Final Project Report

Instrument Tuner

List of Authors (Essmat Almufti, Michael Bowers, Maryam Subhi, Dani Yakoub)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: essmatalmufti@oakland.edu, mkbowers@oakland.edu, subhi@oakland.edu, daniyakoub@oakland.edu,

Abstract—**Instrument tuners detect the pitch frequency of a sound wave and displaying the detected musical note to the user. This project explores the implementation of a microphone to schmitt trigger comparator to produce a pulse wave whose frequency is detected by a digital circuit implemented on a Nexys 4 (Artix-7 based) FPGA development board. The associated note is then displayed to the user. Results show that our circuit detects notes to a high degree of accuracy. Optimization of the microphone/comparator circuit is needed to improve the project suitability for practical applications.**

I. Introduction

This report will cover the background knowledge and methodology behind the design of the instrument tuner. Included is documentation of our project development and experimentation process as well as our obtained results and conclusions.

Our goal for this project was to utilize an audio source as the source input for a digital system, for which an analog-to-digital converter solution would be required. We determined that an instrument tuner would be a suitable project, as commercial tuners are usually designed to pick up external audio via microphone. We used a microphone /amplifier to schmitt trigger circuit to implement a one-bit analog-to-digital converter. Our digital circuitry utilizes pulse detection and counters to detect incoming note frequencies, and 7-segment serialization to display note conditions to the user.

II. Methodology

A. Background

In Western music theory there are a total of twelve notes. Each note represents a perceived pitch and describes a set of frequencies corresponding to an international pitch standard defining an A note above middle C as 440Hz. An octave of any note refers to the doubling of the original note's frequency. For example, 220Hz, 440Hz, 880Hz and 1760Hz are all A notes belonging to different octaves.

Note intervals/ratios are based on an equal-temperament system that divides an octave into twelve equal steps on a logarithmic scale. Reference [1] displays a frequency chart for notes based on an equal-tempered scale - this chart was used to gather standard note frequencies for future calculations.

B. Design

An external breadboard circuit (see Figure 1) was used as an analog-to-digital converter for our input audio. The breadboard was powered by a USB powered HW-131 power supply with voltage regulation for simultaneous 5.0V and 3.3V supply. An LM386 audio amplifier was used to amplify the audio signal from an electret microphone. The datasheets of both ICs used were consulted to determine the pinouts of each. We utilized the typical application circuit from the LM386 Datasheet [2], adding a 10 uF capacitor in between the gain control pins to set the gain value to 200. A 10uF capacitor was also added to the non-inverting terminal as a coupling capacitor to remove DC voltage bias. The LM393 comparator is configured as a schmitt trigger [3]. A voltage divider sets the voltage at 2.5V on the non-inverting terminal. The 470K resistor is used to apply hysteresis to the circuit which changes the threshold voltage whenever the LM393 output changes. When the LM393 output is low, the 470K resistor is effectively in parallel with the lower 100K resistor, which sets a lower threshold voltage at:

$$VL = 5.0V \frac{(1/100K + 1/470K)^{-1}}{100K + (1/100K + 1/470K)^{-1}} = 2.26V$$

When the LM393 outputs high, the 470K resistor is effectively in parallel with the upper 100K resistor, which sets a higher threshold voltage at:

$$VL = 5.0V \frac{100K}{(1/100K + 1/470K)^{-1} + 100K} = 2.74V$$

Reference [4] was used to understand the functioning of the schmitt trigger. Hysteresis ensures that once a threshold voltage is met, AC noise or other waveform irregularities are unlikely to retrigger an unwanted state change as the threshold voltage changes according to the new output state.

This feedback control is crucial for producing a consistent pulse wave. A 10K pullup resistor was placed on the output pin of the LM393 comparator to bring logic high to 3.3V, corresponding to the 3.3V logic standard utilized by the Artix 7.

A two-bit right-shifting shift register was utilized as a pulse detector. An output Q = '1' when the MSB = '1' and LSB = '0', signifying the start of the period of the pulse wave. Output Q is used as a reset for a counter.

On the Nexys 4 we used a clock frequency set 50 MHz to catalog note frequencies as an associated integer quantity. We used this formula to determine the number of clock cycles that occur during one period of a note:

$$\frac{Note\ Frequency}{Clock\ Frequency} = \#\text{ of clock events}$$

The counter counts the number of clock cycles until it is reset again, signifying that one period of the pulse wave has elapsed. When reset = '1' the final count is exported as an integer output dataout and the count is reset back to zero. The final count is fed into a note decoder containing 25 conditional statements of ranges of counts: 12 corresponding to each note, 12 "in-between" conditions when the user plays in between notes, and an "error" condition when no other condition is met.

Range values for each condition were calculated according to the logarithmic equal-temperament scale of musical notes. Reference [5] shows a diagram of octave division in equal-temperament and associated equations. A musical "cent" is a unit of measure used to define the distance between musical intervals. In western music all twelve notes are evenly spaced 100 cents apart over an octave on a logarithmic scale (1200 cents makes an octave). A cent is calculated as:

$$cent(s) = C = 1200 * log_2\left(\frac{f1}{f0}\right)$$

The upper and lower bounds for each note were defined as +/- 5 cents from the note's standard frequency. For a G note at a frequency of 392.0 Hz:

Upper Bound Count (UB):

$$+5 = 1200 * log_2\left(\frac{392.0\,Hz}{f0}\right) \quad f0 = 390.87\text{ Hz}$$

$$UB = \frac{50.0\,MHz}{390.87Hz} = \mathbf{127920}$$

Lower Bound Count (LB):

$$-5 = 1200 * log_2\left(\frac{392.0\,Hz}{f0}\right) \quad f0 = 393.13\text{ Hz}$$

$$UB = \frac{50.0\,MHz}{390.87Hz} = \mathbf{127184}$$

The ranges for the "in between" regions are defined by the count values in between the UB value of one note and the LB value of the next note. Below is a sample figure of a few note conditions. The ranges for the notes are more narrow than the in-between regions because we want the user to be able to accurately reach notes (see Figure 2 for a complete chart of calculated count ranges) .

Note/Condition	Frequency Range	Count Range
E4	328.688 to 330.592	152119 to 151224
In between E4 and E4	330.592 to 348.232	151243 to 143583
F4	348.232 to 350.242	143582 to 142574

.

When a condition is met, bit values are assigned to four logic vector (6:0) outputs (ss0, ss1, ss2, ss3) corresponding to a 7-segment output configuration. The table below shows a sample of our display output scheme.

Cond.	Disp 0	Disp 1	Disp 2	Disp 3
Bb	-------	-------	"B"	"b"
In between Bb and B	"B"	"b"	"-"	"B"
B	-------	-------	-------	"B"

Since only one 7-segment display configuration can be displayed at any time, a 7-segment serializer (display.vhd) was implemented to make multiple configurations visible to the user, utilizing a serialization technique based on a clock divider as was seen in the course.. An internal clock divider divides 50 MHz into four different 4ms intervals, where each interval has an associated clock_out signal. 4ms was used as 4*4ms totals to 16 ms. The 60Hz refresh rate standard for displays corresponds to a 16.67ms image time. Signal clock_out is used as a data selector for the four logic vectors (ss0, ss1, ss2, ss3) holding the bit configurations for its corresponding 7-segment display. Due to limitations in 7-segment display configuration, a common "9" configuration was used to display a G note, and a common "8" was used for a B note. Figure 3 shows our full block diagram of the digital circuit.

III. Experimental Setup

Prototyping of the external circuit was done using a solderless breadboard. A common multimeter and a

Picoscope 2000 series were used to verify the functioning of the external circuit. For our setup we expected the pulse wave frequency to be equal to the frequency of the source note being played. We also expected the voltage swing of the LM386 to be within 0 to +5V. For the LM393 comparator we expected the logic low state to correspond to 0V and the logic high state to +3.3V due to the pull-up resistor. Vivado was used to synthesize and implement the VHDL code and generate a bitstream to program the FPGA on the Nexys 4 board. We used a test bench to confirm the reset action of the two-bit shift register functioning as a pulse detector and a reset for the counter, where output Q = '1' when the logic state of the pulse wave switches back to '1', indicating the start of a new wave period. Testing of the circuit was used with a frequency generator on an IPhone app, and an arduino-based Theremin was used as a test instrument.

IV. Results

Figure 4 shows voltage readings from the Picoscope 2000 series when an 880Hz sine wave is played into the microphone. The output voltage of the LM386 amplifier is shown in red and the output voltage of the LM393 comparator is shown in blue. Saturation is expected from the LM386 output because of the gain value set to 200. We utilized a very high gain to ensure that the output voltage swing would be greater than the voltage range set by the lower threshold (2.26V) and the higher threshold voltage (2.74V) of the schmitt trigger. The bottom of the figure shows specific values measured: the frequency of the digital output is measured at an average of 880.8 Hz, Vmax = 3.238V and Vmin ranges from -305.2 uV to 44.71 mV, which is effectively 0V for our logic low state.

https://youtu.be/yW7LucIZ39Q

The link above is to a video demonstrating the functioning of the instrument tuner. Our design displays the note/condition corresponding to the note being played off of a sine wave generator app off an IPhone. Our results suggest that our external circuit is accurate enough to convert a sinusoidal wave into a square/pulse wave with nearly the same frequency as the input wave, and that our circuit correctly selects a note condition and selects through each seven segment display to display our desired results. Testing with the theremin was largely unsuccessful, as we found that the source audio needed to be very loud in order to get readings.

.

Conclusions

We believe our circuit demonstrates the feasibility of using a schmitt trigger as an ADC for an external sinusoidal audio source. However, we have also determined that the project as it currently stands is not feasible as a practical instrument tuner and that there are some are a number of potential improvements and additions that could be made into this design.

We reasoned that the bottleneck of our design lies in the external circuit. During testing we noticed that we needed to either put the source audio as close as possible to the microphone, or increase the volume of the source audio to impractically high levels to ensure a consistent sinusoidal wave. We partly used a high gain setting on the audio amplifier to combat this issue. This prevented us from doing more faithful tests with the theremin or with other instruments such as a guitar, which was one of our early working goals that we later scrapped. We could also combat the issue by using a higher quality condenser microphone, as our microphone selection was based on availability and not quality.

An issue that wasn't debugged in our digital circuit was that although we intended our circuit to work for frequencies from 250-500 Hz, our circuit worked for frequencies from 500-1000. We designated our clock frequency as 50MHz, but it's possible that during testing/demos our clock frequency was still set at the native frequency of 100MHz, which would result in our note decoder selecting for notes in the octave above the octave range we specified. This issue however illuminated the advantage of using higher clock frequencies to run the counter, as we're able to achieve wider final count ranges for higher frequency notes. We did not incorporate multiple octaves into our counter, as we figured that higher octave conditions would be more difficult to meet as count value ranges get smaller as notes rise in frequency and have shorter periods. It is theoretically possible to add multiple octaves however by dividing or multiplying our current final count values by 2 to scale them up or down to the next octave. Using an index to scale our values up and down was not possible as our count values are whole integers.

We also would have liked to implement the external circuit onto a soldered proto-board to make our circuit more portable and better mimic commercial instrument tuners with smaller form-factors, but we decided this was not necessary.

References

[1] Michigan Technological University (1998). Tuning. Retrieved November 20, 2020, from https://pages.mtu.edu/~suits/notefreqs.html

[2] Texas Instruments (2017). LM386 Low Voltage Audio Power Amplifier. Retrieved December 6, from www.ti.com/lit/ds/symlink/lm386.pdf

[3] Texas Instruments (2018). LLMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators. Retrieved December 6, 2020 from https://www.ti.com/lit/ds/symlink/lm393-n.pdf

[4] A. Kay, T. Claycomb (2014) “Comparator with Hysteresis Reference Design.” Retrieved December 6 2020 from https://www.ti.com/lit/ug/tidu020a/tidu020a.pdf

[5] Georgia State University (2016). Cents. Retrieved December 11 2020 from http://hyperphysics.phy-astr.gsu.edu/hbase/Music/cents.html

FIGURE 1: External Circuit

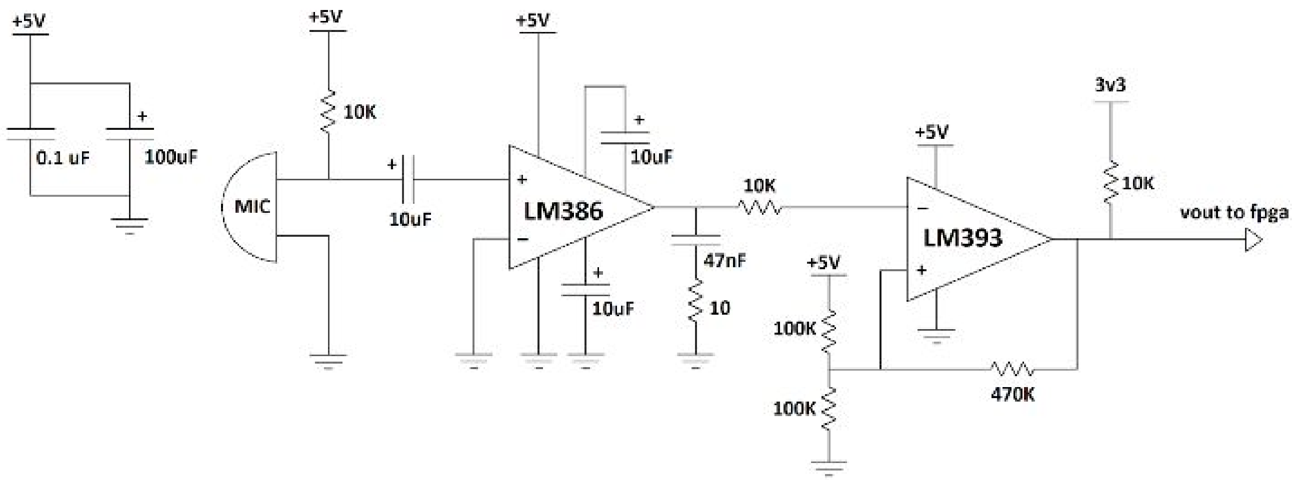

Figure 2: Condition Chart

CONDITION	CLOCK CYCLE	RANGE
C	191092	191658-190549
Between C &Db		190548-180987
Db	180388	180986-179856
Between Db & D		179855-170515
D	170265	170514-169764
Between D &Eb		169763-161174
EB	160704	161173-160247
Between Eb&E		160246-152120
E	151685	125119-151244
Between E&F		151243-143583
F	143172	143582-142575
Between F& Gb		142575 135540
Gb	135139	135540-134745
Between Gb&G		134744-127921
G	127551	127920-127184
Between G&Ab		127183-120737
Ab	120395	120736-120043
Between Ab&A		120042-113965
A	113636	113964-113059
Between A&Bb		113058-107570
Bb	107259	107569-106950
Between Bb&B		101507-100948
B	101239	101507-100948
Between b&C		100947-95830

Figure 3: Block Diagram

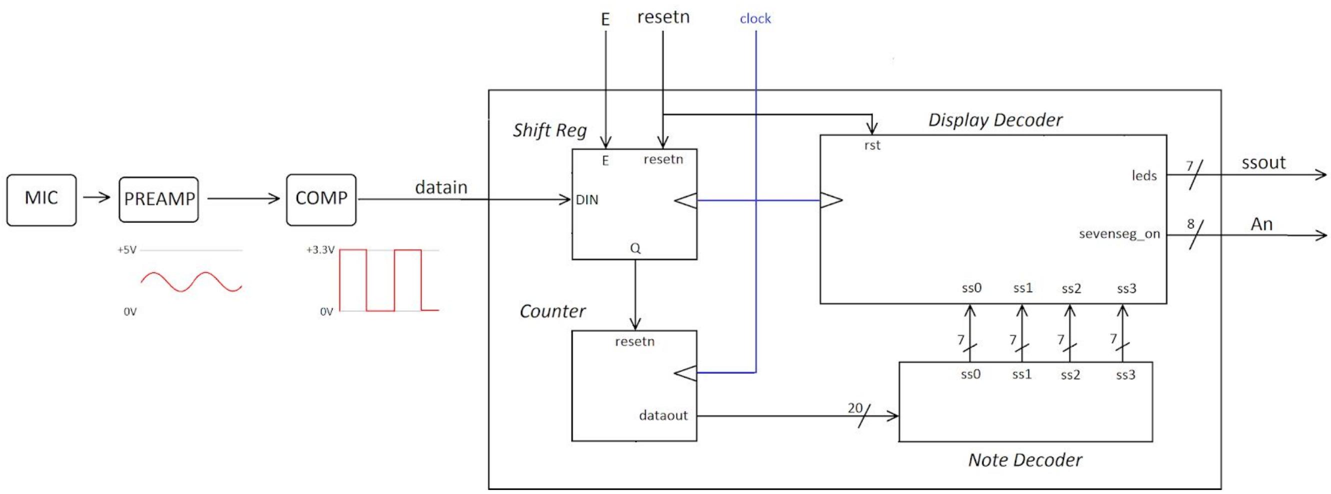

Figure 4: Picoscope Readings