EC01 - Floating Point Matrix Multiplication

Matrix multiplication lies at the heart of numerous applications, from computer graphics and image processing to machine learning and scientific simulations. However, its computational demands often become a bottleneck, hindering real-time performance and large-scale analysis. Dedicated hardware accelerators specifically designed for matrix multiplication offer a compelling solution. These accelerators leverage custom architectures and specialized components to achieve significant speedups compared to general-purpose processors. By offloading the computationally intensive tasks to such hardware, we can unlock faster results, enable real-time processing, and tackle larger and more complex problems, propelling advancements in various fields.

Rules and Regulations

1)Use of Multisim Software is recommended
2)The circuit must be created using basic elements of digital electronics only.
2)Other basic circuit requirements like transistors, resistors, clock, displays, switches, power supply etc. can be used, if required.
3)Use of microcontrollers is not allowed.
4)Input given should be converted to half-precision floating point format and the output should also be of the same format.

Problem Statement

Inputs : A matrix containing marks scored by 5 students in their Cycle Test-1(CT-1), Cycle Test-2(CT-2), Assignment and End semester exam with 20%, 20%, 10%, 50% weightages respectively.

Expected outputs:

Output-1: Floating Point Adder- (weightage -10%).
Output-2: Floating Point Multiplier (weightage -15%).
Output-3: Average of End Semester marks of all the students (weightage -25%).
Outputs -4: (weightage -50%).
A matrix consisting of the consolidated final marks of each candidate out of 100.

Bonus :
Check if the empirical relationship between mean, median and mode is followed (approximately)
Mean – Mode = 3 (Mean – Median)

Test Case:

- For Floating Point Adder and Floating Point Multiplier, the inputs can be of your choice. (like 55.6+66.3 and 69.3*56.0, do not use integers in your test case)
- For Output 3 and output 4 use inputs from the table given below

Name	CT-1	CT-2	Assignment	End Semester
Student-1	56.7	97.6	35.5	55.3
Student-2	58.3	85.4	41.2	57.2
Student-3	59.9	76.2	46.9	59.1
Student-4	61.5	61	52.6	61
Student-5	63.1	48.8	58.3	62.9

Submission Guidelines

Submission should include (Create a drive folder with the following files):

1. Circuit link or file
2. A full circuit downloaded image at high resolution
3. A pdf explaining your logic (with necessary screenshots).
4. A video (screen recording) containing simulation of your circuit by testing it with the provided test case. (Maximum duration: 5 minutes)
5. Approach and other essential information about your circuit that might be important for evaluation.

Partial submissions are also accepted. Remember:
The will to participate matters :)

Please follow the below folder structure:
Root folder name: TRINIT_<TeamName>_EC<ProblemCode>
Sub-folders:
TRINIT_<TeamName>_EC01/SimulationFiles –
All circuit simulation files to be kept here.

If you're using an online simulator like circuitverse, add the circuit link
in a word/google doc.
TRINIT_<TeamName>_EC01/CircuitExplanation –
The PDF explaining your logic (with necessary screenshots) and high-resolution image of the circuit.
TRINIT_<TeamName>_EC01/DemoVideo – Video(s)
(Screen recording) demonstrating the simulation
(maximum duration: 5 minutes total).
Note: Don't forget to update the root drive folder permissions to "viewable by anyone with the link". Failing to do so can result in your
disqualification.
Submitting on D2C:
Create a public github repository with name- "TRINIT_<TeamName>_EC<ProblemCode>" and upload the drive link in the repository's README. You are free to include any extra files/explanations on the repository. Submit the GitHub repository
link on D2C.

Resources:

IEEE Standard 754 Floating Point Numbers
https://www.geeksforgeeks.org/ieee-standard-754-floating-point-numbers/

Digital Electronics Basics
https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/

Multipliers
https://inst.eecs.berkeley.edu/~eecs151/sp18/files/Lecture21.pdf

Adders
https://ieeexplore.ieee.org/document/7164650

Multisim Software Installation
https://youtu.be/no9tLtRsOqQ?si=R68Im07qsK1mM20A

EC02 - Analog Equalizer circuit

The transmission of high-speed data on copper suffers from frequency-dependent losses that can cause significant intersymbol interference. The losses mainly arise from the increase in series resistance at high frequencies due to the skin effect. There are also dielectric losses, which are significant in the GHz range and higher. To correct for frequency dependent losses, a good Equalizer circuit is used in order to get the desired gain back.

Problem Statement:

Have a scenario where you are an Analog design intern working in CERN , switzerland. The operating area of the particle accelerators are massive and the data collected at one end should be processed by the supercomputers which are 12 Kms far. Since the distance is too long , the cables used attenuates the signal at certain frequencies and corrupts the actual signal that's

needed. Your job is to design a continuous time analog equalizer circuit which processes the attenuated signal and nullifies the effect of transmission wire to get the actual input signal back.

The below circuit is the model for the transmission wire. Assume that the cables located at CERN and the circuit below give the same frequency response. Design an EQ circuit for this corresponding system to get a flat frequency response which also matches the given specifications.

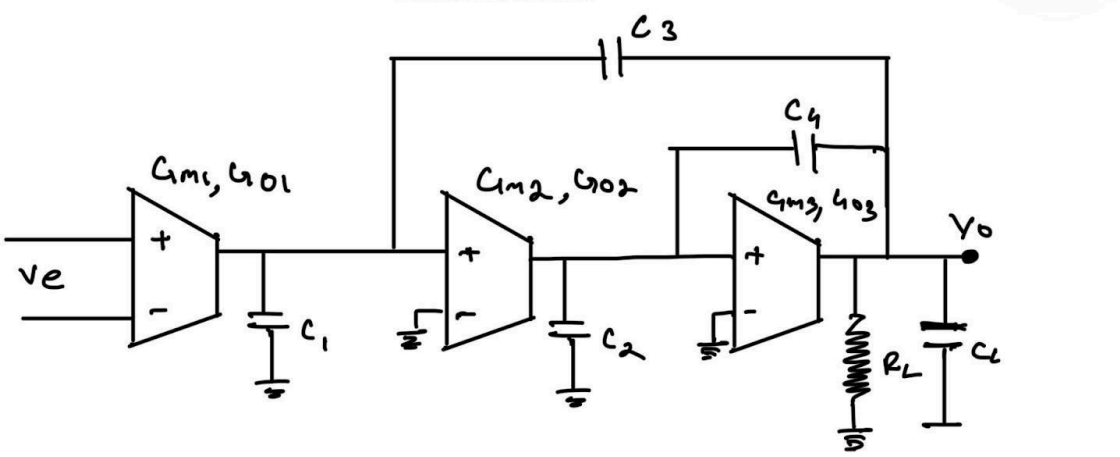


Image 1

Circuit description

The above circuit is a three stage op amp built using transconductance amplifiers with transconductances Gm1 , Gm2 , Gm3. The amplifiers have finite output impedances at the output node whose values are given in terms of conductances Go1 , Go2 and Go3. They also have capacitors C1 , C2 , C3 along with finite impedances which are tied at the output node. C3 and C4 are the miller compensated capacitors whose values are given below. Find and study the behavior of this system considering Vo as output and Ve as input differential voltage.

Hint : Find the frequency response of the above circuit by using the fundamental laws of circuit theory.

The values given for the parameters above are:

10Gm1 = Gm2= 5 Gm3 = 200 micro Siemens
10Go1= Go2 = 20 Go3 = 2 micro siemens
C1 = 2C2 = 6 picofarad
Load capacitance C_L = 10 picofarad
Load resistance R_L = 125 kilo ohm
C3 = 2C4 = 1 picofarad

Rules and Regulations

1)The circuit must be created using basic elements of Analog electronics only.
2)Op amps are not allowed and the use of MOSFETS are necessary. The whole circuit should be a MOSFET based circuit
3)Use of microcontrollers is not allowed.

Circuit specifications:

1) The whole system should give a flat frequency response.
2) Include a power management circuit which shuts down the circuit entirely when the output power of the equalizer crosses certain threshold (fix your threshold according to the VDD used and don't forget to mention your threshold in the doc)
3) Your circuit should consume minimum power and should have less loading effect

4) Your Equalizer circuit should be able to remove the noise from the input signal too. (Hint : think of an amplifier which can remove noise)

Evaluation Criteria:

1. How flat is your system(given circuit and Equalizer circuit combined) response
2. How power efficient is your circuit
3. Working of power protector circuit
4. Your study of the given circuit (image 1)
5. The use of MOSFETS in the circuit

Submission Guidelines:

Submission should include (Create a drive folder with the following files):

1. Circuit link or file
2. A full circuit image downloaded at high resolution
3. A pdf explaining the working of your circuit (with necessary screenshots).
4. Attach the calculations made to study the frequency response of the given system and specify the pole and zero locations
5. The frequency response image of the given circuit (image 1) and the frequency response of your equalizer circuit.
6. Finally, attach the frequency response image of the whole system (given system and your EQ circuit combined)
7. Include a video with a voice over which shows the entire simulation of frequency response of the system and also the working of the power management circuit.

8. Any other document which would help in evaluation are welcomed

Partial submissions are also accepted.
Remember: The will to participate matters :)

Software to be used : LT spice.
You are also free to use open source online tools to get the bode plot etc.

Please follow the below folder structure:
Root folder name: TRINIT_<TeamName>_EC<ProblemCode>

Sub-folders:
TRINIT_<TeamName>_EC02/SimulationFiles –
Add all your spice simulation files here.

TRINIT_<TeamName>_EC01/CircuitExplanation –
The PDF explaining your logic (with necessary screenshots) and high-resolution image of the circuit.

TRINIT_<TeamName>_EC01/DemoVideo – Video(s)
(Screen recording) demonstrating the simulation

(maximum duration: 5 minutes total).

Note: Don't forget to update the root drive folder permissions to "viewable by anyone with the link".

Submitting on D2C:
Create a public github repository with name- "TRINIT_<TeamName>_EC<ProblemCode>" and upload the drive link in the repository's README. You are free to include any extra files/explanations on the repository. Submit the GitHub repository
link on D2C.

Resources:

Basics of bode plot:
https://www.youtube.com/playlist?list=PLUMWjy5jgHK24TCFwngV5MeiruHxt1BQR

Frequency response in Spice:
https://www.youtube.com/watch?v=I-y03jwWq-U

Transconductance amplifiers:
https://www.eeeguide.com/transconductance-amplifier/

Basics of frequency response:
https://ee.sharif.edu/~faez/ch14.pdf

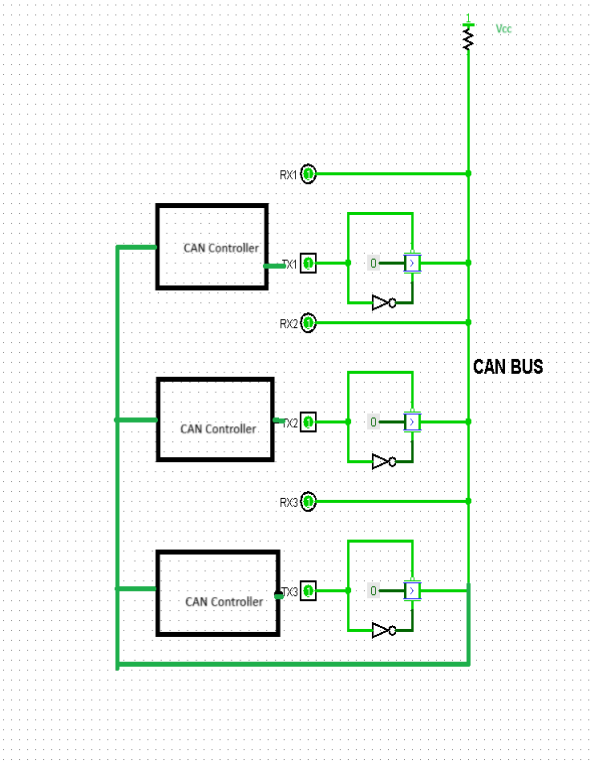
To get an idea of Equalizer circuit:
https://www.homemade-circuits.com/10-band-graphic-equalizer-circuit-for/

https://www.electroschematics.com/3-band-equalizer/

As always , you have an abundance of resources on the internet. Use it wisely :)

EC03 Topic: Implementing Controller Area Network in CAN 2.0B Format Controller

In the automotive industry's evolution, the shift from point-to-point wiring to in-vehicle networks revolutionized electronic device connections, addressing the challenges of bulky, costly wire harnesses. The emergence of CAN, a high-integrity serial bus system, became the standard for networking intelligent devices, leading to the adoption of ISO 11898 in 1993, enhancing efficiency across various markets.



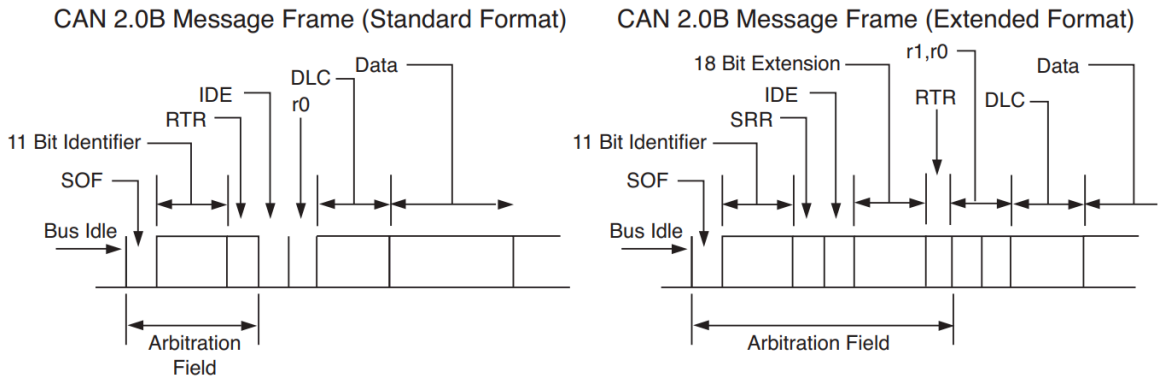
Problem Statement: The problem statement is to design a single-bus CAN digital controller for the given .cir file in the diagram consisting of a CAN bus and three sets of Transmitters and receivers(the CAN bus implemented In the diagram has logic 0 as dominant and logic 1 as recessive state like CAN physical layer).

- Design and Instantiate a CAN controller for each
  transmitter and receiver pair
- The CAN controller processes input messages and the receiver's identifier while monitoring the CAN bus to determine availability. Upon accessing the bus, it generates the message frame for transmission.
- Create a receiver controller that only accepts the CAN 2.0B message having the correct identifier
- Make an arbitrating mechanism if the bus is already being accessed by some other transmitter

Software Used:

LogiSim (Find download link in Resources)

Implementation Rules:



- Use logic 0 as the dominant bit and logic 1 as the recessive
- Only one bit is transmitted at once
- Each message frame must include **SOF**, **Identifier**, (optional) **IDE**, **Data Length(DLC)**, **Data field**, (optional)**CRC**, (optional) **CRC delimiter**, (optional) **ACK**, (optional) **ACK delimiter**, and **EOF**.
- Importing any kind of HDL library is not allowed.
- No hard coding is allowed except for the contents of the state diagram. Actual convolutional code encoding and decoding must take place.
- Good design practices are expected, such as modularising your circuit with subcircuits
- All sequential elements must trigger on Rising Edge.
- All elements must be active high, including asynchronous sets and resets.

Test Cases : (All TEXT should be encoded in utf-8 bytes)

Case 1: Standard ID transmission from TX1 and reception at RX2 or RX3
message.identifier =0x1AB, message.DLC=6. message = “TRINIT”

Case 2: **Extended ID Test transmission from TX1 and reception at RX2 or RX3**
message.identifier =0x10032024, message.DLC=8. message = “TRINIT24”

Case 3: **Arbitration implementation test simultaneous transmission from TX1,TX2, TX3 and reception sequence at RX1 RX2, RX3 is observed should follow the standard arbitration rules**

TX1 message
message.identifier =0x001, message.DLC=6. message = “ONENIT”

TX2 message
message.identifier =0x002, message.DLC=6. message = “TWONIT”

TX3 message
message.identifier =0x003, message.DLC=6. message = “TRINIT”

Submission Rules:

1. You must submit the “.circ” circuit file.
2. Provide the steps to simulate your circuit in a .txt file.
3. Provide a .txt file containing the outputs in the form of English alphabets.
4. Provide a .pdf file explaining the logic behind your circuit. Attach any paperwork like state tables, etc., if needed. Also, describe all subcircuits used in your circuit.
5. Provide a screen recording of you simulating the input message

Brownie Points

1. Implement the option to switch to CAN 2.0B extended format by using the recessive value in the IDE field of the message frame
2. Implement an option to change the data length arbitrarily in the DLC field of the message frame

3. Implement the CRC and ACK mechanism for error handling and successful recipient of messages.
4. Implement a CAN filter and mask mechanism to lighten the burden on the receiver and only listen to the message intended for it

References

1. https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf
2. LogiSim download link