

16.482 / 16.561: Computer Architecture and Design

Spring 2015

Homework #6

Due Thursday, 3/26/15

Notes:

- While typed submissions are preferred, handwritten submissions are acceptable.
- Any electronic submission must be in a single file. Archive files will not be accepted.
- Electronic submissions should be e-mailed to Dr. Geiger at Michael_Geiger@uml.edu.
- This assignment is worth a total of 100 points.

All problems deal with the following three threads. Note that you must determine the number of stall cycles between dependent instructions based on the instruction latencies given below:

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
```

Assume you are using a processor with the following characteristics:

- 6 functional units: 3 ALUs, 2 memory ports (load/store), 1 branch
- 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

1. (25 points) Determine how long the code will take using fine-grained multithreading. Assume the processor uses in-order scheduling.

2. (25 points) Determine how long the code will take using coarse-grained multithreading. Assume the processor uses in-order scheduling, and switch threads on any stall longer than 1 cycle (stalls of 2 or more cycles).

3. (25 points) Determine how long the code will take using simultaneous multithreading. Assume the processor uses in-order scheduling, and that thread 1 is the preferred thread, followed by threads 2 and 3.

4. (25 points) Determine how long the code will take using simultaneous threading if the processor uses dynamic (out-of-order) scheduling. Assume that the processor can issue four instructions per thread in each cycle, and that the order of preferred threads remains the same.