

Diseño Combinacional



La peor decisión es la indecisión.
Benjamín Franklin

	Actividad	Puntos	Fecha límite
PF6	Solución del examen	F	
PF7	Diseño Combinacional con HDL	F	
PF8	Multiplexor	5	
AF3	Decodificador con Display	10	
PF9	Flip Flops	F	
PF10	Pulsos de sincronía	5	
AF4	Diseño Secuencial	10	
AF5	PIA	40	

Codificador o Encoder

Es un circuito que convierte una señal de entrada en un código específico de bits



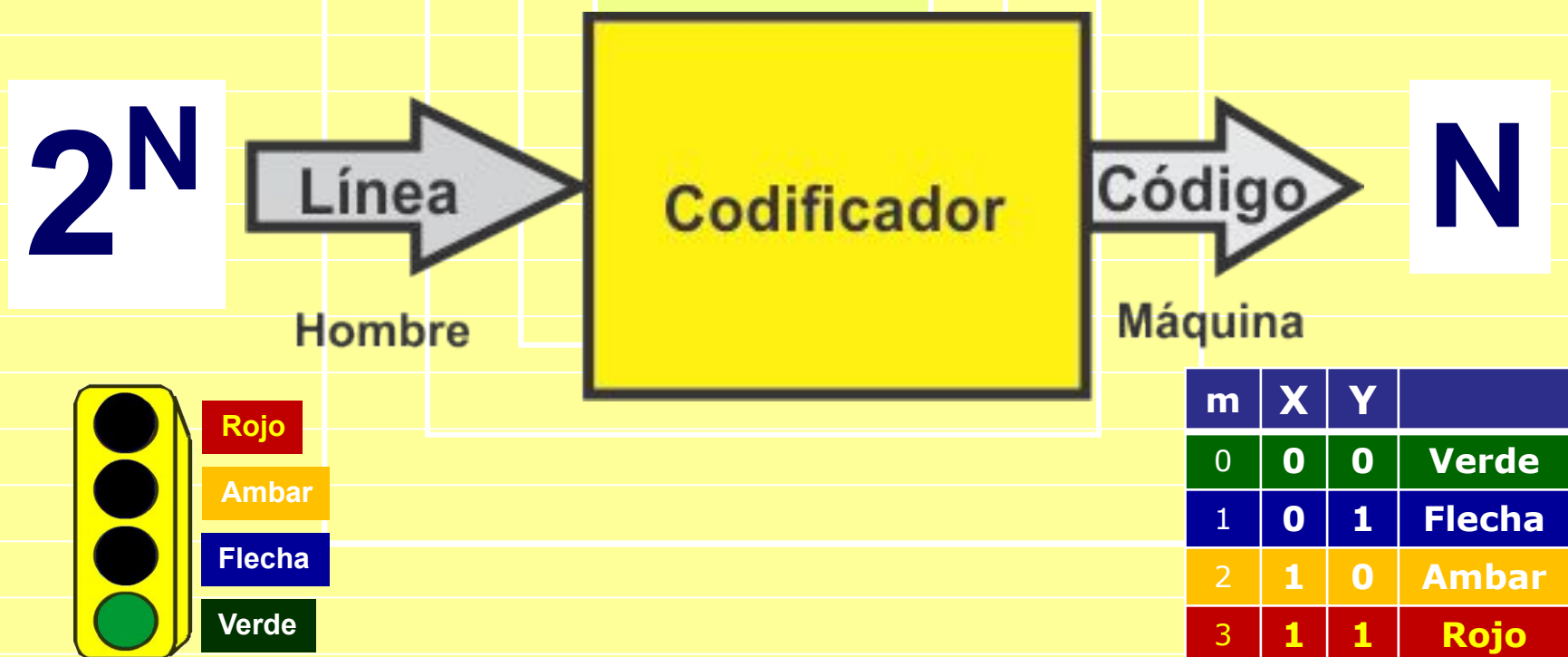
Decodificador o Decoder

Es un circuito que realiza la operación inversa, convirtiendo un código de bits en una señal de salida.



Codificador Encoder

Es un circuito combinacional con 2^N entradas y N salidas.
Su salida es el código binario correspondiente a la entrada activada.



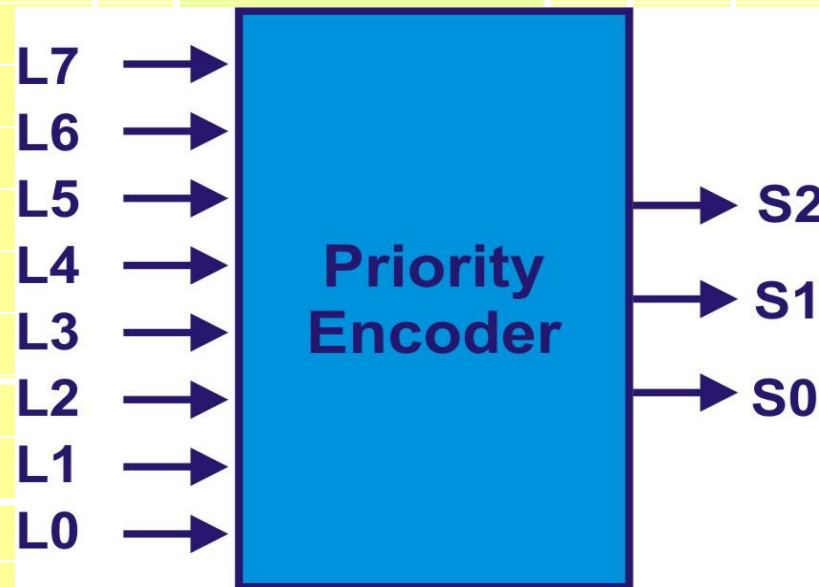
Codificador de prioridad (Priority Encoder)

Definiciones:

- ***Codificador*** Proceso en el cual se obtiene un *código* para cada una de las posibles entradas.
- ***Código*** *Combinación de signos que tiene un determinado valor dentro de un sistema establecido.*
- ***Prioridad*** Anterioridad de algo respecto de otra cosa, en tiempo, en orden.

Codificador de prioridad (Priority Encoder)

**Diseñe un Codificador de prioridad
de 8 líneas de entrada a 3 bit ´s de salida.**



Codificador de prioridad (Priority Encoder)

Las salidas (S2, S1, S0) debe de ser el número en código binario correspondiente a la línea de entrada de mayor valor que contenga un uno.

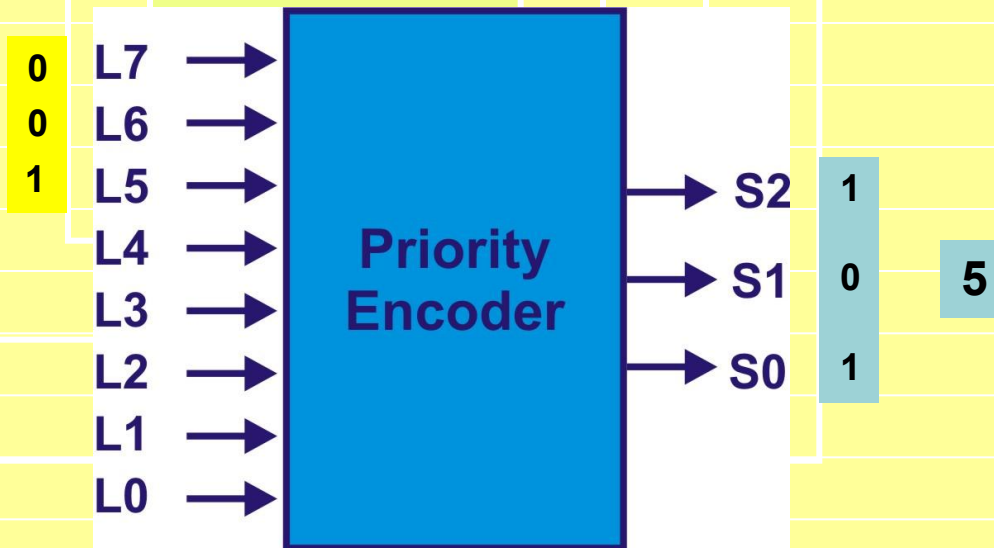


Diagrama de bloques

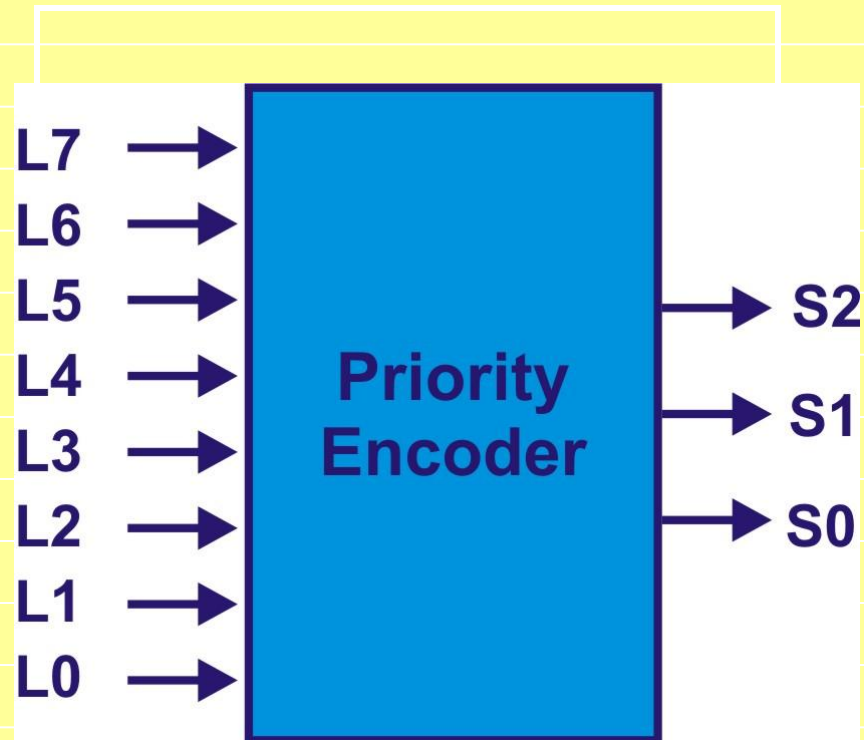


Diagrama de bloques

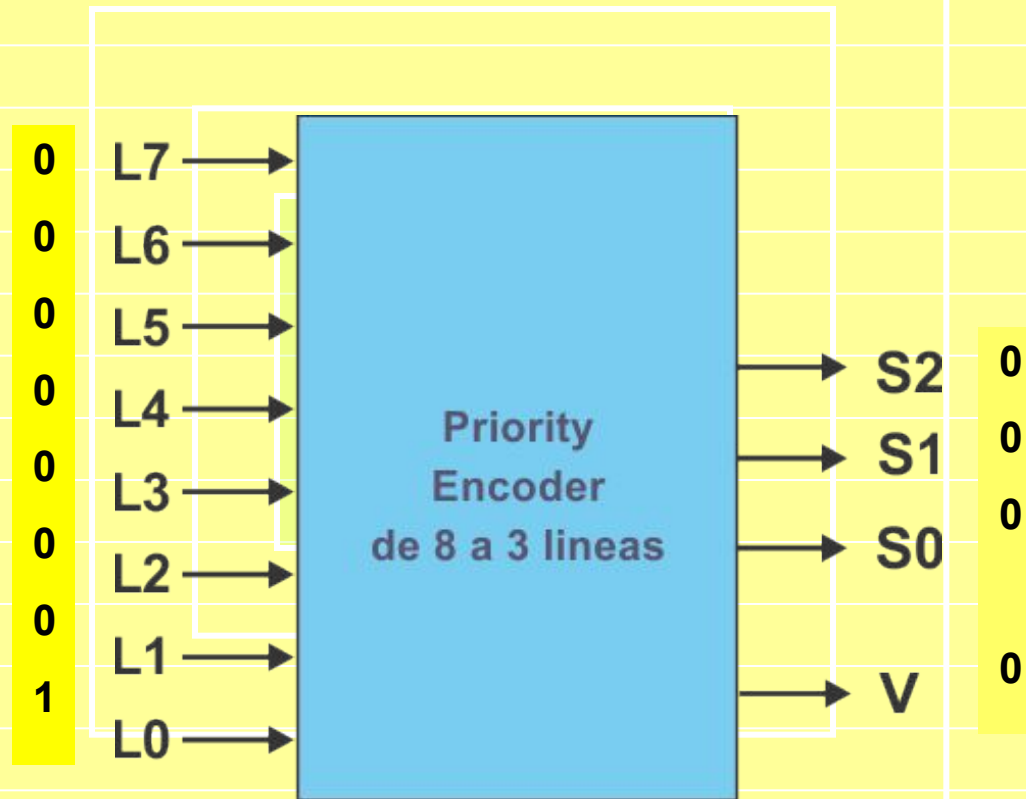


Diagrama de bloques

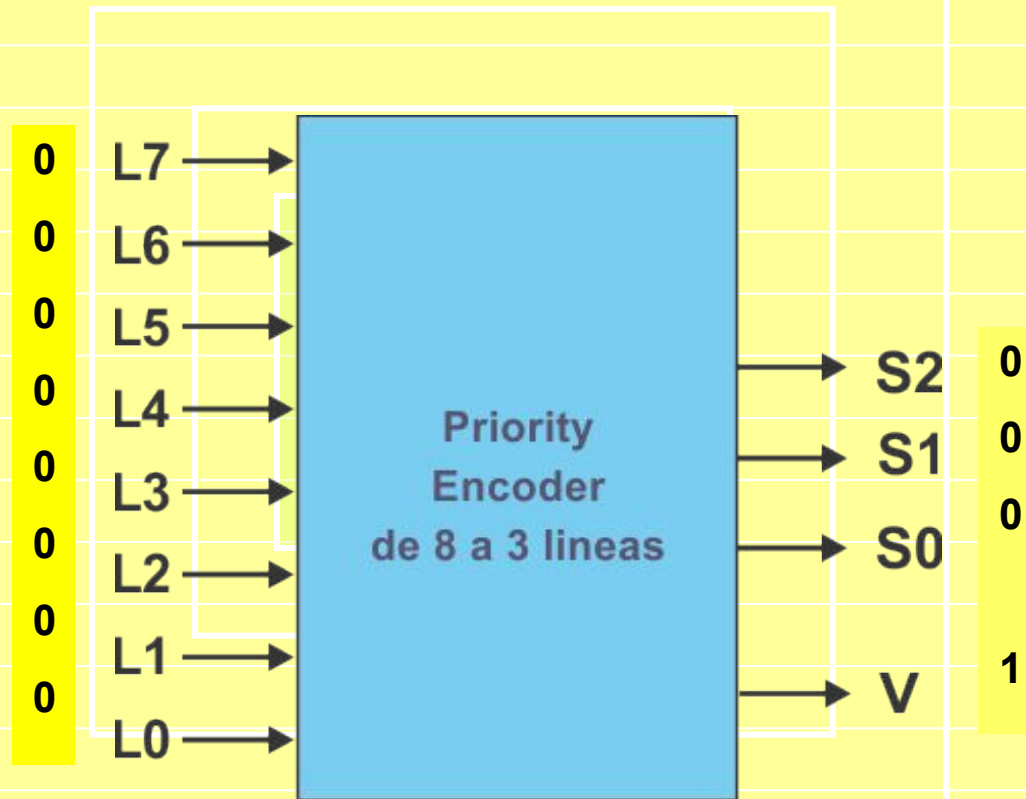


Tabla de Verdad

Con ocho entradas se tienen 256 posibles combinaciones.

En este ejemplo no es necesario listarlas todas ya que gracias al Don't care (.x.).

En las combinaciones de entrada, se puede representar el comportamiento del codificador de prioridad con solo 9 combinaciones como lo muestra la siguiente tabla.

Tabla de Verdad

Entradas								Salidas			
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	V

Tabla de Verdad

Entradas								Salidas			
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	V
1											

Tabla de Verdad

Entradas								Salidas			
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	V
1	X	X	X	X	X	X	X				

Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	$N_{(10)}$	V
1	X	X	X	X	X	X	X	1	1	1	7	0

Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	N ₍₁₀₎	V
1	X	X	X	X	X	X	X	1	1	1	7	0
0	1	X	X	X	X	X	X	1	1	0	6	0

Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	$N_{(10)}$	V
1	X	X	X	X	X	X	X	1	1	1	7	0
0	1	X	X	X	X	X	X	1	1	0	6	0
0	0	1	X	X	X	X	X	1	0	1	5	0

Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	N ₍₁₀₎	V
1	X	X	X	X	X	X	X	1	1	1	7	0
0	1	X	X	X	X	X	X	1	1	0	6	0
0	0	1	X	X	X	X	X	1	0	1	5	0
0	0	0	1	X	X	X	X	1	0	0	4	0
0	0	0	0	1	X	X	X	0	1	1	3	0

Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	N ₍₁₀₎	V
1	X	X	X	X	X	X	X	1	1	1	7	0
0	1	X	X	X	X	X	X	1	1	0	6	0
0	0	1	X	X	X	X	X	1	0	1	5	0
0	0	0	1	X	X	X	X	1	0	0	4	0
0	0	0	0	1	X	X	X	0	1	1	3	0
0	0	0	0	0	1	X	X	0	1	0	2	0
0	0	0	0	0	0	1	X	0	0	1	1	0

Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	N ₍₁₀₎	V
1	X	X	X	X	X	X	X	1	1	1	7	0
0	1	X	X	X	X	X	X	1	1	0	6	0
0	0	1	X	X	X	X	X	1	0	1	5	0
0	0	0	1	X	X	X	X	1	0	0	4	0
0	0	0	0	1	X	X	X	0	1	1	3	0
0	0	0	0	0	1	X	X	0	1	0	2	0
0	0	0	0	0	0	1	X	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0

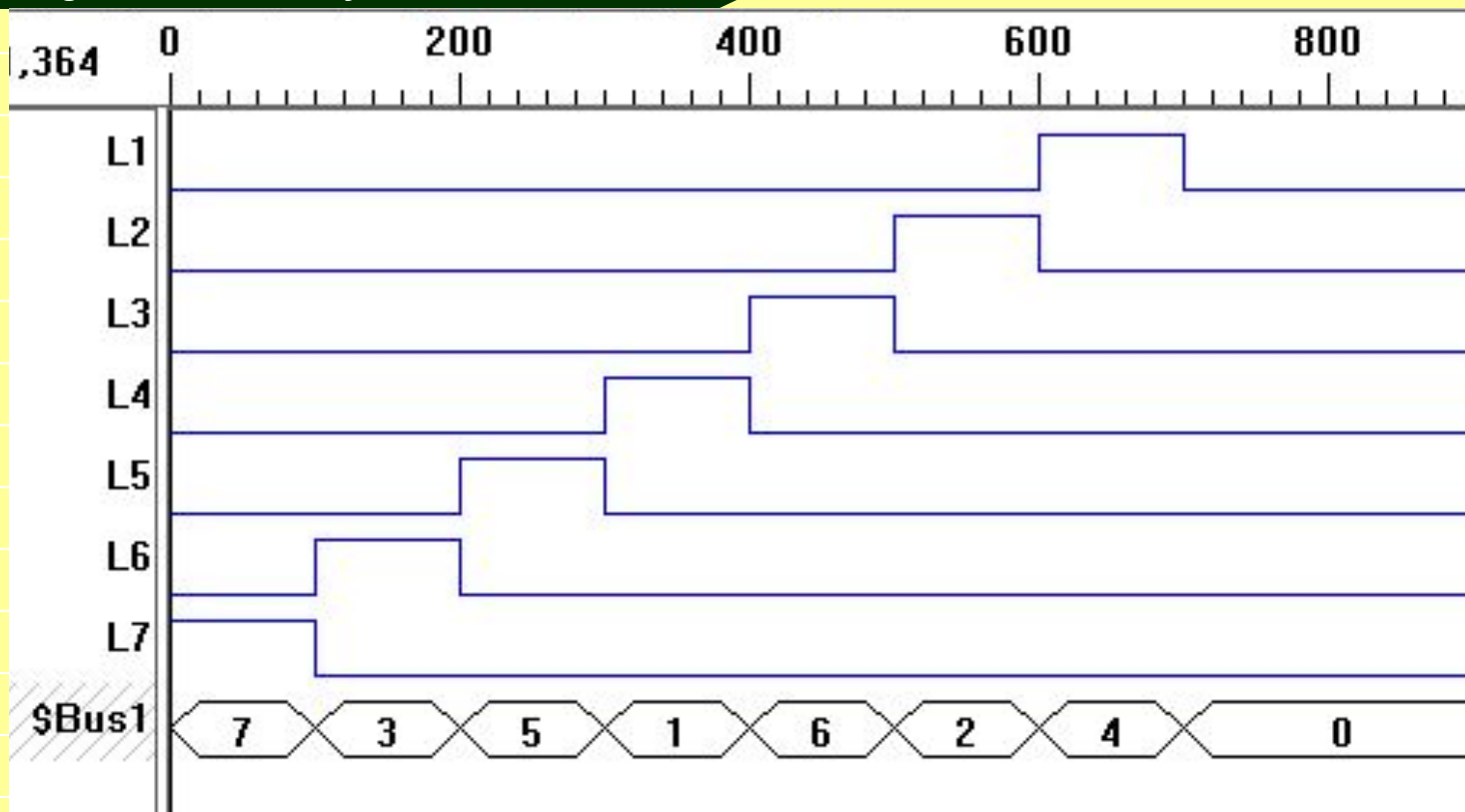
Tabla de Verdad

Entradas								Salidas				
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0	N ₍₁₀₎	V
1	X	X	X	X	X	X	X	1	1	1	7	0
0	1	X	X	X	X	X	X	1	1	0	6	0
0	0	1	X	X	X	X	X	1	0	1	5	0
0	0	0	1	X	X	X	X	1	0	0	4	0
0	0	0	0	1	X	X	X	0	1	1	3	0
0	0	0	0	0	1	X	X	0	1	0	2	0
0	0	0	0	0	0	1	X	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0		1

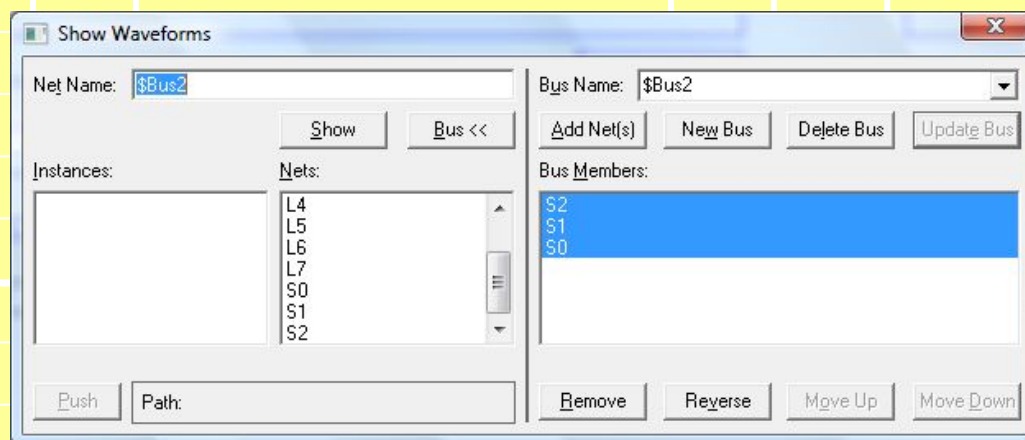
Archivo en ABEL-HDL

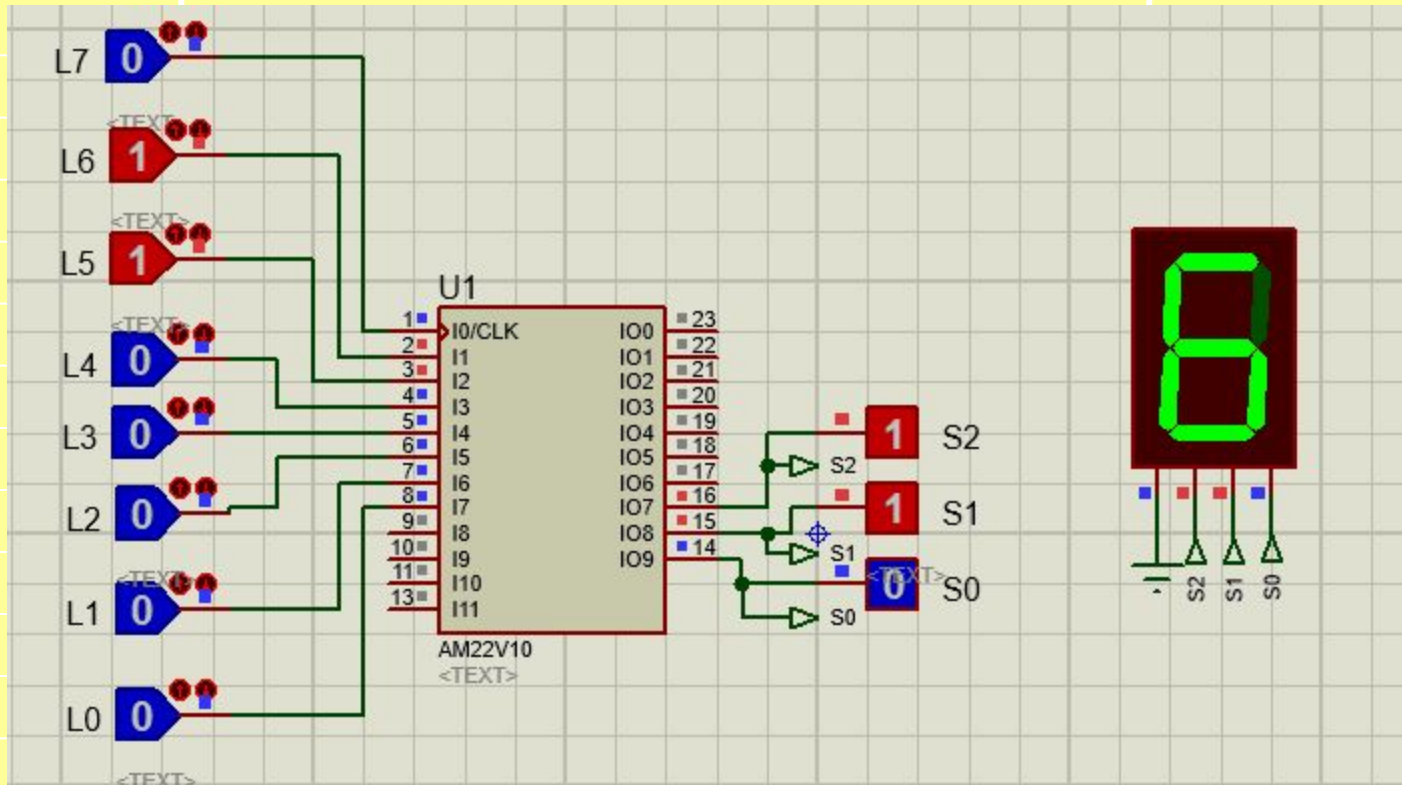
```
MODULE Penco
declarations
X=.x.;
"Entradas
L7..L0 PIN 1..8;
"salidas
S2..S0,V PIN 17..14 istype 'dc,com';
"SET
S=[S2,S1,S0];
"Tabla de verdad
truth_table
([L7,L6,L5,L4,L3,L2,L1,L0]->[S,V])
[1,X,X,X,X,X,X,X]->[7,0];
[0,1,X,X,X,X,X,X]->[6,0];
[0,0,1,X,X,X,X,X]->[5,0];
[0,0,0,1,X,X,X,X]->[4,0];
[0,0,0,0,1,X,X,X]->[3,0];
[0,0,0,0,0,1,X,X]->[2,0];
[0,0,0,0,0,0,1,X]->[1,0];
[0,0,0,0,0,0,0,1]->[0,0];
[0,0,0,0,0,0,0,0]->[0,1];
```

```
test_vectors
([L7,L6,L5,L4,L3,L2,L1,L0]->[
S2,S1,S0,V])
[1,X,X,X,X,X,X,X]->[X,X,X,X];
[0,1,X,X,X,X,X,X]->[X,X,X,X];
[0,0,1,X,X,X,X,X]->[X,X,X,X];
[0,0,0,1,X,X,X,X]->[X,X,X,X];
[0,0,0,0,1,X,X,X]->[X,X,X,X];
[0,0,0,0,0,1,X,X]->[X,X,X,X];
[0,0,0,0,0,0,1,X]->[X,X,X,X];
[0,0,0,0,0,0,0,1]->[X,X,X,X];
[0,0,0,0,0,0,0,0]->[X,X,X,X];
END ;
```



\$Bus1=S2,S1;S0





Decodificadores



Los libros son las abejas que llevan el polen de una inteligencia a otra.
James Russell Lowell (1819-1891) Poeta y escritor estadounidense.

Decodificador Decoder

Convierte un código binario de entrada de N bits de entrada a M líneas de salida.

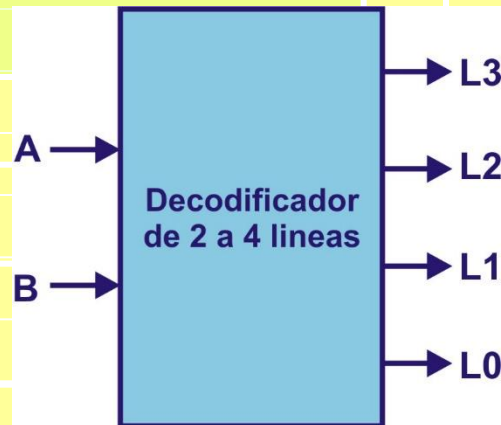


Decodificador de 2 a 4 líneas

Decodificador:

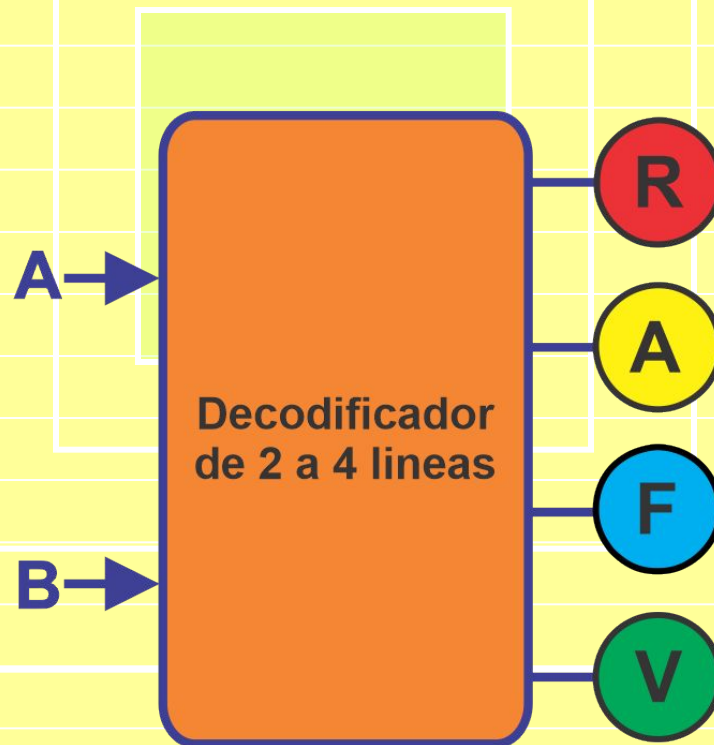
Proceso que permite pasar de un lenguaje codificado (código) a otro legible directamente (línea).

Diagrama de bloques



Decodificador de 2 a 4 líneas

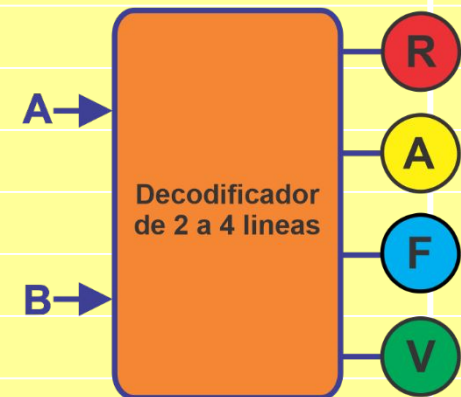
El valor binario de A, B de entrada deberá seleccionar la línea de salida por medio de un nivel alto (uno).



Decodificador de 2 a 4 líneas

Tabla de verdad

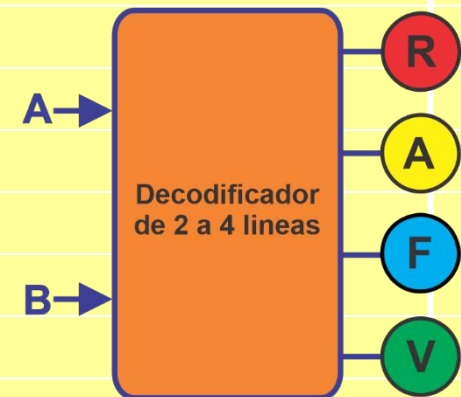
m	Entradas		Salidas			
	A	B	V	F	A	R
0	0	0				
1	0	1				
2	1	0				
3	1	1				



Decodificador de 2 a 4 líneas

Tabla de verdad

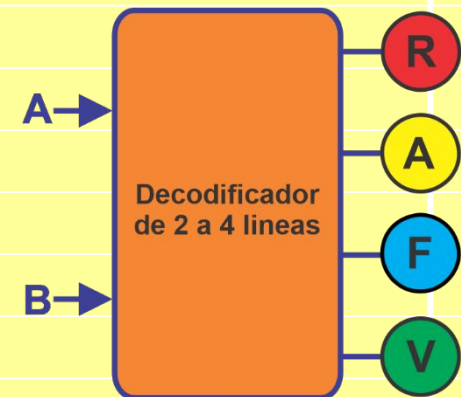
m	Entradas		Salidas			
	A	B	V	F	A	R
0	0	0	0	0	0	1
1	0	1				
2	1	0				
3	1	1				



Decodificador de 2 a 4 líneas

Tabla de verdad

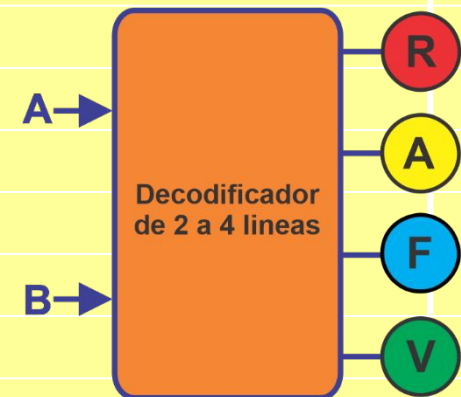
m	Entradas		Salidas			
	A	B	V	F	A	R
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0				
3	1	1				



Decodificador de 2 a 4 líneas

Tabla de verdad

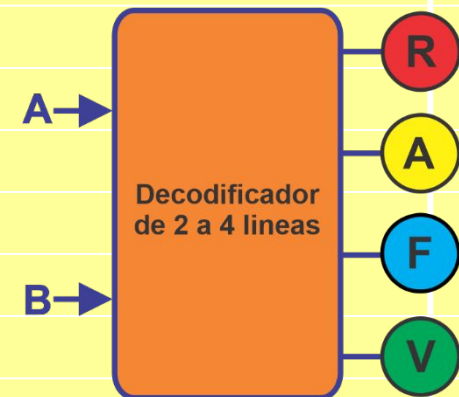
m	Entradas		Salidas			
	A	B	V	F	A	R
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1				



Decodificador de 2 a 4 líneas

Tabla de verdad

m	Entradas		Salidas			
	A	B	V	F	A	R
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0



Decodificador de 2 a 4 líneas

Ecuaciones mínimas

	Entradas		Salidas			
m	A	B	V	F	A	R
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

$$V_{(A,B)} = A' B'$$

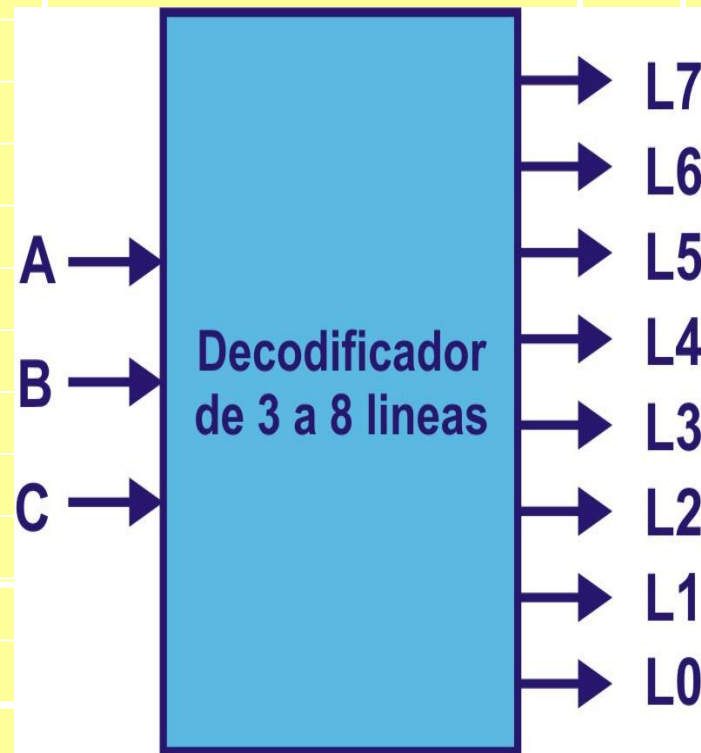
$$F_{(A,B)} = A' B$$

$$A_{(A,B)} = A B'$$

$$R_{(A,B)} = A B$$

Decodificador de 3 a 8 líneas

Diagrama de bloques



Decodificador de 3 a 8 líneas

Tabla de Verdad

M	A	B	C	L7	L6	L5	L4	L3	L2	L1	L0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1								
2	0	1	0								
3	0	1	1								
4	1	0	0								
5	1	0	1								
6	1	1	0								
7	1	1	1								

Decodificador de 3 a 8 líneas

Tabla de Verdad

M	A	B	C	L7	L6	L5	L4	L3	L2	L1	L0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0								
3	0	1	1								
4	1	0	0								
5	1	0	1								
6	1	1	0								
7	1	1	1								

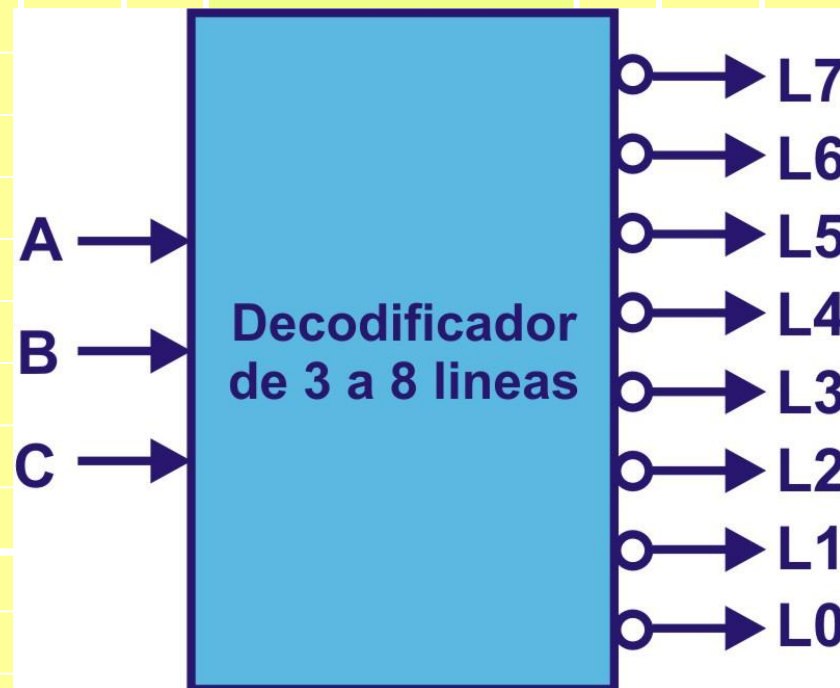
Decodificador de 3 a 8 líneas

Tabla de Verdad

M	A	B	C	L7	L6	L5	L4	L3	L2	L1	L0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

Decodificador de 3 a 8 líneas nivel activo bajo

Diagrama de bloques



Decodificador de 3 a 8 líneas nivel activo bajo

Tabla de Verdad

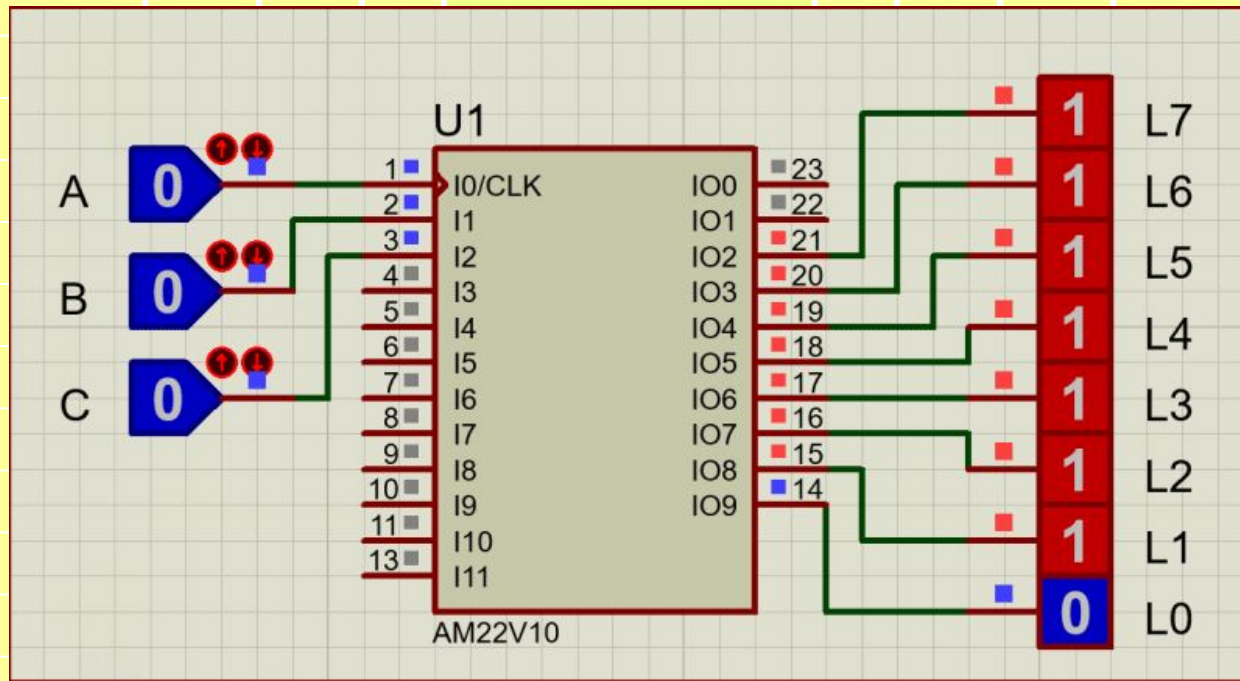
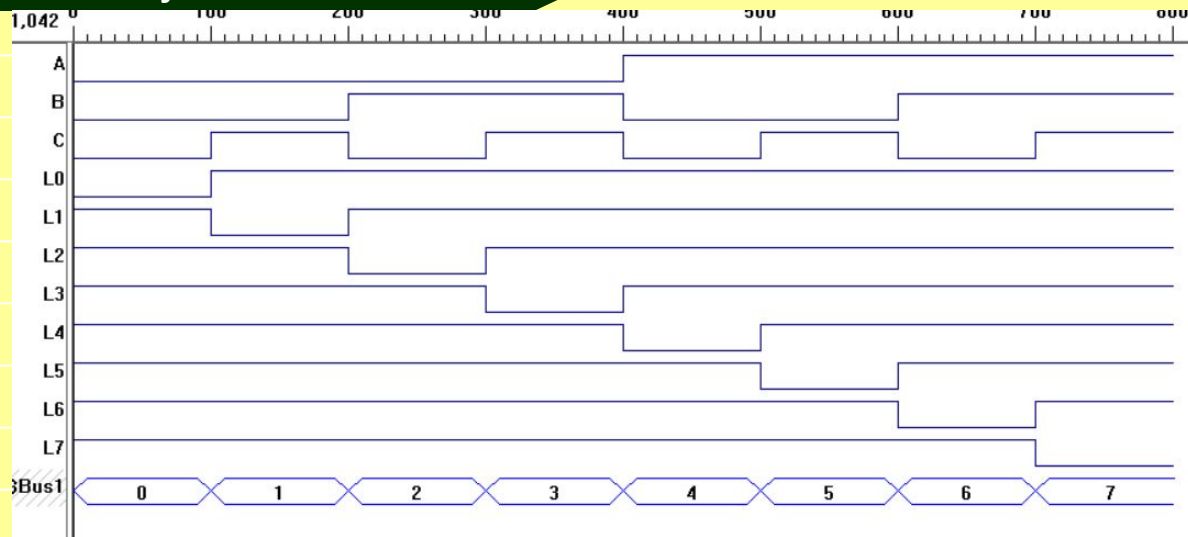
M	A	B	C	L7	L6	L5	L4	L3	L2	L1	L0
0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
2	0	1	0	1	1	1	1	1	0	1	1
3	0	1	1	1	1	1	1	0	1	1	1
4	1	0	0	1	1	1	0	1	1	1	1
5	1	0	1	1	1	0	1	1	1	1	1
6	1	1	0	1	0	1	1	1	1	1	1
7	1	1	1	0	1	1	1	1	1	1	1

Decodificador de 3 a 8 líneas nivel activo bajo

```
MODULE deco
A,B,C PIN 1..3;
L0..L7      PIN      14..21      ISTYPE
'COM';TRUTH_TABLE
([A,B,C]->[L0,L1,L2,L3,L4,L5,L6,L7])
[0,0,0]->[0,1,1,1,1,1,1,1];
[0,0,1]->[1,0,1,1,1,1,1,1];
[0,1,0]->[1,1,0,1,1,1,1,1];
[0,1,1]->[1,1,1,0,1,1,1,1];
[1,0,0]->[1,1,1,1,0,1,1,1];
[1,0,1]->[1,1,1,1,1,0,1,1];
[1,1,0]->[1,1,1,1,1,1,0,1];
[1,1,1]->[1,1,1,1,1,1,1,0];
```

```
TEST_VECTORS
([A,B,C]->[L0,L1,L2,L3,L4,L5,L6,L7])
[0,0,0]->[0,1,1,1,1,1,1,1];
[0,0,1]->[1,0,1,1,1,1,1,1];
[0,1,0]->[1,1,0,1,1,1,1,1];
[0,1,1]->[1,1,1,0,1,1,1,1];
[1,0,0]->[1,1,1,1,0,1,1,1];
[1,0,1]->[1,1,1,1,1,0,1,1];
[1,1,0]->[1,1,1,1,1,1,0,1];
[1,1,1]->[1,1,1,1,1,1,1,0];
```

END

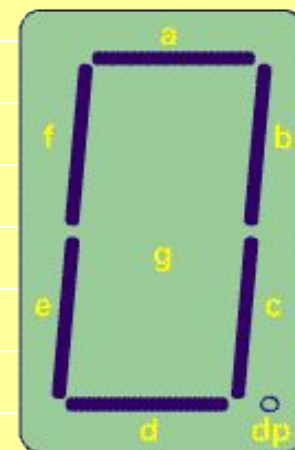


Decodificador de BCD a siete Segmentos

BCD Código Decimal (0 – 9) expresado en Binario (0000 -1001), cada dígito del decimal se representa por cuatro bits ejemplo **4678**₍₁₀₎ **0100 0110 0111 1000**_(BCD)

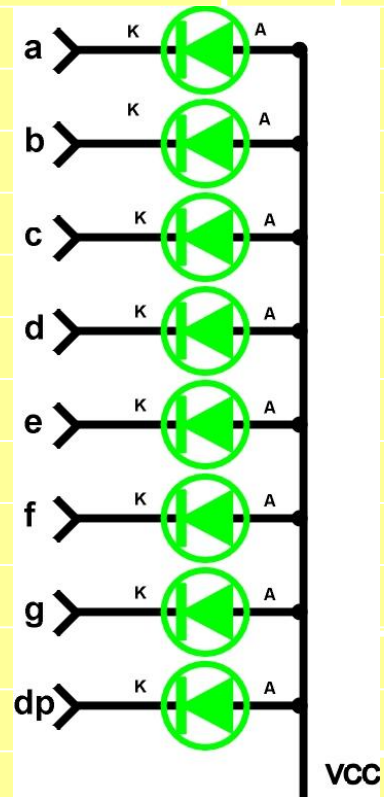
$$\mathbf{4678}_{(10)} = \mathbf{1001001000110}_{(2)}$$

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 al 9.

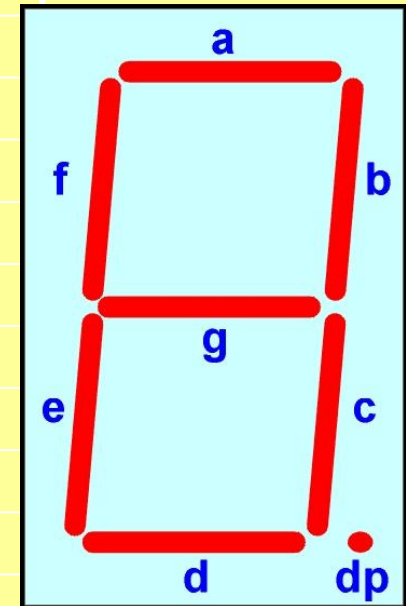
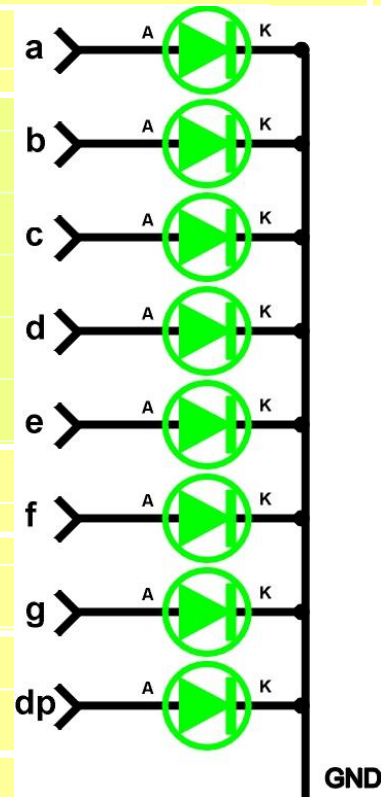


Display de 7 segmentos

Ánodo Común



Cátodo Común



Enciende con un cero

Enciende con un uno

Identificación de terminales del Diaplay

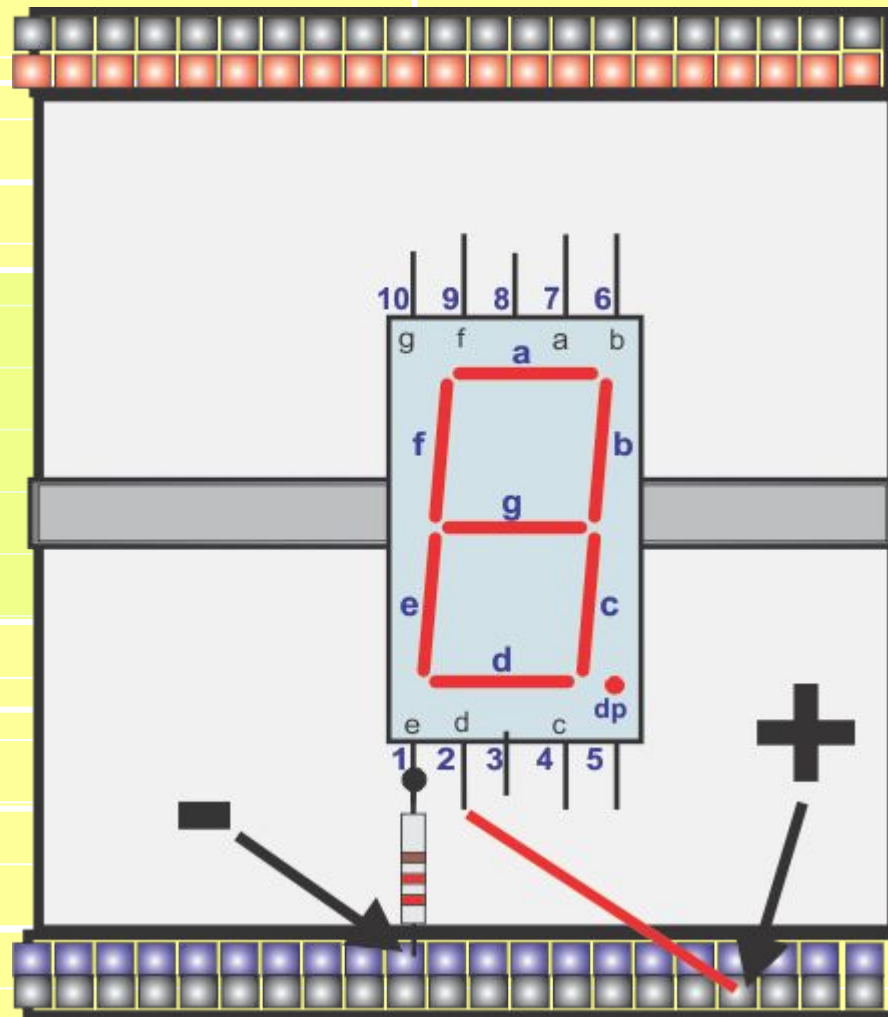
Identificar las terminales comunes

1.- colocar el display en la
tablilla de conexiones

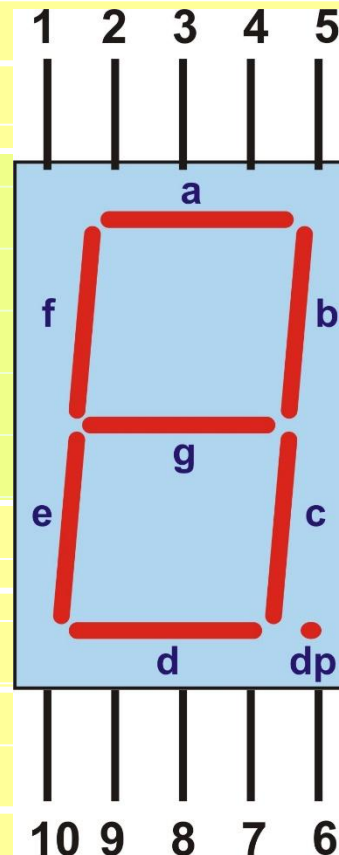
2.- Una de las terminales del display
conectarla al negativo por medio
De un resistor.

3.- con un alambre conectado al
positivo recorrer las demás
Terminales del display e identificar
Con cuales enciende un segmento

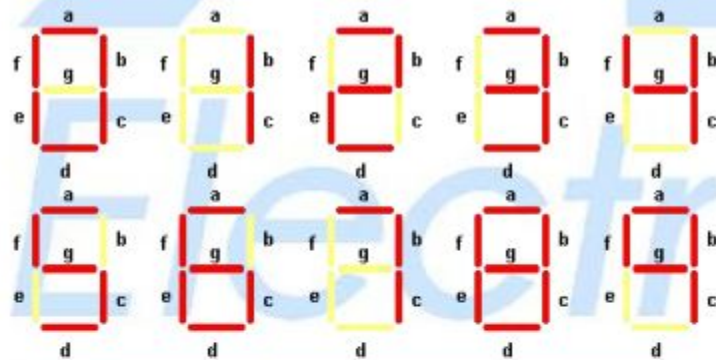
4.- en caso de que no encienda ningún
Segmento, invertir la
Polaridad de la fuente y repetir
El paso 3



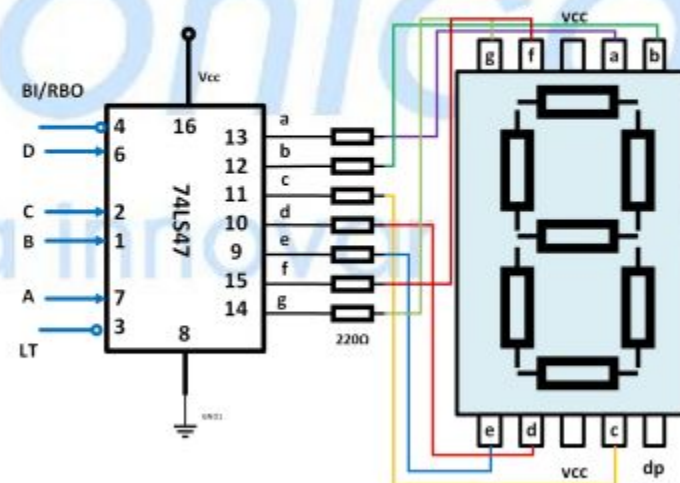
1	g
2	f
3	Común
4	A
5	b
6	dp
7	c
8	Común
9	d
10	e

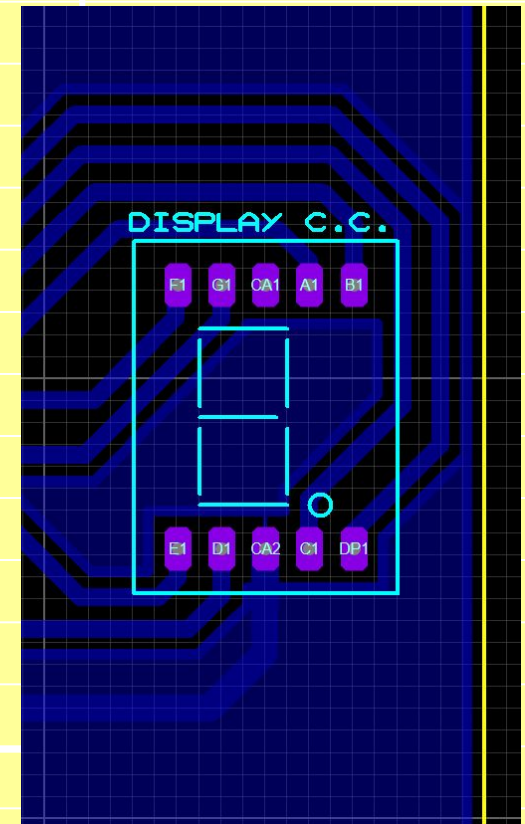
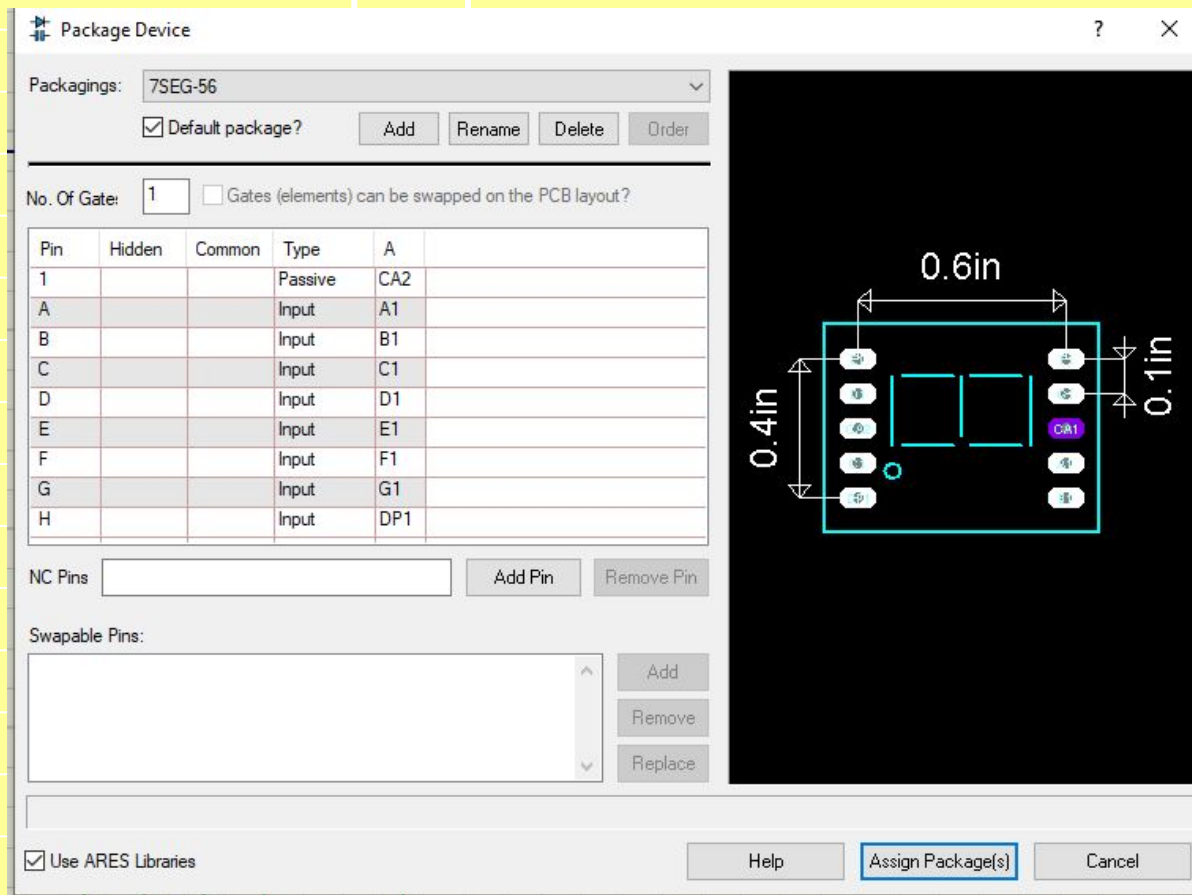


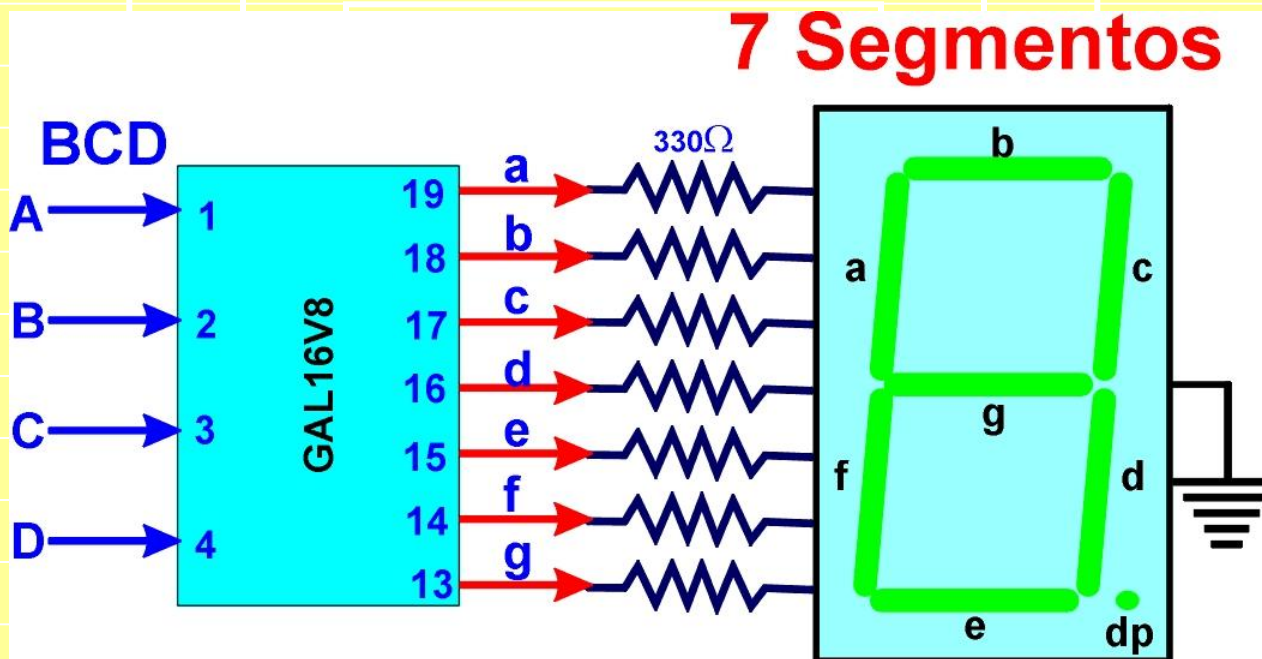
<https://agelectronica.lat/pdfs/textos/G/GM1-5611AURB.PDF>

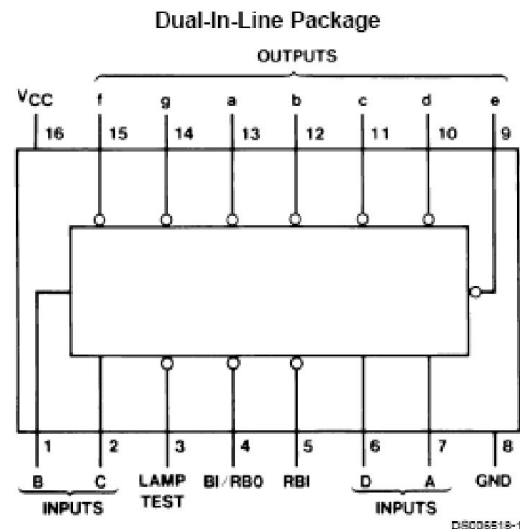


ENTRADAS				SALIDAS							NUMERO
A3	A2	A1	A0	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9



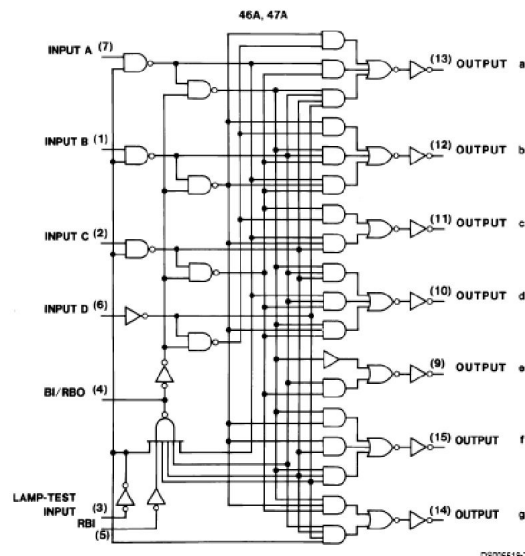






Order Number DM5447AJ, DM7446AN or DM7447AN
See Package Number J16A or N16E

Logic Diagram



Function Table

46A, 47A

Decimal or Function	Inputs						BI/RBO (Note 6)	Outputs							Note
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	(Note 7)
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H	(Note 8)
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	(Note 9)
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	(Note 10)

H = High level, L = Low level, X = Don't Care

Note 6: BI/RBO is a wire-AND logic serving as blanking input (BI) and/or ripple-blanking output (RBO).

Note 7: The blanking input (BI) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input (RBI) must be open or high if blanking of a decimal zero is not desired.

Note 8: When a low logic level is applied directly to the blanking input (BI), all segment outputs are high regardless of the level of any other input.

Note 9: When ripple-blanking input (RBI) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go H and the ripple-blanking output (RBO) goes to a low level (response condition).

Note 10: When the blanking input/ripple-blanking output (BI/RBO) is open or held high and a low is applied to the lamp-test input, all segment outputs are L.

Decodificador de BCD a siete Segmentos

Diagrama de bloques

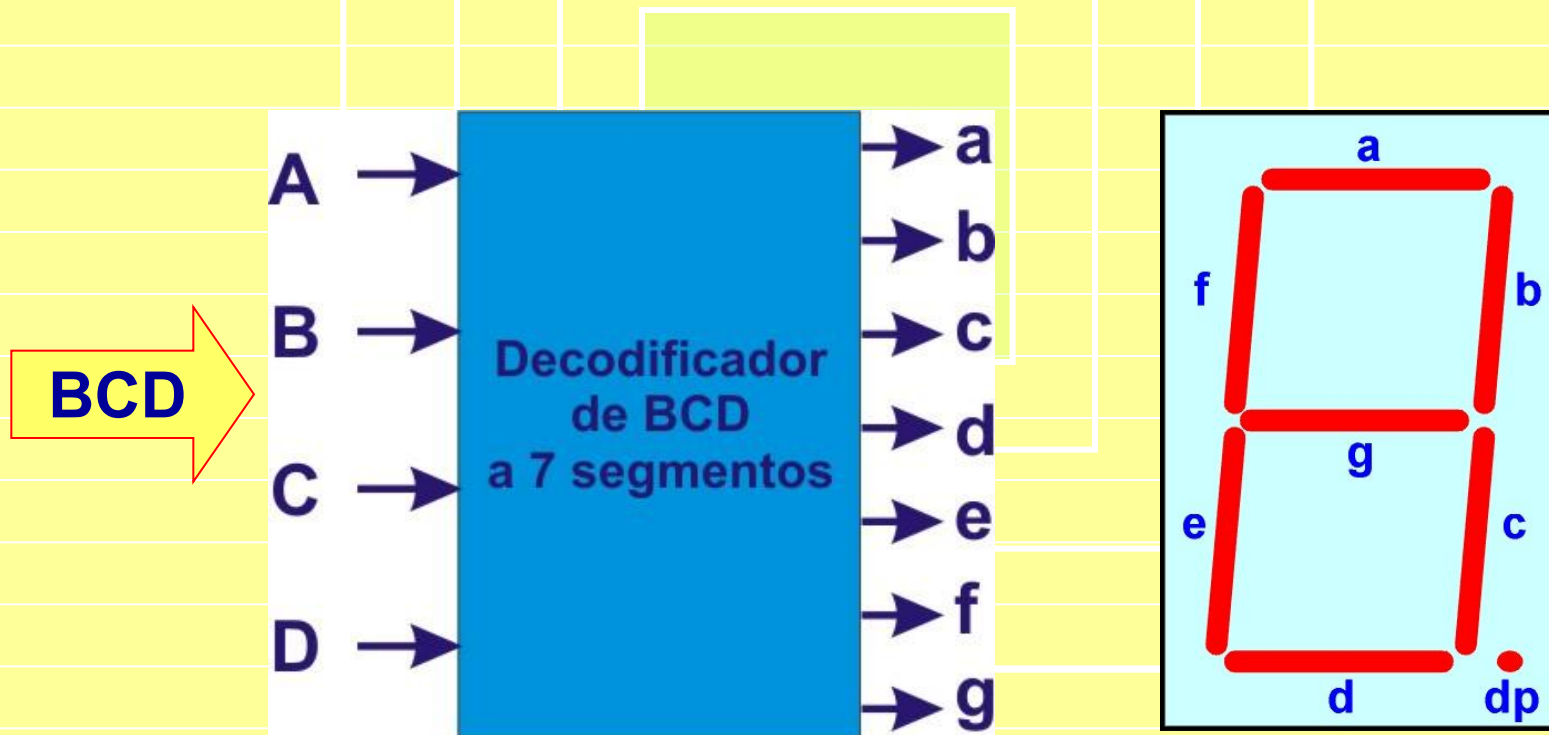
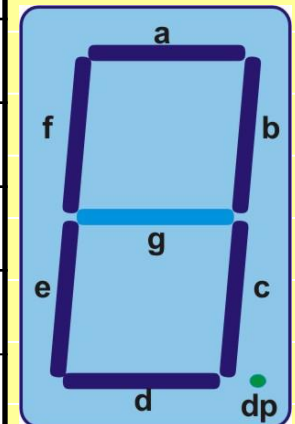
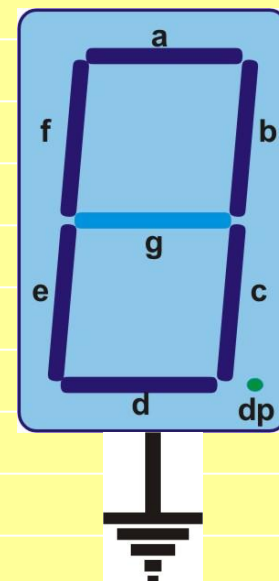


Tabla de Verdad

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0							
1	0 0 0 1							
2	0 0 1 0							
3	0 0 1 1							
4	0 1 0 0							
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



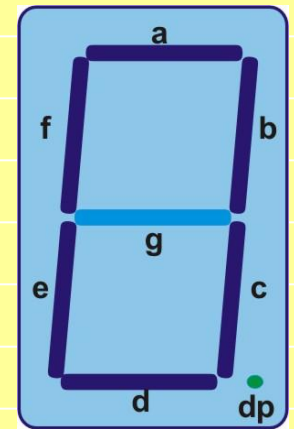
m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1							
2	0 0 1 0							
3	0 0 1 1							
4	0 1 0 0							
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

**Tabla de
Verdad**

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	0	0	0	0	0	0	1
1	0 0 0 1							
2	0 0 1 0							
3	0 0 1 1							
4	0 1 0 0							
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							

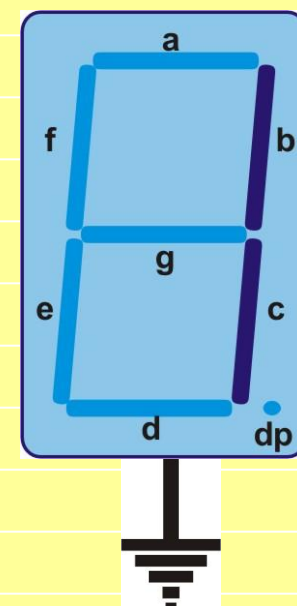


+5 VCD

Considerando
Ánodo Común

**Tabla de
Verdad**

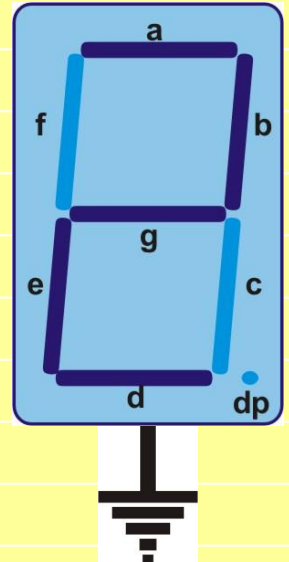
m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0							
3	0 0 1 1							
4	0 1 0 0							
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

**Tabla de
Verdad**

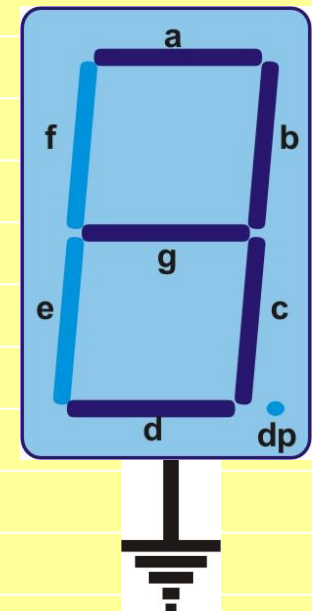
m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1							
4	0 1 0 0							
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

**Tabla de
Verdad**

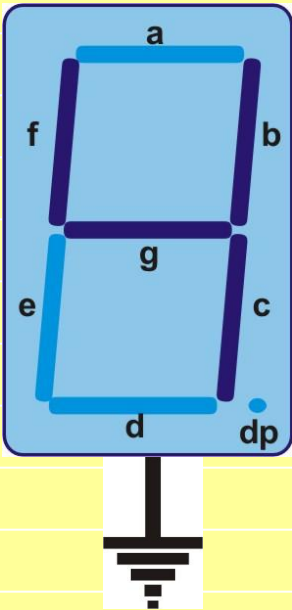
m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0							
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

**Tabla de
Verdad**

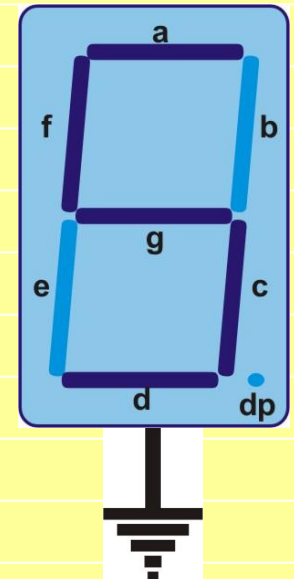
m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1							
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

Tabla de
Verdad

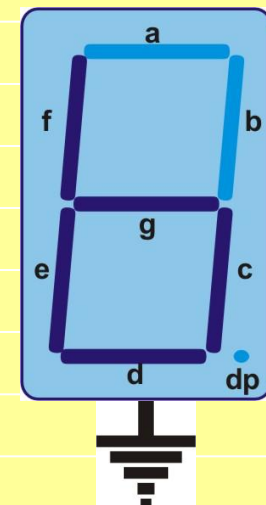
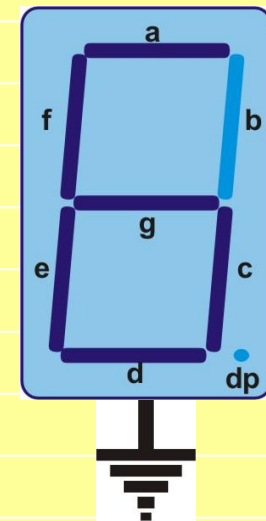
m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0							
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

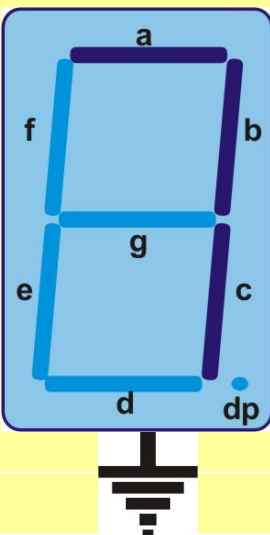
**Tabla de
Verdad**

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	X	0	1	1	1	1	1
7	0 1 1 1							
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



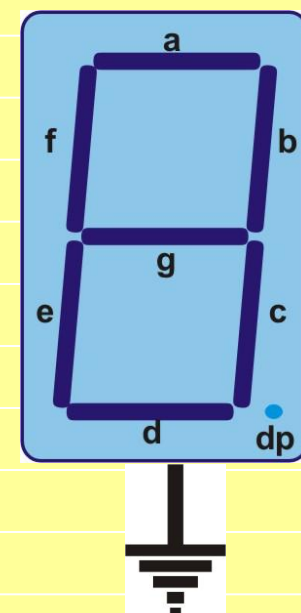
Considerando
Cátodo Común

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	X	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0							
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



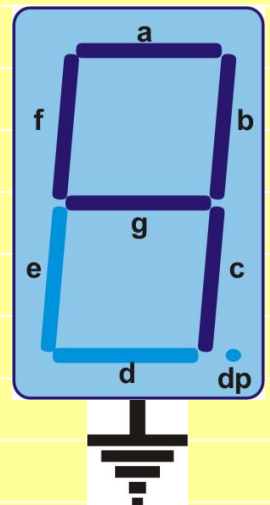
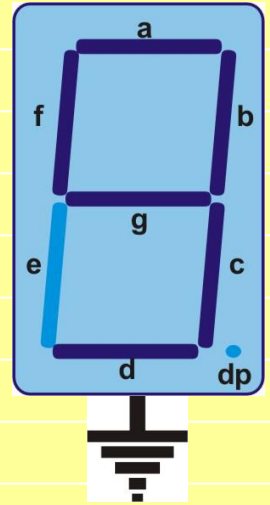
Considerando
Cátodo Común

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	X	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1							
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	X	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1	1	1	1	X	0	1	1
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							



Considerando
Cátodo Común

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	X	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1	1	1	1	X	0	1	1
10	1 0 1 0							
11	1 0 1 1							
12	1 1 0 0							
13	1 1 0 1							
14	1 1 1 0							
15	1 1 1 1							

Código BCD
0 a 9

m	A B C D	a	b	c	d	e	f	g	N ₍₁₀₎	N ₍₁₆₎
0	0 0 0 0	1	1	1	1	1	1	0	126	7E
1	0 0 0 1	0	1	1	0	0	0	0	48	30
2	0 0 1 0	1	1	0	1	1	0	1	109	6D
3	0 0 1 1	1	1	1	1	0	0	1	121	79
4	0 1 0 0	0	1	1	0	0	1	1	51	33
5	0 1 0 1	1	0	1	1	0	1	1	91	5B
6	0 1 1 0	X	0	1	1	1	1	1		
7	0 1 1 1	1	1	1	0	0	0	0	112	70
8	1 0 0 0	1	1	1	1	1	1	1	127	7F
9	1 0 0 1	1	1	1	X	0	1	1	123	7B
10	1 0 1 0	X	X	X	X	X	X	X		
11	1 0 1 1	X	X	X	X	X	X	X		
12	1 1 0 0	X	X	X	X	X	X	X		
13	1 1 0 1	X	X	X	X	X	X	X		
14	1 1 1 0	X	X	X	X	X	X	X		
15	1 1 1 1	X	X	X	X	X	X	X		

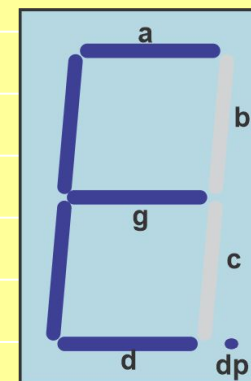
Archivo ABEL-HDL

```
MODULE BCD
" Entradas
A,B,C,D PIN 1,2,3,4;
"Salidas
a,b,c,d,e,f,g pin 16,17,18,12,13,15,14 istype
'com,dc';
S= [A,B,C,D];
" Tabla de verdad para Cátodo Común
truth_table
([S]->[a,b,c,d,e,f,g])
```

```
[0]-> [1,1,1,1,1,1,0];
[1]-> [0,0,1,1,0,0,0];
[2]-> [1,1,0,1,1,0,1];
[3]-> [1,1,1,1,0,0,1];
[4]-> [0,1,1,0,0,1,1];
[5]-> [1,0,1,1,0,1,1];
[6]->
[.x.,0,1,1,1,1,1];
[7]-> [1,1,1,0,0,0,0];
[8]-> [1,1,1,1,1,1,1];
[9]->
[1,1,1,.x.,0,1,1];
```

END

m	A B C D	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	X	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1	1	1	1	X	0	1	1
10	1 0 1 0	1	0	0	1	1	1	1
11	1 0 1 1	1	0	0	1	1	1	1
12	1 1 0 0	1	0	0	1	1	1	1
13	1 1 0 1	1	0	0	1	1	1	1
14	1 1 1 0	1	0	0	1	1	1	1
15	1 1 1 1	1	0	0	1	1	1	1



Con una
E
de error

Archivo ABEL-HDL Con E de error

```
MODULE BCD
```

```
" Entradas
```

```
A,B,C,D PIN 1,2,3,4;
```

```
"Salidas
```

```
a,b,c,d,e,f,g pin 16,17,18,12,13,15,14 istype  
'com,dc';
```

```
S= [A,B,C,D];
```

```
“ Tabla de verdad para Cátodo Común
```

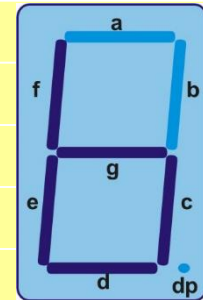
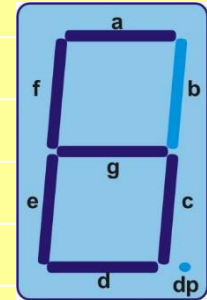
```
truth_table
```

```
([S]->[a,b,c,d,e,f,g])
```

```
[0]-> [1,1,1,1,1,1,0];  
[1]-> [0,0,1,1,0,0,0];  
[2]-> [1,1,0,1,1,0,1];  
[3]-> [1,1,1,1,0,0,1];  
[4]-> [0,1,1,0,0,1,1];  
[5]-> [1,0,1,1,0,1,1];  
[6]-> [.x.,0,1,1,1,1,1];  
[7]-> [1,1,1,0,0,0,0];  
[8]-> [1,1,1,1,1,1,1];  
[9]-> [1,1,1,.x.,0,1,1];  
[10]-> [1,0,0,1,1,1,1];  
[11]-> [1,0,0,1,1,1,1];  
[12]-> [1,0,0,1,1,1,1];  
[13]-> [1,0,0,1,1,1,1];  
[14]-> [1,0,0,1,1,1,1];  
[15]-> [1,0,0,1,1,1,1];  
END
```

Decodificador de Hexadecimal a 7 segmentos

m	A B C D	a	b	c	d	e	f	g	
0	0 0 0 0	1	1	1	1	1	1	0	
1	0 0 0 1	0	1	1	0	0	0	0	
2	0 0 1 0	1	1	0	1	1	0	1	
3	0 0 1 1	1	1	1	1	0	0	1	
4	0 1 0 0	0	1	1	0	0	1	1	
5	0 1 0 1	1	0	1	1	0	1	1	
6	0 1 1 0	1	0	1	1	1	1	1	
7	0 1 1 1	1	1	1	0	0	0	0	
8	1 0 0 0	1	1	1	1	1	1	1	
9	1 0 0 1	1	1	1	X	0	1	1	
10	1 0 1 0	1	1	0	1	1	1	1	A
11	1 0 1 1	0	0	1	1	1	1	1	b
12	1 1 0 0	1	0	0	1	1	1	0	C
13	1 1 0 1	0	1	1	1	1	0	1	d
14	1 1 1 0	1	0	0	1	1	1	1	E
15	1 1 1 1	1	0	0	0	1	1	1	F



Archivo ABEL-HDL

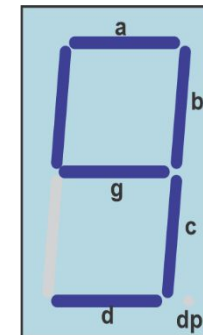
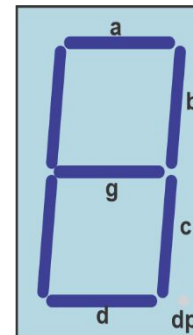
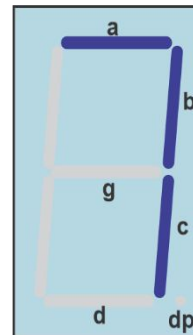
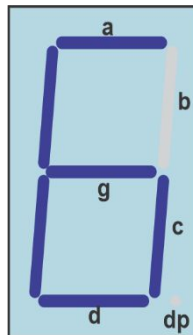
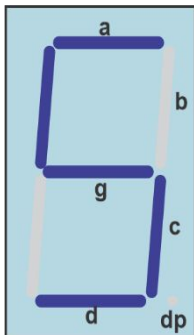
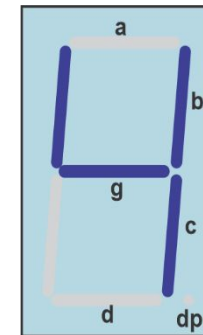
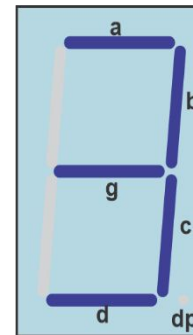
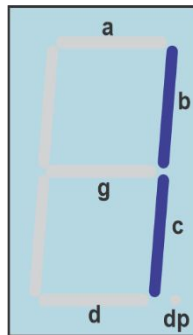
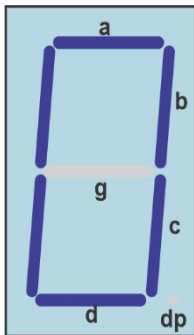
Hexadecimal a 7 segmentos

```
MODULE BCD
" Entradas
A,B,C,D, LT PIN 1,2,3,4,5;
"Salidas
a,b,c,d,e,f,g pin 16,17,18,12,13,15,14 istype
'com,dc';
S= [A,B,C,D];
" Tabla de verdad para Cátodo Común
truth_table
([S]->[a,b,c,d,e,f,g])
```

```
[0]-> [1,1,1,1,1,1,0];
[1]-> [0,0,1,1,0,0,0];
[2]-> [1,1,0,1,1,0,1];
[3]-> [1,1,1,1,0,0,1];
[4]-> [0,1,1,0,0,1,1];
[5]-> [1,0,1,1,0,1,1];
[6]-> [1,0,1,1,1,1,1];
[7]-> [1,1,1,0,0,0,0];
[8]-> [1,1,1,1,1,1,1];
[9]-> [1,1,1,.x.,0,1,1];
[10]-> [1,1,0,1,1,1,1];
[11]-> [0,0,1,1,1,1,1];
[12]-> [1,0,0,1,1,1,0];
[13]-> [0,1,1,1,1,0,1];
[14]-> [1,0,0,1,1,1,1];
[15]-> [1,0,0,0,1,1,1];
END
```

Actividad Fundamental 3 Decodificador de BCD a 7 segmentos

1.- Mostrar los números del 0 al 9.



Prueba de Lámpara

2.- Incluir una entrada adicional al código BCD, por medio de un Push Botón llamada LT (Lamp Test) prueba de lampara, de modo al oprimirlo (LT=1) todos los segmentos deberán de encender incluyendo el punto decimal.

```
MODULE BCD
```

```
X=.x.;
```

```
" Entradas
```

```
A,B,C,D,LT PIN 1..5;
```

```
"Salidas
```

```
a,b,c,d,e,f,g,dp pin 16,17,18,12,13,15,14,19istype 'com,dc';
```

```
“ Tabla de verdad para Cátodo Común
```

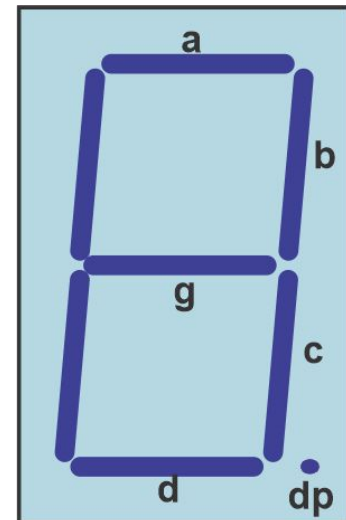
```
truth_table
```

```
([LT,A,B,C,D ]->[a,b,c,d,e,f,g,dp])
```

```
[1,X,X,X,X]->[1,1,1,1,1,1,1,1];
```

```
[0,0,0,0,0]->[1,1,1,1,1,1,0,0];
```

```
[0,0,0,0,1]->[0,1,1,0,0,0,0,0];
```

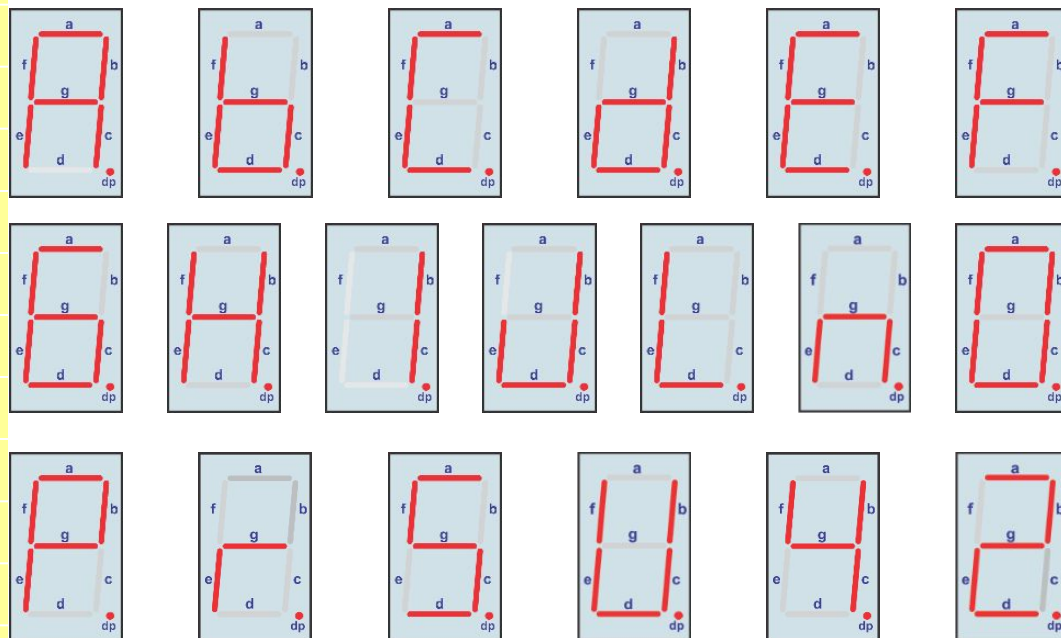


Actividad Fundamental 3 Decodificador de BCD a 7 segmentos

3.- Para las combinaciones del 10 al 15 proponga una palabra de seis caracteres (una letra por cada combinación), además encienda el punto decimal. (hay que registrar la palabra para asegurar que aún no este registrada)

Dicha palabra se deberá de verificar que aún no está registrada en el siguiente enlace: Y de ser así. registra tu propuesta en el siguiente enlace:

<https://uanledu.sharepoint.com/:x:/s/BecariosAE20-21/EcaHwUMPYedOrmGuG23uoVoB4exFQs6vssDckbljdj8QUA?e=GMF7gJ>



	a	b	c	d	e	f	g	
A	1	1	1	0	1	1	1	119
b	0	0	1	1	1	1	1	31
C	1	0	0	1	1	1	0	78
d	0	1	1	1	1	1	0	62
E	1	0	0	1	1	1	1	79
F	1	0	0	0	1	1	1	71
G	0	1	1	1	1	1	1	63
H	0	1	1	0	1	1	1	55
I	0	1	1	0	0	0	0	48
J	0	1	1	1	1	0	0	60
L	0	0	0	1	1	1	0	14
N	1	1	0	1	1	1	0	110
O	1	1	1	1	1	1	0	126
P	1	1	0	0	1	1	1	103
S	1	0	1	1	0	1	1	91
U	0	1	1	0	0	1	0	50
Y	1	1	1	0	0	1	1	115
Z	1	1	1	1	0	1	0	122

Listado de palabras

No contienen las letras M, T, W

1	abedul
2	Abejas
3	ACCESO
4	Acerbo
5	Acerca
6	Acuerdo
7	Afable
8	Ahorca
9	Alegró
10	Algún
11	Aliger
12	Alzado
13	Ancora
14	Anhelo
15	Anillo
16	Ardeos

17	Arduos
18	Arocho
19	Ascend
20	Asedio
21	bailar
22	BALASO
23	BANANA
24	BAncos
25	Bander
26	Blanco
27	Blando
28	Bobina
29	BONDAD
30	Brindo
31	Broche
32	Cadena

33	Cádiz
34	Calder
35	Calzar
36	Canción
37	CANINO
38	Cardeo
39	Cargó
40	Cargón
41	Carino
42	casado
43	CASUAL
44	Caudro
45	CELULA
46	Cerado
47	CESPED
48	CHACHA

49	Cherry
50	Ciego
51	Cigüe
52	Clavar
53	Clavio
54	Cobre
55	COCHES
56	Cochó
57	Cocina
58	Codear
59	CODIGO
60	Colada
61	Colear
62	CONEJO
63	Cordel
64	CORONA

Listado de palabras

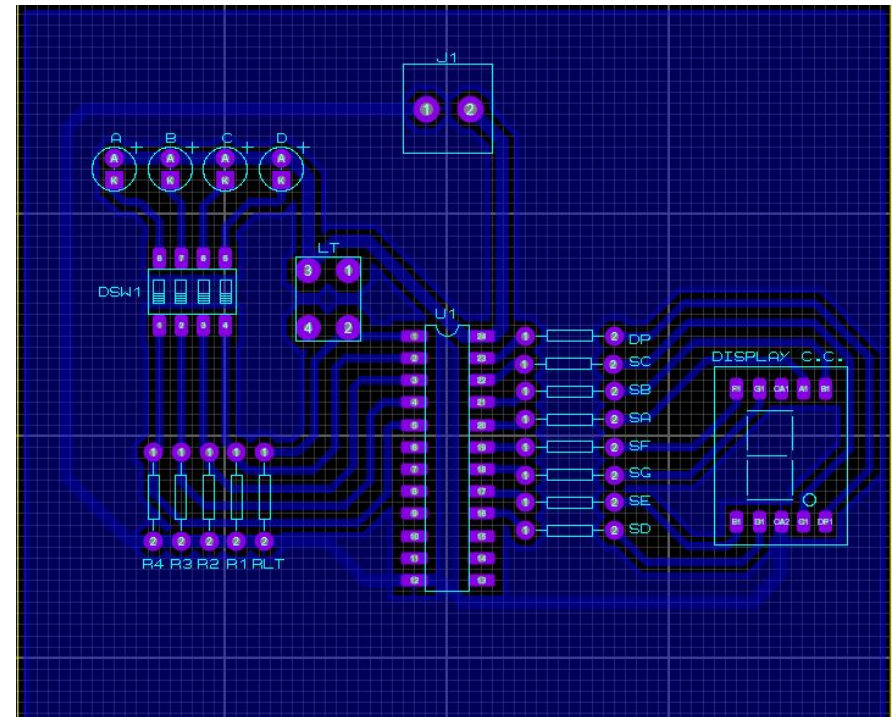
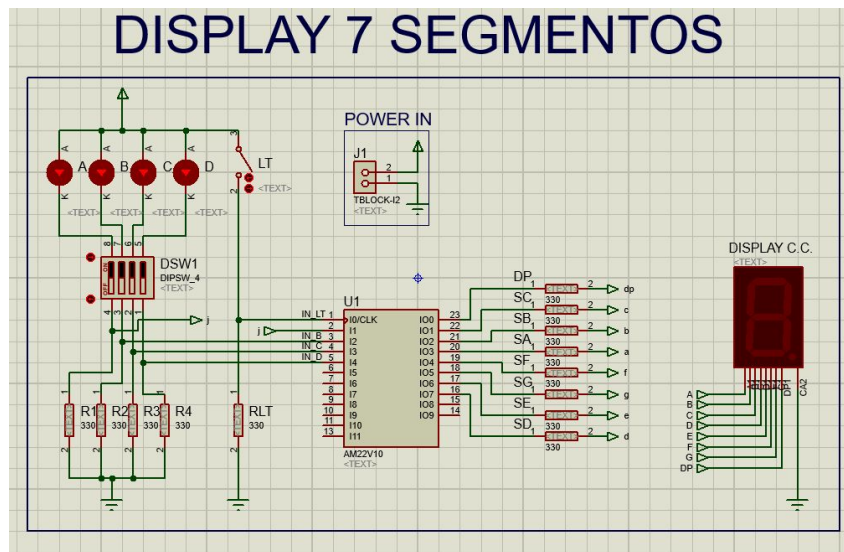
65	Correa	81	DORMIR	97	Escoba	113	Fuerza
66	Corsé	82	Dragon	98	Escudo	114	Fugaz
67	Credal	83	dulces	99	Esgría	115	Fúnebr
68	Críaño	84	Echaro	100	ESPUMA	116	Furgón
69	Cronen	85	Eclips	101	Eunice	117	Galeno
70	Curado	86	Edición	102	Faquir	118	García
71	DANIEL	87	Efesio	103	FELIPE	119	Gárgol
72	Declar	88	Elenco	104	Ferroc	120	Gíndol
73	Decoro	89	Elevó	105	Fibrón	121	Gorjéo
74	Dedazo	90	Eloción	106	Fíjose	122	Grande
75	Deguar	91	Encauz	107	Fijoso	123	Guardó
76	Delgó	92	Encier	108	Flores	124	Hacino
77	Densar	93	Encima	109	Focero	125	Halado
78	Desear	94	Encino	110	FORZAR	126	Hallar
79	dibuja	95	Erosión	111	Fragua	127	helado
80	DIOSES	96	ESCALA	112	Frasco	128	Hilado

Listado de palabras

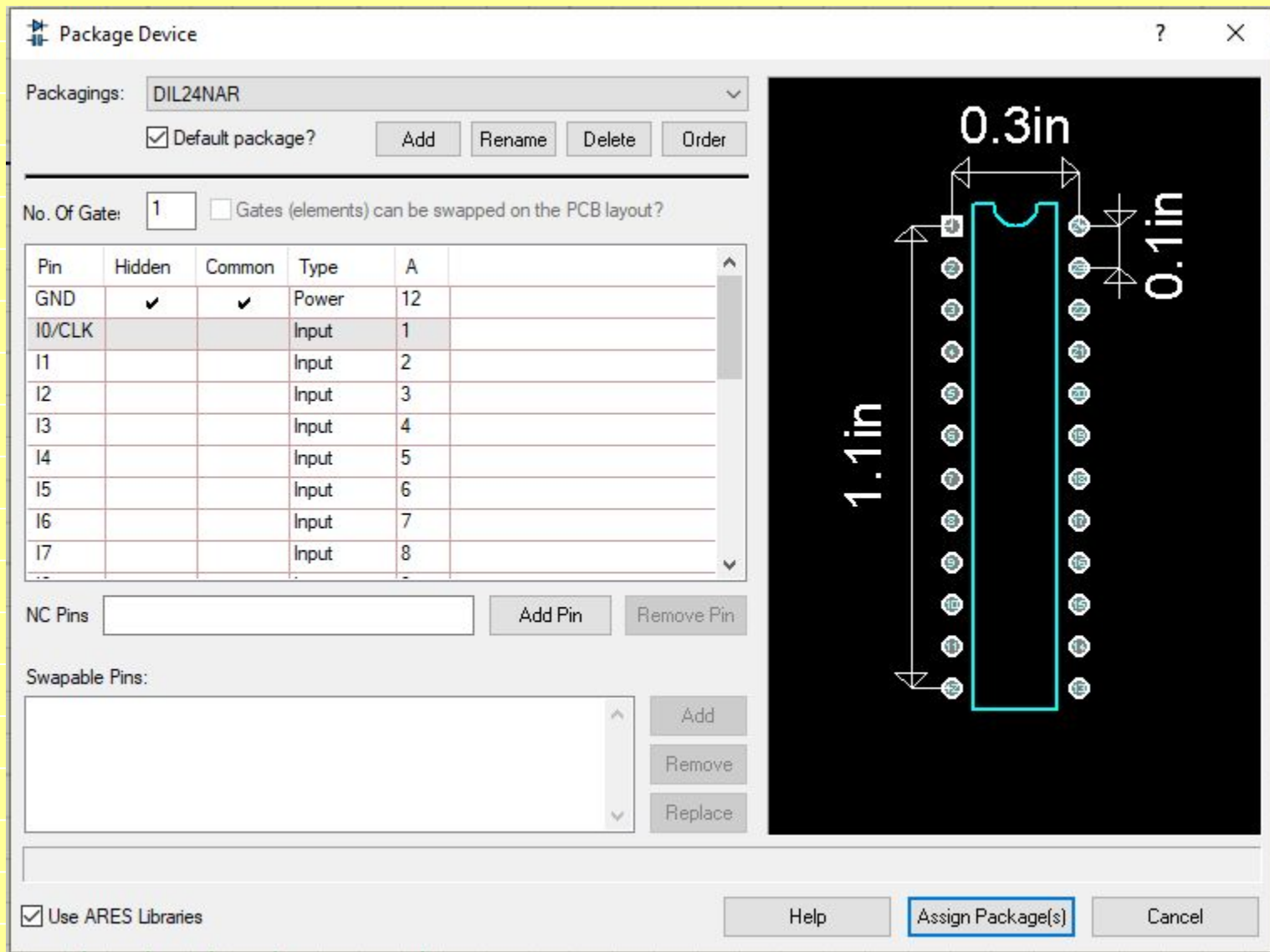
129	Hilvan	145	Lisera	161	Pensar			
130	Hombre	146	Lisero	162	Perros			
131	Hombre	147	Llegó	163	POLLOS			
132	IGUANA	148	Llevar	164	PRUEBA			
133	iloveu	149	Logró	165	PULSOS			
134	INPALA	150	Lucero	166	QUESOS			
135	Iraqui	151	Luego	167	SABADO			
136	Jalone	152	Lugaré	168	SALSAS			
137	Jaulón	153	Lujoso	169	suceso			
138	Jerogl	154	Lupear	170	Unirse			
139	Juegos	155	Lúpulo	171	Volcan			
140	Lacero	156	Obispo	172	ZÓCALO			
141	Ladrón	157	OpciOn					
142	LANZAR	158	origen					
143	Lazare	159	OROPEL					
144	Líbido	160	Papaya					

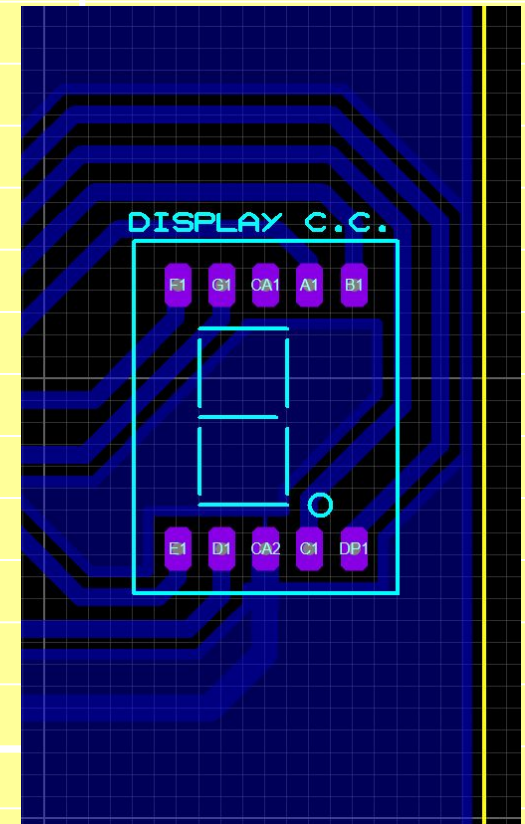
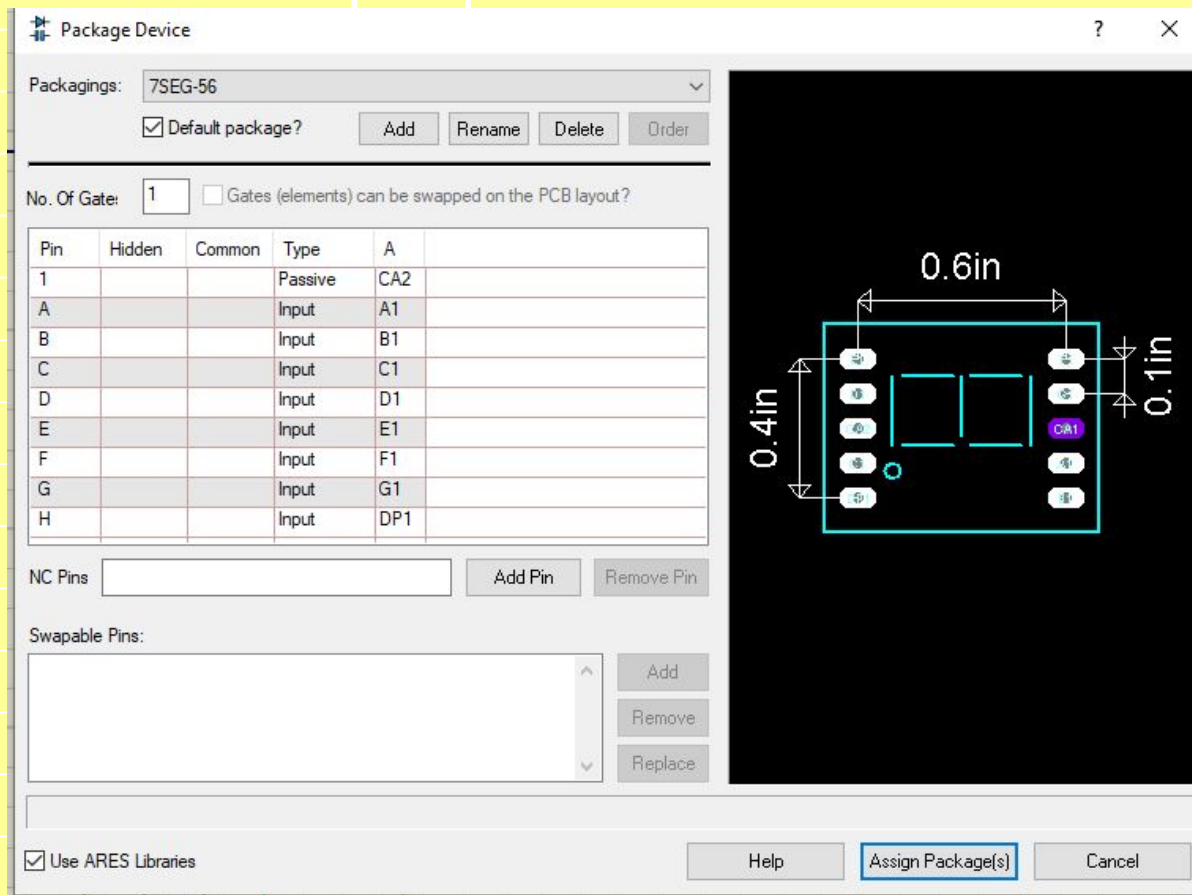
Actividad Fundamental 3 Decodificador de BCD a 7 segmentos

5.- Elabore la simulación y el Layout (PCB) correspondiente al diseño



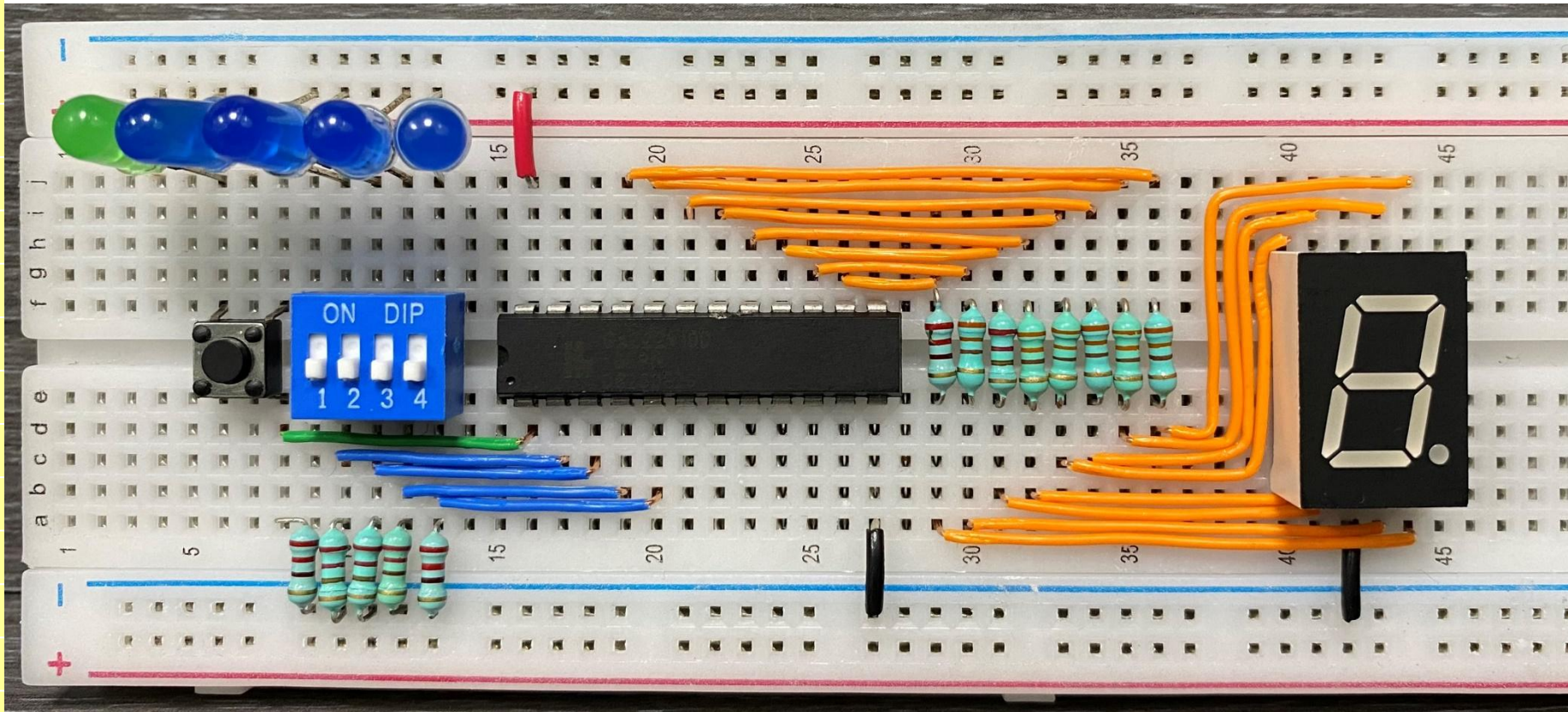
https://www.youtube.com/watch?v=Mghag9dMuV8&t=862s&ab_channel=JuanAngelGarzaGarza

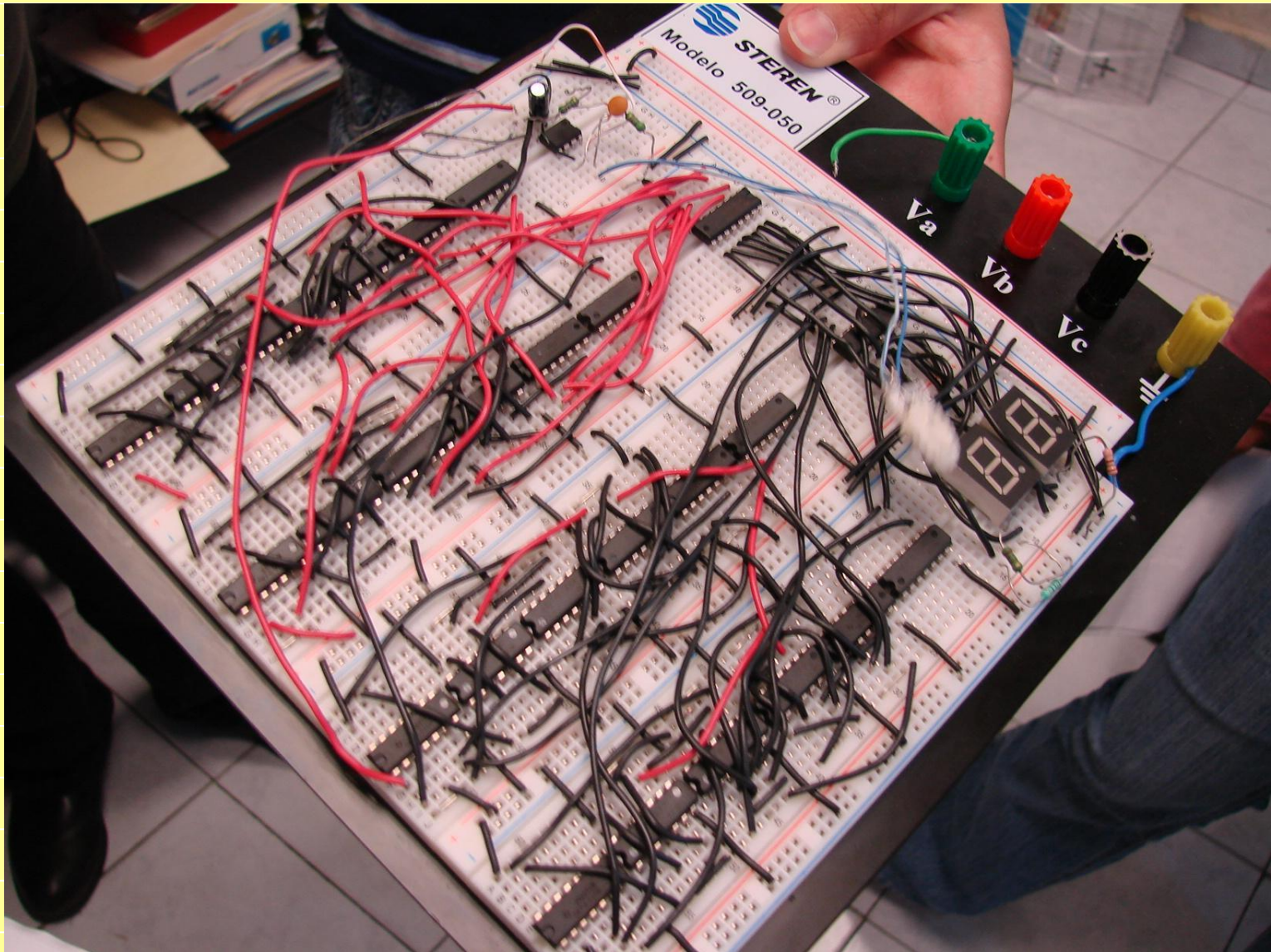




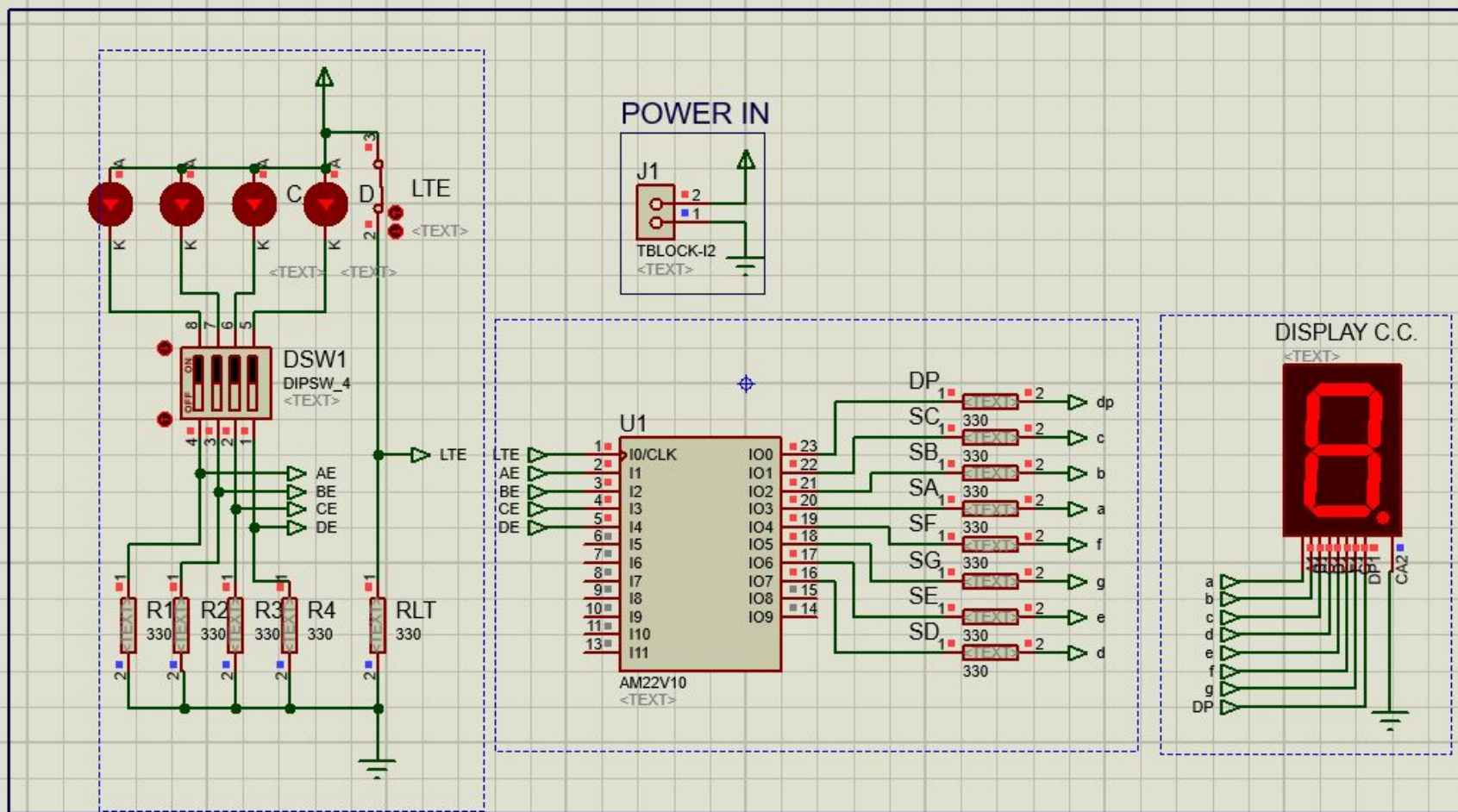
Actividad Fundamental 3 Decodificador de BCD a 7 segmentos

4.- Considere para implementación en un prototipo la asignación necesaria de las terminales de salida de modo que no existan cruces del cableado.

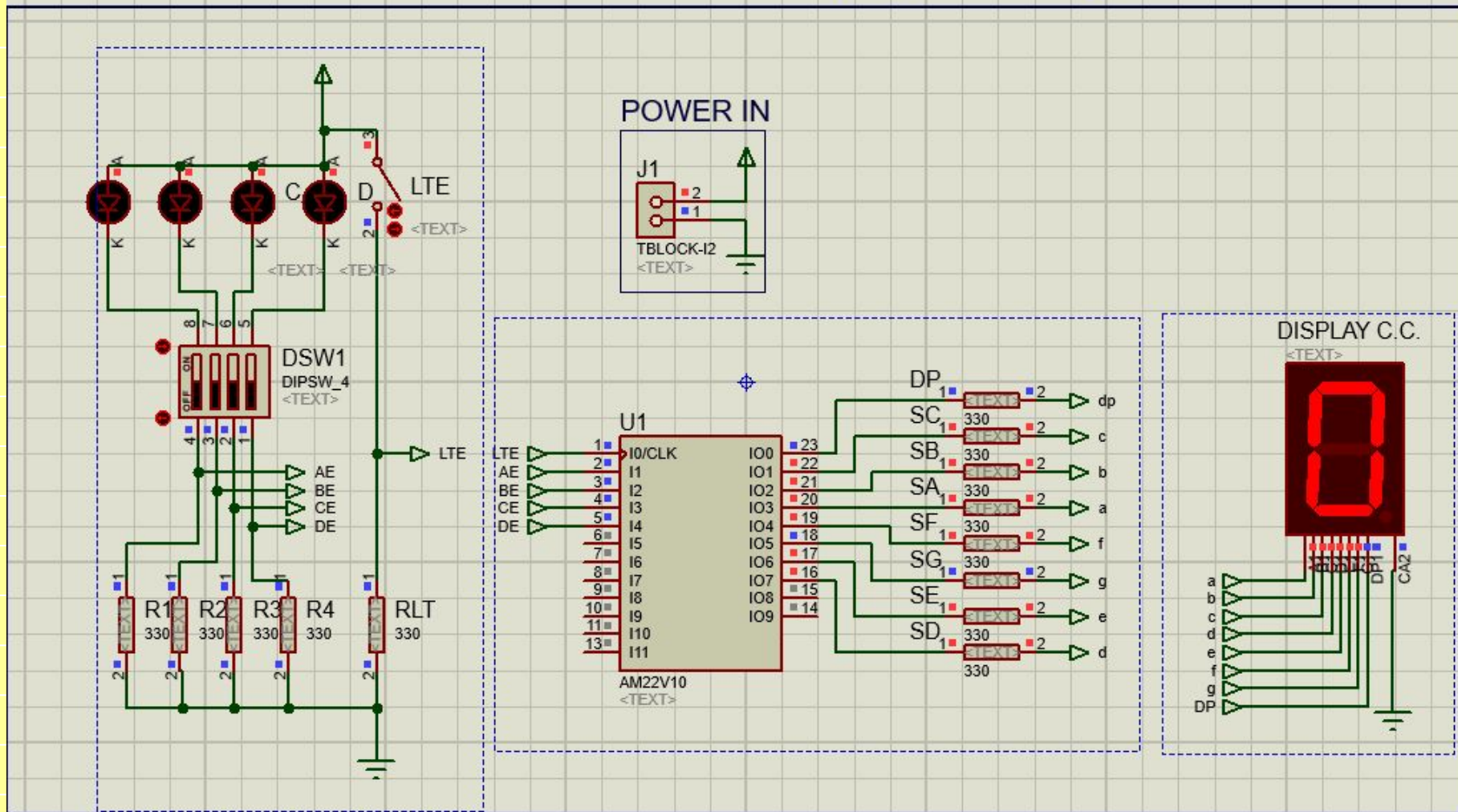




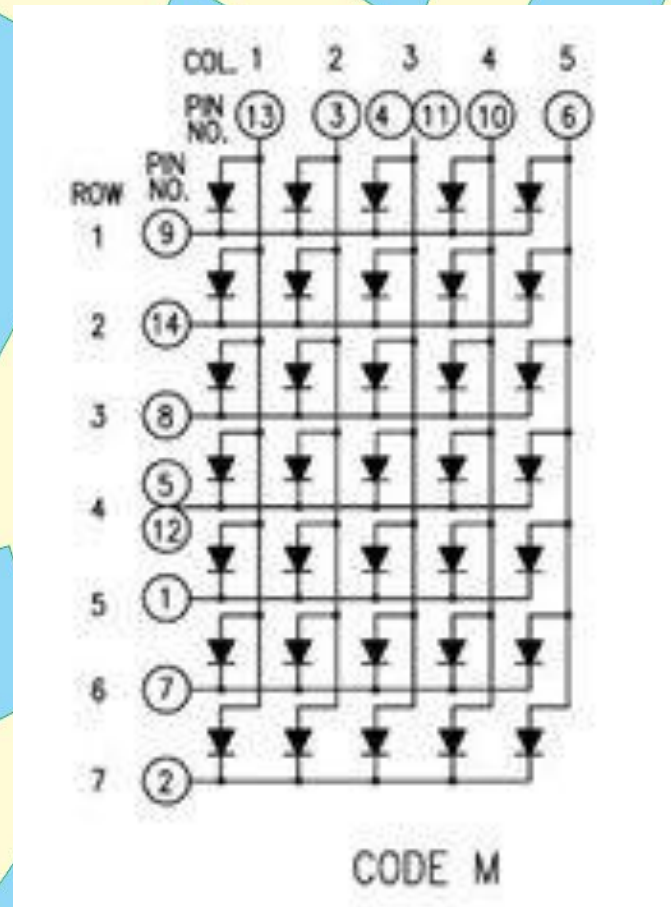
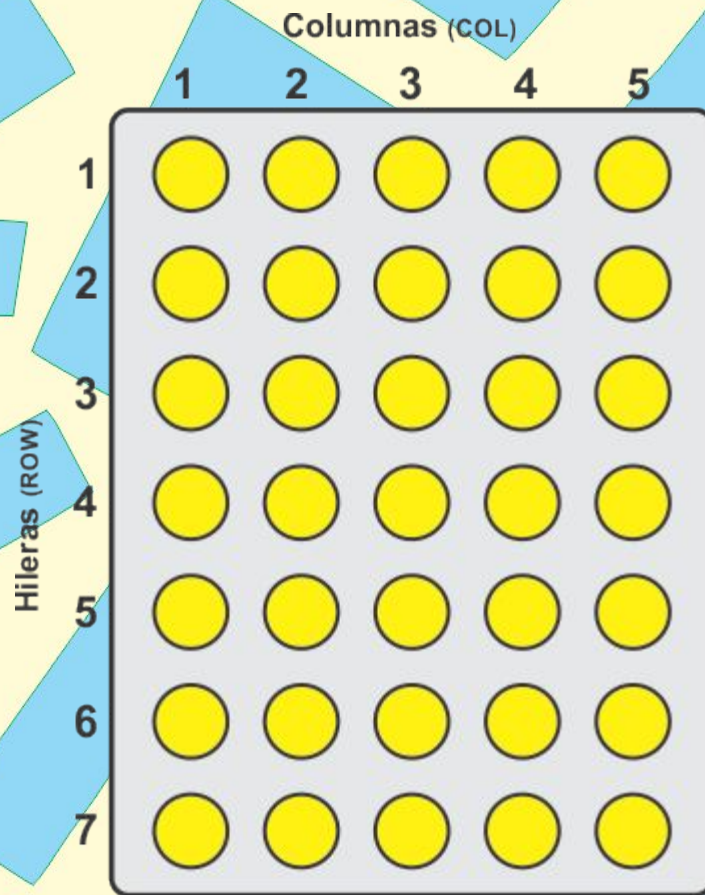
Decodificador de BCD a un Display 7 de segmentos



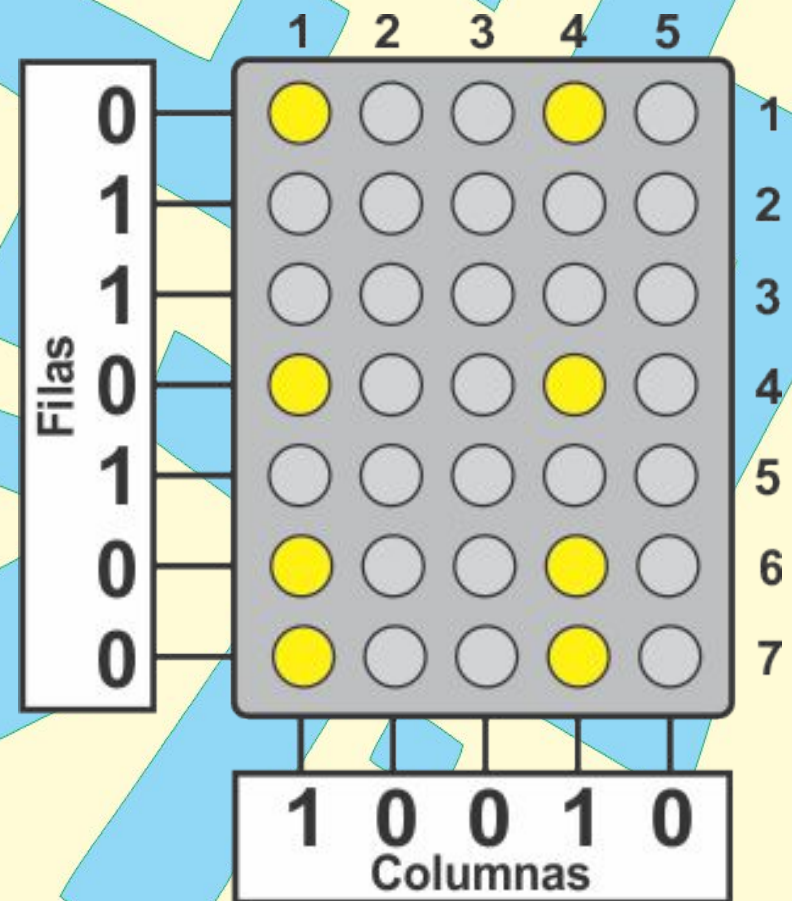
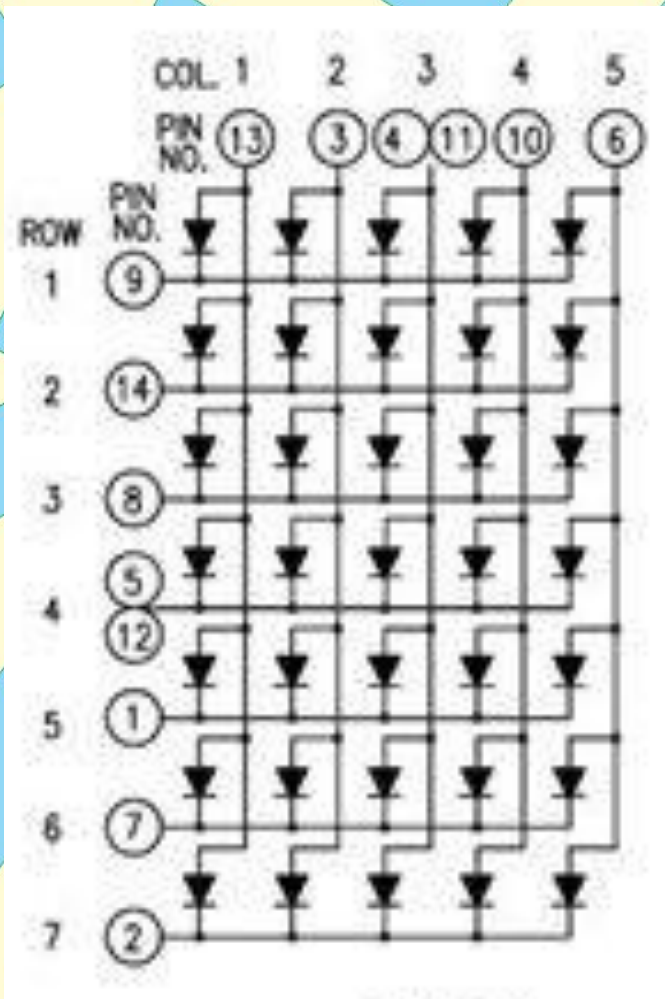
Decodificador de BCD a un Display 7 de segmentos



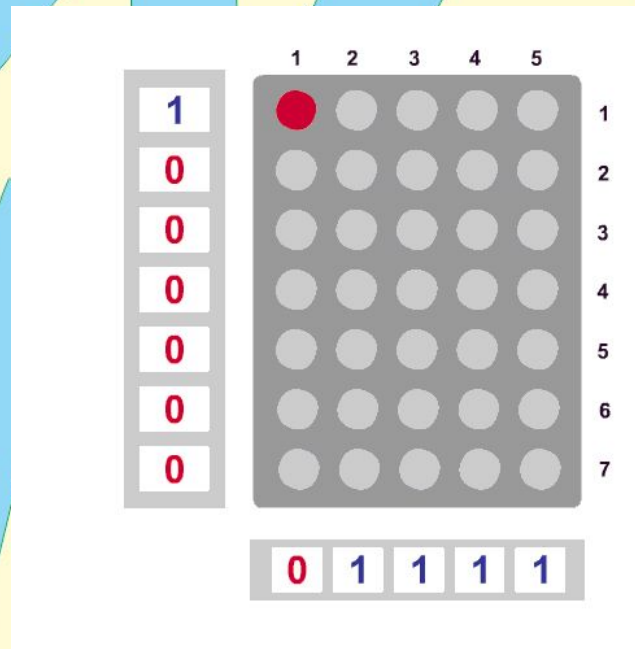
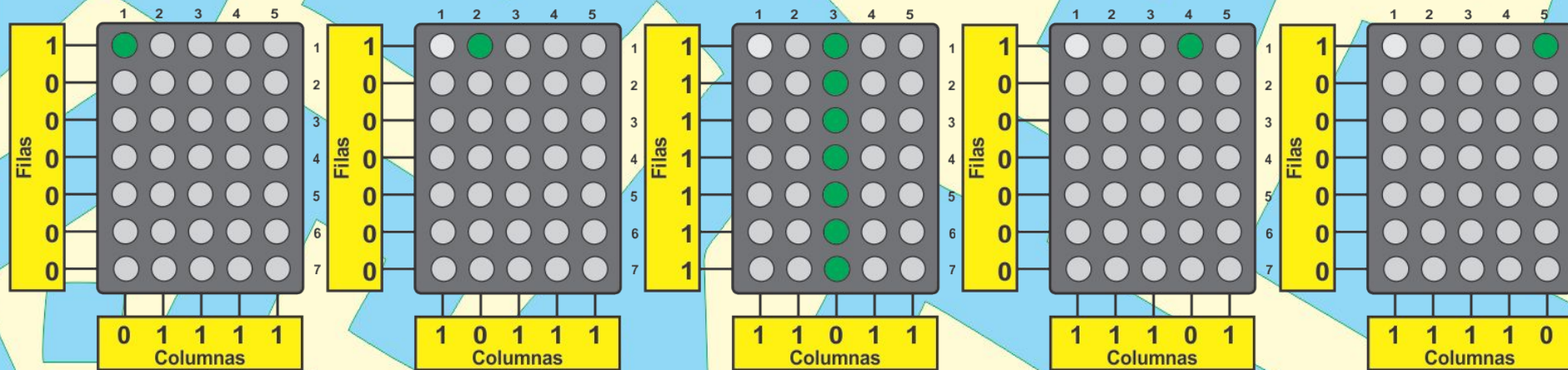
Matriz 5 X 7



Matriz 5 X 7



Matriz 5 X 7



	C1	C2	C3	C4	C5	F1	F2	F3	F4	F5	C	F
0	0	0	0	0	0	0	1	1	1	1	0	15
1	1	0	0	0	0	0	1	1	1	1	16	15
2	0	1	0	0	0	0	1	1	1	1	8	15
3	0	0	1	0	0	0	1	1	1	1	4	15
4	0	0	0	1	0	0	1	1	1	1	2	15
5	0	0	0	0	1	0	1	1	1	1	1	15
6	1	0	0	0	0	1	0	1	1	1	16	23
7	0	1	0	0	0	1	0	1	1	1	8	23
8	0	0	1	0	0	1	0	1	1	1	4	23
9	0	0	0	1	0	1	0	1	1	1	2	23
10	0	0	0	0	1	1	0	1	1	1	1	23
11	1	0	0	0	0	1	1	0	1	1	16	27
12	0	1	0	0	0	1	1	0	1	1	8	27
13	0	0	1	0	0	1	1	0	1	1	4	27
14	0	0	0	1	0	1	1	0	1	1	2	27
15	0	0	0	0	1	1	1	0	1	1	1	27
16	1	0	0	0	0	1	1	1	0	1	16	29
17	0	1	0	0	0	1	1	1	0	1	8	29
18	0	0	1	0	0	1	1	1	0	1	4	29
19	0	0	0	1	0	1	1	1	0	1	2	29
20	0	0	0	0	1	1	1	1	0	1	1	29
21	1	0	0	0	0	1	1	1	1	0	16	30
22	0	1	0	0	0	1	1	1	1	0	8	30
23	0	0	1	0	0	1	1	1	1	0	4	30
24	0	0	0	1	0	1	1	1	1	0	2	30
25	0	0	0	0	1	1	1	1	1	0	1	30

