CMPE 200 HW#6

Due: Monday, Dec 9, 11:59pm
Total Score: /100

Instructor: Hyeran Jeon

Computer Engineering Department, San Jose State University

Cache replacement policy (15pts)

1. We are going to run the following trace of memory accesses on a 4-way set-associative cache that runs LRU replacement policy. Assume that the cache has a total of 8 blocks or lines and is initially filled with the blocks as in the "initial" column. The trace is made of consecutive block addresses dynamically accessed by a program, where each number is block address:

   0 1 0 4 2 1 4 7 5 0 6 1 4 2

   a) Show the cache block status for each access in the following table:

Trace		Initial	0	1	0	4	2	1	4	7	5	0	6	1	4	2
priority	Set															
MRU	0	0														
		4														
		2														
LRU		6														
MRU	1	7														
		5														
		9														
LRU		3														
cache miss																

   b) What is the miss ratio?

Virtual memory (30pts + 30pts)

2. In this problem, we design several structures supporting virtual memory. Consider the following assumptions that are applied to all three sub-problems:
   - Size of virtual address: 32 bits
   - Page size: 64 KB
   - Page table: 3-level page table.
     - The virtual page number is split in 3 fields: 4 bits (first-level), 8 bits (second-level) and 4 bits (third-level)
     - Size of an entry in tables: 32 bits (or 4B)
   - TLB
     - Size: 64 entries
     - Mapping: 2-way set associative

   1) Answer the following questions regarding Page Table:

      a. What is the size of a page table at each level (in bytes)?

      b. What is the amount of virtual memory covered by an entry of page table at each level?

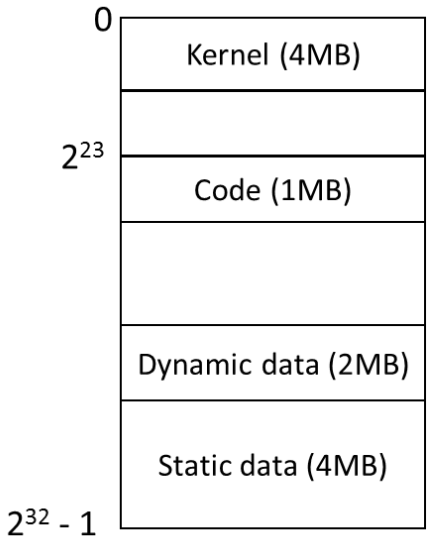
   2) Answer the following questions regarding TLB:

      a. Which bits of the virtual address are used to index the TLB?
      (hint: In the normal cache, a physical address consists of tag, index, and block offset fields. Likely, in the TLB, a virtual address is partitioned into tag, index, and page offset fields.)

      b. Which bits of the virtual address are used as tags in the TLB?

3. This problem is about the structure of page tables to support large virtual-address spaces. Consider the following assumptions:
   - Size of virtual address: 32 bits
   - Size of physical memory: 2 GB
   - Word size: 32 bits (or 4 B)
   - Page size: 4 KB
   - Size of an entry in page tables: 32 bits (or 4 B)

The following figure shows the virtual memory spaces that need to be mapped to the main memory. Assume that for all the specified spaces (kernel, code, dynamic data, static data), virtual addresses need to be translated to physical addresses.



1) What would be the size of a single-level page table?

2) Assume now a 2-level page table. The virtual page number is split into two equal-sized fields for the first-level and the second-level page tables. What is the number of tables in each level page table? What is the total size of all page tables?

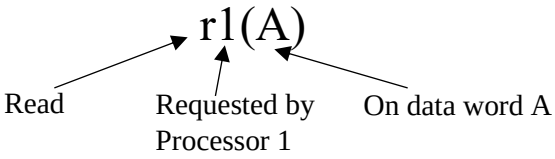
3) Repeat 2) for a 3-level page table where the virtual page number field is split into three fields for the 3-level page tables. The percentage of the first, second, and third level page table in the virtual page number field is 25%, 25%, and 50%, respectively.

Cache Coherence (25 pts)

4. Consider the following reference stream:

r1(A), r2(A), w3(B), w2(A), r3(A), r1(B), r2(B), w3(B), r3(A), r1(A)

All of the references in the stream are to the same cache block but for different data words, A and B within the same cache block. r and w indicate read and write, respectively, and the digit refers to the processor issuing the reference.



We run MOESI protocol. Assume that all caches are initially empty and the accessed cache block is not evicted while executing the reference stream. Use the following cost model:

- Read / write cache hit with no bus access: 1 cycle
- Invalidation broadcasting without requesting the cache block (BusUpgr): 10 cycles
- Request remote processor to send updated a cache block (BusRd / BusRdX): 50 cycles
- Request the memory (or next level cache) to send a cache block (BusRd): 150 cycles

Fill the following table with the coherence state of the three processors, coherence message, and cache hit/miss for each memory references. Each column should show the status of the corresponding data word (either A or B). Show the total number of cycles used for running the reference stream.

	r1(A)	r2(A)	w3(B)	w2(A)	r3(A)	r1(B)	r2(B)	w3(B)	r3(A)	r1(A)
hit/miss	M									
bus	BusRd									
State1	E									
State2	I									
State3	I									
cycles	150									

The total number of cycles : cycles