



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
ACADEMIA DE SISTEMAS DIGITALES



DISEÑO DE SISTEMAS DIGITALES

Práctica de Laboratorio No. 10

“Sensores”

Profesores:

VICTOR HUGO GARCIA ORTEGA
JULIO CESAR SOSA SAVEDRA

OBJETIVO: Diseñar, programar y probar un circuito que determine si una persona entró o salió de una habitación y se contabilice el número de personas en la habitación mediante dos sensores ópticos utilizando SPLD's. La estrategia de diseño será usando máquinas de estados y diseño algorítmico a través del IDE Galaxy. El conteo de debe desplegar en un módulo de 3 displays multiplexados de ánodo común.

MATERIAL Y EQUIPO:

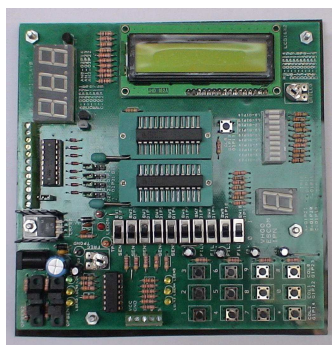
Mesa de instrumentación del laboratorio de sistemas digitales

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 1K Ω
2 Resistencias de 10K Ω
21 Resistencias de 330 Ω
1 Módulo de 3 displays multiplexados de ánodo común.
3 Transistores BC557
2 Opto interruptores H21B1
1 Protoboard
Pinzas y cable para alambrar

1 TEDDi (Tarjeta Educativa para Diseño Digital).



INTRODUCCIÓN TEÓRICA

Existen diversas aplicaciones donde se pueden usar máquinas de estados para resolver cierta problemática. Una de esas aplicaciones consiste en identificar el número de personas que pasan por una puerta de acceso en un sentido u otro. Esta aplicación puede usarse para el conteo de personas en un recinto (salón, teatro, auditorio, estadio, etc), en una fila de un banco, que suben a un autobús, etc.

Vamos a tomar como ejemplo un sistema que pueda contar el número de personas que entran o salen de un salón de clases. La cantidad de personas que ingresan al salón de clases se mostrará en un módulo de 3 displays multiplexados de ánodo común. Para esta aplicación se necesitan dos pares de sensores ópticos (S1 y S0) para detectar la entrada o salida de personas. Los sensores deben de estar colocados a la entrada del recinto como se muestra en la ilustración 1. Cada vez que una persona entre al recinto se incrementarán unos contadores BCD. De forma contraria si una persona sale del recinto los contadores serán decrementados. El valor de los contadores BCD será decodificado y mostrado en el módulo de 3 displays multiplexados.

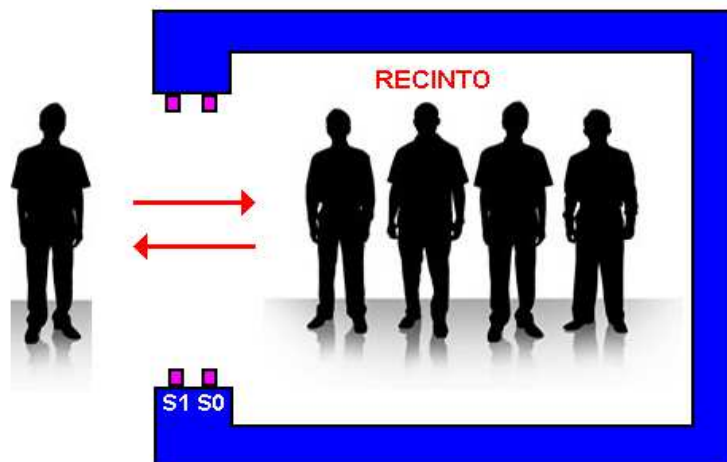


Ilustración 1 Aplicación con sensores.

PROCEDIMIENTO.Antes de asistir al laboratorio:

1. Realizar los programas en VHDL siguiendo la microarquitectura de la Ilustración 2. El diseño propuesto está distribuido en dos PLD 22V10. A continuación se explica los bloques lógicos que contiene cada PLD.

El PLD1 contiene la máquina de estados (FSM – Finite State Machine) para determinar si una persona entra o sale de una habitación dependiendo de cómo se activen los sensores ópticos y el algoritmo para realizar el control en los contadores de década de las unidades y las decenas. Considere lo siguiente:

- Este modelo es una máquina de Mealy la cual contiene 8 estados, tal como se muestra en la ilustración 3.
- La estrategia de diseño a usar en el programa será describiendo la máquina de estados mediante la definición de los estados usando TYPE.
- La salida de la máquina de estados está en función de la tabla 1.

| SAL1 | SAL0 | Operación |
|------|------|--|
| 0 | 0 | Retención |
| 0 | 1 | Conteo Ascendente (entró una persona) |
| 1 | 0 | Conteo Descendente (salió una persona) |

Tabla 1 Salidas de la FSM.

El bloque del algoritmo actualiza los contadores de década dependiendo de las salidas de la máquina de estados. Este algoritmo se muestra en la ilustración 4.

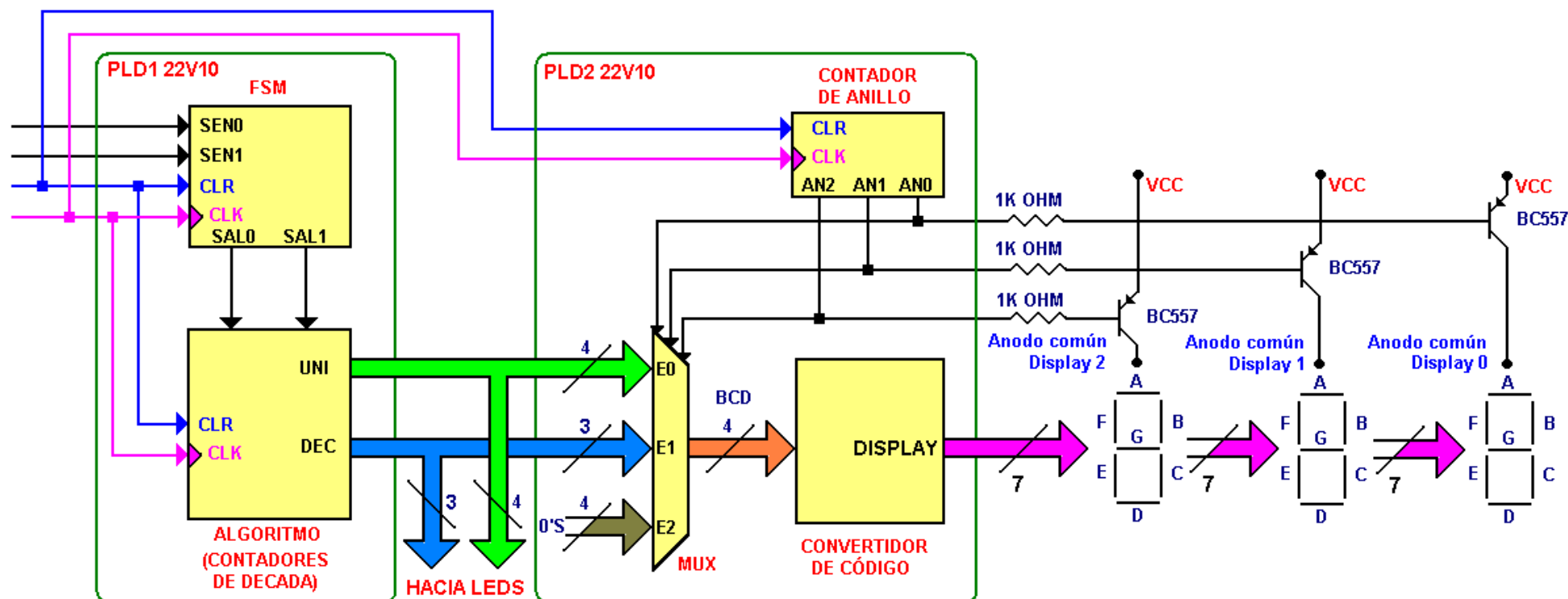


Ilustración 2 Microarquitectura del diseño a realizar.

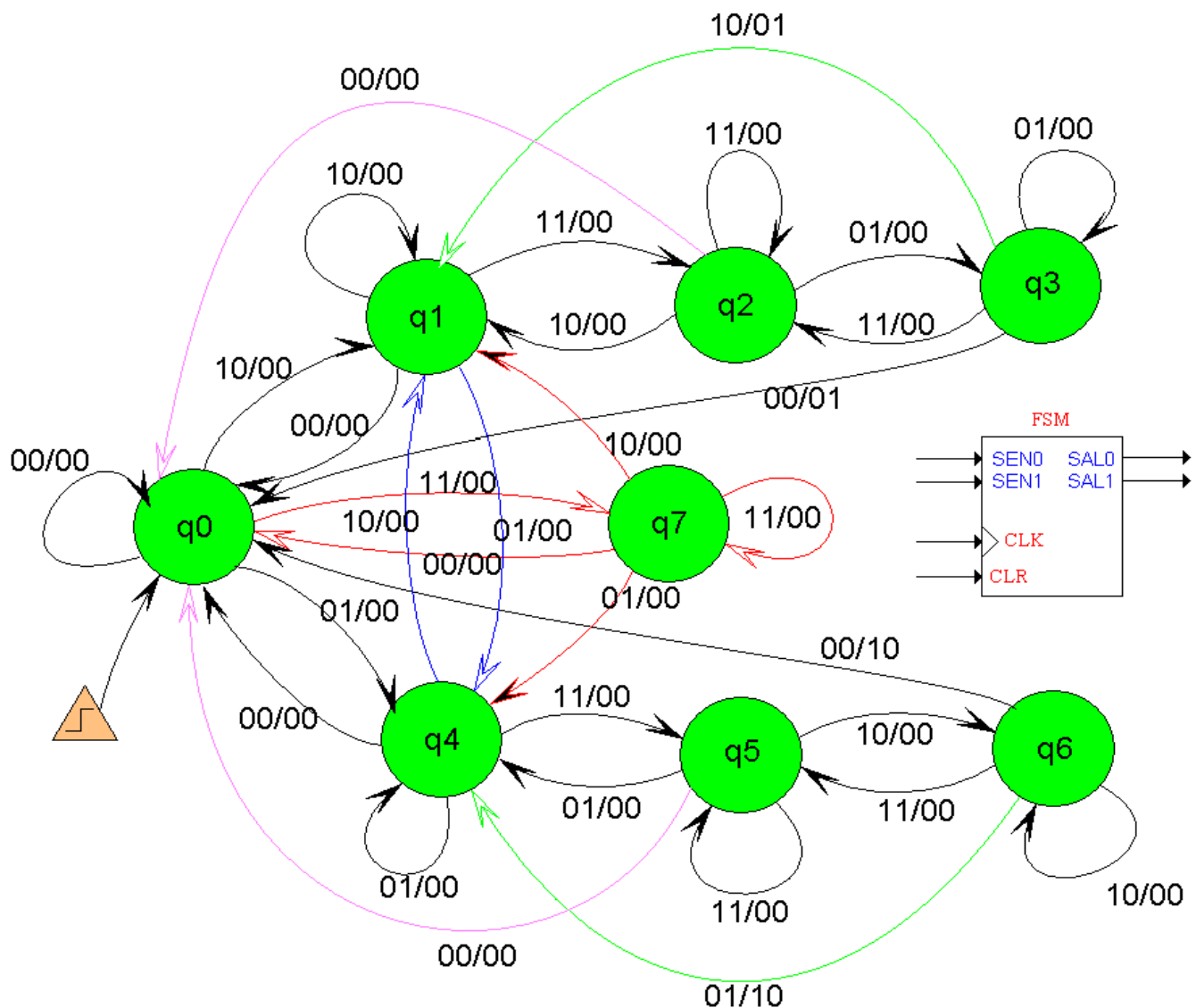


Ilustración 3 Máquina de Mealy para los sensores.

El PLD2 contiene un contador de anillo, un multiplexor de selección de los contadores de década 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 2

| 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 2 Funcionamiento del multiplexor.

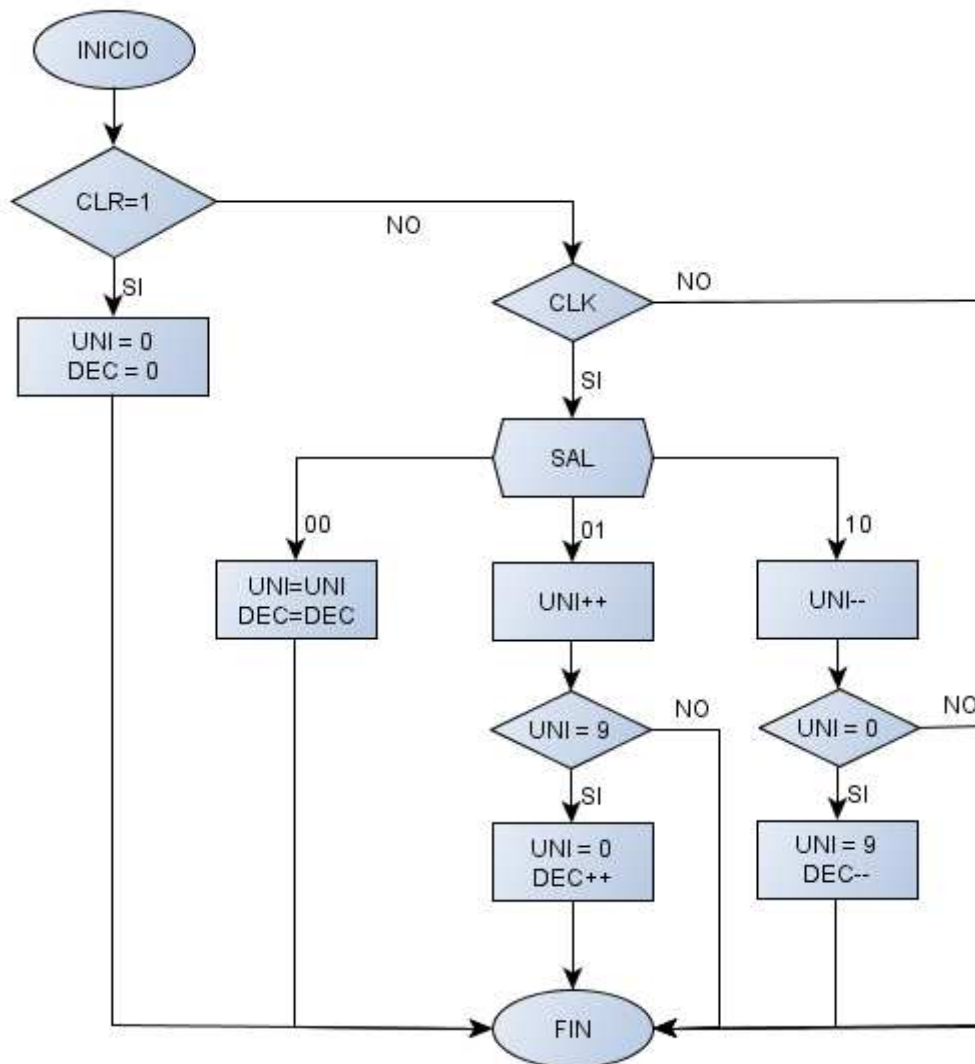


Ilustración 4 Algoritmo de los contadores de década.

- 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 3.

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

Tabla 3 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 la tabla 4.

| Código BCD | | | | Segmentos | | | | | | | Digito |
|------------|--------|--------|--------|-----------|---|---|---|---|---|---|--------|
| 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 |
| 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 |
| 1 | 1 | 1 | 1 | X | X | X | X | X | X | X | X |

Tabla 4 Funcionamiento del convertidor de código.



2. Simular los diseños en el ambiente de desarrollo.
3. Una vez simulado el sistema construir el circuito mostrado en la ilustración 5 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. Pasar una moneda por los sensores ópticos.
4. 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. ¿Qué puedes concluir de esta práctica?

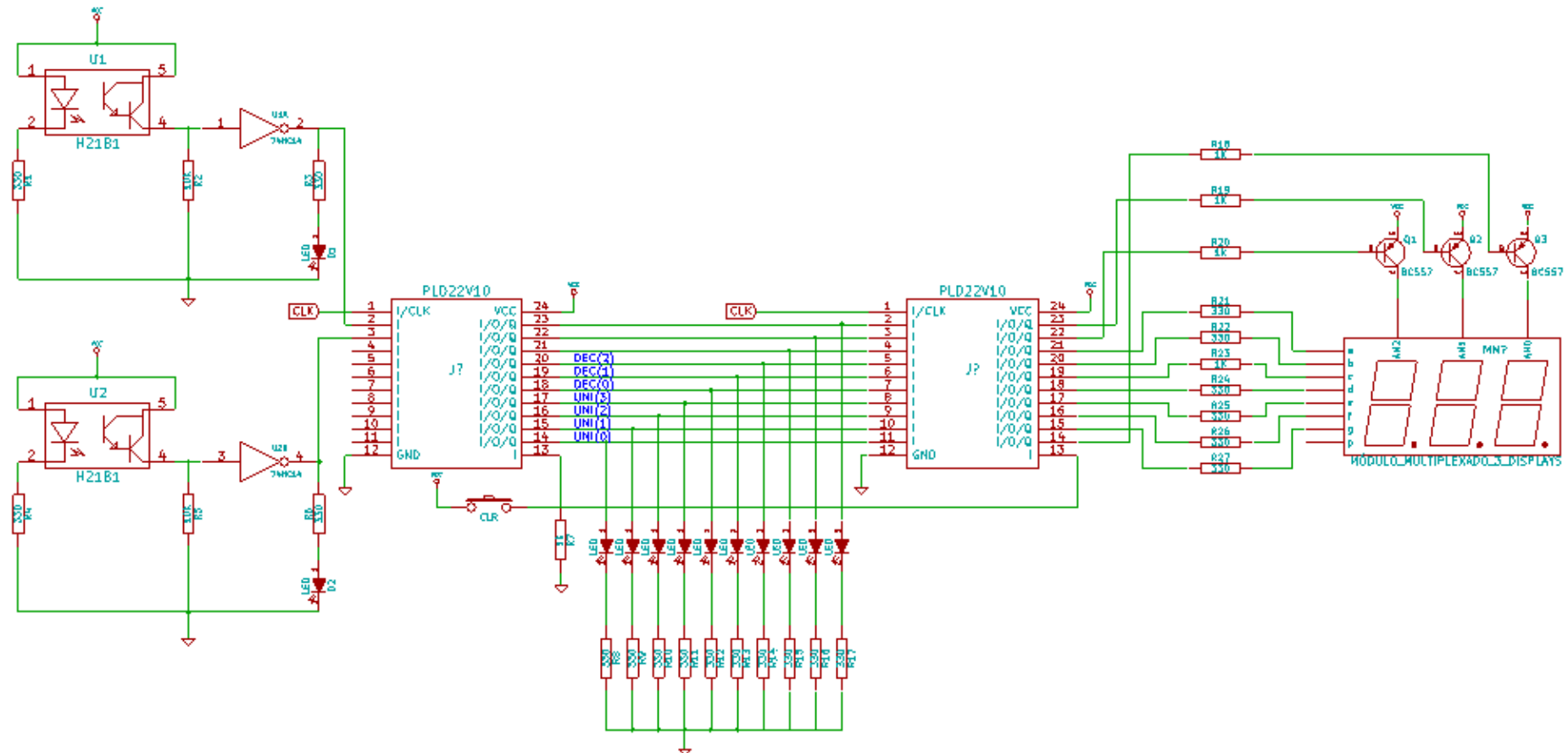


Ilustración 5 Diagrama esquemático.