Comp Arch Final Project Proposal

Project Name: Musical FPGA

Team Members: Deborah Hellen, Ben Kahle, Erika Weiler

Project Description:

We will build an 8 note “piano” in Verilog and load it onto an FPGA. We will use the 8 switches available on the FPGA to control the notes. When a switch is held high, the tone associated with that switch plays and when it is brought back to low, the tone stops. We will create the 8 different tones by producing square waves of different frequencies.

As our first stretch goal, we would like to be able to change octaves using the four buttons on our FPGA. This will probably just involve multiplying our frequencies by a constant. As our second stretch goal, we would like to generate discrete sine waves using verilog to improve the sound quality of our piano.

This project will give us the opportunity to explore logic to manipulate frequencies of signals within the audible range and interface with the inputs and outputs of the physical system of the FPGA. Through the achievement of our stretch goals we will also be able to investigate discrete signal manipulations through the use of digital logic.

Green Features (Minimum Deliverables):

- Produce 8 different tones using square waves
- Each tone is associated with a switch on the FPGA and plays as long as the switch is held high
- Only one tone can be played at a time

Yellow Features (Stretch Goals):

- Support four octaves of output by pressing one of the four buttons on the FPGA
- Generate the 8 tones as triangle waves or discrete sine waves to improve sound quality
- Support chords by toggling multiple switches at once.

Red Features (Out of scope):

- Recording user input and delaying playback

Possible Failure points and mitigations:

- Our verilog code runs into an error that prevents it from synthesizing onto the FPGA
  - Convert more behavioral verilog into structural verilog to have more control over the objects being synthesized.
  - Worst case: run the verilog through ModelSim and either attempt to play signals through computer audio or create a “visual” piano by watching the signals change
- We are unable to interface with the input controls (switches and buttons) on the FPGA

    - Ensure hardware configuration is done correctly and works on a separate FPGA
    - Worst case: “hard code” an input sequence and turn the FPGA into a “played-piano” instead.

Initial 2 steps:

1. Figure out best way to output audio from an FPGA
2. Take a baseline clock and alter it to various audible frequencies