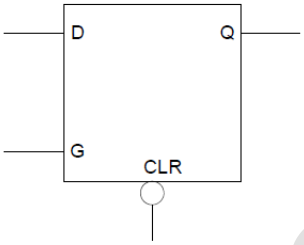
Sequential Logic Design

Design the following circuits using Verilog **and create a testbench** for each design to check its functionality

1) Implement Data Latch with active low Clear



Input	Output
CLR, D, G	Q

Truth Table

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

2) Implement the following latch as specified below

Parameters

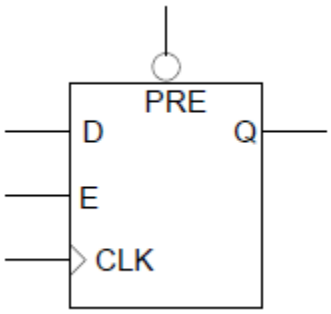
LAT_WIDTH: Determine the width of input data and output q

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.

3) Implement D-Type Flip-Flop with active high Enable and active low Preset.



Input	Output
D, E, PRE, CLK	Q

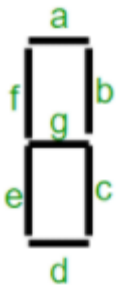
Truth Table

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

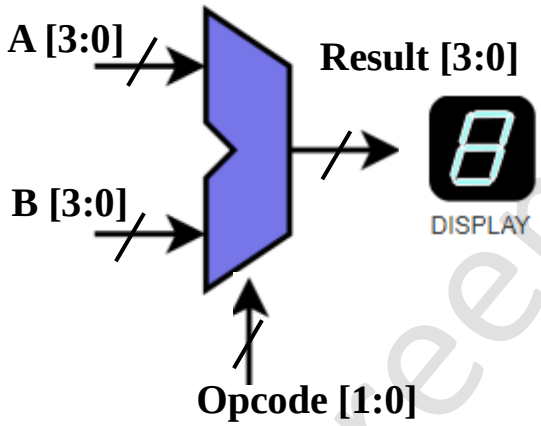
4) Implement 4-bit ALU display on 7 Segment LED Display

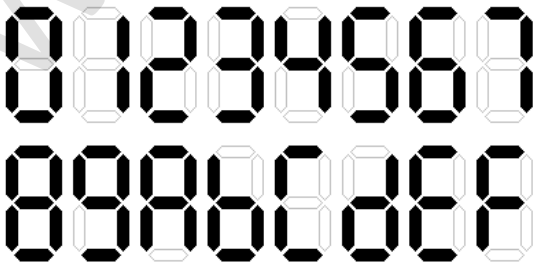
- The design has 4 inputs: A, B, opcode, enable
- The design has 7 outputs (a-g)
- Instantiate the N-bit ALU designed in the previous assignment with parameter N = 4
- ALU should execute the operation on A and B depending on the input opcode
- ALU output should be considered as the digit to be displayed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