Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica

IEE2713 — Sistemas Digitales

Pauta T4

1. Mapa para SOP:

x_3 \ x_2x_1	00	01	11	10
0	0	1	1	1
1	0	1	0	0

Mapa para POS:

x_3 \ x_2x_1	00	01	11	10
0	0	1	1	1
1	0	1	0	0

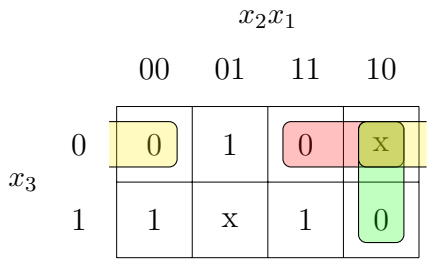
Esto nos permite encontrar las siguientes funciones minimizadas:

$$f(x_3, x_2, x_1) = \bar{x_3}x_2 + \bar{x_2}x_1$$
$$f(x_3, x_2, x_1) = (\bar{x_3} + \bar{x_2})(x_2 + x_1)$$

2. Mapa para SOP:

x_3 \ x_2x_1	00	01	11	10
0	0	1	0	x
1	1	x	1	0

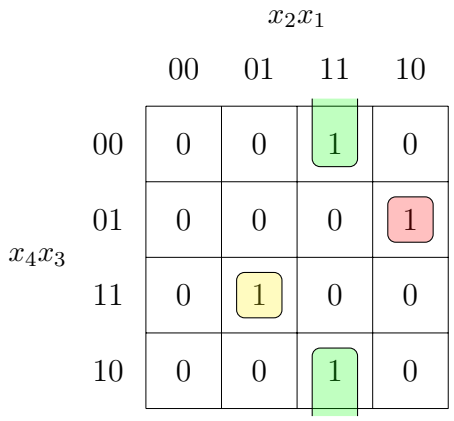
Mapa para POS:



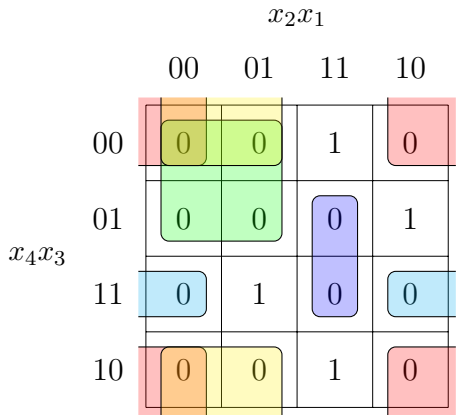
Las funciones resultantes son:

$$f(x_3, x_2, x_1) = \bar{x_2}x_1 + x_3\bar{x_2} + x_3x_1$$
$$f(x_3, x_2, x_1) = (\bar{x_2} + x_1)(x_3 + x_1)(x_3 + \bar{x_2})$$

3. Mapa para SOP:



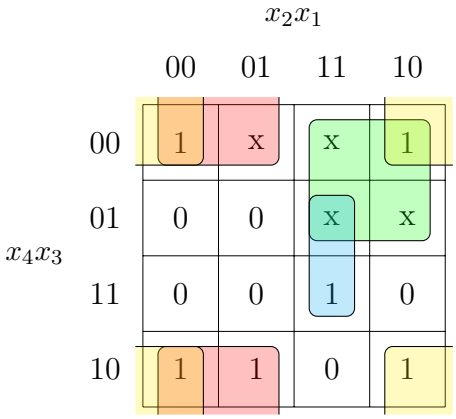
Mapa para POS:



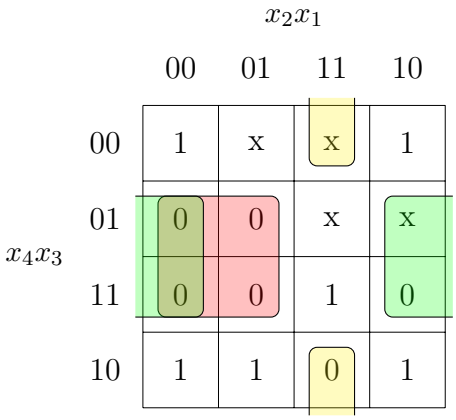
Las funciones resultantes son:

$$f(x_4, x_3, x_2, x_1) = \bar{x_4}x_3x_2\bar{x_1} + x_4x_3\bar{x_2}x_1 + \bar{x_3}x_2x_1$$
$$f(x_4, x_3, x_2, x_1) = (x_3 + x_1)(x_4 + x_2)(x_3 + x_2)(\bar{x_3} + \bar{x_2} + \bar{x_1})(\bar{x_4} + \bar{x_3} + x_1)$$

4. Mapa para POS:



Mapa para SOP:



Las funciones resultantes son:

$$f(x_4, x_3, x_2, x_1) = x_3x_2x_1 + \bar{x_3}\bar{x_2} + \bar{x_4}x_2 + \bar{x_3}\bar{x_1}$$
$$f(x_4, x_3, x_2, x_1) = (\bar{x_3} + x_1)(\bar{x_3} + x_2)(x_3 + \bar{x_2} + \bar{x_1})$$

5. La afirmación es falsa. Como contraejemplo considere $f(x_3, x_2, x_1) = \sum m(0, 5, 7)$. Al calcular notamos que existe una SOP pero dos POS equivalentes:

   - SOP:

$$f = x_1x_3 + \bar{x_1}\bar{x_2}\bar{x_3}$$

- POS:

$$f = (x_1 + \bar{x_3})(\bar{x_1} + x_3)(x_1 + \bar{x_2})$$
$$f = (x_1 + \bar{x_3})(\bar{x_1} + x_3)(\bar{x_2} + x_3)$$

6.

$(0111011110)_2 =$
$0 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 478$

$(1011100111)_2 =$
$1 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 743$

$(3751)_8 =$
$3 \cdot 8^3 + 7 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0 = 2025$

$(A25F)_16 =$
$10 \cdot 16^3 + 2 \cdot 16^2 + 5 \cdot 16^1 + 15 \cdot 16^0 = 41567$

$(F0F0)_16 =$
$15 \cdot 16^3 + 0 \cdot 16^2 + 15 \cdot 16^1 + 0 \cdot 16^0 = 61680$

7.

$(129)_{10} = 128 + 1 = (10000001)_2$
$(\underbrace{10}_{2}\underbrace{000}_{0}\underbrace{001}_{1})_2 = (201)_8$
$(\underbrace{1000}_{8}\underbrace{0001}_{1}) = (81)_{16}$

$(244)_{10} = 128 + 64 + 32 + 16 + 4 = (11110100)_2$
$(\underbrace{11}_{3}\underbrace{110}_{6}\underbrace{100}_{4})_2 = (364)_8$
$(\underbrace{1111}_{15}\underbrace{0100}_{4}) = (F4)_{16}$

$(1906)_{10} = 1024 + 512 + 256 + 64 + 32 + 16 + 2 = (11101110010)_2$
$(\underbrace{11}_{3}\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{010}_{2})_2 = (3562)_8(201)_8$
$(\underbrace{111}_{7}\underbrace{0111}_{7}\underbrace{0010}_{2}) = (772)_{16}$

$(65)_2 = 64 + 1 = (1000001)_2$
$(\underbrace{1}_{1}\underbrace{000}_{0}\underbrace{001}_{1})_2 = (101)_8$
$(\underbrace{100}_{4}\underbrace{0001}_{1}) = (41)_{16}$

8. *Insertar algo sorprendente de la Basys 3.*

11. Asumiremos que el display es de cátodo común (como en la Basys 3), por lo que para encender el segmento éste se debe setear en cero.

    Las letras "B" y "D" son escritas en minúscula para poder diferenciarlas de "8" y "0" respectivamente.

N°	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0
A	1	0	1	0	0	0	0	1	0	0	0
B	1	0	1	1	1	1	0	0	0	0	0
C	1	1	0	0	0	1	1	0	0	0	1
D	1	1	0	1	1	0	0	0	0	1	0
E	1	1	1	0	0	1	1	0	0	0	0
F	1	1	1	1	0	1	1	1	0	0	0

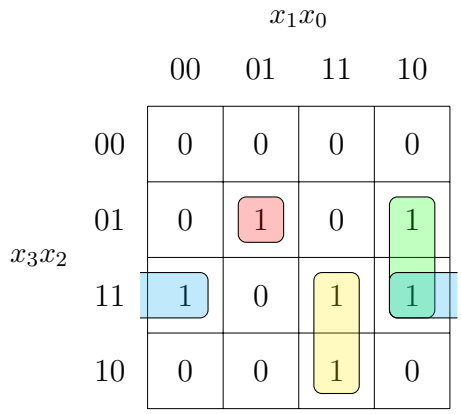
12. Hacemos muchos mapas de Karnaugh:

- a:

x_3x_2 \ x_1x_0	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	1	0	0
10	0	0	1	0

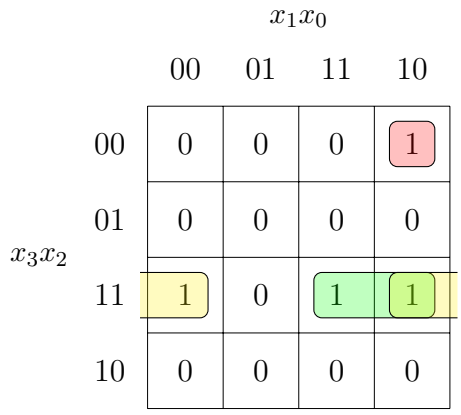
$$a = \bar{x_3}\bar{x_2}\bar{x_1}x_0 + \bar{x_3}x_2\bar{x_1}\bar{x_0} + x_3x_2\bar{x_1}x_0 + x_3\bar{x_2}x_1x_0$$

- b:



$$b = \bar{x_3}x_2\bar{x_1}x_0 + x_2x_1\bar{x_0} + x_3x_2\bar{x_0} + x_3x_1x_0$$

- c:



$$c = \bar{x_3}\bar{x_2}x_1\bar{x_0} + x_3x_2\bar{x_0} + x_3x_2x_1$$

- d:

x_3x_2 \ x_1x_0	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	1	0
10	0	0	0	1

$$d = \bar{x_3}\bar{x_2}\bar{x_1}x_0 + \bar{x_3}x_2\bar{x_1}\bar{x_0}x_3\bar{x_2}x_1\bar{x_0} + x_2x_1x_0$$

- e:

x_3x_2 \ x_1x_0	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	0	0	0	0
10	0	1	0	0

$$e = \bar{x_3}x_2\bar{x_1} + \bar{x_2}\bar{x_1}x_0 + \bar{x_3}x_0$$

- f:

x_3x_2 \ x_1x_0	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	1	0	0
10	0	0	0	0

$$f = x_3x_2\bar{x_1}x_0 + \bar{x_3}\bar{x_2}x_1 + \bar{x_3}\bar{x_2}x_0 + \bar{x_3}x_1x_0$$

- g:

x_3x_2 \ x_1x_0	00	01	11	10
00	1	1	0	0
01	0	0	1	0
11	1	0	0	0
10	0	0	0	0

$$g = \bar{x_3}x_2x_1x_0 + x_3x_2\bar{x_1}\bar{x_0} + \bar{x_3}\bar{x_2}\bar{x_1}$$

13. Notar que la selección del segmento se realiza combinacionalmente mediante el comando assign y el operador ?:.

Listing 1: pregunta_13.v

```
1  //----------------------------------------------
2  // Modulo:  pregunta_13
3  // Archivo: pregunta_13.v
4  // Funcion: Numero binario a 7-seg hexadecimal
5  //----------------------------------------------
6
7  module pregunta_13(input  [3:0] x,
```

```
                    output [6:0] seg);

  always @(*) // Always combinacional
   begin
     case(sw[3:0])
    // Segmentos: 7'bABCDEFG
       15: seg <= 7'b0111000;
       14: seg <= 7'b0110000;
       13: seg <= 7'b1000010;
       12: seg <= 7'b0110001;
       11: seg <= 7'b1100000;
       10: seg <= 7'b0001000;
       9:  seg <= 7'b0000100;
       8:  seg <= 7'b0000000;
       7:  seg <= 7'b0001111;
       6:  seg <= 7'b0100000;
       5:  seg <= 7'b0100100;
       4:  seg <= 7'b1001100;
       3:  seg <= 7'b0000110;
       2:  seg <= 7'b0010010;
       1:  seg <= 7'b1001111;
       0:  seg <= 7'b0000001;
     endcase
   end

endmodule
```

14. Misma idea que la pregunta anterior para los segmentos. Para seleccionar el display, se utiliza un decodificador implementado de manera muy resumida mediante la operación *shift left*.

Listing 2: pregunta_14.v

```
//-----------------------------------------------------
// Modulo   : pregunta_14
// Archivo  : pregunta_14.v
// Function : sw[5:4] seleccionan display.
//            sw[3:0] seleccionan numero.
//-----------------------------------------------------

module pregunta_14(input  wire [5:0] sw,
                   output reg  [3:0] an,
                   output reg  [6:0] seg);

  // Decoder 2-4 para elegir segmento //
  // El 1 se corre a la derecha tantas veces como valor tenga sw[5:4]
```

```
14    assign an = (1 << sw[5:4]);
15
16    always @(*) // Always combinacional
17      begin
18        case(sw[3:0])
19       // Segmentos: 7'bABCDEFG
20           15: seg <= 7'b0111000;
21           14: seg <= 7'b0110000;
22           13: seg <= 7'b1000010;
23           12: seg <= 7'b0110001;
24           11: seg <= 7'b1100000;
25           10: seg <= 7'b0001000;
26           9:  seg <= 7'b0000100;
27           8:  seg <= 7'b0000000;
28           7:  seg <= 7'b0001111;
29           6:  seg <= 7'b0100000;
30           5:  seg <= 7'b0100100;
31           4:  seg <= 7'b1001100;
32           3:  seg <= 7'b0000110;
33           2:  seg <= 7'b0010010;
34           1:  seg <= 7'b1001111;
35           0:  seg <= 7'b0000001;
36        endcase
37      end
38
39  endmodule
```