16.482 / 16.561: Computer Architecture and Design

Spring 2014

Homework #6

Due **Tuesday, 6/17/14**

Notes:

- While typed submissions are preferred, handwritten submissions are acceptable.
- Any handwritten solutions that are scanned and submitted electronically must be clearly legible and combined into a single file—simply sending a picture of each scanned page is not an acceptable form of submission.
- This assignment is worth a total of 100 points.

1. Multithreading (30 points) Given the three threads shown below, determine how long they take to execute using (a) fine-grained multithreading, (b) coarse-grained multithreading, and (c) simultaneous multithreading.

For coarse-grained multithreading, switch threads on any stall longer than 1 cycle. (Note that you must determine the number of stall cycles based on dependences between instructions.) For simultaneous multithreading, treat thread 1 as the preferred thread, followed by thread 2 and thread 3.

Assume you are using a processor with the following characteristics:

- 6 functional units: 3 ALUs, 2 memory ports (load/store), 1 branch
- In-order execution
- The following instruction latencies:
  - L.D/S.D: 4 cycles (1 EX, 3 MEM)
  - ADD.D/SUB.D: 2 cycles
  - All other operations: 1 cycle

Thread 1:

```
L.D F0, 0(R1)
L.D F2, 8(R1)
ADD.D F4, F0, F2
SUB.D F6, F2, F0
S.D F4, 16(R1)
S.D F6, 24(R1)
DSUBUI R1, R1, #32
BNEZ R1, loop
```

Thread 2:

```
DADDUI R1, R1, #24
ADD.D F2, F0, F4
ADD.D F4, F6, F8
ADD.D F6, F0, F6
S.D F2, -24(R1)
S.D F4, -16(R1)
S.D F6, -8(R1)
BEQ R1, R7, end
```

Thread 3:

```
L.D F6, 0(R1)
ADD.D F8, F8, F6
S.D F8, 8(R1)
DADDUI R1, R1, #16
BNE R1, R2, loop
L.D F6, 0(R1)
ADD.D F8, F8, F6
S.D F8, 8(R1)
DADDUI R1, R1, #16
BNE R1, R2, loop
```

2. (20 points) For each of the following memory hierarchies, calculate the average memory access time. If you end up with a fractional number of cycles, round up—there isn't much you can do (besides read/write the register file) in half a cycle!

a. The cache takes 1 cycle to access and has a 10% miss rate, main memory takes 250 cycles to access and has a 6% miss rate, and the disk takes 40,000 cycles to access.

b. The cache takes 3 cycles to access and has a 93% hit rate, main memory takes 450 cycles to access and has a 97% hit rate, and the disk takes 75,000 cycles to access.

c. This problem deals with a multi-level cache, as discussed in class. The cache levels are listed in terms of their order in the memory hierarchy—an access initially goes to the level 1 (L1) cache. If there is a miss in the L1 cache, you then check the level 2 (L2) cache, then the level 3 (L3) cache, and then main memory.

   The L1 cache takes 1 cycle to access, with a 95% hit rate. The L2 cache takes 20 cycles on each access and has a 97% hit rate. The L3 cache takes 60 cycles to access and has a 99% hit rate. Main memory takes 500 cycles to access, with an 85% hit rate, while the disk takes 45,000 cycles to access.

3. (50 points) You are given a system which has a 16-byte, write-back cache with 4-byte blocks. The cache is 2-way set associative.

a. (10 points) If each address uses 8 bits, what size are the offset, index, and tag?

b. (40 points) Assume the initial memory state shown below for the first 16 bytes and last 16 bytes of memory (<u>note:</u> all addresses are listed in decimal):

Address		Address	
0	20	240	15
1	8	241	67
2	27	242	78
3	3	243	19
4	12	244	26
5	44	245	99
6	34	246	9
7	5	247	4
8	110	248	101
9	72	249	71
10	38	250	89
11	127	251	93
12	126	252	106
13	85	253	107
14	2	254	1
15	6	255	11

For each access in the sequence listed on the next page, show the cache state, indicate what register (if any) changes, and indicate if any memory blocks are written back and if so, what addresses and values are written. The cache state should carry over from one access to the next. As above, assume 8-bit addresses. Also, assume the cache is initially empty.

Instruction sequence for Question 3b:

```
lb $t0, 2($zero)
sb $t0, 0($zero)
lb $t1, 240($zero)
sb $t1, 3($zero)
lb $t0, 8($zero)
sb $t1, 248($zero)
sb $t0, 11($zero)
lb $t1, 250($zero)
lb $t3, 249($zero)
lb $t4, 243($zero)
```