

Questions de cours (07pts) : Comblez le vide

Les deux unités qui constituent le processeur 8086 sont :

Unité1 : **Unité d'interface de bus (UIB)**. Son rôle est: **Récupère, stocke les informations à traiter, et établit les transmissions avec les bus du système Recherche les instructions en mémoire et les range dans une file d'attente.**

Unité2 : **Unité d'Exécution (UE)** son rôle est : **Exécute les instructions contenues dans la file d'attente qui lui sont transmises par l'UIB.**

Le prétraitement consiste à **placer les instructions qui suivent dans une file** Lorsque l'UE a fini de traiter une instruction l'UIB lui transmet instantanément l'instruction suivante, et charge la troisième instruction en vue de la transmettre à l'UE.

Les registres sont considérés comme **une mémoire interne** du microprocesseur.

Pour savoir quel type d'opération doit être exécuté, le microprocesseur lit le **premier octet de l'instruction** pointée par le **pointeur d'instruction** et le range dans le **registre d'instruction** Le code opératoire est décodé par **des circuits de décodage** contenus dans l'UIB.

Des signaux de **commande** pour l'UAL sont produits en fonction de l'opération demandée qui est alors exécutée.

Le microprocesseur échange des informations avec **la mémoire et l'unité d'E/S** au moyen d'un ensemble de **connexions** appelé bus. Un bus permet de **transférer de transférer des données** sous forme **parallèle**.

Le Bus d'adresses permet au microprocesseur de spécifier **l'adresse de la case mémoire à lire ou à écrire**. Il est **unidirectionnel**.

Le Bus de données permet les transferts entre le microprocesseur et **la mémoire ou les E/S**. Il est **bidirectionnel**

Le Bus de commande transmet les **ordres de lecture et d'écriture** de la mémoire et les signaux de synchronisation en provenance de l'unité de **commande à destination des E/S**. Il est **directionnel**

Une instruction est **une suite d'octets** contenus dans un fichier **exécutable**. Une instruction comporte **des octets** répartis en champs

Les octets de déplacement sont ajoutés à l'instruction si celle-ci comprend déjà un champ **MOD R/M**. Le champ COP a une taille de **1 octet**, il est codé de plusieurs manières, la plus utilisée est la forme **xxxxxdw**

Exercice 1 (05 pts)

Pour chacune des instructions assembleur suivantes, donnez dans la colonne en face le mode d'adressage utilisé :

Instruction	<u>Réponse</u> : Mode d'adressage
MOV [0ABC], BX	Direct
JMP @2 ; JMP (effectue un saut)	Symbolique ou relatif
MOV [BX+SI+2], 0FFEh	Combiné basé-indexé
MOV AX, [BX]	Indirect registre
MOV [BX- 26], 0500H	basé

Exercice 2 (02 pts)

Quel doit être la taille du bus d'adresse d'un processeur 16 bits pour qu'il puisse accéder à une mémoire de 8 Ko ?

Réponse :

$$8\text{Ko} = 2^3 \times 2^{10} \times 8\text{bits} = 2^{13} \times 8\text{bits} = 2^{12} \times 16\text{bits}$$

12 bits pour le bus d'adresse

Exercice 3 (06 pts)

A. Soit une mémoire de 1 MO, cette mémoire est divisée en blocs de 64 KO.

1. Combien de blocs obtient-on ?
2. Calculez en hexadécimal les adresses de début et de fin de chaque bloc.

B. On suppose maintenant que la mémoire a une capacité de 64 MO et elle est divisée en blocs selon la règle suivante :

$$\text{Taille du Bloc}_0 = \text{Taille du Bloc}_1 = 1\text{MO.}$$

$$\text{Taille du Bloc}_i = 2 * \text{Taille du Bloc}_{i-1}.$$

Répondez à nouveau aux questions 1 et 2.

Réponse :

A.

1- Calculer le nombre de blocs :

Le nombre de blocs = taille globale de la mémoire / taille de chaque bloc.

$$\text{Taille de la mémoire} = 2^{20} \text{ O}$$

$$\text{Taille de chaque bloc} = 2^6 * 2^{10} \text{ O} = 2^{16} \text{ O}$$

$$\text{Le nombre de blocs} = 2^{20} / 2^{16} = 2^4 = 16 \text{ blocs}$$

2- Donner les adresses de début et de fin de chaque bloc.

- Taille de la mémoire = 2^{20} → Les adresses ont une taille de 20 bits = 5 positions en hexadécimal.
- Taille de chaque bloc = 2^{16} → Nous avons besoin de 16 bits pour adresser tous les mots à l'intérieur de chaque bloc = 4 positions en hexadécimal.

→ L'adresse du premier mot de chaque bloc est 0000 0000 0000 0000 = 0H
L'adresse du dernier mot de chaque bloc est 1111 1111 1111 1111 = FFFFH

Le tableau suivant donne les adresses de début et de fin de chaque bloc.

Semestre 2

N du bloc	@ de début	@ de fin
0	00000H	0FFFFH
1	10000H	1FFFFH
2	20000H	2FFFFH
3	30000H	3FFFFH
4	40000H	4FFFFH
5	50000H	5FFFFH
6	60000H	6FFFFH
7	70000H	7FFFFH
8	80000H	8FFFFH
9	90000H	9FFFFH
10	A0000H	AFFFFH
11	B0000H	BFFFFH
12	C0000H	CFFFFH
13	D0000H	DFFFFH
14	E0000H	EFFFFH
15	F0000H	FFFFFH

E

1. Calculer le nombre de blocs

Les tailles des blocs = 64 MO. Et selon la règle donnée, nous calculons la taille de chacun comme suit :

- Taille bloc 0 = 1MO reste 63 MO
- Taille bloc 1 = 1MO reste 62 MO
- Taille bloc 2 = 2MO reste 60 MO
- Taille bloc 3 = 4MO reste 56MO
- Taille bloc 4 = 8MO reste 48 MO
- Taille bloc 5 = 16MO reste 32 MO
- Taille bloc 6 = 32MO reste 0 MO

→ Le nombre de blocs est de 7 blocs.

2. Les adresses de début et de fin de chaque bloc sont données dans le tableau suivant.

RC: Taille de la mémoire = 64 MO = 2^{26} → Les adresses ont une taille de 26 bits = 7 positions en hexadécimal

Numéro du bloc	Taille du bloc	@ Début	@Fin
0	2^{20} MO	0000000H	000FFFFH
1	2^{20} MO	0010000H	001FFFFH
2	2^{21} MO	0020000H	003FFFFH
3	2^{22} MO	0040000H	007FFFFH
4	2^{23} MO	0080000H	00FFFFFFH
5	2^{24} MO	0100000H	01FFFFFFH
6	2^{25} MO	02FFFFFFH	03FFFFFFH

