

CAN (Controller Area Network) Bus – Mixed-Signal Modelling and Verification of a Multi-Node Communication Protocol in eSim

Sourabh Kumar · Thapar Institute of Engineering and Technology · FOSSEE eSim Circuit Simulation Project

1. Introduction

This project models a CAN (Controller Area Network) bus in eSim as a mixed-signal system. The physical layer is simulated in Ngspice and the protocol controller is written in VHDL and integrated through eSim's NGHDL interface. Two controller instances share one differential pair and contend for it.

CAN was chosen over simpler protocols such as UART or SPI because it requires both halves of eSim to work together. Arbitration in CAN is not a logical operation on a data structure: a node drives a recessive level onto the wire, reads the actual voltage back, and withdraws if it reads dominant. That only means anything if the analog bus is genuinely present, with real thresholds and real driver impedance. The same applies to the acknowledge slot and to error signalling, where a node deliberately violates the bit-stuffing rule to destroy a frame the other nodes are receiving.

2. Theory

Physical layer. CAN signals differentially on a twisted pair. The recessive state has both lines biased to 2.5 V, giving zero differential. The dominant state pulls CANH toward the supply and CANL toward ground through driver on-resistance, giving about 2 V differential. Because any node driving dominant overrides all nodes driving recessive, the bus behaves as a wired-AND. ISO 11898-2 places the receiver thresholds at 0.9 V for dominant and 0.5 V for recessive.

Bit timing. There is no clock line. Each bit is divided into time quanta and the receiver samples at a fixed point within the bit, resynchronising on recessive-to-dominant edges. The propagation segment must be long enough that a signal from the far end of the bus arrives before the sample point.

Arbitration. All nodes wishing to transmit begin simultaneously and send their identifiers most-significant bit first. A node sending recessive that reads dominant has lost to a lower identifier and becomes a receiver. The winner is unaffected and no bandwidth is lost, which is why CAN arbitration is called non-destructive.

3. Design parameters

Parameter	Value	Basis
Bit rate	125 kbit/s	design choice
Bit time	8.000 us	1 / bit rate
Time quanta per bit	16 (tq = 500 ns)	design choice
Controller clock	2.000 MHz	one tq per cycle
Segments SYNC/PROP/PS1/PS2	1 / 5 / 6 / 4 tq	PROP from measurement
Sample point	12 tq = 75%	beyond one bus round trip
Driver on-resistance	45 ohm	1.5 V at 33.333 mA
Termination	120 ohm x 2	ISO 11898-2
Bus capacitance	500 pF per line	design choice
Receiver thresholds	0.9 V dom / 0.5 V rec	ISO 11898-2
Identifiers	0x0A5 and 0x123	0x0A5 must win

The propagation segment is derived rather than assumed. Over a modelled 220 m bus the one-way delay measures 1.1136 us and the transceiver contributes 21.6 ns, so the round trip is 2.27 us against the 2.500 us that five time quanta allow, a margin of 9 per cent. Placing the sample point at 75 per cent puts it more than one round trip after the edge.

4. Circuit

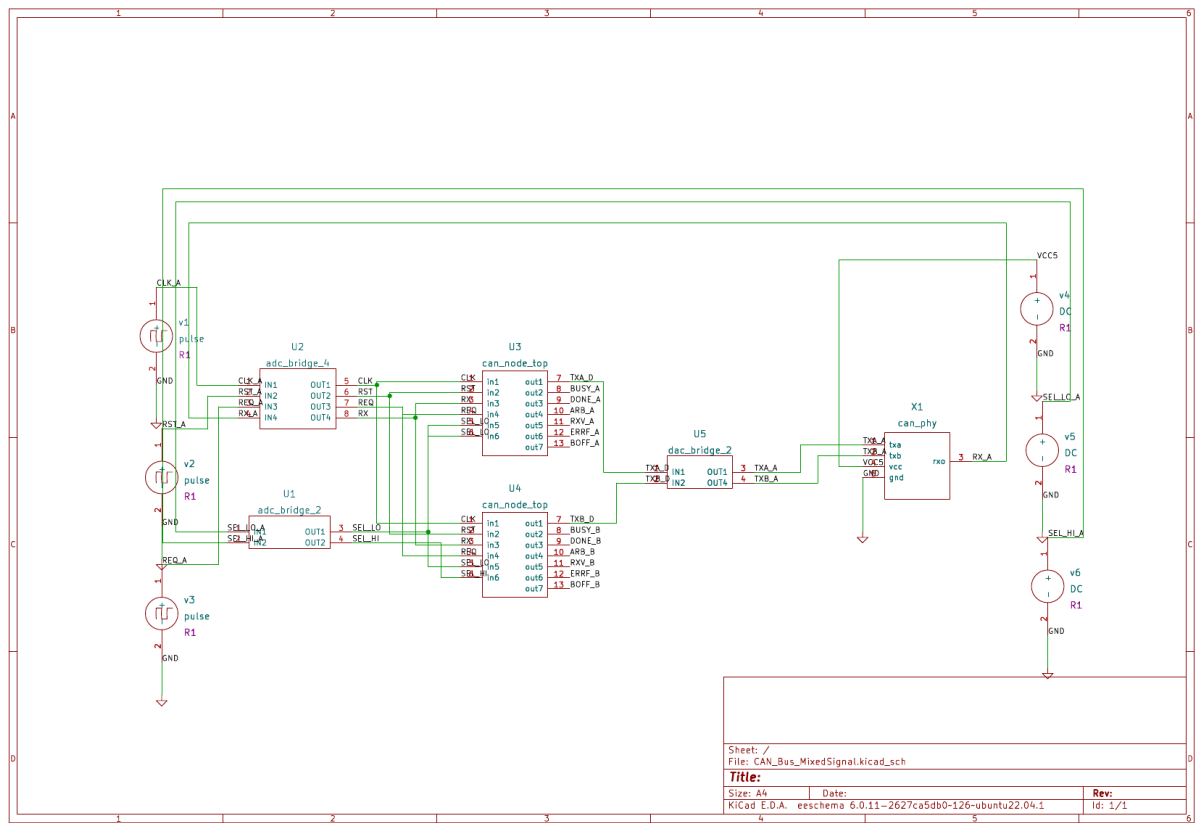


Figure 1. eSim schematic. Two `can_node_top` NGHDL controllers (U3, U4) share the `can_phy` transceiver subcircuit (X1). Bridges U1 and U2 convert the analog stimulus to the digital domain and U5 returns the transmit outputs to analog.

The controller is ten VHDL modules covering bit timing with resynchronisation, bit stuffing, CRC-15 with polynomial 0x4599, frame assembly and decoding, arbitration, acknowledgement, error frames and the transmit and receive error counters that move a node between error-active, error-passive and bus-off. It is compiled by GHDL and runs as an Ngspice code model through NGHDL.

The transceiver is an Ngspice subcircuit containing two driver stages sharing one differential pair, dual termination, the bias network, bus capacitance and a differential comparator centred at 0.7 V. Both driver stages act on the same pair, so the wired-AND behaviour is physical rather than logical.

5. Results

5.1 Python plots

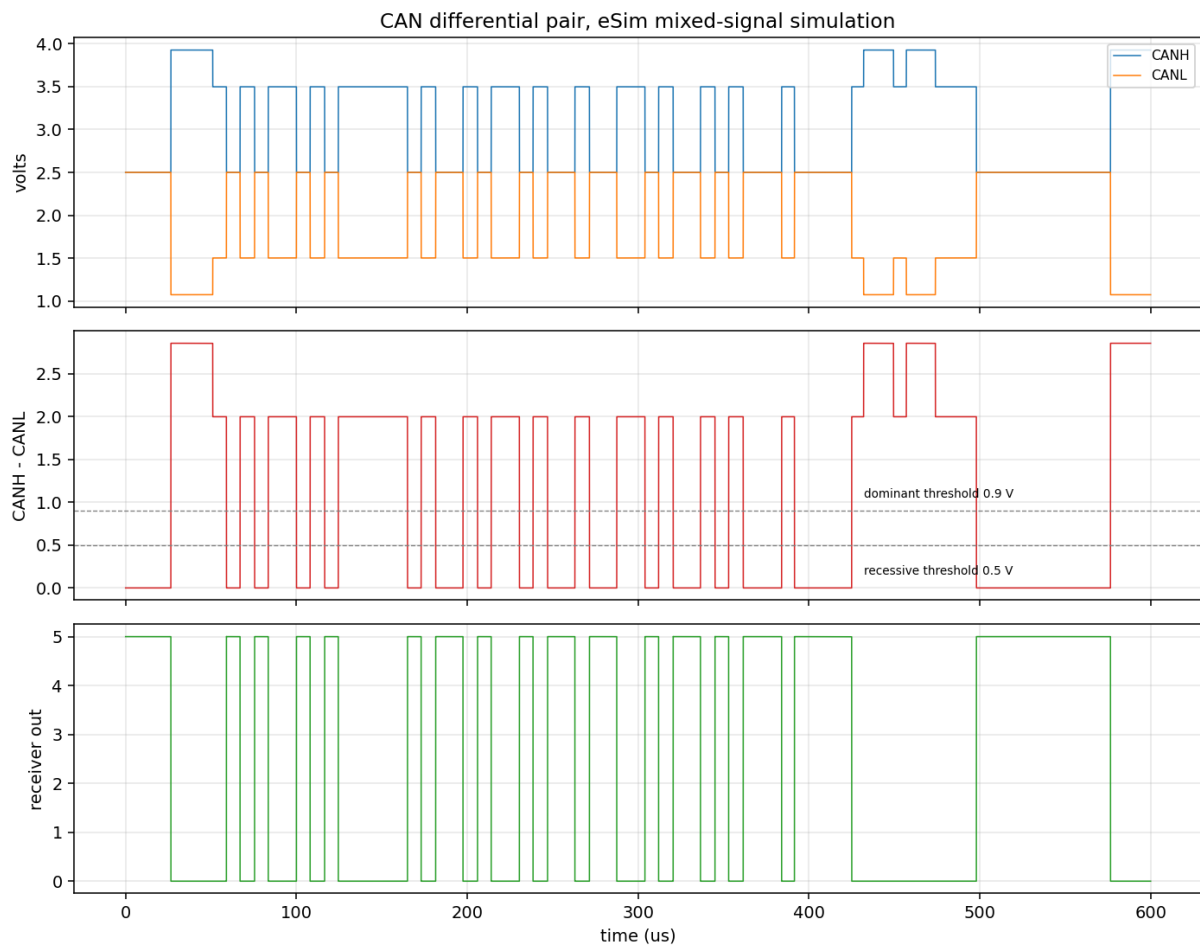


Figure 2. Python plot. Complete frame: CANH and CANL (top), differential with the ISO thresholds marked (middle), receiver output (bottom).

The bus levels match ISO 11898-2: CANH moves from 2.5 V to 3.5 V, CANL from 2.5 V to 1.5 V, and the differential from 0 V recessive to 2 V dominant. The receiver output tracks the differential cleanly with no ambiguous states.

Arbitration is directly visible in the analog waveform. Between roughly 26 us and 55 us the differential reaches 2.85 V rather than 2.0 V, because both nodes are driving dominant at the same time and two 45 ohm drivers in parallel present 22.5 ohm. After 55 us the differential settles to 2.0 V, showing that one node has withdrawn and a single driver remains. That transition is the moment arbitration resolves, and it is readable directly off the bus voltage.

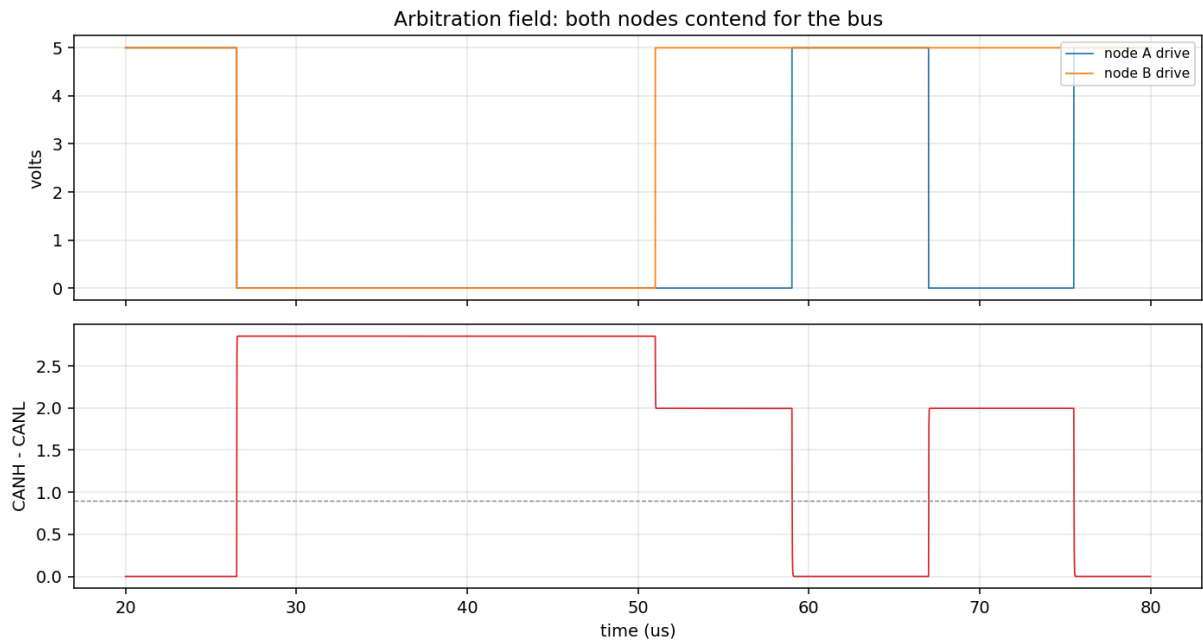


Figure 3. Python plot. Arbitration field: both nodes drive the bus; the elevated differential indicates simultaneous dominant drive.

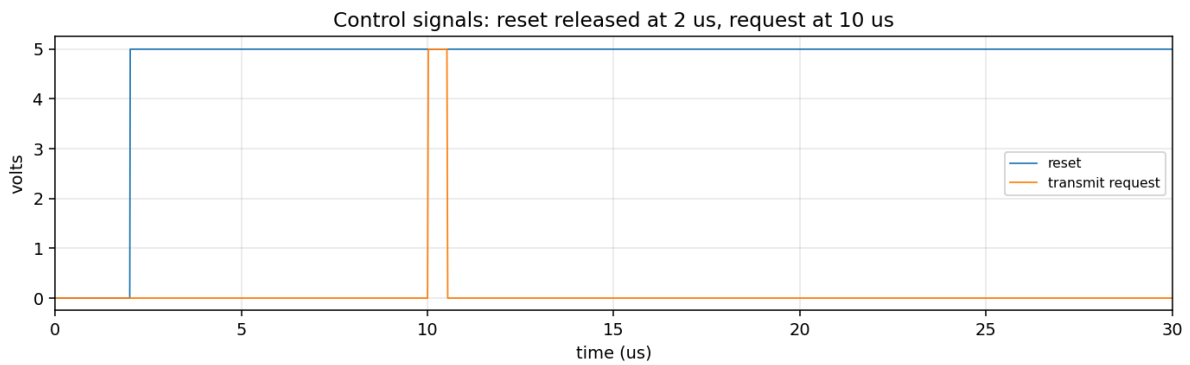


Figure 4. Python plot. Control signals: reset released at 2 us, transmit request asserted at 10 us.

Figures 2 to 4 are plotted in Python from the exported raw data. Figures 5 and 6 are produced by Ngspice itself, from the same run, using the hardcopy command on the loaded raw file. They are included so the results can be seen in the simulator's own output as well as in post-processing.

5.2 Ngspice plots

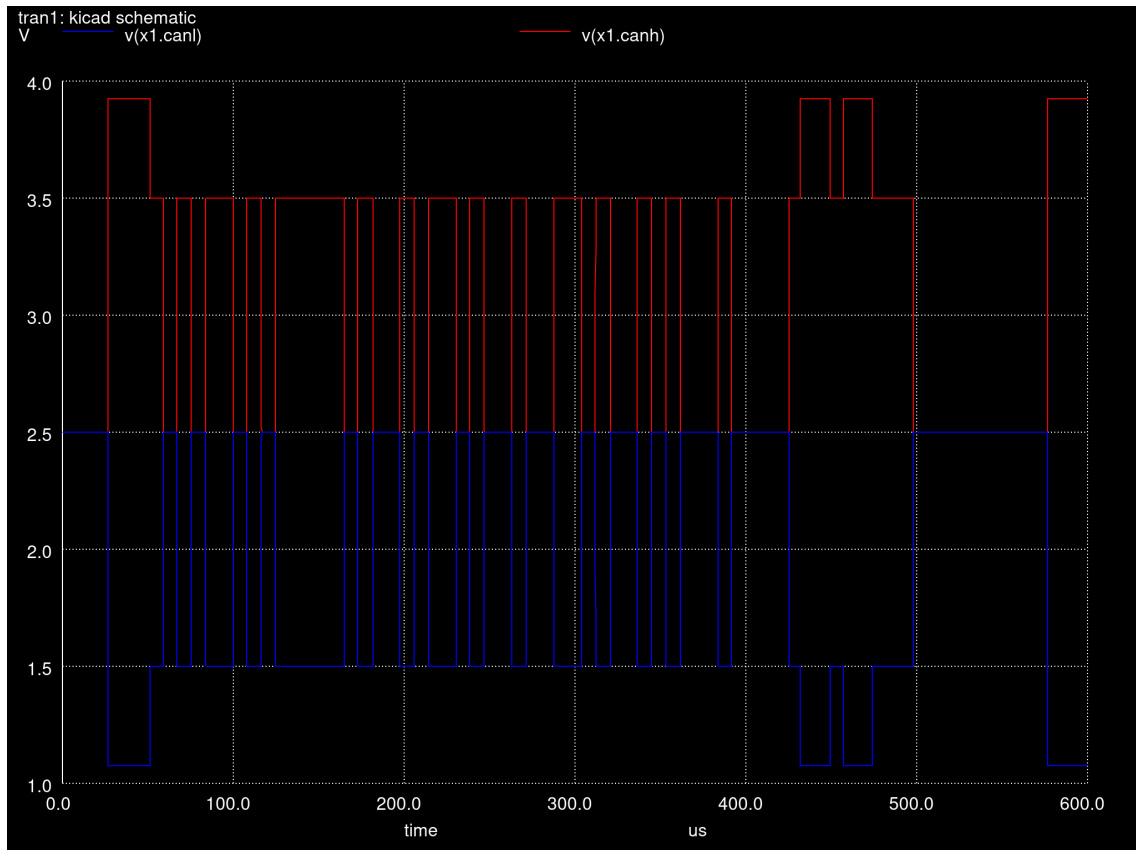


Figure 5. Ngspice plot of CANH and CANL. Both lines rest at 2.5 V recessive and separate to 3.5 V and 1.5 V dominant. The wider excursions are the arbitration field, where two nodes drive dominant at once.

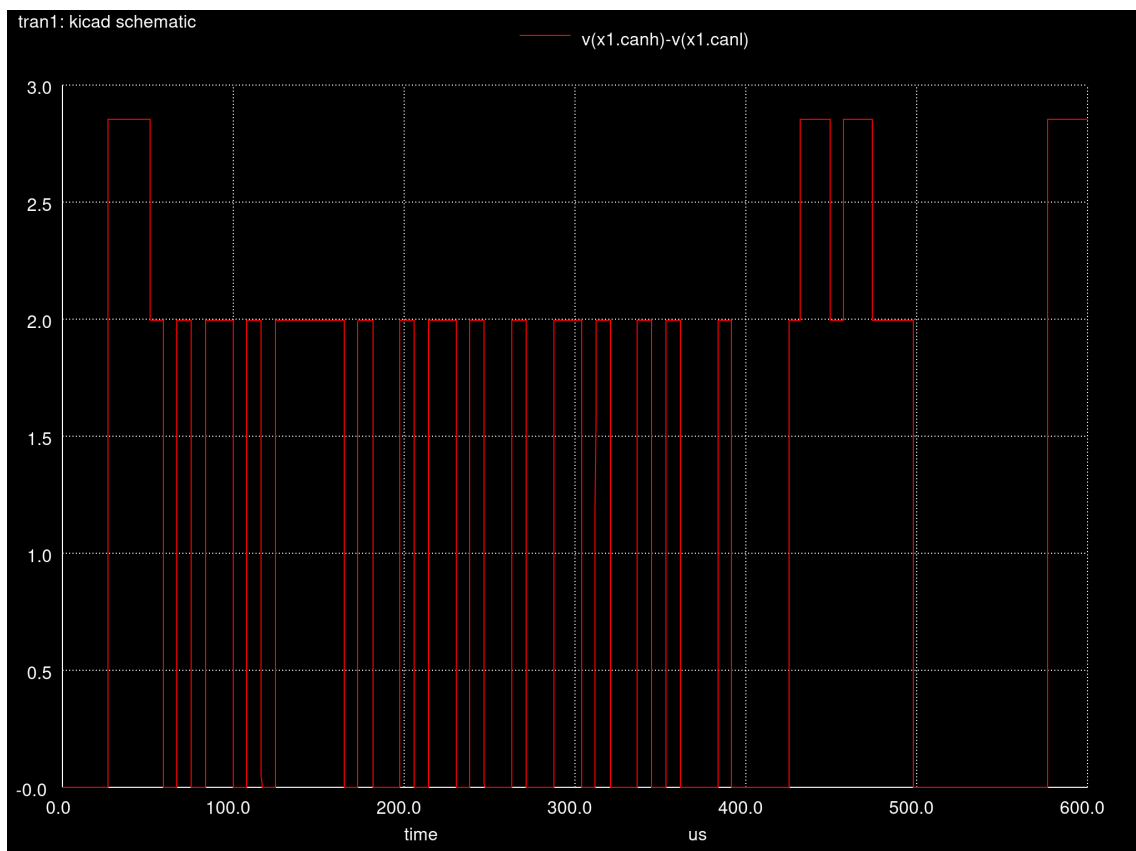


Figure 6. Ngspice plot of the differential voltage. 0 V recessive and 2.0 V dominant, with the 2.85 V plateaus produced by two drivers in parallel during arbitration.

6. Verification

Each block was checked against a reference model written from the ISO specification rather than from the RTL, so that the two could disagree. The CRC implementation was compared over 417 vectors including 400 random frames and matched on every one. Frame assembly was compared bit for bit against independently generated expected output for 68 frames, including the positions of stuff bits, with no mismatches. Arbitration was tested with four nodes whose identifiers predict the order of withdrawal; the three losers withdrew exactly one bit time apart in the predicted order and all of them received the winner's frame. A physical bus fault injected as a low-resistance short across the pair drove the affected node to bus-off while a healthy node on the same wire continued unaffected.

7. Limitations

Standard 11-bit identifiers only; extended identifiers and overload frames are not implemented. A node that loses arbitration abandons the frame rather than retrying. The eSim netlist drives both nodes from one clock, although resynchronisation itself is verified against independent oscillators in simulation: sweeping one node from 0 to 6 per cent skew, reception stays correct to 3 per cent, three times the derived 0.98 per cent bound, and fails safely above that by flagging an error rather than accepting corrupt data. The comparator has no common-mode input range, so the model is optimistic outside the range where real transceivers saturate.

8. Conclusion

A two-node CAN bus was modelled in eSim as a mixed-signal system and its operation verified. The bus levels match ISO 11898-2, a complete frame transfers from transmitter to receiver, and arbitration, acknowledgement and error signalling were each confirmed against a reference model written independently of the RTL. Bit timing holds against oscillator skew up to 3 per cent, three times the derived tolerance, and fails safely beyond it by flagging an error rather than accepting corrupt data. The analog and digital halves were verified together rather than separately, which is what CAN requires: arbitration and acknowledgement are only meaningful when the wired-AND behaviour of the physical bus is genuinely present.

9. References

ISO 11898-1:2015, Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling.

<https://www.iso.org/standard/63648.html>

ISO 11898-2:2016, Road vehicles - Controller area network (CAN) - Part 2: High-speed medium access unit.

<https://www.iso.org/standard/67244.html>

Bosch CAN Specification Version 2.0, Robert Bosch GmbH, 1991.

FOSSEE, IIT Bombay. eSim: an open source EDA tool. <https://esim.fossee.in>

T. Gingold. GHDL, a VHDL simulator. <https://ghdl.github.io/ghdl/>

Ngspice, mixed-level/mixed-signal circuit simulator. <https://ngspice.sourceforge.io>