

```

//-----
// Design Name : lfsr
// File Name   : lfsr.v
// Function    : Linear feedback shift register
//-----
module lfsr    (
in,
out            , // Output of the counter
lineout       , // output to line
enable        , // Enable for counter
clk           , // clock input
reset         , // reset input
);

parameter WIDTH= 4;
//-----Output Ports-----
output [WIDTH-1:0] out;
output lineout;
//-----Input Ports-----
input [WIDTH-1:0] in;
input enable, clk, reset;
//-----Internal Variables-----
reg [WIDTH-1:0] out;
reg lineout;
wire          linear_feedback;

//-----Code Starts Here-----
assign linear_feedback = !(out[0] ^ out[1]);

always @(posedge clk)begin
    if (reset) begin // active high reset
        out <= in ;
    end else if (enable) begin
        out <= {linear_feedback,out[3]
                ,out[2], out[1]};

        // stores single bit output
        // from LSFR
        lineout <= out[1];
    end
end
endmodule // End Of Module counter


module lfsr3b    (
in,
out            , // Output of the counter
lineout       , // output to line
enable        , // Enable for counter
clk           , // clock input
reset         , // reset input
);

parameter WIDTH= 3;
//-----Output Ports-----
output [WIDTH-1:0] out;
output lineout;
//-----Input Ports-----
input [WIDTH-1:0] in;
input enable, clk, reset;
//-----Internal Variables-----
reg [WIDTH-1:0] out;
reg lineout;
wire          linear_feedback;

//-----Code Starts Here-----
assign linear_feedback = !(out[0] ^ out[1]);

always @(posedge clk)begin
    if (reset) begin // active high reset
        out <= in ;
    end else if (enable) begin

```

```
        out <= {linear_feedback ,out[2], out[1]};

        // stores single bit output
        // from LSFR
        lineout <= out[1];
    end
end
endmodule // End Of Module counter
```

```
// D-FF
module DFF (d,q,clock,enable,reset);
//-----Output Ports-----
output q;
//-----Input Ports-----
input d,enable, clock, reset;
//-----Internal Variables-----
reg q;
always @(posedge clock) begin
    if (reset) begin
        q <= 0;
    end else if (enable) begin
        q<=d;
    end
end
end // end always block
endmodule // end DFF module
```