Arithmetic and Logic Unit (ALU)

Begin by reading your lab manual. Then answer the following questions.

1 Complete the ALU

Create a Verilog module called eightbit_alu which has two 8-bit inputs, a and b, and one 3-bit input, sel. The outputs of this module are an 8-bit signal f, a 1-bit signal ovf and a 1-bit take_branch signal. The value of these outputs should change based on the sel signal value which determines the operation. a, b and f are 2's complement numbers.

The ALU operations and their descriptions are shown in Table 1.

s[2:0]	f[7:0]	ovf	take_branch	Description
0 0 0	$a + b$ (add)	overflow	0	a plus b
0 0 1	b (inv)	0	0	Bitwise inversion of b
0 1 0	$a \cdot b$ (and)	0	0	Bitwise AND of a and b
0 1 1	$a \mid b$ (or)	0	0	Bitwise OR of a and b
1 0 0	$a >>> 1$ (sra)	0	0	Arithmetic shift right
1 0 1	$a << 1$ (sll)	0	0	Logical shift left
1 1 0	0	0	$a == b$ (beq)	Branch if Equal
1 1 1	0	0	$a! = b$ (bne)	Branch not Equal

Table 1: ALU operations

Please note that for arithmetic shift right, the sign of the shifted number should be the same as the original number.

Please run your code using Vivado synthesis to make sure there are no syntax errors in the code.

2 Create Test Vectors

Similar to what you did in prelab 2, create a set of test vectors that could test all of the bits in inputs, i.e. a, b and s, and outputs, i.e. f, ovf and take_branch, of the ALU for every operation. Create a table to show your test vectors. Write a testbench.

Submit all your code (design and testbench) as well as a screenshot of your simulation.