LABORATORY PROJECT - 1

21604552 – Emre Hancı

Hacettepe University - Department of Computer Engineering
BBM233 Logic Desing Lab.

-Full Adder Verilog Codes

```
module FullAdder(A,B,Cin,S,Cout,Mod);
        input A,B,Cin,Mod;
        output Cout,S;
        wire xorBMod,xor_AB,and_AB,CinXorAB;
        xor(xorBMod,B,Mod);
        xor(xor_AB,A,xorBMod);
        xor(S,Cin,xor_AB);
        and(and_AB,A,xorBMod);
        and(CinXorAB,Cin,xor_AB);
        or(Cout,and_AB,CinXorAB);
endmodule
```

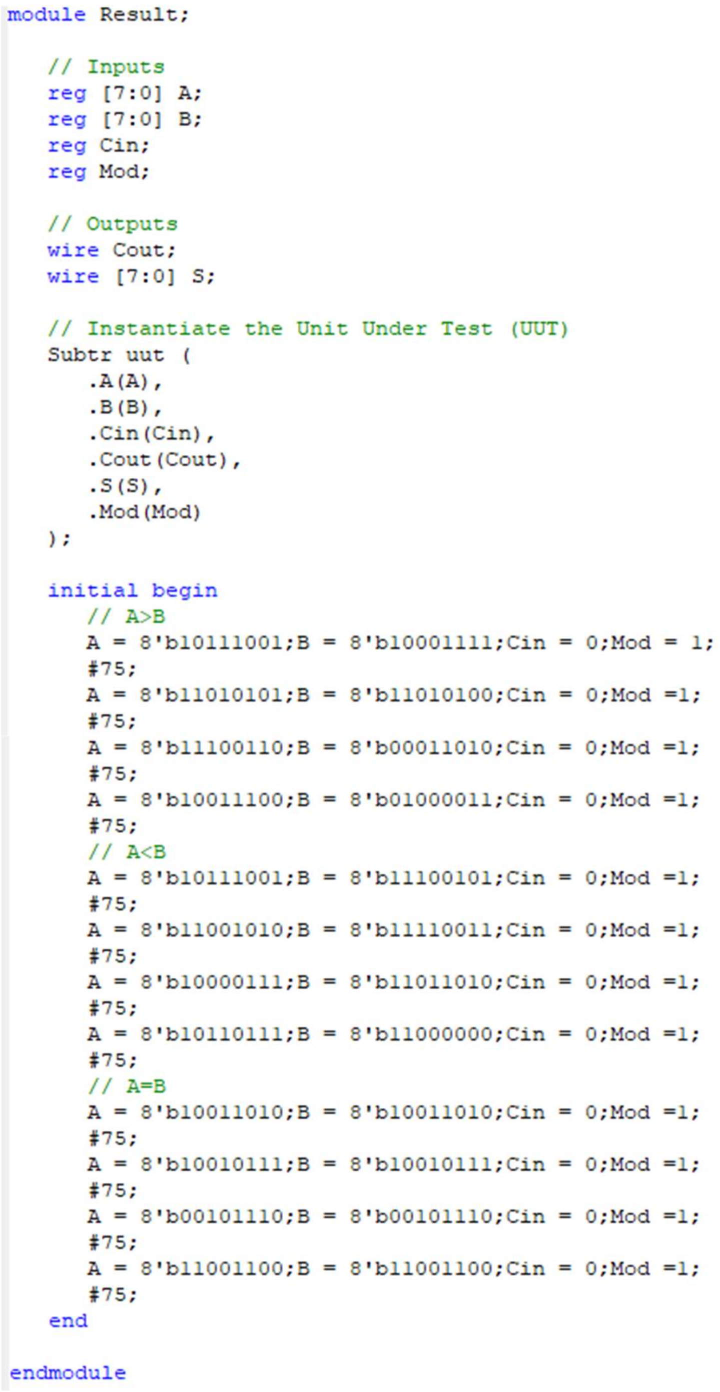
-Subtractor Verilog Codes

```
module Subtr(A,B,Cin,Cout,S,Mod,SW);
  output Cout;
  output [7:0] S;
  output [7:0] SW;
  input Cin,Mod;
  input [7:0] A,B;
  wire Co1,Co2,Co3,Co4,Co5,Co6,Co7,Co8;
  wire Cb1,Cb2,Cb3,Cb4,Cb5,Cb6,Cb7,Cb8;
  FullAdder F1(A[0],B[0],Cin,SW[0],Co1,Mod);
  FullAdder F2(A[1],B[1],Co1,SW[1],Co2,Mod);
  FullAdder F3(A[2],B[2],Co2,SW[2],Co3,Mod);
  FullAdder F4(A[3],B[3],Co3,SW[3],Co4,Mod);
  FullAdder F5(A[4],B[4],Co4,SW[4],Co5,Mod);
  FullAdder F6(A[5],B[5],Co5,SW[5],Co6,Mod);
  FullAdder F7(A[6],B[6],Co6,SW[6],Co7,Mod);
  FullAdder F8(A[7],B[7],Co7,SW[7],Cout,Mod);
  FullAdder F10(SW[0],1,0,S[0],Cb1,0);
  FullAdder F11(SW[1],0,Cb1,S[1],Cb2,0);
   FullAdder F12(SW[2],0,Cb2,S[2],Cb3,0);
   FullAdder F13(SW[3],0,Cb3,S[3],Cb4,0);
   FullAdder F14(SW[4],0,Cb4,S[4],Cb5,0);
   FullAdder F15(SW[5],0,Cb5,S[5],Cb6,0);
  FullAdder F16(SW[6],0,Cb6,S[6],Cb7,0);
  FullAdder F17(SW[7],0,Cb7,S[7],Cb8,0);
  Compare C15(S,Cout,Zero,Leq);
Endmodule
```

-Compare Verilog Codes

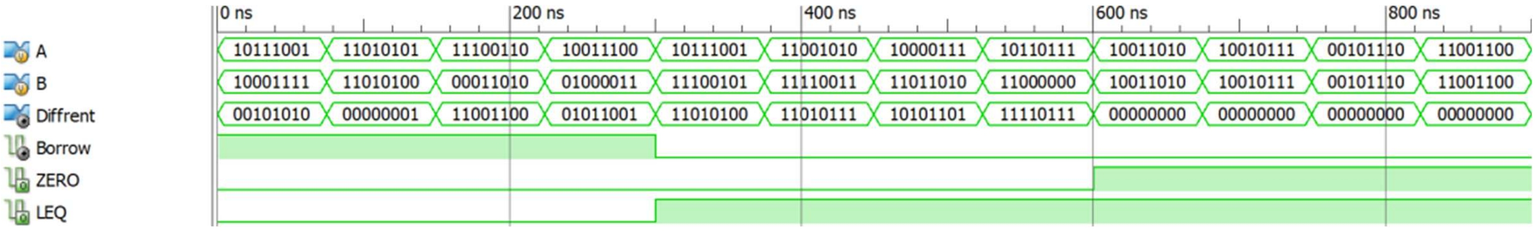
```
module Compare(S,Cout,Zero,Leq);
 input [7:0] S;
 input Cout;
 output Zero,Leq;
 nor(Zero,S[0],S[1],S[2],S[3],S[4],S[5],S[6],S[7]);
 not(Leq,Cout);
endmodule
```

-Testbench



```
module Result;

   // Inputs
   reg [7:0] A;
   reg [7:0] B;
   reg Cin;
   reg Mod;

   // Outputs
   wire Cout;
   wire [7:0] S;

   // Instantiate the Unit Under Test (UUT)
   Subtr uut (
      .A(A),
      .B(B),
      .Cin(Cin),
      .Cout(Cout),
      .S(S),
      .Mod(Mod)
   );

   initial begin
      // A>B
      A = 8'b10111001;B = 8'b10001111;Cin = 0;Mod = 1;
      #75;
      A = 8'b11010101;B = 8'b11010100;Cin = 0;Mod =1;
      #75;
      A = 8'b11100110;B = 8'b00011010;Cin = 0;Mod =1;
      #75;
      A = 8'b10011100;B = 8'b01000011;Cin = 0;Mod =1;
      #75;
      // A<B
      A = 8'b10111001;B = 8'b11100101;Cin = 0;Mod =1;
      #75;
      A = 8'b11001010;B = 8'b11110011;Cin = 0;Mod =1;
      #75;
      A = 8'b10000111;B = 8'b11011010;Cin = 0;Mod =1;
      #75;
      A = 8'b10110111;B = 8'b11000000;Cin = 0;Mod =1;
      #75;
      // A=B
      A = 8'b10011010;B = 8'b10011010;Cin = 0;Mod =1;
      #75;
      A = 8'b10010111;B = 8'b10010111;Cin = 0;Mod =1;
      #75;
      A = 8'b00101110;B = 8'b00101110;Cin = 0;Mod =1;
      #75;
      A = 8'b11001100;B = 8'b11001100;Cin = 0;Mod =1;
      #75;
   end

endmodule
```

-Waveform


0 ns
200 ns
400 ns
600 ns
800 ns
A
B
Diffrent
Borrow
ZERO
LEQ
10111001
11010101
11100110
10011100
10111001
11001010
10000111
10110111
10011010
10010111
00101110
11001100
10001111
11010100
00011010
01000011
11100101
11110011
11011010
11000000
10011010
10010111
00101110
11001100
00101010
00000001
11001100
01011001
11010100
11010111
10101101
11110111
00000000
00000000
00000000
00000000