EE 344 Electronic Design Lab

Discrete Analog to Digital Converter

April 21, 2019

Faculty Mentor - **Prof. Pramod Murali**

Group Details - **DD-13**

Kumar Ashutosh (16D070043)

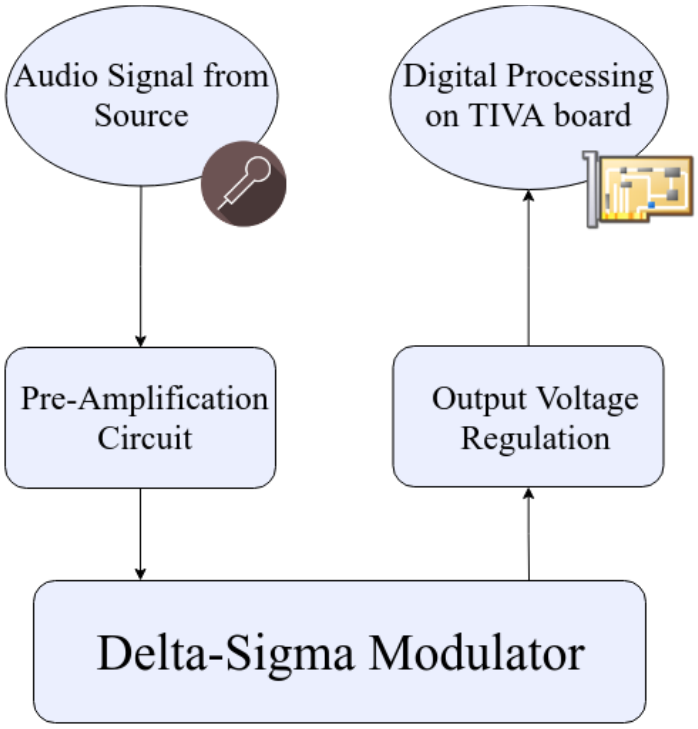
Beni Madhav Agrawal (16D070040)

Yashvardhan Didwania (16D070055)

Objectives -

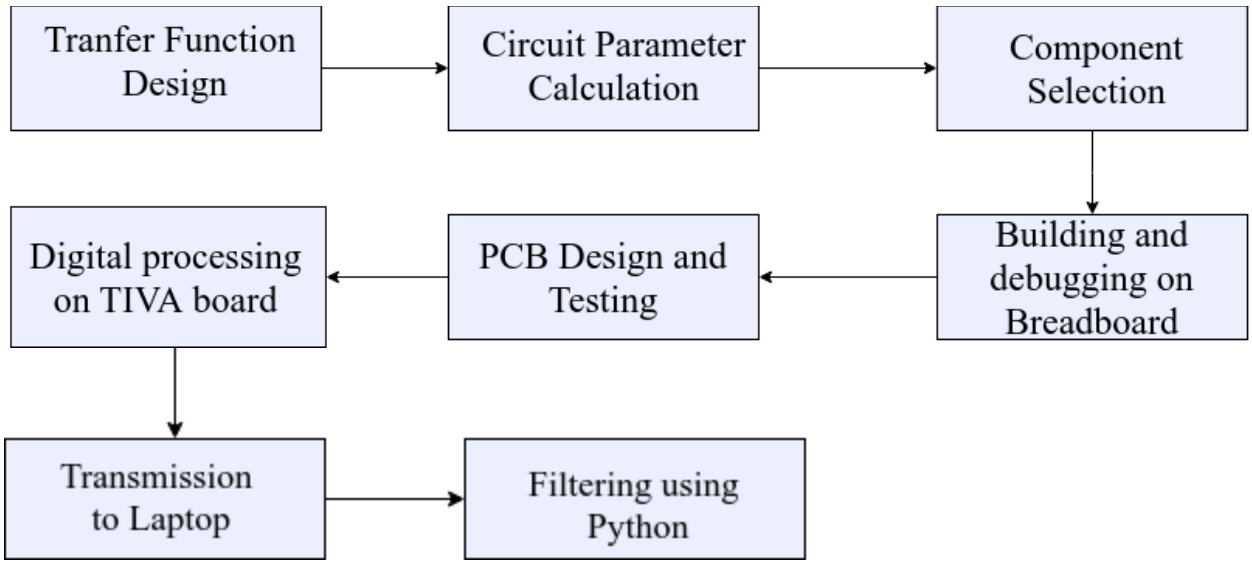
- To design a Second Order Delta-Sigma Modulator.
- Demonstrate the Delta Sigma Modulator for typical audio signals.
- Transmit the bit stream to laptop using TivaC in real time.
- Capture and decimate the received bitstream using Python.
- Obtain a digital version of the Analog Audio Input.

Block Diagram -



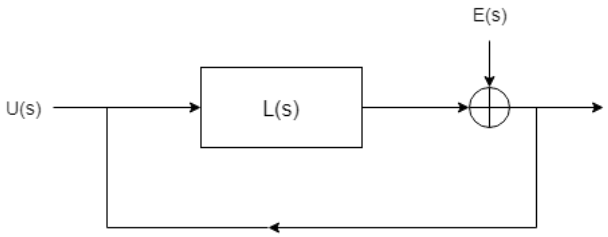
Approach and Workflow -

Each Major component of this workflow is explained in detail in the next sections.



Transfer Function Design -

We used the following Feedback Model to design our circuit:-



Here the sampler (a cascade of Comparator and D-Flip Flop) is modelled as a Noise Input at the output end.

The desired specification of the Delta-Sigma Modulator is chosen as -

- Over-Sampling Ratio - 32
- Order - 2
- Out of Band Noise Magnitude - 1.5

These specifications yield a Noise Transfer Function as follows -

$$NTF\ (z)\ = \frac{(z-1)^2}{z^2\ -\ 1.225z\ +\ 0.4415}$$

This NTF(z) is used to find the signal transfer function S(z) as -

$$STF\ (z) = \frac{0.77485\ (z-0.7208)}{(z-1)^2}$$

Finally, the continuous time Signal Transfer Function L(s) is -

$$L(s) = \frac{\frac{2}{3}\ (s+0.3246)}{s^2}$$

Design Specifications

- **Frequency of Operation -** We chose 1MHz as the clock frequency. This was done to ensure that the clock can be provided by Microcontrollers easily.
- **Oversampling Rate** - The OSR of 32 was chosen so that it is easier to cascade multiple decimation filters to obtain the desired downsampling by 32 after attaining the bitstream. The ratio of clock frequency and OSR gives 31250 Hz which is well above twice the normal audible range.
- **Order of the Delta-Sigma Modulator** - In literature, there are higher order filters but we settled for second order as a tradeoff between simplicity of implementation and the quality of the output waveform. But higher order Delta Sigma Modulator also increases the Non-linearity due to Op-Amps.

Circuit Design to Realise L(s) -

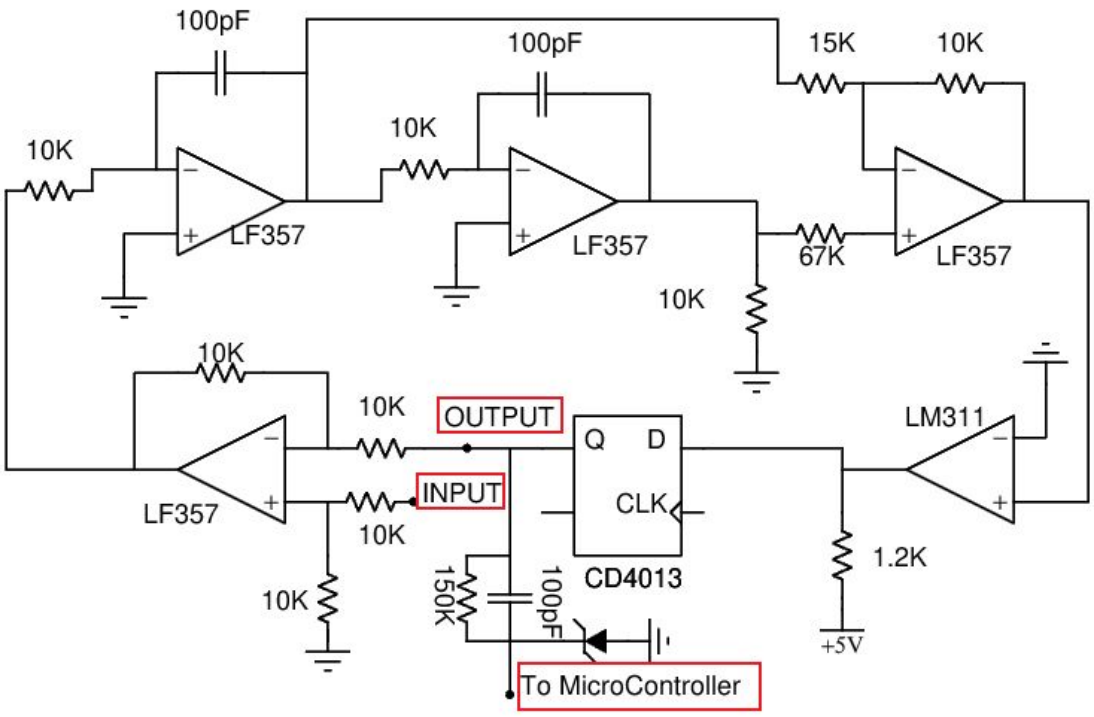
The above-designed transfer function is realized using two integrators in cascade with the outputs from them added to obtain the transfer function of the form

$$L(s) = \frac{a}{s} + \frac{b}{s^2}$$

Also, our cutoff frequency is **1MHz**, hence in both the integrators

$$\frac{1}{RC} = 10^6$$ and thus **R = 10k Ω and C = 100pF**

Other resistor values are chosen so as to satisfy the above transfer function.



Choice of Components -

- **LF357(Opamp) :**
  - **Fast Slew Rate:** 50 V/us
  - **Wide Gain Bandwidth:** 20 MHz
- **LM311(Comparator) :**
  - **Fast Response Time:** 165 ns
  - Controllable slew from pull-down resistor, this open drain IC is isolated from the supply voltage, so we can have desired 5V maximum output.
- **CD4013(D Flip-Flop):**
  - **Good Operation speed** 30ns
  - **Maximum operation speed:** 16Mhz, way higher than required.

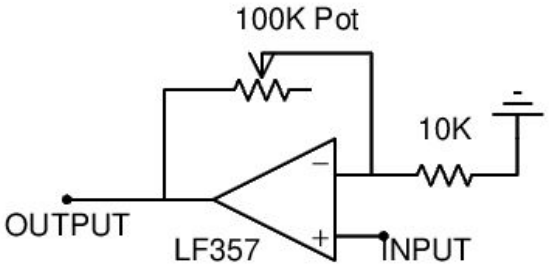
Other Peripheral Circuits Designed -

- **Input Voltage Regulator**

  Since for our circuit, we need +12, +5, GND, -5 and -12, it would be untidy to have all these wires coming out from the Power Supply. Hence we designed a power regulator circuit using **7805** IC which regulates +12 to +5 and a **7905** IC which regulates the -12V to -5V.

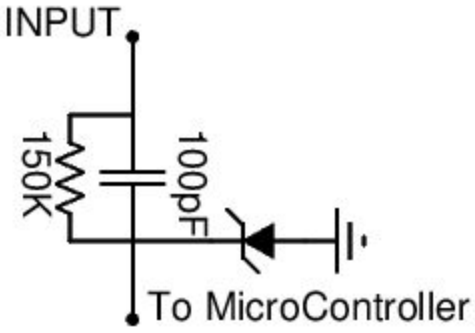
- **Input Audio Amplifier**

  We use an **LM357** opamp and a resistance pot to vary the magnitude of the input voltage before putting it into the delta-sigma circuit.



- **Output Voltage Regulator**

  Since the Microcontroller receives voltage in the range 0 to 3.3V, hence we use a Zener diode to regulate +5V to +3.3V and -5V to -0.7V.



Digital Processing of Bit-Stream -

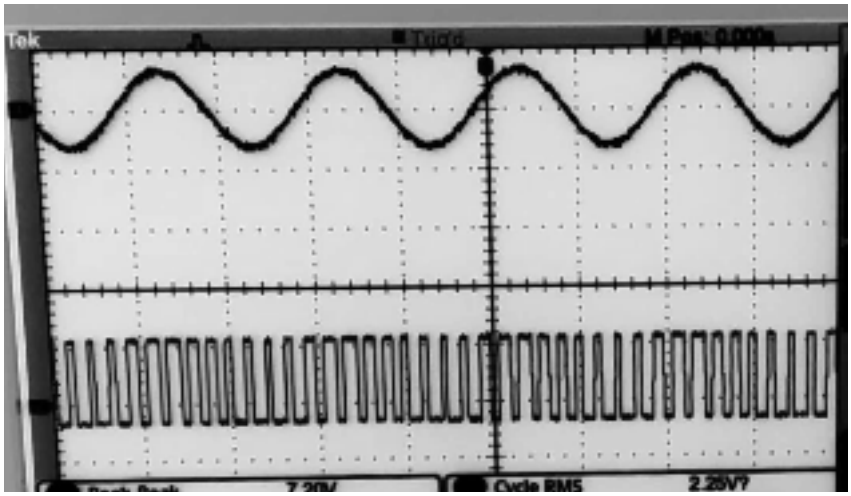
- The task was to sample the bits arriving from the DSM at 1MHz and transmit them to the PC for further filtering and playing the audio. We decided to use TivaC against an FPGA because of the great documentation available online for TivaC.
- We used the TM4C123GH6PM TivaC board as it offered a top operating clock speed of 80MHz. As conventional sampling and storage in a buffer would be constrained in 80 clock cycles at best, it was unreliable to use.
- Hence, we decided to use SPI for storage in a 1 Master and 2 Slave configuration as follows -
    - Master clock of 1MHz and 31.25KHz square waves as FSS1 and FSS2.
    - FSS1 = -FSS2 => The two slaves ports on TivaC sampled and packed 16bit words alternatively from the DSM which was fed as the Master data
    - This was done because we observed 3 out of 16 bits were being dropped by an SPI Slave when a FSS from SPI Master port on TivaC makes a transition.
    - In the above configuration, the sample drop was 1 out of every 64 bits (avg).
- After collecting the bits using SPI, they were transferred to the USB buffers which sent it to the PC at ~50Mbps. A python script captured these bits and stored them in a file.
- **Python Script for audio regeneration**

  Once we have the bitstream stored in a text file, we have another script which reads the file sequentially and filters the output using a predefined FIR filter designed in MATLAB. This produces a bitstream of 1MHz frequency. To store the filtered output as a text file, we decimate the bitstream and save the data using **scipy** audiosave command at a sampling frequency of around 40kHz.

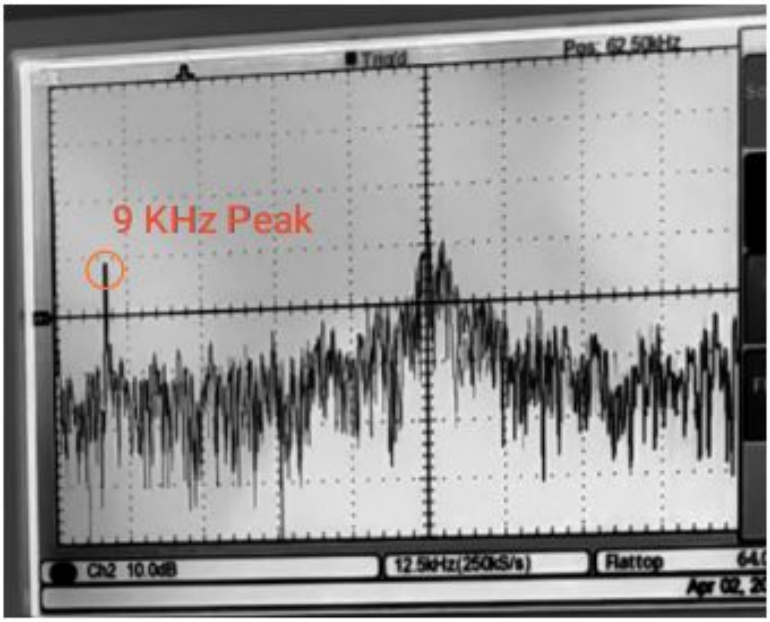
Results -

Input Frequency: 9 KHz

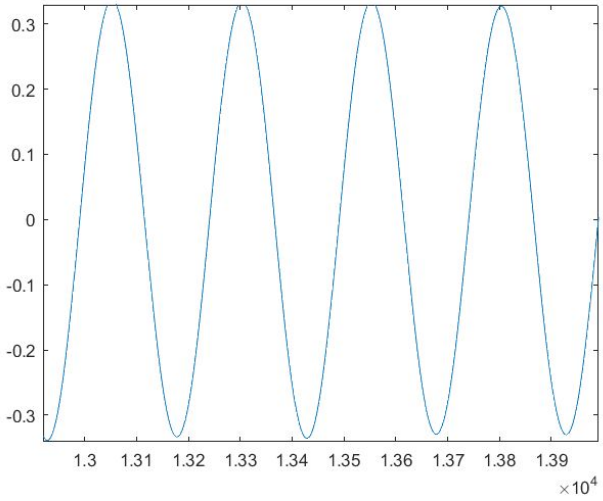
Output voltage swing: 0 - 3.3V



Fast Fourier Transform on DSO:



The reconstructed sine wave on MATLAB:



The 9 kHz peak is clearly visible in the FFT of the output signal. There is also considerable noise shaping which is evident as the noise is comparatively low in the frequencies of interest and much higher outside it. This is one of the major advantages of oversampling.

Output Files:

The output reconstructed audio along with an image of PCB circuit can be found on the following Drive Link:

https://drive.google.com/drive/folders/1DmehXvaA4U-2NN3OT27tKZqO7YZu2l8w

Problems Faced:

- Slew in Comparator: The Comparator LM311 though quite fast, displayed slew at 1MHz signal.
    - Reason: Pulling down the output from +12V required even faster voltage changes.
    - Solution: Pulling down from +5V instead of +12V and reducing the value of pulldown resistance to 1.2K.

- Tiva Board Input Signal voltage: Tiva Board Requires Input voltage in range 0-3.3V, our output was -5 to +5V.
    - Solution: Voltage regulation circuit using Zener diode as mentioned in the circuit diagram.
- Audio input offsets: There were some DC offset in the Audio input initially which caused the output of the open loop block to saturate, thus we used a DC blocker for the audio input from PC.
- The high sample drop rate in SPI was underestimated initially because of which we had to redesign using the 2 SPI Slave configuration with alternating FSS1 and FSS2.
- Stellaris provides USB drivers for receiving information only for Windows. Since we were on Linux, we had to write the code for receiving the bits from the USB port using several software threads.