Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Cuestionario Previo 3

Taller de diseño digital

Integrantes:

Arias Ortiz Kendy Raquel

Pérez Jiménez Jorge

Fabián M. Villegas Bonilla

Adrián Parajeles Alvarado

Profesor:

Dr.-Ing. Roberto Molina Robles

13 de mayo de 2025

Lab. 3: Interfaces con periféricos

Preguntas

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 FIFO es una estructura de almacenamiento temporal que indica que lo primero que entra es lo primero en salir, este pasa por un proceso donde los datos entran, se mantienen en orden y se van procesando conforme entran para salir los primero datos. Este funcionamiento se puede resumir en permitir la entrada (able_write) y guardarla (data_in) en una base de datos creada en el sistema con su respectivo tamaños y posición (rd_data). Ya guardado buscar el dato más viejo (read_data) en el tiempo que se requiere y sacarlo (data_out) además de borrarlo de los datos (borrar) y continuar con estos procesos.

   Un primer uso común es en interfaces de comunicación serial, como UART o SPI, donde la FIFO permite almacenar temporalmente los datos recibidos o por transmitir, evitando pérdidas de información si la CPU no atiende el periférico de inmediato. Un segundo uso se da en transferencias por DMA (Acceso Directo a Memoria), donde las FIFOs permiten desacoplar los tiempos de lectura y escritura entre el periférico y la memoria. Un tercer uso es el sistema de procesamiento de señales.

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.**

   La comunicación en serie UART es un protocolo de transmisión de datos entre dos dispositivos, uno envía y el otro recibe sin estar sincronizados. A pesar de no necesitar sincronización para que funcione el sistema ambos dispositivos deben configurarse con el mismo protocolo de comunicación en serie UART.

   Entre las características más importantes se encuentran el baud rate, que define la velocidad de transmisión por ejemplo de 9600 o 115200 bps, el número de bits de datos, la configuración de paridad, la cual ayuda a detectar errores simples en la transmisión, el número de bits de parada, que indican el final de un byte de datos; y, en algunos casos, los bits de control de flujo (como RTS/CTS), que ayudan a regular el envío y recepción para evitar desbordamientos de datos

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.**

   La integración de bloques escritos en VHDL Very High Speed Integrated Circuits) Hardware Description Language. Esto significa que VHDL permite acelerar el proceso de diseño. Dentro de un entorno mayoritariamente SystemVerilog se basa en la capacidad de los flujos de simulación y síntesis para manejar ambos lenguajes de manera mixta. En la práctica, cada archivo VHDL se compila primero en la librería de trabajo (por ejemplo, con vcom en ModelSim o Vivado) y luego los

archivos SystemVerilog se compilan apuntando a esa misma librería. De esta forma, el simulador reconoce las entidades VHDL como componentes disponibles para instanciar directamente desde un módulo SV, sin necesidad de convertir manualmente cada señal ni escribir “wrappers” adicionales.

La definición de puertos en VHDL difiere de la de SystemVerilog en varios aspectos. En VHDL, los puertos se agrupan en la cláusula port de la entidad, donde tras listar los nombres de señal se indica la dirección (in, out, inout o buffer) y el tipo (std_logic, std_logic_vector, etc.). Además, el VHDL soportan todo tipo de parámetros, que permiten parametrizar anchuras de buses o comportamientos, de forma análoga a los parameter o localparam en SV. En cambio, en SystemVerilog cada puerto se declara de manera individual dentro del módulo, precedido por su tipo (logic, wire) y su dirección (input, output, inout), lo que hace que la definición sea ligeramente más compacta y flexible para declaraciones ad-hoc.

Referencias

1. **Harris, D. M., & Harris, S. L.** *Digital Design and Computer Architecture: ARM Edition* (2nd Edition). Morgan Kaufmann, 2016.

2. Harvie, L. (2025, 21 febrero). *Best Practices for Managing Multiple Clock Domains in FPGA Designs - RunTime Recruitmen*t. RunTime Recruitment. https://runtimerec.com/best-practices-for-managing-multiple-clock-domains-in-fpga-designs/

3. Russell. (2022, 30 junio). *What is a FIFO in an FPGA?* Nandland. https://nandland-com.translate.goog/lesson-8-what-is-a-fifo/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

4. Peňa, E., & Legaspi, M. G. (s. f.). *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter*. Analog Devices. https://www.analog.com/en/resources/analog-dialogue/articles/uart-a-hardware-communication-protocol.html

5. Marcos, S. M. (2014b). *Introducción a la programación en VHDL*. https://docta.ucm.es/entities/publication/3152080b-11fa-466c-88c7-61ed262828a