

PRÁCTICA 11 “CARTA ASM”

Código VHDL

1. Código primera GAL (carta ASM)

```
1 -- Ulises Jesùs Santos Mèndez
2 -- 2CV8
3 -- Practica 11 "Cartas ASM"
4 library ieee;
5 use ieee.std_logic_1164.all;
6
7 entity GAL1 is
8 port (clk,clr,ini: in std_logic;
9       D: in std_logic_vector(5 downto 0);
10      EC,EB,LB: out std_logic;
11      A: inout std_logic_vector(5 downto 0));
12
13     attribute pin_numbers of GAL1: entity is
14       "clr:13 ini:8 D(0):7 D(1):6 D(2):5 D(3):4 "
15 &    "D(4):3 D(5):2 EC:22 EB:21 LB:20 A(5):19 "
16 &    "A(4):18 A(3):17 A(2):16 A(1):15 A(0):14 ";
17
18 end GAL1;
19
20 architecture CARTA of GAL1 is
21 type estados is (e0,e1,e2);
22 signal edo_act,edo_sig: estados;
23 signal LA,EA,z: std_logic;
24 begin
```

```
25 -- Proceso de reseteo y de estado siguiente
26   process(clk,clr)
27   begin
28       if(clr='1')then
29           edo_act <= e0;
30       elsif(rising_edge(clk))then
31           edo_act <= edo_sig;
32       end if;
33   end process;
34 -- Parametrizacion de los estados
35   process(edo_act,ini,z,A)
36   begin
37 -- Señales de salida en 0
38       LA <= '0';
39       EA <= '0';
40       LB <= '0';
41       EB <= '0';
42       EC <= '0';
43       z <= not (A(5) or A(4) or A(3) or A(2) or A(1) or A(0));
44       case edo_act is
45           when e0 =>
46               LB <= '1';
47               if(ini='1')then
48                   edo_sig <= e1;
49
50               else
51                   LA <= '1';
52                   edo_sig <= e0;
53               end if;
54           when e1 =>
55               EA <= '1';
56               if(z='0' and A(0)='1')then
57                   EB<='1';
58                   edo_sig <= e1;
59               elsif(z='0' and A(0)='0')then
60                   edo_sig <= e1;
61               else
62                   edo_sig <= e2;
63               end if;
64           when others =>
65               EC <= '1';
66               if(ini='1')then
67                   edo_sig <= e2;
68               else
69                   edo_sig <= e0;
70               end if;
71       end case;
72   end process;
```

```
73     process(clk,clr)
74     begin
75         if(clr='1')then
76             A <= (others => '0');
77         elsif(rising_edge(clk))then
78             if(EA='0' and LA='1')then
79                 A <= D;
80             elsif(EA='1' and LA='0')then
81                 A(4 downto 0) <= A(5 downto 1);
82                 A(5) <= '0';
83             else
84                 A <= A;
85             end if;
86         end if;
87     end process;
88
89 end CARTA;
```

2. Código de segunda GAL

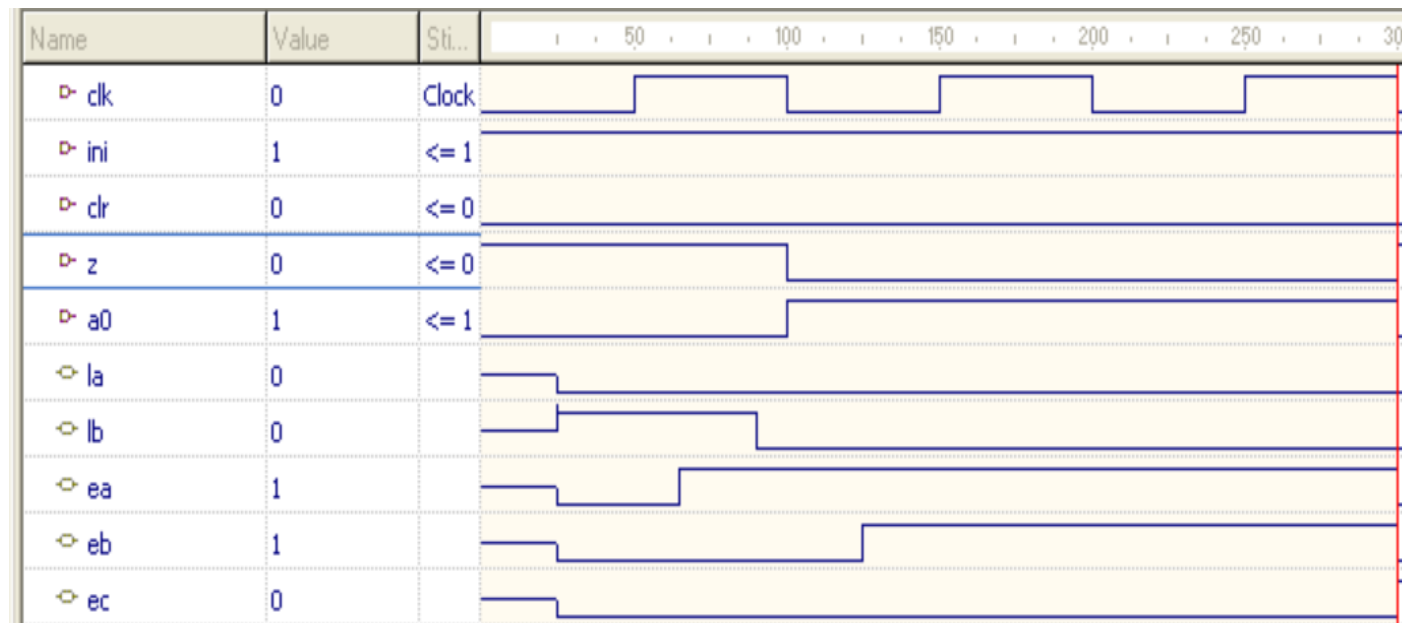
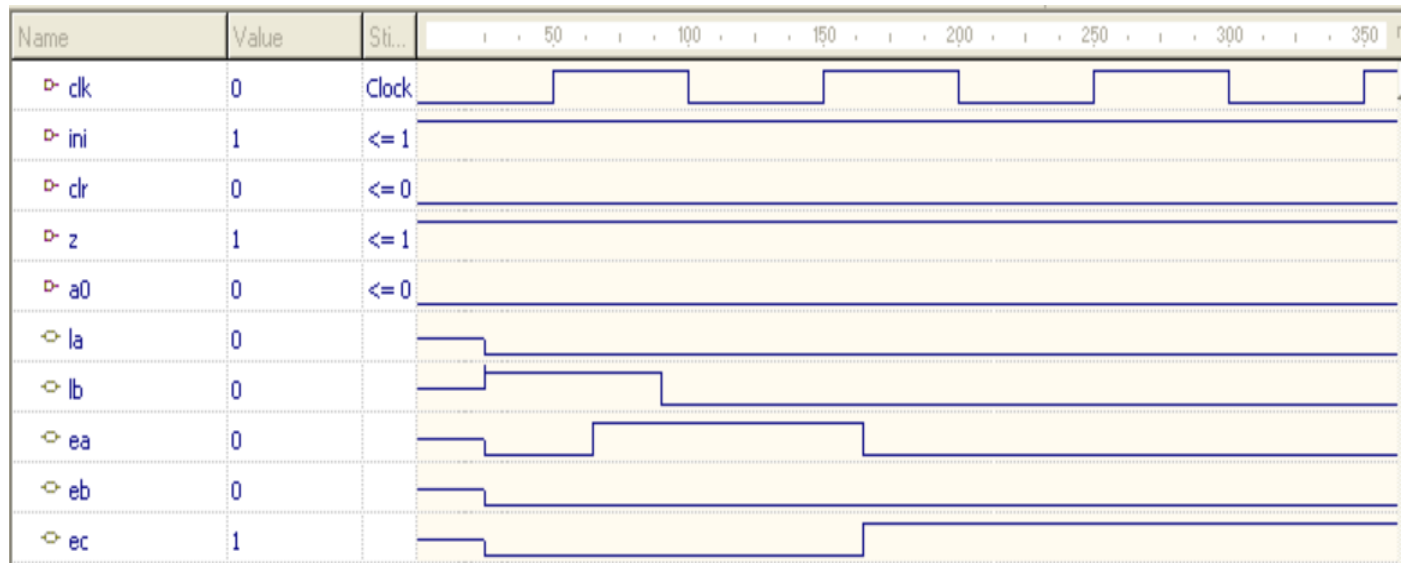
```
1  -- Ulises Jesús Santos Méndez
2  -- 2CV8
3  -- Practica 11 "Cartas ASM"
4
5  library ieee;
6  use ieee.std_logic_1164.all;
7
8  entity GAL2 is
9  port(clk,clr,LB,EB,EC: in std_logic;
10       disp: out std_logic_vector(6 downto 0));
11
12       attribute pin_numbers of GAL2: entity is
13         "clr:13 LB:5 EB:4 EC:3 disp(6):21 disp(5):20 "
14 &     "disp(4):19 disp(3):18 disp(2):17 disp(1):16 "
15 &     "disp(0):15 ";
16
17 end GAL2;
18
19 architecture arq_cont of GAL2 is
20 -- Señal de salida del contador
21 signal qb: std_logic_vector(2 downto 0);
22 -- Representacion de 0-6
23 constant e0: std_logic_vector(2 downto 0):="000";
24 constant e1: std_logic_vector(2 downto 0):="001";
```

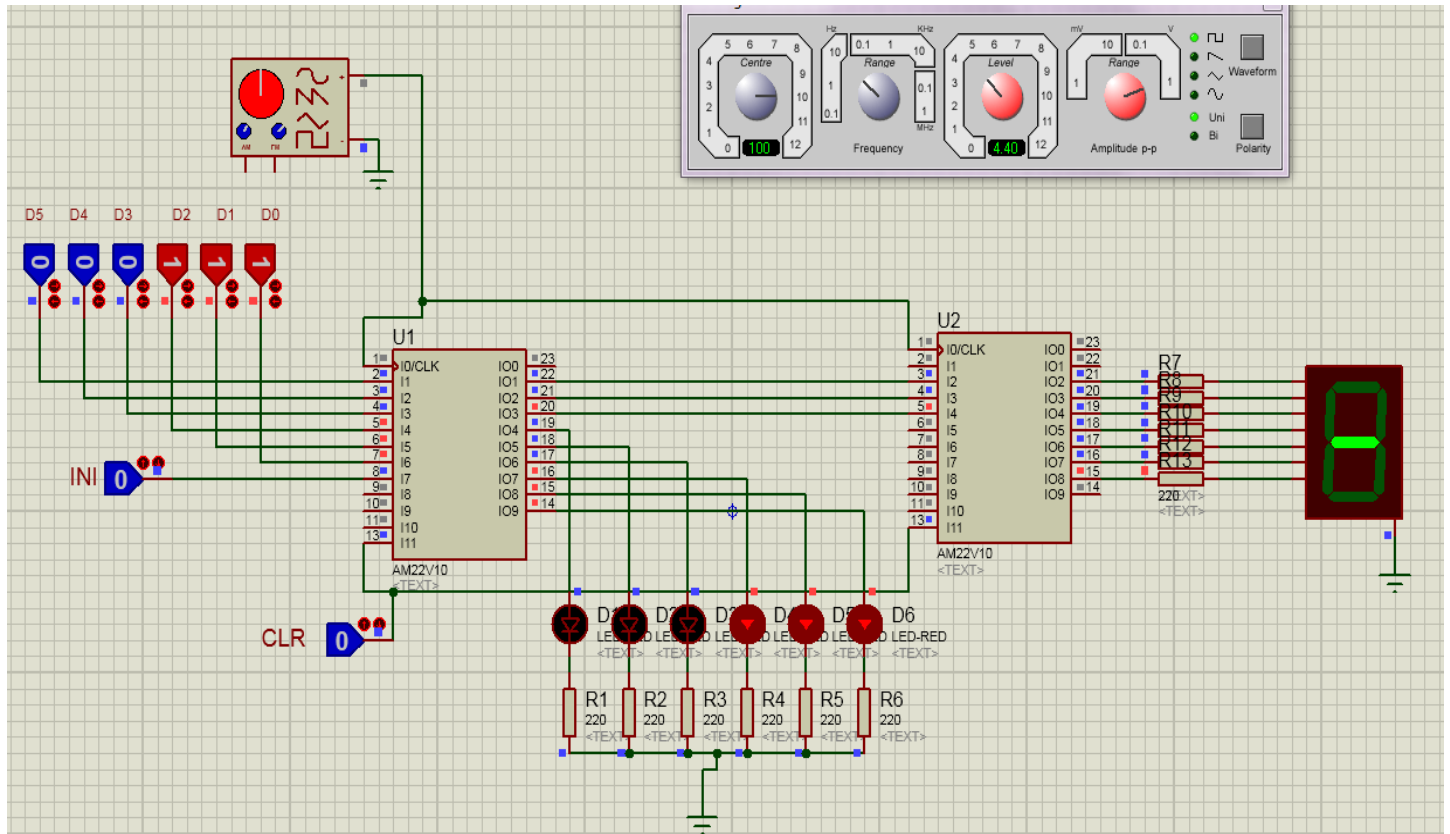
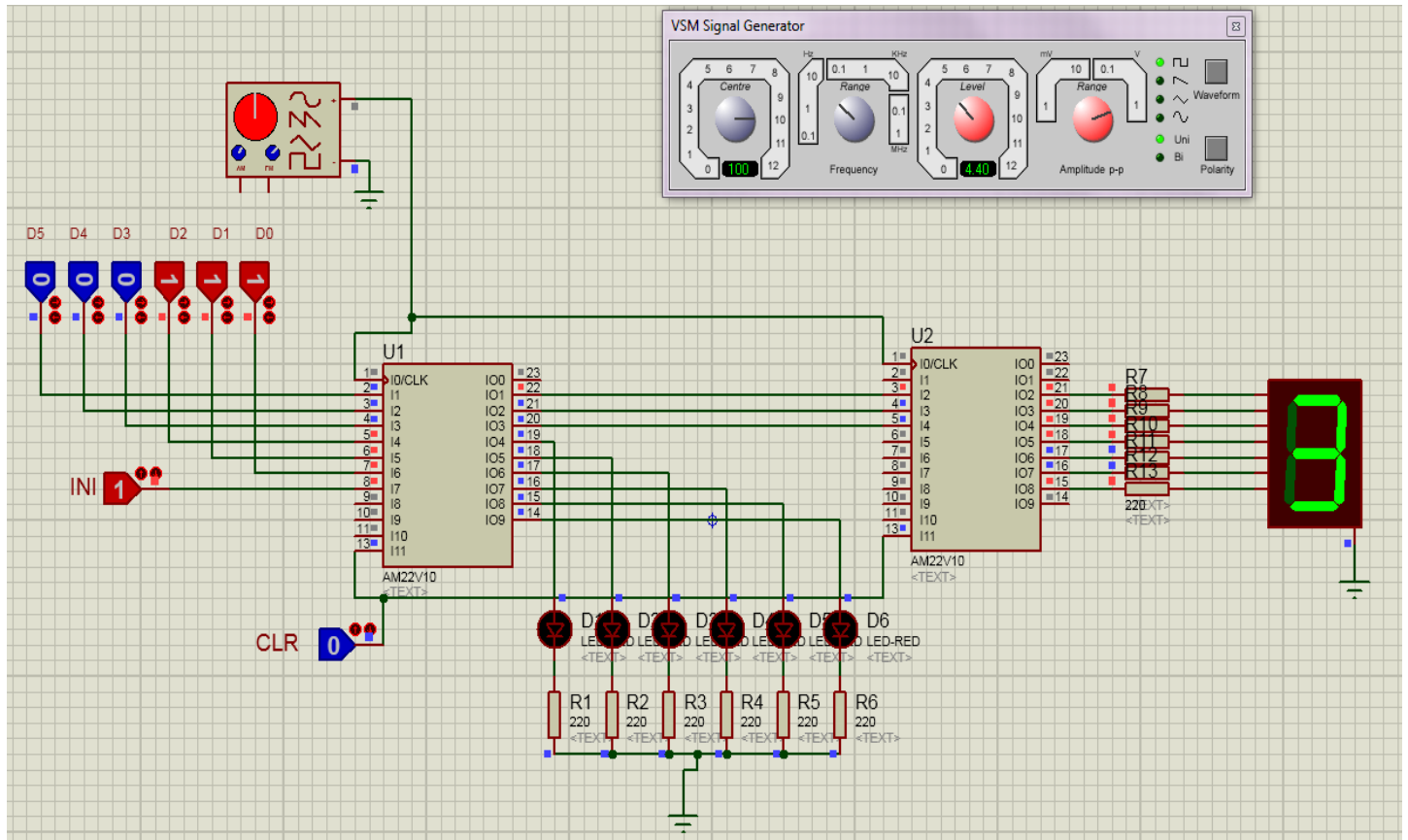
```
25 constant e2: std_logic_vector(2 downto 0) := "010";
26 constant e3: std_logic_vector(2 downto 0) := "011";
27 constant e4: std_logic_vector(2 downto 0) := "100";
28 constant e5: std_logic_vector(2 downto 0) := "101";
29 constant e6: std_logic_vector(2 downto 0) := "110";
30
31 begin
32 -- ////////////CONTADOR//////////
33     process(clk,clr)
34     begin
35         if(clr='1') then
36             qb <= e0;
37         elsif(rising_edge(clk)) then
38             if(LB='1' and EB='0') then
39                 qb <= e0;
40             elsif(LB='0' and EB='1') then
41                 case qb is
42                     when e0 => qb <= e1;
43                     when e1 => qb <= e2;
44                     when e2 => qb <= e3;
45                     when e3 => qb <= e4;
46                     when e4 => qb <= e5;
47                     when others => qb <= e6;
48                 end case;
49
50             elsif (LB='0' and EB='0') then
51                 qb <= qb;
52             else
53                 qb <= "---";
54             end if;
55         end if;
56     end process;
57 -- ////////////DECODIFICADOR Y MUX//////////
58     process(clk,EC,qb)
59     begin
60         if(EC='0') then
61             disp <= "0000001";
62         else
63             case qb is
64                 when e0 => disp <= "1111110";
65                 when e1 => disp <= "0110000";
66                 when e2 => disp <= "1101101";
67                 when e3 => disp <= "1111001";
68                 when e4 => disp <= "0110011";
69                 when e5 => disp <= "1011011";
70                 when others => disp <= "1011111";
71             end case;
72         end if;
```



```
73     end process;  
74 end arq_cont;
```

Simulación en Active HDL-Sim y en Proteus





Questionario

1. ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica?
R= 2 dispositivos PLD 22V10
2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica?
R= 8 4013, 1 555, 6 7408, 6 7432, 5 7404, 1 74ls149.
3. ¿Cuántos pines de entrada/salida de cada PLD 22V10 que se usan en el diseño?
R= en la primera GAL se tienen 9 entradas y 9 salidas, en la segunda GAL se tienen 5 entradas y 7 salidas.
4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total de los PLD 22V10?
R= en la primer GAL se ocupan 34 términos producto y un 86% del PLD, de la segunda GAL se ocupan 22 términos producto y se ocupa un 68% del PLD.
5. ¿Cuántos FF's ocupa el autómata de control de la microarquitectura?
R= 3 Flip Flops
6. ¿Qué puedes concluir de esta práctica?
R= En conclusión se logró observar la facilidad y diferencia en el manejo de máquinas de estados y la comprensión del diagrama de flujo para facilitar los procesos para tener un diseño en alto nivel.