

Suggestion de Solution du Contrôle

Questions de cours (6pts)

1-

SAR : Décalage Arithmétique vers la droite ; Cette instruction décale l'opérande de n positions vers la droite $(0,25)$

SAL : Décalage Arithmétique vers la gauche ; Cette instruction décale l'opérande de n positions vers la gauche $(0,25)$

CMPS :

- L'opérande est un octet : **CMPSB** (Compare String Byte) $(0,25)$

CMPSB compare deux chaînes formées d'une suite d'octets.

- L'opérande est un mot de 16 bits : **CMPSW** (Compare String Word) $(0,25)$

CMPSW compare deux chaînes formées d'une suite de mots

CLI (Clear Interrupt Flag) mise à zéro du masque d'interruption **IF** $(0,25)$

STI (Set Interrupt Flag) mise à 1 du masque d'interruption **IF**. $(0,25)$

2-

Le dépilement engendre deux opérations :

- $destination \leftarrow [SP]$: le sommet de la pile est copié dans destination. $(0,25)$
- $SP \leftarrow SP + 2$: la donnée du sommet étant retirée, on fait avancer **SP** pour pointer sur la donnée suivante $(0,25)$

L'empilement s'effectue en deux étapes :

- $SP \leftarrow SP - 2$: on fait reculer **SP** de deux cases (donnée sur 16 bits), pour qu'il pointe sur le nouveau sommet susceptible de recevoir la donnée à empiler. $(0,25)$
- $[SP] \leftarrow donnée$: le sommet de la pile reçoit la donnée $(0,25)$

3- pour mettre à 0 ses 7 bits de poids faible on utilise l'instruction **AND** avec le masque $(0,5)$
10000000

4- Il y a deux types de commandes que l'on peut adresser au PIC : les **OCW** (Operational Command Words) $(0,5)$ et **ICW** (Initialization Command Words). $(0,5)$

ICW1 7 6 5 4 3 2 1 0 (1)



→ ICW4 est-il requis ? (0 = non, 1 = oui)

1 = un seul PIC existant

0 = PIC en cascade

0 = mode attaque

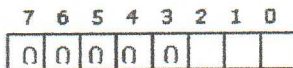
1 = mode palier

ICW2 7 6 5 4 3 2 1 0



→ Numéro de l'interruption à exécuter par IRQ0 $(0,5)$

ICW4



Numéro de la ligne d'interruption par laquelle l'esclave est relié au maître

Exercice 01(4ts)

MOV BX, 5 ; charger 5 dans un registre ; 0,1r
 MOV AX, BX ; copier le nombre dans un 2ème registre
 MUL AX ; $x2 = x \cdot x$; 0,1r
 MOV DL, 2 ; charger 2 dans un registre
 MUL DL ; multiplier x2 par 2 ; 0,1r
 MOV CX, AX ; sauvegarder 2x2 dans un registre ; 0,1r
 MOV AX, BX ; charger x dans AX
 MOV DL, 3 ; charger 3
 MUL DL ; multiplier x par 3 ; 0,1r
 ADD AX, CX ; additionner 2x2 + 3x ; 0,1r
 SUB AX, 1 ; soustraire 1 de 2x2 + 3x ; 0,1r
 MOV [0300H], AX ; sauvegarder le résultat en mémoire ; 0,1r
 HLT ; fin du programme

Exercice 02 (5pts)

MOV CX, 100 ; 0,1r
 MOV BX, 0 ; 0,1r
 Boucle : MOV AX, CX ; 0,1r
 MOV DL, 3 ; 0,1r
 DIV DL ; 0,1r
 CMP AH, 0 ; 0,1r
 JNE Suivant ; 0,1r
 ADD BX, CX ; 0,1r
 Suivant : LOOP Boucle ; 0,1r
 HLT ; 0,1r

- 1- Ce programme calcule la somme des multiples de 3 inférieurs à 100 ; 1,1r
- 2- Bx va contenir cette somme ; 0,1r

Exercice 03(5pts)

MOV AL, 8Bh ; mot de commande ; 0,1r
 MOV DX, 106h ; adresse du registre de contrôle ; 0,1r
 OUT DX, AL ; écriture du mot de commande dans RC ; 0,1r
 MOV DX, 102h ; adresse du port B ; 0,1r
 IN AL, DX ; lecture du contenu du port B dans AL ; 0,1r
 MOV DX, 100h ; adresse du port A ; 0,1r
 OUT DX, AL ; transfert du contenu de AL au port A ; 0,1r