

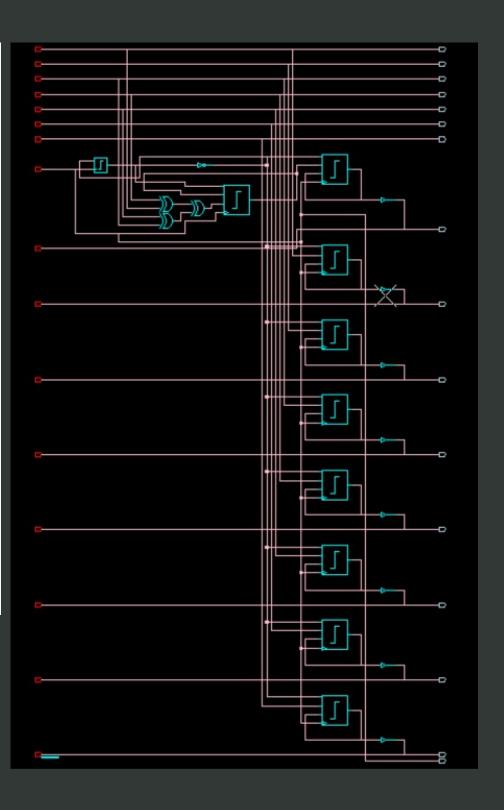
## Código para gerar o Padrão de teste

Bits (n)	Feedback polynomial	Taps	Taps (hex)	Period ( $2^n-1$ )	
2	$x^2 + x + 1$	11	0x3	3	
3	$x^3+x^2+1$	110	0x6	7	
4	$x^4+x^3+1$	1100	0xC	15	4
5	$x^5+x^3+1$	10100	0x14	31	2
6	$x^6+x^5+1$	110000	0x30	63	4
7	$x^7+x^6+1$	1100000	0x60	127	4
8	$x^8 + x^6 + x^5 + x^4 + 1$	10111000	0xB8	255	4

```
5 #define LFSR WIDTH 8
6 #define LFSR TAP BIT1 1
7 #define LFSR TAP BIT2 3
8 #define LFSR TAP BIT3 4
9 #define LFSR TAP BIT4 5
10
11 void updateLFSR(uint8_t*state){
12
           uint8 t lsb=*state & 1;
13
           uint8_t feedback = ((*state>>(LFSR TAP BIT1 - 1)) & 1)^
14
                   ((*state>>(LFSR TAP BIT2 - 1)) & 1)^
15
                   ((*state>>(LFSR TAP BIT3 - 1)) & 1)^
16
                   ((*state>>(LFSR TAP BIT4 - 1)) & 1);
17
           *state>>=1;
18
           if(feedback){
19
                   *state^=(1<<7);//Bit mais significativo(MSB)
20
          }else{
21
                   *state^=(0<<7);//Bit mais significativo(MSB)
22
23 }
24 char *inttostr(int inteiro) {
      char *str = (char *)malloc(32 * sizeof(char));
26
      if (str != NULL) {
27
           sprintf(str, "%d", inteiro);
28
29
      return str;
30 }
31 int main(){
32
           uint8 t lfsr state=0xB8;
33
           int tempo = 0;
34
35
           DEF GENPAT("lfsr sim");
           DECLAR("lfsr state",":2","B",IN,"7 down to 0","");
36
37
           int i;
38
           for(i = 0; i < 256; i++){
39
                   updateLFSR(&lfsr state);
40
                   //fprintf(stdout, "Saida[%d]:%02X\n",i,lfsr state);
41
42
                   AFFECT(inttostr(tempo), "lfsr state", inttostr(lfsr state));
43
                   tempo++;
44
45
           SAV GENPAT();
           return 0;
46
47 }
```

## Código VHDL

```
1 library IEEE;
2 use IEEE.STD LOGIC 1164.ALL;
3 use IEEE.STD LOGIC ARITH.ALL;
4 use IEEE.STD LOGIC UNSIGNED.ALL;
6 entity LFSR is
          Port ( clk : in STD LOGIC;
                 rst : in STD LOGIC;
                 lfsr state : in STD LOGIC VECTOR(7 downto 0) := "01111111";
                 lfsr out : out STD LOGIC VECTOR(7 downto 0);
                 vdd : in STD LOGIC;
                 vss : in STD LOGIC );
13 end LFSR;
16 architecture Behavioral of LFSR is
          signal XOR result : STD LOGIC;
          signal clk borda : bit;
19 begin
          process (clk, rst)
          begin
                  if rising edge(clk) then
                          clk borda <= NOT clk borda; -- detecta a borda
                          if clk borda = '1' then
                                   XOR result <= lfsr state(7) xor lfsr state(5) xor lfsr state(4) xor lfsr state(3);</pre>
                                   lfsr out <= XOR result & lfsr state(7 downto 1);</pre>
                           end if:
                  end if;
          end process;
30 end Behavioral;
```



## Otimizações e Asimut

```
50% area - 50% delay optimization
Reading file 'gerador_t.vst'...
Reading lib '/usr/alliance/cells/sxlib'...
Capacitances on file 'gerador_t.vst'...
Delays on file 'gerador_t.vst'...914 ps
Area on file 'gerador_t.vst'...74000 lamda (with over-cell routing)
       sff2_x4: 9 (72%)
       buf_x2: 8 (10%)
       xr2_x1: 3 (9%)
       sff1_x4: 1 (6%)
Worst RC on file 'gerador_t.vst'...274 ps
Inserting buffers on critical path for file 'gerator_to.vst'...None inserted
Improving RC on critical path for file 'gerator_to.vst'...910 ps
Improving all RC for file 'gerator_to.vst'...
Worst RC on file 'gerator_to.vst'...66 ps
Area on file 'gerator_to.vst'...75000 lamda (with over-cell routing)
Details...
       sff2_x4: 9 (72%)
       buf_x2: 8 (10%)
       xr2_x1: 3 (9%)
       sff1_x4: 1 (6%)
Critical path (no warranty)...910 ps from 'rtlalc_0 0' to 'lfsr_out 0'
Saving file 'gerator_to.vst'...
Saving critical path in xsch color file 'gerator_to.xsc'...
```

```
agic@duza-pies:~$ asimut -b gerador_5o lfsr_sim result
                    0000 0 0
                   9 99 999
                                                           0 00 0
                  @@
                                                              0.0
                                   000 00 000 000 0000
                           0000
                                   000 00 00 00 00
                                   00 00 00
                      000 00 00 00 00 00
                        00 00 00 00 00 00
              00 00 00 00 00 00 00 00
             0000 0 0000 000000 0000 000 000 000
                              A SIMUlation Tool
                Alliance CAD System 5.0,
                Copyright (c) 1991...1999-2023, ASIM/LIP6/UPMC
                           : alliance-users@asim.lip6.fr
       Paris, France, Europe, Earth, Solar system, Milky Way, ...
initializing ...
searching `gerador_5o` ...
BEH : Compiling `gerador_5o.vbe` (Behaviour) ...
making GEX ...
searching pattern file : `lfsr_sim` ...
                                              ###---- processing pattern 207 : 207 ps ----###
                                              ###---- processing pattern 208 : 208 ps ----###
                                              ###---- processing pattern 209 : 209 ps ----###
linking ...
                                              ###---- processing pattern 210 : 210 ps ----###
executing ...
                                               ###---- processing pattern 211 : 211 ps ----###
###---- processing pattern 0 : 0 ps ----### ###---- processing pattern 212 : 212 ps ----###
###---- processing pattern 1 : 1 ps ----### ###---- processing pattern 213 : 213 ps ----###
###---- processing pattern 2 : 2 ps ----### ###---- processing pattern 214 : 214 ps ----###
###---- processing pattern 3 : 3 ps ----###
                                              ###---- processing pattern 215 : 215 ps ----###
###---- processing pattern 4 : 4 ps ----###
                                              ###---- processing pattern 216 : 216 ps ----###
###---- processing pattern 5 : 5 ps ----###
                                              ###---- processing pattern 217 : 217 ps ----###
###---- processing pattern 6 : 6 ps ----###
                                              ###---- processing pattern 218 : 218 ps ----###
###---- processing pattern 7 : 7 ps ----###
                                              ###---- processing pattern 219 : 219 ps ----###
###---- processing pattern 8 : 8 ps ----###
                                              ###---- processing pattern 220 : 220 ps ----###
###---- processing pattern 9 : 9 ps ----### ###---- processing pattern 221 : 221 ps ----###
###---- processing pattern 10 : 10 ps ----### ###---- processing pattern 222 : 222 ps ----###
###---- processing pattern 11 : 11 ps ----### ###---- processing pattern 223 : 223 ps ----###
###---- processing pattern 12 : 12 ps ----### ###---- processing pattern 224 : 224 ps ----###
###---- processing pattern 13 : 13 ps ----### ###---- processing pattern 225 : 225 ps ----###
###---- processing pattern 14 : 14 ps ----### ###---- processing pattern 226 : 226 ps ----###
###---- processing pattern 15 : 15 ps ----### ###---- processing pattern 227 : 227 ps -----###
                                              ###---- processing pattern 228 : 228 ps ----###
###---- processing pattern 16 : 16 ps ----###
                                               ###---- processing pattern 229 : 229 ps ----###
                                               ###---- processing pattern 230 : 230 ps ----###
                                               ###---- processing pattern 231 : 231 ps ----###
                                               ###---- processing pattern 232 : 232 ps ----###
                                               ###---- processing pattern 233 : 233 ps ----###
                                               ###---- processing pattern 234 : 234 ps ----###
                                               ###---- processing pattern 235 : 235 ps ----###
                                               ###---- processing pattern 236 : 236 ps ----###
                                               ###---- processing pattern 237 : 237 ps ----###
                                               ###---- processing pattern 238 : 238 ps ----###
                                               ###---- processing pattern 239 : 239 ps ----###
                                               ###---- processing pattern 240 : 240 ps ----###
                                               ###---- processing pattern 241 : 241 ps ----###
                                               ###---- processing pattern 242 : 242 ps ----###
                                               ###---- processing pattern 243 : 243 ps ----###
                                               ###---- processing pattern 244 : 244 ps ----###
                                               ###---- processing pattern 245 : 245 ps ----###
                                               ###---- processing pattern 246 : 246 ps ----###
                                               ###---- processing pattern 247 : 247 ps ----###
                                               ###---- processing pattern 248 : 248 ps ----###
                                               ###---- processing pattern 249 : 249 ps ----###
                                               ###---- processing pattern 250 : 250 ps ----###
                                               ###---- processing pattern 251 : 251 ps ----###
                                               ###---- processing pattern 252 : 252 ps ----###
                                               ###---- processing pattern 253 : 253 ps ----###
                                               ###---- processing pattern 254 : 254 ps ----###
                                               ###---- processing pattern 255 : 255 ps ----###
                                               magic@duza-pies:~$
```