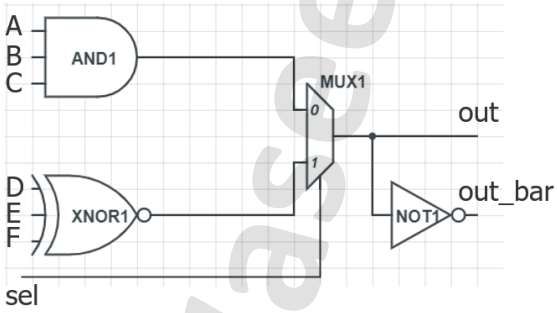
Combinational Circuit Design

Design the following circuits with Verilog using assign statements

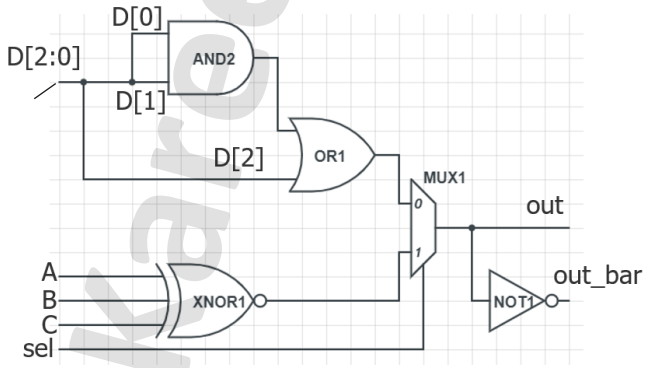
1)

- The design has 7 inputs and 2 outputs
- Use assign statements to design the following



2)

- The design has 5 inputs and 2 outputs
    - Inputs
        i. D -> width = 3
        ii. A, B, C, Sel -> width = 1
    - Outputs
        i. Out, out_bar -> width = 1
- Use Behavioral coding style to implement the following



3) Implement 4-bit adder using addition operator and assign statement

- The design takes 2 inputs (**A, B**) and the summation is assigned to output (**C**) ignoring the carry

4) Implement 2-to-4 Decoder using conditional operator (A logic decoder has n input lines and 2^n output lines. Each output line corresponds to a unique combination of the input values.)

- The design has input **A** (2 bits) and output **D** (4 bits)
- you can use the following format for the conditional operator.
- assign <output_signal> = <condition1> ? <value1> : <condition2> ? <value2> : <default_value>);

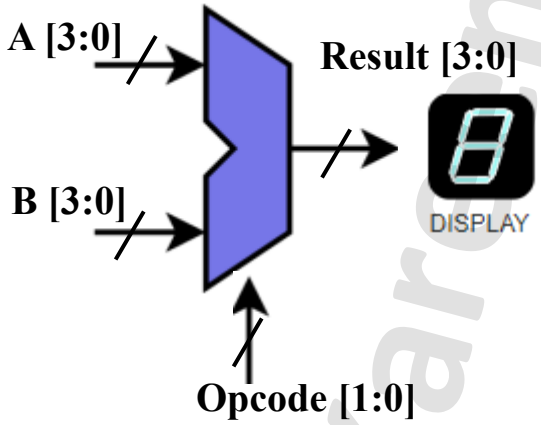
A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

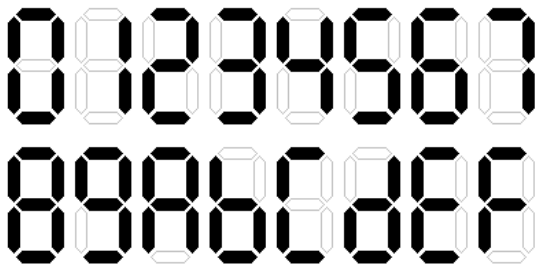
5) Implement an even parity generator module using assign statement. In case you don't know what a parity bit is, please check this link. The design input is a bus where a reduction operator will be used to generate the even parity bit.

- The design has 1 input **A** (8 bits) and 1 output **out_with_parity** (9 bit) where the parity bit calculated will be inserted in the least significant bit of the output bus and the remaining bits will be the input A (Hint: use concatentation).

6)

In this design, you will **connect your 4-bit ALU (done in the lab) to a 7-segment display** so that the result of the ALU operation is shown as a **hex digit (0–F)**.





7-segment decoder

Your design will:

- **Use your ALU module** from the previous lab — with parameter N = 4.
  - You should instantiate it inside your design
- Takes **two 4-bit inputs** A and B.

- Takes a opcode to decide which ALU operation to perform.
- Takes an **enable** input to control whether the display is active.
- Uses a **7-segment display** (with 7 output wires: a to g) to show the ALU result.

What the ALU Does

- The ALU takes A, B, and opcode, and computes a 4-bit result.
- This result will be passed to a **7-segment decoder**.
- The decoder converts the 4-bit result into 7 control signals (a–g) to light up the display.

What the Display Shows

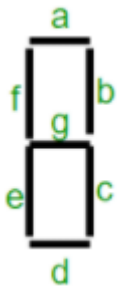
- The 7-segment display shows the **hex value** of the ALU result.
    - If the result is 4'b1010 (10 in decimal), it should show A.
    - If the result is 4'b0001 (1 in decimal), it should show 1.
    - And so on, up to F.

Inputs and Outputs

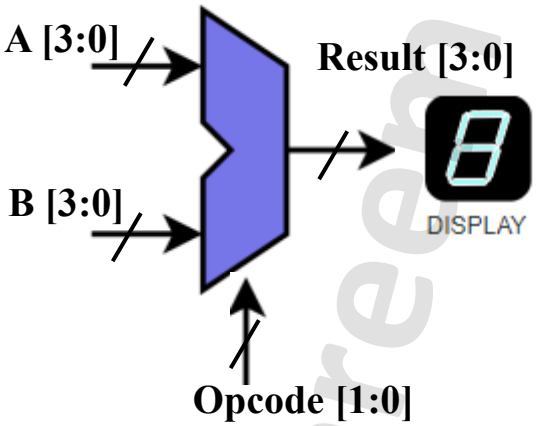
Signal	Type	Description
A	input [3:0]	First ALU operand
B	input [3:0]	Second ALU operand
opcode	input	Selects the ALU operation
enable	input	Enables the display
a–g	output	Controls for the 7-segment LEDs

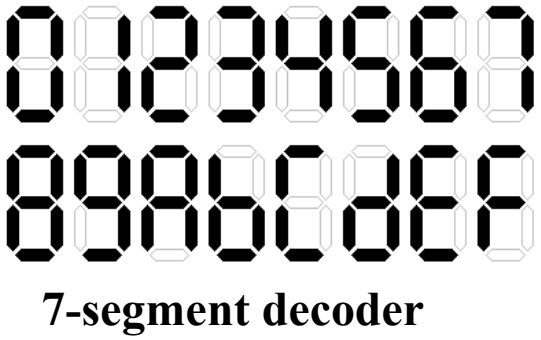
Decoder Truth Table

	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





Deliverables:

Deliverables: The assignment should be submitted as a PDF file with this format <your_name>_Assignment1 for example Kareem_Waseem_Assignment1

Note that your document should be organized as 6 sections corresponding to each design above, and in each section, I am expecting the Verilog code for the design, and waveform snippets forcing different input values to verify the functionality of the design.