

Première épreuve de moyenne durée

Exercice n° 1 (10 points)

Il est demandé d'écrire un programme en Assembleur 8086 qui:

- lit deux nombres décimal au clavier et les sauvegarde dans N1 et N2. La lecture du premier nombre s'arrête quand le caractère lu est un caractère « + » ou « - ». Le second nombre est ensuite lu, la lecture s'arrête quand le caractère lu est le caractère « = ».
- effectue l'opération spécifiée par le caractère séparant les deux nombres saisis
- affiche le résultat en décimal ou en hexadécimal.

Exercice n° 2 (6 points)

Soit la zone mémoire MEM de type octet dont les bits sont positionnés à 1 ou à 0.

Ecrire un programme en Assembleur 8086 qui permet d'afficher les numéros des bits positionnés à 1 (les numéros varient de 0 à 7).

Exercice n° 3 (4 points)

Répondre aux questions suivantes :

- que signifie le bit s : extension de signe
- quels sont les éléments d'une instruction Assembleur à coder pour générer le code machine
- quelle est l'utilité du signal $\overline{\text{BHE}}$
- quels sont les éléments qui influent sur le temps d'exécution d'une instruction Assembleur

Première épreuve de moyenne durée

Exercice1

Soit le segment de données d'un programme donné :

DATA SEGMENT

VAR1 DW 1234H

VAR2 DB 4 DUP(16H, 2 DUP(0AH, 1CH), ABH)

VAR3 DW 5 DUP(?)

VAR4 DW ?

DATA ENDS

Et le segment code du même programme :

CODE SEGMENT

ETIQ1 : MOV SI, 2

ETIQ2 :

MOV AX, VAR1 *

MOV VAR3, AX **

MOV AX, VAR2[SI] ***

INC SI

CMP SI, 3

JNE ETIQ2 ****

JMP ETIQ1 +

CODE ENDS

1. Pour quelle valeur chacun des identificateurs VAR1, VAR3, VAR2[SI], ETIQ1 sera-t-il remplacé dans le code exécutable (résultat de la compilation et édition de lien du programme) ?
2. Quelle est la valeur de AX après exécution de l'instruction *** ?
3. Sachant que CS = 1534H, calculez l'adresse physique de branchement inconditionnel.

Exercice2

Soit une chaîne de caractères, codée en ASCII, nommée CHAINE. Cette chaîne peut être

cas1 : numérique

cas2 : Alphabétique a - tous les caractères sont du type minuscule

b - tous les caractères sont du type majuscule

Ecrire un programme qui :

1- Dans le cas1 : calcule la somme des chiffres correspondants aux caractères de cette chaîne.

2- Dans le cas2 : réalise le tri par ordre alphabétique de la chaîne.

Les codes ASCII sont les suivants

'0' '9' 30H 39H

'A' 'Z' 41H 5AH

'a' 'z' 61H 7AH

Exo 3

Ecrire un programme qui, sans modifier les valeurs des registres AX et BX, transfère :

- 1- Les bits 0-4 du registre AX dans les bits 1-5 du registre CX.
- 2- Les bits 5-8 du registre BX dans les bits 6-9 du registre CX.
- 3- Les bits 9-15 du registre AX dans les bits 10-15 du registre CX.

Première épreuve de moyenne durée

Exercice1

Soit le segment de données d'un programme donné :

```
DATA SEGMENT
VAR1 DW 1234H
VAR2 DB 4 DUP(16H,2 DUP(0AH,1CH),3BH)
VAR3 DW 5 DUP(?)
VAR4 DW ?
DATA ENDS
```

Et le segment code du même programme :

```
CODE SEGMENT
ETIQ1 : MOV SI, 2
ETIQ2 :
        MOV AX, VAR1 *
        MOV VAR3, AX **
        MOV AX, VAR2[SI] ***
        INC SI
        CMP SI, 3
        JNE ETIQ2 ****
        JMP ETIQ1
```

CODE ENDS

- Quelle valeur chacun des identificateurs VAR1, VAR3, VAR2[SI], ETIQ1 sera-t-il remplacé dans l'exécutable (résultat de la compilation et édition de lien du programme) ?
2-Quelle est la valeur de AX après exécution de l'instruction *** ?
3-Sachant que CS = 1534H, calculer l'adresse physique de branchement inconditionnel.

Exercice2

Soit une chaîne de caractères, codée en ASCII, nommée CHAINE. Cette chaîne peut être

Cas1 : numérique

Cas2 : Alphabétique a - tous les caractères sont du type minuscule
b - tous les caractères sont du type majuscule

Ecrire un programme qui :

dans le Cas1 : calcule la somme des chiffres correspondants aux caractères de cette chaîne.
Dans le cas2 : réalise le tri par ordre alphabétique de la chaîne.

Les codes ASCII sont les suivants

'a'.....'z'	61H.....7AH	1000000	1200
'A'.....'Z'	41H.....5AH	0000000	1200
'0'.....'9'	30H.....39H	0000000	1200

Exercice3

Ecrire un programme qui, sans modifier les valeurs des registres AX et BX, transfère :

- 1- Les bits 0-4 du registre AX dans les bits 1-5 du registre CX.
- 2- Les bits 5-8 du registre BX dans les bits 6-9 du registre CX.
- 3- Les bits 9-15 du registre AX dans les bits 10-15 du registre CX.

EMD N°1Exercice 1 : (14 points)

Soit le programme assembleur suivant :

```

Data      SEGMENT
TAB DB    N      DUP (?)
Data      ENDS

Code      SEGMENT
Assume CS: Code, DS: Data
Start:    MOV AX, Data
          MOV DS, AX
          MOV CX, LENGTH TAB
          MOV SI, OFFSET TAB
encore:   MOV AL, [SI]
          MOV [SI+100], AL
          INC SI
          LOOP encore
          MOV AX, 4C00H
          INT 21H
Code      ENDS
          END start

```

- 1) Quel est le rôle du label 'encore' ? (2 points)
- a) Donner les adresses effectives des différentes instructions de ce programme. (2 points)
- c) Lorsque $N = 10$, que fait ce programme ? (2 points)
- d) Quand est-ce que ce programme ne réalisera plus la tâche décrite dans votre réponse à la question (c) ? (2 points)
- e) Donner l'autre écriture de ce programme qui permet de solutionner le problème discuté dans la question (d). (2 points)
- f) En supposant que le data segment est chargé en mémoire à l'adresse segment ABCDH, donner le code machine des trois premières instructions du code segment. (2 points)
- g) Pour une fréquence du microprocesseur de 400 Mhz, donner le temps d'exécution minimal de ce programme. (2 points)

Exercice 2 : (6 points)

Soit la déclaration suivante d'un segment de données :

```

Usthb      SEGMENT
          Un   DB    2      DUP(4, 2 DUP(0), 20)
          ORG 0200H
          Deux DB    01010101B, 'EMD1'
          Trois DW    3 DUP(15, 21)
Usthb      ENDS

```

- a) En supposant que ce segment est chargé en mémoire à l'adresse segment 1234H, donner la représentation de ce segment en mémoire en citant toutes les cases mémoire qu'il occupe (adresses mémoire et contenu en Hexadécimal). (4 points)
- b) Ce data segment tel qu'il est déclaré risque d'influer négativement sur le temps d'exécution du programme auquel il appartient. Quelle modification peut-on apporter à ce data segment pour éventuellement réduire le temps d'exécution du programme. (2 points)

E.M.D N° 1

Exercice 1 : (7 points)

Soient :

(IP) = 2BC0H , (CS) = 0200H , DEPLACEMENT = 5119H , BX = 1200H
DS = 212AH , (224A0H) = 0600H , (275B9H) = 038AH

1. Calculer les adresses effectives de branchement pour les instructions de branchement inconditionnel suivantes :
 - a) Instruction intrasegment direct
 - b) Instruction intrasegment indirect utilisant le mode d'adressage registre et le registre cité est BX
 - c) Instruction intrasegment indirect utilisant le mode d'adressage registre relatif et le registre cité est BX
2. Donner le code machine pour chacun des cas précédents
3. Evaluer le temps d'exécution pour chacun des cas précédents en supposant que la fréquence du microprocesseur est 400 MHz.

Exercice 2 : (6 points)

Donner pour chacune des instructions suivantes le test effectué par le microprocesseur :

JBE OPR , JL OPR , JLE OPR

Les trois instructions interviennent dans le programme après des instructions CMP.

Exercice 3 : (7 points)

Ecrire, en Assembleur, la séquence d'instructions qui permet d'évaluer l'expression booléenne suivante :

$$f(X_7, \dots, X_0) = X_7 \overline{X_6} X_3 X_1 + X_6 X_4 \overline{X_2} X_1 X_0 + \overline{X_7} X_5 X_3 \overline{X_1}$$

Pour se faire, nous supposerons que les bits 7, 6, ..., 1, 0 de AL représentent les valeurs respectives des variables logiques $x_7, \dots, x_0$ et AL contiendra à la fin du traitement la valeur de f.

Remarque : Pour des raisons d'espace mémoire, votre séquence d'instruction ne devra en aucun cas dépasser 18 instructions.

FORMULE DE GENERATION DU CODE MACHINE POUR JMP INTRASEGMENT :

Intrasegment direct short :	7 0 11101011	7 0 DEP.	
Intrasegment direct Near :	7 0 11101001	7 0 DEP. BAS	7 0 DEP. HAUT
Intrasegment indirect	7 0 11111111	7 0 MOD 100 R/M	7 0 DEP. BAS 7 0 DEP. HAUT

Première épreuve de moyenne durée
(DOCUMENTS NON AUTORISÉS)

Exercice n°1 :

Soit le segment de données suivant :

Data segment

TAB1 DB 4 DUP(00H, 32H, 2 DUP(0))

TAB2 DW 2 DUP(0050H, 2 DUP(0,1))

TAB3 DW (0110H, 0140H)

Data ends

1. Écrire les instructions correspondant aux opérations suivantes :

- transfert du 4^{ème} élément de TAB1 dans un registre,
- transfert du 6^{ème} élément de TAB2 dans un registre,
- transfert du 1^{er} élément de TAB3 dans un registre,
- transfert du 6^{ème} élément de TAB2 dans le 2^{ème} élément de TAB3.

2. Préciser pour chaque opération, la valeur affectée à l'opérande destination.

3. Pour chaque accès mémoire mis en jeu lors de l'exécution des instructions précédentes, donner les valeurs de BHE/ et A₀. La valeur de DS est égale à 0010H.

Remarque : Le i^{ème} élément d'une zone donnée dépend du type de base de cette zone.

Exercice n°2 :

Supposons que les variables x, y, w, c et z sont de type DW et considérons la portion de programme assembleur suivante :

MOV AX, Y

MUL W $w \times y \rightarrow 0010H$

MOV BX, AX $60H \rightarrow Y$

MOV CX, DX

MOV AX, X

CWD $00000000H \rightarrow 0010H$

ADD BX, AX

ADC CX, DX

MOV AX, C

CWD

SUB BX, AX

SBB CX, DX

MOV AX, BX

MOV DX, CX

MOV 0020H

MOV Z, AX

1. Que fait ce programme ?

2. Réécrire ce programme (déclarations et instructions) en le simplifiant au maximum.

Exercice n°3 :

Soit un programme dont le code objet occupe 67 Ko et ses données 24 Ko. Le programme génère un espace de pile de 4 Ko et n'utilise aucune données de l'espace « extra-données ».

En supposant que le programme est entièrement résident en mémoire.