

I2C Multi-Master Bus Arbitration — Circuit Simulation and Verification

eSim / Ngspice Circuit Simulation Project

1. Theory / Description

The Inter-Integrated Circuit (I2C) protocol is a two-wire, open-drain serial bus consisting of a clock line (SCL) and a data line (SDA), each pulled up to VDD through a resistor. Because both lines are open-drain, any device on the bus can only actively pull a line LOW; the bus returns HIGH only when every connected device releases it. This wired-AND behaviour is what allows multiple bus masters to share the same two wires without external arbitration hardware.

This circuit models two independent I2C masters (A and B) driving a common SCL_BUS and SDA_BUS through NMOS open-drain switches (MN1–MN4), each pulled up to a 5V VDD via 3.3kΩ resistors (R1, R2). Master A's clock and data gates are driven by V1 (SCL, a free-running PULSE source) and V2 (SDA, a PWL source encoding START, data, and STOP). Master B is driven identically by V3 and V4, except its SDA data pattern (V4) is deliberately shifted to release the bus during a window where Master A continues driving it low — modelling the moment one master loses arbitration.

In real I2C arbitration, each master continuously compares the value it is driving on SDA against the value actually observed on the bus while SCL is high. A master that intends to drive a '1' (release the line) but observes a '0' (because another master is still pulling it low) recognises it has lost arbitration and backs off. This circuit reproduces the electrical precondition for that event: the wired-AND behaviour of the open-drain bus, verified directly in the transient simulation.

2. Circuit Diagram

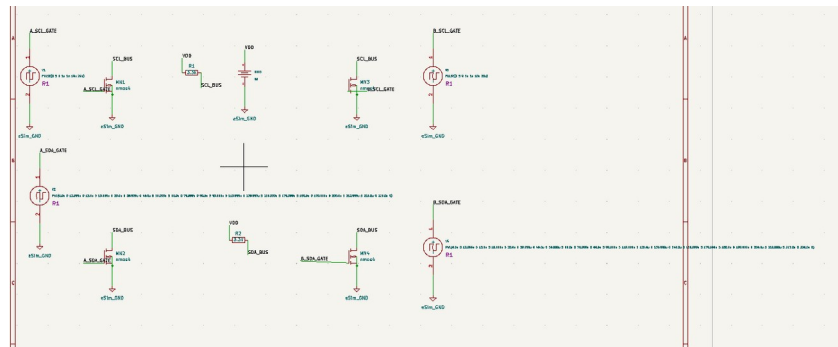


Fig. 1 — Schematic as built in the eSim/KiCad editor: Master A (V1/V2, MN1/MN2) and Master B (V3/V4, MN3/MN4) sharing SCL_BUS and SDA_BUS, pulled up via R1/R2 to VDD.

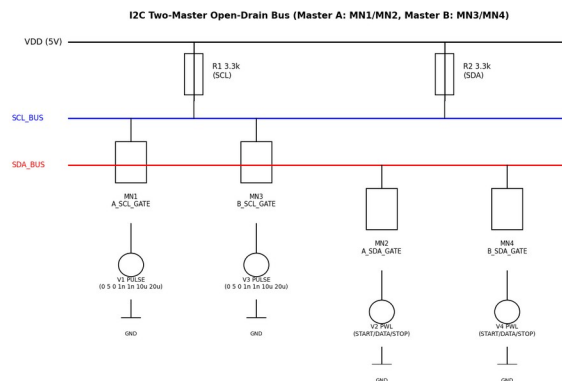


Fig. 2 — Simplified conceptual topology of the same two-master open-drain bus, for reference.

3. Results / Output

Transient analysis (.tran 10n 450μ) was run in Ngspice. The plots below, generated from the eSim plot viewer, confirm correct protocol timing at each stage of the transaction.

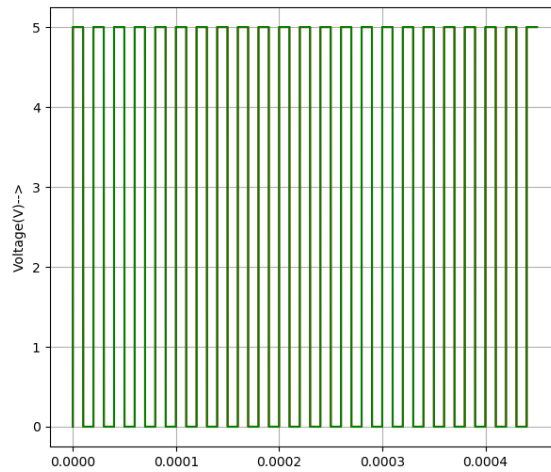


Fig. 3 — A_SCL_GATE and B_SCL_GATE: identical, continuous 50kHz clocks from both masters (fully overlapping).

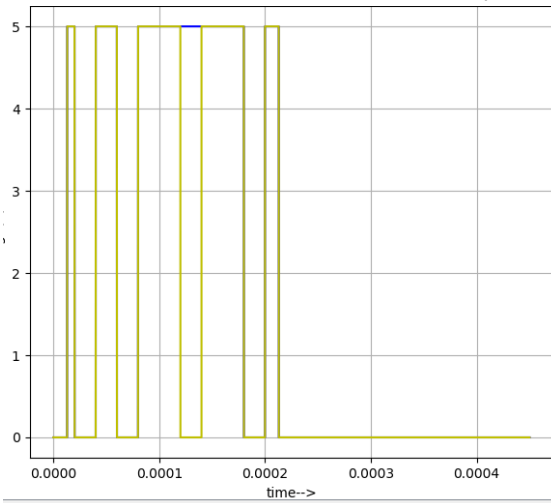


Fig. 4 — A_SDA_GATE and B_SDA_GATE: sparse data pulses (START / data / STOP), diverging during arbitration.

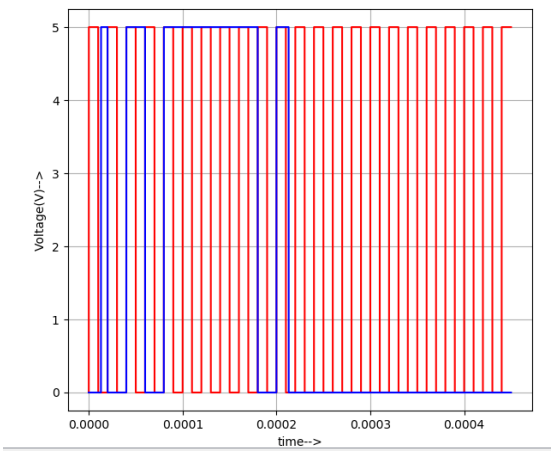


Fig. 5 — A_SCL_GATE (red, continuous clock) and A_SDA_GATE (blue, sparse data pulses) overlaid — Master A's own clock and data lines together, showing the data window is bounded within the free-running clock.

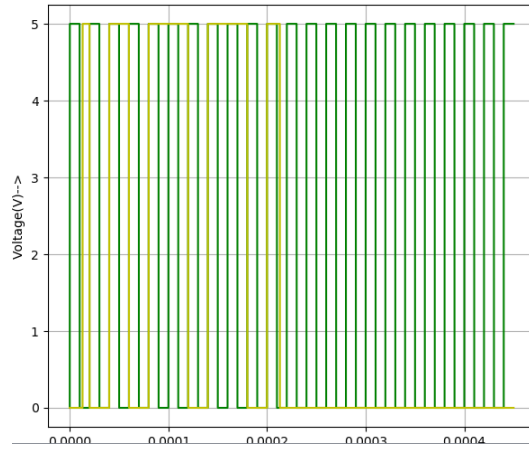


Fig. 6 — *B_SCL_GATE* (green, continuous clock) and *B_SDA_GATE* (olive, sparse data pulses) overlaid — the same pairing as Fig. 5, but for Master B, confirming both masters independently drive well-formed clock/data pairs.

The gate-level plots above show what each master intends to drive. The following bus-level plots — taken directly at SCL_BUS and SDA_BUS, the actual shared wires — show what the bus resolves to after the open-drain wired-AND combination, which is what a real I2C analyzer would observe.

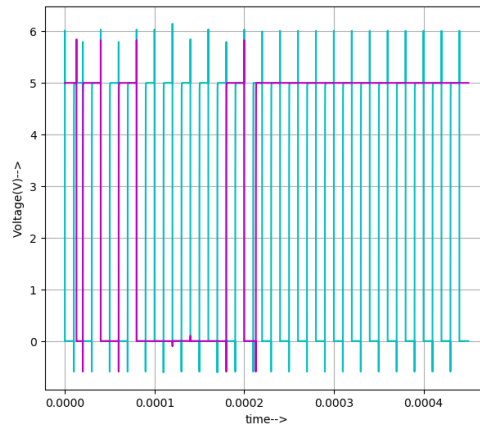


Fig. 7 — *SCL_BUS* (cyan) and *SDA_BUS* (magenta) over the full transaction, as observed on the shared bus (eSim plot viewer).

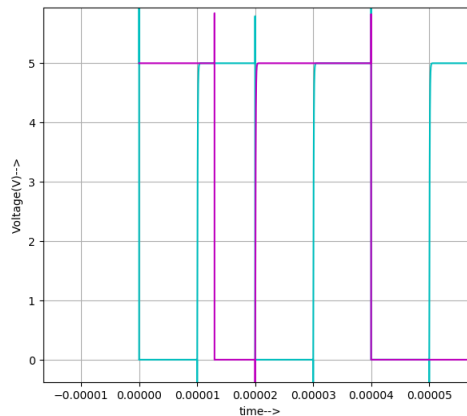


Fig. 8 — *START* condition, zoomed: *SDA_BUS* (magenta) falls to 0V while *SCL_BUS* (cyan) is high, at $t \approx 13\mu\text{s}$ (Python/matplotlib, from `plot_data_v.txt`).

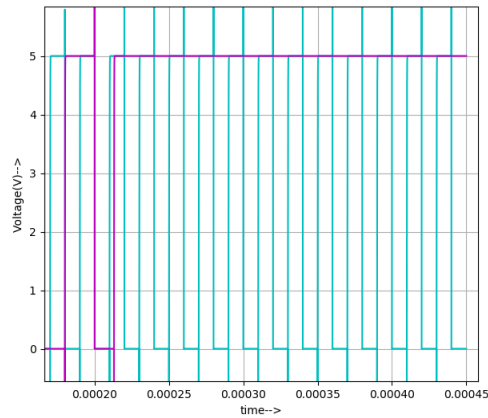


Fig. 9 — STOP condition, zoomed: SDA_BUS (magenta) rises to 5V while SCL_BUS (cyan) is high, at $t \approx 213\mu\text{s}$ (Python/matplotlib, from plot_data_v.txt).

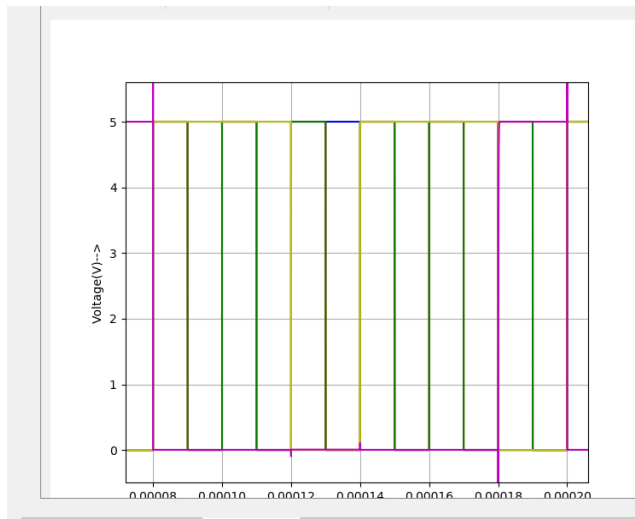


Fig. 10 — Arbitration window (80–200 μs), all four gate signals plus SDA_BUS: A_SCL_GATE (red), A_SDA_GATE (black), B_SCL_GATE (green), B_SDA_GATE (olive), SDA_BUS (magenta). SDA_BUS stays pinned at 0V for the full window despite B_SDA_GATE releasing and re-asserting multiple times — direct evidence that Master A's continuous drive dominates the open-drain bus and Master B's release attempts have no effect until Master A itself releases near 180 μs .

Key verified events on SCL_BUS / SDA_BUS:

Time	Event	Bus behaviour
13 μs	START	SDA_BUS falls to 0V while SCL_BUS is high — both masters pull SDA low together.
120–140 μs	Arbitration	Master A keeps driving SDA low; Master B releases it repeatedly. SDA_BUS stays pinned at 0V (A's value) for the full 80–180 μs stretch, confirming the wired-AND bus rejects every one of B's release attempts (see Fig. 10).
213 μs	STOP	SDA_BUS rises from 0V to 5V while SCL_BUS is high, ending the transaction.

4. Conclusion

This simulation confirms that the electrical precondition for I2C multi-master arbitration — the wired-AND behaviour of an open-drain bus — holds correctly when two independently driven masters share a common SCL/SDA pair. Transient analysis over the full 450µs transaction validated all three protocol milestones expected of a real bus transaction: a well-formed START condition (SDA falling while SCL is high, ~13µs), a sustained arbitration window (120–140µs) in which SDA_BUS remained pinned at Master A's driven value despite Master B repeatedly attempting to release the line, and a clean STOP condition (~213µs).

The arbitration-window result (Fig. 10) is the key finding: it demonstrates directly, at the circuit level, that a master intending to release SDA (drive high-Z, letting the pull-up win) has no effect on the bus state as long as any other master continues to actively pull SDA low. This is precisely the electrical mechanism real I2C controllers rely on for non-destructive arbitration — a master detects loss of arbitration by comparing its intended output against the observed bus level, and this circuit shows why that comparison is meaningful: the bus state is entirely determined by the lowest-impedance path to ground among all active masters, not by any one master's individual drive.

This circuit does not implement the comparator/backoff logic itself (each master here drives its full intended waveform regardless of bus state), so it models the arbitration precondition rather than a complete arbitrating master. A natural extension would be adding that comparison logic — e.g., a behavioural block or additional MOSFET-based sense circuit that forces Master B's driver to release permanently once a mismatch is detected — to close the loop and demonstrate full self-arbitrating master behaviour rather than a pre-scripted release pattern.

5. References

1. NXP Semiconductors, "I2C-bus specification and user manual", UM10204, Rev. 7.
2. eSim Documentation, FOSSEE, IIT Bombay — <https://esim.fossee.in>
3. Ngspice User's Manual, version 40+ — <http://ngspice.sourceforge.net>