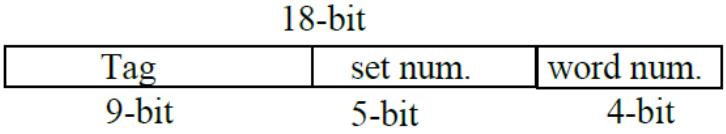
BLG322E Computer Architecture - Recitation 5

A computer system includes **256 KB** (B: Byte) main memory and a cache memory that can hold **1 KB** data. Data transfers between main and cache memories are performed using blocks of **16 bytes**. The cache control unit uses set associative mapping technique where each set contains **two frames** (*2-way set associative*).
In necessary cases FIFO is used as replacement technique.
a) In what fields is the physical address divided by the cache control unit? Give the lengths of the fields.
b) What is the size of the tag memory (how many rows, the length and contents of each row)?
c) The CPU runs the piece of code given on right. This program runs a loop (For i = 1 to 2) two times and reads elements of three arrays.

```
LOOP: For i = 1 to 2
        For j=0 to 19 Read A[j]; //Read 20 elements of A
        For j=0 to 9 Read B[j];  //Read 10 elements of B
        For j=0 to 9 Read C[j];  //Read 10 elements of C
      End of For i
```

The starting address and sizes of the arrays:
A: Starting address: $00010, size: 20 bytes
B: Starting address: $00420, size: 10 bytes
C: Starting address: $01020, size: 10 bytes
Assume that the cache memory is empty. Which arrays are placed to which frames of the cache memory in the **first run** of the loop (For i = 1)? Show the values written to the tag memory.
d) What is the total number of the replacements when the loop runs two times?
e) To increase the hit ratio find another starting address for the array C. Explain shortly.

a) Physical address (fiziksel adres): 18-bit



b)
The tag memory has 64 rows (One tag for each frame).
The length of the tag is 9 bits.
There is also additional information in each row: Valid bit, dirty bit, one bit (flag) for FIFO. Aging counter is not necessary, because the replacement method is not LRU.
Size of the tag memory: 64x12 bits

Takı belleğinde 64 satır vardır (Her çerçeve için bir takı).
Bir takının uzunluğu: 9 bit.
Takı belleğinin her satırında ek bilgiler de vardır: Geçerlilik biti, değişim biti, FIFO için bir bitlik bir bayrak.
Yer değiştirme yöntemi LRU olmadığı için yaşlanma sayacı gerekli değildir.
Takı belleğinin boyutu 64x12 bit.

c)

A[0]-A[15]: $00010-$0001F: 00 0000 0000 0001 xxxx

Cache is empty, cache miss occurs, first 16 bytes of the array A are placed into 1st frame of set 0 0001 of the cache memory. Tag value is 00 0000 000

Cep bellek boş, ıska olur, A dizisinin ilk 16 sekizlisi cep belleğin 0 0001 numaralı kümesinin 1. çerçevesine getirilir.
Takı değeri: 00 0000 000

A[16]-A[19]: $00020-$00023: 00 0000 0000 0010 xxxx

A block including 4 bytes of the array A is placed into 1st frame of set 0 0010 of the cache memory. Tag value is 00 0000 000

A dizisinin 4 sekizlisini içeren blok cep belleğin 0 0010 numaralı kümesinin 1. çerçevesine getirilir.
Takı değeri: 00 0000 000

B[0]-B[9]: $00420-$00429: 00 0000 0100 0010 xxxx

A block including 10 bytes of the array B is placed into 2nd frame of set 0 0010 of the cache memory. Tag value is 00 0000 010 ***(Now, set 2: 00010 is full).***

B dizisinin 10 sekizlisini içeren blok cep belleğin 0 0010 numaralı kümesinin 2. çerçevesine getirilir.
Takı değeri: 00 0000 010 (**Artık küme 2: 00010 doludur**).

C[0]-C[9]: $01020-$01029: 00 0001 0000 0010 xxxx

Set 2: 00010 is full. According to FIFO method the 1st frame in set 2, including the elements *A[16]-A[19] is replaced with C.*
A block including 10 bytes of the array C is placed into 1st frame of set 0 0010 of the cache memory. Tag value is 00 0001 000 .

Küme 2: 00010 doludur. FIFO yöntemine göre küme 2'de A[16]-A[19] elemanlarının yer aldığı 1. çerçeveye C dizisinin elemanları yerleştirilir.

C dizisinin 10 sekizlisini içeren blok cep belleğin 0 0010 numaralı kümesinin 1. çerçevesine getirilir.
ğeri: 00 0001 000 .

d) *There was one replacement in the first run. C[0]-C[9] replaced A[16]-A[19].*
In the second run:
A[0]-A[15]: Hit. No replacement.
A[16]-A[19]: Miss. Replace B[0]-B[9]. +1
B[0]-B[9]: Miss. Replace C[0]-C[9]. +1
C[0]-C[9]: Miss. Replace A[16]-A[19]. +1

Total number of replacements: 3+1 = 4

İlk döngüde bir yer değiştirme olmuştu. C[0]-C[9] ile A[16]-A[19] arasında.
Döngünün ikinci çalışmasında:
A[0]-A[15]: Vuru. Yer değiştirme yok.
A[16]-A[19]: Iska. Yer değiştirme B[0]-B[9]. +1
B[0]-B[9]: Iska. Yer değiştirme C[0]-C[9]. +1
C[0]-C[9]: Iska. Yer değiştirme A[16]-C[19]. +1

Toplam yer değiştirme sayısı: 3+1 = 4

e) *The set number must be different from 2 (00010). Because 3 arrays try to share this set.*
Possible solutions; C: Starting address: $01010 (Set 1)
Or C: Starting address: $01030 (Set 3)

***Special case** (valid only for the given program):*
A[16]-A[19] uses only 4 bytes of a frame. The remaining bytes can be used by C[0]-C[9].
We can place array C just after A.
Starting address of C: $00024
This is the fastest solution, because with A[16]-A[19], also C[0]-C[9] will be placed into the cache.

Küme numarası 2'den (00010) farklı olmalı, çünkü üç dizi de aynı kümeyi paylaşmaya çalışıyor.
Olası çözümler:
C: başlangıç adresi: $01010 (Küme 1)
C: başlangıç adresi: $01030 (Küme 3)

Özel Durum (sorudaki program için geçerlidir):
A[16]-A[19] bir çerçevenin sadece 4 sekizlisini kullanır. Kalan yerlere C[0]-C[9] yerleştirilebilir.
C dizisi hemen A'dan sonraya yerleştirilir.
C: başlangıç adresi: $00024
Bu en hızlı çözümdür, çünkü A[16]-A[19] cep belleğe aktarılırken C[0]-C[9] da cep belleğe alınmış olacak.