Práctica 8.

DISEÑO DE UN TRANSMISOR PARA COMUNICACIÓN SERIAL.

OBJETIVO:

Demostrar a los estudiantes mediante el diseño de un módulo transmisor (TX), empleado en comunicaciones de tipo serial UART (*Universal Asyncrhonous Receiver Transmitter*), la utilidad de este módulo, así como la importancia de su presencia en la arquitectura de un procesador para aplicaciones electrónicas en envío de información.

ESPECIFICACIONES:

Utilizando un FPGA y un switch de 4 posiciones, diseñar un módulo Transmisor serial, el cual sea capaz de leer el valor binario del switch, procesarlo en el FPGA y posteriormente enviarlo a una computadora personal, en donde el dato deberá estar en formato hexadecimal. La conexión entre el FPGA y la computadora deberá realizarse empleando un circuito convertidor USB TTL-Serial. La figura 8.1 muestra el diagrama de bloques del sistema.

DIAGRAMA DE BLOQUES:

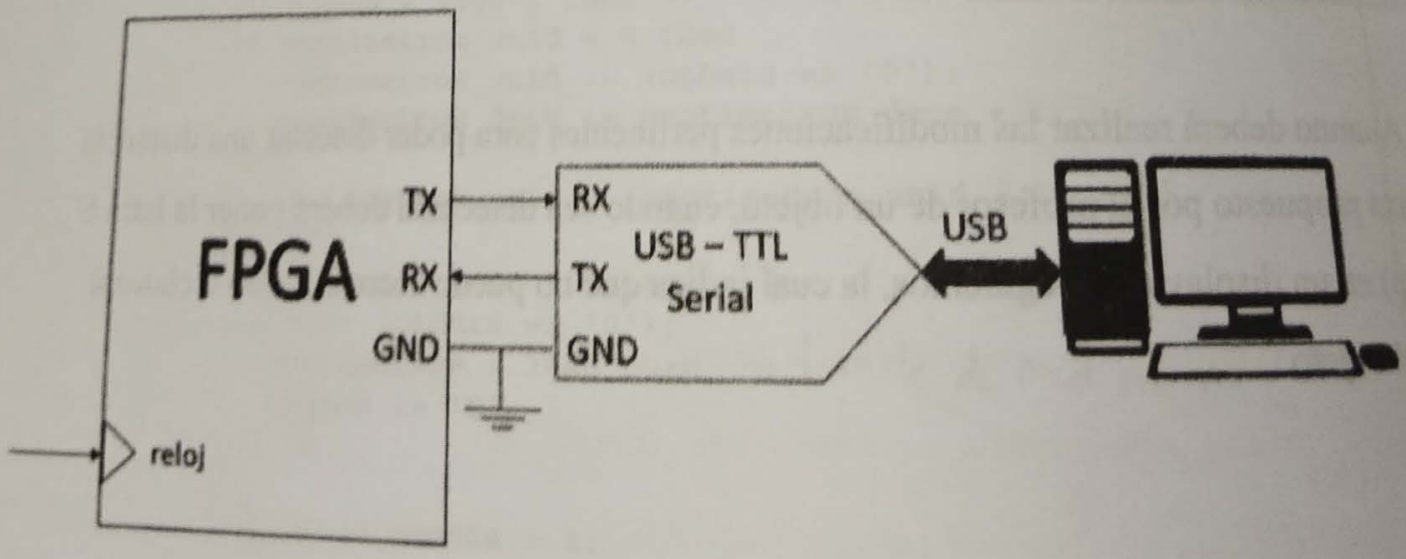

Figura 8.1. Diagrama de bloques para la comunicación serial

Un FPGA es un dispositivo lógico programable, el cual posee la característica de no contar con una arquitectura fija como en el caso de un procesador. Dicha característica trae

Práctica 8.

DISEÑO DE UN TRANSMISOR PARA COMUNICACIÓN SERIAL

OBJETIVO:

Demostrar a los estudiantes mediante el diseño de un módulo transmisor (TX), empleado en comunicaciones de tipo serial UART (*Universal Asyncrhonous Receiver Transmitter*), la utilidad de este módulo, así como la importancia de su presencia en la arquitectura de un procesador para aplicaciones electrónicas en envío de información.

ESPECIFICACIONES:

Utilizando un FPGA y un switch de 4 posiciones, diseñar un módulo Transmisor serial, el cual sea capaz de leer el valor binario del switch, procesarlo en el FPGA y posteriormente enviarlo a una computadora personal, en donde el dato deberá estar en formato hexadecimal. La conexión entre el FPGA y la computadora deberá realizarse empleando un circuito convertidor USB TTL-Serial. La figura 8.1 muestra el diagrama de bloques del sistema.

DIAGRAMA DE BLOQUES:

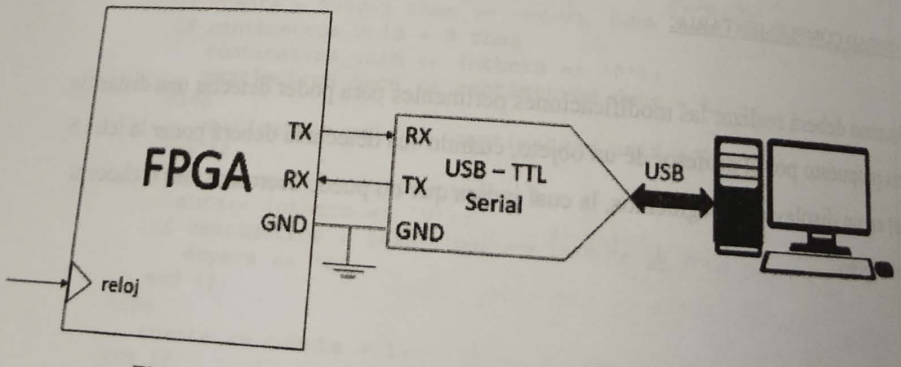

Figura 8.1. Diagrama de bloques para la comunicación serial

Un FPGA es un dispositivo lógico programable, el cual posee la característica de no contar con una arquitectura fija como en el caso de un procesador. Dicha característica trae

consigo la posibilidad de diseñar arquitecturas reconfigurables en donde la cantidad de puertos o módulos periféricos puede ser establecida de acuerdo a las especificaciones de diseño. Así, el diseño de un módulo TX de comunicación UART puede ser elaborado y configurado para realizar tareas específicas consumiendo el mínimo de recursos posible. La figura 8.2 muestra los bloques funcionales del sistema Transmisor, donde las señales se muestran como flechas de color azul, mientras que las terminales físicas se muestran en color rojo. Cada bloque funcional corresponde a un proceso que deberá ejecutarse dentro de la arquitectura.

BLOQUES FUNCIONALES:

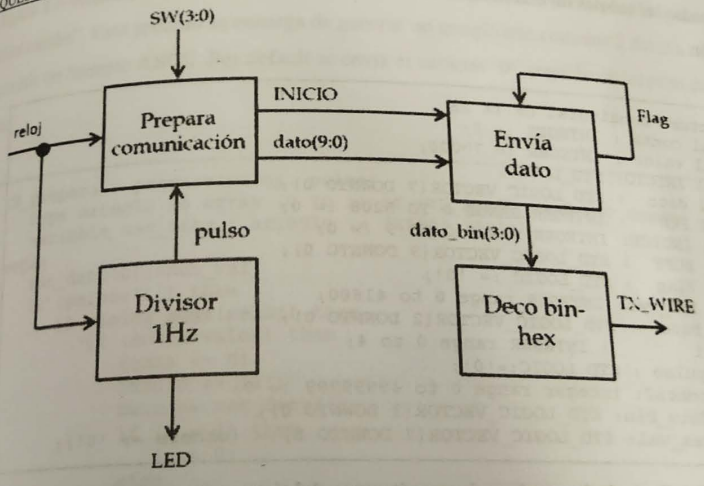

Figura 8.2. Bloques funcionales del sistema transmisor serial

La figura 8.3 muestra la parte entidad del sistema transmisor de comunicación serial. Las terminales físicas corresponden al reloj maestro del FPGA de 50 MHz, cuatro bits de un switch, un LED testigo y la línea de transmisión (TX_WIRE).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity TX is
  port( reloj    : IN  STD_LOGIC;
        SW       : IN  STD_LOGIC_VECTOR(3 downto 0);
        LED      : OUT STD_LOGIC;
        TX_WIRE  : OUT STD_LOGIC);
end entity;
```

Figura 8.3. Entidad del sistema Transmisor Serial TX

La figura 8.4 muestra la parte declaratoria de la arquitectura del módulo **TX.** en donde se declaran todas las señales involucradas en los diferentes procesos del sistema de transmisión.

```
architecture behaivoral OF TX IS
   signal conta : INTEGER := 0;
   signal valor : INTEGER := 70000;
   signal INICIO: STD_LOGIC;
   signal dato  : STD_LOGIC_VECTOR(7 DOWNTO 0);
   signal PRE   : INTEGER RANGE 0 TO 5208 := 0;
   signal INDICE: INTEGER RANGE 0 TO 9 := 0;
   signal BUFF  : STD_LOGIC_VECTOR(9 DOWNTO 0);
   signal Flag  : STD_LOGIC := '0';
   signal PRE_val: INTEGER range 0 to 41600;
   signal baud  : STD_LOGIC_VECTOR(2 DOWNTO 0);
   signal i       : INTEGER range 0 to 4;
   signal pulso : STD_LOGIC:='0';
   signal conta2: integer range 0 to 49999999 := 0;
   signal dato_bin: STD_LOGIC_VECTOR(3 DOWNTO 0);
   signal hex_val: STD_LOGIC_VECTOR(7 DOWNTO 0):= (others => '0');
```

Figura 8.4. Parte declaratoria en la arquitectura del sistema transmisor serial

La figura 8.5 muestra el proceso "TX_divisor" asociado al bloque funcional "Divisor 1 Hz". Éste se encarga de generar una señal denominada "pulso", la cual indica al siguiente proceso cuando es que debe preparar el dato que será transmitido. La configuración mostrada envía un dato cada segundo.

```
begin
  TX_divisor : process(reloj)
  begin
    if rising_edge(reloj) then
      contador<=contador+1;
      if (contador < 140000) then
        pulso <= '1';
      else
        pulso <= '0';
      end if;
    end if;
  end process TX_divisor;
```

Figura 8.5. Proceso Tx_divisor del sistema transmisor serial

La figura 8.6 muestra el proceso "Tx_prepara", en él se implementa al bloque "Prepara comunicación". Este proceso se encarga de generar un arreglo que contiene 2 datos a transmitir en formato ASCII. Por default se envía el carácter '0', seguido de un salto de línea.

```
TX_prepara : process(reloj, pulso)
  type arreglo is array (0 to 1) of STD_LOGIC_VECTOR(7 downto 0);
  variable asc_dato : arreglo := (X"30",X"0A");
begin
  asc_dato(0):=hex_val;
  if (pulso='1') then
    if rising_edge(reloj) then
      if (conta=valor) then
        conta <= 0;
        INICIO <= '1';
        Dato <= asc_dato(i)
        if (i = 1) then
          i <= 0;
        else
          i <= i + 1;
        end if;
      else
        conta <= conta+1;
        inicio <= '0';
      end if;
    end if;
  end if;
end process TX_prepara;
```

Figura 8.6. Proceso TX_prepara del sistema transmisor serial

La figura 8.7 presenta el código del proceso "TX_envia", correspondiente a la descripción del bloque funcional "Envía dato". Dicho proceso es el encargado de generar la velocidad de transmisión "*Baudrate*" y colocar los datos previamente preparados para ser enviados a través de la línea de transmisión.

```
TX_envia : process(reloj,inicio,dato)
begin
   if(reloj'EVENT and reloj = '1') then
      if(Flag = '0' and INICIO = '1') then
         Flag<= '1';
         BUFF(0) <= '0';
         BUFF(9) <= '1';
         BUFF(8 DOWNTO 1) <= dato;
      end if;
      if(Flag = '1') then
         if(PRE < PRE_val) then
            PRE <= PRE + 1;
         else
            PRE<= 0;
         end if;
         if(PRE = PRE_val/2) then
            TX_WIRE <= BUFF(INDICE);
            if(INDICE < 9) then
               INDICE <= INDICE + 1;
            else
               Flag <= '0';
               INDICE <= 0;
            end if;
         end if;
      end if;
   end if;
end process TX_envia;
```

Figura 8.7. Proceso TX_envia del sistema transmisor serial

Finalmente, la figura 8.8 muestra la última parte de la arquitectura del sistema transmisor serial, en donde se realiza la lectura y decodificación del valor binario leído en el switch, para su correspondiente transformación al código ASCII que será transmitido. Así mismo, se presenta la selección de la velocidad de transmisión mediante la señal "baud" dentro de una lista sensible.

```
    LED <= pulso;
    dato_bin<=SW;
    baud<="011";

    with(dato_bin) select
       hex_val <= X"30" when "0000",
                  X"31" when "0001",
                  X"32" when "0010",
                  X"33" when "0011",
                  X"34" when "0100",
                  X"35" when "0101",
                  X"36" when "0110",
                  X"37" when "0111",
                  X"38" when "1000",
                  X"39" when "1001",
                  X"41" when "1010",
                  X"42" when "1011",
                  X"43" when "1100",
                  X"44" when "1101",
                  X"45" when "1110",
                  X"46" when "1111",
                  X"23" when others;

    with (baud) select
       PRE_val <= 41600 when "000",   --  1200 bauds
                  20800 when "001",   --  2400 bauds
                  10400 when "010",   --  4800 bauds
                   5200 when "011",   --  9600 bauds
                   2600 when "100",   -- 19200 bauds
                   1300 when "101",   -- 38400 bauds
                    866 when "110",   -- 57600 bauds
                    432 when others;  --115200 bauds

end architecture behaivoral;
```

Figura 8.8. Código para manipulación de periféricos y selector de velocidad dentro de la arquitectura del sistema transmisor serial

ACTIVIDAD COMPLEMENTARIA:

El alumno diseñará un sistema capaz de enviar el valor del switch en forma binaria, es decir cuatro caracteres, uno por bit leído. La forma en que la secuencia de texto que deberá ser visualizado en la computadora es: **Valor binario=XXXX,** donde XXXX representa el número de 4 bits.