BIST Design Tutorial

The main components of a BIST are **Pattern Generator (PG)**, **Response Analyzer (RA)**, and the BIST Controller.

MLFSR (Maximum Linear Feedback Shift Register) design as PG

Following irreducible polynomials represent two MLSFRs.

- **x^10 + x^7 + 1** for **10 inputs**, taping at 10th and 7th stage
- **x^9 + x^5 + 1** for **9 inputs**

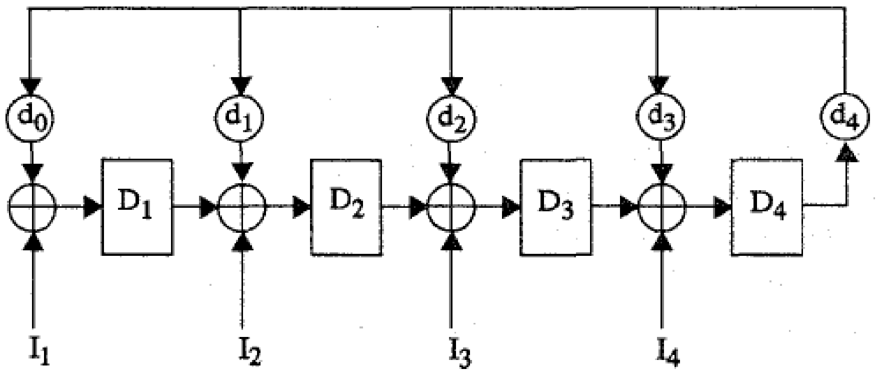
Here is an example source code and testbench for a 10-bit MLSFR.

```
module ML_lfsr(data_out,complete,reset,clock);
input reset;
input clock;
output [9:0] data_out; output complete;
reg complete; reg [9:0] lfsr_reg;
reg [9:0] counter; reg tap;
always@(posedge clock or posedge reset)
begin
   if(reset == 1)
       begin
       lfsr_reg <= 10'b1111111111;
       counter <=  10'b0000000000;
       end
   else
       begin
       tap = lfsr_reg[9] ^ lfsr_reg[6];
       lfsr_reg[9:0] <= { lfsr_reg[8:0], tap };
       counter = counter + 1;
       if(counter < 10'b1111111110)
            complete = 0;
       else
             complete = 1;
       end
   end
       assign data_out = lfsr_reg;
endmodule
```

```
module tb_ML_lfsr;
reg clock,reset;
wire [9:0] dt_out;
ML_lfsr lfsr1(dt_out, complete, reset, clock);
initial
       begin
       clock = 1'b0;
       end
always
       #5 clock = ~clock;
initial
       begin
             reset = 1'b1;
             #10 reset = 1'b0;
       end
endmodule
```

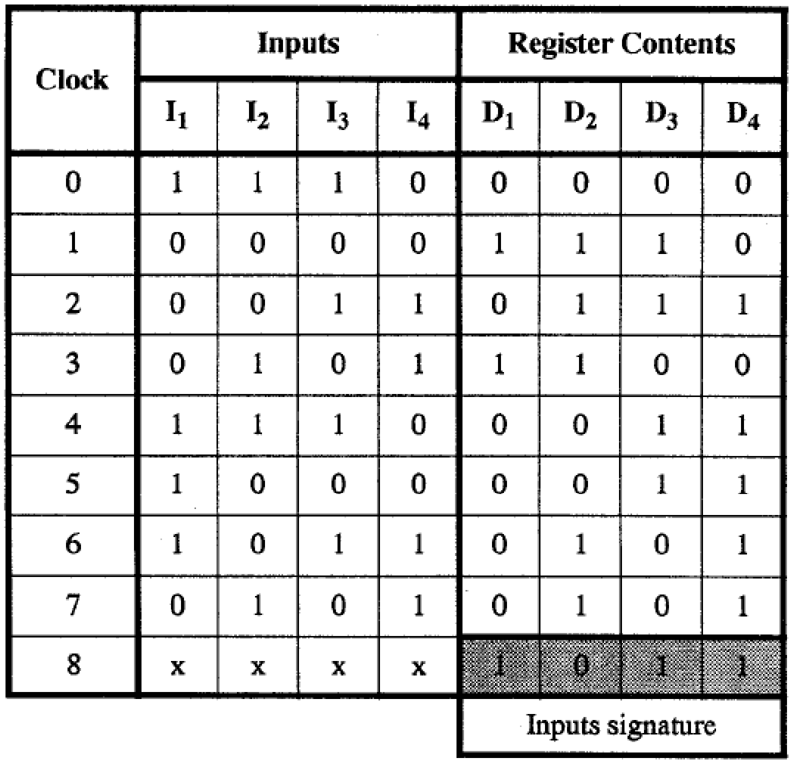
MISR (Multiple Input Signature Register) design as RA

Here is an example for an MISR implementation (please read pp 292-295 in the textbook). The irreducible polynomial used is **1 + x + x^4**, which has four outputs from the circuit under test as it inputs.



4-input MISR

The following figure shows how signature is generated for a given input sequence (please read registers as D0 – D3 instead of D1 to D4).



Clock	Inputs				Register Contents			
	I_1	I_2	I_3	I_4	D_1	D_2	D_3	D_4
0	1	1	1	0	0	0	0	0
1	0	0	0	0	1	1	1	0
2	0	0	1	1	0	1	1	1
3	0	1	0	1	1	1	0	0
4	1	1	1	0	0	0	1	1
5	1	0	0	0	0	0	1	1
6	1	0	1	1	0	1	0	1
7	0	1	0	1	0	1	0	1
8	x	x	x	x	1	0	1	1
					Inputs signature			

MISR signature generation

Here is an example source code and testbench for a MISR:

```
module MISR4bit(dataIn,reset,clock,dataOut);
input [3:0] dataIn;
input reset,clock;
output [3:0] dataOut;
reg [3:0] dataOut;
reg [3:0] misr_temp;

always@(posedge clock or posedge reset)
begin
    if(reset == 1)
       dataOut <= 4'b0000;
    else
         begin
               misr_temp = dataOut;
               dataOut[0] = misr_temp[3] ^ dataIn[0];
               dataOut[1] = misr_temp[3] ^ misr_temp[0] ^ dataIn[1];
               dataOut[2] = misr_temp[1] ^ dataIn[2];
               dataOut[3] = misr_temp[2] ^ dataIn[3];
         end

     end
endmodule
```

```
module tb_MISR4bit;
reg clock,reset;
wire [3:0] dataOut;
reg [3:0] dataIn;

MISR4bit misr4bit1(dataIn,reset,clock,dataOut);

initial
       begin
              clock = 1'b0;
       end
always
       #5 clock = ~clock;
initial
       begin
              reset = 1'b1;
              #10 reset = 1'b0;
              dataIn = 4'b0111;
              #10 dataIn = 4'b0000;
              #10 dataIn = 4'b1100;
              #10 dataIn = 4'b1010;
              #10 dataIn = 4'b0111;
              #10 dataIn = 4'b0001;
              #10 dataIn = 4'b1101;
              #10 dataIn = 4'b1010;
       end
endmodule
```

Controller design

You may need to design a finite state machine.