Combinational Circuit Design

Design the following circuits using Verilog **and create a testbench (directed or randomized)** for each design to check its functionality.

1) Design a 4-bit priority encoder using 2 implementations, the following truth table is provided where x is 4-bit input and y is a 2-bit output. The first implementation should use the casex construct. The second implementation should use if-else construct. Testbench should instantiation the 2 designs. Treat the casex design as the DUT and the if-else design as the reference(golden) model. Generate randomized stimulus and make it a self-checking testbench.

```
x3 x2 x1 x0  y1 y0
------------------
 1  X  X  X   1  1
 0  1  X  X   1  0
 0  0  1  X   0  1
 0  0  0  X   0  0
```

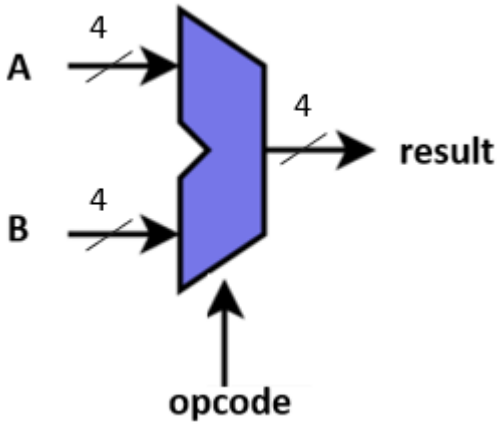
2) Design a decimal to BCD "Binary Coded Decimal" encoder has 10 input lines D0 to D9 and 4 output lines Y0 to Y3 . Below is the truth table for a decimal to BCD encoder. Output should be held LOW if none of the following input patterns is observed.

Input										Output			
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

3) Design N-bit ALU that perform the following operations

- The design has 3 inputs and 1 output
- For the subtraction, subtract B from A "A – B"
- Parameter N has default value = 4.

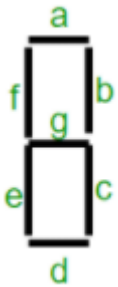
Inputs		Outputs
opcode		Operation
0	0	Addition
1	0	Subtraction
0	1	OR
1	1	XOR



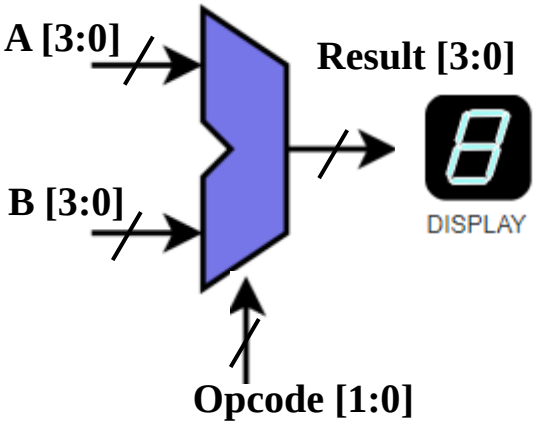
4) Implement 4-bit ALU display on 7 Segment LED Display

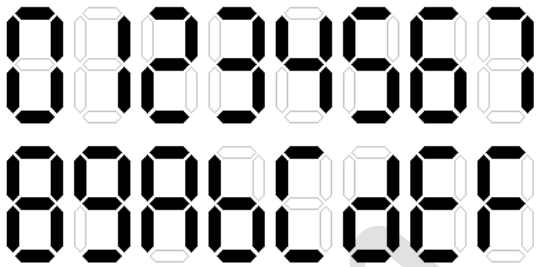
- The design has 4 inputs: A, B, opcode, enable.
- The design has 7 outputs (a-g)
- Instantiate the N-bit ALU designed in the previous design with parameter N = 4
- ALU should execute the operation on A and B depending on the input opcode
- ALU output should be considered as the digit to be displayed on the 7 segment LED display
- Below the truth table of the 7-segment decoder

	Input	Output						
Digit	enable	a	b	c	D	e	f	g
0	1	1	1	1	1	1	1	0
1	1	0	1	1	0	0	0	0
2	1	1	1	0	1	1	0	1
3	1	1	1	1	1	0	0	1
4	1	0	1	1	0	0	1	1
5	1	1	0	1	1	0	1	1
6	1	1	0	1	1	1	1	1
7	1	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1	1
9	1	1	1	1	1	0	1	1
A	1	1	1	1	0	1	1	1
B	1	0	0	1	1	1	1	1
C	1	1	0	0	1	1	1	0
D	1	0	1	1	1	1	0	1



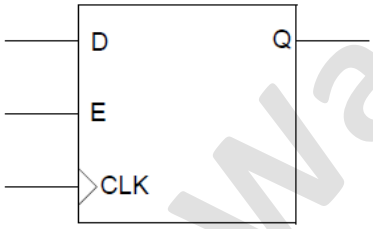
E	1	1	0	0	1	1	1	1
F	1	1	0	0	0	1	1	1
X	0	0	0	0	0	0	0	0





7-segment decoder

5) Implement D-Type Flip-Flop with active high Enable. **DO not write a testbench for this sequential design (We have not learnt how to verify a sequential design yet)**



Input	Output
D, E, CLK	Q

Truth Table

E	CLK	D	Q_{n+1}
0	X	X	Q_n
1	not Rising	X	Q_n
1	↑	D	D

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 and the waveforms snippets