

UNC

Universidad
Nacional
de Córdoba

Cátedra de Arquitectura de Computadoras

Trabajo Práctico N° II - UART

Integrantes:

López Gastón (gopezlaston@gmail.com)

Kleiner Matías (kleiner.matias@gmail.com)

Fecha de entrega:

26 de septiembre de 2018

Introducción

Para el Trabajo Práctico N° 2, se realizó la implementación en Verilog de un módulo UART. Dicho módulo se conecta a la ALU desarrollada en el TP N° 1. El sistema se completa con una PC, a partir de la cual se envían los operandos y el código de operación, para que la ALU los procese. El resultado es enviado de vuelta a dicha PC.

Para la implementación se usará una FPGA, la cual es un dispositivo programable que contiene bloques de lógica que, al interconectarlos, genera una funcionalidad. La mencionada interconexión y funcionalidad se configuran mediante un lenguaje de descripción de hardware (Verilog).

Esquema de la consigna

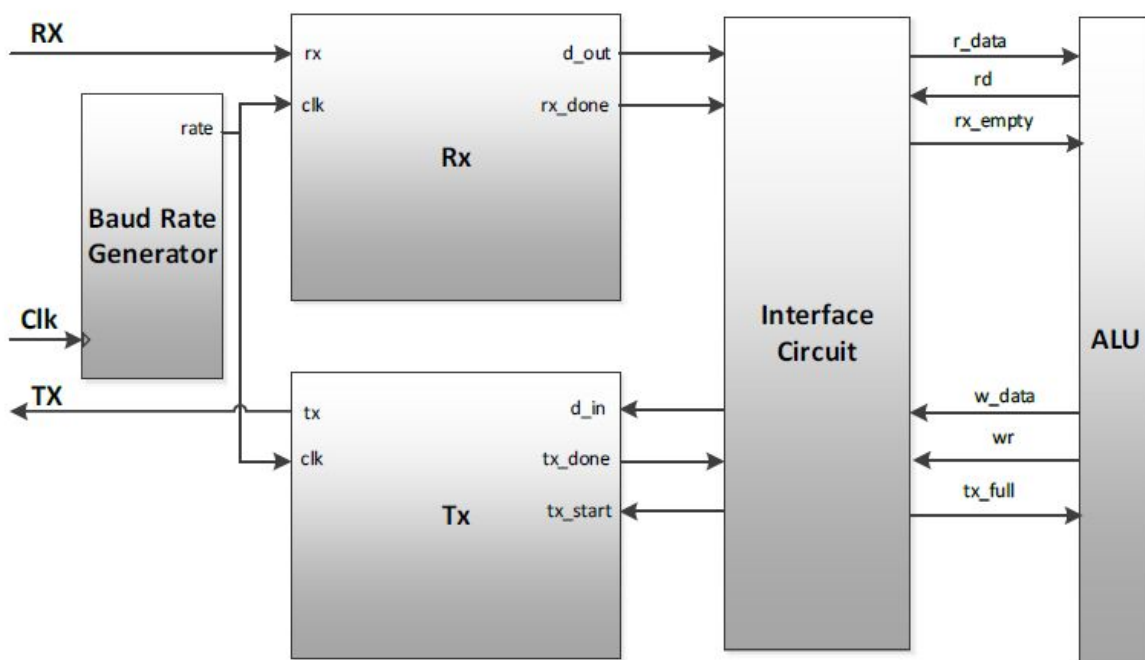


Figura 1 - TP N°2: UART.

Herramientas y elementos utilizados

Entre los dispositivos disponibles para la implementación de la UART se seleccionó la placa de desarrollo ARTY, que puede verse en la siguiente figura.

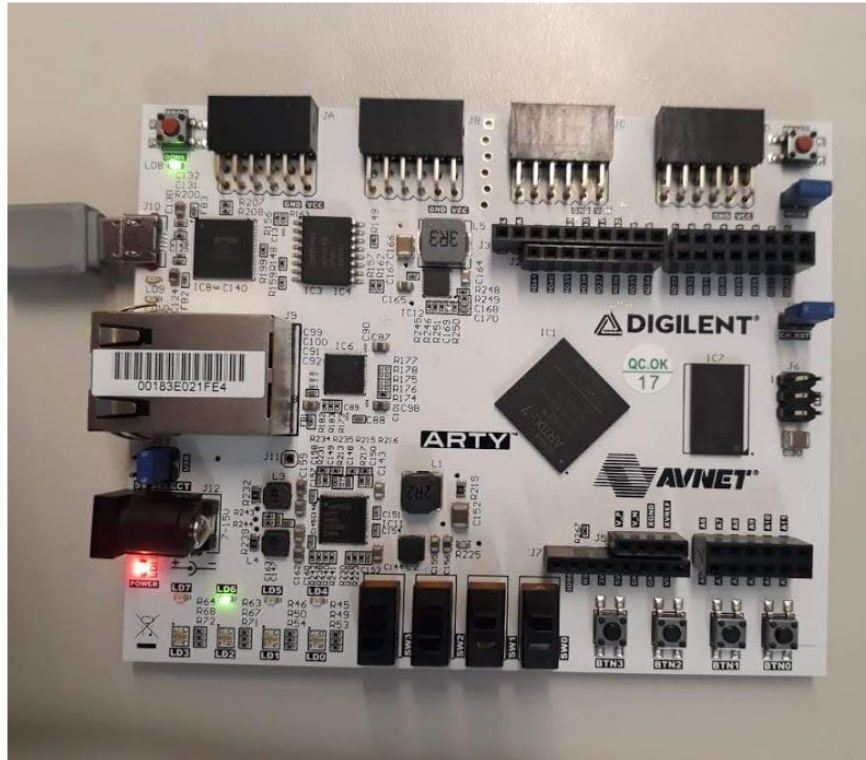


Figura 2 - FPGA Utilizada para el desarrollo del proyecto.

La principal razón de la selección fue que esta placa es una de las que soportan la interfaz de desarrollo Vivado, sobre la cual poseemos algo de experiencia previa. Cabe destacar que la FPGA que contiene la placa es una Artix-7 de Xilinx.

Diseño - Módulo TOP.

Se definió un módulo Top, que contiene la instanciación de los siguientes módulos:

- Tx.
- Rx.
- Baud Rate Generator.
- Alu.
- Interface Circuit.

Las entradas y salidas de este módulo son:

- input i_clock: clock.
- input i_reset: reset.
- input uart_txd_in: transmisor de PC.
- output uart_rxd_out: receptor de PC.
- output [7:0] jc: GPIO de placa para medición con osciloscopio.

El baud rate (parametrizable) utilizado en este trabajo práctico es de 9600 bps, con una frecuencia del clock de 100 MHz. La cantidad de bits de stop es 2 y la cantidad de bits de datos es 8, generando con el bit de start una trama con longitud igual a 11 bits.

Diseño - Módulo Baud Rate Generator.

Este módulo genera ticks que alimenta a los módulos Tx y Rx para coordinar las señales recibidas o transmitidas. La generación de ticks es 16 veces por baud rate. Consiste en un contador, que cuando alcanza un límite, genera un tick. Para calcular dicho límite la fórmula es la siguiente:

$$L\acute{i}mite = \frac{Clock}{Baud\ Rate * 16}$$

Diseño - Módulo RX.

Consiste en una máquina de estados. La representación de estados utilizada es la one-hot. Observando la Figura 3, se tiene que para pasar del estado de Espera al estado de Start se requiere la llegada del bit de start (bit en nivel bajo). Cuando se cuentan 8 ticks se está en la mitad de dicho bit de start que se recibió, por lo que se cambia al estado Read. En este estado, se cuentan los 8 bits de datos (cada 16 ticks se toma el valor recibido debido a que se está en el medio del bit transmitido desde la PC). Luego de contar esos 8 bits (valor parametrizable) se pasa al estado de Stop. Allí se espera una cantidad de bits de stop (en nivel alto) igual al valor definido en la parametrización del módulo. Cuando llegan dichos bits de stop, se coloca el bit Rx Done en alto y en la salida del módulo Rx también se inserta el dato recibido (los 8 bits de dato). Luego se pasa al estado de Espera. En caso de que se necesiten 2 bits de stop, si el primero que llega es un 1 y el segundo un cero, se desincroniza todo el sistema debido a que faltan 8 ticks para pasar el segundo bit de stop. Esta tarea de dejar pasar dicha cantidad de ticks se realiza en el estado de Error.

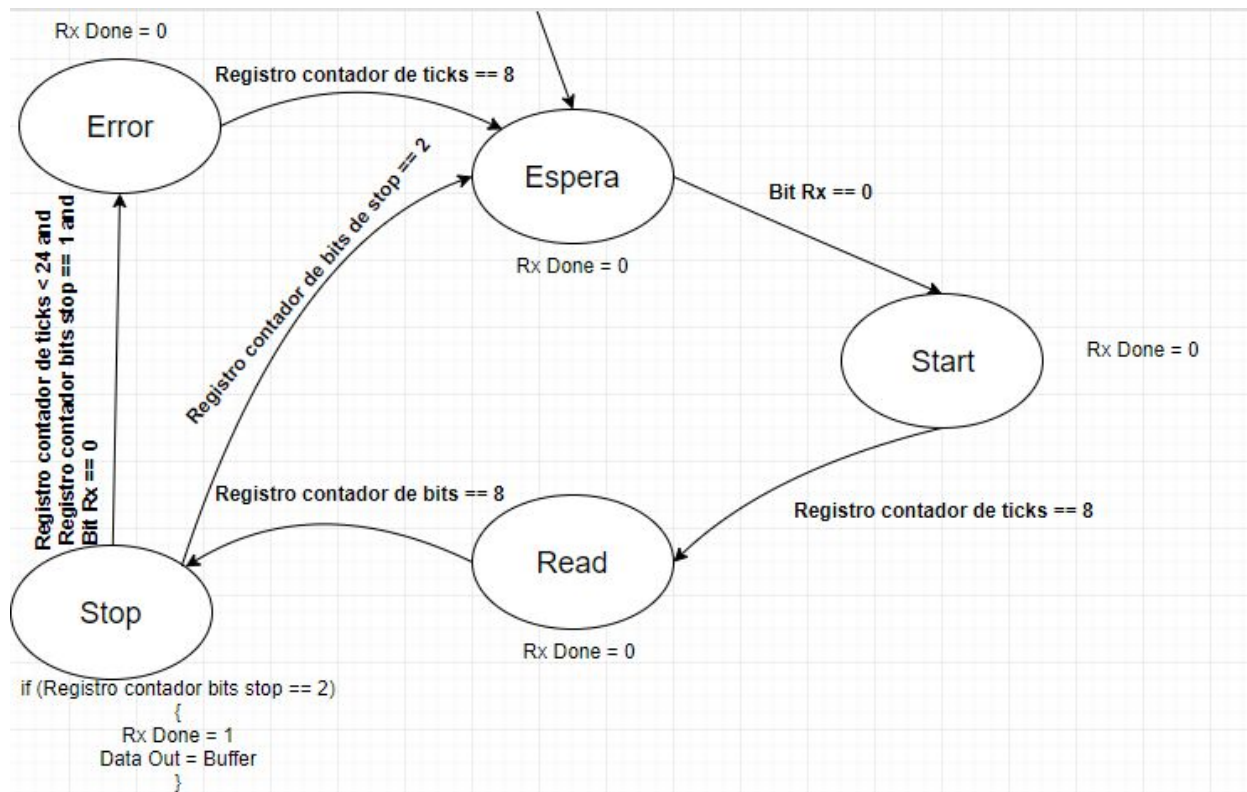


Figura 3 - Máquina de Estados del módulo Rx.

Diseño - Módulo TX.

Consiste en una máquina de estados similar a la anterior. La representación de estados utilizada es la one-hot. Observando la Figura 4, se tiene que para pasar del estado de Espera al estado de Start se requiere un nivel en alto de la señal Tx Start. Luego el pasaje entre los demás estados se debe a la finalización en la generación de los distintos bits de la trama. Mientras se están transmitiendo dichos bits la salida Tx Done se encuentra en un nivel bajo para que el módulo Interface Circuit no envíe otro dato a transmitir. En los estados Read y Stop las salidas están sujetas a condiciones para prevenir errores, colocándose en el diagrama anterior aquellas más representativas.

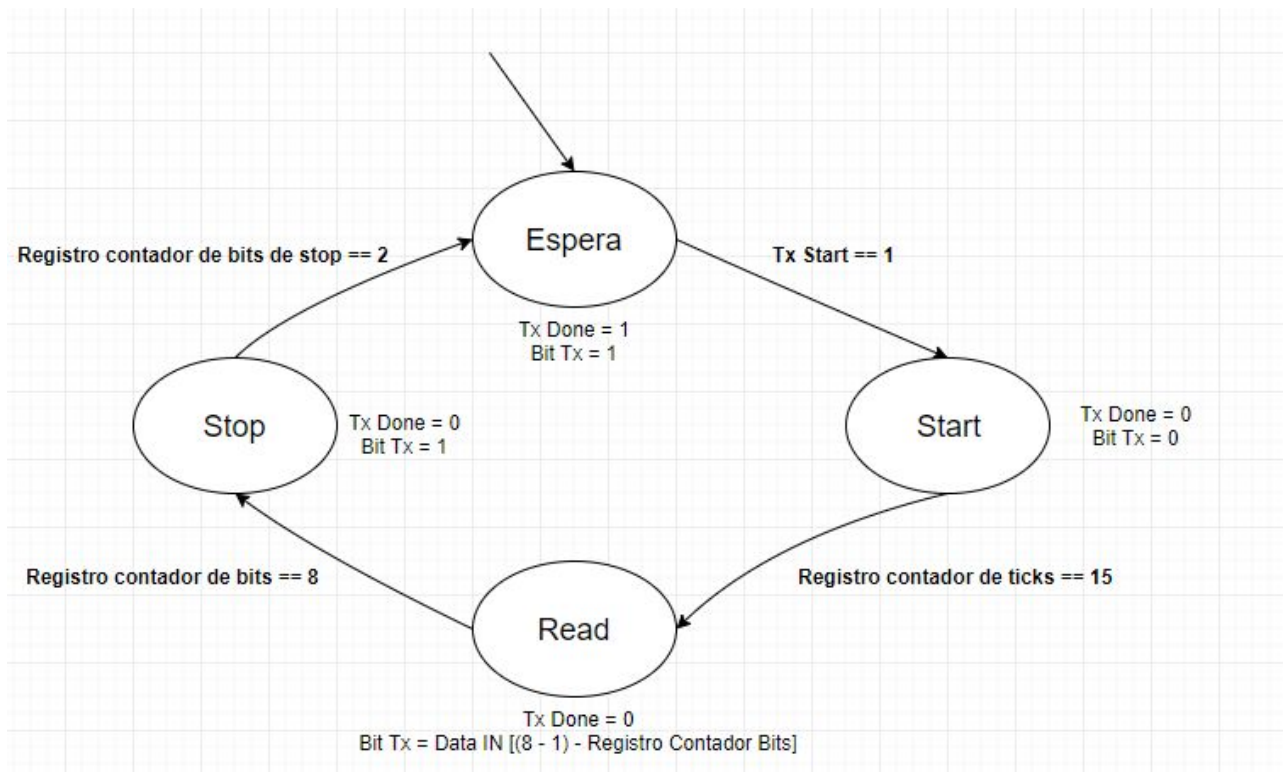


Figura 4 - Máquina de Estados del módulo Tx.

Diseño - Módulo Interface Circuit.

Consiste también en una máquina de estados que se muestra en la Figura 5. Permite la comunicación entre la Alu y los módulos Tx y Rx. Dicha máquina de estados utiliza los flancos ascendentes de las señales Rx Done (salida del módulo Rx que indica que se recibió en forma completa un dato) y Tx Done (señal de salida del módulo Tx que indica que se completó la transmisión de un dato) para efectuar los cambios de estados, los cuales se representan mediante lógica one-hot. Además, la señal Tx Start se pone en uno únicamente cuando se carga el operando 2 y se tiene el resultado de la Alu. Dicha señal inicia la transmisión en el módulo Tx. En los estados Operando1, Operación y Operando2 se carga lo que sale del módulo Rx en el registro correspondiente.

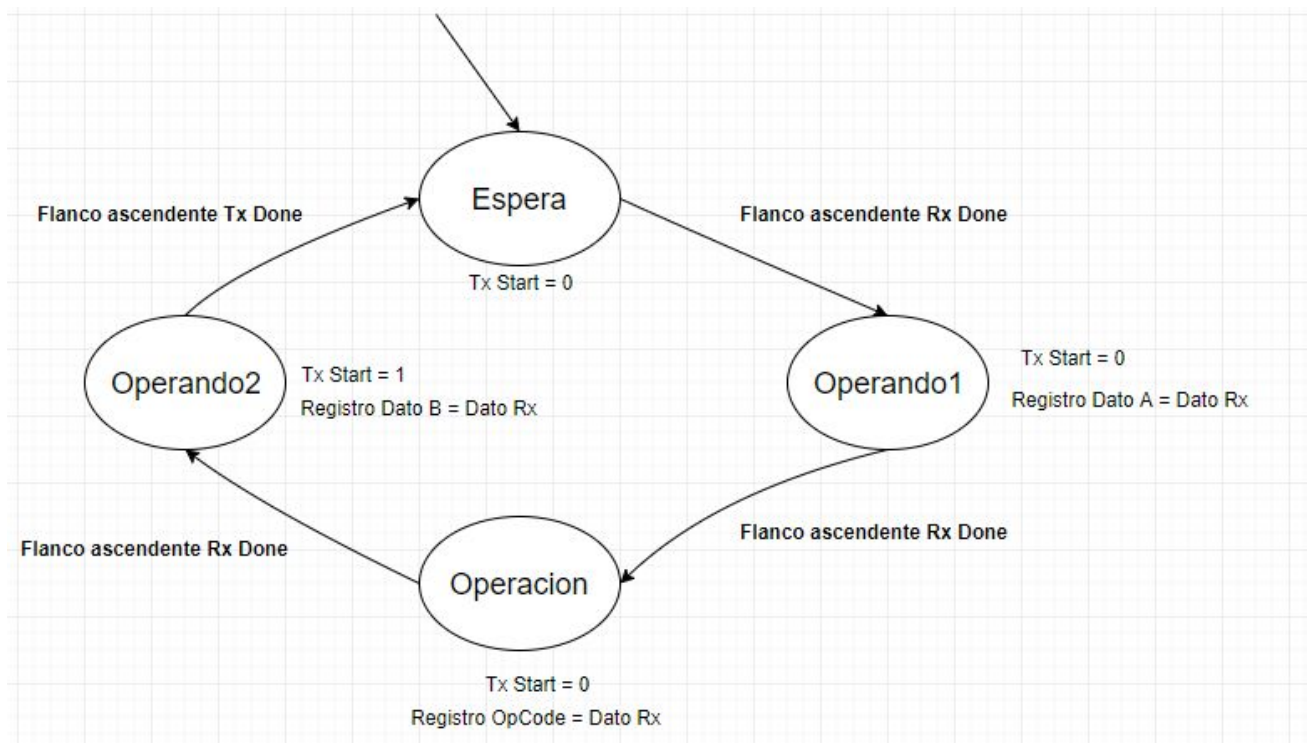


Figura 5 - Máquina de Estados del módulo Interface Circuit.

Diseño - Módulo Alu.

Ídem al del TP N° 1.

Simulaciones de los módulos.

Se efectuaron test benches para todos los módulos y para el top.

- **Top:** se comprueba el reset y que el sistema devuelva el resultado correcto cuando se ingresan operandos y códigos de operación determinados.
- **Alu:** ídem a TP N° 1.
- **Baud Rate Generator:** se comprueba la generación correcta de los ticks.

- **Rx:** se envían bits y se comprueba el pasaje entre los estados del módulo y las salidas en cada uno de dichos estados. Se envían tramas correctas e incorrectas.
- **Tx:** se ingresa un valor a transmitir y se eleva en alto el bit de Tx Start. Se comprueba el pasaje entre los estados del módulo y las salidas en cada uno de dichos estados.
- **Interface Circuit:** se comprueba el pasaje entre los estados del módulo y las salidas en cada uno de dichos estados. Se comprueba que se estén cargando correctamente los valores en los distintos registros que utilizará la Alu como input.

Implementación en FPGA.

Para la implementación, se instanciaron los módulos Tx, Rx, Baud Rate Generator, Interface Circuit y Alu dentro del módulo top, sobre el cual se declararon entradas y salidas que se mapean con pines de la FPGA a utilizar en un archivo denominado constraints. El mismo consiste en un archivo con extensión .xdc.

Explicación de warnings.

En el desarrollo del práctico, apareció el siguiente warning al generar el bitstream que se debe cargar en la FPGA correspondiente:

“[DRC CFGBVS-1] Missing CFGBVS and CONFIG_VOLTAGE Design Properties: Neither the CFGBVS nor CONFIG_VOLTAGE voltage property is set in the current_design. Configuration bank voltage select (CFGBVS) must be set to VCCO or GND, and CONFIG_VOLTAGE must be set to the correct configuration voltage, in order to determine the I/O voltage support for the pins in bank 0. It is suggested to specify these either using the 'Edit Device Properties' function in the GUI or directly in the XDC file using the following syntax:

*set_property CFGBVS value1 [current_design]
#where value1 is either VCCO or GND*

*set_property CONFIG_VOLTAGE value2 [current_design]
#where value2 is the voltage provided to configuration bank 0*

Refer to the device configuration user guide for more information.”

Se buscó información sobre este warning, dando como resultado un conocimiento acerca del origen del mismo. Dicho warning ocurre porque se recomienda especificar el nivel de tensión de los pines I/O del bank 0 de la FPGA en el constraint (ver links al final de esta sección). Dicho nivel de tensión por defecto es de 3.3 V. Además cabe destacar que para poder asociar la placa ARTY con el proyecto en Vivado se debió descargar un paquete de software aparte. Un nivel de tensión erróneo puede causar daños en la placa. Teniendo en cuenta lo mencionado, se agregaron las líneas correspondientes en el constraints, lo que arrojó warnings debido a una incompatibilidad de dichas líneas con el paquete de software asociado a la ARTY, que ya posee los niveles de tensiones por defecto en caso de que no se especifique nada. Es por esto que no se ha corregido dicho warning debido a que no afecta el funcionamiento del proyecto.

Links consultados:

- <https://reference.digilentinc.com/reference/programmable-logic/arty/reference-manual?redirect=1>
- http://www.iamelectronic.com/products/T0005_Artix-7_FPGA_Module/datasheet/

Otro warning que apareció está relacionado a la síntesis del módulo Baud Rate Generator. El mismo mostraba el siguiente mensaje:

“[Synth 8-6014] Unused sequential element reg_contador_reg was removed.”

Si se observa el esquemático RTL, dicho registro aparece sintetizado, por lo que decidimos consultar sobre este error, encontrando varios foros en donde se discutía el mismo. Se concluyó que se trata de un bug que posee la versión 2017.1 de Vivado, la cual es la que estamos utilizando para este proyecto. Es por esto que dicho warning se ignoró.

Link consultado:

<https://forums.xilinx.com/t5/Synthesis/Unused-sequential-element-was-removed-no-idea-why/td-p/801128>

Mediciones con el osciloscopio.

Se observó con el osciloscopio la salida del transmisor instanciado en la FPGA cuya salida utiliza un pin del GPIO (JC). Para observar la señal en el osciloscopio, se optó por generar una transmisión continua de datos. Al tener configurada la ALU de tal forma que un código de operación erróneo provoca que el resultado sea el primer operando, entonces se pasó a utilizar el software PuTTY para enviar los datos desde la PC hacia la placa. Cabe destacar que se envió la letra L continuamente (opcode erróneo), tal como se muestra en la Figura 6. Dicha letra L representa la trama de bits 01001100.

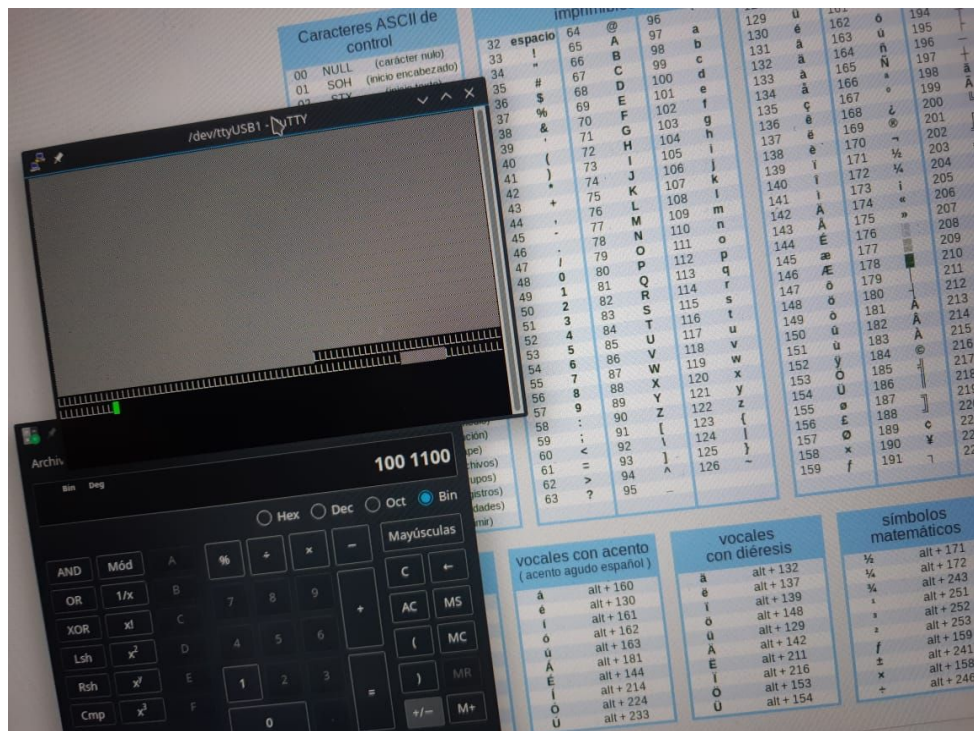


Figura 6 - Envío continuo de la letra L vía PuTTY.

En el osciloscopio se obtuvo la trama de bits correspondientes a la salida del módulo TX instanciado en la FPGA. Cabe destacar que se utilizó una punta atenuadora colocada en un pin del JC de la placa. En la Figura 7 pueden observarse el bit de start, los bits de datos (arrancando por el bit LSB del valor ASCII de la letra "L", la cual actúa de primer operando, segundo operando y opcode) y los 2 bits de stop.

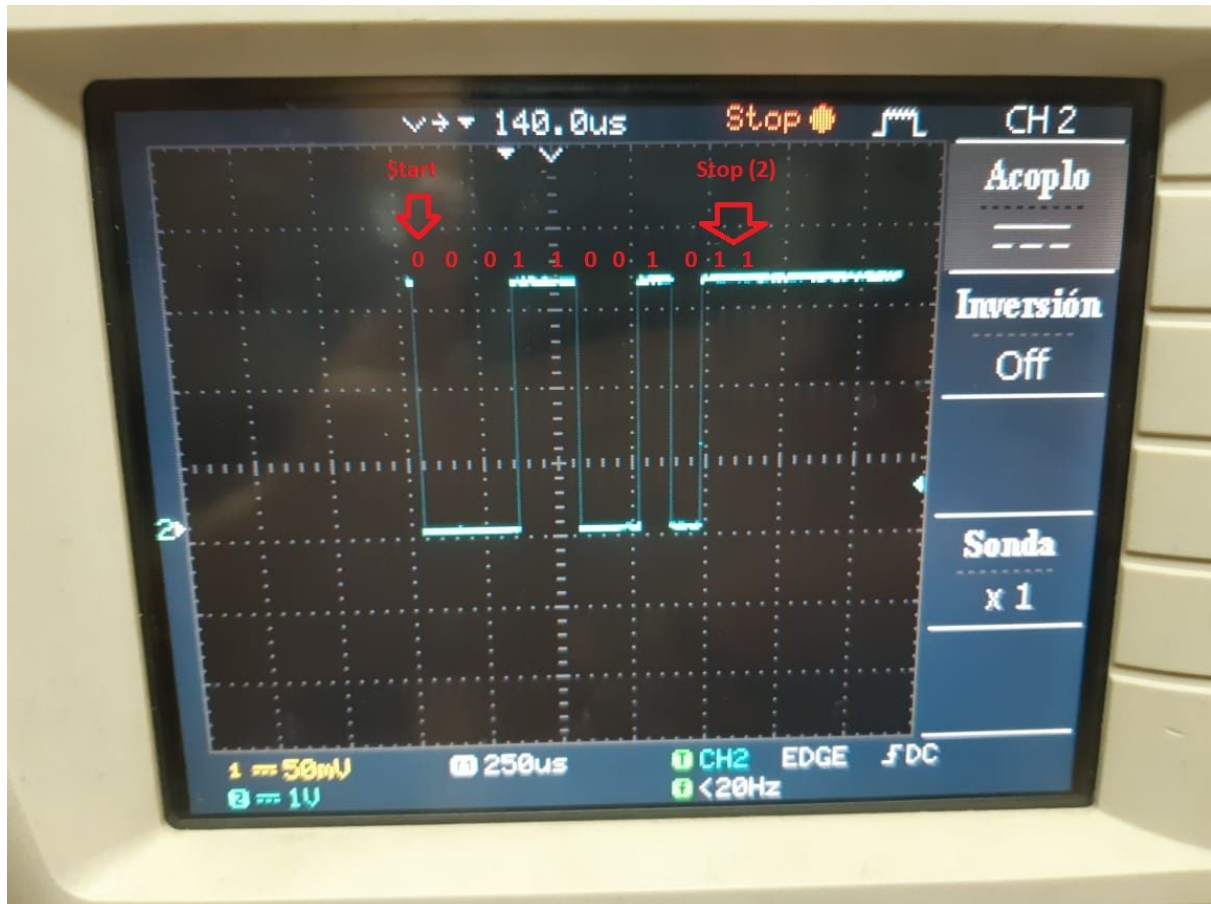


Figura 7 - Trama transmitida por el módulo TX instanciado en la FPGA.

Interfaz en la PC.

Se generó una GUI en Python con la librería Tkinter que permite enviar los operandos y la operación para que la ALU instanciada en la FPGA los procese. Cuando la ALU genera el resultado, dicha GUI lo muestra en pantalla.

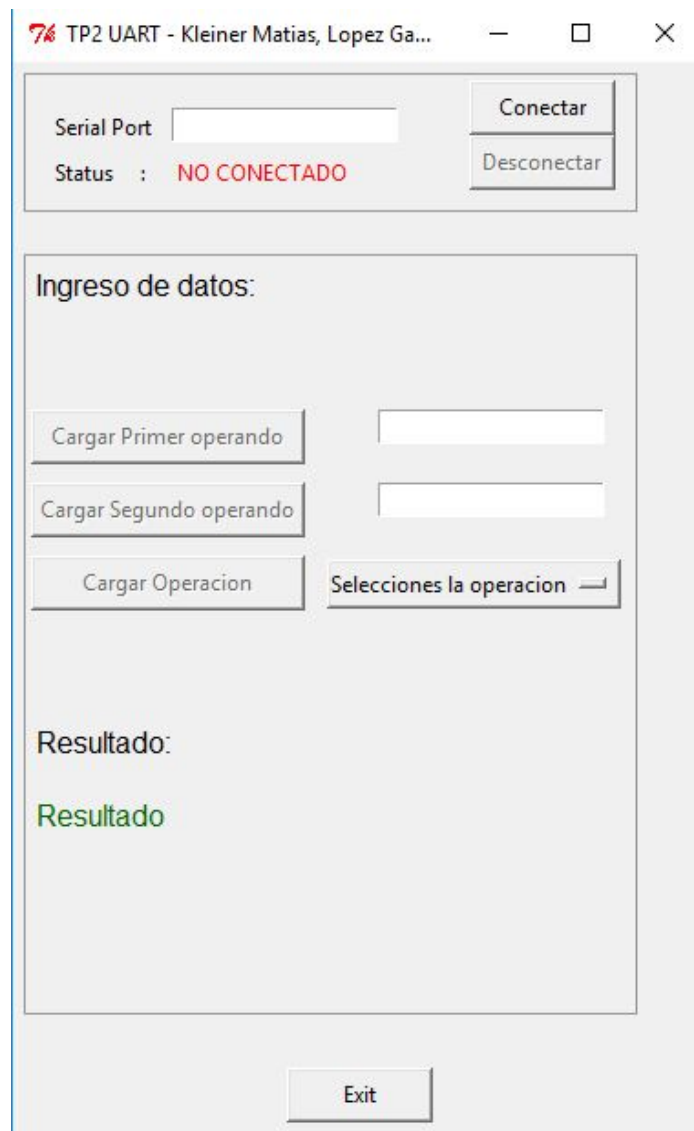


Figura 8 - GUI efectuada en Python.

Conclusión.

Una vez terminado el práctico, a través de la herramienta Vivado se obtuvieron las especificaciones en la utilización de los recursos de la FPGA y en la potencia que consume dicho circuito instanciado. Además, en este práctico se afianzaron los conceptos en cuanto al lenguaje de descripción de hardware Verilog.

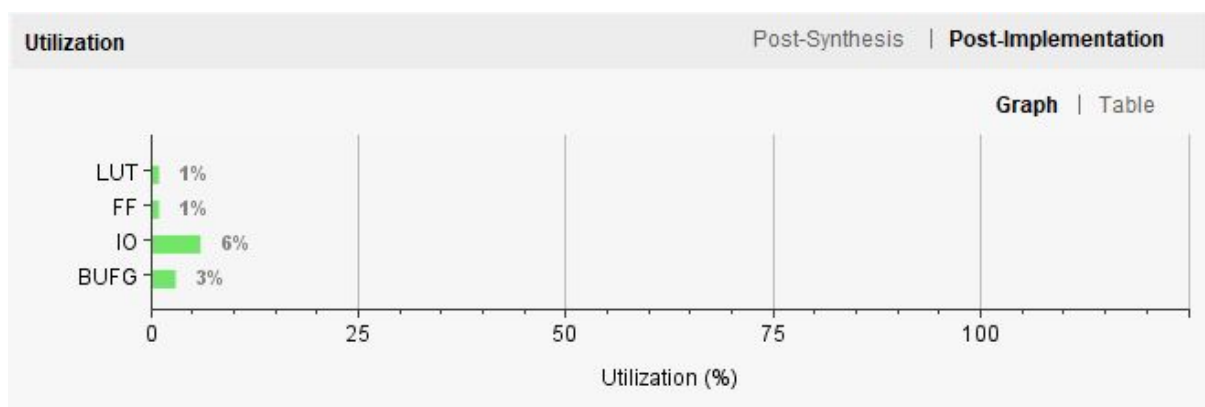


Figura 9 - Recursos utilizados para el desarrollo del proyecto.

Power		Summary On-Chip
Total On-Chip Power:	0.06 W	
Junction Temperature:	25.3 °C	
Thermal Margin:	74.7 °C (15.5 W)	
Effective θ_{JA} :	4.8 °C/W	
Power supplied to off-chip devices:	0 W	
Confidence level:	Medium	
Implemented Power Report		

Figura 10 - Potencia.