CSE3030: Introduction to Computer Systems
Spring 2023

Assignment #3

Due date: June 14th 9:20 AM
Offline submission at the class

1. [50 pt.] Assume a small 16-bit address space special-purpose processor that can be equipped with one of two direct-mapped caches, C1 or C2. Both C1 and C2 have a total capacity of 64 bytes. C1 has a block size of 4 bytes while C2 has a block size of 16 bytes.

(a) Fill in the complete cache parameters (m, C, B, E, S, t, s, b) for C1 and C2. For the meaning of each parameter, refer to our lecture note.

C1 [5pt]:

m	C	B	E	S	t	s	b
16	64	4	1	16	10	4	2

C2 [5pt]:

m	C	B	E	S	t	s	b
16	64	16	1	4	10	2	4

(b) Assume that the cache is initially empty, and we have a program that reads 1-byte data from the following sequence of (hexadecimal) memory addresses: BA00, BA04, AA08, BA05, AA14, AA11, AA13, AA38, AA09, AA0B, BA04, AA2B, BA05, BA06, AA09, AA11.

For each cache option, specify which references are hits (H) and which are misses (M).

C1 [10pt]:

BA 00	BA 04	AA 08	BA 05	AA 14	AA 11	AA 13	AA 38	AA 09	AA 0B	BA 04	AA 2B	BA 05	BA 06	AA 09	AA 11
M	M	m	H	M	m	H	M	H	H	H	M	H	H	H	H

C2 [10pt]:

BA 00	BA 04	AA 08	BA 05	AA 14	AA 11	AA 13	AA 38	AA 09	AA 0B	BA 04	AA 2B	BA 05	BA 06	AA 09	AA 11
m	H	M	M	M	H	H	M	m	H	M	M	H	H	M	H

For each cache option, specify the final data content of the cache. You can use expression “X-Y” to denote the bytes from address X to Y. Leave the cell as empty if the set is not filled.

C1 [10pt]:

set 0:	BA00 - BA03
set 1:	BA04 - BA07
set 2:	AA08 - AA0B
set 3:	
set 4:	AA10 - AA13
set 5:	AA14 - AA17
set 6:	
set 7:	
set 8:	
set 9:	
set 10:	AA28 - AA2B
set 11:	
set 12:	
set 13:	
set 14:	AA38 - AA3B
set 15:	

C2 [10pt]:

set 0:	AA00 - AA0F
set 1:	AA10 - AA1F
set 2:	AA20 - AA2F
set 3:	AA30 - AA3F

2. [50 points]

Consider a cache with parameters m = 32, b = 8, s = 8 and E = 4. The cache is initially empty.

Assume that:

sizeof(element)	= 8 bytes
@x[256]	= AAAA0000
@y[256]	= AABB0000
@a[512]	= AAAA8000
@b[512]	= AABB8000
Only x and y are in main memory	
LRU eviction	

Consider the following code segment:

```
for (i = 0; i < 256; i++) {
    value1 = x[ i ] ∗ y[ i ]
    for (j = 0; j < 2; j++)
        value2 = a[ i×2+j ] + b[ i×2+j ]
}
```

(a) Fill in the complete cache parameters (m, C, B, E, S, t, s, b) for the cache. [5pt]

m	C	B	E	S	t	s	b
32	262144	256	4	256	16	8	8

(b) After the first iteration of the outer loop with "i", which cache sets are filled? [5pt]

A. (set#0 , #128)

(c) After the first iteration of the outer loop with "i", which elements of x[ ], y[ ], a[ ], and b[ ] are stored in cache set #0, #64, #128, and #192? You can use expression “arr[i – j]” to denote the elements from index “i" to index “j”. Leave the cell as empty if the set is not filled. [10pt]

set 0:	x[0-31] , y[0-31]
set 64:	
set 128:	a[0-31] , b[0-31]
set 192:	

(d) How many cache sets are filled after the completion of the outer loop with "i"? [10pt]

A. (24) sets

(e) Which elements of x[ ], y[ ], a[ ], and b[ ] are stored in cache set #143 after the completion of the outer loop with "i"? [10pt]

set 143:	a[480-511], b[480-511]

(f) What is the overall hit rate of this code? **Answer in fraction.** [10pt]

A. ($\frac{31}{32}$)