



İ.T.Ü
Bilgisayar Mühendisliği Bölümü
Prof.Dr. Emre HARMANCI
Yrd.Doç.Dr. Feza BUZLUCA

16.05.2005

BİLGİSAYAR MİMARİSİ DÖNEM SONU SINAVI

....

SORU 3: (50 Puan)

Görüntü adres alanı 512M*8 olan bir bilgisayar sisteminde 2M*8 boyutunda ana bellek bulunmaktadır. Sistemdeki bellek yönetim birimi, 2K*8 boyutunda sayfaların kullanıldığı sayfalı dönüşüm (paged mapping) yöntemine göre çalışmakta ve 4 satırlı bir öngörü tablosundan (TLB) yararlanmaktadır. Aynı sistemde 4K*8'lik veri taşıyabilen bir cep bellek bulunmaktadır. Cep belleğe erişim 32*8'lik blokların kullanıldığı ve her kümede 2 bloğun yer aldığı set-assosiyatif yöntemle sağlanmaktadır. Sistemdeki tüm yer değiştirme işlemleri için **LRU** yöntemi kullanılmaktadır.

- Yukarıda tanıtılan sistemde, görüntü adres – fiziksel adres dönüşümü ve cebe erişim aşamalarında adresler hangi alt alanlara ayrılırlar? Kullanılan tabloların boyutlarını ve hangi verileri içerdiğini açıklayınız.
- Başlangıçta tüm tabloların, ana belleğin ve cep belleğin boş olduğu ve ana belleğin sıfır numaralı adresten itibaren kullanıma açık olduğu varsayımı altında, bir adres başvurusu yapıldığında sistemin davranışının nasıl olacağını maddeler halinde açıklayınız.
- Böyle bir sistemin hangi bölümlerinde hangi durumlarda yer değiştirme algoritmasına gerek duyulur kısaca açıklayınız.
Cep bellekte yer değiştirmeyi gerekli kılacak bir fiziksel adres başvuru zinciri örneği veriniz.
- Cep bellek erişimlerinde set-assosiyatif yöntem yerine doğrudan dönüşüm kullanılması sistemin ortalama bellek erişim süresini nasıl etkileyebilir?
- Sistemin ortalama bellek erişim süresi açısından cep erişiminde kullanılan “Write Though” ve “Write Back” yöntemlerini karşılaştırın.

UYGULAMA

~~2003~~

2005 Final

SORU 3: Görüntü adres alanı 512M x 8

Ana Bellek 2M x 8 ²¹

Sayfa Boyutu 2K x 8 (sayfa dönüşüm yöntemi kullanılabilir)

TLB 4 satırlı

Cep Bellek 4K x 8

Block → 32 x 8

Set → 2 Blok

Set-associatif yöntem.

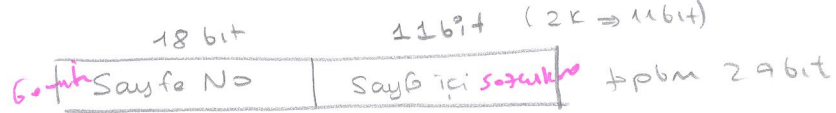
(Tüm yedekleştirme ve LRU yöntemi kullanılır.)

a) Görüntü adres - fiziksel adres dönüşümü ve çeşitli adresleri artırmak? Kullanılan tablo boyutları ve içerik açıklanabilir mi?

$$\text{Görüntü Bellekte } \frac{29 \cdot 10^{20}}{2 \cdot 2^{10}} = 2^{18} \text{ adet sayfa var.} \Rightarrow \text{sayfa-no} = 18 \text{ bit}$$

$$\text{Ana Bellekte } \frac{2 \cdot 2^{20}}{2 \cdot 2^{10}} = 2^{10} \text{ adet blok var} \Rightarrow \text{Block-no} = 10 \text{ bit}$$

⇒ Görüntü adres 29 bit



⇒ Fiziksel adres

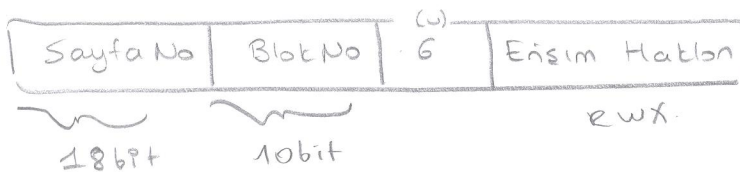


⇒ Sayfa Tablosunun Boyutu sayfa sayısı kadar ⇒ $\frac{2^{18}}{2^8 \cdot 2^{10}} \Rightarrow 256 \text{ K}$

Sayfa Tablosunun 1 satırının içeriği ⇒



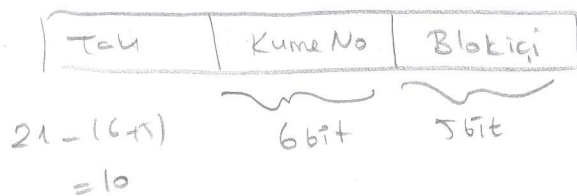
⇒ TLB



⇒ Cep Bellek

$$\frac{4 \text{ K}}{2 \cdot 32} = \frac{2^2 \cdot 2^{10}}{2 \cdot 2^5} = 2^6 \text{ tane küme}$$

$$\text{Block 32} \Rightarrow 5 \text{ bit}$$



b) Başlangıçta tüm tablolar, cep ve ana bellek boş. Ana bellek 0 numaralı adresten itibaren kullanıma açılır. Bir adres başvurusu yapıldığında sistemin davranışı:

— Görüntü adres üretilir

— Aranan sayfanın blok numarası için TLB'ye bakılır

① TLB'de bulunursa:

— Blokno + Blok içi adres'ten fiziksel adres üretilir

— Üretilen fiziksel adres hem cep belleğe hem ana belleğe gönderilir

Küme adresi ile cepten bir küme seçilir, tabii bilgisi ile o küme blokları taranır, aranan blok bulunursa blok içi bitleriyle veri seçilir, bulunmazsa veri cepte yok demektir ana bellekten okunur, ^{SRAM}cebe de alınır.

② Sayfa numarası TLB'de bulunmazsa ⇒ sayfa tabibusunun sonucu beklenir.

Ana sayfa ana bellekte

Sayfanın blok numarası + blok içi ile fiziksel adres üretilir.

Hem cebe hem ana belleğe başvurulur

Küme adresi ile cepten bir küme seçilir, tabii o küme içinde blok aranır. Bulunursa blok içi bitleriyle veri seçilir, bulunmazsa veri cepte yoktur, ana bellekten okunur, ayrıca cebe de alınır.

TLB güncellenir (Sayfa no + blokno TLB'ye eklenir)

Ana sayfa ana bellekte değil

a) Diskten veri okunur, ana belleğe getirilir

b) Sayfa tabibusu güncellenir (Sayfanın bulunduğu blok numarası sayfa tabibusuna yazılır).

c) TLB güncellenir.

c) Böyle bir sistemin hangi bölümlerinde hangi durumda ve değiştirme algoritması gerek duyulur? Cep bellekte ve değiştirme, gerçekli çok bir fiziksel adres basurun için örneği veriniz.

⇒ TLB nin tüm satırları dolu olduğunda, bu adres basurunsunda aranan sayfa TLB'de bulunmazsa TLB güncellenir, son eiser sayfa no + btkno TLB'ye yerleştirir, atılacak satırı seçmek için yer değiştirme algoritması kullanılır.

⇒ Ana Bellek ⇒ aranan sayfa ana bellekte yoksa diskten getirir ve ana belleğin bir bölümüne yerleştirilir, fakat ana bellek doluyorsa hangi btk no bellekten çıkarılacak?

⇒ Cep Bellek ⇒ Aranan cı cepte yoksa bellekten alılır ve cep güncellenir, yeni verinin bulunduğu btk cepte uygun kimeye yerleştirir, peki kime doluyorsa?

	Kırmızı	Btk	
\$0A0A ⇒ 0000	1010 0000	1010	K16'ya yerlesmek ister K=1
\$121D ⇒ 0001 0	010 000	1 1101	K16'ya " " K16 2
\$8A00 ⇒ 1000 1	010 000	0 0000	K16'ya " " ama K16 dolu!

d) Cep Bellek erişiminde set asosyatıf yerine doğrudan dönüşüm kullanılması ortalama bellek erişim süresini nasıl etiler? Arttırır**

Set Asosyatıf daha esnek, doğrudan dönüşüm yerine yerine yerleşeceği kesin ve belli orda btk btk varsa olur, atılır bulunmazsa tekör bellek erişimi vs. esnek olmadı için btk yer okalsa bile istenen btk yerleştirilemez.

Set-as set olmadıkça yer değişimini yok.

e) Sistemin ortalama bellek erişim süresi açısından cep erişiminde kullanılan "Write Through" ve "Write Back" yöntemlerini karşılaştır.

Write Through → yazma (vuru) hercebe her no belleğe

Write Back → sadece cebe, cepten çıkarken no belleğe yazılır.

** NOT: doğrudan dönüşüm kullanılması, vuru olduğunda erişim süresini azaltır (asosyatıf arama olmadığı için), fakat bu sistemde vuru olma olasılığı daha düşüktür. Daha çok yer değiştirme olması ort. bellek erişim süresini arttırır ...