

PRÁCTICA 12 “MARQUESINA CON ROM”

Código VHDL

Contador de 10 bits (se ocupa el mismo en ambas GAL)

```
1  -- Santos Mèndez Ulises Jesùs
2  -- Pràctica 12 "Marquesina con ROM"
3
4  library ieee;
5  use ieee.std_logic_1164.all;
6  use ieee.std_logic_arith.all;
7  use ieee.std_logic_unsigned.all;
8
9  entity contador is
10 port (clk,en: in std_logic;
11        clr: inout std_logic;
12        q: inout std_logic_vector(9 downto 0));
13
14        attribute pin_numbers of contador: entity is
15            "clr:13 en:2 q(9):21 q(8):20 q(7):19 q(6):18 "
16 &      "q(5):17 q(4):16 q(3):15 q(2):14 q(1):23 q(0):22 ";
17 end contador;
18
19 architecture arq_conta of contador is
20 begin
21     process (clk,clr)
22     begin
23         if (clr = '1') then
24             q <= (others => '0');
```

```
25         elsif(rising_edge(clk)) then
26             if(en = '1') then
27                 q<= q+1;
28             else
29                 q <= q;
30             end if;
31         end if;
32     end process;
33 end arq_conta;
```

Segunda GAL para "DISEÑO"

```
1  -- Santos Mèndez Ulises `Jesùs
2  -- Pràctica 12 "Marquesina con ROM"
3
4  library ieee;
5  use ieee.std_logic_1164.all;
6  use ieee.std_logic_arith.all;
7  use ieee.std_logic_unsigned.all;
8
9  entity marq1 is
10 port(clk,clr: in std_logic;
11       dir: in std_logic_vector(2 downto 0);
12       cat: inout std_logic_vector(2 downto 0);
13       display: out std_logic_vector(6 downto 0));
14
15     attribute pin_numbers of marq1: entity is
16         "dir(2):4  dir(1):5  dir(0):6  clr:13 "
17 &     "cat(2):22  cat(1):23  cat(0):14 "
18 &     "display(6):21  display(5):20  display(4):19 "
19 &     "display(3):18  display(2):17  display(1):16  display(0):15 ";
20
21 end marq1;
22
23 architecture arq_mar of marq1 is
24 --/////////Declaracion de salidas de memoria/////////
```

```
25 constant d: std_logic_vector(6 downto 0) := "0111101"; --3D
26 constant i: std_logic_vector(6 downto 0) := "0110000"; --30
27 constant s: std_logic_vector(6 downto 0) := "1011011"; --5B
28 constant e: std_logic_vector(6 downto 0) := "1001111"; --4F
29 constant n: std_logic_vector(6 downto 0) := "1010101"; --55
30 constant o: std_logic_vector(6 downto 0) := "1111110"; --7E
31 constant nu: std_logic_vector(6 downto 0) := "0000001"; --01
32 --////////Salidas de ROM0,ROM1 Y ROM2////////
33 signal disp0: std_logic_vector(6 downto 0);
34 signal disp1: std_logic_vector(6 downto 0);
35 signal disp2: std_logic_vector(6 downto 0);
36 --////////Declaracion de memorias////////
37 type memoria is array (0 to 7) of std_logic_vector(6 downto 0);
38 constant ROM0: memoria := (d,i,s,e,n,o,nu,nu);
39 constant ROM1: memoria := (nu,d,i,s,e,n,o,nu);
40 constant ROM2: memoria := (nu,nu,d,i,s,e,n,o);
41 begin
42
43 -- //////////CONTADOR DE ANILLO////////
44     process(clk,clr)
45     begin
46         if(clr='1') then
47             cat <= "001";
48         elsif(rising_edge(clk)) then
```

```
49         cat <= to_stdlogicvector(to_bitvector(cat)
50         rol 1);
51     end if;
52 end process;
53
54 --////////OPERACIONES DE LECTURA DE LA ROM////////
55     process(dir)
56     begin
57         disp0 <= ROM0(conv_integer(dir));
58         disp1 <= ROM1(conv_integer(dir));
59         disp2 <= ROM2(conv_integer(dir));
60     end process;
61
62 --////////MUX->DISPLAY////////
63     process(cat,disp0,disp1,disp2)
64     begin
65         if(cat="001") then
66             display <= disp0;
67         elsif(cat="010") then
68             display <= disp1;
69         else
70             display <= disp2;
71         end if;
72     end process;
73
74 -
75 end arq_mar;
```

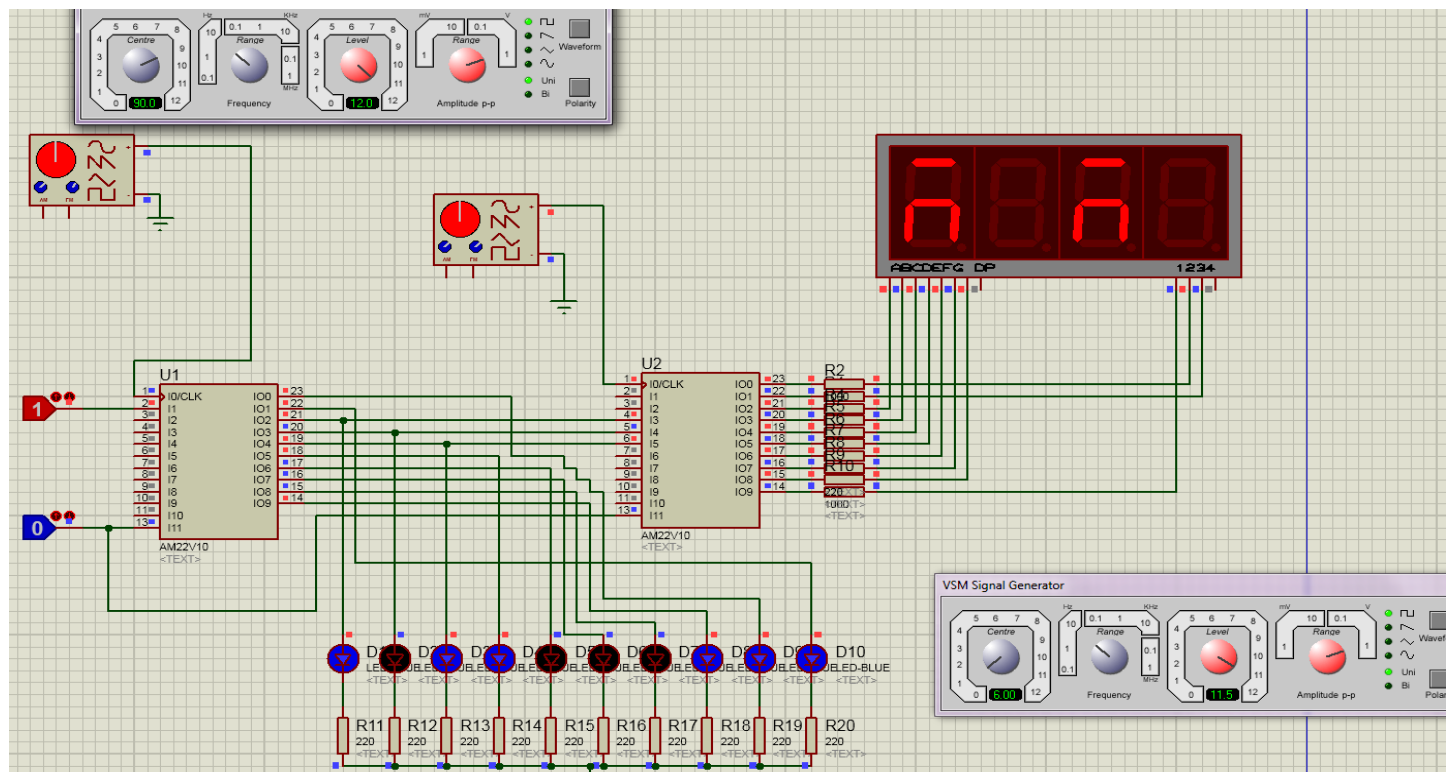
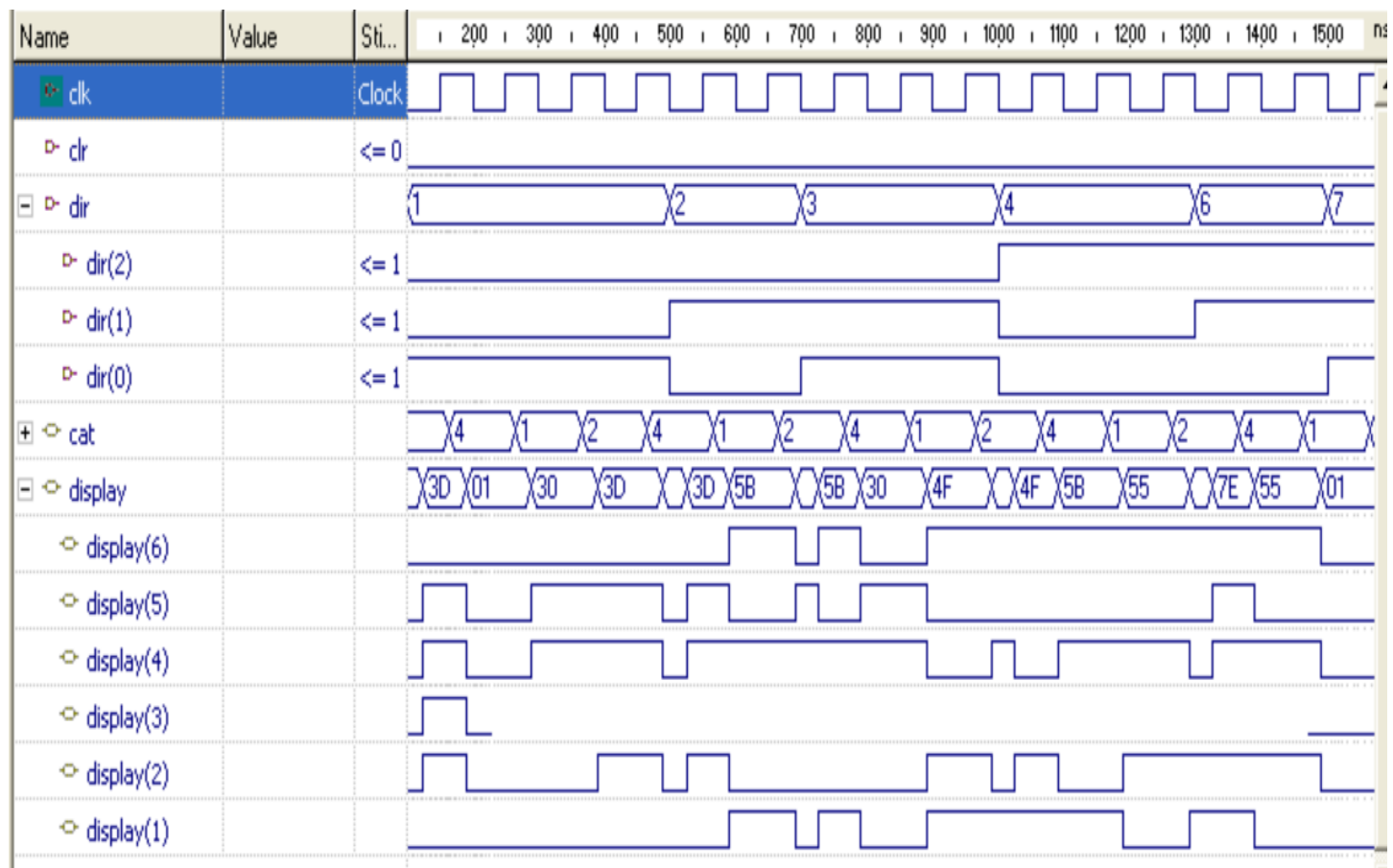
Segunda GAL "DISEÑO DIGITAL"

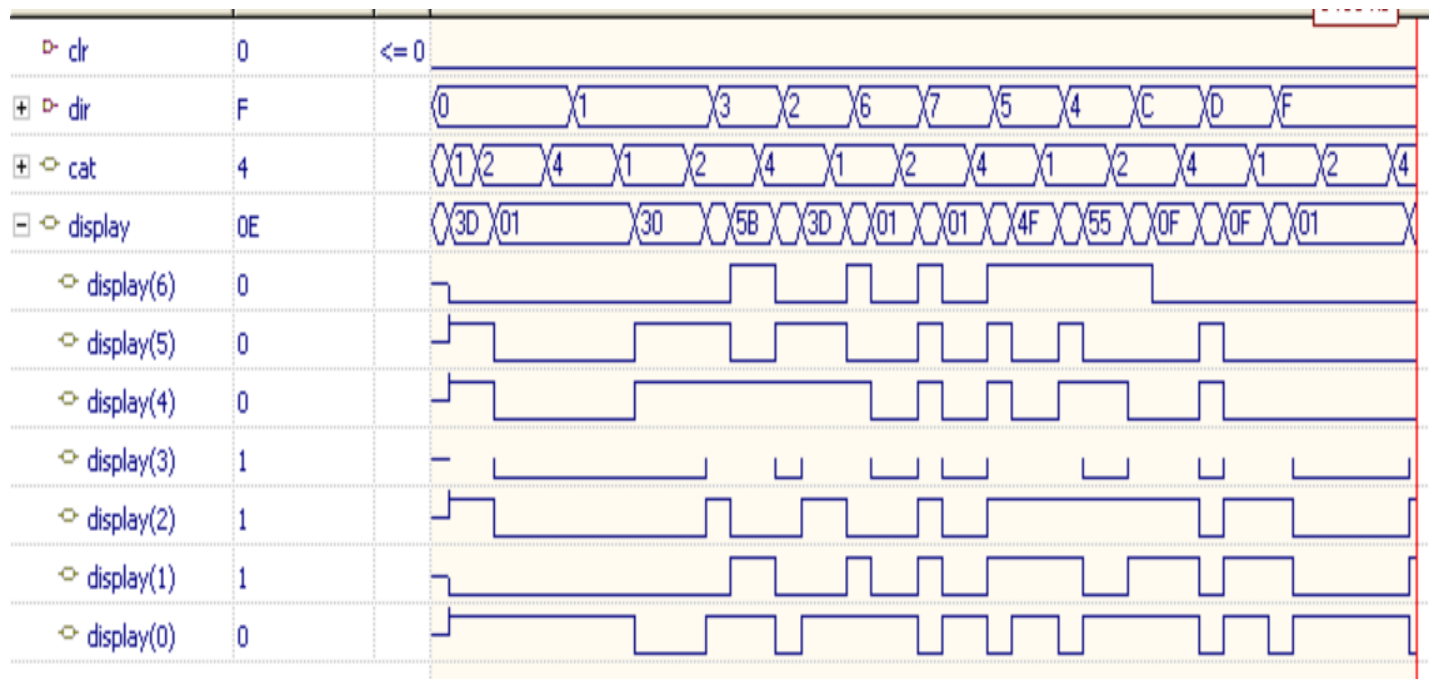
```
1  -- Santos Méndez Ulises `Jesús
2  -- Práctica 12 "Marquesina con ROM"
3
4  library ieee;
5  use ieee.std_logic_1164.all;
6  use ieee.std_logic_arith.all;
7  use ieee.std_logic_unsigned.all;
8
9  entity marq2 is
10 port(clk,clr: in std_logic;
11       dir: in std_logic_vector(3 downto 0);
12       cat: inout std_logic_vector(2 downto 0);
13       display: out std_logic_vector(6 downto 0));
14 end marq2;
15
16 architecture arq_mar2 of marq2 is
17 --////////Declaracion de salidas de memoria como etiquetas////////
18 constant cod_nu: std_logic_vector(3 downto 0) := "0000";
19 constant cod_d: std_logic_vector(3 downto 0) := "0001";
20 constant cod_i: std_logic_vector(3 downto 0) := "0011";
21 constant cod_s: std_logic_vector(3 downto 0) := "0010";
22 constant cod_e: std_logic_vector(3 downto 0) := "0110";
23 constant cod_n: std_logic_vector(3 downto 0) := "0111";
24 constant cod_o: std_logic_vector(3 downto 0) := "0101";
25 constant cod_g: std_logic_vector(3 downto 0) := "0100";
26 constant cod_t: std_logic_vector(3 downto 0) := "1100";
27 constant cod_a: std_logic_vector(3 downto 0) := "1101";
28 constant cod_l: std_logic_vector(3 downto 0) := "1111";
29 constant cod_x: std_logic_vector(3 downto 0) := "1110";
30
31 --////////Salidas de ROM0,ROM1 Y ROM2////////
32 signal disp0: std_logic_vector(3 downto 0);
33 signal disp1: std_logic_vector(3 downto 0);
34 signal disp2: std_logic_vector(3 downto 0);
35 signal smem: std_logic_vector(3 downto 0);
36 --////////Declaracion de memorias////////
```

```
37 type memoria is array (0 to 15) of std_logic_vector(3 downto 0);
38 constant ROM0: memoria := (cod_d,cod_i,cod_s,cod_e,cod_n,cod_o,
39                             cod_nu,cod_d,cod_i,cod_g,cod_i,cod_t,
40                             cod_a,cod_l,cod_nu,cod_nu);
41
42 constant ROM1: memoria := (cod_nu,cod_d,cod_i,cod_s,cod_e,cod_n,cod_o,
43                             cod_nu,cod_d,cod_i,cod_g,cod_i,cod_t,
44                             cod_a,cod_l,cod_nu);
45
46 constant ROM2: memoria := (cod_nu,cod_nu,cod_d,cod_i,cod_s,cod_e,cod_n,cc
47                             cod_nu,cod_d,cod_i,cod_g,cod_i,cod_t,
48                             cod_a,cod_l);
49 begin
50
51 -- //////////CONTADOR DE ANILLO////////
52     process(clk,clr)
53     begin
54         if(clr='1') then
55             cat <= "001";
56         elsif(rising_edge(clk)) then
57             cat <= to_stdlogicvector(to_bitvector(cat)
58                                     rol 1);
59         end if;
60     end process;
61
62 --//////////OPERACIONES DE LECTURA DE LA ROM//////////
63     process(dir)
64     begin
65         disp0 <= ROM0(conv_integer(dir));
66         disp1 <= ROM1(conv_integer(dir));
67         disp2 <= ROM2(conv_integer(dir));
68     end process;
69
70 --//////////MUX->DISPLAY//////////
71     process(cat,disp0,disp1,disp2)
72     begin
73         if(cat="001") then
74             smem <= disp0;
75         elsif(cat="010") then
76             smem <= disp1;
77         elsif(cat="100") then
78             smem <= disp2;
79         else
80             smem <= cod_x;
81         end if;
82     end process;
83
84 -- //////////Convertidor de código//////////
```

```
85     process (smem)
86     begin
87         case smem is
88             when "0000" => display <= "0000001";
89             when "0001" => display <= "0111101";
90             when "0011" => display <= "0110000";
91             when "0010" => display <= "1011011";
92             when "0110" => display <= "1001111";
93             when "0111" => display <= "1010101";
94             when "0101" => display <= "1111110";
95             when "0100" => display <= "1111011";
96             when "1100" => display <= "0001111";
97             when "1101" => display <= "1110111";
98             when "1111" => display <= "0001110";
99             when others => display <= "-----";
100         end case;
101     end process;
102 end arq_mar2;
```

Simulación en Active HDL-Sim y en Proteus





Questionario

1. ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica?
R= 4 dispositivos PLD 22V10
2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica?
R=10 FF's 4013, 2 555, 7 7408, 7 7432, 5 7404, 1 ROM
3. ¿Cuántos pines de entrada/salida del PLD 22V10 se usan en el diseño?
R= La primera GAL ocupa 3 entradas y 10 salidas, la segunda GAL ocupa 5 entradas y 10 salidas
4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?
R=En la segunda GAL se ocupan 70 términos producto y un 68% de las macroceldas
5. ¿Qué puedes concluir de esta práctica?
R= En conclusión las memorias ROM son de mucha utilidad si lo que quieres es tener una gran carga de datos para poder utilizarla con eficiencia así como la declaración de arreglos y distintas sentencias en alto nivel