ECE 540 FINAL PROJECT PROPOSAL

Project Name: FFT Audio Visualizer

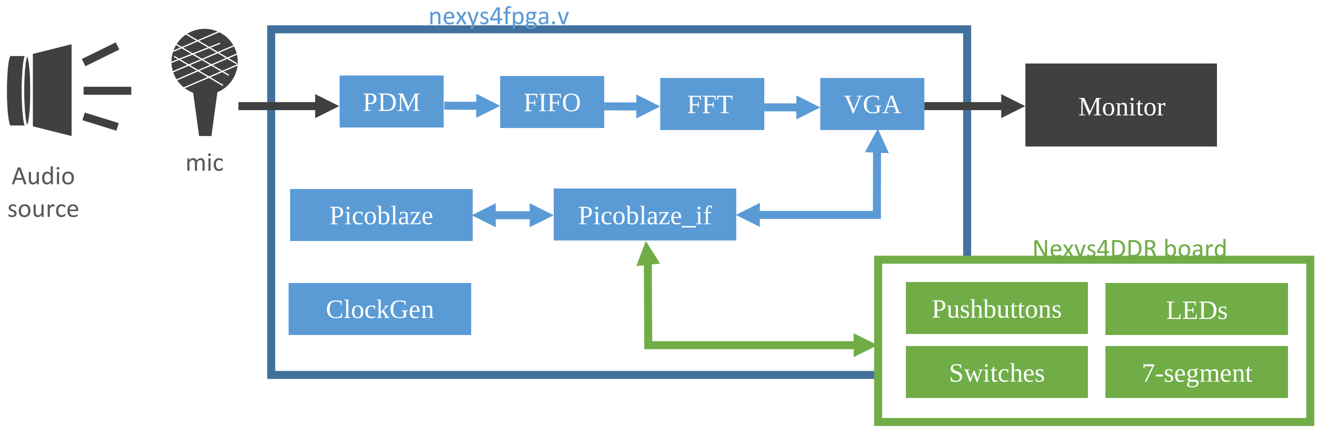
Engineer 1: Rehan Iqbal

Engineer 2: Yogesh Dhada

Engineer 3: Meng Lei

*Project Description [WHAT ARE YOU GOING TO BUILD? WHAT COMPONENT(S) WILL YOU USE? **INCLUDE A BLOCK DIAGRAM OF YOUR DESIGN AS YOU ENVISION IT**]:*

We are building an audio visualizer to display on-screen. We will sample an audio source from the Nexsy4 on-board microphone, which outputs a PDM signal. We will (optionally) store this in a FIFO buffer and then pass to a FastFourier Transform (FFT) module for processing. This module will be optimized to take advantage of the DSP modules in the Artix 7. With the results of the FFT, we will pass this to a VGA controller which will determine whether to draw the background or the spectral bins. The VGA controller will provide the appropriate RGB values for outputting on a monitor connected to the Nexys4 board. The pushbuttons, 7-segment display, and Picoblaze will be used to implement an options menu on the Nexys4 board. You will be able to navigate through the menu tree using pushbutton center as 'select', and modify the spectral display with it. We have not decided on which functions to modify the display, but we're considering number of bins, axis scaling, spectrum color, shift the spectrum up/down on screen, etc.



Design Approach HOW ARE YOU GOING TO BUILD IT? HOW WILL YOU DEMONSTRATE SUCCESS ON THE FPGA DEVELOPMENT BOARD? WHAT ARE YOUR OPTIONS IF YOU START RUNNING OUT OF TIME?]:

1) We are using Github (https://github.com/rsi7/finalproject.git) to manage code and push incremental development. We will establish the architecture and the IO's required for each block, then assign each module to a team member to develop, debug, and complete. No outside hardware should be required save for the monitor. A number of resources are available for FFT algorithms and HDL implementations, so that piece should be fairly straightforward. We will test this functionality on the Nexys4 LEDs to make sure the spectrum is operating as expected. The biggest challenge will probably be the VGA controller block, which will need some pixel detection scheme similar to the 'icon' module in Project 2.

   The majority of interesting work is going on in the FFT and VGA blocks. The FFT is something that would clearly be optimized for hardware MACs, and gives us the chance to play around with DSP toolboxes in MATLAB and possible try some high-level synthesis. The VGA seems rather painful to do in assembly and we already have HDL to handle it. Based on that, we think the Picoblaze would lend itself to implementing some kind of state machine to control the processing being done by (faster) hardware modules.

We considered using interrupts to pass data between modules using the Picoblaze as a hub, but decided that the FFT should connect directly to the VGA for the sake of speed. Another possibility we considered was having it control the onboard PWM as an audio source, connecting it to a cheap speaker and having the on-board mic sample that signal. This way the FPGA is both generating the audio source and reading/interpreting it; we could set it up to do sweeps, chirps, and see the reaction on the spectral display. However, we're not sure if we could realistically deliver that in a few weeks.

2) We will demonstrate success by having an 8-bin audio spectrum displayed on a monitor connected to the board. Each bin corresponds to a frequency range; bin 0 will cover the 0 Hz – 2.5 kHz range, bin 1 will cover 2.5 kHz – 5 kHz, and so on up to 20 kHz (the upper threshold for human hearing). The spectrum should be responsive to the audio and refresh at a reasonable rate.

3) We have intentionally kept the feature set very basic to guarantee deliverables. However, if time runs out we will replace the VGA output with a simpler display (e.g. 8x8 LED array). This should be considerably simpler to implement. If we are running ahead of schedule, we may implement additional features such as: RGB color cycling of the spectrum, a live waveform viewer, a connected RGB LED strip, better microphone connection through USB, DSP modification of the audio input (e.g. delay, reverb, etc.), additional FFT bins.

Milestones/Deliverables [HOW ARE YOU GOING TO DEMONSTRATE THAT YOU'RE MAKING PROGRESS]:

Week of 11/15: Have detailed architecture and IOs listed; divide work, research FFT, start top-level 'nexys4fpga' module.

Week of 11/22 : Audio input into buffer working; Picoblaze code for board IO running. Testing/debugging FFT module.

Week of 11/29 : FFT fully functional and displaying on Nexys4 LEDs; VGA controller in development.

Week of 12/6 : Demonstration and Final Report

Electrical and Computer Engineering Maseeh College of Engineering and Computer Science