



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
ACADEMIA DE SISTEMAS DIGITALES



DISEÑO DE SISTEMAS DIGITALES

Práctica de Laboratorio No. 9

“Tiempos importantes en los Flip-Flops”

Profesores:

VICTOR HUGO GARCIA ORTEGA
JULIO CESAR SOSA SAVEDRA

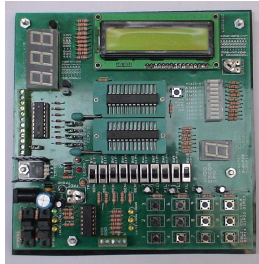
OBJETIVO: Diseñar, programar y probar un circuito que permita ver la importancia de cumplir con los tiempos de *setup* y *hold* de los Flip-Flops mediante un lenguaje de descripción de hardware (HDL) en un PLD 22V10.

MATERIAL Y EQUIPO:

Mesa de instrumentación del laboratorio de sistemas digitales

2 PLD 22v10

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

1 Fuente de 5V 1 Generador de funciones 1 Interruptor de presión 4 Resistencias de 1KΩ 7 Resistencias de 330Ω 1 Módulo de 3 displays multiplexados de ánodo común. 3 Transistores BC557 1 Protoboard Pinzas y cable para alambrar	1 TEDDi (Tarjeta Educativa para Diseño Digital). <div data-bbox="922 602 1182 864" data-label="Image">  </div>
---	--

INTRODUCCIÓN TEÓRICA

Los Flip-Flops (FF's) sincronizados por reloj se utilizan en una amplia variedad de aplicaciones. Estos FF's tienen una entrada de reloj que por lo general se identifica como CLK, CK o CP. Esta entrada de reloj es disparada por flanco de subida o bajada.

Los FF's sincronizados por reloj tienen una o más entradas de control que pueden tener varios nombres, dependiendo de su operación. Las entradas de control no tendrán efecto sobre Q sino hasta que ocurra la transición activa del reloj. Es decir, su efecto esta sincronizado con la señal CLK, a estas entradas se les conoce como **entradas de control síncronas**.

Deben de cumplirse dos requerimientos de sincronización si un FF controlado por reloj va a responder en forma confiable a sus entradas de control cuando ocurra la transición activa de CLK. Estos requerimientos son: el tiempo de Setup o estabilización (t_s) y el tiempo de Hold o retención (t_h).

El **tiempo de Setup o estabilización** (t_s) es el tiempo durante el cual la entrada de control debe mantenerse sin cambio, antes del flanco de la señal de reloj.

El **tiempo de Hold o retención** (t_h) es el tiempo durante el cual la entrada de control debe mantenerse sin cambio, después del flanco de la señal de reloj.

Estos tiempos pueden observarse en la ilustración 1.

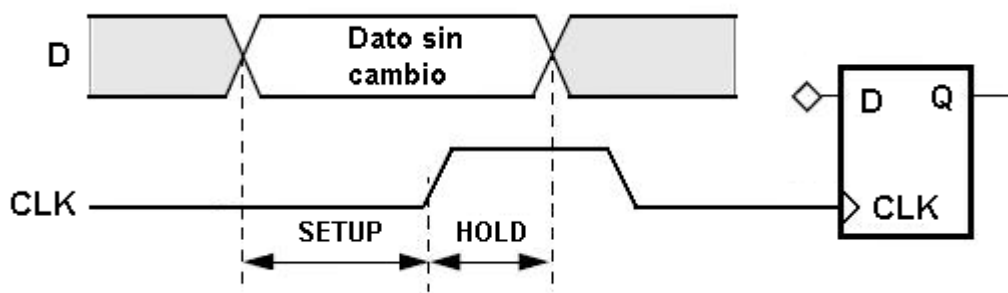


Ilustración 1 Tiempos de Setup y Hold.

Cuando no se cumplen los tiempos de Setup y Hold, ocurre un **asincronismo**. En el asincronismo los FF's pueden tomar el valor de cero lógico, uno lógico o un valor indefinido. El caso más crítico es cuando los FF's toman un valor indefinido, en ese momento los FF's entran en un **estado metaestable**.

Para evitar los asincronismos y el estado metaestable debemos colocar dos FF's conectados en serie en cada una de las entradas del diseño digital. Si ocurre un asincronismo, el primer FF tomará el valor de cero lógico, uno lógico o un valor indefinido (estado metaestable). Si el FF toma alguno de los dos primeros valores, el asincronismo queda eliminado. Esto se muestra en la ilustración 2.

Por diseño electrónico si un FF tiene a la entrada un valor indefinido, colocará a su salida un cero cuando llegue el siguiente flanco activo de la señal de reloj. Usando este hecho, en caso de que el primer FF tenga un estado metaestable, el segundo FF colocará a su salida un cero cuando llegue el siguiente flanco activo de la señal de reloj. Con el segundo FF se elimina el estado metaestable.

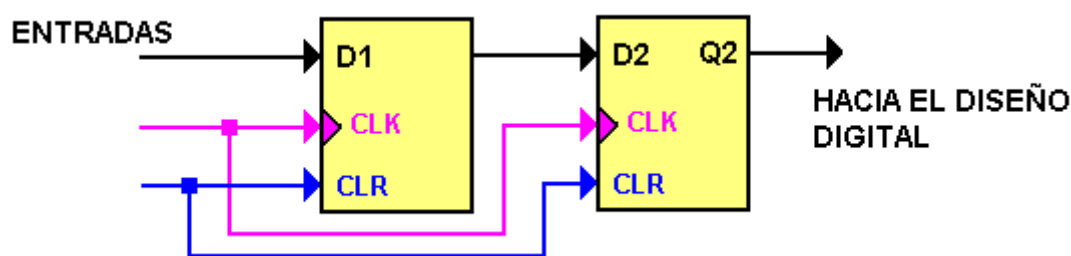


Ilustración 2 Circuito para eliminar asincronismo y metaestabilidad.

PROCEDIMIENTO.

Antes de asistir al laboratorio:

1A. Realizar un programa en HDL que permita implementar el diseño mostrado en el diagrama a bloques de la ilustración 3.

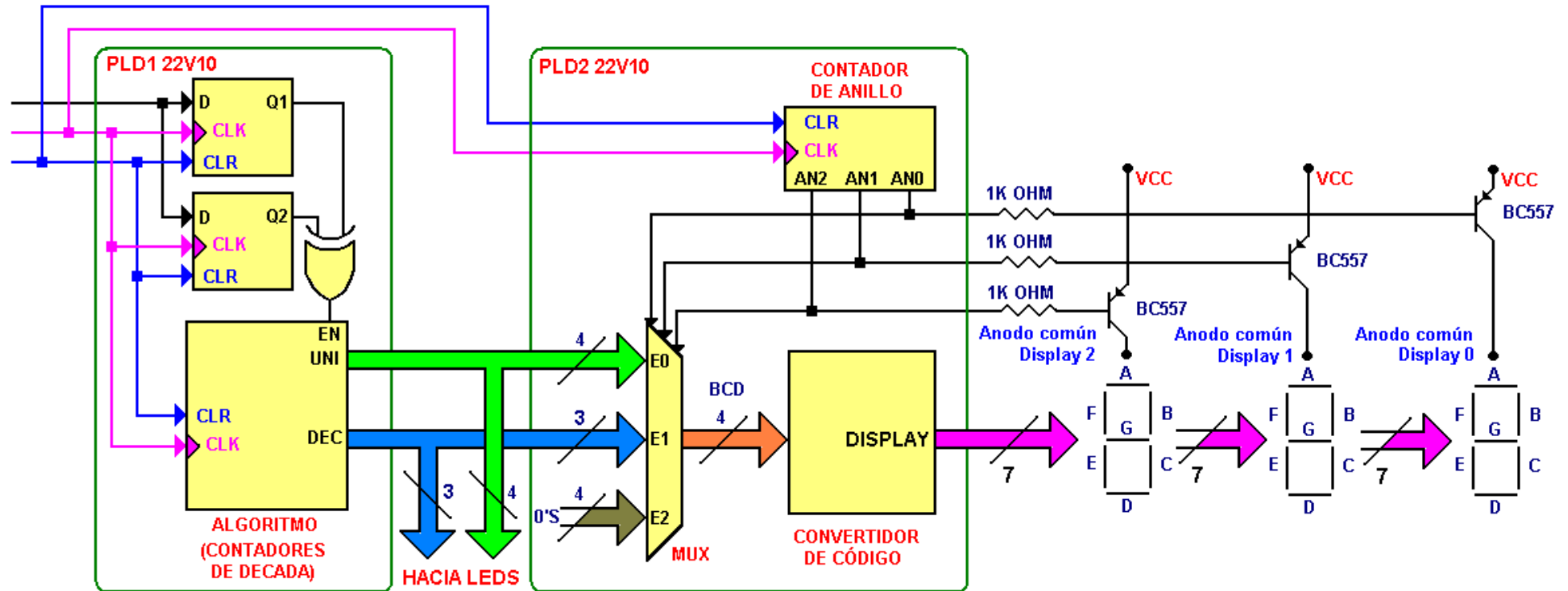


Ilustración 3 Microarquitectura del diseño a realizar.

El diseño propuesto esta distribuido en dos PLD22V10. A continuación se explica los bloques lógicos que contiene cada PLD.

El PLD1 contiene dos FF's que tienen la misma señal de entrada. A la salida de ambos FF's se tiene una compuerta XOR que habilitará a los contadores BCD cuando la salida de los FF's sea diferente.

De los contadores BCD, uno de ellos es de 4 bits y maneja las unidades con el rango de 0 a 9. El otro contador es de 3 bits y maneja las decenas con el rango de 0 a 7. Estos contadores se deben programar mediante un algoritmo. Este algoritmo se muestra en la ilustración 4.

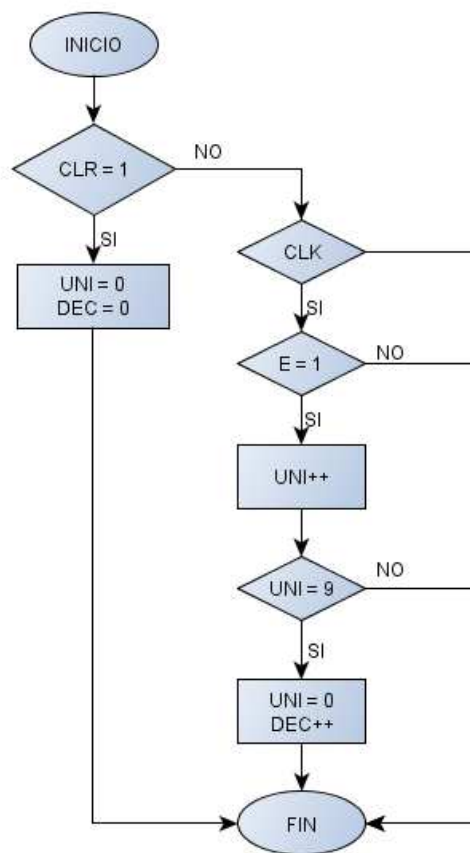


Ilustración 4 Algoritmo de los contadores BCD.

El PLD2 contiene un contador de anillo, un multiplexor de selección de los contadores BCD y el convertidor de BCD a 7 segmentos. Considere lo siguiente:

- El multiplexor tiene como señales de selección las 3 salidas del contador de anillo. Al tener 3 señales de selección es posible tener 8 entradas en el multiplexor, de las cuales solo 3 son de interés para este diseño, por lo que el resto de las entradas de selección se deben tomar como TÉRMINOS NO IMPORTA. Su salida se obtiene de acuerdo a la tabla 1

ENTRADAS			SALIDA
AN2	AN1	AN0	
1	1	0	BCD = E0
1	0	1	BCD = E1
0	1	1	BCD = E2
0	0	0	X
0	0	1	X
0	1	0	X
1	0	0	X
1	1	1	X

Tabla 1 Funcionamiento del multiplexor.

- El contador de anillo es de 3 bits. 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 contador de anillo funciona de acuerdo a la tabla 2.

SALIDAS			DISPLAY SELECCIONADO
AN2	AN1	AN0	
1	1	0	D0
1	0	1	D1
0	1	1	D2

Tabla 2 Funcionamiento del contador de anillo.

- El convertidor de código es un convertidor de BCD a 7 segmentos. Hay que considerar que el módulo de 3 displays multiplexados es de ánodo común, por lo que los segmentos se encienden con ceros. Este convertidor de código se muestra en tabla 3.

Código BCD				Segmentos							Dígito
BCD(3)	BCD(2)	BCD(1)	BCD(0)	A	B	C	D	E	F	G	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7



Práctica 9 – Tiempos importantes en los Flip-Flops

1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9
1	0	1	0	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X

Tabla 3 Funcionamiento del convertidor de código.

1B. Agregar un FF en la entrada D de ambos FF's tal como se muestra en la ilustración 5 y verificar el funcionamiento del diseño.

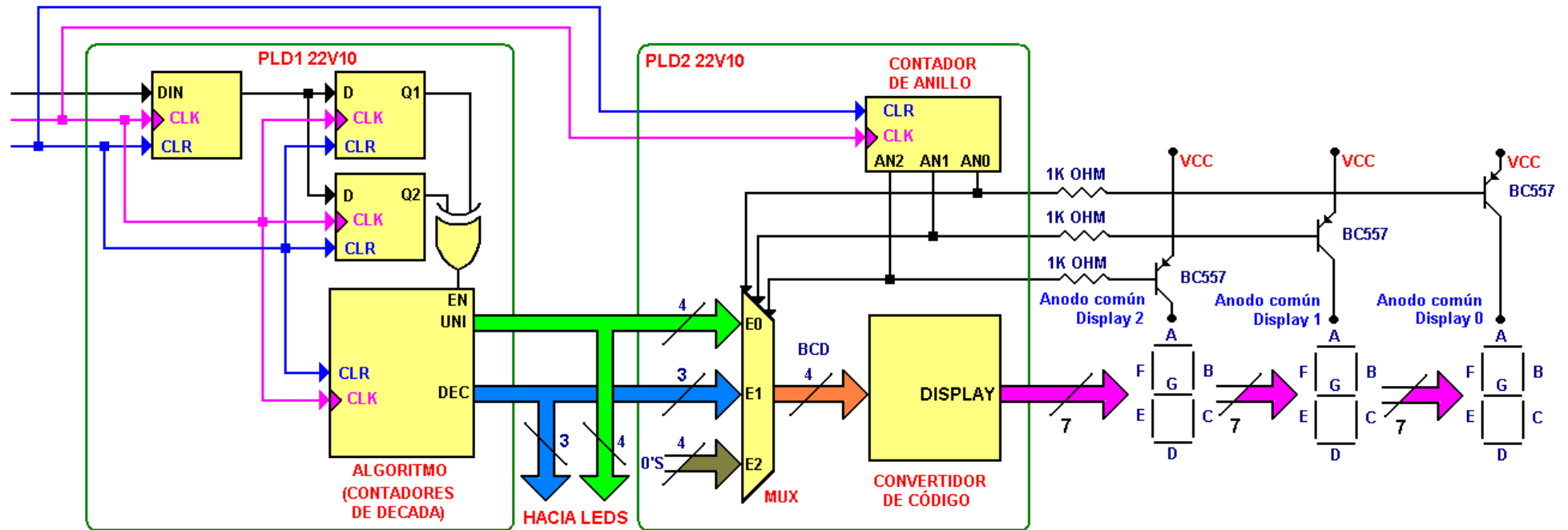


Ilustración 5 Microarquitectura 2 del diseño a realizar.



2. Simular los diseños en el ambiente de desarrollo.
3. Una vez simulado el sistema construir el circuito mostrado en la ilustración 6 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 150 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. Obtener una señal cuadrada del generador de funciones de 1 Hz entre 0V y 5V (TTL Output).
4. Colocar esta señal en la entrada 4 del PLD1 22V10 (Señales D y DIN de las microarquitecturas).
5. Aumentar la frecuencia de la señal de reloj gradualmente hasta alcanzar los 10 MHz.
6. 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 PLD1 22V10 y PLD2 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 PLD1 22V10 y PLD2 22V10?
5. ¿A partir de qué frecuencia se presentan asincronismos?
6. ¿Cómo se eliminan los asincronismos en un diseño digital?
7. ¿Cómo se elimina el estado metaestable en un diseño digital?
8. ¿Qué puedes concluir de esta práctica?

Práctica 9 – Tiempos importantes en los Flip-Flops

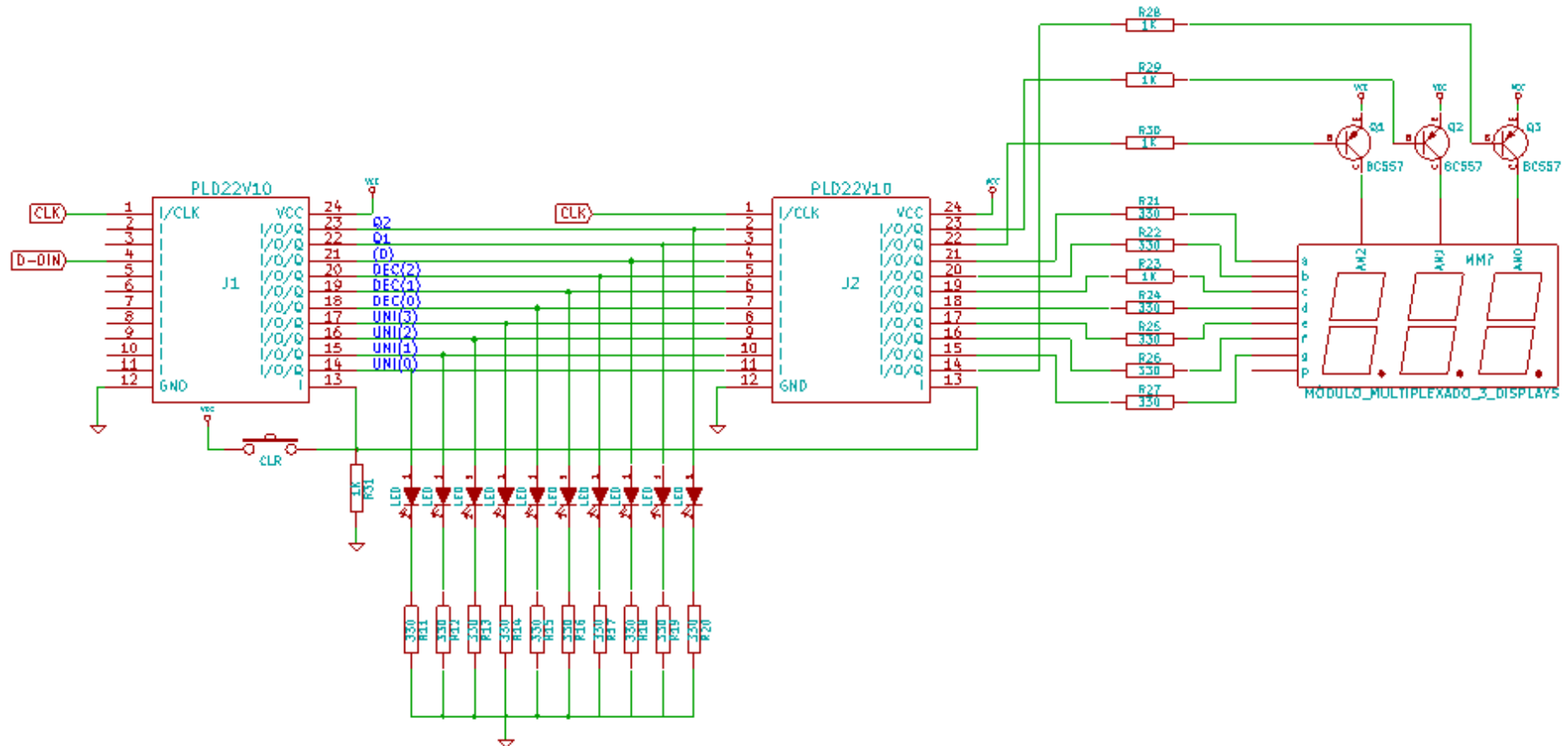


Ilustración 6 Diagrama esquemático.