EEM 335 YARIYIL SONU SINAVI YANITLARI 14.01.2004

1. Aşağıdaki soruları yanıtlayınız

a) ENIAC bilgisayar evrim süreci içinde ne anlam ifade eder?

ENIAC, Dünya'nın ilk genel amaçlı elektronik sayısal bilgisayarı idi.

b) Geleneksel bus mimarisinden, yüksek performanslı bus mimarisine geçiş gereğini doğuran temel etmenler nelerdir? Söz konusu mimariler arasındaki temel fark nedir?

Gelişen teknolojiye paralel olarak, kapasite kullanımı ve iletişim hızlarında da önemli artışlar oldu. Bu çerçevede, hız, kapasite ve işlevselliği artan çevre birimleri (disk, optik disk, tarayıcı, yazıcı, uzak terminal v.b) ile, hem daha büyük hacımlı veriyi daha çabuk aktarmak, hem de yeni uygulama alanları yaratmak (multimedya gibi) olanağı doğdu. Bu da geleneksel tek bus yapısını, birden fazla ve daha yüksek hızda iletişim yapabilen bus'lardan oluşan yapı ile değiştirme gereğini doğurdu.

c) Bellek hiyerarşisini simgeleyen piramitin tepesinden tabana doğru inildiğinde, neler değişir ve ne şekilde değişir? Her iki konum için birer de örnek veriniz.

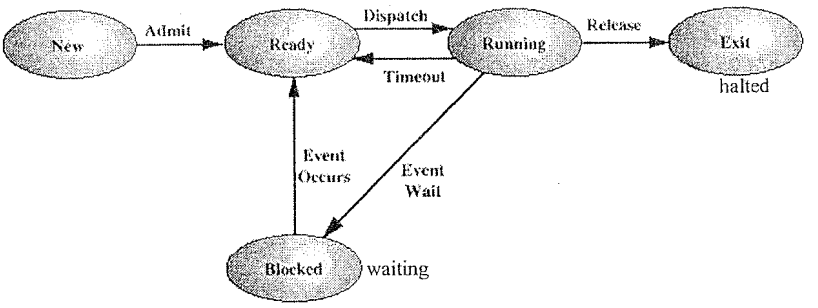
- *Bit başına düşen maliyet azalır*
- *Kapasite artar*
- *Erişim süresi artar (erişim hızı düşer)*
- *İşlemcinin bu birimlere erişme sıklığı azalır*

Örnek :
Hiyerarşinin tepesinde: Yazmaçlar, cache, ana bellek
Hiyerarşinin tabanında: Teyp, MO, WORM

d) Cache kullanımında, "write through" ve "write back" teknikleri neyi sağlar ve nasıl çalışır?

"write through" ve "write back" teknikleri, cache içeriğindeki kayıtlar ile, bu kayıtların ana bellekteki kopyalarının, en son yenilenmiş geçerli kayıtlar olmasını sağlayan tekniklerdir. Cache her zaman dolu bir alan olduğundan, cache'e yazılacak yeni bir kayıtın, bir başka kayıtın yerini alması gerekir (replacement). Bu durumda, mevcut kayıt silinmeden önce, bu kaydın en son yenilenen kopyasının sadece cache'de bulunmadığından emin olmak gerekir. "Write through" tekniğinde, tüm yazma işlemleri, hem cache'e, hem de ana belleğe yapılarak bu sorun çözülür (ancak çok fazla yavaşlamaya neden olur). "Write back" tekniğinde ise, bir kayıt cache'de bulunup yenilendiğinde, bir özel UPDATE biti set olur. Bir kayıtın yerine başkası yazılmadan önce, mevcut kayıtın UPDATE bitine bakılır. Şayet bu bit 1 ise, önce o kayıt bellekteki orijinali üzeine kopyalanır, sonra silinir.

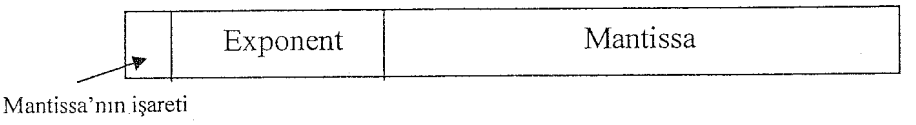
e) "State of a process" kavramını, 5 durumlu "process" modeli çerçevesinde açıklayınız.



Çoklu programlama (multiprogramming) ortamında, birden fazla işin (process), işletim sistemi tarafından en verimli bir biçimde çalıştırılabilmesi için, bu işlerin hangisinin, hangi anda, ne kadar süre ile veya hangi aşamaya kadar işleyeceğine; bellek yeterli değilse, hangi işin askıya alınarak bellekten diske, hangi bekleyen işin, diskten belleğe taşınacağına işletim sistemi (scheduler) tarafından karar verilmesi gerekir. Bunun için de, her yeni işin, işletim için uygun görülüp kuyruğa alınışından (new), tamamlanıp sona erişine kadar (halted) içinde bulunabileceği her evreye, o işin statüsü ya da durumu (state of process) denir. Bellekte işletime hazır bir durumda bekleyen bir iş "ready", işlemekte olan bir iş "running", kaynak kullanımını (I/O gibi) bekleyen bir iş "waiting" durumundadır. Bu durumlar yukarıdaki şekilde gösterilmiştir.

f) Nümerik değerlerin kayan noktalı (floating point) gösterimi ve kayıtlanmasında, "dynamic range" ve "precision" kavramları neyi ifade eder ve nasıl değişir?

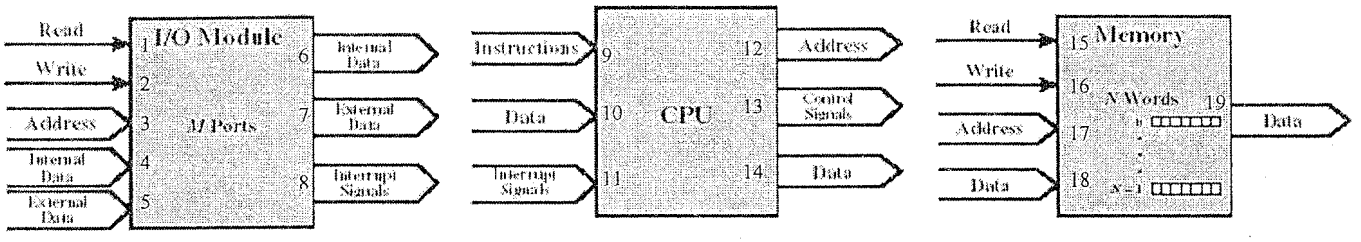
Kesirli bir sayının ifade ettiği matematiksel değerin duyarlığı (precision), o sayının kesir bölümünün (noktadan sonraki) hane sayısına bağlıdır. Bu da, kayan noktalı gösterimde "mantissa" bölümünün boyu demektir. "Dynamic range" ise, kullanılan gösterim kalıbı ile ifade edilebilecek en küçük negatif sayı ile en büyük pozitif sayı arasındaki açılımdır. Bu da, üssel (exponential) gösterimde, üssün (exponent) hane sayısına (±) bağlıdır.



Kayan noktalı bir gösterimde (yukarıdaki şekilde olduğu gibi), kelime boyu (m+n+1 bit) sabit olacağından, mantissa veya exponent'in bit sayısını büyütmek, ancak diğerinin bit sayısını aynı miktarda kısaltmakla olanaklıdır. Bu da, "precision" ile "dynamic range"in, aynı kelime boyu içinde, birlikte büyütülüp küçültülemiyeceği, birinin büyümesinin bedelinin, diğerinin küçültülmesi olacağı anlamına gelir.

2. Aşağıdaki soruları yanıtlayınız

a) Aşağıdaki birimlerin tüm giriş çıkış bağlantıları numaralandırılmıştır. Bu bağlantılar, söz konusu birimlerin sistem bus üzerinden gereği gibi iletişim kurmalarını sağlarlar. Birimlerden beklenen işlevselliğin sağlanabilmesi için, aşağıdaki tablonun en sol sütununda belirtilen uçların, ilk sırasında belirtilen uçlardan hangileri ile bağlantı kurmaları gerekeceğini düşününüz ve karşılık gelen gözlere, bağlantı data bus üzerinden olacaksa "D", adres bus üzerinden olacaksa "A", kontrol bus üzerinden olacaksa "C" yazınız.



	1	2	3	4	5	9	10	11	15	16	17	18
6							D					
7												
8								C				
12			A								A	
13	C	C							C	C		
14				D								D
19						D	D					

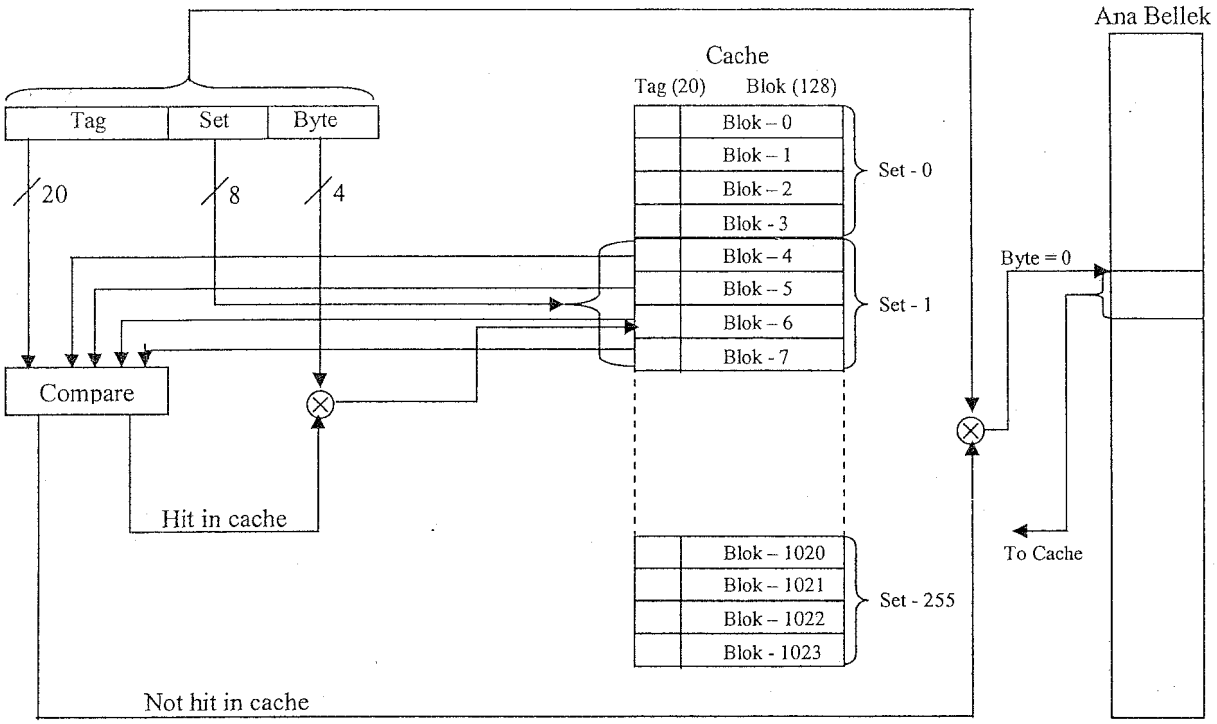
b) Bu yapıya bir yazıcı ile bir tarayıcı (scanner) eklenmek istenirse, yazıcı ve tarayıcının hangi bağlantılarının, bu uçlardan hangilerine bağlanmaları gerekir.

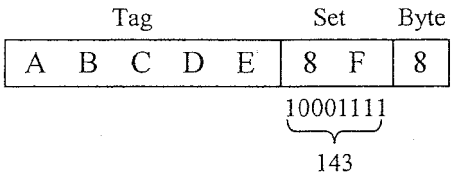
Aslında yazıcı ve tarayıcının, kontrol sinyal iletişimi bulunması gerekir. Ancak şekilde I/O modül bloğunda kontrol sinyalleri gösterilmediğinden, yazıcı ve tarayıcı bloklarında da gösterilmemiştir.

3. Aşağıdaki soruları yanıtlayınız

a) 32-bit'lik bir mikroişlemcinin on-chip, 16 Kbyte, 4-way set-associative cache yapısı bulunmaktadır. Cache'in her satırı (bloğu), "**byte-level" adreslenebilir** nitelikte 4 adet 32-bit kelimeden oluşmaktadır. 32-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 ABCDE8F8 (Hex) adresinde kayıtlı bulunan bir verinin, cache yapısı içinde nerede bulunacağını bulunuz.

Blok büyüklüğü = 4 × 32 bit = 128 bit = 16 × 8 bit = 16 byte
Blok içinde 16 byte adresleyebilmek için gerekli bit sayısı = log2(16) = 4 bit (offset).
Cache içindeki blok sayısı = 16 Kbyte / 16 byte = (16 × 1024) / 16 = 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 = 32 – (8 + 4) = 20
Bu veriler ışığında organizasyon şemasını çizersek:





Belleğin ABCDE8F8 adresinde bukunan bir veri, cache'deki 143_{10} numaralı setin, ABCDE Tag'ini taşıyan bloğunun, 8 numaralı byte'ında (ikinci kelime) bulunacaktır.

b) Yapısında, ana bellek, cache, ve sanal bellek (virtual memory) olarak da kullanılan disk bulunan bir bilgisayarda, bir kelimenin cache'den okunması 20ns süre almaktadır. Aranan kelime cache'de bulunmuyor, ana bellekte bulunuyorsa, kelimenin önce cache'e yüklenmesi 60ns sürmekte ve ardından da cache'den okuma işlemi (yukarıdaki gibi) yapılmaktadır. Aranan kelime ana bellekte de yoksa, kelimenin diskten ana belleğe getirilmesi 12ms sürmekte, sonra, 60ns içinde cache'e yüklenmekte ve oradan, yukarıda açıklandığı gibi okunmaktadır. Arama işlemlerinin isabet oranları (hit ratio), cache için 0.9, ana bellek için ise 0.6 olduğuna göre, bir kelimenin ortalama okuma süresi (ns cinsinden) ne kadardır?
NOT: Hit ratio = 1, %100 isabet anlamına gelir.

Aranan kelimenin bulunduğu yer	Olasılık	Erişim süresi (ns)
Cache'de	0.9	20
Cache'de yok, ana bellekte var	$(0.1)\times(0.6) = 0.06$	$60 + 20 = 80$
Cache'de de yok, bellekte de yok	$(0.1)\times(0.4) = 0.04$	$(12\times10^6) + 60 + 20 = 12000080$

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

$[(0.9) \times (20)] + [(0.06) \times (80)] + [(0.04) \times (12000080)] = 480026$ *ns buluruz.*

4. Aşağıdaki soruları yanıtlayınız

a) "Logical address", "virtual address" ve "demand paging" kavramlarını açıklayınız.

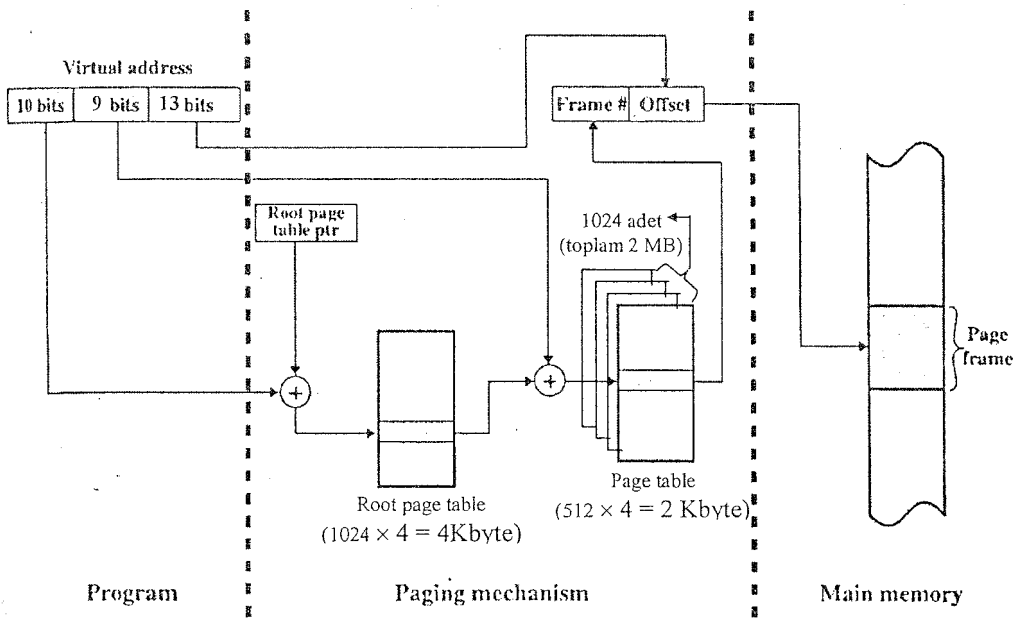
Bir iş (process) belleğe işletilmek üzere getirildiğinde, bu işin belleğin neresine yerleştirileceği bir tür optimizasyon problemi olup, bu sorunu çözmek, işletim sisteminin temel işlevlerinden biridir (memory management). Buna göre, bir işin belleğe her getirilişinde (sanal bellek kullanılmıyor olsa bile), o iş içindeki sözcüklerin göreceli adresleri aynı kalırken, fiziksel adresleri değişecektir. İşte, bu sorunu çözmek için, bir iş içindeki tüm adresler, o işin başlangıç konumu referans alınarak göreceli biçimde ifade edilir ve bu göreceli adreslere "logical address" denir. İşlemci bir programı çalıştırırken, bu "logical address"leri "base address" olarak adlandırılan başlangıç adresi ile toplayarak, fiziksel adrese çevirir.

n-bit'lik adres bus ile, 2^n farklı konum adreslenebilir. Ancak günümüz mimarilerinde, ekonomik gerekçelerle, bir sisteme bu kadar büyük bellek takılmamakta ve disk alanı, bir tür sanal bellek (virtual memory) olarak kullanılmaktadır. Bu durumda n-bit adresin ancak bir bölümünü ana bellekteki bir adrese karşılık gelecektir. İşte özde bir tür "logical adress" olan ve karşılığı kısmen diskte bulunan bu n-bit adrese "virtual address" denir. "virtual adres"i fiziksel adrese dönüştürmek, yukarıda açıklandığı kadar basit bir işlem olmayıp, bunun için muhtelif yöntemler geliştirilmiştir.

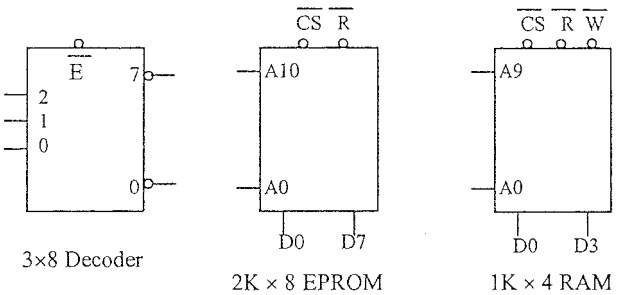
Sanal bellek kullanılan durumlarda, ana bellek kapasitesinden büyük işler de çalıştırılabilir. Bunun için o iş "page" adı verilen sayfalara bölünür ve tümüyle belleğe yüklenmez. Program akışı sırasında, çalıştırılma sırası gelen komutlar şayet bellekte yoksa, işletim sistemi tarafından diskten belleğe taşınır. Bu uygulamaya "demand paging" adı verilir.

b) 32-bit ile "byte-level" adreslenen bir sanal adres ortamı (virtual address space), her biri 8 Kbyte olan sayfalara ayrılmıştır. Sayfa tablosundaki (page table) her kayıt 4-byte uzunluktadır. Buna göre:
- Sanal bellek alanının kaç sayfadan olştuğunu bulunuz
- Gereksinim duyulan sayfa tablosunun büyüklüğünü bulunuz
- 2 seviyeli bir sayfalama düzeni (2-level paging) için kurgulayacağınız yapıyı ve 32-bit sanal adresi nasıl kullanacağınızı çizerek açıklayınız.

*Her bir sayfa içeriği, 8 Kbyte = 2^{13} bit ile adreslenebilir. Sanal adres alanı 32 bit ile adreslendiğine göre, sanal bellek alanı, $2^{32} / 2^{13} = 2^{19}$ sayfadan oluşmaktadır. Bu durumda, sayfa tablosunda (page table) toplam 2^{19} kayıt (PTE) bulunacaktır. Sayfa tablosundaki her kayıt 4 byte olduğuna göre, sayfa tablosunun büyüklüğü $2^{19} \times 2^2 = 2^{21} =$ **2 Mbyte** olur.*
2 seviyeli sayfalama kurgusunu buna göre çizersek:



5.



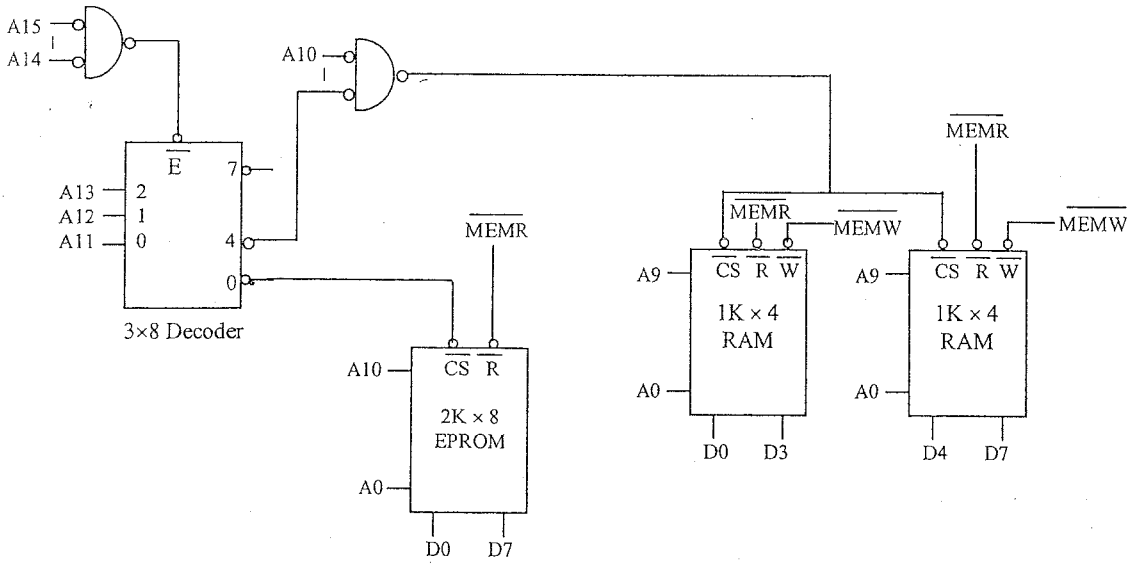
16-bit adres bus ve 8-bit data bus kullanan bir mikroişlemcili yapı için, elinizde yukarıda gösterilen decoder'den 1 adet, EPROM ve RAM chip'lerinden yeteri kadar ve 2 girişli NEGATIVE NAND geçitlerinden de 2 adet bulunduğunu varsayarak, aşağıdaki bellek haritasına uygun bağlantı şemasını çiziniz.

0000H – 07FFH Monitor Program (EPROM'da duracak)
2000H – 23FFH Kullanıcı belleği

Verilen bellek haritasını, 16-bit adres yapısı içinde incelersek:

	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
2K EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000 – 07FF	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1K RAM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2000 – 23FF	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1

"Chip select" mantığını, verilen devre elemanları ile bu haritaya göre kurgularsak:



Puanlar: Soru 1: Her şık 4 puan, toplam 24p
Soru 2: a) 10p, b) 5p, toplam 15p
Soru 3: a) 16p, b) 10p, toplam 26p
Soru 4: a) 10p, b) 15p toplam 25p
Soru 5: 10p

Süre: 120dk.

EEM 335 KISA SINAV-2 **14.11.2007**

1. Blok büyüklüğü 8 byte olan 2^{16} byte toplam ana bellekli bir yapıda, 64 satırdan oluşan "direct mapped cache" kullanılmaktadır.
   a) Bellek adresinin bölümlerini ve bu bölümlerin kaçar bit olduğunu açıklayınız

Blok büyüklüğü $= 2^w = 8 = 2^3 \rightarrow$ ***w = word = 3***
Satır sayısı $= m = 2^r = 64 = 2^6 \rightarrow$ ***r = line = 6***
Adresin tamamı 16 bit olduğuna göre, etiket boyu = 16-3-6 = 7, ***TAG = 7***

   b) Ana bellekte 2C96 adresindeki içerik, hangi cache satırına yerleştirilir?

2C96 = ***0010110***0100101***10*** *Bunu yukarıdaki kalıba göre ayırırsak, ortadaki grup (koyu olmayan) satır bilgisidir. Bu da 010010 = 12H satırı olduğunu gösterir.*

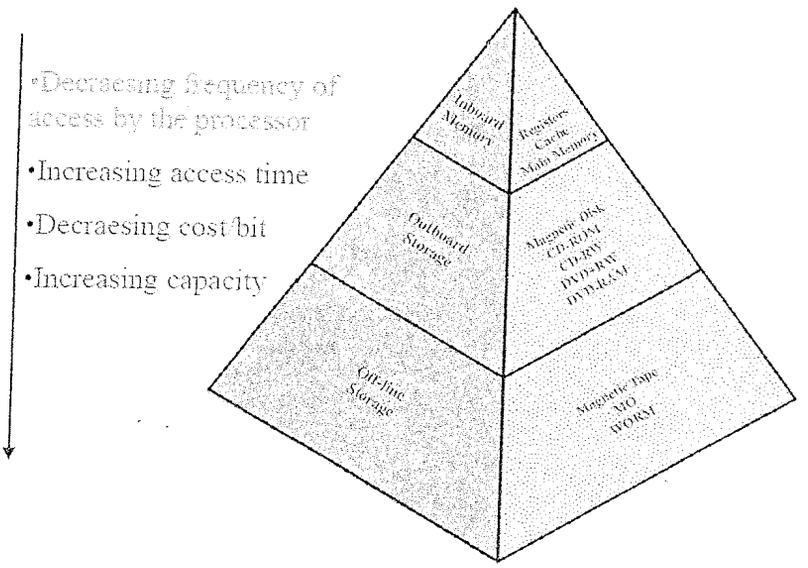
   c) Ana belleğin A043 adresindeki byte cache'e yerleştirildiğinde, bu byte ile birlikte başka hangi ana bellek adreslerindeki byte'lar da cache'e taşınır?

Bir blokdaki ilk sözcük (bu soruda sözcük boyu 1 byte) w = 0 olan sözcüktür. Bu da bu soruda en sağdaki 3 bitin 0 olması anlamına gelir. Bu da A040 adresidir. Blokda toplam 8 sözcük bulunduğuna göre, söz konusu 8 byte, A040 – A047 adresleri arasındaki 8 byte'dir.

2. Bellek organizasyonunda "hiyerarşi" nedir? Ne amaçla kullanılır? Nasıl işlev görür?

Bellek teknolojilerinin günümüzdeki konumunda şu özellikler bulunmaktadır:
- *Erişim hızı arttıkça, bit başına düşen maliyet de artmaktadır*
- *Kapasite arttıkça, bit başına düşen maliyet azalmaktadır.*
- *Kapasite arttıkça, erişim hızı azalmaktadır.*

Kullanıcılar, büyük bellek kapasitesinden yanadır. Çünkü hem kapasiteye gereksinimleri vardır, hem de maliyet düşük olmaktadır. Ne var ki, performans da istenmektedir ve bunun için de daha düşük kapasiteli pahalı bellekler gereklidir. Bu açmazın çözümü, bellek hiyerarşisi uygulamasında bulunmuştur. Buna göre, aşağıdaki hiyerarşik yapıda yukardan aşağı inildikçe, o bellek türüne daha seyrek erişim gereği duyulmakta, erişim hızı ve maliyet azalmakta, kapasite ise artmaktadır.



Soru 1-a : 20p Soru 1-b : 18p Soru 1-c : 17p Soru 2 : 45p

EEM335 QUIZ-2 YANITLARI 24.12.2003

1. Çok sözcük içeren veri aktarımı (multiple word I/O) söz konusu olduğunda, DMA işlemi, niçin kesilmeli (interrupt driven) veya programlanmış giriş/çıkış'dan (programmed I/O) daha verimlidir? Açıklayınız.

Programlanmış I/O'da, CPU tümüyle I/O işlemine atanmış olup, söz konusu işlemi tüm gerekli adımları ile üstlenmiş durumdadır. Konuya salt I/O açısından bakıldığında, söz konusu işlem olası en yüksek hızda icra edilmiş olur ama, CPU da başka hiçbir iş yapamaz. Kesilmeli I/O'da, CPU, I/O biriminin işlemi tamamlayıp tamamlamadığını anlamak için gözlem döngüsünde beklemekten kurtulur ama yine de veri aktarımını yapmak ona düşer. Büyük hacımda veri aktarımı söz konusu olduğunda, (özellikle "multiprogramming" ortamında) bu durum, CPU'nun verimli kullanımını engelleyecektir. Oysa DMA yöntemi uygulandığında, CPU, I/O işlemini tümüyle DMA modülüne havale eder ve başka işlemlere yönelir. DMA modülü, kendi CPU'sunu kullanarak istenen I/O işlemini yerine getirir. Bunun için sistem BUS yapısının kullanılması gereklidir. Bu da, ya BUS yapısının CPU tarafından kullanılmadığı anlarda, ya da "cycle-stealing" yöntemi ile sağlanır.

2. Belleğe haritalanmış giriş/çıkış (memory mapped I/O) ile, yalıtılmış giriş/çıkış (isolated I/O) arasında ne gibi farklılıklar vardır?

Yalıtılmış I/O'da, yazma/okuma işlemi, IOR (I/O Read) ve IOW (I/O Write) sinyalleri tarafından tetiklenir. Bu sinyaller de, I/O işlemleri için tasarlanmış özel komutlar işlenirken üretilir. Bu nedenle, I/O birimleri ile bellek yongalarını aynı adreslere bağlamanın sakıncası yoktur (bellekten okuma/yazma yapmak için tasarlanmış komutlar işlenirken, tetikleme işlemini MEMR (Memory Read) ve MEMW (Memory Write) sinyalleri yapacaktır). Oysa, belleğe haritalanmış I/O'da, özel I/O komutları bulunmaz ve gerek belleğe, gerekse I/O birimlerine yönelik okuma yazma işlemlerinde aynı komutlar (ve MEMR – MEMW sinyalleri) kullanılır. Bu yüzden, toplam adres alanının bir bölümünü I/O birimlerine ayırmak ve bu adreslere bellek yongası bağlamamak gerekir. Belleğe haritalanmış I/O'nun artısı, I/O işlemlerinde de, çok seçenekli ve yetenekli bellek okuma/yazma komutlarından yararlanabilme olanağı (buna karşılık toplam adreslenebilir bellek alanından bir bölümün, I/O'ya ayrılıp, bellek için kullanılamaması) iken, yalıtılmış I/O'nun artısı, bellek alanından kayba uğramaksızın, aynı adres seçeneklerinin hem bellek, hem de I/O için kullanılabilmesi (buna karşılık bellek okuma/yazma komutlarının I/O için kullanılamaması) olmaktadır.

3. İşletim sistemlerinde kaç tür "scheduling" yapılır? Kısaca açıklayınız.

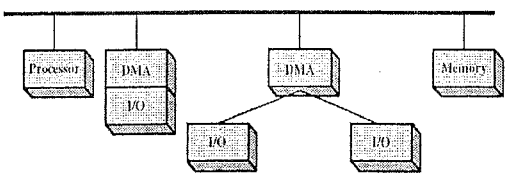
Uzun vadeli planlama (Long-term scheduling): Sırası geldiğinde işleme girebilecek işlerin (task'ların) kabul edilerek "process" statüsü almaları ve kısa vadeli planlayıcı (short-term scheduler) kuyruğuna girmeleri

Orta vadeli planlama (Medium-term scheduling): Çoklu programlama (multiprogramming) gereksinimlerine ve bellek olanaklarına uygun olarak, "swapping-in" ve "swapping-out" (diskten belleğe, bellekten diske taşıma) işlemlerinin planlanması.

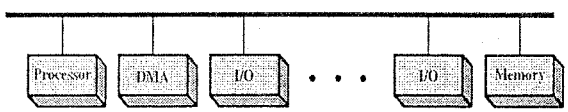
Kısa vadeli planlama (Short-term scheduling): "Process" statüsünde bulunan işlerin, zaman paylaşımı (time sharing) düzeni içinde ve en verimli biçimde işlenmesini veya geçici olarak askıya (bekleme evresine) alınmasını sağlamak.

4. Bir "bus cycle" ortalama 5 milisaniye süre aldığına göre, aşağıdaki DMA işlemlerinde bir word aktarımı (I/O modül ile bellek arasında) ne kadar süre alır?

(a)



(b)



(b) modelinde, I/O işleminin her evresinde, DMA ve Processor aynı BUS yapısını kullanmak durumundadır. Bu nedenle bellek ile I/O birimi arasındaki her "word" aktarımı (bir adet okuma, bir adet yazma) 2 "bus cycle" kullanacaktır. Bu da, 10 ms süre alır.

(a) modelinde ise, "system bus"tan ayrı olarak DMA modülü ile I/O modülleri arasında "bus" yapısı bulunmaktadır. Bu durumda DMA modülü yalnızca bellek erişimi için "system bus"ı kullanacak (okuma veya yazma), I/O modülüne erişim için (yazma veya okuma) ise, processor'dan "bus cycle" almasına gerek kalmayacaktır. Bu da 5ms sürelik sistem zamanı kullanmak anlamına gelir.

NOT: Puanlar eşit olup, süre 25 dakikadır.

EEM335 2. Ara Sınav Yanıtları 16.12.2003

1. Aşağıdaki soruları yanıtlayınız:

a) Yığıt (stack) ile kesilme (interrupt) süreci arasında nasıl bir ilişki bulunduğunu açıklayınız.

Kesilme istemi kabul edildiğinde, program akışı kesilme servis yordamına (ISR) yönlendirilmeden önce yapılacak ilk iki işlem, o anda işlenmekte olan komuttan bir sonra işlemesi gereken komut adresinin (PC içeriği) ve "Processor Status Word" (PSW) olarak adlandırılan, başta akümlatör ve bayrak yazmacı içerikleri olmak üzere, programın kaldığı yerden devamı için en gerekli verinin yığıta yazılarak saklanmasıdır. Bunun dığında kalan yazmaç içeriklerinin saklanması da gerekiyorsa, bu işlemin de ISR içinde gerçekleştirilmesi gerekir. Kesilme dönüşünde, bu saklanan veriler, sırası ile yığıttan geri alınarak eski yerlerine yeniden yazılmalı ve programın kaldığı yerden devamı sağlanmalıdır.

b) "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.

c) "Locality of reference" nedir ve "cache" kullanımında nasıl etkili olur? Örnek vererek açıklayınız.

"Locality of reference" kavramı, program akışı sırasında, bellekte belirli bir küçük zaman dilimi içinde erişilme gereği duyulacak içeriklerin, yüksek bir olasılıkla, son erişilen adresler ve bu adreslere yakın olan adresler çevresinde yoğunlaşacağı öngörüsüdür. "Cache" kullanımı, bu öngörünün gerçekleşmesi oranında başarılı ve anlamlı olmaktadır. Bunun en güzel örneği bellekte yer alan bir programın komut dizilerine birer birer erişim yapmada gözlenebilir. Bir program, uzak noktalara atlama (branching) yapılmadığı sürece, birbirini izleyen adreslerdeki komutların işlenmesi ile çalışır. Bu durumda, "cache" alanına okunan blokların, birbirini izleyen, sözgelimi, 50 komutluk bölümleri içerdiğini varsaysak, 50 komutun işlemi süresince bu komutlar "cache"den okunacak ve

belleğe erişme gereği olmayacaktır. Döngülü durumlarda bu durum daha da çok sayıda komut işlemini kapsayabilir.

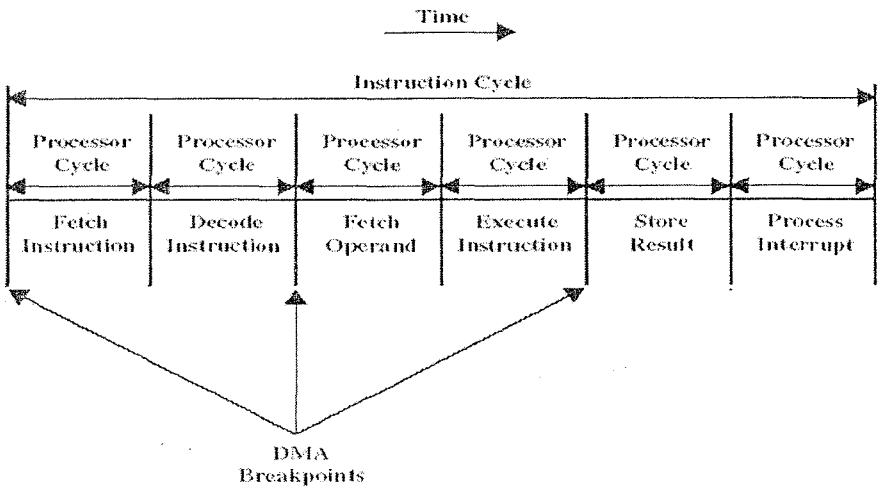
d) 1 Mbyte ana bellek ve her blokda 8 word bulunan bir yapıda, "associative mapping" yöntemi ile cache organizasyonu tasarlandığında, TAG uzunluğu kaç bit olur?

1 Mbyte bellek alanı, log2 (1,048,576) = 20 bit ile adreslenebilir. Bir blok içindeki 8 word ise, log2 (8) = 3 bit ile adreslenir. "Associative mapping" yapısında tolam bellek adresi "Tag" ve "Word" alanlarından oluşacağına ve Word alanı yukarıda açıklandığı gibi 3 bit olacağına göre, "tag" alanı 20 – 3 =1 7 bit boyunda olacaktır.

e) CAV (constant angular velocity) uygulanan bir diskte (hard disk), bit/mm cinsinden kayıt yoğunluğu sektörler ve izler arasında farklılık gösterir mi? Yanıtınızı gerekçeli bir biçimde açıklayınız ve yoğunluk sınırını neyin tayin edeceğini belirtiniz.

Açısal hızın sabit bulunduğu çembersel bir dönüş hareketinde, bir noktanın çizgisel (tangential) hızı, çembersel hareketin yarıçapı ile orantılıdır (v = ω r). Buna göre, merkezden uzaklaşıldıkça çizgisel hız artacaktır. Diskte veri kayıt ve okuma hızı her adres için aynı olduğuna ve bir sektör uzunluğundaki ize aynı hacımda veri kaydedildiğine göre (sektör içindeki iz nerede olursa olsun), merkeze yakın izlerdeki kayıt yoğunluğu, merkezden uzaktaki izlerin kayıt yoğunluğundan fazla olacaktır. Bu durumda, en içteki ize yapılabilecek kayıt yoğunluğu, maksimum kayıt yoğunluğu sınırını da tayin eder. Anlaşılacağı gibi, kayıt yoğunluğu sektörlere göre değil, izlere göre değişmektedir.

f) Bir işlemci (processor), DMA talebini tam olarak ne zaman ve nasıl işleme koyar?



İşlemcinin DMA talebini işleme koyabileceği anlar, işlemcinin sistem bus yapısını kullanma gereği duyacağı anlar olup, yukarıdaki şekilde gösterilmektedir (diğer

dönüllerde işlemci bus yapısını kullanmayacağı için DMA yapmada zaten engel yoktur). İşlemci, bu talebin gereğini yerine getirmek için, DMA talebini aldığı anı izleyen yukarıdaki anlardan ilkinde, kendi işlemini (bus cycle) askıya alır ve talep yapan birime "acknowledge" gönderir. Bu mesaj talep yapan tarafın bus yapısını kullanabileceği anlamına gelir. Bus yapısını her seferinde bir dönül (cycle) için kullanarak ve böylece her seferinde 1 word aktararak işe koyulan DMA modülü (bu işleme "cycle stealing" adı verilir), veri aktarım işlemi tamamlandığında "interrupt" aracılığı ile bu durumu işlemciye bildirir. Uygulanan "arbitration" kurgusuna göre, karşılıklı sinyalleşmelerle, daha uzun sürelerle işlemcinin askıda tutulup, DMA yapılmasına izin verilen uygulamalar da söz konusudur.

2. Aşağıdaki soruları yanıtlayınız:

a) 1 Byte uzunluğundaki nümerik veri, Hamming kod ile kodlandıktan sonra, bir iletim kanalından aktarılmış ve alıcı tarafa 100110010100 olarak ulaşmıştır. SEC (single error correcting) kod uygulaması yapıldığına göre, alınan sayının ondalık karşılığını bulunuz.

8 bit boyundaki bir sözcük Hamming kod ile kodlandığında, $2^k \geq 8+k+1$ olacak sayıda kontrol biti eklemek gerekir. Bu kontrol bitleri, 2'nin kuvveti olan pozisyonlara yerleştirilir. Buna göre k = 4 olur ve sözcükteki veri bitleri (D_i) ile kontrol bitlerinin (C_i) yerleşimi şöyledir:

Pozisyon No:	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*
	D_8	D_7	D_6	D_5	C_8	D_4	D_3	D_2	C_4	D_1	C_2	C_1
	1	*0*	*0*	*1*	*1*	*0*	*0*	*1*	*0*	*1*	*0*	*0*

Kontrol bitleri yerleştirilirken, $C_8C_4C_2C_1$ dizisinin ifade ettiği pozisyon numarası, hatanın yerini gösterecek biçimde (0 = hata yok anlamına gelmek üzere) oluşturulur. Bunun için de, 8,9,10,11,12 pozisyonları çift eşlik üretecek biçimde C_8; 4,5,6,7,12 pozisyonları çift eşlik üretecek biçimde C_4; 2,3,6,7,10,11 pozisyonları çift eşlik üretecek biçimde C_2; ve 1,3,5,7,9,11 pozisyonları çift eşlik üretecek biçimde C_1 hesaplanır. Alınan sözcük için de aynı denetimler yapılırsa:

8,9,10,11,12 pozisyonları için bulunan eşlik : TEK → $C_8 = 1$
4,5,6,7,12 pozisyonları için bulunan eşlik : ÇİFT → $C_4 = 0$
2,3,6,7,10,11 pozisyonları için bulunan eşlik: TEK → $C_2 = 1$
1,3,5,7,9,11 pozisyonları için bulunan eşlik : TEK → $C_1 = 1$

unur. Buna göre pozisyon numarası: $C_8C_4C_2C_1 = 1011 = 11_{(10)}$ olmakta ve 11 aralı pozisyondaki bitin hatalı olduğunu göstermektedir. 11 numaralı syondaki biti değilleyerek (hatayı düzelterek) veri bitlerini yazarsak: $D_7D_6D_5D_4D_3D_2D_1 = 11010011 = 211_{(10)}$ buluruz.

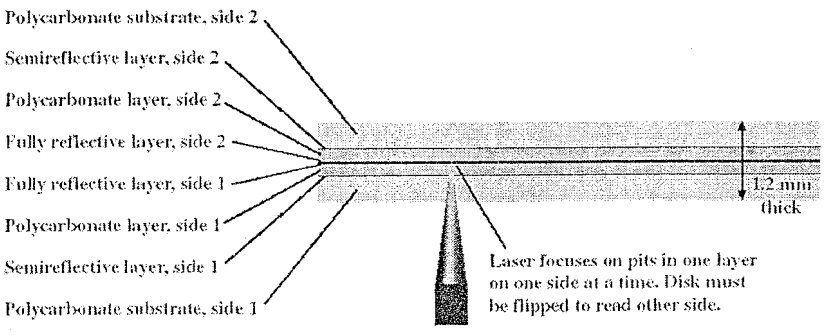
b) Dönme hızı 7500rpm, ortalama arama süresi (average seek time) 9ms olan bir diskte (hard disk), her sektörde 512 Byte ve her izde de (track) 600 sektör bulunmaktadır. Bu disk üzerinden 600 sektör kapsayan bir dosya okunacağını ve bu dosyanın a) tümünün aynı izde, b) tümünün rasgele dağılmış ayrı sektörlerde bulunduğunu varsayarak, en iyi ve en kötü toplam veri okuma sürelerini bulunuz.

NOT: $T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$

7500rpm dönme hızı, r = 7500/60 = 125rps olur. Bir tur dönme süresi = 1/r = 1/125 = 0.008s = 8ms olacaktır. Okunacak 600 sektör boyundaki verinin tümüyle aynı izde bulunduğu durumda, söz konusu veri için aktarım süresi b/rN = (600×512)/(125×600×512) = 8ms olur. Esasen görüleceği gibi bu süre, bir tur dönme süresi kadardır. Buna göre ortalama en iyi toplam veri okuma süresi T_a = 9 + 4 + 8 = 21ms olacaktır.

Okunacak tüm veri muhtelif izlerdeki sektörlere rasgele dağılmış durumda ise, Ts ve 1/2r (ortalama arama süresi ve bulunan izde, okunacak sektöre ulaşmak için geçecek ortalama süre) gecikmelerinin her sektör (512 byte) okuma için hesaba katılması gerekir. Buna göre 1 sektör verinin aktarım süresi b/rN = 512/(125×600×512) = 0.0000133s = 0.0133ms olur. Buna göre bir okuma (512 byte) için ortalama toplam okuma süresi, T_a = 9 + 4 + 0.0133 = 13.0133ms olacaktır. 600 sektör için bu süre 600 kez harcanacağına göre, ortalama en kötü toplam veri okuma süresi, 600×13.0133 = 7807.98ms olarak bulunur.

c) Bir DVD yapısını ve üzerine nasıl veri kaydedildiğini açıklayınız. DVD'lerin kapasitesi CD'lerden niçin ve yaklaşık kaç misli fazladır?



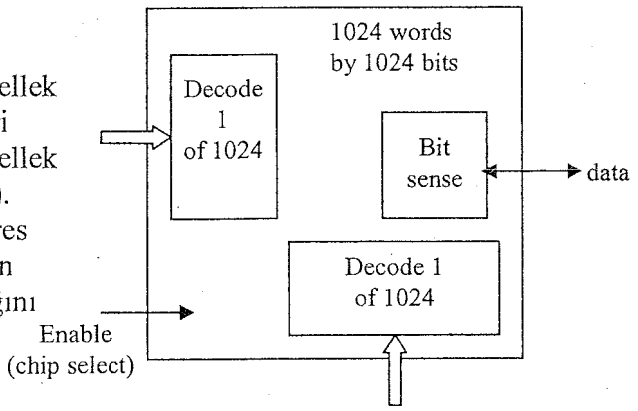
Bir DVD yapısı yukarıdaki şekilde gösterilmektedir. Anlaşılacağı gibi, DVD'nin her iki yüzüne de (her yüzde 2 katmana) kayıt yapılabilmektedir. Yapı 2 katmanlı olduğundan, alttaki katma optik anlamda erişebilmek için, üstteki katman yarıyansıtkan (semireflective) yapılmaktadır. Laser ışını her iki katmana da odaklanabilir ve hangi katmana odaklandıysa, o katmana erişilmiş olur (okuma-yazma anlamında). Sayısal anlamda yazma işlemi, tıpkı CD'lerde olduğu gibi, yüksek şiddette laser ışını kullanılarak ve polycarbonate maddenin üzerine yakılan noktalarda tümsekler (pit), yakılmayan noktalarda çukurlar (land)

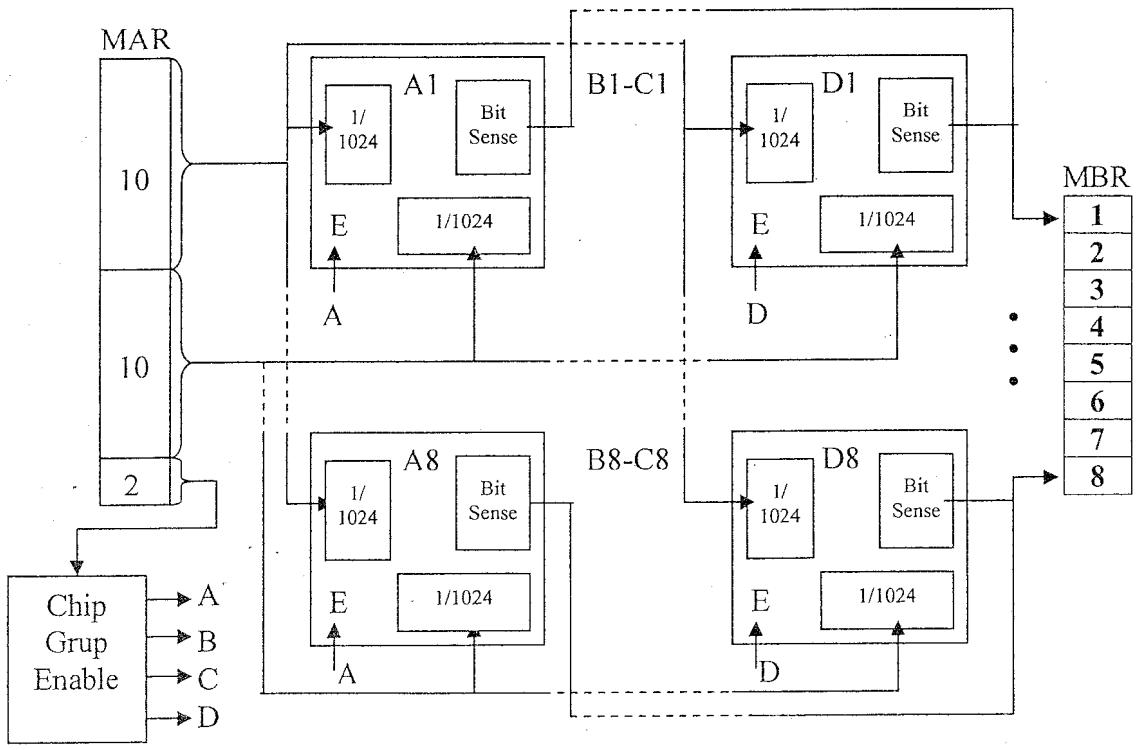
bırakmak suretiyle gerçekleşir. Okuma yapılırken daha düşük şiddette laser ışını kullanılır. Laser ışını çukurlardan daha güçlü, tümseklerden ise daha zayıf yansır ve bu yansıma şiddetindeki değişimler, sayısal veri olarak anlam kazanır (1-0). Okuma (yazma) işlemi, helezon şeklinde bir iz üzerinde, sabit doğrusal hız (Constant Linear Velocity, CLV) yöntemi uygulanarak gerçekleşir. Bunun için de dış kenara yaklaşıldıkça, diskin dönme hızı yavaşlatılır.

Bir DVD'nin toplam kapasitesi, CD'ninkinin yaklaşık 25 katıdır (680MB – 17GB). Bunun nedeni:

a) *DVD daha kısa dalga boylu laser ışını kullanır ve böylece DVD'ye daha yoğun veri kaydedilebilir.*
b) *DVD'nin bir yüzünde 2 katman verdır (CD'de ikinci bir katman yoktur)*
c) *DVD'nin her iki yüzüne de kayıt yapılır (CD'nin tek yüzü kullanılır)*

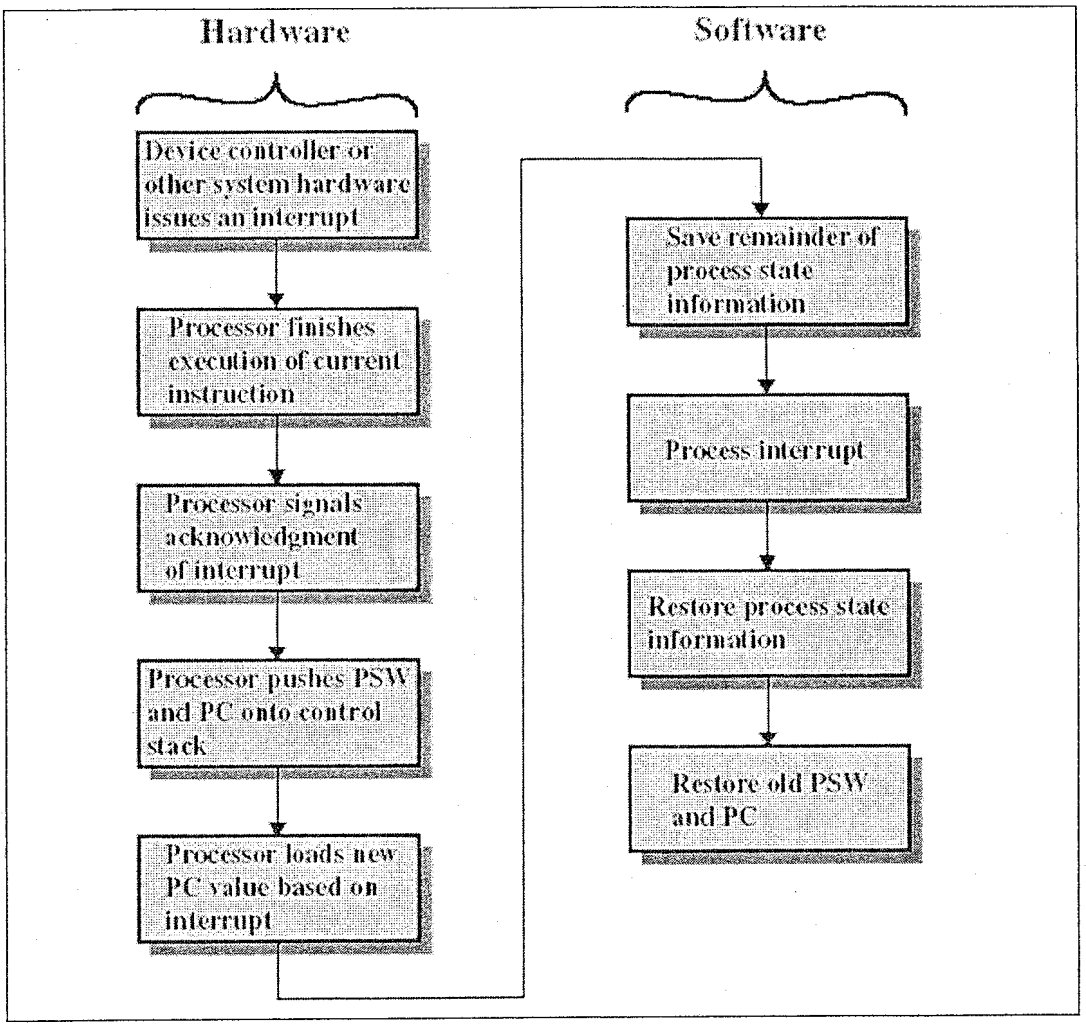
3. Yanda gösterilen (1024×1024×1) bellek yongalarından (memory chip) yeteri kadar kullanarak, 4Mbyte toplam bellek organizasyonu tasarlayınız (çiziniz). Tasarlayacağınız yapının bellek adres yazmacına (MAR) ve bellek tampon yazmacına (MBR) nasıl bağlanacağını gösteriniz.





4.

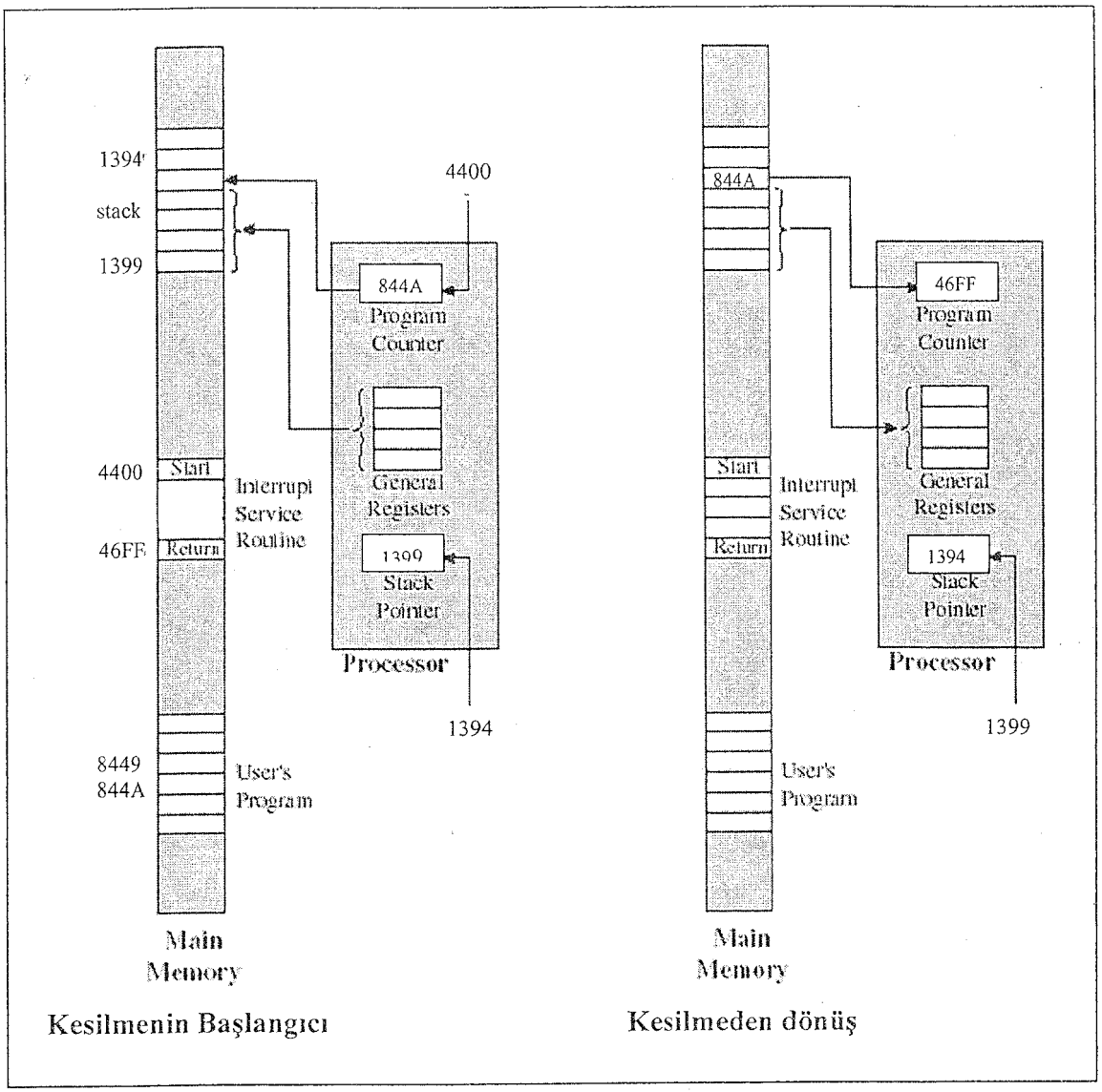
a) Kesilme sürümlü giriş/çıkış (interrupt–driven I/O) sürecini gerek donanım, gerekse yazılım boyutunda olup bitenler bakımından, bir işlem akış şeması düzeninde açıklayınız.



b) Kesilme talebi (interrupt request) geldiği anda, CPU'nun belleğin 8449 (hex) adresindeki program komutunu işlemekte olduğunu, kesilme servis yordamının (interrupt service routine) belleğin 4400 – 46FF adresleri arasında yer aldığını, ve yığıt göstergesinin (stack pointer) 1399 içerdiğini varsayarak:

- Bellek tarafında yığıt içeriğinin
- CPU tarafında PC (program counter), genel yazmaçlar ve yığıt göstergesinin

Kesilme talebinden hemen önce, kesilme talebi işleme konduğu anda ve kesilme servis yordamından dönüş anında nasıl değiştiğini şematik bir anlatımla açıklayınız.



PUANLAR:

Soru 1: Her şık 5p. Toplam 30p
Soru 2: Her şık 10p Toplam 30p
Soru 3: 20p
Soru 4: Her şık 10p Toplam 20p

SÜRE: 120 dakika