4-bits ALU (Simulation only, 2 members in each team)

An Arithmetic Logic Unit (ALU) is a combinational digital circuit that performs arithmetic and bitwise operations on integer binary numbers, serving as a fundamental component of any processor. As illustrated in Figure 1, the ALU takes two 4-bit binary operands as inputs, along with an opcode that specifies the operation to be executed. After calculating the result, the ALU outputs it, along with any carry or overflow information available at the status output.

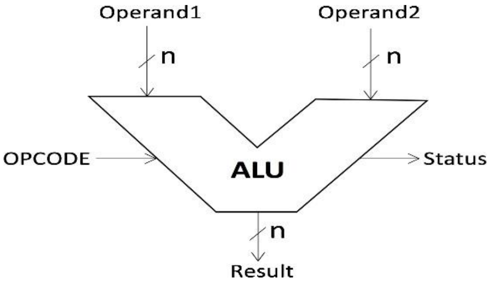

Figure 1: ALU Symbol

Inputs and Outputs:

1. **Operands**: The ALU receives two 4-bit binary numbers, referred to as Operand1 and Operand.
2. **Opcode**: A 3-bit opcode determines the specific operation to be performed on the operands.
3. **Result**: The Result is an 8-bit number, which is the result of the operation being performed.

The ALU uses a set of opcodes to determine the operation to be performed, as outlined in table 1. The ALU takes two 4-bit operands, referred to as

opcode	**operation**	**result**
000	Addition	Result = Operand1 + Operand2
001	Subtraction	Result = Operand1 - Operand2
010	Multiplication	Result = Operand1 * Operand2
011	Greater Than	Result = 1 if Operand1 > Operand2
100	Less Than	Result = 1 if Operand1 < Operand2
101 & 110 & 111	No Operation	Result = 0

Table 1: OpCode table

Operand1 and Operand2, along with a 3-bit opcode that specifies the operation. The result is calculated based on the opcode and then the status is computed. (Note that the output must be 8-bit for the multiplication).

Notes & Advices:

1) **VERY IMPORTANT**: Divide the project into multiple blocks, for example a block for a 4-bit adder, a block for a 4-bit subtractor, a block for an 4-bit multiplier, etc…

2) Do some research about how to implement a subtractor using an adder.
3) Do some research about how to implement the multiplier which is the hardest part of the project.
4) Divide the ”Greater Than” and ”Less Than” blocks into sub-blocks for the design to be easier.
5) Using multiplexers should be only to select the result based on the opcode.