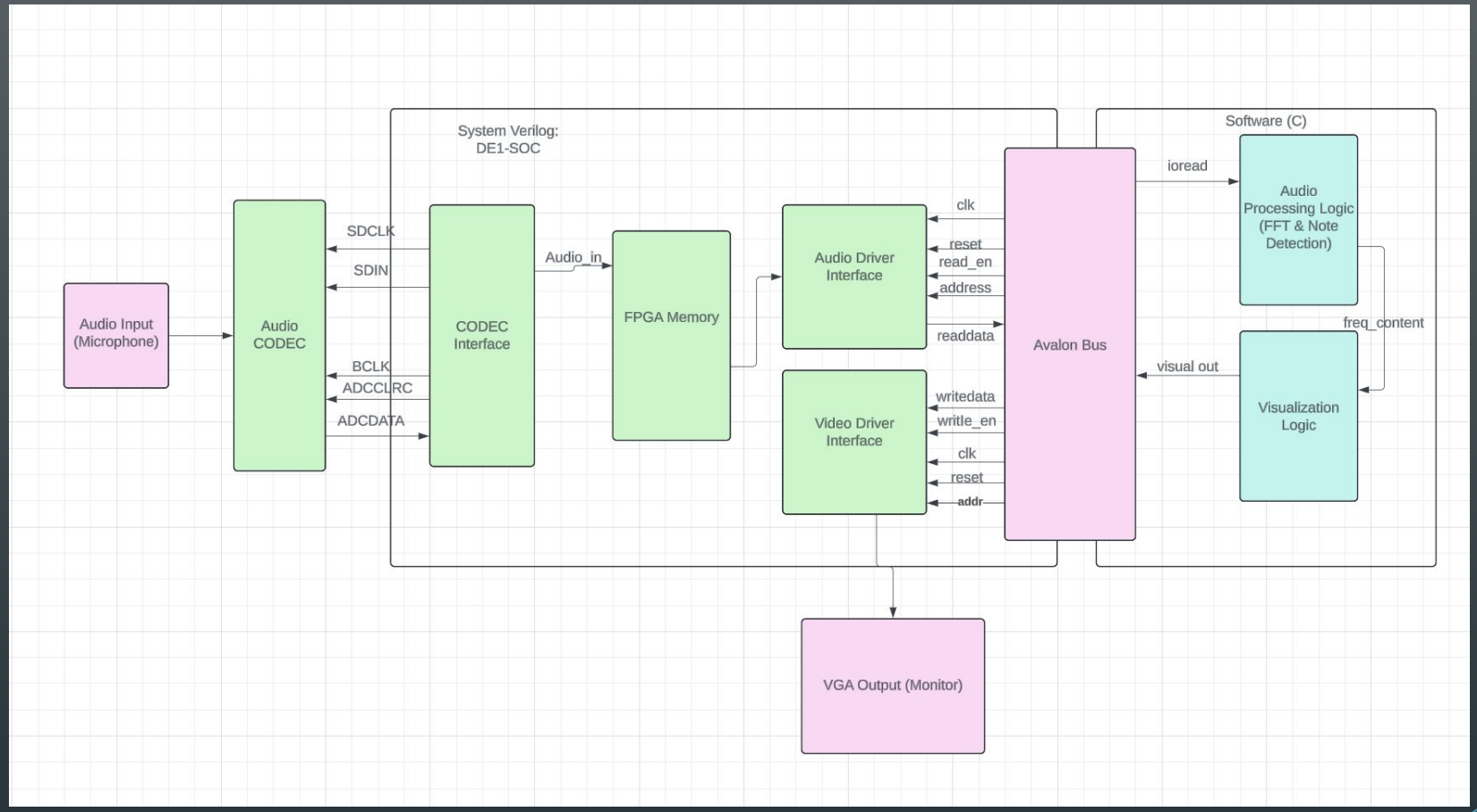
CSEE-4840
Audio Visualizer

Manas Pange, Yaagna Modi, Gaurav Agarwal, Max Lavey

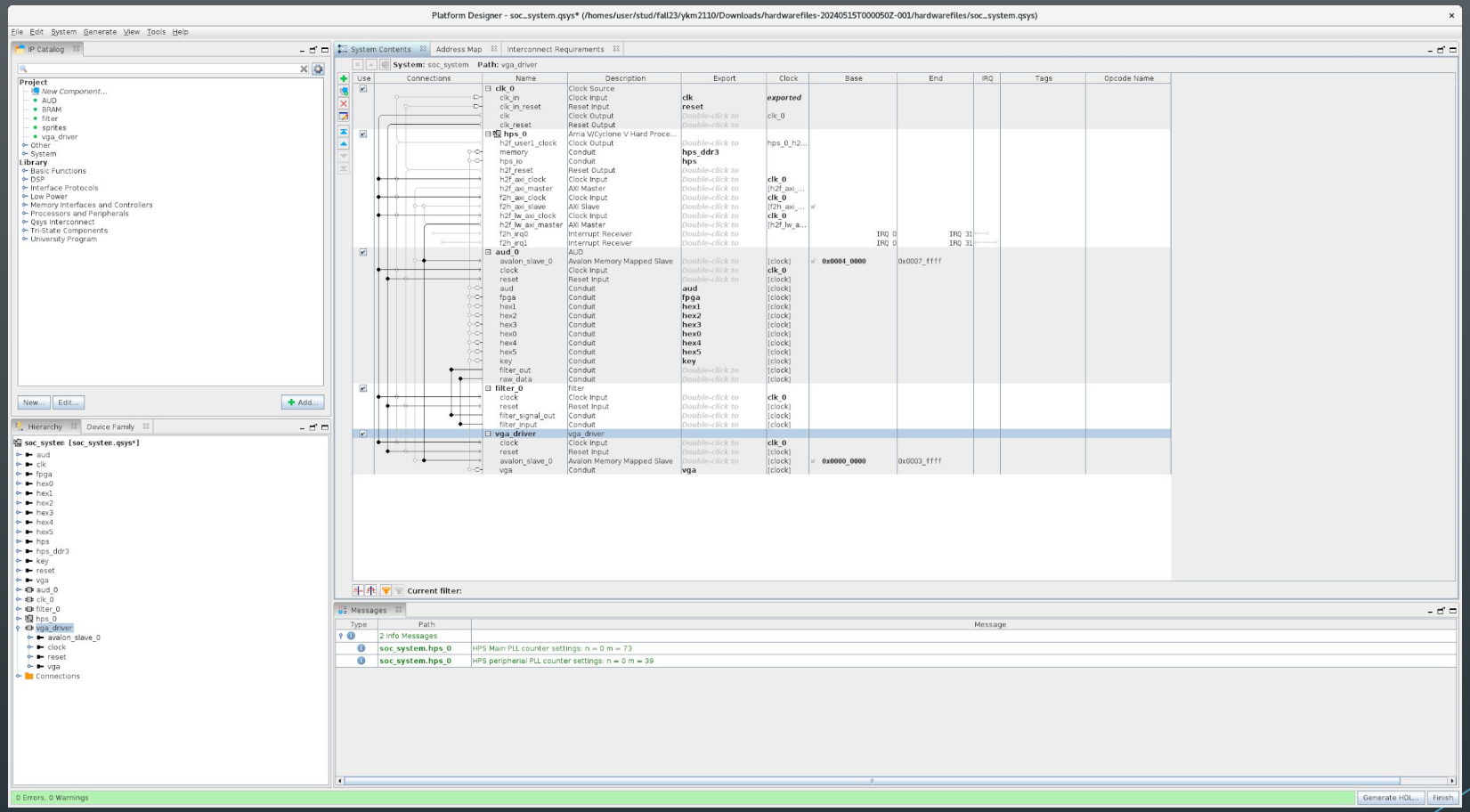
Project Specifications

- The goal of our project is to capture notes from piano and then do FFT on the input data to identify the note that is being played.
- We are using HIYANCO microphone and connecting it to ‘mic in’ input port on FPGA board.
- Convert the analog data into the digital data and save it in an output buffer.
- The sampling rate is 48 kHz.

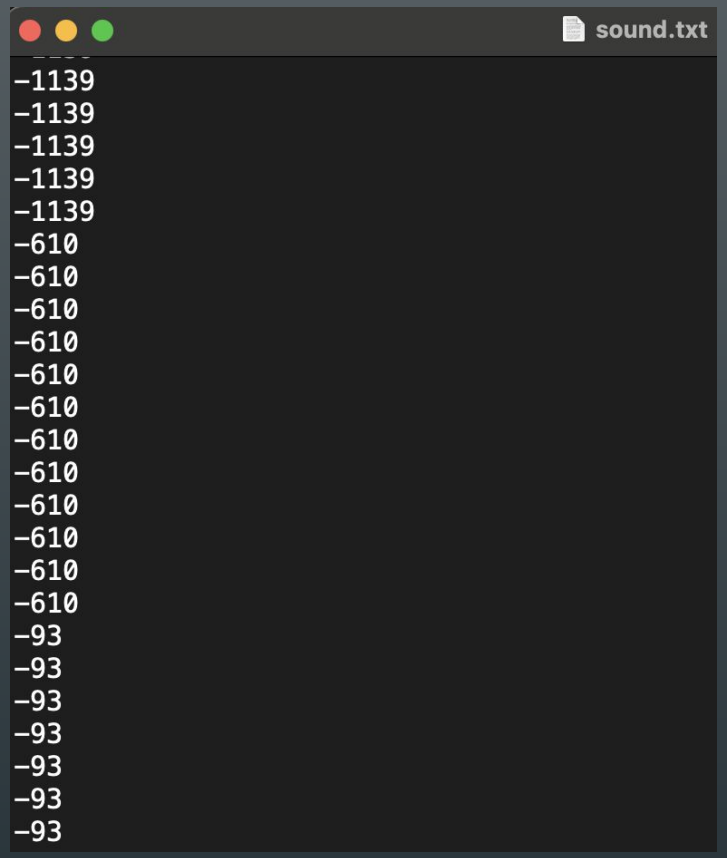
Block Diagram



Qsys


Platform Designer - soc_system.qsys* (/homes/user/stud/fall23/ykm2110/Downloads/hardwarefiles-20240515T000050Z-001/hardwarefiles/soc_system.qsys)
File Edit System Generate View Tools Help
IP Catalog
System Contents
Address Map
Interconnect Requirements
Hierarchy
Device Family
Messages
Current filter:
0 Errors, 0 Warnings
Generate HDL...
Finish

Audio Codec



- The microphone connected to the FPGA provides analog data.
- Then the CODEC takes this data and converts it into digital data.

Verification of CODEC

- After integrating the CODEC and input analog data we connected CODEC's output to the speakers.
- By doing this we verified that the CODEC is working as expected.
- We saved the digital data to a buffer and saved the buffered data in a text file.
- Performed FFT on the data saved in txt file so as to verify that we are getting the right data and accuracy is good.

Hardware-Software Interface

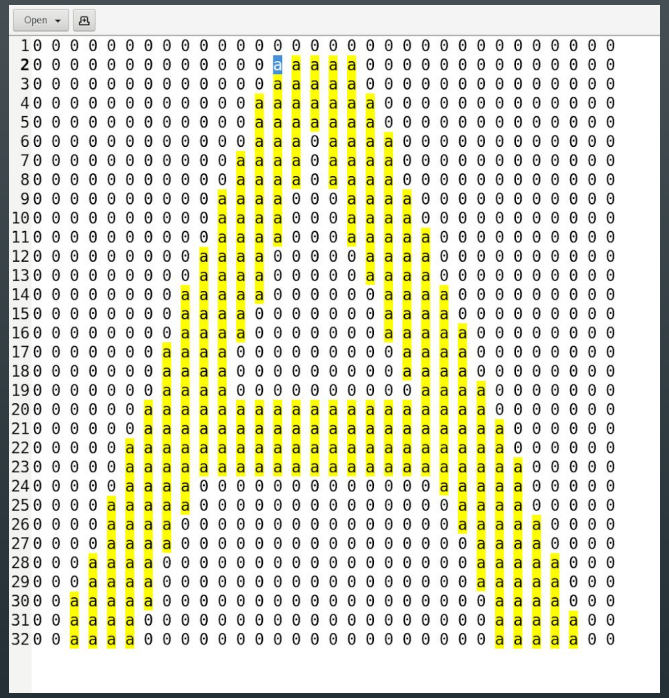
- The hardware of the DE1-SoC board interacts with the software logic to achieve its functionality.
- Writing driver or interface code, controlling memory access, and setting up peripherals are all part of this.
- Following the computation of the note detection, the software modifies the contents of the notes and frequency.
- This is done via a kernel program to the DE1-SoC board's registers.
- The VGA display is updated by the hardware once it reads the data from the registers.

FFT Algorithm

- The software will interface with the hardware and get the buffer data.
- Within the C program, audio samples are accumulated and stored in a time series array. They are then put though the FFT in software with a sampling frequency of 48kHz with 8192 bins.
- From this FFT output array, we are able to search the array for the index of the largest magnitude.
- This frequency is then plugged into a lookup table, where the corresponding note being played is understood.

VGA Interface

- We created custom sprites by drawing the pixel art for them.
- The image size is 32x32 pixels.
- These .txt files are read by Quartus to create ROM modules.



Note Detection and Display

- The VGA portion of the software, takes the identified note and determines the associated sprite, and note frequency to be sent back to the hardware.