

İsim :	2017/1 Mikroişlemci Sistemleri	Soru 1	Soru 2	Soru 3	Toplam
No :	Vize 1 - 9 Kasım 2017	(30p)	(30p)	(40p)	(100p)
İmza :	Süre: 90 dk				

NOT1 : Tüm sorular için izole hafıza haritalama (isolated mapping) kullanın
NOT2 : Tüm μP uçlarının uygun şekilde ayrıştırılmış ve tutulmuş olduğunu varsayın

Soru 1) Aşağıdaki boşlukları uygun şekilde doldurunuz: (30p)

- 1) İlk dijital bilgisayar, elektromekanik elemanlar (röle) kullanılarak Konrad Zuse (kişi) tarafından tasarlanmıştır. (2p)
- 2) PIC mikrodenetleyici ailesinde, veri ve program için farklı yollar ve hafıza elemanları kullanılır. Bu yapıyla PIC ailesi Harvard mimarisindedir. (2p)
- 3) PIC mikrodenetleyici ailesinde komut seti, eşit uzunluktaki (bit bazında), az sayıda ve basit komutlardan oluşmaktadır. Bu yapıyla PIC ailesi RISC mimarisindedir. (2p)
- 4) Mikroişlemcide aritmetik, lojik işlem sonuçlarını saklayan yazmaca akümülatör denir. (2p)

Adres	İçerik
100H	AB H
101H	CD H

- 5) Motorola 6800 mikroişlemcisi ABCDH değerini hafızada ilgili adrese yandaki şekildeki gibi yerleştirir. Bu özelliği ile Motorola 6800 ailesi big endian mimarisindedir. (2p)

- 6) ALU, yazmaçlar, kontrol birimi ve birimler arası yollar içeren, tek çipte tasarlanmış ve üretilmiş, genel amaçlı yapılara mikroişlemci denir. (2p)
- 7) 8086 adres yolu genişliği 20 bittir. (2p)
- 8) 80486 için uygun hafıza arayüzü $1G \times 32$ olduğuna göre, 80486 adres yolu genişliği 32 bittir. (2p)
- 9) 8086 için uygun hafıza arayüzü 512K \times 16 kapasitesindedir. (2p)
- 10) 8086 I/O uzayı 64K byte boyutundadır. (2p)
- 11) Adres çözümleme devresi belirli bir zamanda sadece bir çevre birimin aktif olmasını sağlayarak, farklı entegrelerin aynı anda, aynı yola (bus) farklı veri yazmasını engeller. (2p)
- 12) 8251 arayüzü I/O uzayında 2 byte yer kaplar. (2p)
- 13) 8255 mod 1'de input yönlü ayarlamışsa port uçlarındaki verinin 8255'e otomatik olarak alınmasını sağlayan STB anlaşma (handshaking) ucudur. (2p)
- 14) 8086'da AD_i uçlarından adres bilgisi ayrıştırılırken ALE ucu kullanılır. (2p)
- 15) OUT 21H, AL komutunun işletilmesi sırasında BHE ucu Ø değerini alır. (2p)

Soru 2) 8086 μP 'li bir sistemde bir adet 8255'in A8H adresinden itibaren ardışık çift adreslere ve bir adet 8251'in A9H adresinden itibaren ardışık tek adreslere yerleştirilebilmesi için: (Not: μP ; 8255 veya 8251'i adresleyebilmeli) (30p)

- a) Gerekli adres çözümleme devresini bir adet 3×8 dekodere ve en az sayıda gerekli lojik kapılar kullanarak tasarlayınız. (20p)
- b) 8255 ve 8251 uçlarına, adres çözümleme devresi ve 8086'dan yapılması gereken bağlantıları çizerek gösteriniz. (10p)

2) a) 8255:

A8H :	1010	1000
AAH :	1010	1010
ACH :	1010	1100
AEH :	1010	1110

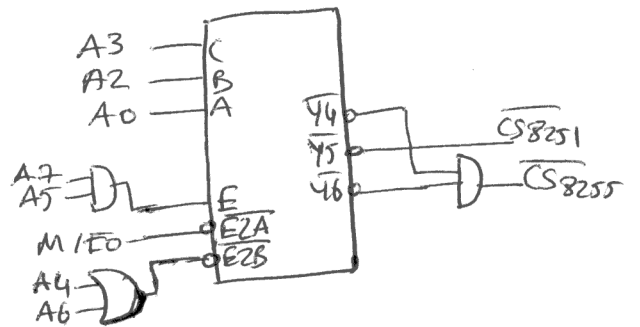
$\rightarrow A1, A2 : 8255$ adres uçlarına

8251:

A9H :	1010	1001
ABH :	1010	1011

$\rightarrow A1 : 8251$ adres ucuna

$$M/\bar{IO} = 0$$

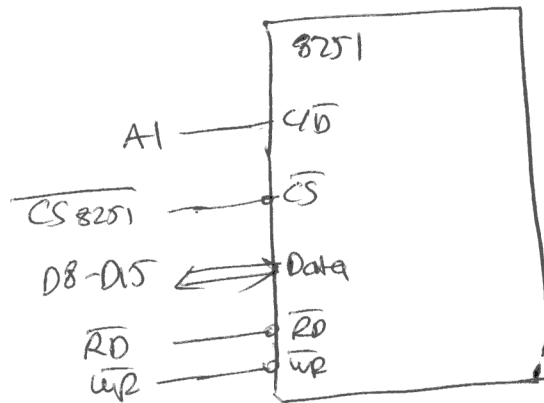
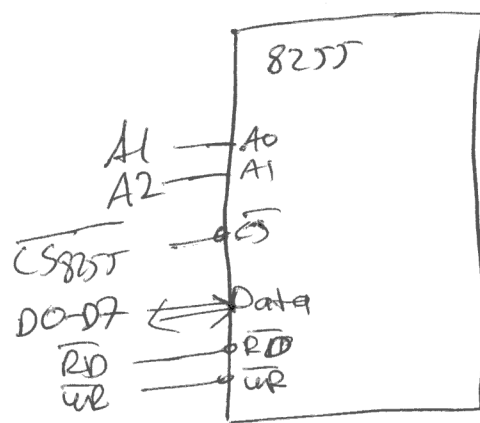


Verilen tüm adres ve işlemlerde ortak uçlar ve değerleri:

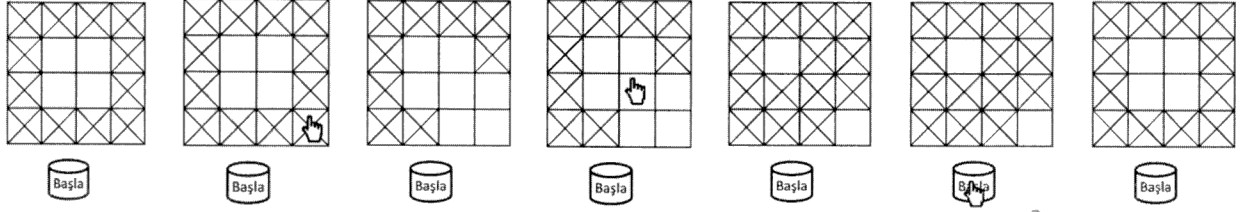
$$\begin{aligned} A7 &= 1 & A6 &= 0 \\ A5 &= 0 & A3 &= 1 \\ A4 &= 1 & M/\bar{IO} &= 0 \end{aligned}$$

$A0, A1, A2$ seçim uçlarına bağlanabilir. (adres aralıklarında değişimi gösterir). Bunlardan herhangi birini seçim ucu olarak bağlamamak \Rightarrow bu ucu 0 veya 1 değeri adres çözümlemeye etkin değil.

2) b)



Soru 3) 8086 μ P ve 8255 kullanarak "Lights Out" isimli oyunun donanımsal kullanıcı arayüzünü tasarlamamız istenmektedir. Oyun 4x4 boyutundaki hücre yapısında bir düzende oynanır. Her bir hücre bir adet LED ve bir adet buton içermektedir. Oyun ilk açıldığında veya BAŞLA isimli tuşa basıldığında, önceden belirlenmiş bir paterne göre oyun tahtasındaki LED'ler yakılır. Oyuncu herhangi bir hücreye basarak, bastığı ve 4 komşuluğundaki hücrelerin her birinin LED'ini yanıyorsa söndürür, söndükse yakar (toggle). Oyunun amacı verilen bu kurala göre, başlangıç durumuna kıyasla, olabildiğince çok ışık söndürmektir. Örnek bir oyun akışı aşağıda verilmiştir.



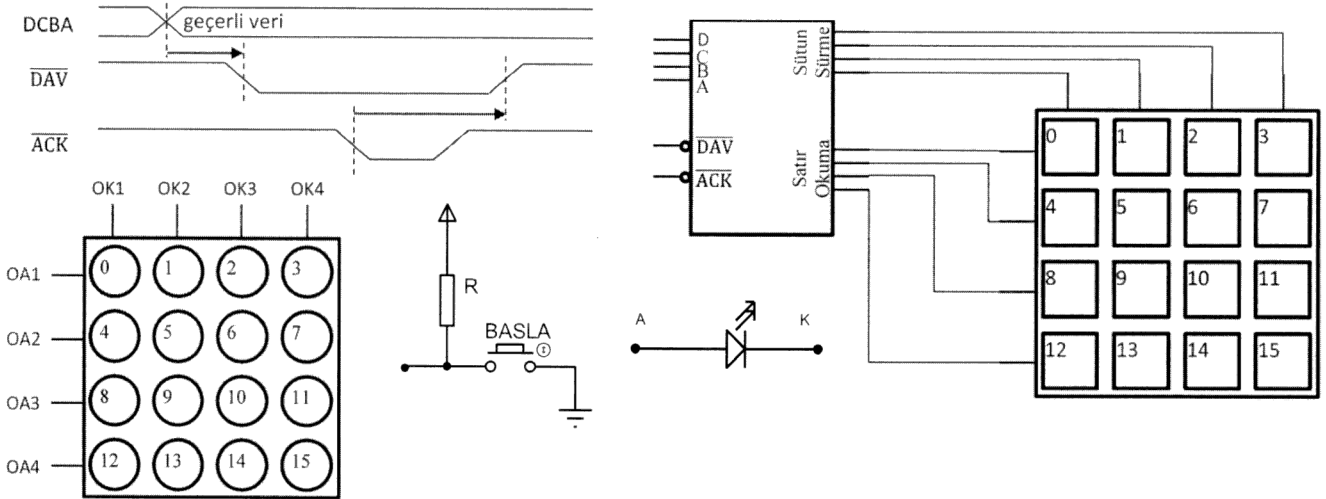
Bu amaçla 4x4 tuş tarama entegresi, 4x4 tuş tarama takımı, 4x4 LED dot-matrix, 1 adet (başla) buton ve 18H adresinden itibaren ardışık çift adreslere yerleştirilmiş bir 8255 kullanılacaktır.

4x4 tuş tarama entegresi ve 4x4 tuş tarama takımının bu entegreye bağlantısı şekildeki gibi hazır olarak verilmiştir. Tarama entegresi, herhangi bir tuşa basıldığında (kutular içinde yazılı değere uygun şekilde) tuş değerini DCBA uçlarından binary olarak çıkartarak, veri hazır (\overline{DAV}) ucunda 0 değeri oluşturur, onay (\overline{ACK}) girişi ucunda 0 değeri okunduğunda ise \overline{DAV} ucunu tekrar 1 yapar. 4x4 tuş tarama entegresinin çalışma mantığı dalga şeklinden de görülebilmektedir.

4x4 LED-dot-matrix'te aynı satırdaki tüm LED'lerin ortak anot uçları OA isimli pinlerle ve aynı sütundaki tüm LED'lerin ortak katot uçları ise OK isimli pinlerle sunulmaktadır. 4x4 LED-dot-matrix yapısı şekil ile verilmektedir.

BAŞLA butonu pull-up direnç ile şekilde verildiği gibi sürülmeştir.

Bu bilgiler çerçevesinde oyun için donanımsal kullanıcı arayüzünü aşağıda istenenlere göre tasarlayınız: (40p)



- 4x4 LED-dot-matrix tek başına düşünüldüğünde, sadece 2 numaralı hücredeki LED'i yakabilmek için tüm OA ve OK uçlarına verilmesi gereken değeri belirleyiniz. (3p)
- 4x4 LED-dot-matrix 8255 PortB ile, 4x4 tuş tarayıcı 8255 GrupA ile Mod 1'de ve BAŞLA butonunu 8255 PortC0 nolu uç ile sürülmek istenmektedir. Bu verilenleri birlikte sağlayan tek bir kontrol kelimesini belirleyiniz. (5p)
- Bulduğunuz kontrol kelimesini 8255'e yazan assembly komutlarını yazınız. (4p)
- 8255 PortC0 ucu ile sürülen BAŞLA tuşuna basılıp basılmadığını tespit eden assembly kod parçası yazınız. (5p)
- 4x4 tuş tarama entegresinin 8255 GrupA Mod1 ile kullanılabilmesi için 4x4 tuş tarama entegresinin uçları ile 8255 uçları arasında yapılması gereken bağlantıyı çizerek gösteriniz. (5p)
- Basılan tuş değerinin, 8255'te hazır olup olmadığını INTRA değeri üzerinden kontrol eden, veri hazır ise bu değeri TUS isimli değişkende saklayan assembly kod parçasının yazınız. (INTEA bitini uygun şekilde ayarlamayı unutmayın) (5p)
- 4x4 LED-dot-matrix'i 8255 PortB ile uygun şekilde sürebilmek için 4x4 LED-dot-matrix ile 8255 uçları arasında yapılması gereken bağlantıyı çizerek gösteriniz. (3p)
- LEDS DB 1, ..., 0, 1 şeklinde tanımlanmış bir dizi (16 elemanlı, 1:LED on, 0:LED off) ile LED'lerin yakılıp yakılmayacağı durumu tutulmaktadır. 4x4 LED-dot-matrix'i LEDS dizisindeki değerlere göre süren, sonsuz döngü şeklindeki assembly kod parçasını yazınız. (tam olarak 5ms bekleme sağlayan `DELAY_5MS` hazır fonksiyonunu kullanabilirsiniz) (5p)

- i) Oyunda basılan tuş ve 4 komşuluğundaki değerlerin toggle ettirilmesi istenmektedir. Basılan tuşun değeri (0-15 arasında bir değer) TUS isimli değişkende mevcutken, LEDS dizisinde istenen toggle işlemlerini yapan assembly kod parçasının işaretli kısmını tamamlayınız. (3p)
- j) En az 1 LED'in yanık olduğu tüm başlangıç durumları göz önüne alındığında, "Lights Out" oyunu için elde edilebilecek en başarılı sonuçta kaç LED yanık kalır? 0 (2p)

(i şikkı için)...

CMP TUS, 05H

JE TUS5

...

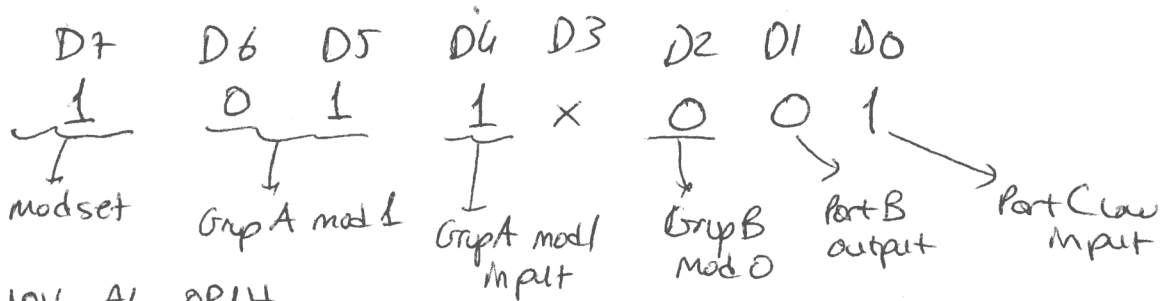
TUS5: ...

TUS6: ...

3) a) OA1 = logik 1 OK1 = logik 1
 OA2 = logik 0 OK2 = logik 1
 OA3 = logik 0 OK3 = logik 0
 OA4 = logik 0 OK4 = logik 1

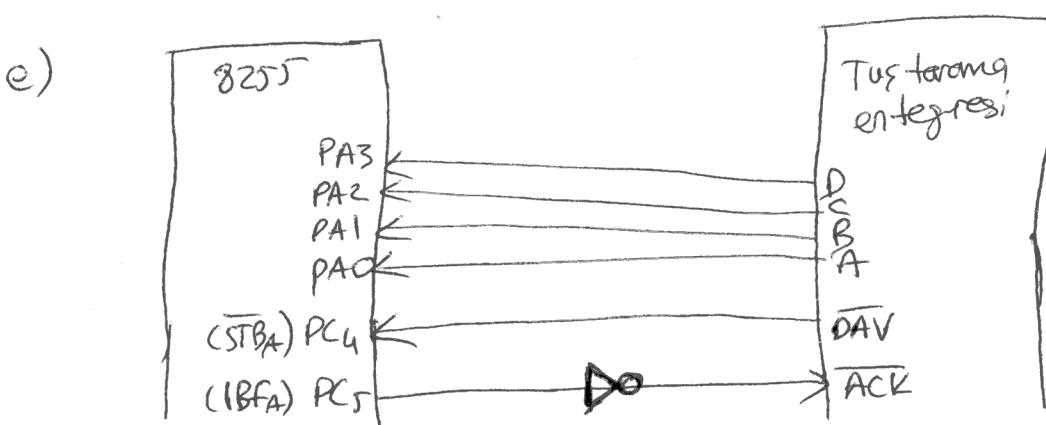
b) PortB → Led matris ⇒ PortB Mod 0 output
 Grup A → tuş tarama entegresi ⇒ Grup A Mod 1 input
 PortC_{low} → buton ⇒ PortC_{low} Mod 0 input

Kontrol kelimesi = B1H



c) MOV AL, 0B1H
 OUT 1EH, AL

d) IN AL, 1CH
 TEST AL, 01H
 JZ TUS_BASILI
 JMP TUS_BASILI-DEFIL



f) MOV AL, 0000 1001B ; BSR modunda PCU lüset et: INTEA=1 olur
OUT IEH, AL

TUS-YOK: IN AL, 1CH ; status reg. oku.

TEST AL, 08H ; INTRA=1'ni kontrol et

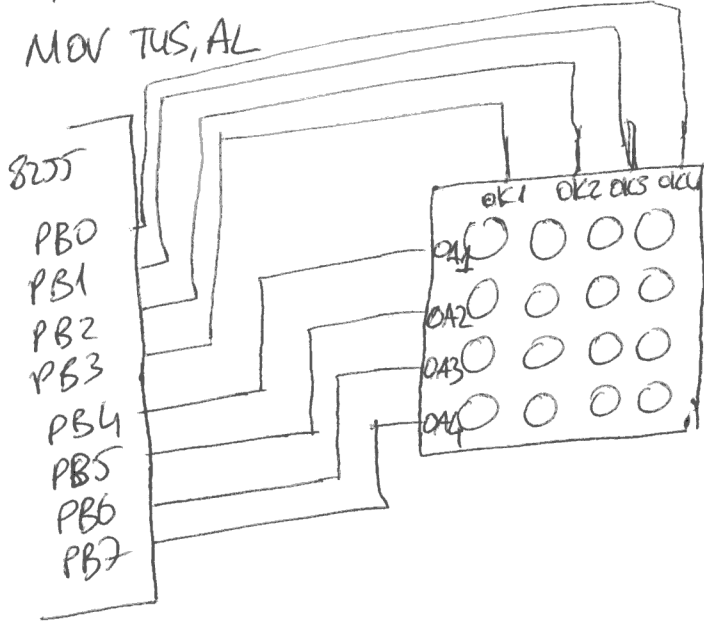
JZ TUS-YOK

IN AL, 1BH ; Port A'daki değeri oku

AND AL, 0FH ; yüksek anlamlı bitleri maskeleye

MOV TUS, AL

g)



h) Sırasıyla her bir satırda yakılmas gereken değer belirlenip sadece 0. satır sürülecek, bir sonraki adımda ise 2. satır yakılacak değerler 0A ualarında düsturulup sadece 2. satır sürülecek, benzer şekilde 3. ve 4. satırlar ~~iamde~~ durum tekrar edilecek. Bu döngü baştan itibaren tekrarlanacak. Satır satır değil her seferinde tek bir LED sönsülse $16 \times 5ms = 80ms \Rightarrow 12,5 Hz$. LED matrix saniyede 125 kez güncellenir göz bunu algılayır. Satır satır sürmede $4 \text{ satır} \times 5ms = 20ms \Rightarrow 50 Hz$, göz bunu algılayamaz.

tüm satırları sürmek için :

L3: MOV BH, 10H
XOR SI, SI
MOV CX, 4

L2: PUSH CX
MOV CX, 4

L1: MOV BL, LEDSESI
RCL BL, 1
RCL AL, 1
INC SI
LOOP L1

LED matrix'in aynı satırındaki LED'lerin aynı değeri AL'in düşük anlamlı 4 bitinde

OR AL, BH
ROL BH, 1
OUT 1AH, AL
CALL DELAY_1MS
POP CX
LOOP L2
JMP L3

seçilecek satırın belirlenmesi

LED Matrix'in 1 satır iami sürülmesi

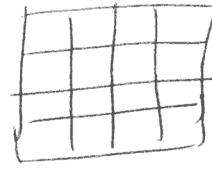
i) TUS 5: XOR LEDS[1], 01H
XOR LEDS[4], 01H
XOR LEDS[5], 01H
XOR LEDS[6], 01H
XOR LEDS[9], 01H
JMP CIKIS

h)



başlangıç durumu

ortadaki
tuşa basıldı →



son durum
tüm LED'ler söndürülmüştür.