TEC | Tecnológico de Costa Rica

Curso: Taller de Diseño Digital

Laboratorio 3: Cuestionario Previo

PROFESOR

Roberto Carlos Molina Robles

Grupo #1

ESTUDIANTES
Oscar David Conejo Cantón 2020234423
Katherine Daniela Salazar Martínez 2014160591
Juan Pablo Solano Solano 2020425831

I semestre 2025

Cuestionario Previo:

1. Investigue sobre el concepto de FIFO (First-In, First-Out). Describa el funcionamiento de estas unidades, las señales básicas que se espera tener y mencione al menos tres usos de estos bloques.

- El concepto de FIFO se refiere a un tipo de memoria o buffer que almacena datos en el orden en que llegan: el primer dato en entrar es el primero en salir. Normalmente se utiliza la analogía de una fila de personas: el que llega primero, se atiende primero. Esto representa correctamente el principio de la memoria.
- Una FIFO se comporta como una cola circular con dos punteros: Write Pointer (puntero de escritura): indica la posición donde se escribirá el siguiente dato y Read Pointer (puntero de lectura): indica la posición de donde se leerá el siguiente dato. Cada vez que se escribe un dato se avanza el puntero de escritura. Y cada vez que se lee un dato el puntero de lectura avanza. Y se hace una comparación entre los punteros para determinar si la FIFO está vacía o llena.
- Usos de estos bloques:
    - Comunicación entre módulos a distinta frecuencia
    - Interfaces de comunicación
    - Procesamiento de señales o datos en paralelo

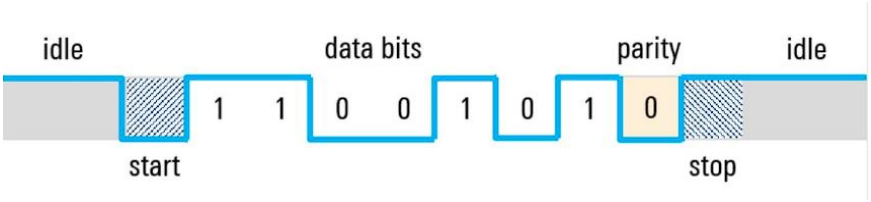
2. Investigue sobre la comunicación serie UART. Preste atención a las diferentes características de configuración necesarias para la comunicación serie mediante UART (por ejemplo, baud rate, paridad, etc). Además, investigue cómo puede utilizar puertos serie en su computadora, considerando el sistema operativo que utilice. En Windows se recomienda utilizar RealTerm.

UART es el nombre que se le da a (universal asynchronous receiver / transmitter) por sus siglas en inglés. Según Rohde “define un protocolo o un conjunto de normas para el intercambio de datos en serie entre dos dispositivos.” (2025). La comunicación UART en serie consiste en líneas como TX (transmit) y RX (receive) y sigue sigue utilizada para aplicaciones de baja velocidad y bajo rendimiento, puesto que es muy simple, económico y fácil de integrar.

Una de las mayores ventajas de UART es que es asíncrono: el transmisor y el receptor no comparten la misma señal de reloj. Si bien esto simplifica en gran medida el protocolo, plantea determinados requisitos en el transmisor y el receptor. Puesto que no

comparten un reloj, ambos extremos deben transmitir a la misma velocidad, previamente concertada, con el fin de mantener la misma temporización de los bits. Las velocidades en baudios más habituales en UART que se utilizan actualmente son 4800, 9600, 19,2 K, 57,6 K, y 115,2 K. Además de tener la misma velocidad en baudios, ambos extremos de una conexión UART deben utilizar también la misma estructura y parámetros de trama. La forma más sencilla de entender esto es observando una trama UART. (Rohde & Schwarz, 2025)

Las señales de recepción de la UART serie se puede apreciar a continuación



Formato de trama en UART

Donde cada uno de estos bits tiene un distinto propósito

Bits de Inicio y parada

Corresponden al bit de inicio que es una transición del estado de reposo alto a un estado bajo. Una vez que finalizan los bits de datos, el bit de parada indica el fin de la carga útil. El bit de parada es una transición de retorno al estado alto o de reposo, o bien la permanencia en el estado alto por un tiempo de bit adicional.

Bits de datos

Corresponde al largo de la información del dato que se quiere procesar, esta información puede tener un largo de 5 a 9 bits de carga útil, pero lo más habitual son 7 u 8 bits

Bit de paridad

Bit final altamente utilizado para detectar errores, la paridad puede ser par o impar

- **Paridad par**: El bit se ajusta de tal modo que el número total de unos en la trama será par.
- **Paridad impar**: El bit se ajusta de tal modo que el número total de unos en la trama será impar.

Baud rate

Corresponde al número de señales o símbolos por segundo que se transmiten por el canal UART. En la mayoría de las implementaciones UART cada bit es un símbolo, por lo tanto. Por lo que la cantidad de símbolos es igual al BaudRate

Además de lo antes mencionado, si se quisiera Interactuar con los puertos de Windows actuales. Para poder interactuar con una computadora moderna es necesario un convertir de entrada de tipo UART a una USB. Una vez conectada se requiere de un software para la interpretación de datos, como RealTerm, donde se selecciona la entrada del puerto que se esté utilizando, se define la velocidad y se recibe la información enviada.

3. Investigue cómo instanciar y usar bloques escritos en VHDL en sistemas escritos mayoritariamente en SystemVerilog. Muestre una comparación de la definición de los módulos (o bloques) en VHDL y SystemVerilog. Haga énfasis en la definición de los puertos de entrada y salida en VHDL.

Definir un módulo en VHDL es bastante más complicado que en SV, requiere definir de una manera más extensa su arquitectura y su comportamiento, mientras que un módulo principal en VHDL se puede definir de la siguiente manera

```
entity Contador is
  Port (
    clk    : in  STD_LOGIC;
    rst    : in  STD_LOGIC;
    enable : in  STD_LOGIC;
    count  : out STD_LOGIC_VECTOR(7 downto 0)
  );
.
.
.
```

El mismo módulo en SV se puede definir de la siguiente manera

```
module top (
  input  logic clk,
  input  logic rst,
  input  logic enable,
  output logic [7:0] count
```

);

end contador;

Debido a su simpleza de definición, es razonable realizar instancias en SV sobre VHDL

Para instanciar un modulo VHDL en sistemas escritos en sistemas escritos mayoritariamente en SystemVerilog se debe tener en cuenta principalmente la herramienta de sintetizado que se esté utilizando. Una vez esto se haya confirmado, la manera de instanciarlo es muy parecido a una instancia de un módulo normal dentro de SystemVerilog. Como se hace a continuación

```
// Instancia del módulo VHDL
contador u_contador (
  .clk    (clk),
  .rst    (rst),
  .enable (enable),
  .count  (count)
);
```

Endmodule

Es frecuentemente usado este tipo de instancias para la implementación de FIFOS u otro tipo de maquinas dentro de entornos de simulación.

Bibliografía

Rohde & Schwarz. (2025). *¿Qué es UART?* Recuperado de Rohde & Schwarz Sitio web: sección “Entendiendo el UART”