CMPE 110 Computer Architecture Fall 2016,
Homework #4
VLADOI MARIAN
mvladoi@ucsc.edu

Submission Guidelines

➔ The homework must be submitted to ecommons by 11:59pm on Wednesday, November 30th
➔ Anything later is a late submission.

Format
➔ The homework must be typed and submitted as a single file in PDF format.
➔ Homework file should be saved as cmpe110-hw1-yourcruzid.pdf.
➔ Please write your name and your UCSC email address.
➔ The homework should be "readable" without too much effort.
➔ To increase readability, use heading styles and a monospaced font (e.g. courier) for code.

Content
➔ Keep your responses coherent and organized or you may lose points.
➔ Clearly state all assumptions and provide details on how to reach a solution.
An answer without explanation gets no credit.

Points 64 = 20 + 24 + 20

1. Address Translation (20 points)

 Below, you are provided with a snapshot of a 4 entry, fully associative TLB (translationlookaside buffer) and a page table.
 ➔ If needed, you may assume that the value of the page table register is 0

PARAMETER	VALUE
page size	4 Kb
associativity	full associative
replacement policy	LRU - the entry that has not been used for the longest time will be evicted if a new entry is to be added

Initial TLB State
TLB INTERPRETATION

COLUMN	VALUE	MEANING
VALID	0 1	entry is not valid (TLB miss) entry is valid (TLB hit)
DIRTY	0 1	data has not been modified data has been modified
LRU	1 4	most recently used least recently used

TLB TABLE

VALID	DIRTY	LRU	TAG	PHYSICAL PAGE#
1	1	2	0101	0111
1	0	3	1100	1011
1	0	1	0001	1001
1	0	4	0010	0101

Initial Page Table State

ADRESS	VALID	PHYSICAL PAGE#
0000	0	Disk
0001	1	1111
0010	0	Disk
0011	1	0010
0100	1	0101
0101	0	Disk
0110	1	0111
0111	1	0110
1000	1	1001
1001	1	1000
1010	0	Disk
1011	1	1010
1100	1	1011
1101	1	1100
…...	…...	…...

Problem

Show the final state of the TLB after processing the 2 virtual address sequence given below. If a page fault happens, the physical page number to be entered into the page table would be 0000; if another page fault happens, use physical page number 0001.

VIRTUAL ADDRESSES SUPPLIED
 1. (MSB) 1010 1111 1110 1010 (LSB) (load instruction)
 2. (MSB) 0011 1100 1001 0001 (LSB) (store instruction)

The final TLB state is as shown below:
- The entry with tag 0010 has been evicted.
- Other LRU bits have been updated accordingly
- The fact that the last entry involves a write will cause the dirty bit to be set to 1.

VALID	DIRTY	LRU	TAG	PHYSICAL PAGE#
1	1	3	0101	0111
1	0	4	1100	1011
1	0	2	0001	1001
1	1	1	1010	1111

2. Cache Coherence (24 points)

Consider a processor with 4 cores. In each of the following parts you will fill out a table for a different cache coherence protocol. For the following memory references, show the state of the cache line containing the variable x in each core's cache. Consider the cacheline only in L1.

2.A VI Protocol (8 points)

Assume for this part that we use the VI protocol for cache coherence. Fill out the table as specified at the beginning of the question.

	CORE	REQUEST TYPE	C0 $ LINE STATE	C1 $ LINE STATE	C2 $ LINE STATE	C3 $ LINE STATE
1	0	Read x	State (I) Load Miss ->(V)	State (I)	State (I)	State (I)
2	1	Read x	State(I)	State (I) Load Miss -> (V)	State (I)	State (I)
3	2	Read x	State (I)	State (I)	State (I) Load Miss ->(V)	State (I)
4	3	Write x	State (I)	State (I)	State (I)	State(I) Store Miss ->(V)
5	1	Read x	State (V) Load Hit	State (I)	State (I)	State (V) Send Data ->(I)

2.B MSI Protocol (8 points)

Assume for this part that we use the MSI protocol for cache coherence. Fill out the table as specified at the beginning of the question.

	CORE	REQUEST TYPE	C0 $ LINE STATE	C1 $ LINE STATE	C2 $ LINE STATE	C3 $ LINE STATE
1	**0**	Read x	**State Invalid (I)** **Load Miss** **->State Shared (S)**	**State Invalid (I)**	**State Invalid (I)**	**State Invalid (I)**
2	**1**	Read x	**State Shared (S)**	**State Invalid (I)** **Load Miss** **->State Shared (S)**	**State Invalid (I)**	**State Invalid (I)**
3	**2**	Read x	**State Shared (S)**	**State Shared (S)**	**State Invalid (I)** **Load Miss** **->State Shared(S)**	**State Invalid (I)**
4	**3**	Write x	**State Shared (S)** **->State Invalid (I)**	**State Shared (S)** **->State Invalid (I)**	**State Shared (S)** **->State Invalid (I)**	**State Invalid (I)** **Store Miss** **-> State Modified (M)**
5	**1**	Read x	**State Shared(S)** **Load Hit**	**State Invalid (I)**	**State Invalid (I)**	**State Modified (M)** **Send Data** **-> State Shared(S)**

2.C MESI Protocol (8 points)

Assume for this part that we use the MESI protocol for cache coherence. Fill out the table as specified at the beginning of the question.

	CORE	REQUEST TYPE	C0 $ LINE STATE	C1 $ LINE STATE	C2 $ LINE STATE	C3 $ LINE STATE
1	**0**	Read x	**State Invalid (I) Load Miss ->State Shared (S) or (State Exclusive (E))**	**State Invalid (I)**	**State Invalid (I)**	**State Invalid (I)**
2	**1**	Read x	**State Shared (S)**	**State Invalid (I) Load MIss ->State Shared (S)**	**State Invalid (I)**	**State Invalid (I)**
3	**2**	Read x	**State Shared (S)**	**State Shared (S)**	**State Invalid (I) Load Miss ->State Shared(S)**	**State Invalid (I)**
4	**3**	Write x	**State Shared (S) ->State Invalid (I)**	**State Shared (S) ->State Invalid (I)**	**State Shared (S) ->State Invalid (I)**	**State Invalid (I) Store Miss -> State Modified (M)**
5	**1**	Read x	**State Shared (S) Load Hit**	**State Invalid (I)**	**State Invalid (I)**	**State Modified (M) Send Data -> State Shared(S)**

3. Virtual Memory (20 points)

An ISA supports an 16-bit, byte-addressable virtual address space. The corresponding physical memory has 256 bytes. Each page contains 32 bytes. A simple, one-level translation scheme is used and the page table resides in physical memory. The initial contents of the frames of physical memory are shown below.

PARAMETER	VALUE
virtual address size	16 bit
addressing type	byte addressable - each byte of memory has its own address
page size	32 B
physical memory size	256 B

LRU	FRAME (PHYSICAL PAGE) #	FRAME CONTENTS
0	1	Page 7
1	6	Empty
2	4	Empty
3	0	Empty
4	2	Page 1
5	3	Page 4
6	5	Page 3
	7	Page Table

Questions

A three-entry TLB that uses Least Recently-Used (LRU) replacement is added to this system. Initially, this TLB contains the entries for pages 3, 4, and 7.

INITIAL TLB

LRU	TAG (VPN)	PHYSICAL PAGE #
1	3	5
2	4	3
3	7	1

3.A TLB MISSES / PAGE FAULTS

For the following sequence of references, put a circle around those that generate a TLB hit and put a rectangle around those that generate a page fault. What is the hit rate of the TLB for this sequence of references?
Note: LRU policy is used to select pages for replacement in physical memory. Assume initial conditions where physical page number (PPN) 0 was used less recently than PPN 4, which was used less recently than PPN 6:
LRU (PPN 0) > LRU (PPN 4) > LRU (PPN 6)

Virtual Page References
(3), (7), 1, 4, [10], (10), 7, [11], (11), (7), [8], 10, (8),(7), [12], [9]
TLB Hit Rate = 7/16

3.B TLB ENTRIES
At the end of this sequence, what three entries are contained in the TLB?
Solution: 12, 7, 9

3.C FRAME CONTENTS
What are the contents of the 8 physical frames?
Solution: Pages 12, 10, 11, 8, 9, 4, 3, Page table