

Universidad Nacional de Córdoba

Facultad de Ciencias Exactas, Física y Naturales

Arquitectura de Computadoras

Trabajo Práctico N°2

Comunicación UART para módulo ALU

16/11/2022

Docentes:

Pereyra, Martin.

Rodríguez, Santiago.

Alumnos:

LAMBERTI, Germán Andrés

PEREZ, Bruno Santiago

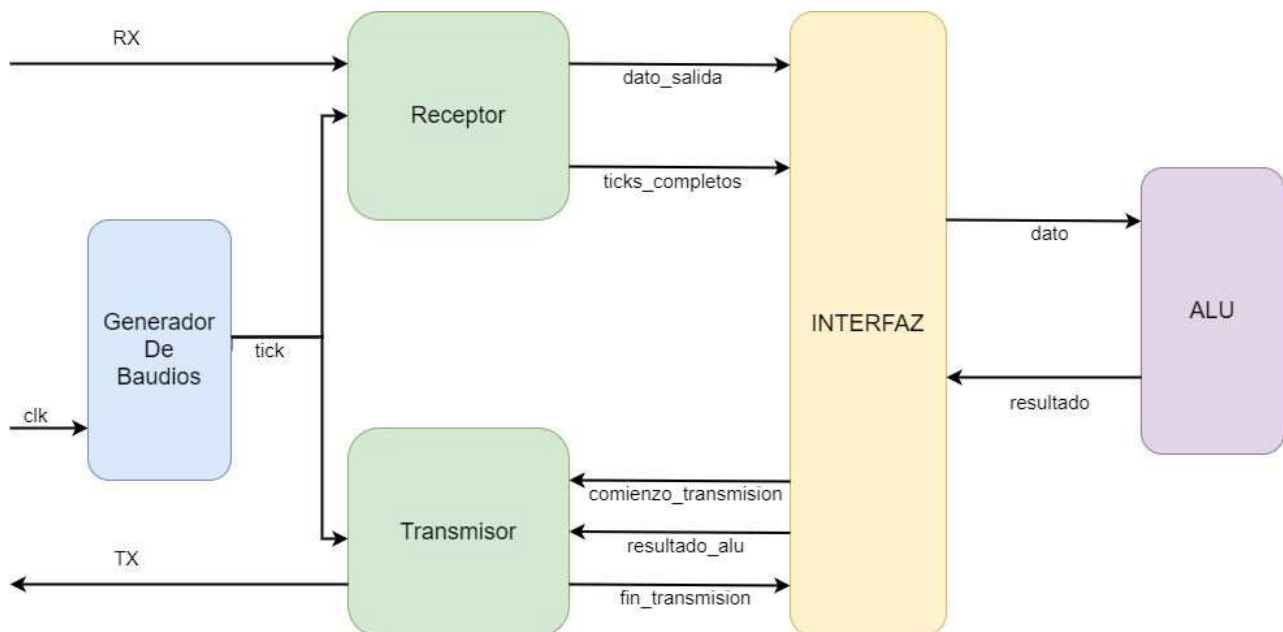
Introducción

En este trabajo se pretende simular a través del lenguaje verilog un módulo de comunicación UART para la Unidad Aritmética Lógica (ALU) desarrollada en el trabajo práctico 1.

El propósito de este trabajo práctico es afirmar los conceptos vistos en clase a través del desarrollo y la implementación de comunicación UART.

Arquitectura a implementar

A través del ide Vivado HLS se implementaron los diferentes módulos y test correspondientes con el objetivo de construir un módulo de comunicación UART para comunicarse con la ALU como mencionamos anteriormente desarrollada en el TP N° 1 de la materia.



¿Qué es un módulo UART?

Es un dispositivo transmisor y receptor asíncrono serial (del inglés Universal Asynchronous Receiver Transmitter). Que sea asíncrono quiere decir que no se utiliza un clock entre el transmisor y receptor ósea que el sincronismo está en los datos que se envían.

Por otro lado, que sea serial quiere decir que solo hay un cable para enviar y otro para transmitir los datos. Al carecer de una señal de clock ambas partes (transmisor y receptor) deben estar de acuerdo con que velocidad (en baudios) se va a realizar la comunicación.

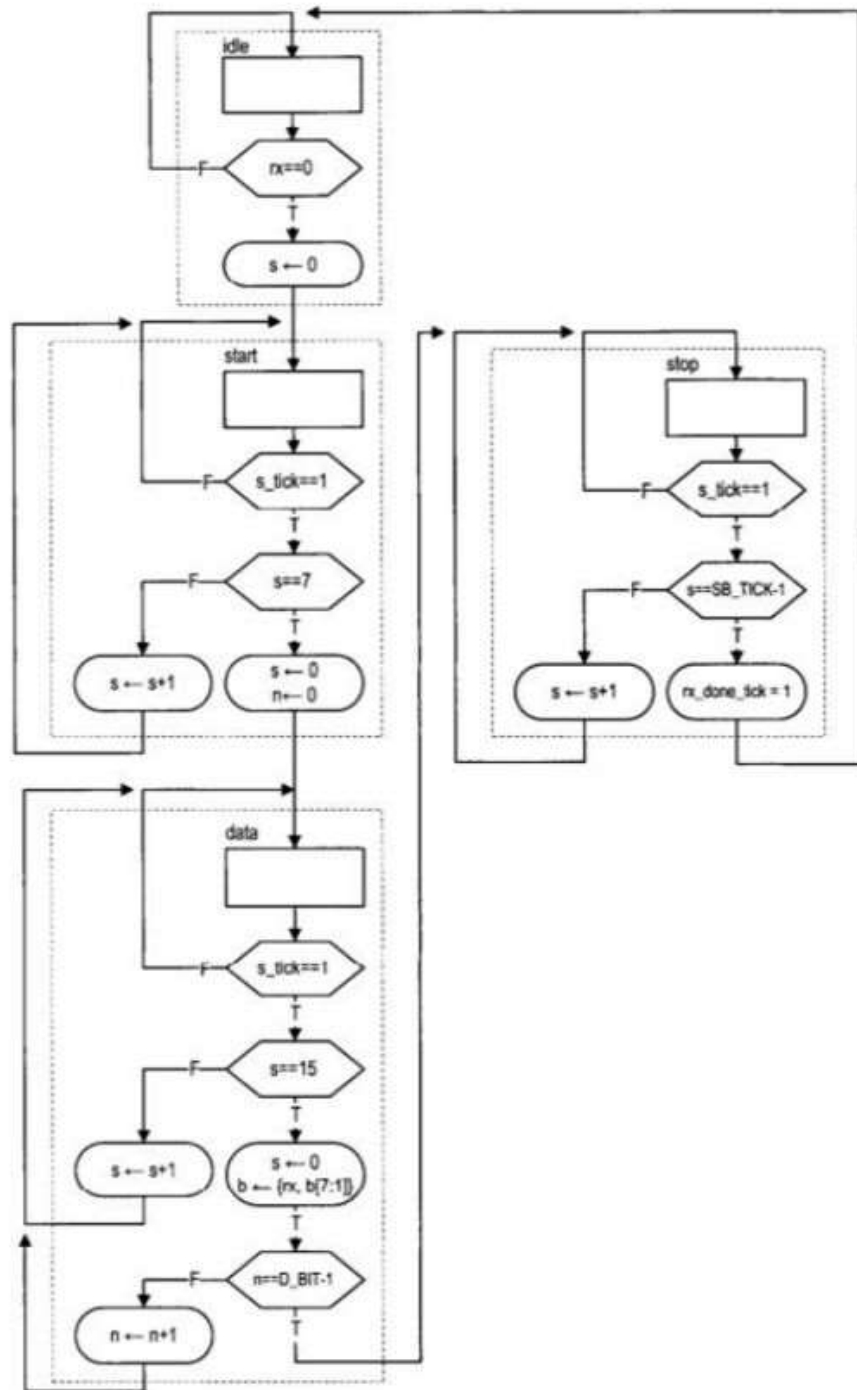
Solución de diseño

Los módulos receptor y transmisor se comportan como máquinas de estados. Estas máquinas de estados son muy similares debido a que tienen los mismos estados.

Los estados son los siguientes:

1. Wait
2. Start
3. Data
4. Stop

Estados del receptor



Wait (Idle): es el estado en el que el receptor está esperando la señal de start por medio de la línea de recepción RX, al recibir un 0 en RX se pasa al siguiente estado llamado **Start**.

Start: En este estado se tiene en cuenta principalmente la cantidad de ticks mandados por el Generador De Baudios. Cuando este contador llegue a 7 quiere decir que estamos en la mitad del bit Start de la trama UART, esto quiere decir que a partir de este bit se van a enviar los datos por ende el nuevo estado es **Data**.

Data: En este estado, como se dijo anteriormente están los bits de datos. Estos se van almacenando en un registro llamado registroDatos. La forma de cargar los datos que llegan por RX a dicho registro, es a través de la concatenación, el primer bit introducido es el más significativo (MSB) y así el último bit es el menos significativo (LSB) En este práctico el dato es de 8 bits, por lo tanto, después de detectar los mismos, el módulo pasa al siguiente estado **Stop**.

Stop: se detecta el bit de stop, que debe ser un "1" lógico. Una vez detectado, el módulo avisa con un pulso por serial_ticks_completos que la salida del receptor, salida_receptor, ya puede ser leída. Luego pasa al estado inicial **Wait** preparado para recibir un nuevo dato.

Estados del transmisor

Wait: el módulo transmisor está esperando la señal Comienzo_TX para comenzar a transmitir lo que tiene en su entrada, entrada_transmisor. Mientras Comienzo_TX no sea un 1 lógico, a la salida del transmisor va a ponerse un 1 lógico. Cuando comienzo_TX pasa a 1, se guarda en el registro de datos interno del transmisor lo que hay en entrada_transmisor y se pasa al estado **Start**.

Start: se comienza a transmitir el bit de start con una duración de 16 pulsos del generador de baudios, para avisar al receptor que se va a comenzar a transmitir datos y se pasa al estado **Data**.

Data: en este estado se comienzan a transmitir los bits que tiene cargado el registro de datos interno del transmisor, se comienza transmitiendo el bit más significativo (MSB) y por último el menos significativo (LSB). La duración de cada bit a la salida del transmisor es de 16 pulsos de ticks generador por el Generador de Baudrate.

Stop: luego de enviar los 8 bits correspondientes al dato, se envía el bit de stop para que el receptor sepa que finalizó la transmisión, este estado tiene la capacidad de avisar a la interfaz que el dato ya se ha enviado por completo mediante una pulso en serial_ticks_completos.

Luego el módulo transmisor vuelve a su estado inicial **Wait** listo para enviar otro dato.

Interfaz UART/ALU

También se la puede ver como una máquina de estados. Estos son:

1. Operando_A
2. Opaerando_B
3. Operación
4. Iniciar_Transmisión
5. Wait_Fin_Transmisión

Operando_A

La interfaz permanecerá en este estado hasta que le llegue la señal de recepción terminada, cuando llegue dicha señal, la interfaz tomará el dato a la salida del receptor y lo colocara en la la posición Dato_A de la ALU.

Operando_B

La interfaz permanecerá en este estado hasta que le llegue la señal de recepción terminada. cuando llegue dicha señal, la interfaz tomará el dato a la salida del receptor y lo colocara en la la posición Dato_B de la ALU.

Operación

La interfaz permanecerá en este estado hasta que le llegue la señal de recepción terminada. cuando llegue dicha señal, la interfaz tomará el dato a la salida del receptor y lo colocara en la la posición Operación de la ALU.

Iniciar_Transmisión

Una vez completada la carga de los registros Dato_A, Dato_B y Operación, se pasa al estado en donde se le indica al transmisor que ya puede comenzar a transmitir, esto se hace mediante una señal, Comienzo_TX.

Wait_Fin_Transmision

En este estado se espera que el transmisor envíe a la interfaz la señal de fin_transmision la cual le indica a la interfaz que la transmisión ha finalizado, lo que implica que ya se está en condiciones para cargar nuevos valores a los campos de entrada de la ALU.