



LABORATORIO 7: TRANSMISOR SERIAL ASINCRONO

OBJETIVO

Diseñar un sistema digital de transmisión asíncrona de datos seriales.

EQUIPOS

Digilent Nexys 4 DDR / CoolRunner II Board
Cable serial
Fuente DC.
Multímetro Digital.
Project Board.
Cables (jumpers)

PRELIMINAR

Cree un nuevo proyecto en Vivado. El proyecto (y el folder que contiene el proyecto y todos los archivos) **será el mismo a utilizar el trimestre completo**. Cada laboratorio nuevo agregará archivos al proyecto. Para implementar un archivo u otro en el FPGA, solo tiene que asegurarse de indicar que este archivo es el Top-Module.

Cuando este en el proceso de crear el proyecto, en la ventana de Project, asegúrese de lo siguiente:

- Family: Xilinx 7 series FPGAs
- Device: XC7A100TCSG324-1
- Preferred Language: VHDL
- Top-Level Source Type: VHDL

Agregue un módulo de VHDL a su proyecto llamado lab7.vhd. Cuando necesite agregar el archivo de restricciones (XDC), hágalo dando click-derecho sobre lab7.vhd y asócielo a este archivo.

DISCUSIÓN

El archivo lab7.vhd es su módulo Top para este laboratorio. En este módulo Top, va a unir/conectar los siguientes dos sub-módulos: Un sub-módulo llamado transmisor el cual se encargará de enviar los datos seriales por la línea de transmisión y un sub-módulo de pruebas el cual se encargará de transmitir los caracteres de su matrícula + el caracter "Carriage Return" que tiene el valor hexadecimal 0x0D. Busque en internet "ASCII Table" para ver una lista de todos los caracteres y sus diferentes valores.

Sobre la Comunicación Serial Asíncrona

La comunicación serial asíncrona utiliza una (1) sola línea de transmisión y los bits de datos son transmitidos uno a la vez a una frecuencia llamada Baud (bit) Rate. Cuando no se están enviando datos, la línea esta inactiva/desocupada, y este estado/nivel se le llama “mark”.

Para enviar datos, el transmisor envía un bit de start (el cual pone la línea en el estado/nivel llamado “space” por la duración de tiempo de 1 bit). Luego de esto, los datos son enviados un bit a la vez, comenzando por el bit menos significativo, en los cuales SPACE=0 y MARK=1. Finalmente, uno o más bits de stop (mark) son enviados, los cuales dejan la línea nuevamente en un estado de mark. Para este laboratorio utilizaremos un baud rate de 9600 baudios (bits por segundo), 8 bits de datos y ningún bit de paridad.

La Nexys 4 DDR cuenta con la capacidad de establecer comunicación mediante una conexión UART ya que cuenta con un circuito integrado puente USB-UART modelo FTDI FT2232HQ, el cual dispone de un conector microUSB tipo B denominado J6. Esta característica le ofrece al usuario la capacidad de comunicarse con aplicaciones de computadoras fácilmente mediante un puerto COM virtual que se crea en la computadora al momento en que se conecte la tarjeta con un puerto USB de esta y se instale el controlador del puerto COM virtual (esto ocurre usualmente de manera automática); el nombre del puerto COM varia entre dispositivos.

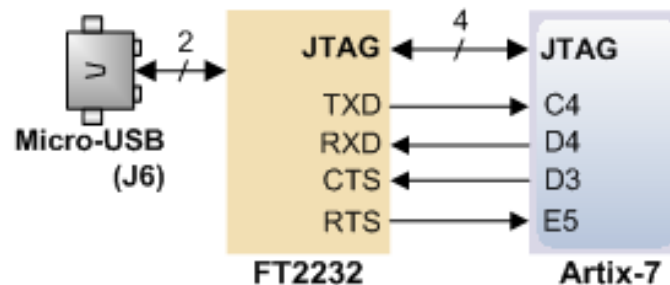


Figure 1 - Conector USB-UART

Note que en los pines de RXD y TXD, que se conectan al FPGA en D4 y C4 respectivamente, la polaridad tiene niveles de voltaje estándar CMOS (3.3V), por lo tanto, mark = high y space = low.

Sobre la Implementación del Módulo de Transmisión

El módulo de transmisión debe contener un registro llamado Transmit Buffer Register (TBR) y otro llamado Transmit Shift Register (TSR). La salida del TSR se conecta a la línea de transmisión (TXD). Al TSR solo puede cargarle datos cuando un bit de STOP ha sido enviado.

Todos los caracteres para transmitir deben ser guardados primero en el TBR (Transmit Buffer Register) y luego al TSR (Transmit Shift Register). Si el TBR contiene el caracter a ser enviado y el TSR no contiene nada, este caracter es pasado del TBR al TSR y la transmisión comienza. Entonces, el TBR se considera listo para recibir otro caracter hasta que le sea escrito un dato nuevamente en él. El TSR se considera ocupado hasta que todos los bits del caracter previamente guardado y el bit de stop han sido enviados.

Refiérase a la sección de “6 USB-UART Bridge (Serial Port)” del manual de usuario del Nexys 4 DDR para más información sobre los pines y conexiones entre el FPGA y el FTDI FT2232HQ:

<https://reference.digilentinc.com/reference/programmable-logic/nexys-a7/reference-manual>

PROCEDIMIENTO

Prepare su cuaderno de laboratorio: escriba el título y una breve descripción del laboratorio. Dibuje un esquemático que incluye los componentes que son parte de este laboratorio. Específicamente, incluya el FPGA (solo los pines utilizados en este laboratorio), el ST3232, el conector D-sub de 9 pines. En cada pin, incluya el número/coordenada de pin.

Diseñe el módulo de transmisión de datos serial. Su módulo debe tener las siguientes entradas: reset, reloj, bus de datos en paralelo (un bus de 8 bits) y una señal que carga el TBR. La señal que carga el TBR está activa durante un solo ciclo de reloj, cuando los datos presentados al bus de datos en paralelo son válidos. El modulo tendrá las siguientes salidas: salida de transmisión de datos (TXD) y una salida de estatus que indica que TRB está lleno o vacío. La entidad de este modulo debe declararla de la siguiente manera:

```
entity transmisor is
  port (
    clk : in std_logic; -- reloj
    reset : in std_logic; -- reset asincrono, activo high
    pdata : in std_logic_vector(7 downto 0); -- datos en paralelo
                                              -- (dato a enviar)
    load : in std_logic; -- cargar TBR
    txd : out std_logic; -- salida de transmision de datos
    empty : out std_logic; -- TBR vacio, active high
  end transmisor;
```

Necesitará un flip-flop interno para mantener el estatus de si el TBR está vacío o no. Tambien necesitara un controlador interno el cual esperará que el bit TBR esté lleno, luego transferirá los datos al TSR (por consecuencia, vaciando el TBR) y luego hacer el proceso de desplazamiento del caracter a enviar (bit de start + contenido del TSR + bit de stop) a un baud rate de 9600. Asuma (por ahora) un reloj de 100MHz. Debe generar el baud (bit) rate de 9600 en base al reloj de 100MHz (debe determinar el tiempo de duración de 1 bit a 9600 bits/s y obtener esta duración con el reloj de 100MHz... aunque no logre exactamente el mismo tiempo, si está por debajo del 1% de diferencia no resulta en problemas).

Escriba un banco de pruebas (Test Bench) que verifique que su módulo pueda transmitir caracteres de manera continua y que muestre el estatus del TBR (si está lleno o vacío). Simule el módulo transmisor, imprima el resultado y péquelo a su cuaderno de laboratorio.

Conecte el Nexys 4 DDR, el programador y el cable de transmisión serial. Utilice un programa de emulación de terminal serial (Termite, PuTTY, etc.) y configure el programa a utilizar un baud rate de 9600, 8 bits de datos, no utilizar bit de paridad y 1 bit de stop. Verifique que los datos que su módulo de pruebas envía están correctos. Demuestre el correcto funcionamiento del sistema al instructor.

- Escriba los procesos que generen las frecuencias (señales de reloj) necesarias para este proyecto.
- Prepare su cuaderno de laboratorio: escriba el título y una breve descripción del laboratorio.
- Dibuje un esquemático que incluye los componentes que son parte de este laboratorio. Específicamente, incluya el FPGA (solo los pines utilizados en este laboratorio), y todas las conexiones.
- Finalmente debe de realizar un vídeo en donde se muestre el circuito funcionando correctamente.

CONCLUSIÓN

Escriba un pequeño resumen de lo que hizo en el laboratorio, lo que aprendió y que sugeriría que pudiera mejorarse del diseño utilizado.