

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0523 Circuitos Digitales II

II ciclo 2022

Tarea 3

## Diseño de un generador de transacciones MDIO

Javier Solera Bolaños B66963

Grupo 1

Profesor: Enrique Coen Alfaro

11/10/2022

# Índice

1. Resumen	1
2. Descripción Arquitectónica	1
3. Plan de Pruebas	2
4. Instrucciones de utilización de la simulación	3
5. Ejemplos de los resultados	3
6. Conclusiones y recomendaciones	6

# 1. Resumen

En este trabajo se realizo una maquina de estados con el protocolo MDIO, el cual puede realizar una lectura o una escritura. En esta tarea se implemento el convertidor de datos de paralelo a serie para la escritura y para la lectura, el convertidor de datos de serie a paralelo. Entonces el programa recibe una palabra de 32 bits, entonces dependiendo del ST (ST es de 2 bits), esta va a realizar la transacción o no, ya que el protocolo que se implemento es para que la transacción se realice siempre que  $ST = 01$ , en caso contrario, no se realizará ninguna transacción. El OPCODE, me define que tipo de transacción es la que se va a realizar, entonces OPCODE es de 2 bits, el generador va a convertir en datos seriales la palabra que esta en  $T\_DATA$ , independientemente si es lectura o escritura, y lo recibe de manera serial el  $MDIO\_OUT$ . El  $MDIO\_OE$  este se va a poner el alto los 32 ciclos del reloj MDC de la transacción, en caso de que sea una transacción de escritura, si es de lectura, se pone en alto solo 16 ciclos del reloj MDC.

También el programa va a convertir de serie a paralelo los datos de  $MDIO\_IN$  a  $RD\_DATA$ , esta es solo cuando se esta realizando la transacción de lectura. Cuando la transacción esta lista, se pone le alto  $DATA\_RDY$  para indicar el la transacción esta lista. Todas estas transacciones se sincronizan con el reloj MDC y tiene una frecuencia que es la mitad de la frecuencia de CLK.

## 2. Descripción Arquitectónica

EL Generador de transacciones que se diseñó, es el siguiente:

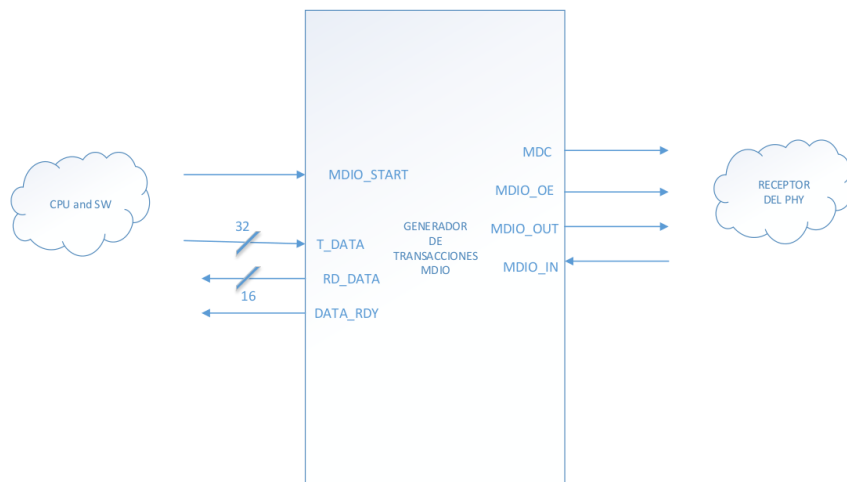


Figura 1: Generador de Transacciones MDIO

La clausula en la que se basó la maquina de estados es la siguiente:

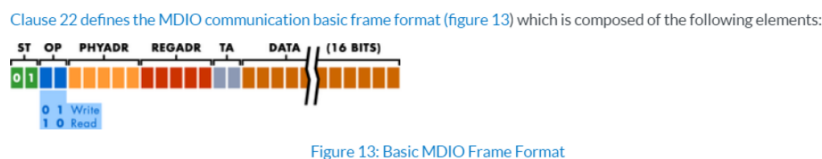


Figura 2: Clausula 22 de IEEE 802.3

Para esta tarea, se realizo la siguiente maquina de estados para representar el protocolo de la clausula 22 de IEEE 802.3, el MDIO:

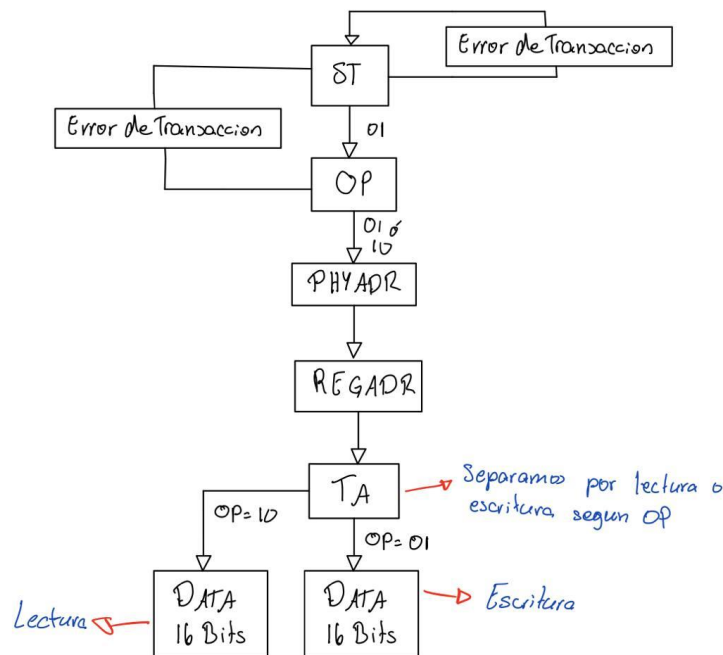


Figura 3: Maquina de Estados para el MDIO

En este diseño, se espera que la maquina de estados, al recibir una palabra de 32 bits y ir preguntando dependiendo de los estados, es decir, preguntar si el estado de ST, que seria los MSB, es igual a 01, de ser correcto continua con el siguiente estado, y de no ser el caso, vuelve al estado inicial. Luego pregunta por el estado OP, que serian los siguientes 2 bits, el cual puede tener valores de 01 o 10, dependiendo de si es escritura o lectura, si es el caso, pasaría al siguiente estado, de no ser así, se devuelve al estado inicial. Luego pasa a los estados PHYADR y REGADR, que para este caso, no se tiene que preguntar nada. Una vez que llega al estado TA, que serian los bits 14 y 15, se hace aquí la separación de si es escritura o lectura, entonces si es de escritura llegara al estado de escritura y lo mismo para la lectura.

### 3. Plan de Pruebas

Para esta tarea, se realizaron 3 pruebas importantes:

1. **Prueba para el RESET:** En esta prueba se trata de implementar que mientras el reset esta en bajo, las señales RD\_DATA, MDIO\_OE, MDIO\_OUT y DATA\_RDY se ponen en bajo, entonces la maquina de estados no funcionaria.
2. Adicionalmente se realizo una prueba, que es cuando el **ST es distinto de 01**, la cual muestra no se realiza ninguna transacción, esto porque los dos MSB de T\_DATA no coinciden con lo especificado por el estándar, es decir, ST es distinto a 01, lo que provoca que cuando se manda un pulso del MDIO\_ no sucede ninguna transacción.
3. **Prueba para la Transacción de Lectura:** Esta prueba se recibe una palabra de 32 bits, que en este caso seria 699A5D6F, entonces, esta palabra llega de forma paralela, y el MDIO\_OUT, va a recibir bit por bit de forma serial, de manera que cada vez que haya

un 1, el MDIO\_OUT, se pone en alto y cuando es un 0 MDIO\_OUT se pone en bajo. Para el MDIO\_OE, este se va a poner en alto toda la transaccion de lectura, es decir, una vez que se detecta que es escritura, el MDIO\_ se pone en alto por 16 ciclos de reloj del MDC. Despues el DATA\_RDY se pone en alto cuando se ha completado la recepci3n de una palabra serial y esta activa el RD\_DATA, de manera que se va a poner los valores desde PHYADR hasta el final de la palabra.

4. **Prueba para la Transacci3n de Escritura:** Esta prueba se recibe una palabra de 32 bits, que en este caso seria 599A5D6F, entonces, esta palabra llega de forma paralela, y el MDIO\_OUT, va a recibir bit por bit de forma serial, de manera que cada vez que haya un 1, el MDIO\_OUT, se pone en alto y cuando es un 0 MDIO\_OUT se pone en bajo. Para el MDIO\_OE, este se va a poner en alto toda la transaccion de escritura, es decir, una vez que se detecta que es escritura, el MDIO\_ se pone en alto por 32 ciclos de reloj del MDC.

## 4. Instrucciones de utilizaci3n de la simulaci3n

Es importante que los archivos Maquina.Estado.v, testbench.v, teste.v y el Makefile se encuentren en la misma carpeta.

Entonces puede abrir la terminal y ubicarse en donde est3n los archivos mencionados anteriormente y correr el comando make, este ejecutar3 el archivo Makefile y compilara y correr3 los archivos.

Tambi3n puede seguir las instrucciones en la terminal son: iverilog -o tb.vvp testbench.v, esta linea lo que hace es compilar los archivos. Luego tienen que escribir en la terminal: vvp tb.vvp, esto lo que hace es correr la simulaci3n. Y por ultimo, se escribe en la terminal: gtkwave tb.vcd, esta lo que hace es abrir el contador en la aplicaci3n gtkwave.

## 5. Ejemplos de los resultados

Prueba para RESET en bajo:

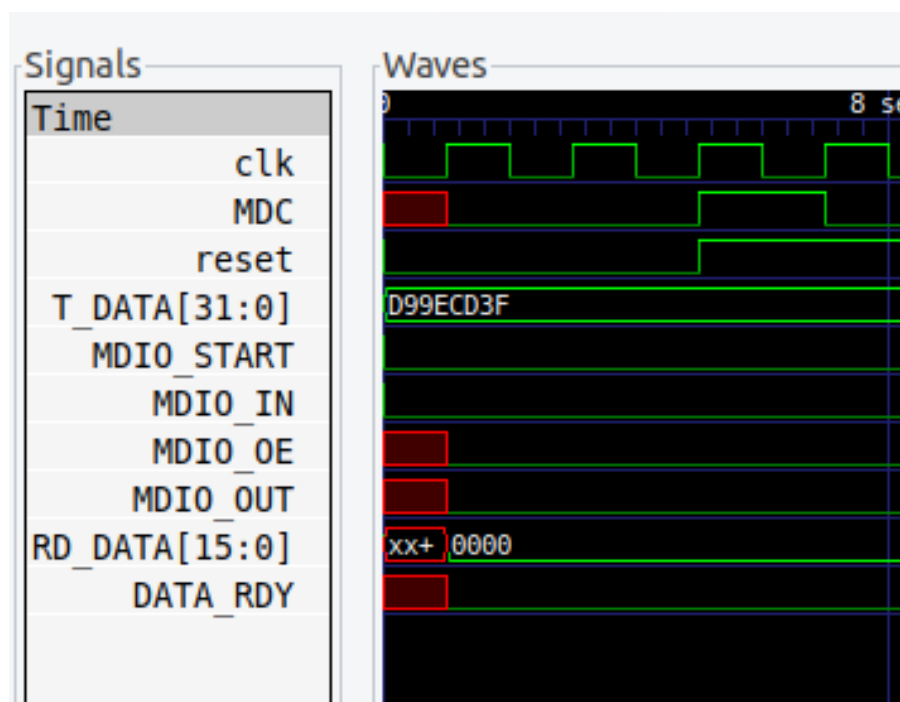


Figura 4: Prueba con el RESET en bajo

Prueba adicional, ST distinto de 01:

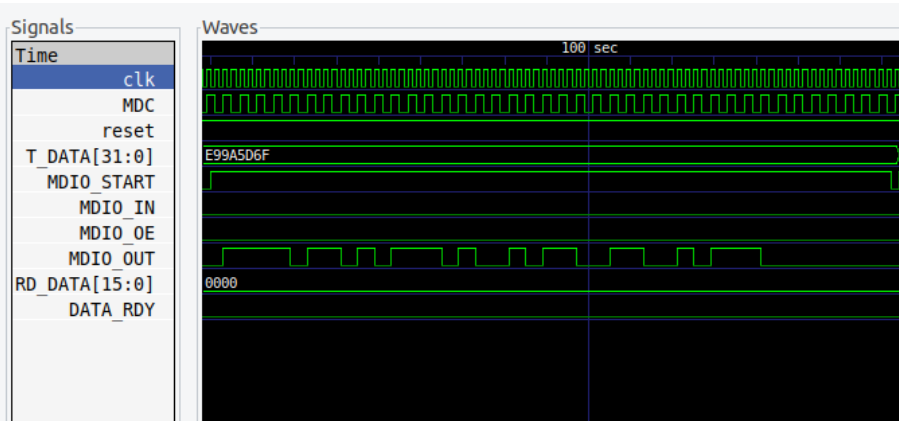


Figura 5: Transacción con ST distinto de 01

Prueba para la transacción de escritura:

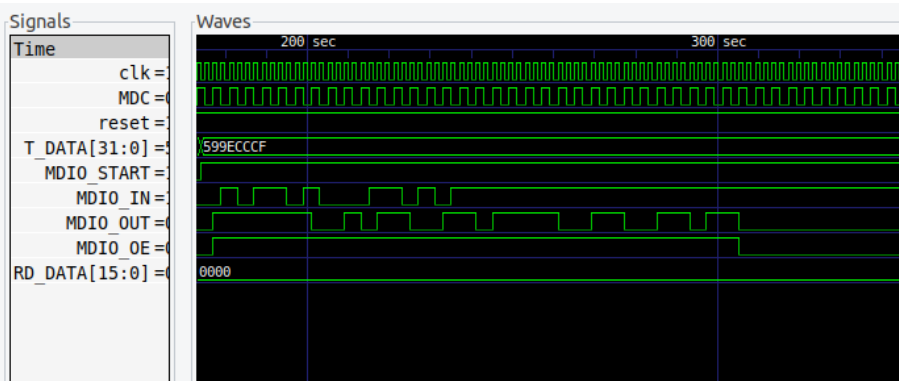


Figura 6: Transacción de escritura

Prueba para la transacción de Lectura:

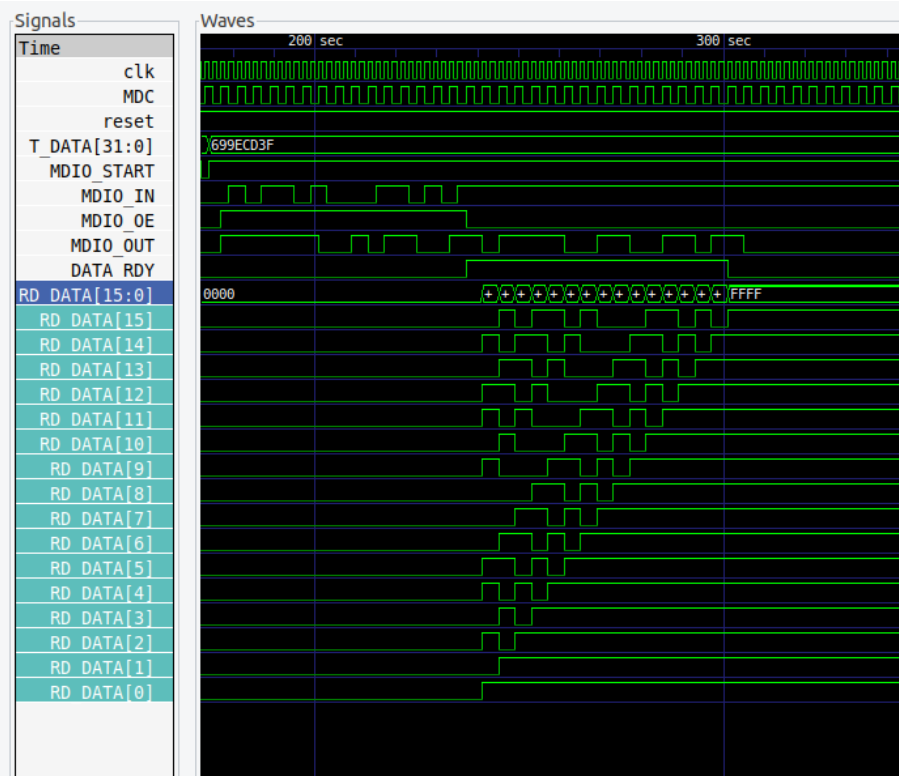


Figura 7: Prueba de Lectura

## 6. Conclusiones y recomendaciones

Se logro implementar correctamente el funcionamiento del protocolo MDIO, por medio de una maquina de estados y así comprender este protocolo, para lectura y escritura de datos, y además poder implementar correctamente una maquina de estados, la cual dependiendo del estado en que se encuentre me va a realizar la lectura o escritura.

También se logro implementar el convertidor de serie a paralelo y de paralelo a serie, para una palabra de 32 bits, y que las señales de *MDIO\_OUT* reciba la palabra en paralelo y la convierta en serie y el *MDIO\_OE* se ponga en alto por 32 ciclos de reloj del MDC en caso de que se trate de una transacción de escritura, mientras que si se trata de una transacción de lectura, este va a permanecer el alto por 16 ciclos de reloj del MDC.

También se logro implementar que la maquina de estados funcione siempre que la entrada en reset se encuentre en alto, y si esta entrada esta en bajo, todas las señales se pone en bajo.

Como recomendación, es bueno estudiar y comprender el protocolo MDIO que esta en la clausula 22 del estándar IEEE 802.3, de manera que se tenga mas claro el como implementar la maquina de estados para lograr realizar el protocolo MDIO.