John McCormack  
DSP Final Project  
4/28/2016  
Audio Wide Circuit

1. Introduction

The program I have written is based off of an analog circuit I helped design for my previous employer. The concept of a wide circuit has existed since the 70s and was previously used quite often to enhance the sound of portable speakers. The essential idea is to emphasis the sounds on the far ends of a stereo mix to make the audio sound wider than normal. To do this, sound from the opposite channel is mixed in, out of phase. This causes the audio in the center of the mix to cancel out, as it is the only data that exists in both channels.

1. Execution

The code is available on github. The first step is to extract the audio from the audio in line into the left and right channels. We do this using a bit mask and then shift the bits to go from a 32 bit integer to two 16 bit integers. We then store the audio in the provided complex number and complex buffer format. The audio data is purely real, so we leave the imaginary part of the complex number set to a float of 0 and load the audio as a float into the real part of the complex buffer. We then use the provided fft function to get the fourier transform of both the left and right channel.

Once the data is loaded into the frequency domain, it can then be processed. We first convert the rectangular complex numbers into polar form. It is important to use only float based math functions as the double based math functions cause errors on the board. Once we have converted to polar form, we then shift the phase of the numbers by pi and convert back to rectangular. We store this new number in a separate buffer in order to preserve the original audio data. Next, the out of phase data is added to the in phase data of the opposite channel. This creates a waveform that does not have the center audio present. This is then added to the original waveform again so that the center audio can still be heard, but now with extra emphasis on the edge data. If we do not add the center audio again then a “karaoke” effect is achieved in most songs, meaning the singer’s voice has been removed.

Finally, the inverse FFT is performed on the audio. To get the inverse FFT, a separate set of twiddle values was calculated where the imaginary component was multiplied by negative one. After performing the iFFT the two audio channels are merged back into the main audio channel using bit shifting and bit masking. Since the iFFT removed the imaginary component, we are only concerned with the real part of the complex buffer. This allows us to transition the data from two 16 bit integers into a single 32 bit integer.

1. Limitations

This form of audio enhancing is fairly simplistic. As a result, the wide effect is limited. To get a better wide sound, reverb is usually employed to give the feeling of depth in addition to width. This creates a more convincing psychoacoustic response. Additionally, the process has introduces a decent amount of white noise. If given more time, it would have been better if I had introduced a low pass filter to cut down on this noise.