
# ECEN 4013

## Final Report

MARCH 12

**Brady Mayfield**
**Michael Thompson**
**Moatasem Alsahafi**
**Ryan Fortson**

## Table of Contents

- ## Introduction

We were tasked with creating an analog-to-digital converter (ADC), which converts a continuous analog wave into a discrete digital representation. There are many different types of ADC's, such as successive approximation (SAR), delta-sigma (ΔΣ), dual-slope, pipelined, and flash. All of these deliver the same output but with different techniques and different structures. Certain types of ADC are not necessarily better than other types, it really depends on the situation and how it is going to be used. However, in our design, we used a successive approximation register analog-to-digital converter (SAR - ADC) and the main goal was to build an old-fashioned ADC, with no kind of microprocessor in it, which was capable of handling an input of 0-5V.

- ### Algorithm and theory:

The successive-approximation analog-to-digital converter circuit typically consists of four main sub-circuits: sample-and-hold circuit, a digital-to-analog converter, SAR, and a comparator of some sort. To begin, the sample-and-hold circuit grabs an analog input and holds it until it is told to grab another sample. This value will be applied to one terminal of the comparator, the other terminal of the comparator is a changing value that comes from the internal DAC. The first value from the DAC is always 1000, which correlates to 2.5V in a 5V reference system. If the sample value is greater than 2.5V, then the most significant bit stays 1. If the sample value is less than 2.5V, then the most significant bit is set to zero. This process repeats all the way down to the LSB, and the digital output is updated. This is just a high-level overview of the system, more technical details will be discussed in the sections for each individual part and the problems that each part had when they are integrated together.

- **Team structure and tasks:**

  No matter what is listed below, all team members were present for the construction of the project and collaborated together to overcome problems that came up.

  - **Brady Mayfield: (electrical engineering and English minor)**
    1. Use Multisim to design and simulate DAC, voltage regulator
    2. Prototype the DAC individually to verify simulations
    3. Construct an external DAC for the purpose of displaying results
  - **Michael Thompson: (Electrical and Computer Engineering, Computer Science minor)**
    1. Use Multisim to design and simulate Control Logic and Registers
    2. Prototype Control Logic and Registers individually to verify simulations
    3. Integrate all parts with control logic
  - **Ryan Fortson: ( electrical engineering )**
    1. Use Multisim to design and simulate the clock and Schmitt Trigger
    2. Prototype clock and Schmitt trigger on a breadboard to verify simulations
    3. Design the PCB and manufacture it at Endeavor
  - **Moatasem Alsahafi: (computer engineering and mathematics)**
    1. Use Multisim to design and simulate sample and hold circuit
    2. Prototype sample and hold circuit individually to verify simulations

- ## Design constraints

  - ### Performance

    The completed ADC needs to be able to handle an input signal ranging from 0V to 5V at, ideally, a sampling frequency of at least 10k samples/second.

  - ### Environmental requirements

    The environment is not a big factor in this project since our circuit was demonstrated at room temperature and controlled humidity.

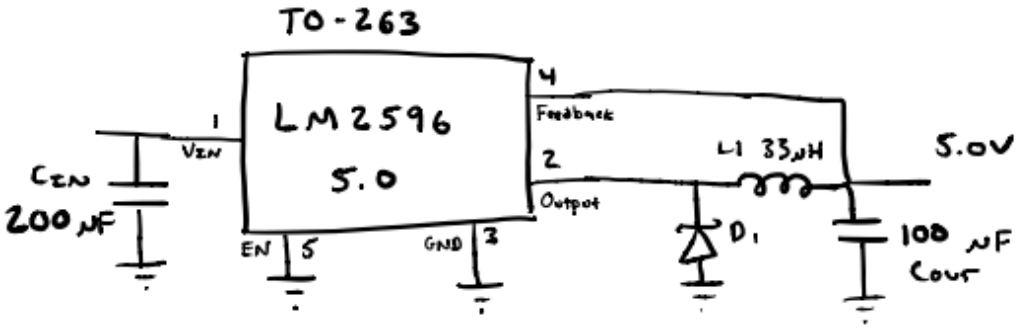
  - ### Applicable issues of public health

    Since Covid-19 is still a huge danger to public health, all the students are required to wear a mask all the time inside the campus. Also, all students have to follow OSU's health protocol.

- ## Design structure

  - ### Voltage Regulator - Brady Mayfield

Initially, we were not planning on using a voltage regulator in our system. The goal was to have everything powered by 5V. However, in the sample and hold circuit, we ran into issues where if the switching transistor only had 5V on the gate then at higher sample values the Vgs drop wasn't enough to switch the transistor. To solve this, we needed a 7V voltage source we could use to bias the transistor. We wanted to still only have one voltage input, so we selected a LM2596S-5.0 Voltage Regulator. This can take a wide range of inputs and step them down to 5.0V. It is capable of sourcing up to 3A if utilized correctly, so I knew it would be more than adequate for anything we would need it for.

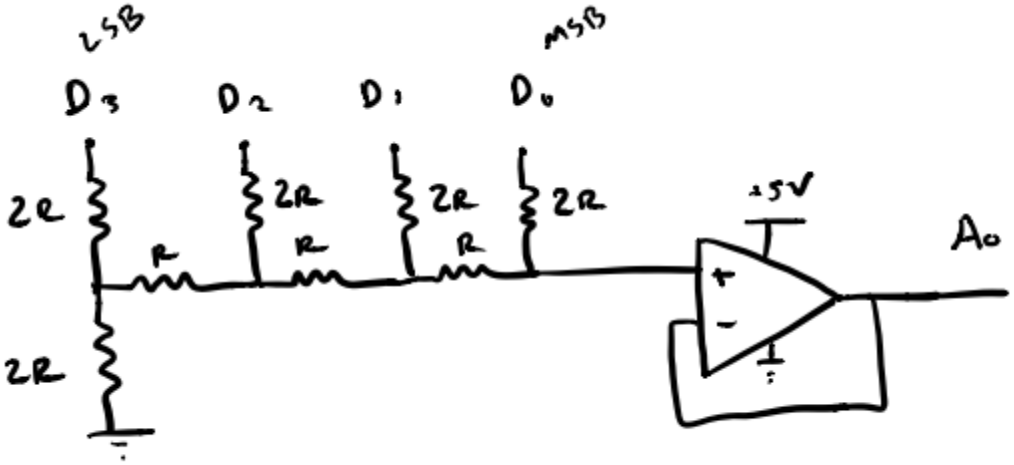



When physically implementing the circuit, we did run into a few small issues with the voltage regulator. I was under the impression this would be a load-independent voltage regulator, but we discovered that when we loaded the regulator more with additional circuitry the output dipped down to about 4.5V. This was easily resolved by upping the input voltage to 8.7V. If there was more time, it would have been good to create a more

reliable voltage regulator. This worked well for what we needed, though. A good feature of the voltage regulator, which may have saved our circuit a couple of times, was the protection circuitry. One time we had our input voltage and ground switched and, even though the voltage regulator got very hot, nothing else in the circuit got hot or was damaged at all.

- **DAC ( Digital to Analog Converter ) - Brady Mayfield**

The initial design for Digital to Analog Converter utilized what was essentially a weighted voltage summer, but this design had one big drawback: it required a wide variety of specific resistor values. It needed an R, 2R, 4R, 8R, and a 16R. We likely could have accomplished this by combining resistors in series, but in researching DAC designs I found another design: the R2R ladder design. This design only required two values for the resistor: R and 2R. The image below shows the schematic for the circuit. When implementing the circuit, 1kΩ and 2kΩ resistors were used.

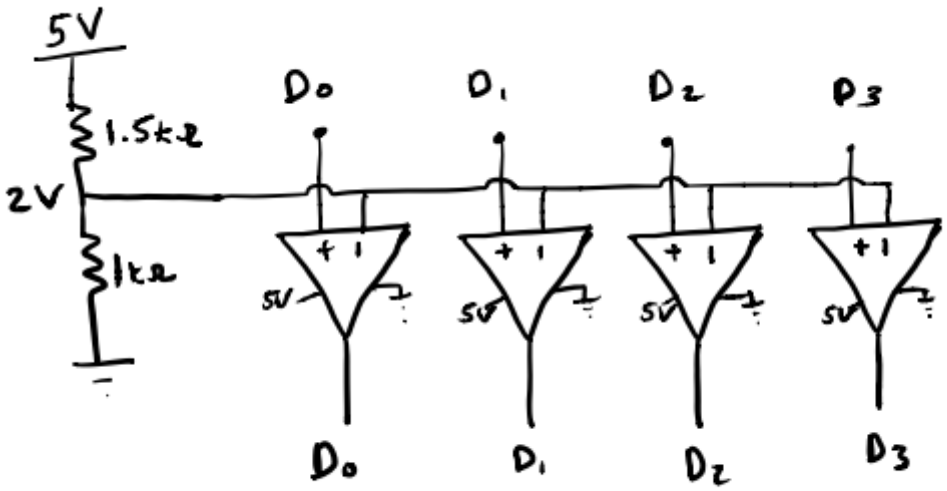


In simulation, this design worked perfectly when interfacing with the digital logic outputs. In reality, though, we realized that the DAC was loading down the digital logic and lowering the digital output voltage level to about 3.4V. For the DAC to be accurate, it is very important that each digital high corresponds to exactly 5V so that the weighted voltage sum is accurate.

Because we wanted to keep the DAC on the PCB, we first attempted to solve this issue by changing resistor values. We reasoned that if the resistor values were higher, they would present a higher input impedance and thus load down the digital outputs less. This was true, to a certain degree. We got our digital logic to output about 4.4V when doing this, but it also worsened the performance of the DAC significantly at higher frequencies.

The resistor solution did not work, so we decided to move the DAC off-board and add a comparator at each digital logic input to the DAC. This means that whenever that comparator receives a digital logic high, it swings up to its rail voltage which we set at 5V. This design is actually better, in our opinion, because it allows for a Vref that is not determined by our digital logic high. In other words, it would only be a few more steps to change this circuit to have a variable Vref set by the user. We did not do this for this project, because of time constraints and it was not required by the project, but it is a way

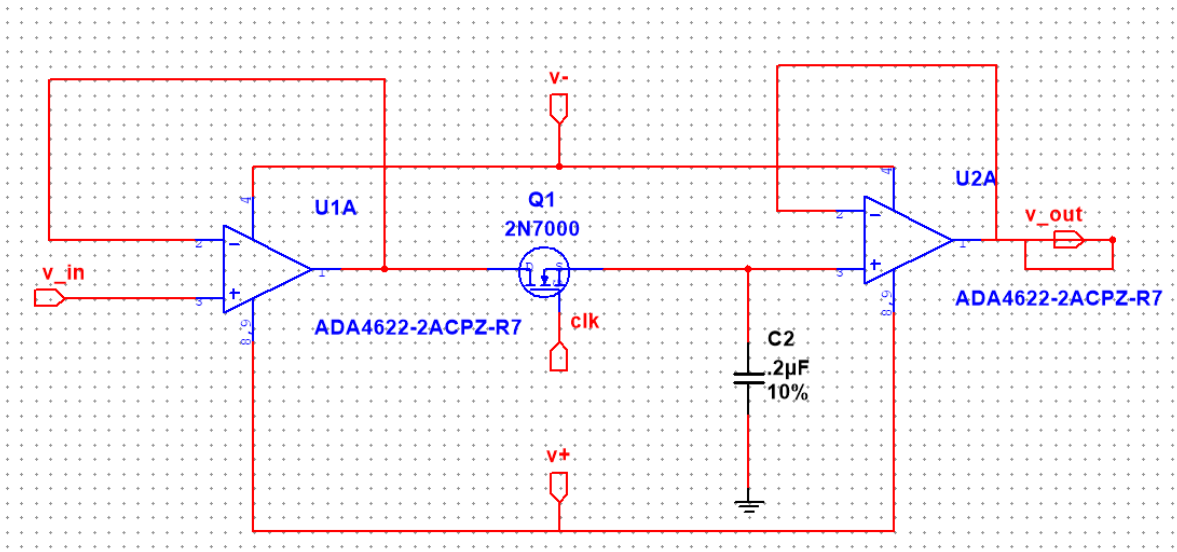
this project could be improved in the future.




Another way the DAC could be improved is by using better components. Dr. Gard pointed out during our presentation that the cause of some of our missing codes could be due to resistor tolerances. We also noticed during our testing that some of our data lines were noisy, which we now think was caused by the resistors. So, if we redid the project we would use higher quality resistors that specified they were low noise as ones with 1% or better tolerances.

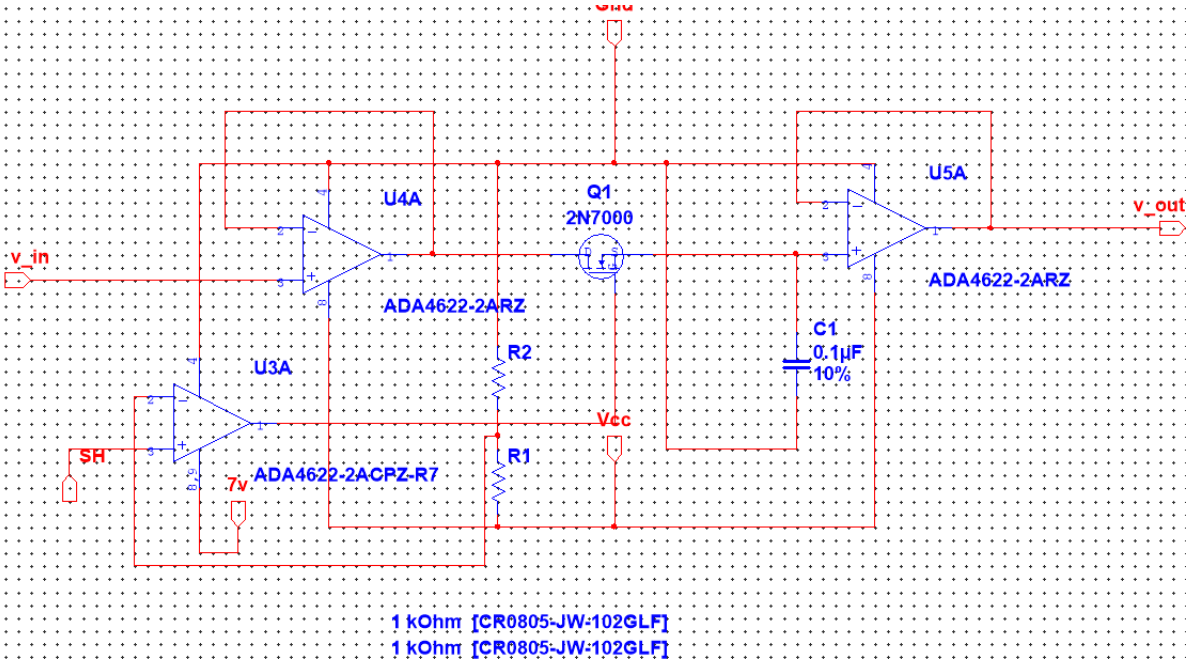
- **Sample and Hold** **-Moatasem Alsahafi**

In general, the job of the sample and hold circuit in the ADC is to sample the input analog signal and hold the sampled signal for n time that is wanted. There are many different techniques to build a sample and hold circuit, but in this circuit, we used two op-amps, a transistor, and a capacitor. The op-amps in the circuit work as a buffer, one is the input buffer and the other one is the output buffer. The input buffer is connected to the input signal that needs to be sampled and it transfers this signal to the switch which is the transistor and from the switch to the capacitor. The capacitor works as a hold, meaning the capacitor holds the analog voltage while the analog is converted to digital. Specifically, the sampled signal goes through the output buffer to the Schmitt trigger. As shown below:




This design worked in the beginning but when we added our circuit together, it did not sample more than 3v. The reason is that the transistor's gate was connected to a 7 voltage source when I did the testing on the circuit. Since our overall circuit use 5v system and the control circuit will only give 5v, the switch could not work as it has to be so I added a

comparator circuit to raise the voltage that goes to the transistor gate from 5v to 7v as shown below:



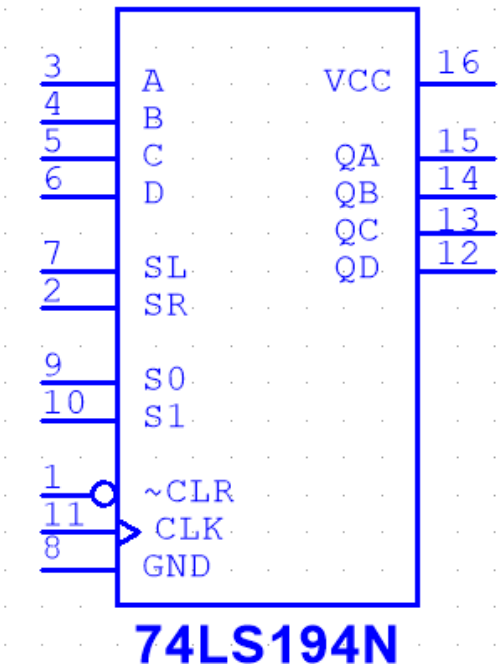

The comparator design here is similar to the one found within the DAC. Essentially, when the comparator receives a digital high it swings up to its voltage rail, which was 7V. The two resistors create a voltage divider that provides a Vref to the comparator of Vcc/2. If I am going to build a sample and hold circuit again, I would use a different switch that is more flexible that could work with different voltages.

- **Schmitt Trigger** **-Ryan Fortson**

The Schmitt trigger was built using an ADA4622 operational amplifier with a 10K ohm resistor going to the positive input, and the negative being connected straight to the DAC. The feedback resistor is a 1M ohm connected between the output and the positive input. In this configuration the op-amp will only output either 0 or 5 volts, it will only output the 5 volts if the voltage input passes the voltage reference, it will keep outputting the 5 volts until the voltage input falls under the voltage reference. The advantage of the Schmitt Trigger versus a classic comparator is that it has hysteresis. That is, the rise voltage is different than the fall voltage. The advantage to this is that it is more stable than a normal comparator when the value being compared to be very close to the Vref. With a comparator, the output value could swing back and forth from high to low in that situation. A schmitt trigger, though, should stay steady which was important for this project to be accurate.

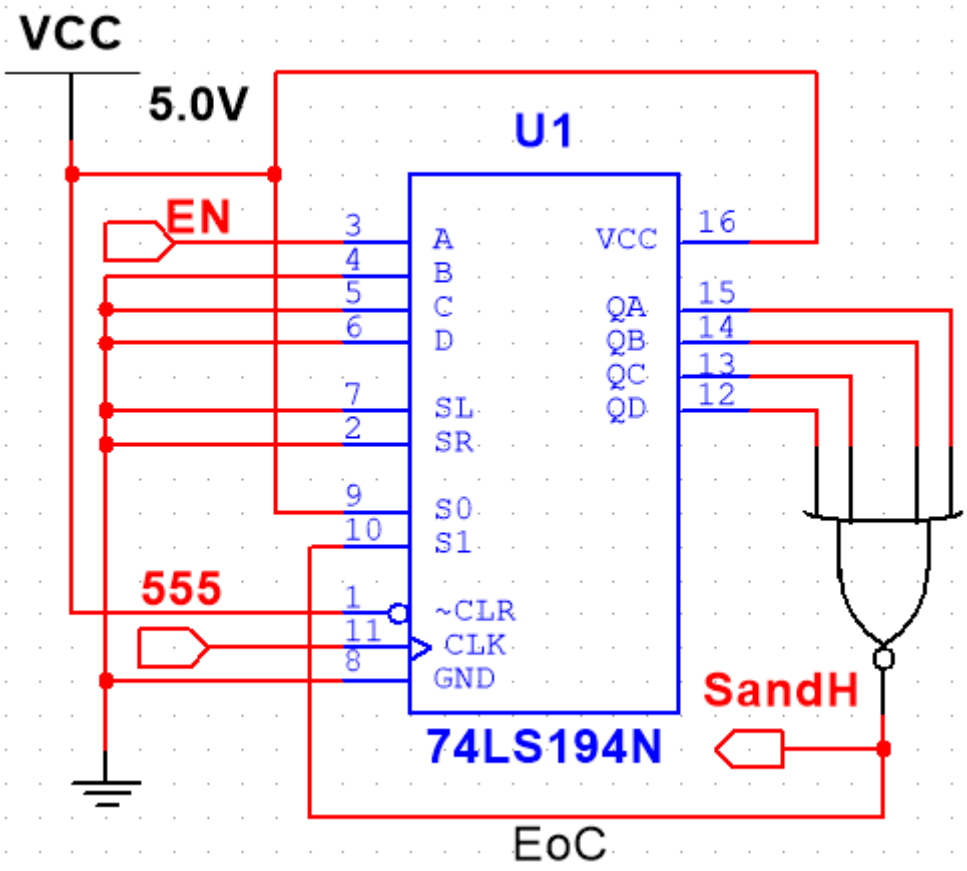
- **Control Logic (SAR)** **-Michael Thompson**

The control logic in the ADC sequentially sends signals to the DAC and then saves or rejects those signals based on the output of the Schmitt trigger. Additionally, the logic sends a signal to the sample and hold circuit in between each cycle. There are three main parts of the logic: The sequential signal, the save register, and the output register.




The sequential signal is straightforward and is implemented by a shift register (shown above). We needed the output (QA-QD) to be 1000 -> 0100 -> 0010 -> 0001 so that each bit could be sent to the DAC in the desired order. To do this, we needed an input before the shift could occur. To accomplish this, we used an external enable signal for pin 3 (A) of the shift register and grounded pins 4-6. At first, we had pins 4-6 floating, but that resulted in 1111 being loaded into the register instead of the desired 1000. Because 0 replaces the most significant bit we need SL and SR to be grounded. To put the register into shift left mode we pull S1 to ground and pull S0 high. To put the register into load mode we pull both S1 and S0 high. Therefore, we can connect S0 to power as it will always need to be high. We don't want the bit to be entered while there are other bits in the shift register so we connected pins 12-15 (register output) to a NOR gate and connect the output to S1 (This signal will be called the end of cycle signal or EoC). That way, when there is a bit in the register, S1 will be low and will be in shift mode and when the register is empty, the register will be in load mode.

If the enable bit is high, pins 15-12 will be 1000 -> 0100 -> 0010 -> 0001 -> 0000 for the next 5 clock cycles (From now on a cycle will be considered 5 clock cycles). However, if the enable signal is low the output will always be 0000. With this set up, the register never needs to be cleared which means we can set pin 1 (~clr) to always be high. When the output is 0000 no comparisons are being made meaning that this is the time to take a sample. The previously set up EoC only outputs high when the output is 0000. Therefore this signal can be sent to the sample and hold circuit as our sample signal. The resulting circuit is below.

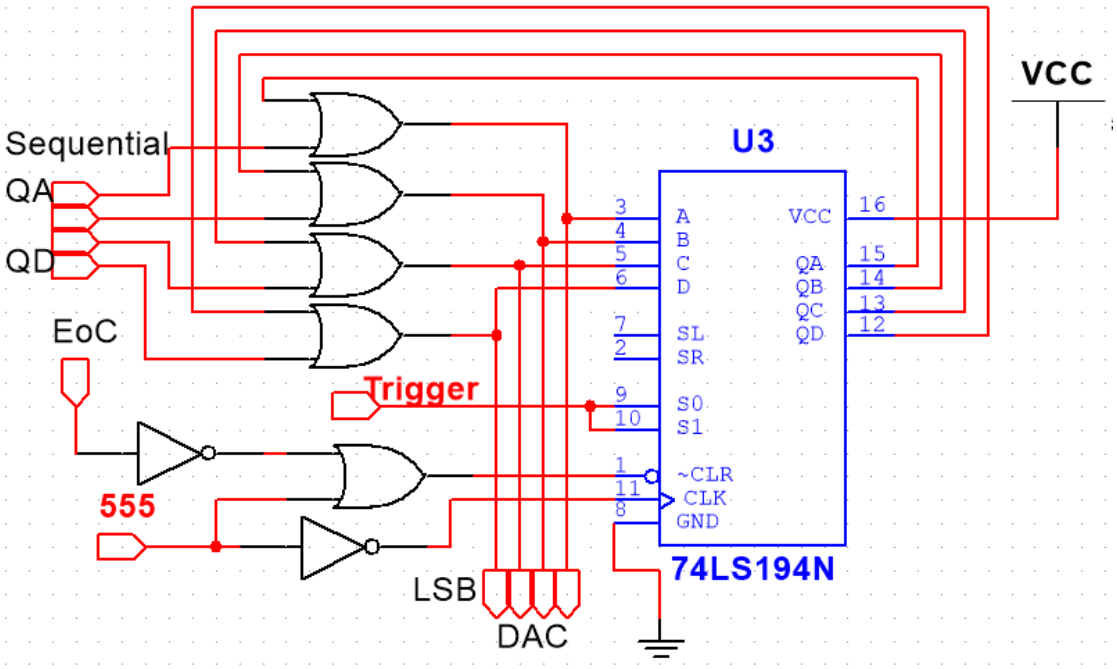



The save register is implemented using a second shift register. The outputs from the sequential circuit need to be combined with the already saved bits in the save register before being sent to the DAC. Therefore output QA of the sequential circuit is ORed

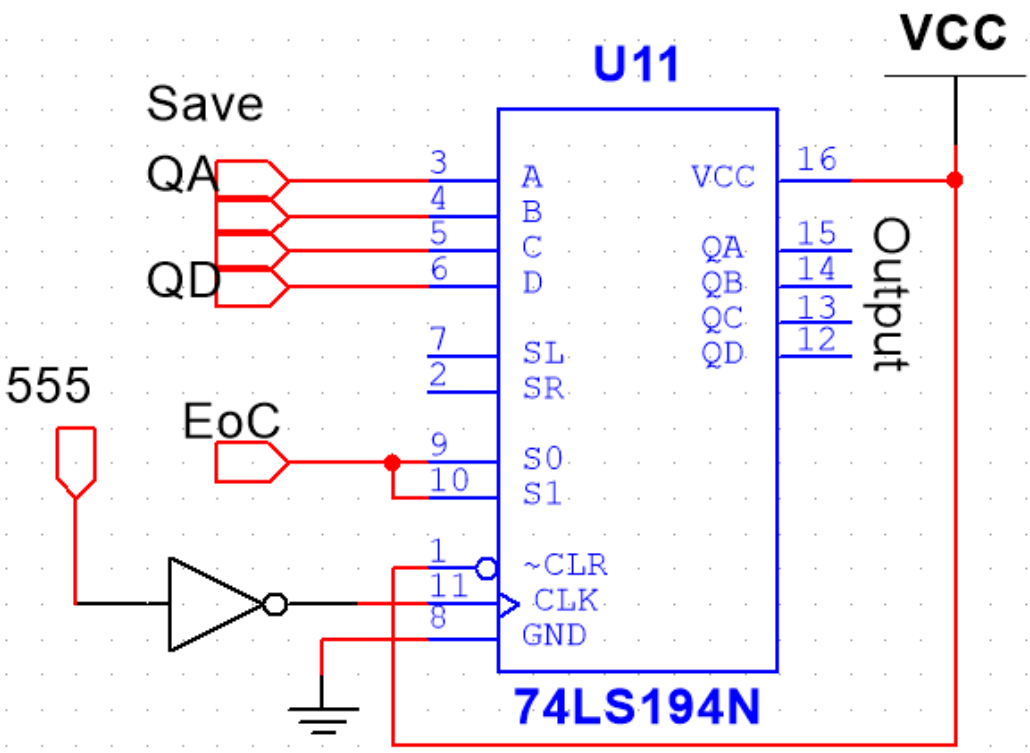
with the output QA of the save register (This is the same for QA-QD). After being ORed, the signal is then sent to the DAC for comparison and also into the input of the save register. To decide whether or not to save the input, the signal from the Schmitt trigger is used. If S0 and S1 are low then the register does nothing but if they are both high then the register is in load mode and the input will be loaded on the next clock cycle. This way, the Schmitt trigger can be attached to both S0 and S1 which will result in the input of the save register to be saved only if the signal from the Schmitt trigger is high.

At the end of the cycle the save register will not automatically reset. This must be done using the ~CLR pin. The ~CLR pin does not rely on the leading edge of the clock and therefore will immediately reset the register when pulled to ground. At the end of a cycle, the save register needs time to send the saved result to the output register. To make sure there is time, both the output register and the save register need to use a NOTed signal from the 555 timer as their clock.

In addition to this, the ~CLR pin needs to be grounded at the end of the cycle but after the save register transfers to the output register. EoC signal is always low until the end of the cycle which is the opposite of what we need. To fix this, that signal needs to be NOTed. Because the ~CLR is clock independent, there needs to be a clock delay used with this signal. With the save register using the NOTed clock we can use the normal clock ORed with the NOTed end of cycle signal. This results in the ~CLR being grounded only when both the 555 timer and the NOTed end of cycle signal are low. The resulting circuit is below.



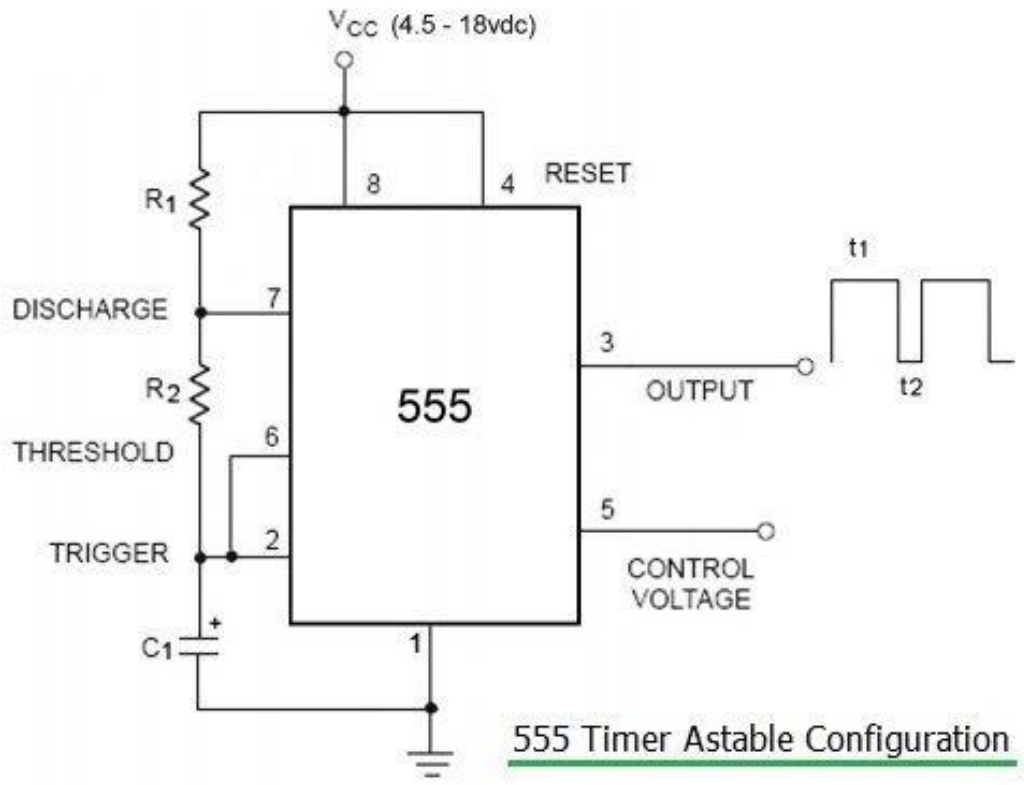

The final part of the control logic is the output register. Again this register is implemented using another shift register. All the output register does is display the results of the save register at the end of every cycle. To do this the outputs of the save register (QA-QD) can be directly imputed into the input of the output register (A-D). To make the register update only at the end of every cycle the S0 and S1 pins are connected to the EoC signal and the clock is the NOTed 555 timer (as previously stated). This register never needs to be reset and therefore the ~CLR pin can be connected to power. The resulting circuit is below.




After this project was completed I realized that instead of hooking the trigger up to S0 and S1, I could have hooked it up to CLK and made S0 and S1 always high. This would allow the save register to update only when the schmitt trigger was high and would not have to wait on the clock signal as well. Similarly the output register could have had EoC attached to CLK instead of S0 and S1 with S0 and S1 always being high. I believe this may remove the chance of some error.

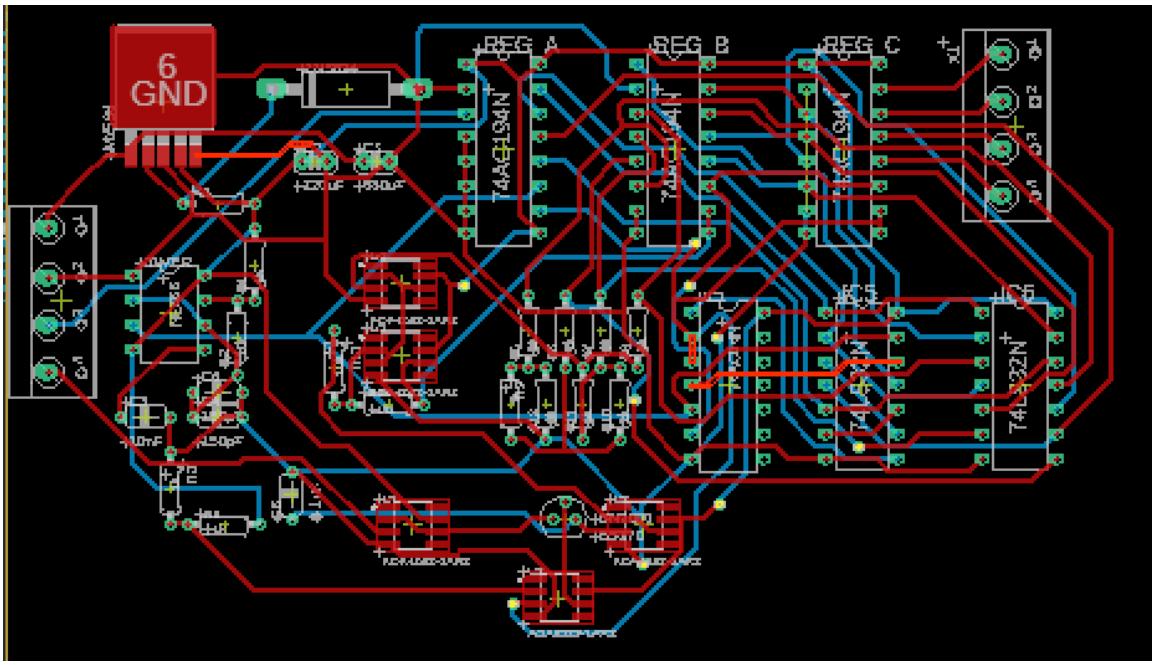
- **555 Timer** **-Ryan Fortson**

The 555 timer was used to generate the clock signal needed for the circuit, and it was configured in stable mode to produce a square wave output. The simulated values for the resistors and capacitor values were R1 was 1k, R2 was 10k and C1 was 150pF. This gave us a frequency of about 460k Hz. When it came to building the timer, we had to change the value of the capacitor in order to slow down the clock so other parts of the circuit would function properly, the actual value used was 200pF, this gave us a frequency of about 235k Hz. The TLC555 IC chip was used because of how well it handled the higher frequency unlike the NE555 which starts to give a noisy output once the clock frequency output exceeds 100k Hz.



- **PCB** **-Ryan Fortson**

Once the simulation in Multisim was done the schematic was transferred to Eagle for the PCB layout. Eagle and Multisum are not compatible, so the transfer was done by going back and forth from both software making sure to replicate the circuit perfectly from Multisim into Eagle. Once the circuit was created in Eagle, each team member looked over their respective parts to make sure that their circuit was recreated perfectly. Once the full team said that their sections looked good, then began the process of laying out where all the parts had to go on the board and connecting all the parts together. Because of how many parts and how complex the circuit was, a double sided PCB was needed. Luckily, Endeavor can do double sided PCBs with their laser PCB etcher. The PCB took about 2 hours to be laser etched and afterwards the board was cleaned and then chemical tinned made it easier to solder to.



- **Integration** **-Michael Thompson**

The first step of integration was through simulation. Each person created their own circuit file with hierarchical inputs and outputs. Because Michael understood the control circuit very well, he had the job of making the master schematic with each hierarchical block. Each block was added one at a time and tested with what was previously added. Any issues were addressed and corrected before the next block was added.

With the control circuit being the center of the overall circuit, it was added first and connected to ideal power and an ideal clock. The next part that was added was the DAC. The DAC and the control circuit were tested together with the trigger input on the control circuit set as high. Next the schmitt trigger was added with the sample and hold input of the schmitt set at a constant voltage and the output of the schmitt attached to the trigger of the control circuit. Then the sample and hold circuit was added with a 1k sine wave as its input and its output connected to the schmitt trigger. The end of cycle signal (SH pin) was connected from the control to the sample and hold.

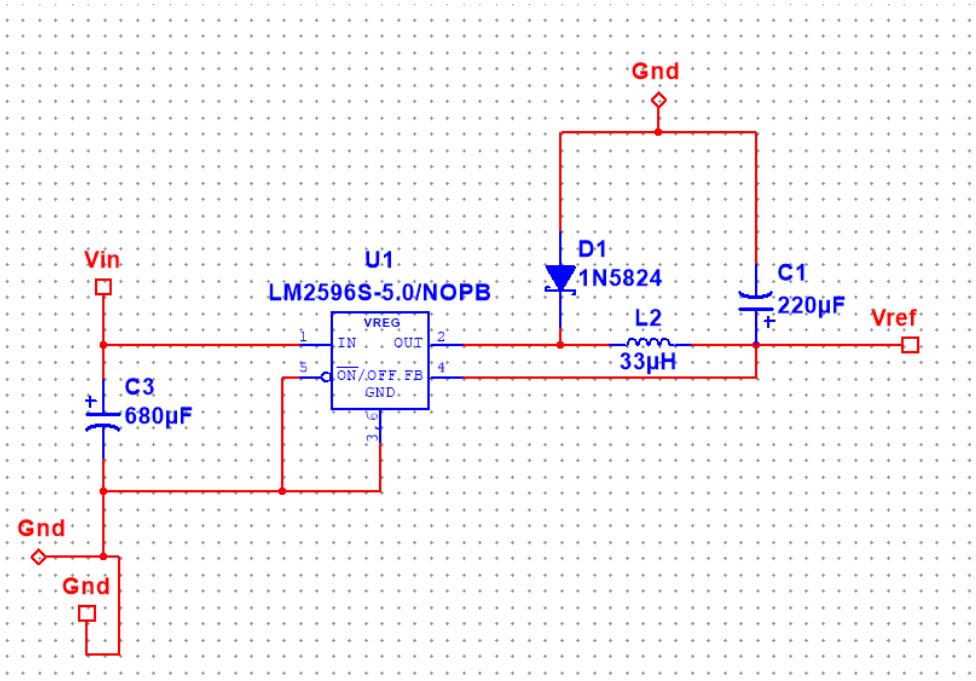
Here we found an issue. The end of cycle signal (the sample signal) is a digital signal with high being 5 and low being 0. The transistor of the sample and hold needs more than 2v of difference between the base and the gate. To achieve this, Mo added a comparator op amp with a breakpoint at 2.5 and a rail voltage of 7v. This solved the issue but we now needed a 7v input. To keep only one power we decided to add a voltage regulator that would take 7v and convert it to 5v. That way we would have access to 7v and to 5v.

The final thing to add was the 555 timer. We determined that 500k was too fast for our op amps to handle and decided to use 215k instead. To test the output, the DAC was duplicated and attached to the output of the control circuit and graphed next to the input. When the timer was added the simulator took forever to simulate anything.

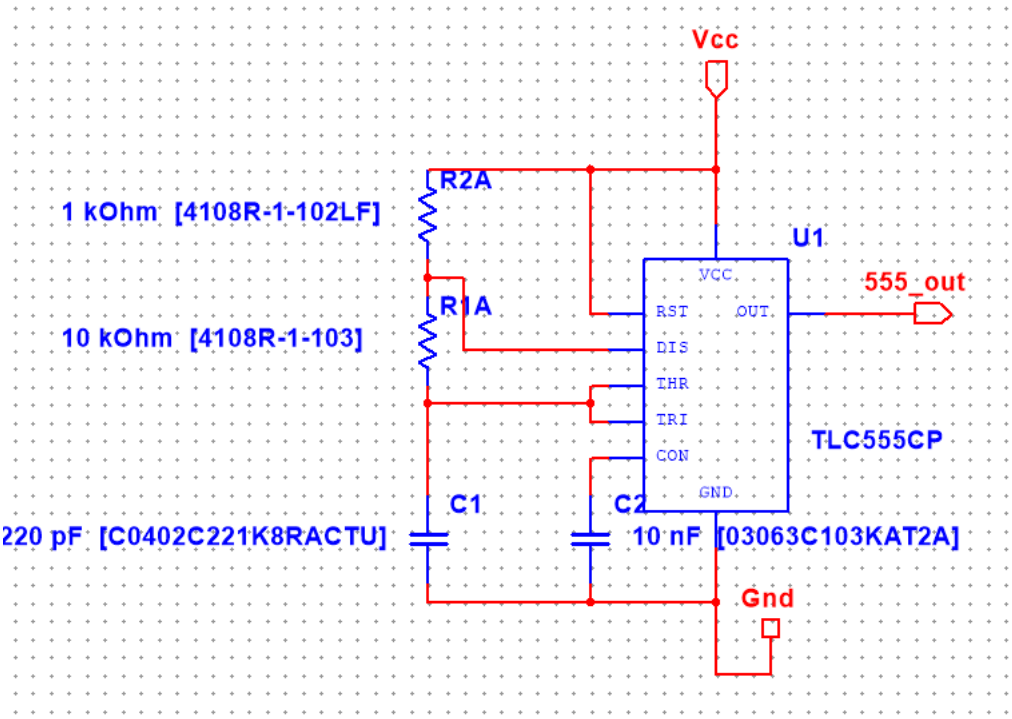
The next part of integration was attaching the hardware. Brady did almost all of the soldering and Michael did the integration testing. This testing consisted of: continuity tests, DC voltage readings, and o-scope readings. The part that took the longest was the continuity tests. As each part was soldered to the PCB it needed to be tested to see if the connections were solid. Due to the lack of through-hole plating most solders had to be done twice (on the top and the bottom of the PCB). Once everything was attached correctly we determined the output of the control circuit was not actually 5v. Instead, the voltage was 3.4v when loaded and 4.5v when not loaded or when there is high impedance. To solve this issue comparators were added for each DAC input (similar to the sample and hold). These comparators had breakpoints at 2v and a rail voltage of 5v that came from the voltage regulator. This had to be done twice because we had a second DAC used to read the output.

The hardest part about the integration was dealing with no solder mask and the lack of through-hole plating. Each solder had to be double and triple checked for a faulty connection or a short to somewhere else. If we did this again, a board with solder mask and through-hole plating would result in a much smoother integration process.

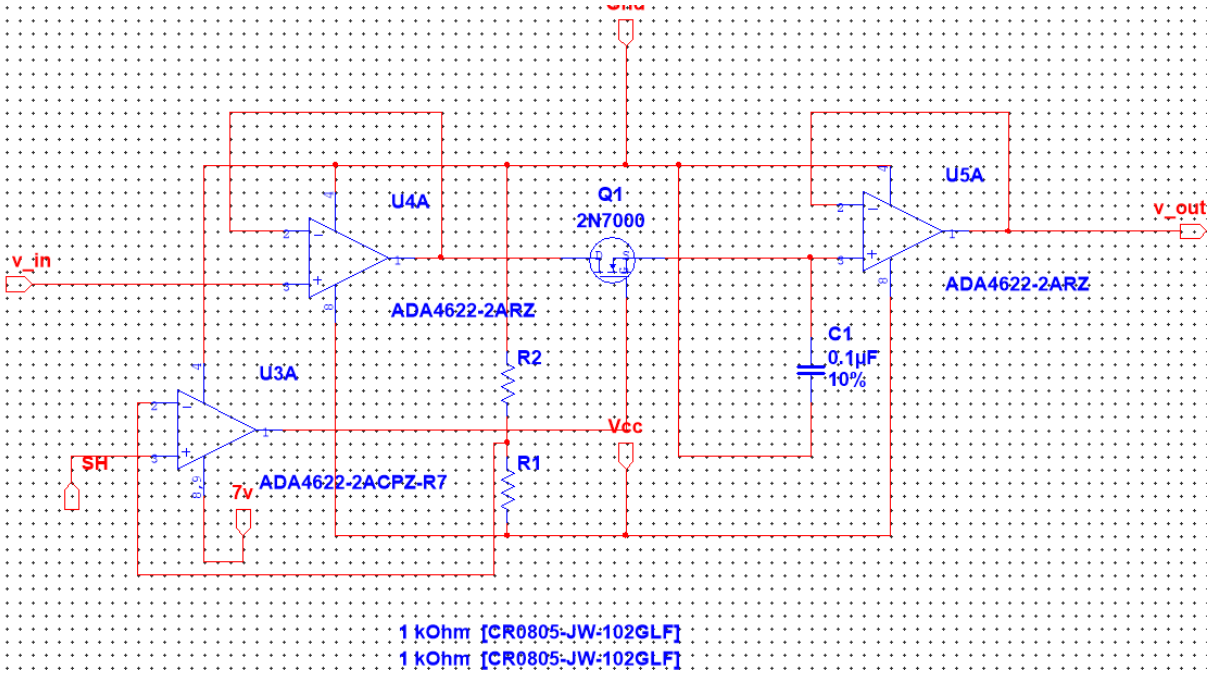
- **Schematic**
  - **Voltage Regulator**



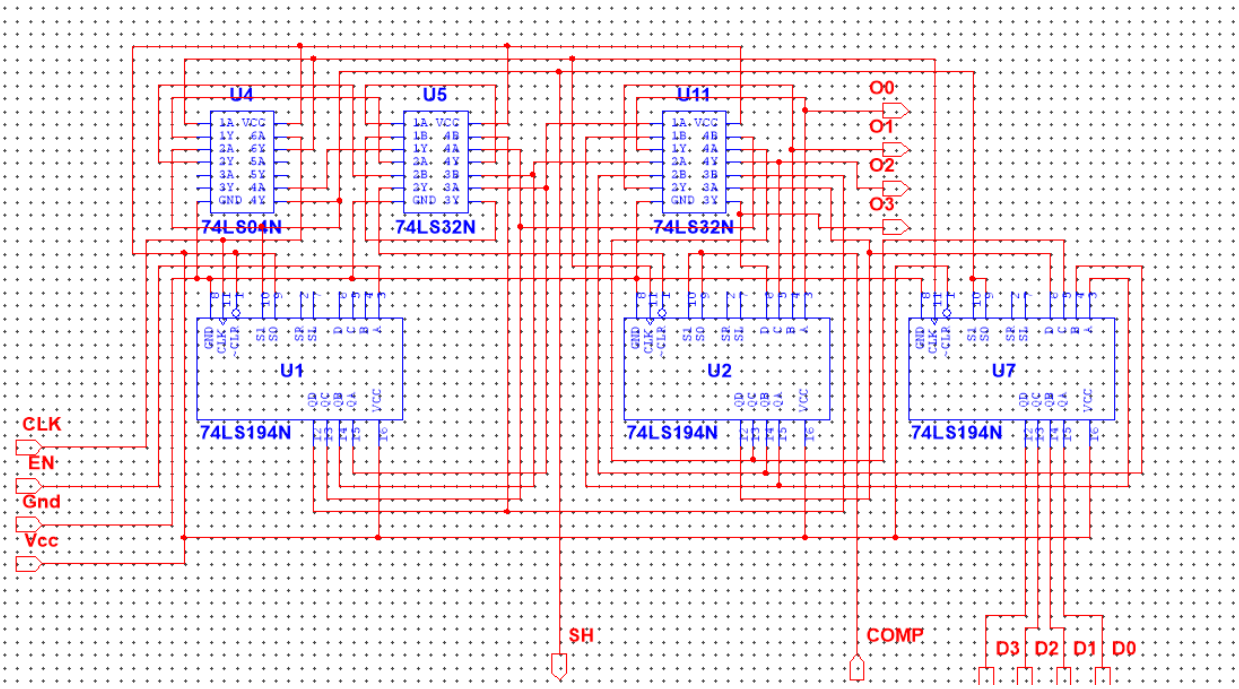

  - **Timer**



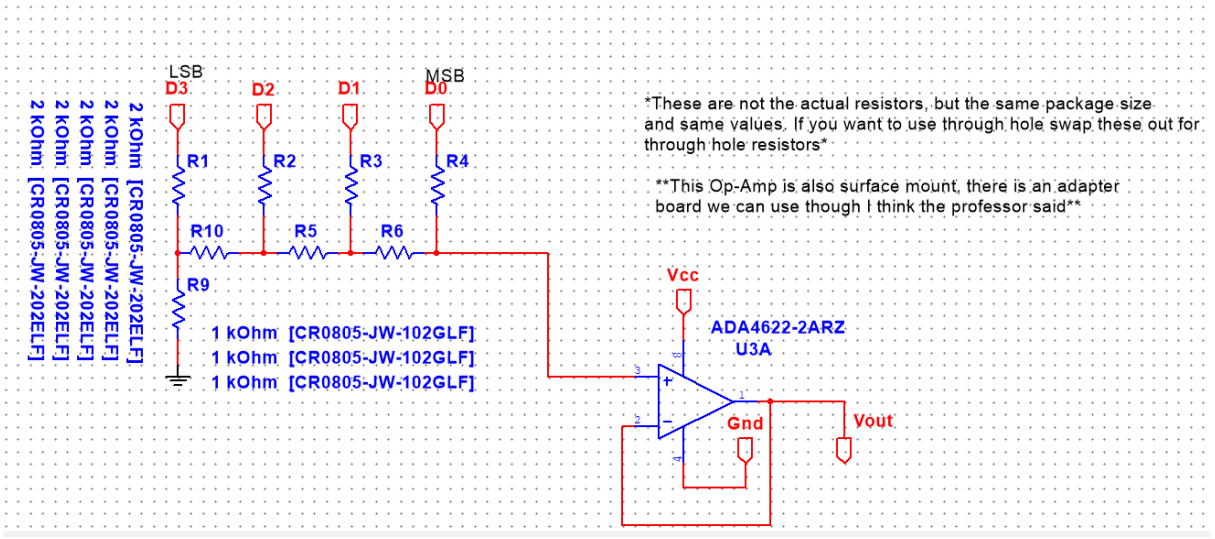
- **Sample and Hold**



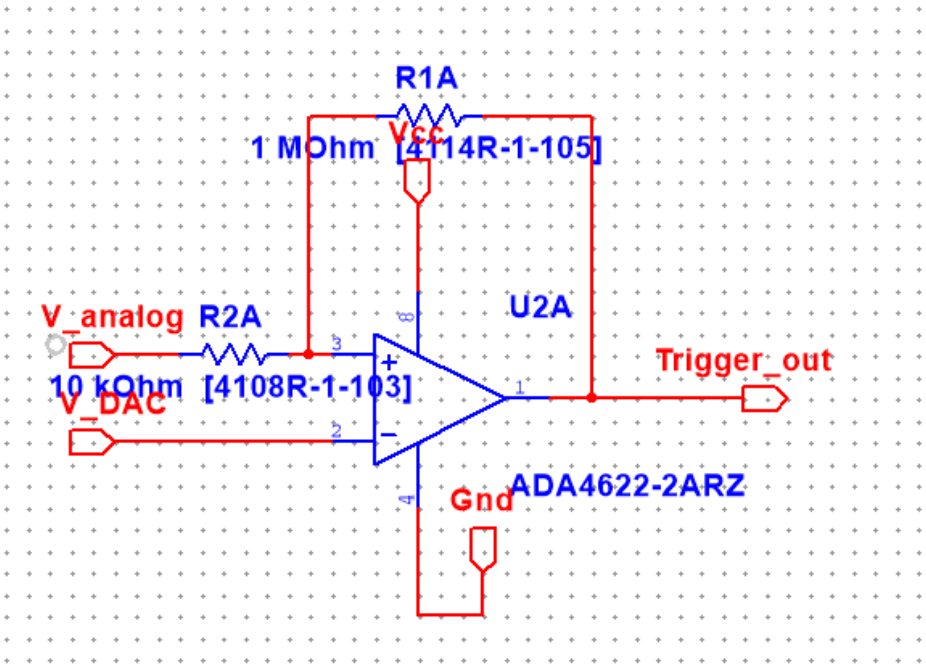

- **Control Circuit**



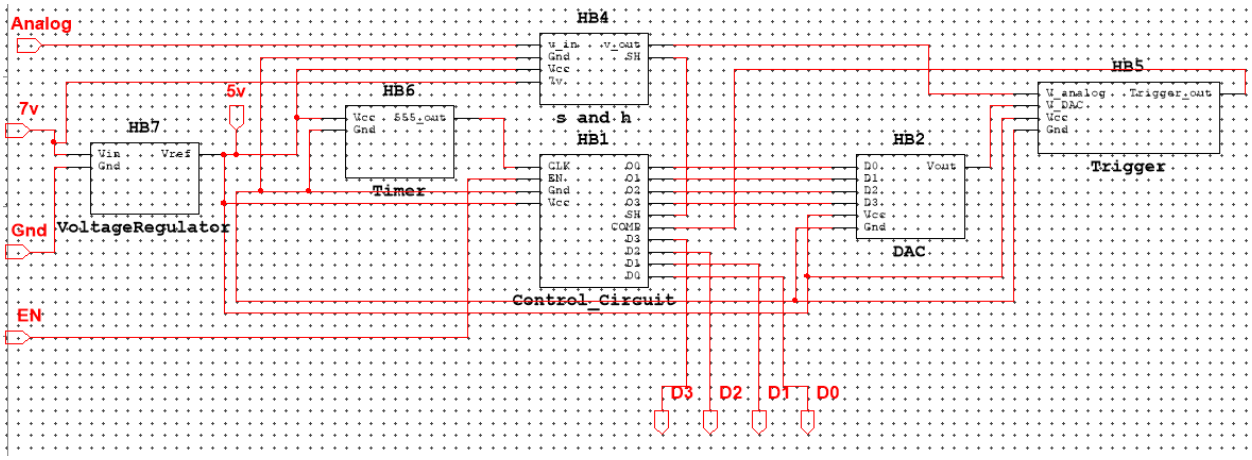
- **DAC**


LSB
D3
D2
D1
MSB
D0
R1
R2
R3
R4
R10
R5
R6
R9
2 kOhm [CR0805-JW-202ELF]
2 kOhm [CR0805-JW-202ELF]
2 kOhm [CR0805-JW-202ELF]
2 kOhm [CR0805-JW-202ELF]
2 kOhm [CR0805-JW-202ELF]
2 kOhm [CR0805-JW-202ELF]
1 kOhm [CR0805-JW-102GLF]
1 kOhm [CR0805-JW-102GLF]
1 kOhm [CR0805-JW-102GLF]
*These are not the actual resistors, but the same package size and same values. If you want to use through hole swap these out for through hole resistors*
**This Op-Amp is also surface mount, there is an adapter board we can use though I think the professor said**
Vcc
ADA4622-2ARZ
U3A
Gnd
Vout

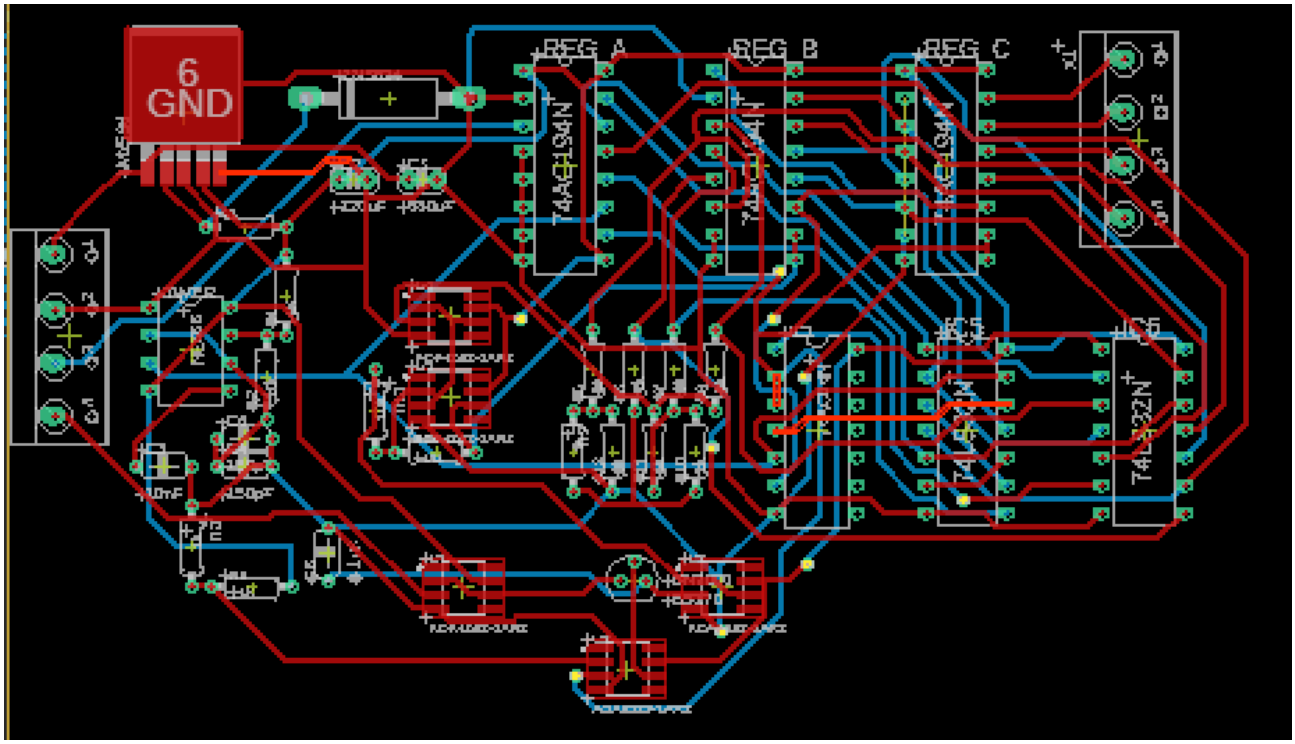
- **Trigger**

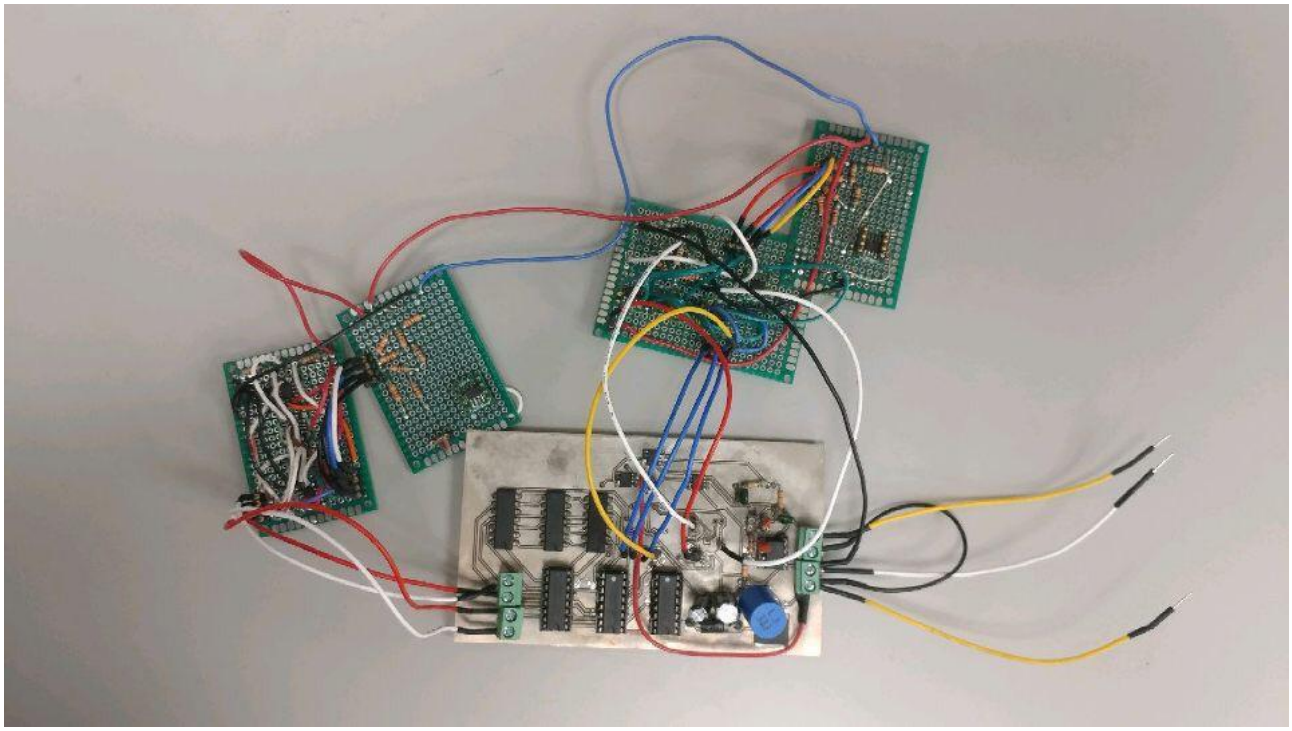

R1A
1 MOhm [4114R-1-105]
Vcc
V_analog
R2A
10 kOhm [4108R-1-103]
V_DAC
U2A
Trigger_out
Gnd
ADA4622-2ARZ

- **Overall Circuit**

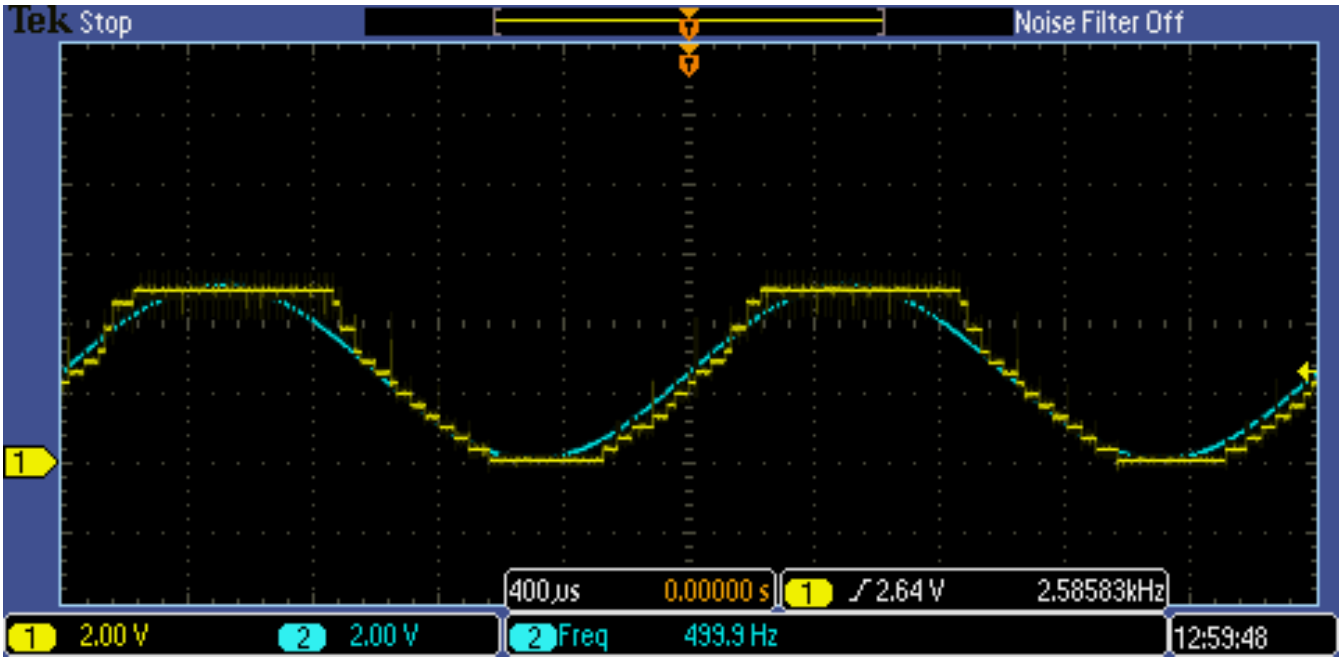



- **PCB and PCB design**

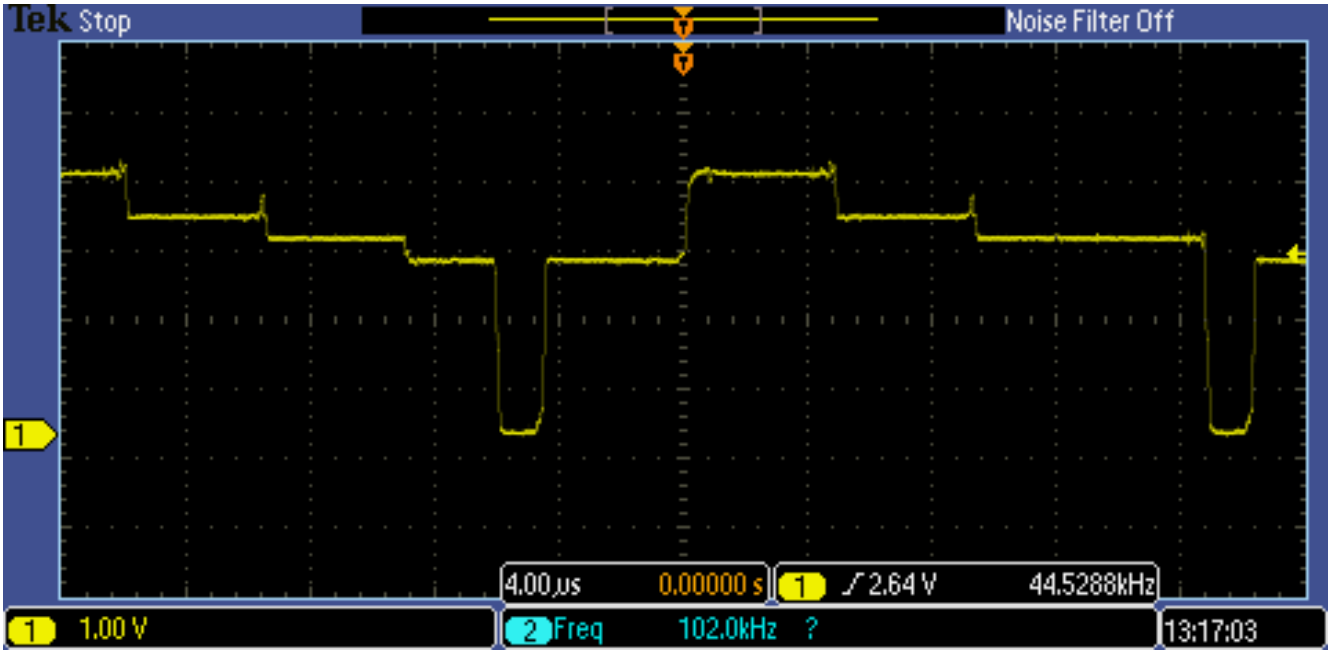




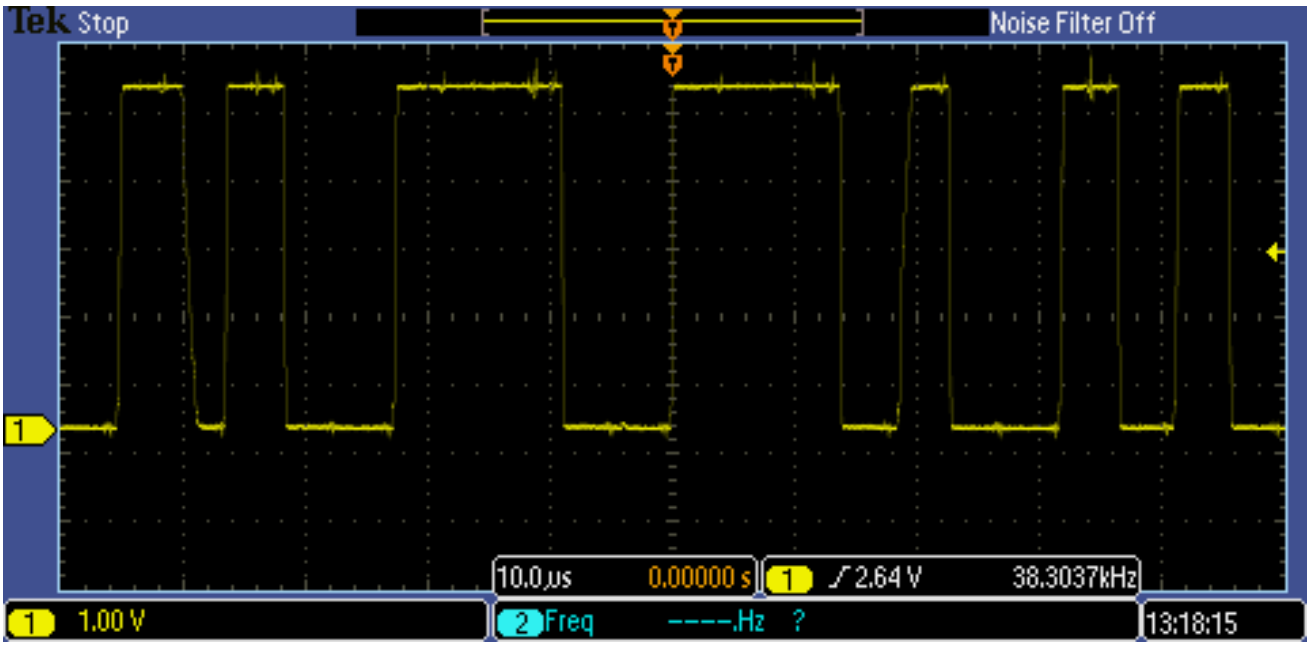
- **Simulation**
  - **ADC output**



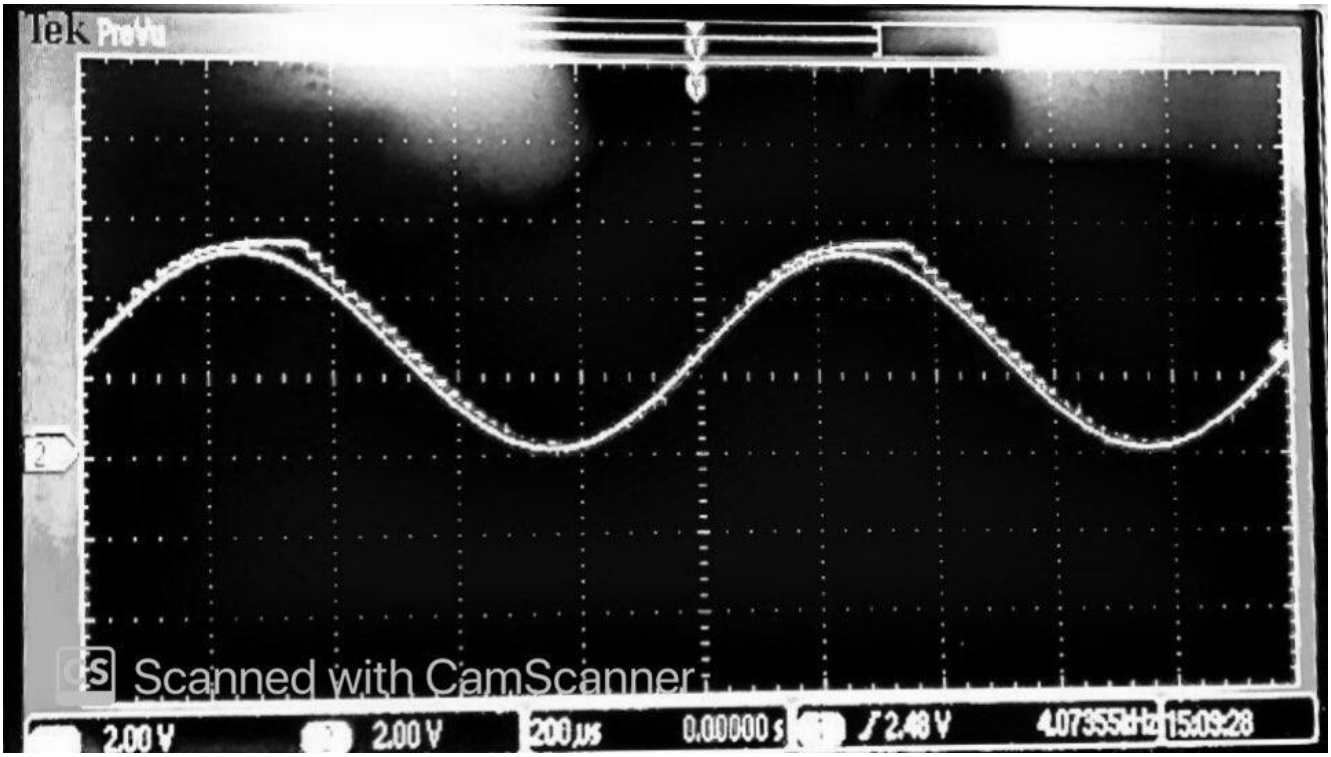
- **DAC output**




- **Trigger output**



- **Sample and hold**



- **Schedule**

| Name | Week1 | Week2 & Week 3 | Week5 & Week6 | Week7 |
|---|---|---|---|---|
| **Moatasem alsahafi** | Plan and choose a part to work on | Design and simulate sample and Hold Circuit. | Prototype sample and hold circuit in breadboard to verify simulations. | Add the sample and hold to the overall circuit and troubleshoots problems. |
| **Brady Mayfield** | Plan and choose a part to work on | Design and simulate DAC, voltage regulator. | Prototype DAC in breadboard to verify simulations. | Add the DAC to the overall circuit and troubleshoots problems. |
| **Michael Thompson** | Plan and choose a part to work on | Design and simulate control Logic and Registers. | Prototype control logic and registers in breadboard to verify simulations. | Add control logic and registers to the overall circuit and troubleshoots problems. |
| **Ryan Fortson** | Plan and choose a part to work on | Design and simulate clock and Schmitt Trigger. | Prototype clock and schmitt trigger on breadboard to verify simulations, manufacture PCB at Endeavor. | Add clock and schmitt trigger to the overall circuit and troubleshoots problems. |

- ## Parts and budget

| Part # | Quantity | Circuit | Member | Stock room/Order |
|---|---|---|---|---|
| **SN74LS194AN** | 3 | Control | Michael Thompson | Order |
| **SN74LS32N** | 3 | Control | Michael Thompson | Order |
| **74LS04N** | 1 | Control | Michael Thompson | Order |
| **LM2596S-5.0/NOPB** | 1 | V. Regulator | Brady Mayfield | Order |
| **ADA4622-1A** | 6 | S/H | moatasem alsahafi | Stock Room |
| **0.1 uF Polyester Film Capacitors** | 2 | S/H | moatasem alsahafi | Stock Room |
| **512-2N7000** | 2 | S/H | moatasem alsahafi | Stock Room |
| **1N5821** | 4 | V. Regulator | Brady Mayfield | Order |
| **ADA4622-1** | 2 | DAC | Brady Mayfield | Stock Room |
| **10k ohm carbon film 1/4W** | 5 | Trigger | Ryan Fortson | Stock Room |
| **1M ohm carbon film 1/4W** | 2 | Trigger | Ryan Fortson | Stock Room |
| **ADA4622** | 2 | Trigger | Ryan Fortson | Stock Room |
| **595-TLC555CP** | 2 | Timer | Ryan Fortson | Stock Room |
| **1k ohm carbon film 1/4W** | 2 | Timer | Ryan Fortson | Stock Room |
| **150 pF KEMET Ceramic Capacitor** | 2 | Timer | Ryan Fortson | Stock Room |
| **10 nF Ceramic Capacitors** | 2 | Timer | Ryan Fortson | Stock Room |
| **8 PIN DIP SOCKET** | 2 | Timer | Ryan Fortson | Stock Room |
| **Carbon Film Resistors 1k** | 2 | S/H | moatasem alsahafi | Stock Room |

In the below is our order from Mouser:

| Part Name | Mouser Part # | # needed | Price per Part | Total Price |
|---|---|---|---|---|
| **Shift Register** | 595-SN74LS194AN | 5 | $1.56 | $7.80 |
| **OR-2 Gate** | 595-SN74LS32N | 5 | $0.65 | $3.25 |
| **NOT Gate** | 595-SN74LS04N | 3 | $0.79 | $2.37 |
| **Voltage Regulator** | 926-LM2596S-5.0/NOPB | 1 | $5.72 | $5.72 |
| **Schottky Diode** | 512-1N5821 | 3 | $0.63 | $1.89 |
| | | | **Overall total** | $21.03 |