



Instituto Tecnológico de Buenos Aires

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

22.13 - ELECTRÓNICA III

Trabajo Práctico N° 2

Implementación de calculadora en FPGA

Grupo N° 3

AGRIPPINO, Franco
fagrippino@itba.edu.ar
61124

ALVAREZ, Matías Ezequiel
matalvarez@itba.edu.ar
62275

HEIR, Alejandro Nahuel
aheir@itba.edu.ar
62496

SBRUZZI, Juan Francisco
jsbruzzi@itba.edu.ar
62517

Cátedra

Ing. DE VERA, Pablo
Ing. WUNDES, Pablo Enrique

Q2 - 2022

Resumen

Se realizó la implementación de una calculadora en Verilog sobre la placa de desarrollo UPduino v3.0, basada en la FPGA iCE40UP5K de Lattice Semiconductor. Se utilizó la placa de teclado matricial del TP1 como método de entrada al sistema, aprovechando su capacidad de operar a 3.3 V. Se destaca el uso de un módulo de 8 dígitos 7 segmentos basado en el integrado MAX7219, permitiendo el control serializado de cada dígito a través de un bus SPI. La implementación también implicó la creación de una nueva placa a modo de interfaz de todas las partes. En el presente informe, se describen las consideraciones de diseño tenidas en cuenta y el desarrollo de los módulos de software y hardware involucrados, así como las características adicionales llevadas a cabo.

Índice

1. Características de la implementación	1
2. Diseño y desarrollo en Verilog	2
2.1. Modularización	2
2.2. Top level, <code>calc.v</code>	2
2.2.1. Consideraciones de clock	2
2.3. Máquina de estados, <code>fsm.v</code>	3
2.4. Teclado, <code>keyboard.v</code>	3
2.5. Unidad aritmético-lógica, <code>alu.v</code>	3
2.6. Display, <code>display.v</code>	4
2.6.1. Interfaz SPI, <code>Max7219.v</code> y <code>spi_master.v</code>	4
3. Diseño y desarrollo electrónico	5
3.1. Distribución	5
3.2. Placa de teclado	6
3.3. Placa de interfaz	6
3.4. Módulo display SPI, MAX7219	8
Apéndice A. Esquemático y PCB, KiCad	10
Bibliografía	14

Índice de figuras

2.1. Diagrama general de módulos.	2
2.2. Funciones de la calculadora.	3
2.3. Diagrama módulo display.	4
3.1. Diagrama de distribución de hardware.	5
3.2. Interconexión de todas las partes, fotografía.	5
3.3. Placa de teclado, fotografía.	6
3.4. Placa de interfaz alimentada, con referencias.	7
3.5. Placa de interfaz, fotografías.	7
3.6. Placa de interfaz, decimal point y control de brillo.	8
3.7. Esquemático display 8 dígitos, MAX7219.	8
3.8. Pinout display 8 dígitos, MAX7219.	9
3.9. Formato de palabra MAX7219.	9
3.10. Registros MAX7219.	9
A.1. PCB 3D, vista superior.	10
A.2. PCB 3D, vista inferior.	10

1. Características de la implementación

Display modular: a través de la comunicación SPI con un MAX7219 se permite el control encadenable de displays, permitiendo fácilmente extender el funcionamiento a números de 8 dígitos.

Brillo variable: mediante una combinación de teclas el usuario puede ajustar el nivel de brillo de los displays.

Punto decimal: se implementa un punto decimal fijo, estando el sistema preparado para operaciones con punto flotante pero siendo limitado por los recursos disponibles.

Operaciones matemáticas: suma, resta, multiplicación, división, potencia (desactivada), tanto de números enteros como reales (funcional bajo ciertas condiciones, actualmente) y positivos o negativos.

Concatenación: el resultado de la cuenta se carga en el primer operando para la siguiente, además de que presionar una operación cuando otra se encuentra pendiente permite encadenar resultados.

Parametrización: módulos clave se encuentran parametrizados para alterar la cantidad de dígitos de la calculadora.

2. Diseño y desarrollo en Verilog

2.1. Modularización

El funcionamiento del sistema fue dividido en 4 módulos característicos, identificados en la figura 2.1, en donde una máquina de estados principal lleva a cabo la lógica central de operación, haciendo de interfaz entre los demás módulos. Esta división permitió el desarrollo coordinado e independiente de cada módulo. En el [repositorio del trabajo](#) en GitHub, pueden encontrarse los códigos de Verilog correspondientes a cada módulo.

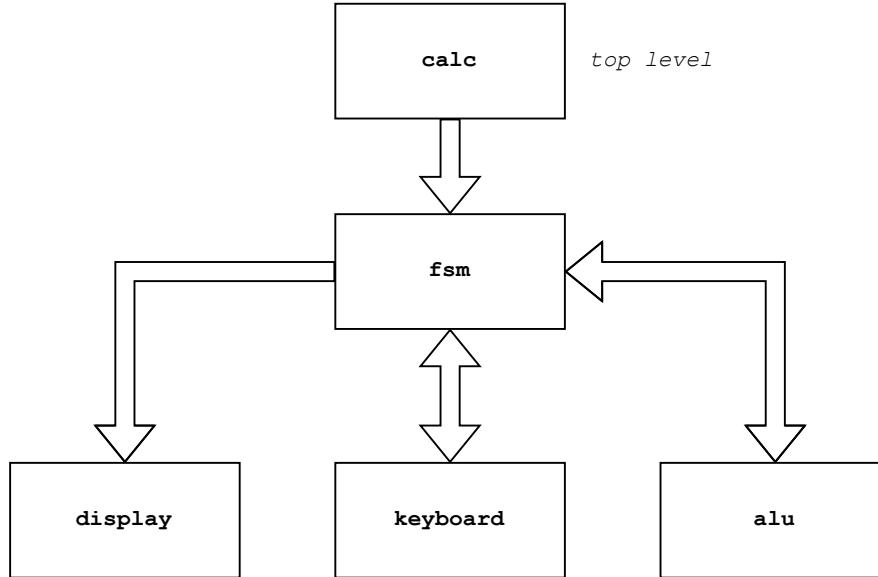


Figura 2.1: Diagrama general de módulos.

2.2. Top level, calc.v

Este es el módulo más superficial del sistema, el cual se encarga de inicializar los módulos subsiguientes y de establecer las referencias a los puertos I/O físicos de la FPGA. Además, inicializa los clocks internos, los cuales son proveídos a cada módulo según corresponda.

2.2.1. Consideraciones de clock

En las primeras instancias de implementación, se utilizaba únicamente el clock interno de alta velocidad, el cual podía ser preescalado hasta 6 MHz. Pronto, se halló que operar el teclado matricial junto a su placa a tan alta frecuencia no era factible, debido principalmente a que los tiempos de propagación máximos de la placa son comparables a dicha frecuencia (órdenes de 200 ns), causado por la lógica discreta utilizada.

Por estos motivos, se decidió utilizar el clock de baja frecuencia, 10 kHz, para todos los módulos, mejorando notablemente la respuesta del teclado, pero, ahora, ralentizando la comunicación con el display, el cual debe trabajar, nominalmente, a 10 MHz.

La decisión final fue implementar ambos clocks, estando el de baja frecuencia dirigido al módulo de teclado, y, el de alta frecuencia, al resto de módulos, dando resultados de funcionamiento satisfactorios.

2.3. Máquina de estados, fsm.v

La máquina de estados central explícita resulta relativamente simple al diseñarse a partir de un diagrama de estados triviales:

`LOADING_OP0` para la carga del primer operando.

`LOADING_OP1` para la carga del segundo operando.

`ALT_INPUT_OP0` para utilizar funciones alternativas del teclado al cargar el primer operando.

`ALT_INPUT_OP1` para utilizar funciones alternativas del teclado al cargar el segundo operando.

En el modo alternativo del teclado, el sistema simplemente ejecuta acciones especiales como guardar un operador especial, modificar el brillo del display, o limpiar los registros de la calculadora. Para la carga en si de los operandos, se implementa otra máquina de estados más implícita, donde cada vez que se presiona un número se desplazan por un dígito BCD a la izquierda al operando actual, además de desplazar el punto decimal si hubiese sido ingresado. No se cambia de estado de carga hasta que se presiona un operador o el botón de "igual". Al realizar cualquiera de esas dos acciones, se calcula el resultado de la operación actual, se lo guarda en el primer operando, y se permite su visualización o modificación. De esta manera, el usuario puede entrar con una secuencia como $2 + 4 + 5 + 6 + 9 + 15 + 8 =$ y obtener el resultado final exitosamente.

La máquina de estados principal se encuentra totalmente preparada para el manejo de números decimales, siendo el límite práctico de la ALU el inconveniente en la implementación. Por otro lado, es parte de las características de esta máquina de estados el permitir la carga de números negativos, incrementando el alcance del trabajo.

1	2	3	+
4	5	6	-
7	8	9	DEL
<i>ALT</i>	0	DP	=

(a) Modo normal.

15.6%	28.1%	40.6%	x
53.1%	65.6%	78.1%	/
90.6%	NA	NA	AC
<i>back</i>	3.1%	NA	NA

(b) Modo alternativo.

Figura 2.2: Funciones de la calculadora.

2.4. Teclado, keyboard.v

El módulo del teclado se debe entender siempre con la interfaz del TP1 en mente. En síntesis, el rol de este módulo se plantea desde el querer una indicación de la tecla presionada, y si es válida esa medición. Para lograr esto, este módulo recorre las cuatro columnas del teclado, verifica si hay teclas presionadas, y almacena el dato (persistiendo a lo largo de toda una rotación de columnas) para evitar glitches.

2.5. Unidad aritmético-lógica, alu.v

La ALU final se preparó para aprovechar las funciones matemática provistas por la FPGA. Se realizó una implementación inicial manejando las operaciones directamente en formato BCD

pero por diversos inconvenientes a la hora de generalizar su funcionamiento se debió dejar de lado. Debido a esto, finalmente la ALU convierte las entradas BCD a binario tomando también la información de signo y lugar del punto decimal.

Las operaciones implementadas resultan ser: suma, resta, multiplicación, división, y potencia (desactivada por limitaciones físicas). Las operaciones con números reales se debieron limitar a punto fijo debido a los recursos disponibles.

2.6. Display, display.v

El módulo de display se encarga íntegramente del control de los dígitos 7-segmentos, recibiendo constantemente de la FSM qué mostrar, dónde, y la configuración de brillo seleccionada por el usuario. Internamente, trabaja con una máquina de estados simple, que inicializa el módulo de display físico y aguarda cambios de parte de la FSM para actualizarlo acordemente.

2.6.1. Interfaz SPI, Max7219.v y spi_master.v

Al trabajar con el driver MAX7219 para el manejo de los dígitos (ver sección 3.4 para detalles), que permite serializar el control por una interfaz SPI, fue necesario lograr una implementación de SPI en la propia FPGA. Dado que dicho tema, estrictamente, escapa de la materia y no resultaba trivial desarrollarlo desde cero, se optó por utilizar la implementación de un master SPI y del MAX7219 en Verilog del usuario de GitHub [cerkit \[1\]](#).

Si bien estos módulos, `Max7219.v` y `spi_master.v`, simplificaron el «desafío», para inicializar correctamente el display y lograr mostrar lo pretendido, fue necesario trabajar constantemente con la documentación del driver [2], donde se especifican los registros internos que deben configurarse para el encendido del módulo, habilitación y modo de control de los dígitos, ajuste de brillo, entre otros aspectos. En la figura 2.3 se sintetiza la topología del módulo display.

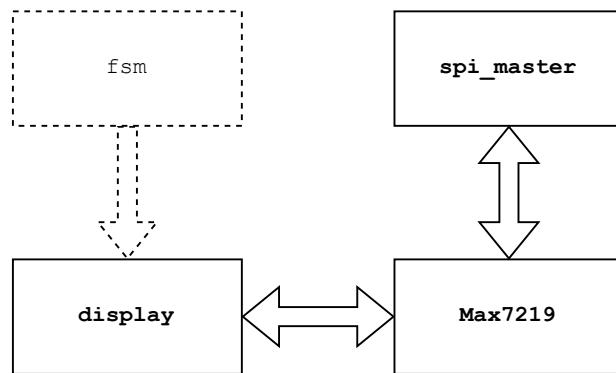


Figura 2.3: Diagrama módulo display.

3. Diseño y desarrollo electrónico

3.1. Distribución

Con el objetivo de utilizar la placa de teclado del TP1, y el módulo de dígitos 7 segmentos basado en driver MAX7219, el diseño electrónico de la calculadora estuvo direccionado por dichos factores, resultando necesaria una placa que haga de interfaz entre la ya existente, el display, y la correspondiente FPGA. La figura 3.1 resume la distribución de las partes, viéndose físicamente implementado en la figura 3.2. Se observa la inclusión de LEDs y botones adicionales, conectados a la FPGA; los mismos, permitieron, inicialmente, hacer debuggeos del sistema, además de proveer inputs y feedbacks directos.

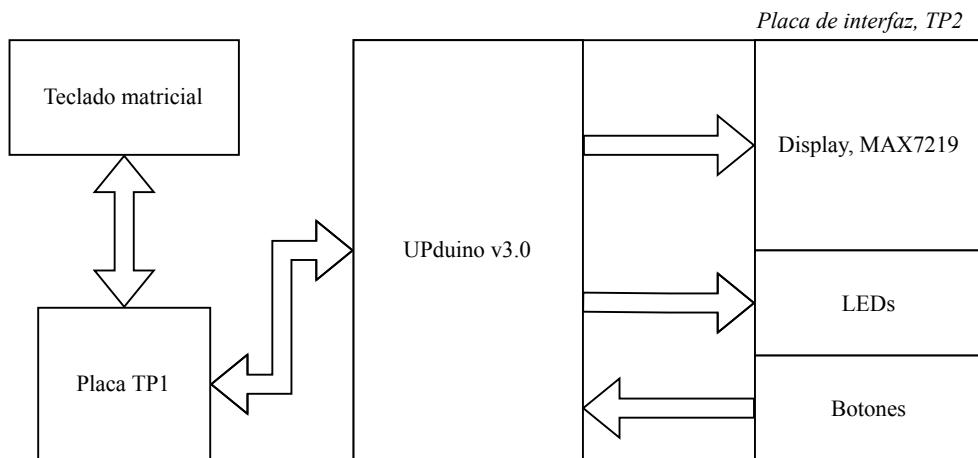


Figura 3.1: Diagrama de distribución de hardware.

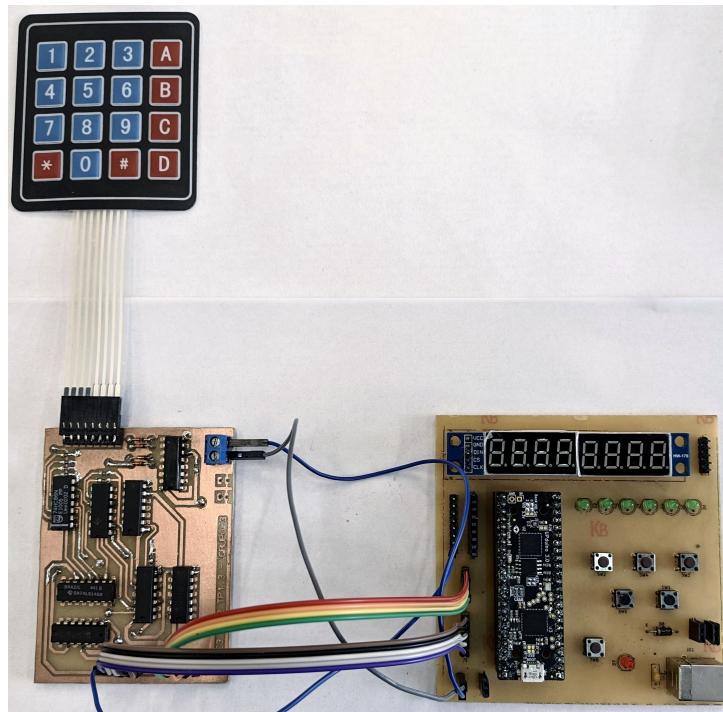


Figura 3.2: Interconexión de todas las partes, fotografía.

3.2. Placa de teclado

Un análisis más profundo de esta placa puede encontrarse en el informe del TP1. Como aspectos relevantes, se mencionan la capacidad de operación a 3.3 V - factor motivante para utilizarla junto a la FPGA sin adaptación de niveles - y tiempos de propagación máximos cercanos a 200 ns - factor contemplado en el uso de clocks, tema abarcado en la sección 2.2.1.

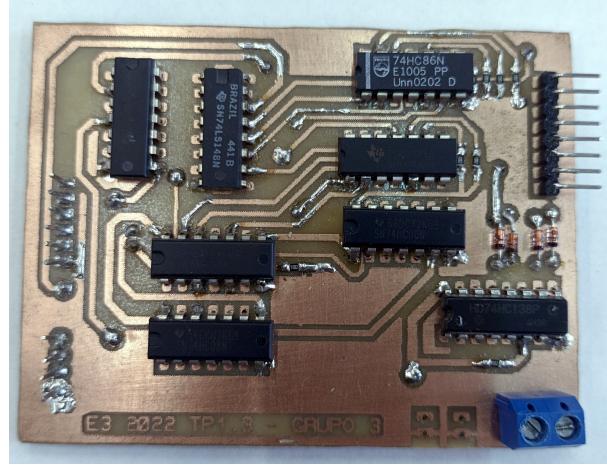


Figura 3.3: Placa de teclado, fotografía.

3.3. Placa de interfaz

Esta placa representa el principal desarrollo de hardware del trabajo. Todo lo referido a archivos de diseño (esquemáticos, PCB) se concentró en el apéndice A, además de disponer del proyecto en KiCad en el [repositorio del trabajo](#) en GitHub.

Como se vio en el diagrama de la figura 3.1, la placa necesita, mínimamente, conexiones a la placa de teclado y al display (bus SPI). Adicionalmente, se tienen

1. Conexión directa al teclado matricial
2. Extensión para encadenar displays
3. Botón de reset (conectado al nRST de la FPGA) y botones de general purpose.
4. LEDs de general purpose, usando 3 de ellos para indicación de signo.
5. Entrada de 5 V por sockets con reverse protection o por USB tipo B; complementario a la alimentación por el micro USB de la UPduino.

Respecto a la alimentación, tanto al FPGA como la placa de display funcionan a 3.3 V, mientras que el MAX7219 y el display operan a 5 V. Si bien en la documentación del driver [2] se especifica un *logic high input voltage* $V_{IH,min} = 3.6$ V, se destaca el aparente correcto funcionamiento con una lógica de 3.3 V.

En la figura 3.4 se observa la placa de interfaz, detallando algunas de sus partes. En la figura 3.5b se muestra la vista inferior de la placa*, destacando el uso de capacitores y resistores SMD.

*Errata: en la leyenda de la placa de interfaz, figura 3.5b, se colocó *Grupo 1*, debiendo haber sido *Grupo 3*

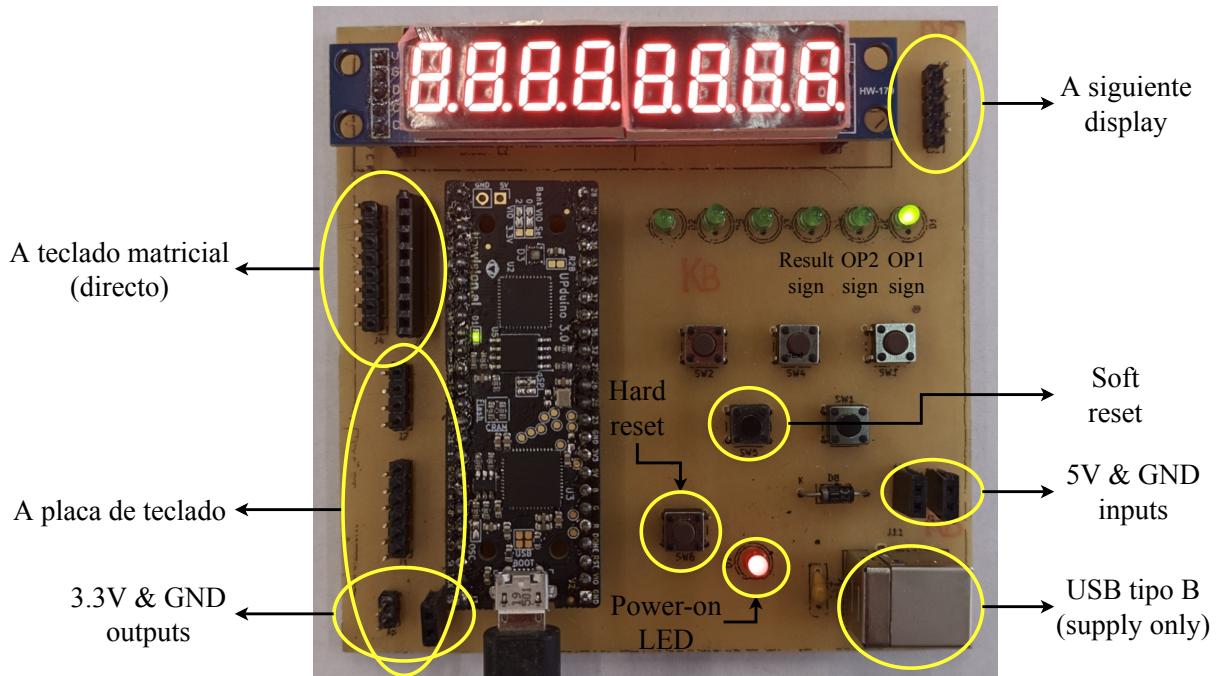
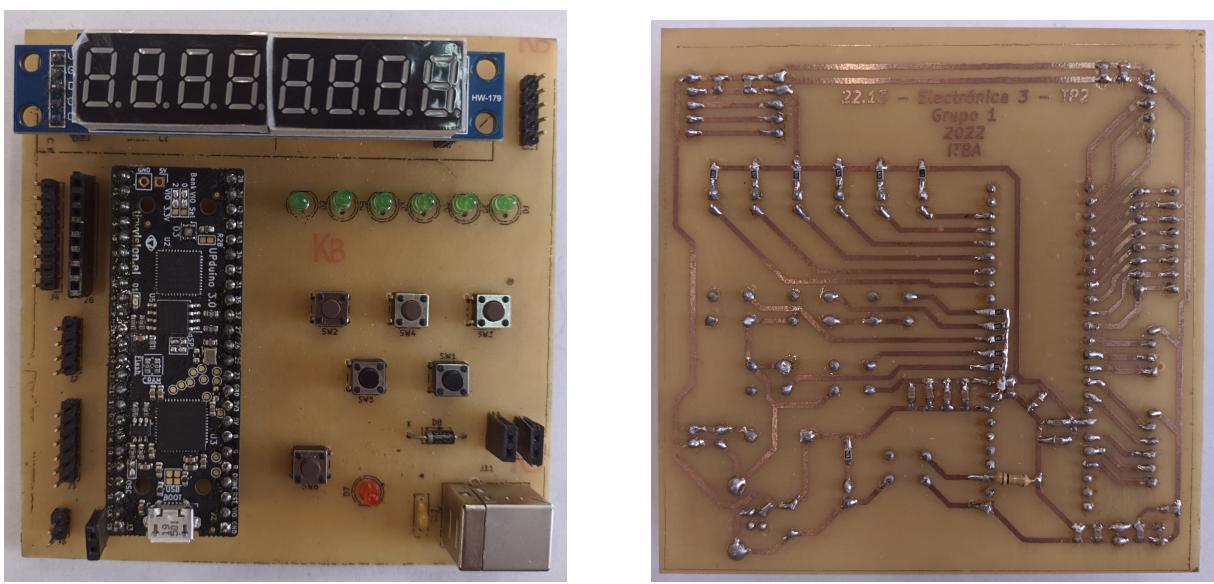


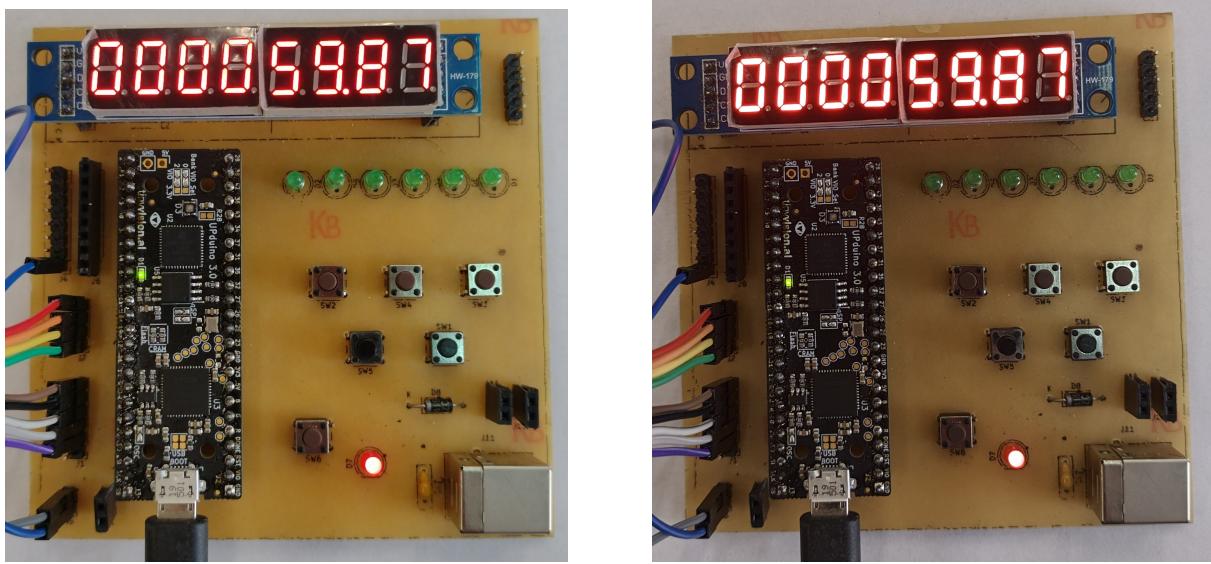
Figura 3.4: Placa de interfaz alimentada, con referencias.



(a) Vista superior.

(b) Vista inferior.

Figura 3.5: Placa de interfaz, fotografías



(a) Brillo bajo.

(b) Brillo máximo.

Figura 3.6: Placa de interfaz, decimal point y control de brillo.

3.4. Módulo display SPI, MAX7219

El módulo display utilizado está basado en el driver MAX7219, el cual permite controlar hasta 8 dígitos 7 segmentos, tanto a nivel lógico como a nivel potencia, a través de una interfaz SPI, en donde actúa como slave, siendo el master la FPGA. En la figura 3.8, se observa el módulo standalone, y, en la figura 3.7, un esquemático equivalente.

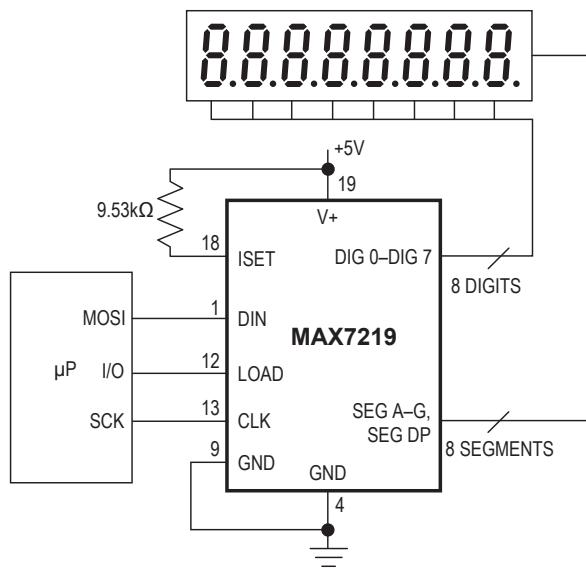


Figura 3.7: Esquemático display 8 dígitos, MAX7219.

La comunicación con este módulo se basa en el envío de paquetes-palabras de 16bits, que contienen la dirección de algún registro interno, y el valor que se le quiere escribir. Existe un total de 16 registros, teniendo cada uno una función específica. En la figura 3.9, se muestra el formato de los paquetes, y, en la figura 3.10, los registros disponibles. Toda esta especificación fue extraída de la documentación del fabricante [2], donde se abarca con más profundidad la funcionalidad de cada registro.



Figura 3.8: Pinout display 8 dígitos, MAX7219.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0		
X	X	X	X	ADDRESS						MSB	DATA						LSB

Figura 3.9: Formato de palabra MAX7219.

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xA
Scan Limit	X	1	0	1	1	0xB
Shutdown	X	1	1	0	0	0xC
Display Test	X	1	1	1	1	0xF

Figura 3.10: Registros MAX7219.

Apéndice A Esquemático y PCB, KiCad

En las siguientes páginas, se adjuntan el esquemático y PCB exportados directamente del software de diseño KiCad. Las imágenes del bottom copper y top silkscreen están al 100 % de escala y con la resolución adecuada para ser impresas.

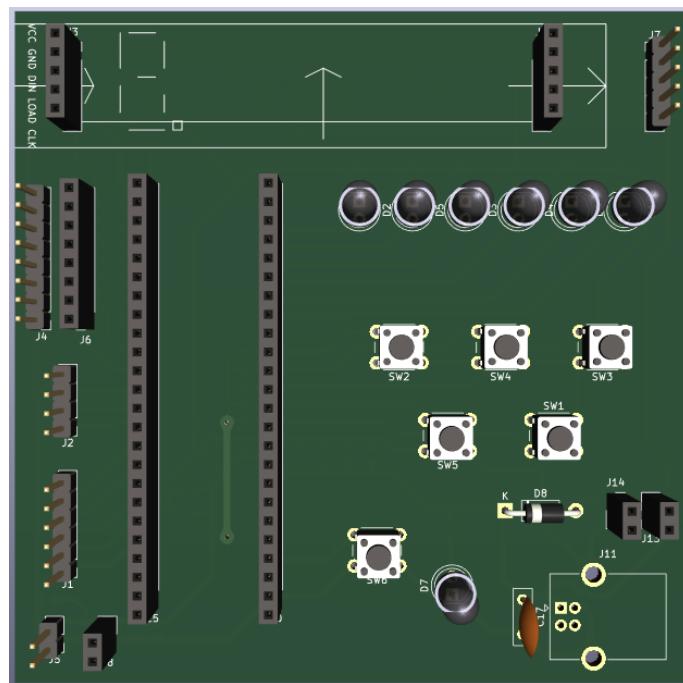


Figura A.1: PCB 3D, vista superior.

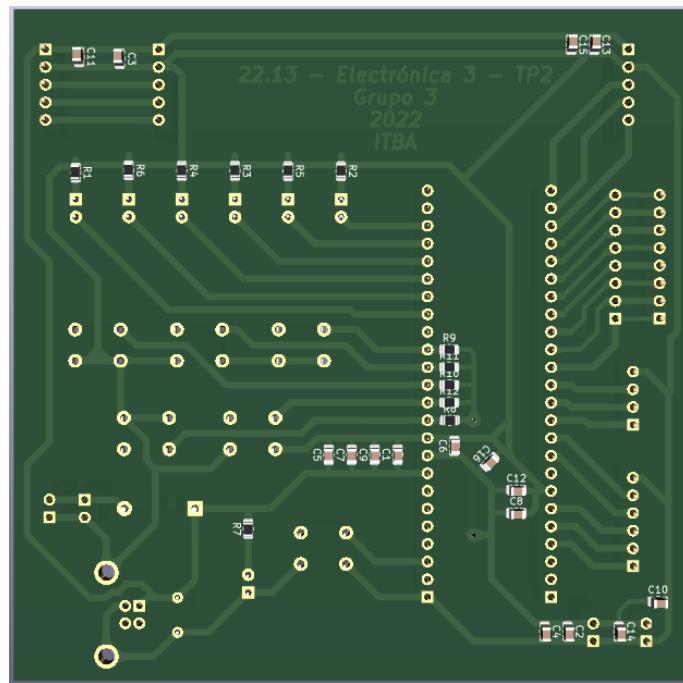
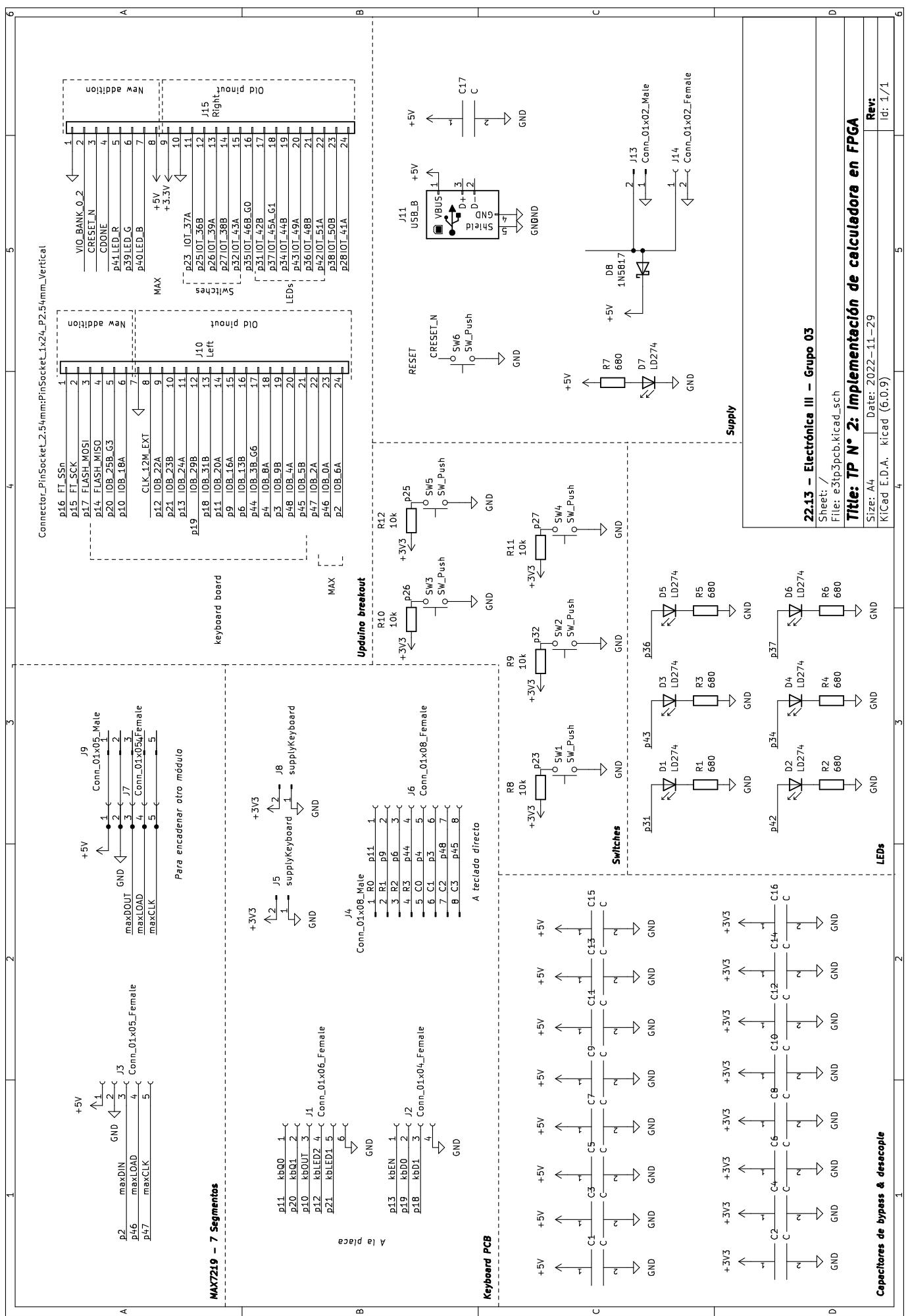
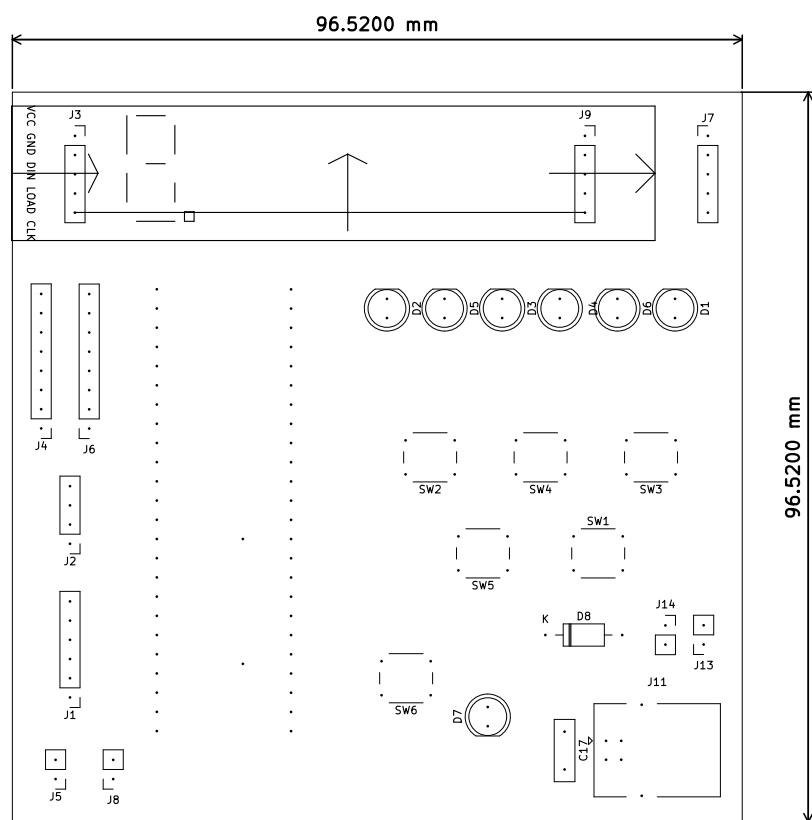
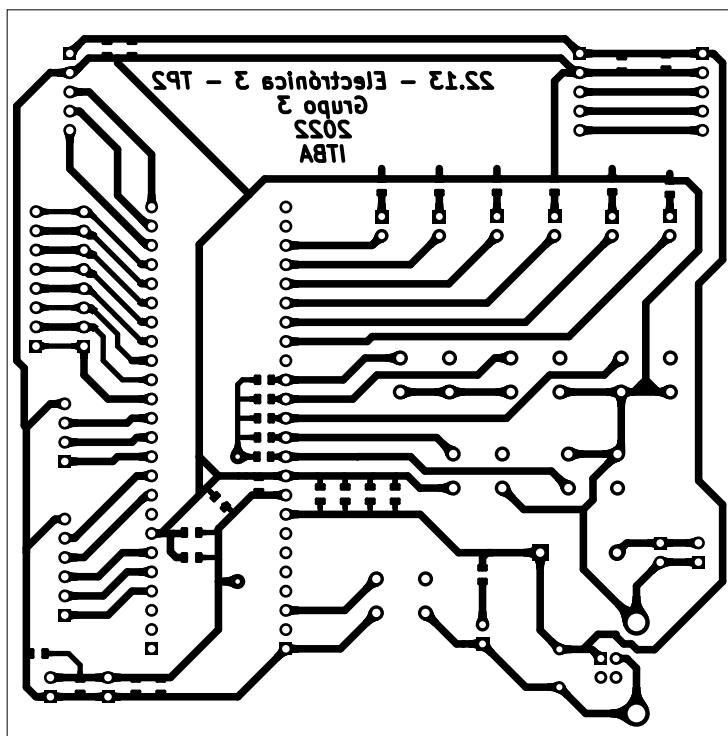


Figura A.2: PCB 3D, vista inferior.







Bibliografía

- [1] Cerkit, *max7219TinyFPGA*, <https://github.com/cerkit/max7219TinyFPGA>, 2019.
- [2] *Serially Interfaced, 8-Digit LED Display Drivers*, MAX7219/MAX7221, Rev. 6, Maxim Integrated, ago. de 2021.