```
module multicycle_tb();

 reg  clock;
 reg  reset;
 wire zeroflag;
 wire IRegen;
 wire [7:0] ALUa;
 wire [7:0] ALUb;
 wire [7:0] ALUout;
 wire [15:0] FetchedInst;
 wire [15:0] Inst;
 wire [3:0] next_state;
 wire [7:0] PCout;
 wire [7:0] R2;
 wire [7:0] R3;
 wire [7:0] RD2;
 wire [2:0] regadr1;
 wire [2:0] regadr2;
 wire [2:0] ShiftCtrl;
 wire [7:0] Shiftin;
 wire [7:0] Shiftout;
 wire [7:0] shiftselectout;
 wire [3:0] state;
 wire [7:0] toReg;
 wire [2:0] wa;


 multicycle DUT(
   reset,
   clock,
   zeroflag,
   IRegen,
   ALUa,
   ALUb,
   ALUout,
   FetchedInst,
   Inst,
   next_state,
   PCout,
   R2,
   R3,
   RD2,
   regadr1,
   regadr2,
   ShiftCtrl,
   Shiftin,
   Shiftout,
   shiftselectout,
   state,
   toReg,
   wa
);

integer j;

initial begin
  clock=0;
  reset = 0;
end

always @(*)
   begin
   for(j=0; j<10000; j=j+1) begin
   clock = ~clock;
         if(R2 != 8'b00000000)
            $display("Error in %1d case",R2);
         else
            $display("No error in %1d case",R2);
   #10;
      end
   #1 $display("t=%t",$time," r2=%1d",R2);
   end


 endmodule
```