



COMPUTER ARCHITECTURE EXAM SOLUTIONS BİLGİSAYAR MİMARİSİ SINAV ÇÖZÜMLERİ

QUESTION 1: SORU 1:

- a) 100 instructions
100 komut:

$$T = (k + n - 1)10 = (4 + 100 - 1)10 = 1030ns$$

Without pipeline:

$$\text{İş hattı olmadan: } 100 \cdot 30 = 3000 \text{ ns}$$

$$\text{Speedup / Hızlanma: } S = \frac{3000}{1030} = 2.9$$

b)

	1	2	3	4	5
Branch	FI	DA	FO	EX	
I1		FI	DA	FO	
I2			FI	DA	
Target					FI

PC is updated
PS güncellenir

Target instruction is fetched in the 5th step instead of the 2nd step.
Penalty: 3 clock cycles.
Dallanmadan sonraki komut 2. adım yerine 5. adımda alınmış oldu.
Ceza: 3 saat periyodudur.

3 clock cycles
3 saat çevrimi

c)

Duration of 1000 instructions without branch penalty:

Dallanma cezası olmadan 1000 komut süresi:

$$T = (k + n - 1)10 = (4 + 1000 - 1)10 = 10030ns$$

70 times branch penalty:

$$70 \text{ defa dallanma cezası: } 70 \cdot 3 \cdot 10ns = 2100ns$$

Total time:

$$\text{Toplam süre: } 10030 + 2100 = 12130ns$$

$$\text{Speedup / Hızlanma: } S = \frac{30000}{12130} = 2.47$$

Another accepted solution:

Kabul edilen diğer bir çözüm:

It is assumed that at the beginning of each iteration first instruction needs k cycles to be completed.

Döngüye her girişte ilk komutun k çevrimde tamamlandığı varsayılırsa.

$$T = 30(4 + 10 - 1)10 + 70(4 + 10 - 1 + 3)10 = 3900ns + 11200ns = 15100ns$$

$$\text{Speedup / Hızlanma: } S = \frac{30000}{15100} = 1.99$$

QUESTION 2: (30 Points)**a)**

Instructions (Komutlar) / Clock cycles (Adımlar)	1	2	3	4	5	6	7	8	9	10	11
LDSU (R0)\$500,R10	I	A	D								
ADD R0,0,R11		I	A								
LDSU (R0)\$502,R11			I	A	D						
ADD R0,0,R12				I	A						
ADD R10,R12,R12					I	A					
SUB R11,1,R11						I	A				
JMPR BHI,LOOP							I	A	D		
ADD R0,R0,R10								I	A		
STL (R0)\$504,R12									I	A	D

There isn't any data conflict.
Veri bağımlılığı yoktur.

b) During the execution of the branch (JMPR), next instruction (ADD) is taken to the pipeline. Therefore, this ADD instruction is also executed in each iteration of the loop if the condition of the JMPR is true.

If this branch conflict is not solved, register R10 will be cleared in every run of the loop and the program will generate wrong result.

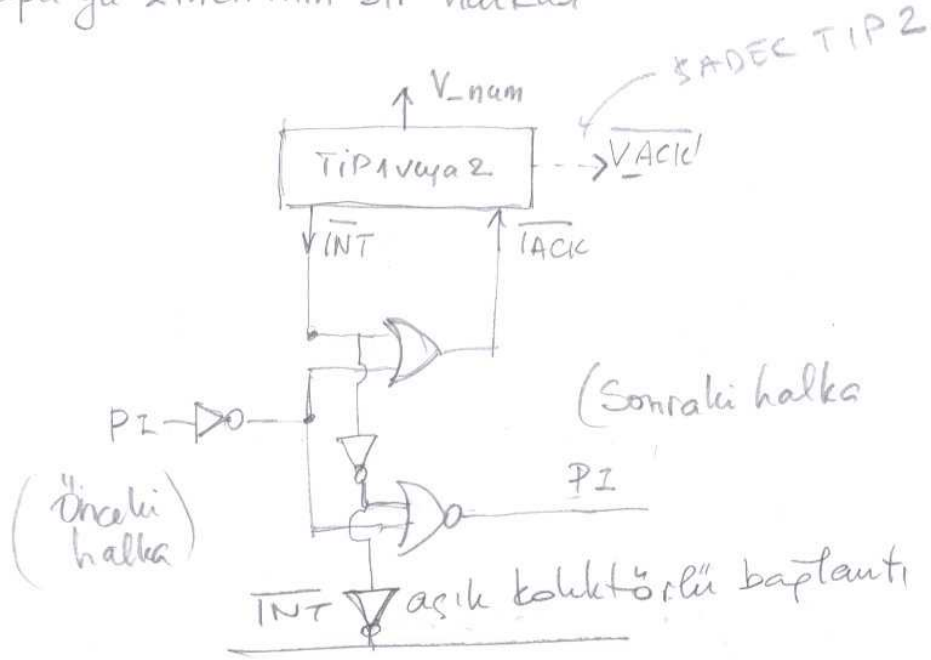
Dallanma komutu yürütülürken, iş hattına sıradaki komut (ADD) alınmaktadır. Bu nedenle koşul sağlandığı halde her döngüde bu komut da yürütülmektedir. Eğer bu sorun çözülmezse R10 saklayıcısı her döngüde sıfırlandığından program yanlış sonuç üretecektir.

c) In order to solve this problem without decreasing the performance of the pipeline, "ADD R10,R12,R12" can be moved after JMPR instruction.

Sorunu çözmek için "ADD R10,R12,R12", JMPR komutundan sonraya taşınır.

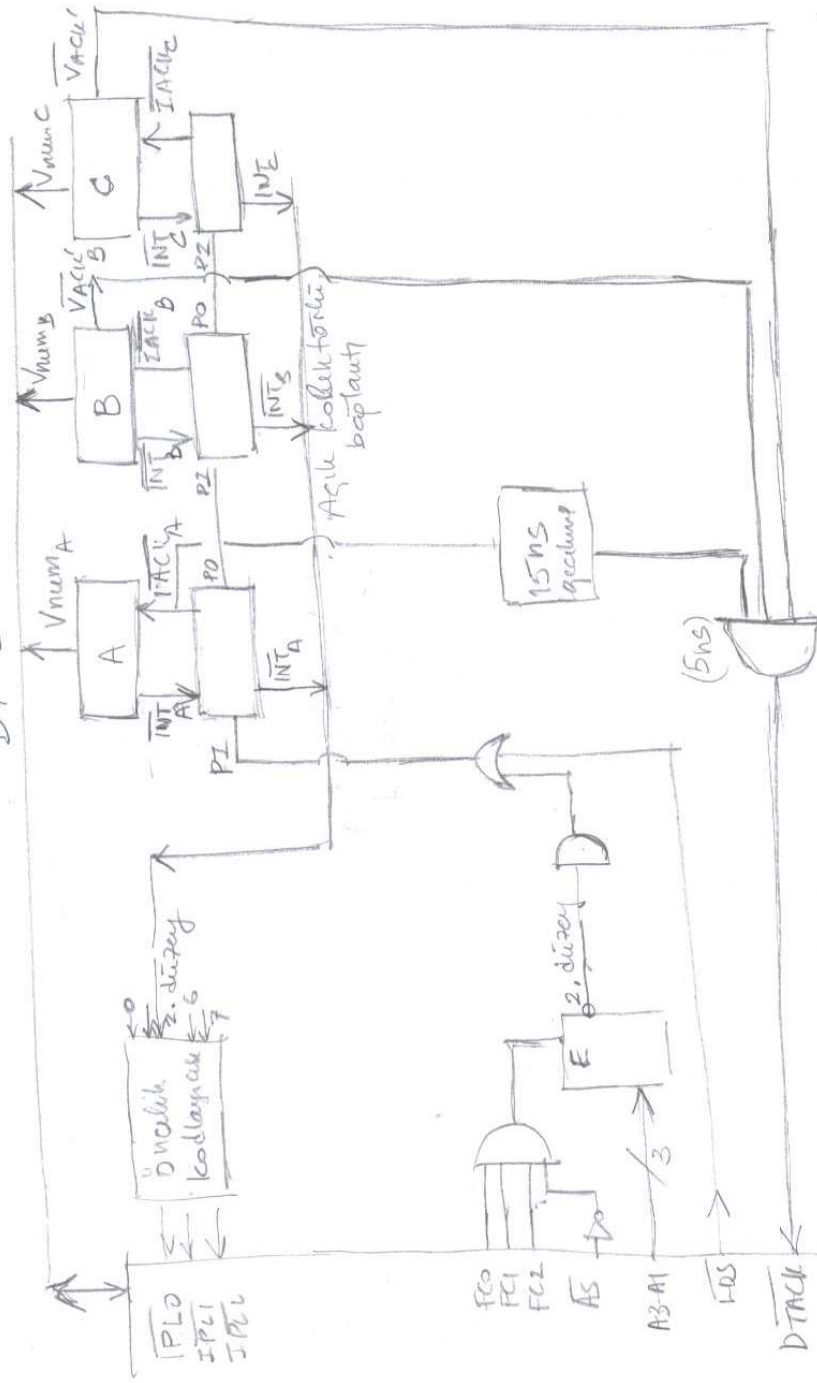
QUESTION 3:
SORU 3:

a) Papatya zincirinin bir halkası



c) C cihazına ait ISR çalışırken A cihazından kesme isteği gelirse C'nin ISR'si (A ile aynı düzeye bağlı olduğu için) kesilmez. A'nın kesme isteği bekletilir. C'nin ISR'i bitince, maske bir alt düzeye iner (0 veya 1), bir komut yürütülür sonra A'nın kesme isteği etkinleşir ve A'ya ait ISR çalışmaya başlar.

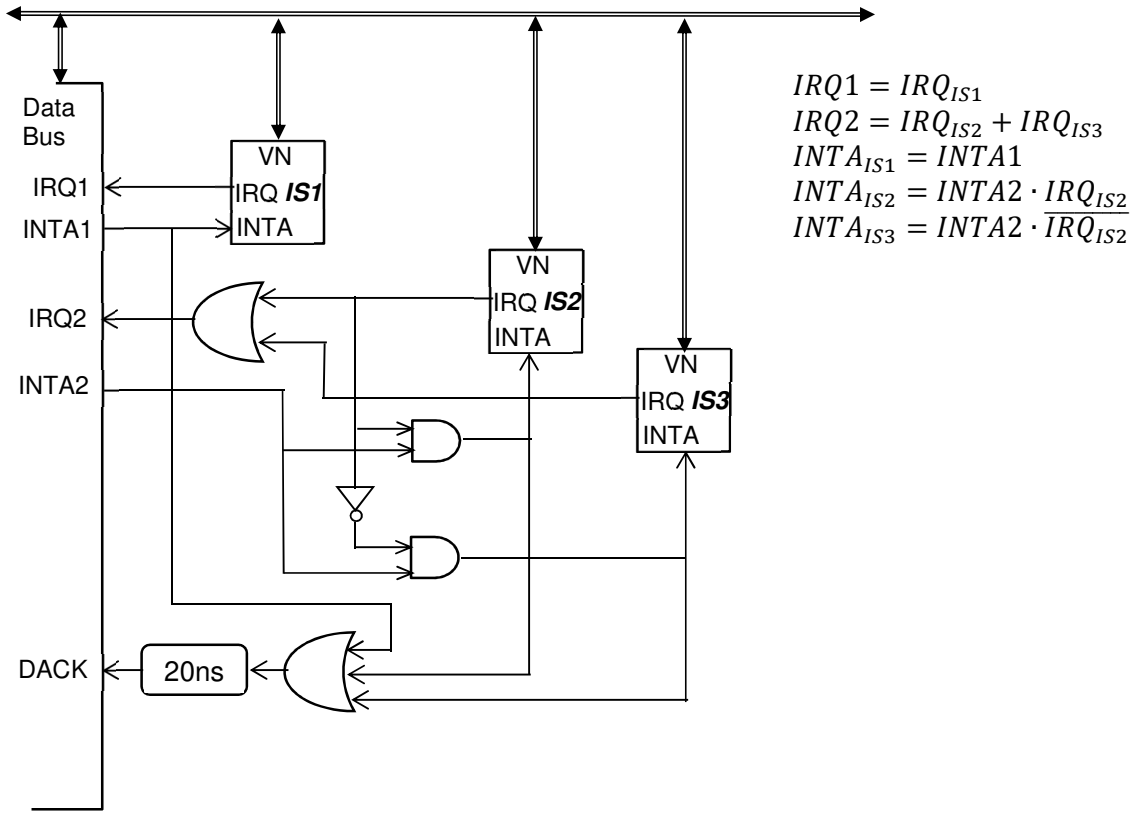
D7--D0



QUESTION 4:

SORU 4:

a)



b)

Remember; interrupt requests are checked after the completion of the instruction.

In this system, an instruction is completed at Clock= 180 ns (60+70+50).

Because of the interrupt request from IS2, the CPU enters the "4. interrupt" phase and starts housekeeping operations for IS2.

During the housekeeping operations the IS1 sends an interrupt request. The housekeeping operations (such as saving the return address and condition code) cannot be interrupted. Remember; interrupt requests are checked after the completion of the instruction.

After the housekeeping operations (200ns) at Clock= 380 ns (180 + 200) the ISR of IS2 starts to run.

ISR of IS2 at Clock= 380 ns.

Now IM1=1 IM2=0. Interrupts from IS1 are enabled.

After the completion of the 1st instruction of the ISR of the IS2 (Clock= 380 + 180 = 560 ns), the CPU checks interrupt requests, and enters the "4. interrupt" phase for IS1.

After the housekeeping operations (200ns) at Clock= 760 ns (560 + 200) the ISR of IS1 starts to run.

ISR of IS1 at Clock= 760 ns.

Hatırlatma: kesme istekleri komut tamamlandıktan sonra kontrol edilip değerlendirilir.

Bu sistemde bir komut Saat = 180 ns'de (60+70+50) tamamlanır.

IS2'den gelen kesme isteği nedeniyle MİB "4. kesme" çevrimine girer ve IS2 için hazırlık işlemlerine başlar.

Hazırlık işlemleri sırasında IS1 kesme isteği gönderir. Hazırlık işlemleri (geri dönüş adresinin ve durum bilgisinin saklanması gibi) kesilemezler Hatırlatma: kesme istekleri komut tamamlandıktan sonra kontrol edilip değerlendirilir.

Hazırlık işlemlerinden sonra (200ns) Saat = 380 ns'de (180 + 200) IS2'nin kesme hizmet programı çalışmaya başlar.

IS2'nin KHP'si Saat = 380 ns.

IM1=1 IM2=0 olur. IS1'den gelen kesme isteklerine izin vardır.

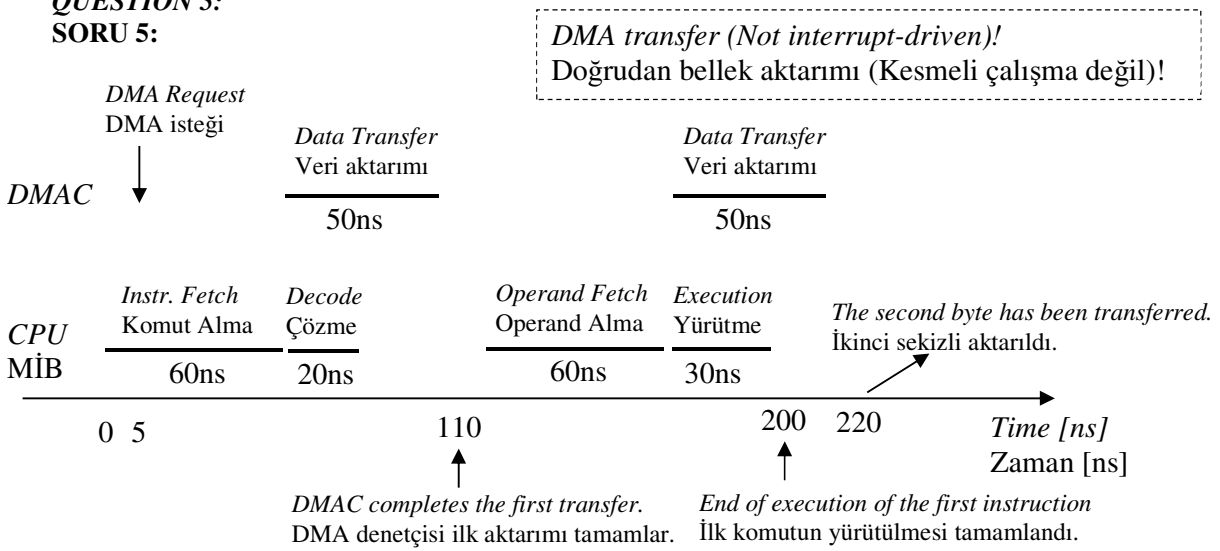
IS2'nin kesme hizmet programını birinci komutu tamamlandıktan sonra (Saat= 380 + 180 = 560 ns), MİB kesme isteklerini kontrol eder ve "4. kesme" çevrimine IS1'in hazırlık işlemleri için girer.

Hazırlık işlemlerinden sonra (200ns) Saat = 760 ns'de (560 + 200) IS1'in kesme hizmet programı çalışmaya başlar.

IS1 KHP'si Saat= 760 ns.

QUESTION 5:

SORU 5:



a)

CPU completes the current bus cycle (Instruction fetch) 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.

Time= 60 + 50 = 110ns

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

Zaman= 60 + 50 = 110ns

b)

Instruction decoding and execution cycles of the CPU can run in parallel with DMA transfers. Since the DMAC uses the **cycle-stealing** technique, after the transfer of the first byte it will give the bus to CPU. After the operand fetch the CPU executes the instruction.

$$\text{Time} = 60 + 50 + 60 + 30 = 200\text{ns}$$

MİB'in komut çözme ve operand alma çevrimleri DMA aktarımları ile paralel yürüyebilir. DMA denetçisi çevrim çalma yöntemiyle çalıştığı için ilk sekizliyi aktardıktan sonra yolu MİB'e verir. MİB operandı aldıktan sonra komutu yürütür.

$$\text{Zaman} = 60 + 50 + 60 + 30 = 200\text{ns}$$

c) (10 points / 10 puan)

During one instruction cycle the DMAC transfers two bytes in 220 ns.

10 bytes are transferred in $5 \times 220 = 1100$ ns.

$$\text{Time} = 5 * 220 = 1100\text{ns}$$

DMA denetçisi bir komut çevriminde **iki sekizliyi 220 ns'de** aktarmaktadır.

10 sekizli $5 \times 220 = 1100\text{ns}$ 'de aktarılır.

$$\text{Zaman} = 5 * 220 = 1100\text{ns}$$

Consider the difference.
Farka dikkat ediniz.

After the transfer of the 10 bytes the CPU runs each instruction in 170 ns.

$$\text{Time} = 1100 + 5 * 170 = 1950\text{ns}$$

On sekizli aktarıldıktan sonra MİB her komutu 170ns'de yürütür.

$$\text{Zaman} = 1100 + 5 * 170 = 1950\text{ns}$$

d)

Remember; interrupt requests are checked after the execution of the instruction. If there is a request the CPU enters the interrupt cycle (given in the question). The data is transferred in the ISR.

$$\text{Time} = 60 + 20 + 60 + 30 + 200 + 500 = 870\text{ns}$$

Hatırlatma; kesme istekleri komut yürütüldükten sonra yoklanır. Eğer kesme varsa MİB kesme çevrimine girer (soruda verilmişti). Veri aktarımı kesme hizmet programının içinde yapılır.

$$\text{Zaman} = 60 + 20 + 60 + 30 + 200 + 500 = 870\text{ns}$$

d) (10 points / 10 puan)

After the execution of each instruction one byte is transferred in the ISR in 870ns.

$$\text{Time} = 10 * 870 = 8700\text{ns}$$

Her komutun yürütülmesinden sonra kesme hizmet programında bir sekizli toplam 870 ns'de aktarılır.

$$\text{Zaman} = 10 * 870 = 8700\text{ns}$$

The execution of the last instruction is completed before the transfer of the last byte. Therefore we subtract the durations of the housekeeping and ISR (200ns+500ns) from total time (8700ns).

$$\text{Time} = 8700 - 700 = 8000\text{ns}$$

Son komutun yürütülmesi son sekizlinin aktarımından önce tamamlanır. Bu nedenle kesme hazırlık işlemleri ve kesme hizmet programı sürelerini (200ns + 500ns) toplam süreden (8700ns) çıkartıyoruz.

$$\text{Zaman} = 8700 - 700 = 8000\text{ns}$$

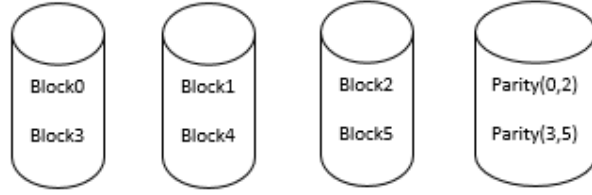
Note: Compare the results obtained by the DMA technique and interrupt-driven technique. The advantage of the DMA technique is obvious.

Not: DMA yöntemi ve kesmeli çalışma ile elde edilen sonuçları karşılaştırınız. Doğrudan bellek erişimi yönteminin avantajı açıkça görülmektedir.

QUESTION 6:

SORU 6:

a)



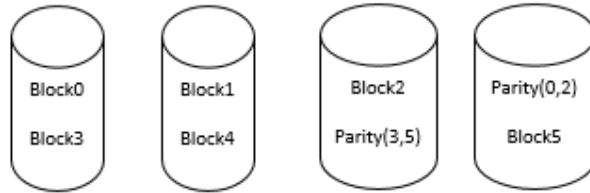
i) *Two different blocks from two different disks can be read in **ta**.*

İki farklı diskteki iki blok paralel olarak **ta** sürede okunur.

ii) *Two read and two write operations (**2ta**) should be performed for an update operation in RAID 4 (See the lecture notes). Since parity update operations cannot be performed independently (in parallel) (there's only one parity disk), it takes **4ta** to update words of two blocks in two different disks.*

RAID 4'te her yazma işlemi iki diski okumayı iki diske yazmayı gerektirir (Ders notlarına bakınız). Okumalar ve yazmalar kendi aralarında paralel olduğu için bir diske yazma **2ta** sürer. Ancak tek bir eşlik diski olduğundan eşlik güncelleme işlemleri paralel olarak yapılamaz. İki farklı diskteki iki bloğa yazma işi **4ta** sürer.

b)



i) *Same as in RAID 4: **ta**.*

RAID 4 ile aynıdır: **ta**.

ii) *For each data update two read and two write operations are necessary. Different from RAID4, now parity update operations can be performed in parallel, because parity strips are distributed to different disks: **2ta**.*

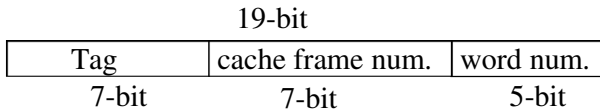
Her veri güncellemesi için iki okuma ve iki yazma işlemi gereklidir. RAID 4'ten farklı olarak eşlik güncelleme işlemleri paralel yapılabilir, çünkü eşlik bilgileri farklı disklere dağıtılmıştır: **2ta**.

SORU 7:

QUESTION 7:

a) Doğrudan dönüşüm (*Direct mapping*),

i) Physical address (fiziksel adres): 19-bit



ii)

Number of frames in the cache:

Cep bellekteki çerçeve sayısı:

$4KB/32B = 2^{12}/2^5 = 2^7 = 128$ frames.

The tag memory has 128 rows.

Takı belleğinde 128 satır vardır.

The length of the tag: Bir takımın uzunluğu: 7-bit

There is also additional information in each row: Valid bit, dirty bit

Takı belleğini her satırında ek bilgiler de vardır: Geçerlilik biti, değişim biti

Note: In direct mapping replacement LRU is not necessary. Therefore there are not aging counters in the tag memory.

Not: Doğrudan dönüşümde LRU yöntemine gerek yoktur. Bu nedenle takı belleğinde yaşlanma sayacı bulunmaz.

iii)

1-\$00053: 0000 0000 0000 0101 0011

Cache is empty, cache miss occurs, the block with the number (14 bits): 000 0000 0000 010 is placed into frame 0000010 of the cache memory. Tag value is 0000000

Block start address: 000 0000 0000 0100 0000

Block end address: 000 0000 0000 0101 1111

Cep bellek boş, iska olur, başvuru blok (000 0000 0000 010) cep belleğin 0000010 numaralı çerçevesine getirilir.

Takı değeri: 0000000

Blok başlangıç adresi: 000 0000 0000 0100 0000

Blok bitiş adresi: 000 0000 0000 0101 1111

2- \$01050: 000 0001 0000 0101 0000

The data is searched in the cache frame 0000 010; tag value in the cache is compared with tag value of the address. They are not same Cache: 0000 000, Address: 000 0001

A miss occurs. The block is placed into the same cache frame 0000 010. Replacement occurs.

Tag value is 000 0001.

Veri cep belleğin 0000 010 numaralı çerçevesinde aranır; cepteki takı değeri ile adresteki takı değeri karşılaştırılır. Değerler eşit değildir. Cep: 0000 000 Adres: 000 0001

Cepte iska olur, ilgili blok cep belleğin aynı çerçevesine (0000 010) getirilir, yer değiştirme işlemi yapılır.

Takı değeri: 000 0001

3- \$0004F: 000 0000 0000 0100 1111

The data is searched in the cache frame 0000 010; tag value in the cache is compared with tag value of the address. They are not same Cache: 0000 001, Address: 000 0000

A miss occurs. The block is placed into the same cache frame 0000 010. Replacement occurs.

Tag value is 000 0000.

Veri cep belleğin 0000 010 numaralı çerçevesinde aranır; cepteki takı değeri ile adresteki takı değeri karşılaştırılır. Değerler eşit değildir. Cep: 0000 001 Adres: 000 0000

Cepte iska olur, ilgili blok cep belleğin aynı çerçevesine (0000 010) getirilir, yer değiştirme işlemi yapılır.

Takı değeri: 000 0000

Bir kümede 2 blok bulunan kümeli çağrışsımlı yöntem (2-way set associative)

i) Physical address (fiziksel adres): 19-bit

19-bit		
Tag	set num.	word num.
8-bit	6-bit	5-bit

ii)

The tag memory has 128 rows (One tag for each frame).

Takı belleğinde 128 satır vardır (Her çerçeve için bir takı).

The length of the tag. Bir takımın uzunluğu: 8-bit

There is also additional information in each row: Valid bit, dirty bit, aging counters

Takı belleğini her satırında ek bilgiler de vardır: Geçerlilik biti, değişim biti, yaşlanma sayaçları

iii)

1-\$00053: 0000 0000 0000 0101 0011

Cache is empty, cache miss occurs, the block with the number (14 bits): 000 0000 0000 010 is placed into frame 0 of set 000 010 of the cache memory. Tag value is 000 0000 0

Cep bellek boş, ıskala olur, başvuru blok (000 0000 0000 010) cep belleğin 000 010 numaralı kümesinin 0 numaralı çerçevesine getirilir.
Takı değeri: 000 0000 0

2- \$01050: 000 0001 0000 0101 0000

The data is searched in the set 000 010; tag values in the set are compared with tag value of the address. They are not same.

A miss occurs. The block is placed into the same set 000 010, but into different frame. Replacement is not necessary

Tag value is 000 0001 0

Veri cep belleğin 000 010 numaralı kümesinde aranır; kümedeki çerçevelerin takı değerleri ile adresteki takı değeri karşılaştırılır.

Cepte ıskala olur, ilgili blok cep belleğin aynı kümesinin (000 010) başka bir çerçevesine getirilir, yer değiştirme gerekli değildir.

Takı değeri: 000 0001 0

3- \$0004F: 000 0000 0000 0100 1111

The data is searched in the set 000 010; tag values of the frames in the set are compared with tag value of the address. 000 0000 0

This block is in the cache; hit occurs.

Veri cep belleğin 000 010 numaralı kümesinde aranır; kümedeki çerçevelerin takı değerleri ile adresteki takı değeri karşılaştırılır. 000 0000 0

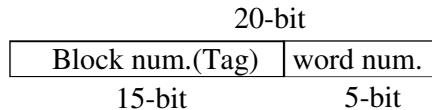
Bu blok daha önce cep belleğe getirilmişti, cepte vuru olur.

QUESTION 8:

SORU 8:

b) Tam çağrışımlı (Full associative)

i) Physical address (fiziksel adres): 20-bit



ii) Each frame of the cache memory has a row (line) in the tag memory.

Cep bellekteki her çerçeve bir takı karşı düşürüldüğünden takı belleğine her çerçeve için bir satır vardır.

Number of frames in the cache:

Cep bellekteki çerçeve sayısı:

$8KB/32B = 2^{13}/2^5 = 2^8 = 256$ frames.

The tag memory has 256 rows.

Takı belleğnde 256 satır vardır.

The length of the tag. Bir takının uzunluğu: 15-bit

There is also additional information in each row: Valid bit, dirty bit, aging counters

Takı belleğini her satırında ek bilgiler de vardır: Geçerlilik biti, değişim biti, yaşlanma sayaçları

iii)

1-\$00025: 0000 0000 0000 0010 0101

Cache is empty, cache miss occurs, the block with the number (15 bits): 0000 0000 0000 001 is placed into frame 0 (first empty frame)of the cache memory. Tag value is 0000 0000 0000 001

Block start address: 0000 0000 0000 0010 0000

Block end address: 0000 0000 0000 0011 1111

Cep bellek boş, ıskala olur, başvuru blok (0000 0000 0000 001) cep belleğin (ilk boş çerçevesine) çerçeve 0'ına getirilir.

Takı değeri: 0000 0000 0000 001

Blok başlangıç adresi: 0000 0000 0000 0010 0000

Blok bitiş adresi: 0000 0000 0000 0011 1111

2- \$0003F: 0000 0000 0000 0011 0101

The block number (tag) is searched in the cache memory (15 bits): 0000 0000 0000 001

This block is in the cache, cache hit occurs.

İlgili blok (takı) cep bellekte aranır: 0000 0000 0000 001

Bu blok daha önce cep belleğe getirilmişti, cepte vuru olur.

3- \$02020: 0000 0010 0000 0010 0000

The block number is searched in the cache memory (15 bits): 0000 0010 0000 001

This block (tag) is not in the cache, cache miss occurs.

The block is placed into cache frame 1.

Tag value is 0000 0010 0000 001

Block starting address: 0000 0010 0000 0010 0000

Blok end address: 0000 0010 0000 0011 1111

İlgili blok cep bellekte aranır: 0000 0010 0000 001

Bu blok (takı) cepte olmadığı için ıskala oluşur.

Blok cep belleğin 1 numaralı çerçevesine yerleştirilir.

Takı değeri: 0000 0010 0000 001

Blok başlangıç adresi: 0000 0010 0000 0010 0000

Blok bitiş adresi: 0000 0010 0000 0011 1111

c) Doğrudan dönüşüm (*Direct mapping*),

i) Physical address (fiziksel adres): 20-bit

20-bit

Tag	cache frame num.	word num.
7-bit	8-bit	5-bit

ii)

Number of frames in the cache:

Cep bellekteki çerçeve sayısı:

$8\text{KB}/32\text{B} = 2^{13}/2^5 = 2^8 = 256$ frames.

The tag memory has 256 rows.

Takı belleğinde 256 satır vardır.

The length of the tag. Bir takımın uzunluğu: 7-bit

There is also additional information in each row: Valid bit, dirty bit, aging counters

Takı belleğini her satırında ek bilgiler de vardır: Geçerlilik biti, değişim biti, yaşlanma sayaçları

iii)

1-\$00025: 0000 0000 0000 0010 0101

Cache is empty, cache miss occurs, the block with the number (15 bits): 0000 0000 0000 001 is placed into frame 0000 0001 of the cache memory. Tag value is 0000 000

Block start address: 0000 0000 0000 0010 0000

Block end address: 0000 0000 0000 0011 1111

Cep bellek boş, ıskala olur, başvuru blok (0000 0000 0000 001) cep belleğin 0000 0001 numaralı çerçevesine getirilir.

Takı değeri: 0000 000

Blok başlangıç adresi: 0000 0000 0000 0010 0000

Blok bitiş adresi: 0000 0000 0000 0011 1111

2- \$0003F: 0000 0000 0000 0011 0101

The data is searched in the cache frame 0000 0001; tag value in the cache is compared with tag value of the address. They are same 0000 000

This block is in the cache, cache hit occurs.

Veri cep belleğın 0000 0001 numaralı çerçevesinde aranır; cepteki takı değeri ile adresteki takı değeri karşılaştırılır. İkisi eşittir: 0000 000

Bu blok daha önce cep belleğe getirilmişti, cepte vuru olur.

3- \$02020: 0000 0010 0000 0010 0000

The data is searched in the cache frame 0000 0001; tag value in the cache is compared with tag value of the address. They are not same Cache: 0000 000, Address: 0000 001

A miss occurs. The block is placed into the same cache frame 0000 0001. Replacement occurs.

Tag value is 0000 001

Veri cep belleğın 0000 0001 numaralı çerçevesinde aranır; cepteki takı değeri ile adresteki takı değeri karşılaştırılır. Değerler eşit değildir. Cep: 0000 000 Adres: 0000 001

Cepte iska olur, ilgili blok cep belleğın aynı çerçevesine (0000 0001) getirilir, yer değiştirme işlemi yapılır. Takı değeri: 0000 001

d) Bir kümede 4 blok bulunan kümeli çağrışımly yöntem (*4-way set associative*)

i) Physical address (fiziksel adres): 20-bit

20-bit

Tag	set num.	word num.
9-bit	6-bit	5-bit

ii)

The tag memory has 256 rows.

Takı belleğnde 256 satır vardır.

The length of the tag. Bir takımın uzunluğu: 9-bit

There is also additional information in each row: Valid bit, dirty bit, aging counters

Takı belleğini her satırında ek bilgiler de vardır: Geçerlilik biti, değışim biti, yaşlanma sayaçları

iii)

1-\$00025: 0000 0000 0000 0010 0101

Cache is empty, cache miss occurs, the block with the number (15 bits): 0000 0000 0000 001 is placed into frame 0 of set 000 001 of the cache memory. Tag value is 0000 0000 0

Block start address: 0000 0000 0000 0010 0000

Block end address: 0000 0000 0000 0011 1111

Cep bellek boş, iska olur, başvuru blok (0000 0000 0000 001) cep belleğın 000 001 numaralı kümesinin 0 numaralı çerçevesine getirilir.

Takı değeri: 0000 0000 0

Blok başlangıç adresi: 0000 0000 0000 0010 0000

Blok bitiş adresi: 0000 0000 0000 0011 1111

2- \$0003F: 0000 0000 0000 0011 0101

The data is searched in the set 000 001; tag values of the frames in the set are compared with tag value of the address.

This block is in the cache, cache hit occurs.

Veri cep belleğın 000 001 numaralı kümesinde aranır; kümedeki çerçevelerin takı değeri ile adresteki takı değeri karşılaştırılır.

Bu blok daha önce cep belleğe getirilmişti, cepte vuru olur.

3- \$02020: 0000 0010 0000 0010 0000

The data is searched in the set 000 001; tag values in the set are compared with tag value of the address. They are not same Cache. Address: 0000 0010 0

A miss occurs. The block is placed into the same set 000 001, but into different frame. Replacement is not necessary

Tag value is 0000 0010 0

Veri cep belleğin 000 001 numaralı kümesinde aranır; kümedeki çerçevelerin takı değerleri ile adresteki takı değeri karşılaştırılır.

Değerler eşit değildir. Adres: 0000 0010 0

Cepte ıskı olur, ilgili blok cep belleğin aynı kümesinin (000 001) başka bir çerçevesine getirilir, yer deęiştirme gerekli değildir.

Takı değeri: 0000 0010 0

QUESTION 9:

SORU 9:

a)

MİB 1 (CPU 1): Modified

MİB 2 (CPU 2): Invalid

b)

1. CPU 2: Read miss;

starts to read main memory; sends "read request A" to other cache controllers.

2. CPU 1: gets "read request A";

blocks memory read operation of CPU2;

writes the block that contains A to the main memory;

changes the state of A to "shared"; sends "shared";

enables memory read operation of CPU 2.

3. CPU 2: gets "shared";

reads the block that contains A from the main memory;

changes the state of A to "shared";

1. MİB 2: Okuma ıskı;

Ana bellekten okuma başlatır; dięer cep bellek denetçilerine "okuma isteęi A" gönderir.

2. MİB 1: "okuma isteęi A" gelir.

MİB 2'nin bellek okuma işlemini bloke edilir;

A'yı içeren bloęu ana belleęe yazar;

A'nın durumunu "shared" ("paylaşılan") olarak deęiştirir; "shared" ("paylaşılan") mesajı gönderir;

MİB 2'nin bellek okuma işlemini serbest bırakır.

3. MİB 2: "shared" ("paylaşılan") mesajını alır;

A'yı içeren bloęu ana bellekten okur;

A'nın durumunu "shared" ("paylaşılan") olarak deęiştirir.