



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
ACADEMIA DE SISTEMAS DIGITALES



DISEÑO DE SISTEMAS DIGITALES

Práctica de Laboratorio No. 8

“Teclado Matricial”

Profesores:

VICTOR HUGO GARCIA ORTEGA
JULIO CESAR SOSA SAVEDRA

OBJETIVO: Realizar un programa que permita decodificar un teclado matricial de 4 x 3 y visualizar el dígito presionado en un display de 7 segmentos mediante un lenguaje de descripción de hardware (HDL) en un PLD 22V10.

MATERIAL Y EQUIPO:

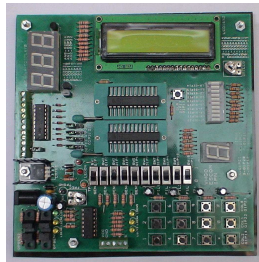
Mesa de instrumentación del laboratorio de sistemas digitales

1 PLD 22v10

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

1 Fuente de 5V
1 Generador de funciones
13 Interruptores de presión
5 Resistencias de 1K Ω
7 Resistencias de 330 Ω
1 Display de ánodo común
1 Protoboard
Pinzas y cable para alambrar

1 TEDDi (Tarjeta Educativa para Diseño Digital).



INTRODUCCIÓN TEÓRICA

Un teclado matricial está constituido por una matriz de interruptores dispuestos en filas y columnas, como se muestra en la ilustración 1. Su intención es reducir el número de líneas necesarias para su interconexión.

Para detectar si hay alguna tecla presionada se aplica a una columna un nivel bajo y a las otras dos columnas un nivel alto. Si se oprime una tecla en la columna donde se aplica el nivel bajo, ese nivel bajo aparecerá en la fila en la que se encuentra el interruptor presionado. Por ejemplo, si se presiona la tecla 1 y se aplica en las columnas 011, ($COL2 = 0$, $COL1 = 1$ y $COL0 = 1$) entonces el nivel bajo aparecerá en la fila 0 ($FIL0 = 0$ y el resto de las filas serán 1). Esta secuencia de valores que se colocan en cada columna se puede generar con un contador de anillo con un cero viajero.

En caso de que no exista ninguna tecla presionada, entonces todas las filas tendrán unos ($FIL0 = 1$, $FIL1 = 1$, $FIL2 = 1$ y $FIL3 = 1$), puesto que tienen resistencias de Pull-Up.

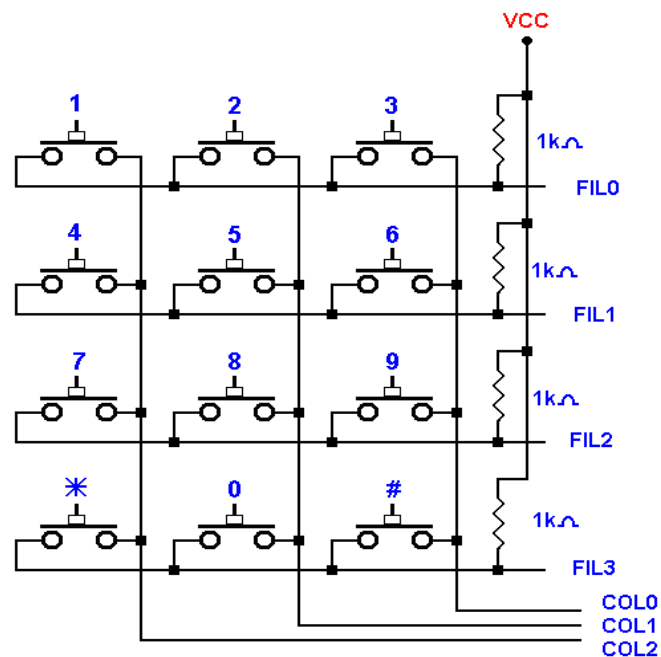


Ilustración 1 Teclado matricial con resistencias Pull-Up.

PROCEDIMIENTO.Antes de asistir al laboratorio:

1. Realizar un programa en HDL que permita implementar el diseño mostrado en la microarquitectura de la ilustración 2.

El contador de anillo es el encargado de colocar un cero en las columnas del teclado, por lo que este contador debe tener un cero viajero. La columna que se selecciona con cada valor del contador se muestra en la tabla 1.

COL2	COL1	COL0	Columna
1	1	0	0
1	0	1	1
0	1	1	2

Tabla 1 Columna que se selecciona en el teclado.

Este contador se recomienda implementar usando el operador ROL, este operador funciona con tipos de datos bit o booleanos, por lo que si se tienen señales del tipo std_logic, se debe usar las funciones pertinentes para hacer el cambio del tipo de dato. Estas funciones son:

```
TO_BITVECTOR( SIGNAL STD_LOGIC_VECTOR )
TO_STDLOGICVECTOR( SIGNAL BIT_VECTOR )
```

Ejemplo:

```
AN <= TO_STDLOGICVECTOR( TO_BITVECTOR(AN) ROL 1 );
```



El registro permite retener la última tecla presionada para poder ser visualizada en el display. Este registro tiene solamente la señal de control L y su funcionamiento se muestra en la tabla 2.

L	Operación
0	Retención
1	Carga

Tabla 2 Funcionamiento del registro.

La compuerta NAND permite cargar al registro la tecla que ha sido presionada. Cuando no se presionan las teclas, el valor que se encuentran en las filas es “1111”, por lo que la salida de la compuerta NAND es 0 y por lo tanto el registro esta en retención. Cuando alguna tecla es presionada la salida de la compuerta NAND es 1 y por lo tanto se habilita la operación de carga.

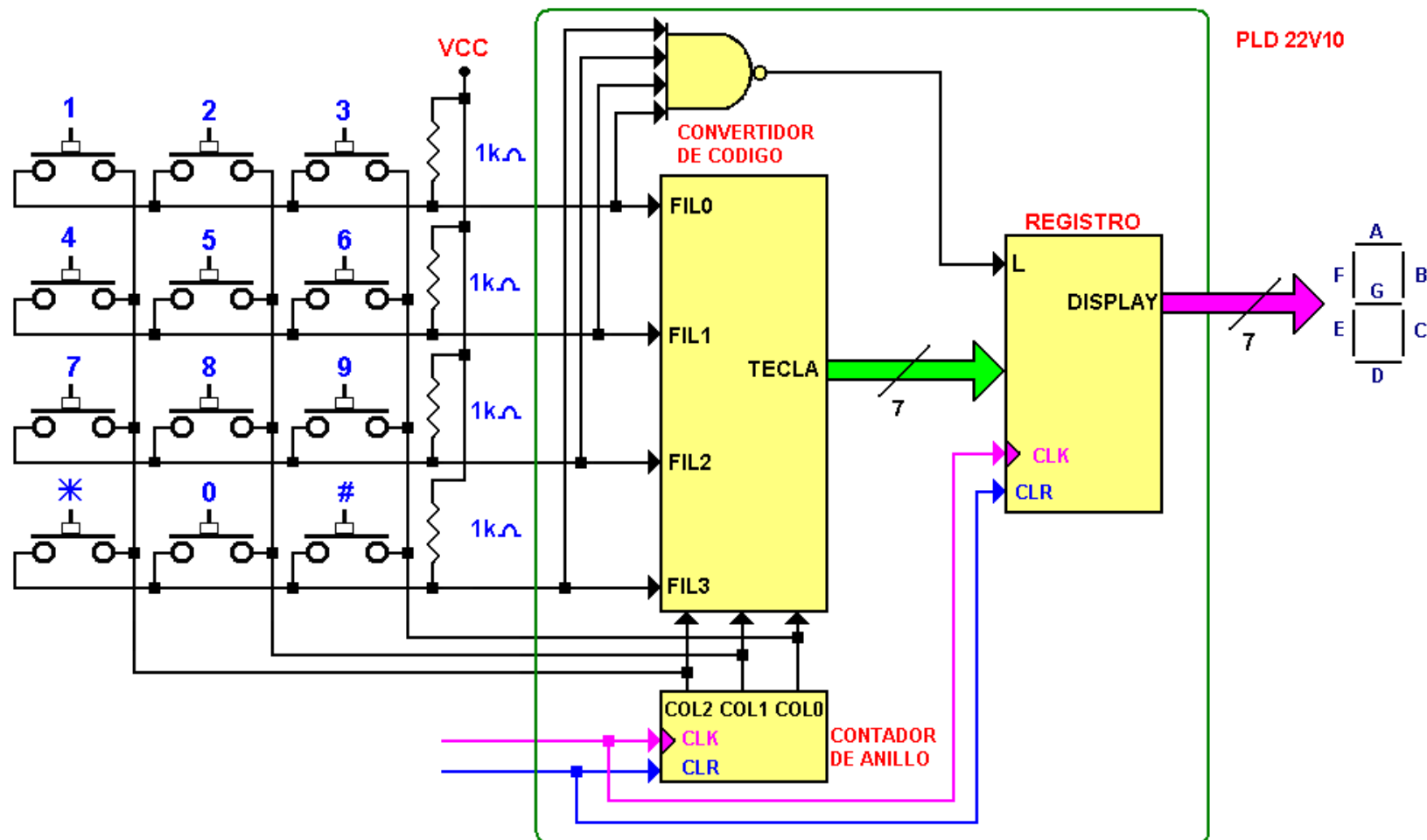


Ilustración 2 Microarquitectura del diseño a realizar.

Práctica 8 – Teclado Matricial

El convertidor de código convierte el valor de fila y columna en un código para enviar a un display de 7 segmentos de ánodo común. El valor en cada fila depende de la tecla que se presionó y del valor del contador de anillo. En la tabla 3 se muestran los valores en filas y columnas para identificar cuando se presionó cada tecla del teclado.

FIL3	FIL2	FIL1	FIL0	COL2	COL1	COL0	TECLA	SIMBOLO
1	1	1	0	0	1	1	1	1
							2	2
							3	3
1	1	0	1	0	1	1	4	4
							5	5
							6	6
1	0	1	1	0	1	1	7	7
							8	8
							9	9
0	1	1	1	0	1	1	*	A
							0	0
							#	g

Tabla 3 Convertidor de código.

NOTA: El alumno debe completar la tabla.

Los símbolos a desplegar deben codificarse como se muestra en la tabla 4.

Segmentos							Símbolo
A	B	C	D	E	F	G	
0	0	0	0	0	0	1	0
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2
0	0	0	0	1	1	0	3
1	0	0	1	1	0	0	4
0	1	0	0	1	0	0	5
0	1	0	0	0	0	0	6
0	0	0	1	1	1	1	7
0	0	0	0	0	0	0	8
0	0	0	0	1	0	0	9
0	0	0	1	0	0	0	A
0	0	0	0	1	0	0	g

Tabla 4 Códigos de los símbolos.

2. Simular los diseños en el ambiente de desarrollo.

3. Una vez simulado el sistema construir el circuito mostrado en la ilustración 3 para probarlo en el laboratorio. **En caso de usar la TEDDi este paso no es necesario.**



En el laboratorio:

1. Programar el PLD 22V10 usando el programador disponible del laboratorio.
2. Colocar la frecuencia de la señal de reloj a 10 HZ. En caso de usar TEDDi ajustar la frecuencia con el potenciómetro "FREC". En caso de haber armado el circuito en protoboard, usar el generador de funciones.
3. 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. ¿Qué frecuencia se debe utilizar para detectar la tecla presionada?
6. ¿Cuáles son las señales que funcionan de manera síncrona y cuales de manera asíncrona?
7. ¿Qué puedes concluir de esta práctica?

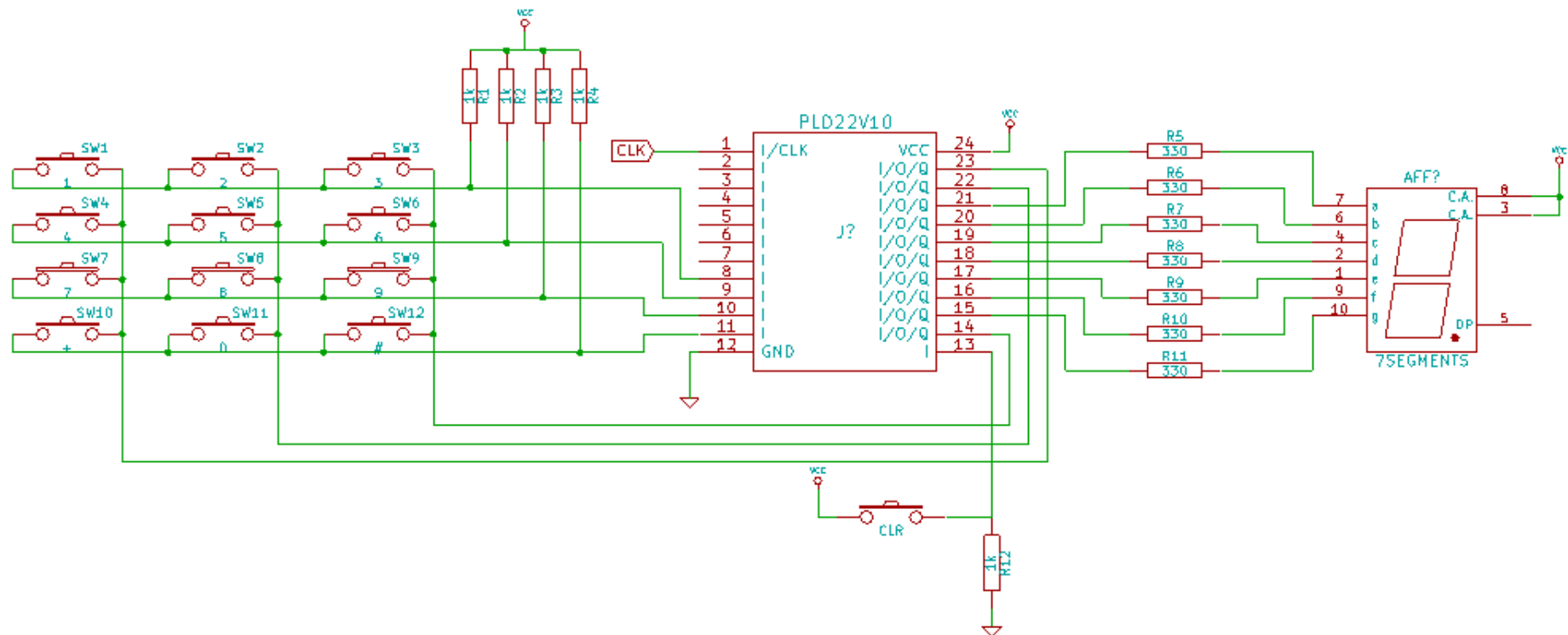


Ilustración 3 Diagrama esquemático.