CMPE 110 Computer Architecture, Fall 2014
Homework #6

Computer Engineering
UC Santa Cruz

December 4, 2014

Name: _____

Email: _____

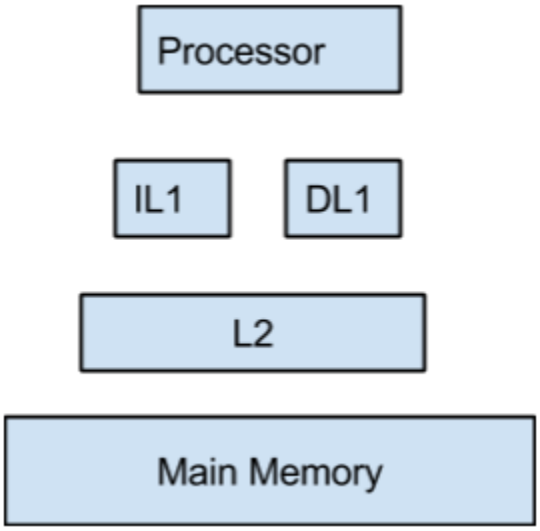
Submission Guidelines:

- This homework is due on Thursday 12/4/14.
- Bring the homework to class before 8pm
  - Anything later is a late submission
- **Please write your name and your UCSC email address**
- **The homework should be "readable" without too much effort**
  - If your handwriting is like mine, type it or risk not being graded
- Provide details on how to reach a solution. An answer without explanation has no credit. Clearly state all assumptions.
- Points: 40 = 10 + 30

Question	Part A	Part B	Part C	Part D	Part E	Part F	Part G	Total
1					-	-	-	
2								
Total								

Question 1. Cache Hierarchies (10 points)

Assume a processor uses a dedicated L1 cache for instructions (IL1) and a dedicated L1 cache for data (DL1). The processor also uses a shared L2 cache that serves as an intermediate level between each of the L1 caches and memory. The figure below shows the hierarchy:



The processor has a cycle time of 1 ns (i.e., operates at 1 GHz frequency). A hit to either IL1 or DL1 takes 1 cycle (time to look up the cache and return the data to processor). IL1 has 3% miss rate. DL1 has 12% miss rate. Assume load/store instructions comprise 25% of the total instructions. Hit access to L2 from either of the IL1 or DL1 caches takes 4 cycles (time to lookup the cache and return the data to either of higher level caches). L2 has total miss rate of 30%. From L2, it takes 20 cycles to access memory.

Question 1.A AMAT (2 points)

What is the average memory access time?

Question 1.B No Caches (2 points)

Assume CPI = 1 if the processor has no memory stalls. Without the caches, each memory access would take 22 cycles. What is the CPI of the processor without any caches?

Question 1.C No L2 Cache (3 points)

Assume CPI = 1 if the processor has no memory stalls. without the L2 cache, each memory access from L1 caches would take 21 cycles. What is the CPI of the processor without the L2 cache?

Question 1.D All Caches (3 points)

Assume CPI = 1 if the processor has no memory stalls. What is the CPI of processor with the all the caches? Remember that it takes 20 cycles to access memory from L2.

Virtual Memory (30 points)

Assume we have a processor that support virtual memory with 4KB page size. The processor uses one page table per program to track the mapping of virtual addresses to physical addresses, and has 28-bit virtual address. Assume we have 4-entry fully associative TLB, with the content specified in table below.

Valid	Tag	Physical Address
0	0x10	0x11
1	0xF	0x1F
1	0x3	0x13
1	0x6	0x16

Assume we start with the following series of accesses: 0x6D10, 0x9000, 0xF200, 0xF800, 0x8800, 0x0000, 0x3100

Also assume the following page table content:

Index	Valid	Physical Mapping
0	0	disk
1	1	0x11
2	1	0x12
3	1	0x13
4	0	disk
5	0	disk
6	1	0x16
7	1	0x17
8	1	0x18
9	0	disk
A	0	disk
B	0	disk
C	0	disk
D	0	disk
E	0	disk
F	1	0x1F

Question 2.A Program Table Entries (4 points)

Assume each entry in page table is 4-bytes. How many entries does the processor need to have in the page table for each program, and what is the total page table size? How about if the processor supports 16KB page size?

Question 2.B TLB and Page Table States (4 points)

Given the address sequence and the initial state of TLB and page table, what would be the final state of the TLB and page table? Assume ideal LRU replacement for the TLB. Specify what accesses hit on TLB, what addresses hit on page table, and what addresses are page fault.

Question 2.C Two-Way Set-Associative TLB (4 points)

Assume the TLB is 2-way set-associative instead of fully-associative. Given the address sequence and the initial state of TLB and page table, what would be the final state of the TLB and page table? Assume ideal LRU replacement for the TLB. Specify what accesses hit on TLB, what addresses hit on page table, and what addresses are page fault.

Question 2.D Direct-Mapped TLB (4 points)

Assume the TLB is direct-mapped instead of fully-associative. Given the address sequence and the initial state of TLB and page table, what would be the final state of the TLB and page table? Assume ideal LRU replacement for the TLB. Specify what accesses hit on TLB, what addresses hit on page table, and what addresses are page fault.

Question 2.E Initial TLB and Page Table state (4 points)

Assume the processor supports 16KB page size. Update the initial TLB state. Note that with 16KB page size, more bits are used for page offset, and less bits are used for tag in the TLB or for index in page table. Also note that TLB cannot have a entry which does not exist in the page table.

Question 2.F Final TLB state (4 points)

This question depends on the the results from part E. We are continuing on the assumption that the processor supports 16KB page size. What would be the state of TLB and Page table after running the given address sequence?

Question 2.G Trade-Offs (6 points)

Discuss the advantages and disadvantages of a larger page size.