

Parte 2

Diseño de un receptor de transacciones MDIO

1. Diseñar un receptor de transacciones MDIO de acuerdo con las especificaciones estipuladas en la cláusula 22 del estándar IEEE 802.3 (disponible en la página de Mediación Virtual del curso), particularmente las secciones 22.2.2.13 y 22.2.2.14.

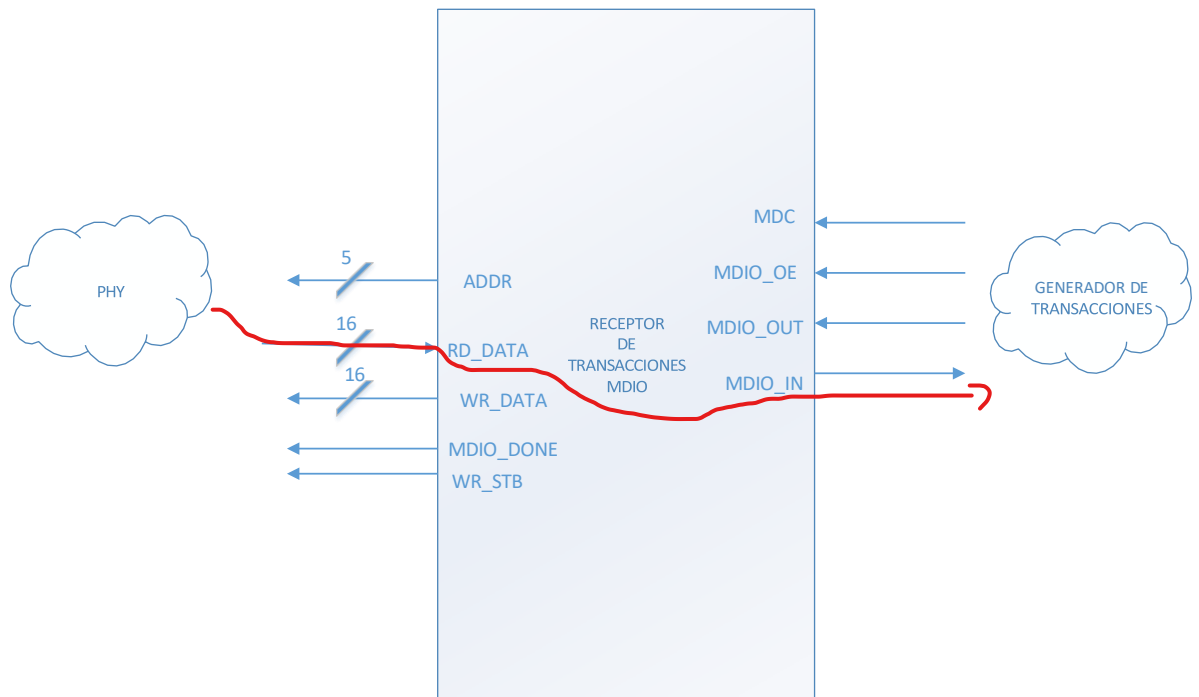


Figura #1: Receptor de transacciones MDIO

Interfaces del receptor:

- Tester** a) **MDC** – Entrada de reloj para el MDIO. El flanco activo de la señal **MDC** es el flanco creciente. Esta entrada debe provenir de un generador de MDIO, o al menos modelar su comportamiento.
- b) **RESET (no se muestra en la figura)** – Entrada de reinicio del generador. Si **RESET=1** el generador funciona normalmente. En caso contrario, el generador vuelve a su estado inicial y todas las salidas toman el valor de cero.
- DUT** c) **MDIO_DONE** – Strobe (pulso de un ciclo de reloj). Salida que indica que se ha completado una transacción de MDIO en el receptor.
- Tester** d) **MDIO_OUT** – Entrada serial. Esta entrada debe provenir de un generador de MDIO, o al menos modelar su comportamiento.
- Tester** e) **MDIO_OE** – Habilitación de **MDIO_OUT**. Esta entrada debe detectar si el valor de **MDIO_OUT** que se está recibiendo es un valor válido habilitado. En una transacción de escritura, debe permanecer en alto durante los 32 ciclos de la transacción, pero ponerse en bajo al terminar la transacción. En una transacción de lectura, debe

permanecer en alto durante los primeros 16 ciclos de la transacción, pero debe ponerse en cero durante los siguientes 16 ciclos, mientras el receptor envía el dato de MDIO_IN. Al final de la transacción de lectura, se espera que esta entrada debe permanecer en cero.

- f) **MDIO_IN** – Salida serie. MDIO_OE 16 en alto y 16 en bajo Durante una operación de lectura (de acuerdo a la cláusula 22 del estándar), se debe enviar a través de esta salida, el dato almacenado en la posición REGADDR, durante los últimos 16 ciclos de la transacción de MDIO.
- g) **ADDR[4:0]** – Salida de dirección. Indica en qué posición de memoria se debe almacenar el dato que se recibe en WR_DATA, o desde cuál posición se debe leer RD_DATA.
- h) **WR_DATA[15:0]** – Salida de datos. Los datos que se presentan en esta salida se escriben en la posición de memoria indicada por ADDR en el ciclo de reloj donde MDIO_DONE = 1 y WR_STB = 1.
- i) **RD_DATA[15:0]** – Entrada de datos. Contiene el valor leído desde la memoria, a más tardar dos ciclos de MDC después de que se cumple que MDIO_DONE = 1 y WR_STB = 0.
- j) **WR_STB** – Esta salida se pone en 1 para indicar que los datos de WR_DATA y WR_ADDR son válidos y deben ser escritos a la memoria.

De acuerdo a la cláusula 22 del estándar IEEE 802.3, una transacción básica de MDIO es una transacción serial de 32 bits, cuyos campos se definen como se muestra en la figura #2 (figura #13 en la referencia):

Clause 22

Clause 22 defines the MDIO communication basic frame format (figure 13) which is composed of the following elements:

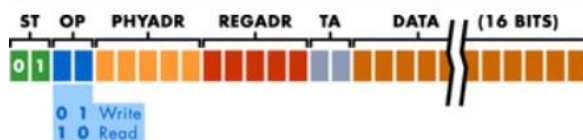


Figure 13: Basic MDIO Frame Format

Figura #2: Transacción de MDIO, tomada de [este sitio web](#).

2. Escribir una descripción conductual del receptor de transacciones de MDIO usando Verilog. Esta descripción servirá como una especificación detallada y formal del funcionamiento del dispositivo diseñado.
3. La descripción en Verilog deberá tener al menos un módulo de banco de pruebas, un módulo probador, y un módulo con la descripción del generador. Use Icarus Verilog.
4. Definir un plan de pruebas para garantizar el funcionamiento del diseño. El plan de pruebas debe cubrir todos los modos de operación del receptor, es decir, transacciones de lectura, de escritura y carga de memoria. El módulo probador debe suministrar las señales necesarias para que las pruebas se realicen.

Rúbrica de Calificación

Documentación y afiche	20 puntos
Receptor de transacciones	40 puntos
Generador de transacciones	40 puntos

Guía para el reporte

Se debe entregar en forma electrónica un documento, a lo sumo de 10 páginas de longitud, que incluya los siguientes puntos:

1. **Resumen:** Breve (1 párrafo) descripción de **todo** el proyecto. Esta sección es fundamental pues puede determinar si el lector se interesa o no en leer los detalles del proyecto. Un resumen mal hecho puede esconder un excelente proyecto. El resumen debería incluir:
 - a) Descripción breve del sistema, es decir, qué hace. Incluya alguna característica que considere que distingue este diseño en particular.
 - b) Las pruebas que se realizaron y qué resultados se obtuvieron. Indique problemas que se tuvieron que considere importante resaltar.
 - c) Conclusiones más importantes y recomendaciones para un diseño posterior.
2. **Descripción Arquitectónica:** Incluye un diagrama de bloques con las señales más importantes que sirve como base para describir el funcionamiento del sistema. La descripción va en términos de lo que se espera que el sistema haga. Es decir, se debe detallar la funcionalidad del sistema, el protocolo de las señales que se usan para que funcionen cada una de las partes y las secuencias de eventos que se deben dar. Esta descripción podría ir acompañada de tablas de verdad, tablas de transición de estados, diagramas de estados, diagramas temporales, etc.
3. **Plan de Pruebas:** Aquí se deben enumerar, esto es, se debe presentar una **lista detallada** de las pruebas que se le van a hacer al diseño para verificar que está funcionando de acuerdo a las especificaciones dadas. La lista debe contener por lo menos los siguientes elementos i) Nombre/número de prueba, ii) Descripción de la prueba, y iii) Una indicación de si el diseño la falló o la pasó. Estas pruebas podrían incluir la generación de vectores de entrada para probar en forma exhaustiva todas las líneas de una tabla de verdad o tabla de estados, patrones aleatorios de entradas para tratar de causar errores en la respuesta del diseño, o patrones específicos que ejerciten un cierto modo de funcionamiento. Cada prueba debería ser claramente enumerada en el plan para que también se pueda hacer referencia a ella en el código del banco de pruebas del diseño.
4. **Instrucciones de utilización de la simulación:** Esta sección debe mostrar los comandos necesarios para hacer funcionar la simulación en todos los casos que especifica el plan de pruebas. Hay que suponer que el diseño de un grupo puede ser utilizado por otro grupo o el profesor. Si los resultados no se pueden repetir porque no se conocen los comandos para hacer funcionar la simulación entonces es como si el diseño no funcionara del todo. Se recomienda crear un Makefile de modo que se pueda correr todas las pruebas del caso con un solo comando en Icarus Verilog y GTKwave.
5. **Ejemplos de los resultados:** Una descripción de los resultados más importantes acompañados de los diagramas temporales de la simulación (GTKWave) o cualquier otra

salida que demuestre claramente el comportamiento descrito. No es necesario incluir una muestra exhaustiva de resultados, sino que los más representativos del diseño. El punto es mostrarle al lector los comportamientos más sobresalientes para formarle una idea clara

del funcionamiento del diseño. Ya verá el lector si desea más detalles, entonces podrá correr una simulación.

6. **Conclusiones y recomendaciones:** Basado en los resultados obtenidos se indica aquí qué se logró con el proyecto. Puede ser que se concluya que con el diseño propuesto se tiene una limitación en la velocidad de respuesta de... etc. O que con ciertas combinaciones de entradas el diseño se vuelve inestable o los resultados no son los esperados. También se puede concluir qué ventajas o problemas encontraron al seguir el plan de trabajo. A raíz de las conclusiones se puede también recomendar cómo se podría mejorar el diseño o qué otras pruebas se le podrían hacer para garantizar su funcionamiento en otras condiciones que al principio no se consideraron, o también cómo se debería planear el siguiente proyecto para poder cumplirlo a tiempo.