0.1 Experiment 5 (Arithmetic Logic Unit)

0.1.1 Aim

In this experiment, your knowledge to implement an Arithmetic Logic Unit (ALU) which is explained in section 4.9 of the course book will be tested.

0.1.2 Problems

You will implement verilog code for a 5-bit arithmetic logic unit (ALU). In ALU, there will be 2-bit select input (S), 5-bit data inputs (X, Y), 5-bit output (F), 1-bit output (C_{out}), 1-bit output ($Overflow$), and the functional table will be as follows:

S	**Overflow**	**C$_{\mathbf{out}}$**	**F**	**Representation**
00	0	X[3:1] × Y[2:0]		Unsigned
01	0	X > Y	00000	Unsigned
10	X + Y			2's Complement
11	X - (4×Y[2:0])			2's Complement

Note:

- Let's say Y = 10110; if the representation is unsigned, Y[2:0] will be 6 in decimal; if the representation is 2's complement signed, Y[2:0] will be -2 in decimal.
- Overflow output is only valid for the operations performed in 2's Complement representation (namely third and fourth operation shown in the above table). For other cases, it is always 0.
- Consider overflow only for addition and subtraction.
- Use **X** for *X*, **Y** for *Y*, **S** for *S*, **F** for *F*, **Overflow** for *Overflow* and **Cout** for C_{out} as input or output names in your verilog files.

0.1.3 Preliminary Work

Before the experiment, you should apply and report 5 step controller process explained in the class as follows:

Before the experiment, you should prepare the following materials:

1. Design circuits with minimum number of components for each function of the ALU.
2. Merge all operations with select inputs and organize outputs.
3. Try to minimize your implementation by using repetitions (see the hint below).
4. Draw final circuit as the final design of the ALU with corresponding. functional tables and calculate the number of chips in your design.
5. Write the behavioral level verilog code of the ALU.

6. Write the verilog code for the testbench waveform in order to test all possible input combinations.
7. Verify the functionality of your implementation.

Then, submit the your code, and your report under the name `<StudentID1>_<StudentID2>_PRE5.zip` through Moodle. One submission per group is enough.