**Modelling of Comparators**

module comparator\_ifelse(a,b,eq,lt,gt);

input [3:0] a,b;

output reg eq,lt,gt;

always @(a,b)

begin

if (a==b)

begin

eq = 1'b1;

lt = 1'b0;

gt = 1'b0;

end

else if (a>b)

begin

eq = 1'b0;

lt = 1'b0;

gt = 1'b1;

end

else

begin

eq = 1'b0;

lt = 1'b1;

gt = 1'b0;

end

end

endmodule

module comparator\_tst;

reg [3:0] a,b;

wire eq,lt,gt;

comparator\_ifelse DUT (a,b,eq,lt,gt);

initial

begin

a = 4'b1100;

b = 4'b1100;

#10;

a = 4'b0100;

b = 4'b1100;

#10;

a = 4'b1111;

b = 4'b1100;

#10;

a = 4'b0000;

b = 4'b0000;

#10;

$stop;

end

endmodule

module comparator(

input [3:0] A, B,

output reg A\_grt\_B, A\_less\_B, A\_eq\_B);

always@(\*) begin

A\_grt\_B = 0; A\_less\_B = 0; A\_eq\_B = 0;

if(A>B) A\_grt\_B = 1'b1;

else if(A<B) A\_less\_B = 1'b1;

else A\_eq\_B = 1'b1;

end

endmodule

module tb;

reg [3:0] A, B;

wire A\_grt\_B, A\_less\_B, A\_eq\_B;

comparator comp(A, B, A\_grt\_B, A\_less\_B, A\_eq\_B);

initial begin

$monitor("A = %0h, B = %0h -> A\_grt\_B = %0b, A\_less\_B = %0b, A\_eq\_B = %0b", A, B, A\_grt\_B, A\_less\_B, A\_eq\_B);

repeat(5) begin

A=$random; B=$random; #10;

end

end

endmodule