HLS ASSIGNMENT

NAME : MUNEES SANID
ROLL NO : KVLSI2501076
QUESTION NO:A
Github

VERILOG CODE

```
// Name : Munees Sanid
// Roll No: KVLSI2501076
// Question No:A

module comparator(
        input [3:0]a,
        input [3:0]b,
        output a_gt_b,
        output a_eq_b,
        output a_ls_b);
wire w0,w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11;
wire l1,l2,l3,l4,l5,l6,l7,l8;
// xnor operation
xnor x1(w4,a[3],b[3]);
xnor x2(w5,a[2],b[2]);
xnor x3(w6,a[1],b[1]);
xnor x4(w7,a[0],b[0]);

// not operation of b
not n1 (w0,b[3]);
not n2 (w1,b[2]);
not n3 (w2,b[1]);
not n4 (w3,b[0]);

// not operation of a
not n5 (l5,a[3]);
not n6 (l6,a[2]);
not n7 (l7,a[1]);
not n8 (l8,a[0]);
// a greater tha b
and a1(w8,a[3],w0);
and a2(w9,a[2],w1,w4);
and a3(w10,a[1],w2,w4,w5);
and a4(w11,a[0],w3,w4,w5,w6);
or o1(a_gt_b,w8,w9,w10,w11);

// a == b operation
and a5(a_eq_b,w4,w5,w6,w7);

// a lesser than b operation
and a6(l1,b[3],l5);
and a7(l2,b[2],l6,w4);
and a8(l3,b[1],l7,w4,w5);
and a9(l4,b[0],l8,w4,w5,w6);
or o2(a_ls_b,l1,l2,l3,l4);
endmodule
~
~
```

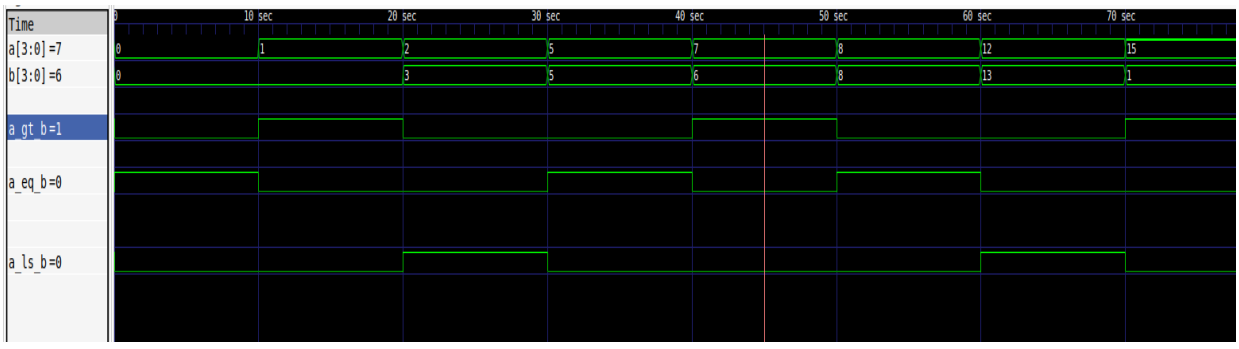
TESTBENCH CODE

```
`include "comparator.v"

module comparator_tb;
reg [3:0] a, b;
wire a_gt_b, a_eq_b, a_ls_b;

comparator dut (.*); // It automatically port the connection

// To display the value
initial begin
$monitor("Time = %0t  a = %b  b = %b   ::   a_gt_b = %b a_eq_b = %b  a_ls_b = %b",$time, a, b, a_gt_b, a_eq_b, a_ls_b);
end
// few combination of 4 bit a and b
initial begin
 a = 4'b0000; b = 4'b0000;
 #10 a = 4'b0001; b = 4'b0000;
 #10 a = 4'b0010; b = 4'b0011;
 #10 a = 4'b0101; b = 4'b0101;
 #10 a = 4'b0111; b = 4'b0110;
 #10 a = 4'b1000; b = 4'b1000;
 #10 a = 4'b1100; b = 4'b1101;
 #10 a = 4'b1111; b = 4'b0001;
 #10 a = 4'b0000; b = 4'b1111;
end

initial begin
 $dumpfile("dump.vcd");
 $dumpvars(0, comparator_tb);
end
endmodule
```

OUTPUT

```
VCD info: dumpfile dump.vcd opened for output.
Time = 0  a = 0000  b = 0000   ::   a_gt_b = 0 a_eq_b = 1  a_ls_b = 0
Time = 10  a = 0001  b = 0000   ::   a_gt_b = 1 a_eq_b = 0  a_ls_b = 0
Time = 20  a = 0010  b = 0011   ::   a_gt_b = 0 a_eq_b = 0  a_ls_b = 1
Time = 30  a = 0101  b = 0101   ::   a_gt_b = 0 a_eq_b = 1  a_ls_b = 0
Time = 40  a = 0111  b = 0110   ::   a_gt_b = 1 a_eq_b = 0  a_ls_b = 0
Time = 50  a = 1000  b = 1000   ::   a_gt_b = 0 a_eq_b = 1  a_ls_b = 0
Time = 60  a = 1100  b = 1101   ::   a_gt_b = 0 a_eq_b = 0  a_ls_b = 1
Time = 70  a = 1111  b = 0001   ::   a_gt_b = 1 a_eq_b = 0  a_ls_b = 0
Time = 80  a = 0000  b = 1111   ::   a_gt_b = 0 a_eq_b = 0  a_ls_b = 1
```

WAVEFORM



QUESTION NO:B

VERILOG CODE

```
module not_mux(input a, output y);
    assign y = a ? 1'b0 : 1'b1;
endmodule

module and2(input a, b, output y);
    assign y = a & b;
endmodule

module or2(input a, b, output y);
    assign y = a | b;
endmodule

module f_realization(input w1, w2, w3, output f);
    wire w1n, w3n;
    wire t1, t2, t3, t4;
    wire o1, o2;

    not_mux n1(w1, w1n);
    not_mux n3(w3, w3n);

    and2 a1(w1n, w3n, t1);
    and2 a2(w1, w3, t2);
    and2 a3(w2, w3, t3);
    and2 a4(w1, w2, t4);

    or2 o_1(t1, t2, o1);
    or2 o_2(t3, t4, o2);
    or2 o_3(o1, o2, f);
endmodule
```

TESTBENCH CODE

```
module testbench;
    reg w1, w2, w3;
    wire f;

f_realization uut(w1, w2, w3, f);

initial begin
$dumpfile("dump.vcd");
$dumpvars(0, testbench);
end

initial begin
$monitor("Time=%0t  w1=%b w2=%b w3=%b :: f=%b", $time, w1, w2, w3, f);
end

initial begin
w1 = 0; w2 = 0; w3 = 0; #5;
w1 = 0; w2 = 0; w3 = 1; #5;
w1 = 0; w2 = 1; w3 = 0; #5;
w1 = 0; w2 = 1; w3 = 1; #5;
w1 = 1; w2 = 0; w3 = 0; #5;
w1 = 1; w2 = 0; w3 = 1; #5;
w1 = 1; w2 = 1; w3 = 0; #5;
w1 = 1; w2 = 1; w3 = 1; #5;

$finish;
end
endmodule
```

OUTPUT

```
VCD info: dumpfile dump.vcd opened for output.
Time=0  w1=0 w2=0 w3=0 :: f=1
Time=5  w1=0 w2=0 w3=1 :: f=0
Time=10  w1=0 w2=1 w3=0 :: f=1
Time=15  w1=0 w2=1 w3=1 :: f=1
Time=20  w1=1 w2=0 w3=0 :: f=0
Time=25  w1=1 w2=0 w3=1 :: f=1
Time=30  w1=1 w2=1 w3=0 :: f=1
Time=35  w1=1 w2=1 w3=1 :: f=1
boolean.v:56: $finish called at 40 (1s)
```

WAVEFORM