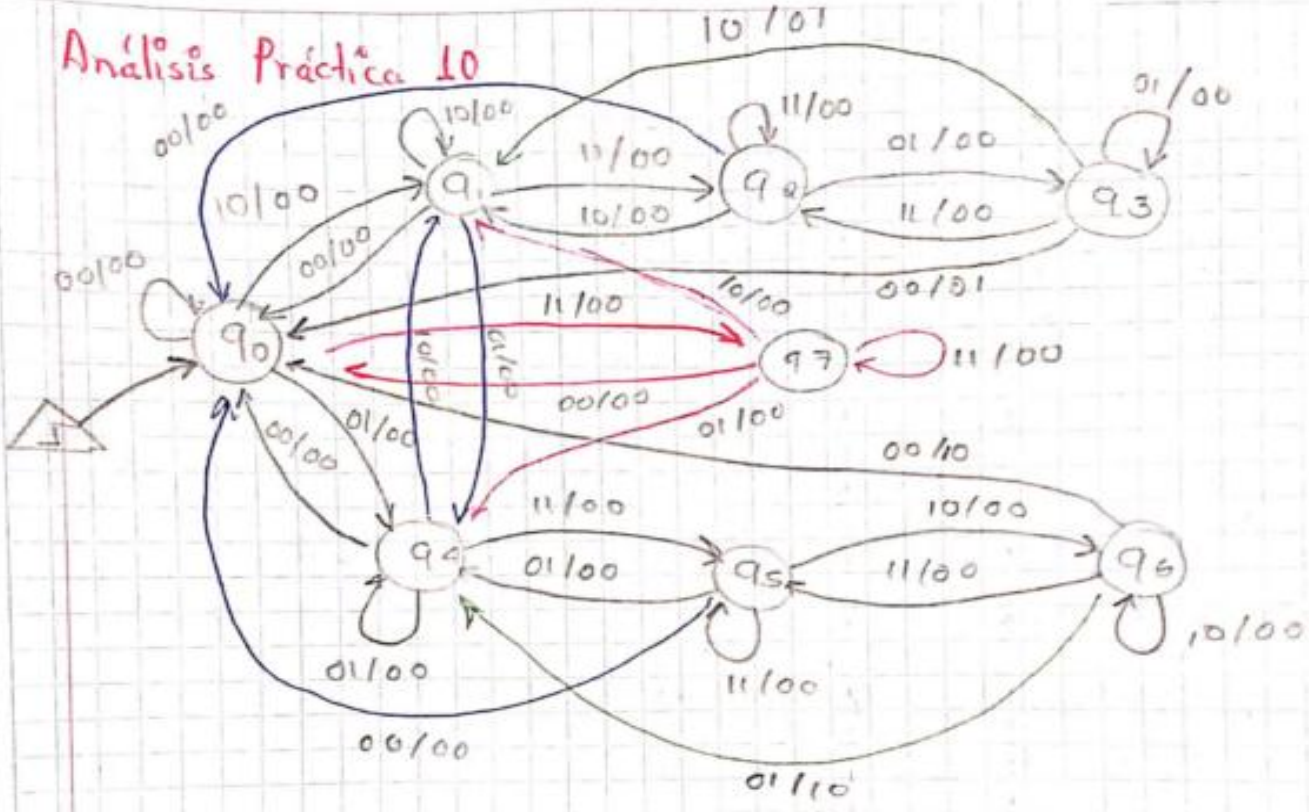
Análisis a mano con FF’JK y Código Grey

Santos Méndez Ulises Jesús 2CV8

Análisis Práctica 10



- Código Gray
- FF’JK
- Se calculan las salidas

Se tienen 8 estados entonces:

$2^n = 2^3 = 8$ entonces necesitamos 3 FFJK

Edo_Act	E	Edo_Sig	S
q0	0 0	q0	0 0
q0	0 1	q4	0 0
q0	1 0	q1	0 0
q0	1 1	q7	0 0
q1	0 0	q0	0 0
q1	0 1	q4	0 0
q1	1 0	q1	0 0
q1	1 1	q2	0 0
q2	0 0	q0	0 0
q2	0 1	q3	0 0
q2	1 0	q1	0 0
q2	1 1	q2	0 0
q3	0 0	q0	0 1
q3	0 1	q3	0 0
q3	1 0	q1	0 1
q3	1 1	q2	0 0
q4	0 0	q0	0 0
q4	0 1	q4	0 0
q4	1 0	q1	0 0
q4	1 1	q5	0 0
q5	0 0	q0	0 0
q5	0 1	q4	0 0
q5	1 0	q6	0 0
q5	1 1	q5	0 0
q6	0 0	q0	1 0
q6	0 1	q4	1 0
q6	1 0	q6	0 0
q6	1 1	q5	0 0
q7	0 0	q0	0 0
q7	0 1	q4	0 0
q7	1 0	q1	0 0
q7	1 1	q7	0 0

q0 → 0 0 0

q1 → 0 0 1

q2 → 0 1 1

q3 → 0 1 0

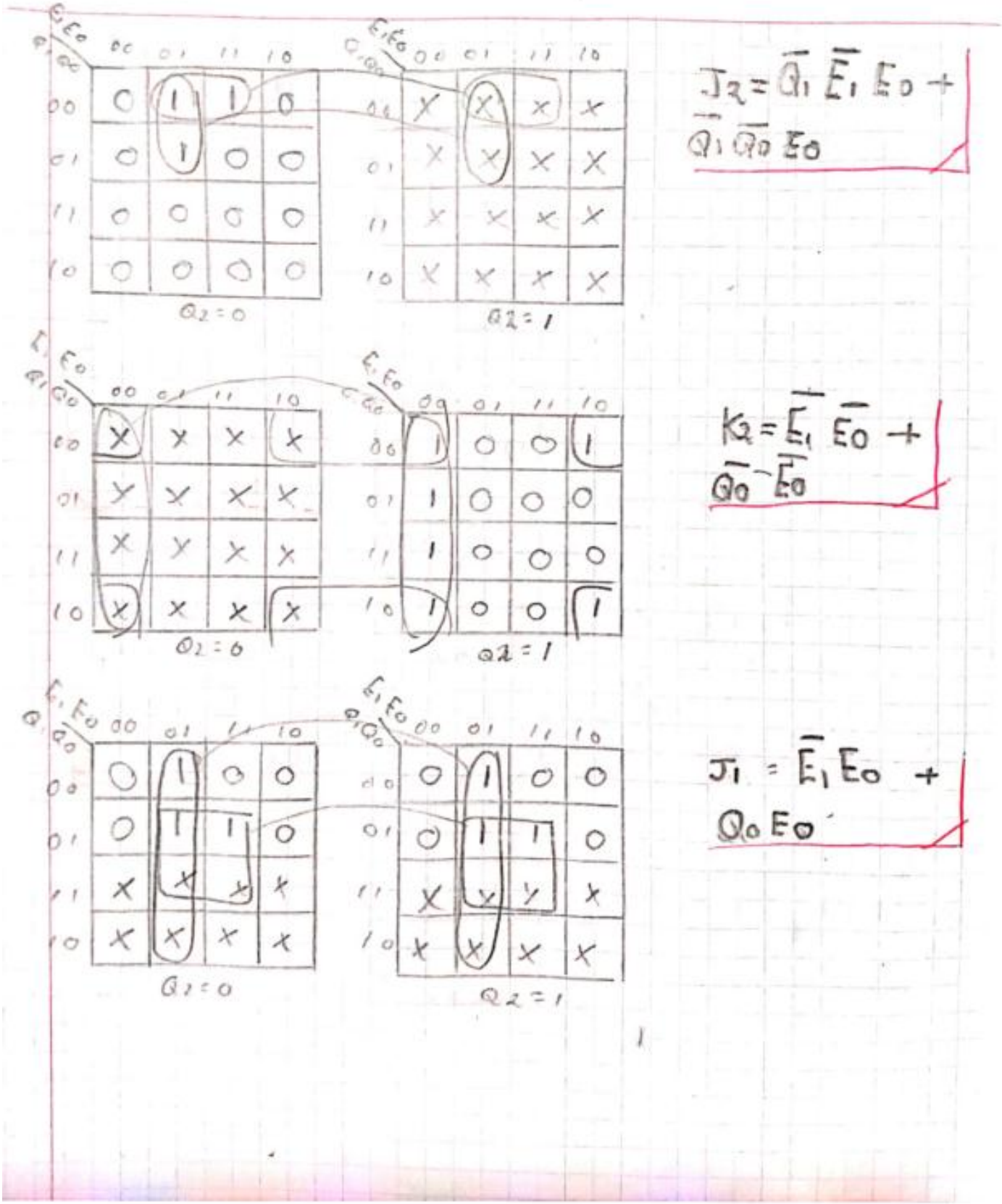
q4 → 1 1 0

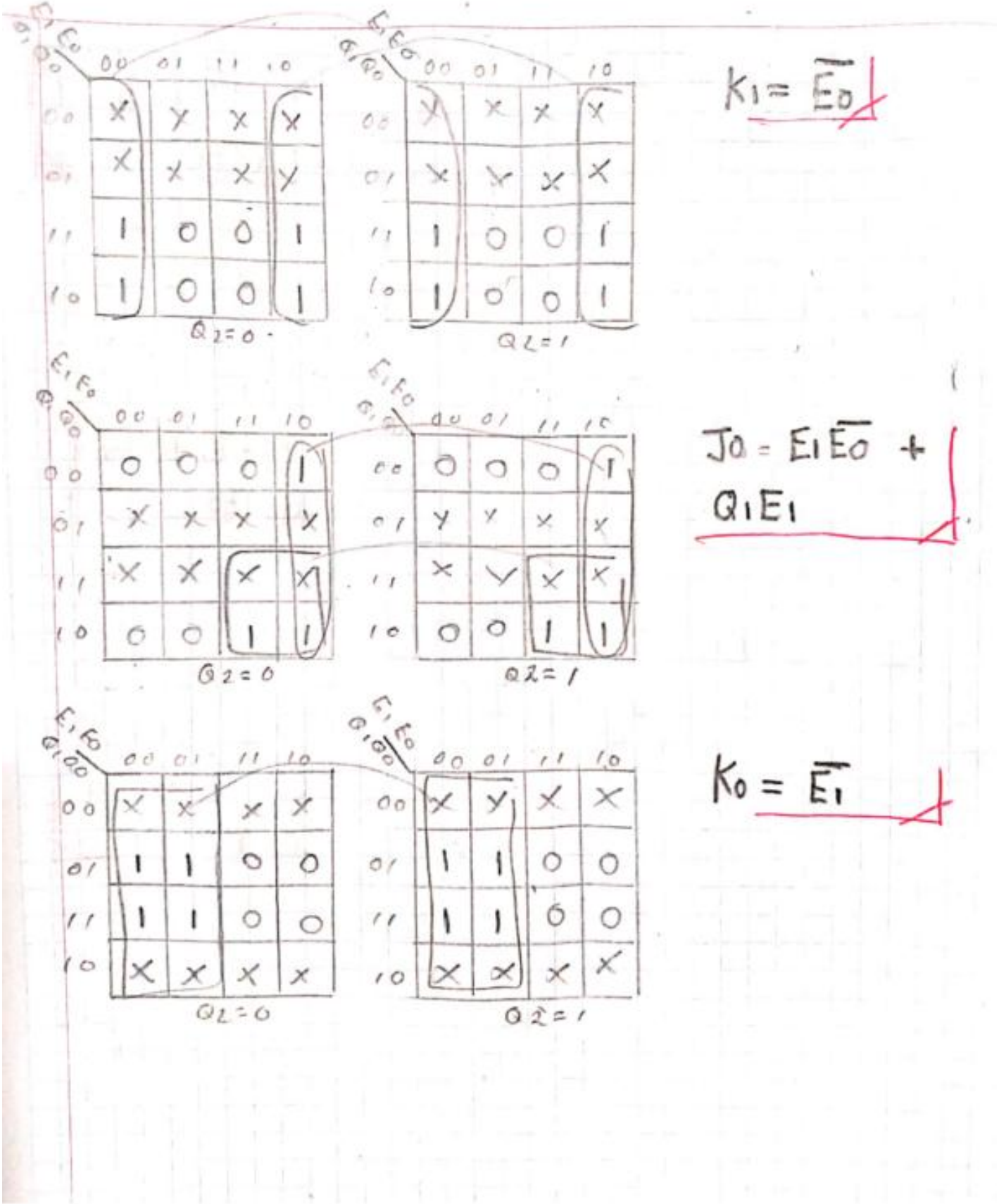
q5 → 1 1 1

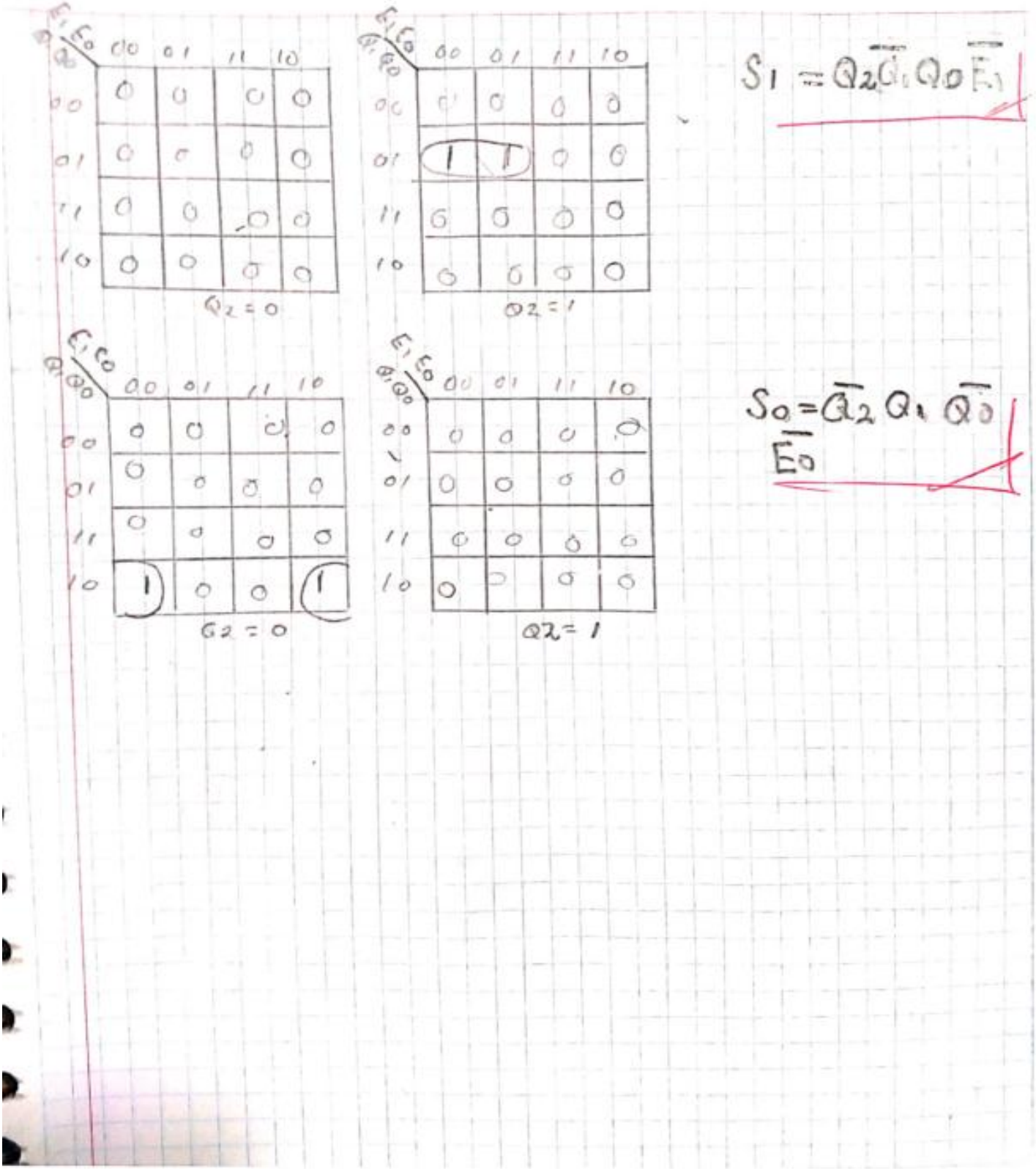
q6 → 1 0 1

q7 → 1 0 0

Edo Act $Q_2\ Q_1\ Q_0$	E $E_1\ E_0$	Edo Sig $Q_2^+\ Q_1^+\ Q_0^+$	S $S_1\ S_0$	J_2	K_2	J_1	K_1	J_0	K_0
0 0 0	0 0	0 0 0	0 0	0	X	0	X	0	X
0 0 0	0 1	1 1 0	0 0	1	X	1	X	1	X
0 0 0	1 0	0 0 1	0 0	0	X	0	X	1	X
0 0 0	1 1	1 0 0	0 0	1	X	0	X	0	X
0 0 1	0 0	0 0 0	0 0	0	X	0	X	X	1
0 0 1	0 1	1 1 0	0 0	1	X	1	X	X	1
0 0 1	1 0	0 0 1	0 0	0	X	0	X	X	0
0 0 1	1 1	0 1 1	0 0	0	X	1	X	X	1
0 1 1	0 0	0 0 0	0 0	0	X	X	1	X	1
0 1 1	0 1	0 1 0	0 0	0	X	X	0	X	0
0 1 1	1 0	0 0 1	0 0	0	X	X	1	X	0
0 1 1	1 1	0 1 1	0 0	0	X	X	0	X	0
0 1 0	0 0	0 0 0	0 1	0	X	X	1	0	X
0 1 0	0 1	0 1 0	0 0	0	X	X	0	0	X
0 1 0	1 0	0 0 1	0 1	0	X	X	1	1	X
0 1 0	1 1	0 1 1	0 0	0	X	X	0	1	X
1 1 0	0 0	0 0 0	0 0	X	1	X	1	0	X
1 1 0	0 1	1 1 0	0 0	X	0	X	0	0	X
1 1 0	1 0	0 0 1	0 0	X	1	X	1	1	X
1 1 0	1 1	1 1 1	0 0	X	0	X	0	1	X
1 1 1	0 0	0 0 0	0 0	X	1	X	1	X	1
1 1 1	0 1	1 1 0	0 0	X	0	X	0	X	1
1 1 1	1 0	1 0 1	0 0	X	0	X	1	X	0
1 1 1	1 1	1 1 1	0 0	X	0	X	0	X	0
1 0 1	0 0	0 0 0	1 0	X	1	0	X	X	1
1 0 1	0 1	1 1 0	1 0	X	0	1	X	X	1
1 0 1	1 0	1 0 1	0 0	X	0	0	X	X	0
1 0 1	1 1	1 1 1	0 0	X	0	1	X	X	0
1 0 0	0 0	0 0 0	0 0	X	1	0	X	0	X
1 0 0	0 1	1 1 0	0 0	X	0	1	X	0	X
1 0 0	1 0	0 0 1	0 0	X	1	0	X	1	X
1 0 0	1 1	1 0 0	0 0	X	0	0	X	0	X


$J_2 = \bar{Q}_1 \bar{E}_1 E_0 + \bar{Q}_1 \bar{Q}_0 E_0$
$K_2 = \bar{E}_1 \bar{E}_0 + \bar{Q}_0 \bar{E}_0$
$J_1 = \bar{E}_1 E_0 + Q_0 E_0$
$Q_2 = 0$
$Q_2 = 1$
$Q_1 = 0$


$K_1 = \overline{E_0}$
$J_0 = E_1\overline{E_0} + Q_1E_1$
$K_0 = \overline{E_1}$
$Q_2 = 0$
$Q_2 = 1$


$S_1 = Q_2\bar{Q}_1Q_0\bar{E}_1$
$S_0 = \bar{Q}_2Q_1\bar{Q}_0\bar{E}_0$
$Q_2 = 0$
$Q_2 = 1$
$Q_2 = 0$
$Q_2 = 1$

Código VHDL

GAL1

```
 1 --Ulises Jesùs Santos Mèndez
 2 --Pràctica 10 "Sensores"
 3 library ieee;
 4 use ieee.std_logic_1164.all;
 5 use ieee.std_logic_unsigned.all;
 6 use ieee.std_logic_arith.all;
 7
 8 entity contador is
 9 port(clk,clr: in std_logic;
10      sen: in std_logic_vector(1 downto 0);
11      uni: out std_logic_vector(3 downto 0);
12      dec: out std_logic_vector(2 downto 0));
13
14      attribute pin_numbers of contador: entity is
15      "clr:13 sen(1):2 sen(0):3 dec(2):20 dec(1):19 "
16 &    "dec(0):18 uni(3):17 uni(2):16 uni(1):15 "
17 &    "uni(0):14 ";
18 end contador;
19
20 architecture arqgal of contador is
21 --Estados de la màquina de Mealy
22 type estados is (q0,q1,q2,q3,q4,q5,q6,q7);
23 signal act,sig: estados;
24 signal sal: std_logic_vector(1 downto 0);
```

```
25 begin
26
27 --Proceso cambio de estado para estado actual
28     process(clk,clr)
29     begin
30         if(clr='1')then
31             act <= q0;
32         elsif(rising_edge(clk))then
33             act <= sig;
34         end if;
35     end process;
36
37 --Proceso para determinar el estado siguiente
38     process(sen,act)
39     begin
40         case act is
41             when q0 =>
42                 if(sen = "00")then
43                     sal <= "00";
44                     sig <= q0;
45                 elsif(sen = "01")then
46                     sal <= "00";
47                     sig <= q4;
48                 elsif(sen = "10")then
```

```
49                      sal <= "00";
50                      sig <= q1;
51                  else
52                      sal <= "00";
53                      sig <= q7;
54                  end if;
55              when q1 =>
56                  if(sen = "00")then
57                      sal <= "00";
58                      sig <= q0;
59                  elsif(sen = "01")then
60                      sal <= "00";
61                      sig <= q4;
62                  elsif(sen = "10")then
63                      sal <= "00";
64                      sig <= q1;
65                  else
66                      sal <= "00";
67                      sig <= q2;
68                  end if;
69              when q2 =>
70                  if(sen = "00")then
71                      sal <= "00";
72                      sig <= q0;
```

```
73                  elsif(sen = "01")then
74                      sal <= "00";
75                      sig <= q3;
76                  elsif(sen = "10")then
77                      sal <= "00";
78                      sig <= q1;
79                  else
80                      sal <= "00";
81                      sig <= q2;
82                  end if;
83              when q3 =>
84                  if(sen = "00")then
85                      sal <= "01";
86                      sig <= q0;
87                  elsif(sen = "01")then
88                      sal <= "00";
89                      sig <= q3;
90                  elsif(sen = "10")then
91                      sal <= "01";
92                      sig <= q1;
93                  else
94                      sal <= "00";
95                      sig <= q2;
96                  end if;
```

```
97              when q4 =>
98                  if(sen = "00")then
99                      sal <= "00";
100                     sig <= q0;
101                 elsif(sen = "01")then
102                     sal <= "00";
103                     sig <= q4;
104                 elsif(sen = "10")then
105                     sal <= "00";
106                     sig <= q1;
107                 else
108                     sal <= "00";
109                     sig <= q5;
110                 end if;
111             when q5 =>
112                 if(sen = "00")then
113                     sal <= "00";
114                     sig <= q0;
115                 elsif(sen = "01")then
116                     sal <= "00";
117                     sig <= q4;
118                 elsif(sen = "10")then
119                     sal <= "00";
120                     sig <= q6;
```

```
121                 else
122                     sal <= "00";
123                     sig <= q5;
124                 end if;
125             when q6 =>
126                 if(sen = "00")then
127                     sal <= "10";
128                     sig <= q0;
129                 elsif(sen = "01")then
130                     sal <= "10";
131                     sig <= q4;
132                 elsif(sen = "10")then
133                     sal <= "00";
134                     sig <= q6;
135                 else
136                     sal <= "00";
137                     sig <= q5;
138                 end if;
139             when q7 =>
140                 if(sen = "00")then
141                     sal <= "00";
142                     sig <= q0;
143                 elsif(sen = "01")then
144                     sal <= "00";
```

```
145                     sig <= q4;
146                 elsif(sen = "10")then
147                     sal <= "00";
148                     sig <= q1;
149                 else
150                     sal <= "00";
151                     sig <= q7;
152                 end if;
153         end case;
154     end process;
155 --Proceso contador de decada
156     process(clk,clr)
157     begin
158         if(clr='1')then
159             uni <= (others => '0');
160             dec <= (others => '0');
161         elsif(rising_edge(clk))then
162             if(sen="00")then
163                 uni <= uni;
164                 dec <= dec;
165             elsif(sen="01")then
166                     if(uni="1001")then
167                         uni <= "0000";
168                         dec <= dec+1;
```

```
169                     else
170                         uni <= uni+1;
171                     end if;
172             elsif(sen="10")then
173                     if(uni="0000")then
174                         uni <= "1001";
175                         dec <= dec-1;
176                     else
177                         uni <= uni-1;
178                     end if;
179             end if;
180         end if;
181     end process;
182 end arqgal;
```

GAL2

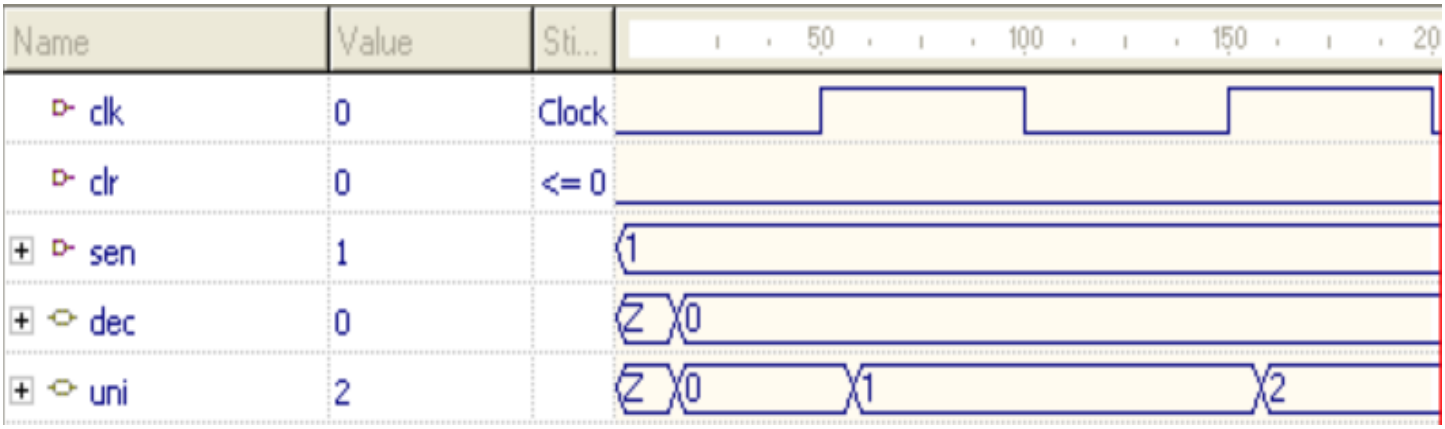
```
1  --Ulises Jesùs Santos Mèndez
2  --Pràctica 10 "Sensores"
3  library ieee;
4  use ieee.std_logic_1164.all;
5
6  entity saldisp is
7  port(clk,clr: in std_logic;
8       uni: in std_logic_vector(3 downto 0);
9       dec: in std_logic_vector(2 downto 0);
10      cat: inout std_logic_vector(2 downto 0);
11      display: out std_logic_vector(6 downto 0));
12
13      attribute pin_numbers of saldisp: entity is
14      "clr:13 uni(3):8 uni(2):9 uni(1):10 uni(0):11 "
15 &    "dec(2):5 dec(1):6 dec(0):7 cat(2):22 cat(1):23 "
16 &    "cat(0):14 display(6):21 display(1):20 display(4):19 "
17 &    "display(3):18 display(2):17 display(5):16 "
18 &    "display(0):15 ";
19 end saldisp;
20
21 architecture arqsal of saldisp is
22 signal su: std_logic_vector(3 downto 0):= "0000";
23 constant sd: std_logic_vector(3 downto 0):= '0'&dec;
24 signal bcd: std_logic_vector(3 downto 0);
```

```
25 begin
26 --Contador de anillo
27     process(clk,clr)
28     begin
29         if(clr='1')then
30             cat <= "011";
31         elsif(rising_edge(clk))then
32             cat <= to_stdlogicvector(to_bitvector(cat)
33             ror 1);
34         end if;
35     end process;
36
37 --Multiplexor
38     process(bcd,uni,dec,cat,sd,cat,su)
39     begin
40             if(cat="011")then
41                 bcd <= su;
42             elsif(cat="101")then
43                 bcd <= sd;
44             elsif(cat="110")then
45                 bcd <= uni;
46             else
47                 bcd <= "----";
48             end if;
```

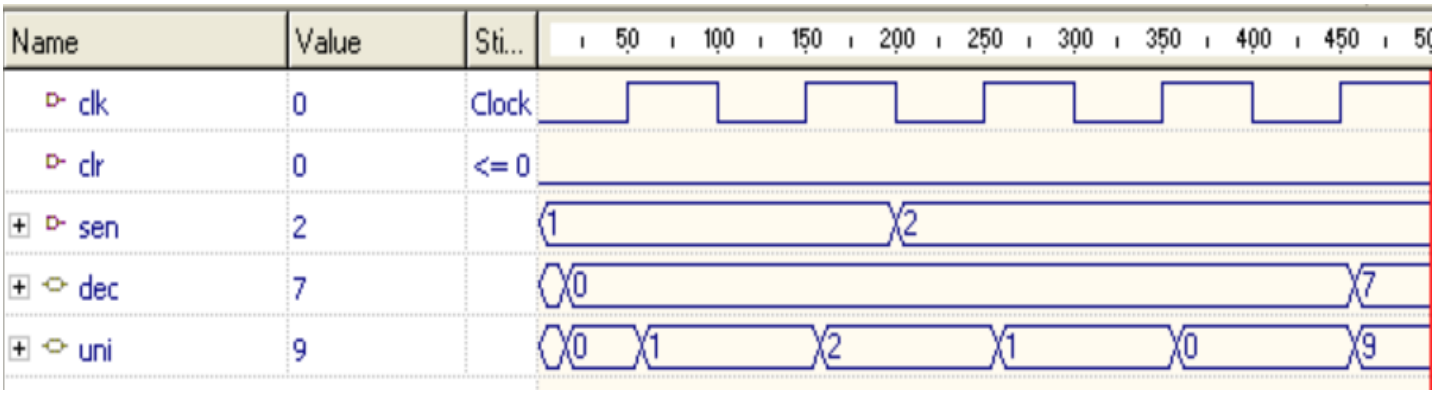
```
49     end process;
50
51 --Decodificador
52     process(bcd)
53     begin
54         case bcd is
55             when "0000" => display <= "1111110";--0//7E
56             when "0001" => display <= "0110000";--1//30
57             when "0010" => display <= "1101101";--2//6D
58             when "0011" => display <= "1111001";--3//79
59             when "0100" => display <= "0110011";--4//33
60             when "0101" => display <= "1011011";--5//5B
61             when "0110" => display <= "1011111";--6//5F
62             when "0111" => display <= "1110001";--7//71
63             when "1000" => display <= "1111111";--8//7F
64             when "1001" => display <= "1110011";--9//73
65             when others => display <= "-------";
66         end case;
67     end process;
68 end arqsal;
```

Simulación en Active HDL-Sim

2 entradas



3 salidas



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= 8 4013, 1 555, 6 7408, 6 7432, 5 7404, 1 74ls149.
3. ¿Cuántos pines de entrada/salida del PLD1 22V10 y PLD2 22V10 se usan en el diseño?
   R=4 pines de entrada y 7 de salida del PLD1 22V10 y 9 pines de entrada y 10 de salida del PLD2 22V10.
4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD1 22V10 y PLD2 22V10?
   R=52 términos producto y se ocupa un 50% del PLD1 22V10, para el PLD2 22V10 se ocupan 65 términos producto, y se ocupa un 86% del PLD.
5. ¿Qué puedes concluir de esta práctica?
   R= Fue muy importante para comenzar a crear proyectos con otro tipo de descripción, en este caso se modelo una máquina con muchos Flip Flops solo con una directiva type, entonces es una buena forma de ocupar las herramientas sobre la que se trabaja.