Assignment #7 – MIPS Processor Design

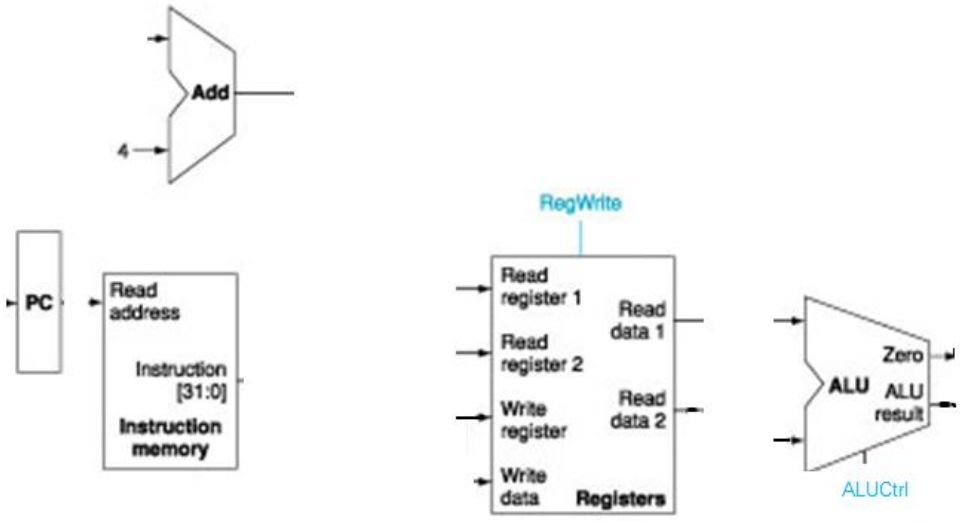
CDA 3100, Computer Organization I

Design a MIPS processor supporting **only** the `R-type` and the `jreq rd, rs, rt` instruction (encoded similarly to the R-types with the registers ordered in rs, rt, rd). The `jreq rd, rs, rt` instruction does the following:

- If `rs!=rt`, the address of the next instruction should be the content of `rt`, and the content of `rd` should be set to be the content of `rs`; otherwise, nothing happens, go to the next instruction.

For this problem, assume that the leading bit of the `opcode` of all `R-type` instructions is 0 and the leading bit of the `opcode` of `jreq` is 1.

(a) (50 points) Show the data path of this processor, add 2-1 MUX when necessary. Besides a group of wires, please show clearly the indices of the bits.



(b) (50 points) The control signals include RegWrite, ALUCtrl, and the signals to control the added 2-1 MUX. Show how to generate all control signals except ALUCtrl. Please fill the following truth table then write down the logic functions for the control signals. We will not need that many 2-1 MUXes.

Certain bits in the instruction can be referred to, for example, as “instruct[31]”. The zero output of the ALU can be simply referred to as “zero”.

	R-type	jreq (rs != rt)	jreq (rs == rt)
RegWrite			
MUXCtrl1			
MUXCtrl2			
MUXCtrl3			
MUXCtrl4			

RegWrite =

MUXCtrl1 =

MUXCtrl2 =

MUXCtrl3 =

MUXCtrl4 =