Audio Mixer

ECE 287: Digital Systems Design

Professor Rajasekhar

Matt Slavney

Nick Mara

The Problem

The Altera DE2-115 FPGA board contains a 24-bit audio CODEC chip. This chip supports microphone-in, line-in, and line-out ports, with a sample rate adjustable from 8 kHz to 96 kHz. What we wanted to do was integrate VHDL logic with this chip in order to implement a program which is able to manipulate an audio sample on the FPGA. Our task was to create an audio sampler that would allow us to record and playback sounds as well as change the volume and add audio effects. This sampler needed to be able to play all of these sounds through a speaker that is connected to the FPGA board.

Description of Design

Our design uses switches that allow the user to interface with the microphone and manipulate the sounds. Each switch has a different function and the switches are as follows:

- The enable - This switch simply allows the program to be turned on or off, without this switch being set to the enabled value, the program will not work. This plays the song continuously on a loop and if no other switches are turned on it will be playing at the normal volume and speed.

- *Record Sound - This switch, as the name implies, allows the user to record a sound with the maximum length being up to one and a third second. *This was later removed due to the static problem (discussed the report below).
- *Playback recording - This switch plays back the recorded sound one time, and depending on what other switches are active, it also plays the applied filters (determined by other switches) with the sound. *This was later turned into the switch that plays the “Shave and a Haircut” tune that we created (see report below).
- Time shift half speed - This switch causes the song to play slower at half the speed as normal.
- Time shift
- Volume up - This switch plays a louder version of the song.
- Volume down - This switch plays a softer version of the song.

When working with the mic and attempting to record audio into the board, we encountered many problems. We were receiving an enormous amount of static and were not able to record audio over this static. We attempted to fix this problem by programming the FPGA to store the audio data as RAM. From this also arose many problems because the board only has enough memory to store around 1.3 seconds worth of audio recording. Not only this but our problem of static persisted and no matter what we did, there was no way around the static. Since this was a crucial part of our project and we were running out of time to find solutions to the problem, we were forced

to change our design from recording audio into the board, to playing back a preset sequence of notes.

When we were using an example code, we discovered as long as the sound playing from the board was a signal sent from the computer, there was no static and the pitch was clean. Therefore we coded a series of notes into the board using math to distinguish each note by its frequency. We also made each of these notes a square wave because it was simpler to code and there isn't any real sound difference. Then to achieve our tune of "Shave and a Haircut", we determined a desired tempo, and changed the clock accordingly. Then we used the sheet music to determine what note to play at what time and implemented the notes using a case statement.

In order to create the effects to change the volume up or down, depending on the switch selected, we needed to implement additional code to our design. To manipulate the volume and make it higher we simply multiplied a number to each of the notes amplitudes. The amplitude of each note is what determines how loud it plays so in order to make it softer we then needed to divide each of the amplitudes by a number. We chose to make the song twice as loud and also twice as soft therefore the number we multiplied by was two.

In order to create the time shifting effect that causes the tune to play faster or slower, we needed to modify the code and cause our clock to change. To make the tune run twice as fast we divided our clock time in half and visa versa to make it play twice as slow. This has no effect on the amplitude or the pitch of the notes because it is only

manipulating the clock, therefore the notes play at the same frequency but for different periods of time.

All of these features can also be overlaid on each other. For example, you can play the volume louder while also making the song twice the speed. If, however, you play the volume louder and softer, you will simply get the original unaltered tone. The same is true if you try to play the song faster and slower at the same time.

Summary

We have strayed from our initial proposal due to unforeseen problems with static in the board as well as memory space and storage issues. Therefore we adapted our design to manipulate audio coming from the computer. Thus, our design plays audio through a speaker connected to the FPGA. Using switches you are able to manipulate this audio by speeding it up and slowing it down, as well as making it louder or softer. Without the use of the audio CODEC chip on the DE2-115 board this would not be possible.

Citations

http://www.alteraforum.com/forum/showthread.php?t=26339 “mbharat”- This was the forum and the username that we found the code that allowed us to use the audio CODEC.