Assignment 3

16-bit ALU

Introduction: -

ALU is the fundamental building block of the processor, which is responsible for carrying out the **unsigned arithmetic**, **logic** functions, **Shift** functions and **Comparison** functions.

Specification:

- ALU **Operands** (A, B)
- ALU **Result** (ALU_OUT)
- ALU operands and output Result are of **16-bit** width.
- ALU **Result** (ALU_OUT) **is registered**.
- The ALU function is carried out according to the value of the **ALU_FUN** input signal stated in the table in the following page and any other value for **ALU_FUN** not stated in the table, **ALU_OUT** must equal to **16’b0**
- **Carry_Flag** represents the carry bit in case of addition and the borrow in case of subtraction.
- **Arith_flag** is activated "High" only when ALU performs one of the arithmetic operations (Addition, Subtraction, Multiplication, division), otherwise "LOW"
- **Logic_flag** is activated "High" only when ALU performs one of the Boolean operations (AND, OR, NAND, NOR, XOR, XNOR), otherwise "LOW"
- **CMP_flag** is activated "High" only when ALU performs one of the Comparison operations (Equal, Greater than, less than), otherwise "LOW"
- **Shift_flag** is activated "High" only when ALU performs one of the shifting operations (shift right, shift left), otherwise "LOW"

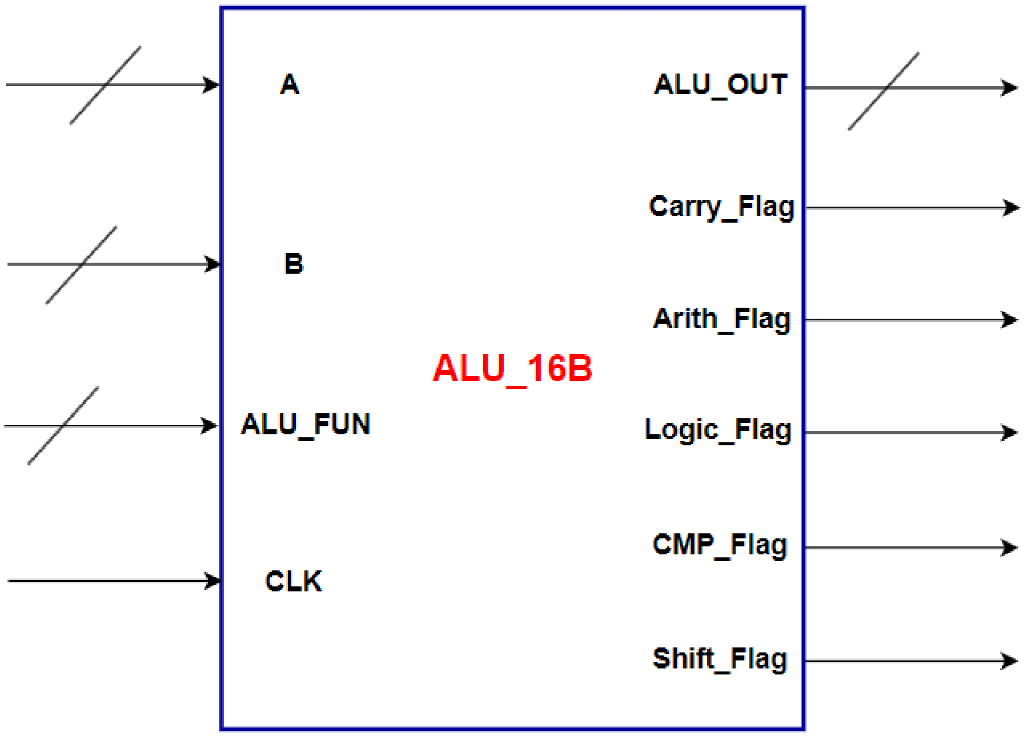
ALU_FUN Table

ALU_FUN	Operation	ALU_OUT
0000	Arithmatic : **unsigned Addition**	
0001	Arithmatic : **unsigned Subtraction**	
0010	Arithmatic : **unsigned Multiplication**	
0011	Arithmatic : **unsigned Division**	
0100	Logic : **AND**	
0101	Logic : **OR**	
0110	Logic : **NAND**	
0111	Logic : **NOR**	
1000	Logic : **XOR**	
1001	Logic : **XNOR**	
1010	CMP: **A = B**	**Equal to 1 else Equal to 0**
1011	CMP: **A > B**	**Equal to 2 else Equal to 0**
1100	CMP: **A < B**	**Equal to 3 else Equal to 0**
1101	SHIFT: **A >> 1**	
1110	SHIFT: **A << 1**	

Hint: Use Case statement to describe the behavior of this table and use default case if needed.

Hint: You can use if statement inside case branches

Block Interface



Requirements

1. Write a Verilog Code to capture the above specifications

2. Write a testbench to test all the ALU functions to include at least the following 16 test cases: -
   - Unsigned Arithmetic Operations (Addition, Subtraction, Multiplication, Division)
   - Logical Operations (AND, NAND, OR, NOR)
   - Compare Operations (Equal, Greater, Less)
   - Shift Operations (A shift right, A shift left, B shift right, B shift left)
   - NOP

3. Write a testbench to test all the ALU functions with operating clock frequency 100 KHz

4. Synthesis Diagram of your code using draw.io online tool