Audio Synthesizer

Submitted to:

Dr. Gregg Chapman

Srinivastin Subramaniyan

Created by:

Group 18

Ansh Pachauri

Rushi Bhatt

Siwei Fan

ECE 2060

The Ohio State University

Columbus, OH

19 March 2024

Executive Summary

The purpose of this lab was to create an Audio Synthesizer which generated various tones corresponding to different push buttons on the Simon game. To begin this lab the group first loaded our previous lab, the Simon Game Project, from Quartus Prime software. Then, the team integrated a 32-bit sine wave synthesizer into the schematic, utilizing calculations to determine the requisite P-value for generating a 200 Hz sine wave. This value was then incorporated into the schematic. Subsequently, the synthesizer was encapsulated into a symbol file labeled "Audio" and integrated with the previously created CODEC. The clock input was then linked to the AUD_DACLRCK clock, facilitating observation of the sine wave output via Signal Tap, with detailed procedural elucidation provided in the Experimental Methodology section.

The subsequent phase of the lab concentrated on refining the audio synthesizer. The team iteratively adjusted the audio schematic with 5 distinct P-values as delineated in the lab 6 guide. Additionally, integration of the debounce block from lab 5 necessitated modifications to inputs and outputs for system refinement. Ultimately, the team crafted a VHDL file named "SoundDecoder" and generated a corresponding symbol file, seamlessly integrating it into the existing CODEC schematic. Following compilation, programming, and testing, the project demonstrated successful operation, with different octaves corresponding to each button press on the Simon Box.

This report extensively delineates the rationale and objectives behind the lab, followed by a comprehensive exposition of the experimental methodology and schematics pivotal for successful completion. The results section encapsulates wave simulation outcomes, VHDL file details, and continued exploration of schematics and VHDL files. Subsequent to the discussion of a series of probing questions pertinent to various facets of the lab, a concise summary reaffirms the attainment of primary objectives. Acknowledgments for each team member and references utilized for both lab completion and report composition are duly included.

Introduction

The objective of lab 6 revolves around the exploration and comprehension of audio synthesizer development. It is pivotal to recognize that the synthesizer's role lies in generating signals to produce varied tones triggered by the activation of individual buttons on the Simon Box, a concept elaborated further in the forthcoming lab report. Through the process of crafting the audio synthesizer, the team has gained proficiency in waveform generation, particularly in generating sine waveforms by cycling through a ROM file defining a single period of the wave. Furthermore, familiarity with Signal Tap implementation via Quartus Prime software and the calculation of diverse P-values has been acquired. Notably, this lab has served as an avenue for the team to apply problem-solving techniques and harness VHDL coding skills garnered from prior labs to formulate a novel VHDL design for the audio synthesizer's implementation. The methodology and outcomes of this endeavor are meticulously expounded upon in the ensuing sections.

Experimental Methodology

In this lab endeavor, Group 18 embarked on the construction of an audio synthesizer, necessitating the utilization of specific materials including a DE2 Board, Simon Game Box, and a speaker. The instructional resources provided encompassed a reference video titled "Synthesizer" along with essential files denoted as "lpm_dff1.vhd" and "sin_16bit.mif". Commencing the lab, Group 18 commenced by viewing the Synthesizer video and proceeded to open the latest iteration of the SimonGame project for schematic design continuation. Within the project framework, an additional schematic file was incorporated, wherein the 32-bit Sine Wave Synthesizer from the provided design was meticulously recreated, bearing the designation "Audio". The sin_16bit.mif file sourced from the lab website was imported into the project folder and seamlessly integrated into the project.

Following schematic creation, Group 18 undertook the calculation of the requisite P-value for generating a 200 Hz sine wave, subsequently substituting the predefined constant value with the calculated parameter. A symbol file of the Sine Wave Synthesizer was generated and christened "Audio", strategically positioned adjacent to the CODEC and linked to the AUD_DACLRCK clock input of the Audio block. Additionally, the output bus was designated as sound[15..0], with subsequent bit shifting to the AUD_outL/R inputs as per the provided video and schematic guidance.

Subsequently, the 200 Hz Sine Wave was observed utilizing Signal Tap, with the project compilation prioritizing CODEC as the top-level entity. SignalTap Logic Analyzer was activated via the Tools menu, configuring the Hardware to USB-Blaster and setting Signal Tap: pre-synthesis as Filter. The AUD_DACLRCK was selected from the designated list, with a Sample Depth of 8k established. Further customization entailed the addition of the sound output to the nodes list and configuration of the Display Format to a Signed Line Chart. Compilation and reprogramming ensued, culminating in the analysis initiation, facilitating observation of the sine wave output.

In a subsequent phase, Group 18 proceeded with the modification of the Audio schematic to accommodate five distinct constant values, rectifying inaccuracies in the originally provided values. The updated symbol file for "Audio" was generated, replacing the existing symbol at the top level. A block was created to facilitate input from the debounce block, with sound output selection contingent upon the activated input. GPIO and LED relations were established accordingly.

Leveraging the SevenSegment.vhd file from the provided resources, Group 18 formulated a new VHDL block, ensuring compatibility with the "STD_Logic" data type to accommodate single-bit inputs. The VHDL code for the "SoundDecoder" was meticulously crafted, followed by the creation of a symbol file and its integration into the CODEC schematic. Subsequent compilation, programming, and verification were conducted to ascertain the functionality of the project.

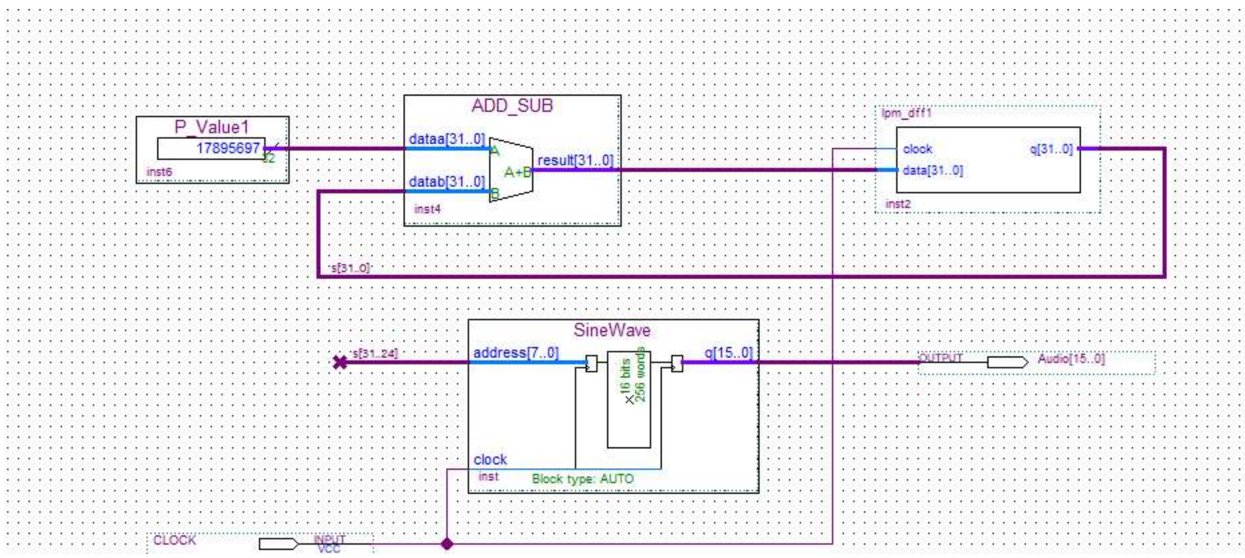

Figure 1 - 32-bit Sine Wave Synthesizer

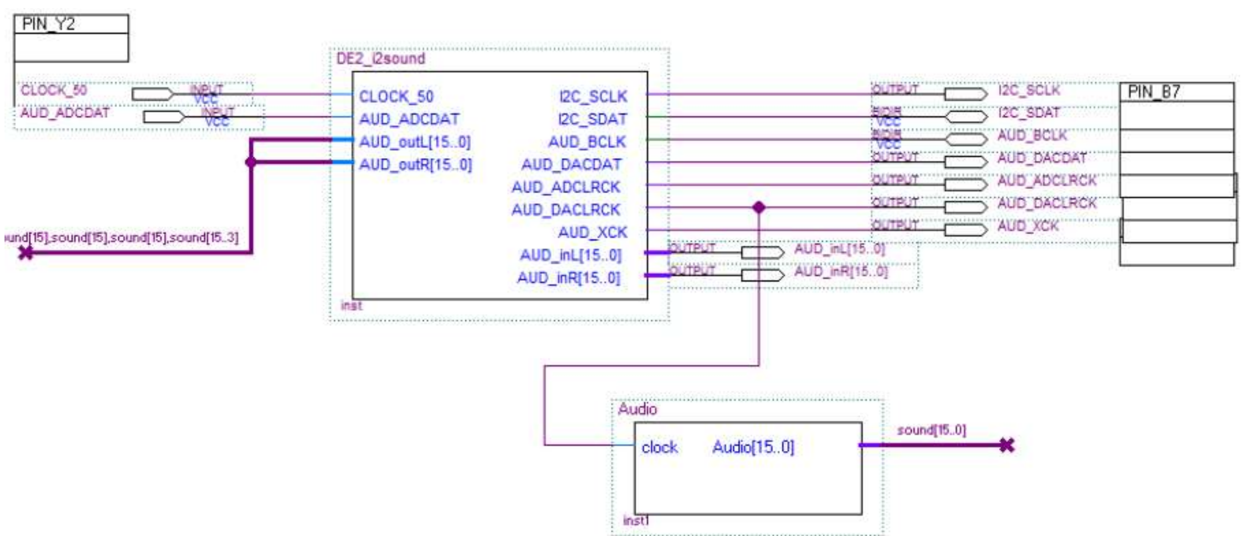

Figure 2 - Audio Top-Level Schematic

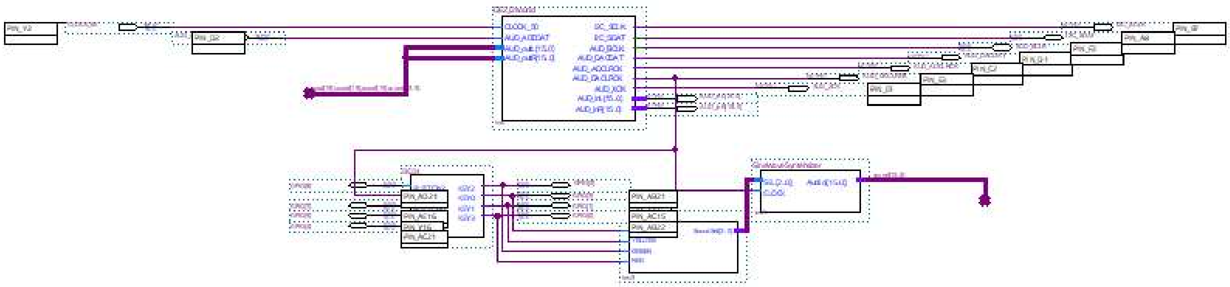

Figure 3 - Modified Audio Schematic

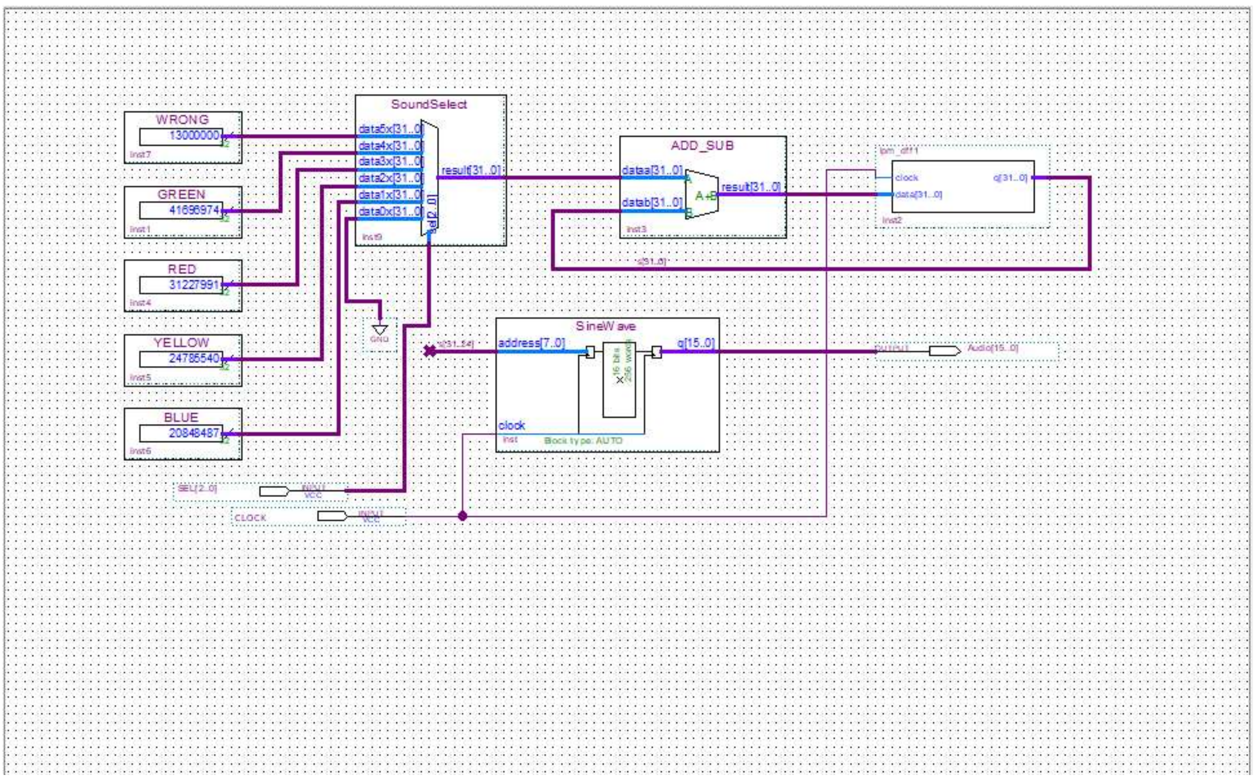

Figure 4 - Sound Decoder Top-Level Schematic

Results

In the aftermath of the project, Group 18's progress is outlined through schematic designs and VHDL code, leading to the following outcomes. Starting with the creation of the Sine Wave synthesizer, the team then made an Audio symbol file from that schematic. This Audio symbol was added to the existing CODEC file from the Simon Game project, connecting to the clock input of the button debounce symbol and the AUD outputs of the DE2_i2sound board. Some adjustments were made to certain inputs for the final schematic design.

Next, the team followed instructions to analyze the 200 Hz Sine Wave using SignalTap. The new CODEC file schematic helped model the sine wave, important for understanding the audio symbol's output. The modified audio schematic now allows for inputs of different colors and values, crucial for the light board. Following this, a block was created to handle the four outputs from the debounce block as inputs and select the appropriate sound output based on the activated input. The final block demonstrating this is shown below.

To ensure the sound decoder schematic works smoothly, the team created VHDL code matching the schematic's functionality, tying everything together. The code's behavior is explained in detail below. After combining the VHDL code and schematic, the resulting sound decoder accurately responded to button presses. Group 18's efforts successfully built a fully functional sound decoder during this lab.

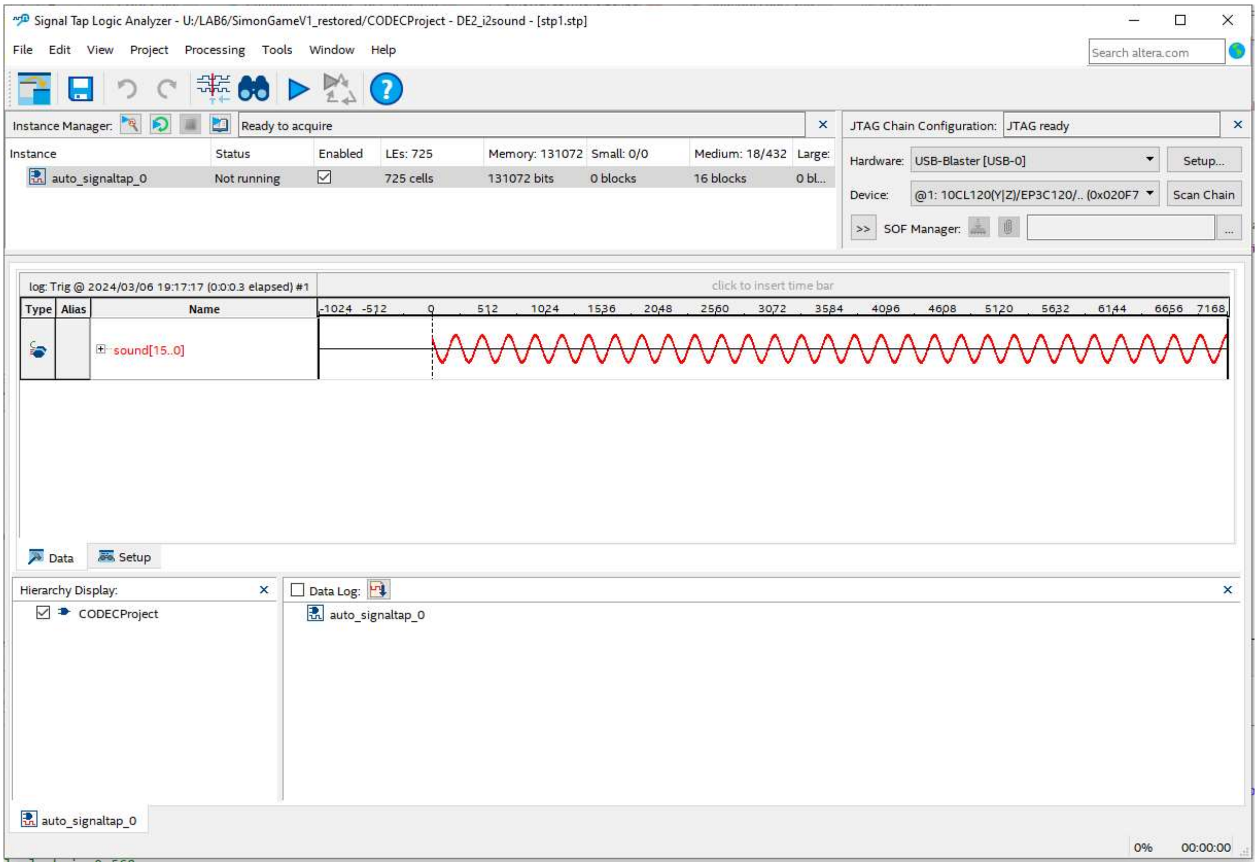

Figure 5 - Signal Tap sine wave simulation

```vhdl
Library ieee;
USE ieee.std_logic_1164.all;

Entity SoundDecoder IS
	Port
	(
		BLUE : IN STD_LOGIC;
		YELLOW : IN STD_LOGIC;
		GREEN : IN STD_LOGIC;
		RED : IN STD_LOGIC;
		SoundSel : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
	);
END SoundDecoder;
ARCHITECTURE Behavioral OF SoundDecoder IS
BEGIN
		SoundSel <= "001" when BLUE = '0' else
				"010" when YELLOW ='0' else
				"100" when GREEN = '0' else
				"011" when RED = '0' else
				"000";
END Behavioral;
```

Figure 6 - Sound Decoder VHDL code

Discussion

Converting the content of a ROM (.mif) file into a continuous synthesized waveform involves several steps. Initially, the .mif file is imported into the designated project folder, followed by adding the ROM as a symbol module to the schematic. Once the schematic is completed, the project is compiled to synthesize the design, and subsequently exported to ModelSim or other relevant software. When designing a ROM with 32-bit values and 512 points of a sine wave, adjustments are made to the Matlab file code depth to 512 and the width to 32. Higher frequencies result in shorter periods of the sine wave, intensifying it within that interval. For instance, the P value required for a 500 Hz sine wave and a width of 16-bits is 2048, determined by the equation P Value = $2\pi f/Fs$, where f is 500 and Fs is 16*10^3. Furthermore, a P value of 447392427 yields a frequency of 5000, obtained by using the same equation but solving for Fs when f = 48000.

Conclusion

To sum up, this lab proved highly beneficial for grasping the implementation of an Audio Synthesizer utilizing Signal Tap and Quartus Prime software. Group 18 engaged in simulating the Sine wave within Signal Tap, observing its shape and frequency. Subsequently, this Sine wave served as a basis for implementing a Sound Decoder in Quartus Prime, alongside an Audio schematic and clock input. Leveraging this schematic, Group 18 devised a four-input circuit, assigning each color button a distinct frequency emission upon pressing, accompanied by corresponding illumination. In essence, this lab provided invaluable hands-on experience in applying classroom-learned skills to a software platform capable of accurately simulating circuit models.

Acknowledgments

All members of Group 18 contributed equally and participated in all lab activities. In terms of the lab report, Siwei wrote the discussion and conclusion sections. Ansh described the executive summary, introduction, and edited the document. Rushi wrote the experimental methodology and results sections.

References

- Lab report template - Lab-Report-Submission-Instructions
- Lab 6 document - encoderdecoderLab