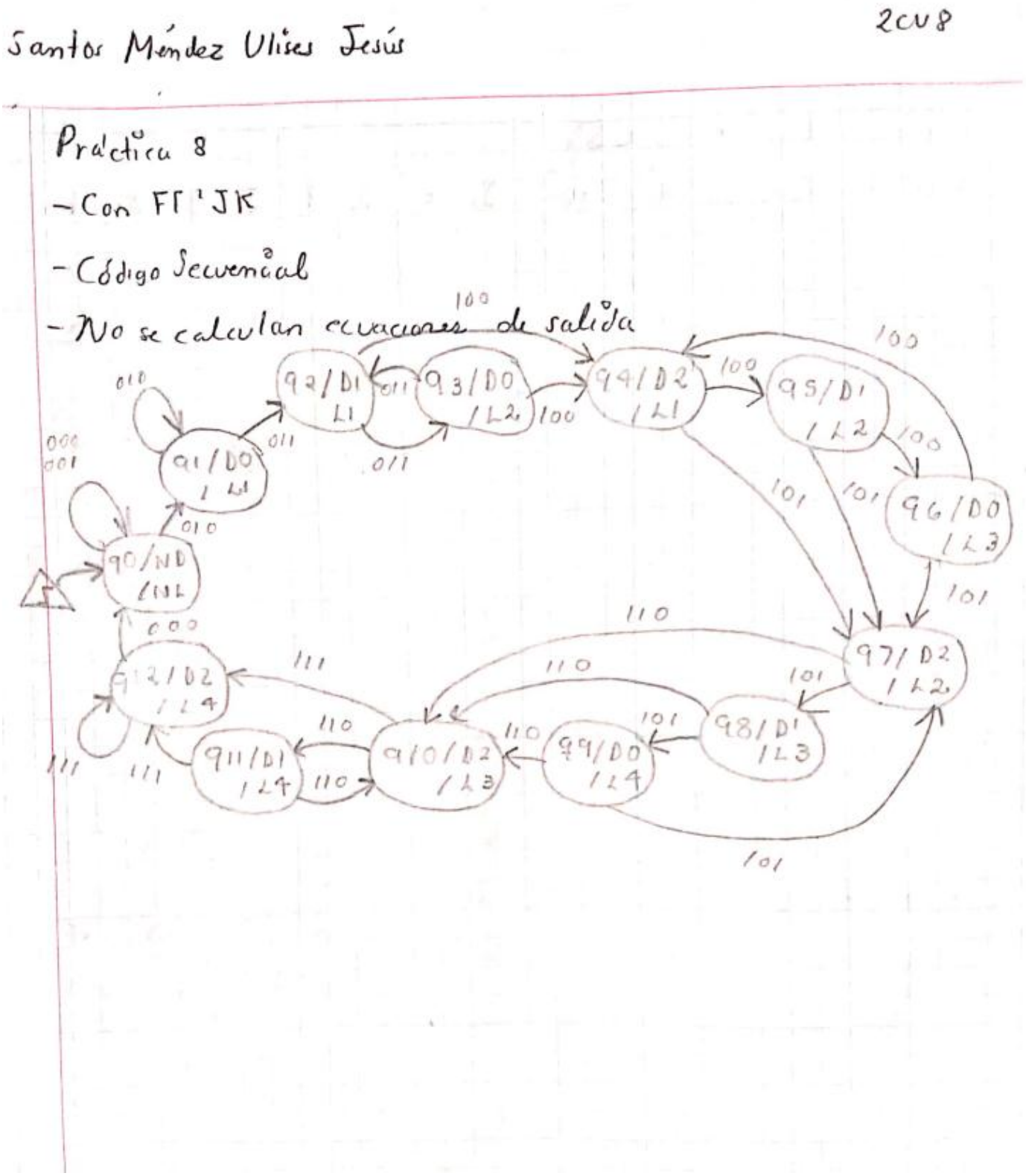
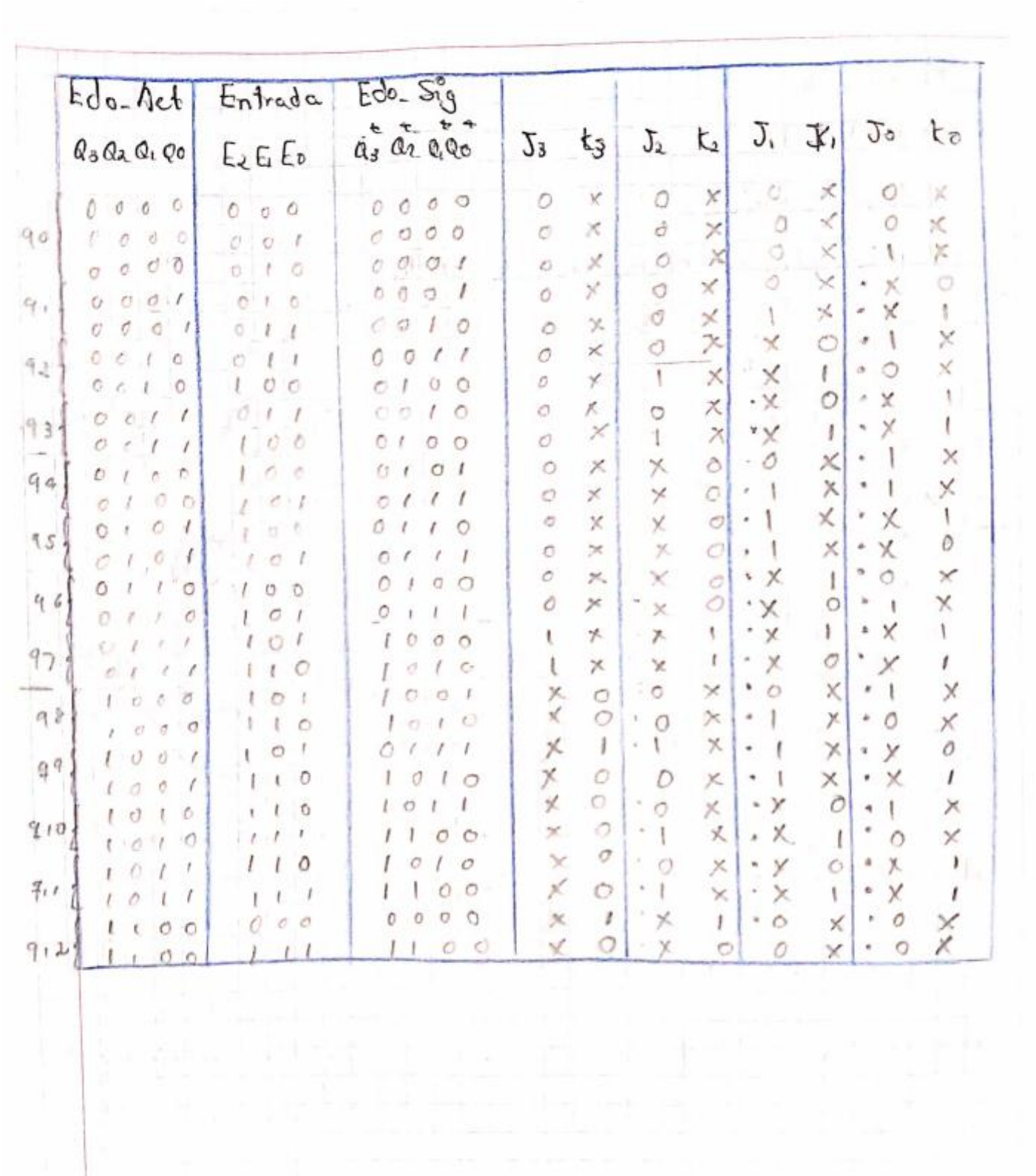
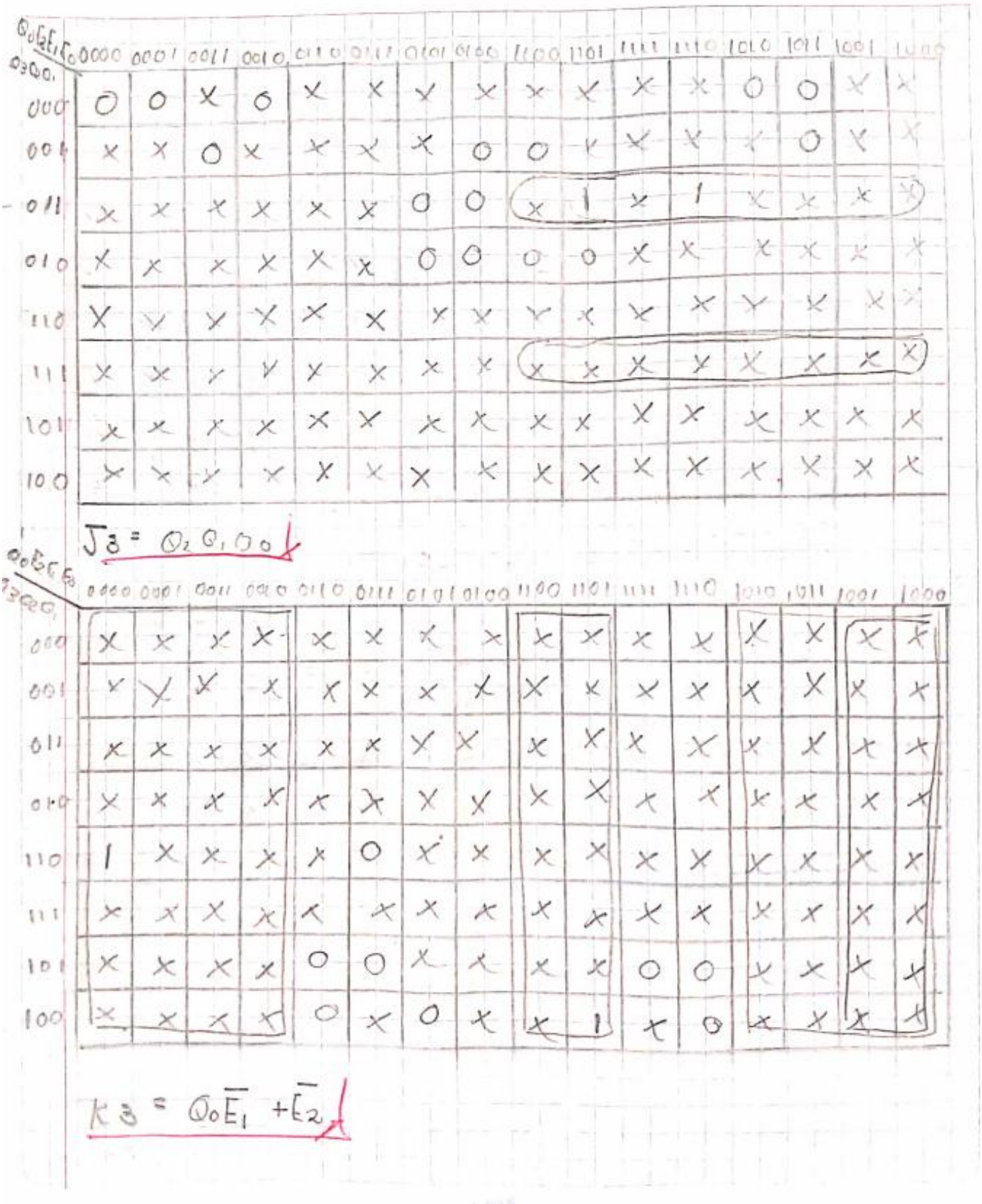
Análisis del circuito con FF’s



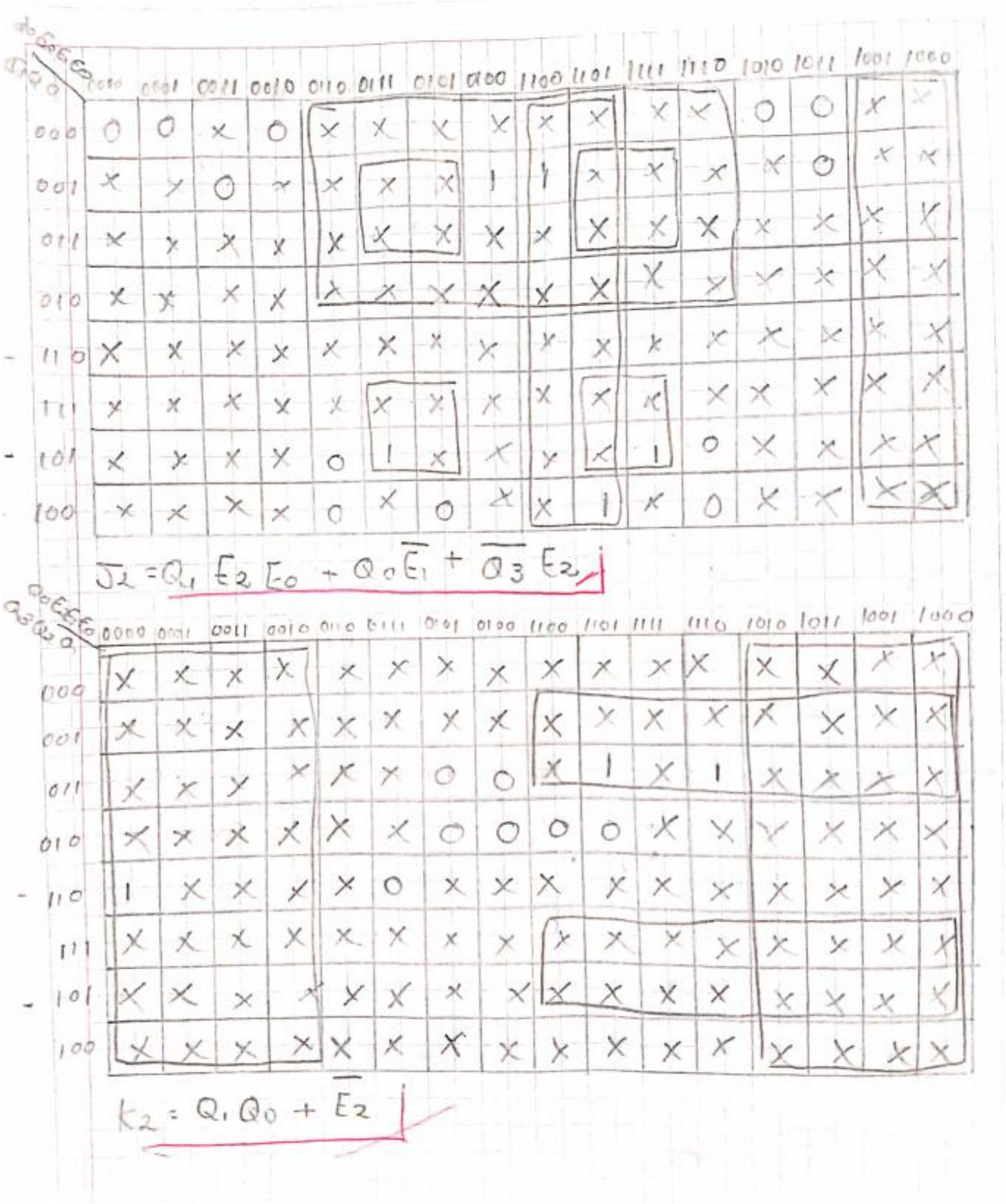


	Edo_Act $Q_3 Q_2 Q_1 Q_0$	Entrada $E_2 E_1 E_0$	Edo_Sig $Q_3^+ Q_2^+ Q_1^+ Q_0^+$	J_3	k_3	J_2	k_2	J_1	K_1	J_0	k_0
90	0 0 0 0	0 0 0	0 0 0 0	0	x	0	x	0	x	0	x
	1 0 0 0	0 0 1	0 0 0 0	0	x	0	x	0	x	0	x
	0 0 0 0	0 1 0	0 0 0 1	0	x	0	x	0	x	1	x
91	0 0 0 1	0 1 0	0 0 0 1	0	x	0	x	0	x	x	0
	0 0 0 1	0 1 1	0 0 1 0	0	x	0	x	1	x	x	1
92	0 0 1 0	0 1 1	0 0 1 1	0	x	0	x	x	0	1	x
	0 0 1 0	1 0 0	0 1 0 0	0	x	1	x	x	1	0	x
93	0 0 1 1	0 1 1	0 0 1 0	0	x	0	x	x	0	x	1
	0 0 1 1	1 0 0	0 1 0 0	0	x	1	x	x	1	x	1
94	0 1 0 0	1 0 0	0 1 0 1	0	x	x	0	0	x	1	x
	0 1 0 0	1 0 1	0 1 1 1	0	x	x	0	1	x	1	x
95	0 1 0 1	1 0 1	0 1 1 0	0	x	x	0	1	x	x	1
	0 1 0 1	1 0 1	0 1 1 1	0	x	x	0	1	x	x	0
96	0 1 1 0	1 0 0	0 1 0 0	0	x	x	0	x	1	0	x
	0 1 1 0	1 0 1	0 1 1 1	0	x	x	0	x	0	1	x
97	0 1 1 1	1 0 1	1 0 0 0	1	x	x	1	x	1	x	1
	0 1 1 1	1 1 0	1 0 1 0	1	x	x	1	x	0	x	1
98	1 0 0 0	1 0 1	1 0 0 1	x	0	0	x	0	x	1	x
	1 0 0 0	1 1 0	1 0 1 0	x	0	0	x	1	x	0	x
99	1 0 0 1	1 0 1	0 1 1 1	x	1	1	x	1	x	x	0
	1 0 0 1	1 1 0	1 0 1 0	x	0	0	x	1	x	x	1
910	1 0 1 0	1 1 0	1 0 1 1	x	0	0	x	x	0	1	x
	1 0 1 0	1 1 1	1 1 0 0	x	0	1	x	x	1	0	x
911	1 0 1 1	1 1 0	1 0 1 0	x	0	0	x	x	0	x	1
	1 0 1 1	1 1 1	1 1 0 0	x	0	1	x	x	1	x	1
912	1 1 0 0	0 0 0	0 0 0 0	x	1	x	1	0	x	0	x
	1 1 0 0	1 1 1	1 1 0 0	x	0	x	0	0	x	0	x



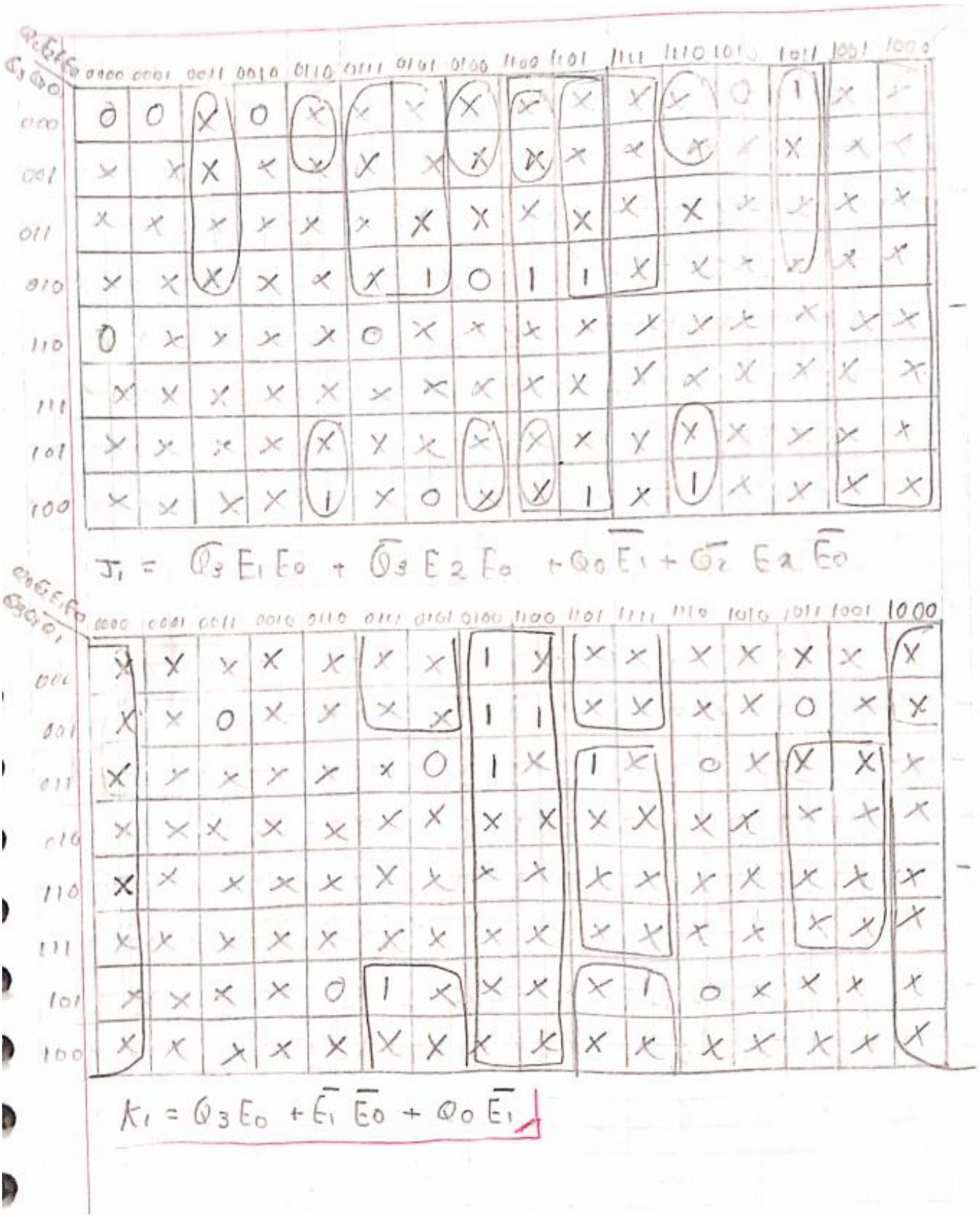
$J_3 = Q_2 Q_1 Q_0$

$K_3 = Q_0 \bar{E}_1 + \bar{E}_2$



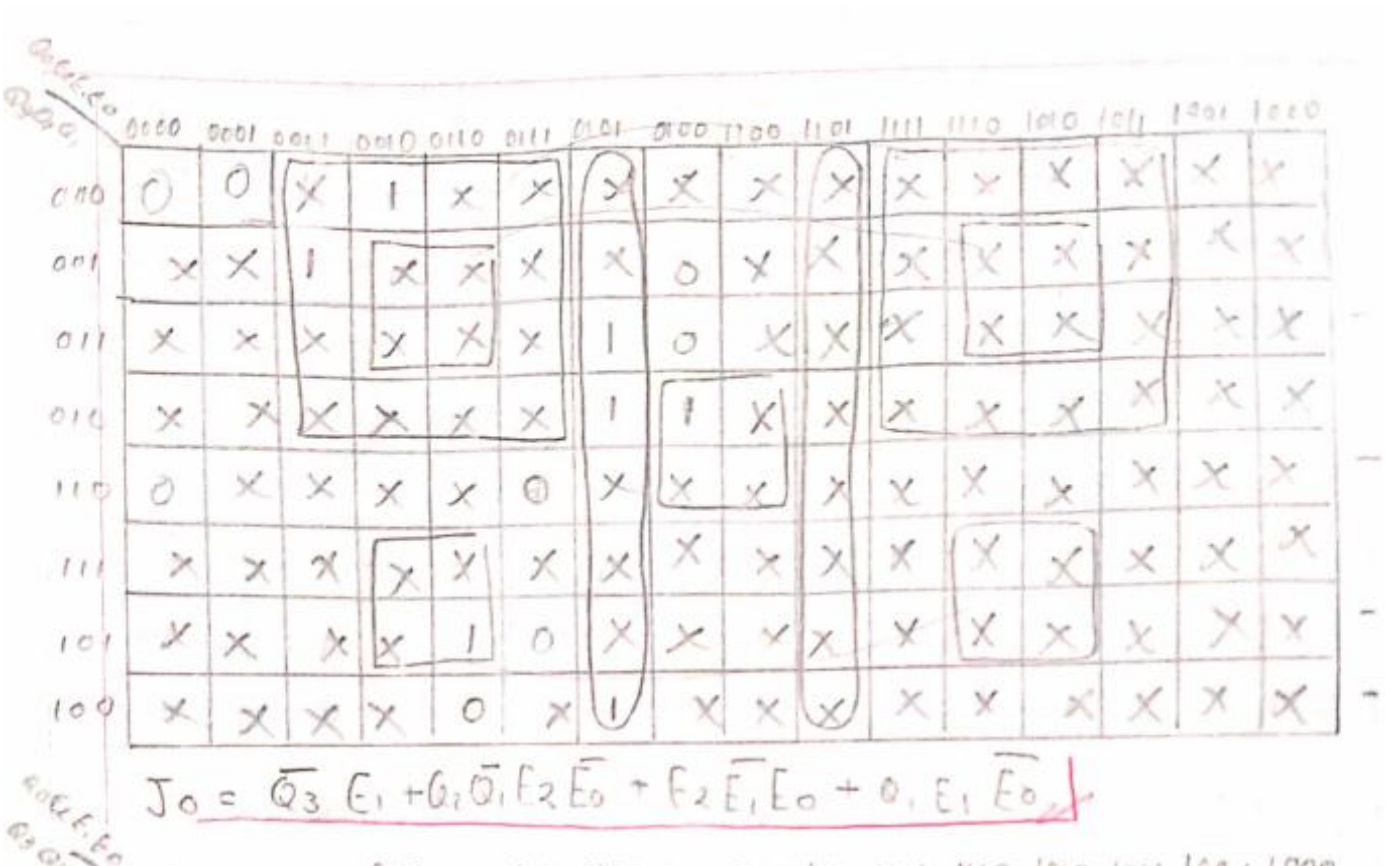
$J_2 = Q_1 E_2 E_0 + Q_0 \overline{E_1} + \overline{Q_3} E_2$

$K_2 = Q_1 Q_0 + \overline{E_2}$



$$J_1 = Q_3 E_1 E_0 + \bar{Q}_3 E_2 E_0 + Q_0 \bar{E}_1 + \bar{Q}_2 E_2 \bar{E}_0$$

$$K_1 = Q_3 E_0 + \bar{E}_1 \bar{E}_0 + Q_0 \bar{E}_1$$



$$J_0 = \bar{Q}_3 E_1 + Q_1 \bar{Q}_1 E_2 \bar{E}_0 + E_2 \bar{E}_1 E_0 + Q_1 E_1 \bar{E}_0$$

$$K_0 = Q_1 + E_2 E_1 \bar{E}_0$$

Código VHDL

```
 1 --Ulises Jesùs Santos Mèndez
 2 --2CV8
 3 --Pràctica 8 "Marquesina"
 4 library ieee;
 5 use ieee.std_logic_1164.all;
 6
 7 entity marq is
 8 port(clk,clr: in std_logic;
 9      e: in std_logic_vector(2 downto 0);
10      display: out std_logic_vector(9 downto 0));
11
12      attribute pin_numbers of marq: entity is
13      "clr:13 display(9):23 display(8):22 display(7):21 "
14 &    "display(6):14 display(5):19 display(4):18 display(3):17 "
15 &    "display(2):16 display(1):15 display(0):20 "
16 &    "e(2):3 e(1):4 e(0):5 ";
17 end marq;
18
19 architecture arqmarq of marq is
20 --NINGUN DISPLAY
21 constant ND: std_logic_vector(2 downto 0) := "111";
22 --DISPLAY DERECHO
23 constant D0: std_logic_vector(2 downto 0) := "110";
24 --DISPLAY IZQUIERDO
```

```
25 constant D1: std_logic_vector(2 downto 0) := "101";
26 --DISPLAY CENTRAL
27 constant D2: std_logic_vector(2 downto 0) := "011";
28 --NINGUNA LETRA
29 constant NL: std_logic_vector(6 downto 0) := "0000001";
30 --H
31 constant L1: std_logic_vector(6 downto 0) := "0110111";
32 --O
33 constant L2: std_logic_vector(6 downto 0) := "1111110";
34 --L
35 constant L3: std_logic_vector(6 downto 0) := "0001110";
36 --A
37 constant L4: std_logic_vector(6 downto 0) := "1110111";
38
39 constant q0:std_logic_vector(9 downto 0):= ND&NL;
40 constant q1:std_logic_vector(9 downto 0):= D0&L1;
41 constant q2:std_logic_vector(9 downto 0):= D1&L1;
42 constant q3:std_logic_vector(9 downto 0):= D0&L2;
43 constant q4:std_logic_vector(9 downto 0):= D2&L1;
44 constant q5:std_logic_vector(9 downto 0):= D1&L2;
45 constant q6:std_logic_vector(9 downto 0):= D0&L3;
46 constant q7:std_logic_vector(9 downto 0):= D2&L2;
47 constant q8:std_logic_vector(9 downto 0):= D1&L3;
48 constant q9:std_logic_vector(9 downto 0):= D0&L4;
```

```
49 constant q10:std_logic_vector(9 downto 0):= D2&L3;
50 constant q11:std_logic_vector(9 downto 0):= D1&L4;
51 constant q12:std_logic_vector(9 downto 0):= D2&L4;
52
53 signal aux: std_logic_vector(9 downto 0);
54
55 begin
56
57     process(clk,clr)
58     begin
59         if(clr = '1')then
60             aux <= q0;
61         elsif(rising_edge(clk))then
62             case aux is
63                 when q0 =>
64                     if(e="000" or e="001")then
65                         aux <= q0;
66                     elsif(e="010")then
67                         aux <= q1;
68                     else
69                         aux <= "----------";
70                     end if;
71                 when q1 =>
72                     if(e="010")then
```

```
73                          aux <= q1;
74                      elsif(e="011")then
75                          aux <= q2;
76                      else
77                          aux <= "----------";
78                      end if;
79                  when q2 =>
80                      if(e="011")then
81                          aux <= q3;
82                      elsif(e="100")then
83                          aux <= q4;
84                      else
85                          aux <= "----------";
86                      end if;
87                  when q3 =>
88                      if(e="011")then
89                          aux <= q2;
90                      elsif(e="100")then
91                          aux <= q4;
92                      else
93                          aux <= "----------";
94                      end if;
95                  when q4 =>
96                      if(e="100")then
```

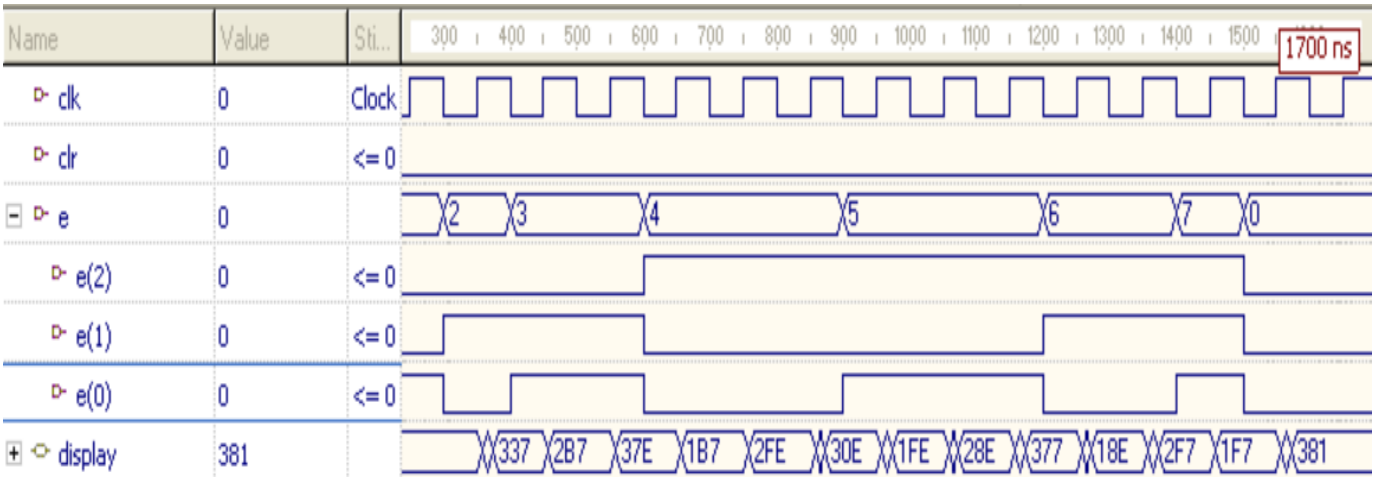
```
 97                          aux <= q5;
 98                      elsif(e="101")then
 99                          aux <= q7;
100                      else
101                          aux <= "----------";
102                      end if;
103                  when q5 =>
104                      if(e="100")then
105                          aux <= q6;
106                      elsif(e="101")then
107                          aux <= q7;
108                      else
109                          aux <= "----------";
110                      end if;
111                  when q6 =>
112                      if(e="100")then
113                          aux <= q4;
114                      elsif(e="101")then
115                          aux <= q7;
116                      else
117                          aux <= "----------";
118                      end if;
119                  when q7 =>
120                      if(e="101")then
```

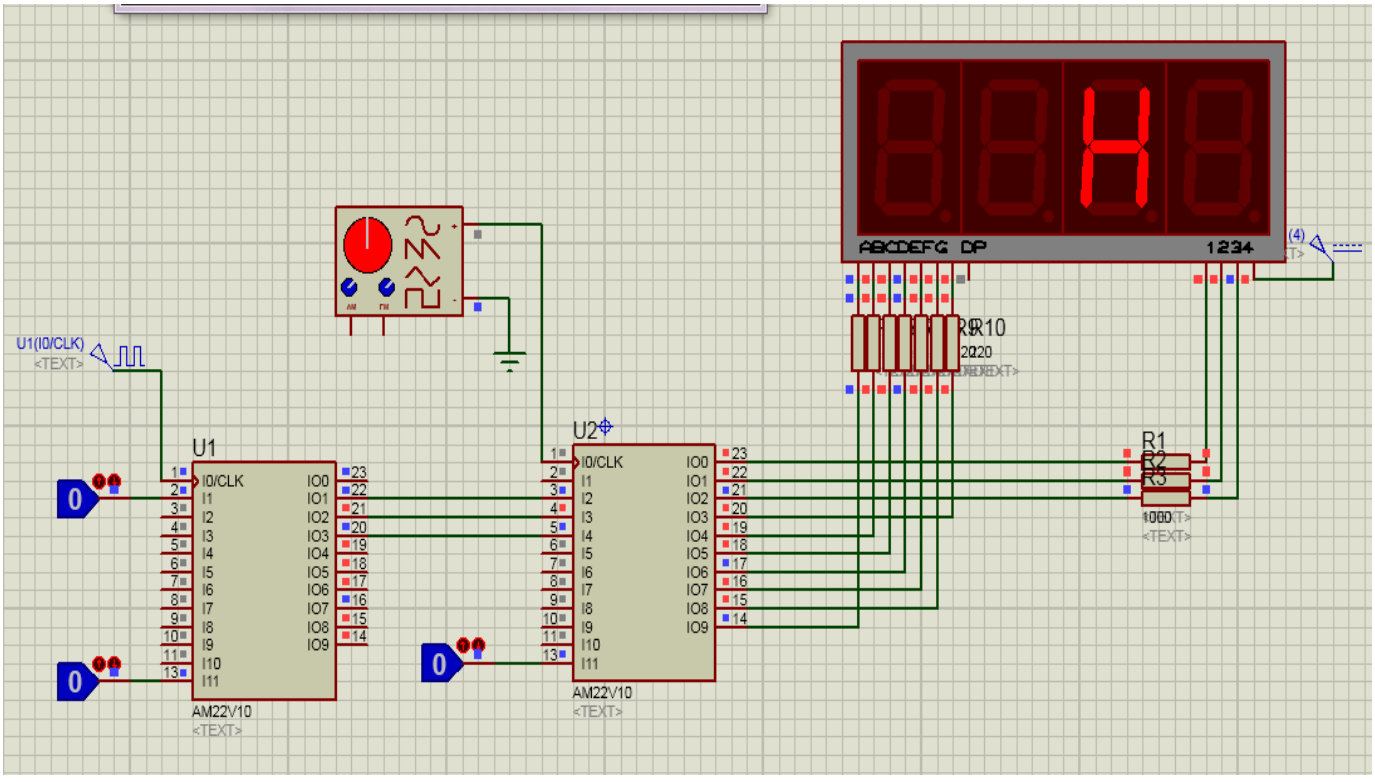
```
121                         aux <= q8;
122                     elsif(e="110")then
123                         aux <= q10;
124                     else
125                         aux <= "----------";
126                     end if;
127                 when q8 =>
128                     if(e="101")then
129                         aux <= q9;
130                     elsif(e="110")then
131                         aux <= q10;
132                     else
133                         aux <= "----------";
134                     end if;
135                 when q9 =>
136                     if(e="101")then
137                         aux <= q7;
138                     elsif(e="110")then
139                         aux <= q10;
140                     else
141                         aux <= "----------";
142                     end if;
143                 when q10 =>
144                     if(e="110")then
```

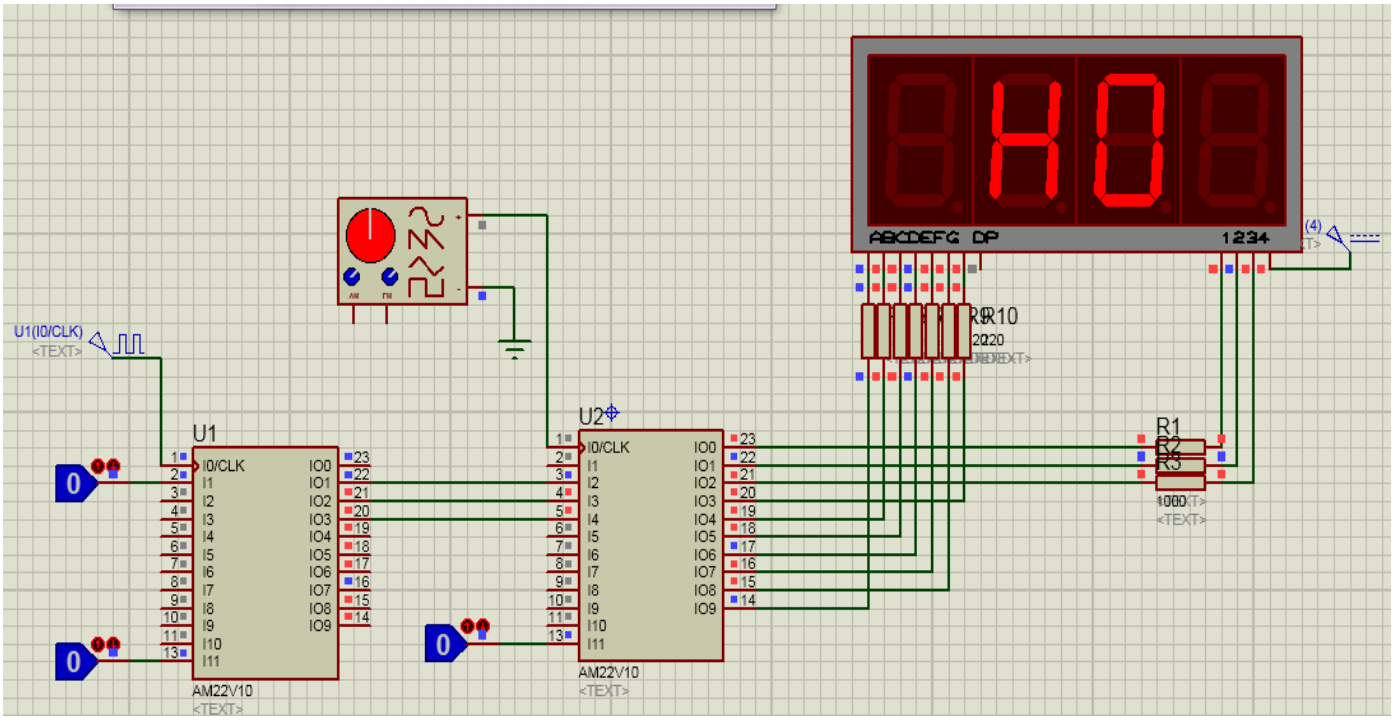
```
145                             aux <= q11;
146                         elsif(e="111")then
147                             aux <= q12;
148                         else
149                             aux <= "----------";
150                         end if;
151                     when q11 =>
152                         if(e="111")then
153                             aux <= q12;
154                         elsif(e="110")then
155                             aux <= q10;
156                         else
157                             aux <= "----------";
158                         end if;
159                     when q12 =>
160                         if(e="111")then
161                             aux <= q12;
162                         elsif(e="000")then
163                             aux <= q0;
164                         else
165                             aux <= "----------";
166                         end if;
167                     when others =>
168                         aux <= "----------";
```

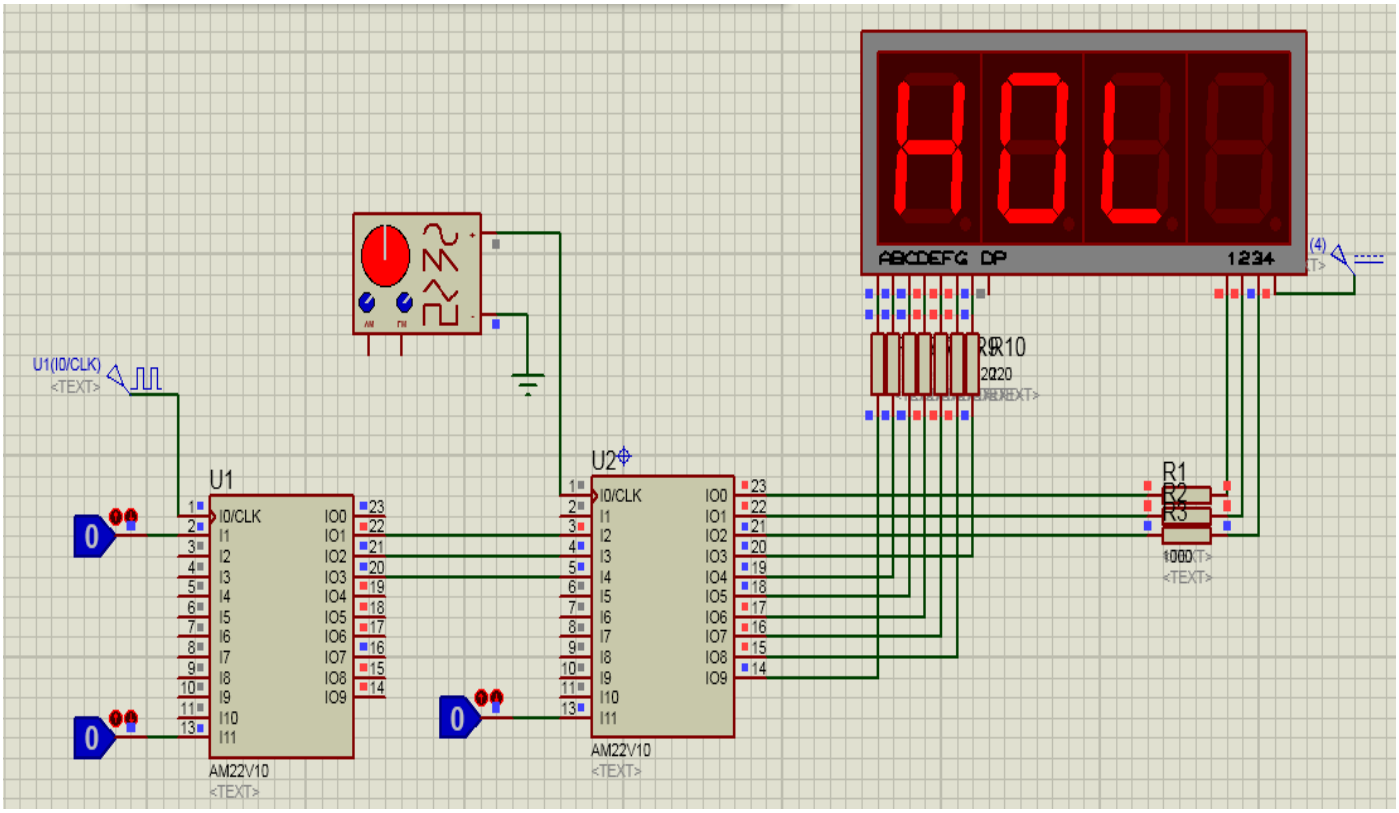
```
169             end case;
170         end if;
171 end process;
172 display <= aux;
173 end arqmarq;
```

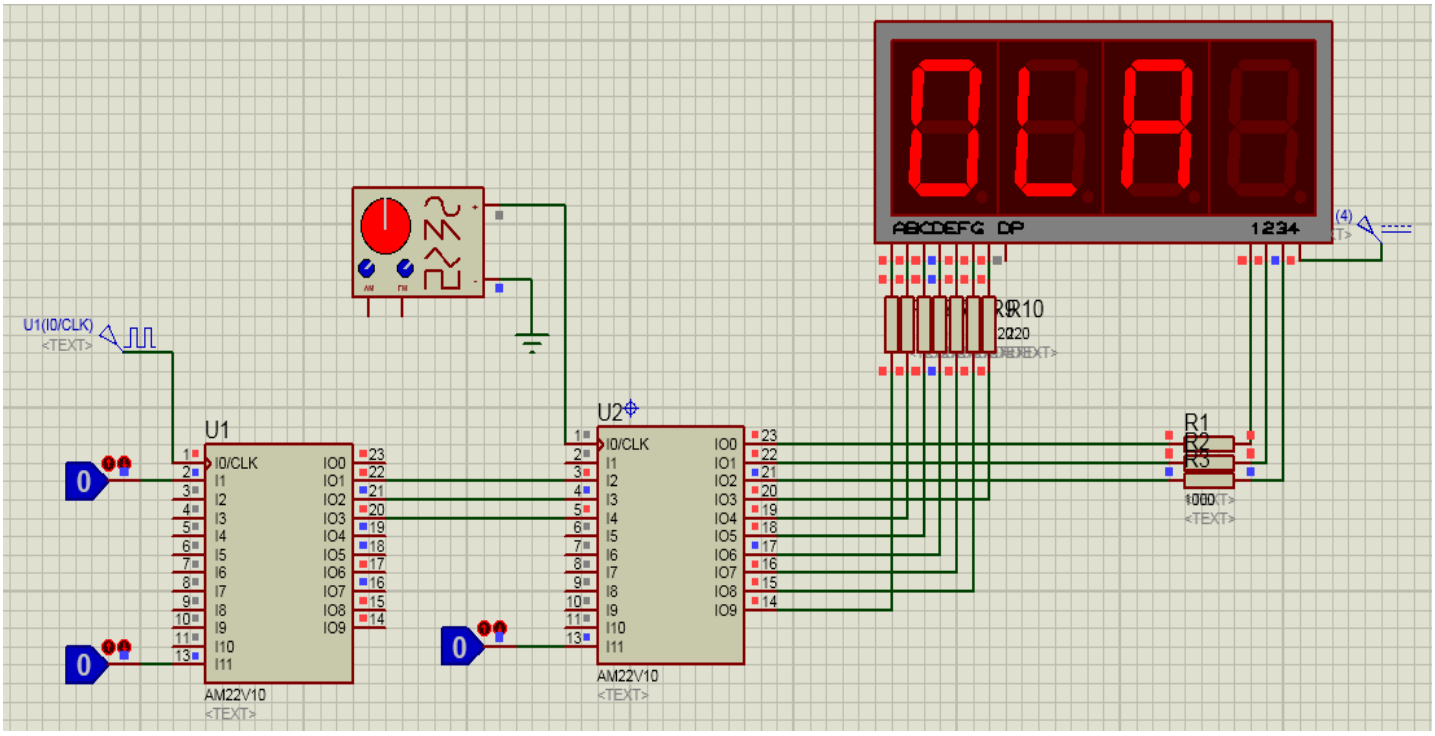
Simulación en Proteus y en Active HDL-Sim

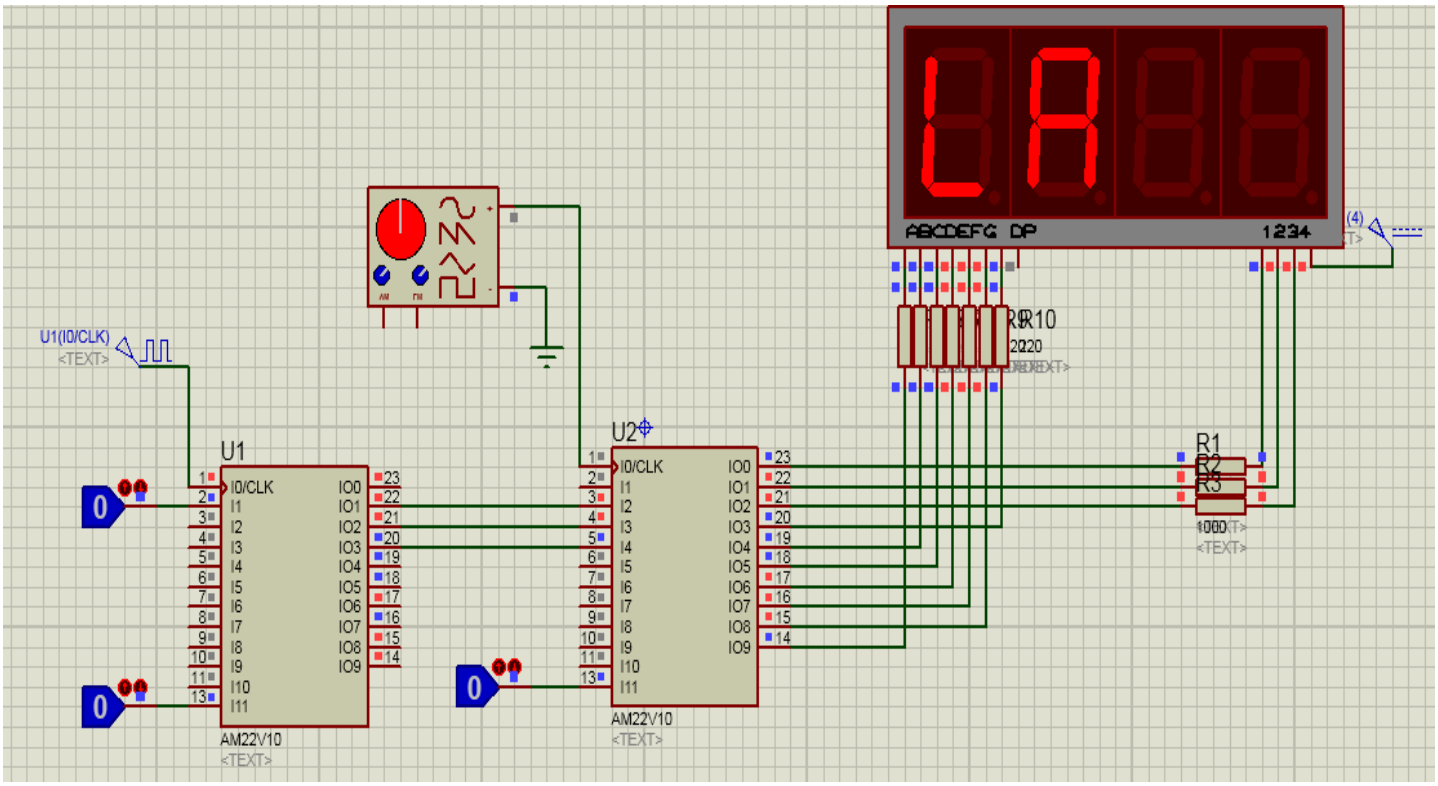


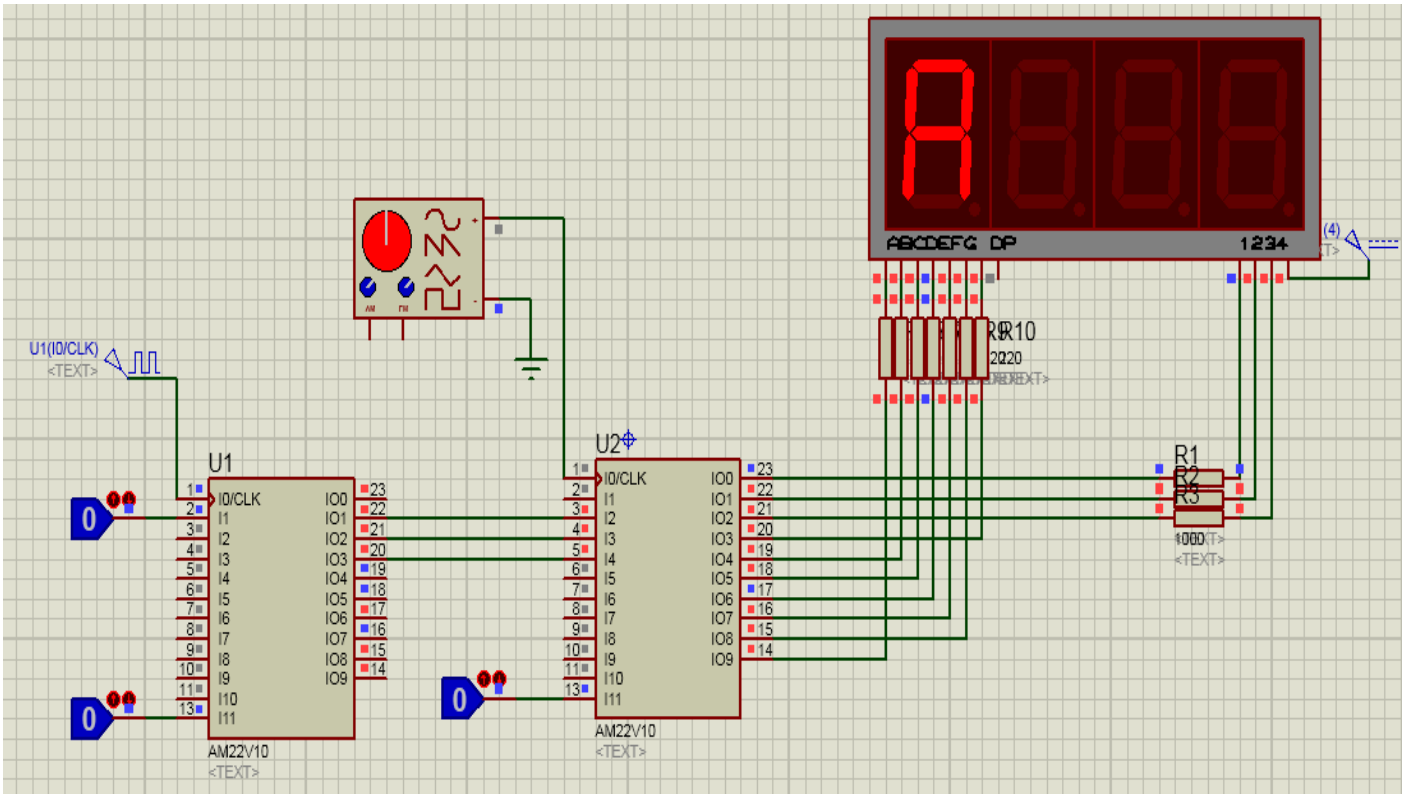


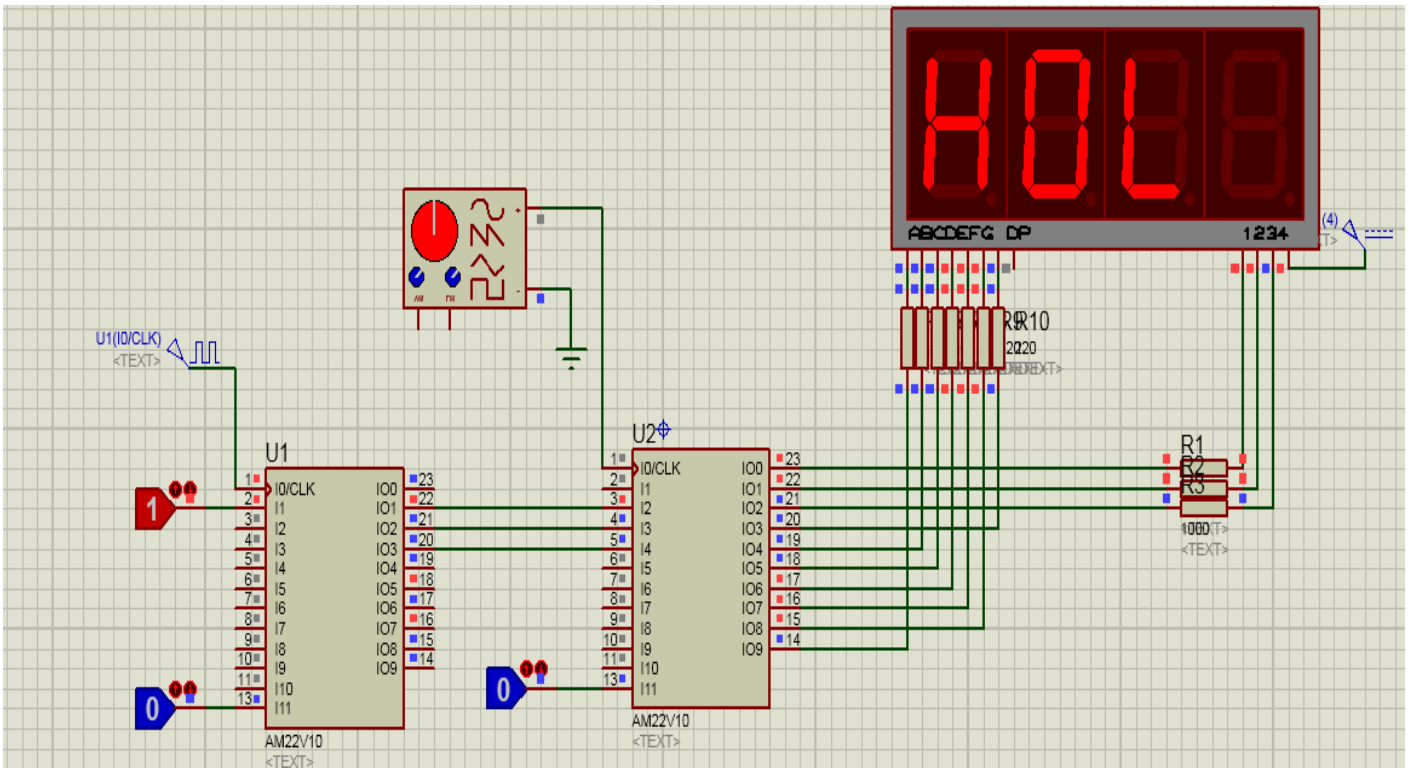

H0
ABCDEFG DP
1234
U1(I0/CLK)
<TEXT>
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
<TEXT>
U2
AM22V10
<TEXT>
R1
R2
R3
1000
<TEXT>
RR10
2020
<TEXT>
(4)


HOL0
ABCDEFG DP
1234
U1(I0/CLK)
<TEXT>
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
<TEXT>
U2
AM22V10
<TEXT>
R1
R2
R3
1000
<TEXT>
RR10
2020
<TEXT>
(4)


0LA0
ABCDEFG DP
1234
U1(I0/CLK)
<TEXT>
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
<TEXT>
U2
R10
R1
R2
R3
1000
0


LA00
ABCDEFG DP
1234
U1(I0/CLK)
<TEXT>
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
<TEXT>
U2
R10
R1
R2
R3
1000
0


U1(I0/CLK)
<TEXT>
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
<TEXT>
U2
AM22V10
<TEXT>
ABCDEFG DP
1234
R1
R2
R3
100
<TEXT>


U1(I0/CLK)
<TEXT>
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
<TEXT>
U2
AM22V10
<TEXT>
ABCDEFG DP
1234
R1
R2
R3
100
<TEXT>

CUESTIONARIO

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=7 FF's 4013, 2 555, 7 7408. 7 7432, 5 7404.
3. ¿Cuántos pines de entrada/salida del PLD 22V10 se usan en el diseño?
   R= 5 pines de entrada y 10 pines de salida.
4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?
   R=44 términos producto, se ocupa un 68 %.
5. ¿Qué codificación se usa con la directiva TYPE?
   R= El poder definir constantes para concatenarlas.
6. ¿Cuál codificación es la que finalmente se pudo sintetizar?
   R= La definición de constantes en un ciclo condicional en la misma lista sensible.
7. ¿Qué puedes concluir de esta práctica?
   R=En conclusión te facilita la codificación y la reducción de salidas en el PLD el definir constantes así como un mejor aprovechamiento de entradas.