Prepare your answers to the following questions in a plain text file. Submit your file to the Curator system by the posted deadline for this assignment. No late submissions will be accepted.

You will submit your answers to the Curator System (www.cs.vt.edu/curator) under the heading HW04.

Questions 1 through 7 refer to the completed single-cycle datapath design, reproduced below, which supports execution of the following MIPS instructions: add, sub, and, or, slt, lw, sw, beq and j.

1. [10 points] For which of the supported instruction(s) must the control line RegDst be set to 0? No justification is necessary.

2. [10 points] For which of the supported instruction(s) must the control line ALUSrc be set to 1? No justification is necessary.

3. [10 points] For which of the supported instruction(s) must the control line MemWrite be set to 1? No justification is necessary.

4. [10 points] For which of the supported instruction(s) might the control line Zero be set to 1? No justification is necessary.

5. [10 points] For which of the supported instruction(s) is the Shift left 2 unit labeled 5 necessary? No justification is necessary.
6. [15 points] For which of the supported instruction(s) does it not matter how the control signal labeled 6 is set? Justify your answer precisely.

7. [15 points] Suppose that an implementation of the datapath is created in which a defect causes the ALUSrc signal to always be 0. Which of the supported instructions would be affected, and how?

8. [20 points] The designers of the MIPS32 instruction set made the decision to include and exclude the following possible bitwise logical instructions from the basic instruction set:

- **# include these**
  - `or $rd, $rs, $rt` # GPR[ rd[i] ] = GPR[ rs[i] ] OR GPR[ rt[i] ]

- **# exclude these**
  - `not $rd, $rs` # GPR[ rd[i] ] = NOT GPR[ rs[i] ]

Recall your Boolean algebra from Discrete Mathematics. Also, recall that NAND means NOT AND, and that XNOR means NOT XOR.

Suppose the designers had made the decision to include `and`, `or`, and `xor`, but were wavering between including `not` and including `nor`. Only one of those two instructions could be included, but which one?

Now, most high-level programming languages will support one or more of the operations that will be excluded, so a compiler must be able to translate all of the excluded instructions into an equivalent form using only the basic instructions.

Make a complete, convincing argument that supports the conclusion to include `nor` and exclude `not`.

**Note:** An argument should involve both mathematical analysis and consideration of the issues mentioned above. Your score on this question will depend both on the quality of your analysis and the quality of your presentation.