}} + t_p(CK \rightarrow Q^*)_{\text{bascule2}} + t_p(CK \rightarrow Q)_{\text{bascule3}} \end{aligned}$$

Les temps de propagation s'ajoutant d'une bascule à la suivante, les compteurs asynchrones de grande taille sont relativement lents. D'autre part, le décalage des sorties les unes par rapport aux autres produits des états transitoires indésirables après chaque front actif d'horloge.

#### II.3.2.1.2. Compteurs / décompteurs asynchrones à cycles incomplets

Les compteurs à cycles complets permettent un comptage binaire de 0 à  $2n - 1$ , si  $n$  est le nombre de bascules du compteur. Pour un comptage modulo  $N$ , où  $N$  n'est pas une puissance de 2, la méthode généralement utilisée est la suivante :

1. On réalise le compteur à cycle complet dont le modulo est immédiatement supérieur à  $N$ .

Ce compteur est constitué de  $n$  bascules, où  $n$  vérifie  $2n-1 < N < 2n$ . Par exemple, pour réaliser un compteur décimal (modulo 10), on part d'un compteur modulo 16 à 4 bascules.

2. On tronque son cycle par un rebouclage asynchrone : lorsque le cycle est terminé, le compteur revient à l'état 0...0 par activation des entrées dynamiques de remise à zéro (clear ou reset) des bascules. Il faut pour cela ajouter une fonction combinatoire de détection de l'état  $N$

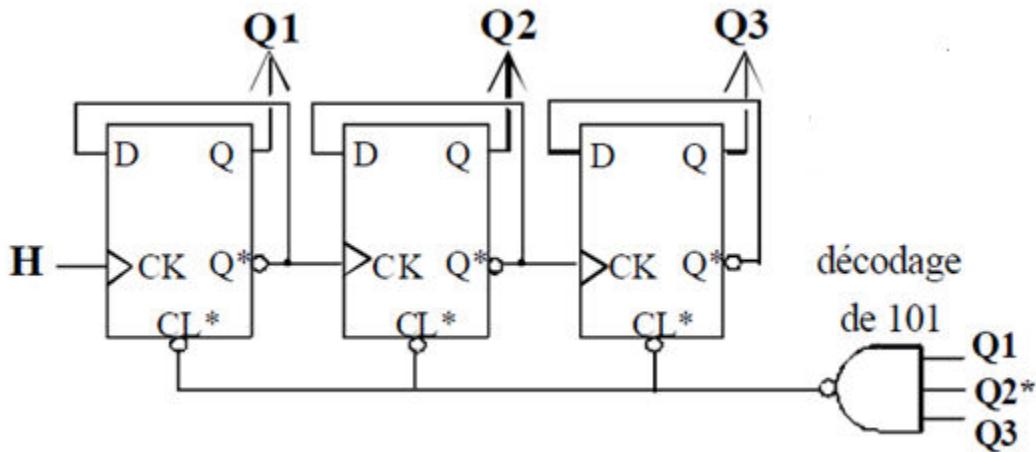


Figure II.27 : schéma fonctionnel du compteur asynchrone

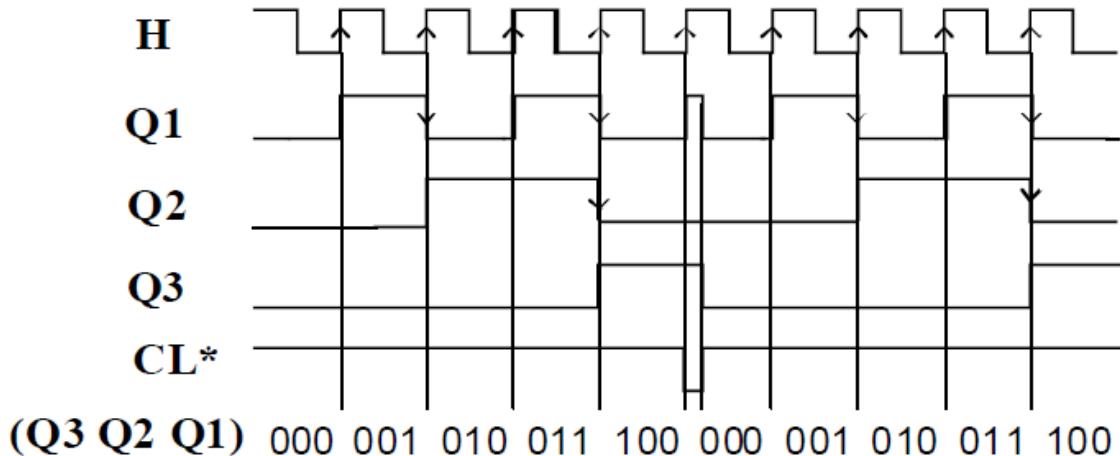


Figure II.28 : chronogramme du compteur asynchrone

## II.4. Exploitation des compteurs asynchrones

L'avantage principal des compteurs asynchrones réside dans leur simplicité de réalisation. En revanche, leur utilisation présente des inconvénients non négligeables :

- “ vitesse de fonctionnement limitée car les temps de propagation s’ajoutent,
- “ Présence d’états transitoires indésirables après chaque front actif de l’horloge,

En pratique, les compteurs à cycles complets sont parfois utilisés pour des applications ne nécessitant pas une fréquence d’horloge très élevée, mais les compteurs à cycles incomplets ne présentant pas une sûreté de fonctionnement suffisante, on leur préfère de loin l’utilisation de compteurs synchrones

## II.5 *Les compteurs synchrones*

La réalisation de compteurs synchrones permet d’éliminer l’ensemble des inconvénients cités ci-dessus. A cette fin, toutes les bascules du compteur doivent être synchronisées par la même horloge

La structure des compteurs synchrones est un peu plus complexe que celle des compteurs asynchrones. En revanche, il est possible de réaliser des compteurs couvrant une gamme beaucoup plus large de fonctions que pour les compteurs asynchrones. Une méthode systématique unique permet de construire facilement des compteurs à cycle complet, à cycle incomplet, des décompteurs, ou de mettre en œuvre des énumérations autres que le comptage binaire naturel.

La réalisation d’un compteur à  $N$  états nécessite, comme dans le cas des compteurs asynchrones,  $n$  bascules, avec  $2^n-1 < N < 2^n$ . La construction du compteur passe par l’écriture de sa table de transition, c’est-à-dire la table donnant pour chaque état du compteur l’état suivant correspondant. Cette table permet ensuite de calculer l’équation logique de l’entrée  $D$  de chaque bascule en fonction des sorties  $Q$  de l’ensemble des bascules. Nous allons traiter plusieurs exemples pour illustrer ce principe.

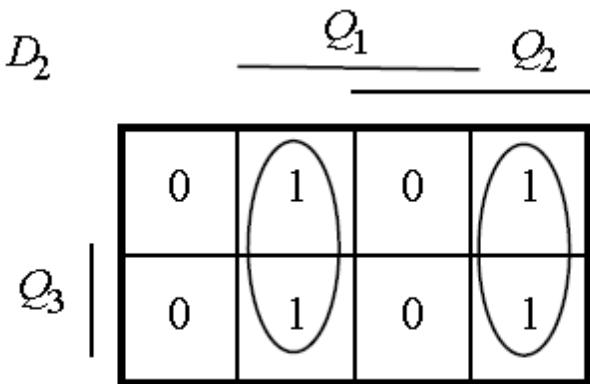
**Exemple :** On veut réaliser un compteur synchrone modulo 8 par trois bascules D

<b>Q3</b>	<b>Q2</b>	<b>Q1</b>	<b>Q+3</b>	<b>Q+2</b>	<b>Q+1</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>

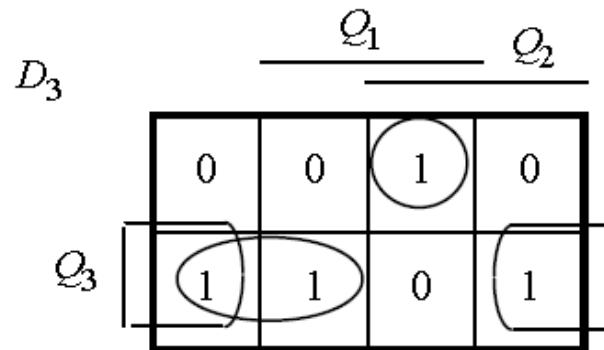
Tableau II.5 : Table de transition d'un compteur synchrone modulo 8

Le compteur est donc constitué de 3 bascules. L'expression des entrées  $D$  des bascules en fonction de  $Q_1$ ,  $Q_2$ , et  $Q_3$  est directement déduite de la table de transition précédente, car pour les bascules D,  $Q_i = D_i$ ,  $i \in \{1,2,3\}$ . Ces expressions peuvent être simplifiées soit directement, soit en utilisant la méthode de Karnaugh.

- L'obtention de  $D_1$  est immédiate :  $D_1 = Q_1$
- Pour  $D_2$  et  $D_3$ , on trace les diagrammes de Karnaugh :



$$\text{On en déduit : } D_2 = Q_1 \bar{Q}_2 + \bar{Q}_1 Q_2 = Q_1 \oplus Q_2$$



On déduit :  $D_3 = \overline{Q_2}Q_3 + \overline{Q_1}Q_3 + Q_1Q_2\overline{Q_3}$ . cette expression peut être écrite sous la forme :

$$D_3 = (\overline{Q_1} + \overline{Q_2})Q_3 + Q_1Q_2\overline{Q_3} = \overline{Q_1}\overline{Q_2}Q_3 + Q_1Q_2\overline{Q_3} = Q_1Q_2 \oplus Q_3$$

La structure du compteur correspondant est donnée en figure 5.35. On a utilisé au maximum les sorties  $Q^*$  des bascules pour minimiser le nombre de portes logiques nécessaires à la réalisation de la fonction «état suivant ». Notamment, la fonction ET ( $Q_1 . Q_2$ ) est réalisée à l'aide d'une porte NOR moins encombrante qu'une porte ET en logique CMOS

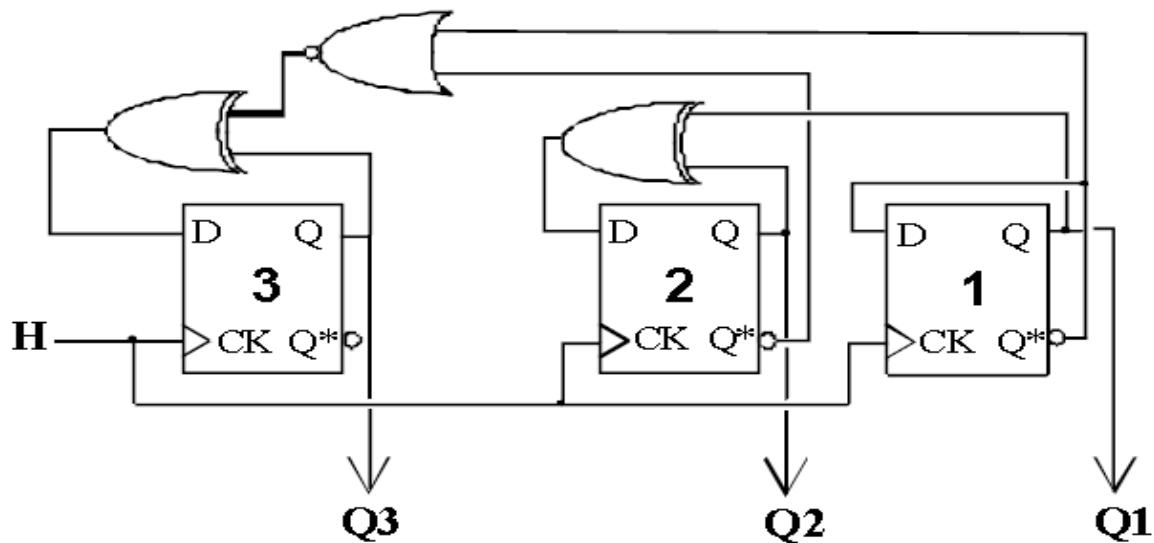


Figure II.28 : schéma fonctionnel du compteur synchrone

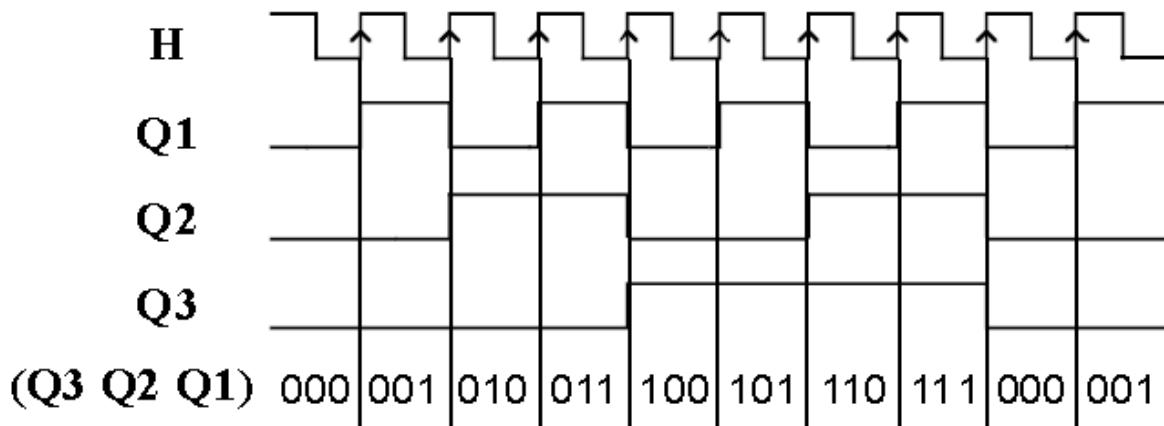


Figure II.29 : chronogramme du compteur synchrone

Le chronogramme obtenu est globalement le même que celui d'un compteur asynchrone, mais toutes les sorties du compteur sont directement obtenues à partir des fronts actifs d'horloge, si bien que, si les temps de propagation des bascules sont identiques, toutes les sorties du compteur commutent en même temps.

### II.5.2. Compteur binaire synchrone à cycle incomplet : compteur modulo 5

Le principe de réalisation d'un tel compteur est la même que celle appliquée précédemment. La table de transition réduite du compteur binaire modulo 5 est la suivante :

Q3	Q2	Q1	Q+1	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	0	0	0

Tableau II.6: table de transition réduite d'un compteur binaire modulo 5

Le compteur est donc constitué de 3 bascules, tout comme le compteur modulo 8. La méthode

d'obtention des expressions des entrées D des bascules en fonction de Q1 , Q2 , et Q3 est identique. On pourra utiliser les états indifférents des tables de Karnaugh lors de la simplification.

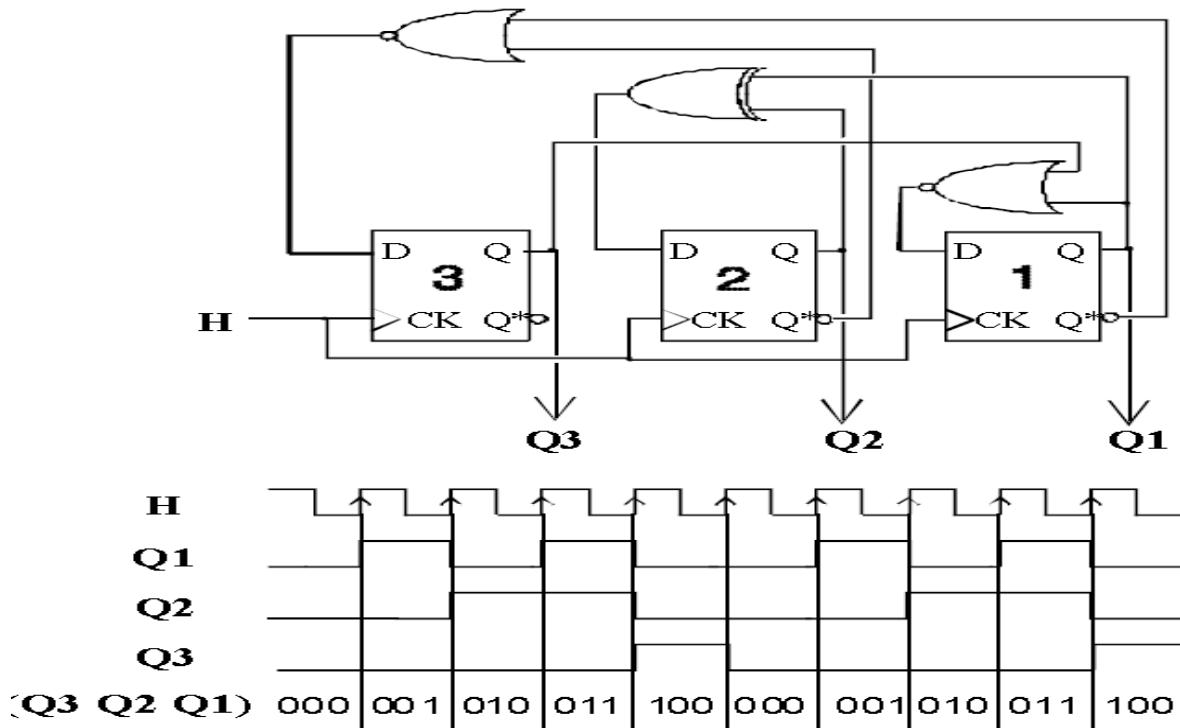


Figure II.30 : Schéma et chronogramme du compteur synchrone incomplet

## II.6. Les compteurs programmables

Les fabricants de circuits intégrés proposent dans leurs catalogues de circuits standard des compteurs qui suivent le code binaire pur et qui possèdent en général les fonctionnalités suivantes :

- Chargement, en général synchrone, d'un état dans le compteur,
- Commande de validation/inhibition du comptage,
- Programmation du sens du comptage (comptage ou décomptage),

Ces circuits sont dits « compteurs programmables ». La figure montre les principales entrées / sorties d'un tel compteur

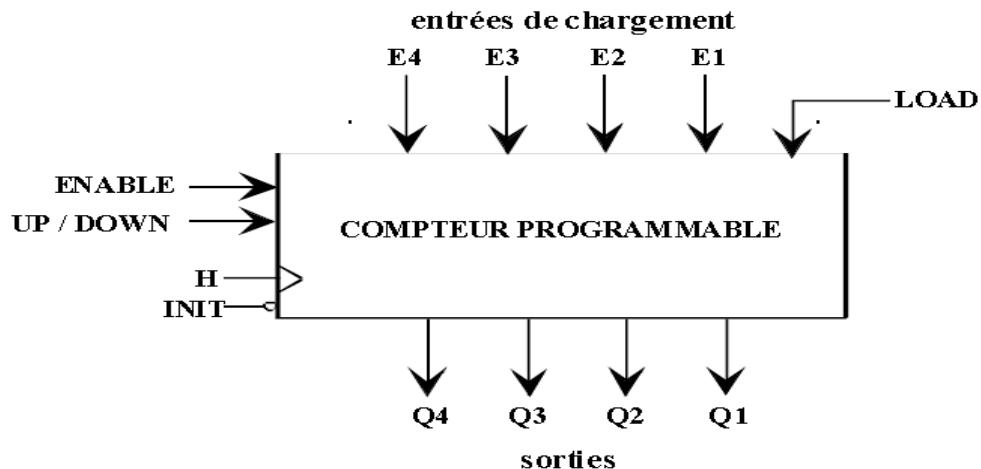


Figure II.31 : compteur programmable

La figure ...montre un exemple de réalisation d'une cellule d'un tel compteur, avec les entrées de commande suivantes :

- □ □  $LOAD = 1$ , mode chargement parallèle,
- □ □  $LOAD = 0$ , mode comptage
- □ □  $ENABLE = 0$ , comptage inhibé,
- □ □  $ENABLE = 1$ , comptage validé,
- □  $UP/DOWN = 1$ , mode comptage,
- □  $UP/DOWN = 0$ , mode décomptage.

Le chargement synchrone à l'aide des entrées  $LOAD$  et  $Ei$  est similaire au chargement parallèle synchrone d'un registre à décalage (cf. 4.3.3). Parfois, certains compteurs possèdent également une entrée de remise à zéro synchrone, qui permet le chargement synchrone direct de l'état 0...0, sans utiliser les entrées  $Ei$ .

L'entrée  $ENABLE$  valide le comptage : si elle est active, le comptage se déroule normalement, sinon il y a rebouclage de la sortie  $Q$  de chaque bascule sur son entrée  $D$ .

Pour un compteur / décompteur, les deux fonctionnalités sont implantées dans le circuit, et l'entrée de programmation  $UP/DOWN$  permet de sélectionner la fonction logique correspondante à appliquer sur l'entrée de chaque bascule.

En général, les circuits programmables possèdent également une sortie de détection de l'état final du comptage (tous  $Qi$  à 1)

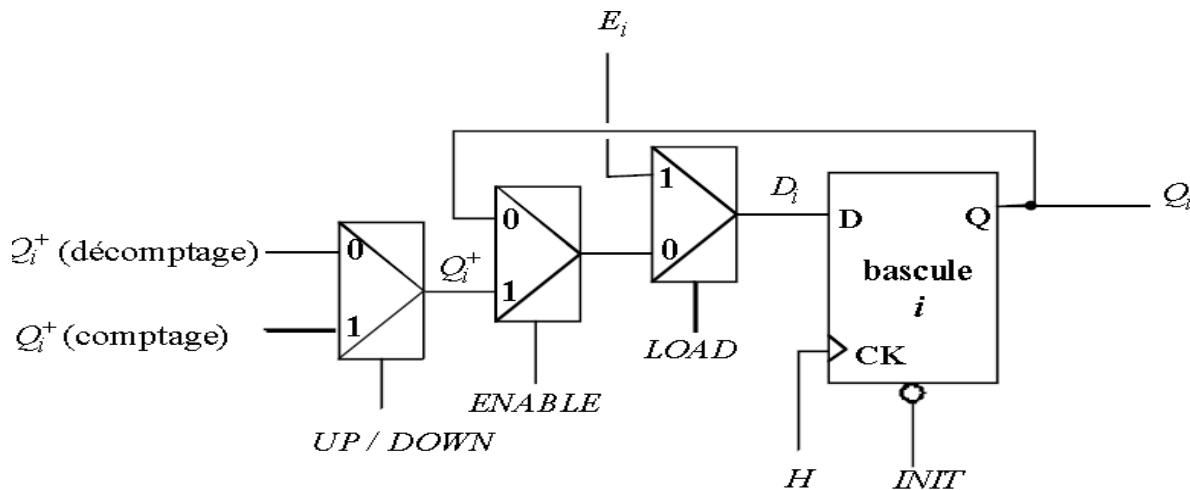


Figure II.32 : cellule d'un compteur programmable

## II.5. Applications des compteurs

Les compteurs sont très couramment utilisés en logique séquentielle. Leurs applications les plus courantes sont les suivantes :

- Comptage d'événements (comptage d'impulsions sur une entrée par exemple).
- Division de fréquence : Dans un compteur binaire à cycle complet, la bascule de rang  $n$  produit en sortie un signal de fréquence  $f / 2^n$ , de rapport cyclique 1/2, où  $f$  est la fréquence de l'horloge du compteur. Pour obtenir des divisions de fréquence de rapports différents de  $2^n$ , on peut utiliser un compteur à cycle incomplet et décoder un état particulier. Par exemple, dans le cas du compteur modulo 5, le décodage par exemple de l'état 000 (à l'aide d'une simple porte NOR à 3 entrées) permet de produire un signal passant à 1 pendant une période de l'horloge sur 5. C'est donc un signal de fréquence  $f / 5$ , mais de rapport cyclique 1/5 et non 1/2.
- Adressage de mémoires par exemple pour la réalisation d'un séquenceur

## II.6. Conclusion

Dans ce chapitre nous avons décris les registres et les compteurs comme des structure de base dans un système séquentiel se caractérisant par la mémorisation et la séquence des données, c'est pourquoi le circuit séquentiel représente une introduction indispensable pour la conception du système à microprocesseur.

## Chapitre III

## Les mémoires à semi-conducteurs

### II.1. Introduction

Contrairement aux systèmes analogiques, les systèmes numériques permettent de mémoriser des informations avant de les traiter, La mémoire est un dispositif électronique qui sert à stocker des informations traitées par le microprocesseur, elle est principalement caractérisée par sa capacité exprimée par un nombre de bits, octets ou ses multiples ( Koctets , Mega-octets , Gega-Octets ) contenant dans un boitier mémoire.

### II.2. Différentes technologies des mémoires

On trouve trois types technologiques de mémoires :

- a- Mémoires magnétiques : sont des mémoires en volume très grande comme les disques durs
- b- Mémoires optiques ; sont des mémoires optiques comme CDROM , DVD leurs lecture / écriture sont effectuées par un signal optique délivré par une diode laser
  - CD-ROM : acronyme en anglais Compact Disc Read Only Mémory est un disque qui permet de stocker 650 MO (qui correspond à 450 disquette haute densité de 1.44 MO ).
  - DVD : acronyme Digital Versatile Disc ou disque numérique polyvalent destiné à la vidéo avec une grande capacité que le CDROM
- c- Les mémoires à semi-conducteurs : elles sont des mémoires sous forme des circuits intégrés caractérisées par leur rapidité mais avec une capacité réduite [04]

### II.3. Notion d'une mémoire à semi-conducteur :

Une mémoire est un circuit intégré à semi-conducteur chargé d'enregistrer, de conserver et de restituer des informations, d'une autre façon elle est un arrangement des cases mémoires, chaque case mémoire peut contenir une donnée, la donnée peut être sous un nombre de bits. Les opérations possibles sur la mémoire sont : la lecture et l'écriture .

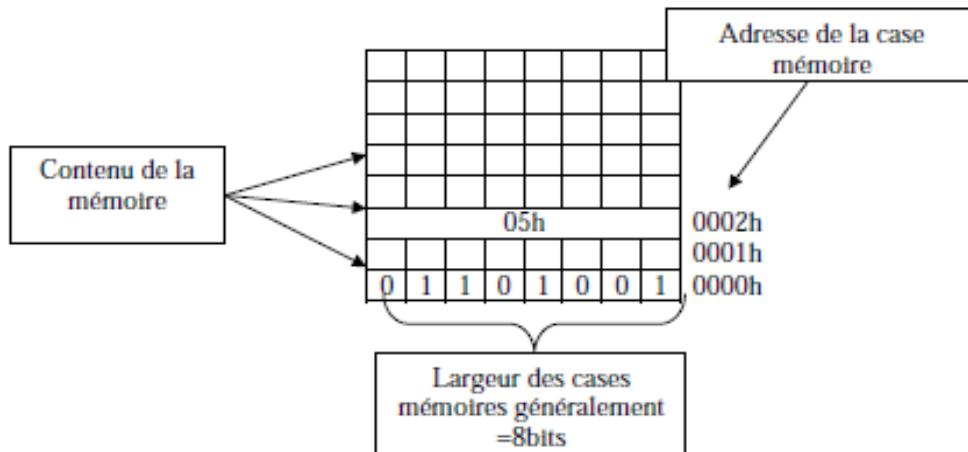


Figure III.33 : structure de la mémoire

Un boitier mémoire doit être connecté avec le microprocesseur par un nombre des lignes comme représenté su le schéma :

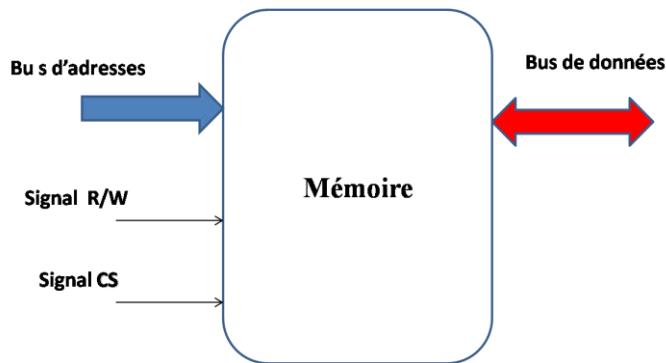


Figure III.34 : Schéma fonctionnel d'une mémoire.

La mémoire possède une broche CS (chip select) de sélection lorsque cette broche est active (état bas) .

#### II.4. Les mémoires à semi-conducteurs [05] :

La mémoire à semi-conducteur est circuit intégré permettant de sauvegarder les données sous forme des bits ou octets, on trouve deux types des mémoires à semi-conducteurs :

##### II.4.1. La mémoire vive ou RAM

La mémoire RAM (Random Access memory) ou mémoire à accès aléatoire, cette mémoire sert à sauvegarder les informations temporairement en fonction de l'alimentation électrique, elle perd ses données lors de coupure d'électricité, elle est dite volatile, cette mémoire est importante au

système à microprocesseur, sa capacité influe sur l'exploitation et le traitement des données au cours de l'opération du système, on distingue deux types : RAM statique et RAM dynamique.

- RAM statique : une RAM fabriquée par un nombre entre 4 et 6 transistors, permettant de stocker les données en quelques bits.
- RAM dynamique : une RAM fabriquée par des capacités MOS et permet de conserver les données en grande volume.



Figure III.35: Image d'une barrette de mémoire RAM.

#### II.4.2. Mémoire ROM :

La mémoire ROM (Read Only Memory) ou mémoire à lecture seule, cette mémoire sert à conserver les informations d'une manière définitive en cas d'une coupure de l'alimentation électrique, elle est utilisée uniquement pour la lecture, elle est dite mémoire non volatile,



Figure III.36: image de mémoire ROM

On trouve plusieurs types des mémoires ROM à savoir :

- Mémoire ROM : Mémoire Read only Memory.
- Mémoire PROM : Mémoire Programmable ROM
- Mémoire EPROM : Mémoire Erasable PROM

- Mémoire EEPROM: Mémoire Electrically EPROM
- Mémoire FLASH-EPROM.
- Mémoire UVPROM: Ultra-Violet PROM.

## II.5. Caractéristiques générales d'une mémoire :

Une telle mémoire se caractérise par les paramètres suivants :

- La capacité : c'est le nombre total de bits que contient la mémoire, la capacité est donnée par la formule :  $C = 2^n \cdot k$  , n est le nombre des adresses, k est le nombre de données  
Exemple : une mémoire de 12 lignes d'adresses et 8 lignes de données, sa capacité  $C = 2^{12} \cdot 8 \text{ Bits} = 4k \text{ octet}$
- Le temps d'accès : c'est le temps qui s'écoule entre l'instant où a été lancée une opération (lecture /écriture) en mémoire et celui où la mémoire place l'information demandée sur le bus de données. Ce temps varie entre 50ns (mémoire rapide) et 300ns (mémoire lente).
- Le temps de cycle : c'est l'intervalle de temps minimum séparant deux demandes successives lecture ou écriture.
- Volatilité : elle caractérise la permanence des informations dans la mémoire, si les données sont stockées instantanément pendant que le système numérique est alimenté par l'électricité , la mémoire est dite volatile, si les informations sont sauvegardées définitivement après l'absence de l'électricité , la mémoire est dite non volatile.

## III.6. Organisation interne d'une mémoire [06]:

Une mémoire est constituée d'un ensemble de registres dont l'un est en relation avec l'extérieur soit pour sa lecture (ROM), soit pour sa lecture ou son écriture (RAM), la taille de registre définit le format de la mémoire soit 8 bits en général et le nombre de registres détermine la capacité de la mémoire.

Pour limiter au maximum l'encombrement du composant, on utilise généralement les mêmes pattes pour les entrées (Input) et les sorties (Output) de la RAM. On gagne ainsi un facteur 2 sur le nombre de broches dédiées aux données, mais il faut bien sûr ajouter une entrée de contrôle supplémentaire pour spécifier si la RAM fonctionne en mode lecture ou mode écriture (entrée OE pour Output Enable).

Pour choisir un registre en contact avec l'extérieur, on doit faire un décodage d'adresse écrit sous forme d'un nombre de bits exemple A<sub>0</sub> A<sub>1</sub> A<sub>2</sub> A<sub>3</sub> A<sub>4</sub> A<sub>5</sub> , le nombre de bits d'adresse détermine la capacité de mémoire sous la relation = 2<sup>n</sup> x 8 bits=2<sup>6</sup> octets= 64octets.

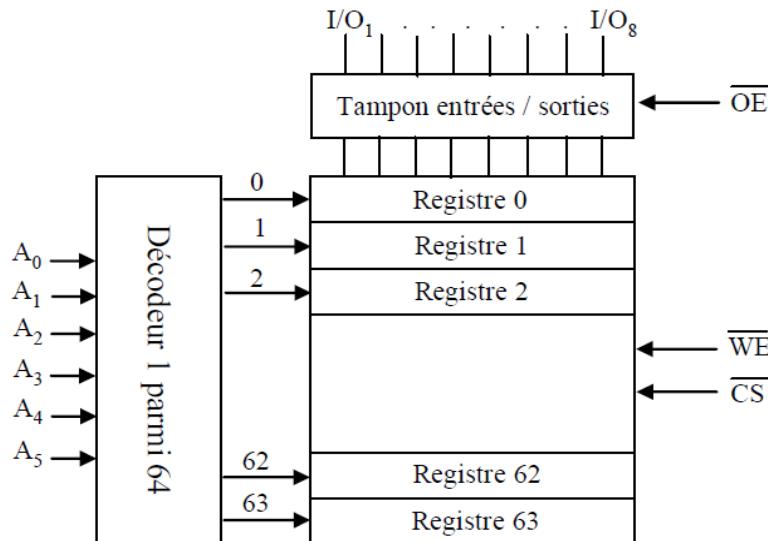


Figure III.37 : organisation d'une mémoire

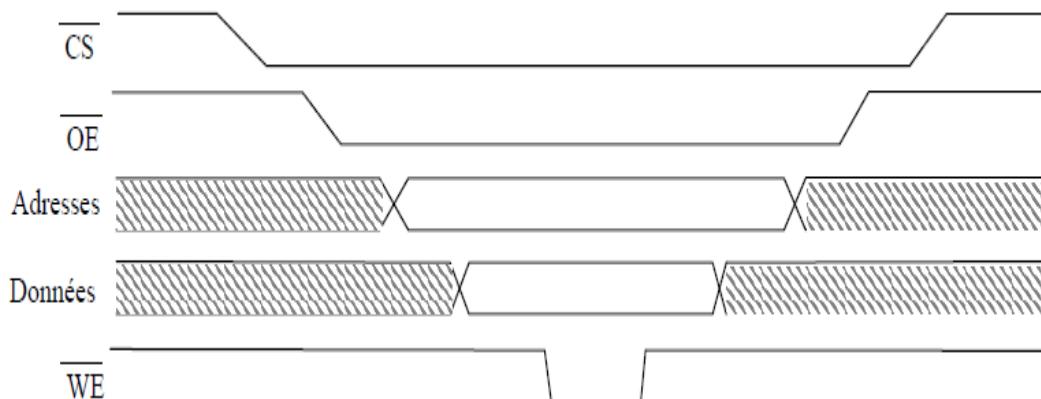


Figure III.38 : cycle de fonctionnement en mode écriture d'une mémoire RAM.

### III.7. Critères du choix d'une mémoire

Une mémoire de stockage est sélectionnée suivant quatre critères lesquels :

- La capacité : c'est la taille de stockage que le microprocesseur permet de conserver soit en mode écriture comme la mémoire RAM ou en mode de lecture en cas de mémoire ROM.
- La vitesse : la vitesse la rapidité d'accès aux informations où le temps d'accès à la mémoire doit être plus court.

- La consommation électrique : la consommation d'énergie représente un facteur primordial permettant au système à microprocesseur de gaspiller l'énergie lors de fonctionnement.
- Le cout : le cout se détermine grâce à les caractéristiques , les performances de la mémoire et la technologie de fabrication.

### III.8. Extension de la mémoire.

Le système à microprocesseur a besoin une grande capacité de mémoire pour conserver les données ou les informations, pour cette mission, il est nécessaire d'organiser les boîtiers mémoires suivant un décodage d'adresses et d'éliminer au même temps la redondance.

Dans ce cas, on veut concevoir une capacité de mémoire à partir un boîtier mémoire élémentaire.

**Exemple :** On veut réaliser une mémoire de **capacité C** , mais nous disposons uniquement de boîtiers ( des circuits ) de **taille inférieur c**.

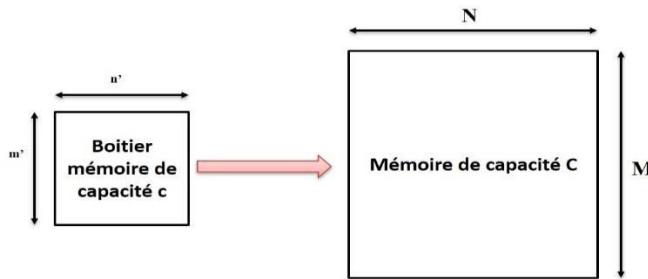


Figure III.39 : Construction d'un espace mémoire

Soit  $M$  une mémoire de capacité  $C$  , tel que  $m$  est le nombre de mot et  $n$  la taille d'un mot.

Soit  $M'$  un boîtier de capacité  $C'$  , tel que  $m'$  le nombre de mot et  $n'$  la taille d'un mot.

On suppose que  $C > C'$  ( $m \geq m'$  ,  $n \geq n'$ )

- Quel est le nombre de boîtiers  $M$ ' nécessaire pour réaliser la mémoire  $M$  ?

Pour connaître le nombre de boîtiers nécessaire, il faut calculer les

Deux facteurs suivants :  $P = m/m'$        $Q = n/n'$

$P$  : permet de déterminer de nombre de boîtiers  $M$ ' nécessaire pour obtenir le nombre de mots de la mémoire  $M$  (extension lignes).

- $Q$  : permet de déterminer le nombre de boîtier  $M$ ' nécessaire pour obtenir la taille de mot de la mémoire  $M$  (extension mots ou extension colonnes).
- $P.Q$  donne le nombre total de boîtiers nécessaire pour réaliser la mémoire  $M$ .

**Exemple :**

Réaliser une mémoire de 1Ko ( la taille d'un mot est de 8 bits) en utilisant des boîtiers de taille 256 mots de 8 bits ? Donner le schéma.

**Solution :**

$(m,n)=(1024,8)$  \_ taille du bus d'adresses est de 10 bits A9 0(A9...A0), taille du bus de données est de 8 bits D70(D7....D0)

$(m',n')=(256,8)$  \_ taille du bus d'adresses est de 8 bits (A7'...A0'), taille du bus de données est de 8 bits (D7'....D0')

- Calculer les deux facteurs d'extension lignes et colonnes :

$$P= m/m' =1024/256=4 \text{ ( extension lignes )}$$

$$Q= n/n' =8/8=1 \text{ (extension colonnes)}$$

- Le nombre totale de boîtiers  $P.Q=4$ .

Le décodeur permet d'activer les quatre mémoires en sélectionnant une seule mémoire pour chaque combinaison binaire.

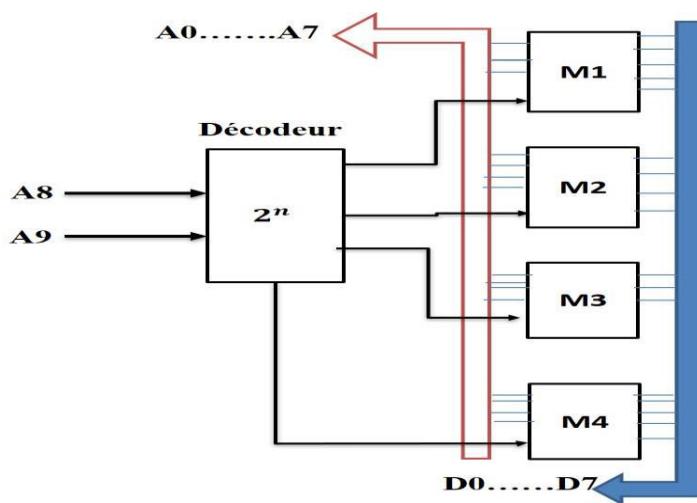


Figure III.40 : extension de mémoire.

**III.9.Conclusion :**

Le chapitre met la lumière sur les mémoires à semi-conducteur et leurs caractéristiques en indiquant le cycle du fonctionnement selon le type mémoire RAM ou ROM, en revanche la technologie des mémoires à semi-conducteur reste à développer à cause de la nécessité du traitement des données et leurs exploitations dans plusieurs domaines d'application.

## Chapitre IV

## Historique et évolution des ordinateurs

### IV.1. Introduction :

Le but de la révolution industrielle était de remplacer l'homme sur les travaux physiques. Cette technologie était principalement orientée vers la transmission, la manipulation et le contrôle de la force. Une technologie parallèle qui a trait à la manipulation et la transmission de l'information est venue la seconder. Cette dernière a pour but de suppléer l'homme sur la monotonie de l'exercice mental ; l'invention des ordinateurs permet de traiter les informations dans n'importe quel d'activité humaine et offre une grande potentialité de résoudre nombreux problèmes dans la vie.

### IV.2. Historique :

L'invention des ordinateurs est provenue d'un chemin scientifique plus longue depuis la préhistoire avant J.C (Jésus Christ), l'homme a commencé à compter avec ses doigts, des bouts de bois et des cailloux, à l'époque beaucoup d'instruments à compter ont existé, l'art de compter n'est pas apparu subitement, il a été progressif, des artisans, des inventeurs et des scientifiques ont mis les jalons nécessaires ont donné naissance aux premiers ordinateurs. Les machines mécaniques à calculer sont peut-être les premiers outils et les plus importantes découvertes conduisant au développement des ordinateurs, Mais le premier instrument aidant l'homme dans les tâches répétitives de comptage sont les tablettes, les bouliers et les abaques. Déjà 3000 ans avant J. C., ces instruments étaient utilisés par les Chinois. Mille ans après, les abaques étaient utilisés par les Babyloniens et en 460 avant J.C. les Egyptiens l'adoptèrent. L'abaque à boule est une pièce en bois sur laquelle des cordes parallèles contiennent des perles qui peuvent glisser le long d'une tige. Selon une méthode de programmation que l'utilisateur doit mémoriser, Toutes les opérations arithmétiques ordinaires pouvaient y être effectuées. L'abaque est toujours utilisé dans certains pays asiatiques.

Vers le VIII<sup>e</sup> siècle, Mohamed Ybn Moussa al-Khawarizmi passe pour être le père de la théorie des algorithmes ainsi que de l'algèbre (de l'arabe Al-jabr signifiant « compensation »)

Une horloge à calculer a été dessinée en 1623 par Wilhelm Schickard (1592-1635) professeur d'astronomie et de mathématiques à l'université de Heidelberg

Avant la seconde guerre mondiale plusieurs inventions dans calculateurs qui utilisaient des quantités physiques, telles que la tension, le courant ou la vitesse de rotation des axes, pour représenter les nombres. Ainsi, ils devaient être reprogrammés manuellement à chaque nouveau problème. Leur avantage par rapport aux premiers ordinateurs numériques était leur capacité à traiter des problèmes plus complexes, avec une certaine forme de parallélisme [07] ,[08].

### **Première génération d'ordinateurs (1936-1956)**

Durant cette période, l'invention des circuits électroniques, tubes à vide, condensateurs et relais remplacèrent leurs équivalents mécaniques et le calcul numérique remplaça le calcul analogique. Les ordinateurs conçus à cette époque forment la première génération d'ordinateur

Vers 1954, les mémoires magnétiques (tores de ferrite pour la mémoire vive, bandes, ensuite disques amovibles puis fixes pour la mémoire de masse) supplantèrent toute autre forme de stockage et étaient dominantes au milieu des années 1960.

De nombreuses machines électromécaniques furent construites avec des capacités diverses. Elles n'eurent qu'un impact limité sur les constructions à venir.

En novembre 1939, John Vincent Atanasoff et Clifford E. Berry achevèrent l'ABC (Atanasoff Berry Computer). Composé de lampes et de tambours pour la mémoire, il est construit pour résoudre des systèmes d'équations linéaires. Bien que n'étant pas programmable, il est basé sur trois idées propres aux ordinateurs modernes :

- L'utilisation du système binaire (plus fiable et plus simple à mettre au point que le système décimal),
- La séparation entre le calcul et la mémoire ;
- L'utilisation de composants électroniques plutôt que des éléments mécaniques pour réaliser les calculs.

### **Deuxième génération (1957-1965)**

La deuxième génération d'ordinateurs est basée sur l'invention du transistor en 1947. Cela permit de remplacer le fragile et encombrant tube électronique par un composant plus petit et fiable. Les ordinateurs composés de transistors sont considérés comme la deuxième génération et ont dominé l'informatique dans la fin des années 1950 et le début des années 1960. Toutefois la notion de « génération », qui est à l'origine un argument commercial, est contestée par les historiens : elle ne tient compte que des technologies de l'unité logique, non des mémoires, de l'architecture ou de la programmation.

En 1955, Maurice Wilkes inventa la microprogrammation, désormais universellement utilisée dans la conception des processeurs. Le jeu d'instructions du processeur est défini par ce type de programmation.

Ramac 305 à l'arsenal de Red River de l'U.S. Army. Au premier plan : deux lecteurs de 350 disques ; au fond : console 380 et unité de calcul 305

En 1956, IBM sortit le premier système à base de disque dur, le Ramac 305 (Random Access Method of Accounting and Control). L'IBM 350 utilisait 50 disques de 24 pouces en métal, avec 100 pistes par face. Il pouvait enregistrer cinq mégaoctets de données et coûtait 10 000 \$ par mégaoctet

### **Troisième génération (1963-1971)**

La troisième génération d'ordinateurs est celle des ordinateurs à circuit intégré qui ont été inventés par Jack Kilby en 1958. C'est à partir de cette date que l'utilisation de l'informatique a explosé.

Les premiers ordinateurs utilisant les circuits intégrés sont apparus en 1963. L'un de leurs premiers usages a été dans les systèmes embarqués, notamment par la NASA dans l'ordinateur de guidage d'Apollo et par les militaires dans le missile balistique intercontinental LGM-30. Le circuit intégré autorise alors le développement d'ordinateurs plus compacts que l'on appelle les mini-ordinateur

### **Quatrième génération (1971 à la fin des années 1980)**

Une définition non universellement acceptée associe le terme de quatrième génération à l'invention du microprocesseur par Marcian Hoff et Federico Faggin (physicien et inventeur italien, spécialisé en physique du solide). En pratique et à la différence des autres changements de génération, celle-ci constitua plus une évolution (presque passée inaperçue) qu'une révolution : les circuits s'étaient miniaturisés de plus en plus depuis l'invention du circuit intégré.

## **IV.3. Organisation d'un ordinateur**

Un ordinateur est tout système électronique à base de microprocesseur permettant le contrôle et le traitement des informations dans un processus industriel ou dans un équipement domestique, on trouve l'ordinateur dans plusieurs domaines ou applications tels que les robots, véhicules, machine à laver, automate programmable, Télécommunications, PC, ...etc. Plusieurs synonymes

utilisés au un micro-ordinateur comme calculateur, machine à calculer, système à micro-ordinateur, en anglais les mots utilisés sont (computer, laptop, system-processor).

Un ordinateur ou système à microprocesseur est généralement composé des éléments suivants :

- D'un microprocesseur ou une unité centrale du traitement (CPU : central processing Unit)
- Une mémoire centrale constituée d'un ensemble des mémoires RAM et ROM.
- D'une unité d'entrée / sortie (interfaces) pour communiquer avec le mode extérieur.
- Unité de transfert de transmission ( Bus de transmission ).
- Des périphériques externes (écran, clavier, disque dur, imprimante ,.....etc).

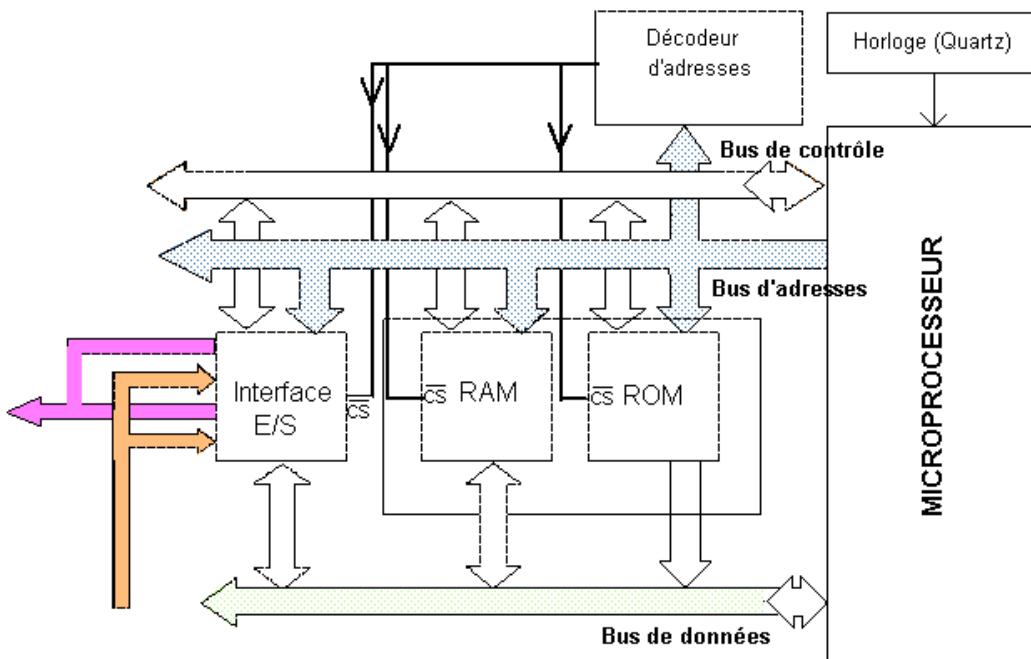


Figure IV.41: organisation d'un système à microprocesseur [09]

#### IV.3.1. Microprocesseur :

Un microprocesseur est un circuit intégré complexe caractérisé par une très grande intégration et doté des facultés d'interprétation et d'exécution des instructions d'un programme. Il est chargé d'organiser les tâches précisées par le programme et d'assurer leur traitement. C'est le cerveau du système. Il est parfois appelé CPU (Central Processing Unit) et est caractérisé par sa fréquence d'horloge (en MHz), sa largeur des bus de données et d'adresse, sa mémoire adressable, le nombre de transistors 1 et la taille de la gravure (en microns). Ses performances sont

caractérisées par le CPI (Cycle Par Instruction) et le MIPS (Millions d'Instructions Par Secondes) qui dépendent directement de la fréquence d'Horloge et son jeu d'instructions.

Un microprocesseur est construit autour de deux éléments principaux :

1. une unité de commande
2. une unité de traitement

Ces éléments sont associés à des registres chargés de stocker les différentes informations à traiter

L'ensemble sont reliés entre eux par des bus internes (de données et d'adresses) permettant l'échange d'informations [ ]



Figure IV.42 : schéma fonctionnel d'un microprocesseur

#### IV.3.2. Mémoire centrale :

La mémoire centrale représente l'ensemble des mémoire RAM et ROM ( voir le chapitre III précédente ) utilisé pour le stockage des données, C'est cette capacité de mémorisation qui explique la polyvalence des systèmes numériques et leur adaptabilité à de nombreuses situations. Les données peuvent être lues ou écrites. Il y a écriture lorsqu'on enregistre des informations en mémoire et lecture lorsqu'on récupère des informations précédemment enregistrées. La mémoire centrale contient :

- Le programme à exécuter : suite d'instructions élémentaires
- Les données à traiter.

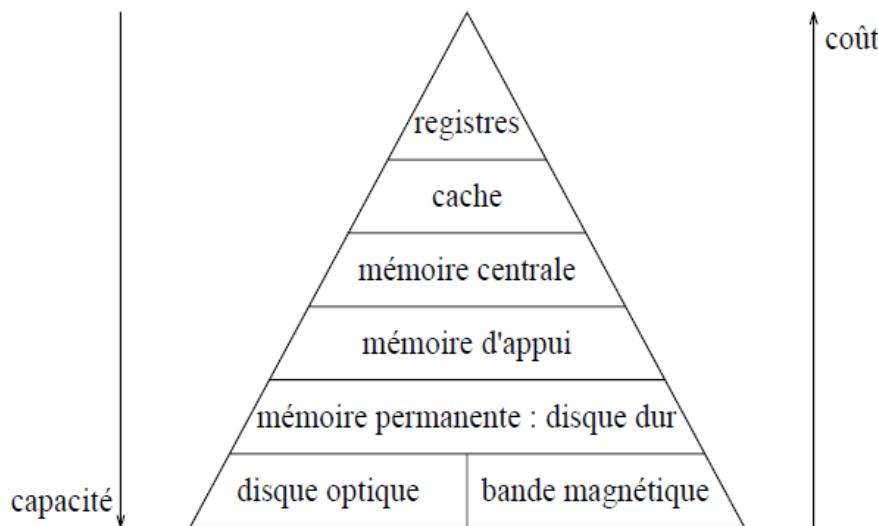


Figure IV.43 : Hiérarchie de mémoires

#### IV.3.3. Unité d'entrée / sortie :

C'est l'unité intermédiaire entre le microprocesseur et le monde extérieur (les périphériques externes) ; cette unité est constitué d'un ensemble des circuits intégrés connectés avec le microprocesseur par des bus de transmission (bus d'adresses, bus de données, bus de commandes), il est possible d'échanger les informations entre cette unité et la mémoire centrale par un accès direct à la mémoire (DMA : Direct Memory Acces).

#### IV.3.4. Unité de transfert.

Le microprocesseur échange des informations avec la mémoire et l'unité d'E/S, sous forme de mots binaires, au moyen d'un ensemble de connexions appelé bus. Un bus est un ensemble de lignes permettant de transférer des données sous forme parallèle, c'est-à-dire en faisant circuler n bits simultanément.

Les bus peuvent ^être décomposés en trois bus distincts :

- Le bus d'adresses permet au microprocesseur de spécifier l'adresse de la case mémoire à lire ou à écrire ;
- le bus de données permet les transferts entre le microprocesseur et la mémoire ou les E/S ;
- le bus de commande transmet les ordres de lecture et d'écriture de la mémoire et des E/S.

#### **IV.4. Les périphériques externes :**

Les périphériques sont tous des équipements ou dispositifs externes connectés avec le microprocesseur via les interfaces d'entrées / sorties comme un clavier, une souris, une imprimante, un disque dur, un capteur, une carte de développement, un actionneur, ,...etc.

#### **IV.5. Horloge :**

L'horloge est un signal carré ou rectangulaire susceptible de synchroniser les informations durant un cycle de fonctionnement du microprocesseur, ce signal se produit par circuit à quartz dont le circuit peut être interne ou externe du microprocesseur

#### **IV.6. Vocabulaire de l'ordinateur :**

Vers la fin des années 30, Claude Shanon démontra qu'à l'aide des contacteurs ( interrupteurs) avec deux états ,fermé pour « vrai » et ouvert pour « faux » , il est possible d'effectuer des opérations logiques en associant le nombre « 1 » pour vrai et « 0 » pour faux , ce codage de l'information est nommée la base binaire, c'est avec ce codage que fonctionnent des ordinateurs.Il consiste à utiliser deux états représentés par 0 ou 1 pour coder l'information.

- Le bit : signifie ( binary digital ) c'est-à-dire 0 ou 1 en numérotation binaire . c'est la plus petite unité d'information manipulable par une machine némurique , physiquement il est possible de représenter cette information par un signal électrique ou magnétique , qui lorsqu'elle atteint une certaine valeur correspond à la valeur 1.
- L'octet : l'octet est une unité d'information composée de 8 bits ; il permet de stocker un caractère telle que une lettre , un chiffre , ...etc
- Le mot : une unité d'information de 16 bits en anglais ( word ) , une unité d'information de 32 bits s'appelle double mot ( double word ) , les multiples sont :  $k_0=2^{10}$  bits=1024 bits ,  $1M=2^{20}$  bits ,  $1G=2^{30}$  bits ,  $1Tera=2^{40}$  bits

#### **IV.7. Architecture d'un ordinateur :**

Il existe deux architectures principales utilisée dans la fabrication des systèmes à microprocesseurs sont :

##### **IV.7.1. Architecture Von Neumann:**

Cette architecture est donnée par Docteur Von Neumann (1906-1957), elle consiste à utiliser un seul bus d'adresse et données multiplexé pour connecter le microprocesseur avec les mémoires et les interfaces d'entrées sorties, en revanche la mémoire est segmenté ou divisée en deux parties l'une pour sauvegarder les données et l'autre pour les programmes.

Cette architecture est économique et moins coûteuse mais son inconvénient est que l'exécution nécessite plus des échanges de données et une période de temps un peu longue.

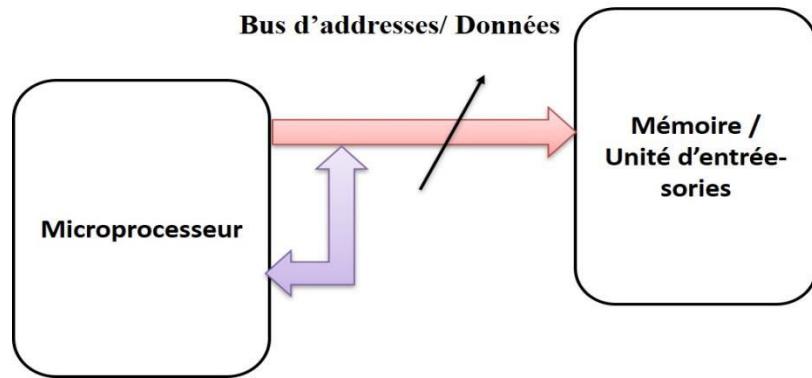


Figure IV.44: architecture Von Neumann

#### IV.7.2. Architecture Harvard:

Cette architecture proposée par l'université Harvard consiste à séparer systématiquement la mémoire de programme, de la mémoire de données, le bus d'adresse et le bus de données sont indépendants, elle est plus performante que celle de Von Neumann mais elle est plus complexe et plus chère, elle est plus utilisée dans conception des microcontrôleurs et des cartes de développement électroniques.

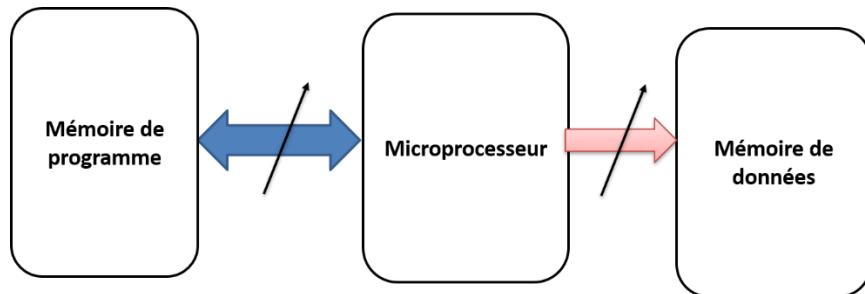


Figure IV.45: architecture Harvard.

#### IV.8. Conclusion:

Ce chapitre a présenté l'historique et l'évolution des ordinateurs suivant le développement de la technologie de semi-conducteurs aboutissant deux architectures principales à la fabrication des systèmes à microprocesseurs, la choix de chaque architecture se procède suivant la performance, la fiabilité, le coût et le champ d'application.

## Chapitre V : Architecture et fonctionnement du microprocesseur

### V.1.Historique

Le premier microprocesseur a été fabriqué par la société américaine INTEL en 1971. C'était un microprocesseur 4 bits baptisé 4004 destiné à équiper des calculatrices de bureau. En 1972 INTEL a produit le premier microprocesseur 8 bits baptisé 8008 par référence au précédent. Ce microprocesseur était destiné à répondre à un contrat de fabrication d'un terminal. En réalité le 8008 s'est révélé trop lent pour satisfaire le cahier des charges du terminal et INTEL a décidé de tenter le lancement de ce produit sur le marché grand public.

L'énorme succès de cette initiative fut à l'origine de la fabrication massive des microprocesseurs. Le 8008 est constitué de 3300 transistors et effectue 60000 instructions par seconde grâce à une horloge à 300KHz. A la suite du succès du 8008, INTEL produisit, dès 1974, le 8080 qui constituera le premier élément de la future famille de microprocesseurs de ce fabricant. Fort de ses 6000 transistors, le 8080, doté d'une horloge à 2MHz effectue 640000 instructions par seconde. En 1974, MOTOROLA, autre fondeur de silicium, décide de lancer le 6800 qui constituera lui aussi le début d'une grande famille.

En 1978, INTEL a réussi d'inventer le premier microprocesseur 8086 à 16 bits étant la base de la famille Intel 80x86 (8086, 80186, 80286, 80386, 80486, Pentium, ...). Plusieurs microprocessus actuels sont disponibles à base de 32bits ou 64 bits pour la fabrication des ordinateurs [V.1]

### V.2.Définition :

Un microprocesseur est un circuit intégré fabriqué par le silicium, dédié à contrôler et traiter les informations dans un micro-ordinateur, il est capable d'exécuter des milliers instructions par secondes suivant un programme

Il faut faire la distinction entre CPU, processeur, microprocesseur et microcontrôleur, DSP [V.2]:

- Le processeur, c'est le nom générique ou commun. C'est un composant (ou une carte électronique) puissant (par opposition au microprocesseur).

- Le microprocesseur, c'est un composant généraliste (il peut tout faire, et n'est optimisé pour rien), employé préférentiellement dans les systèmes informatiques, qui nécessite quelques circuits complémentaires pour pouvoir dialoguer avec son environnement. Selon les types il peut traiter directement des données de 8, 16, 32, 64 et même quelquefois 128 bits. C'est le composant de base de l'informatique, mais en instrumentation on lui préférera généralement des composants plus spécialisés, donc plus optimisés pour les tâches qu'on leur confiera, et donc plus performants.
- Le microcontrôleur est plus adapté aux applications embarquées car il comporte sur sa puce un certain nombre d'interfaces en plus du microprocesseur (Ram, ROM, I/O,...), par contre il est généralement moins puissant en terme de rapidité ou de taille de mémoire adressable et le plus souvent cantonné aux données de 8 ou 16 bits.
- Le processeur de signal (DSP : Digital Signal Processeur) est beaucoup plus spécialisé calcul. Il est optimisé pour le traitement du signal, mais n'est pas prévu pour accomplir des tâches généralistes de bas niveau qu'exécutent les deux précédents



Figure V.46 : Le microprocesseur de Marcian Hoff et Federico Faggin inventé 1971

### V.3.Caractéristiques générales d'un microprocesseur :

Un microprocesseur se définit du point de vue commercial par certaines caractéristiques pour être approprié aux opérations mathématiques et logique [10] :

- **Le bus de données** : C'est la taille de bits de donnée transférée parallèlement vers la mémoire ou l'unité d'entrée, cette caractéristique permet de classifier les microprocessus tels que 8 bits , 16 bits , 32 bits , 64 bits , ..etc

- **Espace adressable** : cette caractéristique représente la capacité de spécifier le nombre des cases mémoires à adresser selon la règle  $2^n$ , n : nombre des lignes d'adresse du microprocessus

- **Fréquence d'horloge** : le signal d'horloge est un signal de synchronisation qui cadence le fonctionnement du microprocesseur, il permet de spécifier la vitesse du microprocesseur, ce signal provient d'un circuit oscillateur, Exemple : horloge à 5 MHz, période  $T = 1/f = 0,2 \mu s$ . Si l'instruction s'exécute en 3 microcycles, la durée d'exécution de l'instruction est:  $3 \times 0,2 = 0,6 \mu s$

Pour augmenter les performances d'un microprocesseur, on peut donc soit augmenter la fréquence d'horloge (limitation matérielle), soit diminuer le CPI (choix d'un jeu d'instruction adapté).

#### V.4.Brochage et signaux :

Un microprocesseur un circuit intégré se présentant sous forme d'un boîtier DIP (Dual In-line Package) , le nombre des broches est symétrique , les broches représentent les entrées et les sorties des signaux du microprocesseur , ils sont regroupés en six ensembles :

- Les signaux d'alimentation (Vcc, Vss, X1, X2)
- Les signaux de commande du microprocesseur (Reset in , Reset out, Ready , clk,S0, S1).
- Les signaux de commande (RD, WR, O/M, Trap, SOD, SID, HOLD, HLDA)
- Les signaux des adresses et des données (AD0,...,AD7, A8,...,A15).
- Les signaux d'état. (RST5.5, RST6.5, RST7.5).
- Les signaux d'interruption (INTR, INTA).

X <sub>1</sub>	1	40	VCC
X <sub>2</sub>	2	39	HOLD
RST OUT	3	38	HLDA
SOD	4	37	CLK OUT
SID	5	36	<u>RST IN</u>
TRAP	6	35	READY
RST7.5	7	34	IO/M
RST6.5	8	33	S <sub>1</sub>
RST5.5	9	32	<u>RD</u>
INTR	10	31	WR
<u>INTA</u>	11	30	ALE
AD <sub>0</sub>	12	29	S <sub>0</sub>
AD <sub>1</sub>	13	28	A <sub>15</sub>
AD <sub>2</sub>	14	27	A <sub>14</sub>
AD <sub>3</sub>	15	26	A <sub>13</sub>
AD <sub>4</sub>	16	25	A <sub>12</sub>
AD <sub>5</sub>	17	24	A <sub>11</sub>
AD <sub>6</sub>	18	23	A <sub>10</sub>
AD <sub>7</sub>	19	22	A <sub>9</sub>
GND	20	21	A <sub>8</sub>

Figure V.47 : Brochage du microprocesseur.

Il est intéressant de dire que chaque type de microprocesseur possède un nombre spécifique des broches différent qu'un autre type.

Dans le chapitre prochain, on va détailler les signaux du microprocesseur à 8 bits.

### V.5. Architecture interne d'un microprocesseur :

Le microprocesseur comprend essentiellement des blocs fondamentaux pour effectuer ses opérations sont :

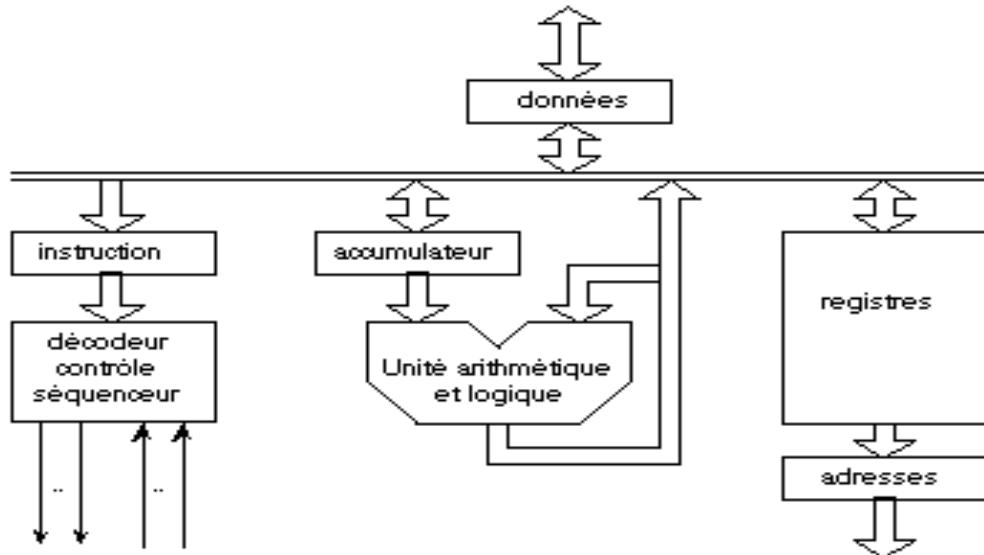


Figure V.48 : architecture interne standard d'un microprocesseur [11]

#### a- Unité arithmétique et logique (UAL) :

L'unité arithmétique et logique est une unité combinatoire servant à exécuter les opérations mathématiques et logiques (addition, soustraction, multiplication, ET, OU, ...)

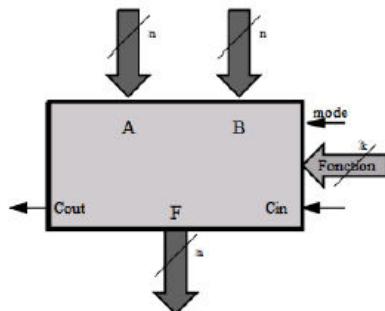


Figure V.49 : Structure de l'unité arithmétique et logique

Une commande mode permet de spécifier le type de calcul (arithmétique ou logique). La commande fonction sur k bits, permet d'indiquer l'une parmi les  $2^k$  fonctions possibles dans le mode considéré. La fonction sélectionnée s'appliquera sur les données A et B pour fournir le résultat F. L'entrée Cin sera utilisée dans le mode arithmétique comme retenue entrante, tandis que la sortie Cout sera la retenue sortante dans ce même mode arithmétique. Dans le cas du mode logique, étant donné que la retenue entrante ne sert pas, nous pourrons utiliser cette entrée comme un bit de commande supplémentaire, de manière à obtenir  $2^{k+1}$  fonctions logiques différentes

**b- Unité de commande et de contrôle :**

C'est l'unité de control qui supervise le déroulement de toutes les opérations au sein du Processeur. Elle est constituée principalement de :

L'horloge : C'est l'horloge qui génère les signaux qui permettent le cadencement et la synchronisation de toutes les opérations. Attention, l'horloge n'est pas une montre au sens commun du terme, c'est juste un signal carré qui a une fréquence fixe (3 Ghz par exemple), à chaque coup (front) d'horloge, le microprocesseur (qui ne l'oublions pas n'est qu'un circuit électronique) réalise une tache élémentaire. L'exécution d'une instruction nécessite plusieurs coups d'horloges.

- Le compteur programme PC : Le compteur programme (PC : program counter) est un registre (pointeur) qui contient l'adresse de la case mémoire où est stockée le prochain élément d'instruction qui devra être chargé dans le processeur pour être analysé et exécuté. Au début de l'exécution d'un programme, le PC est initialisé par le système d'exploitation à l'adresse mémoire où est stockée la première instruction du programme. Le compteur programme est incrémenté automatiquement chaque fois qu'un élément d'instruction est chargée dans le processeur
- Le registre d'instruction RI : C'est là où le CPU stocke l'instruction en cours d'exécution.
- Le décodeur : C'est lui qui va "décoder" l'instruction contenue dans RI et générer les signaux logiques correspondant et les communiquer au séquenceur.
- Le séquenceur : Il gère le séquencement des opérations et génère les signaux de commande qui vont activer tous les éléments qui participeront à l'exécution de l'instruction et spécialement l'ALU.

Le registre d'état : Le registre d'état est formé de plusieurs bits appelés drapeaux ou indicateurs (Flags) qui sont positionnés par l'ALU après chaque opération. Par exemple l'indicateur Z indique quand il est positionné que le résultat de l'opération est égal à Zéro. L'indicateur C indique que l'opération a généré une retenue. Le bit N indique que le résultat est négatif ...

On dispose d'un jeu d'instructions conditionnées par l'état de différents drapeaux

**c- Les registres de stockage temporaire :**

Le processeur utilise toujours des registres, qui sont des petites mémoires internes très rapides d'accès utilisées pour stocker temporairement une donnée, une instruction ou une adresse. Chaque registre stocke 8, 16 ou 32 bits. Le nombre exact de registres dépend du type de processeur et varie typiquement entre une dizaine et une centaine. Parmi les registres, le plus

important est le registre accumulateur, qui est utilisé pour stocker les résultats des opérations arithmétiques et logiques. L'accumulateur intervient dans une proportion importante des instructions.

- d- **Notion de pile :** La pile est une zone de mémoire RAM utilisée pour conserver le contenu des registres lorsque cela est nécessaire, pour appeler un sous-programme ou d'une interruption le pointeur de pile (SP, Stack Pointer) doit effectuer l'adresse de la prochaine instruction à partir de la case mémoire

#### V.6.Format d'une instruction :

Pour qu'un microprocesseur traite une information, il exécute un programme, le programme est une suite d'instructions ordonnées suivant un algorithme structuré, Une instruction est un mot binaire décrivant l'action que doit exécuter le microprocesseur, par exemple copier une case mémoire dans une autre, ajouter le contenu de deux cases mémoires etc.

Ces instructions sont des mots binaires qu'on écrit généralement en hexa, mais pour une plus grande facilité on les désignera par leurs « mnémoniques ». La mnémomique est une abréviation en caractère latin. Une instruction est formée par un ou plusieurs octets. Le premier champ représente l'opération à effectuer et qui est appelée l'opcode. Dans le 8085 l'opcode a toujours une taille de 1 octet. L'instruction est formée soit par un l'opcode ou par l'opcode et un opérande. Ce dernier peut être spécifié par différentes formes. Il peut être une donnée de 8 bits (ou 16 bits), un registre interne ou une adresse d'une position mémoire. Le format d'instruction s'écrit :

**Opération à effectuer      opérande2 ; opérande 1**

#### V.7.Langage machine :

Langage machine est un langage numérique sous forme binaire exécuté par le microprocesseur après une traduction ou compilation qui consiste à transformer un code source lisible par un humain en un fichier binaire exécutable par une machine.

#### V.8.Langage assembleur :

Langage assembleur ou langage d'assemblage est un langage informatique de plus bas niveau consiste à un nombre des symboles appelés mnémonique c'est-à-dire des mots abrégés traduits en langage machine pour exécuter une instruction comme (Add, Soust,Mov, ....) , le langage

machine convertit ces mnémoniques en langage machine , ainsi que les valeurs (écrites en décimal) en binaire et les libellés d'emplacements en adresses, en vue de créer par exemple un fichier objet ou un fichier exécutable.

La méthode de programmation en langage assembleur s'établit comme suivant :

Définir le problème à résoudre : que faut-il faire exactement ?

- D'déterminer des algorithmes, des organigrammes : comment faire ? Par quoi commencer, puis poursuivre ?
- Rédiger le programme (code source) :
  - utilisation du jeu d'instructions (mnémoniques) ;
  - création de documents explicatifs (documentation).
- Tester le programme en réel ;
- Corriger les erreurs (bugs) éventuelles : déboguer le programme puis refaire des tests jusqu'à obtention d'un programme fonctionnant de manière satisfaisante.

### Réalisation pratique d'un programme :

- Rédaction du code source en assembleur `a l'aide d'un éditeur (logiciel de traitement de texte ASCII) :
  - édit sous MS-DOS,
  - notepad (bloc-note) sous Windows,
- Assemblage du code source (traduction des instructions en codes binaires) avec un assembleur :
  - MASM de Microsoft,
  - TASM de Borland,
  - A86 disponible en shareware sur Internet, ...

Pour obtenir le code objet : code machine exécutable par le microprocesseur ;

- Chargement en mémoire centrale et exécution : rôle du système d'exploitation (ordinateur) ou d'un moniteur (carte de d'développement `a base de microprocesseur).

Pour la mise au point (d'débogage) du programme, on peut utiliser un programme d'aide `a la mise au point (comme DEBUG sous MS-DOS) permettant :

- l'exécution pas à pas;
- la visualisation du contenu des registres et de la m'mémoire ;
- la pose de points d'arrêt ...

Exemple :

```

name "add-sub"
org 100h
mov al, 5      ; bin=00000101b
mov bl, 10     ; hex=0ah or bin=00001010b
; 5 + 10 = 15 (decimal) or hex=0fh or bin=00001111b
add bl, al
; 15 - 1 = 14 (decimal) or hex=0eh or bin=00001110b
sub bl, 1
; print result in binary:
mov cx, 8
print: mov ah, 2 ; print function.
        mov dl, '0'
        test bl, 10000000b ; test first bit.
        jz zero
        mov dl, '1'
zero: int 21h
        shl bl, 1
loop print
; print binary suffix:
        mov dl, 'b'
int 21h
; wait for any key press:
        mov ah, 0
int 16h
ret

```

## V.9.Notions d'architecture RISC et CISC

Actuellement l'architecture des microprocesseurs se composent de deux grandes familles :

- e- Architecture CISC (Complex Instruction Set Computer)
- f- Architecture RISC (Reduced Instruction Set Computer).

L'architecture CISC est une architecture avec un grand nombre d'instructions où le microprocesseur doit exécuter des tâches complexes par instruction unique. Pour une tâche

donnée, une machine CISC exécute ainsi un petit nombre d'instructions mais chacune nécessite un plus grand nombre de cycles d'horloge. Le code machine de ces instructions varie d'une instruction à l'autre et nécessite donc un décodeur complexe (microcode)

L'architecture RISC est une architecture dans laquelle les instructions sont en nombre réduit (chargement, branchement, appel sous-programme). Les architectures RISC peuvent donc être réalisées à partir de séquenceur câblé. Leur réalisation libère de la surface permettant d'augmenter le nombre de registres ou d'unités de traitement par exemple. Chacune de ces instructions s'exécutent ainsi en un cycle d'horloge. Bien souvent, ces instructions ne disposent que d'un seul mode d'adressage. Les accès à la mémoire s'effectuent seulement à partir de deux instructions (Load et Store). Par contre, les instructions complexes doivent être réalisées à partir de séquences basées sur les instructions élémentaires, ce qui nécessite un compilateur très évolué dans le cas de programmation en langage de haut niveau.

Le choix dépendra des applications visées. En effet, si on diminue le nombre d'instructions, on crée des instructions complexes (CISC) qui nécessitent plus de cycles pour être décodées et si on diminue le nombre de cycles par instruction, on crée des instructions simples (RISC) mais on augmente alors le nombre d'instructions nécessaires pour réaliser le même traitement.

#### **V.10.Notions sur les interruptions :**

L'interruption est un mécanisme fondamental de tout processeur, Il permet de prendre en compte des événements extérieurs au processeur et de leur associer un traitement spécifique., Il faut noter que l'exécution d'une instruction n'est jamais interrompue ; c'est à la fin de l'instruction en cours lors de l'arrivée de l'événement que le sous-programme d'interruption est exécuté. A la fin de cette procédure, le microcontrôleur reprend le programme principal à l'endroit où il l'a laissé  
Les interruptions peuvent être causées par des sources externes ou par des sources internes

- Sources externes :

- Broches parallèles (Exemples : clavier, alarme)
- Ports séries

- Sources internes

- Timer
- Convertisseur A-N
- Reset

Le dernier chapitre VIII , on va détailler sur les interruptions et leurs méthodes de programmation dans un système à microprocesseur.

### V.11. Conclusion

Dans ce chapitre, on a décrit l'architecture et le fonctionnement d'un microprocesseur d'une façon générale ; mais il important de signaler que chaque type du microprocesseur possède des caractéristiques spécifiques agissant sur ses performances et sa puissance de traitement des informations, le chapitre prochain est consacré pour étudier le microprocesseur à 8 bits comme exemple d'application.

**Chapitre VI :****Etude d'un microprocesseur 8 bits****VI.1. Introduction**

L'étude d'un système à microprocesseur exige une connaissance détaillée sur le cœur du traitement dit le microprocesseur, à ce propos nous allons décrire et analyser l'architecture et le fonctionnement d'un microprocesseur 8 bits ainsi que sa performance pour le traitement des informations suivant ses caractéristiques techniques.

**VI.2. Les différentes familles de microprocesseurs 8 bits :**

Les microprocesseurs 4 bits ont été développés dans les années 1970 en commençant par les Intel 4004. La société Intel a rapidement suivi avec les processeurs 8 bits, et la plupart des concurrents d'Intel ont commencé avec les 8 bits. Les limitations de performances et limites de la mémoire expliquent que les processeurs 4 bits ont rapidement été dépassés sur des applications plus exigeantes.

Un processeur peut être classé sur la base des données auxquelles il peut accéder en une opération unique.

Un processeur 8 bits peut accéder à 8 bits de données en une seule opération, par opposition à un processeur 16 bits qui peut accéder à 16 bits de données en une seule opération.

Voici une liste (très incomplète) de processeurs 8 bits :

Intel 8008, Intel 8080 (source compatible 8008), Intel 8085 (compatible binairement 8080)

Intel 8051 (architecture Harvard), Zilog Z80 (compatible binairement 8080), Zilog Z180 (compatible binairement Z80), Zilog Z8, Zilog eZ80 (compatible binairement Z80), Motorola 6800, Motorola 6803, Motorola 6809 (partiellement compatible 6800), MOS Technology 6502, Microchip PIC10, Microchip PIC12, Microchip PIC16, Microchip PIC18, famille microcontrôleurs Atmel AVR, série de microcontrôleurs NEC 78K0, RCA 1802 (cosmac).[12]

**VI.3. Compatibilité entre microprocesseurs :**

Les microprocesseurs se diffèrent d'un type à un autre suivant la taille et le nombre des registres, le nombre des données et des adresses, la fréquence d'horloge et le nombre des jeux d'instructions, mal ils sont similaires en point de vue du processus du traitement d'informations et l'architecture de base interne, en effet le développement des microprocesseurs atteint actuellement jusqu'à 64 bits dans la majorité des systèmes à microprocesseurs suivant la loi de

Moore qui permet l'évolution la technologie d'intégration des circuits intégrés à chaque 18 mois

### .VI.3.. Prédominance des microprocesseurs Intel et Motorola :

La société Américaine Intel (Intel corporation) est le premier fabricant du microprocesseur, des cartes mères, des mémoires flash et des processeurs graphiques. Intel commercialise en produit phare la série des microprocesseurs dite x86 utilisée par les compatibles PC depuis 1981 avec pour produit le plus vendu la série des Pentium

Motorola Inc. est entreprise américaine spécialisés dans le domaine de télécommunications et électronique, Elle a été compétiteur d'Intel dans le domaine des semi-conducteurs, en particulier avec des gammes de processeurs (680x0, 88000, PowerPC, DragonBall et ColdFire) et de microcontrôleurs (68HC11).

### .VI.4. Etude d'un microprocesseur 8 bits, brochage et signaux externes :

Un microprocesseur 8085 à 8 bits est un circuit intégré fabriqué en 1976 par la société Américaine INTER sous forme DIP (Dual In-line Package) doté de 40 broches, il est compatible avec le microprocesseur 8080, ses broches peuvent être classées en 4 catégories : Signaux d'alimentation et générateur d'horloge, signaux de contrôle et d'état, bus d'adresses, bus des données, signaux d'interruptions et signaux d'interfaces d'entrées-sorties.

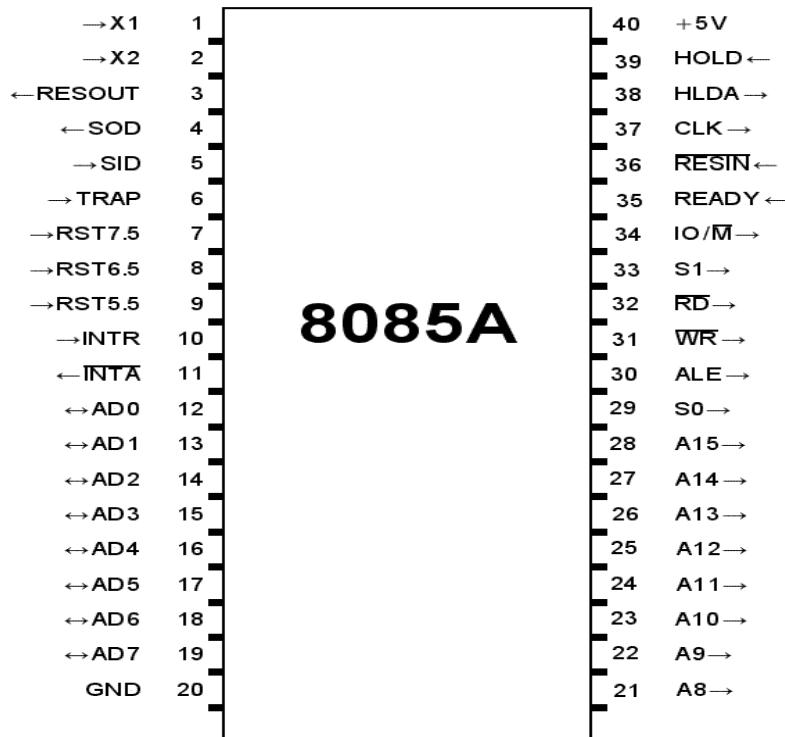


Figure VI.50 : brochage du microprocesseur 8085 [13]

### a- Signaux d'alimentation et générateur de fréquence

**Vcc** : Alimentation +5V

**GND** : Masse de référence

**X1 et X2** : Un Quartz est relié aux deux broches pour générer un signal carré périodique, pour générer une fréquence de 3MHz, le quartz doit avoir une fréquence de 6MHz.

Suivant la formule :

**CLK (OUT)** : Cette broche peut être utilisée comme signal d'horloge pour d'autres circuits

### b- Signaux des adresses et des données :

Le microprocesseur 8085 possède 16 lignes d'adresses divisées en deux poids, un poids fort représenté par un bus d'adresses 8 bits ( A8.....A15) envoyé du microprocesseur vers la mémoire , et un poids faible (AD0-AD7 ) qui est un bus 8 bits multiplexé avec un le bus de données 8 bits , pour les démultiplexer , un circuit verrou (Latch ) de type 74LS373 ,commandé par un signal de commande ALE délivré par le microprocesseur pour valider le pilotage entre les adresses (A0-A7 et ) les données (D0-D7 )

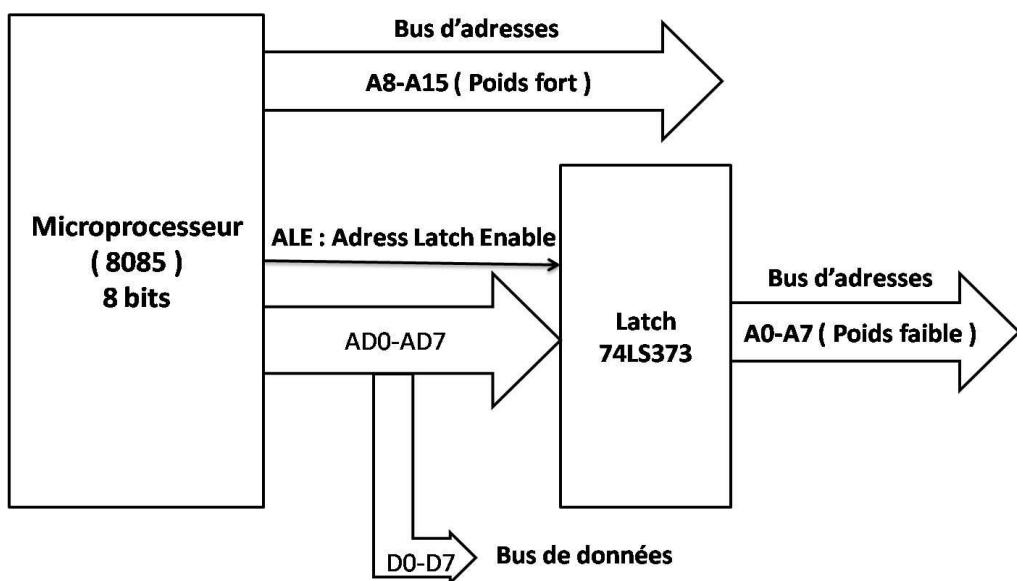


Figure VI.51 : Démultiplexage du bus d'adresses/ données

### a- Signaux de commandes :

**ALE** : A l'état haut, implique que les bits présents sur le bus A/D sont les lignes d'adresses. Un circuit 74373 mémorise ces bits à sa sortie.

**RD** : Read, signal de lecture d'une donnée, actif niveau bas.

**WR** : Write, signal d'écriture d'une donnée, actif niveau bas.

**IO/M** : Input-Output / Memory, indique si le 8085 adresse la mémoire ( $IO/M = 0$ ) ou les entrées/sorties ( $IO/M = 1$ ).

**b- Signaux d'états (flags) :**

**S0 et S1** : Signaux d'état indiquant le type d'opération en cours sur le bus. Rarement utilisés dans les petits systèmes.

**c- Signaux d'interfaces d'entrée-sorties**

Le 8085 possède deux signaux pour l'implémentation de la transmission série : **SID** (Serial

Input Data) et **SOD** (Serial Output Data). Dans la transmission série, les bits de données sont transmis sur une seule ligne l'un après l'autre comme dans le cas de la ligne téléphonique.

**d- Signaux d'interruptions**

Le 8085 possède cinq signaux d'interruptions :

**INTR (Interrupt Request)** : C'est le signal envoyé par une interface indiquant une demande d'interruption.

**INTA** : Le 8085 répond à INTR en envoyant 0 sur la ligne INTA (Interrupt Acknowledge).

**RST7.5 RST6.5 et RST5.5** : Interruptions de redémarrage. Ce sont des interruptions vectorisées qui transfèrent le contrôle à une position mémoire spécifique. Ces lignes sont prioritaires que INTR. Les priorités sont classées selon l'ordre décroissant de 7.5, 6.5 à 5.5.

**TRAP** : (Non Maskable Interrupt) : interruption prioritaire par rapport à INTR.

**HOLD et HLDA** : signaux de demande d'accord d'accès direct à la mémoire (DMA).

**READY** : entrée de synchronisation avec la mémoire.

**RESET IN**: Quand ce signal est à 0, le compteur de programme est remis à 0, les bus sont en haute impédance et le microprocesseur redémarre.

**RESET OUT** : Ce signal indique que le microprocesseur vient d'être redémarrer, ce signal peut être utilisé par d'autres circuits.

## **VI.5. Architecture interne :**

Le microprocesseur 8 bits Intel 8085 construit d'un nombre de blocs se regroupant en trois ou quatre :

**L'unité arithmétique et logique (UAL) :**

Elle effectue les différentes opérations arithmétiques et logiques (+,-,\*/, AND, OR, NOT,..., décalage à gauche ou à droite, ..etc), L'UAL utilise deux registres essentiels, l'accumulateur (registre 8-bit) qui fait partie de chaque opération arithmétique et logique et un registre temporaire (8-bit) pour le stockage de données pendant l'exécution de l'opération, ce registre est inaccessible par le programmeur.

### **Unité de commande :**

Elle décode les instructions et délivre des ordres pour communiquer avec les mémoires et l'unité d'entrée –sortie comme écrire, lire, branchement, interruption,...etc.

Elle décode l'instruction à exécuter et génère les séquences de tous les signaux nécessaires pour l'exécution correcte de l'instruction.

### **Les registres :**

Un registre est une petite mémoire temporaire pour stocker les informations, dans un microprocesseur il y a plusieurs types de registres à savoir :

#### **a. Registres généraux**

- B, C, D, E, H et L : Sont des registres de 8-bit, souvent utilisés pour le stockage temporaire d'informations. Ils peuvent être utilisés seuls ou combinés comme des registres 16-bits, BC, DE, et HL

#### **b. Registres spéciaux**

**Compteur de programme (Program Counter (PC))** : C'est un registre 16-bit, il contient l'adresse de la prochaine instruction à exécuter. L'exécution d'une instruction incrémentera automatiquement son contenu (+1).

**Pointeur de pile (Stack Pointer (SP))** : C'est un registre 16-bit, il indique au microprocesseur l'adresse de la prochaine case disponible dans la pile. La pile est une petite partie de la mémoire RAM utilisée pour sauvegarder les contenus des registres lorsque cela est nécessaire.

**Registre d'instruction (Instruction Register (IR))** : C'est un registre non programmable, il contient l'instruction en cours d'exécution. Ce registre est chargé depuis de la RAM ou de la ROM au début du cycle d'exécution, dont l'adresse est donnée par le compteur de programme PC.

**Registre d'état (Flag Register)** : Il est composé de plusieurs bits appelés drapeaux (Flags) servant à stocker des informations concernant le résultat de la dernière instruction exécutée.

FIGURE 3.3 – Bits du registre d'état (Flag register)

Les rôles des cinq drapeaux (flags) de registre d'état sont les suivants :

✓ S (Sgin Flag (Bit de signe)) : il indique le signe du résultat dans l'accumulateur (1-négatif, 0-positif). Pour une opération d'addition ou de soustraction, S=1 si le bit le plus significatif du résultat est égal a "1".

✓ Z (Zero Flag) : Ce drapeau est mis a 1 si le résultat de l'opération est nul.

✓ AC (Auxillary carry Flag) : Ce bit est égal a 1 s'il y a une retenue du quarter de poids faible dans le quarter de poids fort.

✓ P (Parity Flag) : Ce bit est mis a 1 si le résultat contient un nombre pair de 1 (la parité est paire) et mis a 0 si le résultat contient un nombre impair de 1 (la parité est impaire).

✓ CY (Carry Flag) : Quant une retenue ou une report est générée lors d'une opération, le CY est mis a 1.

### c. Registres non programmables

Registre d'instruction (Instruction Register (IR)) : Il est déjà présent dans les registres spéciaux.

Registre de décodage d'instructions : Il décode l'instruction trouvée dans le registre d'instruction.

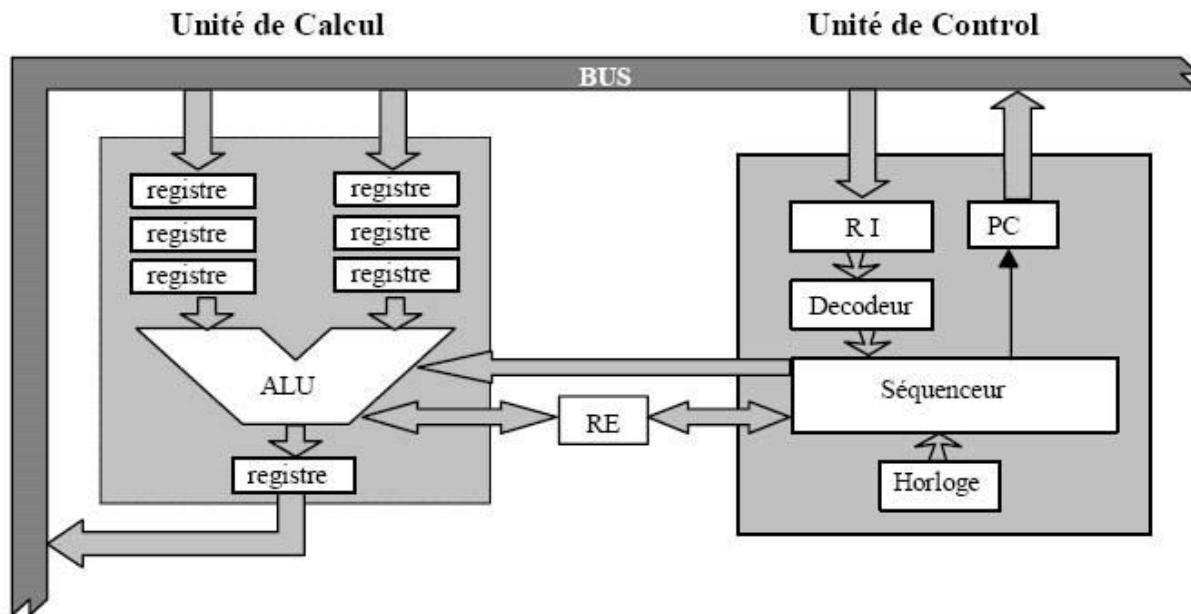


Figure VI.52 . : Structure interne du microprocesseur 2085 [14]

## VI.6. format d'une instruction :

Le programme est une suite d'instruction. Une instruction est un mot binaire décrivant l'action que doit exécuter le microprocesseur, par exemple copier une case mémoire dans une autre, ajouter le contenu de deux cases mémoires etc.

Ces instructions sont des mots binaires qu'on écrit généralement en hexa, mais pour une plus grande facilité on les désignera par leurs « mnémoniques ». La mnémonique est une abréviation en caractère latin. Une instruction est formée par un ou plusieurs octets. Le premier champ représente l'opération à effectuer et qui est appelée l'**opcode**. Dans le 8085 l'opcode a toujours une taille de 1 octet. L'instruction est formée soit par un l'opcode ou par l'opcode et un opérande. Ce dernier peut être spécifié par différentes formes. Il peut être une donnée de 8 bits (ou 16 bits), un registre interne ou une adresse d'une position mémoire.

Opération à effectuer opérande 2, opérande1

L'ensemble des instructions du 8085 est divisé en trois groupes :

- Instructions de tailles 1 octet (pas d'opérande)
- Instructions de tailles 2 octets (opérande de 8 bits)
- Instructions de tailles 3 octets (opérande de 16 bits)

## VI.7. Modes d'adressage

Les façons de désigner les opérandes constituent les : modes d'adressage, Le microprocesseur 8085 possède quatre modes d'adressage :

- Mode d'adressage par registre.
- Mode d'adressage immédiat.
- Mode d'adressage direct.
- Mode d'adressage indirect.

### 7.1. Adressage par registre :

Ce mode consiste à transférer une information ou une donnée d'un registre à l'autre. Dans ce mode l'opérande sera stocké dans un registre interne au microprocesseur.

Exemple :

MOV A, B : déplacer le contenu du registre B vers le registre A.

### 7.2. Adressage immédiat :

Dans ce mode, le contenu d'une adresse est transféré immédiatement dans le registre.

**Exemple :**

MVI A, 30H : transférer le contenu de l'adresse 30H vers le registre A.

### 7.3. Adressage direct :

Dans ce mode d'adressage, on spécifie directement l'adresse de l'opérande dans l'instruction.

Par exemple pour charger l'accumulateur A par le contenu de l'adresse 2000H, on doit écrire :

LDA 2000H

### 7.4. Adressage indirect :

Dans ce mode d'adressage l'adresse de l'opérande est stockée dans un registre qu'il faut bien évidemment le charger au préalable par la bonne adresse. L'adresse de l'opérande sera stockée dans un registre pair B, D, H ou SP.

**Exemple :**

LXI H,1234H ; adressage immédiat codé par 21H 34H 12H

MOV A,M ; adressage indirect codé par 7E

## VI.8. Jeu d'instructions :

Le microprocesseur possède nombre d'instructions se divise en quatre groupes comme indiquée dans les tableaux suivants :

### 8.1. Instructions de transfert

Instruction	Explication	Exemple
MOV	Copier un octet de données depuis un registre ou adresse mémoire	MOV A,b
MVI	Charger un octet de donnée	MVI A,14H
OUT	Envoyer un octet de donnée vers une interface	OUT 08H
IN	Recevoir un octet de donnée d'une interface	IN 09H
LXI	Charger un mot de donnée	LXI B,1400H
LDA	Charger l'accumulateur par un contenu d'une adresse mémoire	LDA, 2000H
LDAX	Charger l'accumulateur par un contenu d'une adresse mémoire spécifiée dans un registre A ou D	LDAX B
STA	Stocker l'accumulateur dans une case mémoire	STA 3000H
STAX	Stocker l'accumulateur dans une case mémoire dont l'adresse est spécifiée dans un registre A ou D	STAX D

LHLD	Charger le registre pair H par le contenu de la case mémoire	LHLD 3000H
SHLD	Charger le contenu de la case mémoire par le contenu registre pair H	SHLD 3000H
XCHG	Echange les contenus entre HL et DE	XCHG
XTHL	Echange le top de la pile avec HL	XTHL
SPHL	Copier les contenus des registres HL dans le registre SP	SPHL
PCHL	Copier les contenus des registres HL dans le registre CP	PCHL

## 8.2. Instructions arithmétiques

Instruction	Explication	Exemple
ADD	Additionner le contenu de l'accumulateur A avec le contenu du registre ou case mémoire.	MVI A ,04H MVI B,03H ADD B
ADI	Addition immédiat	MVI A ,FFH ADI 01H
SUB	Soustraction	MVI A,FFH MVI B, 03H SUB B
SUI	Soustraction immédiat	MVI A,05H SUI 02H
INR	Incrémantation	MVI A,05H INR A
DCR	Décrémentation	MVI A ,01H DCR A
INX	Incrémentation 16 bits	LXI SP ,2000H INX SP Result : 2000H+1=2001H

DCX	Décrémentation 16 bits	LXI B,3000H DCX B Résultat : $3000-1=2FFDH$
ADC	Additionner registre 8bits ou une cas mémoire avec A et CY	MVI A,30H MVI B ,F0H ADD B ADC B
ACI	Additionner une donnée 8bits avec A et CY	MVI A,30H MVI B ,F0H ADD B ACI 30H Résultat : $30H+F0H=20H$ et CY=1 $20H+30H=51H$
SBB	Soustraire registre 8 bits ou case mémoire et report CY de A	MVI A ,30H MVI B,20H ADI FFH SBB B Résultat : $30H+FFH=2FH$ et CY=1 $2FH-20-1=0EF$
SBI	Soustraire donnée 8bitede donnée et report CY de A	MVI A,30H ADI FFH SBI 20H Résultat: $30H+FFH+2FH$ et CY de A $2FH-20-1+0EF$

DAA	Ajustement décimal de l'accumulateur.	MVI A,38H MVI B , 43H ADD B DAA Résultat : A=7BH , A=81H
-----	---------------------------------------	---

### 8.3. Instructions logiques

Instruction	Explication	Exemple
ANA	AND logique entre l'accumulateur A et le contenu du registre	MVI A,81H MVI B ,77H ANA B
ANI	AND logique entre le contenu de l'accumulateur A et une valeur 8 bits.	MVI A, 55H ANI 01H
ORA	OR logique entre l'accumulateur A et le contenu du registre ou une case mémoire.	MVI A,81H MVI B,7EH ORA B
ORI	OR logique entre le contenu de l'accumulateur A et une valeur 8 bits.	MVI A,55H ORI 02H
XRA	OU exclusif entre le contenu de l'accumulateur A et une valeur 8 bits.	MVI A, 80H MVI B ,7EH XRA B
XRI	OU exclusif entre le contenu de l'accumulateur et une valeur 8 bits	MVI A,55H XRI 02H
CMA	Complémenter l'accumulateur A (inverser tous les bits de A)	MVI A,55 CMA Résultat : AAH
RLC	Rotation à gauche de A	
RAL	Rotation à gauche de A par CY	

RRC	Rotation à droite de A	
RAR	Rotation à droite de A par CY	

#### 8.4. Instructions de comparaison

Instruction	Explication	Exemple
CMP	Comparaison avec l'accumulateur A	
CPI	Comparaison en immédiat avec l'accumulateur A	
CMP R	Compare le contenu du registre avec le contenu de A	
CMP M	Compare le contenu de case mémoire avec le contenu de A	

#### 8.5. Instructions de branchement

Instruction	Explication	Exemple
JMP	Saut	
JC	Saut si retenue CY=1	
JNC	Saut si pas de retenue y=0	
JZ	Saut si le résultat est nul	
JNZ	Saut si le résultat est non nul	
JP	Saut si plus grand	
JM	Saut si plus petit	
JPE	Saut si pair	
JPO	Saut si impair.	
STC	Mise à 1 de CY	
CMC	Complémenter CY	

#### VI.9. La gestion de la pile

Une pile dans un système à microprocesseur peut être décrite comme un ensemble de cases mémoire en lecture et écriture. Cette zone mémoire est utilisée pour stocker les octets

temporairement durant l'exécution d'un programme. Le début de la pile est défini dans le programme par l'utilisation de l'instruction LXI SP, qui charge l'adresse mémoire de la pile dans le registre pointeur de pile (SP). En général SP pointe vers le top de la pile. Par exemple si SP est chargé par 2099H (LXI SP,20099H), le stockage des données commence à partir de 2098H et continu selon l'ordre 2097H, 2096H,... La taille de la pile dépend de la taille de la mémoire disponible. Les contenus des registres pairs peuvent être stockés dans la pile (deux octets en même temps) par l'utilisation de l'instruction PUSH. Pour transférer les données depuis la pile vers les registres POP est utilisée. Deux octets sont transférés de/vers la pile, ce qui implique que lors d'un PUSH SP se décrémente de 2 et lors d'un POP SP s'incrément de 2.

**Exemple :**

LXI B,1234H

LXI SP, 2000H

PUSH B

POP B

## VI.10. Programmes en assembleur [15]

Un programme en assembleur a une forme bien particulière. Chaque ligne d'une source assembleur comporte une instruction. Chaque ligne est composée de champs. De gauche `a droite, on a :

- le champ étiquette, qui peut être vide
- le champ mnémonique (le nom de l'instruction)
- le champ opérande (les arguments de l'instruction), qui peut être vide
- le champ commentaire, qui peut être vide

**Exemple :**

; Forme générale d'un fichier source en assembleur 8085

; Nom du fichier : modele.asm

; Date d'écriture : 30 Nov. 2022

; Objet : Allumer des LED

; Programme, comment on l'utilise, ...

Etiquette	mnémonique	opérande	commentaire
Donnée 1	MOV	B, 12 H	; déplacer l'adresse 12 H vers le registre B
.....	.....	.....	; commentaire

```
.....      .....      ..... ; commentaire
.....      .....      ..... ; commentaire
.....      .....      ..... ; commentaire
.....      .....      ..... ; commentaire.
.....      END          ; fin de source
```

## VI.11. Conclusion

Dans ce chapitre, nous avons détaillé la structure et le fonctionnement du microprocesseur 8 bits comme un élément de base dans un ordinateur, suivant la loi de Moore qui favorise l'intégration des circuits intégrés chaque 18 mois, les microprocesseurs sont développés en plusieurs types 16 bits, 32 bits et 64 bits pour traiter les données, mais chaque microprocesseur possède ses caractéristiques permettant de réaliser les taches désirées.

**Chapitre VII : Les interfaces d'entrées-sorties****VIII.1. Introduction :**

L'exploitation des résultats obtenus du traitement des données doit être à la portée de l'utilisateur sous plusieurs formes comme des graphes, images, textes ou des tableaux,...etc donc il est indispensable d'utiliser autres dispositifs électroniques adaptés avec le système à microprocesseur pour effectuer ces taches, ces dispositifs sont appelés les périphériques connectés avec le microprocesseur à travers une unité des interfaces entrée/sorties.

**VII.2. Définition**

Une interface entrée/ sortie est un circuit intégré permettant au microprocesseur de communiquer avec les périphérique ou le monde extérieur d'un système à microprocesseur tels que : clavier, souris, imprimante, capteur, caméra, processus industriel .....etc.

Les entrées sont les données envoyées par un périphérique ( clavier , souris , capteur,...etc) destinées vers le microprocesseur.

Les sorties sont les données émises par le microprocesseur vers un périphérique (imprimante, écran, lampe, moteur,...etc)

Les interfaces sont connectées avec le microprocesseur par les bus d'adresses, les bus de données et les bus de commandes, les points d'accès sont appelés les ports

Il y a deux types d'interfaces : interface parallèle et interface série, d'où on trouve un port parallèle et un port série.

Une interface parallèle permet de transmettre un mot de données parallèlement sous forme d'un octet ou multiple entre le microprocesseur et un périphérique.

Une interface série permet de transmettre les données en série bit par bit entre le microprocesseur et le périphérique.

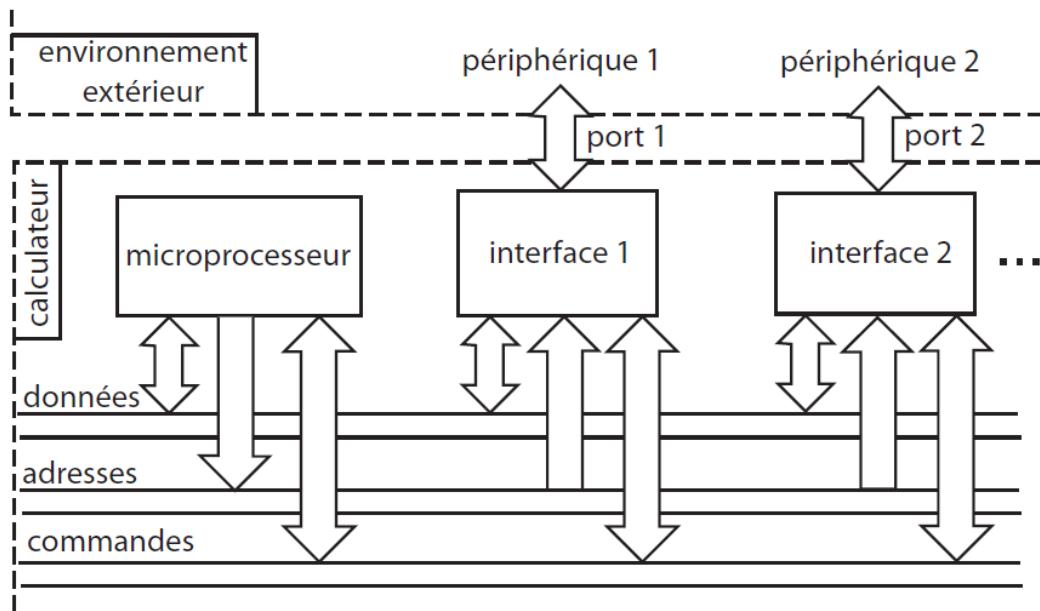


Figure VII.53 : microprocesseur et interfaces [16]

### VII.3. Communication entre le microprocesseur et interface

Le microprocesseur 8085 dispose un espace mémoire  $2^N = 64$  KO de 16 bits lignes d'adresses et d'interface d'entrée-sortie de 256 Octets

Pour sélectionner entre la mémoire et l'interface, le microprocesseur émet un signal I/OM

Pour accéder à la mémoire, le signal I/OM=0.

Pour accéder aux d'interfaces d'entrée-sortie , le signal I/OM=1.

### VII.4. Différents types d'interfaces

Il existe plusieurs types d'interfaces d'entrée sortie de la famille INTEL à savoir: l'interface périphérique programmable 8255A, l'interface série 8250, le contrôleur d'interruptions 8259A, le Timer 8254 et le DMA 8237.

#### VII.4.1 .Interface parallèle 8255

Le 8255 est un circuit programmable de 40 broches (voir figure 1) d'interface d'entrée/sortie parallèle qui a été conçu pour travailler avec les microprocesseurs de la famille INTEL. Il est formé par trois ports d'entrées/sorties, chaque port est de 8 bits qui peuvent être programmés en entrée ou en sortie avec trois modes différents (mode zéro, mode un et mode deux).

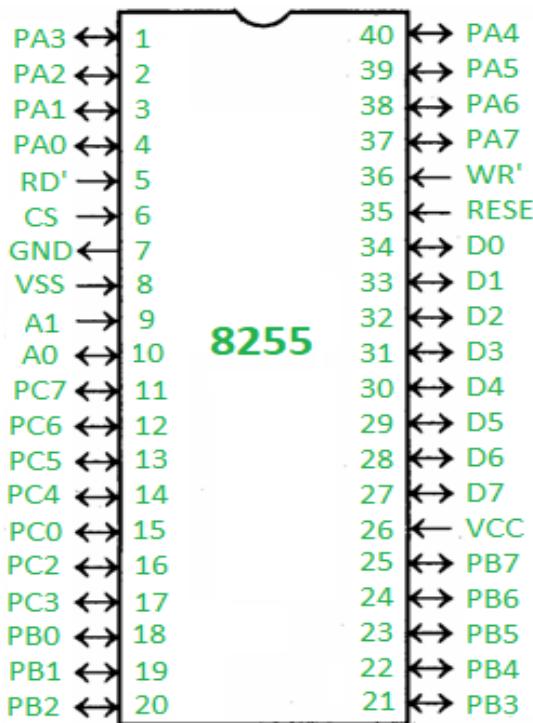


Figure VII.54 : Brochage d'interface 8255

Le schéma bloc des 8255 montre bien qu'il est divisé en deux groupes :

Groupe A : formé par le port A et le port C haut.

Groupe B : formé par le port B et le port C bas.

Le registre de données (Data buffer Bus) assure la liaison entre le bus de données extérieur et le registre de contrôle ainsi que les ports d'entrées/sorties.

La sélection du 8255A se fait par l'intermédiaire de la pine CS (qui est en général fournie par une logique de décodage qui permet la sélection du 8255A : voir plus loin les exemples d'applications)

<b>A<sub>1</sub> A<sub>0</sub></b>	<b>Registre</b>
0 0	Port A
0 1	Port B
1 0	Port C
1 1	Registre de commande

Tableau VII.5 : les adresses de sélection d'un port

Le bus d'adresse du 8255A est formé essentiellement par deux pines (A0et A1) qui permettent de sélectionner les ports ainsi que le registre de contrôle comme le montre le tableau suivant : Un registre de données assure la liaison entre le bus de données extérieur et le registre de contrôle ainsi que les ports d'entrées/sorties [17].

La logique de commande est composée de six lignes :

- **RD** : Ce signal permet d'effectuer une opération de lecture. Quand ce signal est à 0, le microprocesseur lit la donnée du port sélectionné du 8255A.
- **WR** : Ce signal permet d'effectuer une opération d'écriture. Quand ce signal est à 0, le microprocesseur écrit la donnée dans le port sélectionné du 8255A.
- **Reset** : C'est un signal actif à l'état haut, il permet à la fois d'effacer le registre de contrôle et mettre les ports en mode entrée.
- **CS, A0, A1** : Ce sont les signaux de sélection. CS à l'état bas sélectionne le circuit 8255A et A0, A1 spécifient le port sélectionné ou le registre de contrôle comme le montre le tableau suivant :

<b>A<sub>1</sub></b>	<b>A<sub>0</sub></b>	<b>RD</b>	<b>WR</b>	<b>CS</b>	<b>Opération de lecture (Read)</b>
0	0	0	1	0	Le bus de données = Port A
0	1	0	1	0	Le bus de données = Port B
1	0	0	1	0	Le bus de données = Port C
<b>Opération de écriture (Write)</b>					
0	0	1	0	0	Le bus de données = Port A
0	1	1	0	0	Le bus de données = Port B
1	0	1	0	0	Le bus de données = Port C
1	1	1	0	0	Le bus de données = reg_com
<b>Le 8255 hors fonctionnement</b>					
X	X	X	X	1	Bus de donnée troisième état
X	X	1	1	0	Bus de donnée troisième état

Tableau VII.6 : adresses détaillant la sélection des ports

#### VII.4.1.2. Programmation de l'interface 8255 :

On peut programmer le 8255A selon trois modes :

Mode 0 : Entrée/sortie de base.

Mode 1 : Entrée sortie échantillonnée.

Mode 2 : Bus bidirectionnel.

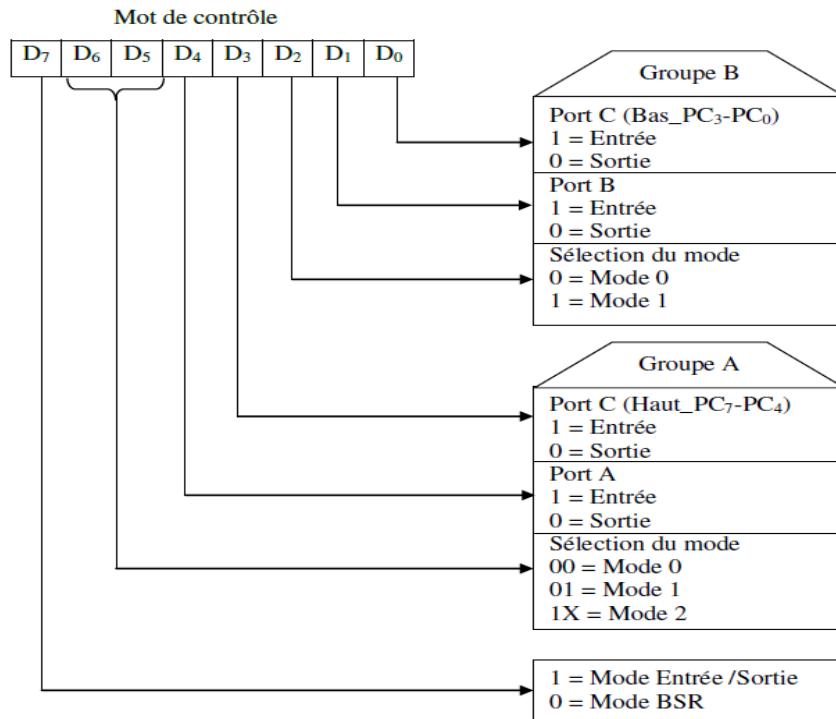


Figure VII.55 : Modes de programmation de l'interface 8255.

### Le mode 0 du 8255

En mode zéro les ports du 8255A peuvent être programmés en entrée ou en sortie : 8 bits pour le port A , 8 bits pour le port B et le port C est formé de deux quartes ( un quarte haut et un quarte bas ) , donc il y a 16 combinaisons possibles :

<b>Mot de commande</b>	<b>PORT A</b>	<b>PORT B</b>	<b>PORT C</b>	
			<b>PC<sub>4</sub> à PC<sub>7</sub></b>	<b>PC<sub>0</sub> à PC<sub>3</sub></b>
<b>10000000</b>	S	S	S	S
<b>10000001</b>	S	S	S	E
<b>10000010</b>	S	E	S	S
<b>10000011</b>	S	E	S	E
<b>10001000</b>	S	S	E	S
<b>10001001</b>	S	S	E	E
<b>10011010</b>	S	E	E	S
<b>10001011</b>	S	E	E	E
<b>10010000</b>	E	S	S	S
<b>10010001</b>	E	S	S	E
<b>10010010</b>	E	E	S	S
<b>10010011</b>	E	E	S	E
<b>10011000</b>	E	S	E	S
<b>10011001</b>	E	S	E	E
<b>10011010</b>	E	E	E	S
<b>10011011</b>	E	E	E	E

Tableau VII.7 : mode 0 du 8250

**Le Mode 1 du 8255A :**

En mode 1 , les ports A et B sont utilisés en entrée ou en sortie gérées par les accès du portC .

**Mode 1 : en entrée :**

En entrée les ports A et B sont configurés en entrée alors que le PC0, PC1 et PC2 assure le handshake pour B et PC3, PC4 et PC5 pour le A (PC6 et PC7 peuvent être programmés en entrée ou en sortie). STB : (Strobe input) : Au niveau bas , charge les données dans le verrou d'entrée . IBF : (Input Buffer Full) : Au niveau haut les données ont été verrouillées :

IBF = 1 par STB au niveau 0

IBF = 0 par le front montant d'un RD (read) INTR : (Interrupt Request) : Au niveau haut : demande d'interruption vers le microprocesseur :

INTR = 1 par STB=1, IBF=1 et INTE=1

INTR =0 par le front descendant d'un RD (read)

**Mode 1 : en sortie**

La figure VI.7 montre les signaux de contrôle quand les ports A et B sont configurés comme ports de sortie. Ces signaux sont définis comme suit :

- OBF (Output Buffer Full) : C'est un signal de sortie. Mis à l'état bas quand le microprocesseur écrit la donnée dans le latch de sortie du 8255A. Ce signal indique à un périphérique de sortie qu'une nouvelle donnée est prête pour la lecture. Il sera remis à l'état haut après la réception de ACK par le 8255A de la part d'un périphérique.
- ACK (Acknowledge) : C'est un signal d'entrée qui provient d'un périphérique qui doit envoyer un 0 quand il reçoit une donnée des ports du 8255A.
- INTR (Interrupt Request) : C'est un signal de sortie. Il est mis à 1 par un front montant du signal ACK . Ce signal peut être utilisé pour interrompre le microprocesseur pour demander la prochaine donnée. Le signal INTR est mis à 1 quand OBF, ACK et INTE sont tous à 1. INTR est remis à 0 au front descendant de WR.
- INTE (Interrupt Enable) : C'est une bascule interne du port, elle et a besoin d'être à 1 pour générer le signal INTR. INTEA et INTEB sont commandés par les bits PC6 et PC2 respectivement à travers le mode BSR.
- PC4,5 : Ces deux lignes peuvent être configurées en entrée ou sortie.

**Mode 2 : Transfert bidirectionnel des données**

Ce mode est utilisé en premier lieu dans des applications comme le transfert entre deux microordinateurs ou contrôleur d'interface des disques amovibles. Dans ce mode, le port A peut être configuré en mode bidirectionnel et le port B en mode 0 ou en mode 1. Le port A utilise cinq signaux du port C comme signaux handshake pour le transfert des données. Les

trois signaux restants du port C peuvent être utilisés comme des E/S simple ou comme un handshake pour le port B. La figure VI.9 illustre ce mode

### Mode BSR :

Ce mode concerne seulement les huit bits du port C, qui peuvent être mis à 1 ou à 0 par configuration du registre de contrôle. Un mot de contrôle avec le bit D7=0 décrit le mode BSR. Les opérations d'E/S des ports A et B ne sont pas affectées par le mode BSR. Pour la mise à 1 ou à 0 dans le port C, un mot de contrôle est écrit dans le registre de contrôle et non pas dans le port C. Le mot de contrôle est spécifié par ce qui suit :

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
0	X	X	X	Sélection des bits			S/R	
↓ Mode BSR	Non utilisés						↓	
				0	0	0	= Bit 0	
				0	0	1	= Bit 1	
				0	1	0	= Bit 2	
				0	1	1	= Bit 3	
				1	0	0	= Bit 4	
				1	0	1	= Bit 5	
				1	1	0	= Bit 6	
				1	1	1	= Bit 7	

Tableau VII.8 : Registre de contrôle

### Exemple :

Ecrire un sous programme du mode BSR pour la mise à 1 des bits PC7 et PC3 et ensuite leurs

remise à 0 après 10ms. On suppose que le délai 10ms est disponible.

Pour la mise à 1 du bit PC7, le mot de contrôle = 0 0 0 0 1 1 1 1 = 0FH

Pour la mise à 0 du bit PC7, le mot de contrôle = 0 0 0 0 1 1 1 0 = 0EH

Pour la mise à 1 du bit PC3, le mot de contrôle = 0 0 0 0 0 1 1 1 = 07H

Pour la mise à 0 du bit PC3, le mot de contrôle = 0 0 0 0 0 1 1 0 = 06H

L'adresse du registre de contrôle = 83 H (figure VI.2)

Sous programme :

BSR: MVI A,0FH ; Charger l'octet 0FH dans l'accumulateur

OUT 83H ; PC7=1

MVI A,07H ; Charger l'octet 07H dans l'accumulateur

OUT 83H ; PC3=1

CALL DELAY ; Délai de 10ms

## Chapitre VI Systèmes à microprocesseurs

73

MVI A,06H ; Charger l'octet 06H dans l'accumulateur

OUT 83H ; PC3=0

MVI A,0EH ; Charger l'octet 0EH dans l'accumulateur

OUT 83H ; PC7=0

RET

**VII.4.2. Interface série**

Malgré sa rapidité la liaison parallèle n'est pas adaptée à quelque application lors de la communication entre le microprocesseur et ses périphériques, en effet ce type de liaison a des faiblesses lorsqu'il s'agit de communiquer avec un organe pour une longue distance, à cet effet on préfère utiliser la liaison série malgré son faible débit contre la liaison parallèle.

Une interface série permet d'échanger les informations bit par bit entre le microprocesseur et la périphérique ,

La liaison entre l'émetteur et le récepteur en mode série peut être :

- Simplexe.
- Half duplexe
- Full duplex

La figure suivante montre la différence entre les trois modes

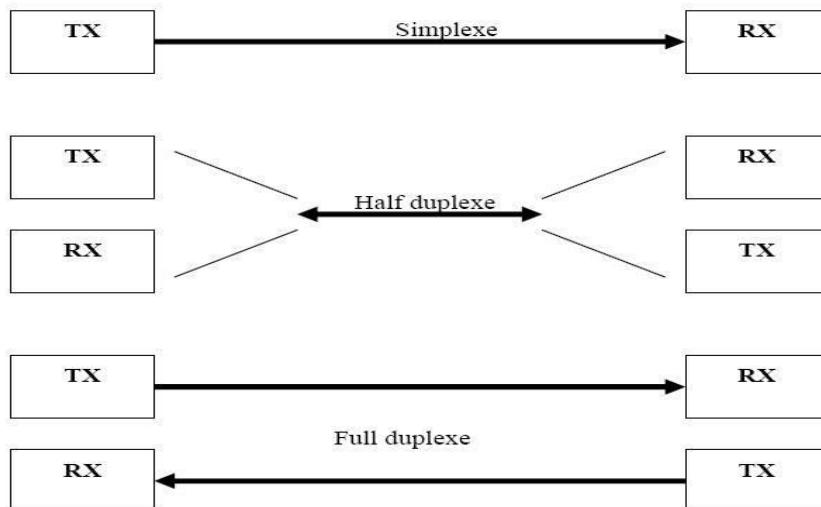


Figure VII.56 : modes d'interface série

TX : ligne d'émission. RX : ligne de réception.

La vitesse de transmission est mesurée en bps (bits par seconde), mais aussi BAUD (nombre des changements du signal par Seconde). Il existe 2 types de Communications Série: Synchrone et Asynchrone :

### I-1 ) Communications asynchrones

Ici la transmission s'effectue caractère par caractère.

Un caractère comprend :

- un bit de départ (START),
- des bits de donnée selon un format variable (5, 6, 7,8 bits),
- un bit de parité (optionnel),
- un ou plusieurs bits d'arrêt (Stop).

### I-2 ) Communication synchrone :

Avec la communication synchrone les données sont transmises d'une manière continue. Il est donc nécessaire d'effectuer la synchronisation des caractères au début du bloc de données. Cette synchronisation peut être interne ou externe :

La synchronisation interne : le récepteur détecte un ou plusieurs caractères de synchronisation en tête de message, en cherchant une correspondance bit à bit entre un caractère de synchronisation préétabli et le train de bits arrivant.

La synchronisation externe : nécessite une ligne spécialisée qui fournit un top d'horloge à chaque début de transmission de bloc.

Remarque 1 :

Dans tous les cas, une horloge cadence la transmission des bits qui ne sont donc émis qu'à des instants déterminés.

Remarque 2 :

Il existe deux circuits pour les Communications Série: UART (Universal Asynchronous Receiver-Transmitter, (les Ports COM d'IBM PC utilisent l'UART8050/16550 de NS ) et USART (Universal Synchronous- Asynchronous Receiver-Transmitter (Le 8251 de Intel).

Exemple interface série 8250

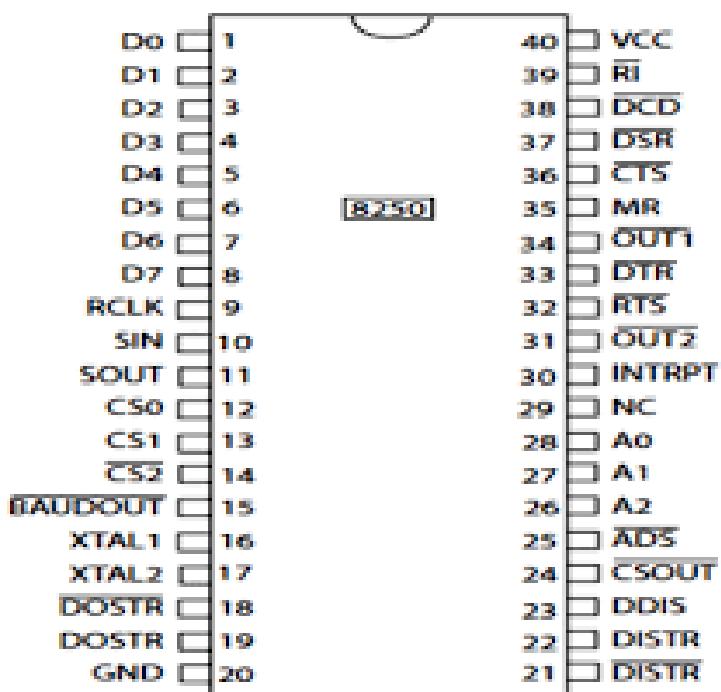


Figure VII.57 : Brochage de l'interface série 8250

N° pin	Nom	Notes
Pin 1:8	D0:D7	Bus de données
Pin 9	RCLK	Receiver Clock Input. La fréquence de cette entrée doit être égale au baud du réception * 16.
Pin 10	RD	Ligne de réception de données
Pin 11	TD	Ligne de transmission de données
Pin 12	CS0	Chip Select 0 - Active haut
Pin 13	CS1	Chip Select 1 - Active Haut
Pin 14	CS2	Chip Select 2 - Active bas
Pin 15	nBAUDOUT	Baud Output = (Baud Rate x 16)
Pin 16	XIN	External Crystal Input
Pin 17	XOUT	External Crystal Output
Pin 18	nWR	Write Line - Inverted
Pin 19	WR	Write Line - Not Inverted
Pin 20	VSS	GND
Pin 21	RD	Read Line - Inverted

Pin 22	nRD	Read Line -
Pin 23	DDIS	= 0 lorsque le microprocesseur lit les données à partir du 8250, = 1 pour inhiber un transceiver se trouvant sur le bus de données.
Pin 24	nTXRDY	Transmit Ready
Pin 25	nADS	Address Strobe. Utiliser lorsque les signaux ne sont pas stables pendant le cycle d'écriture ou de lecture.
Pin 26	A2	Address Bit 2
Pin 27	A1	Address Bit 1
Pin 28	A0	Address Bit 0
Pin 29	nRXRDY	Receive Ready
Pin 30	INTR	Interrupt Output
Pin 31	nOUT2	User Output 2
Pin 32	nRTS	Request to Send

Pin 33	nDTR	Data Terminal Ready
Pin 34	nOUT1	User Output 1
Pin 35	MR	Master Reset
Pin 36	nCTS	Clear To Send
Pin 37	nDSR	Data Set Ready
Pin 38	nDCD	Data Carrier Detect
Pin 39	nRI	Ring Indicator
Pin 40	VDD	+ 5 Volts

Tableau VII.9 : brochage d'interface 8250 [17]

#### VII.4.3. Ports USB (Universal Serial Bus)

Le nouveau Bus USB de connexion série, promu par les principaux constructeurs informatiques et éditeurs de logiciels (Compaq, DEC, IBM, Intel, Microsoft, NEC, Nortel) et dont les premières spécifications doivent permettre le raccordement sur une prise unique de plusieurs équipements divers (imprimante, téléphone, modem, fax, clavier, souris, scanners, écrans...). On désire ainsi éviter la multiplication actuelle des connecteurs sur les PC. Ce bus permet la transmission de données en série. Cette nouvelle technique se doit d'être rapide, bidirectionnelle, synchrone, de faible coût et l'attachement d'un nouveau périphérique doit être dynamique. De plus l'alimentation des équipements est possible.

Le débit brut (émission plus réception) peut aller jusqu'à 12 Mbit/s au maximum. Le débit brut doit être partagé entre tous les appareils connectés au bus

Le port USB possède 4 broches à savoir : une paire d'alimentation (VCC et GND) et une paire torsadée de données inversés (-DATA et +DATA). Au repos +DATA et -DATA sont à l'état haut et bas respectivement

### Topologie d'un réseau USB

Un réseau USB a une topologie en étoile.

- Le port USB est contrôlé entièrement par un contrôleur unique appelé hôte (“host”). Souvent le PC, il initie toutes les communications, et est le maître absolu du bus.
- Les “hubs” permettent de relier plusieurs appareils à un seul port USB.
- Le rôle principal des hubs est de transférer les données de l'hôte aux périphériques.
- Chaque hub contrôle ses ports afin de savoir si un appareil s'y connecte
- Il peut y avoir 5 niveaux de hub en plus du hub racine.
- Il y a 127 appareils maximum dans un réseau USB.

Chaque appareil a son adresse.

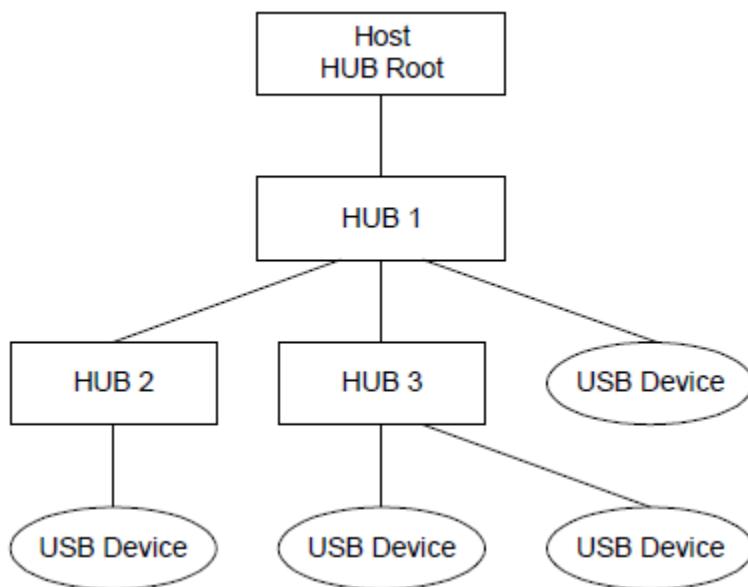


Figure VII.58 : topologie d'un réseau USB

### VII.5. Application sur l'interface parallèle 8255 :

1. Identifier les adresses du port de la figure suivante :

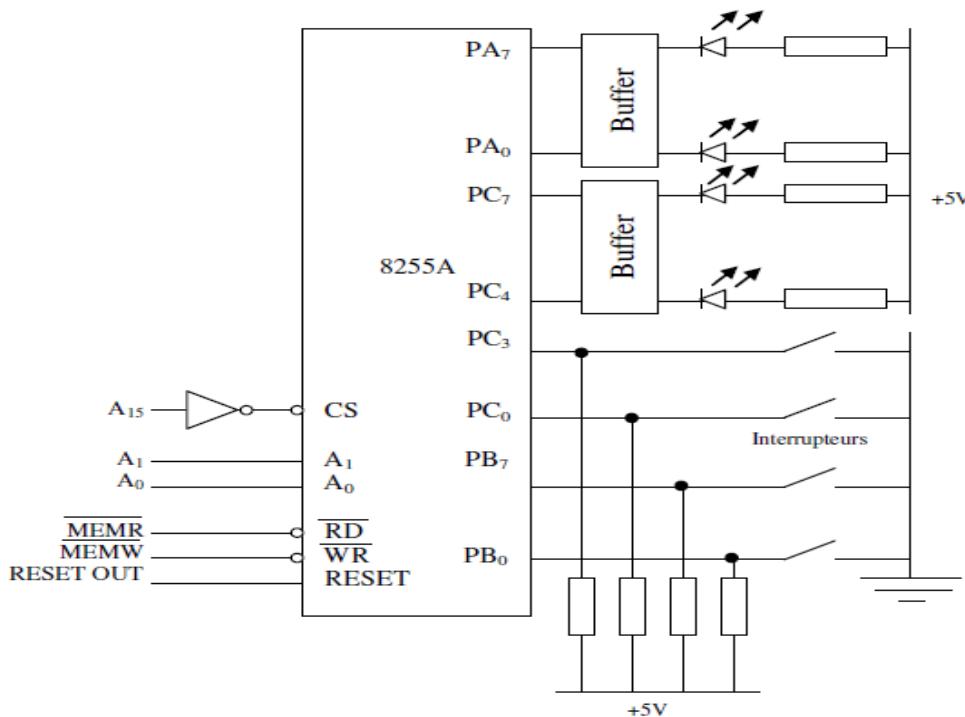


Figure VII.59 : Exemple d'application du 8255A en mode 0 en entrée et en sortie [18]

2. Identifier le mot de contrôle du mode 0 pour configurer le port A et le port CHaut en tant que ports de sortie et le port B et le port CBas en tant que ports d'entrée.
3. Ecrire le programme pour lire les états des interrupteurs et afficher la lecture du port B au port A et la lecture du port CHaut au port CBas.

Solution :

1. Adresses des ports et registre de contrôle :

Port A : 8000H (A1=0 et A0=0)

Port B : 8001H (A1=0 et A0=1)

Port C : 8002H (A1=1 et A0=0)

Registre de contrôle : 8003H (A1=1 et A0=1)

2. Mot de contrôle :

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	=83H
1	0	0	0	0	0	1	1	

3. Programme :

MVI A,83H ; Charger l'accumulateur avec le mot de contrôle

STA 8003H ; Ecrire le mot dans le registre de contrôle pour initialiser les ports

LDA 8001H ; Lire les interrupteurs du port B

STA 8000H ; Afficher la lecture au port A

LDA 8002H ; Lire les interrupteurs du port C

ANI 0FH ; Masquer les quatre bits de poids fort du port C ; Ces bits ne sont pas des entrées

RLC ; Faire la rotation et placer la donnée au poids fort de l'accumulateur

RLC

RLC

RLC

STA 8002H ; Afficher la lecture au port C de poids fort

HLT

### **VII.5. Conclusion :**

Dans ce chapitre nous avons expliqué le principe et les modes du fonctionnement des interfaces entrées-sorites suivant les protocoles adoptées par le constructeur et la configuration du programmeur, mais la gestion des interfaces continue à développer en fonction du nombre et du type des interfaces grâce à l'évolution de la technologie numérique notamment actuellement qui est développée par l'intégration de l'intelligence artificielle et la science des données.

**Chapitre VIII :****Les interruptions****VIII.1. Introduction :**

Le traitement des données est généralement difficile lorsque les événements sont nombreux et complexes, ensuite il y a quelques fois des événements cruciaux comme la coupure d'alimentation ou un composant électronique indispensable tombe défectueux, tous ces facteurs donnent la notion des interruptions qui nécessite une méthode de gestion appropriée.

**VIII.2. Définition**

Une interruption est un évènement apparu lorsque l'interface d'entrée-sortie délivre ou reçoit une information exigeant un empêchement d'un programme du traitement exécuté par le microprocesseur, dans ce cas le microprocesseur doit exécuter l'interruption comme sous-programme suivant un mécanisme ou protocole exigé par le constructeur ou le programmeur. Un système à base de microprocesseur peut comporter plusieurs périphériques tels que : un Ecran, une souris, une imprimante, un disque dur, un convertisseur numérique analogique (CNA) ,..... etc

**VIII.3. Protocoles d'échanges de données :**

Le dialogue avec les périphériques se fait par trois choix de communication :

- En questionnant le périphérique de façon continue pour vérifier que des données peuvent être lues ou écrites (Polling).
- En établissant une communication directe entre un périphérique et la mémoire (DMA : Direct Memory Access).
- En l'interrompant le programme principal lorsqu'un périphérique est prêt à lire ou écrire des données (interruption)

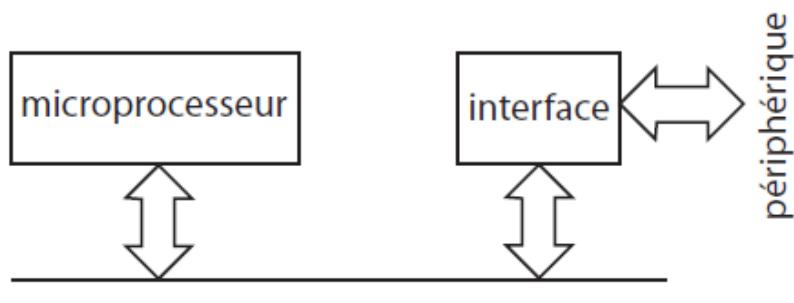


Figure VIII.60 : Interconnexion microprocesseur-interface.[19]

**VIII.3.1. La scrutation périodique ( Polling )**

Le programme principal contient des instructions qui lisent cycliquement ou systématiquement l'état des ports d'E/S ou un test d'état périodique des périphériques  
Avantage : facilité de programmation.

Inconvénients :

- perte de temps s'il y a de nombreux périphériques à interroger ;
- de nouvelles données ne sont pas toujours présentes ;
- Les données peuvent être perdues si elles se changent rapidement

### VIII.3.2. Accès direct à la mémoire : DMA (Direct Memory Access) :

En cas de transfert de données de grande quantité entre deux périphériques, il peut être plus efficace de laisser la communication entre ces deux périphériques plutôt que de solliciter le processeur. Ce type de communication avec les entrées/sorties s'appelle gestion de transfert par DMA, Le processeur indique au contrôleur de DMA quels sont les périphériques qui doivent communiquer, le nombre et éventuellement l'adresse des données à transférer, puis il initie le transfert. Le processeur n'est pas sollicité durant le transfert, et une interruption signale au processeur la fin du transfert.

### VIII.3.3. Interruption :

Lorsqu'une donnée apparaît sur un périphérique, le circuit d'E/S le signale au microprocesseur pour que celui-ci effectue la lecture de la donnée : c'est une demande d'interruption (IRQ : Interrupt Request) :

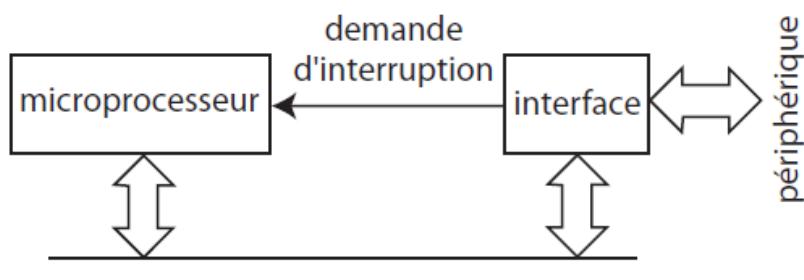


Figure VIII.61 : la demande d'interruption [20]

Avantage : le microprocesseur effectue une lecture des ports d'E/S seulement lorsqu'une donnée est disponible, ce qui permet de gagner du temps et d'éviter de perdre des données.  
Les demandes d'interruptions sont divisées en deux types : interruption masquable et interruption non masquable soit d'une source matérielle ou logicielle.

### VIII.4. La classification des interruptions :

Les interruptions sont subdivisées en deux catégories selon la source:

#### a- les interruptions matérielles :

Une interruption matérielle est une interruption signalée au processeur par un évènement matériel sur une borne spéciale, la demande d'interruption est effectuée par une interface E/S, la gestion des interruptions matérielles s'effectue par un circuit contrôleur ( PIC :Programmable Interrupt Controller ), le processeur traite l'interruption dès la fin de l'instruction qu'il était en train d'exécuter. Le traitement de l'interruption consiste soit :

- A l'ignorer et passer normalement à l'instruction suivante : c'est possible uniquement pour certaines interruptions, nommées interruptions masquables , il est en effet parfois nécessaire de pouvoir ignorer les interruptions pendant un certains temps, pour effectuer des traitements très urgents par exemple. Lorsque le traitement est terminé, le processeur démasque les interruptions et les prend alors en compte.
- A exécuter l'"interruption (interruption non masquable), un traitant d'interruption est un programme qui est appelé automatiquement lorsqu'une interruption survient, l'adresse de début du traitant est donnée par la table des vecteurs d'interruptions .Lorsque le programme d'interruption traitant a effectué son travail, il exécute l'instruction spéciale IRET qui permet de reprendre l'exécution à l'endroit où elle avait été interrompue,

### b- Interruptions du logiciel

Une interruption logicielle représente un déclenchement par une instruction spéciale générée par un programme d'application ou par un système d'exploitation, autrement dit une interruption logicielle est un arrêt de l'exécution d'un programme pour exécuter une routine d'interruption du BIOS ou DOS (pour le microprocesseur 8086 à 16 bits ))

Le microprocesseur 8086 à 16 bits peut gérer jusqu'à 256 interruptions. Chaque interruption reçoit un numéro compris entre 0 et 255 appelé type de l'interruption.

Trois sortes d'interruptions sont reconnues par le 8086 :

- Interruptions matérielles produites par l'activation des lignes INTR et NMI du microprocesseur ;
- interruptions logicielles produites par l'instruction INT n, où n est le type de l'interruption
- Interruptions processeur générées par le microprocesseur en cas de dépassement, de division par zéro ou lors de l'exécution pas à pas d'un programme

Les interruptions du 8086 sont vectorisées. La table des vecteurs d'interruptions doit obligatoirement commencer à l'adresse physique 00000H dans la mémoire centrale. Chaque vecteur d'interruption est constitué de 4 octets représentant une adresse logique du type CS : IP.

### VIII.5. Interruptions masquables et interruptions non masquables

Lorsque certaines séquences critiques d'un programme sont en cours d'exécution, on peut vouloir interdire toute prise en compte d'une interruption quelconque qui risquerait de le perturber gravement. Cette interdiction s'appelle le masquage des interruptions (interruptions masquables). La méthode courante consiste à positionner un bit d'état, ou plusieurs bits d'état dans le mot d'état du microprocesseur. Il s'agit des bits de masquage. On les bascule du mode inactif au mode actif. Dès lors, le microprocesseur refuse toute prise en compte d'interruption. Le positionnement de ces bits d'état se fait par logiciel, via une commande insérée au bon endroit dans un programme. La séquence de principe du programme est ainsi la suivante :

1. Positionnement des indicateurs de masquage des interruptions pour les interdire.
2. Exécution d'une séquence critique du programme.
3. Démasquage des interruptions qui peuvent de nouveau librement intervenir.

Une autre méthode de masquage consiste à verrouiller le codeur de priorités ou le circuit de gestion des interruptions s'il a été prévu pour cette fonction.

Une interruption masquable se fait généralement en positionnant un flag dans le registre d'état. Il existe aussi une interruption non masquable qui est toujours prise en compte par le microprocesseur comme interruption prioritaire exemple : Reset.

### VIII.6. La gestion des interruptions

- Le fonctionnement des périphériques se fait en général d'une manière asynchrone, donc plusieurs interruptions peuvent être déclenchées à m'importe quel instant sans référence du système d'horloge, il existe souvent un contrôleur d'interruptions destiné à gérer les conflits entre interruptions.
- Lorsqu'un périphérique reçoit des données pour le processeur, il envoie au processeur un signal d'interruption. Si celui-ci peut être interrompu (à condition qu'il ne soit pas en train de communiquer avec un périphérique de plus haute priorité), il stoppe la tâche en cours, sauve son état (ces registres) en mémoire (la zone pile) et appelle la routine correspondant au numéro d'interruption.
- les interruptions peuvent être générées par le microprocesseur lui-même en cas de problèmes tels qu'une erreur d'alimentation, une division par zéro ou un circuit mémoire défectueux (erreurs fatales). Dans ce cas, la demande d'interruption conduit à l'arrêt du microprocesseur

### VIII.7. Processus de traitement d'une interruption,

Pour que le microprocesseur exécute une interruption, il se déroule par certaines étapes :

- Terminer l'instruction du programme en cours.
- Sauvegarder l'état du processeur.
- Interdire les interruptions du niveau moins élevé.
- Accéder au registre d'interruption en service (ISR : In Service Registre)
- Exécuter l'interruption dans ISR.
- Restituer l'état du processeur
- Reprendre l'exécution du programme abandonné.

### VIII.8. Interruptions du microprocesseur 8085

Le microprocesseur 8085 possède un service d'interruptions [20]:

#### a- Interruptions logicielles données par l'instruction REST

Le microprocesseur 8085 à 8 bits comprend huit instructions RST (restart) , Ce sont des instructions à un seul octet. Ces instructions sont des appels qui permettent de transférer l'exécution des programmes à des positions mémoire bien spécifiées. Un retour au programme principal sera assuré par la récupération du CP de la pile dès l'exécution de l'instruction RET.

Interruption	Adresse
REST0	CALL 0000H
REST1	CALL 0008H
REST2	CALL 0010H
REST3	CALL 0018H
REST4	CALL 0020H
REST5	CALL 0028H
REST6	CALL 0030H
REST7	CALL 0038

Tableau VIII.10 : Interruptions logicielles du 8085

**Interruptions matérielles** produites par l'activation d'au moins de l'une des cinq lignes RST7.5, RST6.5, RST5.5, INTR et TRAP du microprocesseur, ce type d'interruptions sont reparties en deux catégories : vectorisées et non vectorisées.

#### Interruption non vectorisée INTR

Dès la réception d'un signal INTR de la part d'une interface d'entrée, le microprocesseur envoie un accusé de réception sur la ligne INTA pour indiquer à l'interface que l'interruption a été acceptée (si EI=1). La position mémoire du sous programme d'interruption (ISR) est inconnue. Donc c'est à l'interface d'envoyer au microprocesseur l'adresse de l'ISR.

## Interruptions vectorisées

Le 8085 possèdent quatre interruptions vectorisés par ordre de priorité : TRAP (Interruption non masquable), RST 7.5, RST 6.5, RST 5.5. Les positions des sous programmes d'interruption sont comme suit :

**TRAP** est reconnu par NMI (Non Maskable Interrupt). Cette interruption ne peut être inhibée. C'est une interruption utilisée dans les situations critiques comme problème d'alimentation

### RST 7.5, RST 6.5, et RST 5.5 :

Ce sont des interruptions masquables activés par EI ( Enable Interrupt ) et SIM (Set Interrupt Mask) dont la description est donnée en ce qui suit.

7	6	5	4	3	2	1	0
SOD	SDE	XXX	R7.5	MSE	M7.5	M6.5	M5.5

SOD= Donnée de sortie série ( Serial output Data) , ignoré si D6=0

SDE= autorisation de donnée série ( Serial Data Enable ) si D6=1 ; le bit 7 est envoyé.

XXX= ignoré

R7.5= Reset RST7.5 , si 1 ,la bscule RST7.5 est en bas.

MSE= autorisation maquée ( Masked Set Enable ) sit D3=0 les bits 0-2 ignorés , si D3=1, le maquage est défini

M= Masked ( si est 1 RST7.5 , RST6.5 et RST5.4 sont masqués , si 0 sont non masqués).

### Exemple :

1. Pour accepter les interruptions dans un système à base 8085.

EI

MVI A,08H

SIM

2. Pour la mise à 0 de l'interruption 7.5

MVI A, 18H

SIM

## Interruptions en attente

Le fait qu'il existe plusieurs lignes d'interruption, quand une interruption est servie, l'autre interruption doivent attendre. Le 8085 possède une deuxième instruction appelée RIM (Read Interrupt Mask) pour savoir les interruptions en attente.

### \* Instruction RIM

C'est une instruction à un seul octet. Cette instruction est définie comme suit :

7	6	5	4	3	2	1	0

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
-----	------	------	------	----	------	------	------

SID=bit de donnée d'entrée série

Interruption en attente (I7.5, I6.5, I5.5)=1

Etat d'activation d'une interruption (IE) =1 activé, 0 non activé

Masquer des interruptions : M7.5, M6.5 ,M5.5 =1 masqué

- Pour lire les masques d'interruptions, cette instruction RIM charge l'accumulateur avec 8 bits en indiquant le masque d'interruption.
- Pour identifier les interruptions en attente, les bits D4, D5 et D6 identifient les interruptions en attente.
- Pour recevoir une donnée série ,le bit D7 est utilisé pour cet événement.

### Exemple :

On suppose que le 8085 est entrain d'exécuter une demande d'interruption RST 7.5. Une vérification de l'attente de RST 6.5 par exemple. S'il y a une attente, une activation de RST 6.5 doit être faite sans l'affectation des autres interruptions, sinon retour au programme principal.

```

RIM ; Lecture du masque d'interruptions
MOV B,A ; Sauvegarder le masque dans B
ANI 20H ; Vérifier s'il y a une attente de RST 6.5
JNZ SUIVANT
EI
RET ; RST 6.5 non pas en attente, retour au programme principal
SUIVANT: MOV A,B ; RST est en attente
ANI 0DH ; Valider RST 6.5 par mise à 0 de D1
ORI 08H ; Valider SIM par mise à 1 de D3
SIM
JMP SERV ; Aller exécuter le service de la routine RST 6.5

```

### VIII.9. Contrôleur programmable d'interruptions

The 8259A est un circuit intégré permettant de gérer les interruptions à travers les signaux (INTR/ INT) avec le microprocesseur 8085, ses caractéristiques sont [21]:

- Il gère 8 demandes d'interruptions.
- Il peut vectoriser une demande d'interruption dans n'importe quelle case dans la mémoire via un control programme sans ajouter un matériel hardware, cependant les 8 demandes d'interruptions sont positionnées dans l'intervalle 4 endroits ou 8 endroit.
- Il peut résoudre 8 niveaux d'interruptions prioritaires en modes différents.

- Si on ajoute un autre circuit identique, on peut gérer jusqu'à 64 niveaux.

La grande restriction de ce circuit est que toutes les demandes sont localisées dans la page de l'adresse 00H qui est réservée pour ROM ou EPROM et l'accès à ces endroits est difficile après le système est dimensionné, en plus le processus de sélectionner les priorités est limité et un extra-hardware est nécessaire pour insérer les instructions Restart. Le 8259 surmonte ces contraintes et fournit plusieurs options flexibles . il peut être utilisé avec un microprocesseur 16 bits comme Intel 8086/8088.

Le contrôleur 8259A porte une autre caractéristique additionnelle telle que la lecture des règles et le changement du mode d'interruption durant le programme d'exécution.

#### Le brochage :

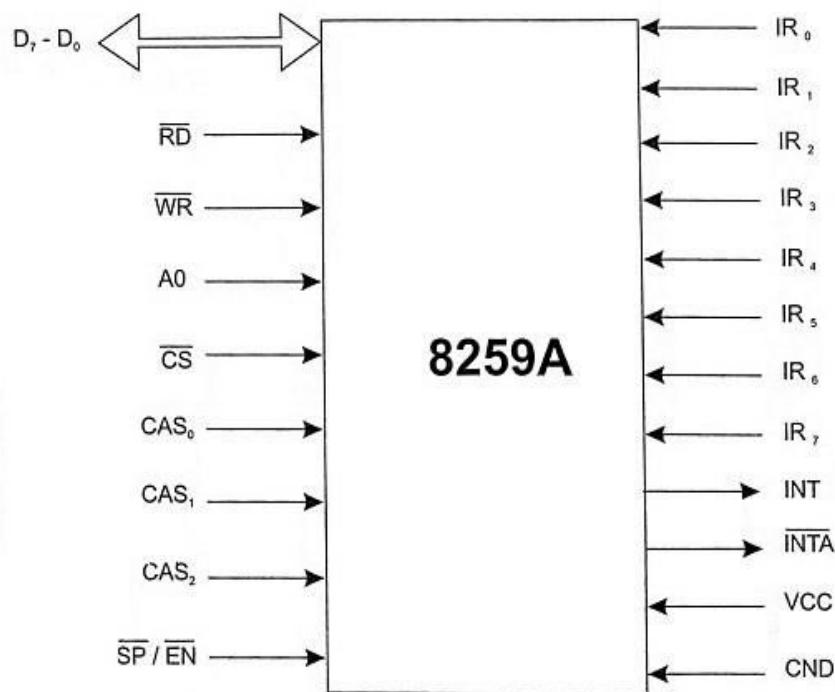


Figure VIII.62 : Brochage du contrôleur 8259A

**A0** : entrée pour sélectionner une de deux adresses internes.

**CS** : chip select , un signal pour activer ou désactiver le composant.

**RD et WR** : sont deux signaux pour lecture et écriture respectivement.

**INT** : signal d'interruption

**INTA** : signal d'accusé d'interruption.

**VCC et GND** : signal d'alimentation et la masse respectivement.

**CAS0, 1, 2:** Utilisées dans la mise des PIC8259s en cascade pour étendre le nombre d'interruptions du microprocesseur jusqu'à 64. En mode Esclave, les CAS0,1,2 sont ignorés.

**IR0...7** : sont des entrées de requête (Interrupt Request) ; suivant l'initialisation, on peut

choisir des entrées sensibles au niveau haut ou au front montant.

**SP/EN** (Slave Programming/Enable): (SP/EN = '1' Maître, SP/EN = '0' Esclave)

**D0...D7** : bus de données.

Le 8259A contient une logique du control, des registres pour les demandes d'interruptions, sélectionneur de priorités, logiques en cascade pour additionner un autre contrôleur 8259A, et bus de données , ces blocs sont illustrés la figure suivante :

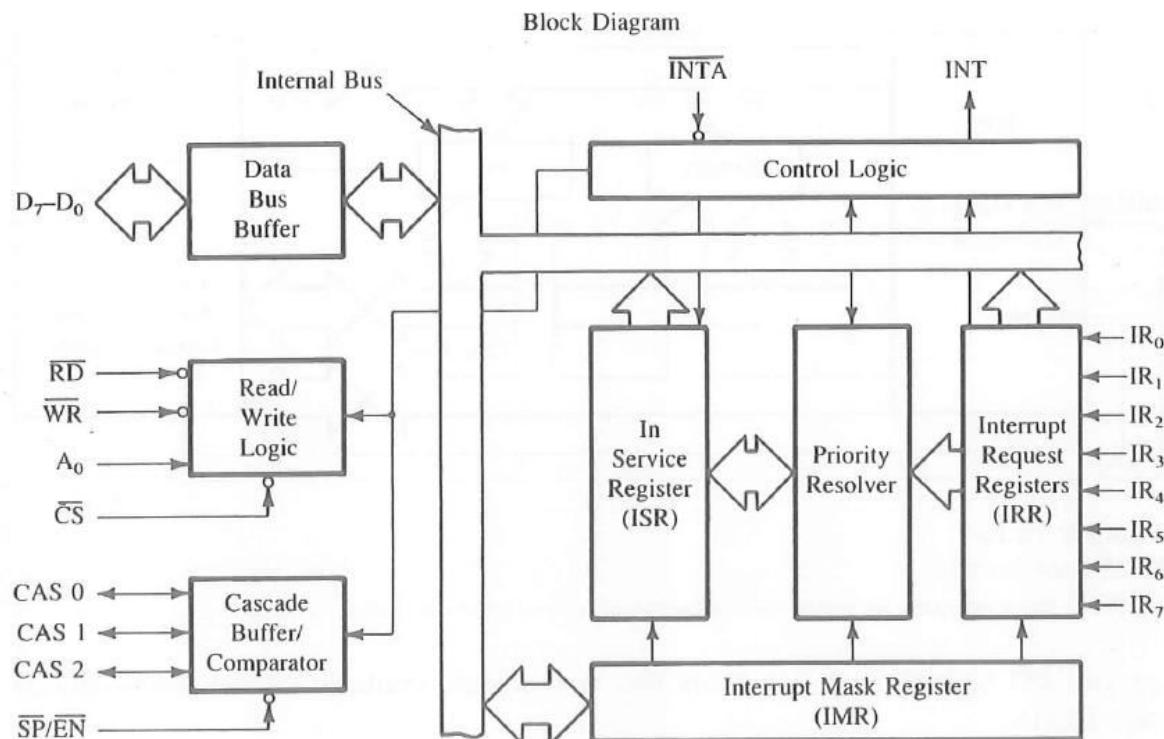


Figure VIII.63 : architecture interne du contrôleur 8259A

**Remarque :** plusieurs détails concernant l'initialisation, la configuration, procédure de programmation des interruptions peuvent être donnés dans un cours spécial sur le contrôleur 8259A.

### VIII.10. Conclusion :

Dans ce chapitre, nous avons donné les principes de base des interruptions utilisées dans un système à microprocesseur, la gestion des interruptions suivant les priorités programmées par le constructeur ou exigées par le système d'exploitation, ensuite l'encombrement des interruptions durant l'opération du système à microprocesseur exige un microcontrôleur chargé pour faire la gestion des différents évènements.

## ***Conclusion Générale***

Dans ce manuscrit, nous avons présenté en détails la description, l'architecture et le fonctionnement du système à microprocesseur à 8 bits comme une connaissance de base permettant l'étudiant de maîtriser les microprocesseurs pour passer au niveau supérieur.

Le système à microprocesseur est le système de base en électronique digitale et introduit dans nombreux d'applications technologiques et informatiques, en addition l'intelligence artificielle vient de permettre d'élargir l'exploitation du système à microprocesseur dans les domaines techniques, énergétiques et économiques.

- Mais il est intéressant d'attirer l'attention que les microprocesseurs sont différents suivant leurs architectures et caractéristiques et leurs tailles de transfert des données , dans ce cas l'étudiant doit enrichir sa connaissance par d'autres références et documents.
- Ce manuscrit permet l'étudiant de traverser vers l'étude des microcontrôleurs et ses différentes applications notamment des cartes de développement comme la famille des Arduino , les cartes FPGA , les cartes Rasph Peri ....etc.
- Il est important pour l'étudiant d'apprendre la technique de programmation d'un microprocesseur par l'un des langages connus comme Assembleur, langage C , PIC basic, miKro C ,.....etc
- La maîtrise du système à microprocesseur est le fondement des systèmes électroniques embarqués.

## Références

- [01] Etienne Messerli Yves Meyer , Electronique numérique : Systèmes combinatoires 1 er Tome , HES-SO 2010
- [02] Camille Diou ,Cours d'électronique numérique ,Université Paul-Verlaine- Mertz 2009
- [03] Catherine Douillard ,Gérald Ouvradou ,Michel Jézéquel, Logique séquentielle , techniques d'intégration , Ecole Nationale Supérieure des Télécommunications de Bretagne ,2006
- [04] Mabrouk Issam , Microprocesseurs et Microcontrôleurs , support des cours ,Institut supérieur des études technologiques , Tunisie , 2014.
- [05] Ben Amar M & Gaaloul K. cours des systèmes logiques , ISET Nabeul , 2015
- [06] T. Dumartin , Architecture des ordinateurs , Note de cours 2005
- [07] Eric Magarotto , cours d'informatique industrielle ,Université de Caen 2006.
- [08] Philippe Breton, Une histoire de l'informatique. Paris, éditions du Seuil , 1990.
- [09] Nabil Boukhennoufa , Systèmes à microprocesseurs Cours & Exercices , université Mohamed Boudiaf – M'sila.
- [10] John Crisp, Introduction to Microprocessors and Microcontrollers, Elsevier; Second Edition 2004
- [11] Steve Health, Microprocessor Architectures and Systems RISC CISC and DSP , BHNEWNES , OXFORD 1991
- [12] Abid Sabeur, les microprocesseurs à 8 bits ,Ecole supérieure des sciences et techniques 2003.
- [13] Abdallah Zegaoui ,les microprocesseurs & les microcontrôleurs ,université Hassiba Ben Bouali chlef , polycopié tome 1 , 2011.
- [14] Guichi Amar , Cours de Systèmes à microprocesseurs, polycopié , université de M'sila,2020
- [15] Philippe Preux, Méthodologie de programmation en assembleur , 1997.
- [16] Jlassi khaled , Microprocesseurs et microcontrôleurs , Université Virtuelle de Tunis,2008
- [17] Mustapha Abdel El-barr et Hesham El-Rewini, Fundamentals of computer , organization and Architecture , Wiley Interscience ,2005.
- [18] R.C Holland, Microprocessors and their operating systems , APPLIED ELECTRICITY AND ELECTRONICS SERIES , PERGAMON PRESS, UK ,1989.
- [19] Tayari Lassaad „Support de cours Microprocesseurs ,ISET GABES , Tunisie ,2014.
- [20] YOSRA RKHSSI KAMMOUN, Cours microcontrôleurs, Polycopie ISET Nabeul,2017
- [21] Ramesh S.Gaonkar ,Microprocessor Architecture , Programming and Applications with the 8085 , Fifth Edition , Prentice Hall ,2002.