New York University

CSCI-UA 201, Section 3

Computer Systems Organization, Spring 2015

Lab Assignment 2: The Cache Lab

Student:
Peter Mountanos
(*pjm*419)

Professor:
Joanna Klukowska
(*joannakl*)

April 18, 2015

Problem 1

For this problem, you need to tweak the functions `level_1(),...,level_5()` to make them produce lower number of cache misses. For each function that you optimize, you need to provide an explanation of why your improvement results in a smaller number of cache misses. Provide that in the comments above each function.

- `level_1()`: See function docstring for improvement (found in `optimized.c`).
- `level_2()`: See function docstring for improvement (found in `optimized.c`).
- `level_3()`: See function docstring for improvement (found in `optimized.c`).
- `level_4()`: See function docstring for improvement (found in `optimized.c`).
- `level_5()`: See function docstring for improvement (found in `optimized.c`).

Problem 2

The following table gives the parameters for a number of different caches, where m is the number of physical address bits, C is the cache size (number of data bytes), B is the block size in bytes, and E is the number of lines per set. For each cache, determine the number of cache sets (S), tag bits (t), set index bits (s), and block offset bits (b).

Cache	m	C	B	E	S	t	s	b
1.	32	1024	4	4	64	24	6	2
2.	32	1024	4	256	1	30	0	2
3.	32	1024	8	1	128	22	7	3
4.	32	1024	8	128	1	29	0	3
5.	32	1024	32	1	32	22	5	5
6.	32	1024	32	4	8	24	3	5

Problem 3
Consider the following matrix transpose routine:

```
typedef int array[4][4];

void transpose(array dst, array src) {
  int i, j;
  for (i = 0; i < 4; i++) {
    for (j = 0; j < 4; j++) {
      dst[j][i] = src[i][j];
    }
  }
}
```

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

- `sizeof(int) == 4`
- The `src` starts at address 0 and the `dst` starts at address 64 (decimal).
- There is a single L1 data cache that is direct-mapped, write-through, write-allocate, with a block size of 16 bytes.
- The cache has a total size of 32 data bytes and the cache is initially empty.
- Accesses to the `src` and `dst` are the only sources of read and write misses.

Part A
For each `row` and `col`, indicate if the access to `src[row][col]` and `dst[row][col]` is a hit (h) or a miss (m). For example, both reading `src[0][0]` and writing to `dst[0][0]` are misses.

`dst` array

	Col 0	Col 1	Col 2	Col 3
Row 0	m	m	m	m
Row 1	m	m	m	m
Row 2	m	m	m	m
Row 3	m	m	m	m

`src` array

	Col 0	Col 1	Col 2	Col 3
Row 0	m	m	h	m
Row 1	m	h	m	h
Row 2	m	m	h	m
Row 3	m	h	m	h

Part B
Repeat **Part A**, but for cache with total size of 64 data bytes.

`dst` array

	Col 0	Col 1	Col 2	Col 3
Row 0	m	m	h	h
Row 1	m	m	m	h
Row 2	m	h	m	m
Row 3	m	h	h	m

`src` array

	Col 0	Col 1	Col 2	Col 3
Row 0	m	m	h	h
Row 1	m	h	m	h
Row 2	m	h	h	m
Row 3	m	h	h	h