CSE 560 – Practice Problem Set 4

1. Use the following RISC code fragment:

```
loop:  load   r1, 0(r2)
       addi   r1 ← r1, #1
       store  0(r2), r1
       addi   r2 ← r2, #4
       sub    r4 ← r3, r2
       bnez   r4, loop
```

   You may assume that the initial value of r3 is r2 + 396.

   (a) Show the timing of this instruction sequence (i.e., draw a pipeline diagram) for our 5-stage pipeline without any forwarding or bypassing hardware but assuming a register read and a write in the same clock cycle "forwards" through the register file. Assume that the branch is handled by flushing the pipeline (use the notation **f** in the pipeline diagram). If all memory references hit in the cache (i.e., take one cycle), how many cycles does this loop take to execute?

   (b) Show the timing of this instruction sequence for our 5-stage pipeline with normal forwarding and bypassing hardware. Assume that the branch is handled by predicting it as not taken. If all memory references hit in the cache, how many cycles does this loop take to execute?

2. Your job is to build the latest and greatest handheld device, and your company has purchased the rights to use a family of processor designs (notice the similarity to ARM's business model, think of your company as a customer of ARM). The immediate task is to choose which micro-architecture (within the family) that best meets the needs of your new device. The two choices are B and L (for big.LITTLE, ARM systems that support two different types of cores in a single chip, a point that is irrelevant to the current problem).
   - Since both designs under consideration (B and L) are ARM processors, the compiler and the ISA can be considered to be the same.
   - Both designs use the same 5-stage pipeline with stages F, D, X, M, W
   - The mix of instruction types is as follows:
     i. 20% conditional branch instructions
     ii. 20% load instructions
     iii. 10% store instructions
     iv. 50% data manipulation (ALU) instructions (of which, 20% of these are floating point instructions)
   - In design B, floating point instructions are fully pipelined and therefore have a negligible number of pipeline stalls (i.e, you can ignore them).
   - In design L, the floating point execution hardware is not pipelined, resulting in an average stall of 3 clock cycles for each floating point instruction.

- In design B, a hybrid branch predictor is present in the fetch (F) pipeline stage that predicts correctly 90% of the time. Mispredictions trigger a 2-clock cycle flush.
- In design L, a simpler branch predictor is present in the decode (D) pipeline stage that predicts correctly 80% of the time. Mispredictions trigger a 2-clock cycle flush. Don't forget that correct predictions will have a 1-clock cycle bubble for this design.
- Both designs can be clocked at 2 GHz.

(a) Write an analytical expression for CPI_B and CPI_L, including identifiable components for each instruction type.
(b) Substitute numbers into the above expressions, giving numerical values for CPI_B and CPI_L.
(c) For a benchmark program with 10 billion (10^{10}) instructions, what is the time to completion for each design?
(d) Which processor is faster, and what % faster is it?