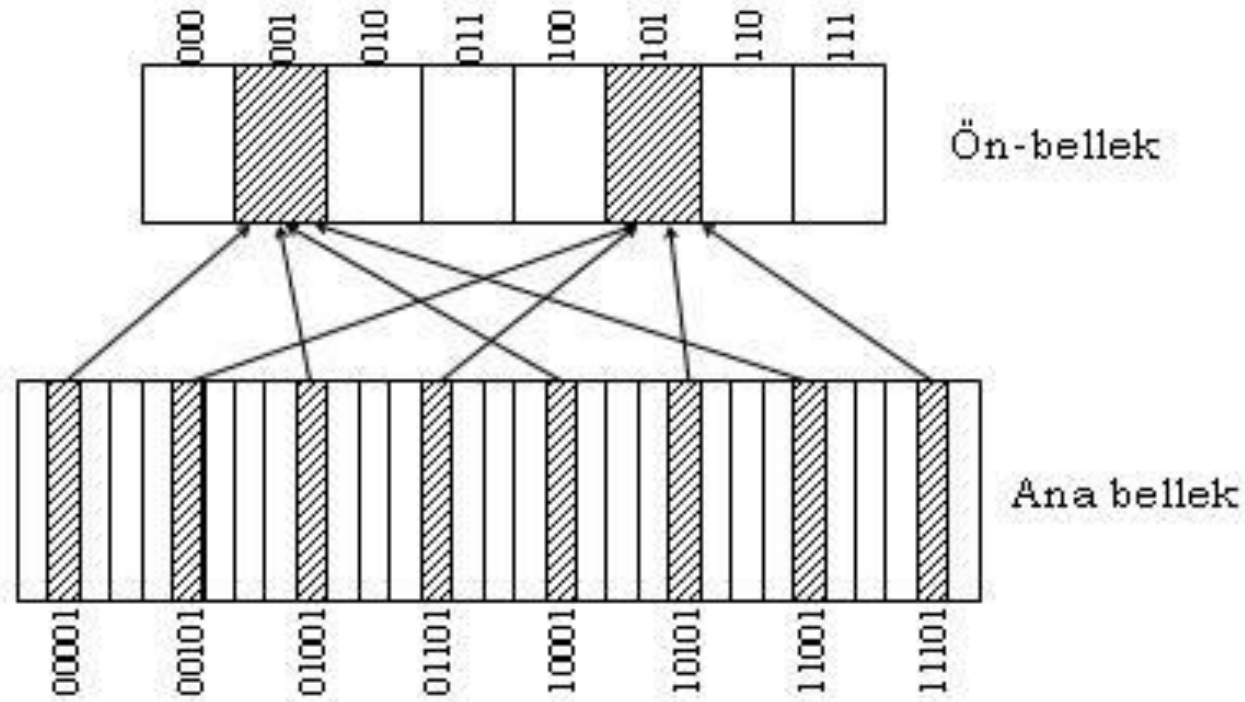


# BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU



## 11. BÖLÜM - DİREKT HARİTALAMA YÖNTEMİ

# Direkt Haritalama Yöntemi



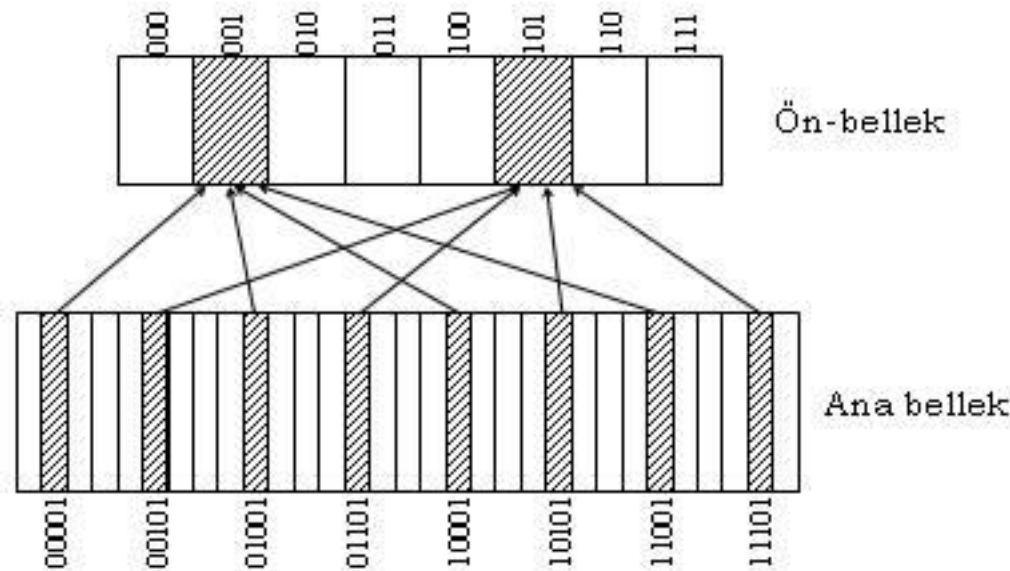
1



❖ İşlemci enformasyon (bilgi) aradığında önce ön-belleğe başvurur. Peki aranan bu enformasyonun ön-bellekte olup olmadığını ve böyle bir enformasyon varsa, onun aranan olup olmadığını nasıl belirlemektedir? İşlemci enformasyonu, onun adresine göre aramaktadır. Ön-bellekte ise böyle bir adresleme yöntemi kullanılmaz. Direkt haritalama yönteminde ana belleğin belirli adresindeki enformasyon grubu (buna blok veya modülde denir) ön-belleğin belirli grupları (blok veya modül) arasında bir uygunluk oluşturulur. Hem de bu uygunluk  $2^m$  modülüne göre oluşturulur ki, burada  $m$  ön-bellekte oluşturulan blokların sayısını göstermektedir. Burada dikkat edilmesi gereken husus odur ki ön-belleğin bir bloğundaki veriler, ana bellekteki tek bir bloktan değil çeşitli bloklardan getirmektedir.

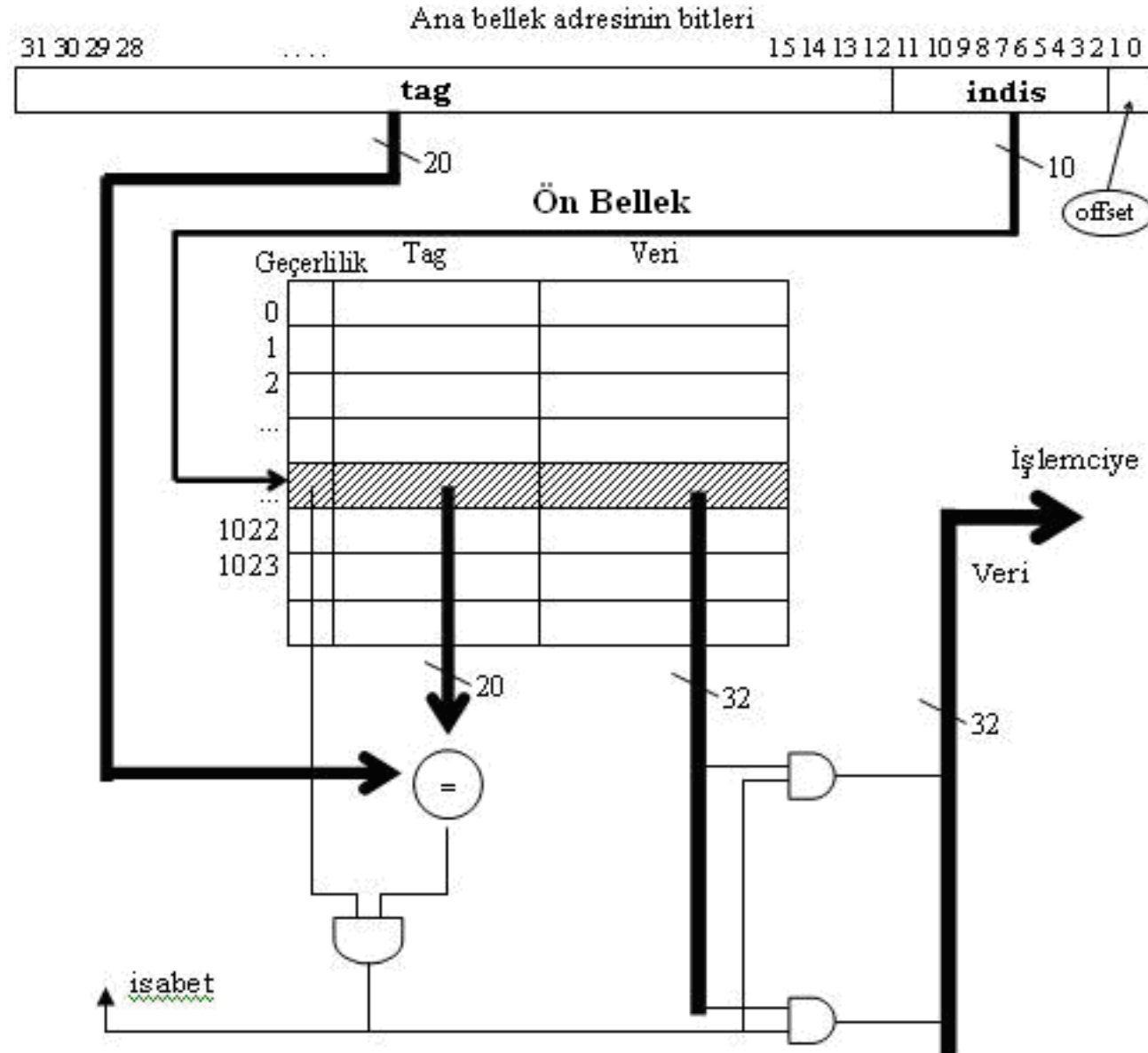


- ❖ Böylece her bir ön-bellek adresi çeşitli ana bellek adreslerinden gelen içerikleri depolayabilir. Bu durumda ön bellekteki içeriğin talep edilen enformasyon olup olmadığını belirlemek için tag olarak adlanan ek enformasyon kullanılır. Bütün bunları daha iyi anlamak için bir örneğe bakalım. Ön-belleğin kapasitesi 8 (bayt, kelime vs.) Ana belleğin 00001 ile 11101 ( $29_{10}$ ) adresleri arasındaki bloğun ön-belleğe nasıl haritalanacağına bakalım. 30 adres söz konusu olduğundan haritalama  $\log_2 30 = 5$  ön-bellek adresine yapılmaktadır.





❖ Şekilde ön-bellekle ana belleğin arasındaki ilişki gösterilmektedir. Şekilden de görüldüğü gibi ana bellek adresi 32 bit olduğunda bunun ilk 2 biti bayt offsetini, sonraki 10 bit ön-bellekteki verinin yerini belirleyen adresini ve daha sonraki 20 bit ise tag için ayrılan kısmını ifade etmektedir.





❖ Ön-belleğin tag kısmı ana bellek adresinin son 20 bitinden teşkil edilmiştir. Ön-belleğin  $2^{10}=1024$  blok kapasitesi ve 1 bloğun 1 kelimedenden oluştuğu varsayılmaktadır. Bu yüzden ön-bellekteki bloklara erişmek için 10 bit yeterli olmaktadır. Ana bellek adresinin indis olarak da adlanan kısmı ön-bellekteki bloğu seçmek için kullanılır. Ön-bellekteki bir adres seçildiğinde bu adresteki tag, ana bellek adresinin son 20 biti ile karşılaştırılır. Bunların her ikisi aynı olduğunda ve geçerlilik biti de “1” olduğunda ön-belleğe müracaat isabetli sayılır. Bu durumda ön-belleğin aynı adresindeki 32 bit veri okunur ve işlemcide işleme sokulur. Aksi durumda ıskalama söz konusudur ve isabet sinyali “0” olacağından veri okunamıyor (uygun “ve” kapısı kapalıdır).





❖ Ön-bellek için talep edilen bitlerin genel sayısı ön-belleğin kapasitesinin ve adres uzunluğunun bir fonksiyonudur. Adres uzunluğu  $m$  bit, ön-bellek kapasitesi  $2^n$  kelime olduğunda tag için gereken uzunluk  $m-(n+2)$  bit olacaktır. Buradaki 2 bit bir kelimedeki baytı göstermek için kullanılmaktadır. Bir kelimedeki 32 bit olduğu varsayılırsa bir ön-belleğin kapasitesi:

$$K=2^n (32 + (m - n - 2) + 1) \text{ olur.}$$

Eğer  $m=32$  kabul edersek;

$$K=2^n \cdot (32 + (32 - n - 2) + 1) = 2^n \cdot (63 - n) \text{ olur.}$$

$N=14$  olduğunda ön-belleğin kapasitesi ne kadar olmalıdır?

$$K = 2^{14} (32 + (32 - 14 - 2) + 1) = 2^{14} \cdot 49 = 784 \cdot 2^{10} = 784 \text{ KB}$$

# Örnek Problem -1



6



**ÖRNEK-1)** Belleğin bir kelimesinin uzunluğu **16 Byte**, bir bloğun kapasitesi **4 kelimedir**. Esas belleğin kapasitesi **256 KB** ve ön belleğin kapasitesi **16 KB**'tır.

- a)** Önbellek blok sayısını ve esas bellek blok sayısını bulunuz.
- b)** Direk haritalama yöntemine göre, önbellek blok adresi **8**, tag numarası **6** olan bloğun esas bellek blok adresini hesaplayınız?

# Örnek Problem -2



**SON**



**ÖRNEK-2)** Belleğin bir kelimesinin uzunluğu **8 Byte**, bir bloğun kapasitesi **16 kelime** olan bir sistemde; önbelleğin kapasitesi **8 KB** ve esas belleğin kapasitesi **128 KB**'tır.

- a)** Önbellek blok sayısını ve esas bellek blok sayısını bulunuz.
- b)** Esas bellek bloğunun adresi **MB=641** olan blok direk haritalama yöntemine göre önbelleğin kaçınıcı bloğunda yedeklenir ve tag numarası hesaplayınız?