Title of the experiment:

Simulation of I2C Communication Protocol using eSim.

Theory:

The Inter-Integrated Circuit or I2C protocol can be understood as a synchronous, multi-master, multi-slave, packet-switched, and single-ended serial communication bus that uses two bidirectional and open-drain lines for data transfer and timing. This protocol makes use of the Serial Data Line (SDA) and the Serial Clock Line (SCL) and pulls them to voltage supply rails via resistor types. This bus has devices that operate under a Master-Slave relationship, wherein synchronism and control originate from the Master device that transmits data through generating a Start bit and subsequently sends out a unique 7-bit Address to pinpoint any particular Slave device and accompany it with a Read/Write bit. Upon receipt of such specific data and control, the addressed Slave sends back an Acknowledge bit that sets the SDA line to low voltage level.

This experiment aims to simulate this physical layer of an I2C bus and its use of NMOS transistors that simulate the open drain line to perform a mixed signal analysis of addressed data, data sending, and multiple slaves, offering better integration and performing multiple functions effectively.

Schematic Diagram:

The circuit schematic of the Multi-Slave I2C bus designed in eSim is as shown below:

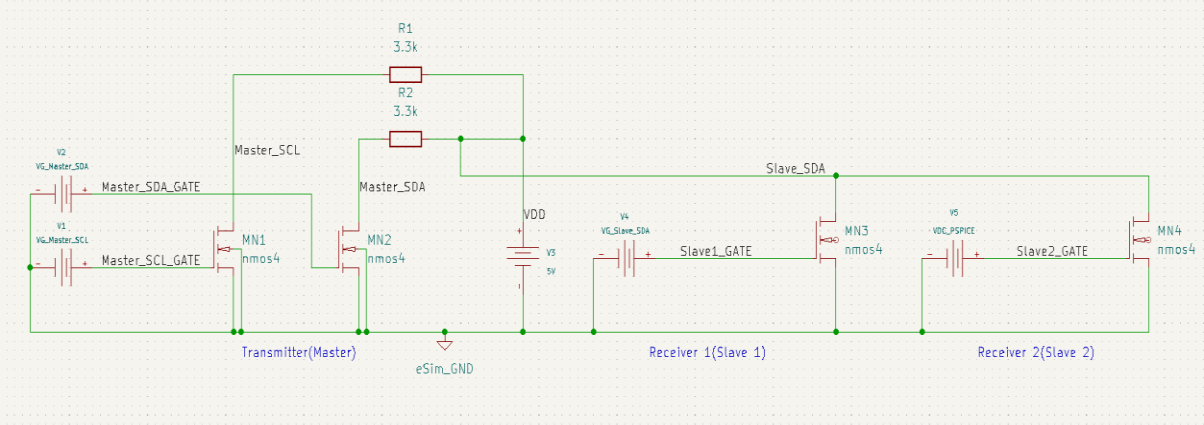

Figure 1: Multi-Slave I2C Bus Schematic

Protocol Requirements & Reference Circuit:

To verify the design, the simulation was tested against standard I2C physical layer requirements (NXP UM10204 Specification):

1. **Bus Topology:** Two bidirectional Open-Drain lines (SDA, SCL) pulled up to VDD.
2. **Logic Levels:** Logic '0' (< 0.3VDD) and Logic '1' (> 0.7VDD).
3. **Start Condition:** SDA transitions High-to-Low while SCL is High.

4. **Stop Condition:** SDA transitions Low-to-High while SCL is High.
5. **Acknowledge (ACK):** The receiver pulls SDA Low during the 9th clock pulse.

The reference circuit used corresponds to the standard Open-Drain configuration using NMOS transistors (for pull-down) and 3.3kΩ resistors (for pull-up), as defined in standard literature [1].

Simulation Results:

1. Ngspice Plots

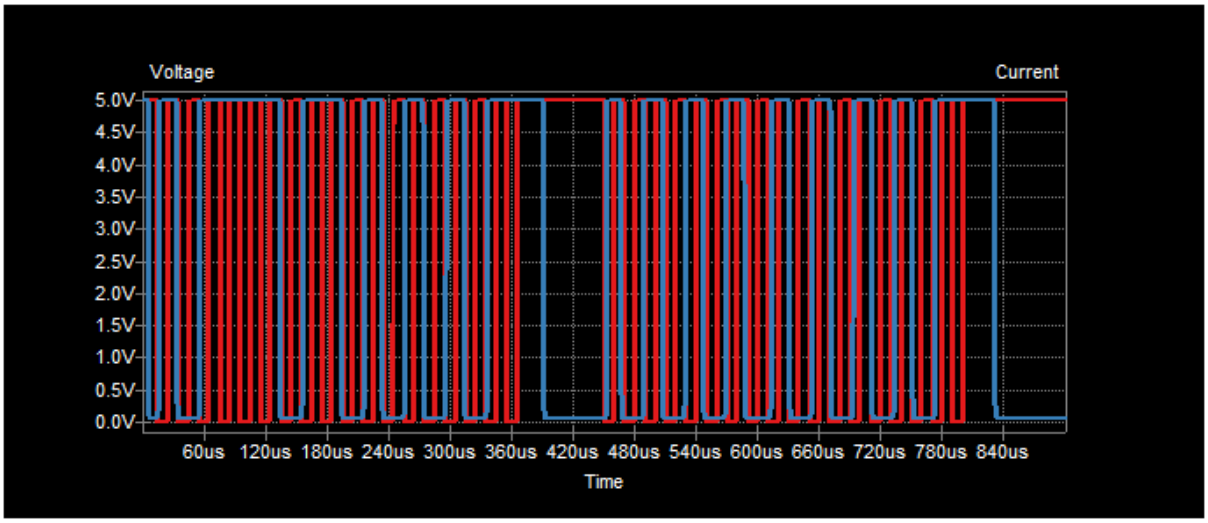

Figure 2: Ngspice Output Plot showing Multi-Slave Addressing and Data Transmission

2. Waveform Validation

The following table compares the required I2C protocol states with the observed simulation timestamps (refer to Ngspice Plot):

Protocol State	Requirement	Observed in Simulation	Status
Idle Bus	SCL and SDA both High	0µs - 10µs	**Pass**
Start Condition 1	SDA falls while SCL is High	~14µs	**Pass**
Address (Slave 1)	Binary 1010000 (0x50)	18µs - 150µs	**Pass**
ACK 1 (Address)	SDA Low during 9th Clock	~170µs	**Pass**
Data Payload 1	Byte 0xAA (10101010)	190µs - 350µs	**Pass**
ACK 2 (Data)	SDA Low during 18th Clock	~370µs	**Pass**
Stop Condition 1	SDA rises while SCL is High	~400µs	**Pass**
Start Condition 2	SDA falls while SCL is High	~450µs	**Pass**
Address (Slave 2)	Binary 1100000 (0x60)	460µs - 610µs	**Pass**

ACK 3 (Address)	SDA Low during 9th Clock	~630µs	**Pass**
Data Payload 2	Byte 0x55 (01010101)	650µs - 790µs	**Pass**
ACK 4 (Data)	SDA Low during 18th Clock	~810µs	**Pass**
Stop Condition 2	SDA rises while SCL is High	~830µs	**Pass**

Inference:

The simulation results demonstrate two distinct transactions: Transaction 1 (Slave 1 receiving 0xAA) and Transaction 2 (Slave 2 receiving 0x55). The simulation accurately reproduces all mandatory protocol phases (Start, Address, Data, ACK, Stop). The voltage levels verify the correct operation of the Open-Drain/Pull-up configuration (Low ≈ 0V, High ≈ 5V).

Conclusion:

Thus, we have studied the I2C communication protocol by simulating a mixed-signal, multi-slave architecture using eSim. The simulation successfully verified the open-drain physical layer characteristics, independent slave addressing, and accurate data transmission as observed in the output waveforms.

References:

1. https://www.i2c-bus.org/specification/
2. https://www.electronics-tutorials.ws/combination/comb_1.html
3. Spoken Tutorial Project (eSim Tutorials): https://spoken-tutorial.org/tutorial-search/?search_foss=eSim&search_language=English