

İsim :	2018/1 BLM3031 Mikroişlemci Sistemleri Vize - 21 Kasım 2018 Süre: 75 dk	S1 (40p)	S2 (60p)	Σ (100p)
No :				
İmza :				

**NOT1 : Tüm sorular için izole hafıza haritalama (isolated mapping) kullanın.**

**NOT2 : Tüm  $\mu P$  uçlarının uygun şekilde ayrıştırılmış ve tutulmuş olduğunu varsayın.**

**Soru 1) Aşağıdaki boşlukları uygun şekilde doldurun (40p)**

- a) 8086 veri yolu genişliği 16 bittir. (2p)
- b) 8086 adres yolu genişliği 20 bittir. (2p)
- c) 8086 için uygun hafıza arayüzü 512 × 16 kapasitesindedir. (2p)
- d) 8086 I/O uzayı 64K byte boyutundadır. (2p)
- e) 8086 hafıza uzayı 1M byte boyutundadır. (2p)
- f)  $M\overline{IO}$  ucu sadece izole I/O haritalamada adres çözümleme için kullanılır. (3p)
- g) 8088 için uygun hafıza arayüzü 1M×8 olduğuna göre, 8088 adres yolu genişliği 20 bittir. (2p)
- h) 80486 için uygun hafıza arayüzü 1G×32 olduğuna göre, 80486 adres yolu genişliği 32 bittir. (2p)
- i) 8086'da  $AD_i$  uçlarından adres bilgisi ayrıştırılırken ALE ucu kullanılır. (3p)
- j) 8086'da  $AD_i$  uçlarından veri bilgisi ayrıştırılırken DEN ve  $DT\overline{R}$  uçları kullanılır. (3p)
- k) 8086, veri ve program hafızalarının **aynı** veri yolunu kullanması açısından von Neumann mimarisindedir. (2p)
- l) 8086, anlamlı byte'ın hafızada saklanma yeri açısından little endian mimarisindedir. (2p)
- m) 8086, komut seti yapısı itibarıyla CISC mimarisindedir. (2p)
- n) Adres çözümleme devresi belirli bir zamanda sadece bir çevre birimin aktif olmasını sağlayarak, farklı entegrelerin aynı anda, aynı yola (bus) farklı veri yazmasını engeller. (3p)
- o) Mikroişlemcide aritmetik, lojik işlem sonuçlarını saklayan yazmaca akümülatör (AX) denir. (2p)
- p) 8086'da adres ve veri bilgisinin aynı uçlarda farklı zamanlarda bulunması zaman çoğullama olarak adlandırılır. (3p)
- q) 8255 mod 1'de input yönlü ayarlanmışsa port uçlarındaki verinin 8255'e otomatik olarak alınmasından sorumlu anlaşma (handshaking) uçları  $\overline{STB}$  ve IBF dir. (3p)

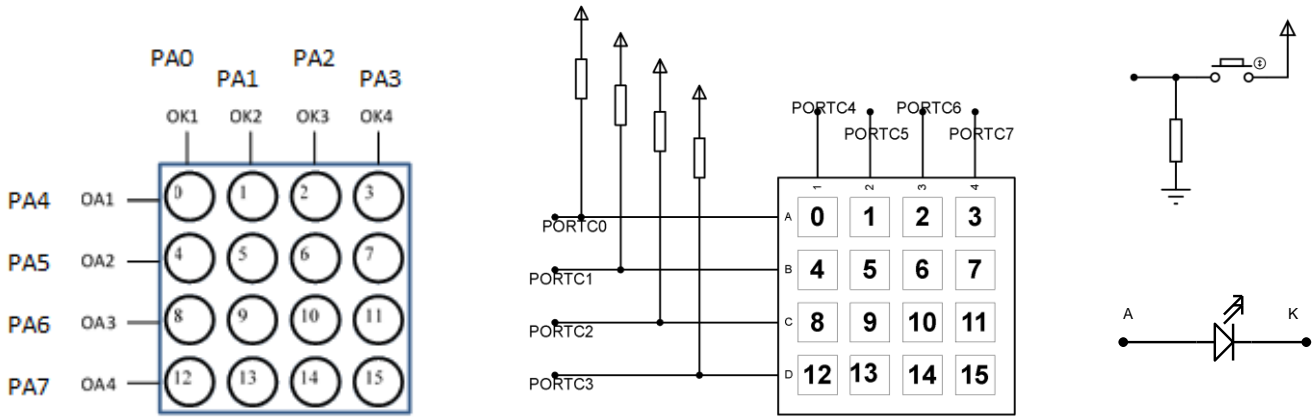
**Soru 2)** Game of Life oyununun donanımsal olarak gerçekleştirilmesi istenmektedir. İlk durum oluşturulup oyun hızı belirlendikten sonra hücresel bir otomat uygulaması olan Game of Life oyununda adım adım sonraki nesilleri oluşturulur.

Game of Life oyunu aşağıdaki kurallara göre çalışmaktadır:

- Bir canlı hücrenin, iki'den daha az canlı komşusu varsa "yalnızlık nedeniyle" ölür
- Bir canlı hücrenin, üç'ten daha fazla canlı komşusu varsa "kalabalıklaşma nedeniyle" ölür
- Bir canlı hücrenin, iki ya da üç canlı komşusu varsa değişmeden bir sonraki nesile kalır
- Bir ölü hücrenin tam olarak üç canlı komşusu varsa canlanır.

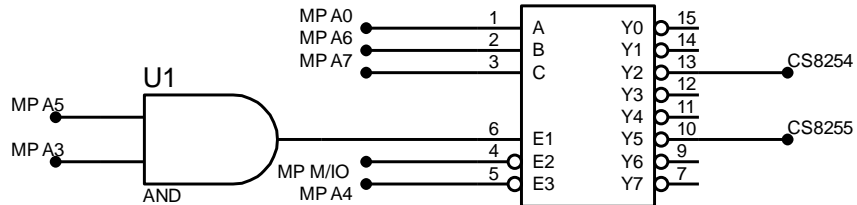
Bu oyun için 4×4 LED matris, 4×4 tuş tarama sistemi, 4 adet buton, 8255 ve 8254 kullanılacaktır. Oyun 4×4 boyutundaki grid yapısında bir düzende oynanır. Oyuncu 4×4 tuş tarama sisteminde herhangi bir tuşa basarak, bastığı numaralı LED'leri toggle (yanık söndürebilir, söndükse yakabilir) ettirir. Daha sonra HIZ+ ve HIZ- butonları ile oyun hızını ayarlayarak, BASLA tuşu ile hücresel otomat çalıştırır. Her bir yeni nesil öncesi HIZ+, HIZ- tuşları ile belirlenmiş bir süre beklenerek nesiller hesaplanır. Oyuncu isterse SIFIRLA tuşuna basarak oyunu durdurup, 4×4 LED matriste tüm LED'leri söndürebilir.

4×4 LED matris yapısı şekil ile verilmiştir. Aynı satırdaki LED'lerin anot uçları birleştirilerek dışarı verilmektedir (OA uçları), benzer şekilde aynı sütundaki LED'lerin katot uçları birleştirilerek dışarı verilmektedir (OK uçları). Bir LED'e ilişkin uç tanımları ilgili şekil ile verilmiştir. Basıldığında ilgili satır ve sütunu kısa devre yapan butonlar içeren, 4×4 tuş tarama sistemi konfigürasyonu şekil ile verilmiştir. Kullanılan ayırık butonlar (BASLA, HIZ+, HIZ-, SIFIRLA) için ortak konfigürasyon şeklinde verilmiştir. Bunlara göre:



- a) A9H adresinden itibaren ardışık tek adreslere 8255'i, 68H adresinden itibaren ardışık çift adreslere ise 8254'ü yerleştirebilmek için uygun adres çözümlenme devresini 1 adet 3×8 dekodör ve gerekli en az sayıda basit lojik kapılar kullanarak gerçekleştiriniz. (10p)

- A9 → 1 0 1 0 1 0 0 1  
 AB → 1 0 1 0 1 0 1 1  
 AD → 1 0 1 0 1 1 0 1  
 AF → 1 0 1 0 1 1 1 1  
 68 → 0 1 1 0 1 0 0 0  
 6A → 0 1 1 0 1 0 1 0  
 6C → 0 1 1 0 1 1 0 0  
 6E → 0 1 1 0 1 1 1 0



A7, A6, A0 seçim uçları, A5=1, A4=0, A3=1,  $M\overline{IO}=0$ , A2 ve A1 8255 ve 8254 adres uçlarına

- b) Verilen adres değerlerine göre mikroişlemci ve adres çözümlenme devresinden 8255 ile 8254'e yapılması gereken bağlantıları çizerek gösterin. (4p)

8255	8254
D0-7 → MP D8-15	D0-7 → MP D0-15
A0 → MP A1	A0 → MP A1
A1 → MP A2	A1 → MP A2
RD → MP RD	RD → MP RD
WR → MP WR	WR → MP WR
CS → CS8255	CS → CS8254

- c) 4×4 LED matris 8255 PORTA'ya (OK1-4→PORTA0-3, OA1-4→PORTA4-7), 4 adet ayrı buton PORTB'ye (BASLA→PORTB0, HIZ+→PORTB1, HIZ-→PORTB2, SIFIRLA→PORTB3), 4×4 tuş tarama sistemi ise PORTC'ye (A,B,C,D→PORTC0-3, 1,2,3,4→PORTC4-7) bağlanacaktır. 8255 için uygun bir kontrol kelimesini belirleyin. **(4p)**

PORTA →output, PORTB→ input, PORTCL →input, PORTCU→output, 8255→ MOD0  
1 0 0 0 0 0 1 1

- d) 8255'e belirlemiş olduğunuz kontrol kelimesini aktarmak için gerekli assembly kodunu yazın. **(3p)**

MOV AL, 83H ; PORTA →output, PORTB→ input, PORTCL →input, PORTCH→output, 8255→ MOD0  
OUT 0AFH, AL

- e) 8254, CNTR0'ın MOD0'da 100ms ile 1sn arasında zamanlayıcı olarak kullanılması istenmektedir (HIZ+ veya HIZ- tuşlarına basıldıkça oyunda adımlar arası bekleme süresi 100ms ile 1sn arasında 100ms'lik adımlarla değişecektir (1kHz'lik saat işareti mevcutken), varsayılan bekleme süresi ise 500ms'dir). Bunun için uygun 8254 kontrol kelimesini belirleyin. **(3p)**

0011 0000 → CNTR0 mod0'da LSB+MSB, binary, 100ms-1s arası 100ms adımlarla sayabilmek için sayma değeri 100-1000 arasında olur

- f) Yukarıda bahsedilen durum ile 1kHz'lik saat işareti mevcutken, 8254 için uygun kontrol kelimesini ve ilk sayma değerlerini (500ms'lik zamanlama için) aktaran assembly kodunu yazın. **(4p)**

MOV AL, 0011 0000B ; CNTR0 mod0'da LSB+MSB, binary  
OUT 6EH, AL  
MOV AX, 500 ;varsayılan bekleme suresi  
OUT 68H, AL  
MOV AL, AH  
OUT 68H, AL

- g) BASLA isimli butona basılıp basılmadığını tespit eden assmebly kodunu yazınız. **(3p)**

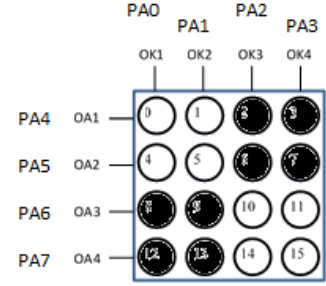
TEKRAR:  
IN AL, 0ABH  
TEST AL, 01H ; BASLA tuşu PORTB 0'da 0000 0001B ile test edilir, tuş basılı değilse sonuç 0 olur  
JZ TEKRAR  
BASLABASILİ:

- h) SIFIRLA tuşuna basıldığını tespit eden ve tuş basılmışsa 4×4 LED matrisi bir seferde temizleyen assembly kodunu yazın. **(5p)**

TEKRAR:  
IN AL, 0ABH  
TEST AL, 08H ; SIFIRLA tuşu PORTB 3'te 0000 1000B ile test edilir, tuş basılı değilse sonuç 0 olur  
JZ TEKRAR  
SIFIRLABASILİ:  
MOV AL, 0FFH ; PORTA'dan FF göndererek tüm OK uçları 1 yapılır, sonuçta tüm LED'ler söner  
OUT 0A9H, AL

- i) Yanda verilen durumu (varsayılan ilk durum) 4×4 LED matriste oluşturabilmek için gerekli assembly kodunu yazınız (yanan LED'ler siyah ile gösterilmiştir). (6p)

```
TEKRAR: MOV AL, 0CCH
        OUT 0A9H, AL
        MOV AL, 33H
        OUT 0A9H, AL
        JMP TEKRAR
```



- j) 4×4 tuş tarama sisteminde 5 ve 9 numaralı tuşlarından hangisine basıldığını tespit eden assembly kodunu yazın. (6p)

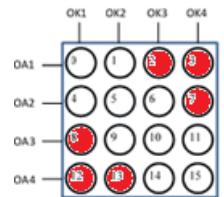
```
TEKRAR: MOV AL, 0DFH; PORTC5 → 0, PORTC4,6,7 → 1
        OUT 0ADH, AL; keypadin sadece 2 numaralı ucu 0 ile sürülüyor
        IN AL, ADH
        TEST AL, 02H; PORTC5 aktifken, B numaralı uç PORTC1'de 0 olduyorsa 5'e basılmıştır
        JZ TUS5
        TEST AL, 04H; PORTC5 aktifken, C numaralı uç PORTC2'de 0 olduyorsa 9'a basılmıştır
        JZ TUS9
        JMP TEKRAR
```

- k) 8254'ün zamanlama miktarını, HIZ+ butonuna basıldığında 100ms arttıran, HIZ- butonuna basıldığında ise 100ms azaltan assembly kodunu yazın (zamanlama miktarının 100ms ile 1s arasında kalması gerekmektedir). (6p)

```
TEKRAR: IN AL, 0ABH; PORTB oku
        TEST AL, 02H; HIZ+ için test
        JZ HIZARTI
        TEST AL, 04H; HIZ- için test
        JZ HIZEKSI
        HIZARTI: ADD COUNT8254, 100
        CMP COUNT8254, 1000
        JBE DEVAM
        MOV COUNT8254, 1000
        JMP DEVAM
        HIZEKSI: DEC COUNT8254, 100
        CMP COUNT8254, 100
        JAE DEVAM
        MOV COUNT8254, 100
        DEVAM: MOV AX, COUNT8254
        OUT 68H, AL
        MOV AL, AH
        OUT 68H, AL
        JMP TEKRAR
```

- l) 8254'ün zamanlamasının bittiğini tespit eden assembly kodunu yazınız. (4p)

```
TEKRAR: OUT 6EH, 1110010B; CNTR0 status latch için readback komutu
        IN AL, 68H; CNTR0 status okuma
        ROL AL, 1; CNTR0 output status 7'de, ROL ile carry'e aktarılır
        JNC TEKRAR; MOD0 'da sayma devam ederken output 0'da
        CNTR0TIMEOUT:
```



- m) i şıkındaki başlangıç konumundan bir sonraki nesli, verilen boş şekle çizerek gösterin. (2p)