CAN COMMUNICATION BETWEEN THE MICRO’s WITHOUT CAN TRANSCEIVER

Scope: Hardware Design

Application: Battery Management System (BMS)

When it comes to keeping rechargeable batteries in check, the hardware circuit design of a Battery Management System (BMS) is key. This circuit is like the brain of electric vehicles, making sure everything runs smoothly. Modifications were required in the CAN communication between the microcontrollers to remove some components, aiming to reduce costs to the existing design.

Hardware Design – Challenge & SoW

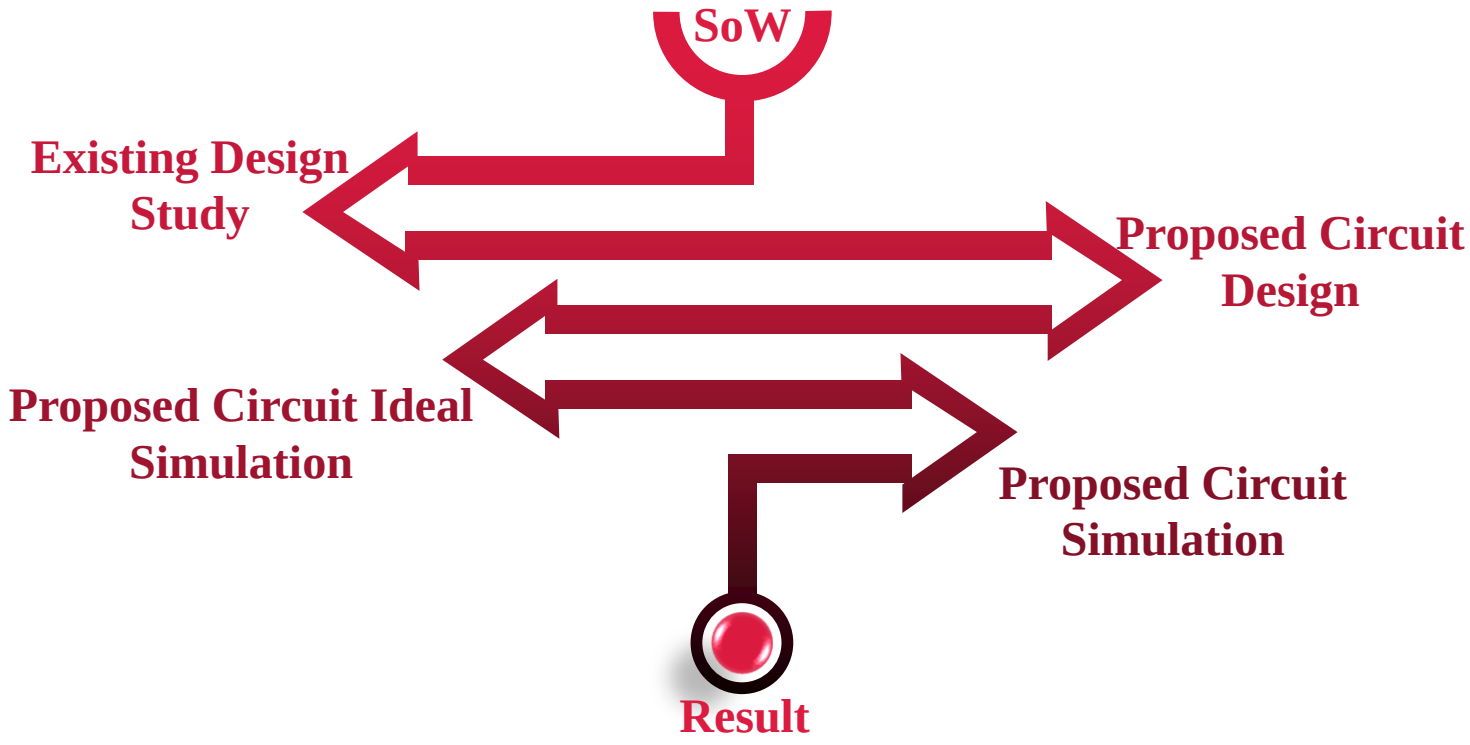
We were tasked to remove CAN transceiver component without change or reducing the functionality of the existing circuit design.

Challenge:

- Functionality should not be changed

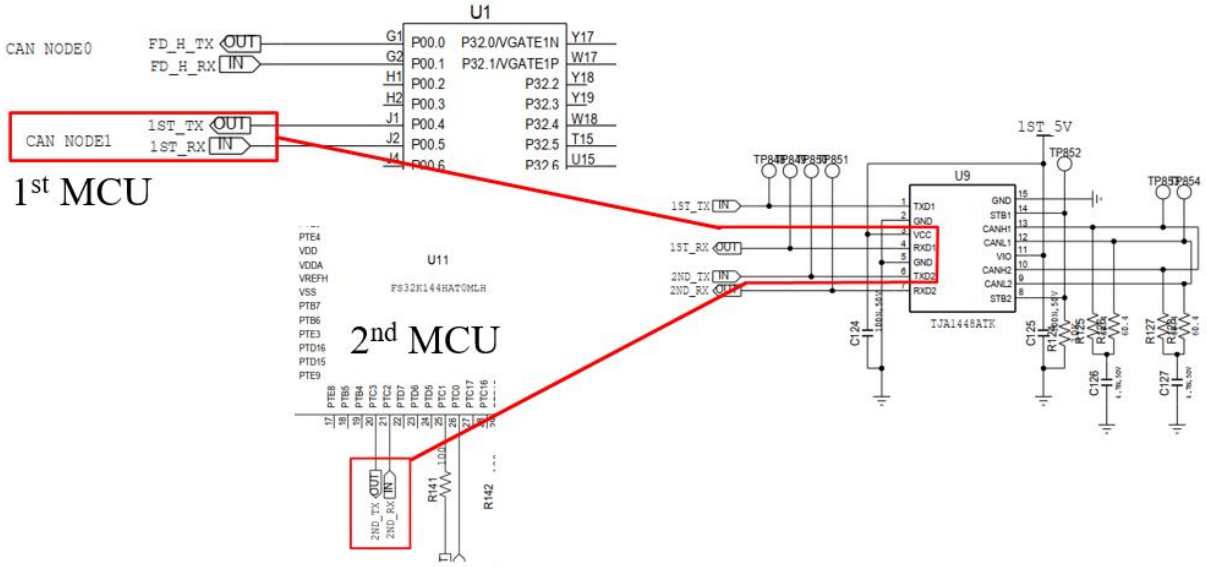
Functionality:

- Clock frequency - 20MHz
- Data transfer rate - 2Mbps
- No bit error should occur

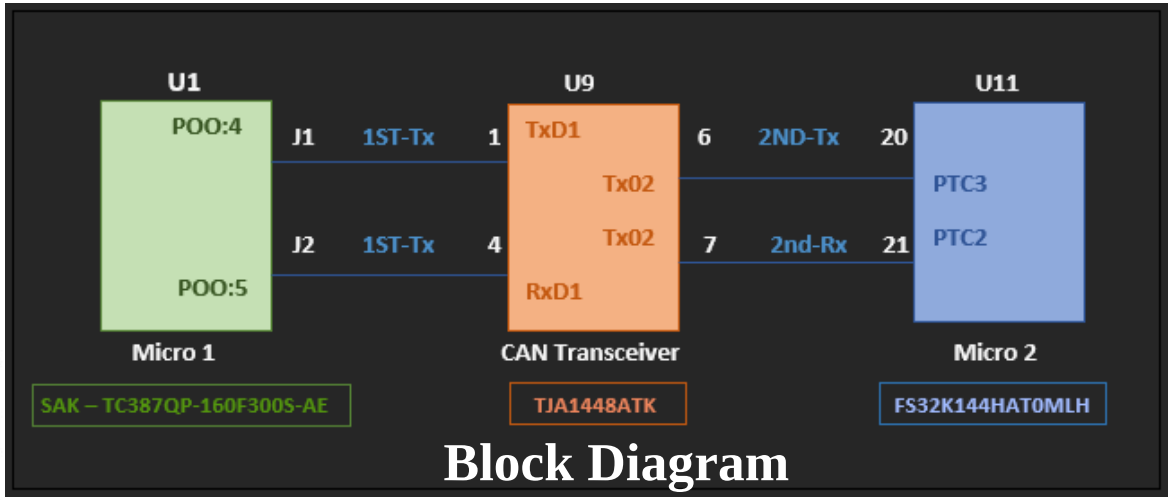


Every stages of work have multiple brainstorming and review with the client.

Existing Design Study



Existing Schematic

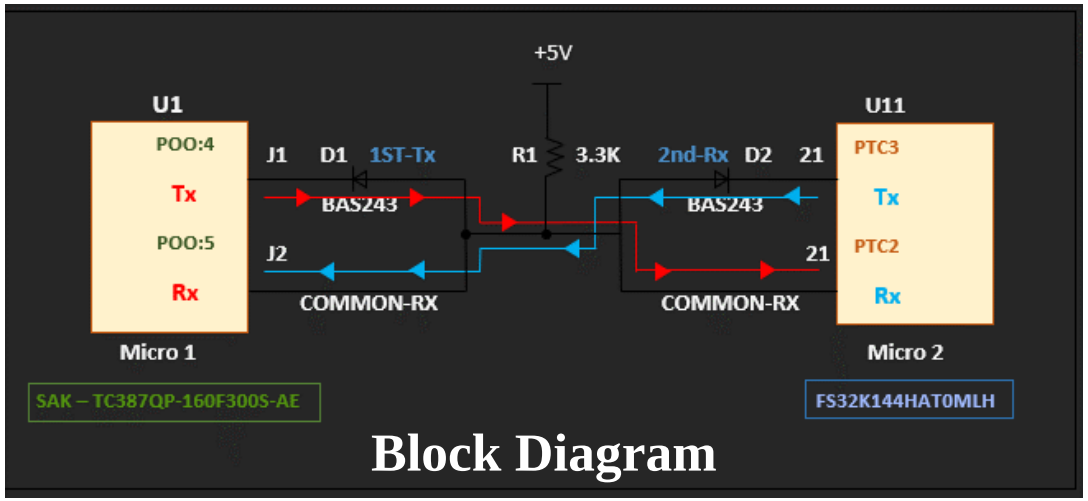


Block Diagram

Based on the Existing schematic study, the two microcontroller are connected through CAN communication, using CAN Transceiver to boost the signal.

In layout, two MCU are connected less than 6 inches which makes the possibility to remove the transceiver.

Proposed Circuit Design



Block Diagram

In the proposed design, the CAN transceiver is replaced with two diodes and resistor.

With this configuration the two wire combined to single wire.

Proposed Design Working

When the Tx of the micro1 send's data from the pin in open drain configuration, it makes the diode D1 forward bias and D2 reverse bias.

The data will flow from Micro1 to Micro2. After receiving the data, the micro2 will send the acknowledge bit which makes the D2 Forward bias and D1 reverse bias.

This process is vice versa for both microcontroller (Data flow from Micro2 to Micro1.

Proposed Circuit Simulation Preparation

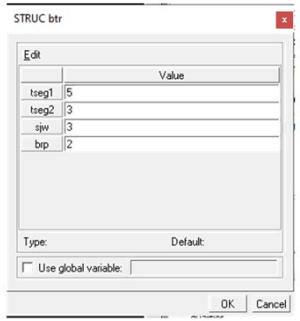
Design Inputs

- Frequency = 20Meg
- No. of Bits Send = 7
- Bit send → 0,1,0,1,0,1,0

Design Models

- Diode → BAS16LT1G(Spice Model)
- MCU1 → saxtc387qp160f300s_a(IBIS Model)
- MCU2 → kfa_64lqfp_5p0(IBIS Model)
- CAN Engine → Synopsys SaberRD(MAST Model)

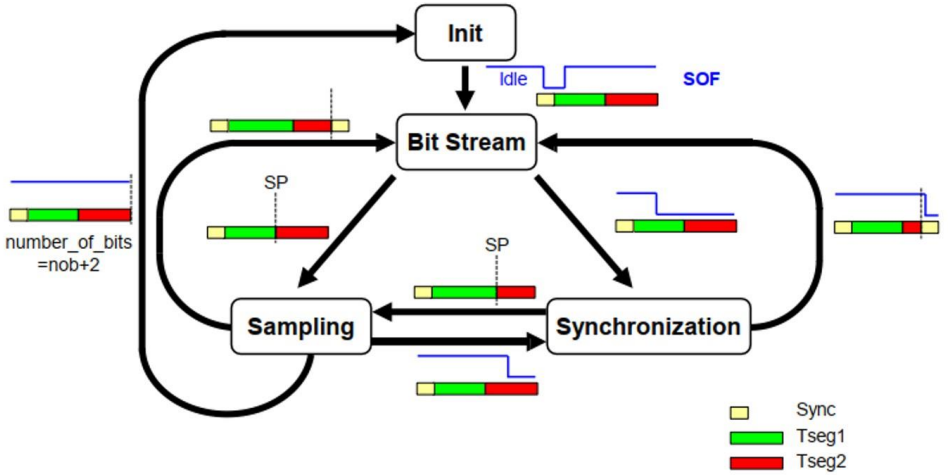
BTR Structure



Design Software

- Synopsys SaberRD Software

CAN BTL State Machine



Proposed Circuit Simulation Working

MCU1 → MCU2 CAN Communication:

During CAN transmission, the CAN engine generate 7 bits of data from the transmitter CAN engine (MCU1_Tx). These 7 bits need to receive at the receiver (MCU2_Rx). If the date is received, the acknowledge bit is transmitted from the received CAN engine (MCU2_Tx).

MCU1 → MCU2 CAN Communication:

The above process is repeated with MCU2 as transmitter and MCU1 as receiver.

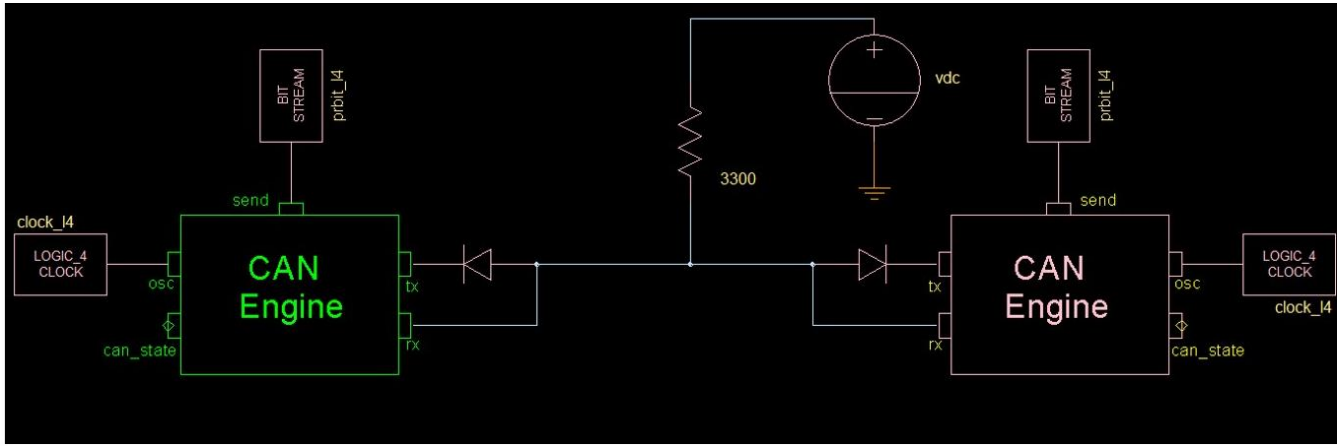
Ideal:

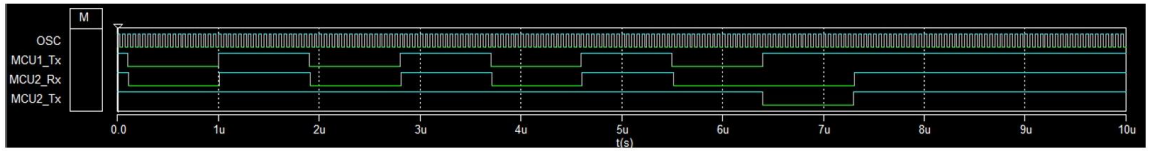
This setup utilizes only a resistor and 2 diode.

With IBIS:

The real time simulation result utilizes the MCU's IBIS model.

MCU1 → MCU2 CAN Communication:

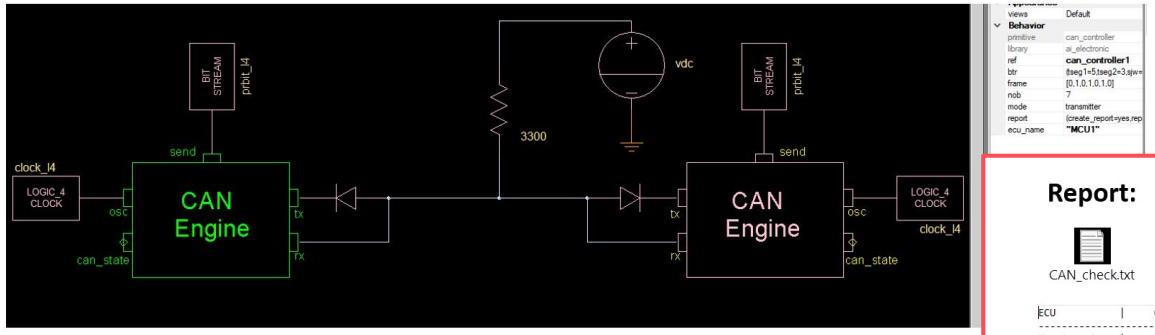




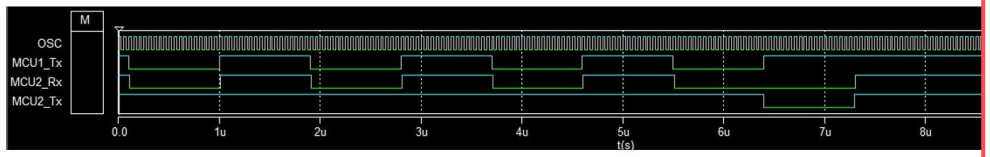
Proposed Circuit Ideal Simulation

MCU1 → MCU2 Ideal CAN Communication

Schematic:



Output:

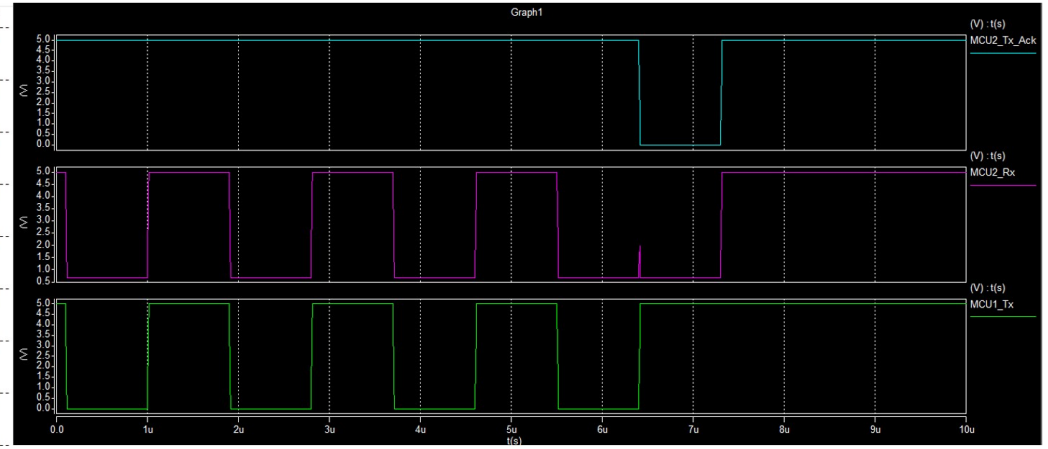


Report:

CAN_check.txt

ECU		0	1	Ref
Bit 1	value	0	0	0
	time	7e-07	7e-07	
	status	OK	OK	
Bit 2	value	1	1	1
	time	1.6e-06	1.6e-06	
	status	OK	OK	
Bit 3	value	0	0	0
	time	2.5e-06	2.5e-06	
	status	OK	OK	
Bit 4	value	1	1	1
	time	3.4e-06	3.4e-06	
	status	OK	OK	
Bit 5	value	0	0	0
	time	4.3e-06	4.3e-06	
	status	OK	OK	
Bit 6	value	1	1	1
	time	5.2e-06	5.2e-06	
	status	OK	OK	
Bit 7	value	0	0	0
	time	6.1e-06	6.1e-06	
	status	OK	OK	
Bit 8	value	0	0	0
	time	7e-06	7e-06	
	status	OK	OK	
Bit 9	value	1	1	1
	time	7.9e-06	7.9e-06	
	status	OK	OK	

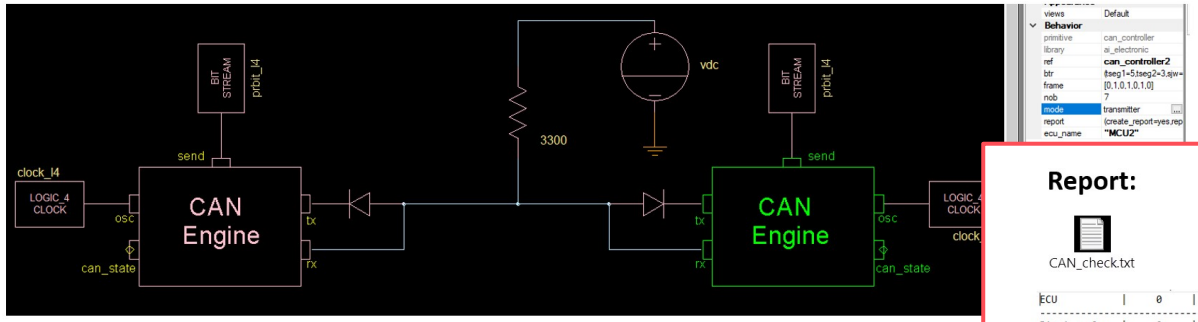
ECU label:
0 : MCU1 (1111111)
1 : MCU1 (0101010)
ECU with highest priority bitstream is MCU1



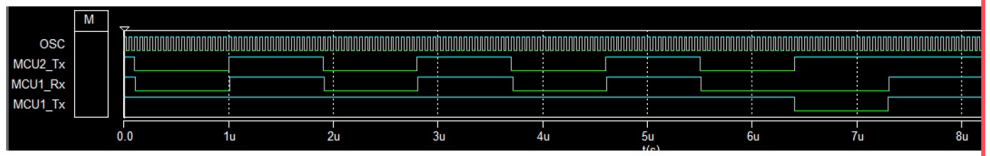
Proposed Circuit Ideal Simulation

MCU2 → MCU1 Ideal CAN Communication

Schematic:



Output:

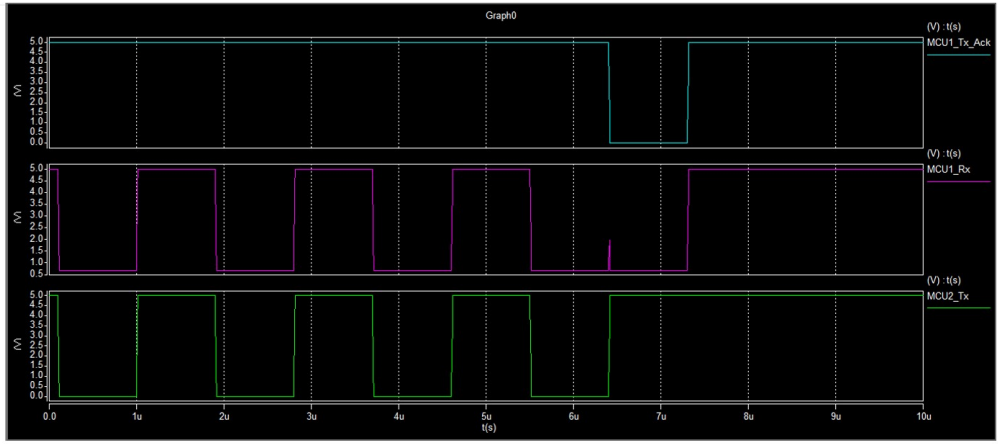


Report:

CAN_check.txt

ECU		0	1	Ref
Bit 1	value	0	0	0
	time	7e-07	7e-07	
	status	OK	OK	
Bit 2	value	1	1	1
	time	1.6e-06	1.6e-06	
	status	OK	OK	
Bit 3	value	0	0	0
	time	2.5e-06	2.5e-06	
	status	OK	OK	
Bit 4	value	1	1	1
	time	3.4e-06	3.4e-06	
	status	OK	OK	
Bit 5	value	0	0	0
	time	4.3e-06	4.3e-06	
	status	OK	OK	
Bit 6	value	1	1	1
	time	5.2e-06	5.2e-06	
	status	OK	OK	
Bit 7	value	0	0	0
	time	6.1e-06	6.1e-06	
	status	OK	OK	
Bit 8	value	0	0	0
	time	7e-06	7e-06	
	status	OK	OK	
Bit 9	value	1	1	1
	time	7.9e-06	7.9e-06	
	status	OK	OK	

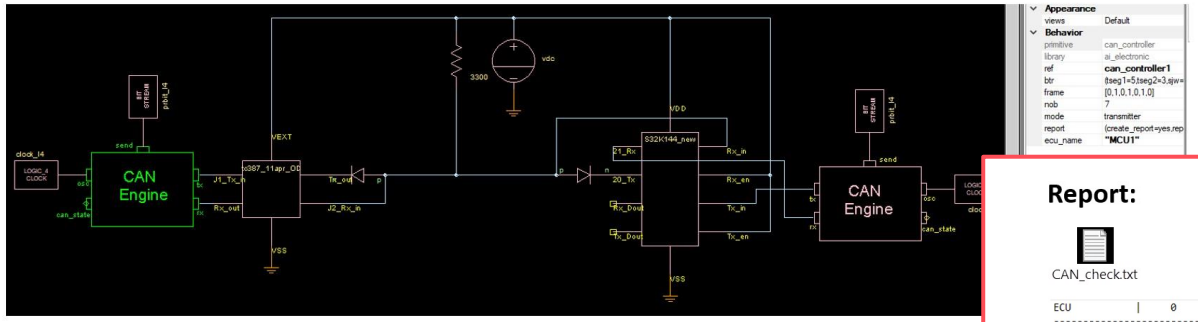
ECU label:
0 : MCU2 (0101010)
1 : MCU1 (1111111)
ECU with highest priority bitstream is MCU2



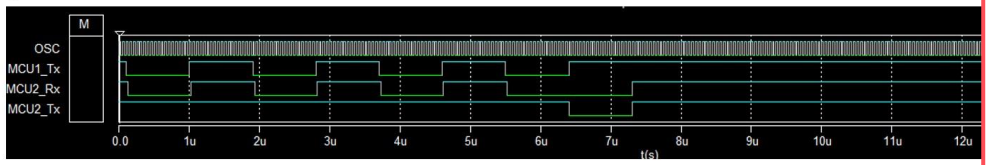
Proposed Circuit Simulation

MCU1 → MCU2 CAN Communication with IBIS models

Schematic:



Output:

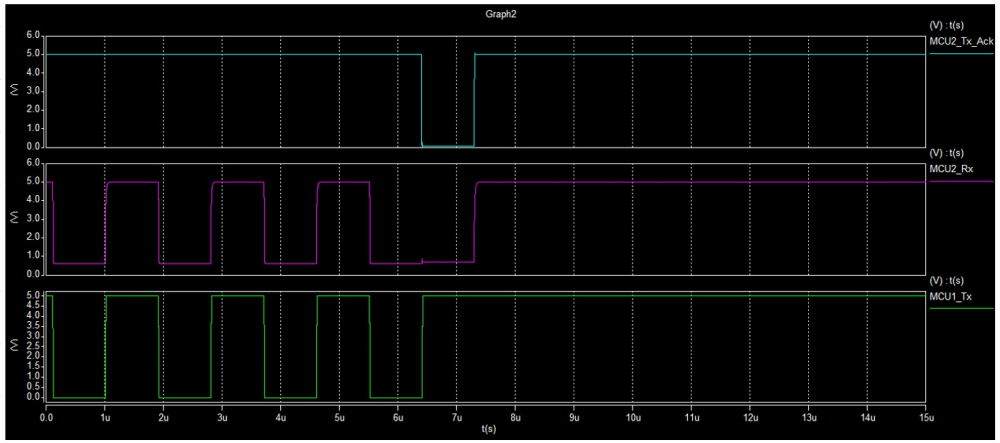


Report:

CAN_check.txt

ECU		0	1	Ref
Bit 1	value	0	0	0
	time	7e-07	7e-07	
	status	OK	OK	
Bit 2	value	1	1	1
	time	1.6e-06	1.6e-06	
	status	OK	OK	
Bit 3	value	0	0	0
	time	2.5e-06	2.5e-06	
	status	OK	OK	
Bit 4	value	1	1	1
	time	3.4e-06	3.4e-06	
	status	OK	OK	
Bit 5	value	0	0	0
	time	4.3e-06	4.3e-06	
	status	OK	OK	
Bit 6	value	1	1	1
	time	5.2e-06	5.2e-06	
	status	OK	OK	
Bit 7	value	0	0	0
	time	6.1e-06	6.1e-06	
	status	OK	OK	
Bit 8	value	0	0	0
	time	7e-06	7e-06	
	status	OK	OK	
Bit 9	value	1	1	1
	time	7.9e-06	7.9e-06	
	status	OK	OK	

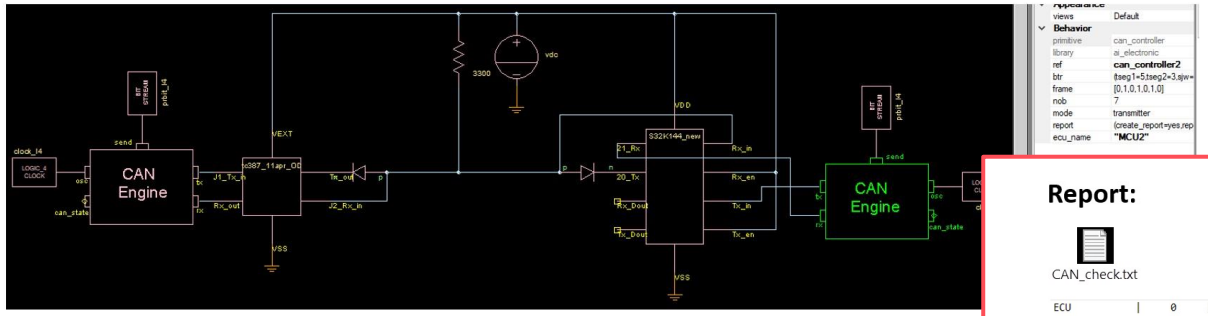
ECU label:
0 : MCU2 (1111111)
1 : MCU1 (0101010)
ECU with highest priority bitstream is MCU1



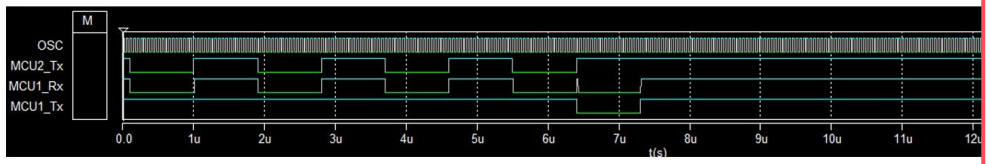
Proposed Circuit Simulation

MCU2 → MCU1 CAN Communication with IBIS models

Schematic:



Output:

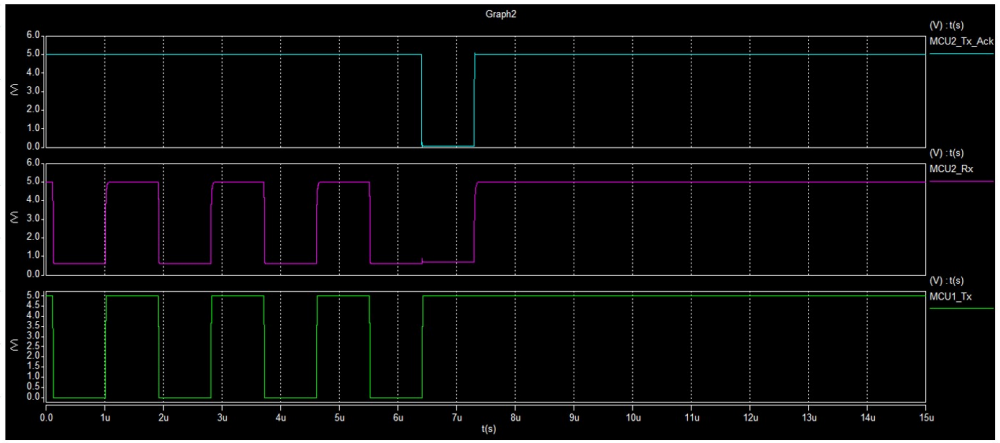


Report:

CAN_check.txt

ECU		0	1	Ref
Bit 1	value	0	0	0
	time	7e-07	7e-07	
	status	OK	OK	
Bit 2	value	1	1	1
	time	1.6e-06	1.6e-06	
	status	OK	OK	
Bit 3	value	0	0	0
	time	2.5e-06	2.5e-06	
	status	OK	OK	
Bit 4	value	1	1	1
	time	3.4e-06	3.4e-06	
	status	OK	OK	
Bit 5	value	0	0	0
	time	4.3e-06	4.3e-06	
	status	OK	OK	
Bit 6	value	1	1	1
	time	5.2e-06	5.2e-06	
	status	OK	OK	
Bit 7	value	0	0	0
	time	6.1e-06	6.1e-06	
	status	OK	OK	
Bit 8	value	0	0	0
	time	7e-06	7e-06	
	status	OK	OK	
Bit 9	value	1	1	1
	time	7.9e-06	7.9e-06	
	status	OK	OK	

ECU label:
0 : MCU2 (1111111)
1 : MCU1 (0101010)
ECU with highest priority bitstream is MCU1

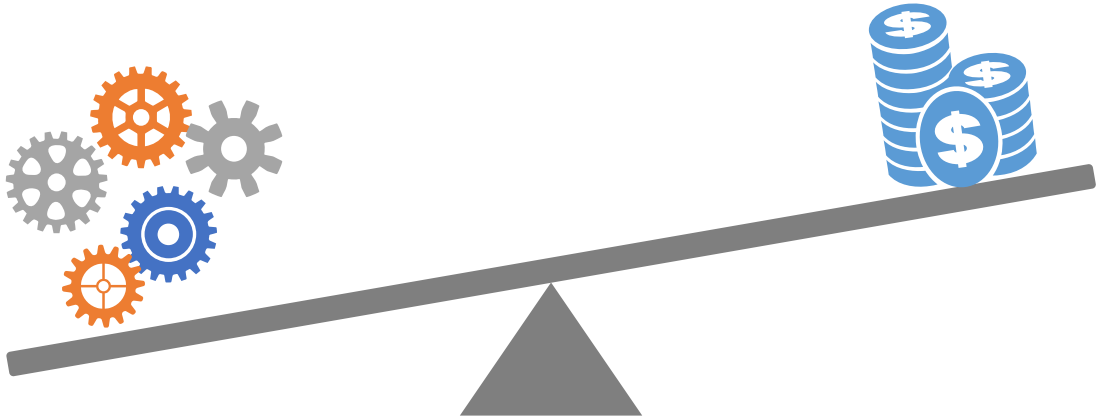


Result

From the simulation result, it is concluded that without CAN transceiver the data are transmitted without bit error.

This circuit can function without any error until 6 Mbps (Required – 2 Mbps).

This proposed design reduce the cost from $3.297 to $0.0464, which is 66 times less cost than the existing design.



COST

Existing				
			Unit Price	Final Price
Capacitors	100nF	2	0.06	0.12
Capacitors	4.7nF	2	0.17	0.34
Resistor	10k	1	0.03	0.03
Resistor	60.4	2	0.0573	0.1146
Resistor	128	2	0.0412	0.0824
IC	TJA1448	1	2.61	2.61
			Total Cost	3.297

Proposed				
			Unit Price	Final Price
Resistor	3300	1	0.013	0.013
Diode		2	0.0167	0.0334
			Total Cost	0.0464

GigHz
A Global Engineering partner

An ISO-27001 ISMS Certified Company

A Heartfelt Customer's Voice

I am truly impressed with the exceptional work and dedication demonstrated by the team in overcoming the challenges in our hardware design project. Their innovative solution of replacing the CAN Transceiver with a simplified circuit has not only maintain performance but also reduced costs significantly. The quick turnaround time in completing this design showcases their efficiency and commitment. This modified design aligns perfectly with our requirements, and we are delighted with the cost-effective yet unmodified performance solution provided. It's a significant milestone in our journey, and we look forward to continued collaboration with such a talented and reliable team.

Conclusion

- In summary, the challenge in create a hardware design, where we replaced the complex CAN Transceiver with a simpler circuit and analyzed real-time data transfer which aimed to reduce the overall cost by significant amount.
- By combining two wire to one which makes the PCB layout design better as it will be routed as 50ohm instead of 100ohm.
- Provided the cost effective design, without compromising the performance, which makes the product design best suited to the client requirement.
- With completing this design in short time period, marks a significant milestone in our journey.