ECEN302 Lab 7 - TCL scripts and Packaging IP

Joshua Benfell - 300433229

October 2, 2020

1 Introduction

The primary objective of this lab was to wrap up a module as an IP Block. This was done so that an understanding was gained of how to compartmentalise and modularise the program written for an FPGA was acquired for the purpose of speeding up development time in future endeavors.

2 Methodology

The module being packed up in this lab is a PWM sine wave generator. This module takes the first nine switches as inputs to provide a binary weighting to a unity sine wave. Flicking the switches on would increase the time it takes for a point on the output sine wave to be generated; this in turn lowers the frequency of the output sine wave. The AUD_SD port of the module was used to turn off the low pass filter which stops the audio signal from outputting. The AUD_PWM is the output port for the generated tone.

To set up the project a tcl script was run which performed all the connections that needed to be made in vivado to create the PWM audio master module. tcl scripts are a good option to use for saving projects as it recreates them with the same commands that vivado runs automatically as you make the project from the ground up. This allows easy version control and modularity in created components.

Once it was created it was opened in vivado and any dependencies were updated to the latest versions to ensure that everything can be found. This module makes use of a direct digital synthesizer to generate a sine wave of a certain frequency as detailed above and then uses a converter to output a PWM equivalent.

This was then uploaded to the FPGA to verify it's functionality, and then it was abstracted so that it was no longer specific to the board being used, but rather the chip being used. This provides greater flexibility to where the IP core can be applied. It was then packaged with the IP packager tool and categorized as a waveform synthesis module to make it easier to find in the future. This was then validated by creating a managed IP project and doing a test import.

Now that an IP block has been created, it was imported into a Vivado project through the block design editor. In addition to this, the external ports and clock interface were added to the design so that the IP block could function correctly. Finally, it was downloaded to the FPGA after creating a new constraints file and the functionality was verified to be the same as the pre-packaged version.