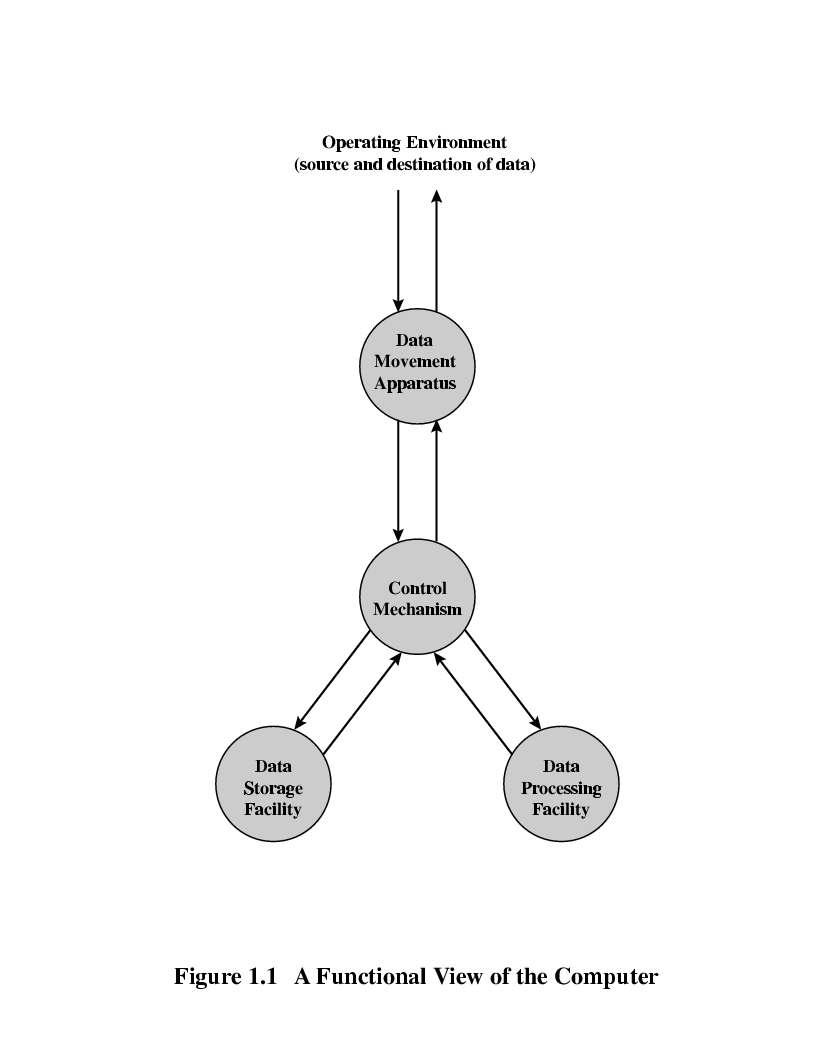
S-1 ) Bilgisayarların temel fonksiyonu nedir ve bunu nasıl gerçekleştirirler?

C-1) Bir bilgisayarın temel fonksiyonu istenen verilerin işlenmesi (proccesing) işlenen verinin hafızada saklanması ve istenilen anda hafızadaki veri üzerinde işlem yapabilme (storage), hafızasından veya hafızaya veri taşıma (movement), test ve bölüm klavuzları sayesinde data sayısı ve benzeri denetimleri yapmaktır (control).

Kısaca 4 ana fonksiyonu vardır:

- Data processing

- Data storage

- Data movement

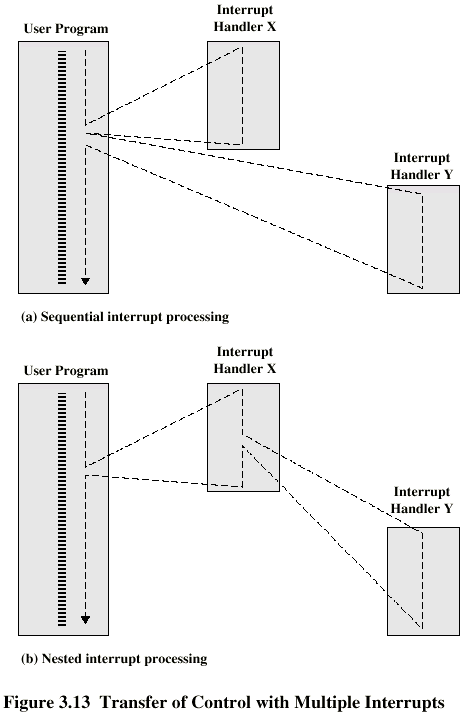
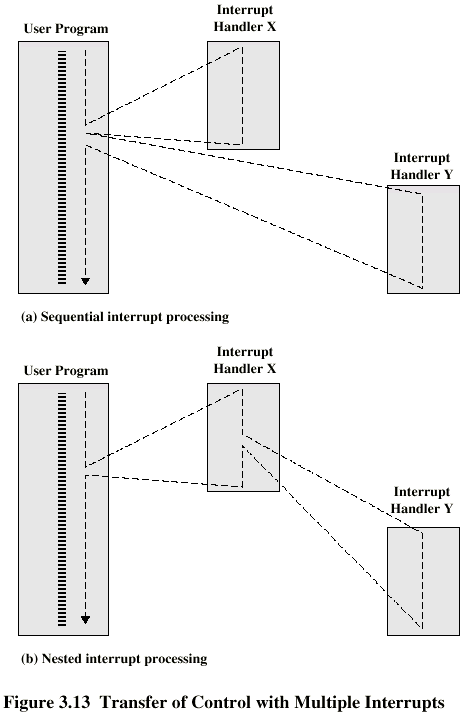
- Control

Bunu anabileşenleri sayesinde yaparlar. Bu ana bileşenler CPU, main memory, I/O devices, system interconnection olmak üzere 4 kısımda incelenir.

S-2 ) Kesme nedir, çoklu kesme nedir, bilgisayar fonksiyonunu gerçekleştirirken kesme olursa ne olur şekil çizerek açıklayınız.

C-2) CPU’nun komut işleme hızına göre çok yavaş olan ve ne zaman sonuçlanacağı belirlenemeyen (yazıcının bir döküm içini sonuçlandırılması gibi), ya da ne zaman gerçekleşeceği bilinmeyen (kullanıcının klavyede bir tuşa basması gibi) işlemlerin söz konusu olduğu durumlarda, bu durumun gerçekleşip gerçekleşmediğini bir program döngüsü, bir timer, I/O devices ya da harware sinyali olarak anlamak ve böylece döngü için harcanacak CPU zamanı başka işlerde kullanmayı olanaklı kılma işidir. Yani kısaca bir işlem yapılırken önceliği daha yüksek olan işlem geldiğinde önceliği düşük olan işlemin durdurulup önceliği yüksek olan işlem gerçekleştikten sonra ve önceliği düşük olan işleme devam edilmesidir.

Çoklu kesme ise zaten kesilmiş olan bir işlemin daha yüksek bir önceliğe sahip başka bir işlem ile kesilmesidir. Öncelikleri aynı ise birbirlerini kesmez sıra ile işlem yaparlar.



Çoklu Kesme

Sıralı Kesme

S-3) Bus nedir, mimari türleri nedir, açıklayınız.

C-3 ) Sistem kaynaklarını iyi kullanabilmek için bilgisayar içindeki bileşenler arasında veri taşıyan kanallara bus (veri yolu) denir. Bus 8,16,32 ve 64 bir gibi kapasitelere sahiptir.

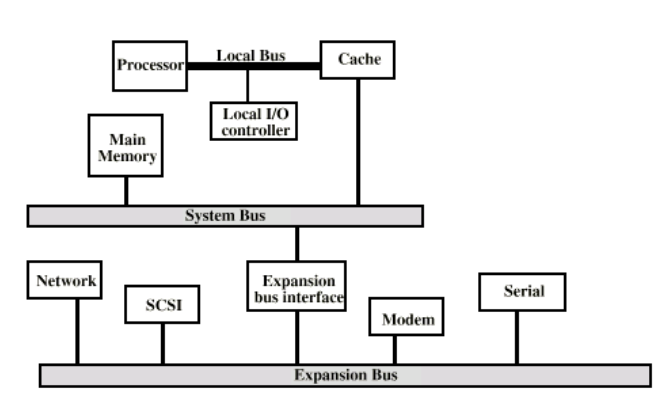
Bus Yapısı belirli anda yalnızca bir birim tarafından kullanılabileceği için söz konusu ortak kullanım anca zaman paylaşımı ile olanaklıdır. Hanngi birimin öncelik ile ve ne zaman bus yapısını kullanabileceği “bus arbitaration” tarafından belirlenir. Bus mimarisi iki “geleneksel” ve “yükse hızlı” olmak üzere iki şekilde incelenir. Burada önemli olan kullanılan bus tipleri ve deviceların bu buslar üzerindeki yerleşimidir. Performans soruunlarına göre gruplanarak farklı buslar üzerine koymak uygun görülmüştür.

Local Bus: Genellikle en yüksek trafiği yaratan CPU ve Cache erişimine ayrılır.

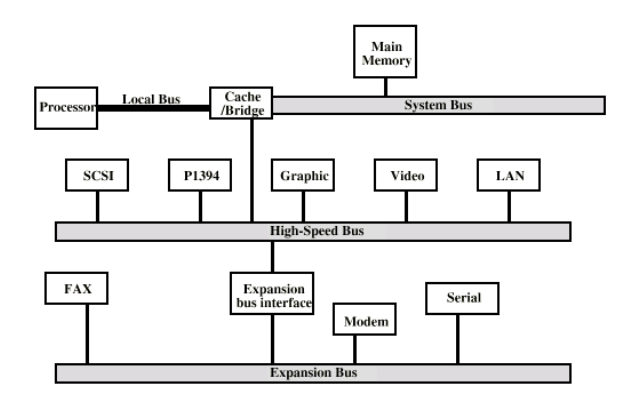
System Bus: İkinci yüksek trafiği yaratan ana bellek erişimine ayrılır.

Expension Bus: Bunların dışında geri kalan birimler için ayrılmıştır.

High Speed Bus: Yüksek hızlı bus mimarisi gibi bazı yapılarda öncelik amacı ile kullanılır.



Geleneksel BUS mimarisi



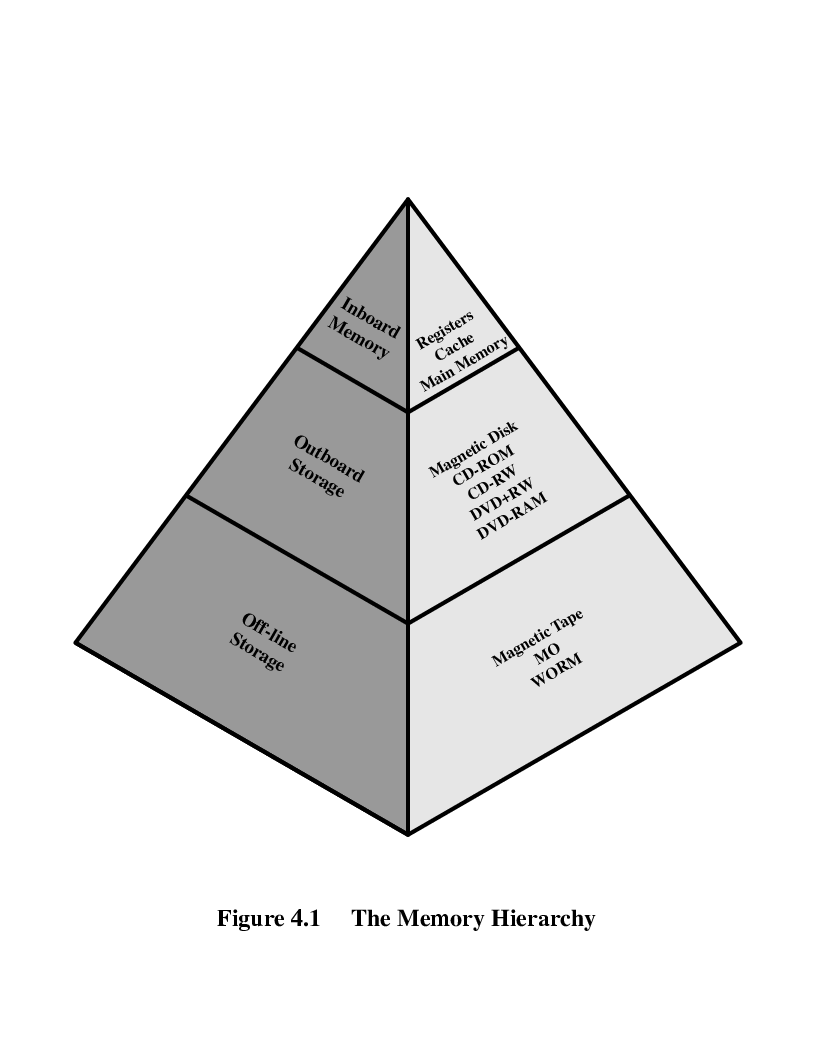
Yüksek hızlı BUS mimarisi

S-4 ) Bilgisayar memory hiyerarşisini çiziniz avantaj ve dezvantajları ile açıklayınız.

C-4 ) Belek teknolojilerinin günümüzdeki konumunda şu özellikler bulunmaktadır.

* Erişim hızı arttıkça, bit başına düşen maliyet artmaktadır.
* Kapasite artıkça, bit başına düşen maliyet azalmaktadır.
* Kapasite arttıkça, erişim hızı azalmaktadır.

Kullanıcılar büyük bellek kapasitesinden yanadır çünkü hem kapasite gereksinimleri vardır hem de maliyet düşük olmaktadır. Fakat performansta önemlidir. Bunun için de daha düşük kapasiteli bellekler gerekmektedir. Bu çıkmazın çözümü belek hiyerarşisi uygulamasında bulunmuştur.



Birim maliyet artar

Erişim süresi azalır

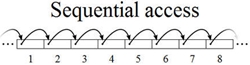
Boyut azalır

CPU’ya yaklaşır

Kullanım sıklığı artar

S-5) Access metodlarını açıklayınız ve her biri için matematiksel örnek veriniz.

c-5) Access methodları

Sequential: Okuma /yazma mekanizması bulunulan konumdan istenen konuma kadar tüm kayıtları okuyarak gider. (Örn:. Tape)

100000 kayıt, her kayıt 400 bytes, block size 2400 bytes. Sequential search ile 10000 tane kayıta bakmak için gerekli süre nedir

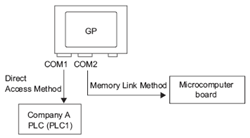
- Her probe da bir block okunabilir

- (100000\*400)/2400 = 16667 block

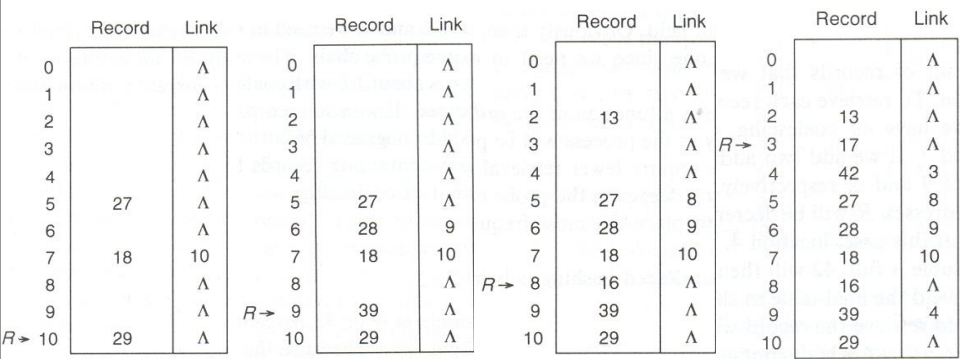
- 0.84ms bir block için okuma süresi (IBM 3380)

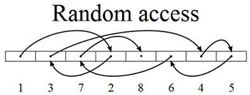
- Her kayıt için gerekli süre (16667/2)\*0.84 = 7 sec.

- 10000 kayıt için gerekli süre 7sec \* 10000 = 19 saat

Direct: İstenen konuma doğrudan konumlanılır. İstenen konum okuma/yazma mekanizmasının altına gelene kadar beklenir. Erişim süresi önceki bulunulan konuma bağlıdır. (Örn:. HDD, CD)

Hash (key) = key mod 11 olacak şekilde sayıları (27, 18, 29, 28, 39, 13, 16) atayınız.

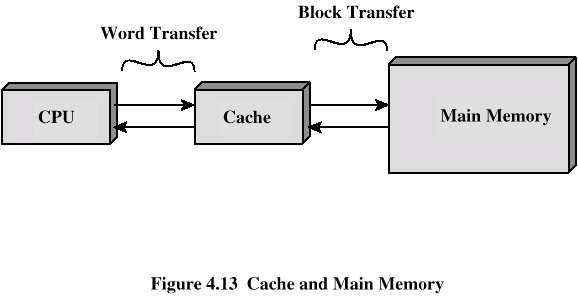


Random: İstenen konuma doğrudan gidilir. Erişim süresi önceki konuma bağlı değildir. (Örn:. Main memory)

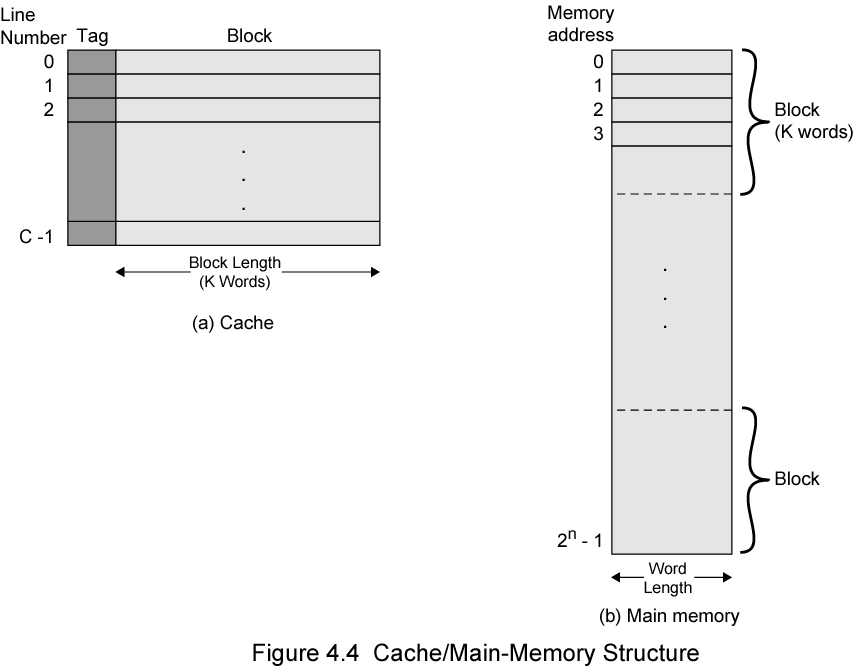
Asociative: Arama adrese göre değil içeriğe göre yapılır. Aranan veriyle tüm hafıza alanları eşzamanlı karşılaştırılır. (Örn:. Cache)

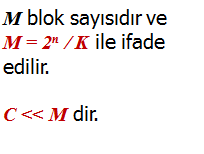
S-6) Ön bellek nedir, çalışma prensibi nasıldır.

C-6 ) Cache (ön belek ) kullanımındaki temel amaç pahalı hızlı ve küçük bir bellek alanına ucuz, yavaş ve büyük bir bellek alanının en sık erişilmek istenen kopyaları bulundurmak, böylece bir yandan büyük bir bellek kapasitesine daha ucuz bellek yongaları ile sağlarken diğer yandan pahalı bellek yongalarının hızına yakın bir ortalama performans elde etmektir. Ya da, bellekte en son erişilen adresteki ve bu adrese yakın adresteki veriye erişim ihtiyacının yüksek bir olasılıkla bir süre daha devam etmesi olgusu gerçekleşmesi oranında yarar sağlar. Buna göre okunacak veri önce önbellek alanında aranır olursa ana belleğe erişim tekrar ulaşıma gerek kalmaz. Bulunmazsa aralarında okunacak verinin de bulunduğu küçük bir blok ana bellekten cache belleğe kopyalanır.



Cache ile memory arasındaki ilişki

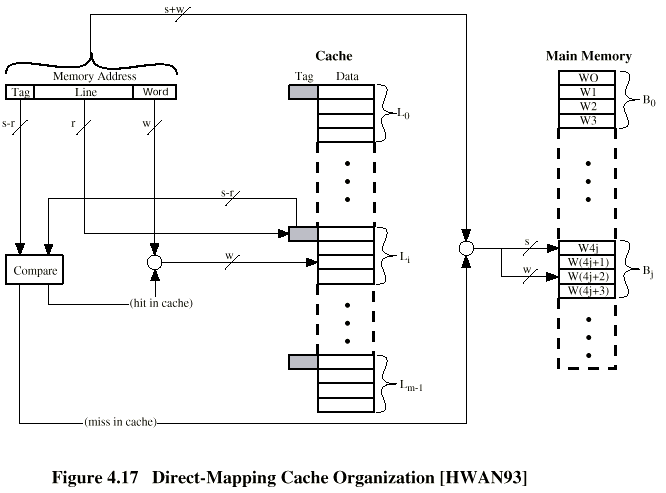


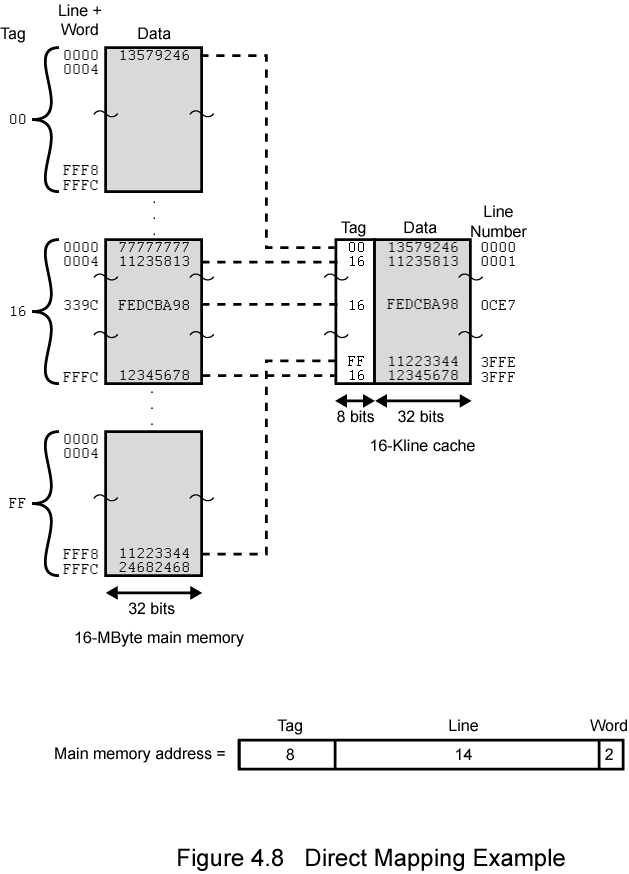


S-7) Ön bellek mapping nedir, çeşitleri nelerdir, her birini örnekle açıklayınız.

C-7 ) Ön bellek mapping, main memoryden cache gelecek olan verinin belirli yöntemlerle aktarılmasıdır. Bu haritalama işlemi direct ve associative olarak 2 farklı yol ile yapılır. Associative 2 farklı yönteme ayrılır.

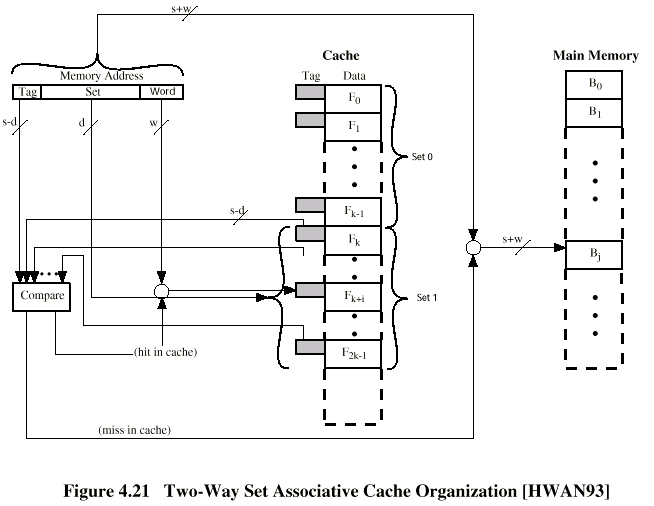
- Direct mapping



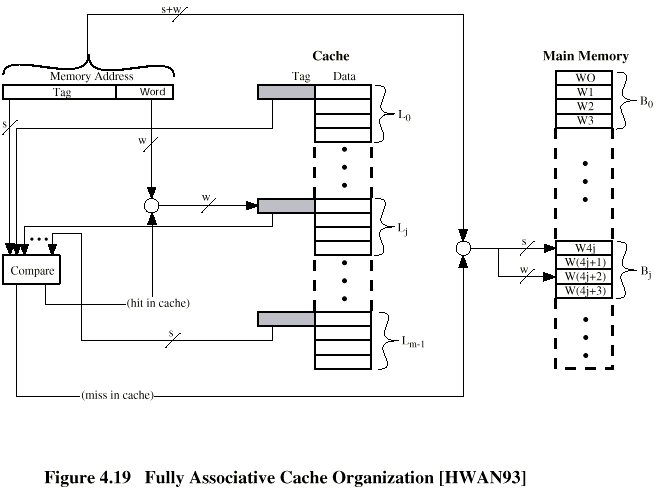


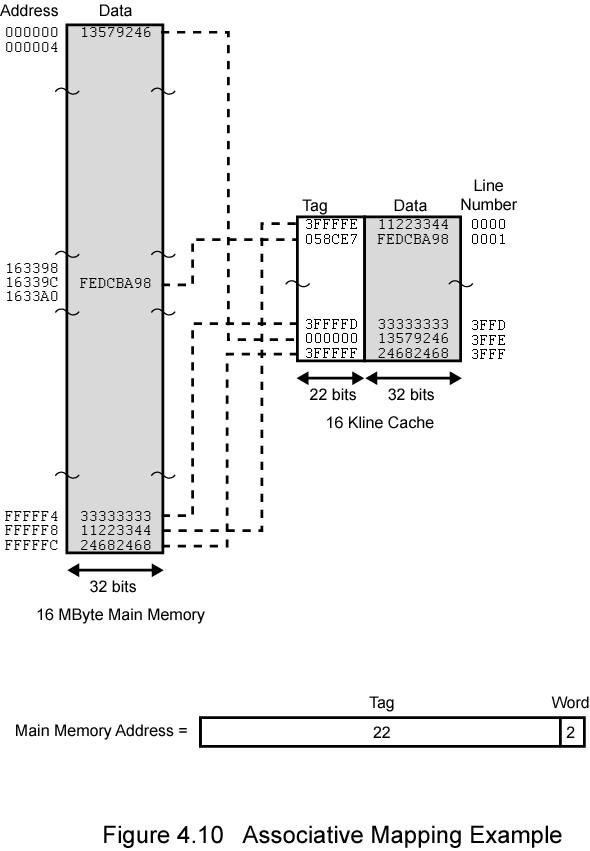
Direct Mapping  
 Örneği

- k-way set associative mapping



- Fully associative mapping





Associative   
Mapping Örneği