CS 520: Computer Architecture and Organization
Fall 2015: Prof. Timothy N. Miller

Homework 1

Due date: Wednesday, September 16

As always homework assignments are to be worked on individually. Sharing of solutions with other students is prohibited.

Problem 1

(10 points)
The Java VM and the Dalvik VM compile Java code to two different machine-independent low-level bytecode representations. You will likely need to look these up on Google and Wikipedia. A core design difference pertains to how VM instruction operands are specified. What is that difference? Briefly explain some advantages and disadvantages to one design choice or the other.

Problem 2

(5 points) Brief answers:
3.1: How does a cache take advantage of spatial locality?
3.2: How does a cache take advantage of temporal locality?

Problem 3

(5 points)
What do you get when you multiply time (t) by power (P)?

Problem 4

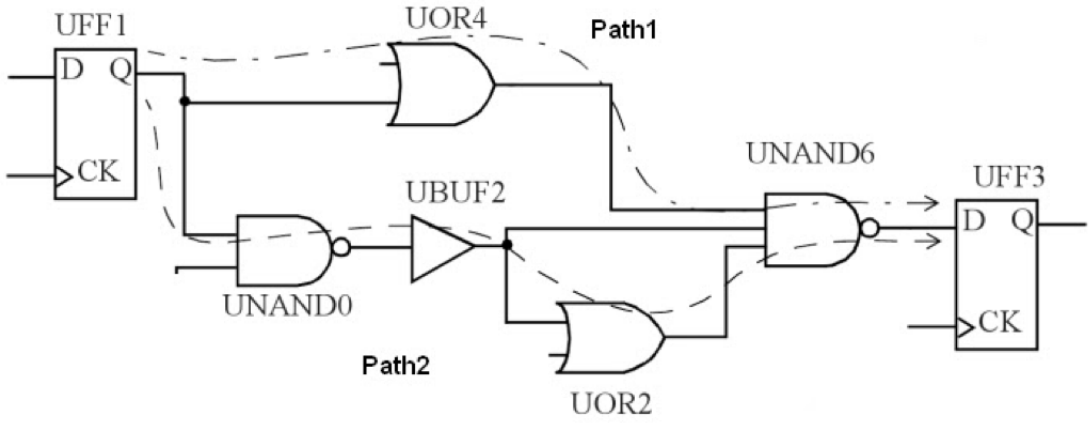
(10 points)
5.1: Briefly name two benefits of pipelining.
5.2: In the limit (given an infinite number of instructions), derive a formula for the speedup (S) that can be achieved from a given number of pipeline stages (K).

Problem 5

(5 points)
Explain the distinction between these two types of registers:

- Pipeline register
- Architectural register

Figure 1: Example circuit with pipeline registers



Problem 6

(15 points)

Consider Figure 1. After the rising edge of clock signal CK, output Q of UFF1 becomes valid, and signals propagate through the logic. Eventually, the D input to UFF3 becomes stable. Only after that time is it sensible to have another clock edge.

7.1: Of Path1 and Path2, which one dictates the minimum clock period that will be valid for both paths?

7.2: What would happen if the clock period used were shorter than the maximum signal propagation time through the circuit?

7.3: Why do electrical signals take time to propagate through combinatorial logic?

(If you have forgotten how edge triggered flip flops work, you may also want to read https://courses.cs.washington.edu/courses/cse370/08wi/pdfs/lectures/14-Flip-flops.pdf)

Problem 7

(5 point)

What is a pipeline stall, and what might cause one to occur?

Problem 8

(20 points)

We often define logic gate delay in terms of "FO4", which is the switching delay through a CMOS inverter whose output is connected to the input of four other logic gates. (FO4 means "fan-out of four".) The FO4 delay can often be treated as a constant for a given technology, so **we will treat it as a *unit of time***.

Let us design a CPU where typical pipeline stage circuit delay is 16 FO4, a slave latch has a delay of 1 FO4, and a master latch has a delay of 1 FO4. The latch overhead is constant for each pipeline stage.

Question: What is the maximum clock rate of this processor in units of 1/FO4?

In a later revision, we wish to more deeply pipeline the CPU. For instance, instead of pipeline stages with delay of 16 FO4, we will divide each 16 FO4 stage in the original design into two stages with delay of 8 FO4 in the new design.

Question: What is the maximum clock rate of the new design in units of 1/FO4?

Question: What is the maximum speedup according to Amdahl's law?

(continued on next page…)

Answer these questions in a table of this form:

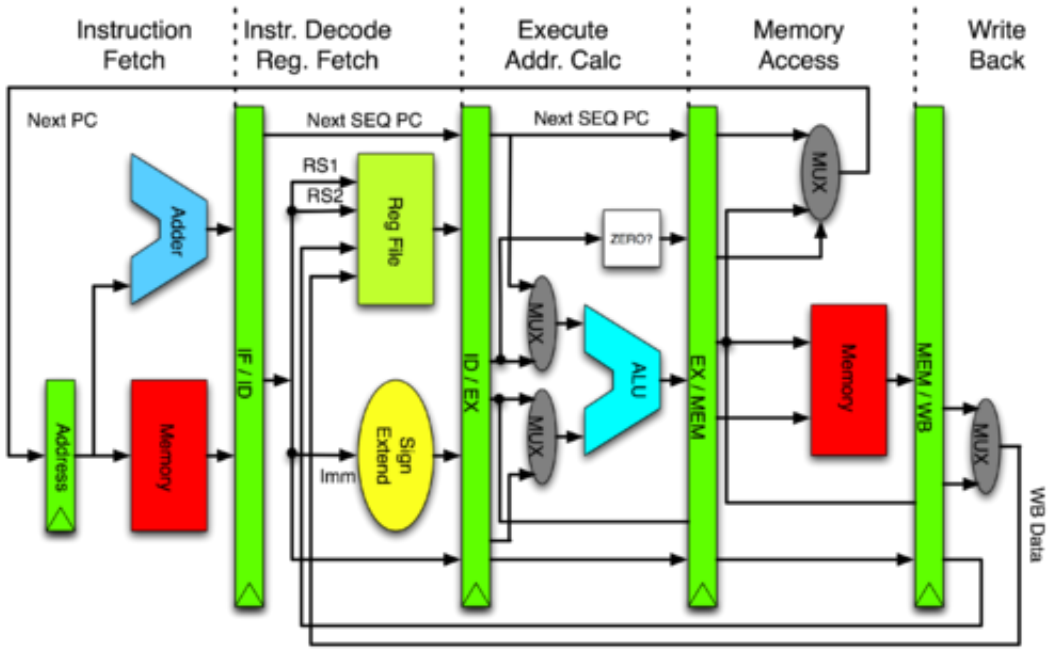
Pipeline logic delay (FO4)	*Total pipeline stage delay (FO4)*	*Clock rate (1/FO4)*	*Maximum speedup vs. 16 FO4*
16			1.0
8			
4			
2			
1			

Problem 9

(5 points)
In a reservation table:

A. What does it mean if there are two X's in a row?
B. What does it mean if there are two X's in a column?

Figure 2: More detailed MIPS pipeline



Problem 10

(20 points)
See Figures 2 and pages 28 and 33 of the class notes. In the D/RF stage, the register file is a *memory block* with two read ports for reading instruction source operands and one write port for write-back of results.

Consider an architecture where register file memory blocks are restricted to having two ports: one read port and one write port. Nevertheless, we want to support an execution model similar to the familiar pipeline of Figures 3, where each instruction can potentially have two input operands and one output operand. We will solve this problem by providing a second register file.

Problem: Diagram a pipeline similar to Figure 3 where the two register files are placed in parallel. Expand the detail on appropriate pipeline stages to show the memory blocks and their connections for reads and write-back. Describe any implications with regard to energy, area, and performance.