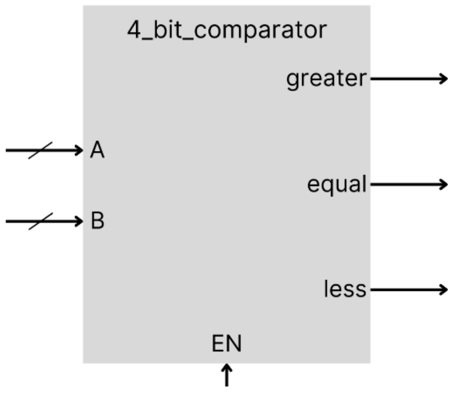
Session 2 assignment

Q1) Design using Verilog HDL a 4-bit comparator circuit, with the following specifications:

- Use Always block.

Ports	Type	Size	Description
A	input	4-bit	--
B	input	4-bit	--
EN	input	1 bit	The comparator works when EN is high.
equal	output	1 bit	When A = B.
greater	output	1 bit	When A > B.
Less	output	1 bit	When A < B.



Q2) Design the following ALU (Arithmetic Logic Unit) using Verilog HDL, with the following specifications:

Ports	Type - Size	Description
A	Input – 4-bit	The first input port.
B	Input – 4-bit	The second input port.
cin	Input – 1 bit	Carry in, used only for addition operation.
opcode	Input – 2-bit	Operation code.
pass_A	Input – 1 bit	When it's high, pass A to out port and ignore the opcode.
pass_B	Input – 1 bit	When it's high, pass B to out port and ignore the opcode.
out	Output – 4-bit	The main output.
cout	Output – 1 bit	The carry out, it's equal zero for all operation instead of arithmetic operations

Note: pass_A is the highest priority, the second is pass_B, and the least is opcode operation

opcode signal	Operation
00	AND (A & B)
01	Addition (A + B)
10	Subtraction (A - B)
11	Reduction XOR operation on B

Q3) Designing a Multi-bit Adder Using Module Instantiation

Tasks:

1. **Design a 1-bit Full Adder:**
   - Create a Verilog module named full_adder.
   - Use one assign statement to compute the **Sum** and **Carry Out (Cout)**.
     Hint: use concatenation (curly braces).

   - Inputs: **A, B, and Cin (carry-in).**
   - Outputs: Sum and Cout.

2. **Extend to a 2-bit Adder:**
   - Create a Verilog module named adder_2bit.
   - Instantiate **two** full_adder modules inside adder_2bit.
   - Connect them properly so that the carry propagates from **LSB to MSB**.
     Hint: (The first full adder generates an intermediate carry (**Cout_1**), which is passed as the carry-in (**Cin**) to the second full adder. And the carry out of the adder_2bit will be the carry bit generated by the second full adder. This ensures correct addition of multi-bit numbers, allowing the carry to ripple through successive stages)

   - Inputs: A[1:0], B[1:0], and Cin.
   - Outputs: Sum[1:0] and Cout.

Submission Guidelines:

- The submitted file must be a **PDF file**.
- Name the file in the following format:
  StudentName _assignment2.pdf
- The PDF must include:
  - Verilog design codes
  - Waveform snippets from **QuestaSim** showing test results.