Mikroişlemci ve Assembly

(Uygulamalar)

Dr. Cahit Karakuş

2017

İçindekiler

1. Ekrana Yazma Ve Okuma

Örnek: Verilen diziyi okusun ve ekrana yazsın

```
ORG    100h
mov si, 0h
MOV    BX, OFFSET a    ; get address of VAR1 in BX.

atla:     mov dl, a[si]

mov ah,2
int 21h    ; dl'nin içerigini ekrana yazar.

cmp si,18
je stop
inc si
jmp atla
stop:     ret

a db 41h,42h,43h,44h, 45h,46h,47h,48h,49h,4ah,4bh, 4ch,4dh,'N', 'O', 'P', '1', '8', 'z'
```

Açıklama: Dizinin yerinin DS'de bulunması, DS:0700: Bx'in içeriği yazıldığında dizinin yeri bulunur.

Örnek: Diziyi okuyan toplamını bulun yazılım

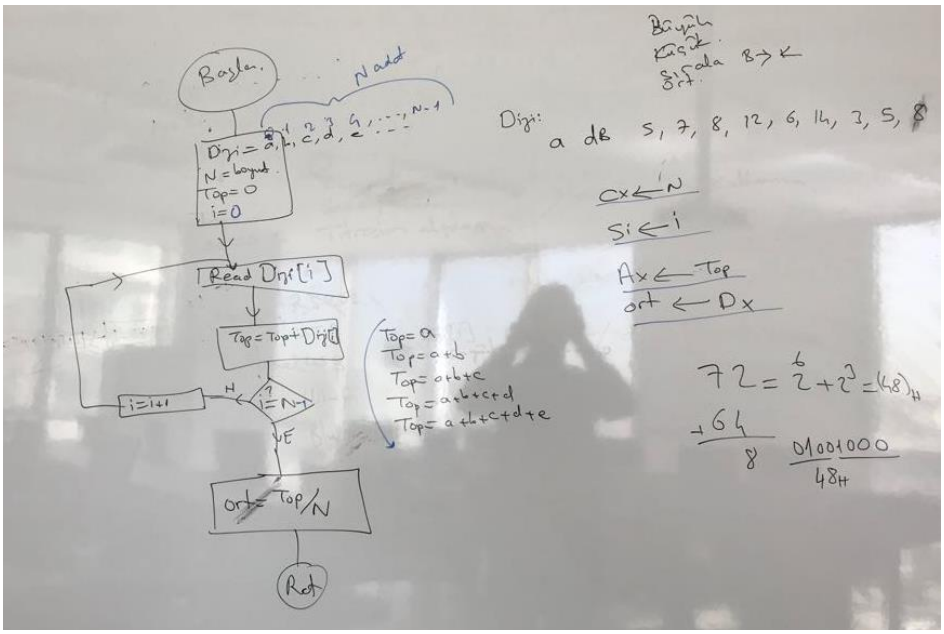
```
            ORG    100h
            mov si,0h
            mov al,0h

atla:       cmp si,12
             je stop

            add al, a[si]
            inc si
             jmp atla
stop:       ret
```

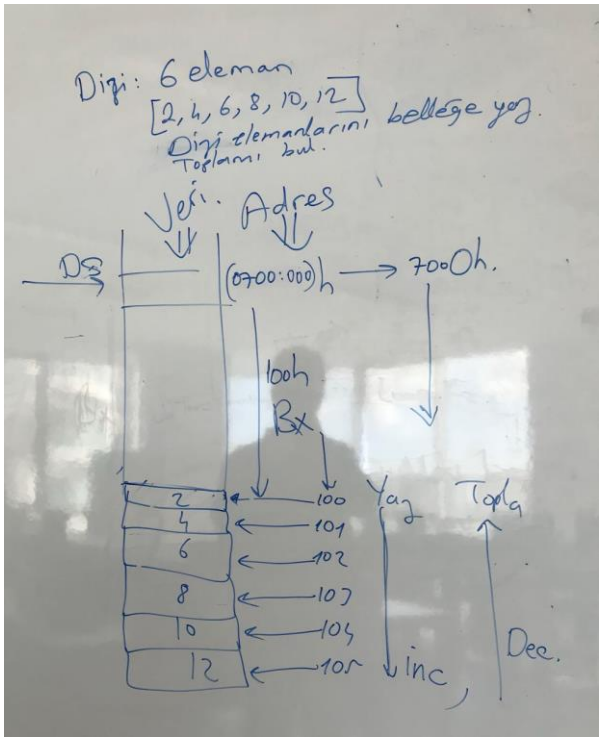
a db 1h, 2h, 3h, 4h, 5h,4h,7h,8h,4h,0ah,0bh, 0ch, 0dh

Belleğe dizi elemanlarını yazma, okuma ve toplamını bulma



; DS: 0700: 0100 =7100h

```
            ORG    100h
            mov bx, 100h
             ;belleğe yazma
            mov [bx],2
            inc bx
            mov [bx],4
            inc bx
            mov [bx],6
inc bx
            mov [bx],8
            inc bx
            mov [bx],10
inc bx
            mov [bx],12
       ; Toplama bulma
             mov cx,6
             mov ax,0
 topla:  mov dl, [bx]
            add al,dl
            dec bx
         loop topla
```



```
        ret
```

EKRANA YAZI YAZMA

```
        org 100h
        mov dx, offset msg
        mov ah, 9
        int 21h
        mov cl,0
        mov bx,offset msg
atla:   mov al,[bx]
        cmp al,024h
        je git
        inc bx
        inc cl
        jmp atla
git:    mov bx,offset msg
        mov dl,0
git3:   mov al,[bx]
        cmp al,'u'
        je git2
        inc bx
        loop git3
        jmp cik
git2: inc dl
      inc bx
      jmp git3
cik:   ret
      msg db "HEP SEN EZILEN $"
```

EKRANA YAZI YAZAR

```
ORG 100h          ; this directive required for a simple 1 segment .com program.
MOV AX, 0B800h    ; set AX to hexadecimal value of B800h.
MOV DS, AX        ; copy value of AX to DS.

MOV CL, 'A'       ; set CL to ASCII code of 'A', it is 41h.
MOV CH, 10011100b ; set CH to binary value.
MOV BX, 15Eh      ; set BX to 15Eh.
MOV [BX], CX      ; copy contents of CX to memory at B800:015E

MOV CL, 'Z'
MOV [BX+2],CX

MOV CL, 'R'
MOV [BX+4],CX

MOV CL, 'A'
MOV [BX+6],CX


RET               ; returns to operating system.
```

KLAVYEDEN KAREKTER OKUMA VE EKRANA YAZMA

```
        org 100h    ; Example 3: Reading and displaying a character:

        mov ah,01h   ; keyboard input subprogram
        int 21h        ; read character into al

        mov dl,al ; copy character to dl
        mov ah, 02h ; character output subprogram
        int 21h ; display character in dl

mov dl,' '
        mov ah, 02h
        int 21h

mov dl,'C'
        mov ah, 02h
        int 21h

mov dl,'A'
        mov ah, 02h
        int 21h

mov dl,'F'
        mov ah, 02h
        int 21h

mov dl,'E'
        mov ah, 02h
        int 21h


mov dl,'R'
        mov ah, 02h
        int 21h
```

2. Aritmetik İşlemler

Toplama

```
ORG    100h
mov ax,5
mov cx,4
mov bx,00000111b
add ax,cx
add ax,bx
add ax,5
mov bx, 100h
mov [bx],0
inc bx
mov [bx],0
dec bx
mov [bx],3
add [bx],ax
ret
```

Toplama

```
org 100h
mov dx, offset MESAJ1 ; Mesaj1 ekrana yazilir.
mov ah, 9
int 21h

mov ah, 1   ; ilk sayi girilir
int 21h
mov bl,al

mov dx, offset MESAJ2  ; Mesaj2 ekrana yazilir.
mov ah, 9
int 21h

mov ah, 1    ; ikinci sayi girilir
int 21h

add bl,al   ; iki sayi toplanir. Dikkat ASCI karsilikleritoplanır.
```

```
ret
MESAJ1 db "Input Number -1$"
MESAJ2 db "Input Number -2$"
```

Aritmetik işlem - 1

```
org 100h
mov ax, 011bbh
mov bx, 022cch
add ax,bx   ; for addition
sub ax,bx   ;for subtraction
mul bx      ; for multiplication
div bx      ; for division
ret
```

Aritmetik işlem - 2

```
org 100h
mov ax, 5 ; load 5 into ax
add ax, 3 ; add 3 to the contents of ax, and ax now contains 8
inc ax    ; add 1 to ax and  ax now contains 9
dec ax    ; subtract 1 from ax and ax now contains 8
sub ax, 6 ; subtract 4 from ax and  ax now contains 2
ret
```

Dört işlem

```
org 100h
mov ax, 0aabh
mov bx, 0dc0h
add ax,bx  ; for addition
sub ax,bx, ;for subtraction
mul bx  ; for multiplication
div bx ; for division
ret
```

1'den N'e kadar toplamı bulma

```
        org 100h
        mov cx,16
        mov ax,0
d:      add ax,cx
        loop d
        ret
```

1'den N'e kadar toplamı bulma

```
        ORG   100h      ;TOP= N*(N+1)/2
        mov cl,20       ;N
        mov al,cl       ;N
        inc al          ; N+1
        mul cl          ; N*(N+1)
        mov bl,2
        div bl          ; N*(N+1)/2
        ret
```

Çıkarma

```
        ORG   100h
        mov al,100
        mov cl,99
        sub al,cl
        add al,cl

        mov bx,100h
        mov [bx], 10
        mov al,11
        sub al,[bx]
        add al,[bx]

        mov al,20
        sub al,5
        add al,6
        ret
```

Daire alanı

```
org 100h
mov ah,01h
int 21h
sub al,30h
mul al
mov bl,3
mul bl
ret
```

Bölme

```
ORG  100h
mov ax,1000
mov cl,10
div cl
mov cx,1000
mul cx
div cx
ret
```

Çarpma

```
ORG  100h
mov al,255
mov cl,255
mul cl
mov bx,100h
mov [bx], 254
mov al, 02h
mul [bx]
mov ax,0c0deh
mov cx,0dff0h
mul cx
mov ax,0h
mov al,-10
mov cl,20
imul cl
mov ax,0c000h
mov cx,0d000h
```

```
        mul cx
        mov al,20d
        mov cl,20d
        mul cl
        sub al,120d
        ret
```

Çarpma

```
        ORG   100h
        mov al,10
        mov cl,20
        mul cl
        mov al,-10
        mov cl,20
        imul cl
        mov ax,0c000h
        mov cx,0d000h
        mul cx
        ret
```

Ortalama bulur

```
        ORG   100h
        mov cx,9
        mov si,cx
        dec si
        mov ax,0
        mov dx,0
git:    mov dl, a[si]
        add ax,dx
        cmp si,0
        je cik
        dec si
        jmp git
cik:    div cl
        ret
        a db 5,6,3,2,7,8,5,4,5
```

Faktoriyel

```
            org 100h
            ; 0 ile 99 arasinde desimal sayi girer
don:        mov ah,1
            int 21h
            cmp al,30h
            jb don
            cmp al,39h
            ja don
            sub al,30h
            mov dl,10
            mul dl
            mov cl,al
don1:       mov ah,1
            int 21h
            cmp al,30h
            jb don1
            cmp al,39h
            ja don1
            sub al,30h
            add al,cl

      ; faktoriyel bulur ( 8 e kadar)
            mov cl,al
            mov ch,0h
            mov ax,1
 eben:      mul cx
            loop eben
cik:        ret
```

DİZİDE BÜYÜK ELAMANI BULMA

```
 org 100h
   mov cl,1
   mov ch,N       ; dizi uzunlugu
   lea bx,vec    ; dizinini baslangic adresi
   mov dh,[bx]    ; max=vec[1]

              L3:   inc cl
            inc bx
            mov dl,[bx]
            cmp dh,dl   ; max > vec[i]
            ja L1
            mov dh,dl

             L1:
            cmp ch,cl
            jne L3
          ret

            N    db 8
            vec  db 43h, 1h, 3h,75h, 9h,7h,2h,66h
         end
```

DİZİDE BÜYÜK BULAN

```
org 100h
 ;********** max bulur
       mov cl, dizi_n   ; dizi uzunlugu
       lea bx, vec ; dizinini baslangic adresi
       mov dl,[bx]; max=vec[1]
       dec cl
cevrim:  inc bx
       mov al,[bx]
       cmp dl,al   ; max > vec[i]
       ja git
       mov dl,al
   git: loop cevrim
 ;********** max  index bulur
              lea bx, vec
              mov cl,dizi_n
        gita:    cmp dl,[bx]
              je cik

              inc bx
              inc index
              loop gita
          cik: mov ax,index
 ;********** max  index led icin rol bulur

              mov cl,al
              dec cl
              mov al,1
              rol al,cl
 ;********** led on
           mov dx, 0378h
           out dx,al
             nop
             ret
  dizi_n db 8
  index  dw 1
  vec    db 43h, 1h, 3h,5h, 9h,7h,2h,66h
```

3. BİLGİSAYAR PARELEL PORT KONTROL

8 BIT PORT KONTROLÜNDE ON/OFF ÇALIŞTIRMA

```
        ; program port control yapar
       org 100h
           mov bx, 0200h ; ledlerin durumu yazildigi ram gozleri
          mov cl,30h   ; ledler 1.....8
don2:
        mov dx, offset mesaj1  ; mesaj1 yazar
       mov ah, 09h
       int 21h
       ; led no ascii karekter ekrana yazar
       mov dl, cl
       mov ah,2
       int 21h
       mov dx, offset mesaj2  ; mesaj2 yazar
       mov ah, 09h
       int 21h
don1:  mov ah,1h   ; led durumu 0 mi 1 mi  kontrol eder
       int 21h
       cmp al,030h
       je cik1
       cmp al,031h
       je cik1
       jmp don1


    cik1:
         sub al,030h     ; led durumu al de
          mov [bx], al    ; ram e led durumunu yazar
           inc bx         ; ram arti
          inc cl          ; led artirildi
           cmp cl,38h
         jne don2
      ret
  mesaj1 db 0dh,0ah,"LED $"
 mesaj2 db "  [0 / 1] : $"
```

BİLGİSAYAR PARELEL PORT KONTROL

```
        ; program port control yapar
       org 100h
  atla:
           mov al,0h
           mov dx,378h
           out dx,al
           in al,dx
           cmp al,0h
           jne atla
          mov bx, 0200h ; ledlerin durumu yazildigi ram gozleri
          mov cl,31h   ; ledler 1.....8
          ; cursor set bilgileri ram e yazilir
          mov [260h],0  ; sayfa
          mov [261h],0   ; satir
          mov [262h],16   ; sutun

 don2:
       mov dx, offset mesaj1  ; mesaj1 yazar
       mov ah, 09h
       int 21h
       ; led ascii karekter ekrana yazar
       mov dl, cl
       mov ah,2
       int 21h
       mov dx, offset mesaj2  ; mesaj2 yazar
       mov ah, 09h
       int 21h
 don1:  mov ah,1h   ; led durumu 0 mi 1 mi  kontrol eder
       int 21h
       cmp al,030h
        je cik1
        cmp al,031h
        je cik1
          mov [250h],bh    ; register lar ram e yazilir
          mov [251h],dh
          mov [252h],dl
```

```
        ; cursor set bilgileri ram den okunur
       mov bh,[260h]  ; sayfa
       mov dh,[261h]   ; satir
       mov dl,[262h]   ; sutun
       mov ah, 2
                int 10h

       ; register lara ram den tekrar data tarsfer edilir
       mov bh, [250h]
       mov dh, [251h]
       mov dl, [252h]
      jmp don1

cik1:
      sub al,030h     ; led durumu al de
       mov [bx], al    ; ram e led durumunu yazar
       inc bx         ; ram arti
       inc cl         ; led artirildi

  mov dl, 0dh          ; enter
  mov ah,2
  int 21h
          mov dl, 0ah   ; satir besle
  mov ah,2
  int 21h
     cmp cl,039h         ; led 8 den buyudumu
  je cik2
  inc [261h]
  jmp don2
 cik2:
    mov bx,0200h       ; ram den okur
    mov ch,0
    mov cl,0
don3:     mov al,[bx]
    rol al,cl
    add ch,al
    cmp cl,8
```

```
        je cik3
        inc cl
        inc bx
        jmp don3

    cik3:
         nop
         nop

        mov dx, 0378h
        mov al,ch
        out dx,al
         nop
          mov ah,1
          int 21h
          cmp al, 0dh
          je atla

     ret
         mesaj1 db "LED $"

    mesaj2 db "  [0 / 1] : $"
```

ASCII TO BINARY

```
        org 100h
        mov bx,0200h
        mov ah,1h   ; Klavyeden character okur
        int  21h
        mov dh,al
        mov dl,"="  ; EKRANA = YAZAR
        mov ah,2h
        int 21h
        mov dl,1h
        mov cx,8
don:  and al,dl
        mov [bx],al
        mov al,dh
```

```
        ror al,1
        mov dh,al
        inc bx
        loop don
        mov cx,8h
        mov bx,0207h
don1: mov dl,[bx]
        add dl,030h
        mov ah,2
        int 21h
        dec bx
        loop don1
        mov dl,"b"
        mov ah,2
        int 21h
        ret
```

ALT PROGRAM

```
ORG    100h
MOV    AL, 1
MOV    BL, 2
CALL   m2
CALL   m2
CALL   m2
CALL   m2
RET              ; return to operating system.
m2     PROC
MUL    BL          ; AX = AL * BL.
RET              ; return to caller.
m2     ENDP
END
```

4. ALT PROGRAM

Example-1:

```
org    100h
   mov ax,0ccbbh
   mov bx,0ddeeh
   push ax
   push bx
   mov    AL, 5
   mov    BL, 2
    mov cl,5
don: call   m2
   loop don
   mov dx,ax
   pop bx
   pop ax
   RET              ; return to operating system.

m2     PROC
MUL    BL           ; AX = AL * BL.
RET               ; return to caller.
m2     ENDP
END
```

Example-2:

```
ORG    100h
MOV    AL, 1
MOV    BL, 2
CALL   m2
CALL   m2
CALL   m2
CALL   m2
RET               ; return to operating system.

m2     PROC
MUL    BL           ; AX = AL * BL.
RET               ; return to caller.
m2     ENDP

END
```

Example-3: to print a Hello World! message

```
ORG    100h
LEA    SI, msg       ; load address of msg to SI.
CALL   print_me
RET                  ; return to operating system.

; ========================================================
; this procedure prints a string, the string should be null
; terminated (have zero in the end),
; the string address should be in SI register:
print_me     PROC

next_char:
   CMP  b.[SI], 0    ; check for zero to stop
   JE   stop         ;

   MOV  AL, [SI]     ; next get ASCII char.

   MOV  AH, 0Eh      ; teletype function number.
   INT  10h          ; using interrupt to print a char in AL.

   ADD  SI, 1        ; advance index of string array.

   JMP  next_char    ; go back, and type another char.

stop:
RET                  ; return to caller.
print_me     ENDP
; ========================================================

msg    DB  'Hello World!', 0   ; null terminated string.

END
```

CURSOR SET

```
org 100h
mov bh,0
mov dh,10
mov dl,20
mov ah,2
int 10h
mov dl,"S"
 mov ah,2h
int 21h
mov bh,0
mov dh,20
mov dl,10
mov ah,2
int 10h
mov dl,"T"
mov ah,2h
 int 21h
ret
```

CHARACTER TO HEX CONVERTER

```
org 100h
  ; 0... 15 RAM gozune yazildi
 mov bx,0200h
 mov cx,16
 mov al,0
  don:    mov [bx],al
 inc bx
 inc al
  loop don
   ;  ram den okur ekrana yazar
   mov cx,16
    mov bx,0200h
don1:   mov al, [bx]
    cmp al,9
    ja git1
    add al,30h
     jmp git2
git1:   add al,37h
```

```
git2: mov [bx+20h],al
          inc bx
       loop don1
       mov ah,1h  ; KAREKTER OKUR
       int 21h
         mov cl,al
          mov dl,"="
          mov ah,2h
          int 21h
       mov bx,220h
          mov al,cl
       and al,0f0h
       ror al,4
         mov ah,0
         add bx,ax
         mov dl,[bx]
          mov ah,2h
          int 21h
        mov bx,220h
          mov al,cl
       and al,0fh
         mov ah,0
         add bx,ax
        mov dl,[bx]
          mov ah,2h
          int 21h
          mov dl,"h"
          mov ah,2h
          int 21h
          ret
```

5. ASCII

```
; ASCII
            org 100h    ; Example 3: Reading and displaying a character:
            mov ah,01h   ; keyboard input subprogram
            int 21h        ; read character into al
            mov cl,255
don:  mov dl,cl   ; keyboard input subprogram
            mov ah,2
            int 21h
            loop don
            ret
```

6. SORULAR

Soru-1: Bellek fiziksel başlangıç adresi (A000)h, bellek fiziksel bitiş adresi (BFFF)h ise

a) Belleğin başlangıç indis adresi 0 alınırsa bitiş indis adresi nedir?

Bellek bitiş indis adresi= Bellek fiziksel bitiş adresi - Bellek fiziksel başlangıç adresi

Bellek bitiş indis adresi = (BFFF)h –(A000)h=(1FFF)h

b) Belleğin kapasitesi kaç Kbyte'dır?

Bellek kapasitesi= Bellek bitiş indis adresi + 1

(1FFF)h +1=(2000)h

(0010 0000 0000 0000)b ise indis=2^13 byte=8Kbyte

Soru-2:

ROM bellek: U0 =24Kbyte

RAM bellek: U1 = 31Kbyte

RAM bellek: U2= 45Kbyte

RAM bellek: U3= 88Kbyte

RAM bellek: U4= 188Kbyte

a) Herbir bellek kaç byte alınır?

U0, ROM bellek: U0 =24Kbyte; 32Kbyte=2^15byte;

U1, RAM bellek: U1 = 31Kbyte; 32Kbyte=2^15byte;

U2, RAM bellek: U2= 45Kbyte; 64Kbyte=2^16byte;

U3, RAM bellek: U3= 88Kbyte;128Kbyte=2^17byte;

U4, RAM bellek: U4= 188Kbyte;256Kbyte=2^18byte;

b) Herbir belleğin adres hattı sayısı nedir? Herbir belleğin adres hatlarını indisleyiniz.

U0, ROM bellek: Adres hattı sayısı=15adet; Adres Bus İndis: A14, A13, … , A1, A0

U1, RAM bellek: Adres hattı sayısı=15adet; Adres Bus İndis: A14, A13, … , A1, A0

U2, RAM bellek: Adres hattı sayısı=16adet; Adres Bus İndis: A15, A14, … , A1, A0

U3, RAM bellek: Adres hattı sayısı=17adet; Adres Bus İndis: A16, A15, … , A1, A0

U4, RAM bellek: Adres hattı sayısı=18adet; Adres Bus İndis: A17, A16, … , A1, A0

c) Maksimum indisi bulunuz.

A17

a) Bellekleri seçmek için adres dekoding devresinin çıkışları kullanılmaktadır. Girişleri ise adres hattı olarak CPU'dan gelmektedir.

Adres Dekoding devresi girişi, n=2^m dir. Burada n: bellek sayısı, m= CPU'dan çıkan ve adres dekoding devresine giriş yapan adres hattı sayısıdır. O halde n=5 ise m değerlerini bulunuz.

5=2^m; m=3 alınır.

b) CPU'dan çıkıp adres dekoding devresinin girişine gelen "m" adres bus hatlarını indisleyiniz.

Maksimum Adres Bus indis: A16 olduğundan adres dekoding devresine gelen adres bus hatları indisi: A20, A19, A18 olur.

c) Adres dekoding devresine gelen adres hatlarından herbir belleği seçecek lojik devrenin doğruluk tablosunu oluşturun. Hangi giriş durumunda hangi bellek seçilir.

Doğruluk Tablosu							
Giriş			Çıkış				
A20	A19	A18	U0	U1	U2	U3	U4
0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Giriş			Çıkış
A20	A19	A18	
0	0	0	U0
0	0	1	U1
0	1	0	U2
0	1	1	U3
1	0	0	U4
1	0	1	Y0
1	1	0	Y1
1	1	1	Y2

d) CPU'dan çıkan toplam adres hatlarını indisleyin.

CPU'dan çıkan toplam adres hatları indisleri: A20, A19, …, A1, A0

e) CPU'dan çıkan toplam adres hattı sayısı nedir?

CPU'dan çıkan toplam adres hattı sayısı=21

f) CPU bellek adresleme kapasitesi kaç Mbyte'dır?

CPU bellek adresleme kapasitesi= 2^21=2Mbyte

Soru-3: Aşağıdaki soruları yanıtlayınız.

1) Toplam kapasite= 2^n byte olarak hesaplanır. Buradaki n ifadesi adres hattı sayısıdır. Adres hattı sayısı, n=43 ise bilgisayar sistemin adresleme kapasitesi kaç Terabyte'dır.
   8Tbyte

2) Bilgisayar sisteminin önemli birimi olan adres decoding devresinin amacı nedir?
   Belek ya da I/O birimlerinden birini seçmek

3) CPU mimarisindeki hangi segmenteki verilere erişmek için SP ve BP indis register'ları kullanılır.
   SS

4) Tek transistör ve yarı iletken depolama kapasitörlerinde bitler şarj olarak saklanır. Şarj etme sızıntısı vardır bu nedenler güç verildiğinde bile yenilenmesi gerekir. Daha basit yapıdadır. Bit başına bir transistör vardır. Ucuzdur. Yenileme devrelerine ihtiyacınız var (birkaç milisaniyede bir). Yavaş. Ana Bellek RAM olarak kullanılır. Bu bellek sınıfına ne ad verilir?
   DRAM

5) (1011 1011 1110 1100)b ikili değerinin hex karşılığını bulunuz.
   (BBEC)h

6) (AD7E)h hex değerinin ikili karşılığını bulunuz.
   (1010, 1101 0111 1110)b

7) İşlemcinin 98 Gbyte kapasiteli belleğe erişmesi için kaç adres hattı gerekir?
   98Gbyte yerine 128 Gbyte alınır. 128Gbyte=2^7*2^30=2^37, 37 adet adres hattı gerekir.

8) CS=A000h, DS=B000h, SS=C000h, ES: D000h, Bx=1000h, DI=2000h, SI=4000h, SP=5000h, BI=3000h ise Mov Ax, SS:[BI] komutunda bellek gözünün fiziksel adresi nedir?
   Fiziksel adres=C000$\mathbf{0}$ + 3000h = C3000h

Soru-4: Aşağıdaki soruları yanıtlayınız.

1) Başlangıç Adresi stack segment'te kayıtlı olan belleğin fiziksel adresi BC000h olan belleğin gözünün indis register'ı olan SP'nın içeriği nedir? Stack Segment Register içeriği A000h alınacaktır. İlgili bellek gözüne (AA)h değerini yazan assemble kodu yazın.

   Belleğin gözünün indis register'ının içeriği= Bellek gözünün fiziksel adresi - Stack Segment Register içeriği*10h
   Belleğin gözünün indis register'ının içeriği, SP=(BC000)h- (A000)h*10h
   Belleğin gözünün indis register'ının içeriği, SP=(BC000)h- (A0000)h=(C000)h

   Mov Ax, 0A000h
   MOV SS, Ax
   BP: 0C000h
   Mov SS:[SP], (0AC)h

2) (22)h desimal sayısının tersini bulan assemble kodu yazın. Sonuç ne olur?
   MOV AL, 22h
   NOT AL
   (DD)h
3) 9 ve 5 sayılarını AL ve BL register'larına transfer eden, toplayan ve değerini 4 ile çarpan assemble kodu yazın. Sonuç hangi rester içerisindedir ve değeri nedir?
   MOV AL, 9
   MOV BL, 5
   ADD AL, BL
   MOV CL, 4
   MUL CL

4) Aşağıda kodu yazılmış assemble programın sonucunda AL register içeriği ne olur?
   MOV AL, (B)h
   MOV BL, (7)h
   OR AL, BL
   (F)h

7. Kaynaklar

1) *http://history.acusd.edu/gen/recording/computer1.html*
2) *http://www.cs.virginia.edu/brochure/museum.html*
3) *http://www.columbia.edu/acis/history/650.html*
4) *http://www.piercefuller.com/collect/pdp8.html*
5) *http://www.computer50.org/kgill/transistor/trans.html*
6) *The History of The Microprocessor, Bell Labs Technical Journal, Autumn, 1997*
7) *http://www.intel.com*
8) *www.cs.sjsu.edu/faculty/lee/chapter3_presentation2.ppt*
9) *https://profs.basu.ac.ir/.../722.1869.file_ref.1998.2468.ppt*
10) *www.abandah.com/.../22446_S11_Intro_to_microprocessor...*
11) *rise.cse.iitm.ac.in/people/faculty/kama/prof/x86_1.ppt*
12) *www.cse.unsw.edu.au/~cs2121/.../week3_notes.pp*
13) *https://users.cs.jmu.edu/.../IntelProcessors4004ToPentiumPr...*
14) *Microprocessor, Atul P. Godse, Deepali A. Gode, Technical publications, Chap 11*
15) *The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware, and Applications (4th Edition) Paperback – Aug 29 2002 by Walter A. Triebel (Author), Avtar Singh (Author).*
16) *Dandamudi S., 2003. Fundamentals of Computer Organization and Design, Springer-Verlag, Heidelberg*
17) *Mano, M.M., 2007. Bilgisayar Sistemleri Mimarisi, Literatür Yayıncılık, İstanbul*
18) *Buzluca F., "Bilgisayar Mimarisi Ders Notları", Bilgisayar Mühendisliği Bölümü, İTÜ Bilgisayar ve Bilişim Fakültesi*
19) ***https://tr.wikipedia.org/wiki/Aritmetik_mant%C4%B1k_birimi***
20) *CSE 675.02: Introduction to Computer Architecture, Slides by Gojko Babiç*