Graham Sider
Michael Hong

Final Report - "The iHarp" **ECE241**

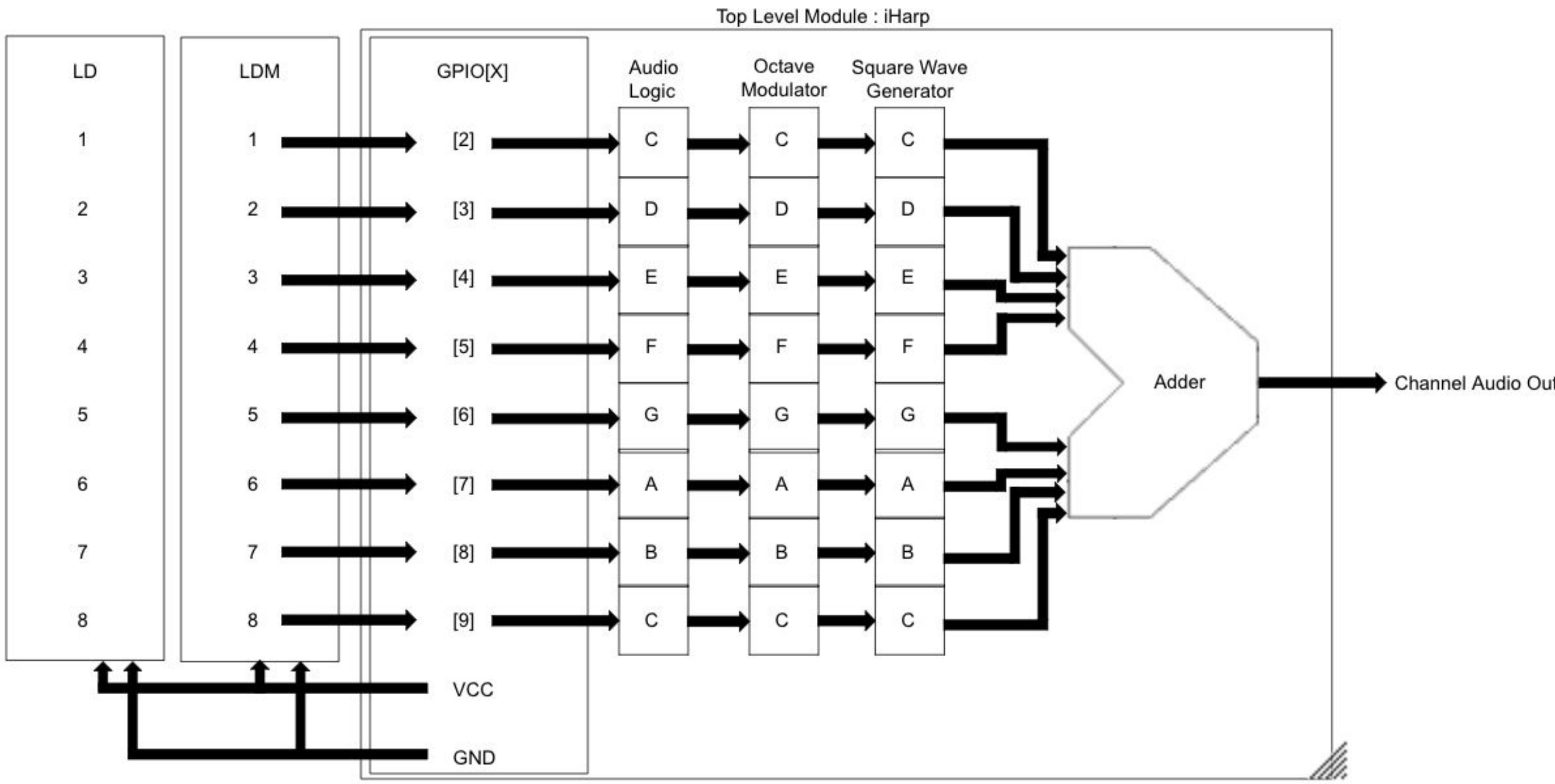
1. Introduction

A harp produces sound by setting nearby air molecules into vibrational motion when a string is plucked or 'vibrated'. However, an engineer might ask: Does a harp really need strings to produce sound? We began this project with the simple intention of intertwining music with engineering. We first discussed the idea of an "invisible harp" back in early September, and decided to take on the challenge once we noticed the majority of people on the project ideas page using the VGA adapter. The initial goals of this project were to design an instrument-like object off the FPGA itself that would play various notes (including sharps and flats) depending on which sensor detected an object above or below it.

The physical design of the iHarp consists of 8 laser modules and 8 corresponding laser detector modules which detect a beam of light (or the absence of a beam of light) from a laser module. The lasers will always be hitting the sensors at its default state; thus, when an object such as a hand passes through a laser, the hand will block the beam from reaching the detector. Each sensor has an output which is connected to a pin on the FPGA board; a signal is sent to the board when the sensor detects obstruction. By the state of the signal, the FPGA determines whether or not to play the corresponding note. Following this, we implemented a variety of logic such as: an octave modulator, the ability to go up (or down) a semitone, the ability to change the duration a note gets played, etc.

2. The Design

Block Diagram:



Verilog design:

Square Wave Generator:

Eight assignments each consisting of multiple conditional statements to produce three possible 32-bit signed output scenarios which control the amplitude of each note:

I. No wave: outputs a constant amplitude of 1’d0. This occurs when the laser is unobstructed and the “note sustain” and “string-like” audio logic functions (described below) are not in effect.

II. Default wave: the output alternates between a positive amplitude (0x10000000) and a negative amplitude (0xF0000000) at a frequency determined by a counter that counts to the appropriate duty cycle given by Octave Modulator module.

III. Dampened wave: outputs the wave described in (II) for 0.5s and ends the note with the “string-like” audio effect for the latter 0.5s (i.e the note fades out). Only occurs when the “string-like” switch is high. More information on “string-like” provided below.

Square Wave Summation:
Sums the eight values produced above into a single 32-bit value to send to the Audio Core module. To be certain that this value was no larger than 32 bits, we divided the absolute maximum possible value (positive being 0x7FFFFFFF) by the number of notes we wished to include in the design (8). This resulted in a base positive amplitude of 0x10000000 for each note, as was mentioned above.

Octave Modulator:
Outputs the appropriate duty cycle for each note (calculated via dividing the note frequency by the clock frequency). The octave (and thus the corresponding duty cycles) is dependant on which key is pressed/whether the reset switch is flipped (KEY[0] = C2 to C3, KEY[1] = C3 to C4, reset = C4 to C5, KEY[3] = C5 to C6, KEY[4] = C6 to C7).

Audio Logic:
The audio logic we implemented can be split up into two categories:

I. Note duration logic:
   A. Default: occurs when both “sustain” and “stringlike” are low. A given note will only play *while* its corresponding laser is being obstructed.
   B. Sustain: occurs when the “sustain” switch (SW[0]) is high. If a given laser is obstructed, the corresponding note will only stop playing when the “sustain” switch goes low or the “reset” switch goes high. By summing the square waves (as mentioned above), this feature can be used to produce various harmonies.
   C. String-like: occurs when the “string-like” switch (SW[1]) is high. Whilst a given laser is obstructed, the corresponding note will play normally. Once the laser becomes unobstructed, the note will continue modulating between its default positive and negative amplitudes for 0.5s. It will then decrease the absolute values of the two amplitudes by 0x2BB1 every (1/48000)s for another 0.5s. Thus, one second after the laser becomes unobstructed, both amplitudes evenly reach a value of 1’d0, resulting in a “fade-out” effect. If both the “sustain” and “string-like” switches are both high, notes will sustain as described in (B). If the “sustain” switch is then subsequently flipped low, all sustained notes will fade-out with the “string-like” effect as described above.

II. Semitone logic: uses SW[8] and SW[7]. When SW[8] is flipped high, “semitone mode” is enabled. In this state, all notes will go down by one semitone if SW[7] is low, and up by one semitone if SW[7] is high. This works for any octave and any note that is already being sustained.

Physical Design:

The physical design was made entirely of wood and included four rectangular slabs, two on top and two on bottom, to create two small, rectangular boxes. It was all held together by four cylindrical poles (one for each corner of the design). The top box includes wiring that connects the Vcc and GROUND wires to eight LDs (Laser Diodes) that are placed into the lower of the two top slabs of wood. The lower box includes a breadboard that connects the Vcc and GROUND to these LDs, as well as to eight LDMs (Laser Detector Modules) that sit on top of the upper of the bottom two slabs. The LDMs are aligned such that the beams of light from the LDs hit the photodiode of each LDM directly. Also connected to the breadboard are the outputs of the LDMs (which produce a high signal when a laser is hitting the photodiode, and a low signal otherwise). Vcc, GROUND, and the eight LDM outputs are all connected to the the FPGAs external GPIO_0 pins via a 10-pin ribbon.

3. Report on Success

Overall, the outcome of the design was a success. There was nothing in particular that we did not accomplish from our goals. All the logic and physical design worked as required and no unexpected failures occurred during the presentation.
There are aspects in which we could have improved; however, given the time and information, we were able to achieve all of our desired logic and create a well-functioning and aesthetically pleasing design.

An obstacle that was overcome with extreme precision was aligning the lasers with the photodiodes of the LDMs. With a distance of 12 inches between them, the diameter of the photodiode was very small and a miniscule change in angle of the LD would drastically change the end location of the beam of light.

One feature that we failed to implement (but was not part of our objectives) was displaying the notes currently playing on-screen via the VGA adapter; however, due to limited time we were not able to create a version that was fully functional and thus decided against showcasing it in the presentation.

4. What would you do differently?

If we were to do re-do this project, we would make sure to figure out in advance exactly where to get all of the necessary equipment (i.e. LDs, LDMs, etc.). Originally, our plan was to ship the equipment from the United States, as we had found a website that seemed to have everything we required; however, do to shipping issues, the parts arrived far later than anticipated. Had we known that nearby there was a store that sold everything we required, we would have been able to complete the physical design much quicker and possibly add more features (such as VGA).

On that note, even though VGA was not part of our main goals, it would have been nice to have an on-screen display of the current notes being played. Since VGA is a separate component to the design and does not require other logic to be completed, had we had one more day we probably could have sorted out the bugs and implemented the feature.
We played around with the idea of storing .mif files in memory to allow the harp to play various sounds that were not simply square waves (e.g. the sound of an actual harp string being plucked). If we were to start over with the knowledge we currently have of the project, this is a feature we definitely would attempt to implement.

To conclude, although it would have been nice to implement all these different features, given the limited timeframe of the project, we were satisfied with the outcome of our project.