



Facultad de Ingeniería Mecánica y Eléctrica

Actividad Fundamenta 3

Diseño simulación y construcción de un prototipo de un Display de 7 segmentos con mensaje personalizado



El objetivo principal de esta actividad es diseñar, simular e implementar un prototipo de un sistema digital combinacional que funcione como un decodificador de BCD a 7 segmentos con un mensaje personalizado. Para ello, se utilizará un Dispositivo Lógico Programable (PLD) y se programará en el lenguaje ABEL-HDL, utilizando el comando Truth_Table

Fundamento teórico, Definiciones clave

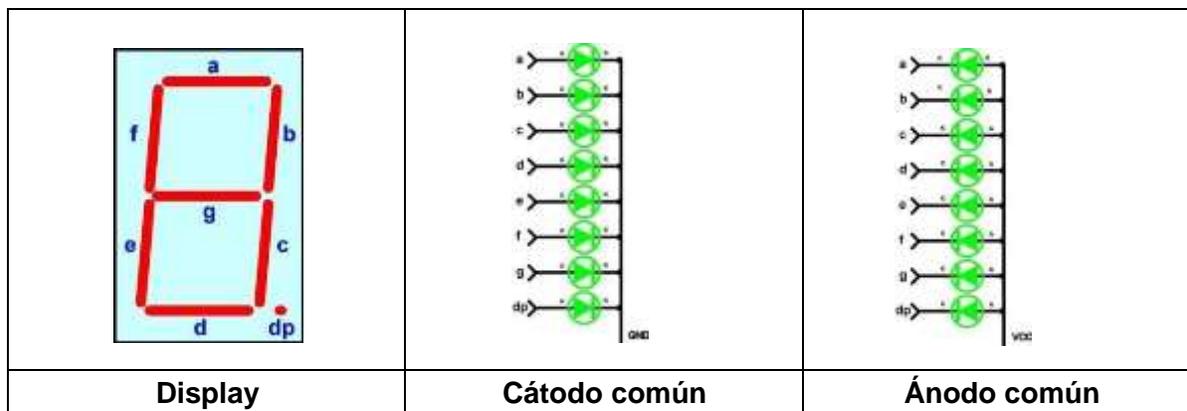
Decodificador: Circuito que convierte un código binario a una representación legible o utilizable por otro sistema, como un Display de 7 segmentos.

BCD (Binary-Coded Decimal): Código numérico en el que cada dígito decimal se representa mediante 4 bits en su equivalente binario. ejemplo el numero $9\ 6\ 7\ 3_{(10)} = 1001\ 0110\ 0111\ 0011_{(BCD)}$

N(10)	9	6	7	3
N(BCD)	1001	0110	0111	0011

y el mismo número expresado en binario sería $9\ 6\ 7\ 3_{(10)} = 0010010111001001_{(2)}$.

7 Segmentos se refiere a un DISPLAY (dispositivo para mostrar resultados) compuesto por LED's (Diodos emisores de luz) distribuidos de tal suerte que se puedan mostrar los dígitos del 0 al9.



Procedimiento:

1.- Elaborar una tabla de verdad para el decodificador de BCD a 7 segmentos que incluya 5 entradas: 4 bits para el código BCD (D, C, B, A) y 1 bit adicional llamado LT (Lamp Test, prueba de lámpara).

Para la entrada LT, se debe cumplir que cuando $LT = 1$, sin importar el valor del código BCD, todas las 8 salidas que controlan los segmentos (a, b, c, d, e, f, g, dp) deben activarse para encender completamente el Display de 7 segmentos, incluyendo el punto decimal. Esto permitirá verificar que todos los segmentos funcionen correctamente y que considerar que Para el caso de Cátodo común:

M	LT	A	B	C	D	a	b	c	d	e	f	g	DP	V ₍₁₀₎
16→31	1	X	X	X	X	1	1	1	1	1	1	1	1	255

Para el caso de Ánodo común:

M	LT	A	B	C	D	a	b	c	d	e	f	g	DP	V ₍₁₀₎
16→31	1	X	X	X	X	0	0	0	0	0	0	0	0	0

Nota: la última columna, denominada V(10), corresponde al valor decimal que se obtiene a partir de los valores binarios de las variables a, b, c, d, e, f, g y DP.

Tabla de verdad

M	BCD					7 segmentos							V ₍₁₀₎	
	LT	A	B	C	D	a	b	c	d	e	f	g		DP
0	0	0	0	0	0									
1	0	0	0	0	1									
2	0	0	0	1	0									
3	0	0	0	1	1									
4	0	0	1	0	0									
5	0	0	1	0	1									
6	0	0	1	1	0									
7	0	0	1	1	1									
8	0	1	0	0	0									
9	0	1	0	0	1									
10	0	1	0	1	0									
11	0	1	0	1	1									
12	0	1	1	0	0									
13	0	1	1	0	1									
14	0	1	1	1	0									
15	0	1	1	1	1									
16, 31	1	X	X	X	X									

2.- En el rango de combinaciones del 10 al 15, se utilizará un modo de texto.

Se deberá seleccionar de una lista proporcionada una palabra de seis letras, una letra por cada combinación. Además, el punto decimal (DP) deberá encenderse junto con cada palabra para indicar que está en modo texto.

La palabra propuesta deberá ser verificada previamente para asegurarse de que no esté registrada en el documento de Excel compartido para asegurar que sea diferente a las propuestas de los demás compañeros.

3.- Identifique las terminales del Display seleccionado.

Terminal		Segmento
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

4.- Elabore el archivo ABL correspondiente al diseño del decodificador solicitado deberá ser elaborado mediante el programa Isp Starter, utilizando el lenguaje Abel-HDL y el comando Truth_Table.

5.- Para su implementación, se deberá considerar la asignación más conveniente de terminales de entrada y salida, evitando cruces en el cableado. La disposición de terminales del Display seleccionado se muestra en la figura 1 y deberá ser considerada en el diseño del decodificador

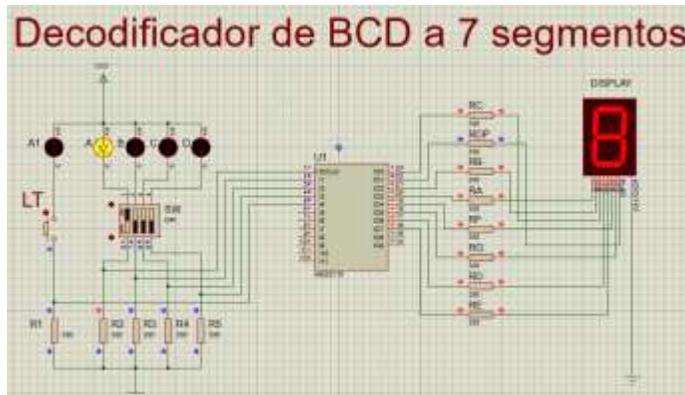


Figura 1: Diagrama de alambrado propuesto para la Simulación en PROTEUS

6.- Para comprobar el correcto funcionamiento se debe realizar la simulación del decodificador utilizando el programa PROTEUS.

7.- Si el funcionamiento del decodificador es correcto, se deberá elaborar una animación en formato gif que muestre cada una de las combinaciones de la simulación de Proteus.

8.- Para obtener información adicional sobre el diseño de PCB BCD 7 segmentos en Proteus, se recomienda consultar el siguiente video:

https://www.youtube.com/watch?v=Mghag9dMuV8&ab_channel=JuanAngelGarzaGarza

9.- Elabore el diseño del PCB en el programa PROTEUS como se muestra en la figura 2.

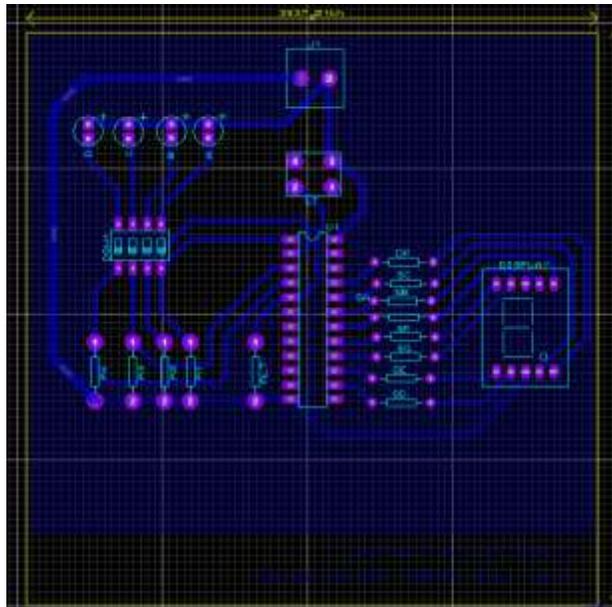


Figura 2: Imagen del Layout del circuito impreso en PROTEUS

10.- Construya el prototipo del decodificador solicitado.

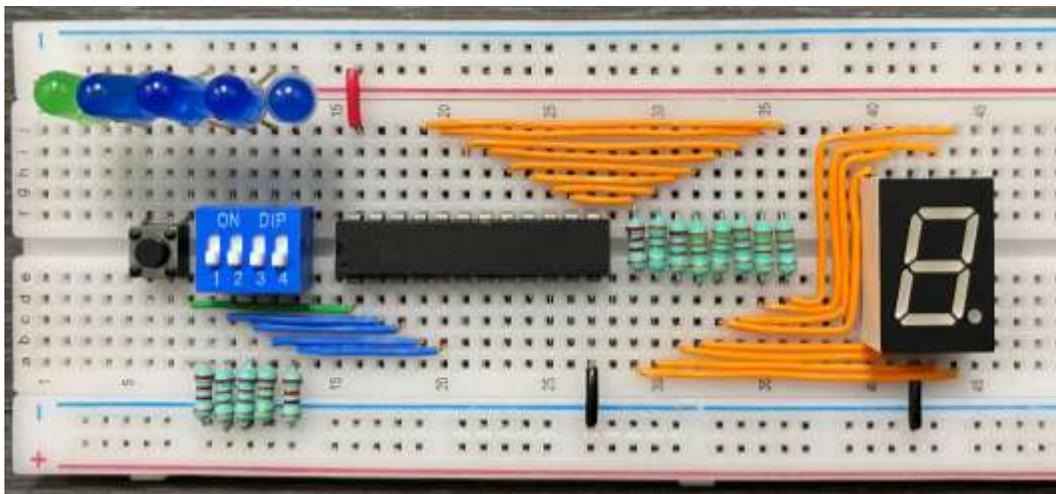


Figura 3: Imagen del Prototipo armado en físico.

11.- Elabore el reporte

Reporte (lista de Cotejo, Check List)

1	Portada.
2	Redacción del Problema (Incluir palabra de seis letras registrada)
3	Diagrama de Bloques (entradas y salidas)
4	Tabla de verdad (como la que se muestra en el inciso a)
5	Archivo del código en formato ABEL-HDL Module.
6	Distribución de terminales (Pin Out)
7	Simulación en Proteus
8	Imagen del Layout del circuito Impreso (incluyendo el plano de tierra)
9	Foto del prototipo implementado
10	Conclusiones
11	Recomendaciones
12	Referencias bibliográficas y hojas del fabricante

Nota: Agregar notas de pie a todas las figuras

Presentación con las siguientes diapositivas:	
1	Portada.
2	Redacción del problema. Incluir palabra de seis letras registrada
3	Código ABEL-HDL
4	Imagen de la distribución de terminales (pin out).
5	Simulación de Proteus en un Gif animado.
6	Imagen del Layout del circuito impreso
7	Foto del prototipo implementado
7	Conclusiones
8	Recomendaciones

Entregables en Google Classroom

Archivos entregables en Zip o RAR nombre= hora y numero de lista	PDF	ABL	JED	Animación	PROTEUS	PPT
--	-----	-----	-----	-----------	---------	-----

Una vez cumplido con lo anterior agende su entrevista



“He fallado una y otra vez en mi vida, por eso he conseguido el éxito.”



Dr. Santos Guzmán Lopez
Rector UANL

Reflexiona sobre tu desempeño en esta evaluación y encontraras áreas de mejora.

¡Sigue adelante en tu formación como ingeniero!



Dr. Arnulfo Treviño Cubero
Director FIME