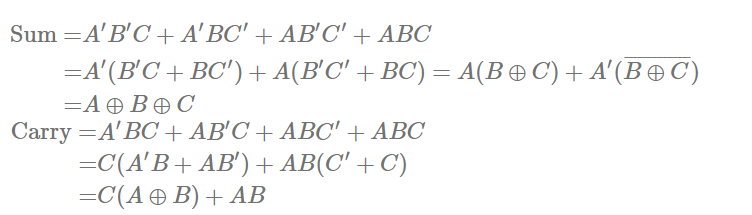
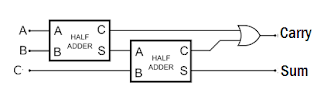
**Lab Assignment 4**

**ECSE104L**

Example of Hierarchical Designing in Verilog: -

Full adder Boolean expression-





**Verilog code for Half Adder**

**Half Adder:**  
//Declare the ports of Half adder module  
**module** half\_adder(  
    Data\_in\_A,  
    Data\_in\_B,  
    Data\_out\_Sum,  
    Data\_out\_Carry  
    );  
  
    //what are the input ports.  
    **input** Data\_in\_A;  
    **input** Data\_in\_B;  
    //What are the output ports.  
    **output** Data\_out\_Sum;  
     **output** Data\_out\_Carry;  
       
     //Implement the Sum and Carry equations using Verilog Bit operators.  
     **assign** Data\_out\_Sum = Data\_in\_A ^ Data\_in\_B;  //XOR operation  
     **assign** Data\_out\_Carry = Data\_in\_A & Data\_in\_B; //AND operation  
      
**endmodule**

**Verilog code for full adder-**

//declare the Full adder verilog module.  
**module** full\_adder(  
    Data\_in\_A,  //input A  
    Data\_in\_B,  //input B  
    Data\_in\_C,  //input C  
    Data\_out\_Sum,  
    Data\_out\_Carry  
    );  
  
    //what are the input ports.  
    **input** Data\_in\_A;  
    **input** Data\_in\_B;  
     **input** Data\_in\_C;  
    //What are the output ports.  
    **output** Data\_out\_Sum;  
     **output** Data\_out\_Carry;  
     //Internal variables  
     **wire** ha1\_sum;  
     **wire** ha2\_sum;  
     **wire** ha1\_carry;  
     **wire** ha2\_carry;  
     **wire** Data\_out\_Sum;  
     **wire** Data\_out\_Carry;  
  
     //Instantiate the half adder 1  
    half\_adder  ha1(  
        .Data\_in\_A(Data\_in\_A),  
        .Data\_in\_B(Data\_in\_B),  
        .Data\_out\_Sum(ha1\_sum),  
        .Data\_out\_Carry(ha1\_carry)  
    );  
      
    //Instantiate the half adder 2  
    half\_adder  ha2(  
        .Data\_in\_A(Data\_in\_C),  
        .Data\_in\_B(ha1\_sum),  
        .Data\_out\_Sum(ha2\_sum),  
        .Data\_out\_Carry(ha2\_carry)  
    );  
  
    //sum output from 2nd half adder is connected to full adder output  
    **assign** Data\_out\_Sum = ha2\_sum;    
    //The carry's from both the half adders are OR'ed to get the final carry./  
    **assign** Data\_out\_Carry = ha1\_carry | ha2\_carry;  
      
**endmodule**

Ques 1 Write Verilog code for half adder. Test using university wave form.

Ques 2. Design a four-bit combinational circuit 2’s complementor using exclusive-OR gates and half adder. Write Verilog code in Quartus tool then test using university waveform.

Ques 3. Write Verilog code for full adder. Test using university wave form.

Ques 4. Write Verilog code for 4-bit ripple carry adder using full adder. Test using university wave form.