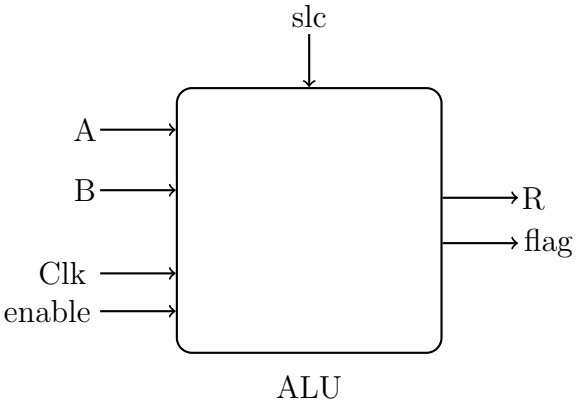
ALU

Your task is to program the behavior of an entity called "ALU". This entity is declared in the attached file "ALU.vhdl" and has the following properties:

- Inputs: Clk and enable with type std_logic; these are clock and enable signals
- Inputs: A and B with type std_logic_vector; these are operands
- Inputs: slc with type std_logic_vector; this is a selector for choosing one of the operators
- Outputs: R with type std_logic_vector; this is the result of operation with the same length as inputs
- Outputs: flag with type std_logic; this is a one-bit flag



Do not change the file "ALU.vhdl"!

The "ALU" entity shall behave according to the following:
It has two input data which we assume to be unsigned, a clock signal, an enable signal and a selector to select which instruction has to be performed. The output only changes at the rising edges of the clock. The length of data is 4 bits. In a case that the result of the operation is 5 bits only the four least significant bits are used for the output on the port R. The instruction set includes SUB, AND, Comparator and Shift Left and the value of selector for each instruction is as follows:

- "00": SUB
- "01": AND
- "10": Comparator

- "11": Shift Left

The instruction description is as below:

- SUB: subtracts B from A and outputs the result on R.
- AND: operates logical AND between A and B and outputs the result on R.
- Comparator: compares A with B, and outputs A on R.
- Shift Left: shifts all bits of A one place to the left (the most significant bit is discarded), sets the least sigificant bit to 0, and outputs the four least significant bits on R.

When the ALU is disabled, the outputs (R and flag) are zero and when it is enabled it performs the selected instruction (based on slc) and then outputs the result. The length of result is equal to the length of input. There is also another output which is a one-bit flag and should be calculated for each selected operation. The flag for each instruction is determined based on the result of each operation and is defined as follows:

- SUB –> Zero: the flag is 1, when the result is zero.
- AND –> Zero: the flag is 1, when the result is zero.
- Comparator –> Carry: if A>=B then the flag bit will be 1, if A<B then the flag bit will be 0.
- Shift Left –> Flag: the flag is the most significant bit of A.

This behavior has to be programmed in the attached file "ALU_beh.vhdl".

To turn in your solution, write an email to vhdl-dis+e384@tuwien.ac.at with Subject "Result Task 5" and attach your file "ALU_beh.vhdl".

Good Luck and May the Force be with you.