CS 2200 - Introduction to Systems

Fall 2016

Homework 9

Rules:

- Please print a copy of the assignment and handwrite your answers. No electronic submissions are allowed. **Please print as one double-sided page.**
- This is an individual assignment. No collaboration is permitted.
- Due Date: 16th November 2016 – 6:05 PM . Bring your BuzzCard.

Name(print)_____RUBRIC_____ GT Login:_____ Section_____

Problem 1:

Consider a two-level paging system with the following characteristics:

- Each page is 4 KB (4096 bytes)
- Virtual addresses and physical addresses are 32 bits long
- Each page table entry occupies 4 bytes. Space not used for the physical frame number is reserved for other flags, such as valid bits, etc.
- Both inner and outer page tables fill a single page (i.e. their size is 4 KB)

A) Given the specifications above, mark the virtual address shown below with how many bits should be allocated to each portion:

Outer Virtual Page Number	Inner Virtual Page Number	Page Offset
__10____ bits	**___10___ bits**	**___12____ bits**

B) For each virtual address given, fill out the corresponding physical address using the page tables given below.

Page Table Base Register

PTBR	**0x3FF**

Page Tables

Page table at 0x3FF

VPN	PFN	V
0x3b8	0x12345	1
0x3b9	0xC3AFF	1
0x3ba	0x56789	1
0x3bb	0xFF00D	0

Page table at 0x56789

VPN	PFN	V
0x3fc	0x0BEEF	1
0x3fd	0xCAFFE	0
0x3fe	0x0DEAD	0
0x3ff	0xDECAF	1

Page table at 0x12345

VPN	PFN	V
0x123	0xBADA5	1
0x124	0x00BED	1
0x125	0xB0000	1
0x126	0x0FACE	0

Write the **physical address** in hexadecimal for each virtual address. If there is no valid translation, write "**page fault**". If you cannot determine the answer from the given information, write "**not enough information**". Converting virtual addresses to binary may be helpful.

Virtual Address	Physical Address
0xeebff000	**Answer: 0xDECAF000**
0xf6bc0190	**Answer: Not enough information**
0xee123012	**Answer: BADA5012**
0xeeaff022	**Answer: Not enough information**

NOTE: We will accept page fault as an answer for not enough information entries, however since the entire page table was not shown, you cannot guarantee that those accesses will cause a page fault. We had meant for the last entry in table to be 0xeebff022, which would have resulted in a page fault, however it seems that I made a typo.

Problem 2:

A virtually indexed, physically tagged cache with the following specifications is given:

- **4 way** set associative
- Cache data size is **64 KB**
- **64 bit** virtual addresses
- **32 bit** physical addresses
- **16 byte** blocks
- LRU replacement

A) How big should the page size of the virtual memory system be so that the cache and the TLB can be accessed in parallel? Show your work for full credit.

Tag (18 bits)	**Index (10 bits)**	**Offset (4 bits)**

VPN (50 bits)	**Offset (14 bits)**

Page size should be 2^14 bytes

B) If the page size is 4KB, which bits of the virtual page number must not change during address translation? Draw a diagram to supplement your answer.

Since the number of bits required to index into the cache are 10, and the block offset is 4 bits, 14 (least significant) bits of the virtual address must not change. The page offset is however only 12 bits long, hence **2 least significant bits of the vpn** must not change during address translation. This called 'page coloring' if you want to read more about it.