🡪code for which test bench is written:-

// Code your design here

//`timescale 1ns / 1ps

module votingMachine(

input clock,

input reset,

input mode,

input button1,

input button2,

input button3,

input button4,

output [7:0] led

);

wire valid\_vote\_1;

wire valid\_vote\_2;

wire valid\_vote\_3;

wire valid\_vote\_4;

wire [7:0] cand1\_vote\_recvd;

wire [7:0] cand2\_vote\_recvd;

wire [7:0] cand3\_vote\_recvd;

wire [7:0] cand4\_vote\_recvd;

wire anyValidVote;

assign anyValidVote = valid\_vote\_1|valid\_vote\_2|valid\_vote\_3|valid\_vote\_4;

buttonControl bc1(

.clock(clock),

.reset(reset),

.button(button1),

.valid\_vote(valid\_vote\_1)

);

buttonControl bc2(

.clock(clock),

.reset(reset),

.button(button2),

.valid\_vote(valid\_vote\_2)

);

buttonControl bc3(

.clock(clock),

.reset(reset),

.button(button3),//

.valid\_vote(valid\_vote\_3)

);

buttonControl bc4(

.clock(clock),

.reset(reset),

.button(button4),

.valid\_vote(valid\_vote\_4)

);

voteLogger VL(

.clock(clock),

.reset(reset),

.mode(mode),

.cand1\_vote\_valid(valid\_vote\_1),

.cand2\_vote\_valid(valid\_vote\_2),

.cand3\_vote\_valid(valid\_vote\_3),

.cand4\_vote\_valid(valid\_vote\_4),

.cand1\_vote\_recvd(cand1\_vote\_recvd),

.cand2\_vote\_recvd(cand2\_vote\_recvd),

.cand3\_vote\_recvd(cand3\_vote\_recvd),

.cand4\_vote\_recvd(cand4\_vote\_recvd)

);

modeControl MC(

.clock(clock),

.reset(reset),

.mode(mode),

.valid\_vote\_casted(anyValidVote),

.candidate1\_vote(cand1\_vote\_recvd),

.candidate2\_vote(cand2\_vote\_recvd),

.candidate3\_vote(cand3\_vote\_recvd),

.candidate4\_vote(cand4\_vote\_recvd),

.candidate1\_button\_press(valid\_vote\_1),

.candidate2\_button\_press(valid\_vote\_2),

.candidate3\_button\_press(valid\_vote\_3),

.candidate4\_button\_press(valid\_vote\_4),

.leds(led)

);

endmodule

module voteLogger(

input clock,

input reset,

input mode,

input cand1\_vote\_valid,

input cand2\_vote\_valid,

input cand3\_vote\_valid,

input cand4\_vote\_valid,

output reg [7:0] cand1\_vote\_recvd,

output reg [7:0] cand2\_vote\_recvd,

output reg [7:0] cand3\_vote\_recvd,

output reg [7:0] cand4\_vote\_recvd

);

always @(posedge clock)

begin

if(reset)

begin

cand1\_vote\_recvd <= 0;

cand2\_vote\_recvd <= 0;

cand3\_vote\_recvd <= 0;

cand4\_vote\_recvd <= 0;

end

else

begin

if(cand1\_vote\_valid & mode==0)

cand1\_vote\_recvd <= cand1\_vote\_recvd + 1;

else if(cand2\_vote\_valid & mode==0)

cand2\_vote\_recvd <= cand2\_vote\_recvd + 1;

else if(cand3\_vote\_valid & mode==0)

cand3\_vote\_recvd <= cand3\_vote\_recvd + 1;

else if(cand4\_vote\_valid & mode==0)

cand4\_vote\_recvd <= cand4\_vote\_recvd + 1;

end

end

endmodule

module modeControl(

input clock,

input reset,

input mode,

input valid\_vote\_casted,

input [7:0] candidate1\_vote,

input [7:0] candidate2\_vote,

input [7:0] candidate3\_vote,

input [7:0] candidate4\_vote,

input candidate1\_button\_press,

input candidate2\_button\_press,

input candidate3\_button\_press,

input candidate4\_button\_press,

output reg [7:0] leds

);

reg [30:0] counter;

always @(posedge clock)

begin

if(reset)

counter <= 0; //Whenever reset is pressed, counter started from 0

else if(valid\_vote\_casted) //If a valid vote is casted, counter becomes 1

counter <= counter + 1;

else if(counter !=0 & counter < 5)//If counter is not 0, increment it till 5

counter <= counter + 1;

else //Once counter becomes 100000000, reset it to zero

counter <= 0;

end

always @(posedge clock)

begin

if(reset)

leds <= 0;

else

begin

if(mode == 0 & counter > 0 ) //mode0 -> voting mode, mode 1 -> result mode

leds <= 8'hFF;

else if(mode == 0)

leds <= 8'h00;

else if(mode == 1) //result mode

begin

if(candidate1\_button\_press)

leds <= candidate1\_vote;

else if(candidate2\_button\_press)

leds <= candidate2\_vote;

else if(candidate3\_button\_press)

leds <= candidate3\_vote;

else if(candidate4\_button\_press)

leds <= candidate4\_vote;

end

end

end

endmodule

module buttonControl(

input clock,

input reset,

input button,

output reg valid\_vote

);

reg [30:0] counter;

//1 sec / 10ms = 100000000

always @(posedge clock)

begin

if(reset)

counter <= 0;

else

begin

if(button & counter < 6)

counter <= counter + 1;

else if(!button)

counter <= 0;

end

end

always @(posedge clock)

begin

if(reset)

valid\_vote <= 1'b0;

else

begin

if(counter == 5)

valid\_vote <= 1'b1;

else

valid\_vote <= 1'b0;

end

end

endmodule

🡪testbench:-

// Code your testbench here

// or browse Examples

// Code your testbench here

// or browse Examples

module votingMachine\_tb();

reg clock,reset,mode,button1,button2,button3,button4;

wire [7:0]led;

votingMachine vvt(clock,reset,mode,button1,button2,button3,button4,led);

initial

begin

$monitor($time,"clock=%b,reset=%b,mode=%b,button1=%b,button2=%b,button3=%b,button4=%b,led=%b",clock,reset,mode,button1,button2,button3,button4,led);

clock=0;reset=0;mode=0;button1=0;button2=0;button3=0;

button4=0;

#24 button1=1;mode=0;

#24 button2=1;button1=0;

#24 button3=1;button2=0;

#24 button4=1;button3=0;mode=1;

#24 button1=1;

#24 $finish;

end

always #2 clock=~clock;

endmodule

  