

CSE 361S Intro to Systems Software Assignment #4

Due: Thursday, Nov. 17, 2011.

1. 3M decides to make Post-It notes by printing yellow squares on white pieces of paper. As part of the printing process, they need to set the CMYK (cyan, magenta, yellow, black) value for every point in the square. 3M hires you to determine the efficiency of the following algorithm on a machine with a 2048-byte direct-mapped data cache with 32-byte blocks. You are given the following definitions:

```
struct point_color {
    int c;
    int m;
    int y;
    int k;
};

struct point_color square[16][16];
int i, j;
```

Assume the following:

- `sizeof(int) == 4`.
- `square` begins at memory address 0.
- The cache is initially empty.
- The only memory accesses are to the entries of the array `square`. Variables `i` and `j` are stored in registers.

Determine the cache performance of the following code:

```
for (i=0; i<16; i++) {
    for (j=0; j<16; j++) {
        square[i][j].c = 0;
        square[i][j].m = 0;
        square[i][j].y = 1;
        square[i][j].k = 0;
    }
}
```

- a. What is the total number of writes?
- b. What is the total number of writes that miss in the cache?
- c. What is the miss rate?

2. Given the assumptions in problem 1, determine the cache performance of the following code:

```
for (i=0; i<16; i++) {
    for (j=0; j<16; j++) {
        square[j][i].c = 0;
        square[j][i].m = 0;
        square[j][i].y = 1;
        square[j][i].k = 0;
    }
}
```

- a. What is the total number of writes?
- b. What is the total number of writes that miss in the cache?
- c. What is the miss rate?

3. Given the assumptions in problem 1, determine the cache performance of the following code:

```
for (i=0; i<16; i++) {
    for (j=0; j<16; j++) {
        square[i][j].y = 1;
    }
}
for (i=0; i<16; i++) {
    for (j=0; j<16; j++) {
        square[i][j].c = 0;
        square[i][j].m = 0;
        square[i][j].k = 0;
    }
}
```

- a. What is the total number of writes?
- b. What is the total number of writes that miss in the cache?
- c. What is the miss rate?

4. Consider the following memory sub-system (including cache):

- The memory is byte addressable.
- Memory accesses are to **1-byte words** (not to 4-byte words).
- Addresses are 13 bits wide.
- The cache is two-way set associative ($E = 2$), with a 4-byte block size ($B = 4$) and eight sets ($S = 8$).

The contents of the cache are as follows, with all numbers given in hex:

2-way set associative cache												
	Line 0						Line 1					
Set index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	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	CD	88	37

List all of the hex memory addresses that will hit in sets 1 and 6. Note: this problem builds off of practice problem 6.13 in the 2nd edition of the text.

- In a memory allocator, determine the minimum block size for each of the following combinations of alignment requirements and block formats. Assumptions: Explicit free list, four-byte pred and succ pointers in each free block, zero-size payloads are not allowed, and headers and footers are stored in four-byte words.

Alignment	Allocate block	Free block	Minimum block size (bytes)
Single-word	Header and footer	Header and footer	
Single-word	Header, but no footer	Header and footer	
Double-word	Header and footer	Header and footer	
Double-word	Header, but no footer	Header and footer	