Sebastian Krupa & Ashuka Xue

E155 Final Project Proposal

November 4, 2014

The goal of this project will be to create a digital keyboard capable of producing a variety of sounds and notes by varying frequency and waveform. The project will consist of a 12 note keyboard that will send data to the PIC when a key is being pressed. The PIC will then translate these inputs into the desired frequencies and send them to the FPGA using SPI interfacing. The FPGA will produce waves with the specified frequencies and waveforms, outputting them to a set of I/O pins which will be converted into an analog signal sent to a speaker.

KEYBOARD

The keyboard will consist of 12 3D printed keys to signify an octave. These keys will be attached to push buttons which will toggle the system between ground and 3.3 V. All 12 of these signals will be sent to 12 separate I/O pins to avoid scanning. This board will also include a switch to select between the different waveforms. Only one waveform will be able to be produced at a time for all 12 keys. Finally, there will also be an octave increment/decrement switch pair which will shift all 12 keys by an octave depending on user input.

PIC32

The PIC will be responsible for interpreting input from the keyboard and sending appropriate signals to the FPGA. It will read all 12 input pins to decide which keys are currently being pressed. It will also read which octave and waveform is currently selected. Using the octave and note data, it will send a signal to the FPGA which holds the correct frequency for each of the notes being pressed. The frequency will be determined by creating a clock signal that is generated by taking the note frequency and multiplying it by a desired sampling frequency. This sampling frequency will be dependent on the number of registers used by each FPGA module.

FPGA

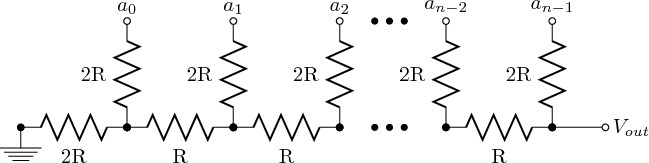
The FPGA will contain 4 shift register modules each representing one of 4 waveforms. These will be a square wave, a sawtooth wave, a triangle wave, and a sine wave. Each of these waves will have 4 bit accuracy in amplitude which creates a fairly accurate digital approximation of each of these signals.

During implementation, 3 modules will be running at the same time to produce 3 instances of the desired wave so that a maximum of 3 notes can be pressed at any time to produce sound. The outputs from the notes will then be added together to produce a complex 6 bit output wave. In order for the output to have the same volume regardless of how many notes are being played, the output will be scaled depending on the number of notes. If 3 notes are being pressed, the output remains the same. If only 2 notes are being played, their output is added to itself bit-shifted right by 1, and if only 1 note is being played, it will be added to itself bit-shifted left by 1.

This 6 bit output will then be output to 6 I/O pins connected to a resistor ladder.

RESISTOR LADDER

The resistor ladder will work as a way to create an analog signal from our 6 bit digital signal. This will be achieved by using common resistor ladder conventions as shown below:



http://commons.wikimedia.org/wiki/File:R2r-ladder.png

In this case, Vout would range between ~0V and ~3.3V. Although not exactly 3.3V will be output, since all output values will be affected by the same constant multiple, the result will still be scaled properly. This Vout is then used to drive the speaker circuit.

SPEAKER

Finally, the speaker circuit will be similar to the one used for lab 5 in order to play the desired sounds.

BUDGET

|  |  |  |
| --- | --- | --- |
| Item | Quantity | Unit Cost ($) |
| SLB1470 4 Position Slide Switch | 1 | 1.69 |
| GF0668-ND 8 OHM speaker | 4 | 4.87 |
| 35RASMT2BHNTRX 3.5 mm Phone Connector | 3 | 0.66 |
|  |  |  |
|  | TOTAL | 23.15 |