Trabajo Práctico 2

Comunicación UART

Materia: **Arquitectura de Computadoras**

Alumnos: **Coutinho, Juan Agustín** **Mat.:36548307**
Malatini, Hernán **Mat.:37196715**

Profesor: **Santiago Rodriguez**

Año Lectivo: **2016**

Introducción

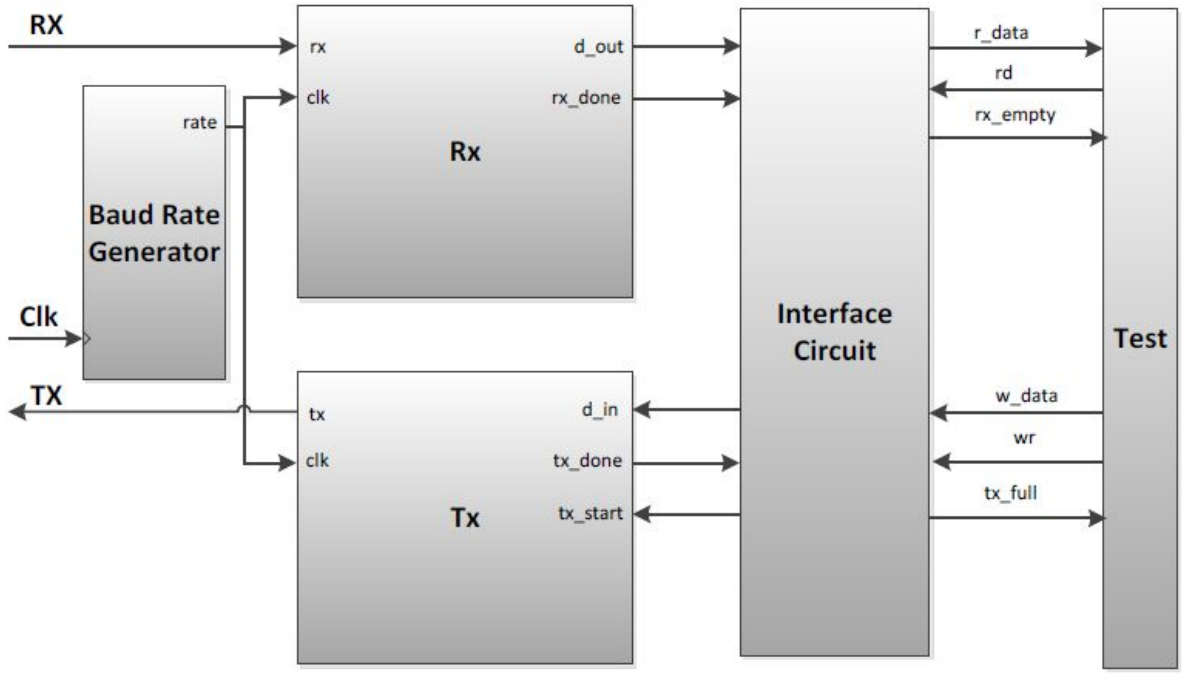
En el presente trabajo práctico se implementará un Universal Asynchronous Receiver-Transmitter (UART) sobre la placa de desarrollo Nexys 3, la cual contiene una FPGA Spartan E6.

El lenguaje de programación utilizado será Verilog y el entorno de desarrollo será el proporcionado por Xilinx (ISE Design Suite 14.9).

Desarrollo

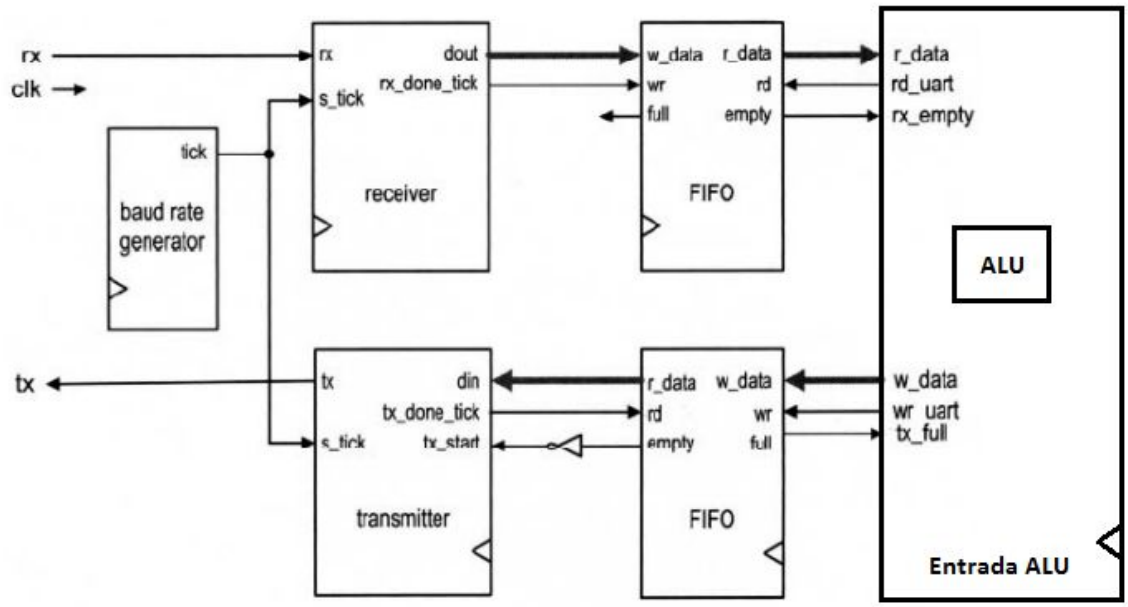
Funcionamiento Deseado

El esquema simplificado de la implementación que debe realizarse sobre la placa de desarrollo es la siguiente:



La placa de desarrollo recibirá datos (operandos y código de operación) a través de RX desde la PC, los cuales serán procesados por una ALU (Unidad Aritmético Lógica) en la FPGA, la cual devolverá un resultado a través de TX hacia la PC.

Diagrama de bloques del sistema desarrollado

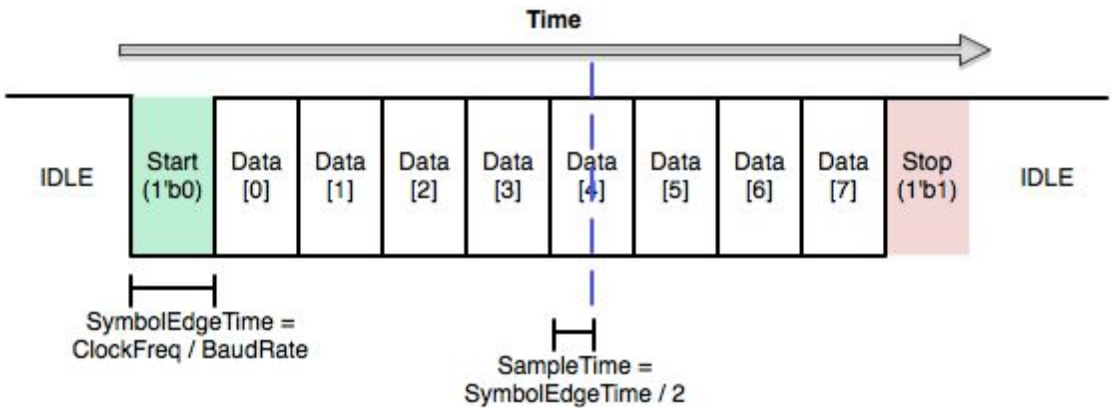


Baud Rate Generator

Este bloque es el encargado de generar una señal tick, a través del clock. La frecuencia con la que se genera depende de la fórmula $\frac{CLOCK}{BAUDRATE * 16}$. En nuestro caso utilizamos un clock de 100 MHz proporcionado por la placa, y un baudrate de 19200, lo cual nos da un período de 325.5 cuentas de clock (100 Mhz). Es decir, se generará un pulso (tick) cada $3.25x10^{-6}$ segundos. La duración del pulso de tick, es de un periodo del clock.

Receiver

Este bloque se encargará de muestrear el dato transmitido desde la PC a través de "rx", considerando un Baudrate de 19200 (bits por segundo). Cada dato recibido tiene el siguiente formato:



Una vez que se ha muestreado todo el dato, se lo coloca en "dout" y luego se envía una señal de "rx_done_tick", que le indicará a la FIFO que el dato está listo para ser apilado.

Transmitter

Este bloque recibirá una secuencia de bits a través de la entrada “din”, y cuando se reciba un pulso por la entrada “tx_start”, comenzará a transmitir de a un bit, a través de la salida “tx”. Una vez que finalice la transmisión de esos datos, se pondrá en uno la salida “tx_done_tick”.

FIFO (First Input First Output)

Éste bloque es utilizado tanto por el receptor como por el transmisor. La FIFO del receptor será el encargado de almacenar los tres datos recibidos de la PC (Código de Operación, Operando 2 y Operando 1) y la FIFO del transmisor almacenará el resultado de la salida de la ALU (con un dato en la FIFO ya comenzará el proceso de transmisión, ya que se colocará en 0 la salida “empty” del módulo FIFO, y dicha señal negada le indica al módulo “Transmitter” que inicie el envío del dato).

Entrada ALU

Este bloque, contiene una secuencia de estados, la cual se encargará de ir pasando los datos de la FIFO del receptor a la ALU una vez que se hayan recibido los tres valores necesarios. Los 3 datos se desapilan y colocan en las entradas correspondientes de la ALU en la siguiente secuencia: Operando 1, Operando 2 y Código de Operación. Una vez que el circuito puramente combinacional “ALU” posee los tres datos necesarios, realiza el cálculo, y devuelve el resultado de la operación, el cual se enviará a la FIFO del transmisor.

Diagramas de flujos

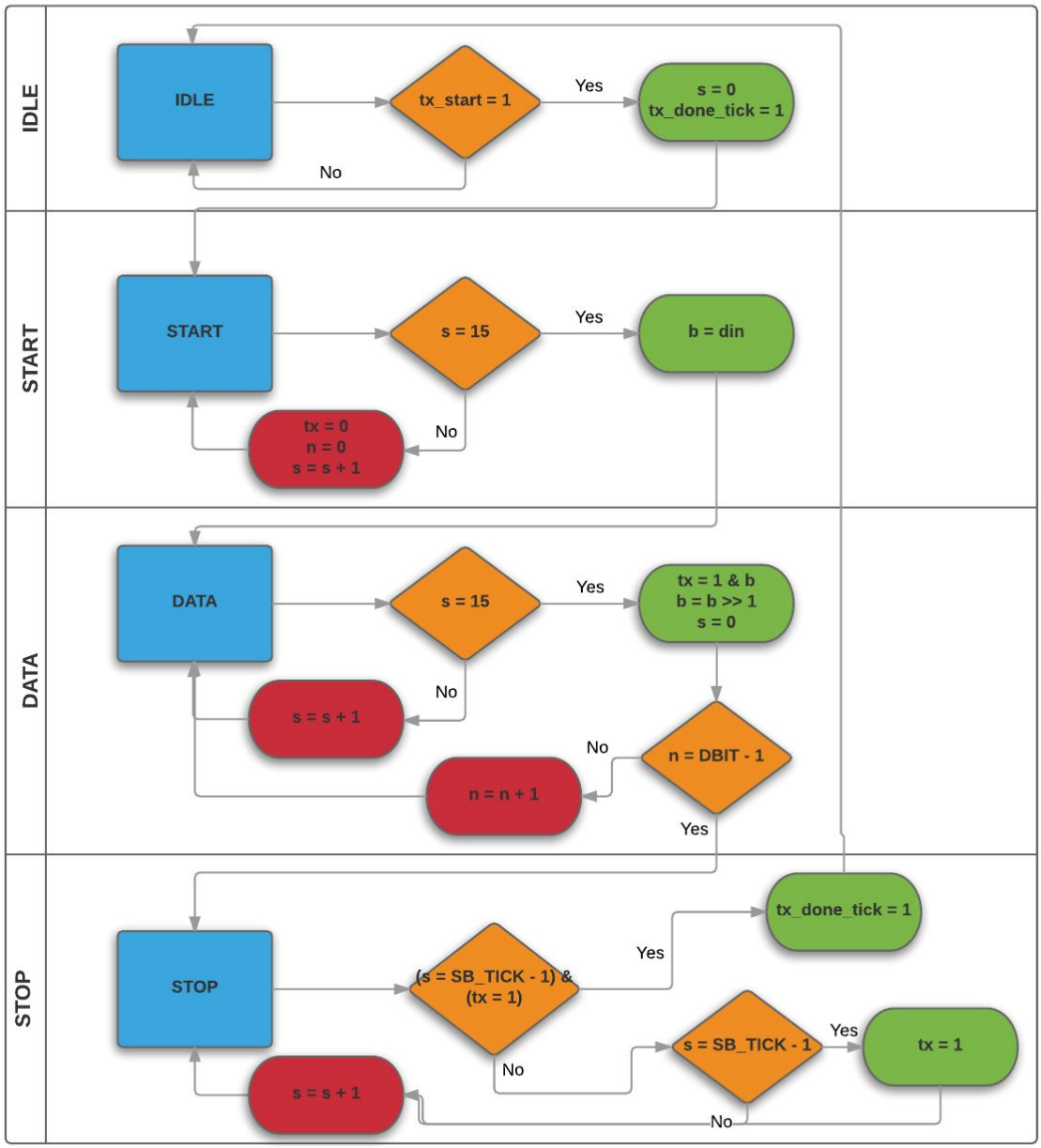
Para un entendimiento del código fuente, y además del funcionamiento de los bloques, se realizarán los diagramas de flujo de los bloques Receptor, Transmisor y Entrada ALU, que son los bloques que cuentan con máquinas de estado.

RECEPTOR

Referencias de variables (Rx)

- b: Registro donde se irán almacenando los bits recibidos.
- s: Contador de lo “s_tick” de entrada que provienen del baudrate.
- n: Contador de la cantidad de bits recibidos hasta el momento.
- DBIT: Constante pasada como parámetro a “Receptor” y “Transmisor” que indica la cantidad de bits de palabra utilizado en el UART.
- stop_bit: Bandera utilizada en el receptor para que al leer el último bit recibido (bit de stop), se cuenten 7 “s_tick” más para que al pasar al estado IDLE ya no se esté sobre dicho bit.

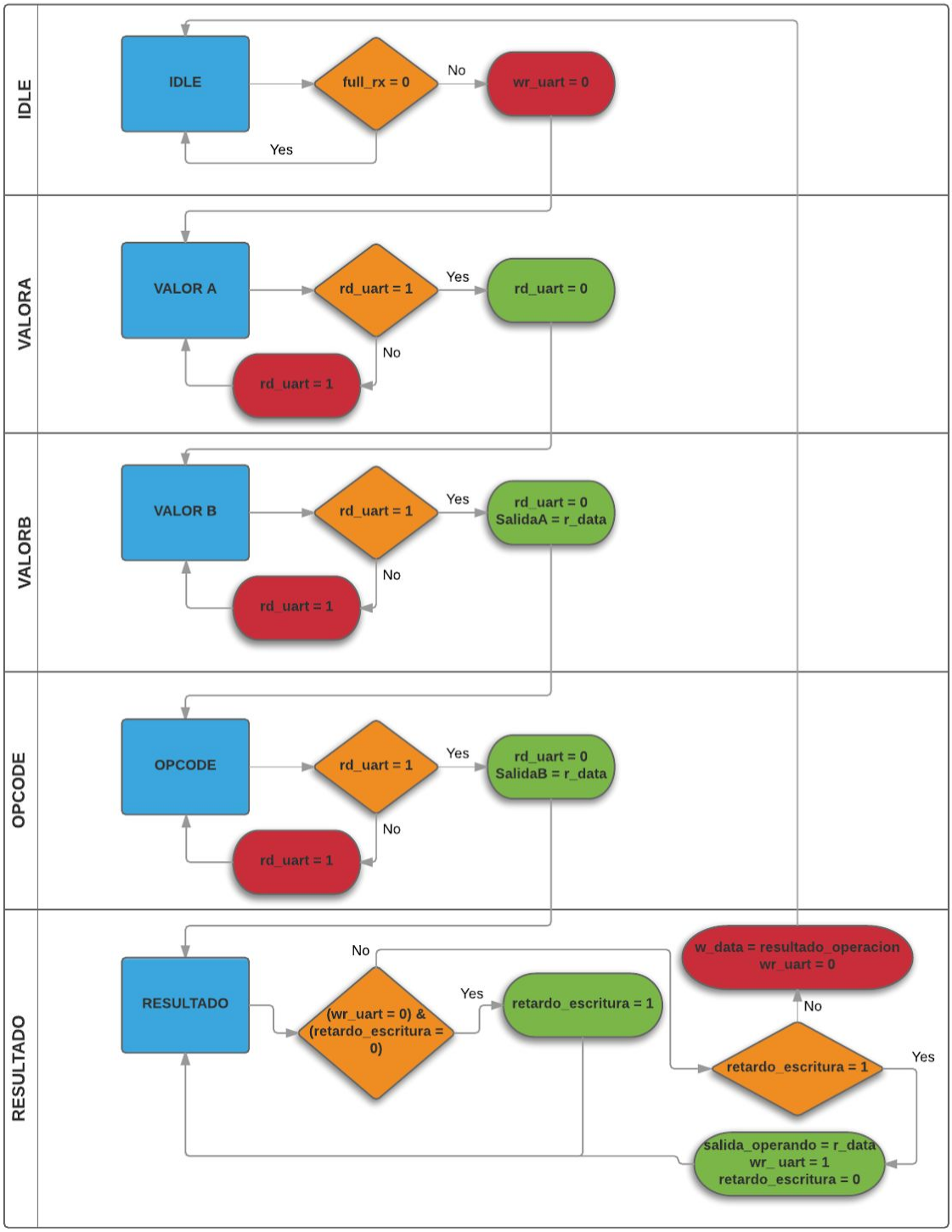
TRANSMISOR



Referencias de variables (Tx)

- b: Registro donde se almacenará los bits a enviar.
- s: Contador de los “s_tick” de entrada que provienen del baudrate.
- n: Contador de la cantidad de bits recibidos o enviados hasta el momento.
- DBIT: Constante pasada como parámetro a “Receptor” y “Transmisor” que indica la cantidad de bits de palabra utilizado en el UART.

ENTRADA ALU


IDLE
IDLE
full_rx = 0
No
wr_uart = 0
Yes
VALORA
VALOR A
rd_uart = 1
Yes
rd_uart = 0
No
rd_uart = 1
VALORB
VALOR B
rd_uart = 1
Yes
rd_uart = 0
SalidaA = r_data
No
rd_uart = 1
OPCODE
OPCODE
rd_uart = 1
Yes
rd_uart = 0
SalidaB = r_data
No
rd_uart = 1
RESULTADO
RESULTADO
No
(wr_uart = 0) &
(retardo_escritura =
0)
Yes
retardo_escritura = 1
w_data = resultado_operacion
wr_uart = 0
No
retardo_escritura = 1
Yes
salida_operando = r_data
wr_ uart = 1
retardo_escritura = 0

Referencias de variables (Entrada ALU)

- SalidaA, SalidaB, OpCode: Son las entradas a la ALU (puramente combinacional).
- retardo_escritura: Variable utilizada como bandera para retardar la escritura un ciclo de reloj más y de ésta manera tener tiempo de colocar el OpCode a la entrada de la ALU y obtener el resultado antes de que sea leído para colocarlo en la FIFO del transmisor.

Test Bench

Archivo “test_main_completo”

El test bench realizado consiste en enviar a través de “rx” los siguientes datos (los valores en rojo son el “start bit” y el “stop bit” respectivamente) :

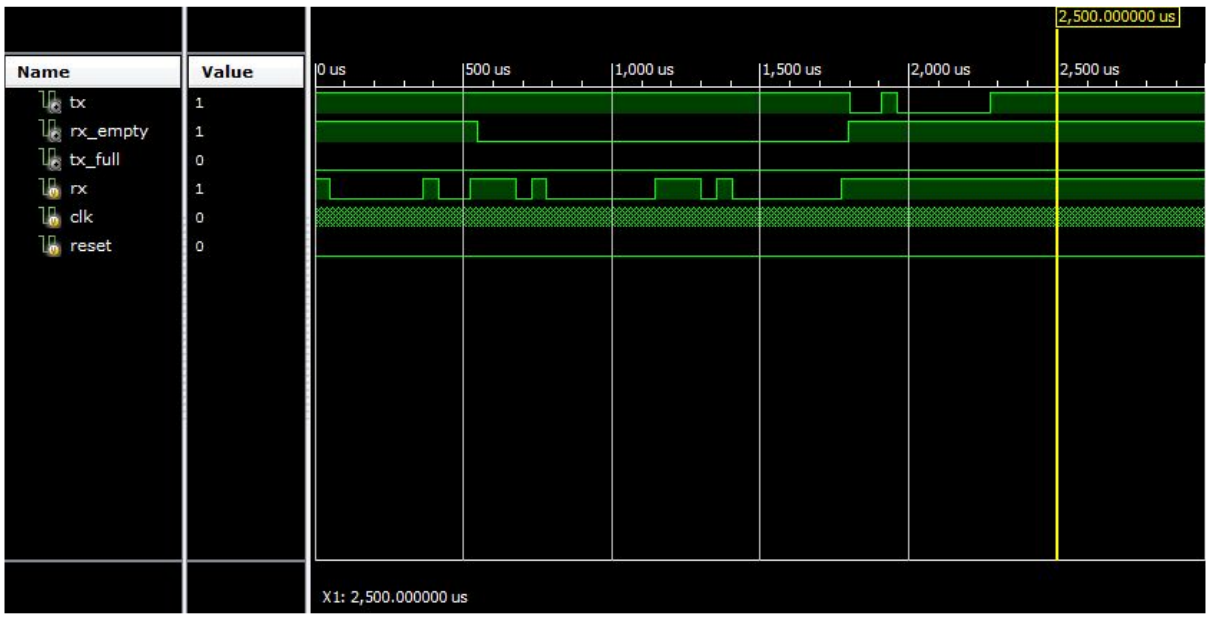
- 0**00000100**1 -> Suma
- 0**10000000**1 -> Operando2
- 0**10000000**1 -> Operando1

Cabe aclarar que los datos desde la PC a la placa son enviados comenzando por los bits menos significativos a los más significativos.

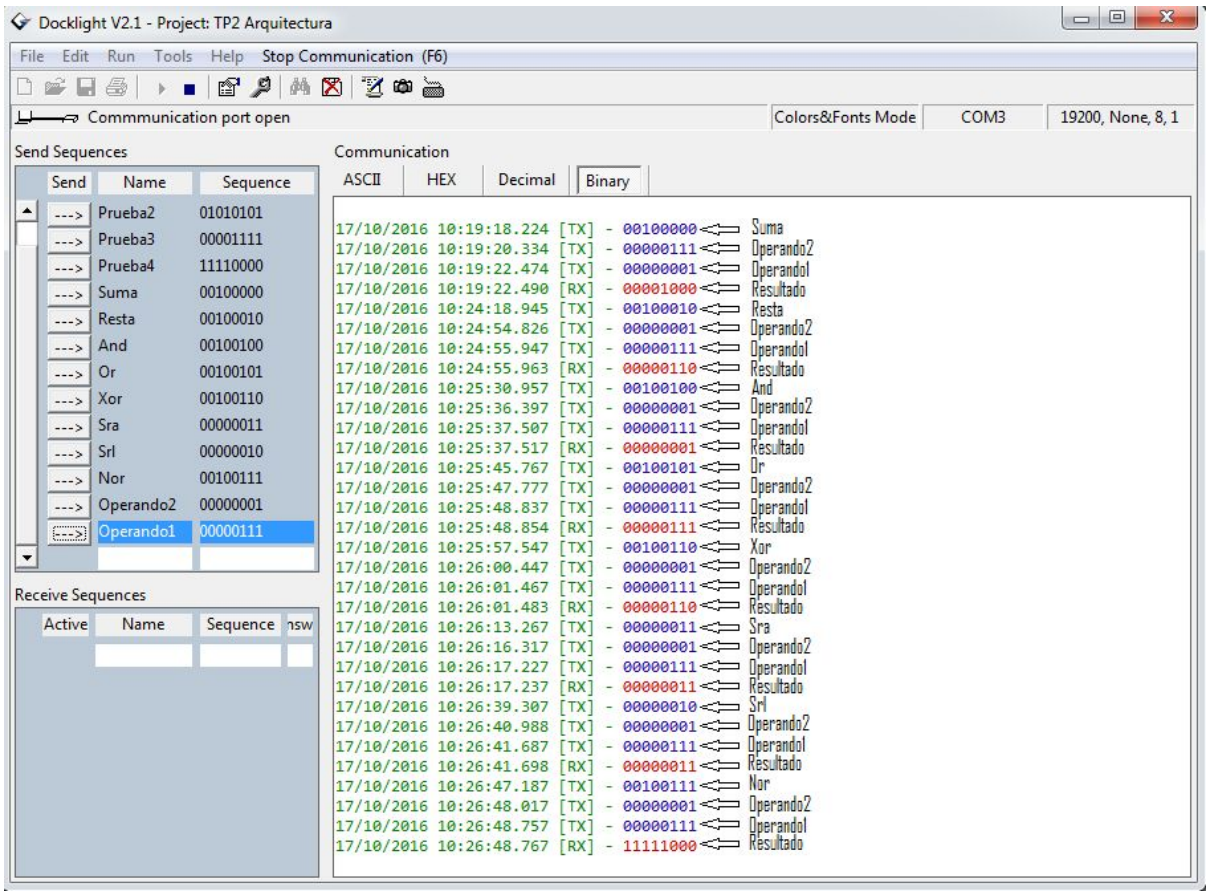
Se observa que el resultado por “tx” es:

- 0**01000000**1

Siendo el resultado correcto.



Prueba de Funcionamiento



Se ha utilizado el software “Docklight” el cual nos permite visualizar los datos enviados y recibidos en formato binario.

En la imagen los datos enviados a la FPGA por UART son los valores en azul y la respuesta por parte de la FPGA (resultado de la operación) son los valores de color rojo.