Answer :

module bcd\_counter(input Clock , Enable , nReset , output reg [3:0]Cout , output reg NextEn) ;

always@(posedge Clock) begin

if(nReset == 1'b0 || (!Enable)) begin

Cout = 4'b0;

NextEn = 0 ;

end

else

begin

Cout = Cout + 1'b1 ;

if(Cout == 9) begin //if cout = 9 , nexten will be set high

NextEn = 1;

end

if(Cout == 10) // if cout= 10 then cout is assigned with 0

begin

Cout = 0 ;

end

end

end

endmodule

module BCD\_TOP(input clock , nrst , enable , output reg [3:0]C01 , C10 , C100 ) ;

//instantiation of bcd conter

wire nexten1 , nexten2,nexten3 ;

bcd\_counter BCD1 (clock , nrst , enable , C01 ,nexten1);

bcd\_counter BCD2 (clock , nrst , nexten1 , C10 ,nexten2);

bcd\_counter BCD3 (clock , nrst , nexten2 , C100 ,nexten3);

endmodule

///////test bench

module testbench;

//input and output port declaration

reg clk ;

reg nrst ;

reg en ;

wire [3:0]Co1 , Co10 , Co100 ;

//clock generation

always #5 clk = ~clk ;

//dut instantiation

BCD\_TOP DUT(clk , en , nrst , Co1 , Co10 , Co100);

initial begin

nrst = 0 ;

clk = 1 ;

en = 0 ;

#20 ;

nrst = 1 ;

en = 1 ;

#2000 ;

$finish() ;

end

//comment this if you dont want to run waveform

initial begin

$dumpfile("dump.vcd");

$dumpvars;

end

endmodule