# 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 (1) | Disk (1101) |
| 0010 | 0 (1) | Disk (1110) |
| 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 # |
|---|---|---|---|
| 1 | **4** | 1011 | 1100 |
| 1 | **3** | 0111 | 0100 |
| 1 | **2** | 1001 | 0110 |
| **1** | **1** | **0000** | **0101** |

This reference is a: TLB miss, Page table hit

**Address 2:**

| | | | | |
|---|---|---|---|---|
| 2 | 0111 | 1010 | 0010 | 1000 |

| Valid | LRU | Tag | Physical Page # |
|---|---|---|---|
| 1 | 4 | 1011 | 1100 |
| 1 | 1 | 0111 | 0100 |
| 1 | 3 | 1001 | 0110 |
| 1 | 2 | 0000 | 0101 |

This reference is a: TLB hit.

**Address 3:**

| | | | | |
|---|---|---|---|---|
| 3 | 0011 | 1101 | 1010 | 1101 |

| Valid | LRU | Tag | Physical Page # |
|---|---|---|---|
| 1 | 1 | 0011 | 0001 |
| 1 | 2 | 0111 | 0100 |
| 1 | 4 | 1001 | 0110 |
| 1 | 3 | 0000 | 0101 |

This reference is a: TLB miss, Page table hit

**Address 4:**

| | | | | |
|---|---|---|---|---|
| 4 | 0011 | 1010 | 1001 | 1000 |

| Valid | LRU | Tag | Physical Page # |
|---|---|---|---|
| 1 | 1 | 0011 | 0001 |
| 1 | 2 | 0111 | 0100 |
| 1 | 4 | 1001 | 0110 |
| 1 | 3 | 0000 | 0101 |

This reference is a: TLB hit

**Address 5:**

| | | | | |
|---|---|---|---|---|
| 5 | 0001 | 1100 | 0001 | 1001 |

| Valid | LRU | Tag | Physical Page # |
|---|---|---|---|
| 1 | 2 | 0011 | 0001 |
| 1 | 3 | 0111 | 0100 |
| **1** | **1** | **0001** | **1101** |
| 1 | 4 | 0000 | 0101 |

This reference is a: Page fault

**Address 6:**

| | | | | |
|---|---|---|---|---|
| 6 | 0001 | 0000 | 0000 | 0000 |

| Valid | LRU | Tag | Physical Page # |
|---|---|---|---|
| 1 | 2 | 0011 | 0001 |
| 1 | 3 | 0111 | 0100 |
| 1 | 1 | 0001 | 1101 |
| 1 | 4 | 0000 | 0101 |

This reference is a: TLB hit

**Address 7:**

| | | | | |
|---|---|---|---|---|
| 7 | 0010 | 0010 | 1101 | 0000 |

| Valid | LRU | Tag | Physical Page # |
|---|---|---|---|
| 1 | 3 | 0011 | 0001 |
| 1 | 4 | 0111 | 0100 |
| 1 | 2 | 0001 | 1101 |
| **1** | **1** | **0010** | **1110** |

This reference is a: Page fault

**Question B: (10 points)**

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

Advantage – Less Page faults, Smaller page tables
Disadvantage – Larger Page size => more fragment, Longer disk transfer, More page faults if no locality.

**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 |

Page size = 16KB = $2^{14}$
Page table entry size = $2^3$
So number of bits = 64 – 14 = 50. i.e. $2^{50}$ page table entries each with 8 bytes.
So Page table size = $2^{53}$ and for 5 applications, total page table size = 5 * $2^{53}$

Hence Inverted tables are used.

**SIMPLE SCALAR SIMULATOR** **(70 points)**

Answers vary.