6.101 Final Project Report

Xavier Hubbard, Kayla Statham

May 15, 2014

Contents

1 Introduction(Kayla)

A light organ is a device that takes an audio signal and produces rythmic light effects from it. These devices were extremely popular in the 1970s at discos and other party venues. They are also still quite popular as do-it-yourself projects for home use. However, these devices can easily become unintersting, and so our project was to create a novel light organ with extended features. These features included the use of RGB LEDs that have the ability to change colors, a touch switch that would control the on off state of the entire device upon contact, and a power converter so that the proper amount of voltage, 12V, could be drawn from the wall to run the LED strips. The entire block diagram for our device is shown below.

2 Light Organ

The main portion of the project stemmed from the light organ. The aim of this module was to take in audio, separate the signal into three sections, high, mid, and low, check the relative loudness of the signals, and finally use the checked signals to drive a set of RGB LED strips. The light organ also had the ability to cycle through different colors over time. The light organ received the audio signal via a 3.5mm jack, which could be plugged in to most electronic devices. This signal was DC biased around 6V, and then put through three different second order filters to create the three specific frequency bands. Afterwards the signal was amplified for easier use. To check the loudness of the signals, they were each put through a series of comparators, whose outputs would drive specific strips of LEDs within one frequency section. In order to allow for color oscillations we used a 3-inverter ring oscillator in combination with a series of AND gates to individually drive each color in the LED strips on and off at varying frequencies.

2.1 Filters(Kayla)

There were three filters in the circuit, a high pass for high frequency instrumentation, band pass for mid frequency instrumentation, and low pass for bass. The high pass filter used was a second order Sallen-Key configuration with a cutoff frequency of 1.5 kHz. With a Sallen-Key filter, the cutoff frequency f_0 is given by:

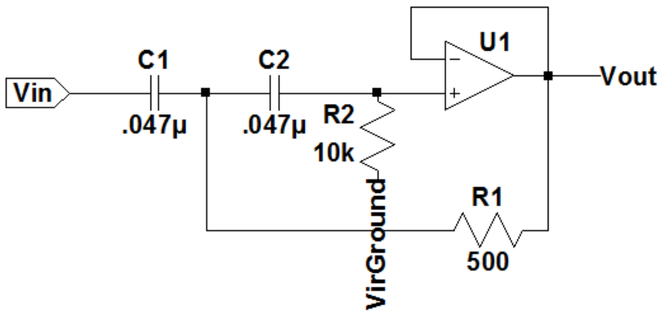


Figure 1: Highpass Filter

$f_0 = \frac{1}{2\pi\sqrt{R_1R_2C_1C_2}}$

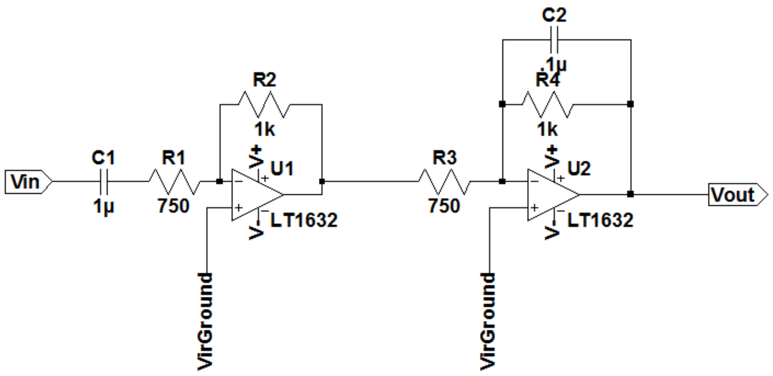


Figure 2: Bandpass Filter

The band pass filter was a second order filter constructed of a first order high pass filter and first order low pass filter. The cutoff frequency for the high pass filter and low pass filter are given by:
$f_{highpass} = \frac{1}{2\pi R_1C_1}$
$f_{lowpass} = \frac{1}{2\pi R_4C_2}$

This will give the bandpass filter a low frequency cutoff of 212 Hz and a high frequency cutoff of 1.6 kHz. In practice, this filter had roughly the same bandwidth, but it was shifted lower down the frequency spectrum, most likely due to the coupling capacitance used to bias our input signal around 6V. The low pass filter was also a second order Sallen-Key configuration, and

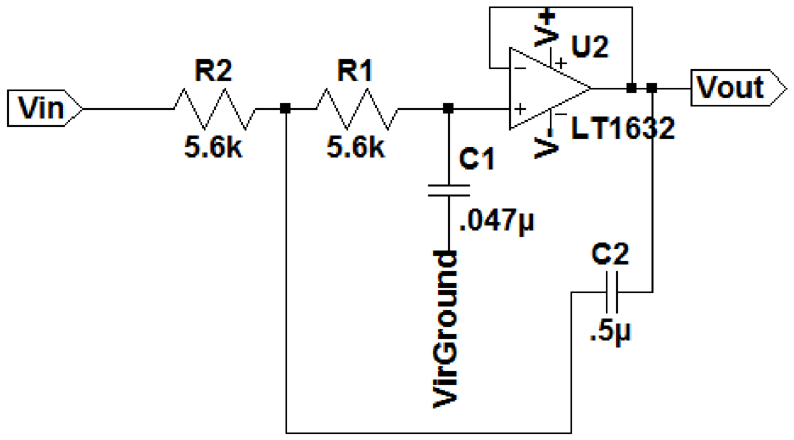


Figure 3: Lowpass Filter

was set to have a cutoff frequency of 185Hz. These frequency sections were chosen based on the frequency spectrum of different instruments common in music. Instruments like drums, bass guitar, and parts of the keyboard all fall in the range of 185 Hz and below, whereas instruments like violins, winds, and some synthesizer sounds fall in the range of 1.4 kHz and above.

2.2 Amplifiers(Kayla)

From the filters, the signal is then amplified for easier use later down the signal path. Unamplified, the voltage levels for most songs was on the order of tens of millivolts. We used an inverting amplifier configuration to increase this to volts. Each amplifier had a gain of 500, so that we could reach peak to peak voltages of a maximum of 3-4V, so as to amplify the signal without railing to either 12V or ground. We were able to use an inverting amplifier

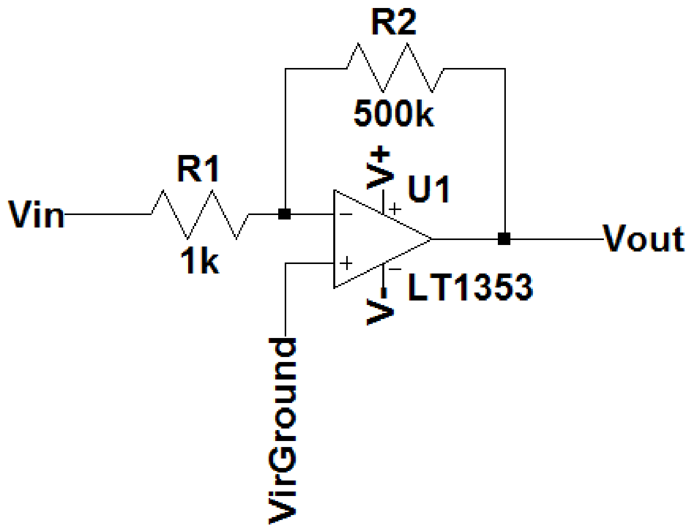


Figure 4: Inverting Amplifier

even though we had no negative supply by setting up a virtual ground at 6V, where our input signal was originally biased. This allowed us to invert around 6V rather than ground, and so our signal was amplified without a need for a negative power supply. We also chose to have the amplification stage after the filtering stage in order to not amplify any noise present in the original signal.

2.3 Comparators(Kayla)

After amplification, the signal was then put four different comparators so that we could check the loudness of the signal, and if it was over a certain threshold, it would light up an individual strip of LED's for a specific frequency band. The thresholds were set up in such a way to mimic a log scale. This was because of the way our ears percieve loudness, which is a logarythmic relative relation. The thresholds we used were 7.5V, 8.3V, 8.7V, and 9V.

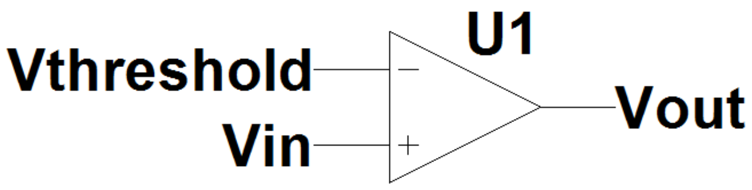


Figure 5: Comparator

2.4 Oscillator(Xavier)

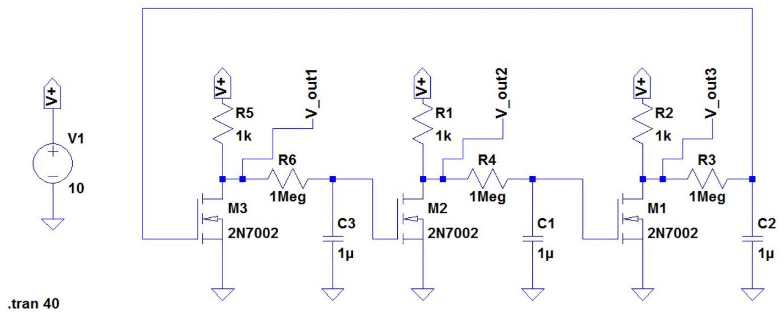


Figure 6: Ring Oscillator

To successfully produce color oscillations we needed to drive each of the red, green and blue sections of the LED strip individually. We created this effect by utilizing a simple 3-inverter ring oscillator shown in Figure 6. We can create different waveforms that are produced thanks to the astable nature of an odd number of inverters. Simply put, the inverters are constantly turning each other off and on, since there is no stable configuration. There is a threshold gate voltage that a MOSFET needs to turn on, so it is not instantaneous. By adding in a resistor and capacitor network, we can slow the rate at which voltage changes across gate thereby delaying the rate at which the inverters turn on. The output waveforms of the ring oscillator can be seen in Figure 6. As seen, there are three different wave forms with the same asymmetric duty cycle of approximately 33% all $\frac{2\pi}{3}$ rad out of phase with one another. The construction of the ring oscillator was relatively simple and posed no major hiccups.

2.5 Drive Stage(Xavier)

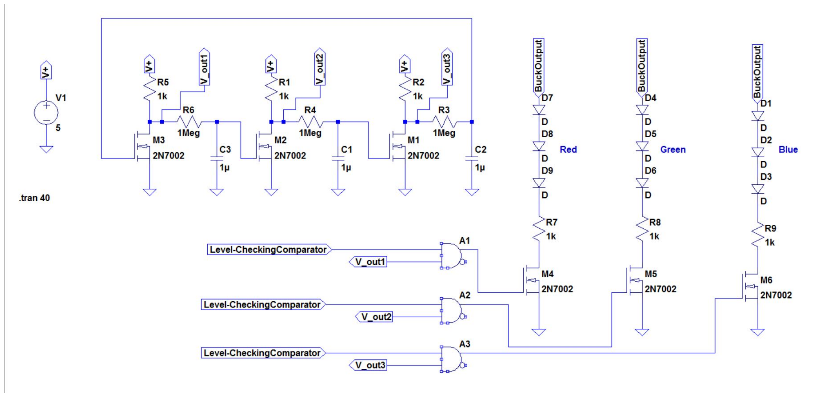


Figure 7: MOSFET Driver

To utilize the waveforms produced by the oscillator, we fed the output of our amplitude checking comparators into an AND gate with each of the 3 oscillator waveforms. Since one LED strip was used to represent a single frequencys amplitude, the same level-checking comparator output was ANDed with the 3 different outputs of the ring oscillator. To turn on the lights themselves, we used a 2N7000 N-Channel MOSFET as an on off switch whose gate voltage was controlled by the output of the AND gates. Each AND gate was used to drive one color in one LED strip effectively assigning 3 gates to each LED strip for a total of 36 AND gates. Therefore, the schematic shown for a single LED strip driver shown in Figure 7 was to be repeated a total of 12 times. Due to time constraints and various debugging problems we were only able to repeat this process 4 times and completing only one of the frequency bands. However, the versatility of the circuit schematic allowed us to quickly switch between the frequency bands that we wanted to show for the checkoff. The most difficult part about the drive stage was the sheer magnitude of the things to be wired. Furthermore, it couldnt really be tested until all previous stages were up and running which meant that most of the wiring had to be done without test. Thankfully, it worked on the first try!

3 Buck Converter(Xavier)

The LED strips that we chose to use had a very specific power requirement of 12 Volts. Given that our supply voltage was at 20V we needed some way to step down our voltage. Instead of using a simple linear regulator, we chose to go with the more efficient and more difficult buck converter.

3.1 Buck Converter

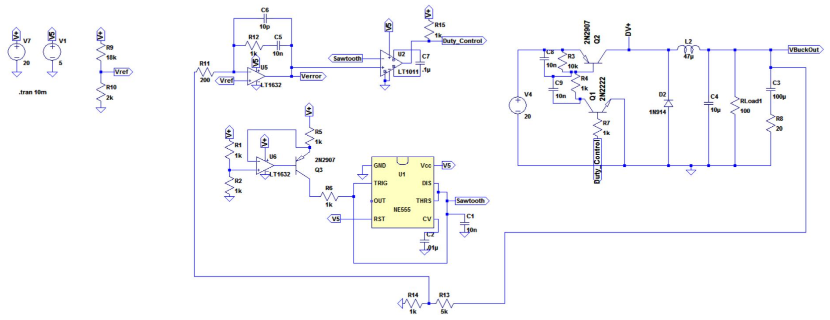


Figure 8: Buck Converter

Buck Converters are commonly chosen as DC-DC converters due to their high efficiency. Where non-switching converters waste all unused voltage as heat, switching converters take a different approach. You can see the general form of a buck converter in Figure 8. There are 6 main components. The input voltage, V_{in}, the output voltage across the load, V_{out}, the switching transistor, an inductor, a capacitor, and a freewheeling diode. This gives the simplest layout for a buck converter. It performs by chopping the DC voltage, V_{in} by turning the transistor on and off with some duty cycle, D, at some frequency f_s. We then use an LC filter on the chopped waveform to produce a DC voltage whose value is found by $V_{out} = DV_{in}$. The purpose of the freewheeling diode is to allow the current in the inductor to continue to flow even when the switch is off. The values of the inductor, capacitor and switching frequency were all chosen to be big with $C = 10\mu F$, $L = 47\mu H$ and switching frequency of $f_s = 300kHz$. This, however, is a very simplified explanation of the buck converter, but is sufficient for the purposes of our project.

3.2 Feedback Control

While the general buck converter seems quite straightforward, it does not perform the same under varying loads. It needs a feedback loop in order to maintain a constant output voltage under varying loads. Implementing this using a microcontroller or digital means is straightforward, however doing it with just analog components proved to be a rather tricky process. The general Idea was to create a reference voltage V_{ref} whose value would be equal to V_{out} after it was attenuated by a voltage divider. The difference between the attenuated signal and V_{ref} was then fed into a PID controller to produce V_{error}. To create the duty on/off duty cycle for the switch V_{error} was then fed into the positive terminal of a LT1011A comparator while a sawtooth waveform operating at f_s was fed into the negative terminal. The output of the comparator acted as the base drive voltage for the Sziklai pair that served as our switch. Each of these components will be explored in further detail below.

3.2.1 Generating the Sawtooth

The sawtooth waveform was generated rather simply using a 555 timer configured in its astable oscillator mode. The schematic for ths can be seen in Figure 9. The capacitor, C_1 was charged with the current from a voltage-controlled current source. The astable setup of the 555 shorts the capacitor to ground through a BJT after it reaches 66% of the supply voltage and allows it to charge again once it falls to 33% of the supply voltage. The sawtooth can befound simply from the differential equation, $I = C\frac{dV_c}{dt} \Rightarrow V_c = \frac{1}{C}\int Idt$ It can be seen that if I is constant then the voltage on the capacitor should be affine, which leads to the sawtooth appearance of V_c. With the proper choice of current we could control the time required to charge the capacitor which allowed us to set the frequency of the sawtooth to f_s. A current of approximately 10mA was used with a capacitor of 10nF which gave us an f_s of approximately 300kHz.

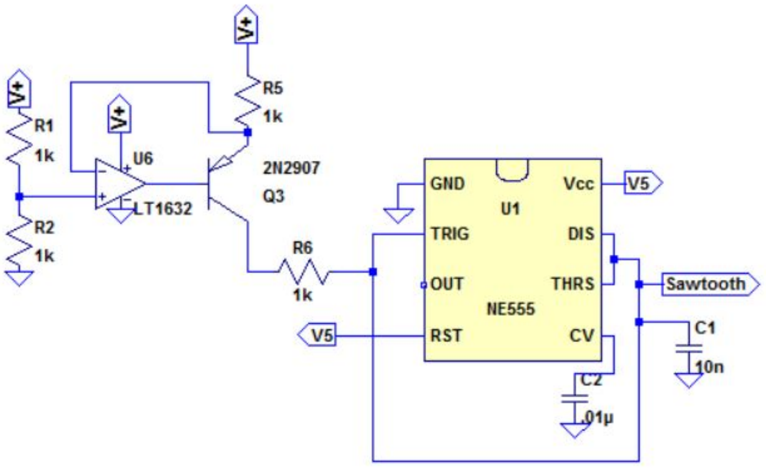


Figure 9: Sawtooth Generator

3.2.2 Compensator

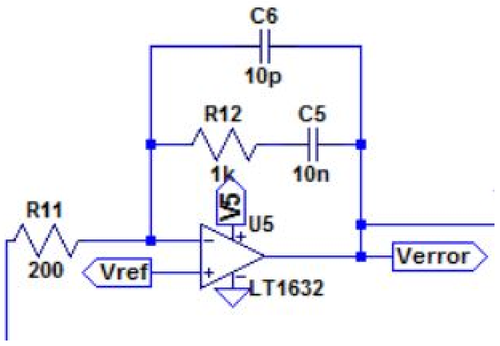


Figure 10: Compensator

A simple PID controller was used to mitigate errors on the output of the buck converter. Its transfer function can be seen below:
$\frac{1}{sR_{11}} \cdot \frac{R_{12}C_5s+1}{R_{12}\frac{C_5C_6}{C_5+C_6}s+1} \cdot \frac{1}{C_5+C_6}$

The utility of the compensator was simply to help improve the phase margin of the system and reduce steady state error. V_{error} is a voltage that represents the difference between V_{ref} and the attenuated V_{out}. The schematic for the controller is shown in Figure 10.

3.2.3 Creating the Duty Cycle

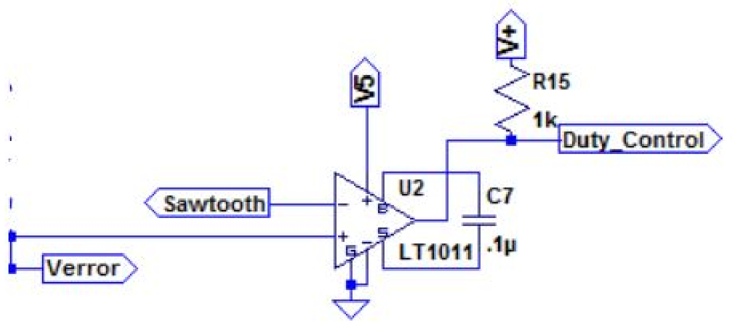


Figure 11: Duty Cycle Generator

It was previously mentioned that the duty cycle was created using a sawtooth waveform and a comparator. The idea behind this is actually rather ingenious. If V_{out} is too high then that drives V_{error} lower which reduces the duty cycle and pulls V_{out} lower. If V_{out} is too low then that drives V_{error} higher which increases the duty cycle and pulls V_{out} higher. The waveforms can be seen clearly in Figure 11

3.3 The Switch

The switch we chose to use was a Szikiel pair with current limiting resistors. This switch configuration was chosen thanks to the very small amounts of current required to turn it on. The configuration for the switch can be seen below. The base of the NPN transistor was driven directly with the output of the LT1011A comparator. Despite the low current required to switch on the transistor pair, the PNP transistor required a heat sink as all current that passed through the system was drawn through it. The schematic for the Szikiel pair can be seen in Figure 12

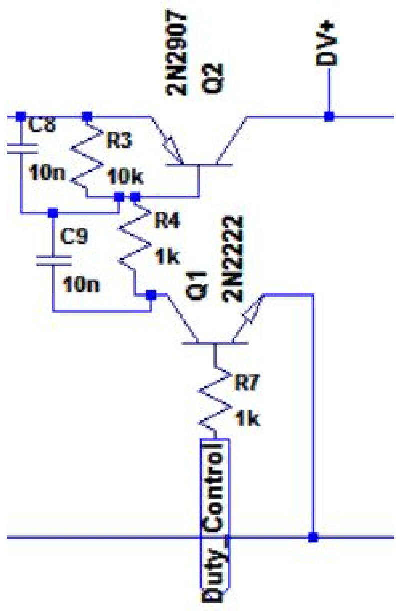


Figure 12: Szikiel Pair

3.4 Problems

I ended up making two different buck converters using the same schematic. The first was done on a breadboard and worked very consistently. However, to allow the final product to be more mobile we decided to use perfboards instead of breadboards. This was a mistake in our planning. Perfboarding is not only more time consuming, but recreating the buck converter on the perfboards proved to be challenging for unknown reasons. The sawtooth generated by the 555 timer worked sporadically without any changes to the circuit and was the source of most of the problems. Several different timers were used, but the problem persisted and a solution was not found. We could only get a consistent 12V about half the time, so for the purposes of the checkoff we chose to use the more consistent 12V supply from our labkits.

4 Touch Switch(Kayla)

The touch switch used in this project stems from a monostable oscillator configuration of a 555 timer, which is then split, with part of the signal being delayed to use as a clock signal, and the other being fed directly into a JK flip flop to change the state of the circuit. When the user touches the touch pad of the switch, a rising clock edge will be sent into the flip flop, as well as the inputs to toggle the state of the flip flop, which will turn the light organ on and off.

4.1 Monostable Oscillator

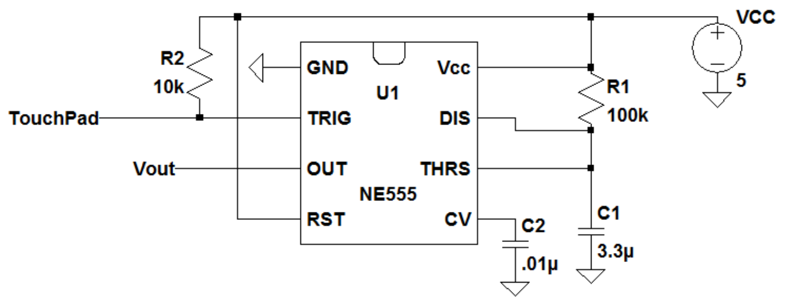


Figure 13: Monostable Touch Switch

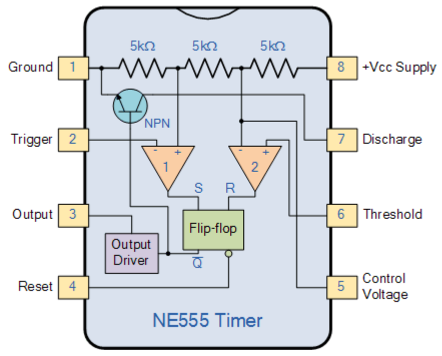


Figure 14: 555 Timer Internal Structure

The monostable oscillator is achieved by connecting a touch plate to the trigger (pin 2) of the 555 timer. When the plate is touched, a connection to ground is made, sending the trigger pin to zero. This will cause comparator one to output high, which activates the set feature of the SR flip flop. The flip flop will then set Q to high, and $\overline{Q}$ to low. $\overline{Q}$ going low will turn off the BJT that is shorting the discharge pin (pin 7) to ground, thus allowing the capacitor connected to discharge to begin charging up. Once the voltage across the capacitor becomes higher than 2/3 Vcc, comparator two will output high, causing the flip flop to reset, sending Q to zero, and turning the BJT back on. The capacitor will then discharge, and the circuit will then return to its initial and stable state until the trigger is touched again. The output of this oscillator will be a single square wave pulse of Vcc, or 5V.

4.2 Signal Delay

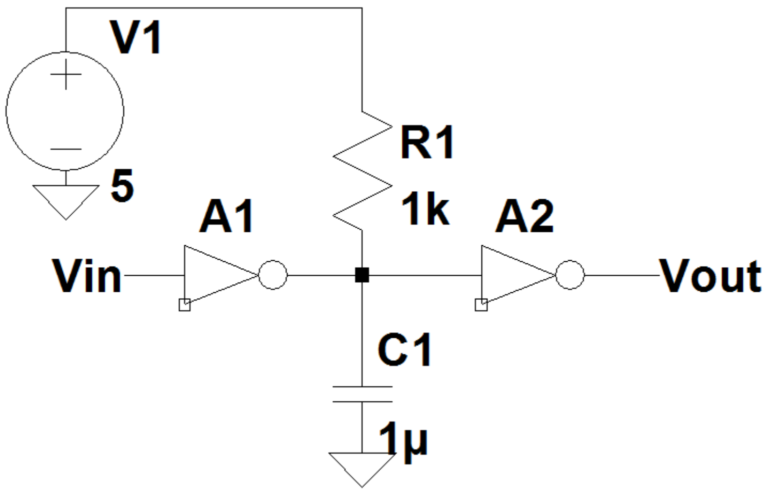


Figure 15: Signal Delay Circuit

The oscillator output is then fed into a series of inverters, which will delay the signal by a certain propagation time, and then fed into the last inverter via a capacitor. The time it takes to charge up the capacitor will further delay the signal, on the order of RC, and be output to the JK flip flop. The function of this delay is to ensure that there is always a rising clock edge shortly after the monostable oscillator registers a touch. This way, the flip flop will never miss an input touch due to there being no rising edge.

4.3 Flip Flop

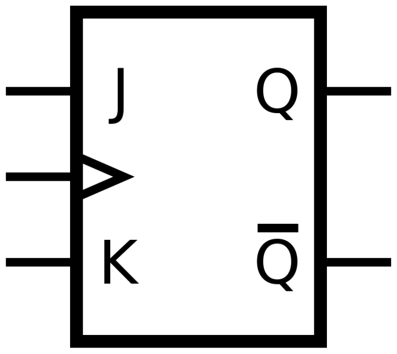
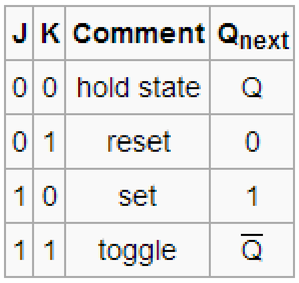


Figure 16: JK Flip Flop



J	K	Comment	Q_{next}
0	0	hold state	Q
0	1	reset	0
1	0	set	1
1	1	toggle	$\overline{Q}$

Figure 17: JK Flip Flop Truth Table

Finally, the undelayed and delayed signal are fed to the JK flip flop.Pins J and K are both tied to the output from the monostable oscillator. With this configuration, the flip flop will only ever toggle between high and low, or on and off respectively. The output of the flip flop will be either a constant 5V when the state is high, or a constant 0V when the state is low. This constant voltage is then supplied to a set of comparators that will output a constant 12V when the input voltage is greater than 2.5V, and a constant 0V otherwise. These comparators also have an additional push pull output stage of an npn BJT and pnp BJT tied together at their emitters. These BJTs serve to allow a significant amount of current draw without damaging the comparator op amps which have a set amount of current they can supply.

5 Conclusion(Xavier)

Overall, we were very happy with our final product. Unfortunately, we were pinched for time at the end and didnt manage to solve all of our issues. For unknown reasons, the output of the touch sensor module stopped working when tied to the inputs on a comparator. The 555 timer was finicky and only worked sometimes. One of the main issues that lead to our lack of time was the initial decision to use perfboards instead of breadboards for the project. Perfboards require much more time to be done properly due to the soldering requirements and the difficulty in correcting mistakes. As a result we lost a lot of time at the beginning of the project debugging systems that didnt work on the perfboard but did work on a breadboard. The only reason we were able to finish the project as much as we did was thank to the decision to abandon the use of perfbaords. Another issue that was present was the sheer bulk of the wiring that was required. Our original plan was for 5 different frequency bands and 5 levels which leads to 125 different LED driver circuits. We had a hard time even after scaling this down significantly.

Thankfully, with some slight modifications, our Light Organ worked as well as wed hoped it would. It correctly filtered sound, amplified it, and showed the amplitude of the sound in the number of LED strips that were lit in each frequency band. The individual modules worked separately but we were unable to make the work together. Given more time, it would have been nice to figure out how to make all of the modules work cohesively. Several improvements that could be made would be to allow the organ to take input from a microphone, to use the capacitive touch switch to control the brightness levels of the LEDs, the make the buck converter provide a variable output voltage instead of a static 12V to name a few. In the end we considered our project a great success and a wonderful learning opportunity that provides room for improvement while still giving us a solid final product.