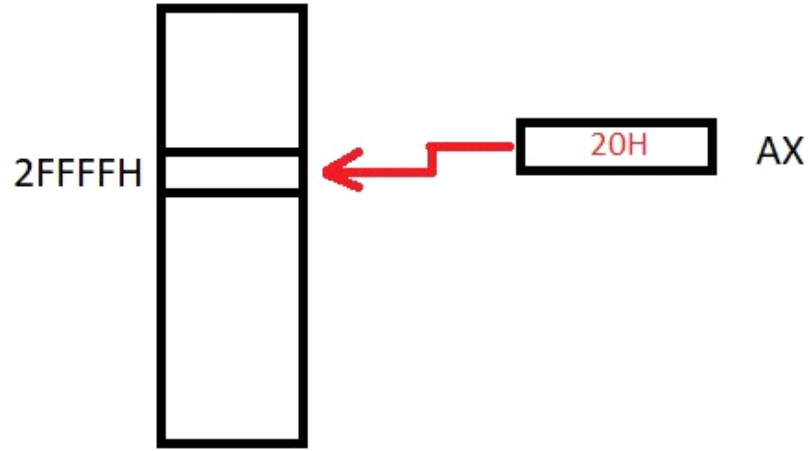


ADI: CANSU**SOYADI: DAL****NO: 18253039**

1. Aşağıdaki şıklarda verilen her bir komutun fiziksel adreslerini hesaplayıp, hedef kaynak ilişkisini gösteren blok çizimini çiziniz. (30p)

BX: 1000H BP: ABCDH AX: 0020H SI: 1000H IP: CFFFH SP: A000H DS: 2000H
SS: 8000H CS:0000H ES:EA00H ARRAY: 1C00H SUM: DDH

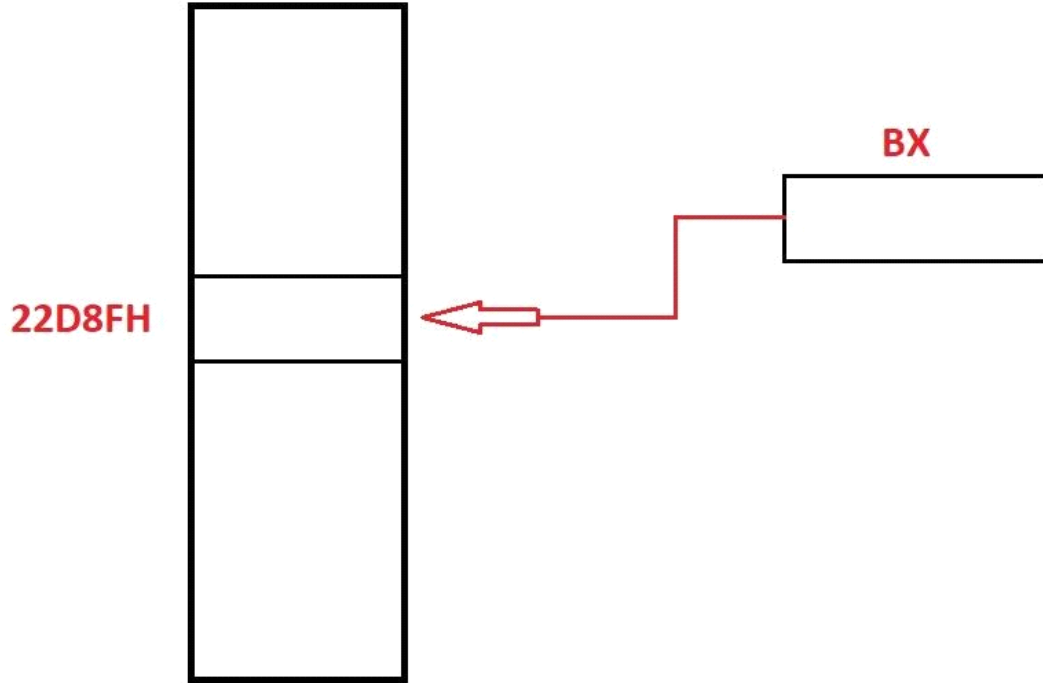
ÖRNEK: MOV [FFFFH],AX



a) MOV ARRAY[SI+18FH],BX

Yanıt :F.A=(DS)*10H+ARRAY+SI+BX

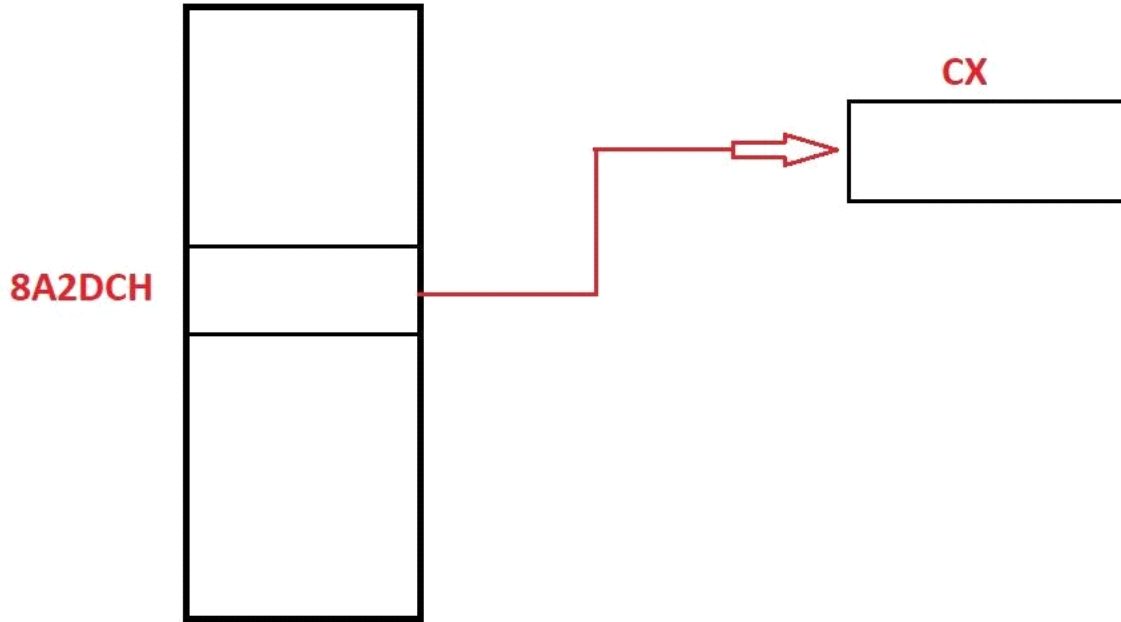
$$F.A = 20000H + [1C00H + 1000H + 18FH] = 22D8FH$$



b) MOV CX, SUM[SP+1FFH]

Yanıt: F.A = (SS) * 10H + [SUM + SP + 1FFH]

$$F.A=80000H+[DDH+A000H+1FFH]=8A2DCH$$



2. F2H adresli giriş birimine bağlı klavyeden girilen son iki adet sayı değerlerini toplayıp, sonuçlarını 30000H adresinden başlamak üzere yükleyen 8086 sistemi için prosedür programı yazılacaktır. Ayrıca program, toplama sonucun '10' dan büyük yada küçük olduğunu F2H adresli çıkış birimine bağlı 7_Seg_Led üzerinde göstermektedir. Eğer sayı '10' dan büyük ise D2H adresli çıkış birimine bağlı Led üzerinde 'G' harfi, eğer küçük ise 'S' harfi yazmaktadır. Led'ler genel anoda göre bağlanmıştır. Toplanacak sayıların her biri 8 bit dir. Toplam 100 adet toplama işlemi yapıldıktan sonra Prosedür Programı sonlanacaktır. Sayının 10'dan büyük yada küçük olma testi **ALT_PRO_LED_YAZ** adındaki alt program çağırılarak kontrol edilmektedir. Buna göre aşağıda verilen boşlukları doldurunuz. (70p)

```
PROC NEAR TUS_TOPLA_2
```

```
MOV DX, 3000H // taban
```

```
MOV DS,DX
```

```
MOV SI,0000H
```

MOV BL,00H // Birinci sayı girişi için temizleme.

MOV CL, 00H // İkinci sayı giriş için temizleme.

DEVAM IN AL,F2H

MOV BL, AL

IN AL, F2H

MOV CL,AL

ADD BL,CL

MOV [SI], BL

CALL ALT_PRO_LED_YAZ

INC SI

CMP SI,64H // 100'ü geçtiyse döngüden çıkılır.

JNE DEVAM

ENDP

ALT_PRO_LED_YAZ

CMP BL,0AH

JG BUYUKYAZ

MOV BL,49H

OUT F2H,BL

JMP SON

BUYUKYAZ MOV BL,41H

OUT D2H,BL

SON HLT

ENDP