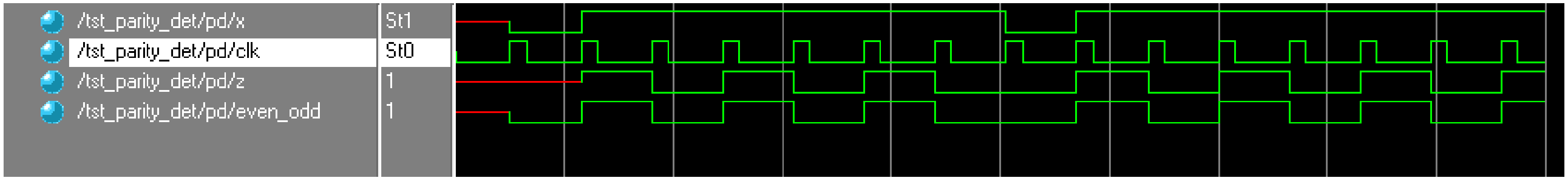
```
module parity_det (x, clk, z);
     input x, clk;
     output z; reg z;

     reg even_odd;

     parameter EVEN = 0, ODD = 1;

     always @(posedge clk)
          case (even_odd)
               EVEN: begin
                    z <= x? 1:0;
                    even_odd <= x? ODD:EVEN;
               end
               ODD: begin
                    z <= x? 0:1;
                    even_odd <= x? EVEN:ODD;
               end
               default: even_odd <= EVEN;
          endcase

endmodule
```

```
module tst_parity_det;

     reg x, clk;
     wire z;


     parity_det pd (x, clk, z);

     initial   begin
          clk = 0;

          forever begin
               #10 clk = ~clk;
               #3 clk = ~clk;
          end
     end

     initial
          repeat (10) @(posedge clk)
               x = $random;

endmodule
```



/tst_parity_det/pd/x St1
/tst_parity_det/pd/clk St0
/tst_parity_det/pd/z 1
/tst_parity_det/pd/even_odd 1