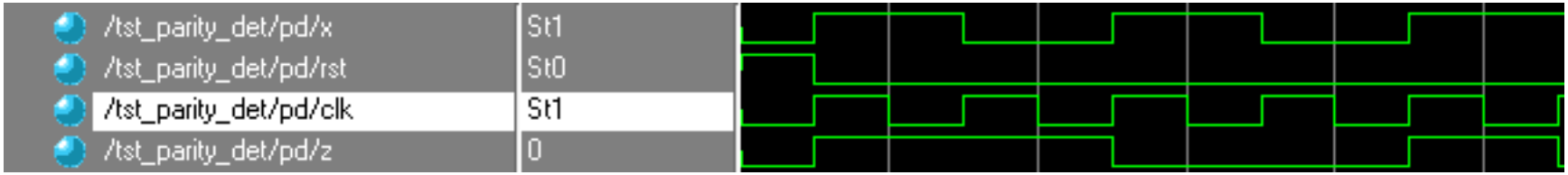
```
module parity_det_1always (x, rst, clk, z);

     input x, rst, clk;
     output z; reg z;

     reg even_odd;

     parameter EVEN = 0, ODD = 1;

     always @(posedge clk, rst)
          if (rst) begin
               even_odd <= EVEN;
               z <= 0;
          end
          else
               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
               endcase

endmodule
```

```
module tst_parity_det;

     reg x, rst, clk;
     wire z;

     parity_det_1always pd (x, rst, clk, z);

     initial   begin
          clk = 0;
          x = 0;
          #10 x = 1;
          #20 x = 0;
          #20 x = 1;
          #20 x = 0;
          #20 x = 1;
     end

     initial   begin
          rst = 1;
          #10 rst = 0;
     end

     always
          #10 clk = ~clk;

endmodule
```

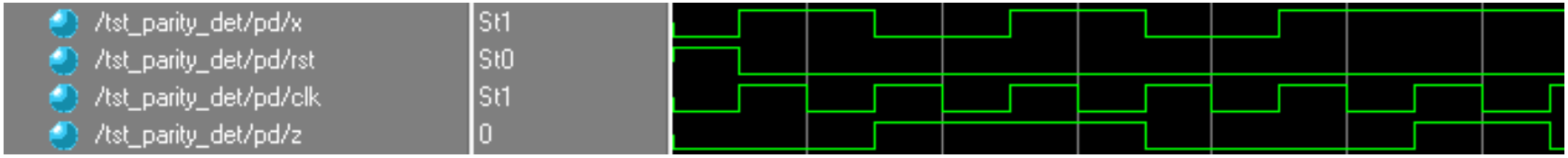
```
//Output of parity_det_1always
```



```
module parity_det_moore (x, rst, clk, z);

     input x, rst, clk;
     output z; reg z;

     reg PS, NS;

     parameter EVEN = 0, ODD = 1;

     always @(posedge clk or rst)
          if (rst)
               PS <= EVEN;
          else
               PS <= NS;
     always @(PS)
          case (PS)
               EVEN: z = EVEN;
               ODD:  z = ODD;
          endcase

     always @(PS or x)
          case (PS)
               EVEN: NS = x? ODD:EVEN;
               ODD:NS = x? EVEN:ODD;
          endcase

endmodule
```

```
module tst_parity_det;

     reg x, rst, clk;
     wire z;

     parity_det_moore pd (x, rst, clk, z);

     initial   begin
          clk = 0;
          x = 0;
          #10 x = 1;
          #20 x = 0;
          #20 x = 1;
          #20 x = 0;
          #20 x = 1;
     end

     initial   begin
          rst = 1;
          #10 rst = 0;
     end

     always
          #10 clk = ~clk;

endmodule
```

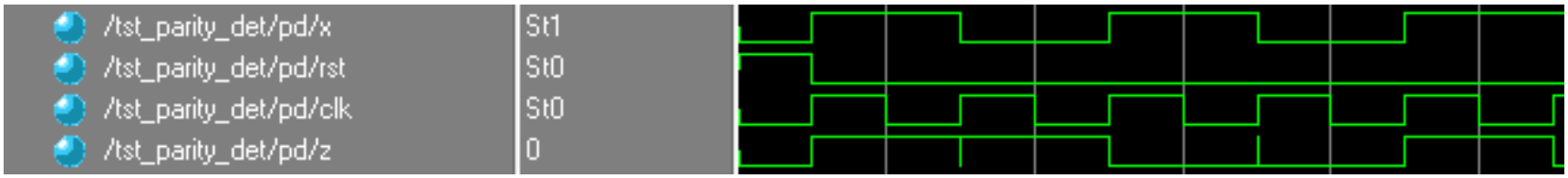
//Output of parity_det_moore



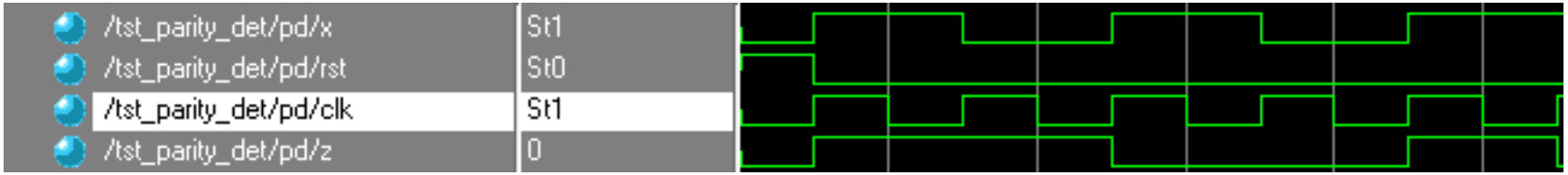
```
module parity_det_mealy (x, rst, clk, z);

     input x, rst, clk;
     output z; reg z;

     reg PS, NS;

     parameter EVEN = 0, ODD = 1;

     always @(posedge clk or rst)
          if (rst)
               PS <= EVEN;
          else
               PS <= NS;

     always @(PS or x)
          case (PS)
               EVEN: begin
                    z = x? 1:0;
                    NS = x? ODD:EVEN;
               end
               ODD: begin
                    z = x? 0:1;
                    NS = x? EVEN:ODD;
               end
          endcase

endmodule
```

```
module tst_parity_det;

     reg x, rst, clk;
     wire z;

     parity_det_mealy pd (x, rst, clk, z);

     initial   begin
          clk = 0;
          x = 0;
          #10 x = 1;
          #20 x = 0;
          #20 x = 1;
          #20 x = 0;
          #20 x = 1;


     end
     initial   begin
          rst = 1;
          #10 rst = 0;
     end

     always
          #10 clk = ~clk;

endmodule
```

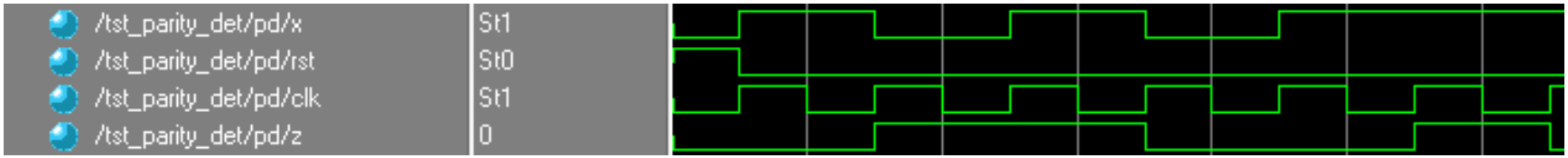
```
//Output of parity_det_mealy
```



//Output of parity_det_1always



//Output of parity_det_moore



//Output of parity_det_mealy