Soos Qasem & Huiwen Zhang

EELE 468

4/23/2023

**Final Project Report**

A wah-wah bandpass filter is a type of audio filter commonly used in music production and guitar effects pedals. It creates a sweeping sound effect by selectively filtering certain frequency ranges of an audio signal. The effect is achieved by modulating the center frequency of a bandpass filter over time using a pedal or an automated circuit. As the center frequency moves up and down, it creates a distinctive "wah-wah" sound that can be heard in many classic rock and funk guitar riffs.

In a system implementation, a wah-wah bandpass filter typically consists of an audio input, a bandpass filter with a variable center frequency controlled by a pedal or automation circuit, and an audio output. The audio input is fed into the bandpass filter, which selectively filters out frequencies outside of a specific range. The pedal or automation circuit modulates the center frequency of the filter, causing the sweeping effect. The filtered audio signal is then output through the audio output, creating the distinctive wah-wah sound effect.

The signals processing block that is called stateVariableFilter, has the following inputs. The original audio signal that needs to be filtered, the damp factor and delta which are used for filtering the input signal, and the minimum and maximum frequencies that set the limits of sweeping method.

A picture containing diagram, text, line, design

Description automatically generated

Fig. 1

The output of the signal processing block is the result of a bandpass filter that goes into a wetDryMix block to be combined with the original audio signal. After the filtered signal is combined with the original signal, the output of the wetDryMix is then controlled by two parameters. The first parameter is the enable control register that if set to 1, the filtered signal will be pushed to the AudioOut port, and if set to 0, the original signal will be pushed out. The second parameter is the volume controller that ranges in values from [0 1] to control the outputs’ volume. Fig. 2 shows the implementation of the filter’s signal flow that implements the lowpass, bandpass, and highpass.

The generation of the wah-wah filter is dependent on the difference equations which are:

* yl(n) = F1yb(n) + yl(n − 1)
* yb(n) = F1yh(n) + yb(n − 1)
* yh(n) = x(n) − yl(n − 1) − Q1yb(n − 1)

where F1 and Q1 are the tuning coefficients:

* F1 = 2sin(π\*fc/fs)
* Q1 = 2\*d

Diagram, schematic

Description automatically generated

Fig. 2

For implementing the sound effect and controlling the tuning parameters, we had to write device tree files that control the actual sound effect and the adc input that initializes the three potentiometers. As well as writing a device driver that connects the inputs of the potentiometers into the sound effect registers to be able to control the tuning parameters.

Our favorite aspect was debugging the device driver and tree files since we enjoy low-level programming, as well as working with Quartus and making sure everything was whereas it should be. The most difficult aspect of the project was having to learn using Simulink to generate systems using vhdl compatible blocks. This project, bringing everything we learned through the course together, was a great way to finish EELE468 FPGA2.