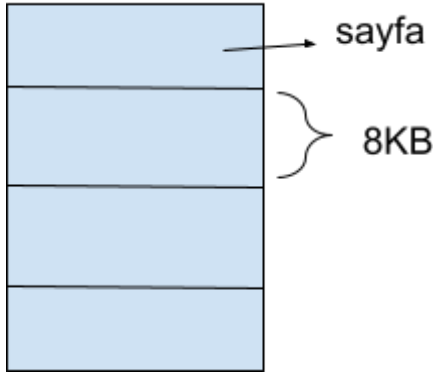


1.) Sanal bellek kullanan bir işlemcide: veri öbeği genişliği 8 bayt, bellek uzayı 4GB ve bir sayfanın büyüklüğü 8KB'tır. Bu işlemcinin önbelleğe tek vuruşta erişebildiğini biliyoruz.

- bu işlemcinin kullandığı adres bitlerini, boyutlarını ve nedenlerini detaylandırarak gösteriniz.(offset, dizin, etiket kaç bittir? neden? bir yazmaç üzerinde çizip gösterin)
- bu işlemci, eğer doğrudan eşlemeli bir önbellek kullanıyorsa, önbellek boyutu ne olur? Kaç adet karşılaştırıcı kullanılır ve bu karşılaştırıcıların girişleri kaç bittir? Önbelleği çizerek anlatınız.
- bu işlemci, 4 yollu kümeli ilişkili önbellek kullanırsa, önbellek boyutu ne olur? Kaç adet karşılaştırıcı kullanılır ve bu karşılaştırıcıların girişleri kaç bittir? Önbelleği çizerek anlatınız.
- bu işlemcide en fazla kaç adet sayfa bulunabilir?

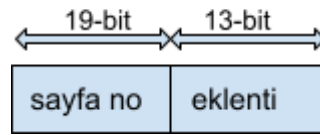
**Çözüm:**

bellek uzayı 4GB  $\rightarrow 2^{32}$  bayt  $\rightarrow$  32-bit adresleme yapıyor. (4096 MB\*1024 KB\*1024 bayt)



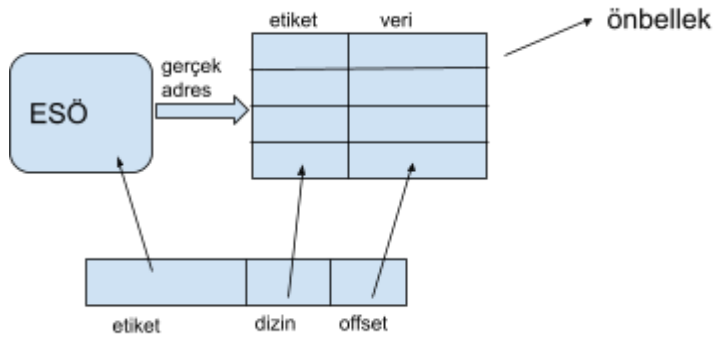
sayfa büyüklüğü = 8KB  $\rightarrow 2^{13}$  bayt (1024\*8 bayt) (sayfa eklentisi offset)

toplam sayfa sayısı =  $\frac{4GB}{8KB} = 2^{19}$  bayt



Adresin yorumlanması(sanal bellek açısından)  $\rightarrow$

İşlemcinin önbelleğe tek vuruşta erişebilmesi için aynı saat vuruşunun başında hem ESÖ'ye hem de önbelleğe erişmesi gerekir. Adresin etiket kısmı ESÖ'ye gönderilecek, ESÖ'den gerçek etiket adresi alınıp, önbellekteki ilgili satırın etiketiyle karşılaştırılacak.



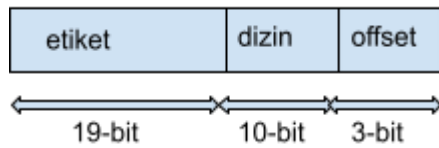
Etiket uzunluğu, sayfa numarası bit uzunluğuna eşit olmalı !

etiket = 19-bit

veri öbeği = 8 bayt → offset = 3-bit

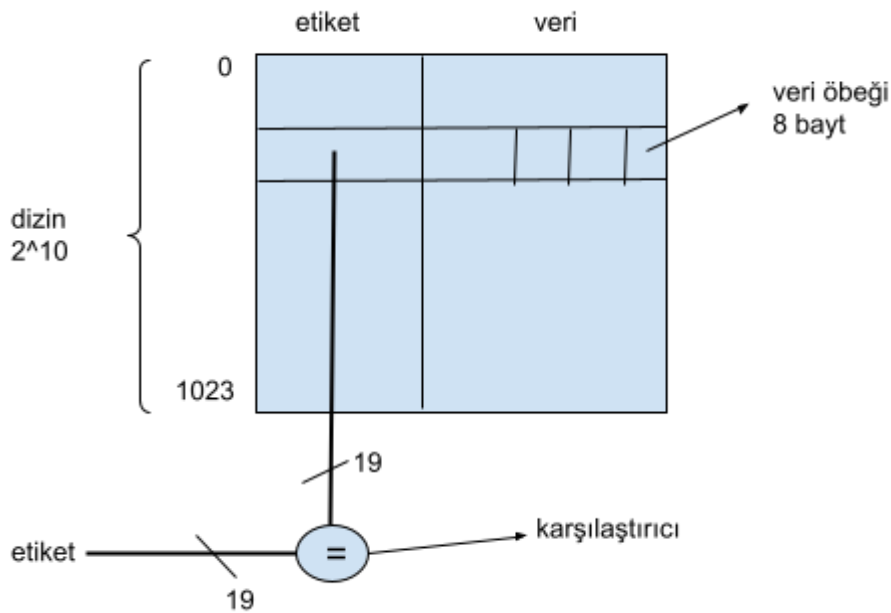
dizin = 32 - 19 - 3 = 10-bit

a.)

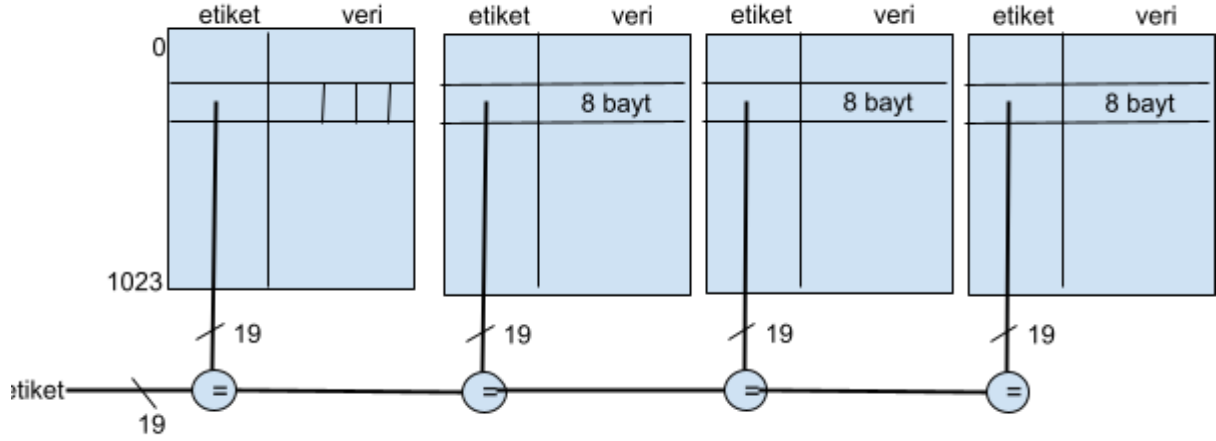


b.) doğrudan eşlemeli kullanılırsa önbellek boyutu  $2^{13} = 8 \text{ KB}$

1 adet karşılaştırıcı kullanılır ve 19-bitliktir.



c.) Adresleme bölümlerinde hiçbir değişiklik olmaz. Boyut  $4 \times 8 \text{ KB} = 32 \text{ KB}$  olur. 4 adet karşılaştırıcı devre kullanılır. Devrenin girişleri 19-bit olur.



d.) sayfa sayısı =  $\frac{4 \text{ GB}(\text{toplam bellek})}{8 \text{ KB}(\text{sayfa büyüklüğü})} = 2^{19}$  adet sayfa bulunur.

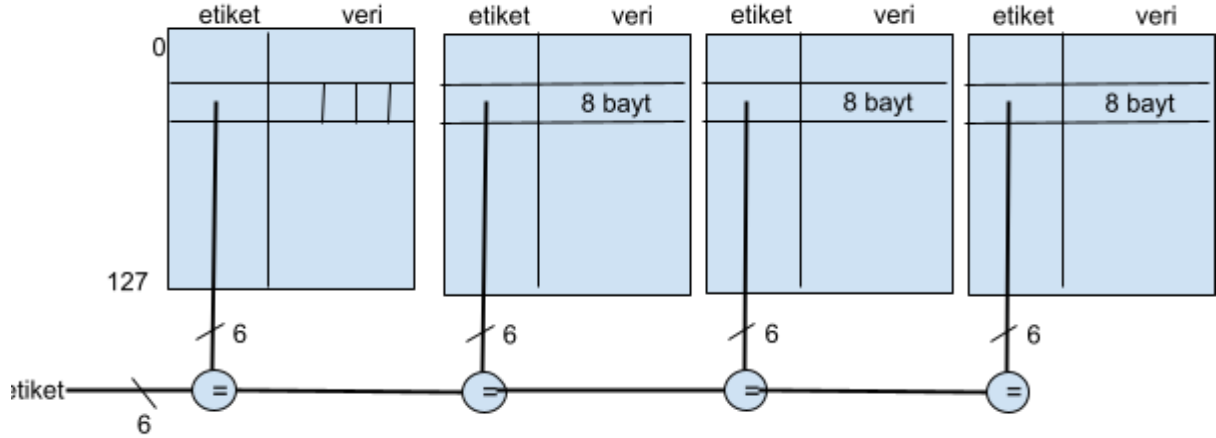
2.) 8-bitlik verilerle işlem yapan bir işlemcide aşağıdaki kod bloğu **120 defa çalışacak** şekilde ve dolaylı adresleme yapacak şekilde (bir yazmacın değeri ve bir offset değerinin toplanması ile adrese erişiliyor) kodlanmıştır. 16 bitlik adresleme kullanan bir işlemcide 4 kilobaytlık dört yollu kümeli ilişkili bir önbellek yapısı kullanılmaktadır ve veri öbeği 8 bayttır. R1 yazmacı 8-bit, R2 ve R3 yazmaçları ise 16-bit genişliğindedir. offset herhangi bir sabit sayıdır. Ana bellekten veri getirilmek istendiği zaman, veri önce önbelleğe getirilmekte, daha sonra önbellekten veriye ulaşabilmektedir. Yani ana belleğe erişim sadece önbellekten yapılıyor, işlemci de sadece önbellekten veri okuyabiliyor. Ana belleğe erişim süresi(verileri ana bellekten önbelleğe getirmek, ya da önbellekten ana belleğe götürmek) 100 saat vuruşu, önbelleğe erişim(veriyi önbellekten yazmaca aktarmak, ya da yazmaçtan önbelleğe aktarmak) 2 saat vuruşu kadar sürmektedir. Sanal bellek kullanılmaktadır.

- Önbelleğin yapısını detaylı çizip bileşenleri açıklayınız(kaç adet kaç bitlik karşılaştırmacı devre var, etiket genişlikleri, adresleme yaparken hangi bitler neleri ifade ediyor...).
- Load işlemi için kaç defa önbelleğe, kaç defa ana belleğe erişim gerçekleşmiştir? Load işlemi içi BBÇ nedir?

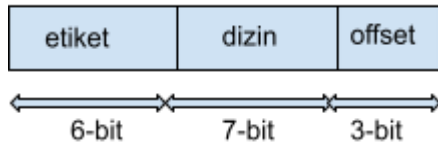
```
Load R1 ← mem[R2 + offset]
Store R1 → mem[R3 + offset]
R2 ← R2 + 1
R3 ← R3 + 1
```

**Çözüm:**

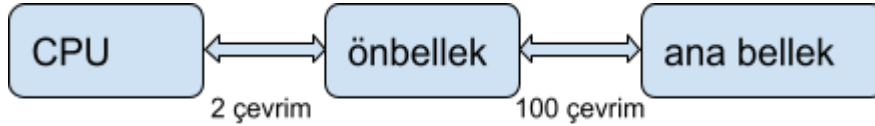
4 kilobaytlık dört yollu kümeli ilişkili → 1 küme 1KB büyüklüğünde → dizin + offset = 10-bit  
 veri öbeği 8 bayt → bayt offset = 3-bit  
 dizin = 10 - 3 = 7-bit  
 etiket = 16 - 10 = 6-bit



4 adet 6-bitlik karşılaştırıcı devre kullanılır.



b.) Load R1 ← mem[...]; R1'e 1 baytlık veri yazılıyor(120 defa)



- işlemci ilk defa veri istediğinde ön bellekte bulamayacak. Ana belleğe gidip bir veri öbeği kadar veriyi ön belleğe getirecek. 8 bayt veri olduğu için 8 kere ön bellekten veriyi alabilecek. Daha sonra tekrar miss edip aynı döngüyü 120 / 8 = 15 kere yapacak.

döngü 8 kere döndüğünde;

1 → ana bellek

1+8 → ön bellek erişimi olacak (baştaki 1'ler ilk miss olayı için)

bu 15 kere tekrar edecek → 15 ana bellek erişimi 135 ön bellek erişimi gerçekleşir. **not:**

normalde 105 kere ön belleğe erişecek 15 kere de ana belleğe erişecek fakat soruda belirtildiği gibi ana bellekten direkt okunamadığı için ana bellekten ön belleğe getirilip daha sonra ön bellekten okunacak. Yani +15 daha eklenip 135 ön bellek erişimi olmuş olacak.

Toplam çevrim = 15 x 100 + 135 x 2 = 1770

BBÇ(load) = 1770 / 120 = 14.75