In this lab you will complete the MIPS CPU design. You will write structural Verilog code to implement the CPU as shown below.

You will use the datapath which you implemented in lab 4. The memory will be given to you and you will design the controller.

The controller design should have the following structure.

The block labeled **Controller** will generate the control sequences for all of the control signals besides the ALU control signals. The block labeled **ALU Control** is responsible for generating the control signals for the ALU based on the ALUOp values. The following figure summarizes the steps needed to perform each instruction type.
Remember to support the addi instruction as well.

The zero, opcode, and funct signals come from the instruction register (IR) in the datapath, so you will need to modify your datapath to output those signals. The ALUOp signal is coded as follows: 00 is ADD, 01 is SUBTRACT, and 10 is “the funct field determines the operation”.

To test your final design you should modify the memory module to initialize it with the following simple program:

```
addi $r1, $r0, 1
addi $r2, $r0, 2
add $r3, $r1, $r2
sw $r3, 32($r0)
```

You should execute this program and you should ensure that the correct result is in $r3.

**What to submit:**

Submit the verilog code of your design and your testbench electronically via EEE.

Submit simulation evidence that register $r3 has the correct data in it after the execution of the program shown above.