```
module consecutive(input CLOCK_50, input x, output reg z);
reg [3:0]state;
always @(posedge CLOCK_50)
begin

      case(state)
         4'b0001://one zero
            begin
               z<=1'b0;
               if(~x)state<=4'b0010;
               else state<=4'b0101;//one one
            end
         4'b0010://two zero
            begin
               z<=1'b0;
               if(~x)state<=4'b0011;
               else state<=4'b0101;//one one
            end
         4'b0011://three zero
            begin
               z<=1'b0;
               if(~x)state<=4'b0100;
               else state<=4'b0101;//one one
            end
         4'b0100://four zero
            begin
               z<=1'b1;
               if(~x)state<=4'b0100;
               else state<=4'b0101;//one one
            end
         4'b0101://one one
            begin
               z<=1'b0;
               if(x)state<=4'b0110;
               else state<=4'b0001;//one zero
            end
         4'b0110://two one
            begin
               z<=1'b0;
               if(x)state<=4'b0111;
               else state<=4'b0001;//one zero
            end
         4'b0111://three one
            begin
               z<=1'b0;
               if(x)state<=4'b1000;
               else state<=4'b0001;//one zero
            end
         4'b1000://four one
            begin
               z<=1'b1;
               if(x)state<=4'b1000;
               else state<=4'b0001;//one zero
            end
         endcase
```