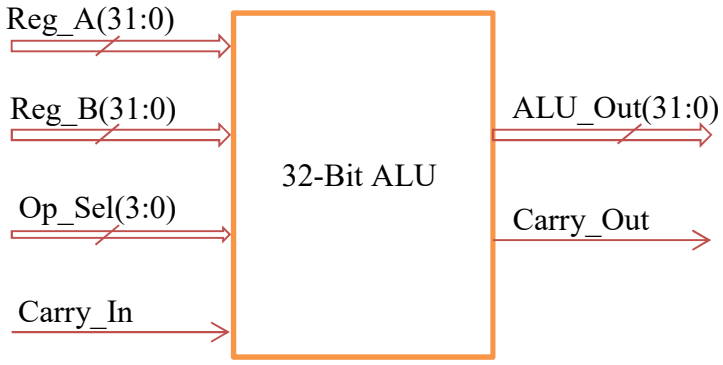
Lab #10 (Final Project): 32-bit ALU Design

1. Requirement

Design a 32-bit ALU which can perform arithmetic and logic operations:

- The design has two 32-bit inputs, input A and input B: they are unsigned binary number
- The design must be able to perform:
  - Addition, increment, decrement and transfer (arithmetic operations)
  - AND, OR, NOT, XOR (logical operations)
  - Right shift, left shift (shift operations)
- The design must have 4-bit select line called Operation Select, which would direct the unit as to which operation to perform
- The unit has a Carry-in and also a Carry-out.

The block diagram of this ALU is as shown below:



The interface of this design is as below:

```
entity ALU_32Bits_Design is
      port(
            Reg_A     :  in std_logic_vector(31 downto 0);
            Reg_B     :  in std_logic_vector(31 downto 0);
            Op_Sel    :  in std_logic_vector(3 downto 0);
            Carry_In  :  in std_logic;
            Carry_Out :  out std_logic;
            ALU_Out   :  out std_logic_vector(31 downto 0)
      );
end ALU_32Bits_Design;
```

The Function Table of this ALU is as shown in **Table 1.** The ALU performs different operations according to the select lines (operation selection), i.e. the Op_Sel signal dictates the operation performed by the Unit.

Table 1: Function Table

Operation Select (Op_Sel)				Carry_In	Operation	Function
Op_Sel(3)	Op_Sel(2)	Op_Sel(1)	Op_Sel(0)			
0	0	0	0	x	ALU_Out = A	Transfer Reg_A
0	0	0	1	x	ALU_Out = A + 1	Increment Reg_A
0	0	1	0	x	ALU_Out = A - 1	Decrement Reg_A
0	0	1	1	0 or 1	ALU_Out = A + B + Carry_In	Addition
0	1	0	0	x	ALU_Out = NOT A	Not Reg_A
0	1	0	1	x	ALU_Out = A AND B	Reg_A and Reg_B
0	1	1	0	x	ALU_Out = A OR B	Reg_A or Reg_B
0	1	1	1	x	ALU_Out = A XOR B	Reg_A xor Reg_B
1	0	x	x	x	ALU_Out = Shift A to the right by amount of bits defined by B	Right shift for Reg_A
1	1	x	x	x	ALU_Out = Shift A to the left by amount of bits defined by B	Left shift for Reg_A

2. **Pre-lab**
   - Study the concept of an ALU
3. **Lab**
   - Write the VHDL code for this unit
   - Simulate using Xilinx ISE simulator
4. **Deliverables**
   - VHDL program
   - VHDL test bench with enough test cases which cover all the operations of this design
   - Waveform with comments
   - A Formal Lab Report
5. **Guidelines**
   - The students are expected to record their Prelab/Design work in the Lab Journal
   - In addition to turning in the Lab notebook, you are required to submit a Formal Lab Report
   - The reports are individual, even though you work in teams.
6. **Guidelines**
   - Working ALU Implementation: 50 points
   - Good test-bench: 20 points
   - Well-structured code including meaningful comments: 15 points
   - Good lab report: 15 points