Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

Total points: 100

CDA 4101
Assignment # 2

---There are 10 questions in this assignment-------------------------------------

Submission Details: You can insert your answer directly in this Word Document. You can also insert the figure directly in this document where needed. Finally, Submit the PDF of your solution in Canvas by the deadline.

Honor Code: Honor Code This assignment should be completed individually to maximize learning. It is important that you adhere to the Course Policies, particularly the section on Programming Assignments. Also this assignment should follow FIU Honor Code Policy. Authorized help for all assignments are limited to the lab handouts, the LAs, and the Professor. Copying work from another student or the Internet is an honor code violation as is giving help to another student, which will result in a zero on the assignment and possibly further sanctions. Your code may be subject to evaluation by MOSS (Measure of Software Similarity), which is used to detect inappropriate similarities among programs.

Tools required: I encourage you to use LogiSim software. You may draw circuits into LogiSim and insert the circuit diagram in your report as a figure.

- You can download it from here: http://www.cburch.com/logisim/download.html
- Here is a video tutorial of Logisim (you can find many similar videos yourself online): https://www.youtube.com/playlist?list=PL9Tu_yD7oJURQqPEAQ78FggiDeiK7MqVb

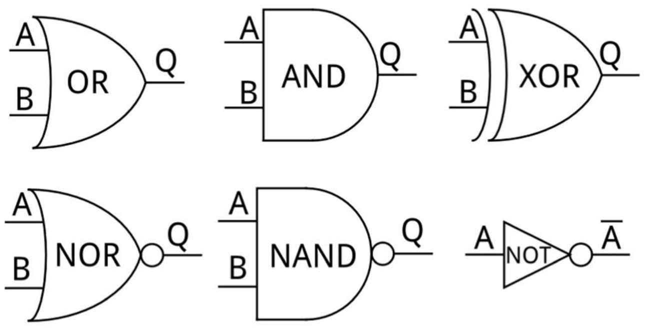
Goal: In this practice, you are going to use the basic building blocks we have learned in digital logic to put together a basic design of a MIPS microprocessor for a simple subset of the instruction set. These instructions illustrate most aspects of the processor:

Memory reference: lw, sw
Arithmetic/logical: add, sub, and, or, slt
Control transfer: beq, j

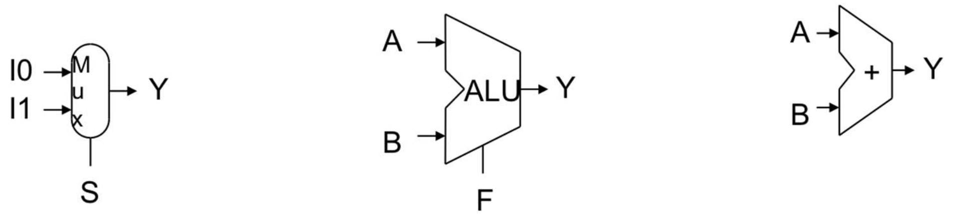
Combinational Elements: Remember that you can use gates (AND, OR, XOR, NOT), Adders, ALUs, and Multiplexors as we have studied so far and as are diagrammed on the attached sheet called combinational elements. When looking at a circuit diagram for a combinational element we think of electricity (imagine it flowing like water) flowing from the left side of the diagram to the right side. Each wire, represented by a single line, represents and input with electricity flowing in at a high level (1) or a low-level (0).

Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

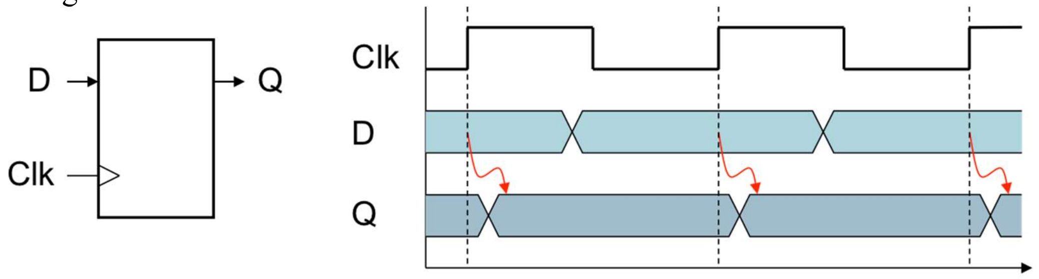
Combinational Elements



- Multiplexer
  - Y = S ? I1 : I0
- Arithmetic/Logic Unit
  - Y = F(A, B)
- Adder
  - Y = A + B



Sequential Elements: We will also need to use something we discussed briefly earlier called a sequential element. A sequential element is one that introduces a formal sequencing into a circuit and only changes on the clock edge. The typical sequential element that we are interested in is a register. A register has an input D, and an output Q. There is always a signal on D and the goal is to have that signal reflected on Q constantly; allowing Q to change only when we are looking for that change.

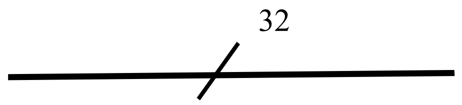


Consider the above diagram on sequential elements. Notice in the right figure how the value reflected on the Q output only changes on the rising edge of the clock (i.e. when the clock value goes from 0 to 1).
Note that the D register shown this diagram is a one-bit register. When we are dealing with the processor design we will be using registers that are the size of a word in the machine we are

Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

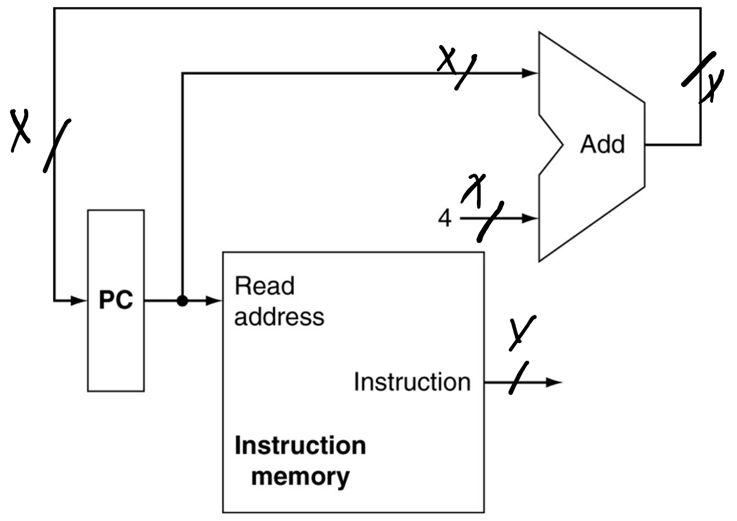
designing a data path for. MIPS has 32-bit words so most of our registers and hence our "buses" or lines will be 32-bits. We can just draw a long box with a line coming in and a line coming out to represent a register of more than one bit.

We show a line is more than one bit or one value wide by drawing a diagonal line through it and writing the "width" of the bus or line above the diagonal like this:



This diagram represents 32-bits or a 32-bit wide bus (32 lines) coming into a device.

Now we have the tools to be able to draw a picture representing the register called the Program Counter addressing instruction memory and sending out an instruction to the processor.



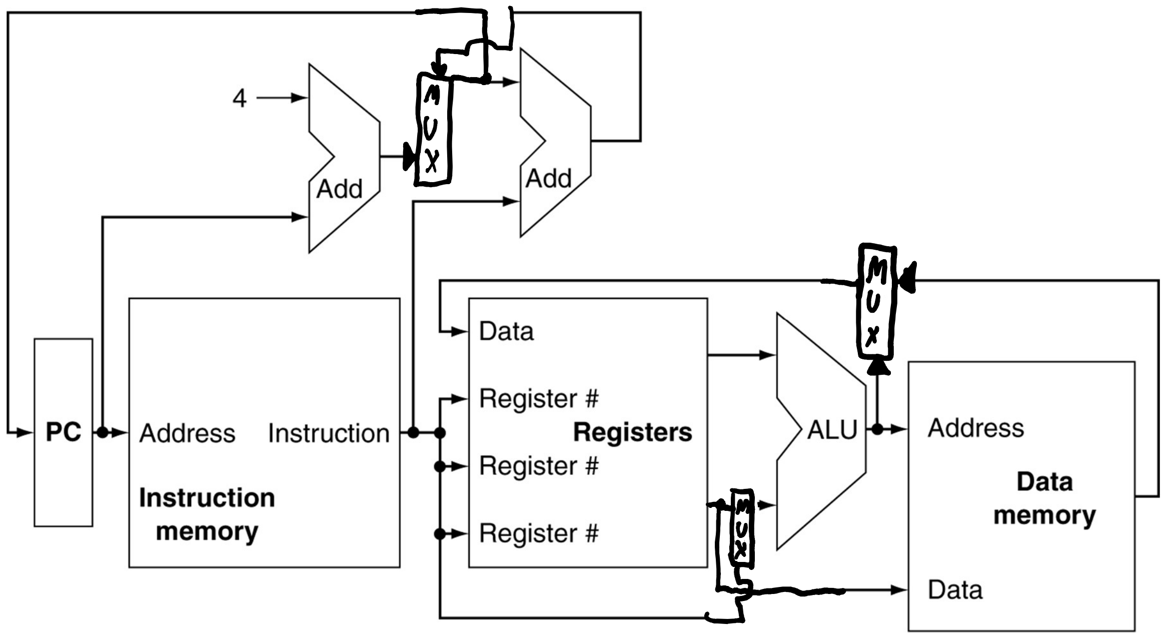
X = size of instruction memory
Y = length of each instruction

P.1) [5 pt] In the above diagram, add diagonal lines to indicate the size of each of the lines coming into and going out of each logic element.

To continue in this exercise we will need to have one more concept down. This is the idea that we may want more than one "input" in general to a particular logic component. For example in the above diagram we may want to sometimes get a new program counter that is just the next instruction (hence the add 4 to the current program counter) or if the instruction is a branch or jump we will want to get some information from the instruction to change to the new program counter. When we want two different inputs, we need to use a multiplexor to change between the two rather than have two outputs (old PC and instruction) both going into the same input (new PC).

The diagram below shows an "overview" of the whole processor design we will be modifying.

Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2



P.2) [10 pt] Redo the diagram inserting multiplexors where needed.

Now that you have a new diagram with multiplexors, you will be focusing on elaborating on each section of the overview diagram to account for different parts of the execution of the machine for different instructions. To make sure that you understand the overview diagram, answer the following questions with respect to it.

P.3) [10 pt] What bits of the instruction will be sent to the second adder in the circuit computing the new PC – why?

The 16 lower order bits, as they specify the immediate offset value. These 16 bits are then sign extended to 32 and then sent to the second adder.

P.4) [10 pt] What bits of the instruction will be going into each of the Register # inputs of the Register file in the center of the machine?

Bits 6-10 go to one register and bits 11-15 go to another one.

P.5) [10 pt] For what kind of instruction will the output of the data memory be going into the input of the register file? Explain.

The load instructions, since they write the value read from the output of the data memory into the register file.

Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

P.6) [5 pt] Now add to your overview picture, an oval for "control" showing lines coming in and lines going out to control the multiplexors. Label each line going into a multiplexor with a descriptive word to describe what control we want selecting the multiplexor input.

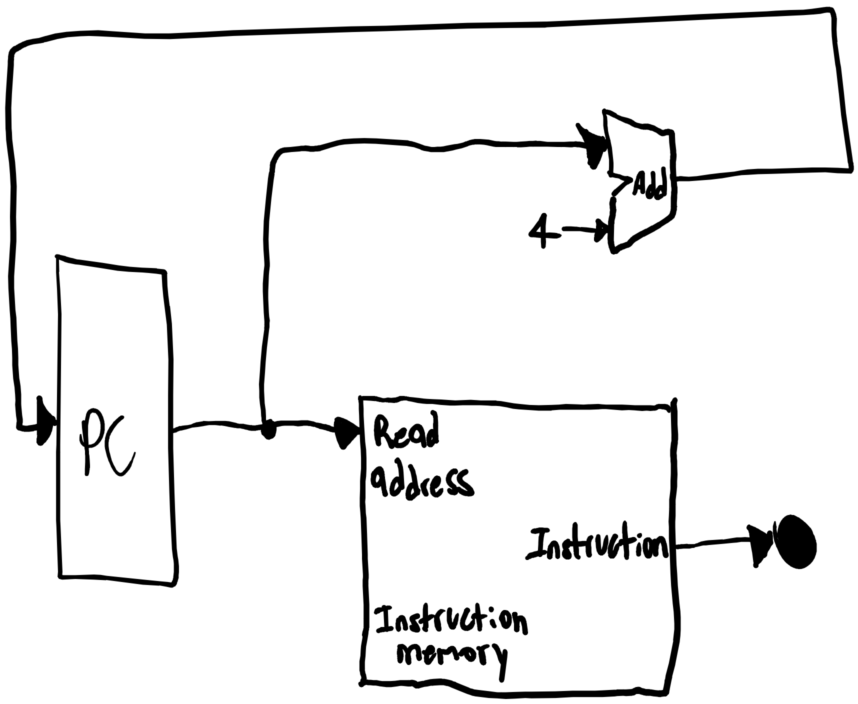
?

Now for each set of instructions you will be making a list of what the instruction type needs to do and draw basic block diagram of only the components relevant to these particular instructions.

Instruction Fetch/Branch Instructions: Refine the PC generation part of the block diagram and explain what needs to happen for a beq instruction and a jump instruction.

P.7) [10] Now, for "Instruction Fetch" list necessary actions and draw a revised circuit diagram for the components required for instruction fetch.

For Instruction Fetch, we need to fetch the instructions from the memory. To then execute the next instruction, we also need to increment the program counter so that it can then point to the next instruction.

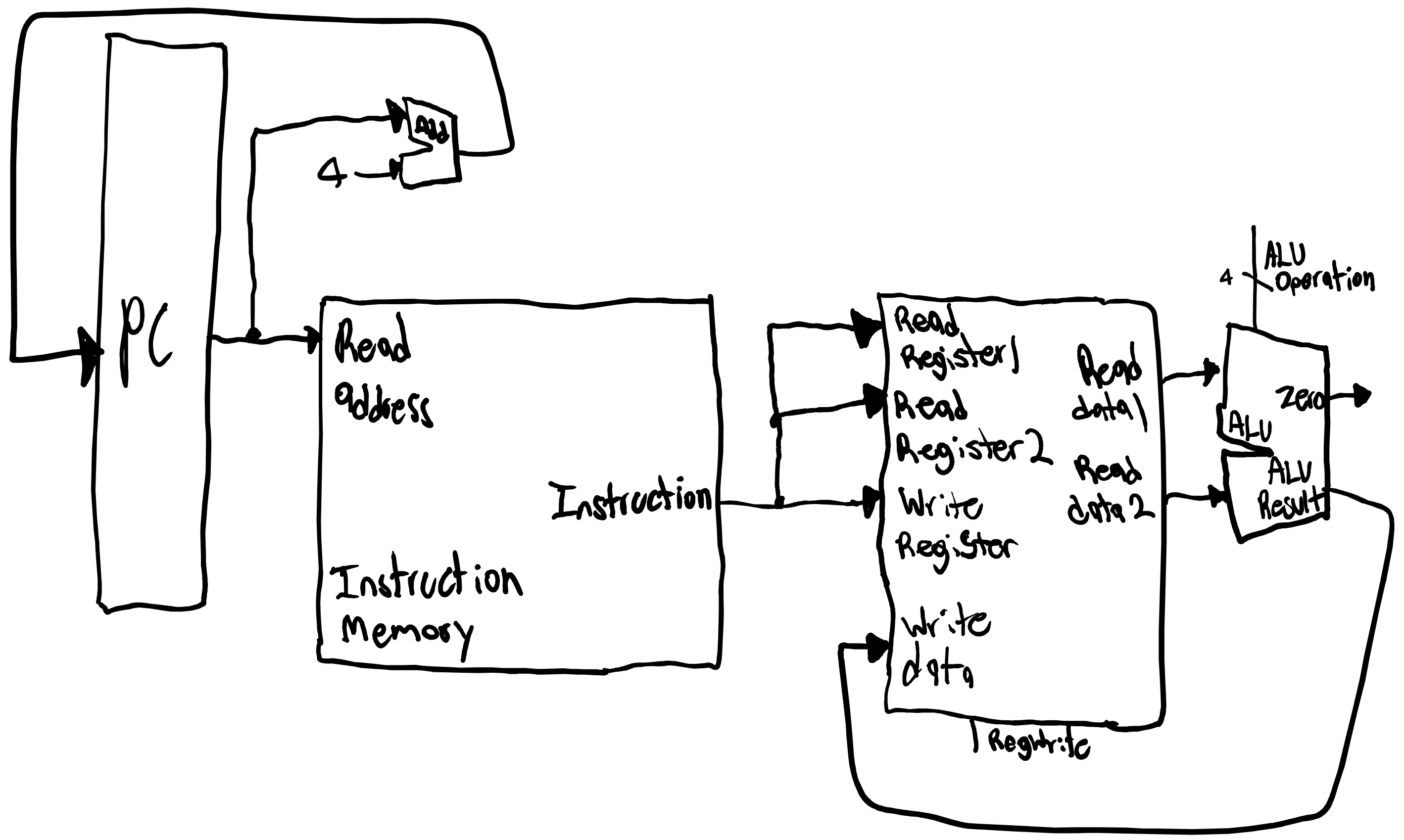


Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

R-Format Instructions: The R-format instructions are the add, sub, and, or, slt.

P.8) [15 pt] Now for "R-Format instructions", list necessary actions and draw a revised circuit diagram for the components required in the execution of an R-format instruction.

For R-Format instructions, we need to read 2 registers, perform an ALU operation on the contents of the registers, and then write the result to another register.

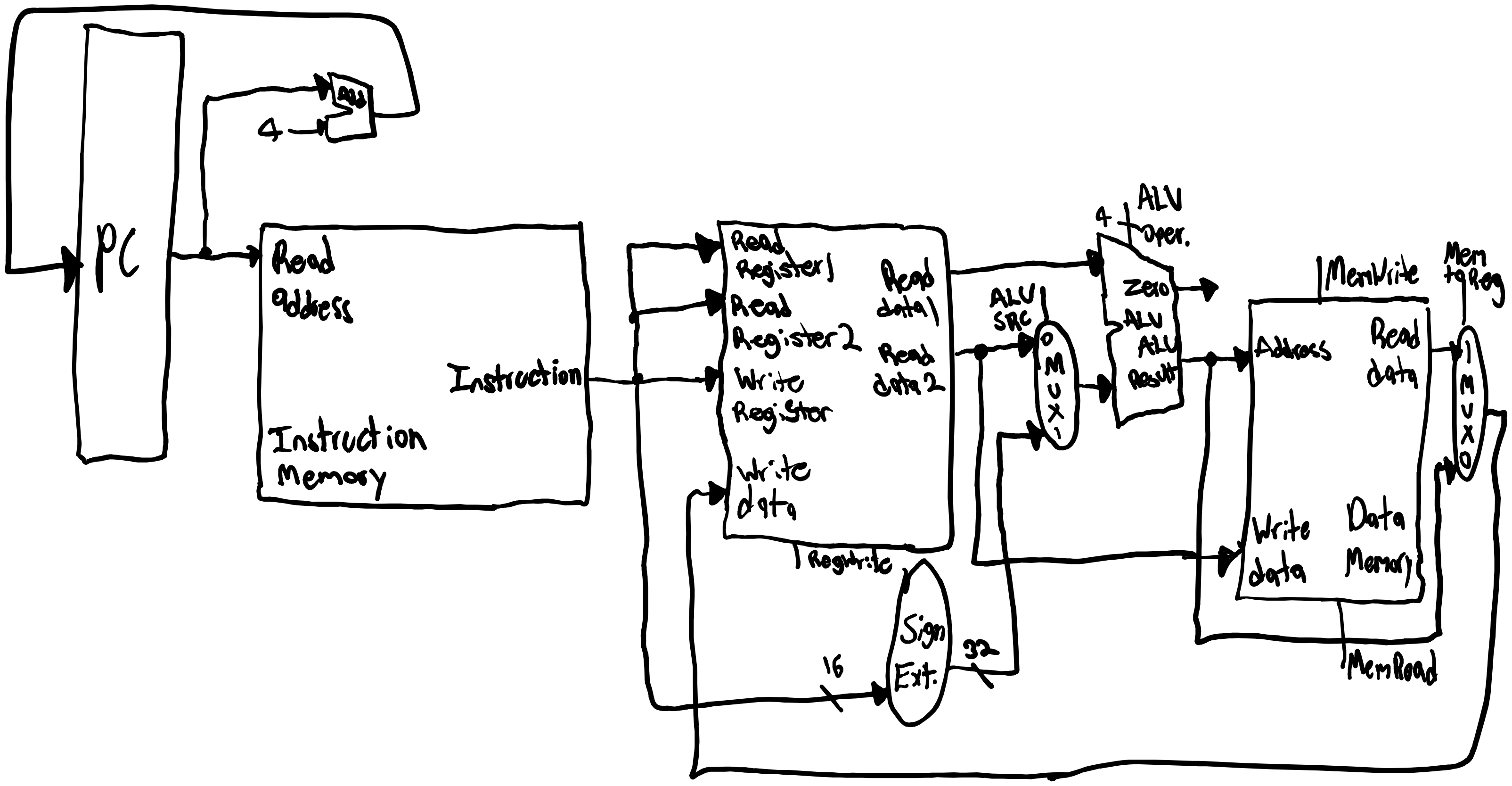


Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

Load/Store Instructions: The load/store instructions are lw and sw.

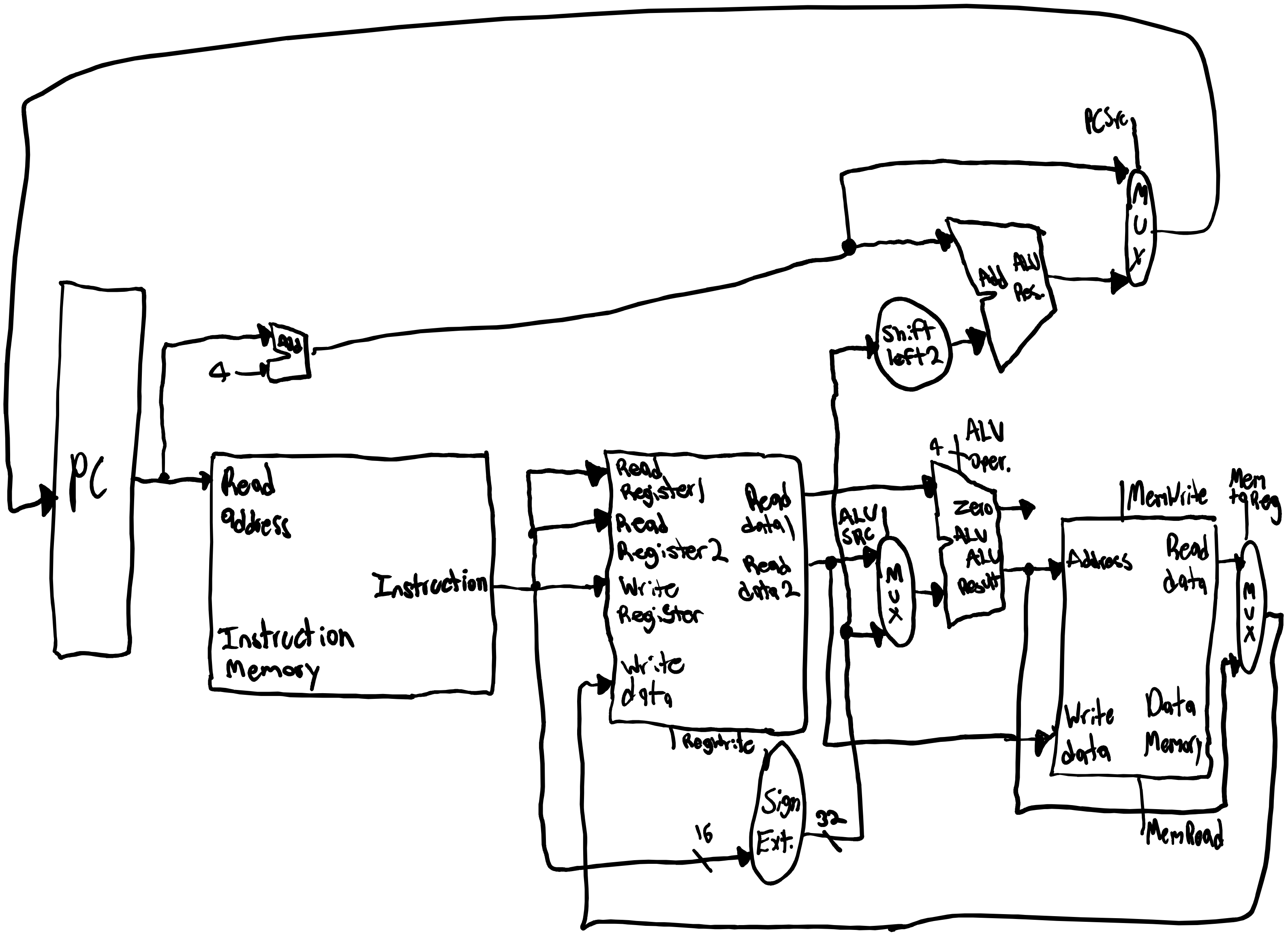
P.9) [15 pt] Now for "Load/Store" list necessary actions and draw a revised circuit diagram for the components required in the execution of a load/store instruction.

For Load/Store, we need the memory unit, a read signal, the sign extension unit, and the write signal to be able to store and write to memory.



Joshuan Jimenez (ID: 6059847)
Cristian Caro
CDA4101 - RVBB
Assignment #2

P.10) [20 pt] Now incorporate your smaller circuit diagrams into a larger overview diagram with all of the pieces together. Note that you can just label the control lines coming out of an oval appropriately labeled to indicate the signal you want generated by the control.



---------------------------------------Questions END here--------------------------------