

Planeación de Protocolo de envío y recepción de datos UART.

## **Plan de Verificación y Validación**

### **Versión 1.0**

## **Contenido**

- I.   Objetivos
- II.  Introducción
- III. Alcance
- IV.  Áreas Funcionales
- V.   Elementos de Verificación
- VI.  Conclusión

## Objetivo

El objetivo de esta verificación es la de corroborar el correcto funcionamiento de la implementación en Lenguaje Descriptivo de Hardware Verilog del protocolo de comunicación UART para envío de datos a través de un puerto serial estándar.

## Introducción

La comunicación serial consiste en el envío secuencial de un bit a de información entre dos o más dispositivos, con un correcto programa para la precisión en la recepción y transmisión del dato y de esta manera tener una lectura limpia.

El bit de inicio es un 0 lógico, y el de alto es un 1. El protocolo se repite para cualquier valor enviado, y debe ser especificado en la programación interna de ambos módulos, puesto que la correcta comunicación entre ellos es el funcionamiento primario del módulo principal.

Debemos tener en cuenta la velocidad a la que se envían los bits serialmente, es de suma importancia. Ambos dispositivos deben estar a la misma velocidad para transmitir y recibir bits, si no lo están, simplemente la transmisión serial no sería correcta.

De esta forma se pueden enviar cualquier tipo de información: letras, números y caracteres. Se puede enviar el código ASCII de una letra seguida de otra y así sucesivamente para formar palabras y valores numéricos.

Una vez el protocolo pase los diferentes niveles de verificación, el módulo tendrá la posibilidad de ser entregado al cliente.

El objetivo de la verificación es poder encontrar la mayor cantidad de defectos tempranamente con el objetivo de entregar un software de calidad al cliente, y evitando las cuestiones de rediseño ocasionadas por el descubrimiento de fallas sobre el final del proceso de desarrollo.

Para lograr esto, se requiere tener tanto una visión global del sistema, como visión en cada uno de los subsistemas que lo componen, y también de cada unidad de código fuente que componen los subsistemas. Por ese motivo la verificación se separa en 3 diferentes pruebas:

- Pruebas unitarias: Se verifica cada unidad de código fuente, para determinar si contiene defectos en su implementación, comprobándolo contra el diseño del mismo.
- Pruebas de integración: Se verifica el correcto funcionamiento de las interacciones entre las unidades de código fuente para determinar que no existen defectos en la comunicación de los mismos.
- Pruebas funcionales: Se verifica que el sistema cumple con todas las funcionalidades especificadas en los requerimientos del mismo

## Alcance

Para definir el alcance del proceso de verificación y validación es necesario especificar las responsabilidades que deben asumir los integrantes del mismo.

La verificación unitaria será responsabilidad del programador o implementador que le sea asignada, y consistirá en ir revisando de manera cautelosa cada unidad terminada o avanzada, así como crear un programa de verificación **test bench** que pueda evaluar el modulo en un conjunto de casos estipulados.

Las pruebas funcionales consistirán en pruebas basadas enteramente en el documento de especificaciones de requerimientos, para de esta manera comprobar que el sistema cumple con ellas.

## Áreas funcionales

El sistema de comunicación está conformado por 3 elementos:

- **Transmisor serial:** que se encarga de manipular las variables internas del mismo y con ello llevar a cabo la transmisión ordenada después de un bit de inicio y finalizar la misma con un bit de alto.
- **Receptor serial:** Se encarga de recibir los bits tanto de inicio y de paro, como los de datos, así como la manipulación de los datos para ser presentados al usuario una vez la transmisión finalice.
- **Modulo Top:** Es el modulo principal, y agrupa a los módulos de receptor serial y transmisor serial para llevar a cabo el protocolo de comunicación de manera simplificada para el usuario, y presentar los datos en un sistema de entradas y salidas mínimas.

La comunicación serial entre el transmisor y receptor requiere de 4 señales, con 2 conexiones entre ellos:

- **TxD**
- **RxD**
- **Ready**
- **Clr\_ready**

El primero para la transmisión de datos, el segundo para la recepción de los mismos, y el tercero y cuarto indican cuando un proceso de envío o recepción se completa. Los datos son transferidos a una velocidad de transmisión predefinida, en bits por segundo y conocida técnicamente como **baud rate**. Los valores más comunes son 4800, 9600, 56000 y 115200.

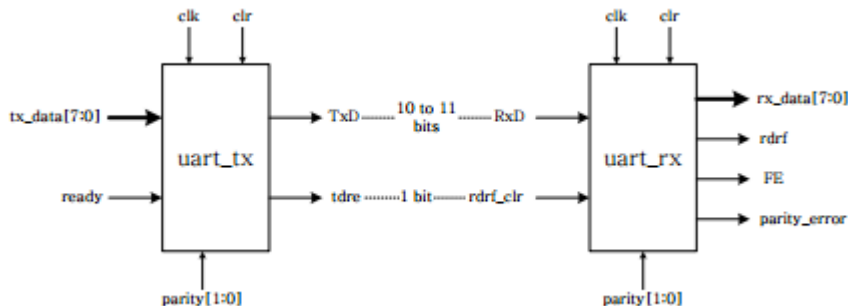
Además del baud rate, también deben ser predefinidos los siguientes valores dentro del módulo UART:

- **Bit de inicio**
- **Bits de datos**
- **Paridad**
- **Bits de alto**

Es posible enviar cualquier dato en forma binaria de manera serial, siempre que exista un protocolo de transmisión y recepción finamente acoplado y funcionando con exactitud en los tiempos de envío y lectura entre ambos dispositivos.

La transmisión está sincronizada a un bit de inicio, al que le procede la transmisión de datos, finalizando la misma con un bit de alto indicando que la transmisión ha sido completada.

El funcionamiento de los módulos los obliga a comunicarse cuando empiezan y terminan eventos, por lo que la conexión **ready** indica al módulo transmisor cuando empezar a enviar los bits que conforman el protocolo, así como ejecutar el reset para la salida del receptor indicando que los datos han sido entregados con éxito.



## Elementos de verificación

Cada submódulo deberá ser probado, por lo que pasará por las siguientes etapas para poder ser integrado al proyecto final:

- Diseño del comportamiento del módulo, que debe cumplir con los requerimientos estipulados en el documento de especificaciones y optimizar los distintos bloques de comportamiento.
- Implementación del comportamiento en lenguaje descriptivo de hardware Verilog, utilizando el programa Questa Sim que contiene el paquete de herramientas necesarias para la codificación, simulación, verificación, monitoreo y almacenamiento de resultados, así como una interfaz clara del comportamiento del mismo en relación al tiempo.
- Definir los elementos del módulo a ser verificados: entradas, salidas, variables, valores y parámetros.
- Codificación del test bench, teniendo en cuenta las conexiones del submódulo así como el monitoreo de los datos entregados, incluyendo diferentes casos de comportamiento para comprobar que todo funcione correctamente, desde los cambios en los valores de las variables para obtener diferentes resultados acordes a los distintos casos a lo que se

someterá el modulo, hasta la simulación digital del **clk** que marcará los tiempos en el protocolo

- Interpretación de resultados, obtenidos por medio de simulaciones digitales al submodulo externo test bench. De igual manera con el software Questa Sim se llevará a cabo la visualización de los pulsos de cambio en las salidas de los módulos, así como los valores para presentar los resultados esperados.

El proceso completo se llevara a cabo en una computadora portátil, con los requerimientos recomendados para la ejecución del programa Questa Sim, así como todas las herramientas que componen la suite del simulador.

## Conclusión

Es necesario tomar en cuenta muchas cosas antes del diseño, y tener un plan que agrupe todos los aspectos que deben ser cubiertos, es indispensable. Aunque el diseño es la parte que encara al usuario, atrás de este se encuentra un programa que reúne todos los objetivos de las actividades de verificación y validación por la cual pasó antes de llegar a ser utilizado. Algunas de estas son valorar y mejorar la calidad de los productos del trabajo generados durante el desarrollo y modificación del software. Los atributos de la calidad deben ser la corrección, la perfección, la consistencia, la confiabilidad, la utilidad, la eficacia, el apego a los estándares y la eficacia de los costos totales, y de esta manera tener un buen producto como resultado.