Session 4 assignment

.....

Q1) Build a 64-bit arithmetic shift register, with synchronous load. The shifter can shift both left and right, and by 1 or 8 bit positions, selected by signal amount.

An arithmetic right shift shifts in the sign bit of the number in the shift register (q[63] in this case) instead of zero as done by a logical right shift. Another way of thinking about an arithmetic right shift is that it assumes the number being shifted is signed and preserves the sign, so that arithmetic right shift divides a signed number by a power of two.

There is no difference between logical and arithmetic left shifts.

Verify your code with testbench.

Inputs:

- **load: Loads shift register with data[63:0] instead of shifting.**
- **en: Chooses whether to shift or not.**
- **amount: Chooses which direction and how much to shift.**
  - **2'b00: shift left by 1 bit.**
  - **2'b01: shift left by 8 bits.**
  - **2'b10: shift right by 1 bit.**
  - **2'b11: shift right by 8 bits.**

Output:

- **q: The contents of the shifter.**

Q2) design a paramatrized Johnson counter, with a reset value of 0

Inputs: reset, clk.

Output: (out) parallel output of counter with parametrized width.

Verify your code with testbench, default width is 4.

The states should be as as following

CP	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Hint: if you didn't get the design idea from the states you can search about Johnson counter.

Q3) Use Verilog generate blocks and module instantiation to create a parameterized operator that performs addition or multiplication based on a Type parameter.

Variables	Type	Size
A	input	WIDTH
B	input	WIDTH
out	input	2*WIDTH

Operations Table:

Parameter "Type"	Operation
0	N-bit Adder
1	N-bit Multiplier

Task:

1. **Create two separate modules:**
    - **Adder: Performs N-bit addition.**
    - **Multiplier: Performs N-bit multiplication.**
2. **Inside Operator, use a generate block with conditional instantiation based on Type.**
    - **If Type == 0: instantiate the Adder**
    - **If Type == 1: instantiate the Multiplier**
3. **Use a testbench to verify both modes, check on the output. First make "TYPE" = 0, take your screens, then make "TYPE" = 1.**

Hint: you have 2 parameters "WIDTH" and "TYPE" choose "WIDTH" = *8;* and "TYPE" by default = 0;

Hint: you have to make 3 modules adder, multiplier, and "Top" where you use generate block and instantiate the two other modules.

Submission Guidelines:

- **The submitted file must be a PDF file.**
- **Name the file in the following format: StudentName _assignment4.pdf**
- **The PDF must include:**
  - **Verilog design codes**
  - **Testbenches for all questions and the DO file**
  - **Waveform snippets from QuestaSim showing test results.**