CSSE 232 – Computer Architecture I
Rose-Hulman Institute of Technology
Computer Science and Software Engineering Department

Exam 2

Solution

Name:_____ Section: 1 2 3

This exam is **closed book**. You are allowed to use the reference card from the book. You may not use a computer, calculator, pda, etc. during the examination, except you may use a calculator **only** for question **??**.

Write all answers on these pages — use the back if necessary. Be sure to **show all work** and document your code. Do not use instructions that we have not covered (e.g. no `mult` or `div` but you can use `sll`, `srl`).

MIPS code is judged both by its correctness and its efficiency. You may use MIPS pseudoinstructions when writing MIPS code. However, the length of your instruction sequence is judged by the actual instructions to which the sequence expands.

All numbers are expressed in decimal unless specifically stated otherwise.

You are encouraged to read the entire exam before beginning.

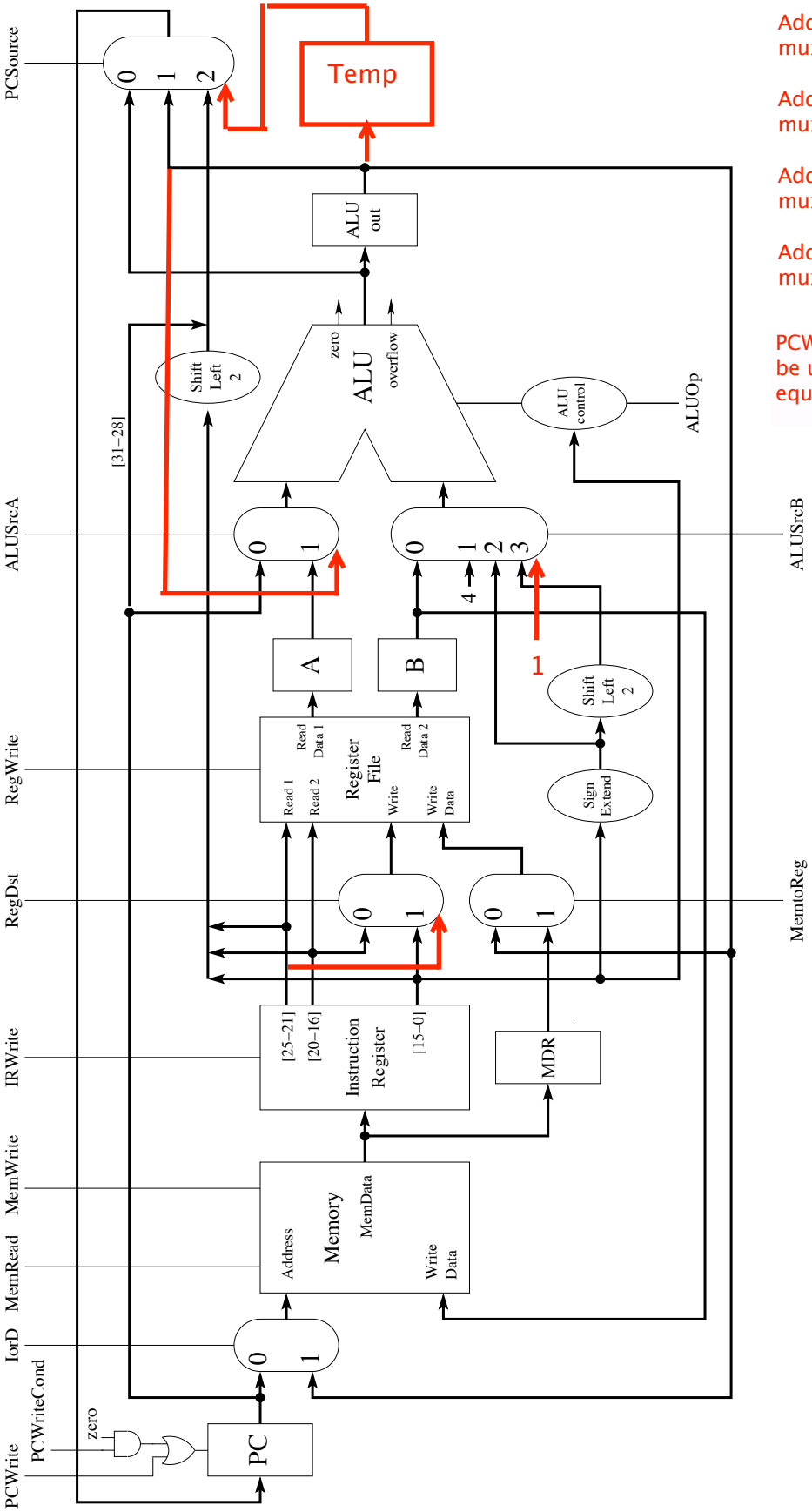
	Points available	Your marks
1	15	
2	15	
3	15	
4	20	
5	35	
Total	100	

Problem 1 (15 points) You are the lead designer for a **multi-cycle** implementation of a MIPS-like processor. Your team is considering adding a *decrement and branch if not equal* instruction to make loops faster. `dnbne $s0, $s1, label` decrements (subtracts 1 from) the value in register $s0 and stores it back into register $s0. If the result (the decremented value of register $s0) is not equal to the value in $s1, the instruction branches to `label`.

Modify the Register Transfer Language (RTL) shown in the following table to include the new `dnbne` instruction. Use as few cycles as possible. Remember that this is a `MIPS`-like processor. **Be sure to make it clear how your solution works**.

Step	R-type	`lw/sw`	`beq`	`dnbne`
Inst Fetch	IR = Mem[PC] PC = PC + 4			
Inst Decode Register Fetch	A = Reg[IR[25-21]] B = Reg[IR[20-16]] ALUOut = PC + (SE(IR[15-0])<<2)			
Execution Address comp `beq/bne` done	ALUOut = A op B	ALUOut = A + SE(IR[15-0])	if (A==B) then PC = ALUOut	Temp = ALUout ALUout = A - 1
Mem access R-type done `sw` done	Reg[IR[15-11]] = ALUOut	`lw:` MDR = Mem[ALUOut] `sw:` Mem[ALUOut] = B		Reg[IR[25-21]] = ALUout if(ALUout != B) PC = temp
`lw` done		`lw:` Reg[IR[20-16]] = MDR		

Problem 2 (15 points) Modify the multi-cycle datapath on the next page as necessary to support the new `dnbne` instruction. Be sure these modifications are consistent with the RTL in Problem **??** and add as little to the datapath as possible. List any additional control signals added and their purpose. Be sure it is clear how your changes work.



Add new input to PCSource mux for temp value

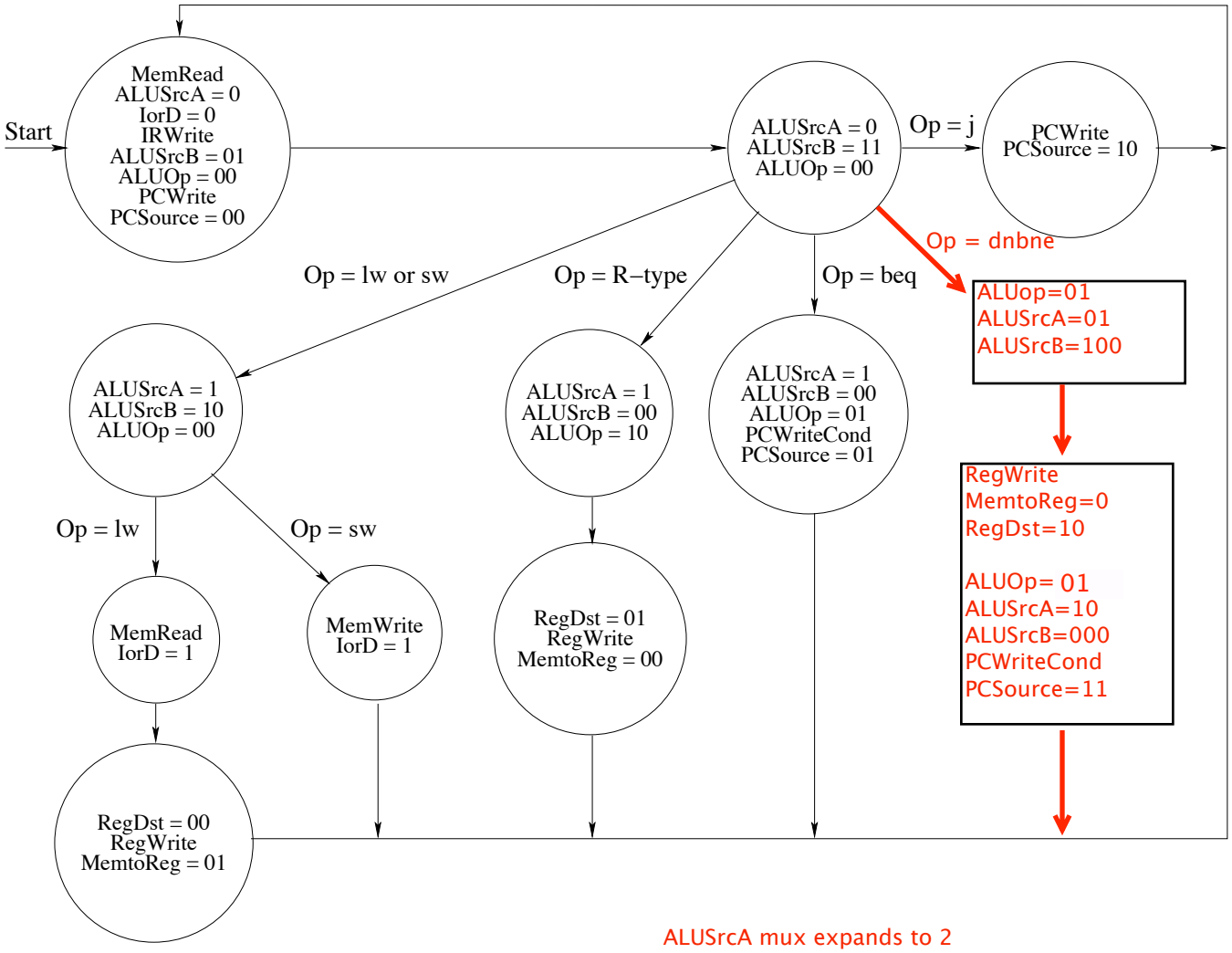
Add new input to ALUSrcA mux for ALUout

Add new input to ALUSrcB mux for 1 constant

Add new input to RegDst mux for IR 25–21 bits

PCWriteCond logic needs to be updated to support 'not equal' comparison

Problem 3 (15 points) Modify the Finite State Diagram below as necessary to support the new `dnbne` instruction. Be sure these modifications are consistent with the RTL in Problem ?? and the datapath in Problem ??. In addition, consider the impact upon any existing states. Note: `ALUOp = 00` generates an add, `ALUOp = 01` generates a subtract, and `ALUOp = 10` generates an operation based on the function bits.



ALUSrcA mux expands to 2 bits. Original signals are prefixed with 0

ALUSrcB mux expands to 3 bits. Original signals are prefixed with 0

RegDst mux expands to 2 bits. Original signals are prefixed with 0

PCSource mux expands to 4 bits.

Problem 4 (20 points) Your design team has tasked you with evaluating the impact of including the new `dnbne` instruction on the performance of your multi-cycle processor.

(a) (5 points) The current processor design has the following instruction categories and frequencies for the suite of programs that will be commonly run.

Type	Cycles	Frequency
R-type	4	0.41
`lw`	5	0.17
`sw`	4	0.14
`beq/bne`	3	0.20
jumps	3	0.08

What is the average CPI of the original machine?

(b) (5 points) Assume 1/2 of the branches are `bne` instructions, all of the `bne` instructions are used in loops which decrement a register, `dnbne` requires 4 cycles, and adding `dnbne` to the processor will slow the clock speed by 10%. **This may not be the correct answer to questions ??–??.**

Assuming `dnbne` is used to replace instructions where possible, fill in the table below with the frequencies for each type of instruction.

Type	Cycles	Frequency
R-type	4	
`lw`	5	
`sw`	4	
`beq/bne`	3	
`j`	3	
`dnbne`	4	

(c) (5 points) What is the average CPI of the new machine?

(d) (5 points) Which version (original or modified) is faster? What, if any, additional information is needed? Be sure to show **all** your work and justify your answer!