

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0523 - Circuitos Digitales II

I ciclo 2023

Tarea 5

# Diseño de un generador y un receptor de transacciones I2C

Isaí David Vargas Ovares B88263

Grupo 1

Profesor: Enrique Coen Alfaro

4 de junio de 2023

# 1. Resumen

I2C (Inter-Integrated Circuit), también conocido como IIC, es un protocolo de comunicación en serie síncrono inventado por Philips Semiconductors en 1982. Es ampliamente utilizado para conexiones de corta distancia entre circuitos integrados periféricos de baja velocidad y procesadores y microcontroladores. Como se menciona en [1], el protocolo I2C utiliza dos líneas bidireccionales:

Línea de datos en serie (SDA): esta línea se utiliza para transferir datos entre los dispositivos maestro y esclavo. Línea de reloj serie (SCL): esta línea se utiliza para sincronizar la transferencia de datos entre dispositivos maestros y esclavos. I2C utiliza un enfoque maestro-esclavo, lo que significa que hay un maestro que controla la comunicación y uno o más esclavos responden a los comandos del maestro.

## 2. Descripción Arquitectónica

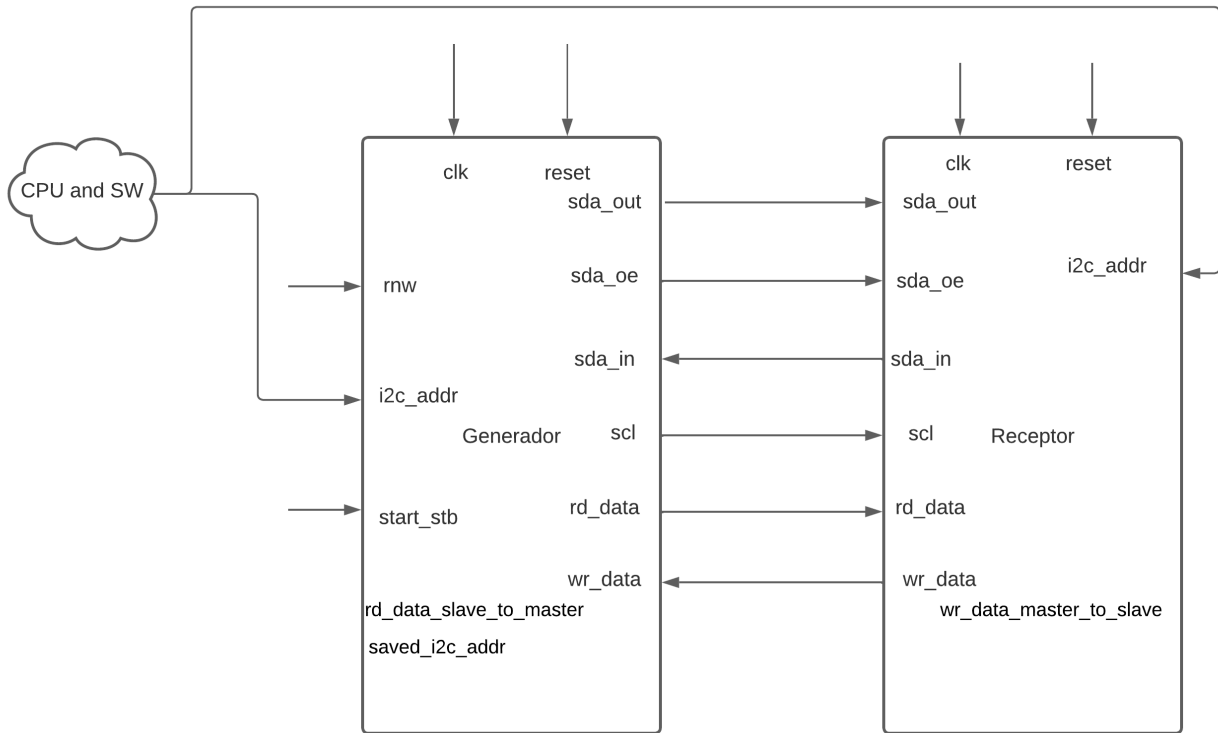


Figura 1: Interfaz generador y un receptor de transacciones I2C.

Un generador de transacciones I2C y un receptor de transacciones I2C, siguiendo las especificaciones del documento I2C-bus Specification and User Manual revisión 7.0 [1] y I2C Bus, Interface and Protocol [2]. Ambos componentes tienen interfaces específicas que permiten la comunicación y el intercambio de datos a través del protocolo I2C. El generador envía una señal de inicio en la línea sda mientras la línea scl está alta. Esto informa al receptor que una nueva comunicación está a punto de comenzar. Luego, el generador envía la dirección de 6 bits con la que desea comunicarse. Un esclavo con una dirección coincidente afirma la dirección bajando la línea sda mientras que la línea scl está alta. El generador puede entonces enviar datos o

leer datos del receptor. Cuando se completa la comunicación, el generador envía una señal de parada colocando la línea sda en nivel alto mientras que la línea scl está en nivel bajo.

A continuación se presenta un resumen de las especificaciones y características clave para cada uno de los componentes:

## **2.1. Generador de transacciones I2C:**

- CLK: Entrada del reloj desde el CPU con una frecuencia determinada.
- RESET: Entrada de reinicio del generador.
- START\_STB: Pulso de un ciclo de reloj que indica al generador que el CPU desea iniciar una transacción de I2C.
- RNW: Señal que indica si la transacción es de lectura o escritura (lectura = 1, escritura = 0).
- I2C\_ADDR[6:0]: Entrada de la dirección del receptor de transacciones con quien el generador se quiere comunicar.
- SCL: Salida del reloj para el I2C, este debe ser un 25 % de clk.
- SDA\_OUT: Salida serial con el comportamiento especificado por el protocolo I2C.
- SDA\_OE: Habilitación de SDA\_OUT, según quien tiene el control del bus.
- SDA\_IN: Entrada serial proveniente del probador.
- WR\_DATA[15:0]: Entrada paralela que contiene los 16 bits a enviar en una transacción de escritura.
- RD\_DATA[15:0]: Salida paralela que produce los 16 bits recibidos durante una transacción de lectura.

## **2.2. Receptor de transacciones I2C:**

- CLK: Entrada del reloj desde el CPU con una frecuencia determinada.
- RESET: Entrada de reinicio del generador.
- I2C\_ADDR[6:0]: Entrada de la dirección del receptor de transacciones.
- SCL: Entrada del reloj para el I2C.
- SDA\_OUT: Entrada serial con el comportamiento especificado por el protocolo I2C.
- SDA\_OE: Habilitación de SDA\_OUT.
- SDA\_IN: Salida serial enviada desde el receptor hacia el generador de transacciones.
- WR\_DATA[15:0]: Salida paralela que contiene los 16 bits recibidos durante una transacción de escritura.
- RD\_DATA[15:0]: Entrada paralela que produce los 16 bits a enviar durante una transacción de lectura.

## 2.3. Estados

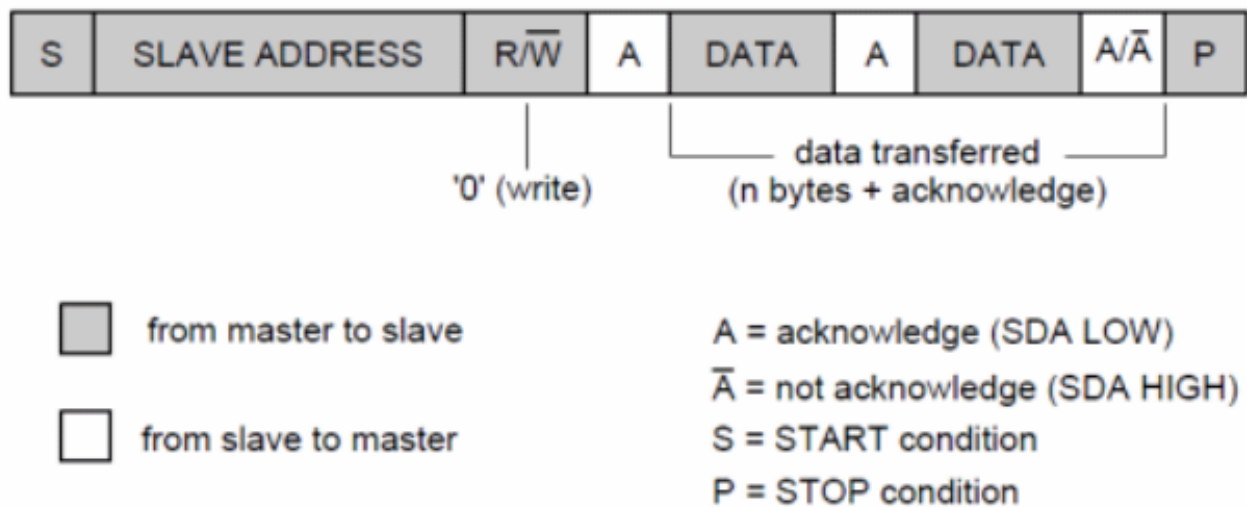


Figura 2: Transacción de escritura I2C.

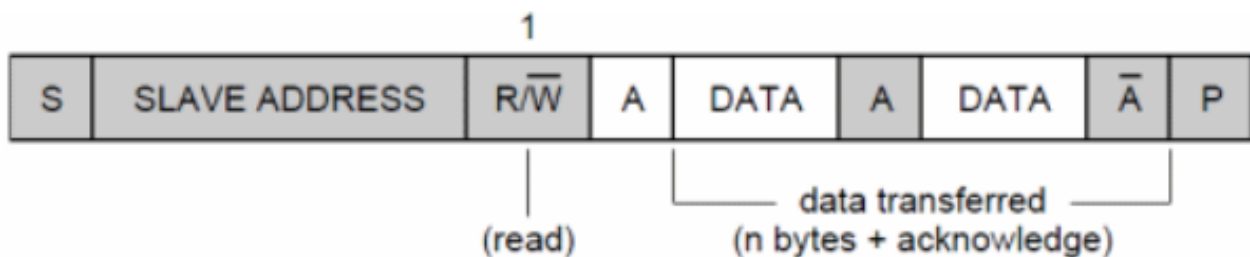


Figura 3: Transacción de lectura I2C

Cada modulo tiene sus estados según sea el modulo generador o el receptor, también se debe considerar que cuando el generador este en estado de escritura el receptor estará en estado de lectura de que lo que se esta escribiendo y viceversa, también se debe considerar el comportamiento de las imágenes anteriores, los de generador son:

- IDLE: Estado de reposo, se espera la señal strat\_stb
- START: Se recibió la condición de inicio.
- ADDRESS: Se envía la dirección y el rnw.
- READ\_ACK: Se lee el A de confirmación del receptor, si la dirección es correcta se pasa a WRITE\_DATA o READ\_DATA.
- WRITE\_DATA: Se escribe el primer byte a enviar, se espera el A del receptor y se escribe el segundo byte.
- WRITE\_ACK: En los casos de lectura de debe escribir un A al receptor, después de leer el primer byte.
- READ\_DATA: Se lee el primer byte a enviar, se envía el A al receptor y se escribe el segundo byte.

- READ\_ACK2: Se lee el segundo A de confirmación del receptor.
- WRITE\_ACK2: En los casos de lectura debe escribir un A al receptor, después de leer el segundo byte.
- STOP: Se recibió la condición de parada.

Estados del receptor:

- READ\_ADDR: Se lee la dirección enviada por el generador.
- SEND\_ACK: Se compara la dirección y si es correcta se envía A y se pasa a READ\_DATA o WRITE\_DATA, considerando que cuando el generador está escribiendo el receptor debe estar leyendo y viceversa.
- READ\_DATA: Se lee el primer byte escrito por el generador y envía A, se lee el segundo byte y se envía el cambio de A correspondiente.
- WRITE\_DATA: Se envía el primer byte y se lee el A enviado por el generador, se envía el segundo byte y se lee el A enviado por el generador.
- SEND\_ACK2: Se envía el A/ $\bar{A}$ .
- READ\_ACK: Se lee el primer A enviado por el generador.
- READ\_ACK2: Se lee el segundo A enviado por el generador.

## 2.4. Señales internas

- rd\_data\_slave\_to\_master: Registro interno del generador que es la unión de los dos bytes leídos, este sirve de comprobación que la transacción de lectura se realizó correctamente, ya que debe ser igual a rd\_data.
- saved\_i2c\_addr: Registro interno del generador que une la dirección y rnw.
- wr\_data\_master\_to\_slave: Registro interno del receptor que es la unión de los dos bytes escritos, este sirve de comprobación que la transacción de lectura se realizó correctamente, ya que debe ser igual a wr\_data.

## 3. Plan de Pruebas

- Prueba 1: Transacción de escritura estableciendo rnw=0, se desea escribir un número de 16 bits, en este caso se desea escribir CCCC=1100110011001100. En esta además se comprueba el correcto funcionamiento del clk, reset, star\_stb, scl, sda\_out, sda\_in, sda\_oe, wr\_data\_master\_to\_slave.
- Prueba 2: Transacción de lectura, estableciendo rnw=1, se desea leer un número de 16 bits, en este caso AAAA=1010101010101010. En esta además se comprueba el correcto funcionamiento del clk, reset, star\_stb, scl, sda\_out, sda\_in, sda\_oe, rd\_data\_slave\_to\_master.
- Prueba 3: Dirección distinta, se establece la dirección en la que el generador quiere escribir y se establece una distinta a esta para la dirección del receptor.

## 4. Instrucciones de utilización de la simulación

Esta tarea fue desarrollada en el entorno de EDAPlayground por lo que basta con ingresar al siguiente link y correr la simulación.

- <https://www.edaplayground.com/x/mPYH>

## 5. Resultados

- **Prueba 1:** Se puede ver que se realizó de manera correcta la transacción de escritura, este caso se escribe CCCC en forma de dos bytes en binario, el scl se estableció a un 25 % de clk, el reset se levanto y se permitió el funcionamiento normal, se dio el start\_stb para dar inicio, por lo que sda\_out se levanto para provocar la condición de start, una vez se dio la condición se empezó a transmitir la dirección junto con el rnw, por lo que después de la condición de start se transmitió el 63=01111111 junto con el 0 del rnw, para la señal de sda\_oe se tiene en alto en los tiempos correspondientes al control del bus por parte generador y las ocasiones donde pasa a bajo es porque se está recibiendo el A de confirmación del receptor. Para el wr\_data\_master\_to\_slave que es un registro interno del receptor se puede ver que efectivamente se escribió CCCC. Por lo que en general la prueba fue exitosa.

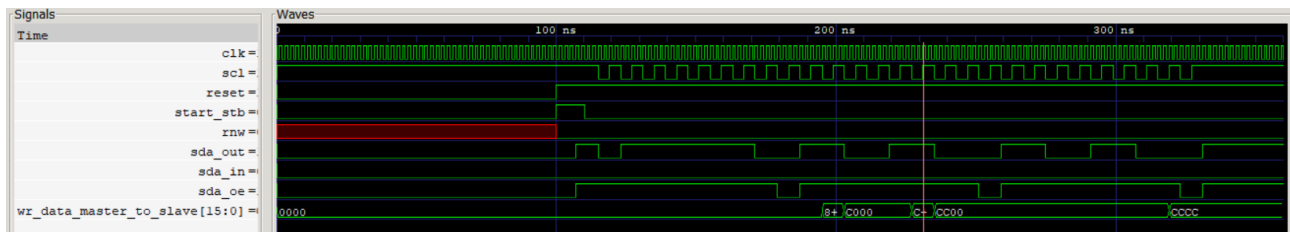


Figura 4: Prueba 1. Diagramas de tiempo.

- **Prueba 2:** Se puede ver que se realizó de manera correcta la transacción de lectura, este caso se lee AAAA en forma de dos bytes en binario, el scl se estableció a un 25 % de clk, el reset se levanto y se permitió el funcionamiento normal, se dio el start\_stb para dar inicio, por lo que sda\_out se levanto para provocar la condición de start, una vez se dio la condición se empezó a transmitir la dirección junto con el rnw, por lo que después de la condición de start se transmitió el 63=01111111 junto con el 1 del rnw, para la señal de sda\_oe se tiene en alto en los tiempos correspondientes al control del bus por parte generador y las ocasiones donde pasa a bajo es porque se está recibiendo el A de confirmación del receptor y cuando el receptor está transmitiendo por medio de sda\_in. Para el rd\_data\_slave\_to\_master que es un registro interno del generador se puede ver que efectivamente se escribió AAAA. Por lo que en general la prueba fue exitosa.

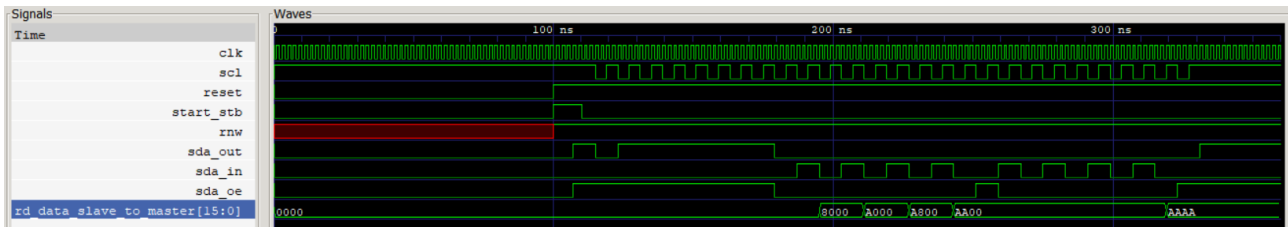


Figura 5: Prueba 2. Diagramas de tiempo.

- **Prueba 3:** En este caso se espera que la transacción no sea exitosa, ya que se establecen direcciones distintas, por lo que el receptor debe ignorar la solicitud de lectura, se puede ver que sucede de esa manera, ya que sda\_in no varía en ningún momento, esto a pesar de que el generador envió la dirección, pero al no ser la dirección del receptor no permite la lectura y esto se confirma con rd\_data\_slave\_to\_master que permanece en 0, con respecto a las demás señales se tiene el comportamiento esperado para una indicación de lectura por parte del generador, donde se dio la condición de start después de la indicación de start\_stb y stop de manera correcta. Por lo que en general la prueba fue exitosa.

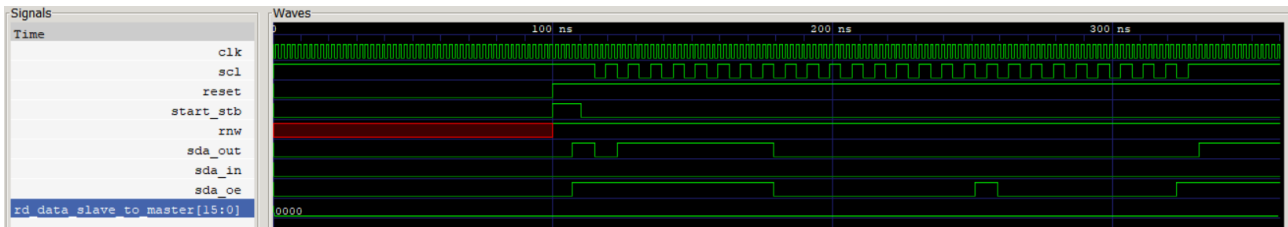


Figura 6: Prueba 3. Diagramas de tiempo.

## 6. Conclusiones

- Se logró de manera correcta la transacción de escritura y lectura de dos bytes, evidenciando la gran versatilidad que tiene este protocolo para realizar la comunicación con distintos dispositivos, ya que este al utilizar una dirección se puede seleccionar con qué dispositivo se quiere realizar la transacción.
- Resultó fácil aprender el comportamiento del protocolo, ya que se podría resumir todo en las señales de sda y scl, en este caso se pidió utilizar sda\_in y sda\_out por aparte, sin embargo este se podría implementar con un inuot y realizarlo de manera bidireccional e indicando quien controla el bus con la señal de sda\_oe, esto implicaría un gran ahorro de recursos y diseño.
- Se podría aumentar la cantidad de bytes enviados de manera sencilla, aunque se puede notar que es un protocolo algo lento ya que se debe enviar cierta cantidad de bit y esperar una confirmación para luego continuar.
- Dado que cada receptor tiene su dirección propia sería muy fácil añadir más dispositivos, incluso se podría cambiar la velocidad de transmisión para adaptarse a los requisitos del receptor, como se menciona en [1], esta velocidad puede ir desde los 100kbit/s hasta los 5Mbit/s.

## Referencias

- [1] N. Semiconductors, “I2c-bus specification and user manual,” 2021.
- [2] i2c Info. (2023) I2c bus specification.