CSE 140 Lab/HW#5 – 4/12 11:59PM

Caches (30pts + 30pts)

1. Assume that the system uses 32-bit address.
   a. Consider a cache with 64MB of data, 4-way set associativity, and 128-byte line (block) size. What is the size of each line's tag in bits?

   b. Consider a cache with 2048 sets, 4-way set associativity, and 12-bit tags. How many bytes can the cache store? (What is the data capacity not including overhead of storing tags, valid bits, etc.?)

   c. Consider a cache with a 256-byte line size and 14-bit tags. The cache has 4096 lines. What is the set associativity of the cache?

2. In this problem, we compare cache replacement policies for a given trace of accesses to a set. A trace is made of consecutive block addresses dynamically accessed by a program. Take the following trace, where each letter is a block address:

   abcdcbaeffgfaefgdcbaefga

   Assume that the fully associative cache that has four cache lines is cold (empty) at the beginning. What is the miss rate under the following replacement policies:

   - FIFO
   - LRU

MIPS assembly for testing cache performance (40pts)

3. In this problem, you will evaluate cache performance with your MatrixMul program that was implemented in HW2. Use any MatrixMul code. If you couldn't finish the MatrixMul implementation in HW2, you can use other students' code. We won't check plagiarism for the code itself. But, the evaluation should be done by yourself. This is NOT a group work. You can use MARS tool to collect cache hits/misses.

   a. Overwrite the input and output matrixes in your original MatrixMul code with the following matrixes.

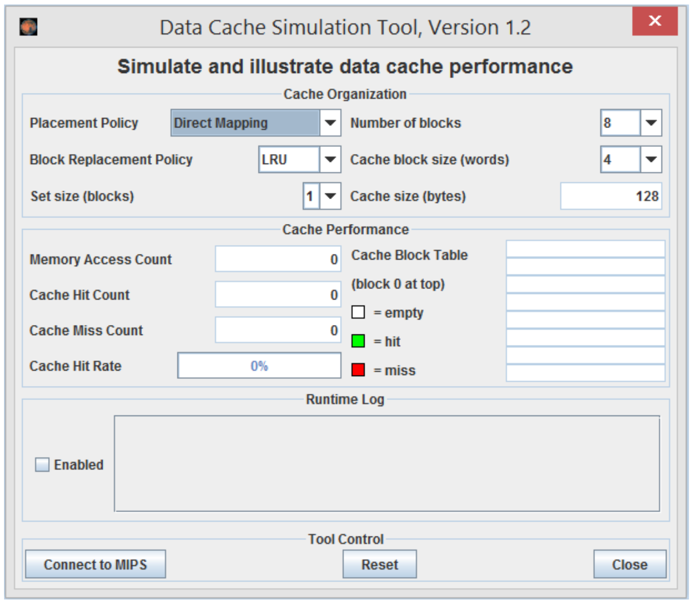
```
matrix_a:
       .word   1,  2,  3,  4,  5,  6,  7,  8,  9,
       .word  13, 14, 15, 16, 17, 18, 19, 20, 21,
       .word  25, 26, 27, 28, 29, 30, 31, 32, 33,
       .word  37, 38, 39, 40, 41, 42, 43, 44, 45,
       .word  49, 50, 51, 52, 53, 54, 55, 56, 57,
       .word  61, 62, 63, 64, 65, 66, 67, 68, 69,
       .word  73, 74, 75, 76, 77, 78, 79, 80, 81,
       .word  85, 86, 87, 88, 89, 90, 91, 92, 93,
       .word  97, 98, 99,100,101,102,103,104,105

matrix_b:
       .word 133,134,135,136,137,138,139,140,141,
       .word 121,122,123,124,125,126,127,128,129,
       .word 109,110,111,112,113,114,115,116,117,
       .word  97, 98, 99,100,101,102,103,104,105,
       .word  85, 86, 87, 88, 89, 90, 91, 92, 93,
       .word  73, 74, 75, 76, 77, 78, 79, 80, 81,
       .word  61, 62, 63, 64, 65, 66, 67, 68, 69,
       .word  49, 50, 51, 52, 53, 54, 55, 56, 57,
       .word  37, 38, 39, 40, 41, 42, 43, 44, 45

matrix_c:
```

```
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0,
.word   0,  0,  0,  0,  0,  0,  0,  0,  0
```

b. You can measure cache performance by using MARS Data Cache simulator plugin. Click “Tools->Data Cache Simulator” in the MARS menu bar. Then, the cache simulator tool will appear like below:


Data Cache Simulation Tool, Version 1.2

Simulate and illustrate data cache performance

Cache Organization

Placement Policy: Direct Mapping
Number of blocks: 8
Block Replacement Policy: LRU
Cache block size (words): 4
Set size (blocks): 1
Cache size (bytes): 128

Cache Performance

Memory Access Count: 0
Cache Hit Count: 0
Cache Miss Count: 0
Cache Hit Rate: 0%

Cache Block Table (block 0 at top)
☐ = empty
🟩 = hit
🟥 = miss

Runtime Log
☐ Enabled

Tool Control

Figure 1 Data Cache tool

The configurable parameters for the cache include Placement Policy (a.k.a. mapping scheme), Block Replacement Policy (always leave at LRU), Set size (a.k.a. K-Ways), Number of blocks in cache and cache block size. The total cache size (in bytes) is simply calculated as: (# of Blocks) * (Cache block size in words) * (4 bytes per word). Note: There is an oddity that statistics are not kept for any cache with 512 blocks or more even though those options exist in the drop down box.

To run data cache tool, click “Connect to MIPS” button of the Data Cache tool so the tool can begin monitoring the MARS simulation. **Everytime you run your program, you should also be sure to RESET the cache simulator** by clicking the Reset button.

b. To configure the cache, run the following experiments on your baseline code.

Cache misses can be categorized as compulsory, capacity, or conflict misses. Determine the number of compulsory misses. To do this, set your block size at **4 words per block** and set **replacement policy to LRU**. Think about what mapping scheme should be used if you do not want any conflict misses. Then continue to increase the size of the cache **by increasing the number of blocks** until you can correctly determine the number of compulsory misses. Find the smallest cache size that only produces compulsory misses.

c. Evaluate the cache performance of your program.

1. Given the smallest cache size that only produced compulsory misses (determined in step b), decrease the size of cache **to** 50%, 25%, and 12.5% **by decreasing the number of blocks**. Keep the block size (4 words per block), the replacement policy (LRU), and the placement policy (as you determined in step b) as is.
2. Change the placement policy to the other two types (e.g., if you set direct mapping in step b, use fully set associative and N-way set associative; feel free to set the N size as you wish) and find the configuration that provide better performance than the results in step c.1. (e.g., for my MatrixMul code, 4-way set associative cache shows more cache hits than direct-mapped cache that was used for determining the cache size)

d. In the homework submission, include the answers of the following questions based on your evaluations above.

1. What placement policy did you set to find the cache size in step b?

2. What is the cache size that has compulsory misses only (the size that you found in step b)?

2. How many compulsory misses did you have?

3. Put a table or a graph that shows “cache miss count” and “cache hit rate” when you decrease cache size to 50%, 25%, and 12.5%.

4. Explain what cache configuration showed better cache hit rate when the cache size is decreased to 50%, 25%, and 12.5%. Show the cache miss count and hit rate of the configuration for each cache size. And, add your reasoning why the configurations show better cache hit rates by emphasizing the memory access pattern of the MatrixMul code.

Submission Guideline

- Submit your solution in an MS Word or a pdf format to the CatCourse.
- Deadline: **4/12 11:59PM** (All sections have the same deadline for this HW)