# ALU Decoder

## Problem Statement

The following truth table describes an Arithmetic/Logic Unit (ALU) Decoder used in a microprocessor controller. The circuit has seven inputs and three outputs. The inputs are a 2-bit ALUOp signal, a 3-bit funct3 signal, 1 bit called op_5, and one bit called funct7_5. The last two inputs are treated as a pair in the truth table. The output is a 3-bit ALUControl signal. "Instruction" doesn't imply any hardware; it's just a human readable note about the type of instruction that will make more sense when we build the processor later. The outputs are only specified for certain input combinations; they may be treated as Don't Care for unspecified inputs combinations.

| ALUOp | funct3 | {$op_5$, $funct7_5$} | ALUControl | Instruction |
|---|---|---|---|---|
| 00 | x | x | 000 (add) | `lw, sw` |
| 01 | x | x | 001 (subtract) | `beq` |
| 10 | 000 | 00, 01, 10 | 000 (add) | `add` |
| | 000 | 11 | 001 (subtract) | `sub` |
| | 010 | x | 101 (set less than) | `slt` |
| | 110 | x | 011 (or) | `or` |
| | 111 | x | 010 (and) | `and` |

The goal of this lab problem is for you to design a gate-level implementation of the ALU Decoder, model it with structural SystemVerilog, and debug your design. Use only AND, OR, and NOT gates in your solution.

Start with the files below. Modify the aludec module but don't change the testbench or test vectors because that would mess up the hash.

If you are using a lab computer with a professional version of ModelSim, be sure to uncheck Enable Optimization in the Start Simulation dialog. Otherwise, ModelSim may optimize away signals so you won't be able to see them in the waveform viewer.