

ANEXA 4

Model Probleme Examen Microprocesoare I

1. Se dă un șir de 15 elemente, definite pe octet, în segmentul de date, format din litere mari majuscule) și cifre zecimale. Să se genereze două șiruri: unul format din literele mari și unul din cifrele șirului initial.

Se cere:

a. Definiți, în formă prescurtată, segmentul de date care sa conțină: **1p**

-șirul initial (SIR_INIT) format din elementele:

2,'F','A',3,9,4,'T','E','P',8,5,7,'W',1,'J'

-cele doua șiruri destinație denumite CIFRE respectiv LITERE

b. Să se scrie secventa de generare a celor două șiruri. **2p**

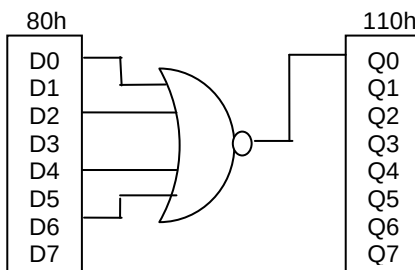
c. Să se afiseze pe randuri diferite, șrurile obținute, folosind funcții ale întreruperii BIOS int 10h. **2p**

Obs:

- pentru elementele definite sub forma 'X', se returnează codul ascii corespunzător caracterului;

- codurile ASCII ale cifrelor: 0....9 sunt 30h....39h
literelor mari: A....Z sunt 41h....5Ah

2. Să se scrie o secvență de program care emulează circuitul logic combinațional prezentat în următoarea figură: **3p**



Soluție probleme:

1.a)

;definirea sirului initial si a celor doua siruri destinatie

.data

```
SIR_INIT db 2,'F','A',3,9,4,'T','E','P',8,5,7,'W',1,'J'
CIFRE db 15 dup (?)
LITERE db 15 dup (?)
```

```
b) mov cx, 15 ;contor elemente sir
    mov si, offset SIR_INIT ;adresa initiale siruri
    mov di, offset CIFRE
    mov bx, offset LITERE
```

```
et1: mov al, [si] ;pentru cifrele din SIR_INIT se
            ;returneaza valoarea efectiva iar
            ;pt litere codul ascii coresp.
            cmp al, 40h ;fiecare element din sir va fi
            ;comparat cu valoarea 40h.
            jb et2 ;daca valoarea e mai mica decat
            ;40h elementul repr. o cifra
            mov [bx], al ;daca valoarea e mai mare decat
            ;40h elementul repr. o litera
            inc bx ;se depune majuscula in sirul
            ;LITERE
            jmp et3
et2: mov [di], al ;se depune cifra in sirul CIFRE
            inc di
            jmp et3
et3: inc si ;se trece la urmatorul element
            dec cx ;procedeul se repeta pentru toate
            ;elementele sirului
            jnz et1
```

```
c) ;pozitie cursor cifre:
    mov ah, 02h ;se util functia 02h a int 10h
    mov bh, 0h
    mov dh, 6 ;se alege randul si coloana
    mov dl, 4
    int 10h
    ; afisare cifre
    mov cx, length CIFRE
    mov di, offset CIFRE
    mov ah, 0eh ;pt afisare functia 0eh a int 10h
afis1: mov al, [di]
            add al, 30h ;se determina codul ascii
            ;corespunzator cifrelor
            mov bh, 0
            int 10h
            inc di
            dec cx
            jnz afis1
```

```

                                ;pozitie cursor litere:
mov ah, 02h
mov bh, 0h
mov dh, 7                        ;se alege randul si coloana
mov dl, 4
int 10h

                                ;afisare litere
mov cx, length LITERE
mov bx, offset LITERE
mov ah, 0eh
afis2: mov al, [bx]             ;elementele sirului LITERE sunt
                                ;reprezentate de codul ascii - nu
                                ;mai e necesara nici o conversie

mov bh, 0
int 10h
inc di
dec cx
jnz afis2

2.   mov dx, 80h                ;se citeste portul de intrare 80h
                                ;in registrul al
in al, dx                        ;se real functia logica and intre
                                ;octetul citit si valoarea 55h
and al, 55h                     ;55h=0101.0101b
jz et                            ;daca valoarea obtinuta este 0
                                ;salt la eticheta et
mov al, 0000.0000b              ;daca valoarea obtinuta este 1
                                ;inseamna ca cel putin unul din
                                ;bitii D0,D2,D4,D6 a fost 1,
                                ;rezultatul fuctiei SAU negat e 0
                                ;valoare ce se trimite la portul
                                ;de iesire 110h

mov dx, 110h
out dx, al
jmp go_to
et:   mov al, 0000.0001b        ;inseamna ca bitii D0,D2,D4,D6 au
                                ;fost toti 0, rezultatul fuctiei
                                ;SAU negat este 1
mov dx, 110h                    ;valoare trimisa la portul 110h

out dx, al
go_to: .....

```