// Code design here- asynchronous d flip flop

`timescale 100ns / 10ps

module my\_dff(q,qb,d,clk,rst);

output q,qb;

input d,clk,rst;

reg q,qb;

always @(posedge clk or negedge rst)

begin

qb=1;

if (rst==1)

q <= 0;

else

q<=d;

end

endmodule

// Code testbench here

`timescale 10ns / 1ps

module my\_dff\_testbench;

reg d,clk,rst;

wire q,qb;

initial begin // Dump waves

$dumpfile("dump.vcd");

$dumpvars(1);

end

my\_dff m1(q,qb,d,clk,rst);

initial begin

d=1;

clk=1;

end

always begin

rst=1;

#10;

rst=0s;

#10;

end

always

#1.2 clk= ~clk;

always

#6 d= ~d;

initial

#100 $finish;

endmodule