ECE341 Homework No. 6 Solution

Problem No. 1

Cycles 1-3: PC_enable = 0, PC = 0x34000
Cycle 4: PC_enable = 1, PC changes to 0x34004 after cycle-4
Cycle 5: PC_enable = 0, PC = 0x34004

Problem No. 2

(a) For Load, Or and Store instructions, *PC_enable* = 0
For Call_Register instruction, *PC_enable* = 1

(b) For Load instruction, *Mem_read* = 1, *Y_select* = 01
For Or instruction, *Mem_read* = 0, *Y_select* = 00
For Store instruction, *Mem_read* = 0, *Y_select* = don't care
For Call_register instruction, *Mem_read* = 0, *Y_select* = 10

(c) For Load instruction, *RF_write* 1, *C_select* = 00
For Or instruction, *RF_write* = 1, *C_select* = 01
For Store instruction, *RF_write* = 0, *C_select* = don't care
For Call_register instruction, *RF_write* = 1, *C_select* = 10

Problem No. 3

(a) The key idea behind pipelining is to arrange the processor hardware in such a way that *multiple* instructions can be processed at the same time.
Pipelining does not cause an individual instruction to execute faster. However, by overlapping the execution of multiple instructions, pipelining increases the instruction completion rate (no. of instructions completed per second or *throughput*), and thereby improves processor performance.

(b) Without pipelining: Each instruction will require 5 cycles. There will be no overlap amongst successive instructions.
Number of cycles = 150 * 5 = 750

With pipelining: Each pipeline stage will process a different instruction every cycle. First instruction will complete in 5 cycles, then one instruction will complete in every cycle, due to ideal overlap.
Number of cycles = 5 + ((150-1)*1) = 154

(c) Speedup obtained with pipelining = Exec. Time without pipelining / Exec. Time with pipelining
= 750 / 154 = **4.87**

(d) For a 5-stage pipeline, the upper bound on the obtainable speedup is **5**.

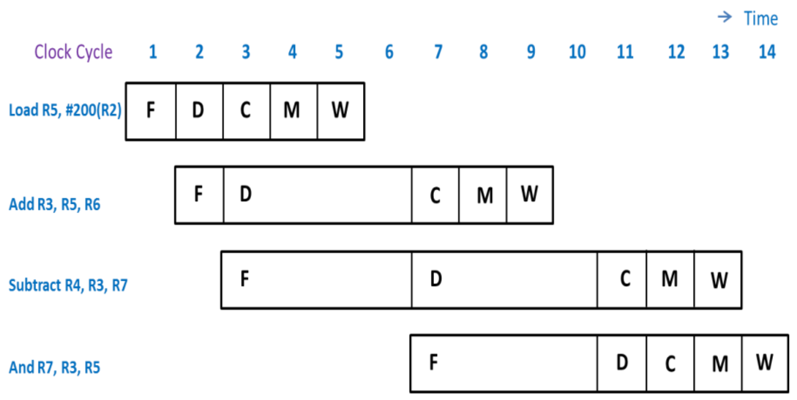
The speedup in part(c) is less than 5 because there is an initial overhead of filling the pipeline with instructions. It takes 5 cycles for the first instruction to complete, resulting in loss in throughput during the first 4 cycles. As long as these first 4 cycles are significant relative to the total execution time, the speedup would be significantly less than 5.

Problem No. 4

(a) There are four data dependencies in the given instruction sequence:

The Add instruction depends on the Load instruction for register R5's value
The Subtract instruction depends on the Add instruction for register R3's value
The And instruction depends on the Add instruction for register R3's value
The And instruction depends on the Load instruction for register R5's value

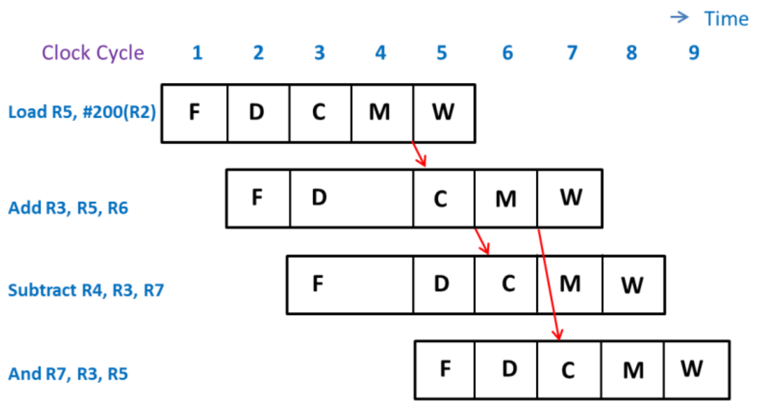
(b) The following figure shows the flow of instructions through the pipeline during each clock cycle:



(c) Execution time for non-pipelined processor = 4 instructions * 5 cycles per instruction = 20 cycles
Execution time for pipelined processor = 14 cycles (from the above figure)
Speedup due to pipelining = 20/14 = **1.43**

Problem No. 5

(a) The following figure shows the flow of instructions through the pipeline:



The arrows show the three forwarding operations that take place in the above pipeline:

(i) The Load instruction forwards its result from register “RY” (the pipeline register between “M” and “W” stages) to the ALU input “InA” in the “C” stage.

(ii) The Add instruction forwards its result from register “RZ” (the pipeline register between “C” and “M” stages) to the ALU input “InA” in the “C” stage.

(iii) The Add instruction forwards its result from register “RY” (the pipeline register between “M” and “W” stages) to the ALU input “InA” in the “C” stage.

These three forwarding operations mitigate the first three data dependences mentioned in Problem No. 1(a). The fourth data dependency (between Load and And instructions) does not require operand forwarding because the Load instructions has already completed (cycle no. 5) before the And instruction enters the decode stage (cycle no. 6).

(b) Execution time without forwarding = 14 cycles (from Problem 1(b))
Execution time with forwarding = 9 cycles (from Problem 2(a))
Speedup due to pipelining = 14/9 = **1.55**