

## Ingeniería en electrónica

### Técnicas Digitales 1

#### Trabajo Práctico N°2: Conversor BCD a 7 Segmentos

##### Profesores

Titular: Ing. Toledo

JTP: Ing. Mercado

##### GRUPO: 06

Alumnos: Lutz Franco Leg: 90735 (3R3)

Nieto Tiago Leg: 89859 (3R2)

## Índice

• <i>Introducción</i>	pág. 2
• <i>Desarrollo</i>	pág. 2
• <i>Materiales</i>	pág. 9
• <i>Dificultades/inconvenientes</i>	pág. 10
• <i>Conclusión</i>	pág. 10

## Introducción

En este proyecto nos proponemos llevar a cabo el testeo y análisis del circuito integrado CD4511. Adicionalmente se realizará la descripción de hardware de dicho circuito integrado el cual se ejecutará en una placa de desarrollo FPGA.

## Desarrollo

El integrado CD4511 se trata de un conversor de cuatro entradas binarias

a siete salidas orientadas al manejo de un display de siete segmentos. Para su demostración utilizando el minilab dispusimos un display de cátodo común, ya que es el mismo tipo de display que tiene el kit CPLD. Este tipo de configuración consiste en proveer un uno lógico a cada segmento que se desee encender, y al estar todos los cátodos de los segmentos conectados al polo negativo de la alimentación, al enviar un estado alto en los segmentos estos se encenderán.

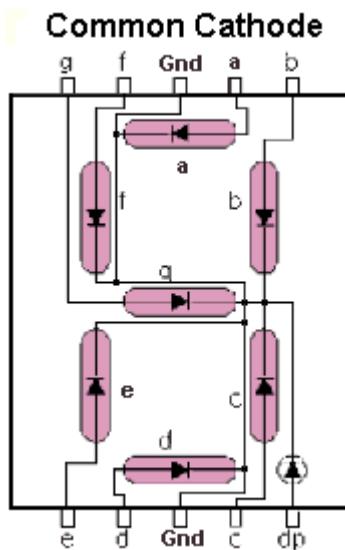


Imagen 1: Diagrama display cátodo común

El integrado CD4011 cuenta con 3 entradas de control adicionales a las cuatro del BCD, estas entradas son “lamp test”, “blanking” y “latch enable”.

Lamp test nos permite visualizar el correcto funcionamiento del display con un nivel de entrada bajo. De estar el nivel de entrada alto el display se apagará.

Blanking nos permite efectuar la conversión BCD a los siete segmentos del display con un nivel de entrada alto.

Latch enable, de poseer un nivel de entrada alto, generara como salida del integrado las salidas correspondientes al ultimo valor BCD previo a ser activado.

A continuación, se dispone una tabla de verdad del integrado CD4011 para su mayor comprensión:

LE	B	C	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	Blank
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	0	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	0	0	1	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	0	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	0	1	1	1	1	1	1	1	0	0	0	8
0	1	1	0	1	1	1	1	1	1	1	1	0	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	0	1	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	0	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	1	0	0	0	0	0	0	0	Blank
1	1	1	X	X	X	X	*	*	*	*	*	*	*	*

Imagen 2: Tabla de verdad integrado CD4011

Dispusimos el circuito integrado en la protoboard y corroboramos el correcto funcionamiento del integrado.

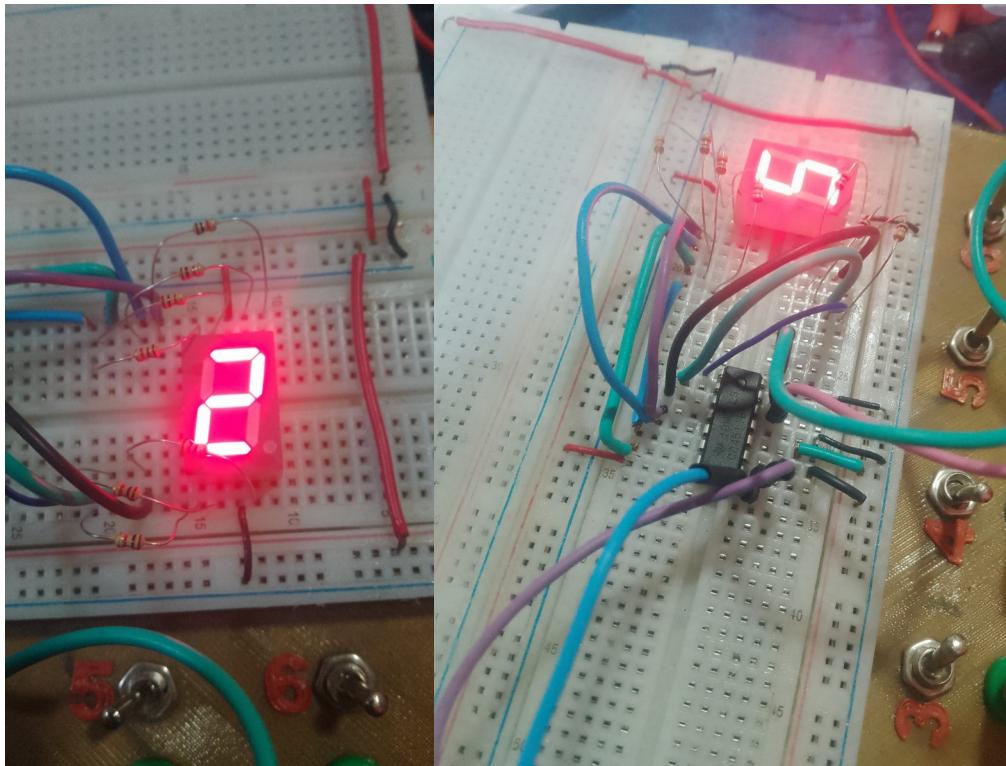


Imagen 3 y 4: Testeo CD4011 dentro del entorno minilab

Habiendo concluido con las pruebas en el minilab procedemos a realizar la descripción de hardware. Para eso realizamos mapas de Karnaugh de cada salida en base a la tabla de verdad anteriormente propuesta en la imagen dos.

Salida A:

		AB	
		00	01
CD	00	1	0
	01	1	1
	11	0	0
	10	0	1

Salida B:

		AB	
		00	01
CD	00	1	0
	01	1	1
	11	0	0
	10	1	1

Salida C:

		AB	
		00	01
		11	10
CD	00	1	0
	01	0	1
	11	0	0
	10	1	1

Salida D:

		AB	
		00	01
		11	10
CD	00	1	0
	01	1	1
	11	0	0
	10	1	0

Salida E:

		AB	
		00	01
CD	00	1	0
	01	1	1 1
	11	0	0
	10	1	0 1

Salida F:

		AB	
		00	01
CD	00	1	0
	01	0	1 1
	11	0	0
	10	1	0 1

### Salida G:

		AB			
		00	01	11	10
CD	00	1	0	0	0
	01	1	1	1	1
	11	0	0	0	0
	10	1	0	1	1

Posteriormente procedimos a utilizar el software Xilinx para realizar la correspondiente descripción de hardware del circuito integrado.

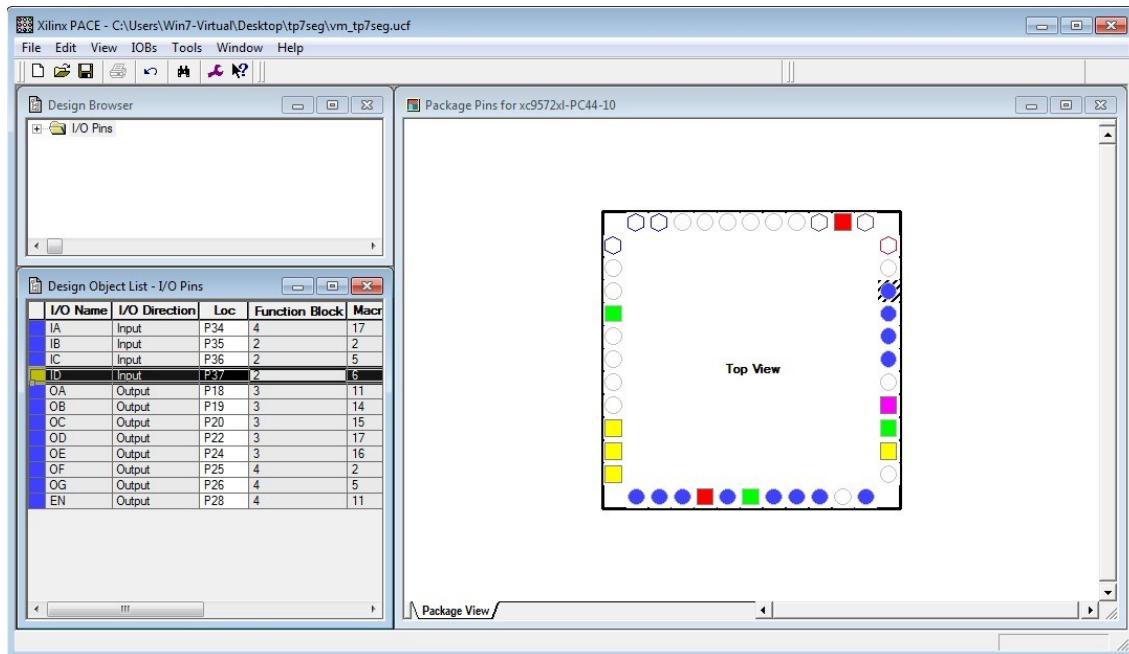
Realizamos el modulo en verilog con las respectivas funciones obtenidas de los mapas de karnaugh:

```

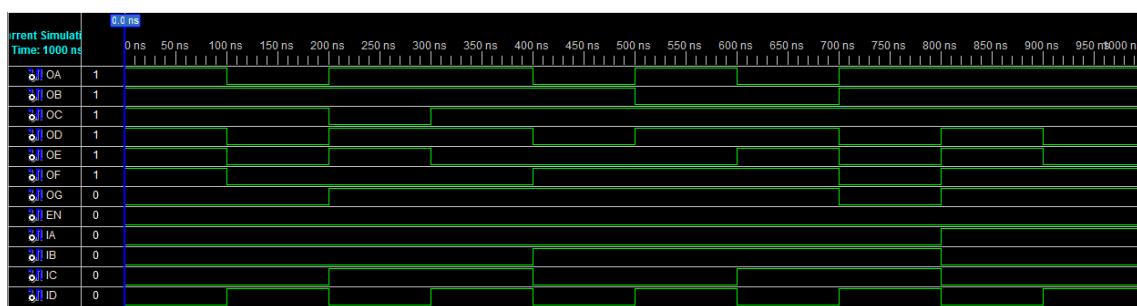
21 module vm_tp7seg(
22     input IA,
23     input IB,
24     input IC,
25     input ID,
26     output OA,
27     output OB,
28     output OC,
29     output OD,
30     output OE,
31     output OF,
32     output OG,
33     output EN
34 );
35
36 assign OA = ~IA&IB&ID | IA&~IB&~IC | ~IA&~IB&IC | ~IA&~IB&~ID;
37 assign OB = ~IA&~IB | ~IB&~IC | ~IA&~IC&~ID | ~IA&IC&ID;
38 assign OC = ~IA & IB | ~IA&~IC | ~IA&ID | ~IB&~IC;
39 assign OD = ~IB&~IC&~ID | ~IA&IC&~ID | ~IA&~IB&IC | ~IA&IB&~IC&ID;
40 assign OE = ~IA&IC&~ID | ~IB&~IC&~ID;
41 assign OF = IA&~IB&~IC | ~IA&IB&~IC | ~IB&~IC&~ID | ~IA&IB&~ID;
42 assign OG = IA&~IB&~IC | ~IA&~IB&IC | ~IA&IB&~IC | ~IA&IC&~ID;
43
44 assign EN=0;
45
46 endmodule

```

A continuación, dispusimos la selección de pines teniendo en cuenta entradas y salidas de datos y una salida “EN” en bajo nivel para encender el transistor que enciende el display del CPLD.



Luego utilizamos el dispositivo de prueba de verilog y simulamos para corroborar que los datos de entrada y salida sean congruentes:



### **Materiales que se utilizaron:**

- -Minilab.
- -CD4011.
- -Kit CPLD.
- -Resistencias.

### **Dificultades/ Inconvenientes:**

Los inconvenientes que surgieron a la hora de realizar este trabajo práctico se dieron a la hora de emplear el kit CPLD, ya que debido a una mala conexión del bus de datos entre la computadora y el kit, el software Xilinx no reconocía la placa de desarrollo. Este inconveniente se solucionó siendo más delicados y presentes a la hora de realizar todas las conexiones.

### **Conclusión:**

Este trabajo práctico nos permitió tener una clara puesta en práctica del kit CPLD, para el cual debimos basarnos en la descripción de hardware previamente realizada. Luego de este proyecto podemos afirmar que contamos con la experiencia para poner en práctica nuestros futuros proyectos que involucren descripción de hardware.