

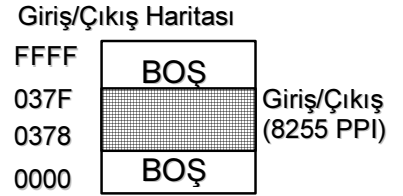
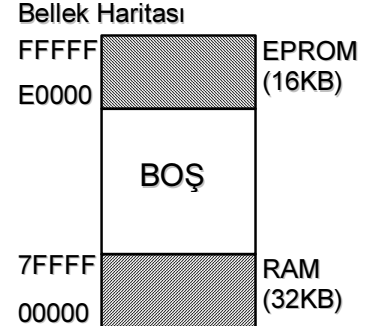
| |
|---|
| 1 |
| 2 |
| 3 |
| T |

Öğrenci Numarası:

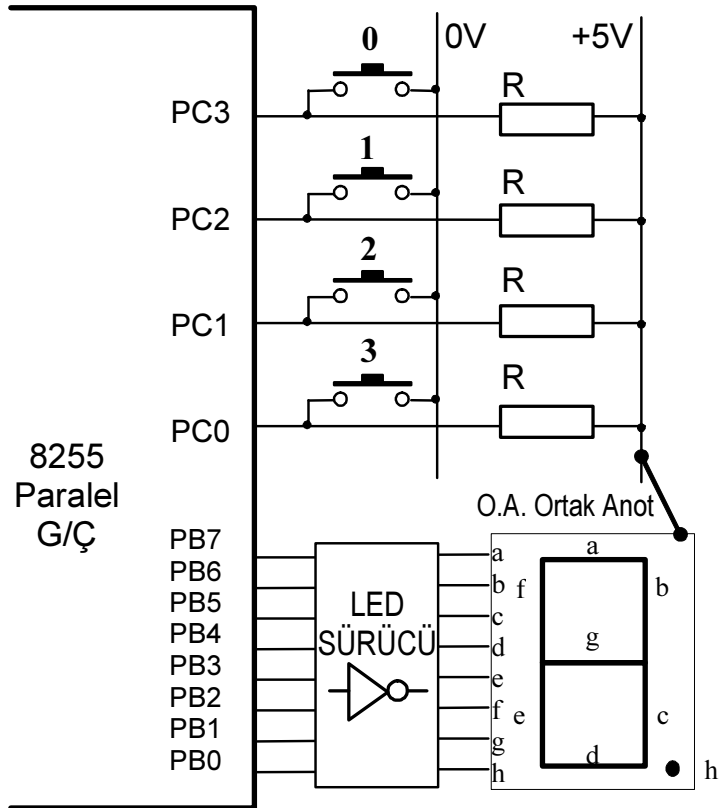
Adı ve Soyadı:

S1(35p). Yanda bellek haritası ve giriş/çıkış haritası verilen minimum mod 8086 mikroişlemci temelli bir sistem tasarlanacaktır.

- Sistemin adres çözümü devresini en az sayıda kod çözücü tümeleşik devre kullanarak ayrıntılı olarak tasarlayınız. (Ek olarak yalnız NOT kapısı kullanılabilir)
- Sistemin blok diyagramını yol, ad ve genişliklerini belirterek çiziniz.
- Bu sistem kullanılarak 378H çıkış adresinden başlayarak 16-Bit büyüklüğündeki veri paralel çıkış birimine yazılacaktır. Giriş/Çıkış birimi donanımını, ayrıntılı blok diyagramını çizerek tasarlayınız. Yazılımını yazarak kısaca açıklayınız.



S2(30p). Aşağıda verilen 8255 paralel ve 8251 seri giriş/çıkış birimi bulunan 8086 mikroişlemci temelli bir sistemde basılan tuşun değerini göstergede görünmesini ve ASCII kodunun seri veri olarak uzağa gönderilmesini sağlayan makine dili yazılımı tasarlayarak yazınız ve nasıl çalıştığını kısaca AÇIKLAYINIZ.



S3(35p). Aşağıda 8086 temelli bir sisteme güç verildiğinde ilk olarak çağrılan 8086 makine dili alt programın analizi yapılacaktır.

- Programdaki her satırının açıklamasını yanına yazınız. (*) işaretli satırdaki komutların makine kodlarını bulunuz.
- Programın analizini, kullanılan bellek gözleri, yazmaçlar ve durum yazmacının Z (sıfır) biti üzerinde yapınız.
- Programda (*) işaretli satırdaki komutların çalışma süresini bulunuz.

Yazmaçların ilk durumu:

AX=0001 BX=0002 CX=0003 DX=0004 SP=0000 BP=0000 SI=0000 DI=0003
DS=0CCC ES=0CCC SS=0CCC CS=0CCC IP=0022 NV UP EI PL NZ NA PO NC

Veri belleğinin ilk durumu:

0CCC:0000 EB 1A 01 00 34 12 09 00 00 00 0C 01 FF FF FE
0CCC:0010 FD FF FC FF 01 00 01 00 01 00 01 00 B9 02 00
0CCC:0020 04 00 BF 14 00 03 44 14 E8 11 00 03 44 04 D1

makina dili program:

```
.MODEL small
.STACK 100h
.DATA
S0 DW 1
A0 DW 1234h,9,0,10Ch
A1 DW -1,-2,-3,-4
M0 DW 4 DUP (1)
.CODE
ORG 22h
(*) MOV CX,2
(*) LEA SI,A0
LEA DI,M0
L1: ADD AX,WORD PTR [SI+OFFSET M0]
(*) CALL TASI
(*) ADD AX,WORD PTR [SI+OFFSET A0]
SAL DX,1
ADD DX,WORD PTR [SI+OFFSET A1]
MOV AX,DX
DEC SI
DEC SI
(*) JZ SON
LOOP L1
SON: RETF
TASI: STOSW ;AX → ES:[DI] ve, eğer DF = 0 ise DI = DI + 2 değilse DI = DI - 2
RET
END
```

SÜRE: 90 dakika.

BAŞARILAR.