# Decodificador Binario a Decimal

Estilo de desarrollo: Estructural, funcional y flujo de datos

## ¿Qué es lo que hará la arquitectura?

La entidad principal tiene como mision convertir una entrada binaria de 8 bits en su equivalente entendible para nosotros los humanos, en otras palabras, convertir un numero binario a numero decimal.

Nombre de la entidad: **dec\_ctrl\_bin2dec7seg**

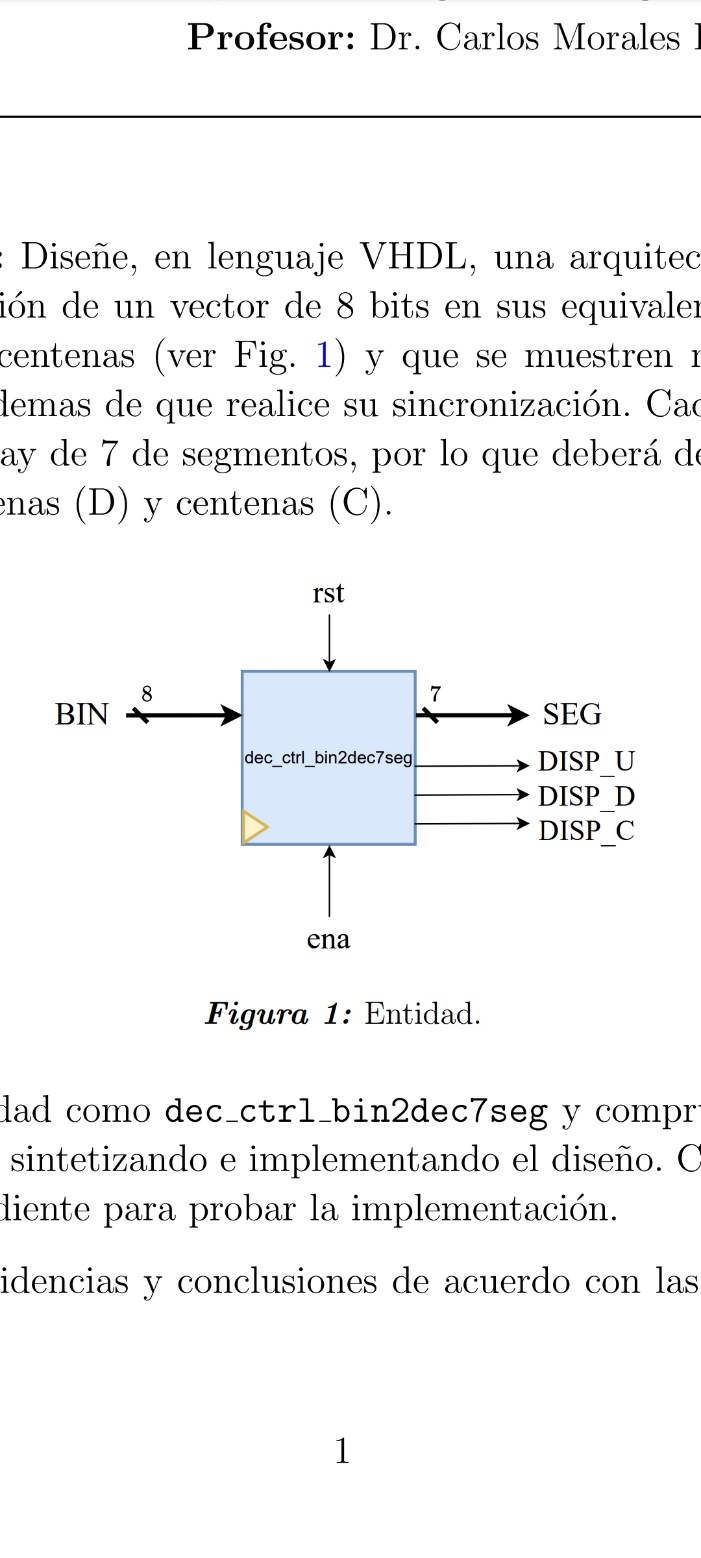


Fig. 1 Entidad principal: dec\_ctrl\_bin2dec7seg

## Desarrollo

### Arquitectura planteada

Para llevar a cabo el proceso deseado se implementó la siguiente arquitectura

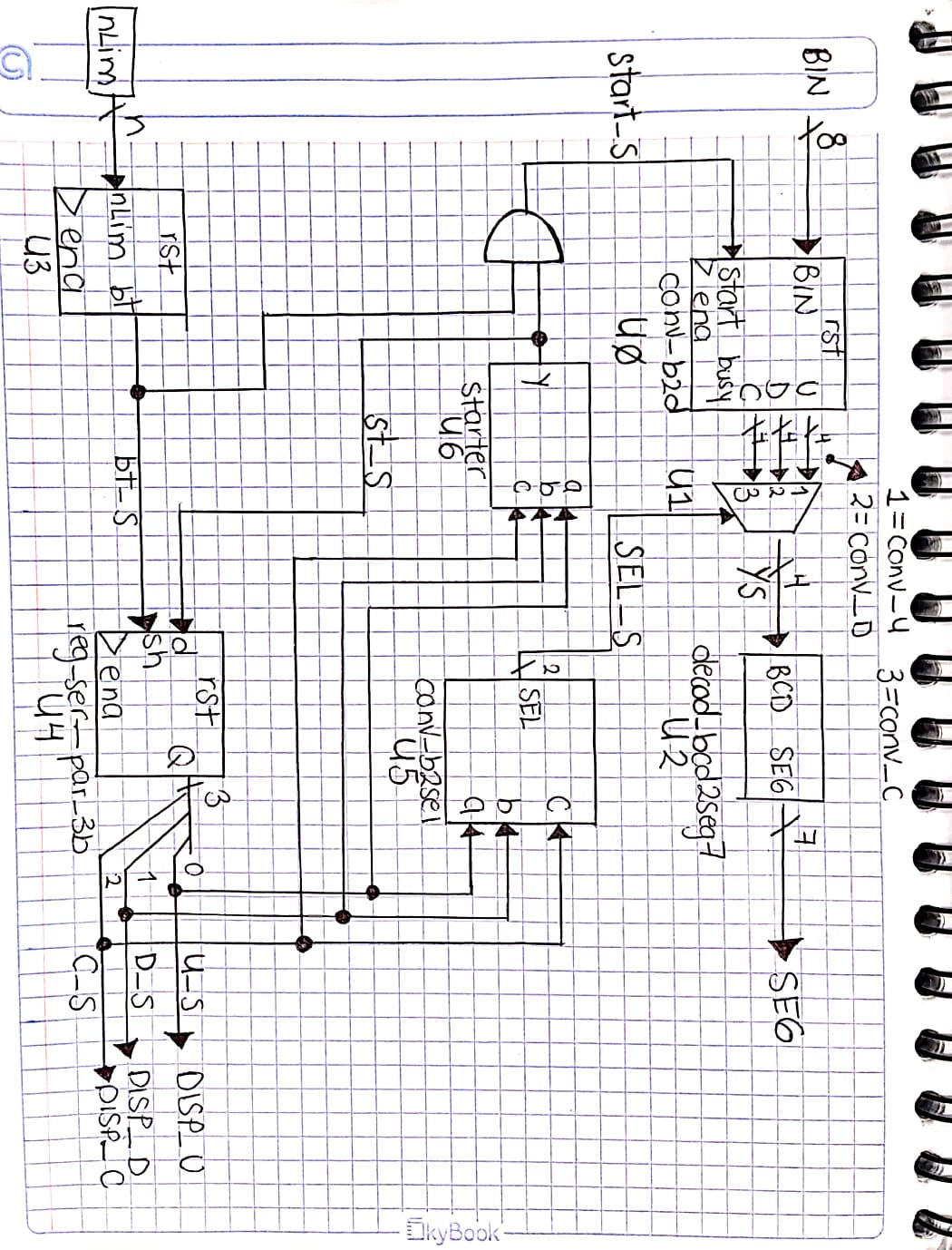


Fig. 2 Arquitectura de la entidad principal

### Identificación de los componentes

|  |  |
| --- | --- |
| No. de dispositivo | Nombre de la entidad |
| U0 | conv\_b2d |
| U1 | mux\_3\_to\_1\_4b |
| U2 | decod\_bcd2seg7 |
| U3 | base\_time reg\_n\_bits comp\_n\_bits adder\_nbits |
| U4 | reg\_ser\_par\_3b |
| U5 | conv\_b2sel |
| U6 | Starter |

Para ver las descripciones completas:

### Entradas, señales y salidas de la arquitectura

|  |  |  |
| --- | --- | --- |
| **Entradas** | **Señales** | **Salidas** |
| BIN (7 downto 0)  clk ena rst | **Señales de 1 bit** bt\_s  st\_s  start\_s  **Señales de 2 bits**  SEL\_s  **Señales de 4 bits**  conv\_U  conv\_D  conv\_C  Ys | SEG (6 downto 0) DISP\_U DISP\_D DISP\_C |

## Comportamiento de las entidades usadas

### **U0: conv\_b2d**

Es la entidad de mayor impotancia dentro de la arquitectura, por medio de un algoritmo se encarga de convertir una entrada binaria de 8 bits a un numero decimal de 3 digitos.

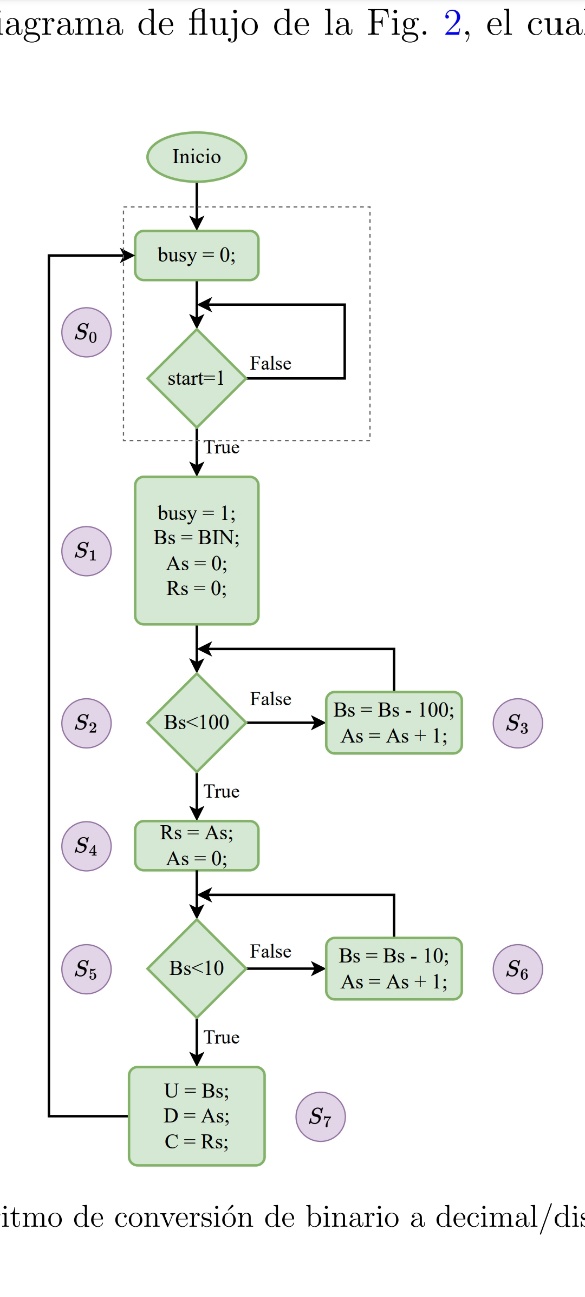
**

Fig. 3 Algoritmo conv\_b2d

### **U1: mux\_3\_to\_1\_4b**

Durante la ejecución de la descripción de hardware, la función del multiplexor es seleccionar que “bus” de datos estará a la salida de acuerdo al selector del mismo.

Para este caso en concreto:

|  |  |
| --- | --- |
| S | Y |
| 00 | -- |
| 01 | Unidades |
| 10 | Decenas |
| 11 | Centenas |

Donde S es la entrada de la entidad y Y es la salida seleccionada

### **U2: decod\_bcd2seg7**

La función del decodificador es traducir un numero binario de 4 bits a una salida de 6 bits que corresponde al numero binario entrante.

La tabla de verdad a continuación demuestra el comportamiento en las descripción del archivo:

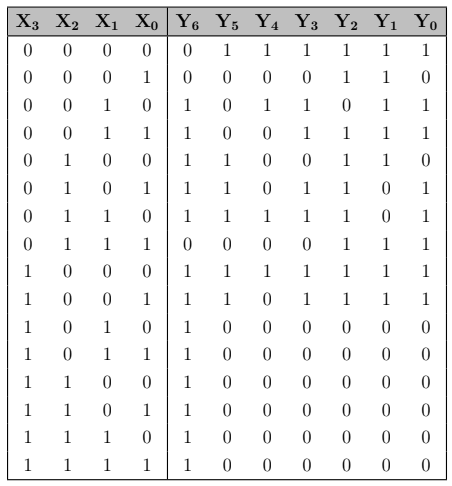


Fig. 4Tabla de verdad decod\_7seg

### **U3: base\_time**

Esta entidad a grandes rasgos, controla la velocidad de multiplicación de los displays; es decir, gobierna la velocidad de respuesta de todos los demás componentes.

Dentro de esta entidad existe una señal a la cual se le debe asignar un numero en concreto la cual determina dicha velocidad “nLim”.

Cálculos para nLim:

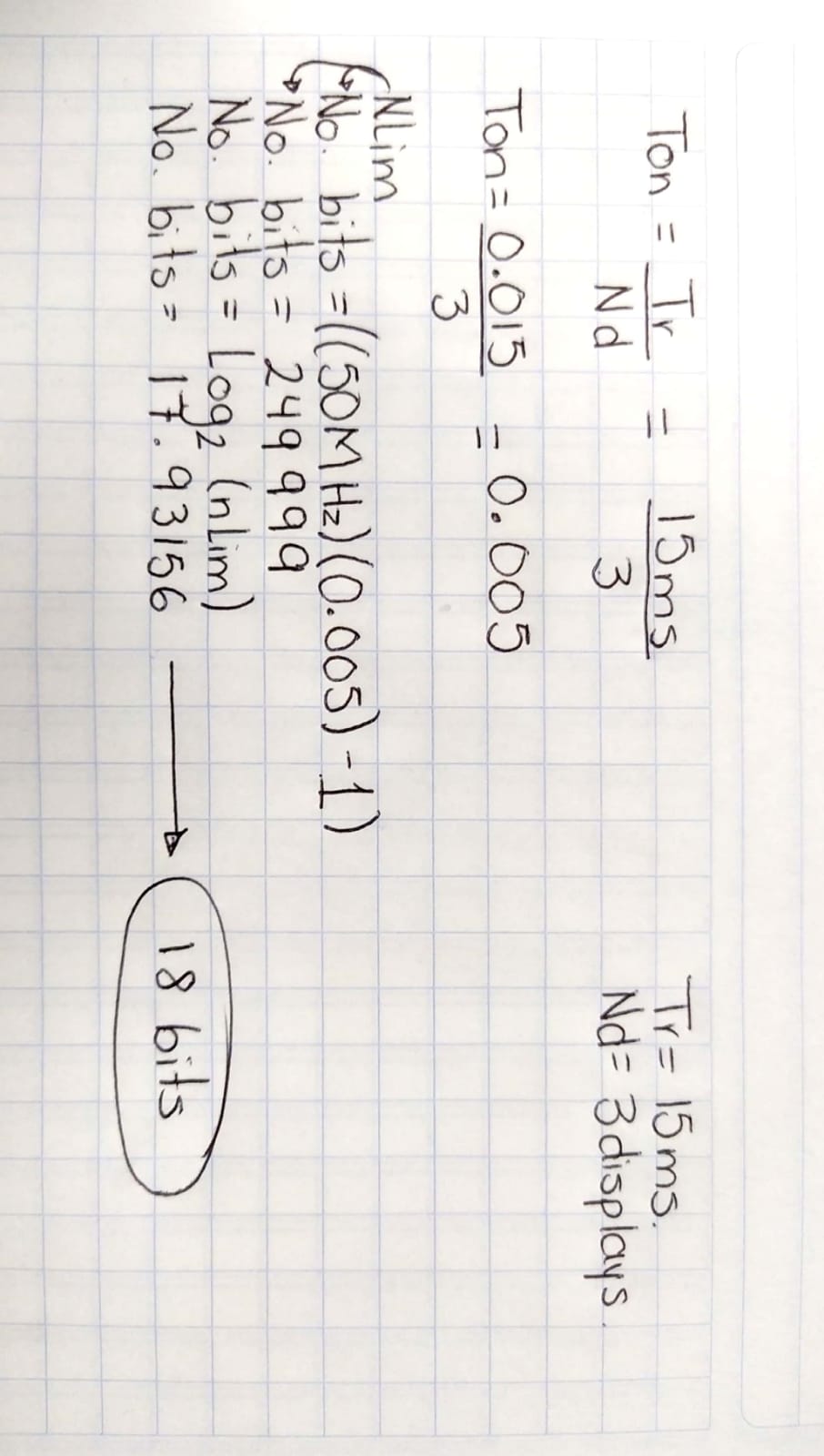


Fig. 5 Cálculos para nLim

Para la descripción se usó el numero decimal 249,999 que en hexadecimal es “3d08f”

signal nLim : std\_logic\_vector (19 downto 0):=x"3d08f"; --249,999 15ms

### **U4: reg\_ser\_par\_3b**

Un registro de 3 bits serie-paralelo consta en una serie de flip-flops dispuestos uno tras de otro de tal manera que cuando se detecte un flanco de subida de la señal de reloj, los bits de la única entrada se irán recorriendo para que a cada salida se vea reflejado el bit que le corresponde.

### **U5: conv\_b2sel**

Este dispositivo se construyo con estilo de descripcion de flujo de datos.

Con este dispositivo obtendremos la señal que seleccionara que dato se decodificara con “decod\_bcd2seg7”.

Tabla de verda del dispositivo descrito en VHDL

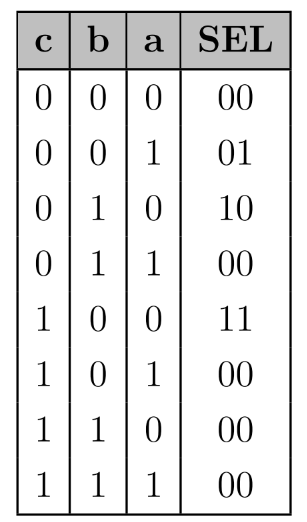


Fig. 6 Tabla de verdad conv\_b2sel

### **U6: starter**

Es el dispositivo donde se obtendrá la señal que inicializará el algoritmo del dispositivo “con\_b2d”

Tabla de verdad:

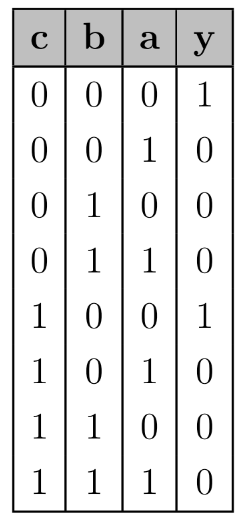


Fig. 7 Tabla de verdad starter

## Esquemático de conexiones fisicas

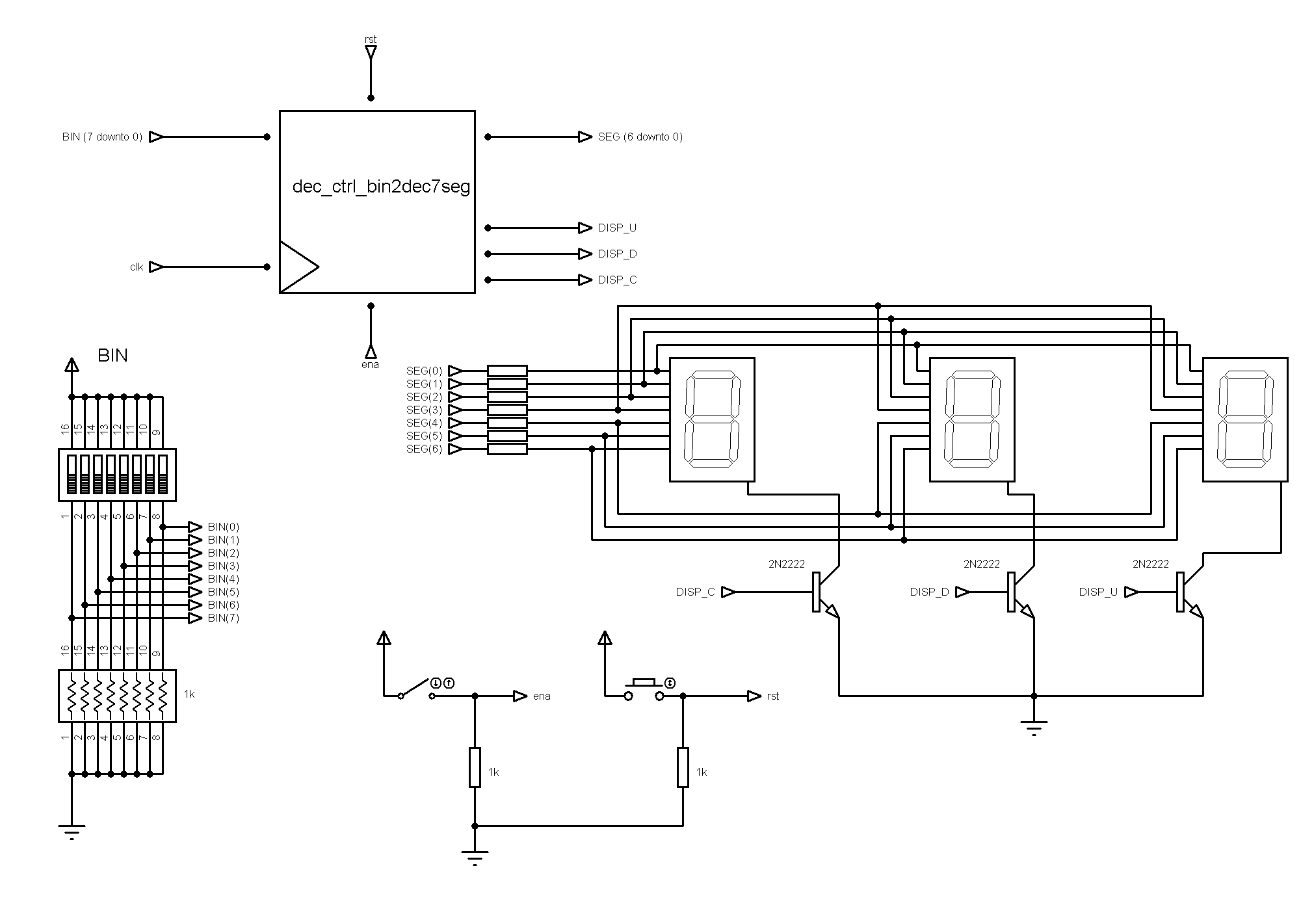


Fig. 8 Esquemático

## Asignación de pines a entradas y salidas en la placa de desarrollo

Conexiones Displays SEG

|  |  |  |  |
| --- | --- | --- | --- |
| Vector | Segmento | **PIN** | Color |
| Seg(0) | A | **70** | Morado |
| Seg(1) | B | **71** | Azul |
| Seg(2) | C | **72** | Verde |
| Seg(3) | D | **73** | Amarillo |
| Seg(4) | E | **74** | Naranja |
| Seg(5) | F | **75** | Rojo |
| Seg(6) | G | **76** | Café |

Conexiones para los transistores

|  |  |
| --- | --- |
| Vector | **PIN** |
| DISP\_U | **3** |
| DISP\_D | **2** |
| DISP\_C | **1** |

Conexiones para los Deep Switch BIN

|  |  |  |
| --- | --- | --- |
| Vector | **PIN** | Color |
| BIN(0) | **41** | Gris |
| BIN(1) | **40** | Blanco |
| BIN(2) | **39** | Negro |
| BIN(3) | **38** | Café/dorado |
| BIN(4) | **37** | Rojo |
| BIN(5) | **36** | Naranja |
| BIN(6) | **35** | Amarillo |
| BIN(7) | **34** | Verde |

Conexiones para ena y rst

|  |  |
| --- | --- |
| Conexión | **PIN** |
| ena | **30** |
| Rst | **33** |

## Resumen de configuración en Quartus

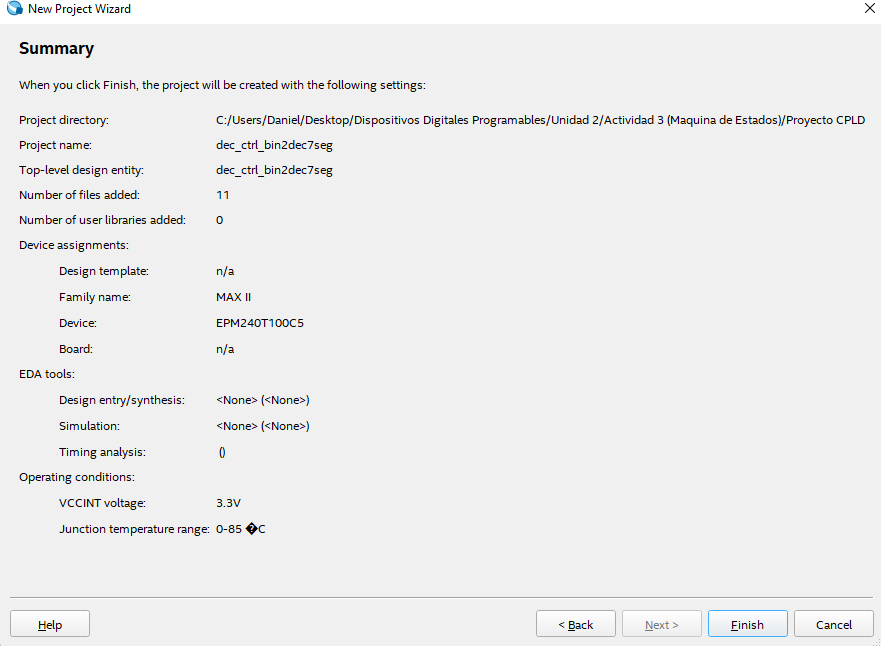


Fig. 9 Summary

## Evidencia fotográfica

Circuito en físico con placa de desarrollo

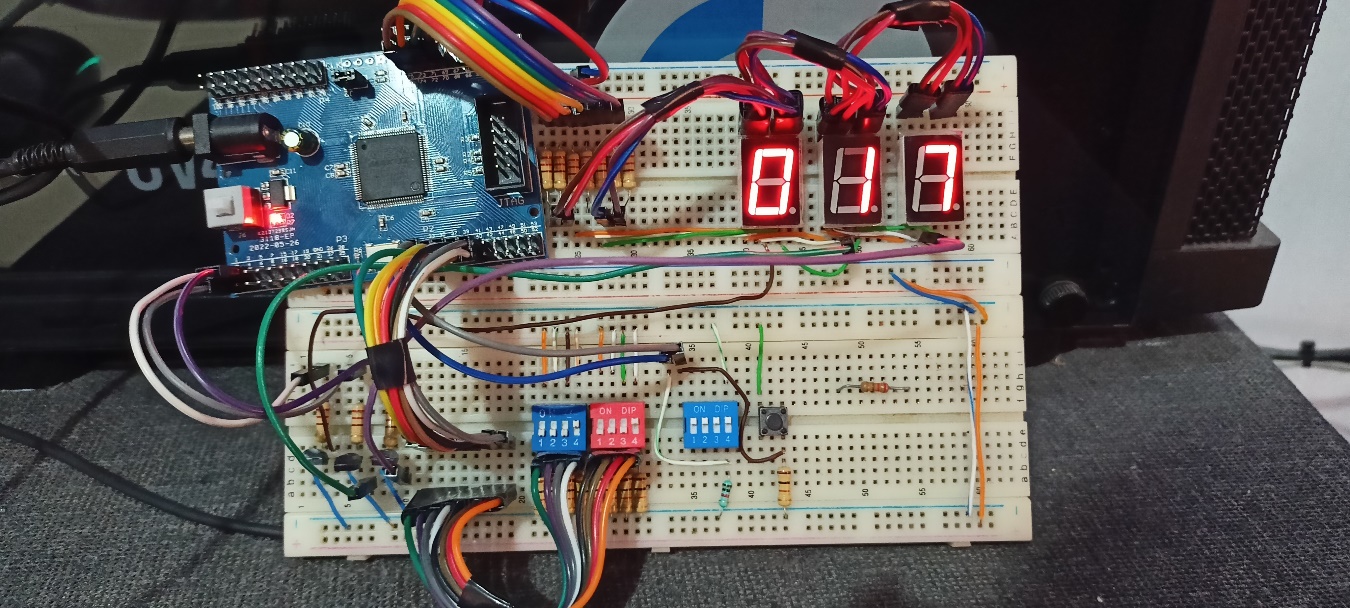


Fig. 10 Binario: 10001

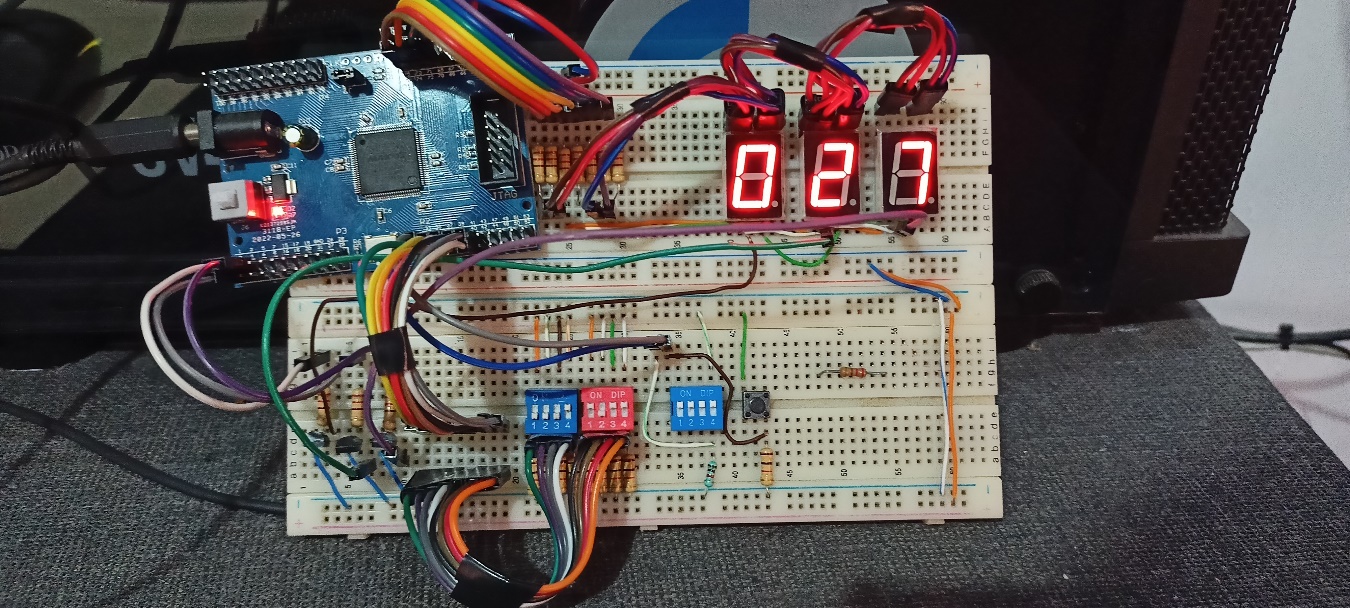


Fig. 11 Binario: 11011

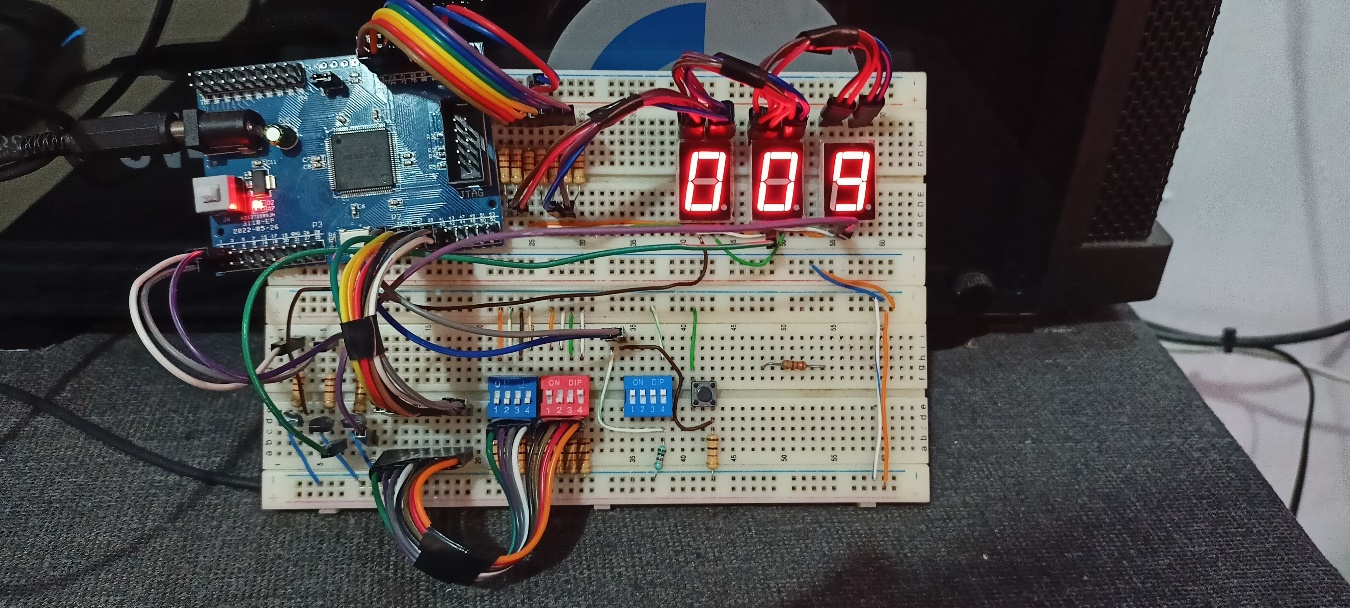


Fig. 12 Binario: 1001

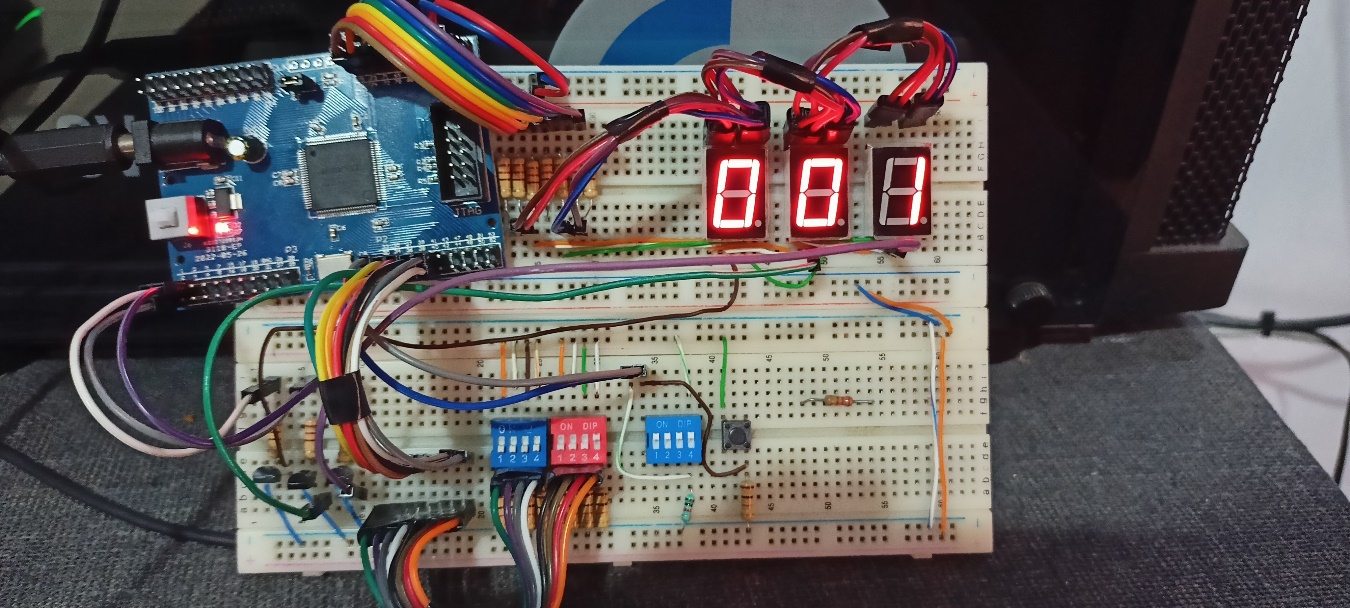


Fig. 13 Binario: 1

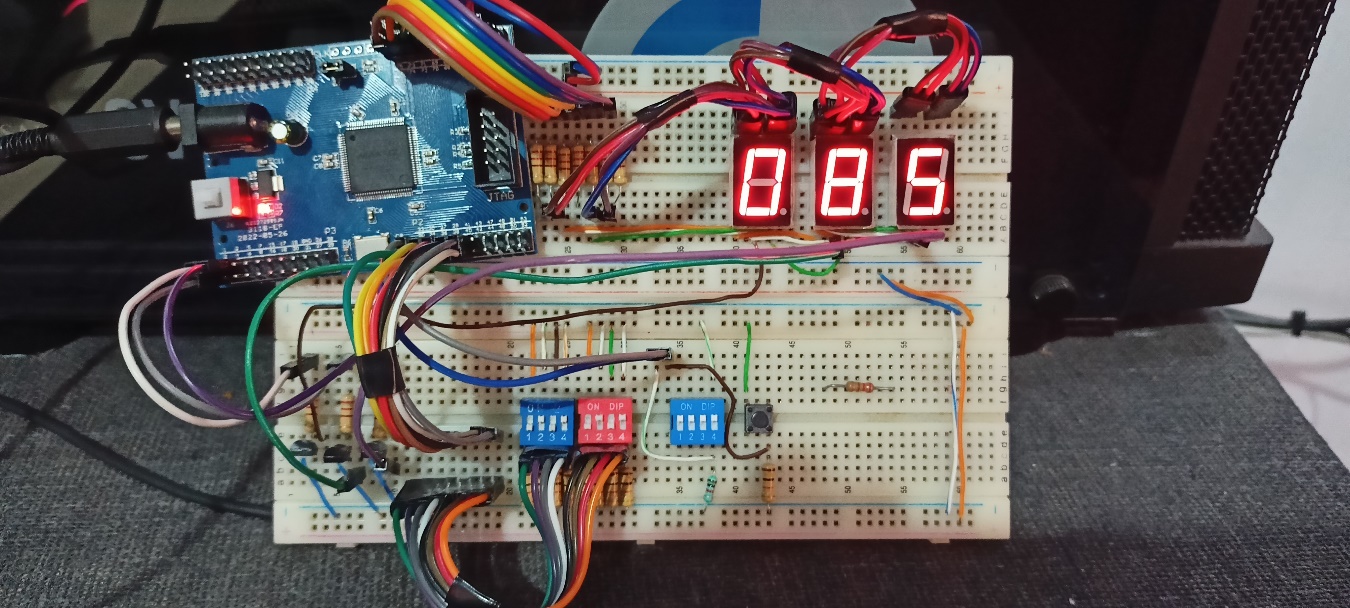


Fig. 14 Binario: 1010101

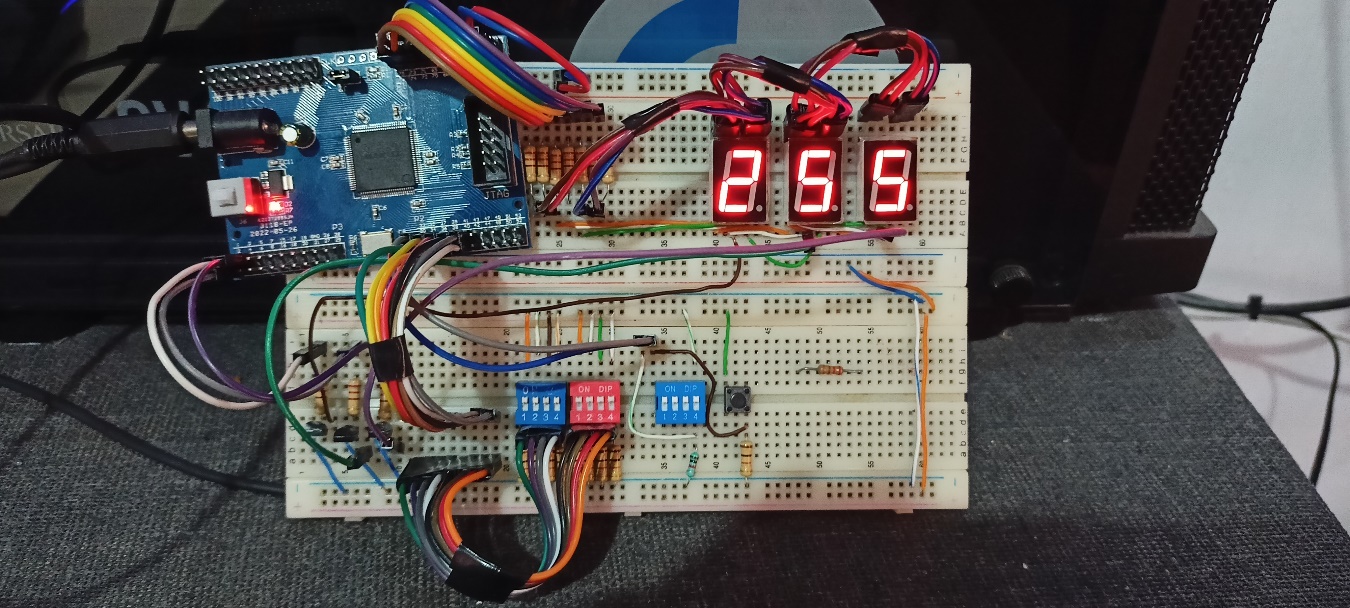


Fig. 15 Binario: 11111111