`timescale 1ns / 1ps

module test\_bench();

reg a,b;

initial

begin

a=1;

#200 a=0;

#200 $stop;

end

initial

begin

b=0;

#100 b=1;

#175 b=0;

#75 b=1;

#50 $stop;

end

endmodule

`timescale 1ns / 1ps

module test\_bench();

reg rst\_n,D,clk;

initial clk=0;

always #5 clk=~clk;

initial

begin

rst\_n=0;

#27.5 rst\_n=1;

#27.5 $stop;

end

initial

begin

D=0;

#12.5 D=1;

#25 D=0;

#17.5 $stop;

end

endmodule

`timescale 1ns / 1ps

module d\_ff\_r(

input clk,rst\_n,d,

output reg q

);

always@(posedge clk)

begin

if(rst\_n==0)

q <=1'b0;

else

q <= d;

end

endmodule

`timescale 1ns / 1ps

module test\_bench();

reg rst\_n,D,clk;

wire q;

d\_ff\_r u1(.clk(clk),.rst\_n(rst\_n),.d(D),.q(q));

initial clk=0;

always #5 clk=~clk;

initial

begin

rst\_n=0;

#27.5 rst\_n=1;

#27.5 $stop;

end

initial

begin

D=0;

#12.5 D=1;

#25 D=0;

#17.5 $stop;

end

endmodule

module decode(

input [2:0] a,

output reg [7:0] o

);

always@(\*)

begin

case(a)

3'b000: o=8'b00000001;

3'b001: o=8'b00000010;

3'b010: o=8'b00000100;

3'b011: o=8'b00001000;

3'b100: o=8'b00010000;

3'b101: o=8'b00100000;

3'b110: o=8'b01000000;

3'b111: o=8'b10000000;

default: o=8'b0;

endcase

end

endmodule

module test\_bench();

reg [2:0] a;

wire [7:0] o;

integer i;

decode u1(.a(a),.o(o));

initial

begin

a=3'b000;

for(i=0;i<7;i=i+1)

#20 a=a+3'b001;

#20 $stop;

end

endmodule