Write a Verilog **STRUCTURAL** model HDL program for an **even up-down Binary counter**. Additional specifications below:

*Implement using D flip flops.

*The sequential circuit has the following inputs: Y, reset, and clock.

*When Y==1 and reset==1, the counter is an upstream state transition:

0000-->0010-->0100-->0110-->1000-->1010-->1100-->1110, back to 0000, and repeat.

*When Y==0 and reset==1, the circuit is a downstream state transition:

1110-->1100-->1010-->1000-->0110-->0100-->0010-->0000, back to 1110, and repeat.

*If reset==0 and regardless of the value of Y, output is 0000.

*reset is a negative edge.

*clock is positive edge with half cycle of 5 time units (i.e., #5 clock = ~clock).

Use the following test cases for the testbench:

a.) Initially: reset=0, then after 10 time units, change the reset value to 1.

b.) Then x=0 and ensure enough simulation time to show the entire counter transition.

c.) Then, x=1 and ensure enough simulation time to show the entire counter transition.

d.) Print/monitor the value of the inputs (clk, reset, X) and outputs.

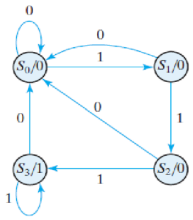
Use control transfer construct (i.e., for-loop, repeat, or always) in the testbench. Brute force construct is not allowed.

The Testbench monitor should display all inputs and outputs

Verilog by structural modelling

```
//Consecutive three or more 1's detector using structural modelling
module FF_D(Q,D, clk, rst);
        output reg Q;
        input D, clk, rst;
        always @(posedge clk, negedge rst)
                if (rst==0)
                        Q <= 1'b0;
                else
                        Q <= D;
endmodule

module detect3(y,A,B,x,clock,reset);
        output y,A,B;
        input x, clock,reset;
        wire DA, DB;
// assign Flip-flop input and output equations
        assign DA = (A&x) | (B&x);
        assign DB = (A&x) | ((~B)&x);
        assign y = A&B;
// instantiate D FF
        FF_D F1(A, DA, clock, reset);
        FF_D F2(B, DB, clock, reset);
endmodule
```



Testbench

```
//Consecutive three or more 1's detector using structural
modelling (with external input and output w/reset)
module detect3_tb();
	wire t_y;
	wire [1:0] t_state;
	reg t_x, t_clk, t_reset;

	detect3 dut(t_y,t_state[1],t_state[0],t_x,t_clk,t_reset);

	initial
		begin
			t_clk = 1'b0;
			forever #5 t_clk = ~t_clk;
		end

	initial
		begin
			t_reset = 1'b0;
			#20 t_reset = 1'b1;
		end
```

```
initial
	begin
		t_x = 1'b1;
		#70 t_x = 1'b0;
	end
initial
	begin
		$display("Program by Sensei RLUy 3 consecutive 1
detector with external input and output using structural
modelling");
		$monitor("time=%03d, Clk=%b, reset=%b, x=%b,
state=%b, y=%b ",$time,t_clk,t_reset,t_x,t_state,t_y);
		$dumpfile("Seq_io_struct.vcd");
		$dumpvars();
	end

initial #100 $finish;
endmodule
```