CS 250 Spring 2017 Homework 07
Due 11:58pm Friday, March 10, 2017
Submit your typewritten file in PDF format to Blackboard.

1. Text exercise 5.8. Assume that after the last instruction of the given code snippet is fetched, that the instructions fetched are denoted "???". Note that the pipeline of Figure 5.5 cannot read an operand that is written into the register unit on clock cycle N until clock cycle N+1. The comment to line 3 of the assembly code should read "# put 20 in register 9".
   Answer: Let your answer start as follows:

Fetch Clock Cycle	Instr.	Operands	Comment
1	loadi	r7, 10	# Put 10 in register 7, a load immediate instruction, # where 10 is carried within a field of the loadi instruction # and will be written into the register unit on clock cycle 5.
2	loadi	r8, 16	# Put 16 in register 8,
3	loadi	r9, 20	# Put 20 in register 9
4	nop		# wait for register 7 and register 8
5	nop		# wait for register 7 and register 8
6	nop		# wait for register 7 and register 8
7	addrr	r10, r7, r8	# add register 7 and register 8, store in register 10
8	movr	r12, r9	# copy register 9 value to register 12
9	movr	r11, r7	# copy register 7 value to register 11
10	nop		# wait for register 11
11	nop		# wait for register 11
12	nop		# wait for register 11
13	nop		# wait for register 11
14	addri	r14, r11, 27	# add register 11 and 27, store in register 14
15	addrr	r13, r12, r11	# add register 11 and register 12, store in register 11

2. Text exercise 6.4.
   From Figure 6.6, we know that an offset value is treated as a constant. Adding a register address would result in that register value never being retrieved.

3. Text exercise 6.6.
   The circuit is not an infinite loop because the Adder takes another input along with the output of the program counter. If a value is not supplied, the circuit will continue outputting the same instruction from memory.

4. Explain how a CPU with two execution modes uses those modes to run an operating system and also execute code written by a user.
   User code and OS code are executed in two different modes. OS code can execute all possible instructions since it has the highest privilege. User code will have some instructions deemed illegal since it would be unsafe to give a user complete control over

the processor. As such, User code has less privilege than OS code.

5. You are designing a microcoded version of the CPU in chapter 6 of our textbook. You have built a fully-functional circuit that includes all the components shown in Figure 6.9. What additional component do you need? To what should its input and its output be connected in Figure 6.9?
   If the microcoded version of the CPU is done via Vertical Microcode, there needs to be something to convert macro instructions to micro instructions. This could be done with an FPGA or additional microcontroller that would be connected to instruction memory. If using a Horizontal Microcode, the CPU needs a data transfer mechanism and macro general purpose registers. The data transfer mechanism output would to the ALU and the Register Unit. It's input would be instruction memory.

6. Assume that the IF (instruction fetch), ID (instruction decode), EX (instruction execute), MEM (data memory access, read or write), and WB (write-back to a register) stages of a 5-stage pipelined version of Figure 6.9 have propagation delays of 20, 30, 50, 50, and 10 nanoseconds (ns), or 10^{-9} seconds, respectively.
   a. What is the fastest clock rate for this pipeline?
   $$\frac{1}{20 + 30 + 50 + 50 + 10} = \frac{1}{160} = 0.00625GHz = 6.25MHz$$

   b. Assume that the stage registers required for pipelining each have 3 ns of propagation delay to record an input. What was the propagation delay of the processor before pipelining?
   $(20 - 3) + (30 - 3) + (50 - 3) + (50 - 3) + (10 - 3) = 17 + 27 + 47 + 47 + 7$
   $142ns$

   c. Ideally, how much faster is this processor once it is pipelined?
   20%

7. Could the multiplexer controlled by the signal RegDst in Lecture 15, slide 52 be moved from the EX stage to the MEM stage without compromising the correct functioning of the pipeline? Could this multiplexer be moved all the way to the WB stage and have correct operation of the pipeline be preserved? Why does the design in this slide place the multiplexer in the EX stage?
   It could be moved back to the MEM stage but not the WB stage because the result is expected at the end of the MEM stage. The multiplexer is placed in the EX stage because it gets it's result at the end of the ID stage, so the EX stage is the first possible stage it can execute.