VERILOG PROJECT 1

Name : Muhammed Said
Surname: Kaya
Student Number : 21627428
Section : 02
Subject : Two n-bit Binary Number Comparision by Adder/Subtractor

Module of İmplementing The Function

```
`timescale 1ns / 1ps

module HalfAdder(input i1 , input i2 , output Sum , output Carry );

        xor(Sum,i1,i2);
        and(Carry,i1,i2);

endmodule

module FullAdder(input i1 , input i2 , input Cin , output Sum , output Cout);

        wire sum1,carry1,carry2;

        HalfAdder ha1(i1 ,i2 , sum1 , carry1);
        HalfAdder ha2(sum1, Cin , Sum , carry2);

        or(Cout,carry2,carry1);

endmodule

module EightBitAdderSubtractor(
    input [7:0]A,B,
    input Cin,
    output [7:0]Sum,
    output Cout
    );

        wire c1,c2,c3,c4,c5,c6,c7;

FullAdder fa1( A[0] ,B[0]^Cin, Cin , Sum[0] , c1 );
FullAdder fa2( A[1] ,B[1]^Cin, c1  , Sum[1] , c2 );
FullAdder fa3( A[2] ,B[2]^Cin, c2  , Sum[2] , c3 );
FullAdder fa4( A[3] ,B[3]^Cin, c3  , Sum[3] , c4 );
FullAdder fa5( A[4] ,B[4]^Cin, c4  , Sum[4] , c5 );
FullAdder fa6( A[5] ,B[5]^Cin, c5  , Sum[5] , c6 );
FullAdder fa7( A[6] ,B[6]^Cin, c6  , Sum[6] , c7 );
FullAdder fa8( A[7] ,B[7]^Cin, c7  , Sum[7] , Cout);

endmodule

module Compare(Sum,Cout,LEQ,ZERO);
        input [7:0]Sum;
        input Cout;
        output LEQ;
        output ZERO;

        wire [7:0]Sum;
        wire Cout;
        reg LEQ;
        reg ZERO;

        always @(Sum or Cout)
        begin
                if(Sum==0 )begin
                        ZERO=1;
                        LEQ =1;
                end
                if(Sum!=0 && Cout == 1)begin
                        ZERO=0;
                        LEQ =0;
                end
                if(Sum!=0 && Cout == 0)begin
                        ZERO=0;
                        LEQ =1;
                end

        end
```

```
endmodule


module main(
        input [7:0]A,B,
        output ZERO,LEQ );


        wire [7:0]Sum;
        wire Cout;


        EightBitAdderSubtractor
f1(A,B,1,Sum,Cout);
        Compare f2(Sum,Cout,LEQ,ZERO);



endmodule
```

Testbench Module Of The Function

```
`timescale 1ns / 1ps


module Testbench;
	// Inputs
	reg [7:0] A;
	reg [7:0] B;

	// Outputs
	wire ZERO;
	wire LEQ;
// Instantiate the Unit Under Test (UUT)
	main uut (
		.A(A),
		.B(B),
		.ZERO(ZERO),
		.LEQ(LEQ)
	);
	initial begin
		A = 0;  B = 0;  #100;
		A = 20; B = 15; #100;
		A = 15; B = 15; #100;
		A = 15; B = 20; #100;
		A = 30; B = 1;  #100;
		A = 1;  B = 30; #100;
		A = 100; B = 109; #100;
		A = 121; B = 121; #100;
		A = 79;  B = 43;  #100;
		A = 09;  B = 09;
		// Add stimulus here
	end
endmodule
```

Waveform from The Simulation