Introduction to Pipelining

Tom Kelliher, CS 220

1 Administrivia

Today's Objectives

1. Calculate the speed-up advantage of pipelining.
2. Discuss implementation issues involved with pipelining.
3. Understand pipeline hazards

Next Up

Read 7.4–7.6

1. Understand the effect of branches on pipelined performance.
2. Study techniques for reducing the effect of branches on pipelined performance.

2 Warm-Up

1. Compared to a flow-through processor, the execution of a single instruction in a pipelined processor takes

   (a) less time

   (b) the same amount of time

   (c) more time

2. If a flow-through processor is modified to be pipelined, the cycle time of the pipelined version will be f/n, where f is the cycle time of the flow-through processor and n is the number of pipeline stages.

   True/False

3. Adding additional pipeline stages to a pipelined processor will always result in improved performance.

   True/False

4. In a pipelined processor, the execution of two given instructions will always take the same number of cycles.

   True/False

5. A pipeline of n stages and an instruction sequence of i instructions with no branch instructions will require how many cycles to execute?

   (a) n

   (b) i

   (c) $n + i$

   (d) i/n

   (e) None of the above.

6. A pipeline of n stages and an instruction sequence of i instructions with branch instructions will require how many cycles to execute?

   (a) n

   (b) i

   (c) $n + i$

   (d) i/n

   (e) None of the above.

7. The instruction sequence

```
add r4, r4, #1
ldr r0, [r1, r4, lsl #2]
```

contains what kind of hazard?

(a) WAW

(b) WAR

(c) RAW

3 Problem Set 14.0

1. The pipelining speed-up formula is

$$S = \frac{ni}{n + (i - 1)}$$

   In what ways is this formula flawed?

2. A processor executes an instruction in the following six stages. The time required by each stage in picoseconds (1 ps = 10^{-3} nanoseconds (ns) = 10^{-12} seconds).
   IF, instruction fetch, 300 ps
   ID, instruction decode, 150 ps
   OF, operand fetch, 250 ps
   OE, execute 350 ps
   M, memory access, 700 ps
   OS, operand store (register writeback), 200 ps

   (a) What is the time to execute an instruction if the processor is not pipelined?

   (b) What is the time taken to fully execute an instruction assuming that there is an additional penalty of 20 ps per stage for pipeline latches?

   (c) Does this penalty affect instruction latency, instruction throughput, or both. Explain.

   (d) Once the pipeline is full, what is the average instruction rate?

   (e) If 25% of the instructions are taken branches that cause a three cycle penalty, what is the effective instruction execution time?

3. Assuming a four stage pipeline, identify *all* of the RAW data dependencies in the following code. Which dependencies are data hazards that will be resolved by forwarding? Which dependencies are data hazards that will cause a bubble?

```
add r3, r4, r2
sub r5, r3, r1
ldr r6, [r3, #20]
add r7, r3, r6
```