Your Name:_____

Compsci/ECE 250 Midterm II Exam

Prof. Alvin R. Lebeck
10:05am to 11:20am

One 8.5" x 11" sheet, no calculators, no computers/tablets/phones, no electronic communication, nothing but a pen or pencil

Answer all questions, state all your assumptions, clearly mark your final answer.

Be sure you have all six (6) pages of the exam.

Write your name on each page of the exam.

Read all questions and use your time accordingly.

Good Luck!

By signing your name below you agree to abide by the Duke Community Standard

Name:_____ **NetID**:_____

1. _____ (25 pts)
2. _____ (25 pts)
3. _____ (25 pts)
4. _____ (25 pts)

Total _____ **(100 pts)**

Your Name:_____

Question 1. (25 pts) Combinational Logic Design

Design a combinational circuit that takes as input a 3-bit 2's complement binary value and computes if the value is less than -2 or greater than 3.

[9 pts] Write the truth table for this circuit.

[8 pts] Write the boolean expression for your output.

[8 pts] Draw the circuit diagram for an implementation that uses only AND, OR and NOT gates. There are no constraints on the number of inputs to the gates.

Your Name:_____

Question 2. (25 pts) Finite State Machines

You are tasked with designing a finite state machine to identify potential malicious attacks. Our system starts in an iniatial state, and then receives a continuous stream of bits as input. A malicious attack occurs whenever one of the bit patterns 01 or 11 appears **anywhere in the continuous stream**. Your output is a single bit that indicates if one of the malicious patterns occurs in the two most recent bits. Note your system runs continuously.

a.) [10 pts] Draw the state diagram for your predictor, clearly label states, transitions input, and output.

b.) [8pts] Write the truth table for your finite state machine.

c.) [7pts] Draw the circuit diagram for your finite state machine malicous attack classifier.

Your Name:_____

Question 3. (25 pts) Datapath

Modify the single cycle datapath and control (use the figure on the next page) to support the jump register+register instruction **jregs rs, rt.** Where a new PC value is set to the result of adding register **R[rs]** and **R[rt]**. The register transfer language for this instruction is:

PC = R[rs] + R[rt];

The ALUOp0 and ALUOp1 signals control the ALU in the following manner. Note this is a subset of the ALU signals actually required.

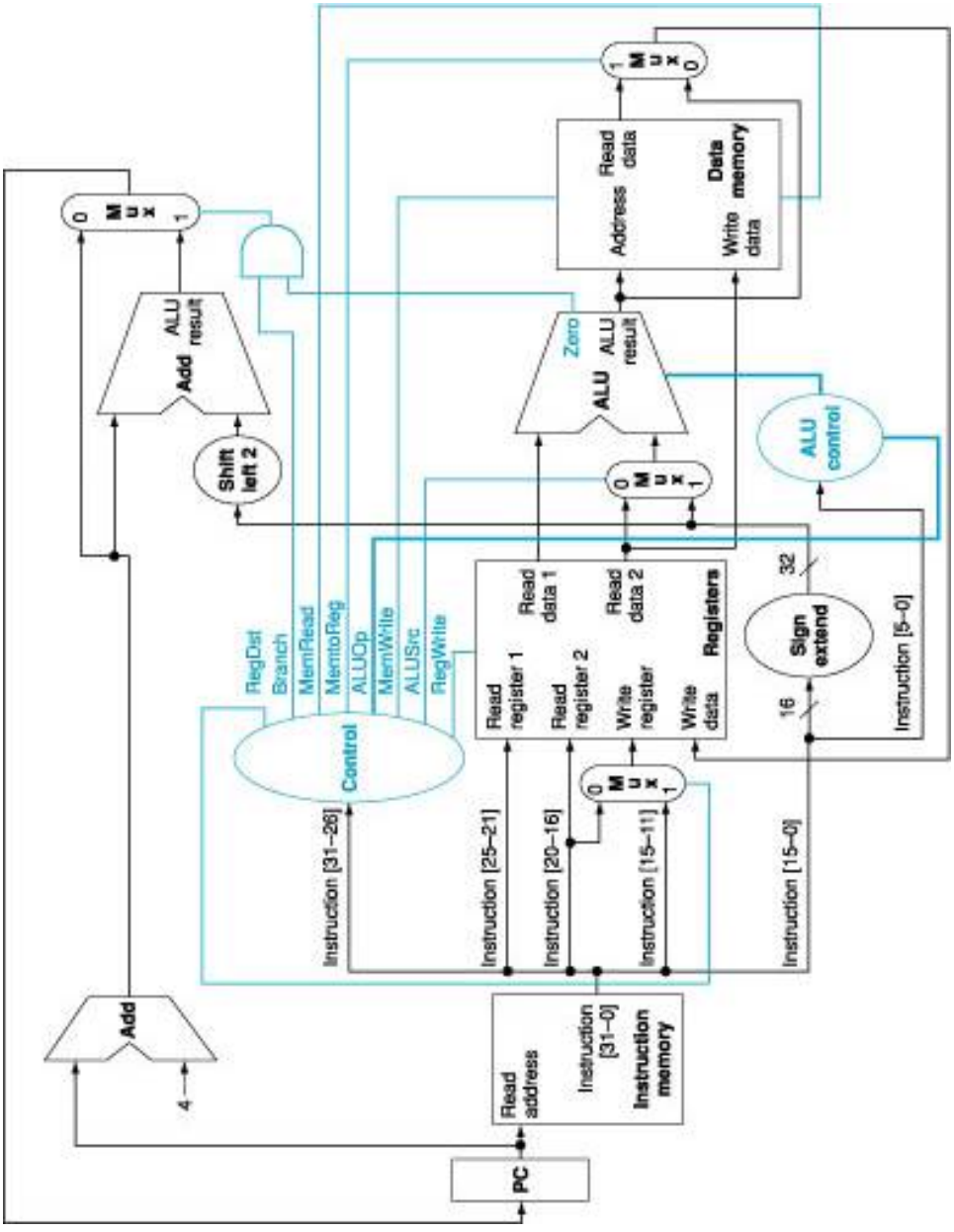
ALU Operation	ALUOp1	ALUOp0
A + B	0	0
A – B	0	1
A AND B	1	0
A OR B	1	1

Modify the structures, datapath and control on the next page to support the new **jregs rs, rt** instruction. You can add new logic blocks if needed. Indicate the values of the control signals in the table below when the new instruction is executing, add any new control signals to the table (you can add more rows as needed). **All existing instructions must continue to work properly.**

Control Signal	jregs
RegDst	
ALUsrc	
MemtoReg	
RegWr	
MemRd	
MemWr	
Branch	
Jump	
ALUOp1	
ALUOp0	

Your Name:_____

Question 3. Continued



Your Name:_____

Question 4. (25 pts) Processor Design

a.) [5 pts] How does a processor know if a set of bits is an instruction or data?

b.) [5 pts] MIPS requires that the stack pointer be stored only in register $29, true or false? Explain your answer.

c.) [15pts] Fill in the data path control signal values (0, 1, or X) for the instructions listed in the table. Use the datapath diagram on page 5 and ALUop values on page 4 for reference.

Control Signal	`jr $r31`	`slt $r2,$r3,$r5`	`lui $r5,xaddr`
RegDst			
ALUsrc			
MemtoReg			
RegWr			
MemRd			
MemWr			
Branch			
Jump			
ALUOp1			
ALUOp0			