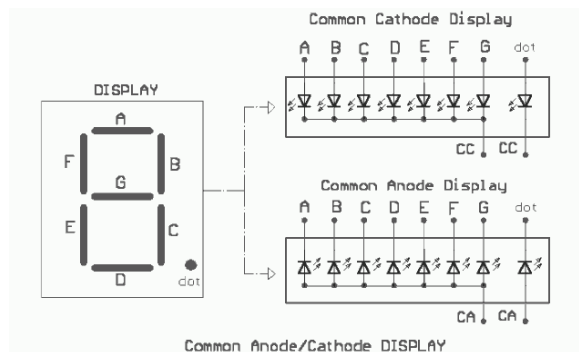


<b>Nombre:</b> Gustavo Alejandro Solorio Ramos	<b>Correo:</b> gustavo.solorio@iteso.mx
<b>Nombre del curso:</b> Diseño de Sistemas Digitales	<b>Nombre del profesor:</b> Dr. Omar Longoria Gándara
<b>Práctica 1</b>	<b>BCD-&gt; 7 Segmentos</b>
<b>Fecha:</b> 11 de octubre de 2021	

Las siguientes tablas describen el desarrollo de un decodificador BDC a 7 segmentos, mostrando los resultados para los números del 1 al 9, siendo los casos restantes indistintos. Se toma como referencia la siguiente imagen tomando como los segmentos A – G los equivalentes para su asignación en la tarjeta de desarrollo DE10 lite.



**Imagen 1.** Configuración de display de 7 segmentos + punto decimal, se toma configuración de ánodo común

Decimal	BCD (0 a 9) a 7 Segmentos											
	Entradas				Salidas							
	SW3	SW2	SW1	SW0	HEX0[7]	HEX0[6]	HEX0[5]	HEX0[4]	HEX0[3]	HEX0[2]	HEX0[1]	HEX0[0]
0	0	0	0	0	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	0	0	1
2	0	0	1	0	1	0	1	0	0	1	0	0
3	0	0	1	1	1	0	1	1	0	0	0	0
4	0	1	0	0	1	0	0	1	1	0	0	1
5	0	1	0	1	1	0	0	1	0	0	1	0
6	0	1	1	0	1	0	0	0	0	0	1	0
7	0	1	1	1	1	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	0	0	0	0	0
9	1	0	0	1	1	0	0	1	0	0	0	0
A	1	0	1	0	X	X	X	X	X	X	X	X
B	1	0	1	1	X	X	X	X	X	X	X	X
C	1	1	0	0	X	X	X	X	X	X	X	X
D	1	1	0	1	X	X	X	X	X	X	X	X
E	1	1	1	0	X	X	X	X	X	X	X	X
F	1	1	1	1	X	X	X	X	X	X	X	X

A continuación, se realiza la simplificación de los mapas de Karnaugh, así como las asignaciones equivalentes para su código en el archivo de Verilog utilizado en el proyecto

HEX0[7]		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	1	1	X	1
	"01"	1	1	X	1
	"11"	1	1	X	X
	"10"	1	1	X	X

$$\text{HEX0}[7] = 1$$

```
assign HEX0[7]= 1'b1
```

HEX0[6]		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	1	0	X	0
	"01"	1	0	X	0
	"11"	0	1	X	X
	"10"	0	0	X	X

$$\text{HEX0}[6] = \text{SW3}'\text{SW2}'\text{SW1}' + \text{SW2 SW1 SW0}$$

```
assign HEX0[6] = (~SW[3] & ~SW[2] & ~SW[1]) | (SW[2] & SW[1] & SW[0]);
```

HEX0[5]		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	0	0	X	0
	"01"	1	0	X	0
	"11"	1	1	X	X
	"10"	1	0	X	X

$$\text{HEX0}[5] = \text{SW1} * \text{SW0} + \text{SW2}' * \text{SW1} + \text{SW3}' * \text{SW2}' * \text{SW0}$$

```
assign HEX0[5] = (SW[1] & SW[0]) | (~SW[2] & SW[1]) | (~SW[3] & ~SW[2] & SW[0]);
```

HEX0[4]		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	0	1	X	0
	"01"	1	1	X	1
	"11"	1	1	X	X
	"10"	0	0	X	X

$$\text{HEX0}[4] = \text{SW0} + \text{SW2} * \text{SW1}'$$

```
assign HEX0[4] = (SW[0]) | (SW[2] & ~SW[1]);
```

HEX0[3]		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	0	1	X	0
	"01"	1	0	X	0
	"11"	0	1	X	X
	"10"	0	0	X	X

$$\text{HEX0}[3] = \text{SW3}'\text{SW2}' * \text{SW1}' * \text{SW0} + \text{SW2 SW1}'\text{SW0}' + \text{SW2 SW1 SW0}$$

```
assign HEX0[3]=
```

```
(~SW[3]&~SW[2]&~SW[1]&SW[0]) |(SW[2]&~SW[1]&~SW[0])|(SW[2]&SW[1]&SW[0]);
```

		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	0	0	X	0
	"01"	0	0	X	0
	"11"	0	0	X	X
	"10"	1	0	X	X

$$\text{HEX0}[2] = \text{SW2}' \text{SW1} \text{SW0}'$$

```
assign HEX0[2] = (~SW[2] & SW[1] & ~SW[0]);
```

		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	0	0	X	0
	"01"	0	1	X	0
	"11"	0	0	X	X
	"10"	0	1	X	X

$$\text{HEX0}[1] = \text{SW2} \text{SW1}' \text{SW0} + \text{SW2} \text{SW1} \text{SW0}'$$

```
assign HEX0[1] = (SW[2] & ~SW[1] & SW[0]) | (SW[2] & SW[1] & ~SW[0]);
```

		SW3,SW2			
		"00"	"01"	"11"	"10"
SW1,SW0	"00"	0	1	X	0
	"01"	1	0	X	0
	"11"	0	0	X	X
	"10"	0	0	X	X

$$\text{HEX0}[0] = \text{SW3}' \text{SW2}' * \text{SW1}' * \text{SW0} + \text{SW2} \text{SW1}' \text{SW0}'$$

```
assign HEX0[0] = (~SW[3] & ~SW[2] & ~SW[1] & SW[0]) | (SW[2] & ~SW[1] & ~SW[0]);
```

Una vez simplificadas las funciones por los mapas de Karnaugh, se procede a elaborar el archivo de Verilog con las declaraciones y asignaciones correspondientes, como muestra el siguiente código:

```
1 //=====
2 // This code is generated by Terasic System Builder
3 //=====
4 module TAREA_1(
5 /////////////// SEG7 ///////////////
6 output [7:0] HEX0,
7 /////////////// SW ///////////////
8 input [3:0] SW
9 );
10
11 //=====
12 // REG/WIRE declarations
13 //=====
14
15 //=====
16 // Structural coding
```

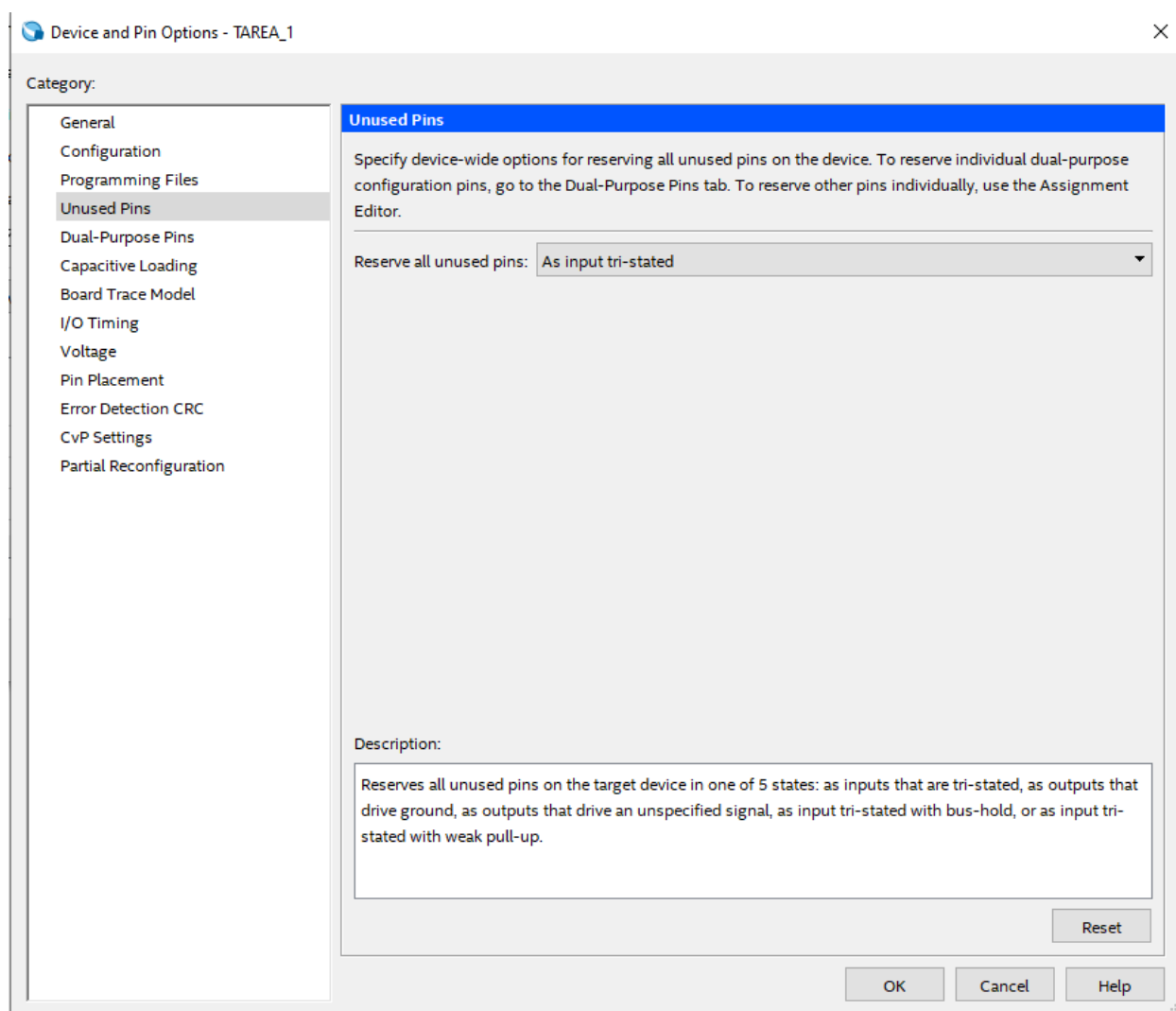
```

17 //=====
18 assign HEX0[7]= 1'b1; // dot
19 assign HEX0[6]= (~SW[3] & ~SW[2] & ~SW[1]) | (SW[2] & SW[1] & SW[0]); //G
20 assign HEX0[5]= (SW[1] & SW[0]) | (~SW[2] & SW[1]) | (~SW[3] & ~SW[2] & SW[0]); // F
21 assign HEX0[4]= (SW[0]) | (SW[2] & ~SW[1]); // E
22 assign HEX0[3]= (~SW[3] & ~SW[2] & ~SW[1] & SW[0]) | (SW[2] & ~SW[1] & ~SW[0]) | (SW[2] & SW[1] &
  SW[0]); // D
23 assign HEX0[2]= (~SW[2] & SW[1] & ~SW[0]); // C
24 assign HEX0[1]= (SW[2] & ~SW[1] & SW[0]) | (SW[2] & SW[1] & ~SW[0]); // B
25 assign HEX0[0]= (~SW[3] & ~SW[2] & ~SW[1] & SW[0]) | (SW[2] & ~SW[1] & ~SW[0]); // A
26 endmodule

```

**Código 1.** Programación ejecutada en el archivo de Verilog, asociado como el “Top Level” del proyecto de Quartus II.

Como paso adicional, antes de la compilación y programación del proyecto, se designan los pines sin usar como un estado de entrada triestado, para que se mantengan apagados los pines que no estén en uso para esta implementación.



**Imagen 2.** Configuración del dispositivo MAX 10 10M50DAF484C7G para las directivas de los pines sin usar