

önbellek ve ESÖ'de bulma oranı

8KB kümeli ilişkili veri önbelleği

veri öbekleri 64 bayt

3 elemanlı tam ilişkili ESÖ

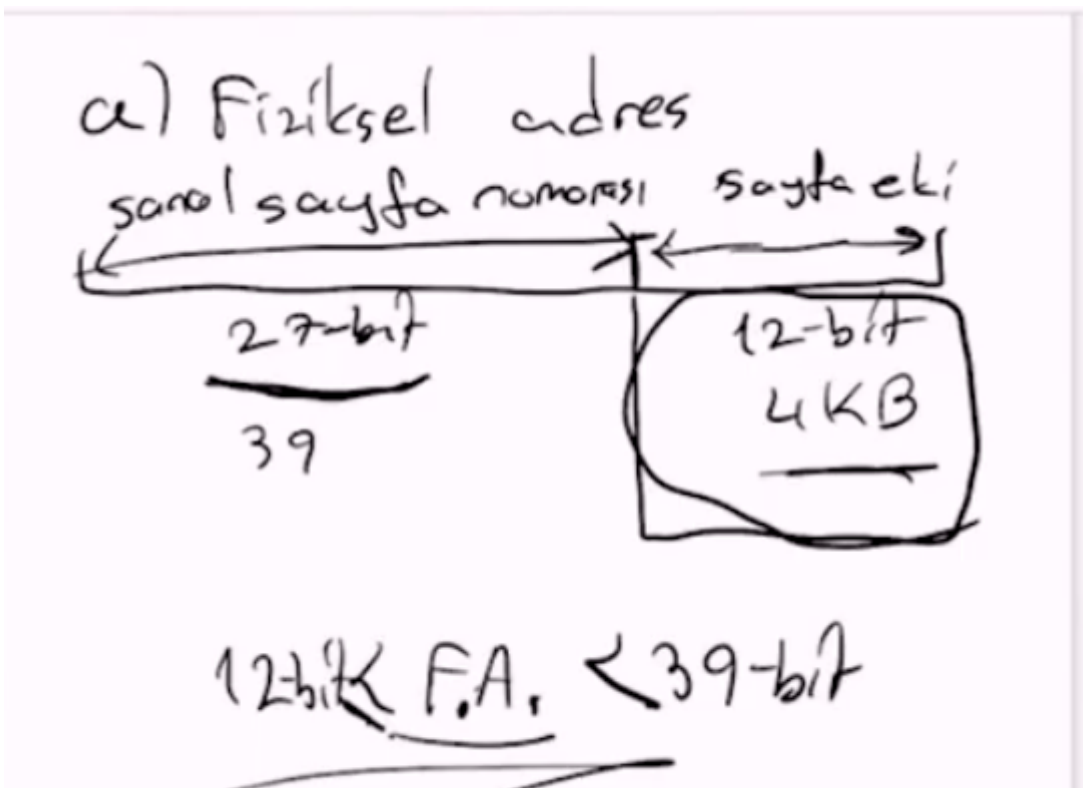
sayfa boyutu 4KB <-- **sayfa eki**

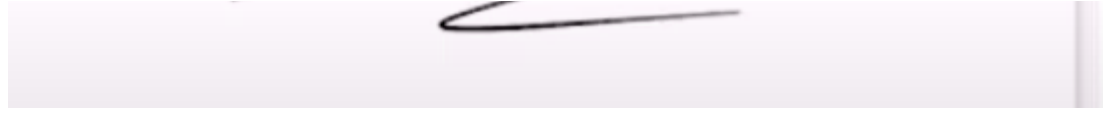
ESÖ ve veri önbelleğine aynı saat vuruşunda erişebiliyor

sanal adresler 39 bit genişliğinde

```
N = 1024;  
int A[N][N];  
int B[N][N];  
  
fonksiyon gez()  
{  
  for(x=1; x<N-1; x++)  
  {  
    for(y=1; y<N-1; y++)  
    {  
      toplam = A[x][y];  
      toplam += A[x+1][y];  
      toplam += A[x-1][y];  
      toplam += A[x][y-1];  
      toplam += A[x][y+1];  
  
      B[x][y] = toplam;  
    }  
  }  
}
```

a) fiziksel adresler kaç bit genişliğinde olabilir?





b) kaç tane fiziksel ve sanal sayfa vardır?

$12 \leq F.A \leq 39$ bit olarak

$2^{(12-12)} \leq FSN \leq 2^{(39-12)}$ sayfa sayısı olarak (en az sayfa eki kadar olacak en fazla sanal sayfa numarası kadar)

$39-12 = 27$ -bit sanal sayfa numarası olmalıdır. 2^{27} sanal sayfa vardır.

c) kod parçasının önbellekte bulma oranı nedir?

integer 4 bayt.

1.)örüntü

X A[1][1] -->001: **0000 0000 0100 (12 bayt sayfa eki)**

X A[2][1] -->010: 0000 00 00 0100

X A[0][1] -->000: 0000 00 00 0100 --> **0000 00 => satır no 00 0100 => öbek eki**

X A[1][0] -->001: 0000 00 00 0000

O A[1][2] -->001: 0000 00 00 1000

X B[1][1] -->001: 0000 00 00 0100

sayfa boyutu 4KB => 4096 bayt 2^{12} 2^6 veri öbeği boyutu olduğu için $2^{12} / 2^6 = 2^6 = 64$ satır

ESÖ ve veri önbelleğine aynı anda erişim olduğu için

hepsinin satır numarası aynı yani aynı satıra denk geliyor hepsi, 2 yollu önbelleğimiz olduğu için aynı satırda farklı etiketlere sahip en fazla 2 elemanı önbellekte tutabileceğiz.

önbellekte B'yi asla bulamıyoruz.

2.)örüntü

O A[1][2] -->001: 0000 00 00 1000

X A[2][2] -->010: 0000 00 00 1000

X A[0][2] -->000: 0000 00 00 1000

X A[1][1] -->001: 0000 00 00 0100

O A[1][3] -->001: 0000 00 00 0110

X B[1][1] -->001: 0000 00 00 0100

X B[1][2] --> 001:0000 00 00 0100

.

.

.

3.)örüntü

O A[1][15] --> 001: 0000 00 11 1100

X A[2][15] --> 010: 0000 00 11 1100

X A[0][15] --> 000: 0000 00 11 1100

X A[1][14] --> 001: 0000 00 11 1000

X A[1][16] --> 001: **0000 01** 00 0000 => **burada farklı bir satıra geçiyoruz.**

X B[1][15] --> 001:0000 00 11 1000

4.)örüntü

O A[1][16] --> 001: 0000 01 00 0000 => *burada dikkat et elemanlar 1.satırda*

X A[2][16] --> 010: 0000 01 00 0000

X A[0][16] --> 000: 0000 01 00 0000

O A[1][15] --> 001: 0000 00 11 1100

X A[1][17] --> 001: 0000 01 11 0100

X B[1][16] --> 001: 0000 01 00 0000

5.)örüntü

O A[1][17] --> 001: 0000 01 00 0100

X A[2][17] --> 010: 0000 01 00 0100

X A[0][17] --> 000: 0000 01 00 0100

X A[1][16] --> 001: 0000 01 00 0000

O A[1][18] --> 001: 0000 01 00 1000

X B[1][17] --> 001: 0000 01 00 0100

veri öbeği 64 bayt olduğu için 1 öbekte 16 integer olabiliyor. yani A[1][0] - A[1][15] arası aynı öbekte saklanıyor.

her A[1][15] A[1][16] geçişinde 4. yaptığımız örüntü gerçekleşecek. bu da bir döngüde $1024 / 16 = 64$ kez gerçekleşecek. $1024 * 64$ toplam gerçekleşmesi olacak

A[1][1] --> A[1][15] 2.örüntü 13 kez toplamda 1 kez

A[1][16] --> A[1][31] 5.örüntü 14 kez toplamda 63 kez

$(14 * 63 + 13) * 2 = \text{bulma}$

$(14 * 63 + 13) * 4 = \text{bulamama}$

$(14 * 63 + 13) * 1024 * 2 + 63 * 3 * 1024 \Rightarrow \text{bulma sayısı buna} = A \text{ diyelim}$

$A / (A + (14 * 63 + 13 + 63 * 9) * 1024 \rightarrow \text{bu da bulma oranıdır.}$

d) Kod parçasının ESÖ'de bulma oranı nedir?

burada satırdan çok hangi sütuna erişildiğine bakacağız.

diziye göre erişilen sayfalar

1.sayfa X

2.sayfa X

0.sayfa X

1.sayfa O => **3 elemanlı tam ilişkili dediğimiz için burada 1.sayfayı bulabiliyoruz.**

1.sayfa O

B --> X B'yi asla bulamayacağımız için genel bir terim olarak B diyoruz.

ESÖ = 1 B 2

1 O

2 O

0 X --> 1 0 2

1 O

1 O

B X --> 1 0 B

ESÖ = 1 0 B

1 O

2 X --> 1 2 B

0 X --> 1 2 0

1 O

1 O

B X --> 1 B 0

ESÖ = 1 B 0

1 O

2 X --> 1 B 2

0 X --> 1 0 2

1 O

1 O

B X --> 1 0 B

ESÖ = 1 0 B

örüntü yakaladık.

bu örüntü içerdeki döngü bitene kadar devam edecek. yani bir kez y N-1'e ulaşana kadar bu şekilde devam edecek.

sayfa geçişlerindeki değişimlere bakacağız.

2 X --> 1 2 B

3 X --> 3 2 B

1 X --> 3 2 1

2 O

2 O

B+1 X --> B 2 1

ESÖ = B 2 1

2 O

3 X --> B 2 3

1 X --> 1 2 3

2 O

2 O

B+1 X --> 1 2 B

ESÖ = 1 2 B

2 O

3 X --> 3 2 B

1 X --> 3 2 1

2 O

2 O

B+1 X --> B 2 1

ESÖ = B 2 1

3 O --> $(1022 * 1021 - 2)^3 + 6 + (1022 * 2)$ bulma

3 X --> $(1022 * 1021 - 2)^3 + 6 + (1022 * 4)$ bulmama