Advanced Computer Architecture CMSC 611
Homework 5
Due in class at 1.05pm, Nov 21st, 2012

(If you wish to **go green,** then you can submit the entire Homework electronically. Make sure you include the string **"CMSC 611 Homework"** in your subject line. You will get a canned response immediately if your message is filtered correctly! I use an exact substring match for the filter so make sure you include the exact string in your subject line. Deadline remains the same)

Please **DO NOT** email your homework to Dr. Olano!! **DO NOT** include him in the CC either!! There is a strong chance it won't be graded if you do!! **Send it only to <abhay1@umbc.edu>**

1) For this problem, you should assume the following: (30 points)

   1. Pages are 4 KB
   2. There is a 4-entry, fully-associative TLB
   3. The TLB uses a true, least-recently-used replacement policy

   For the pattern of virtual addresses shown below, comment on whether the entry is:

   a. Found in the TLB
   b. Not found in the TLB but is found in the Page Table
   c. Not found in the TLB or the Page Table (thus, there is a page fault)

   The initial TLB and page table state is shown below.

   If there is a page fault, and you need to replace a page from disk, assume the page number is one higher than the current highest page in the page table. (This is currently 1100_2 / 12_{10}. Thus, the next page would be $1101_{(2)}$ / $13_{(10)}$, the next would be $1110_{(2)}$ / $14_{(10)}$, etc.)

Stream of Virtual Addresses:

	MSB			LSB
1	0000	1111	1111	1111
2	0111	1010	0010	1000
3	0011	1101	1010	1101
4	0011	1010	1001	1000
5	0001	1100	0001	1001
6	0001	0000	0000	0000
7	0010	0010	1101	0000

Initial TLB State:

(Note that '1' = "Most Recently Used and '4' = "Least Recently Used")

Valid	LRU	Tag	Physical Page #
1	3	1011	1100
1	2	0111	0100
1	1	1001	0110
0	4	0000	-----

Initial Page Table State:

	Valid	Physical Page #
0000	1	0101
0001	0	Disk
0010	0	Disk
0011	1	0001
0100	1	1001
0101	1	1011
0110	0	Disk
0111	1	0100
1000	0	Disk
1001	0	Disk
1010	1	0011
1011	1	1100

Question A (15 points)

Given the virtual addresses above, show how the state of the TLB changes by filling in the tables:

Address 1:

1	0000	1111	1111	1111

Valid	LRU	Tag	Physical Page #

This reference is a:

Address 2:

2	0111	1010	0010	1000

Valid	LRU	Tag	Physical Page #

This reference is a:

Address 3:

3	0011	1101	1010	1101

Valid	LRU	Tag	Physical Page #

This reference is a:

Address 4:

4	0011	1010	1001	1000

Valid	LRU	Tag	Physical Page #

This reference is a:

Address 5:

5	0001	1100	0001	1001

Valid	LRU	Tag	Physical Page #

This reference is a:

Address 6:

6	0001	0000	0000	0000

Valid	LRU	Tag	Physical Page #

This reference is a:

Address 7:

7	0010	0010	1101	0000

Valid	LRU	Tag	Physical Page #

This reference is a:

Question B: (10 points)

What would some of the advantages of having a larger page size be? What are some of the disadvantages?

Question C: (5 points)

Given the parameters in the table above, calculate the total page table size for a system running 5 applications.

Virtual Address Size	Page Size	Page Table Entry Size
64 bits	16 KB	8 bytes

SIMPLE SCALAR SIMULATOR **(70 points)**

2) You were introduced to the simple scalar simulator in Homework 4. Now that you have a good grip on how to run the simulators, we will move onto the cache simulator. The cache simulator can be run without arguments using `./sim-cache` to find out all the arguments it takes. For this problem we will use the anagram benchmark. So to run the cache simulator a command will look like this

`./sim-cache -redir:sim` **`simOutputFilePath`** `-redir:prog` **`progOutputFilePath`** `-cache:il1 dl1 -cache:il2 none -cache:dl1 ul1:4096:16:1:l -cache:dl2 none ./tests-pisa/bin.little/anagram ./tests-pisa/inputs/words < ./tests-pisa/inputs/input.txt`

The above command uses a unified L1 instruction and data cache and does not have an L2 cache (Note that the last parameter in `ul1:1024:32:1:l` is a small letter L which signifies LRU replacement policy and not the number One)

Note: Please run `./sim-cache` or `./sim-cache -h` to see the list of all parameters it takes. If you don't understand them, a simple web search of the parameter should help you understand what they represent.

Create your own input file – To make sure the simulation runs are not shared among peers, you are expected to run the anagram over different input files. Here are the steps to create an input file and the expected format.

In your **home directory**, create a file `hw5.txt` (**DO NOT** create your input file in `~olano/simplesim-3.0/`)

Your input file must have the following 3 line format

Line number 1*:* <*your first name*> and <*your last name*> (Use only one word per field if you have multiple words in your name)

Line number 2: < *first name*> and < *last name*> of your favorite scientist!

Line number 3: < *first name*> and < *last name*> of your favorite celebrity!

(More lines will take longer to execute! If Line number 1 and 2 has a large number of characters, Line number 3 can be omitted)

So the argument to anagram will now look like this,
`./tests-pisa/bin.little/anagram ./tests-pisa/inputs/words < ~<yourUmbcUsername>/hw5.txt`

You will have to submit or mention the contents of the input file used!

a) For the first task, (25 points)

- You will use the anagram benchmark with your input file.
- You will use a separate instruction and data L1 cache and a unified L2 cache.
- Assume a direct mapped cache.
- Assume a LRU replacement policy
- Vary the number of index lines in the cache from 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K.
- Vary the block size as 8, 16, 32, 64.

Fill the table below with the **L1 data cache** miss ratios (i.e % of cache misses)

	1K	2K	4K	8K	16K	32K	64K	128K
8								
16								
32								
64								

Plot the results as Miss ratio (y-axis) vs Cache size (x axis) for 4 different block size (Plot them in the same graph, not 4 different graphs!) A rough representation is good enough! Nothing fancy needed.

b) For the second task, (25 points)

- You will use the anagram benchmark with your input file
- You will use a separate instruction and data L1 cache and use a unified L2 cache.
- Assume a LRU replacement policy
- Vary the number of sets as 32, 64, 128, 256 and 512 sets
- Vary the associativity as 1-way, 2-way, 4-way, 8-way and 16-way
- Fixed block size of 16 bytes.

Fill the table below with the **L1 data cache** miss ratios (i.e % of cache misses)

	32	64	128	256	512
1way					
2way					
4way					
8way					
16way					

Plot the results as Miss ratio (y-axis) vs associativity (x axis) for the 5 different cache set size (Plot them in the same graph, not 5 different graphs!) A rough representation is good enough! Nothing fancy needed.

c) **Analysis** **(20 points)**

Based on the simulator output obtained, answer the following questions

i. For a given number of sets, what effect does increasing associativity have on the miss ratio?

ii. For a given associativity, what is the effect of increasing the number of sets?

iii. For a given cache size, how does the miss ratio change when going from an associativity of one to two to four?

iv. If you were to design an Instruction cache, limited to a total cache size of 4 Kbytes, which cache organization would you choose, based solely on performance? Justify your answer.

v. If you were to design a data cache, limited to a total cache size of 4 Kbytes, which cache organization would you choose, based solely on performance? Justify your answer.