Série de TD N° 2

Exercice 01 :

1. Quel type d'architecture le 8086 a-t-il ? (**CISC** ou **RISC**).
2. En quelle année le 8086 a été fabriqué ?
3. Quel est le nombre de broches du 8086 ?
4. Quel est le nombre total des registres du 8086 ?
5. Citez les registres de segments, et dans quelle partie du microprocesseur se trouvent-ils ?
6. Citez les registres d'index est dans quelle partie du microprocesseur se trouvent-ils ?
7. Quelle est la taille de la file d'attente du microprocesseur 8086. Dans quelle partie du microprocesseur se trouve-t-elle ?
8. Quelle est la taille en octet de la mémoire adressable par le 8086.
9. Une instruction est composée de deux champs, citez-les ?
10. Quelle unité du microprocesseur est responsable de charger l'adresse physique (20 bits) sur le bus d'adresses.

Exercice 02 :

Répondez par **(vrai)** ou **(faux)** aux affirmations suivantes :

1. Un segment est une zone mémoire de 64 Ko.
2. Les bits qui composent le registre d'état sont indépendants les uns des autres.
3. Les registres de segment CS, DS, ES et SS sont chargés de sélectionner les différents segments de la mémoire en pointant sur le début de chacun d'entre eux.
4. L'UE n'a aucune connexion avec les bus (adresses, données et commande).
5. La BIU est chargée de remplir la file d'attente des instructions cherchées.
6. Le registre IP contient l'adresse de l'emplacement mémoire où se situe la prochaine instruction à exécuter.
7. L'indicateur de parité PF est à 1 si le résultat de l'opération est pair.
8. La BIU cherche les instructions à exécuter dans la mémoire et les stocke dans la file d'attente du 8086.
9. La BIU calcule les adresses physiques sur 16 bits.
10. Le langage machine est le seul langage que le processeur puisse traiter.
11. Le segment de pile est organisé en octets (sur 8 bits).
12. L'instruction PUSH permet d'incrémenter de 2 l'adresse contenue dans le registre SP.
13. Le registre (IR : Instruction Register) peut être manipulé par le programmeur.

Exercice 03 :

1. Si le registre du code segment CS contient la valeur 2300H. Déterminez l'adresse physique du début et de la fin du segment de code.
2. Si le registre du data segment DS contient la valeur 0100H. Déterminez l'adresse physique du début et de la fin du segment de données.
3. Après l'exécution des instructions ci-dessous, donnez l'état des drapeaux (CF, ZF, SF et PF) du registre d'état (registre indicateur)

b)
```
mov AX, 70D
mov BX, 200D
add AX, BX
```

a)
```
mov AX, 22H
mov BX, 52H
sub AX, BX
```

c)
```
mov AX, 22H
mov BX, AX
sub AX, BX
```

4. Soit la partie du programme assembleur ci-dessous

```
Mov   cx, 55h        ; le registre SP contient la valeur 2244 H
Push  ax
Push  bx
Push  cx
Pop   ax
```

Déterminez le contenu des registres SP et AX après exécution du programme.

Exercice 04 :

Choisissez la bonne réponse

1. Le rôle principal de la 'cache mémoire' est
- **a)** d'augmenter la capacité de stockage des données de la mémoire centrale.
- **b)** de compenser la faible vitesse et accélérer l'accès à la mémoire centrale.
- **c)** d'améliorer l'horloge du microprocesseur.

2. Le temps d'exécution d'une instruction dépend de
- **a)** la complexité de l'instruction et du mode d'adressage.
- **b)** l'architecture du microprocesseur.
- **c)** la taille de la mémoire centrale.

3. L'opcode de l'instruction permet de
- **a)** déterminer sa nature.
- **b)** localiser l'opérande en mémoire.
- **c)** indiquer le type des opérandes.

4. Dans le segment de pile
- **a)** le premier élément introduit est toujours le premier sorti.
- **b)** le premier élément introduit est toujours le dernier sorti.

Exercice 05 :

Après avoir exécuté le programme ci-dessous. Déterminez le contenu des registres AX, BX et CX. Déterminez le contenu de registre SP aux positions mentionnées. Sachant qu'initialement SP contient FFFEH

```
MOV  AX, 1122H
MOV  BX, 3344H
MOV  CX, 5566H

PUSH AX
PUSH BX  ;  déterminez le contenu du registre SP après exécution de l'instruction en cours SP = … ?
PUSH CX

POP  BX
POP  AX  ;  déterminez le contenu du registre SP après exécution de l'instruction en cours SP = … ?
POP  CX
```