IC220: HW 9

Due: 26 Apr 2019

Full Name: _____ **Alpha:** _____

Circle Your Section: Aviv/1001 Aviv/2001 Aviv/4001 Choi/5001 Missler/5002

Total Points: 70

Preliminary: Carefully do the assigned reading for Chapter 5 (5.1-5.7)

---

1. **[5 points]** Consider a direct mapped cache. Consider the address/memory pairing, and the series of instructions. Ignoring the tag and using just the index, show the series of address and data placements in the cache, and indicate the total number of hits and misses.

Addr.	Data
20	7
21	3
22	27
23	32
24	101
25	78
26	59
27	24
28	56
29	87
30	36
31	98

Index	Address	Data
0		
1		
2		
3		

Read 30
Read 31
Read 30
Read 26
Read 25
Read 28
Read 23
Read 25
Read 28

Hits _____ Misses _____

2. [**5 points**] Consider a direct mapped cache. Consider the address/memory pairing, and the series of instructions. Ignoring the tag and using just the index, show the series of address and data placements in the cache, and indicate the total number of hits and misses.

Addr.	Data
20	7
21	3
22	27
23	32
24	101
25	78
26	59
27	24
28	56
29	87
30	36
31	98

Index	Address	Data
0		
1		
2		

Read 30
Read 31
Read 30
Read 26
Read 25
Read 28
Read 23
Read 25
Read 28

Hits _____ Misses _____

3. [**5 points**] Looking back at the last two questions:

   - Identify **two** different kinds of reasons for why there might be a cache miss.
   - Provide a technique for addressing each type of miss.

4. [**5 points**] Consider a direct mapped cache. Consider the address/memory pairing, and the series of instructions. Ignoring the tag and using just the index, show the series of address and data placements in the cache, and indicate the total number of hits and misses.

   In this cache the block size is 2! If you read an odd number address, you should load two blocks from the lowest even number. For example, address 3 would access address 2 and 3.

Addr.	Data
20	7
21	3
22	27
23	32
24	101
25	78
26	59
27	24
28	56
29	87
30	36
31	98

Index	Address	Data1	Data2
0			
1			
2			
4			

Read 24
Read 22
Read 25
Read 21
Read 28
Read 21
Read 31
Read 25
Read 27

Hits _____ Misses _____

5. Provide the correct formula for calculating the cache index given the cache parameters

   (a) **[2 points]** N=10, Block Size=4

   (b) **[1 point]** N=8, Block Size=1, Associativity=4

   (c) **[2 points]** N=16, Block Size=8, Associativity=2

6. **[5 points]** Consider the associative cache below. Consider the address/memory pairing, and the series of instructions. Ignoring the tag and using just the calculated index, show the series of address and data placements in the cache, and indicate the total number of hits and misses.

   In this N=8, the associativity is 4, and the block size is 2!. Since the block size is still 2, if you read an odd number address, you should load two blocks from the lowest even number. For example, address 3 would access address 2 and 3.

Addr.	Data
20	7
21	3
22	27
23	32
24	101
25	78
26	59
27	24
28	56
29	87
30	36
31	98

Index	Address	Data1	Data2
0			
1			

Read 24
Read 25
Read 26
Read 24
Read 21
Read 26
Read 24
Read 26
Read 27

Hits _____ Misses _____

7. Assume you have an empty cache with a total of 16 blocks, with each block having 2 bytes (block size of 2!). The cache is directly mapped. Below is a series of byte addresses operations.

Address	**Hit/Miss**	**Index**
2		
3		
11		
16		
21		
10		
80		
48		
39		
11		
3		
80		

(a) **[8 points]** Above, fill in the Hit/Miss column and the index column.

(b) **[2 points]** What is the formula for the index in this cache?

(c) **[5 points]** Draw the cache and it's final contents below:

8. Assume you have an empty cache with a total of 16 blocks, with each block having 2 bytes (block size of 2!), but this time the cache is two-way associative. Assume LRU for eviction.

Address	**Hit/Miss**	**Index**
2		
3		
11		
16		
21		
10		
80		
48		
39		
11		
3		
80		

(a) **[8 points]** Above, fill in the Hit/Miss column and the index column.

(b) **[2 points]** What is the formula for the index in this cache?

(c) **[5 points]** Draw the cache and it's final contents below:

9. Suppose a 32 bit address uses 4 bits for the index and 3 bits for the offset for every address.
   (a) **[1 point]** What is the block size for a **Direct Mapped cache**?

   (b) **[1 point]** What is the total number of blocks for a **Direct Mapped cache**?

   (c) **[1 point]** What is the total size of the cache for a **Direct Mapped cache**?

   (d) **[1 point]** What is the block size for a **4-Way Associative cache**?

   (e) **[1 point]** What is the total number of blocks for a **4-Way Associative cache**?

   (f) **[1 point]** What is the total size of blocks for a **4-Way Associative cache**?

   (g) **[1 point]** What is the tag size in **either** a **4-Way Associative cache** or a **Direct Mapped cache**?

10. **[3 points]** Suppose a cache uses 4-byte blocks and has 128 blocks total. Break up the 32-bit address into tag, index, and byte offset.

```
0000 1000 0101 1100 0001 0001 0111 1001
```