

1	2	3	4	Toplam
2	3, 4	2, 3	3, 4	PÖÇ

1. Briefly explain how priorities are resolved in software by a microprocessor to identify and service highest priority interrupt source among many interrupt generating sources. // Kesme üreten kaynaklardan en yüksek öncelikli olanını mikroişlemcinin yazılımla nasıl belirlediği ve servis ettiğini kısaca açıklayınız.

Yazılımla kesme önceliklerinin belirlenmesi için kesme servis programında yüksek öncelikten düşük önceliğe doğru tüm kesme girişlerinin kesme üretip üretmediği sırayla denetlenmelidir. Bu sayede, iki farklı öncelikteki kaynaktan gelen kesme işaretleri nedeniyle başlatılan kesme servis programının en başında, en yüksek öncelikliden başlayarak denetlendiği için önce daha yüksek öncelikli olan istek değerlendirilebilmektedir.

2. a. A parking lot owner wants to display the number of available parking positions in a digital display at the entrance in order not to waste drivers' time to find a place to park before they decide to use the parking lot. Write two interrupt service routines, one interrupt generated by exiting cars to increment available parking position and one interrupt generated by entering cars to decrement available parking position, to update available positions and a program (main function) to display available parking positions in C. You may assume that a digital display indicating available parking positions can be updated by using outport (address, unsigned int data) C function to output an unsigned integer to address DISPLAY. State all other assumptions you have to make. // Bir otopark yeri sahibi sürücülerin park yeri içinde boş bir park yeri bulmak için vakit kaybetmelerine engel olmak için sürücülere otoparkı kullanmaya karar vermeden önce boş park yeri sayısını girişteki sayısal bir göstergede göstermek istemektedir. Otoparkta bulunan müsait park yeri sayısını, çıkan araçların ürettiği kesme ile boş yer sayısını artırarak ve giren araçların ürettiği kesme ile boş yer sayısını azaltarak güncelleyen iki kesme servis programını ve gösterge güncelleyen ana programı (main fonksiyonunu) C dilinde yazınız. Müsait park yeri pozisyon sayısını belirten sayısal bir göstergenin, outport (adres, unsigned int veri) C fonksiyonu ile güncellenebildiğini varsayabilirsiniz. Tasarlarken yapmak zorunda kaldığınız diğer tüm varsayımları belirtiniz.

```

void _interrupt CarExit(void)          void _interrupt CarEntry(void)          void main(void)
{
    if (boş_yer < 50)
        boş_yer ++;
    return;
}

{
    if (boş_yer > 0)
        boş_yer --;
    return;
}

{
    int boş_yer=50;
    while (true)
        outport(GÖSTERGE, boş_yer);
}

```

- b. Code interrupt service routines and main program in Assembly. // Kesme servis programları ve ana programı Assembly ile kodlayınız.

```

CarExit :    cmp boş_yer, #50          CarEntry :    cmp boş_yer, #0          mov boş_yer, #50
             jae Done                  jbe Done                  Döngü :    out GÖSTERGE, boş_yer
             inc boş_yer                dec boş_yer                  j Döngü
Done :       ret                      Done :       ret

```

3. a. Even though magnetic discs suffer from seek time and limited data transfer rate, they are still developed and used as lower cost alternative to SSDs. Suggest an Input/Output technique in order not to slow down microprocessors considerably. // Erişim süresi ve sınırlı veri aktarım hızları nedeniyle manyetik diskler sorun yaşasa da SSD'lere daha düşük maliyetli bir alternatif olarak hala geliştirilmekte ve kullanılmaktadır. Mikroşlemcileri kayda değer oranda yavaşlatmaması için bir Giriş/Çıkış tekniği öneriniz.

Manyetik diskler, tampon olarak kullandıkları statik RAM'den blok veri aktarımı yaptıkları için **Doğrudan Bellek Erişimi** kullanılmalıdır.

- b. List the sequence of events before and during a 32kB data block transfer from disc buffer to the main memory address 3A4B. // Disk tamponundan ana belleğin 3A4B adresine 32kB bir veri bloğu aktarımı öncesi ve aktarım süresince gerçekleşen olayları listeleyiniz.

Öncesinde Program : Belleğe aktarımın yapılacağı başlangıç adresi olan 3A4B₁₆ DMA Denetleyici **Adres Kaydedicisine** yükler

DMA Denetleyici **Aktarım Sayacı**na 32768 değeri yükler

Okuma/Yazma yönünü belirten DMA **Yön Kaydedicisine** dış cihazdan okuma yapılacağına dair bilgi yükler

DMA Denetleyici : Yol İstek (**Bus Request**, BR) işaretini bekler, BR aldığı anda Yol Kullanımına İzin Verildi (**Bus Granted**, BG) işaretini çıkarır

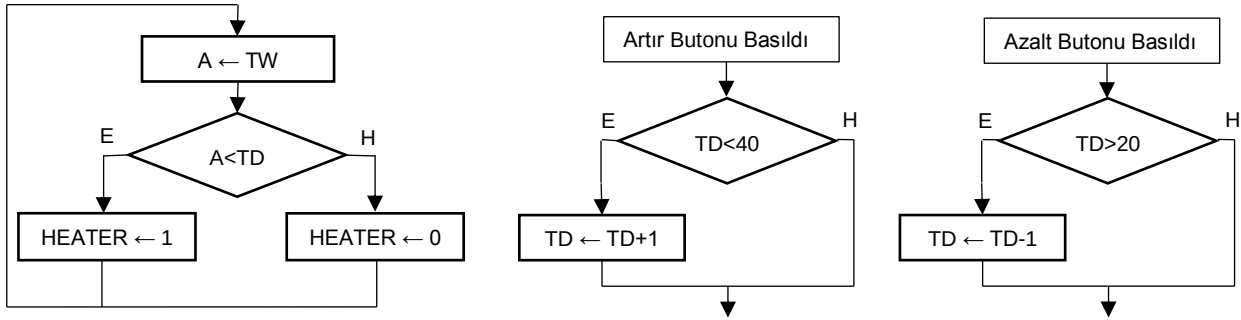
Aktarım süresince : Adres Kaydedici değeri adres yoluna aktarılır,

Diskten belleğin DMA Adres Kaydedici ile gösterilen adresine bir byte aktarım (cihazdan okuma, belleğe yazma) yapılır

DMA Adres Kaydedici değeri bir artırılır ve Aktarım Sayacı bir azaltılır ve aktarım sayacı 0'a ulaşana kadar bu işlemler tekrarlanır

4. From Wikipedia, the free encyclopedia: A thermostat exerts control by switching heating or cooling devices on or off, or by regulating the flow of a heat transfer fluid as needed, to maintain the correct temperature. A thermostat can often be the main control unit for a heating or cooling system, in applications ranging from ambient air control, to such as automotive coolant control. Thermostats are used in any device or system that heats or cools to a setpoint temperature, examples include building heating, central heating, air conditioners, as well as kitchen equipment including ovens and refrigerators and medical and scientific incubators. Design a microprocessor based user programmable thermostat to maintain a desired water temperature (TD) of an electric water heater by turning a heater resistance on or off as required. Heating resistance current is controlled by an electronic circuit at symbolic address HEATER. Programmable thermostat should output a value of 1 to HEATER which turns on heater if water temperature read from temperature sensor (TW) is less than TD and should output a 0 to turn the heater off if TW is greater than or equal to TD. There are two push buttons on the water heater to increment or decrement desired water temperature by pushing corresponding button. State all other assumptions you have to make in order to design the user programmable thermostat. // Vikipedi, özgür ansiklopedi'den: Bir termostat, doğru sıcaklığı sağlamak için ısıtma veya soğutma aygıtlarını açar veya kapatır veya ısı transfer sıvı akışını regüle eder. Termostat, ortam hava denetiminden otomobil motor soğutma suyu sıcaklığına kadar olan ısıtma ve soğutma sistemlerinde ana denetim birimi olabilir. Termostatlar, herhangi bir aygıt ve sistemde, örneğin bina ısıtma, merkezi ısıtma, klimalar ve hatta fırın, buzdolabı gibi mutfak araçlarında ve bilimsel inkubatörlerde kullanılmaktadır. Gereklikçe bir ısıtıcı rezistansı açarak veya kapatarak elektrikli bir su ısıtıcısında istenen su sıcaklığını (TD) sağlayacak kullanıcı programlanabilir mikroişlemcili termostatu tasarlayınız. Isıtıcı rezistansının akımı, HEATER sembolik adresindeki bir elektronik devre tarafından denetlenmektedir. Programlanabilir termostat, sıcaklık algılayıcıdan okuyacağı su sıcaklığı (TW) TD'den küçükse 1 değerini HEATER adresine çıkarak ısıtıcıyı açmalı ve TW TD'den büyük veya eşitse 0 çıkarak ısıtıcıyı kapatmalıdır. Su ısıtıcısı üzerinde bulunan iki basılabilir anahtardan ilgili olanına basılarak istenen su sıcaklığı artırılabilir ve azaltılabilir. Tasarlarken yapmak zorunda kaldığınız diğer tüm varsayımları belirtmeniz gerekmektedir.

- a. Draw the flowchart(s) of the programmable thermostat algorithm. // Programlanabilir termostat algoritmasının akış çizge(ler)ini çiziniz.



- b. Code programmable thermostat algorithm in C. // Programlanabilir termostat algoritmasını C dili ile kodlayınız.

```

void main (void)
{
    (while (true)
    {
        A = inport(TW); // Su sıcaklığını oku
        if (A < TD)
            output(HEATER, 1); // Isıtıcıyı aç
        else
            output(HEATER, 0); // Isıtıcıyı kapa
    }
}

void _interrupt artır (void)
{
    if (TD < 40)
        TD++;
    return;
}

void _interrupt azalt (void)
{
    if (TD > 20)
        TD--;
    return;
}
  
```

- c. Code programmable thermostat algorithm in Assembly language. // Programlanabilir termostat algoritmasını Assembly dili ile kodlayınız.

```

döngü :   in A, TW           // Su sıcaklığını oku           artır :   cmp TD, #40           azalt :   cmp TD, #20
          cmp A, TD          // İstenenle (TD) kıyasla       jae Done
          jb isi_kapat       // su sıcaksa ısıtıcıyı kapat   inc TD
          mov A, #1          // açma işareti (1)           Done :   ret
          out HEATER, A      // HEATER'a çıkar
isi_kapat : mov A, #0        // kapama işareti (0)
          out HEATER, A      // HEATER'a çıkar
          j döngü
  
```