**Lab 1**

* 1. Write Verilog code to describe the following functions

f1 =ac’+bc+b’c’

f2 = (a+b’+c)(a+b+c’)(a’+b+c’)

Check whether f1 and f2 in question 1 are functionally equivalent or not.

* 1. Simplify the following functions using K-map and implement the circuit using logic gates.

1. f(A,B,C,D) = ∑m(1,3,4,9,10,12) + D(0,2,5,11)
2. f(A,B,C,D) = ∏M(6,9,10,11,12) + D(2,4,7,13)
   1. Minimize the following expression using K-map and simulate using only NAND gates.

f(A,B,C,D)= πM(2,6,8,9,10,11,14)

* 1. Minimize the following expressions using K-map and simulate using only NOR gates.

f(A,B,C,D)= ∑m(0,1,2,5,8,9,10)

* 1. Simulate a circuit that has four inputs, x1, x2, x3, and x4, which produces an output value of 1 whenever three or more of the input variables have the value 1; otherwise, the output has to be 0.

**Lab 2**

Write behavioral Verilog code to implement the following and simulate

1. Half adder and full adder
2. Four-bit adder/ subtractor
3. 2-bit multiplier
4. Single-digit BCD adder using a four-bit adder(s).
5. Write the Verilog code to implement a 4-bit adder
6. Design and simulate a circuit that determines how many bits in a six-bit unsigned number are high.

**Lab 3**

1. Using **for** loop, write behavioral Verilog code to convert an N bit grey code into equivalent binary code.
2. Write and simulate the Verilog code for a 4-bit comparator using 2-bit comparators.
3. Write behavioral Verilog code for

* an 8 to 1 multiplexer using **case** statement
* a 2 to 1 multiplexer using the **if-else** statement.

Using the above modules write the hierarchical code for a 16 to 1 multiplexer.

1. Implement F(a,b,c,d) = a’b + ac’ + abd’ + bc’d using 8 to 1 multiplexer and write the Verilog code for the same.
2. Write and simulate the Verilog code for a BCD to Excess 3 code converter using 8 to 1 multiplexers and other necessary gates.
3. Write behavioral Verilog code for a 4 to 1 multiplexer using **conditional** operator. Use this to write the hierarchical code for a 16 to 1 multiplexer.
4. Implement a 3 input majority function using only 2 to 1 multiplexers and write the Verilog code for the same.

**Lab 4**

1. Write behavioral Verilog code for a 2 to 4 decoder with active low enable input and active high output using **case** statement. Using this, design a 4 to 16 decoder with active low enable input and active high output and write the Verilog code for the same.
2. Write behavioral Verilog code for 16 to 4 priority encoder using **for** loop.
3. Design and simulate a combinational circuit with external gates and a 4 to 16 decoder built using a decoder tree of 2 to 4 decoders to implement the functions below.

F= ab’c + a’cd + bcd’ , G=acd’ + a’b’c and H=a’b’c’ + abc + a’cd

1. Design and implement a full adder using 2 to 4 decoder(s) and other gates.
2. Implement the function f(w1,w2,w3)=∑m(0,1,3,4,6,7) by using 3 to 8 binary decoder and an OR gate. Write the Verilog code to implement the same.
3. Design and implement a 3 input majority function using 2 to 4 decoder(s) and other necessary gates.

**Lab 5**

1. Write behavioral Verilog code for a negative edge triggered T FF with asynchronous active low reset.
2. Write behavioral Verilog code for a positive edge-triggered JK FF with synchronous active high reset
3. Design and simulate the following counters
4. 4-bit ring counter.
5. 5 bit Johnson counter.
6. 4 bit synchronous up counter
7. 3 bit synchronous up/down counter with a control input up/. If up/ = 1, then the circuit should behave as an up counter. If up/ = 0, then the circuit should behave as a down counter.
8. Write behavioral Verilog code for positive edge-triggered D FF with synchronous reset.
9. Write Verilog code for an N bit shift register.
10. Write Verilog code for a 2-bit asynchronous up counter.
11. Assume a 4-bit signal A and a clock as inputs and a 4-bit signal Y as output. Design a circuit which repeats the operations cyclically with a clock as follows:

Clock(input) Y(output)

1st clock cycle rotate A by 1 bit to the left

2nd clock cycle rotate A by 2 bit to the left

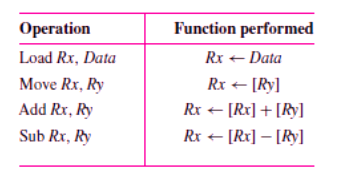
3rd clock cycle rotate A by 3 bit to the left

4th clock cycle complement of A

And then repeats in the same pattern.

**Lab 6**

1. Write and simulate the Verilog code to swap the contents of two registers using multiplexers.
2. Simulate a simple processor that can perform the following functions:



1. Write and simulate the Verilog code to swap the contents of two registers using tristate drivers.