



# Diseño de una UART

## Diseño de una UART

El objetivo de este proyecto es el diseño para FPGA de un módulo **Universal Asynchronous Receiver Transmitter** (UART) que permita la transmisión/recepción serie de datos de 8 bits a 9600 baudios. Para ello, se diseñará en VHDL el módulo transmisor y el módulo receptor, de acuerdo con la funcionalidad que se describe a continuación. Se utilizará la placa de desarrollo [Spartan-3 Starter Kit](#) de Digilent, basada en una FPGA Spartan-3 400 de Xilinx. El diseño funcionará con la señal de reloj **CLK** de 50MHz generada por el oscilador incluido en dicha placa.

### Transmisor

El módulo transmisor enviará por la salida **TX\_OUT**, de 1 bit, el dato de 8 bits en la entrada **TX\_DATA** cuando se active un pulsador conectado a la entrada **BTN\_IN**. La activación del pulsador genera un cambio de '0' a '1' en la señal **BTN\_IN**.

Para implementar la funcionalidad del transmisor se describirá un registro de desplazamiento *Transmitter Shift Register* de 10 bits. Cuando se active la transmisión, este registro se cargará con '1' en su bit más significativo, el valor del dato a transmitir en los 8 bits siguientes y con un '0' en el bit menos significativo. Este último bit, siempre a '0', supone el bit de inicio en el protocolo de la comunicación, y el primero, siempre a '1', el bit de parada. Como el estándar RS-232 considera que el primer bit del paquete de datos que se envía es el menos significativo, el desplazamiento del registro debe ser a la derecha. De acuerdo con el estándar de la comunicación, el estado por defecto de **TX\_OUT** debe ser '1'.

Adicionalmente, el módulo transmisor tiene una señal de salida **TX\_READY** que indica con el valor '1' que el transmisor está disponible para enviar un paquete de datos, tomando valor '0' cuando se está realizando una transmisión.

Considera una señal de reset asíncrono **RESET** que asigna el valor inicial por defecto adecuado a todos los componentes del diseño que sea necesario.

La Figura 1 muestra el esquema RTL simplificado del circuito que se propone, basado en el registro de desplazamiento *Transmitter Shift Register* mencionado anteriormente y el contador *Transmitter counter* que lleva la cuenta, una vez iniciado el proceso de transmisión, del número de bits transmitidos. En la Figura 1, el *Transmitter BaudRate Generator* se encarga de generar la señal de actualización con la temporización adecuada para la velocidad de transmisión de 9600 baudios. El control del proceso de transmisión es llevado a cabo mediante una Máquina de Estados Finitos para la cual se propone el diagrama de estados de la Figura 2. Debe diseñarse como una máquina de estados de Moore.

La detección de la activación del pulsador conectado a **BTN\_IN** que inicia la transmisión se realizará mediante una FSM para la detección de flancos ascendentes en una señal.



# Diseño de una UART

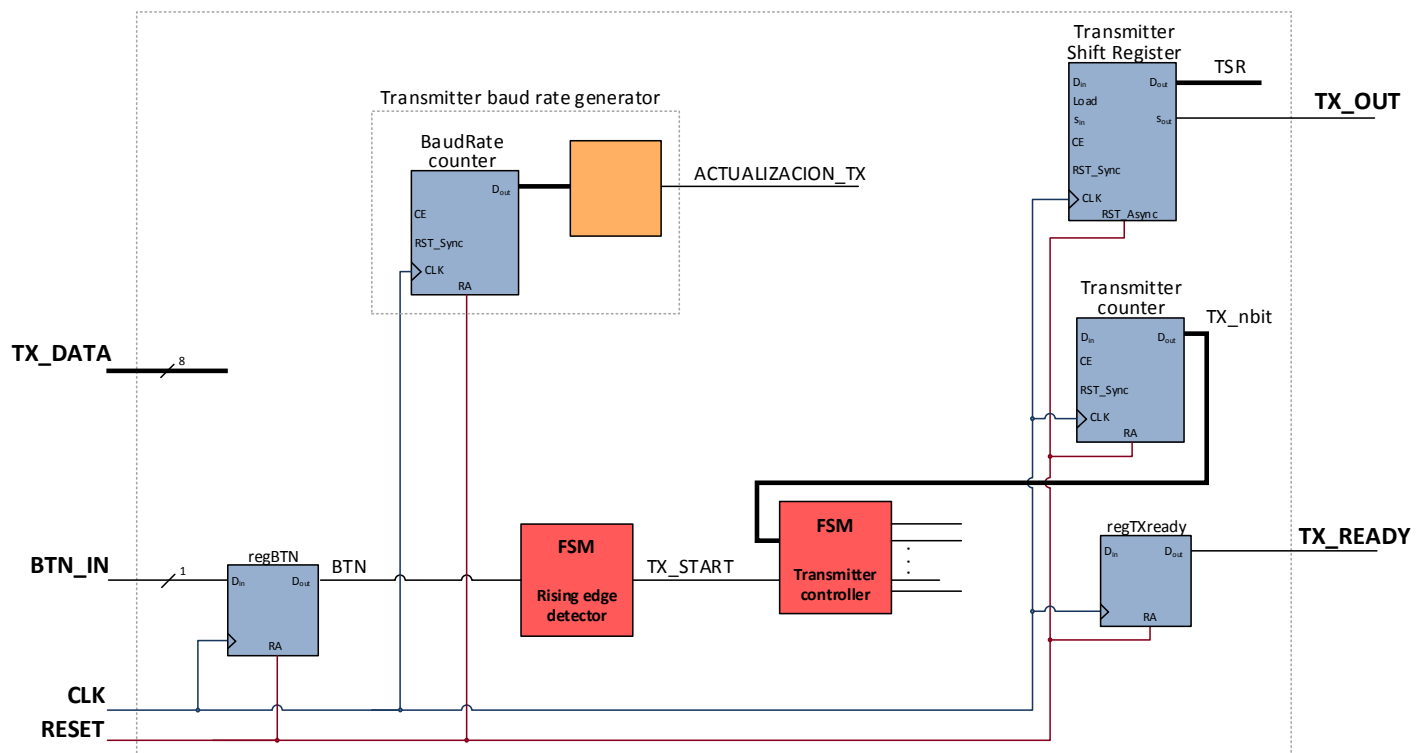


Figura 1. Esquema RTL simplificado del módulo Transmisor.

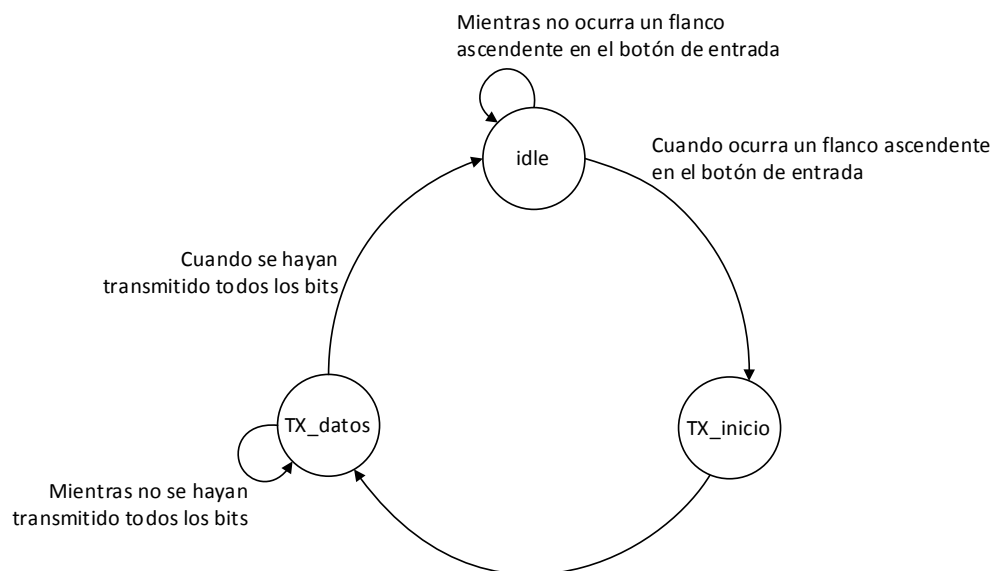


Figura 2. Diagrama de la Máquina de Estados Finitos propuesta para el control del proceso de transmisión.



# Diseño de una UART

---

## Receptor

El módulo receptor recibe una secuencia de bits por la señal de entrada **RX\_IN**, de 1 bit. Cuando se haya completado la recepción del dato, éste estará presente en la señal de salida **RX\_DATA**, de 8 bits.

Para implementar la funcionalidad del módulo de recepción se propone el esquema RTL mostrado en la Figura 3, basado en el registro de desplazamiento *Receiver Shift Register*, de 9 bits. El primer bit que se recibe será el bit de inicio enviado por el transmisor, que siempre tiene valor '0'; los siguientes 8 bits forman el dato recibido. La activación del proceso de recepción tendrá lugar, por tanto, cuando estando en estado de espera de recepción de datos se detecte un '0' en la señal de entrada **RX\_IN**. En ese momento debe comenzar un proceso que espere recibir, a una velocidad de 9600 baudios, un paquete de 9 bits, incluyendo ese primer bit a '0'. El contador *Receiver counter* lleva la cuenta de los bits recibidos. El *Receiver BaudRate Generator* se encarga de generar la señal de actualización con la temporización adecuada para la velocidad de recepción de 9600 baudios.

Adicionalmente, el módulo receptor tiene una señal de salida **RX\_NEWDATA**, de 1 bit, que indica que se ha recibido un dato. Para ello, esta señal tomará valor '1' cuando se haya completado la recepción del dato, únicamente durante un ciclo de reloj.

El control del funcionamiento de todos estos componentes es realizado por la Máquina de Estados Finitos *Receiver controller*, para la cual se propone el diagrama de estados de la Figura 4.

Considera una señal de reset asíncrono **RESET** que asigne un valor inicial por defecto adecuado a todos los componentes del diseño que sea necesario.



# Diseño de una UART

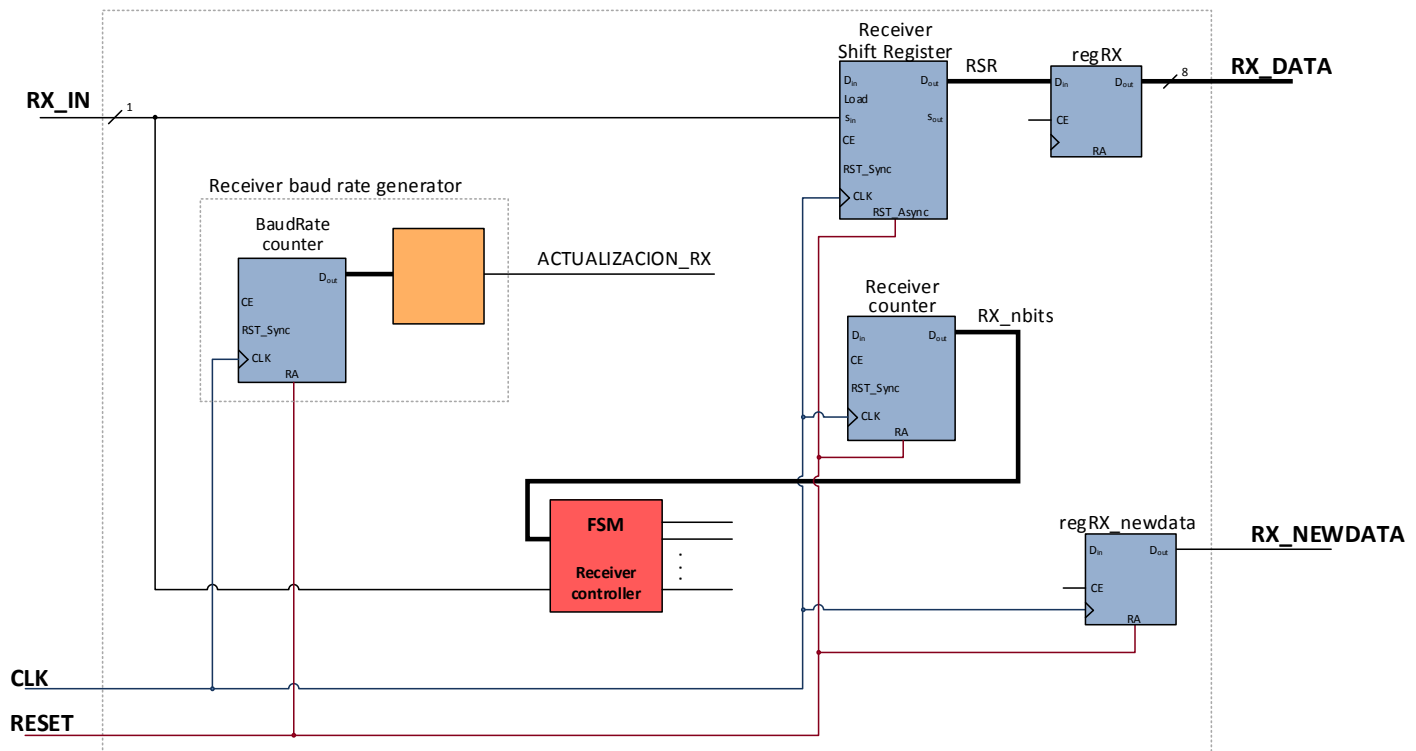


Figura 3. Esquema RTL simplificado del módulo Receptor.

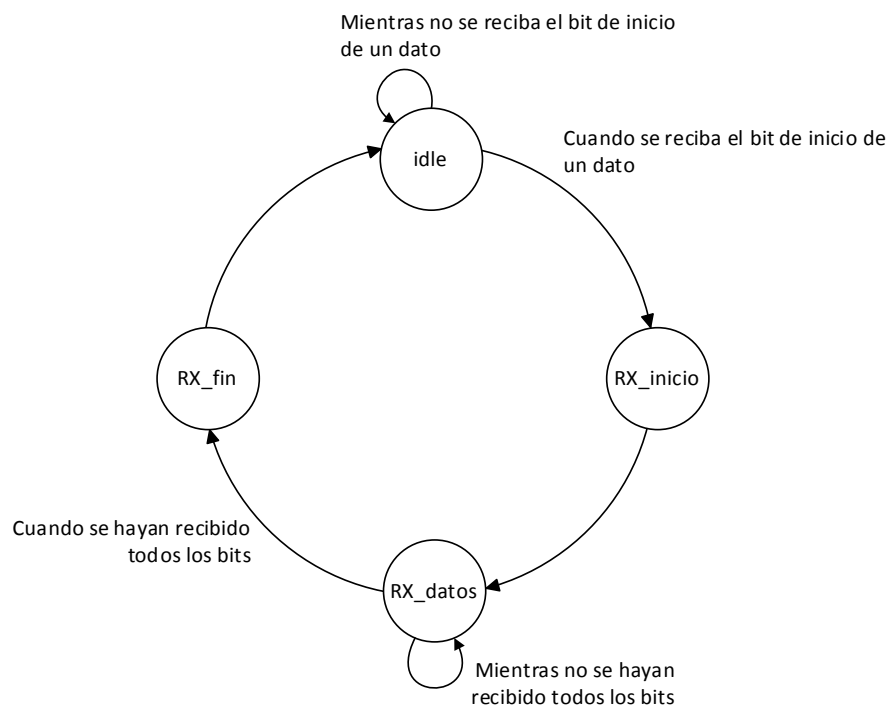


Figura 4. Diagrama de la Máquina de Estados Finitos propuesta para el control del proceso de recepción.



# Diseño de una UART

## Consideraciones sobre la velocidad de la comunicación.

Como se ha indicado, la UART debe recibir y transmitir datos a 9600 baudios. Al mismo tiempo, el circuito que la implementa debe funcionar con una señal de reloj de 50 MHz. Por tanto, es necesario controlar cuándo se actualizan los componentes secuenciales del diseño mediante la correspondiente señal de habilitación.

La Figura 5 muestra el aspecto de la señal que debe encargarse de esto. Se trata de una señal de frecuencia 9600Hz: durante cada periodo de tiempo, esta señal estará a '1' un único ciclo de reloj, permaneciendo en '0' el tiempo restante hasta completar el periodo.

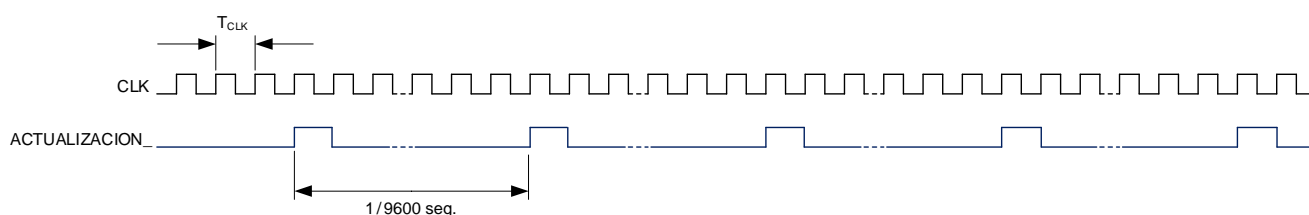


Figura 5. Señal de actualización.

Para lograr el funcionamiento independiente de los módulos transmisor y receptor es necesario que cada uno de ellos gestione su propia temporización y, en consecuencia, su propia señal de actualización. En los esquemas RTL de las Figuras 1 y 3 estas señales se han denominado ACTUALIZACION\_TX y ACTUALIZACION\_RX para los módulos transmisor y receptor, respectivamente. De la generación de estas señales se encargan los correspondientes *BAUDRATE Generator* incluidos en los esquemas RTL.

## Simulación e implementación en la placa *Spartan-3 Starter Kit*.

La funcionalidad de la UART debe validarse mediante la correspondiente simulación funcional, utilizando para ello el *testbench* adecuado.

Una vez validado su correcto funcionamiento, se completará el proceso de implementación y generación del *bitstream* para la descarga en la placa FPGA. Para ello, es necesario añadir al proyecto el archivo UART\_Spartan3.ucf disponible en la página en Aula Virtual de la asignatura. Este archivo asigna a cada puerto de entrada o de salida el pin de la FPGA adecuado para la implementación en la placa *Spartan-3 Starter kit* de este diseño.