CS 240 SI Worksheet #17
Valerie Zhao
4/11/17

Cache (Part 1)

1. Given a cache with the following specifications and content:

- byte-addressable memory, each memory access is to 1 byte
- 13-bit addresses
- 2-way set associative, 4-byte block size, 8 sets

Set Index	Line 0 Tag	Line 0 Valid	Line 0 Byte 0	Line 0 Byte 1	Line 0 Byte 2	Line 0 Byte 3	Line 1 Tag	Line 1 Valid	Line 1 Byte 0	Line 1 Byte 1	Line 1 Byte 2	Line 1 Byte 3
0	09	1	86	30	3F	10	00	0	-	-	-	-
1	45	1	60	4F	E0	23	38	1	00	BC	0B	37
2	EB	0	-	-	-	-	0B	0	-	-	-	-
3	06	0	-	-	-	-	32	1	12	08	7B	AD
4	C7	1	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	-	-	-	-
6	91	1	A0	B7	26	2D	F0	0	-	-	-	-
7	46	0	-	-	-	-	DE	1	12	C0	88	37

a. (CSAPP Practice Problem 6.12) What is the format of an address for this cache? Fill in CO for cache block offset, CI for cache set index, and CT for cache tag:

12	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

b. (6.16) List all of the hexadecimal memory addresses that will hit in set 3:

c. (6.13) Will a 1-byte word access to address `0x0D53` result in a cache hit?

2. (6.17) For the following code:

```
typedef int array[2][2];
void transpose1(array dst, array src) {
      int i, j;
      for (i = 0; i < 2; i++) {
            for (j = 0; j < 2; j++) {
                  dst[j][i] = src[i][j];
            }
      }
}
```

Assume the code runs on a machine with the following properties:

- `sizeof(int)` = 4
- `src` array starts at address `0x00`, `dst` array starts at address `0x10`
- Single L1 data cache: direct-mapped, write-through, write-allocate, 8-byte block size
    - initially empty

a. If the cache has a total size of 16 bytes, indicate whether access to `src[row][col]` and `dst[row][col]` is a hit or a miss:

`dst` **array**			`src` **array**		
	Column 0	**Column 1**		**Column 0**	**Column 1**
Row 0	**m**		**Row 0**	**m**	
Row 1			**Row 1**		

b. Repeat for cache with total size of 32 bytes:

`dst` **array**			`src` **array**		
	Column 0	**Column 1**		**Column 0**	**Column 1**
Row 0	**m**		**Row 0**	**m**	
Row 1			**Row 1**		

3. (6.19) Given the machine with the following specifications:

- `sizeof(int) = 4`
- direct-mapped cache of 2,048 bytes in total, with 32-byte blocks; initially-empty

And the following code:

```
struct algae_position {
     int x, y;
}

struct algae_position grid[32][32]; // grid begins at memory address 0
int total_x = 0, total_y = 0;
int i, j;
for (i = 31; i >= 0; i--) {
     for (j = 31; j >= 0; j--) {
          total_x += grid[j][i].x;
          total_y += grid[j][i].y;
     }
}
```

a. **What is the total number of reads?**

b. **What is the total number of reads that hit in the cache?**

c. **What is the hit rate?**

d. **What would the miss rate be if the cache were twice as big?**