

BM-311 Bilgisayar Mimarisi

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

- **Hafıza sistemleri karakteristikleri**
- Hafıza hiyerarşisi
- Önbellek prensipleri
- Önbellek tasarım bileşenleri
 - Cache size
 - Mapping function
 - Replacement algorithms
 - Write policy
 - Line size
 - Number of caches

Hafıza sistemleri karakteristikleri

- **Location**
 - Internal (regs, cache, main memory)
 - External (optical disk, magnetic disk, tape)
- **Capacity**
 - Number of words
 - Number of bytes
- **Unit of transfer**
 - Word
 - Block
- **Access method**
 - Sequential
 - Direct
 - Random
 - Associative
- **Performance**
 - Access time
 - Cycle time
 - Transfer rate
- **Physical type**
 - Semiconductor
 - Magnetic
 - Optical
 - Magneto-optical
- **Physical characteristics**
 - Volatile/nonvolatile
 - Erasable/nonerasable
- **Organization**

Hafıza sistemleri karakteristikleri

Location

- **Internal:**
 - Register'lar, control unit memory, cache, main memory.
- **External:**
 - HDD, tape, I/O kontrol ile erişilenler.

Hafıza sistemleri karakteristikleri

Capacity

- **Number of words:**

- Toplam satır sayısını ifade eder.
- Genellikle main memory için kullanılır.
- Bir word 8, 16 veya 32 bit olabilir.

- **Number of bytes:**

- İkincil depolama birimlerinde kapasite byte olarak ifade edilir.

Hafıza sistemleri karakteristikleri

Unit of transfer

- **Word:**

- Bir seferde bir word boyutunda veri okunur veya yazılır.
- Main memory için kullanılır.

- **Block:**

- Bir seferde bir blok veri okunur veya yazılır.
- External memory için kullanılır.

Hafıza sistemleri karakteristikleri

Access method

- **Sequential:** Okuma /yazma mekanizması bulunulan konumdan istenen konuma kadar tüm kayıtları okuyarak gider. (Örn.: Tape)
- **Direct:** İstenen konuma doğrudan konumlanılır. İstenen konum okuma/yazma mekanizmasının altına gelene kadar beklenir. Erişim süresi önceki bulunulan konuma bağlıdır. (Örn.: HDD, CD)
- **Random:** İstenen konuma doğrudan gidilir. Erişim süresi önceki konuma bağlı değildir. (Örn.: Main memory)
- **Associative:** Arama adrese göre değil içeriğe göre yapılır. Aranılan veriyle tüm hafıza alanları eşzamanlı karşılaştırılır. (Örn.: Cache)

Hafıza sistemleri karakteristikleri

Performance

- **Access time (latency):**
 - RAM için adres bilgisinin verilmesinden verinin alınmasına/yazılmasına kadar geçen süredir.
 - Diğerleri için okuma/yazma mekanizmasının istenen konuma ulaşması için geçen süredir.
- **Memory cycle time:**
 - RAM için iki erişim süresi arasındaki toplam süredir.
- **Transfer rate:**
 - Veri aktarım hızıdır.
 - RAM için $(1/\text{cycle time}) \times \text{block size}$ ile ifade edilir.
 - Non-random access memory için $T_N = T_A + (n / R)$ ile ifade edilir.
 - T_A Ortalama erişim süresi; n bit sayısı ve R aktarım hızı (bps).
 - T_N , n adet bit için okuma veya yazma süresidir.



Hafıza sistemleri karakteristikleri

Physical type

- **Semiconductor:**
 - Random access memory'lerde kullanılır.
- **Magnetic:**
 - Disk ve tape ünitelerinde kullanılır.
- **Optical ve Magneto-optical:**
 - CD ve DVD'lerde kullanılır.



Hafıza sistemleri karakteristikleri

Physical characteristics

- **Volatile/nonvolatile:**
 - Elektrik kesildiğinde veri kaybolan (semiconductor)
 - Elektrik kesildiğinde veri kaybolmayan (magnetic-surface)
- **Erasable/nonerasable:**
 - İçeriği silinebilen (EEPROM)
 - İçeriği silinemeyen (ROM)



Hafıza sistemleri karakteristikleri

Organization

- Random access memory için word oluşturmak için bitlerin yerleşimini ifade eder (interleaved, sequential).



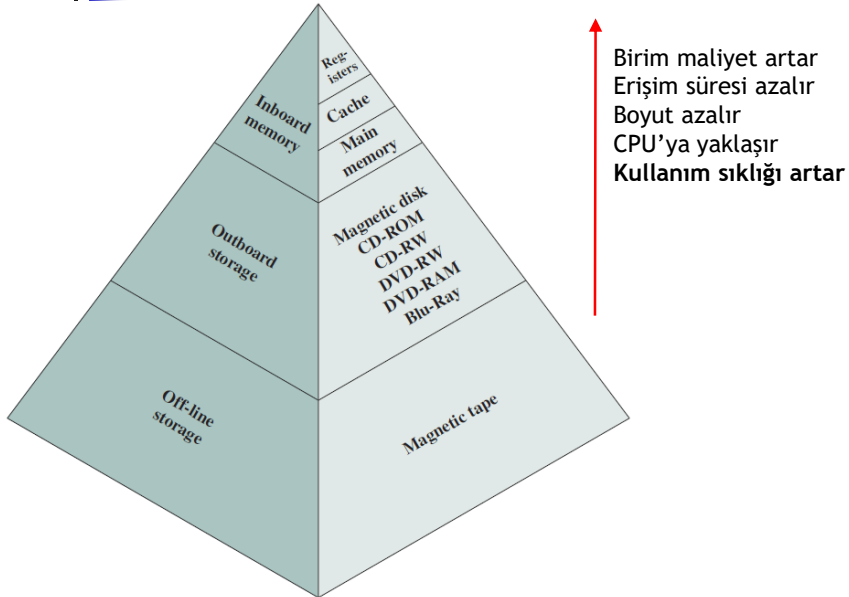
Konular

- Hafıza sistemleri karakteristikleri
- **Hafıza hiyerarşisi**
- Önbellek prensipleri
- Önbellek tasarım bileşenleri
 - Cache size
 - Mapping function
 - Replacement algorithms
 - Write policy
 - Line size
 - Number of caches

Hafıza hiyerarşisi

- **CPU tarafından** hafıza birimlerinde **erişim süresi kısaldıkça, bit başına maliyet artar.**
- Hafıza birimlerinde **kapasitesi arttıkça bit başına maliyet düşer.**
- Hafıza birimlerinde **kapasitesi arttıkça erişim süresi artar.**

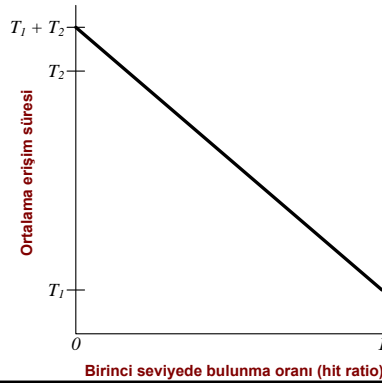
Hafıza hiyerarşisi



Hafıza hiyerarşisi

Örnek:

- Bir CPU iki seviyeli hafızaya erişiyor.
- **Birinci seviye 1000 word** ve **ikinci seviye 100.000 word** kapasitededir.
- Birinci seviye hafızaya **erişim süresi 0,01μs** ve ikinci seviye hafızaya **erişim süresi 0,1μs** dir.
- CPU ilk önce birinci seviyeye yoksa ikinci seviyeye erişmektedir.



Hafıza hiyerarşisi

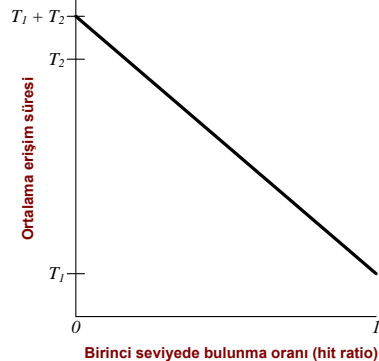
Örnek:

- Şekilde, verinin birinci seviye hafızada bulunma oranına göre erişim süresi görülmektedir. **Birinci seviye önbellegi ifade eder.**
- Verinin birinci seviyede bulunma oranı %95 ise ortalama erişim süresi nedir?

$$\begin{aligned}\text{Ortalama erişim süresi} &= (0,95)(0,01\mu\text{s}) + (0,05)(0,01\mu\text{s} + 0,1\mu\text{s}) \\ &= 0,0095 + 0,0055 = 0,015\mu\text{s}\end{aligned}$$

Ortalama erişim süresi **0,01μs** daha yakındır çünkü %95 oranında birinci seviyede bulunmuştur.

Şekilde, T_1 **birinci seviyeye**
 T_2 **ise ikinci seviyeye erişim süresidir.**



Hafıza hiyerarşisi

CPU'nun erişim sıklığına göre hafıza hiyerarşisi

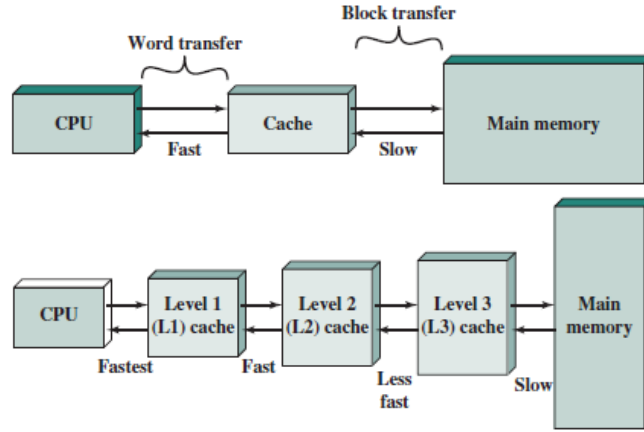
- Registers
- L1 Cache
- L2 Cache
- L3 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape

Konular

- Hafıza sistemleri karakteristikleri
- Hafıza hiyerarşisi
- **Önbellek prensipleri**
- Önbellek tasarım bileşenleri
 - Cache size
 - Mapping function
 - Replacement algorithms
 - Write policy
 - Line size
 - Number of caches

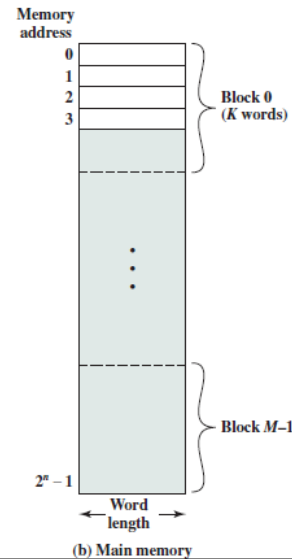
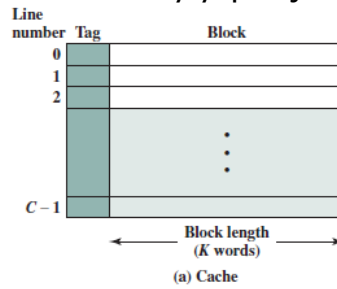
Önbellek prensipleri

- Önbellek, **main memory ile CPU arasına yerleştirilir.**
- CPU önce önbelleğe erişir, aranan veri yoksa main memory'ye erişir.
- Eğer aranan veri main memory'de ise içinde bulunduğu blok ile birlikte alınır. Önbelleğe ve CPU'ya aktarılır.



Önbellek prensipleri

- Önbellek ve main memory yapısı şekilde görülmektedir.



M blok sayısıdır ve
 $M = 2^n / K$ ile ifade edilir.

$C \ll M$ dir.

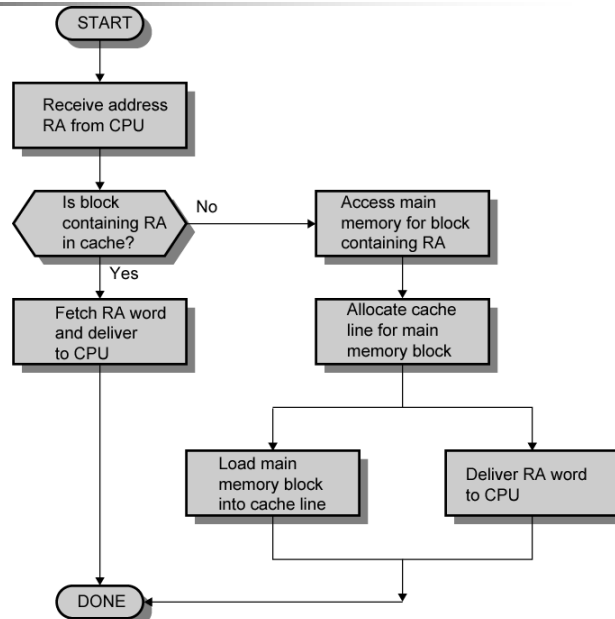
Önbellek prensipleri

Önbellek işlemi

- CPU bir adres içeriğini ister.
- Önbelleğe bakılır.
- Önbellekte bulunursa alınır.
- İstenen veri önbellekte yoksa hafızada içinde bulunduğu blok alınır ve önbelleğe aktarılır.
- Önbellekten CPU'ya aktarılır.

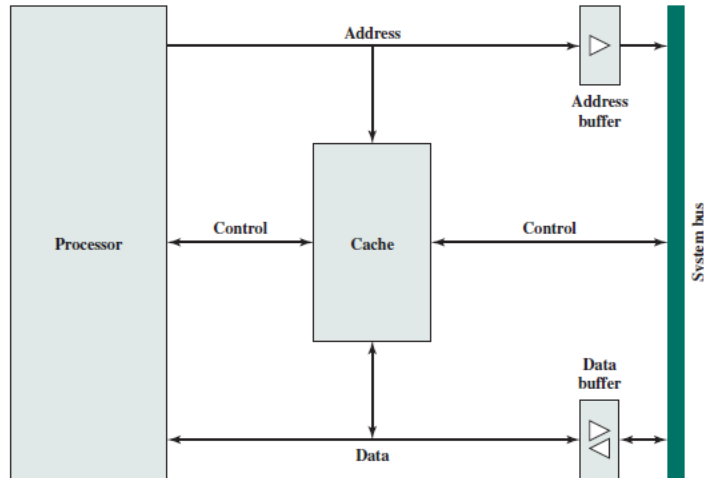
Önbellek prensipleri

Önbellek işlemi



Önbellek prensipleri

Önbellek organizasyonu



Konular

- Hafıza sistemleri karakteristikleri
- Hafıza hiyerarşisi
- Önbellek prensipleri
- **Önbellek tasarım bileşenleri**
 - Cache size
 - Mapping function
 - Replacement algorithms
 - Write policy
 - Line size
 - Number of caches

Önbellek tasarım bileşenleri

- **Cache size**
 - Cache satır boyutu
 - Cache satır sayısı
- **Mapping function**
 - Direct
 - Associative
 - Set associative
- **Replacement algorithms**
 - Least recently used (LRU)
 - First in first out (FIFO)
 - Least frequently used (LFU)
 - Random
- **Write policy**
 - Write through
 - Write back
 - Write once
- **Line size**
 - Satıra alınan blok boyutu
- **Number of caches**
 - Single or multilevel
 - Unified or split

Önbellek tasarım bileşenleri

Cache size

- **Önbellek boyutu azaldıkça** toplam **maliyet düşer.**
- **Önbellek boyutu arttıkça** hit oranı arttığı için **veriye erişim süresi düşer.**
- **Önbellek boyutu arttıkça** kullanılan **devre daha karmaşık hale gelir** ve az da olsa yavaşlama olur.

Önbellek tasarım bileşenleri

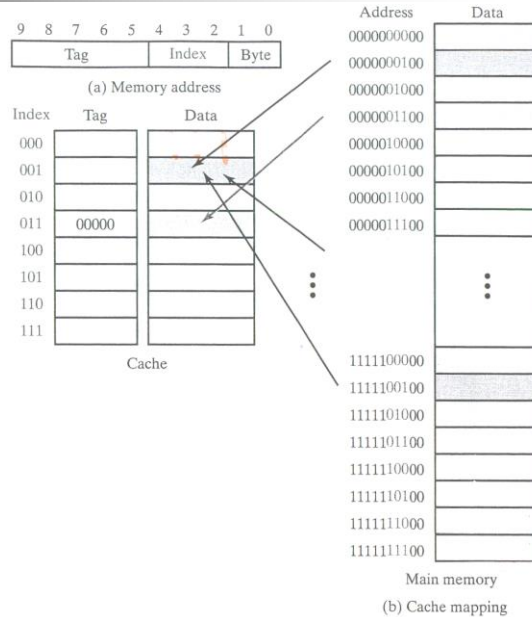
Mapping function

- **Önbellekteki satır sayısı main memory'den çok az** olduğu için eşleştirme fonksiyonu kullanılarak aktarma yapılır.
- Mapping function **hafızadaki bir bloğun önbelleğe nasıl yerleştirileceğini belirler.**
- **Direct, associative** ve **set associative** olarak üç yöntem kullanılır.

Önbellek tasarım bileşenleri

Direct mapping

- Şekilde önbellek satır boyutu 32-bit alınmıştır.
- Hafızada her blok 4-byte kapasitedir (Şekilde 1 satır).
- Önbellek adresleme 3-bit ile yapılmaktadır.
- Hafızadaki 256 blok önbellekteki 8 satıra eşleştirilmektedir.



Önbellek tasarım bileşenleri

Direct mapping – devam

- Eşleştirme modüler aritmetiğe göre yapılır.

$$i = j \bmod m$$

i = önbellek satır numarası

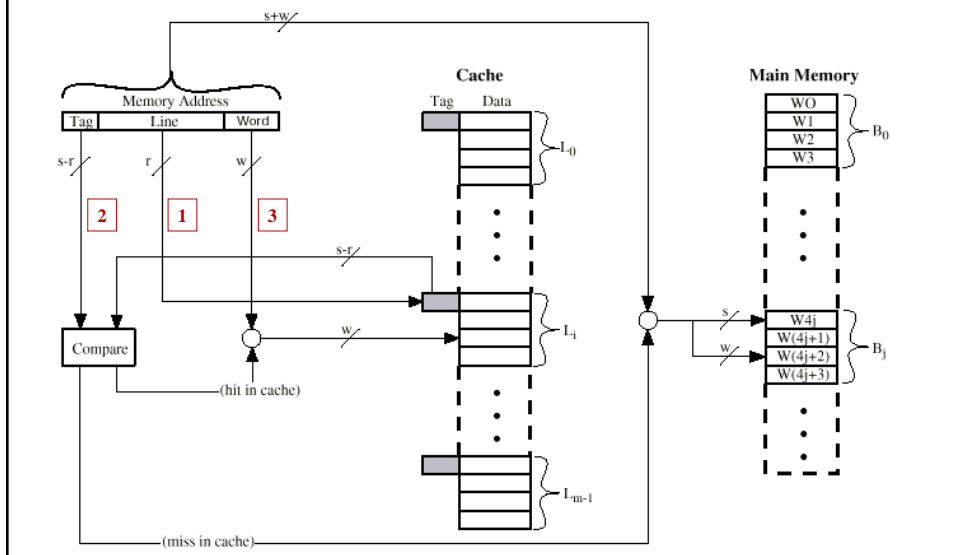
j = main memory blok numarası

m = önbellekteki satır sayısı

- Her hafıza adresi 3 alandan oluşur. **Tag**, **Line** ve **Word**.
- Line önbellekte satırı seçmek için, tag seçilen satırın etiketini belirtmek için word ise bulunan satırın bir elemanını seçmek için kullanılır.

Önbellek tasarım bileşenleri

Direct mapping – devam



Önbellek tasarım bileşenleri

Direct mapping – devam

- Adres boyutu = $(s+w)$ bit
- Adreslenebilir alan sayısı = 2^{s+w} word
- Blok boyutu = satır boyutu = 2^w word
- Hafızadaki blok sayısı = $(2^{s+w}) / (2^w) = 2^s$ word
- Önbellekteki satır sayısı = 2^r
- Önbellek boyutu = 2^{r+w}
- Tag boyutu = $(s - r)$ bit

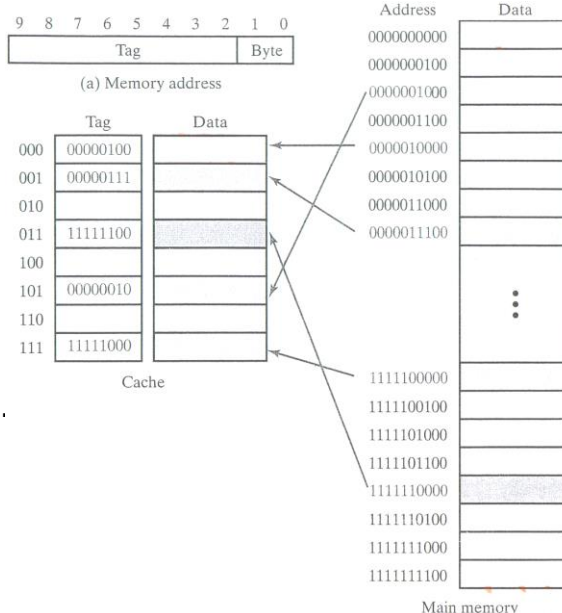
Avantaj / dezavantaj

- Oluşturmak basit ve ucuzdur.
- Bir blok sadece bir satıra yazılabilir.
- Aynı satıra eşleşen iki blok sürekli çalıştığında performans düşer.

Önbellek tasarım bileşenleri

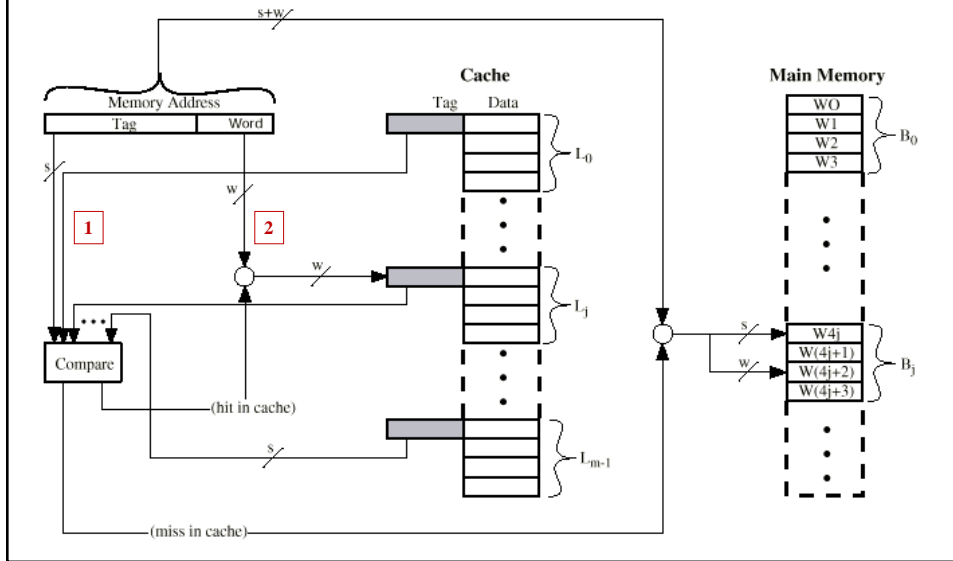
Associative mapping

- Direct mapping'teki bir bloğun sadece bir satıra eşleştirilmesi dezavantajı ortadan kaldırılır.
- Bir blok önbellekte istenilen satıra eşleştirilir.**
- Aranan bilginin önbellekte olup olmadığı tüm satırlarda eşzamanlı kontrol edilir (içerik arama).
- Hafızadan alınan bloğun önbellekte hangi satıra yazılacağı replacement algoritmasıyla belirlenir.



Önbellek tasarım bileşenleri

Associative mapping – devam



Önbellek tasarım bileşenleri

Associative mapping – devam

- Adres boyutu = $(s+w)$ bit
- Adreslenebilir alan sayısı = 2^{s+w} word
- Blok boyutu = satır boyutu = 2^w word
- Hafızadaki blok sayısı = $(2^{s+w}) / (2^w) = 2^s$ word
- Önbellekteki satır sayısı = tanımlı değil
- Tag boyutu = s bit

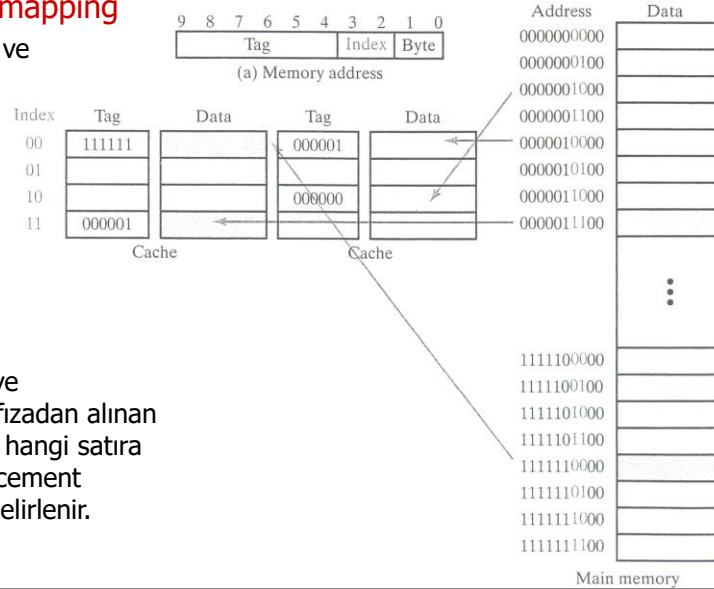
Avantaj / dezavantaj

- Yapısı karmaşıktır.
- Bir blok uygun olan bir satıra yazılabilir.
- Önbellekte eşzamanlı arama hızı düşüktür.

Önbellek tasarım bileşenleri

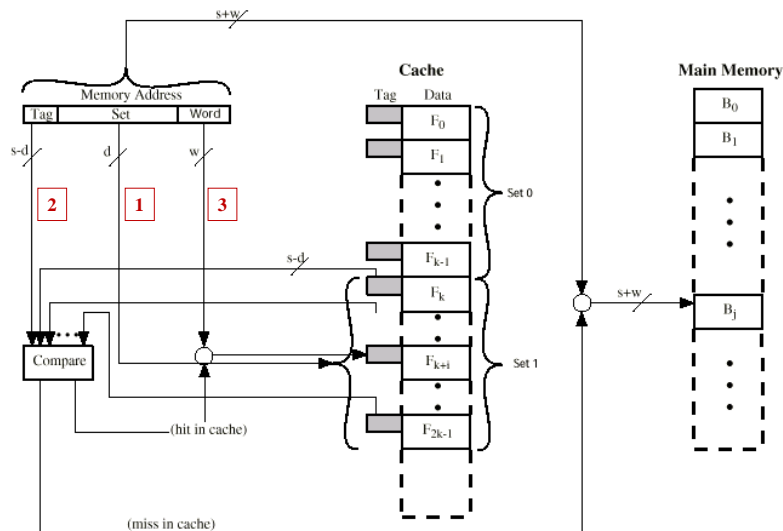
Set associative mapping

- Direct mapping ve associative mapping birleşimidir.
- Önbellek belirli sayıda kümeden oluşur.**
- Her küme kendi içinde associative yapıdadır ve hafızadan alınan bloğun kümede hangi satıra yazılacağı replacement algoritmasıyla belirlenir.



Önbellek tasarım bileşenleri

Set associative mapping – devam



Önbellek tasarım bileşenleri

Set associative mapping– devam

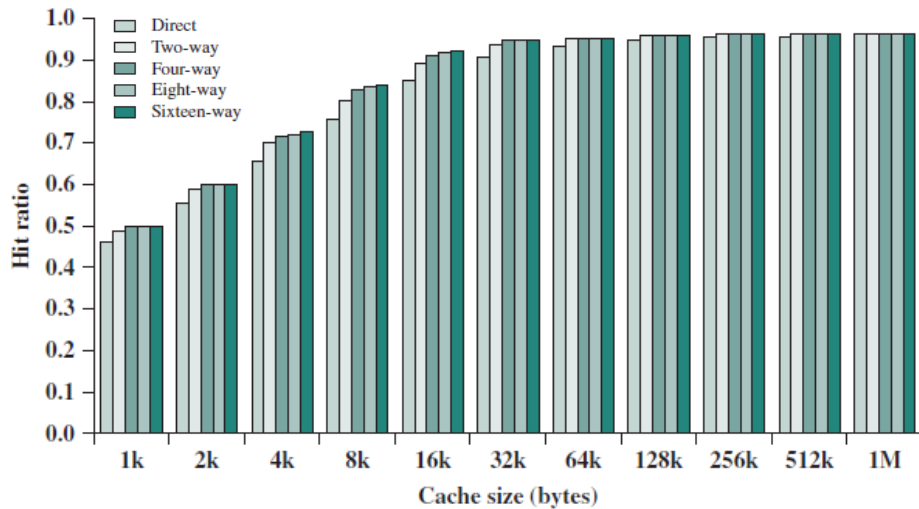
- Adres boyutu = $(s+w)$ bit
- Adreslenebilir alan sayısı = 2^{s+w} word
- Blok boyutu = satır boyutu = 2^w word
- Hafızadaki blok sayısı = $(2^{s+w}) / (2^w) = 2^s$ word
- Kümedeki satır sayısı = k
- Küme sayısı = 2^d
- Önbellekteki satır sayısı = $k \cdot 2^d$
- Tag boyutu = $(s-d)$ bit

Avantaj / dezavantaj

- Yapısı direct mapping'e göre karmaşıktır.
- Bir blok sadece kendisine ait bir kümedeki istenilen satıra yazılabilir.
- Önbellekte küme içinde eşzamanlı arama hızı düşüktür.

Önbellek tasarım bileşenleri

Cache boyutuna göre hit ratio (direct ve set-associative cache)



Önbellek tasarım bileşenleri

Replacement algorithms

- Direct mapping'te sadece bir satır seçilebildiği için replacement algoritması kullanılmaz.
- Associative ve set associative eşleştirmede replacement algoritmaları kullanılır.
- **Least recently used (LRU):**
 - En uzun süre kullanılmayan satıra yazılır.
 - Her satır için USE field kullanılır.
- **First in first out (FIFO):**
 - İlk gelen satıra yazılır.
- **Least frequently used (LFU):**
 - En az kullanılan satıra yazılır.
 - Her satır için counter kullanılır.
- **Random:**
 - Rastgele bir satır seçilir ve o satıra yazılır.

Önbellek tasarım bileşenleri

Write policy

- Önbellekteki veri değişmişse üzerine yeni veri yazılmadan önce hafızaya aktarılması gerekir.
- Eğer hafıza birden fazla cihaz tarafından ortak kullanılıyorsa, önbellekteki değişimin hafızaya aktarılma yöntemi çok önemlidir.
- **Write through:** Önbellekteki her yazma işlemi doğrudan hafızaya da aktarılır. Birden çok işlemcili sistemlerde bus sürekli izlenir ve değişiklikler güncellenir.
- Bus üzerindeki trafik fazladır.
- **Write back:** Önbellekteki bir veri değişir değişmez değil, sadece atılacağı zaman hafızaya yazılır. Her satır için UPDATE field kullanılır.
- Bus trafiği write through'a göre daha azdır.
- **Write once:** Birden çok işlemcili sistemlerde cache coherence için kullanılan protokoldür. İlk değişiklikler hemen yazılır diğerleri hemen yazılmaz.

Önbellek tasarım bileşenleri

Line size

- Önbellekte bir veri bulunamadığında, hafızadan sadece o veri değil bir blok alınır.
- Satır boyutu arttıkça ilk önceleri hit ratio artar daha sonra düşmeye başlar.
- Blok boyutu arttıkça yakın zamanda kullanılmayacak verilerde alınmaya başlar.

Önbellek tasarım bileşenleri

Number of caches

- Önbellekler birden fazla seviyede, veri ve komut için ayrı ayrı da oluşturulabilirler.

Multilevel caches

- On-chip ve off-chip olarak oluşturulabilir. L1, L2 ve L3 şeklinde üç seviyeli kullanımı vardır.
- On-chip önbellek CPU'nun external bus trafiğini azaltır.

Unified / split caches

- Split önbelleklerin bir kısmı komut için bir kısmı ise data için kullanılır.
- Unified önbelleklerde hit oranı yüksektir. Çünkü komut ve data arasındaki fetch yoğunluğuna göre kendini update eder.
- Unified önbelleklerde tek önbellek kullanıldığı için oluşturulması basittir.
- Split önbelleğin en önemli avantajı, instruction cycle'da instruction fetch/decode unit ile execution unit'i bağımsız hale getirir. **Pipelining için önemlidir.**



Ödev

- Multicore ve çok işlemcili sistemlerde cache coherence için kullanılan protokoller hakkında detaylı bir araştırma ödevi hazırlayınız.