* 1. “Bus arbitration” nedir ve hangi durumlarda, nasıl gerçekleşir?

En basit olanları hariç, hemen tüm bilgisayar sistemlerinde, işlemci (processor) dışında kalan birimlerin de sistem bus yapısını doğrudan kullanmaları gerekebilir. Bu gereksinimin tipik bir örneği, bir I/O modülün doğrudan bellek erişimi (DMA) yapmasıdır. Bus yapısı, belirli bir anda yalnızca bir birim tarafından kullanılabileceği için, sözkonusu ortak kullanım, ancak zaman paylaşımı ile olanaklıdır. Hangi birimin, hangi öncelik sırası ile ve ne zaman bus yapısını kullanacağına karar verme işlemine “bus arbitration” denmektedir. “Bus arbitration” işlemi genellikle merkezi (centralized) ve dağıtılmış (distributed) olmak üzere 2 yöntemle gerçekleştirilir. Merkezi yöntemde, bu işi bir “bus arbiter” (bazen “bus controller” olarak da adlandırılır) yapar. Dağıtılmış yöntemde ise bir merkezi “arbiter” bulunmaz, her modül kendi erişim kontrol mantığına sahiptir ve bus yapısını bu mantık kurgusu çerçevesinde paylaşırlar.

* 1. Ortalama erişim süresi (average access time) 2.5ms, veri aktarım hızı (transfer rate) 12 Mbit/s olan bir disk biriminde, 10Kbyte verinin ortalama okuma/yazma süresi ne kadardır?

*TN = TA + N/R Burada, TN: N-bit okuma/yazma için ortalama süre, TA: Ortalama erişim süresi, N: Aktarılacak toplam bit sayısı, R: b/s cinsinden aktarım hızını göstermektedir. Buna göre:*

*N=10Kbyte = 10x1024x8 = 81,920bit, R=12Mbit/s = 12x1,048,576 =12,582,912bit/s*

*TN = 2x10-3 + 81,920/12,582,912 = 0,009 = 9 ms bulunur.*

2)Aşağıdaki yapının bağlı bulunduğu bilgisayarın adres uzayını (address space) belirtiniz ve bellek haritasını bulunuz. Bu yapıya sadece mevcut RAM yongalarından ekleyerek elde edilebilecek en büyük bellek kapasitesi ne olur? Nasıl sağlanır?

°

°

°

MEMR

MEMR

MEMW

3×8 Decoder

A10

A0

D0 D7

CS R

2K × 8

EPROM

2

1

0

0

7

E

A15

°

3

D0 D3

A11

A0

CS R W

4K × 4

RAM

D4 D7

A11

A0

CS R W

4K × 4

RAM

A14

A13

A12

MEMR

A11

MEMW

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 : 0000H

2K EPROM

0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 : 07FFH

0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 : 7000H

4K RAM

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 : 7FFFH

3x8 decoder üzerinde 1,2,3,4,5,6 çıkışları boştur. Buralara da şekildeki gibi 4Kx4 RAM

Yonga çiftleri bağlanabilir. Böylece mevcut 4KB RAM kapasitesi 4x7=28KB olur. Bu

28KB RAM alanı, bellek haritasında 1000H – 7FFFH arasında yer alır.

1. Yapısında, ana bellek, cache, ve sanal bellek (virtual memory) olarak da kullanılan disk bulunan bir bilgisayarda, bir kelimenin cache’den okunması 15ns süre almaktadır. Aranan kelime cache’de bulunmuyor, ana bellekte bulunuyorsa, kelimenin önce cache’e yüklenmesi 50ns sürmekte ve ardından da cache’den okuma işlemi (yukarıdaki gibi) yapılmaktadır. Aranan kelime ana bellekte de yoksa, kelimenin diskten ana belleğe getirilmesi 10ms sürmekte, sonra, 50ns içinde cache’e yüklenmekte ve oradan, yukarıda açıklandığı gibi okunmaktadır. Arama işlemlerinin isabet oranları (hit ratio), cache için 0.85, ana bellek için ise 0.7 olduğuna göre, bir kelimenin ortalama okuma süresi (ns cinsinden) ne kadardır?

NOT: Hit ratio = 1, %100 isabet anlamına gelir.

|  |  |  |
| --- | --- | --- |
| **Aranan Kelimenin Yeri** | **Bulunma Olasılığı** | **Erişim Süresi (ns)** |
| Cache’de | 0.85 | 15 |
| Cache’de yok, ana bellekte var | (0.15)×(0.7) = 0.105 | 50 + 15 = 65 |
| Cache’de de yok, bellekte de yok | (0.15)×(0.3) = 0.045 | (10×106) + 50 + 15 = 10000065 |

*Ortalama erişim süresi için, 3 olası durumdaki erişim sürelerini, o durumların olasılıkları ile çarpar ve bulduğumuz sonuçları toplarsak:*

*[(0.85) × (15)] + [(0.105) × (65)] + [(0.045) × (10000065)] = 450022.5 ns buluruz*

24-bit “address bus”, 16-bit “data bus” yapısı olan bir mikroişlemcide, 8 Kbyte, 4-way set-associative cache bellek bulunmaktadır. Cache’in her satırı (bloğu), “**byte-level” adreslenebilir** nitelikte 4 adet 16-bit kelimeden oluşmaktadır. 24-bit bellek adresinin hangi boylarda, kaç bölüme ayrılacağını ve bu yapının gerek cache, gerekse ana bellek adreslemede (“byte-level”) nasıl kullanılacağını, organizasyon şemasını çizerek açıklayınız Belleğin FF096A (Hex) adresinde kayıtlı bulunan bir verinin, cache yapısı içinde nerede bulunacağını bulunuz.

Aynı bus yapıları ve bellek büyüklükleri “direct mapping” ve “associative mapping” biçiminde organize edilseydi, 24-bit bellek adresi hangi bölümlerden oluşurdu?

*Blok büyüklüğü = 4 × 16 bit = 64 bit = 8× 8 bit = 8 byte*

*Blok içinde 8 byte adresleyebilmek için gerekli bit sayısı = log2(8) = 3 bit (offset).*

*Cache içindeki blok sayısı = 8 Kbyte / 8 byte = (8 × 1024) / 8 = 1024*

*Cache içindeki set sayısı = Cache içindeki blok sayısı / bir setteki blok sayısı*

*= 1024 / 4 = 256*

*256 adet seti adresleyebilmek için gerekli bit sayısı = log2(256) =* ***8***

*TAG için kullanılacak bit sayısı = 24 – (8 + 3) =* ***13***

*Bu veriler ışığında adres yapısını çizersek:*

Tag = 13

Set = 8

Byte = 3

*Verilen FF096A adresinin açılımı:* 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 0 1 1 0 1 0 1 0

*Görüldüğü gibi, set adresi (altı çizili 8 bit) = 00101101 = 2DH = 4510 olmaktadır. Buna göre verilen adres içeriği, cache’in 2D adresli setinin 4 bloğundan birinin, 2 numaralı byte’ında (byte = 010), 1111111100001 = 1FE1H etiketi ile bulunabilir*

*“Direct Mapping” yapılırsa, her blok (satır) yine 8 byte olacağına göre, cache alanı 8K/8 = 1024 satırdan oluşacaktır. 1024 satır, log2(1024) = 10 bit ile adreslenebilir. Yine Byte = 3 bit olacağına göre, Tag = 24-(10+3) = 11 bit olur. Buna göre adres yapısı şöyle olur:*

Tag = 11

Line = 10

Byte = 3

*“Associative” yapıda, “byte” ve “Tag” bulunacağından, adres yapısı aşağıdaki gibi olur:*

Tag = 21

Byte = 3

“Set Associative” yapının organizasyon şeması aşağıda gösterilmektedir:

Byte = 0

Ana Bellek

⊗

Tag (20) Blok (128)

Tag Set Byte

Blok – 0

Blok – 1

Blok – 2

Blok - 3

Set - 0

Blok – 4

Blok – 5

Blok – 6

Blok - 7

Set - 1

Blok – 1020

Blok – 1021

Blok – 1022

Blok - 1023

Set - 255

Cache

Compare

⊗

Hit in cache

13

8

3

Not hit in cache

To Cache

4. Aşağıda bir bellek kesiminin adres ve içerik dökümü gösterilmektedir. Bir varsayımsal (hypothetical) makinenin komut kümesi içinde ise, şu 3 komutun yer aldığıbilinmektedir:

# HEX Kod Komut Açıklama

5XXX Mov M, AC XXX bellek adresi içeriğini AC’ye aktar

6XXX Mov AC, M AC içeriğini XXX bellek adresine aktar

7XXX Add M, AC XXX bellek adresi içeriğini AC’ye topla

Aşağıdaki tablonun boş gözlerini, belirtilen noktadan itibaren, her satır bir adıma karşılık gelecek ve tüm adımlar ilgili sütun başlıklarının içerik değişimini gösterecek biçimde doldurunuz. Bunu yaparken içeriğinin ilgisiz bulduğunuz gözler olursa, buralara X koyunuz.

NOT: “Cycle” sütununda, “Fetch” ya da “Execute” dönüllerinden hangisi olduğu belirtilecektir. Tablodaki satır sayısı rastgele seçilmiş olup, yazmanız gereken adım sayısını göstermeyebilir.

Bellek:

|  |  |
| --- | --- |
| Adres | İçerik |
| 4B8 | 57F8 |
| 4B9 | 77F9 |
| 4BA | 67FA |
| : | : |
| 7F8 | 5200 |
| 7F9 | 2187 |
| 7FA | 3DFF |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cycle | 7F8 | 7F9 | 7FA | PC | AC | IR | MAR | MBR |
| Fetch | 5200 | 2187 | 3DFF | 4B8 | X | 57F8 | 4B8 | 57F8 |
| Execute | 5200 | 2187 | 3DFF | 4B9 | 5200 | 57F8 | 7F8 | 5200 |
| Fetch | 5200 | 2187 | 3DFF | 4B9 | 5200 | 77F9 | 4B9 | 77F9 |
| Execute | 5200 | 2187 | 3DFF | 4BA | 7387 | 77F9 | 7F9 | 2187 |
| Fetch | 5200 | 2187 | 3DFF | 4BA | 7387 | 67FA | 4BA | 67FA |
| Execute | 5200 | 2187 | 7387 | 4BB | 7387 | 67FA | 7FA | 7387 |

Adres / İçerik

Puanlar: Soru-1 : 30p Soru-2: 30p Soru-3: 25p Soru-4: 15p

1. Ortalama erişim süresi (average access time) 4ms, veri aktarım hızı (transfer rate) 12Mbit/s olan bir disk biriminde, 10Kbyte verinin ortalama okuma/yazma süresi ne kadardır?

*TN = TA + N/R Burada, TN: N-bit okuma/yazma için ortalama süre, TA: Ortalama erişim süresi, N: Aktarılacak toplam bit sayısı, R: bps cinsinden aktarım hızını göstermektedir. Buna göre:*

*N = 10Kbyte = 10x1024x8 = 81,920bit, 12Mbit = 12x1,048,576 =12,582,912bit*

*TN = 4x10-3 + 81920/12582912 = 4x10-3+ 0,0065 s = 10.5 ms bulunur.*

1. Bir sistem yapısında kullanılan kesilmeler (interrupt) için öncelik sıralaması ve ISR (kesilme servis yordamı) işleyiş süreleri ve kesilme istemlerinin yapıldığı anlar aşağıdaki gibidir (zaman birimi önemli değildir):

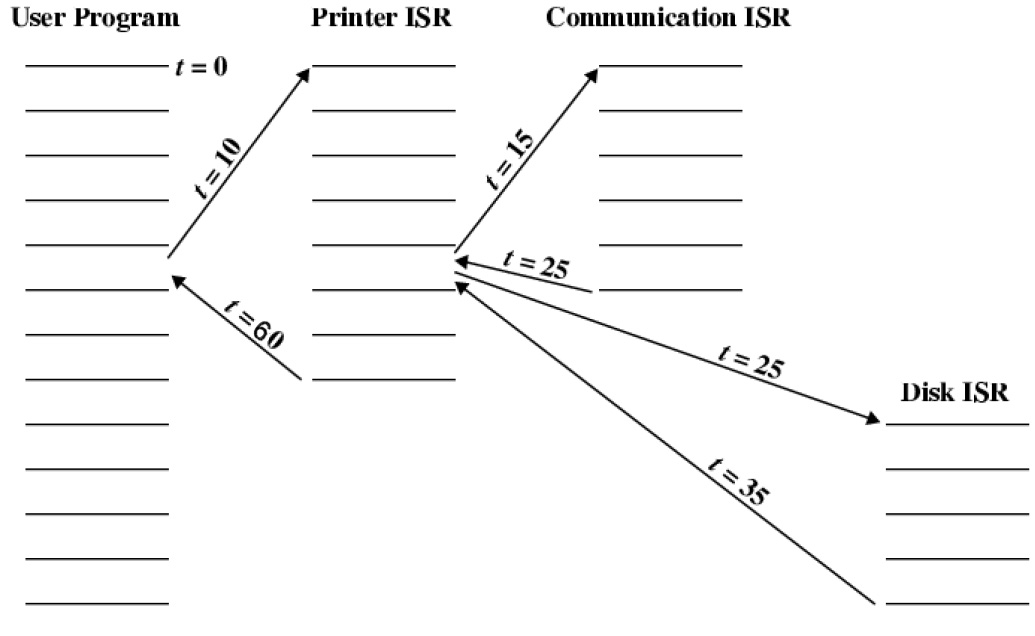
Öncelik Kesilme İstemcisi ISR süresi Kesilme istemini yaptığı an (t)

1 İletişim Arayüzü 10 15

2 Disk Sürücü 10 20

3 Yazıcı 30 10

Ana programın t = 0 anında başladığını varsayarak, yazıcı ISR’sinin tamamlanış zamanını (t = ?) bulunuz. Gerekçenizi açıklayınız.



Printer ISR, şekilde görüldüğü gibi t = 60 da tamamlanmaktadır.

1. Ortalama erişim süresi (average access time) 4ms, veri aktarım hızı (transfer rate) 12Mbit/s olan bir disk biriminde, 10Kbyte verinin ortalama okuma/yazma süresi ne kadardır?

*TN = TA + N/R Burada, TN: N-bit okuma/yazma için ortalama süre, TA: Ortalama erişim süresi, N: Aktarılacak toplam bit sayısı, R: bps cinsinden aktarım hızını göstermektedir. Buna göre:*

*N = 10Kbyte = 10x1024x8 = 81,920bit, 12Mbit = 12x1,048,576 =12,582,912bit*

*TN = 4x10-3 + 81920/12582912 = 4x10-3+ 0,0065 s = 10.5 ms bulunur.*

**1.**

Byte (word) seviyesinde adresleme yapılabilen bir “Direct Mapping cache” organizasyonunda adres yapısı yukardaki gibidir. Buna göre:

a) Ana belleğin büyüklüğünü b) Cache belleğin büyüklüğünü

c) Ana bellekteki blok sayısını d) Blok (satır) büyüklüğünü

bulunuz.

1. *Adres uzunluğu : 8+13+1 = 22 Bit olduğuna göre bellek büyüklüğü : 222 =* ***4Mbyte***
2. *Cache 213 = 8K satırdan oluşmaktadır. Her satır 21 = 2 Byte (word) içerdiğine göre, cache büyüklüğü = 8K × 2 =* ***16Kbyte*** *olmaktadır.*
3. *Her Blok 2 byte olduğuna göre, ana bellek 4M/2 =* ***2M*** *blok içerir.*
4. *Word adresleme işlemi w = 1 bit ile yapıldığına göre, blok büyüklüğü 21 = 2 byte olacaktır.*

Bu yapıda, belleğin 25DB2D (Hex) adresindeki sözcük, cache alanının hangi satırına ve hangi sözcüğüne yerleşir?

*Verilen adresi 22-bit olarak yazarsak:*

w

Line

Tag

1 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1

2 5 D B 2 D

*Bu adresin “Line” bölümü = 0 1 1 0 1 1 0 0 1 0 1 1 0 = 0D96 olduğundan verilen adresteki sözcük cache alanının* ***0D96 Hex adresli satırının ikinci sözcüğüne*** *(w = 1) oturur.*

İçeriği aynı cache satırına yerleşebilecek en küçük bellek adresi nedir?

*“Line” adresi 0D96 olan 22 bit uzunluğundaki en küçük adresi yazmak istersek:*

Tag Line w

0 0 0 0 0 0 0 0 **0 1 1 0 1 1 0 0 1 0 1 1 0** 0

0 0 1 B 2 C

*Belleğin 001B2C adresi aynı satıra yerleşecek en küçük bellek adresi olmaktadır.*

**4.** “Associative Mapping” için aynı kurgunun bellek adresleme yapısı nasıl olurdu?

21

1

Tag w

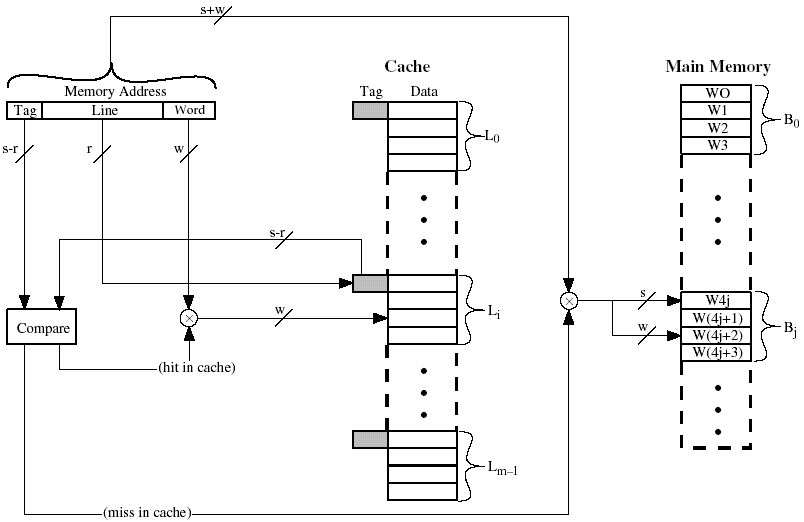
*Yukarıda gösterildiği gibi Word = 1-bit aynı kalırdı, adresin geri kalan kısmı Tag olurdu.*

1. 64 KBytes ana bellekli bir yapıda, 2 KBytes cache bellek kullanılması ve her blokda 16 word bulunması istenmektedir. Bellek organizasyonu, 1 Byte = 1 wordolaraktasarlanmıştır**.**
   1. Bu yapıda, en az kaç bit bellek adresi kullanılması gerektiğini belirtiniz.

*log2 (65,536) = 16, En az 16 bit adres gerekir. (64K = 216)*

* 1. “Direct Mapping” yöntemi için, söz konusu bellek adresinin hangi boylarda, kaç bölüme ayrılacağını ve bu yapının gerek cache, gerekse ana bellek adreslemede nasıl kullanılacağını, organizasyon şemasını çizerek açıklayınız.

*Her blokda 16 word olacağına ve toplam cache alanı da 2048 Bytes (2 KBytes) olduğuna göre, cache alanı toplam 2048/16 = 128 blokdan (satırdan) oluşacaktır. 128 bloğu adresleyebilmek için, log2(128) = 7 bit gerekir (27 = 128). O halde, r = 7 olmalıdır. Öte yandan, her blokdaki 16 word’ü adresleyebilmek için de log2(16) = 4 bit gerekir (w = 4). Bu durumda, toplam adres boyu 16 bit olduğuna göre (s+w = 16), Tag için ayrılacak bit sayısı: (s + w) – (w + r) = 16 – (4 +7) = 5 olacaktır. bellekten bir word okunmak istendiğinde, önce adresin r bit bölümü (line) ile cache satırına erişim yapılır ve adresin ilk 5 biti (tag) o satırın etiketi (tag) ile karşılaştırılır. Tag’lar aynı ise (hit), adresin son 4 biti (w) kullanılarak istenen word okunur. Tag’lar aynı değilse, önce s+w bit adres ile (blok başına ulaşmak için w=0000 kullanılır) bellekten bir blok cache’a kopyalanır ve adresin son 4 biti kullanılarak istenen word okunur.*



:

:

:

:

:

W0

W15

W15

W127

* 1. Cache alanının kaç satırdan (kaç bloktan) oluşacağını ve ana bellekteki toplam blok sayısını belirtiniz. Ana bellekte ilk bloğun “Blok-0” olarak numaralandığını ve cache satırlarının “Satır-0”dan başlandığını varsayarak, bellekteki “Blok-132”nin hangi cache satırına haritalandığını bulunuz.

*(b) bölümünde açıklandığı gibi, cache alanı 128 blokdan (satırdan) oluşmaktadır. Benzer düşünüşle, ana bellek 64Kbytes ve her blok da 16 word (bytes) olduğuna göre, ana bellekdeki toplam blok sayısı: 65,536/16 = 4096 olacaktır (64K/16 = 4K).*

*i = j mod m (i = cache satır no., j = ana bellek blok no., m = cache’daki toplam satır sayısı) olduğuna göre, bellekteki Blok-132, 4. cache satırına yerleşecektir (132/128 işleminin artanı).*

* 1. Aynı temel yapı, “2-way Set Associative Mapping” yöntemi için organize edilse idi, bellek adresi hangi boylarda, kaç bölüme ayrılırdı? Bu durumda, bellekdeki “Blok-132”, cache alanında hangi sete (“Set-0”dan başladığını varsayarak) haritalanırdı?

*“2-way Set Associative Mapping” yönteminde, her set içinde 2 blok bulunacaktır. Bir blokdaki word sayısı ise, problem tanımı uyarınca yine 16 olacaktır. Bloklardaki word sayısı aynı kaldığı için de, cache alanındaki toplam blok sayısı (128) ve ana bellekteki toplam blok sayısı da (4096), direct-mapping ile aynı kalacaktır. Her set 2 blok içereceğine göre, cache alanı 128/2 = 64 set’ten oluşacaktır. 64 seti adreslemek için, log2(64) = 6 bit adres kullanmak gerekir. Her blokdaki 16 adet word ise, yine 4 bit ile adreslenecektir. Buna göre Tag için kalan bit sayısı, bellek adresi boyundan (16) bu iki sayı çıkartılarak kolayca bulunur: 16 – (6+4) = 6. Sonuç olarak, “2-way Set Associative Mapping” yönteminde bellek adresi 6-bit Tag, 6-bit Set ve 4-bit word alanlarından oluşacaktır. Bu kez, i = j mod v (i = cache set no., j = ana bellek blok no., v = toplam set sayısı) olacağından, Blok-132, 4. sete yerleşecektir (132/64 işleminin artanı).*