378 MIDTERM

Spring 2006

Name: ____________________

Q1 (10pts): ____________

Q2 (10pts): ____________

Q3 (10pts): ____________

Q4 (10pts): ____________

Q5 (20pts): ____________

Q6 (20pts): ____________

Q7 (20pts): ____________

Total (out of 100): ____________

Question 1 (10 points): Write a program that computes $t0 \times t1$ and leaves the result in register $t2$. You can use *only the following instructions*: ADDI, ADD, SUB, SLL, SRL, NAND, BEQ. Do not worry about efficiency or following MIPS calling conventions. Assume that before your program runs $t0$ and $t1$ contain the values you need to multiply. Furthermore, you can assume that both $t0$ and $t1$ are positive integers. Do not worry about overflow.

Question 2 (10 points): Write a program that computes $t0 \times t1$ and leaves the result in register $t2$. You can use *only the following instructions*: ADD, SUB, SLL, SRL, NAND, BEQ. Do not worry about efficiency or following MIPS calling conventions. Assume that before your program runs $t0$ and $t1$ contain the values you need to multiply. Furthermore, you can assume that both $t0$ and $t1$ are positive integers. Do not worry about overflow. (Note this question is different than question 1).

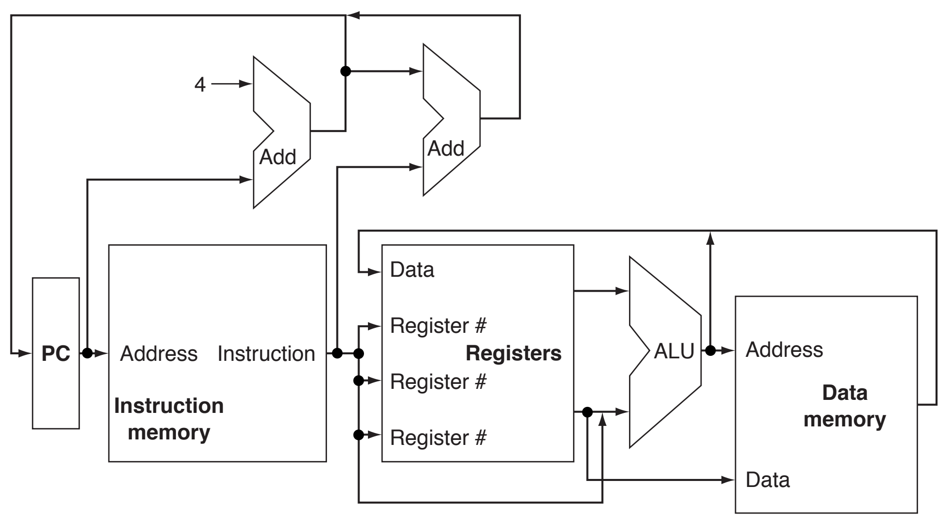
Question 3 (10 points): Write a program that computes $t0 \times t1$ and leaves the result in register $t2$. You can use *only the following instructions*: ADD, SLL, SRL, NAND, BEQ. Do not worry about efficiency or following MIPS calling conventions. Assume that before your program runs $t0$ and $t1$ contain the values you need to multiply. Furthermore, you can assume that both $t0$ and $t1$ are positive integers. Do not worry about overflow. (Note this question is different than question 2).

Question 4 (10 points): Write a program that computes $t0 \times t1$ and leaves the result in register $t2$. You can use *only the following instructions*: ADD, SLL, SRL, NAND. Do not worry about efficiency or following MIPS calling conventions. Assume that before your program runs $t0$ and $t1$ contain the values you need to multiply. Furthermore, you can assume that both $t0$ and $t1$ are positive integers. Do not worry about overflow. (Note this question is different than question 3).

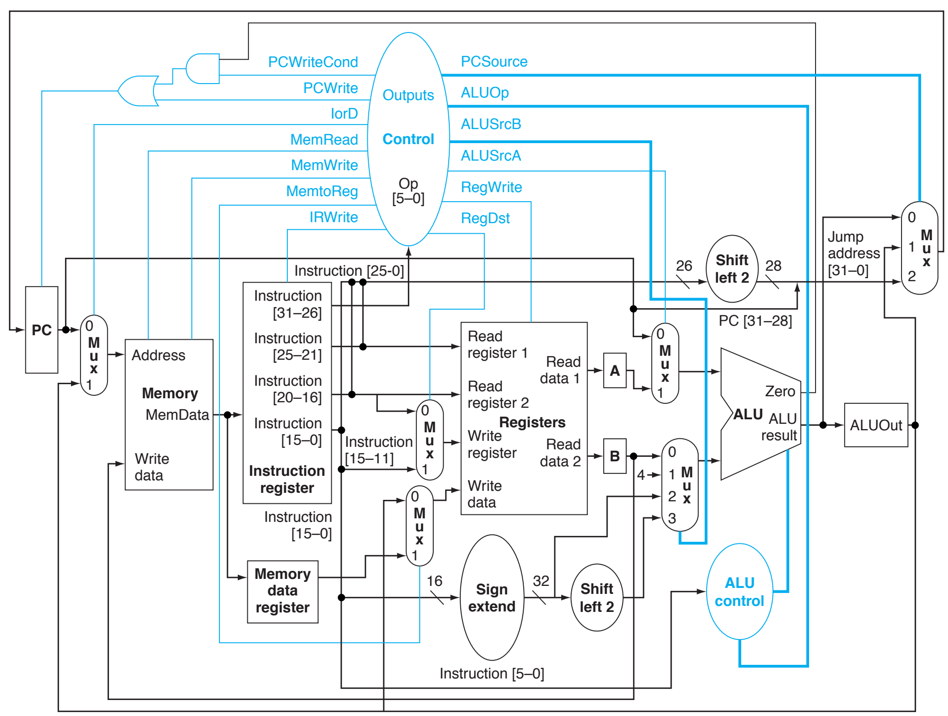
Question 5 (20 points): Modify the following machine to support the following new instruction: JRLZ. It stands for Jump-to-register if less than zero. The instruction is used in assembly like (as an example):

JRLZ $T0, $T1

The result of the instruction is to cause the machine to branch to the address specified in $t0 if $t1 is less than zero.



Question 6 (20 points): The following diagram looks a lot like Lab 1, but is subtly different. Explain how to make this processor support loads and stores. Write the control signals required to do this and explain their operation.



Question 7 (20 points): Suppose the following instructions were fetched:

ADD	\$T1, \$T3, \$T4
LW	\$T0, 0(\$T1)
BEQ	\$T0, \$T2, SOMEWHERE

Suppose the LW instruction is in the memory stage. Draw on the diagram where the BEQ and ADD instructions are (which stage). Label all forwarding networks that are being used in that clock cycle and indicate what data values are being forwarded. It is possible you will have to modify the diagram to support your answer and make the processor properly execute. If this is the case, draw these modifications too.