Combinational Circuit Design & Sequential Logic Design

Design the following circuits using Verilog **and create a randomized testbench** for question 1.

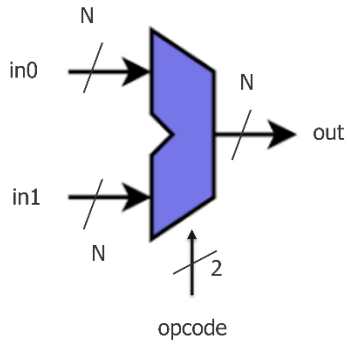
1) Implement a N-bit ALU with the following specifications

Parameters:

- N: defines the ports width (default is 4)
- OPCODE: defines the operation to be done (default is 0)

Ports:

- In0 (Width: N)
- In1 (Width: N)
- Out (Width: N)



Inputs		Outputs
Opcode		Operation
0	0	Addition
0	1	OR
1	0	Subtraction (in0-in1)
1	1	XOR

After designing the ALU, create 4 testbenches where each testbench will verify different parameter configurations as follows:

- Testbench 1: N=4 & OPCODE = 0
- Testbench 1: N=4 & OPCODE = 1
- Testbench 1: N=4 & OPCODE = 2
- Testbench 1: N=4 & OPCODE = 3

In the testbenches, you have to write to make the testbench self-checking while randomizing the inputs.

Draw on a piece of paper the 4 expected schematics if we synthesized the design with OPCODE = 0, OPCODE = 1, OPCODE = 2, and OPCODE = 3.

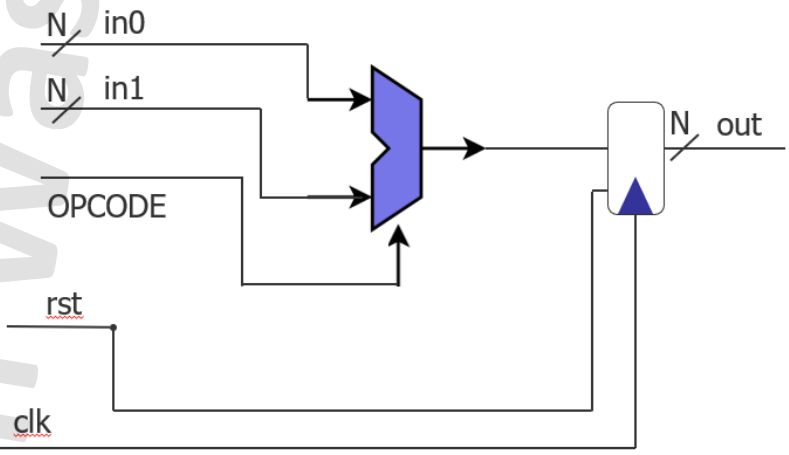
2) Modify on question 1 design so that the output “out” is evaluated with the positive edge of a register as follows

Parameters:

- N: defines the ports width (default is 4)
- OPCODE: defines the operation to be done (default is 0)

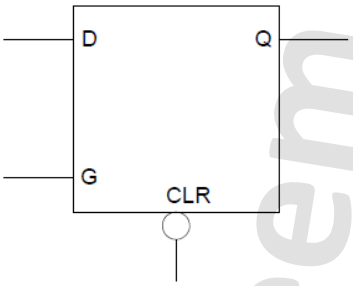
Ports:

- In0 (Width: N)
- In1 (Width: N)
- Out (Width: N)
- Clk (new port added)
- Rst (new port added)
    - Active high synchronous reset



- No testbench is needed.
- Redraw on a piece of paper the 4 expected schematics if we synthesized the design with OPCODE = 0, OPCODE = 1, OPCODE = 2, and OPCODE = 3.

3) Implement Data Latch with active low Clear. No testbench is needed.



Input	Output
CLR, D, G	Q

Truth Table

CLR	G	D	Q
0	X	X	0
1	0	X	Q
1	1	D	D

4) Implement the following latch as specified below. No testbench is needed.

Parameters

LAT_WIDTH: Determine the width of input data and output q

Ports

Name	Type	Description
aset	Input	Asynchronous set input. Sets q[] output to 1.
data[]		Data Input to the D-type latch with width LAT_WIDTH
gate		Latch enable input
aclr		Asynchronous clear input. Sets q[] output to 0.
q[]	Output	Data output from the latch with with LAT_WIDTH

If both aset and aclr are both asserted, aclr is dominant. aset signals sets **all** output bits to 1.

5)

A. Implement T-type (toggle) Flipflop with active low asynchronous reset (forces q to 0 and forces qbar to 1). T-Flipflop has input t, when t input is high the outputs toggle else the output values do not change. No testbench is needed.
    - Inputs: t, rstn, clk
    - Outputs: q, qbar
B. Implement Asynchronous D Flip-Flop with Active low reset (forces q to 0 and forces qbar to 1). No testbench is needed.
    - Inputs: d, rstn, clk
    - Outputs: q, qbar
C. Implement a parameterized asynchronous FlipFlop with Active low reset with the following specifications.
    - Inputs: d, rstn , clk
    - Outputs: q, qbar
    - Parameter: FF_TYPE that can take two valid values, DFF or TFF. Default value = “DFF”. Design should act as DFF if FF_TYPE = “DFF” and act as TFF if FF_TYPE = “TFF”. When FF_TYPE equals “DFF”, d input acts as the data input “d”, and when FF_TYPE equals “TFF”, d input acts the toggle input “t”.
    - No testbench is needed.

Deliverables:

1) The assignment should be submitted as a PDF file with this format <your_name>_Assignment2 for example Kareem_Waseem_Assignment2
2) Snippets from the waveforms captured from QuestaSim for each design with inputs assigned values and output values visible

Note that your document should be organized as 5 sections corresponding to each design above, and in each section, I am expecting the Verilog code for the design, testbench for question 1 and the waveforms snippets