CSEE 4840 Project Proposal: FPGA Music Mixer

Joshua Zhou (jz3311), Joy He (jh4337), Harrison Riley (hjr2128), Oli Macgregor (owm2106)

Overview

We would like to design an FPGA DJ that takes two song recording files and performs a hardware-accelerated FFT algorithm on the waveforms. The user can then use switches on the FPGA to decide which effects they would like on each song (half the switches will be for one song, half for the other). They can also decide whether to play one song or a weighted addition of both songs.

Software Component

We will store preselected song files on an SD card, which can be selected by the user. We will use C or C++ to load these files (either as WAV or MP3 depending on our assessment of available resources), perform any necessary pre-processing, and send the files in chunks (to not overload the hardware) to hardware. We will then receive the frequency analysis from the hardware and perform the “dj effects” in software. Our hope is to have three effects: high-pass/low-pass filtering, echo-ing, and reverb, but we will assess the time constraints as the

project progresses. We will take inspiration from previous audio and image-processing projects in this class on how to best design the interface between hardware and software and deal with large files.

Hardware Component

Although the fast fourier transform is already an optimization of the original fourier transform in software, we would like to see if this can be accelerated more by being performed in hardware. There are several papers that discuss the procedure to do so, including a STFT (short-time fourier transform) that performs FFT operations in parallel with each other. We plan to use the code blocks in these papers to save time. Additionally, the user interface will begin at the hardware. We plan to use half the switches to toggle the effects for the first song the user selects and half to toggle the effects of the other song. We also plan to use the LED lights on the FPGA, and potentially the VGA monitor, to indicate volume.

Parts

- VGA monitor (to display which songs are being played)
- FPGA Altera Cyclone V
- SD Card with Ubuntu
- Speaker
- Microphone (optional stretch)

Milestones

1) Flesh out hardware-software interfacing

2) Get audio files on SD Card, read them on software, and send files to hardware (make sure the speaker can play the songs individually and combined)
3) Implement hardware-accelerated FFT in SystemVerilog, send frequency analysis back to software
4) Perform audio effects on software → figure out whether to perform inverse FFT in software or hardware
5) Implement switch logic on FPGA

Potential Issues

- Audio files are memory intensive. We will have to split computation into chunks
- FFT acceleration in hardware is not very common (may try to use a "knockoff" version of the fourier transform that is similar enough)
- Potential inefficiencies in interfacing between hardware and software

References

- https://web.mit.edu/6.111/www/f2015/projects/asloboda_Project_Final_Report.pdf
- https://www.twam.info/general/building-a-custom-usb-audio-mixer
- https://web.mit.edu/6.111/www/f2017/handouts/FFTtutorial121102.pdf
- https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=040c8b6516ae83a284e0e8f5482f3c1c6314ed86
- https://www.ijrte.org/wp-content/uploads/papers/v8i1/A9240058119.pdf