

Architecture des ordinateurs

**Programmation et Fonctionnement
d'un CPU**

Pr. Zaynab EL KHATTABI

Plan du cours

- ✓ Introduction
- ✓ Présentation de CPU
- ✓ Programmation d'un CPU
- ✓ Architecture du CPU - Intel 8086
- ✓ Intel 8086 – les registres
- ✓ Intel 8086 – Segmentation de la mémoire
- ✓ Langage assembleur

Présentation de CPU

- Le processeur ou CPU est le composant principal de l'ordinateur.
- Intel est le premier fabricant de microprocesseurs.
- Tous les trois ou quatre ans environ, Intel mettait sur le marché un nouveau processeur qui comportait des améliorations par rapport au précédent.
- Depuis 1978, tous ses processeurs sont basés sur l'architecture du **8086**.
- D'autres constructeurs se sont aussi basés sur le 8086. Le plus connu étant AMD, principal concurrent d'Intel.
- Les microprocesseurs basés sur l'architecture du 8086 forment la famille x86.
- Il existe d'autres familles de microprocesseurs (AMD, Cyrix, Rise Technology...) compatibles entre eux.
- **MASM** ne fonctionne que sur la famille x86.

Présentation de CPU

Définition

- La programmation d'un CPU consiste à utiliser un **ensemble d'instructions** appelé **jeu d'instructions** fixé par le constructeur qui sont reconnus par ce CPU et représentées sous forme symbolique définies dans un langage appelé **assembleur**.

Outils

- Les outils nécessaires pour programmer un CPU sont les suivants:
 - Processeur: Famille, type, mode d'organisation de la mémoire et registres
 - Jeu d'instructions.
 - Langage d'assembleur
 - IDE

Programmation d'un CPU

Définition

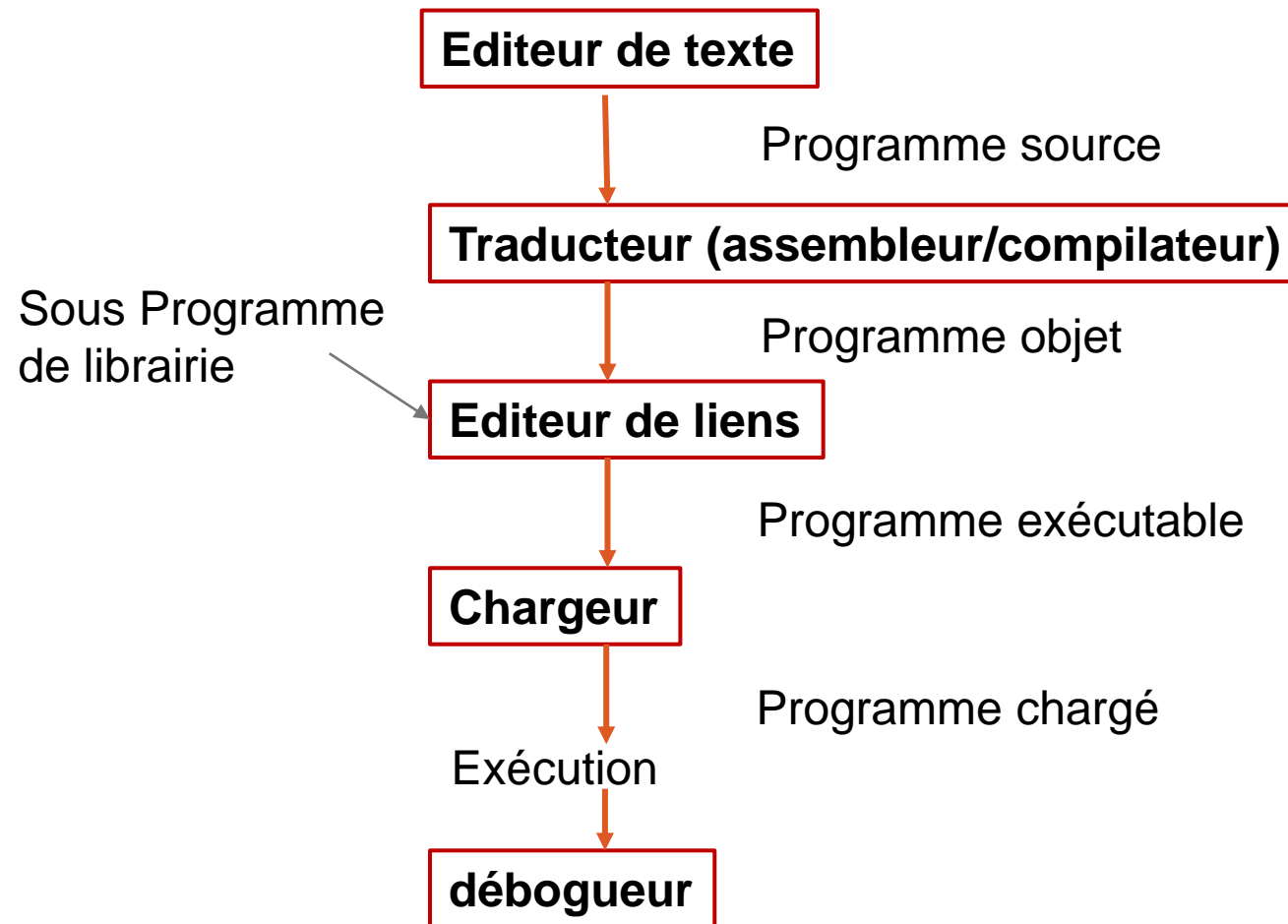
- Le jeu d'instruction est l'ensemble des instructions qui sont définis par un microprocesseur, le nombre d'instructions reconnues varie d'un processeur à l'autre.
- Le processeur est composé de deux éléments principaux :
 - **L'UAL** : L'Unité Arithmétique et Logique.
 - **Les registres.**

Exemple

- Instructions de transfert de données.
- Instructions arithmétiques.
-

Programmation d'un CPU

Etapes de préparation d'un programme



Programmation d'un CPU

- **L'éditeur de textes** : Un éditeur de texte (text editor) est un logiciel interactif qui permet de saisir du texte à partir du clavier et le stocker dans un fichier.
- **Le Traducteur** : Le traducteur est un logiciel qui, traduit un programme source écrit en langage de haut niveau en un programme objet.
- **L'éditeur de liens** : Un éditeur de liens (linker) est un logiciel qui permet de combiner plusieurs programmes objet en un seul programme exécutable.
- **Le Chargeur** : Le programme exécutable, obtenu après l'édition de liens, doit être chargé en mémoire centrale pour être exécuté. Le chargeur (loader) s'occupe de cette tâche.
- **Le débogueur** : Le débogueur (debugger) est un logiciel qui facilite la mise au point de programmes (détection et correction des erreurs, ou bugs). Il permet d'examiner le contenu de la mémoire ainsi que le contenu des différents registres. Ainsi, on peut suivre l'exécution pas à pas, c'est à dire instruction après instruction.

Programmation d'un CPU

L'Assembleur

Définitions:

Le langage assembleur : Il possède les mêmes instructions que le langage machine. A la seule différence que ces instructions sont écrites en lettres humaines. Chaque instruction en langage machine a une instruction en langage assembleur qui lui correspond et qui veut dire exactement la même chose.

L'Assembleur : C'est le nom du programme qui traduit du langage assembleur en langage machine. A ne surtout pas confondre avec le langage assembleur. Ce dernier étant le langage dans lequel on écrit, alors que l'Assembleur est le programme traducteur.

Programmation d'un CPU

L'Assembleur

Langage d'assembleur

- Le langage d'assembleur permet de représenter symboliquement les instructions d'un microprocesseur. Parmi les langages d'assembleur on trouve par exemple:
 - MASM: Microsoft Assembler.
 - TASM: Turbo Assembler.
 - NASM: Netwide Assembler.
 - FASM: Flat Assembler.
 - HLA: High level assembly.
 - ...

Programmation d'un CPU

- L' Environnements de développement Intégré (IDE) est un programme qui combine les éléments suivants:
 - **L'éditeur de textes**
 - **L'assembleur**
 - **L'éditeur de liens**
 - **Le débogueur**

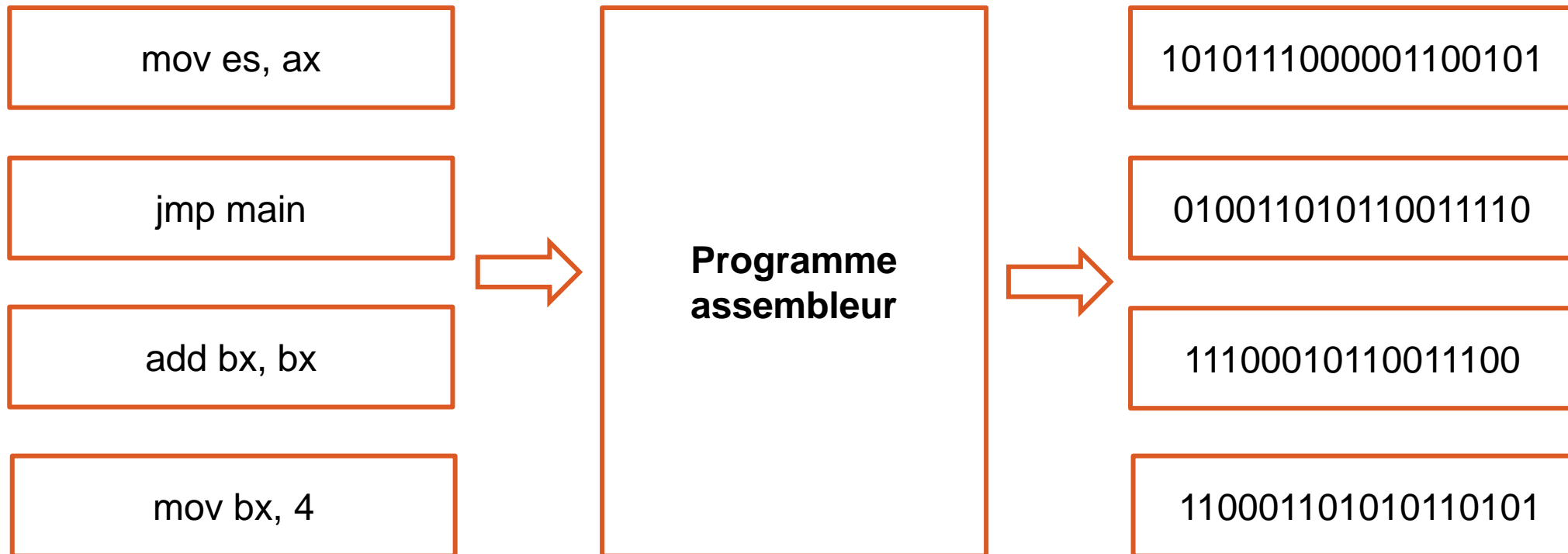
Exemples:

- RadAsm, WinAsm Studio, AsmEditor, Easy Code

Programmation d'un CPU

L'Assembleur

Schéma d'un programme écrit en langage assembleur:



Architecture d'un CPU

Intel 8086

Les registres: sont utilisés pour stocker des données de petite taille dont l'accès doit être très rapide.

Les registres sont divisés en cinq groupes :

- **Les registres de travail**
- **Les registres d'index**
- **Les registres de segments**
- **Le registre flags**
- **Le registre ip**

Architecture d'un CPU

Intel 8086 - Les registres

Contrairement aux emplacements mémoire qui sont désignés par des adresses, les registres ont chacun un nom.

Voici les principaux, 14 en tout :

- **Registres de travail : ax, bx, cx, dx.**
- **Registres d'index : di, si, sp, bp.**
- **Registres de segments : cs, ds, es, ss.**
- **Le registre flags.**
- **Le registre ip.**

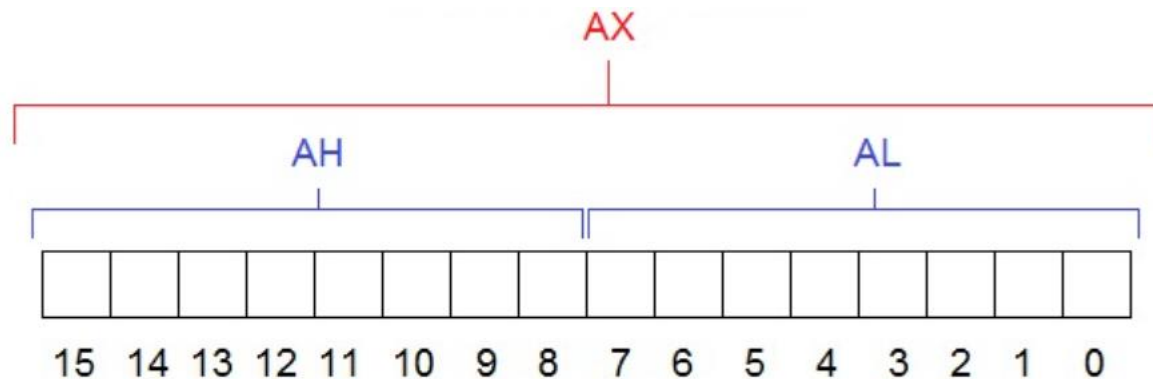
Architecture d'un CPU

Intel 8086 - Les registres

- L'ensemble des registres se nomme le jeu de registres.
- Chacun des registres est constitué de 16 bits (2 octets).
- Chaque registre peut contenir un nombre de 0 à 65 535.
- Chaque registre a en général une fonction particulière,

Exemple: Schéma du registre de travail AX

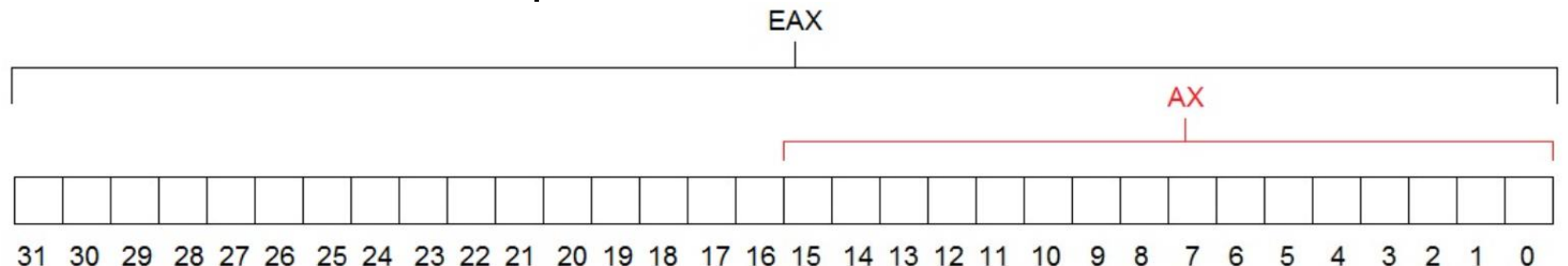
- Le registre est divisé en deux parties: l'octet "haut" (de poids fort) et l'octet "bas" (de poids faible).



Architecture d'un CPU

Intel 8086 - Les registres

- Tous les registres du 8086 existent encore dans les processeurs actuels.
- Il existe deux façon d'améliorer le jeu de registre, sans empêcher l'utilisation des anciens registres:
- **L'ajout de nouveaux registres :**
 - **Exemple:** dans le 80386, Ajout de deux registres de segments, gs et fs.
- **L'extension d'anciens registres :** C'est l'agrandissement de registres déjà existants.
 - **Exemple: Registre EAX**
 - eax fait 32 bits et possède **ax** comme mot de poids faible.



Architecture d'un CPU

Intel 8086 - Les registres

Tableau descriptif

Groupe	registres	divisé en 2 partie	étendu à 32 bits
Registres de travail	ax, bx, cx, dx	oui	oui
Registres d'index	di, si, sp, bp	non	oui
Registres de segments	cs, ds, es, ss	non	non
Registre Flags	flags	non	oui
Registre ip	ip	non	oui

Architecture d'un CPU

Intel 8086 - Les registres

Registres de travail généraux:

- Registres Accumulateur: **EAX** (32 bits), **AX** (16 bits), AH(8 bits) et AL(8 bits)
- Registres de Base: **EBX** (32 bits), **BX** (16 bits), BH(8 bits) et BL(8 bits)
- Registre Compteur: **ECX** (32 bits), **CX** (16 bits), CH(8 bits) et CL(8 bits)
- Registre Donnée: **EDX** (32 bits), **DX** (16 bits), DH(8 bits) et DL(8 bits)

Utilisation:

- Servent à **contenir temporairement des données**.
- Le registre C sert souvent de compteur, par exemple dans l'instruction loop.
- Le registre D sert d'extension au registre A pour enregistrer des nombres de double précision pour la multiplication et la division.
- Les registres de 8 et 16 bits sont utilisées par les CPU 8086 et plus.
- Les registres de 32 bits sont utilisés seulement par le CPU 80386 et plus

Architecture d'un CPU

Intel 8086 - Les registres

Registre de pointeurs et d'index:

- Pointeur de pile (Stack Pointer): **SP** (16 bits) et EPS (32 bits)
 - Registre de déplacement dans le segment pile
- Pointeur de base (Base Pointer): **BP**(16 bits) et EBP(32 bits)
- Source Index: **SI** (16 bits) et ESI (32 bits)
- Destination Index: **DI** (16 bits) et EDI (32 bits)
 - SI et DI sont des registres de déplacement dans les segments des données DS.
- Les registres de 16 bits sont utilisées par les CPU 8086 et plus.
- Les registres de 32 bits sont utilisés seulement par les CPU 80386 et plus

Architecture d'un CPU

Intel 8086 - Les registres

Registres de segment:

- **CS** : Registre de segment de code.
- **DS** : Registre de segment de données.
- **SS** : Registre de segment de pile.
- **ES** : extra registres de segment supplémentaire pour les données (FS et GS dans le 80386).
- Les registres de segments, associés aux pointeurs et aux index, permettent au microprocesseur d'adresser l'ensemble de la mémoire.

Registre IP Pointeur d'instruction :

- Compteur ordinal et contient l'adresse de la prochaine instruction à exécuter.
- Registre de déplacement dans le segment CS.
- Nombre de bits: un registre (EPI) sur 32 bits et un registres (PI) sur 16 bits.

Architecture d'un CPU

Intel 8086 - Les registres

Registre d'état:

- Le processeur possède un certain nombre d'indicateurs. Ce sont des bits d'états qui sont mis à jour lors de l'exécution de la plupart des instructions pour indiquer au programme le résultat de l'instruction.
- Il existe deux types d'indicateurs: Systèmes (utilisés par le système d'exploitation) et application (utilisés par le programmeur).
- Deux types d'indicateurs d'application: **d'états et de contrôle**

Architecture d'un CPU

Intel 8086 - Les registres

Registre d'état:

Indicateurs de contrôle:

- **TF:** Indicateur d'exécution pas à pas (Trap Flag), TF=1 si le microprocesseur exécute les instructions pas à pas, TF=0 sinon.
- **IF:** Indicateur d'autorisation d'interruption (Interrupt Flag), IF=1 si une interruption externe est autorisé, IF=0 sinon.
- **DF:** Indicateur de direction (Direction Flag), indique le sens croissant ou décroissant pour les adresses mémoires (ESI et EDI) pendant les opération sur les chaînes de caractères. DF=0 pour l'incrémentement et DF=1 pour décrémentement.

Architecture d'un CPU

Intel 8086 - Les registres

Registre d'état:

Indicateurs d'état:

CF = Carry Flag ou retenue

PF = Parity Flag ou parité

AF = Auxiliary Carry Flag ou retenue auxiliaire

ZF = Zero Flag ou indicateur de zéro

SF = Sign Flag, ou indicateur de signe

DF = Direction Flag ou indicateur de direction

OF = Overflow Flag ou indicateur de débordement de capacité

Architecture d'un CPU

Intel 8086 - Les registres

Registre d'état:

Indicateurs d'état:

- **CF:** Indicateur de retenue, **CF = 1** s'il y a une retenue après une opération arithmétique, **CF= 0** sinon.
- **Exemple:** A=0xFF, B=0x01 , A+B=0x00 → CF=1

0 1 0 0
+ 0 1 1 0
CF=0 1 0 1 0

1 1 0 0
+ 0 1 1 0
CF=1 0 0 1 0
1

1 1 1 1
+ 0 0 0 1
CF=1 0 0 0 0
1

Architecture d'un CPU

Intel 8086 - Les registres

Registre d'état:

Indicateurs d'état:

- **PF:** Indicateur de parité, **PF=1** si le nombre de 1 est pair dans les 8 bits du poids faible du résultat d'une opération (arithmétique ou logique), **PF=0** si le nombre des 1 est impair.
- **ZF:** Indicateur de zéro, **ZF=1** si tous les bits d'un résultat d'une opération (arithmétique ou logique) sont à 0, **ZF=0** sinon.

Architecture d'un CPU

Intel 8086 - Les registres

Registre d'état:

Indicateurs d'état:

- **SF**: Indicateur de signe, **SF=1** si le bit du poids fort du résultat d'une opération (arithmétique ou logique) est 1, **SF=0** si non.

	0 1 0 0
	+ 0 1 1 0
SF=1	1 0 1 0

	1 1 0 0
	+ 0 1 1 0
SF=0	0 0 1 0

	1 1 1 1
	+ 0 0 0 1
SF=0	0 0 0 0

OF: Indicateur de débordement, OF=1 s'il y'a un débordement de capacité après une opération arithmétique, OF=0 si non.

- Un **résultat négatif d'opérandes positifs** (ou vice versa) est un **débordement**.
 - Exemple: A=0x30, B=0x6F , A+B=0x9F (0x9F est négatif)
 - OF=1, pour indiquer qu'un débordement s'est produit lors de l'opération.

Architecture d'un CPU

Intel 8086 - Les registres

Exemple : Somme de deux nombres a et b

a=25H

b=5BH

- Quel est le résultat de l'opération et les indicateurs du registre d'état: CF, ZF, SF et OF.

Solution : $a+b = 80H = 10000000$

- CF=0, pas de retenu
- ZF=0, pas null
- SF=1, négatif
- OF=1, résultat négatif (10000000), d'opérandes a=00100101 et b=01011011 positifs => débordement

Architecture d'un CPU

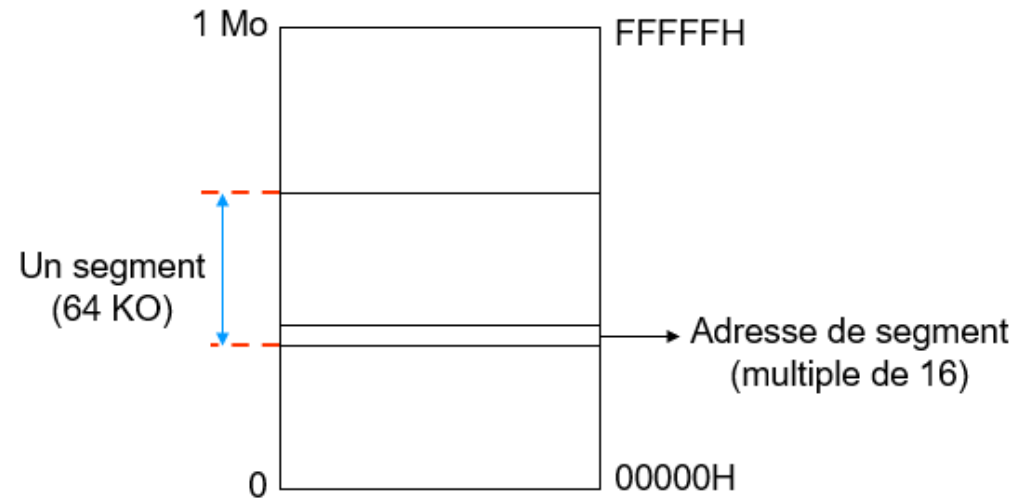
Intel 8086 – Segmentation de la mémoire

- L'espace mémoire adressable ($1\text{ Mo}=2^{20}$ Octets, 20 broches du bus d'adresse) du 8086 est divisé en quatre segments logiques allant jusqu'à **64 Ko chacun**.
- L'accès à ces espaces est direct et simultané, or le compteur de programme est de 16 bits, donc la possibilité d'adressage de 2^{16} Octets=64 Ko, (ce qui ne couvre pas la totalité de la mémoire), alors on utilise 2 registres pour indiquer une adresse au processeur.
- Chaque segment débute à l'endroit spécifié par le registre segment.
- Le déplacement (**offset**) à l'intérieur de chaque segment se fait par un registre de décalage qui permet de trouver une information à l'intérieur du segment.

Architecture d'un CPU

Intel 8086 – Segmentation de la mémoire

- Chaque segment est adressé par deux registres:
 - **Registre Segment ou registre Base:** pointe sur le début du segment (Base).
 - **Registre de décalage:** pointe sur l'information à l'intérieur du segment (Offset)



- **L'adresse physique = Base * 16 + Offset.** (en décimal)
- **L'adresse physique = Base * 10 + Offset.** (en Hexadécimal)

Intel 8086 – Segmentation de la mémoire

- L'adresse segment (base) et l'adresse de déplacement (offset) sont des adresses sur 16 bits (4 chiffres hexadécimaux), adressées par des registres sur 16 bits.
- Tandis que l'adresse physique est une adresse sur 20 bits (5 chiffres hexadécimaux), adressées par le bus d'adresses de 20 broches.
- **Exemple:** Une adresse 20 bits formée à partir du segment et du déplacement sur 16 Bits.

Bits	19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																		
	Base																0 0 0 0		
+	0 0 0 0				Offset														
=	Adresse Physique																		

Architecture d'un CPU

Intel 8086 – Segmentation de la mémoire

- L'adresse logique est le couple (segment, offset) qui est notée par:
 - **segment : offset.**

Exemple:

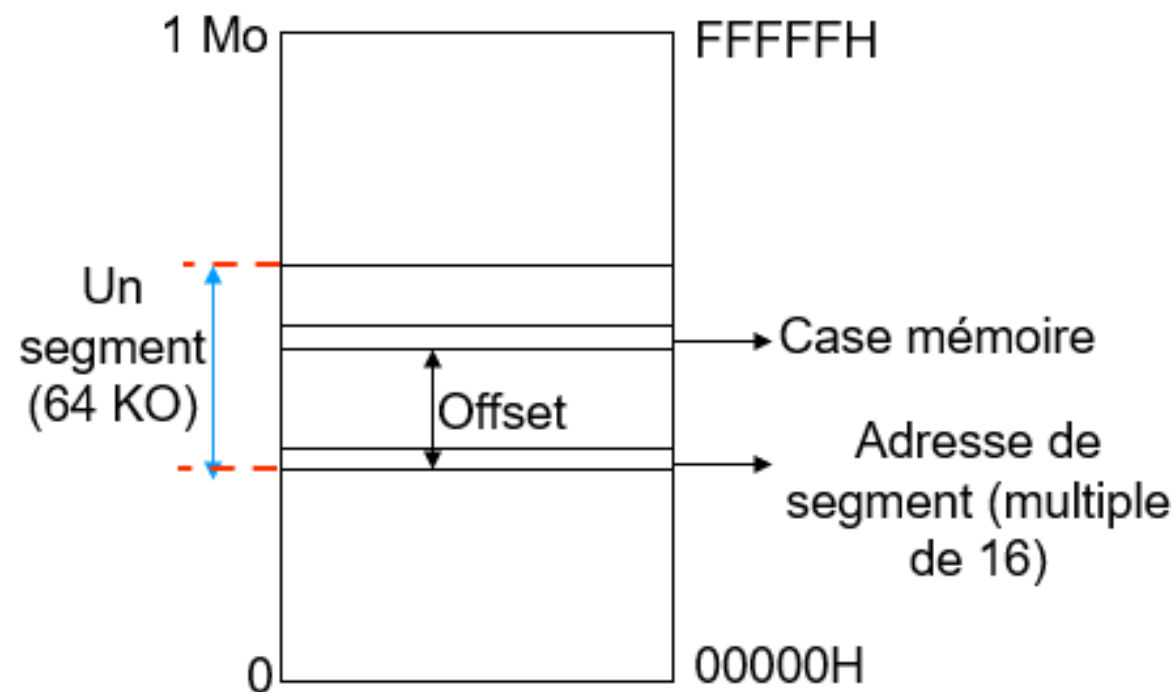
CS=1000H

IP=2006H



@Phy=10000H+02006H=12006H

@log=(CS:2006H)



Architecture d'un CPU

Intel 8086 – Segmentation de la mémoire

Exemple de configuration de 3 segments de la mémoire:

Case mémoire

