1. Construct a finite state machine for the 3-bit Gray code counter. The counter is to be designed with one input terminal (which receives pulse signals) and one output terminal. It should be capable of counting in the Gray system up to 7 and producing an output pulse for every 8 input pulses. After the count 7 is reached, the next pulse will reset the counter to its initial state, i.e., to a count of zero.

The state transitions will be: S0: 000, S1:001, S2:011, S3:010, S4:110, S5:111, S6:101, S7:100 --> S0

Output will be high only from S7-->S0.

a) Construct the state assignment, the state diagram, state table, transition and output table, excitation table.

b) Then build the K-map for the same and design the circuitry or logic diagram.

c) Write the Verilog code module to implement the 3-bit Gray code counter.

d) Now, add a test bench to test the 3-bit Gray code counter.

2. Write a detailed description of eight-bit adder/subtracter to add/subtract two eight-bit two's complement numbers. and it's working with the proper circuit diagram in a PDF file. Then write the Verilog code module to implement an eight-bit adder/subtracter. It will be implemented in two modules. First, module implements a one-bit adder/subtracter with four inputs a, b, cin, and opcode, and two outputs sum and carry. For the addition operation the input opcode will be 0 and 1 for subtraction operation. The top module implements the eight-bit adder/subtracter using the one-bit adder/subtracter module. There will be two inputs for this module the two input numbers and the opcode and produces the sum and whether there is an overflow as the outputs. Now, add a test bench to test the eight-bit adder/subtracter. Your test bench must have fifteen different inputs. Put five-time unit delay between consecutive inputs.