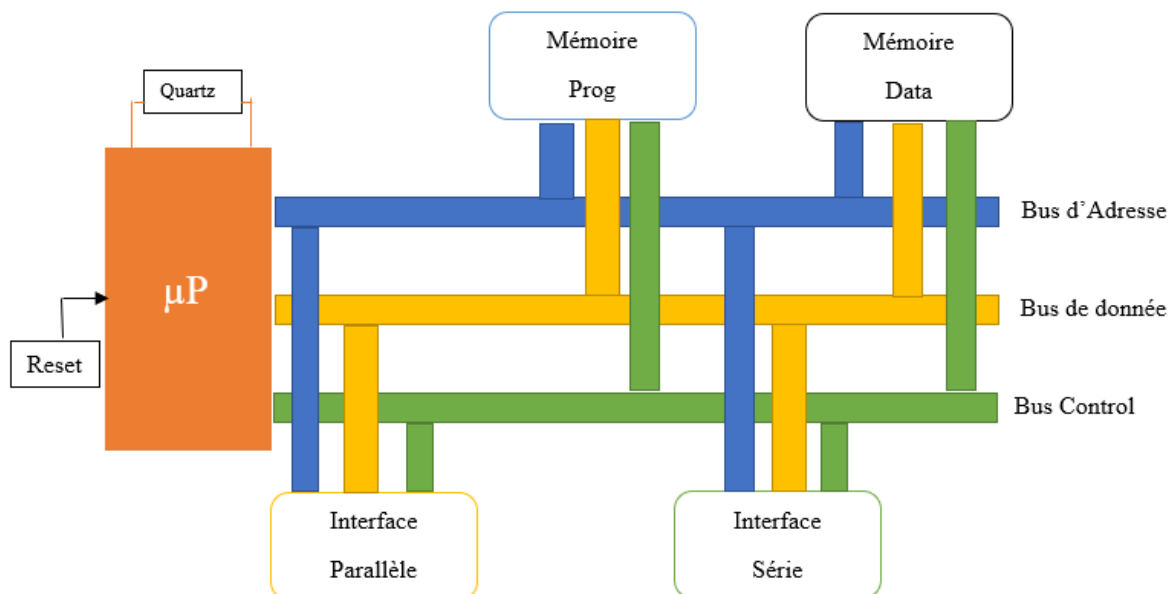


Chapitre 2.

Structure d'un système à μP



2.1 Structure externe du μP :

- Les bus du μP permettent à ce dernier de communiquer avec les circuits mémoires et circuits interfaces. Pour chaque μP on distingue 3 types de bus :
 - 1- Bus d'adresse : Unidirectionnel (μP \rightarrow mémoire et interfaces) de niveau TTL, la longueur du Bus (nombre de lignes) permet de déterminer la capacité d'adressage du μP (exp I8085 : bus d'adresse 16 lignes A_0-A_{15} d'où une capacité d'adressage $2^{16}=64Ko$)
 - 2- Bus de donnée : Bidirectionnel (μP \leftrightarrow mémoire et interfaces) de niveau TTL, la longueur du bus détermine la longueur de mots d'opération dans l'UAL.

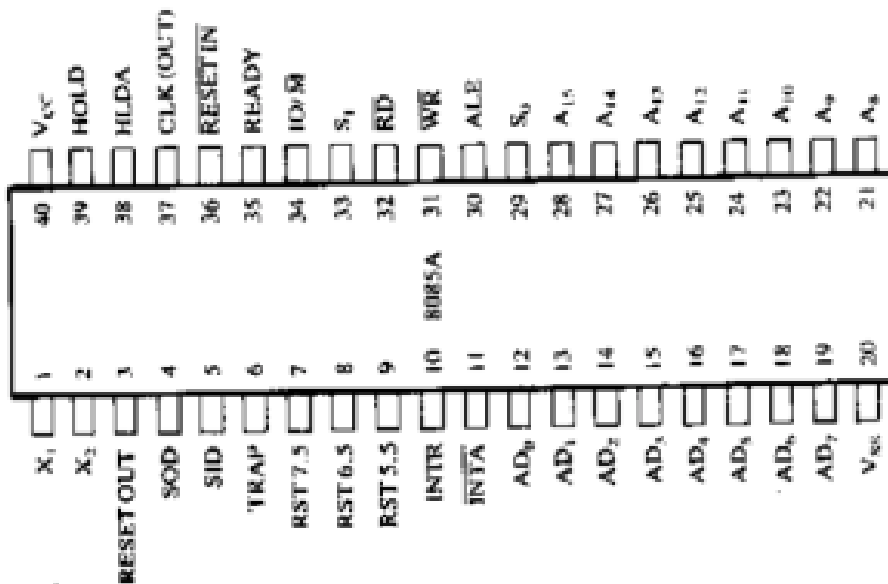
Exp : Cas du I8085 \rightarrow bus data 8 bits niveau TTL.

- 3- Bus de control : constitué de quelques lignes de niveau TTL, Unidirectionnel (μP vers mémoire et périphériques : R/W , IO, M, ENABLE) (périphérie vers μP : interruption)
 - R : Read (demande de lecture : μP \xrightarrow{Data} mémoire à interface)
 - W : Write (demande d'écriture : μP \xrightarrow{Data} mémoire ou interface)
 - Enable : activer un circuit (mémoire ou interface)
 - IO : sélection interface (μP \rightarrow interface)
 - M : sélection mémoire (μP \rightarrow mémoire)
 - Int : demande d'interruption (μP \rightarrow interface)
 - ACK : acknowledge (μP \rightarrow interface)

- A voir sur internet :
- Différences de types de mémoires.
- Amplificateur à 3 octets (buffer) (74HC245 et HC244)

- Latch à 3 états exp : 74HC573
- Decodeur :74HC138

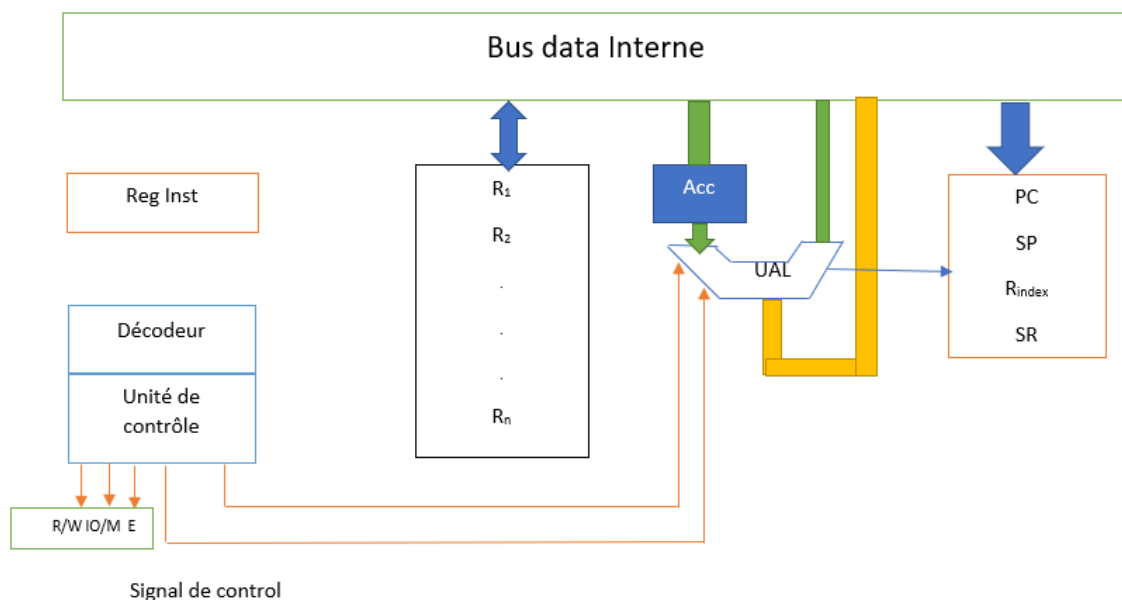
Etat des lignes	R/W	IO/M	sélection
	0	0	Ecrire dans la mémoire
	0	1	Ecrire dans l'interface
	1	0	Lire mémoire
	1	1	Lire interface



2.2 Structure interne du μP cas 8085:

- Définition :

- Un μP est un C.I de technologie minimum (LSI), c.à.d résultat d'une intégration à grande échelle de plusieurs milliers de transistors, cette intégration a permis d'insérer dans une puce (C.I) une UAL, une unité de contrôle et un ensemble de registres.
- Le μP est constitué de :
 - 1- Une UAL : pour effectuer toutes les opérations arithmétiques et logique aussi que de tests.
 - 2- Une unité de contrôle : que va chercher dans la mémoire chaque instruction à exécuter décode et généré tous les signaux nécessaires pour l'exécution correcte de l'instruction.
 - 3- Un ensemble de registres : utiliser pour stockage provisoire des données, certains registres ont une tâche bien précise (exp : SP, PC, SR, Acc, R_{index})



Structure Interne du Microprocesseur 8 bits.

PC : Program Counter (Compter Prog)

SP: Stack Pointer (Pointeur de pile)

SR : Status Register (Registre d'état)

R_{index} : register indexe.

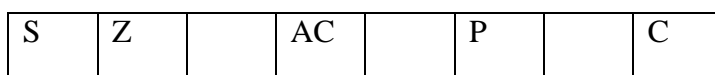
2.3 Les registres interne du μP cas 8085:

- 1- Accumulateur Ac : registre à 8bits très utilisés dans le μP , il est exploité dans toutes les opérations arithmétiques et logiques.

Exp : LDA Adresse ; Acc — M(Adresse) : LDA 8000

STA Adresse ; Acc ← M(Adresse) : STA 8005
 ADI data ; Acc ← Acc+Data : ADI 25

- 2- Le compteur programme (PC: Prog Counter) : registre 16 bits, il pointe à l'adresse de la prochaine instruction à exécuter.
- 3- Le pointeur PILE (SP : Stack Pointer) : registre à 16 bits, utilisé dans une mémoire RAM de type LIFO (Last In First Out), il pointe au top de cette pile (le haut de la mémoire LIFO) il est incrémenté et décrémenté automatiquement à chaque opération de chargement et stockage dans la pile.
- 4- Le registre Index (i.d.R : index reg) : registre à 8bits, ou 16 bits il permet d'utiliser un mode d'adressage indexé. Dans le cas du μP 8085 le registre HL (16 bits) est utilisé comme registre d'index
- 5- Le registre d'état (SR : Status Register : Flag) : registre à 8bits indique l'état du μP après exécution d'une instruction, il fournit l'information concernant la dernière opération exécutée. Dans le cas du μP 8085 les bits du Flag sont comme suit :



a) Le bit R (C) Carry : indique une retenue après l'opération :

C=0 ~~pas de~~ retenue

C=1 ~~retenue~~

Exp : MVI A, F0

ADI 25 A ~~E0+25~~

b) Le bit P (Parity) : indique si le résultat est pair/impair :

P=1 ~~résultat~~ pair

P=0 ~~résultat~~ impair

c) Le bit Ac (Axxillary Carry) : indique un demi retenu dans l'opération

d) Le bit Z (zéro) : indique si le résultat est nul :

Z=1 ~~résultat~~ nul

Z=0 ~~résultat~~ ≠0

Exp : MVI A,1C

ANI 00 A ~~A and 0~~

e) Le bit S (signe) : indique le signe de l'opération positive ou négative :

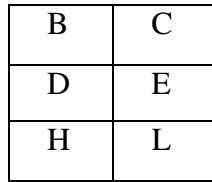
S=1 ~~résultat~~ négative

S=0 ~~résultat~~ positive

Exp : MVI A, 25

SUI 30 S=1

- 6- Registre d'instruction : toutes les instructions d'un programme sont en mémoire ROM ou RAM, le processeur va chercher l'instruction en mémoire pour le ranger dans ce registre avant exécution.
- 7- Les registres d'ordre générale : dans le cas du μP 8085 on a 6 registres à 8 bits qui peuvent être utilisés aussi comme 3 registres à 16 bits selon leurs architectures :



Exp : MOV reg1,reg2 reg1 reg2

MOV A,B A ← B

• Exercice :

Ecrire un programme qui permute le contenu de deux registres B et C :

➤ Solution :

Adresse :	Code Opérateur :	Assembleur :	Commentaire :
7000	06 55	MVI B,55	B ← 55
7002	0E FF	MVI C, FF	C ← FF
7004	78	MOV A,B	A ← B
7005	41	MOV B,C	B ← C
7006	4F	MOV C,A	C ← A
7007	CF	RST 1	FIN

2.4 Les modes d'adressage :

- Les modes d'adressage caractérisent la façon dont est obtenue l'adresse du 2eme opérande d'une instruction. Pour un μP avec différents modes d'adressage la programmation sera puissante parmi les modes d'adressages le plus réponsus dans le μP :

1- Le mode implicite : l'opération n'a pas besoin de données ou adresse pour être exécuter.

Exp : INR A, RST 1, DCR B, NOP...

2- Le mode immédiat : la donnée suit immédiatement l'opération dans une instruction.

Exp : MVI A,data

LXI H, adresse

3- Le mode direct : l'adresse de l'opérande (donné) suit directement l'opération.

Exp : LDA Adresse ; LDA 8000 ; A M(8000)

STA Adresse ; STA 8010 ; M(8010) A →

25.02.2018

Adressage direct :

- 1- Direct page zéro : Dans le mode d'espace mémoire est partagé en pages de 256 octets donc on peut avoir 256 pages. Les cases mémoires adressable en page '0' sont accédé avec le mode « Direct page zéro »

Exp : LDA 50 ; l'adresse effective : 0050.

- 2- Direct étendu : Dans ce mode l'adresse est définie par 2 octets qui suivent le code opération.

Exp : LDA 1240 ; Ac \longleftarrow M(1240)

Adressage Indirect :

L'adresse effective = le contenu du registre ou de l'adresse mémoire.

Syntaxe : code opératoire [reg ou adresse]

Exp : LDA [1240]

- Cette technique d'adressage donne lieu à différents modes :
 - Adressage indirect par mémoire.
 - Adressage indirect par registre.
 - Adressage indirect par registre avec déplacement.
- Code opération [reg], D

Adresse effective = (contenu du registre) + déplacement

- Code opération [reg, D] ;

Adresse effective = (contenu du registre + déplacement)

Exp : LDA [HL], 40

- Adressage indirect par registre post incrémentation :
 - Code opération [reg]+
 - Exp : LDA [HL]+
 - Adressage indirect pré-décément par registre :
 - Code opération [reg]-
 - Exp : LDA [HL]-

Adressage indexé :

Dans ce mode le μ P utilise des registres internes spéciaux qu'on appelle registre d'index.

Dans le cas de I-8085 le registre d'index est le registre pair 'HL'.

Ce mode permet de manipuler un bloc facilement :

- Code opération R_{index}
 Adresse effective : contenu (R_{index})
 Exp : LDA R_{index} ; Ac \longrightarrow M(R_{index})
 MOV A,M ; A \longleftarrow M(HL)
- Charger l'Acc par le contenu de la case mémoire pointé par HL.

- Exercice :

- Ecrire un programme qui permet d'effacer l'espace mémoire 8000 à 8010.

➤ Solution :

Adresse :	Code op :	Assembleur :	Commentaires :
7000	2E 00	MVI L, 00	HL ← 8000
7002	26 80	MVI H, 80	
7004	06 10	MVI B, 10	B=10
7006	36 00	MVI M, 00 etiq	M(HL) ← 00 etiq
7008	2C	INR L	L ← L+1
7009	05	DCR B	B ← B-1
700A	C2 06 70	JNZ etiq	Si B≠0 va à etiq
700D	CF	RST 1	FIN

- Travail à la maison à voir sur internet :
- L'adressage indirect-indexé avec exemple.
- L'adressage indexé-indirect avec exemple.

2.5 Cycle d'exécution d'une instruction

Le microprocesseur ne comprend qu'un certain nombre d'instructions qui sont codées en binaire. Le traitement d'une instruction peut être décomposé en trois phases.

a) Phase 1 : Recherche de l'instruction à traiter

1. Le PC (compteur programme) contient l'adresse de l'instruction suivante du programme. Cette valeur est placée sur le bus d'adresses par l'unité de commande qui émet un ordre de lecture.

2. Au bout d'un certain temps (temps d'accès à la mémoire), le contenu de la case mémoire sélectionnée est disponible sur le bus des données.

b) Phase 2 : Décodage de l'instruction et recherche de l'opérande

Le registre d'instruction contient maintenant le premier mot de l'instruction qui peut être codée sur plusieurs mots. Ce premier mot contient le code opératoire qui définit la nature de l'opération à effectuer (addition, rotation,...) et le nombre de mots de l'instruction.

1. L'unité de commande transforme l'instruction en une suite de commandes élémentaires nécessaires au traitement de l'instruction.

2. Si l'instruction nécessite une donnée en provenance de la mémoire, l'unité de commande récupère sa valeur sur le bus de données.