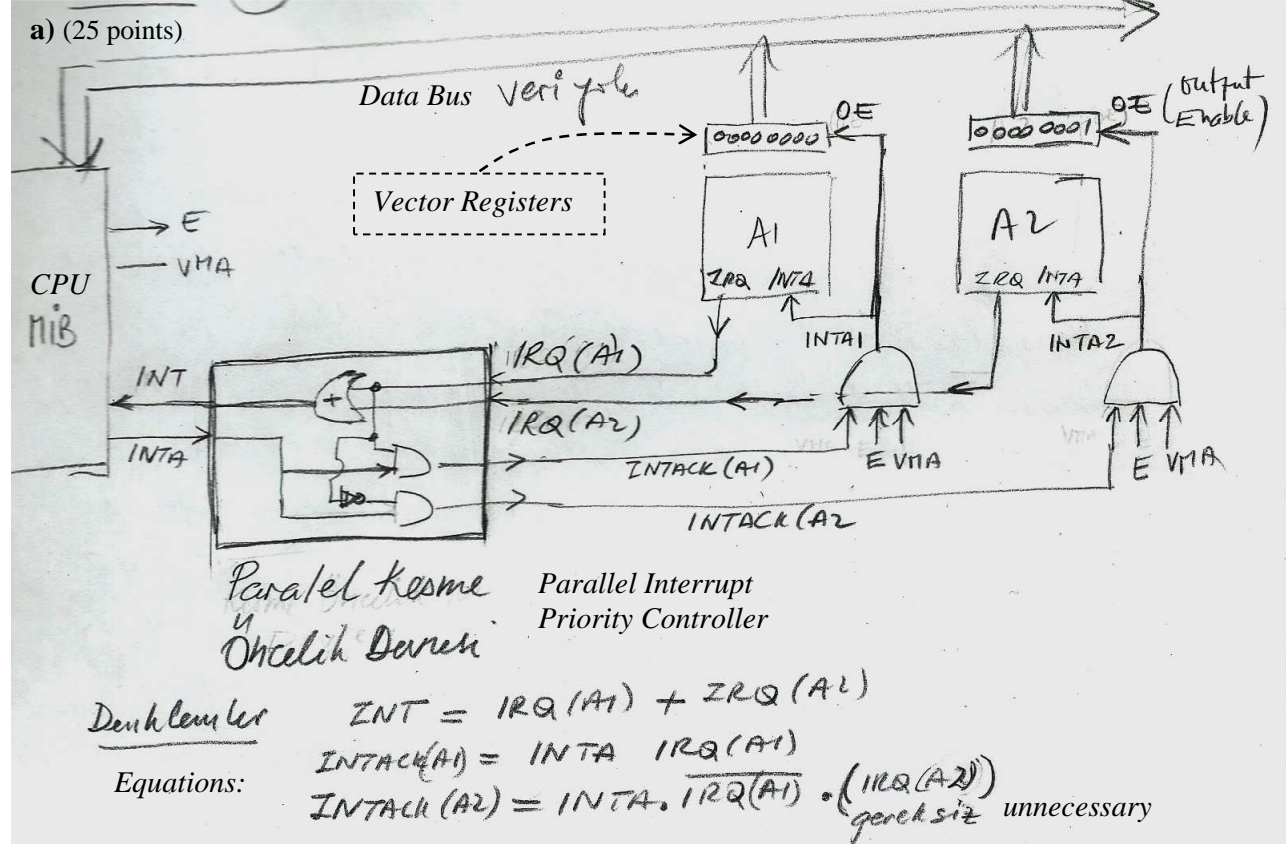




COMPUTER ARCHITECTURE 2nd MIDTERM
BİLGİSAYAR MİMARİSİ 2nci YILIÇI SINAVI
21.04.2016

SOLUTION / ÇÖZÜM 1: (35 Points / Puan)



b) (10 points)

i) Software poll

After the interrupt cycle the CPU jumps to the interrupt service routine (ISR).

All interrupt requests are disabled.

In the ISR the program reads (checks) the status flags of the interrupt sources according to a predefined order.

If the interrupt status flag of a device is set, the source is identified and the program calls the subroutine of this device.

In the subroutine of the device the status flag is cleared and the interrupt request is removed.

After the subroutine of the device the program returns to the ISR.

After returning from the ISR interrupts are enabled.

If there is a pending interrupt request the CPU will jump to the ISR and the same operations will be repeated from another device.

ii) Hardware poll mechanism is implemented as a daisy chain.

When the CPU accepts an interrupt request it activates the interrupt acknowledge output (INTA).

The INTA signal is received by the first link of the chain. If the device connected to the first link has a request, then the INTA signal is sent to this device and its interrupt vector is put to the data bus.

The next link in the chain cannot get the INTA signal.

The interrupt vector identifies the interrupt source.

The CPU finds the start of the ISR using the interrupt vector of the device and jumps to the ISR of the related device.

After getting the INTA the requesting device removes its request.

If the first device does not have request, then the INTA signal is sent to the next link in the chain.

1) Yazılım yoklaması: Kesme isteği etkinleşince kesme çevrimine girilir ve yoklama yazılımının başlangıç adresine dallanılır. (Genelde bellek alanının en üst adreslerinden biri). Yoklama yazılımı olası kesme kaynaklarının ilgili bayraklarını belirlenmiş bir sıraya göre birer birer okur. Bu okuma sırasında ilk rastlanan etkin bayrağa hizmet verilir yani o bayrağa ilişkin hizmet programına dallanılır ve bu hizmet programı bitene kadar başka kesme kabul edilmez. Hizmet programı I/O işlemini bitirirğinde kendini başka kesme bayrağını sıfırlar ve yoklama yazılımından çıkılır. O sırada başka bir kesme isteği oluşmuşsa, yine den kesme çevrimine girilir ve yoklama programına dallanılır. Öncelik yoklama sırası ile belirlenir

2.1) Donanım yoklaması (Hardware poll)

Papatya zinciri (Daisy Chain) adı verilen donanım ile yapılır. Kesme HIB'e eriştiğinde kesme çevrimine girilir ve kesme anlaşıldı işaretini etkinleştirilir (INTA). Bu işaret papatya zincirinin ilk halkasına iletilir (P2) giriş. Her halkanın PI girişi ve PO çıkışı vardır. Eğer birinci halkada kesme isteği yoksa, işaret ikinci halkaya, orada da kesme isteği yok sonraki halkaya iletilir. Kesme isteğine rastlanan ilk halkadan sonra işaret iletilmez. O halkaya bağlı olan cihaza vektörünü yola koymak ve kesme isteğini kaldırmanın için bir işaret gönderilir. Yola konan vektör yardımı ile kesme hizmet programının başlangıç adresi belirlenir (Tablo yardımıyla) ve bu programa dallanılır. Program sona erene kadar yeni bir kesme kabul edilmez. Program sona erince kesilen programa döner. Bu sırada başka bir kesme oluşmazsa veya beklemeleri ise (pending interrupt) yine den kesme çevrimine girilir ve aynı işlemler tekrarlanır. Papatya zincirinde halkaların yoklama sırası öncelik sıradır

SOLUTION / ÇÖZÜM 2: (35 Points / Puan)

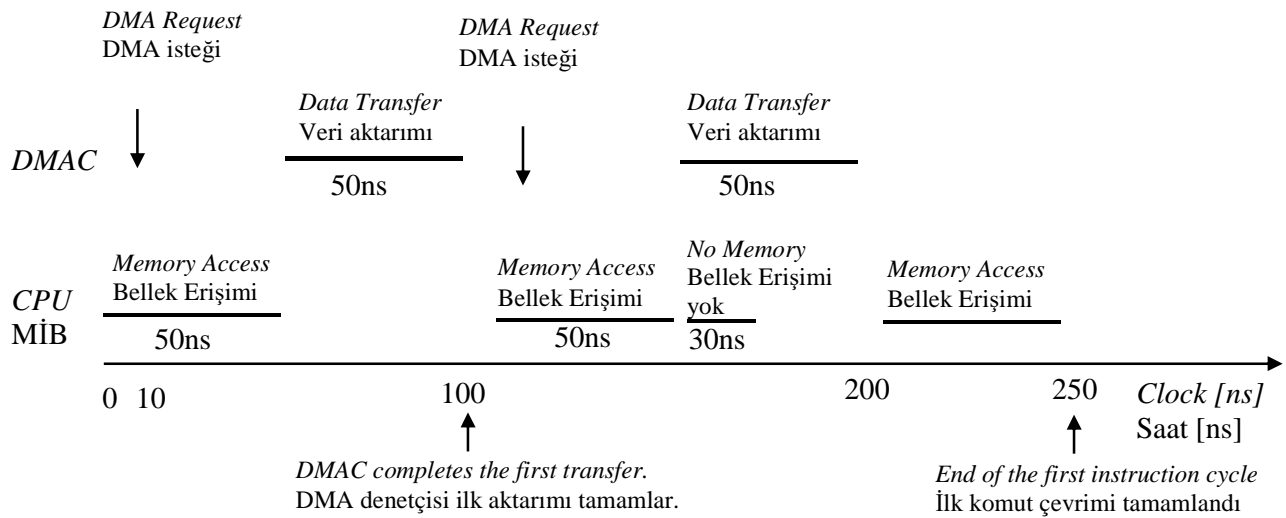
a) (10 points / 10 puan)

CPU completes the current bus cycle and isolates itself from the system bus. The DMAC transfers the first data. Since the type of the DMAC is **fly-by** (Implicit) data is transferred in 50 ns.

$$\text{Clock} = 50 + 50 = 100\text{ns}$$

MİB o andaki yol çevrimini tamamlar ve kendini sistem yolundan yalıtır. DMA denetçisi ilk sekizliyi aktarır. DMA denetçisi örtülü olarak çalıştığı için veri 50ns’de aktarılır.

$$\text{Saat} = 50 + 50 = 110\text{ns}$$



b) (10 points / 10 puan)

For the explanation the the figure.

$$\text{Clock} = 50 + 50 + 50 + 50 + 50 = 250\text{ns}$$

Açıklama için yukarıdaki şekle bakınız.

$$\text{Saat} = 50 + 50 + 50 + 50 + 50 = 250\text{ns}$$

c) (15 points / 15 puan)

a) A **flow-through** (explicit) DMAC first reads a word from the I/O interface then writes it to the memory. Therefore, the transfer of one word takes $2 \times 50 = 100$ ns.

$$\text{Clock} = 50 + 2 \cdot 50 = 150\text{ns}$$

Açık DMAC önce bir sözcüğü G/Ç arabiririminden okur sonra belleğe yazar. Bu nedenle bir sekizlinin aktarımı $2 \times 50 = 100$ ns sürer.

$$\text{Saat} = 50 + 2 \cdot 50 = 150\text{ns}$$

c) b)

In **burst** (block) mode the DMAC transfers all words until the word counter is zero, when it takes the bus from the CPU. 10 words: $10 \times 100 = 1000$ ns.

$$\text{Clock} = 50 + 1000 + 30 + 50 + 50 = 1180\text{ns}$$

Blok aktarım yönteminde çalışan DMAC yolu MİB’ten aldıktan sonar sözcük sayacı sıfır oluncaya kadar tüm sözcükleri aktarır. 10 sözcük: $10 \times 100 = 1000$ ns.

$$\text{Saat} = 50 + 1000 + 30 + 50 + 50 = 1180\text{ns}$$

SOLUTION / ÇÖZÜM 3: (30 Points / Puan)

a)

i) (2 Points/ Puan)

One Parity disk. Total = $N + 1 = 3$ disks.

Bir eşlik diski gereklidir. Toplam = $N + 1 = 3$ disk.

ii) (6 Points/ Puan)

Remember; the disks in RAID 3 are synchronized and strip size is small. All heads are located at the same position.

Shortest: Two words are at the same location (same track/sector) in different disks. They can be read in parallel.

Time = t_a

Longest: Two words are at different locations (different track/sector) in different disks. They can be read sequentially, because disks are synchronized.

Time = $t_a + t_a = 2 \cdot t_a$

Hatırlatma; RAID 3'te diskler senkronizasyon ve şerit boyu küçüktür. Tüm disklerin kafaları aynı pozisyona konumlanırlar.

En kısa: İki sözcük farklı disklerin aynı fiziksel adreslerindendir (aynı iz/sektör). Bu sözcükler paralel olarak okunabilir.

Zaman = t_a

En uzun: İki sözcük farklı disklerin farklı fiziksel adreslerindendir (farklı iz/sektör). Bu sözcükler ancak peş peşe okunabilir, çünkü diskler senkron çalışırlar.

Zaman = $t_a + t_a = 2 \cdot t_a$

iii) (8 Points/ Puan)

Shortest: Two words are at the same location (same track/sector) in different disks. They can be written in parallel.

Also the parity can be calculated and written at the same time, because disks are synchronized. The heads are at the same and correct position. There is no need to spend seek and rotation time for the parity disk.

Time = t_a

Longest: Two words are at different locations (different track/sector) in different disks. They can be written sequentially. To write the first word into disk 1, the word at same location in disk 2 is read, the parity is calculated and the parity is written into parity disk. Writing disk, reading the word at same location in disk 2 and parity calculation can be performed in parallel, hence in t_a . To write the parity to parity disk, we need an extra t_a . Therefore, time to write 1 word is $2 \cdot t_a$; consequently, time to write 2 words is $4 \cdot t_a$.

Time = $2 \cdot t_a + 2 \cdot t_a = 4 \cdot t_a$

Another solution for the longest time: Actually, in systems, where parity can be calculated very fast, the operations writing the first word in disk 1, reading the word at same location in disk 2 and writing the parity into parity disk can be performed in parallel. Because the heads are at the same and correct position, there is no need to spend seek and rotation time for the parity disk. In this case, time to write 1 word is t_a , and time to write 2 words is $2 \cdot t_a$.

Time = $t_a + t_a = 2 \cdot t_a$

If proper explanations are given (not only the numbers), then both solutions for "iii) longest" are accepted.

En kısa: İki sözcük farklı disklerin aynı fiziksel adreslerindendir (aynı iz/sektör). Bu sözcükler paralel olarak yazılabilir. Eşlik biti de aynı anda hesaplanarak eşlik diskine yazılabilir çünkü bütün diskler senkron çalışır. Tüm kafalar aynı ve doğru konumdadır. Eşlik diski için ayrıca konumlandırma ve dönüş zamanı harcamaya gerek olmaz.

Zaman = t_a .

En uzun: İki sözcük farklı disklerin farklı fiziksel adreslerindendir (farklı iz/sektör). Bu sözcükler ancak peş peşe yazılabilir. Birinci sözcüğü diske 1'e yazarken disk 2'nin aynı konumundaki sözcük okunur, eşlik hesaplanır ve eşlik bilgisi eşlik diskine yazılır. Disk 1'e yazarken, disk 2'nin aynı konumundaki sözcüğü okumak ve eşliği hesaplamak paralel olarak t_a süresinde yapılabilir. Eşlik bilgisini eşlik diskine yazmak için ek bir t_a süresine gerek olur. Bu nedenle 1 sözcük yazmak için $2 \cdot t_a$; 2 sözcük yazmak için ise $4 \cdot t_a$ gerekli olur.

Zaman = $2 \cdot t_a + 2 \cdot t_a = 4 \cdot t_a$

En uzun süre ile ilgili diğer bir olası çözüm: Eşlik bilgisinin hızlı hesaplanabildiği sistemlerde, birinci sözcüğü diske 1'e yama, disk 2'nin aynı konumundaki sözcüğü okum ve eşlik bilgisi eşlik diskine yazma işlemleri paralel olarak yapılabilir. Tüm kafalar aynı ve doğru konumda olduğundan eşlik diski için ayrıca konumlandırma ve dönüş zamanı harcamaya gerek olmaz. Bu durumda bir sözcük yazma süresi t_a ve 2 sözcük yazma süresi $2 \cdot t_a$ olur.

Zaman = $t_a + t_a = 2 \cdot t_a$.

Eğer uygun açıklamalar verildiyse (sadece sayılar değil) iii) en uzun süre için her iki çözüm de kabul edilmiştir.

b)

i) (2 Points/ Puan)

Mirror disks. Total = $2 \cdot N = 4$ disks.

Ayna diskleri gereklidir. Toplam = $2 \cdot N = 4$ disk.

ii) (6 Points/ Puan)

Shortest and longest durations are same. Because disks are not synchronized, if two words are in different disks, they can be read in parallel.

Time = t_a .

En kısa ve en uzun süreler aynıdır. Diskler senkron olmadığından eğer iki sözcük farklı disklerdeyse bu sözcükler paralel olarak okunabilir.

Zaman = t_a .

iii) (6 Points/ Puan)

Shortest and longest durations are same. Because disks are not synchronized, if two words are in different disks, they can be written in parallel. Their mirrors are also in different disks.

Time = t_a .

En kısa ve en uzun süreler aynıdır. Diskler senkron olmadığından eğer iki sözcük farklı disklerdeyse bu sözcükler paralel olarak yazılabilir. Bu sözcüklerin aynaları da farklı disklerde dir.

Zaman = t_a .