



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**  
**Dpto. de Ing. en Sistemas Computacionales**  
**Academia de Sistemas Digitales**  
**Diseño de Sistemas Digitales**



Práctica de Laboratorio No. 11 – **Marquesina con ROM.**

**OBJETIVO:** Realizar un programa que permita crear una marquesina usando un módulo de 3 displays multiplexados de ánodo común mediante un lenguaje de descripción de hardware (HDL) en un PLD 22V10.

**MATERIAL Y EQUIPO:**

Mesa de instrumentación del laboratorio de sistemas digitales

1 PLD 22v10

Además de lo anterior, se puede optar por alguna de estas dos opciones:

1 Fuente de 5V 1 Módulo de 3 displays multiplexados de ánodo común. 1 Push Button 4 Resistencias de 1K $\Omega$ 7 Resistencias de 330 $\Omega$ 3 Transistores BC557 1 Protoboard Pinzas y cable para alambrar	1 TEDDi (Tarjeta Educativa para Diseño Digital).
--	--

**INTRODUCCIÓN TEÓRICA**

Una ventaja de los sistemas digitales sobre los analógicos es la capacidad de almacenar información digital por periodos cortos o largos. Los dispositivos en los cuales se realiza dicho almacenamiento de información se conoce como **memoria**.

Las memorias más utilizadas son aquellas basadas en semiconductores (memorias electrónicas), debido a que son las más rápidas. La clasificación de estas memorias se muestra en la ilustración 1.

Las memorias de solo lectura (ROM – Read Only Memory) son un tipo de memorias que están diseñadas para guardar datos que son permanentes o que no cambian con mucha frecuencia.

En algunas ROM los datos que están almacenados tienen que grabarse durante el proceso de fabricación y para otros tipos de ROM esto se puede hacer en forma eléctrica. El proceso de grabar datos se conoce como *programación* de la ROM.

Algunas características más importantes de las ROM son:

- Dependiendo de la tecnología, algunas ROM no pueden cambiar sus datos una vez programadas; otras pueden borrarse y reprogramarse varias veces.
- Las memorias ROM son no volátiles, es decir, los datos almacenados permanecen cuando se desconecta la energía eléctrica.

AUTOR: VICTOR HUGO GARCIA ORTEGA

- Durante la operación normal, no pueden escribirse datos en una ROM, pero si se pueden leer.
- El tiempo de acceso de una ROM es grande en comparación con una memoria RAM.

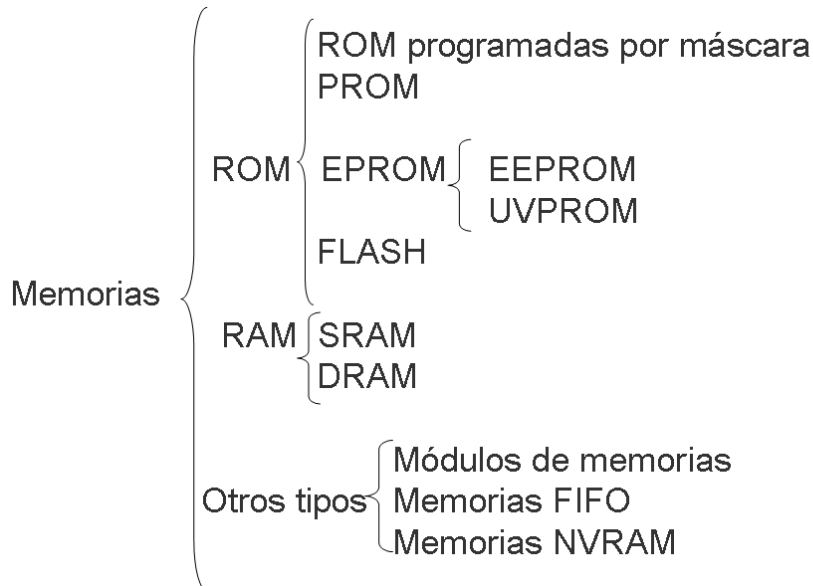


Ilustración 1 Clasificación de memorias

La implementación de memorias ROM usando un lenguaje de descripción de hardware se hace mediante arreglos. En esta práctica se programarán 3 microarquitecturas diferentes para crear un diseño digital que permita implementar una marquesina usando memorias ROM para almacenar los caracteres a desplegar.

## PROCEDIMIENTO.

### Antes de asistir al laboratorio:

1. Realizar el programa de la microarquitectura mostrada en la ilustración 2. Este diseño consiste de un contador de 10 bits que se encuentra en el SPLD1. Este contador usa sus 3 bits más significativos para proporcionar la dirección a las memorias ROM que se encuentran en el SPLD2. Al usar los bits más significativos, el cambio en estos bits se da a la frecuencia dada por:

Para un contador de n bits la frecuencia de salida en cada uno de sus bits esta dada por:

$$Q_n = \frac{FCLK}{2^{n+1}}$$

Si se considera una frecuencia de reloj ( $FCLK$ ) de 256Hz y si se toma la salida de los 3 bits MSB ( $Q_7$ ,  $Q_8$  y  $Q_9$ ) del contador, la frecuencia del mensaje en modo marquesina será la frecuencia de  $Q_7$ , es decir:

$$Q_7 = \frac{FCLK}{2^{7+1}} = \frac{256}{256} = 1Hz$$

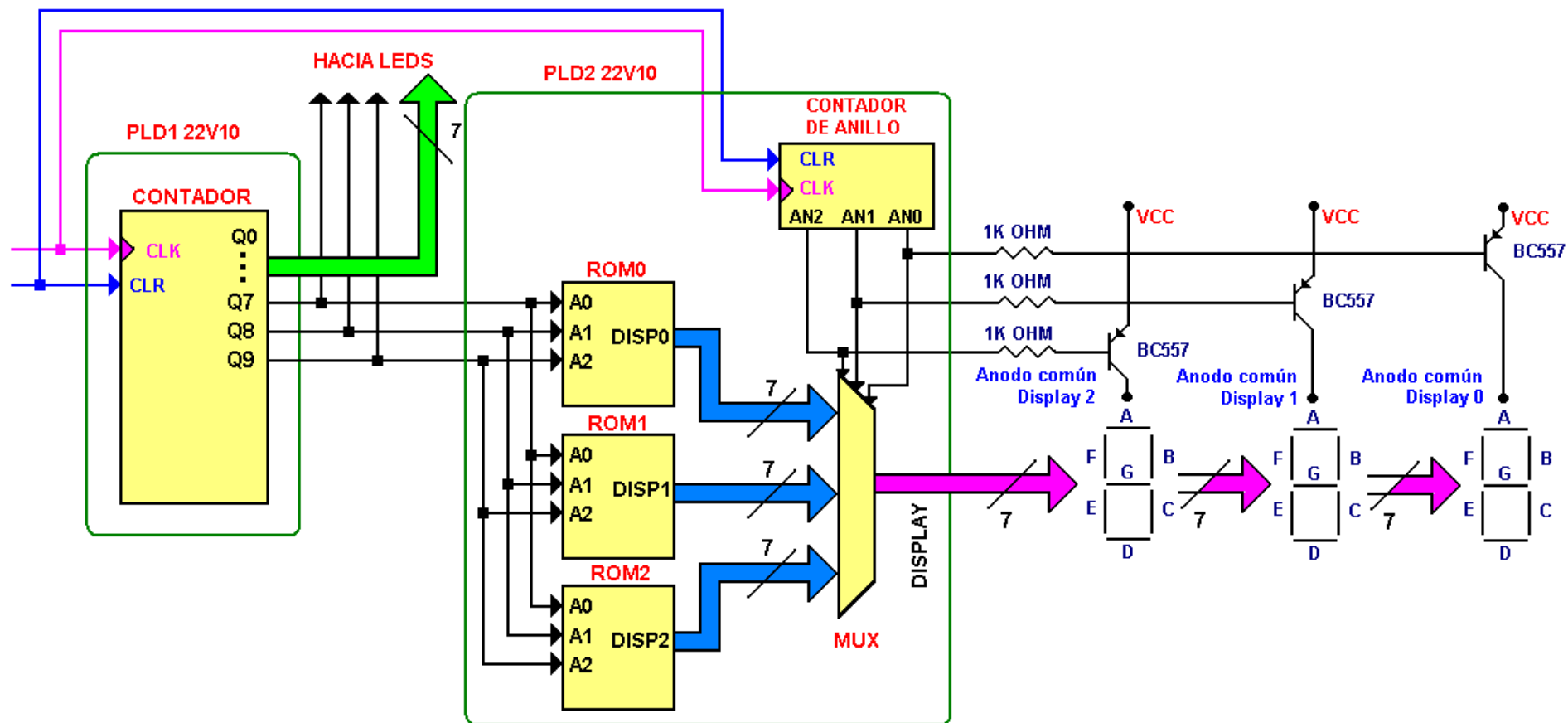


Ilustración 2 Microarquitectura 1.

Por lo que las direcciones en las memorias ROM cambiarán de valor cada segundo.

En el SPLD2 se encuentran las memorias ROM que contienen los códigos de 7 segmentos para que se muestren las letras del mensaje “dISEÑO” en el módulo de 3 displays multiplexados. Las memorias tienen una organización de 8x7, por lo que se tienen:

- 3 bits en el bus de direcciones.
- 7 bits en el bus de datos.
- Capacidad de 56 bits.

Por las 3 memorias se tiene una capacidad total de 56 bits x 3 memorias = 168 bits. Los datos almacenados en cada memoria se muestran en la tabla 1.

Dirección	ROM2	ROM1	ROM0
000	—	—	d
001	—	d	l
010	d	l	S
011	l	S	E
100	S	E	ñ
101	E	ñ	O
110	ñ	O	—
111	O	—	—

Tabla 1 Contenido de las memorias ROM

La dirección proveniente del contador del SPLD1 se coloca en el bus de direcciones de las 3 memorias simultáneamente, esto provoca que se lean 3 datos que son multiplexados para enviarse a cada uno de los displays del módulo de 3 displays multiplexados. El selector del multiplexor es un contador de anillo que selecciona el display y al mismo tiempo el dato de la memoria ROM que se tiene que mostrar en dicho display.

Para el contador de anillo del SPLD2 considere los códigos mostrados en la tabla 2.

AN2	AN1	AN0	Display
1	1	0	D0
1	0	1	D1
0	1	1	D2

Tabla 2 Secuencia del contador de anillo.

Para el multiplexor del SPLD2 considere los códigos mostrados en la tabla 3.

AN2	AN1	AN0	Salida
1	1	0	DISPLAY = DISP0
1	0	1	DISPLAY = DISP1
0	1	1	DISPLAY = DISP2

Tabla 3 Multiplexor

- Realizar el programa de la microarquitectura mostrada en la ilustración 3. Este diseño consiste de un contador de 10 bits que se encuentra en el SPLD1. Este contador usa sus 4 bits más significativos para proporcionar la dirección a las memorias ROM que se encuentran en el SPLD2. Al usar los bits más significativos, el cambio que se da en estos bits se da a la frecuencia dada por:

Para un contador de n bits la frecuencia de salida en cada uno de sus bits esta dada por:

$$Q_n = \frac{FCLK}{2^{n+1}}$$

Si se considera una frecuencia de reloj ( $FCLK$ ) de 256Hz y si se toma la salida de los 4 bits MSB ( $Q_6$ ,  $Q_7$ ,  $Q_8$  y  $Q_9$ ) del contador, la frecuencia del mensaje en modo marquesina será la frecuencia de  $Q_6$ , es decir:

$$Q_6 = \frac{FCLK}{2^{6+1}} = \frac{256}{128} = 2Hz$$

Por lo que las direcciones en las memorias ROM cambiarán de valor cada medio segundo.

A diferencia de la microarquitectura 1, en el SPLD2 se encuentran las memorias ROM que contienen códigos de 4 bits, que representan cada una de las letras del mensaje "dISEÑO dIgItAL" a mostrar en el módulo de 3 displays multiplexados. La codificación elegida es la GRAY. Estos códigos se muestran en la tabla 5.

C3	C2	C1	C0	Etiqueta	Letra
0	0	0	0	COD__	_
0	0	0	1	COD_d	d
0	0	1	1	COD_l	l
0	0	1	0	COD_S	S
0	1	1	0	COD_E	E
0	1	1	1	COD_ñ	ñ
0	1	0	1	COD_O	O
0	1	0	0	COD_g	g
1	1	0	0	COD_t	t
1	1	0	1	COD_A	A
1	1	1	1	COD_L	L
1	1	1	0	COD_X	

Tabla 4 Códigos asignados a las letras de los mensajes

**Estos códigos se asignan con la finalidad de reducir el tamaño de la memoria, pero se requiere de forma adicional un convertidor de código para convertir el código de 4 bits al código de 7 segmentos que se requiere para mostrar la letra en el display. Si no se asignan estos códigos las memorias serían más grandes!!!. Esta es una técnica que permite reducir el tamaño de las memorias y hacer una compresión de los datos almacenados.**

Las memorias tienen una organización de 16x4, por lo que se tienen:

- 4 bits en el bus de direcciones.
- 4 bits en el bus de datos.
- Capacidad de 64 bits.

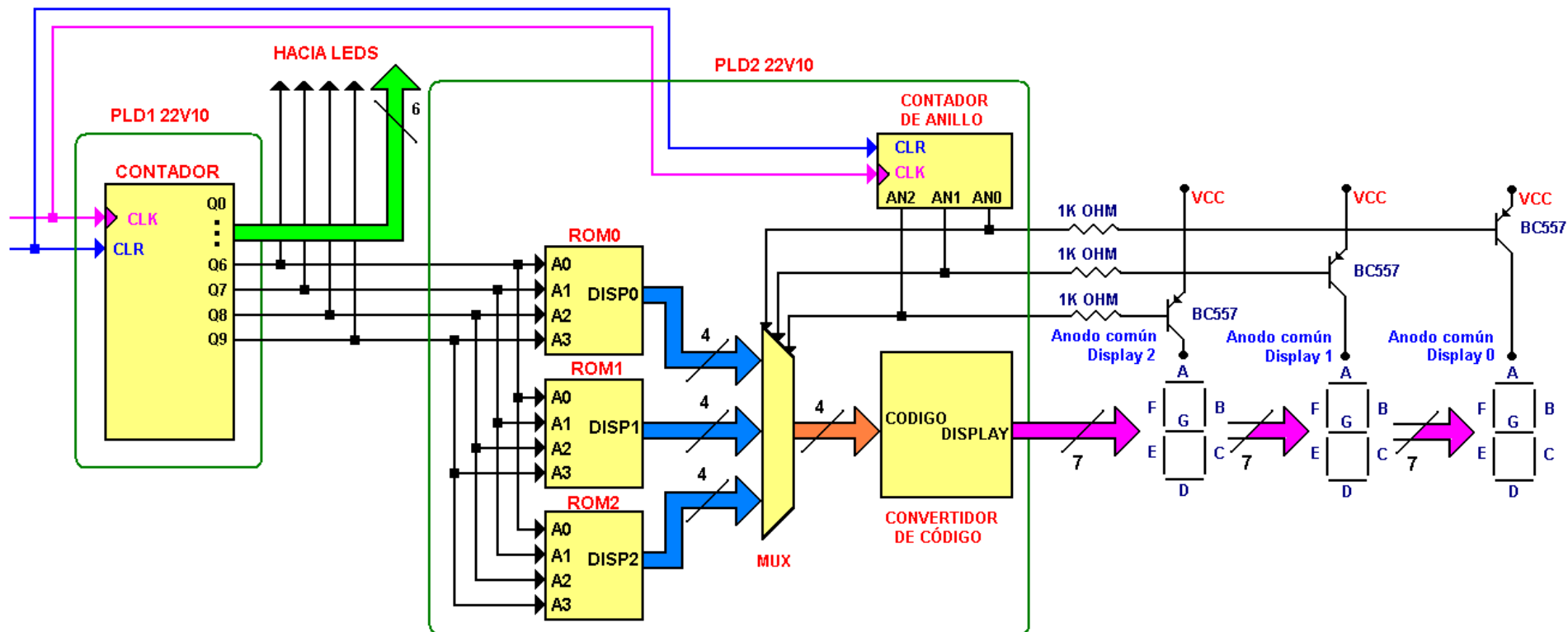


Ilustración 3 Microarquitectura 2.

Por las 3 memorias se tiene una capacidad total de 64 bits x 3 memorias = 192 bits. Los datos almacenados en cada memoria se muestran en la tabla 5.

Dirección	ROM2	ROM1	ROM0
0000	COD__	COD__	COD_d
0001	COD__	COD_d	COD_I
0010	COD_d	COD_I	COD_S
0011	COD_I	COD_S	COD_E
0100	COD_S	COD_E	COD_ñ
0101	COD_E	COD_ñ	COD_O
0110	COD_ñ	COD_O	COD__
0111	COD_O	COD__	COD_d
1000	COD__	COD_d	COD_I
1001	COD_d	COD_I	COD_g
1010	COD_I	COD_g	COD_I
1011	COD_g	COD_I	COD_t
1100	COD_I	COD_t	COD_A
1101	COD_t	COD_A	COD_L
1110	COD_A	COD_L	COD__
1111	COD_L	COD__	COD__

Tabla 5 Contenido de las memorias ROM

La dirección proveniente del contador del SPLD1 se coloca en el bus de direcciones de las 3 memorias simultáneamente, esto provoca que se lean 3 datos que son multiplexados para enviarse a cada uno de los displays del módulo de 3 displays multiplexados. El selector del multiplexor es un contador de anillo que selecciona el display y al mismo tiempo el dato de la memoria ROM que se tiene que convertir al código de 7 segmentos a mostrar en dicho display.

Para el contador de anillo del SPLD2 considere los códigos mostrados en la tabla 2.

Para el multiplexor del SPLD2 considere los códigos mostrados en la tabla 6.

AN2	AN1	AN0	Salida
1	1	0	CODIGO = DISP0
1	0	1	CODIGO = DISP1
0	1	1	CODIGO = DISP2
0	0	0	CODIGO = CODX
0	0	1	CODIGO = CODX
0	1	0	CODIGO = CODX
1	0	0	CODIGO = CODX
1	1	1	CODIGO = CODX

Tabla 6 Multiplexor

**El código CODX se utiliza para sintetizar un circuito con menos términos productos a las salidas.**

Para el convertidor de código del SPLD2 considere los códigos mostrados en la tabla 7.

C3	C2	C1	C0	A	B	C	D	E	F	G	LETRA
0	0	0	0	1	1	1	0	1	1	1	—
0	0	0	1	1	0	0	0	0	1	0	d
0	0	1	1	1	0	0	1	1	1	1	l
0	0	1	0								S
0	1	1	0								E
0	1	1	1								ñ
0	1	0	1								O
0	1	0	0								g
1	1	0	0								t
1	1	0	1								A
1	1	1	1								L
1	1	1	0	X	X	X	X	X	X	X	
1	0	1	0	X	X	X	X	X	X	X	
1	0	1	1	X	X	X	X	X	X	X	
1	0	0	1	X	X	X	X	X	X	X	
1	0	0	0	X	X	X	X	X	X	X	

Tabla 7 Convertidor de código

El alumno debe completar la tabla 7. **El término X (no importa) se utiliza para sintetizar un circuito con menos términos productos a las salidas.**

- Realizar el programa de la microarquitectura mostrada en la ilustración 4. Este diseño consiste de un contador de 10 bits que se encuentra en el SPLD1. Este contador usa sus 3 bits más significativos para proporcionar la dirección a la memoria ROM que se encuentran en el SPLD2. Al usar los bits más significativos, el cambio que se da en estos bits se da a la frecuencia dada por:

Para un contador de n bits la frecuencia de salida en cada uno de sus bits esta dada por:

$$Q_n = \frac{FCLK}{2^{n+1}}$$

Si se considera una frecuencia de reloj (  $FCLK$  ) de 256Hz y si se toma la salida de los 3 bits MSB (  $Q_7$ ,  $Q_8$  y  $Q_9$  ) del contador, la frecuencia del mensaje en modo marquesina será la frecuencia de  $Q_7$ , es decir:

$$Q_7 = \frac{FCLK}{2^{7+1}} = \frac{256}{256} = 1Hz$$

Por lo que las direcciones en las memorias ROM cambiarán de valor cada segundo.

En el SPLD2 se encuentran la memoria ROM que contienen los códigos de 7 segmentos para que se muestren las letras del mensaje “ESConn” en el módulo de 3 displays multiplexados. La memoria tiene una organización de 10x7, por lo que se tienen:

- 4 bits en el bus de direcciones.
- 7 bits en el bus de datos.
- Capacidad de 70 bits.



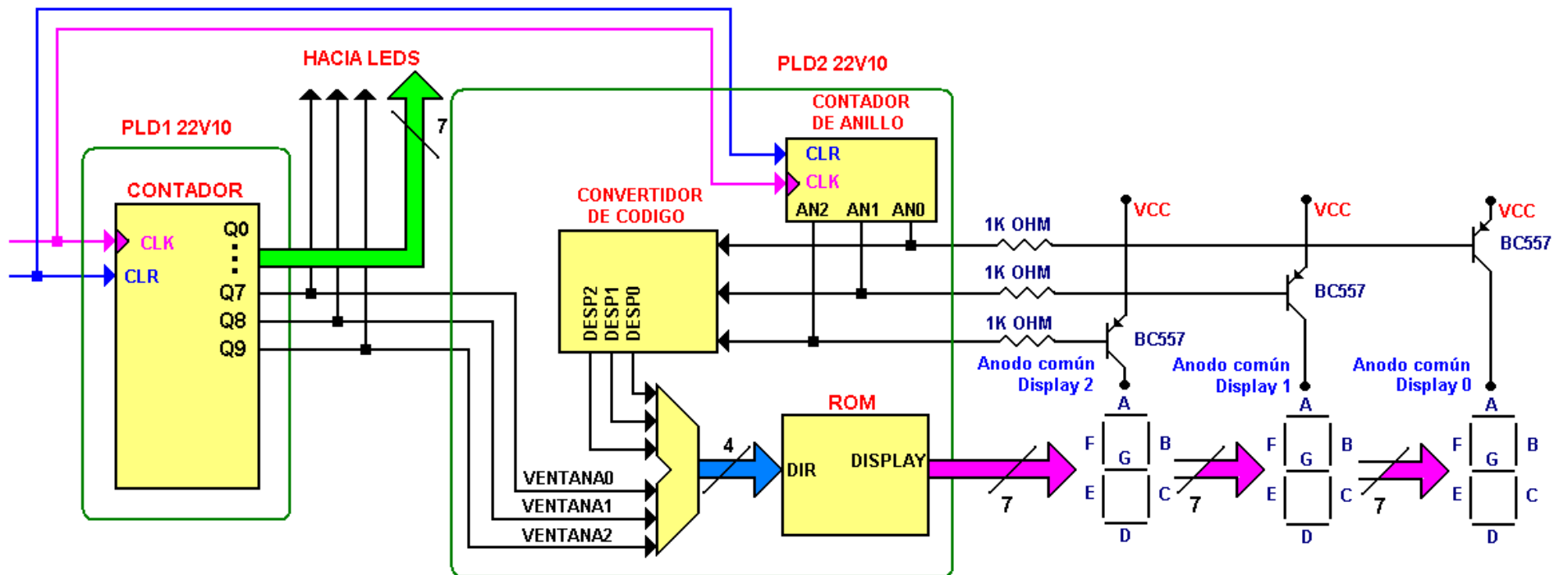


Ilustración 4 Microarquitectura 3.

Los datos almacenados en cada memoria se muestran en la tabla 8.

Dirección	ROM
0000	—
0001	—
0010	E
0011	S
0100	C
0101	O
0110	n
0111	n
1000	—
1001	—

Tabla 8 Contenido de la memoria ROM.

A diferencia de las microarquitecturas anteriores, la dirección de la memoria ROM se forma con la suma del valor proveniente del contador del SPLD1 y del convertidor de código. Esta suma de 4 bits se coloca en el bus de direcciones de la memoria para leer el dato que se va a enviar al display correspondiente del módulo de 3 displays multiplexados.

El valor del contador proporciona el inicio de una ventana de 3 letras que se va desplazando sobre el mensaje a desplegar en los displays cada segundo. Una vez que se tiene el inicio de la ventana, se realiza un desplazamiento con el valor proveniente del convertidor de código. Este desplazamiento cambia a una velocidad igual a la frecuencia del reloj ( $FCLK$ ).

**El circuito sumador debe implementarse usando ecuaciones para que se pueda llevar a cabo la síntesis correctamente.**

Para el contador de anillo del SPLD2 considere los códigos mostrados en la tabla 2.

Para el convertidor de código del SPLD2 considere los códigos mostrados en la tabla 9.

AN2	AN1	AN0	DESP2	DESP1	DESP0
1	1	0	0	0	0
1	0	1	0	0	1
0	1	1	0	1	0
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	1	0	0	0

Tabla 9 Convertidor de código

**El valor “000” en las últimas 5 combinaciones se utilizan para sintetizar un circuito con menos términos productos a las salidas.**

**Programar las memorias ROM mediante arreglos.**

**4. Simular los diseños en el ambiente de desarrollo.**

AUTOR: VICTOR HUGO GARCIA ORTEGA

5. Una vez simulados los diseños, construir el circuito mostrado en la ilustración 4 para probarlo en el laboratorio. **En caso de usar la TEDDI este paso no es necesario.**

#### Ilustración 4 Diagrama esquemático

#### En el laboratorio:

- 1.- Programar el PLD 22V10 usando el programador disponible del laboratorio
- 2.- Verificar el correcto funcionamiento del diseño.

#### **CUESTIONARIO**

1. ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica?
2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica?
3. ¿Cuántos pines de entrada/salida del PLD 22V10 se usan en el diseño?
4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?
5. ¿Cómo se implementó el sumador en la microarquitectura 3?
6. ¿Qué puedes concluir de esta práctica?