Lab #6 Analog-to-Digital Convertor

name: Evan Liu

partner: Zachary, Kenneth, Henrique

Table of Contents

1. Objective

The objective of this lab is to understand how an Analog-to-Digital Convertor works, and construct one with an Analog to Time Convertor. We will study and use five components:

1. Trigger Pulse Generator (TPG)

2. Clock Pulse Generator (CPG)
3. Counter
4. Analog to Time Convertor (ATC)
5. Control Logic (CL)

2. Background

The method we will use involves an Analog to Time Convertor. A trigger and stop pulse will be generated, with the time between them proportional to the input voltage. Within this time window, a counter will count the amount of pulses generated from a clock. This converts our analog voltage into a binary representation. We will use the following materials:

1. SN74HC14N Schmitt Trigger - DATASHEET
2. SN74HC00N NAND Gates - DATASHEET
3. CD74HC390E Counter - DATASHEET
4. LM311P Comparator - DATASHEET
5. LF411CN Comparator - DATASHEET

3. Components

3.1 Trigger Pulse Generator

3.1.1 Schmitt Trigger

First, we want to measure the two threshhold values on our Schmitt-trigger. We can do this by inputting a triangle wave into our invertor, and measuring the input voltage and output voltage. The circuit is setup as shown:

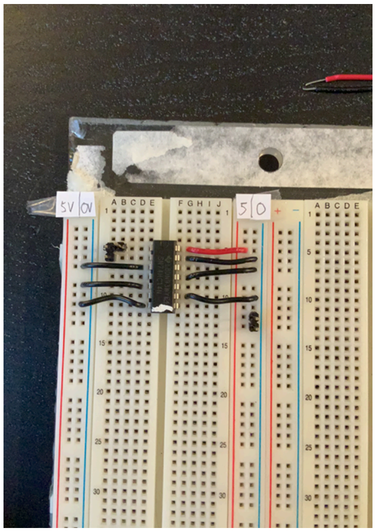


Figure 1 - Schmitt-Trigger with input triangle wave, and oscillescope set up around input and output voltage.

Using wavegen, we input a triangle wave with: 1khz period, 5V amplitude. We use an oscilloscope to measure and compare the inputs and outputs.

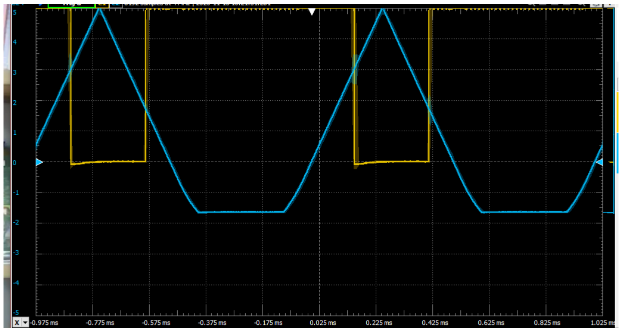


Figure 2 - Oscilloscope output from Schmitt-Trigger with blue as input, yellow as output.

Our output is as expected, the scmitt-trigger turns on a falling edge at a threshold, and turns off on rising edge at a different threshold value. We will use cursors to measure these thresholds. The uncertainties were taken to be the range of the value oscillation.

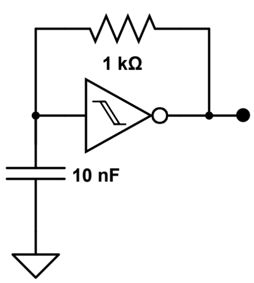
$$V_{H->L} = 2.97V \pm 0.03V$$

$$V_{L->H} = 1.72V \pm 0.03V$$

Schmitt-triggers provide good noise immunity because of their threshold value hysteresis. Hysteresis means that something will depend on its history. In this case, the trigger value depends on if it is high or low. Once the trigger turns off at a threshold, it will turn on at a different threshold value. This allows any mini noise oscillations to not trigger a state change by mistake.

3.1.2 Square Wave Oscillator

Now we will construct a square wave oscillator from the circuit diagram in Fig. 3.



a

Figure 3 - Circuit diagram

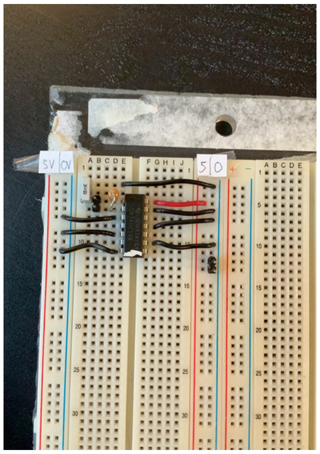


Figure 4 - Constructed circuit

We will set up an oscilloscope on the voltage at the input, and the voltage at the output.

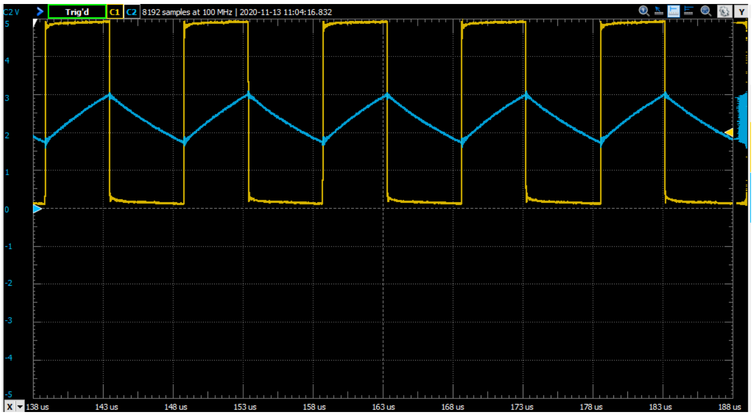
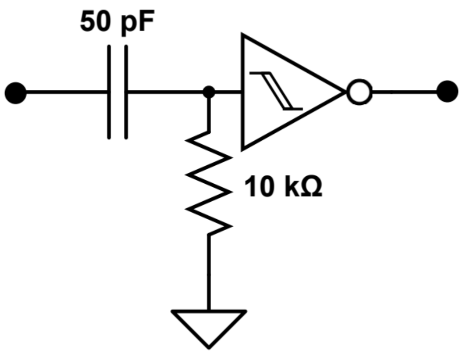


Figure 5 - Oscilloscope with blue as input, and yellow as output.

We can think of the input as starting as LOW, and the therefore the output is HI. the HI voltage is linked back to the input with a capacitor that is linked to ground. The capacitor at the input will accumulate voltage, and once it reaches a threshold where the Schmitt-Trigger interprets HI, it outputs LO. The capacitor will then discharge through the resistor until it reahes a threshold value to turn the trigger on again. This cycle creates a square wave oscillator.

3.1.3 Trigger Pulse Generator

To make this oscillator generate short pulses every few seconds, we hook up a short pulse rectifier to the output of the oscillator.



b

Figure 6 - Circuit diagram of short pulse rectifier. The node on the left connects to the output of the Square Wave Generator.

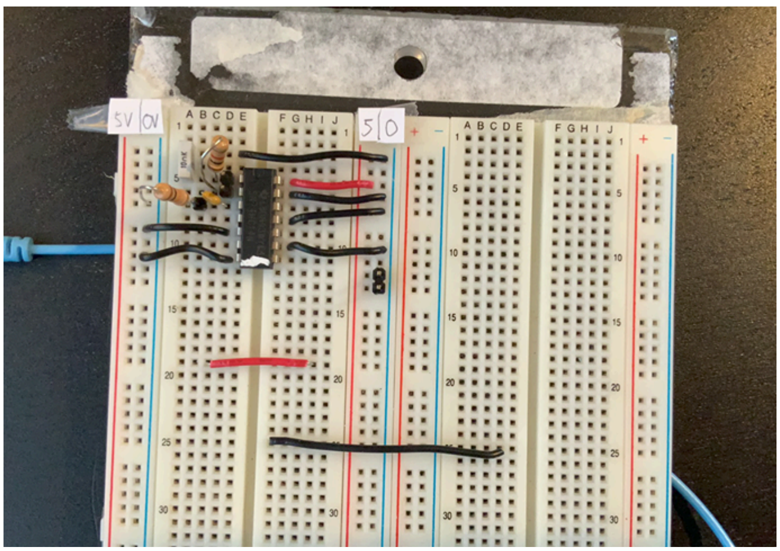


Figure 7 - Constructed Circuit

To verify this circuit performs our intended function, we hook up an oscilloscope around the input of the Schmitt-trigger on the square wave oscillator, and the input of the short pulse rectifier. Below is a screenshot of the oscilloscope reading.

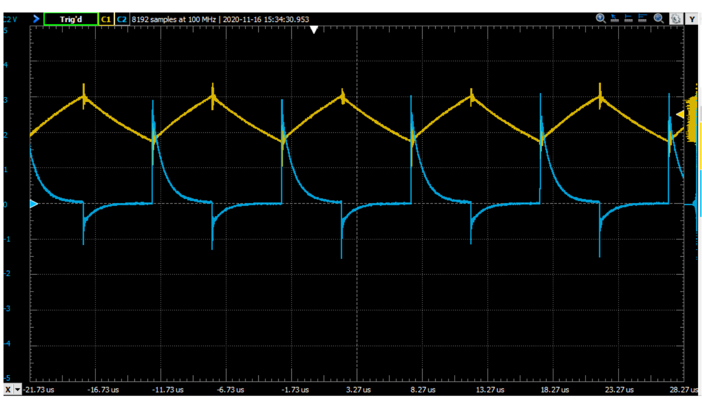


Figure 8 - Scope Reading

As shown, the voltage at the input to the second schmitt-trigger spikes when the first trigger ouput changes state. This spike can be used to generate a pulse in the second trigger. The rising edge turns the schmitt output off, and shortly after the falling edge turns the ouput back on. We confirm this by hooking an oscilloscope channel to the output of the TPG.

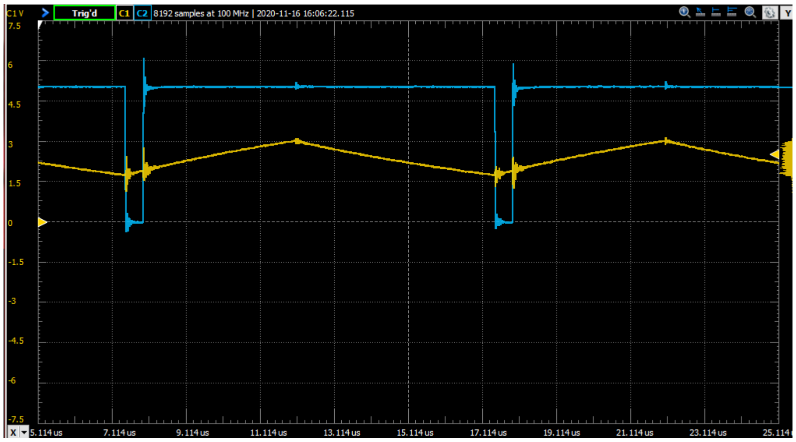


Figure 9 - Oscilloscope around schmitt output

As expected, when the first trigger ouptut switches to high, there is a short pulse to ground generated.

To decrease the oscillation frequency, we change the 10nF capacitor to a 1 μF in parallel with a 220 μF capacitor to form a $221 \mu F$ capacitor.

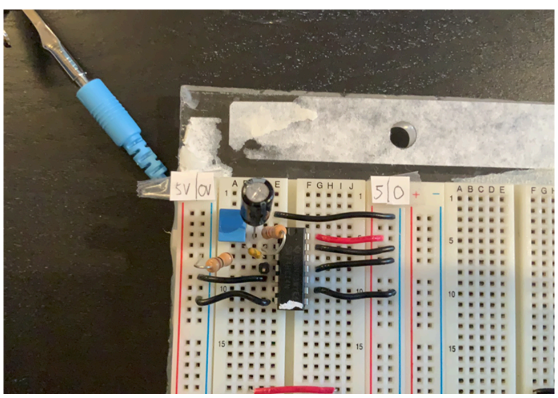


Figure 10 - Replaced Capacitors

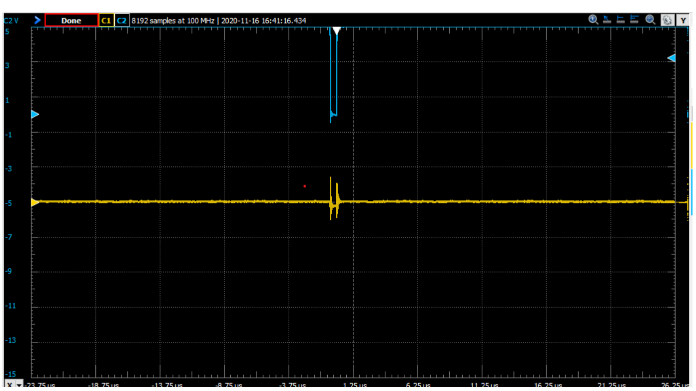


Figure 11 - Oscilloscope Single Shot

We can see from a single shot measurement form the scope that it generates a trigger pulse relatively infrequently. This is because by increasing capacitance value, we increase the time constant of the RC section of our circuit. This increases the time for charge and discharge, slowing the rate of pulses generated.

3.2 Clock Pulse Generator

We now will construct a CPG as shown in Fig. 12.

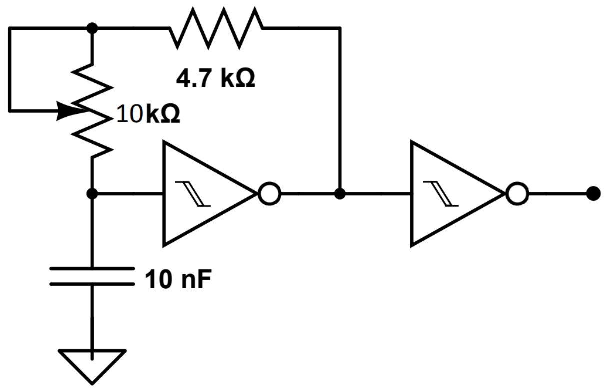


Figure 12 - Clock Pulse Generator Diagram

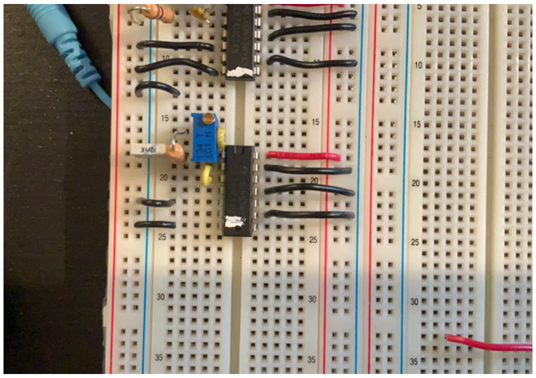


Figure 13 - Built Clock Pulse Generator

Hooking up an oscilloscope to the output of the CPG, we can verify that it produces a square wave output.

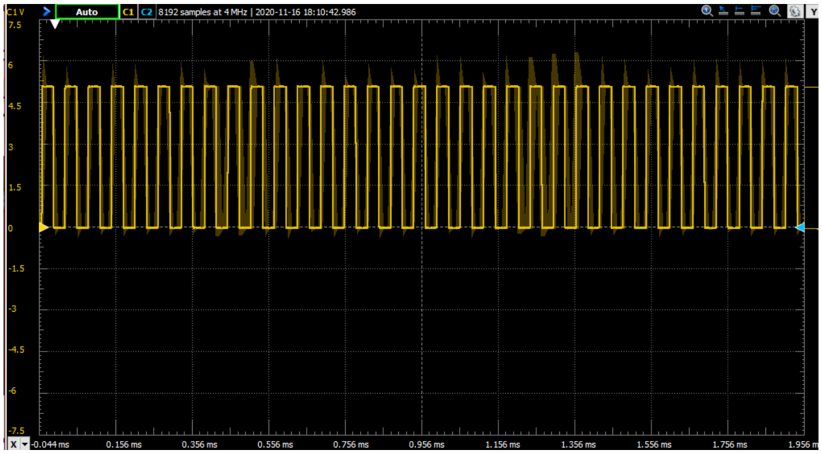


Figure 14 - Square wave output

By adjusting the potentiometer, we verify that the frequency can be adjusted from $7.5kHz \pm 0.3kHz$ to $23.0kHz \pm 0.5kHz$. These values were measurements taken by the oscilloscope, and the uncertainty was taken to be the maximum variation of the measurements.

3.3 Counter

Now we construct a divide-by-100 counter as shown in the following schematic:

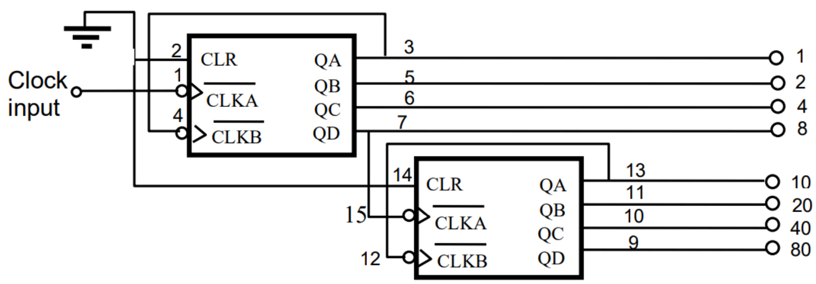


Figure 15 - Hundred Counter

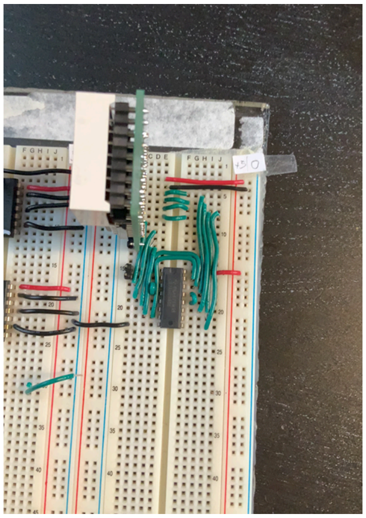


Figure 16 - Assembled counter.

It's operation can be verified in the following video: https://www.youtube.com/watch?v=edlV9Yeyc_A&feature=youtu.be

3.4 Analog to Time Convertor

This component consists of two parts: a voltage ramp generator, and a voltage comparator.

The generator will generate a voltage that increases linearly with time on a given start pulse (given from the TPG). The voltage comparator will generate a stop pulse once the voltage ramp has exceeded our input voltage we want to convert. This will also reset the ramp generator, and the control circuit flip-flop, stopping the readings from the decade counter.

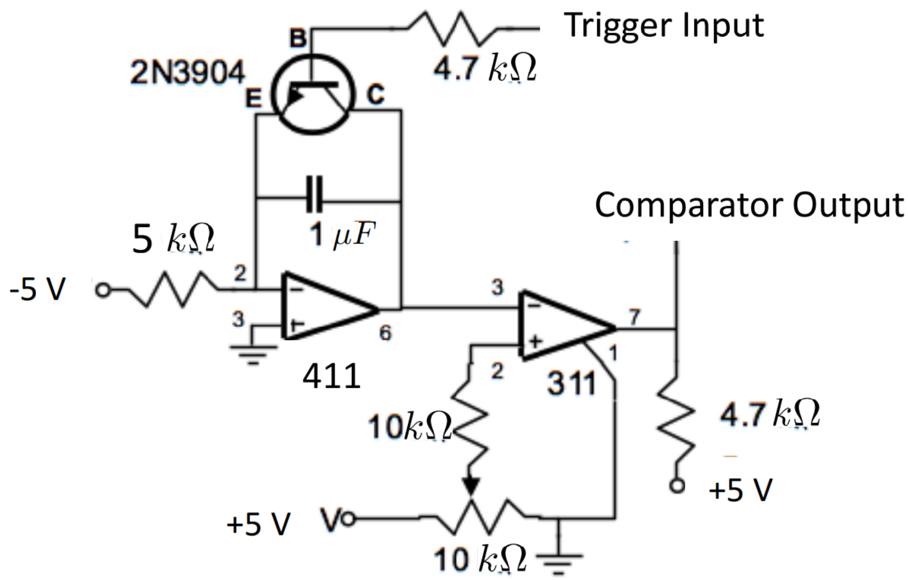


Figure 17 - ATC Circuit Diagram

Starting with the first Op-Amp circuit, assuming the capacitor starts with no charge, and the transistor is off, we can verify it produces a ramp as follows:

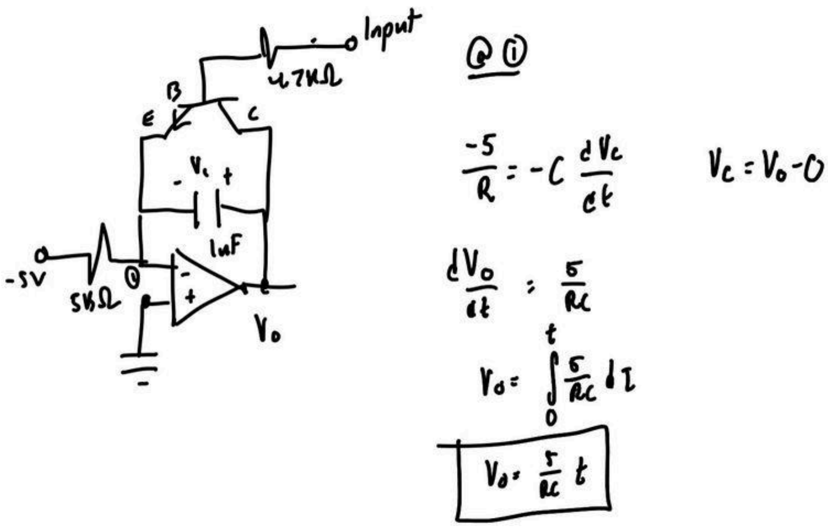


Figure 18 - Derivation of voltage ramp

We can see that V_{out} has a linear relationship with time. When the trigger input activates the transistor, the capcitor is short circuited and the voltage is reset to 0V.

For the comparator OpAmp, the output will start HI, and once the voltage ramp exceeds the voltage set by the pot, will quickly switch to LO.

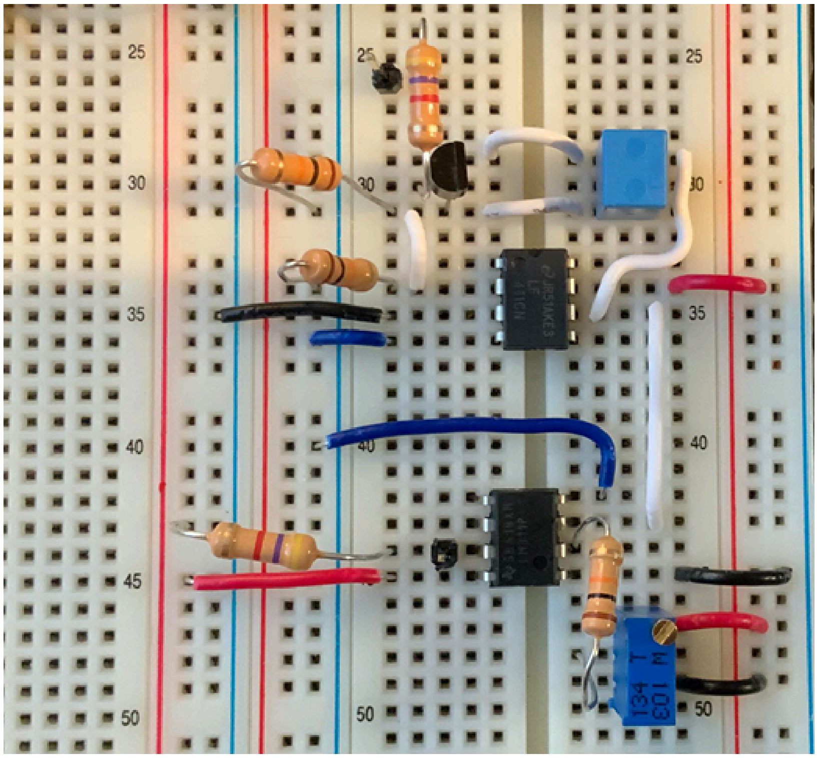


Figure 19 - Constructed ATC

To test this function, we apply a square wave signal to the input of the transistor switch using a pulse with amplitude 5V, and frequency 100hz. We hook up the yellow scope around the input comparator voltage, and the blue scope around the voltage ramp output.

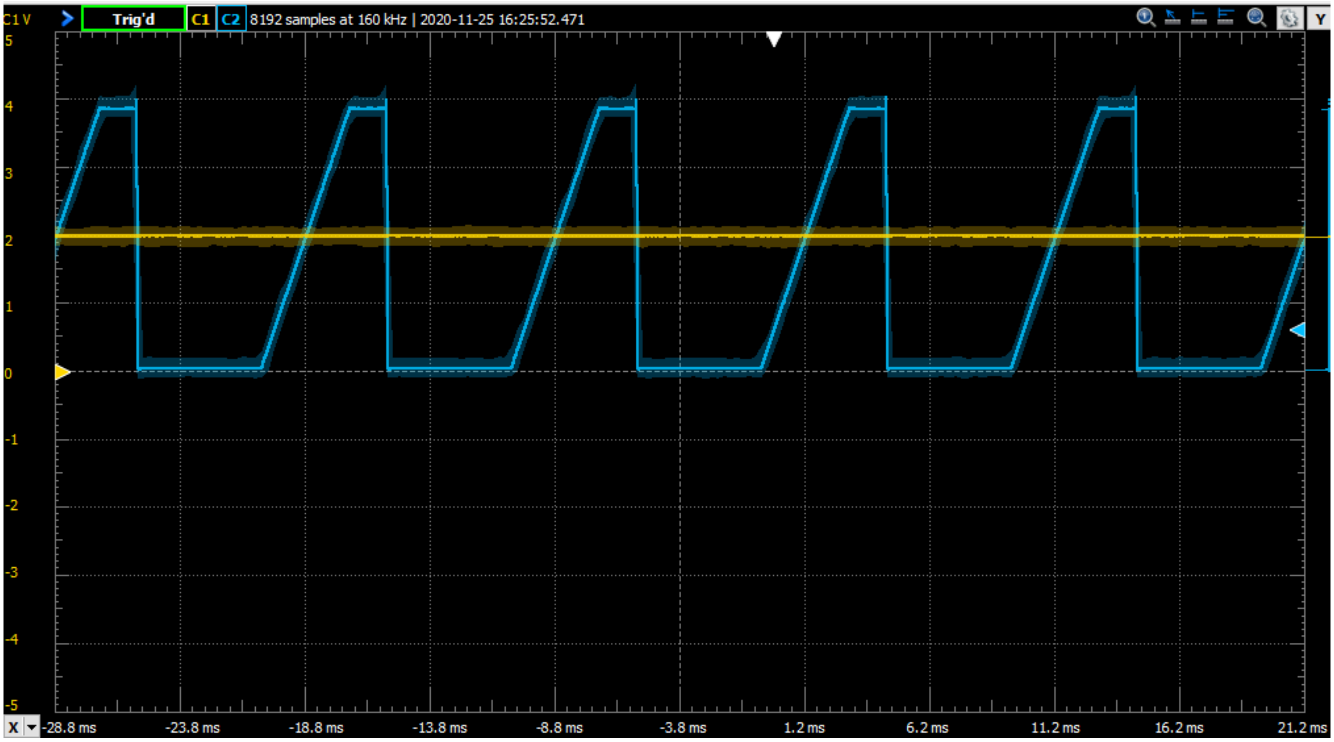


Figure 20 - Scope reading

Here we can see that the input comparator voltage is set at 2V. Now we move the yellow scope to the comparator output.

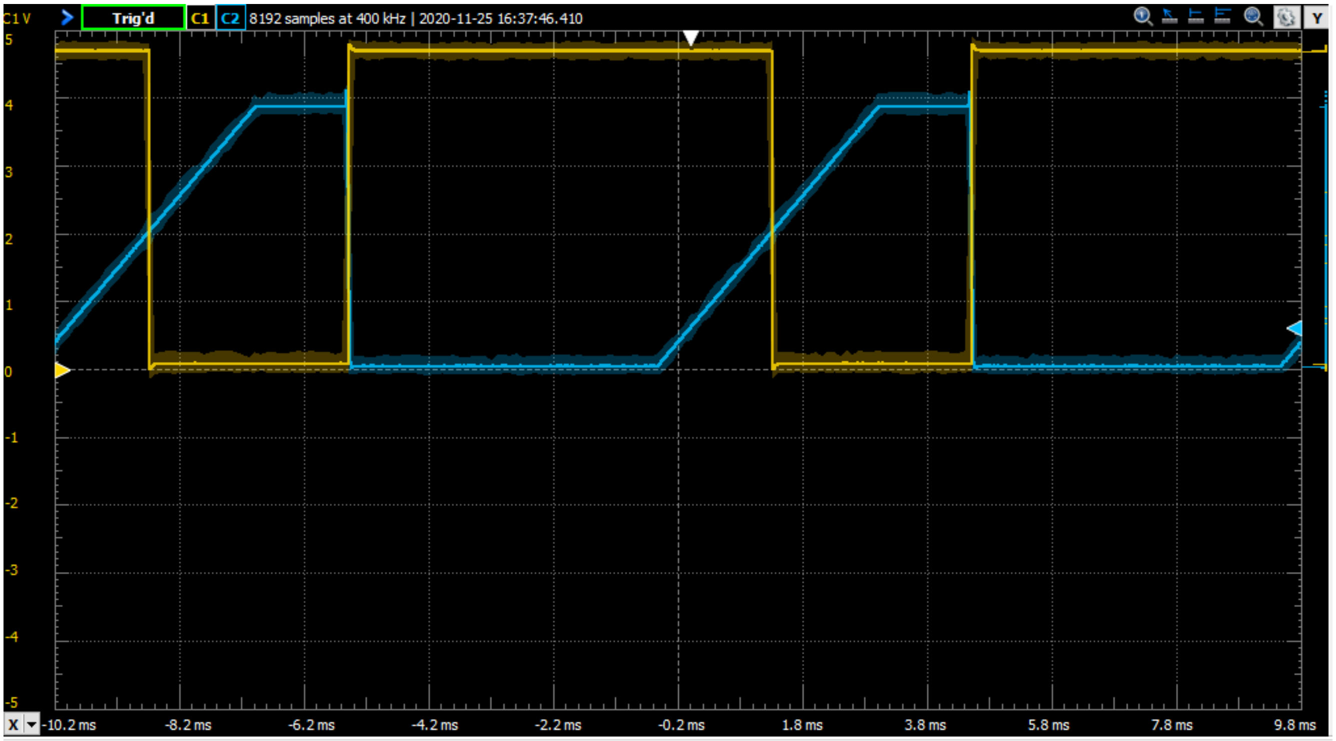


Figure 21 - Yellow scope is around comparator output, blue around output voltage ramp

We can see that once the ramp exceeds 2V, the comparator output becomes LO. When the ramp resets to 0, the output becomes HI again. This agrees with our desired behaviour.

3.5 Control Logic

Now we want to construct our Control Logic as shown in Fig. 22. This will coordinate our component inputs/ outputs together.

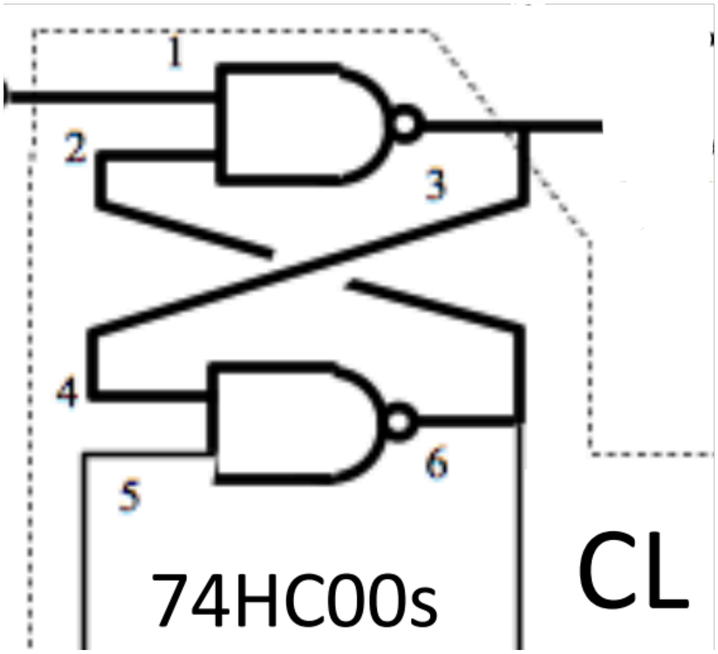
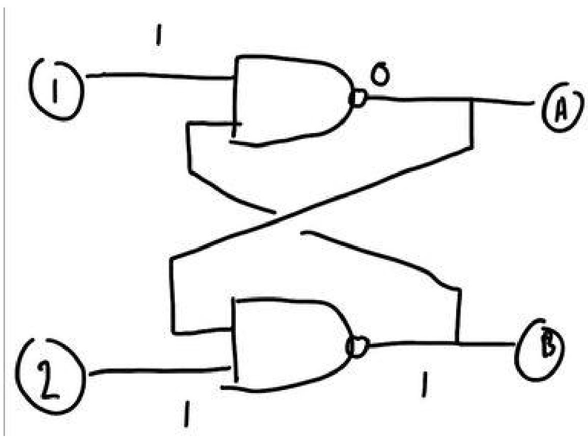


FIgure 22 - Control logic circuit diagram



Control Logic

1	2	A	B
0	0	1	1
1	0	0	1
0	1	1	0
1	1	prev	prev

Figure 23 - Truth table

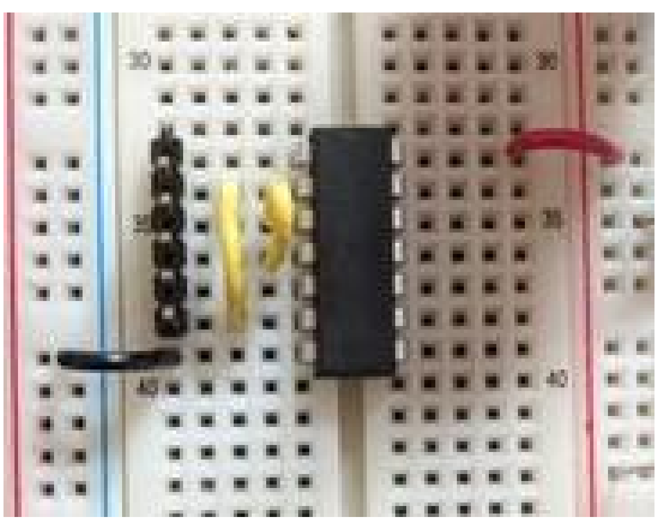


Figure 24 - Constructed Circuit

To test our truth table, we input the four different different combinations and measure their output.

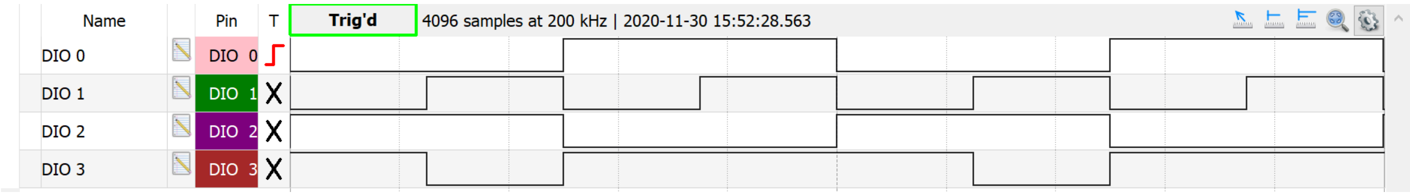


Figure 25 - Logic IO. DIO0 is the input to DIO3, and DIO1 is the input to DIO2.

We can verify that our CL matches the truth table.

4. Integration of Components into an ADC

We will now assemble the full ADC. We also need to add an additional NAND gate, and two more invertors.

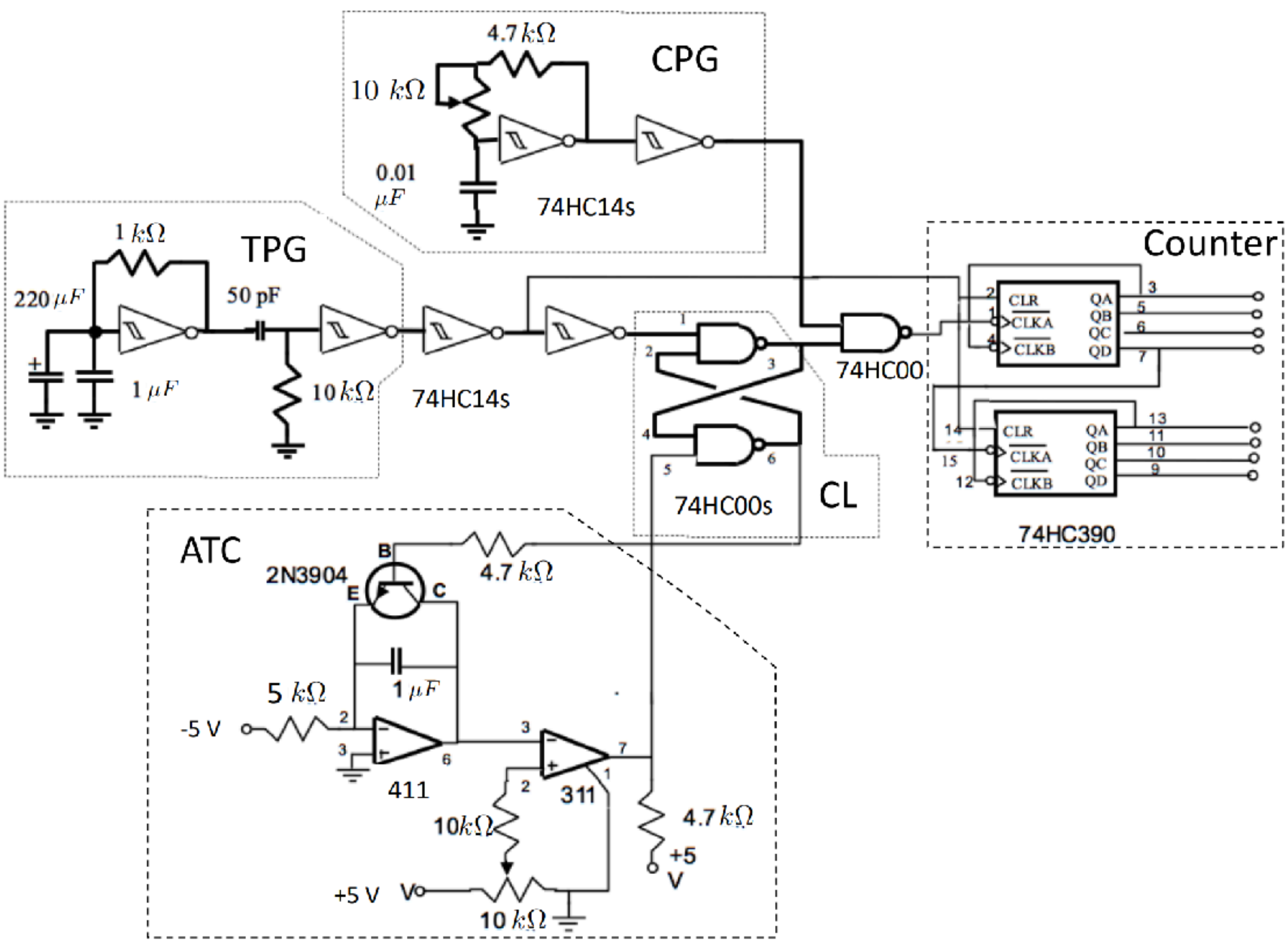


Figure 25 - Full ADC constructed

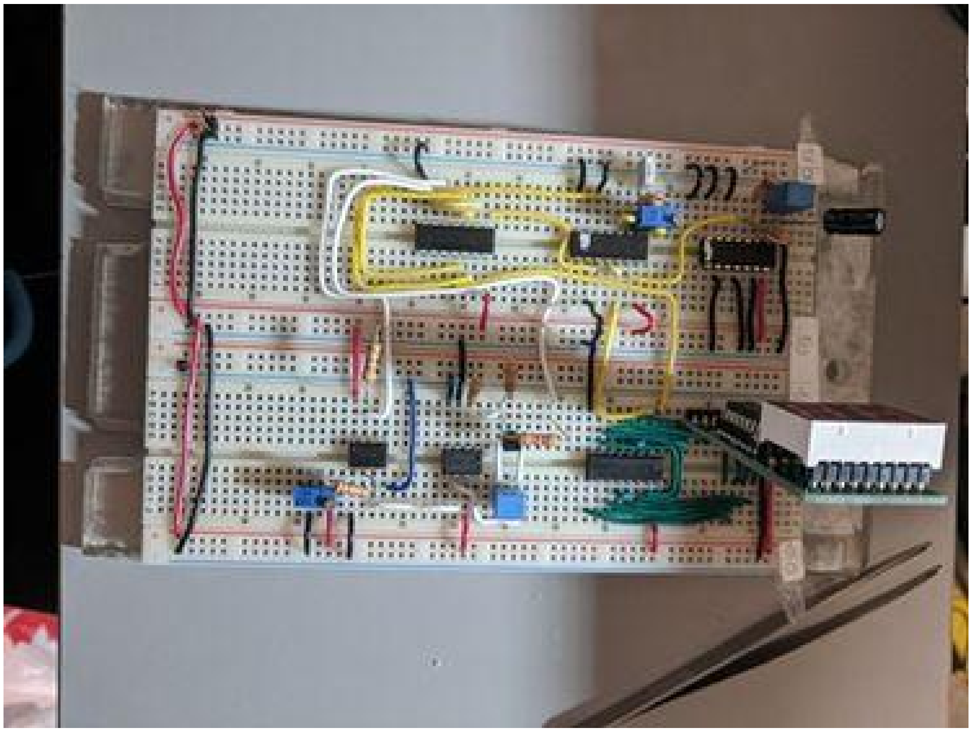


Figure 26 - Constructed ADC

4.1 Mode of Operation

When I first plugged in the power, the display did not turn on and so I had to troubleshoot.

4.1.1 Troubleshooting

Since I have already verified each component individually, I decided I would first check the connections between components. I noticed that the wire connecting my TPG to the other Schmitt Trigger had been plugged into the input of the trigger, and not the output. After correcting the connection, the display turns on.

4.1.2 Operation

When we plug in power, the display turns on, and its value is changeable by the potentiometer in the ATC. Hooking up an oscilloscope to the output of the voltage potentiometer, and comparing it to the display we see the value is a bit off. We can calibrate this by changing the voltage pot to 2.5V, and adjusting the resistor pot in the CPG until the values match.

Now we want to produce a timing diagram, so we hook up a digital input to the output of all our components, and a scope to the integrator ramp. We set the voltage to 1V for this measurement.

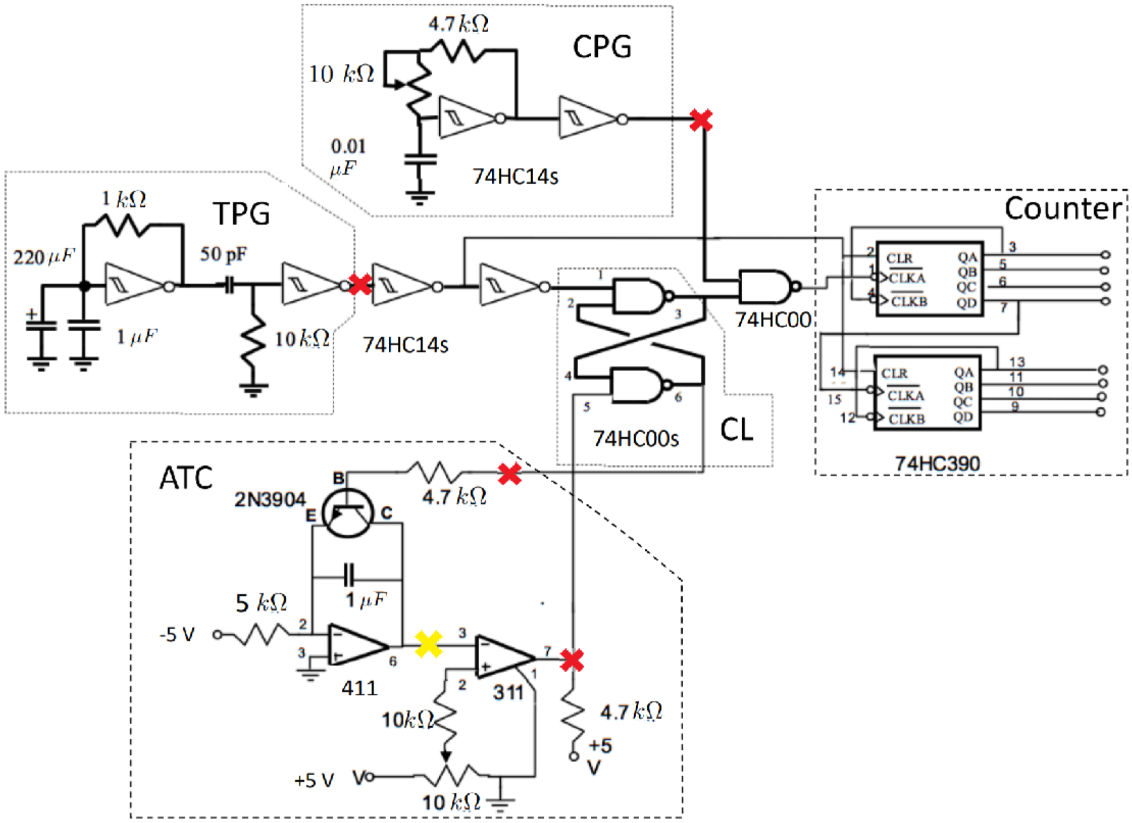


Figure 27 - Red cross signifies where we read digital input, yellow cross signifies where we read our scope.

4.1.2 Importing Waveform Data

Script for importing data from the following text file:

```
filename: C:\Evan\Eng\MatLab\Enph 259\Lab6\logic.csv
```

Auto-generated by MATLAB on 02-Dec-2020 15:09:59

Setup the Import Options and import the data

```
opts = delimitedTextImportOptions("NumVariables", 5);

% Specify range and delimiter
opts.DataLines = [1300, 6000];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["Times", "CPG", "TPG", "Comparator", "CL"];
opts.VariableTypes = ["double", "double", "double", "double", "double"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Import the data
```

```
tbl = readtable("C:\Evan\Eng\MatLab\Enph 259\Lab6\logic.csv", opts);
```

Convert to output type

```
Times = tbl.Times;
CPG = tbl.CPG;
TPG = tbl.TPG;
Comparator = tbl.Comparator;
CL = tbl.CL;
TPG = TPG + 1.02;
Comparator = Comparator + 2.02;
CL = CL + 3.06;
```

Clear temporary variables

```
clear opts tbl
```

Script for importing data from the following text file:

```
filename: C:\Evan\Eng\MatLab\Enph 259\Lab6\scope.csv
```

Auto-generated by MATLAB on 02-Dec-2020 15:13:59

Setup the Import Options and import the data

```
opts = delimitedTextImportOptions("NumVariables", 3);

% Specify range and delimiter
opts.DataLines = [3500, 8000];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["ScopeTimes", "Channel1V", "Var3"];
opts.SelectedVariableNames = ["ScopeTimes", "Channel1V"];
opts.VariableTypes = ["double", "double", "string"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, "Var3", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Var3", "EmptyFieldRule", "auto");

% Import the data
tbl = readtable("C:\Evan\Eng\MatLab\Enph 259\Lab6\scope.csv", opts);
```

Convert to output type

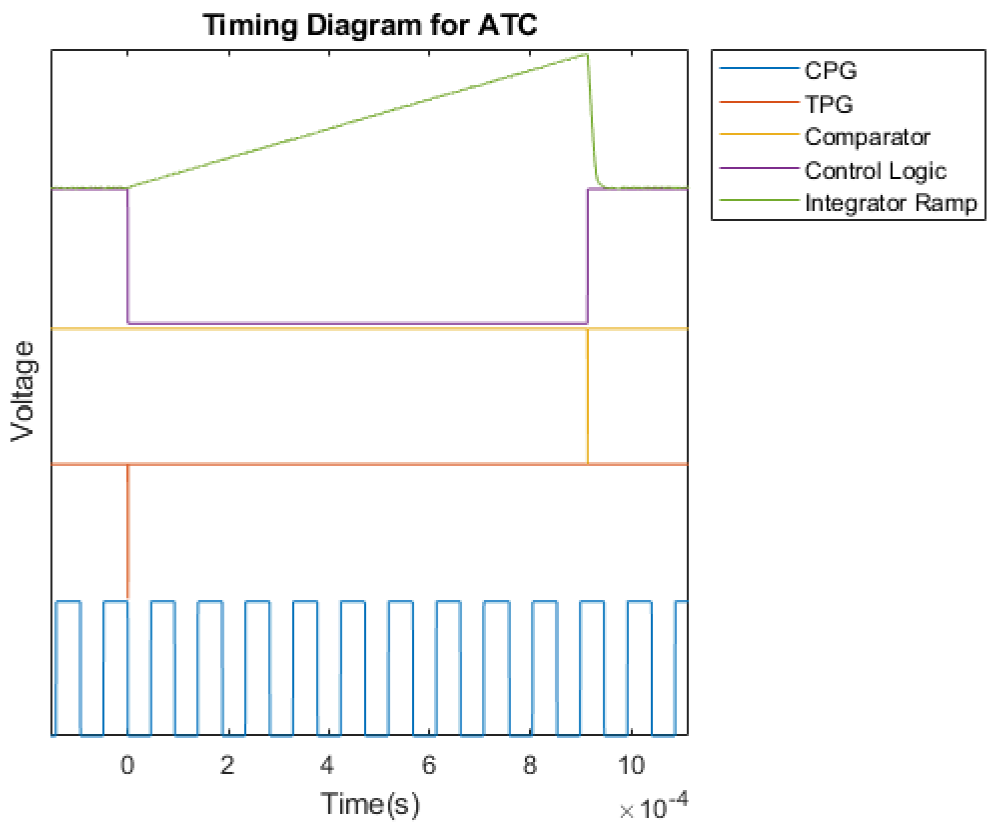
```
ScopeTimes = tbl.ScopeTimes;
Channel1V = tbl.Channel1V;
Channel1V = Channel1V + 4.02;
```

Clear temporary variables

```
clear opts tbl
```

4.1.3 Graph Scope and Logic Data

```
plot(Times, CPG, Times, TPG, Times, Comparator, Times, CL, ScopeTimes,Channel1V)
ylim([0, 5.1])
xlim([-0.00005, 0.001])
legend("CPG", "TPG", "Comparator", "Control Logic", "Integrator Ramp", 'Location', 'bestoutside
set(gca, 'ytick',[]);
title('Timing Diagram for ATC')
xlabel('Time(s)')
ylabel('Voltage')
```



Our timing diagram looks as expected.

The TPG gives a very short pulse, and starts the data colection, while the Comparator gives a very short pulse to end the data collection.

The control logic going to GND represents the timing window in which we want to count pulses, and the Integrator Ramp represents the climbing voltage.

In this case, the time it takes our voltage ramp to reach 1V is around 0.9ms.

4.2 Conversion Table

We want to compare the voltages we set, and what the display shows. We can measure the voltages using the DMM on the Analog Discovery. We construct the table.

Voltage Set (V)	DMM Display Number
0.70 ± 0.01	06
1.40 ± 0.01	13
2.10 ± 0.01	20
2.80 ± 0.01	28
3.50 ± 0.01	35

Figure 28 - Voltage Conversion Table

From here we can see that our ADC is only very precise for a small range around the voltage we calibrated at. We can conclude that any calibration of the ADC will only be precise for a small range.

5. Conclusion

First, we studied the Schmitt Trigger and determined its hysteresis threshold voltage values.

$$V_{H->L} = 2.97V \pm 0.03V$$

$$V_{L->H} = 1.72V \pm 0.03V$$

We constructed a Trigger Pulse Generator using two components. A square wave generator, and a short pulse rectifier. Each component uses a Schmitt Trigger, and takes advantage of its hysteresis to prevent noise interfering with our circuit. The hysteresis of the threshold values allows the invertor to not trigger on any noise.

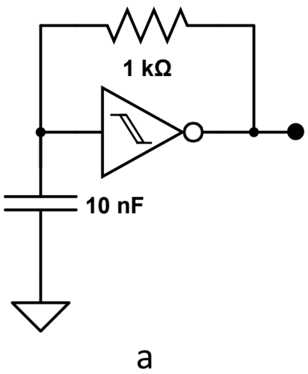


Figure 28 - Circuit diagram for square wave generator

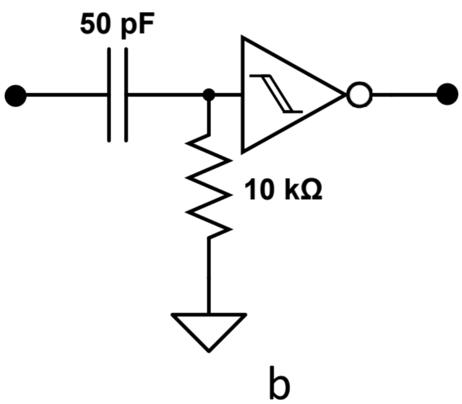


Figure 29 - Circuit diagram of short pulse rectifier.

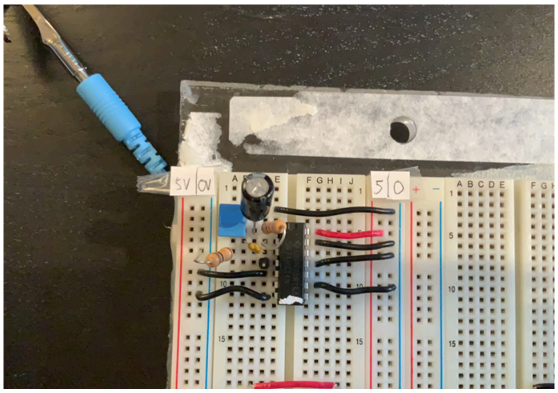


Figure 30 - Constructed TPG.

For our TPG, the left node of the short pulse rectifier is hooked up to the right node of the square wave generator. We verify its function by hooking up the the scope to the output of the TPG, and using single shot to confirm it emits a pulse.

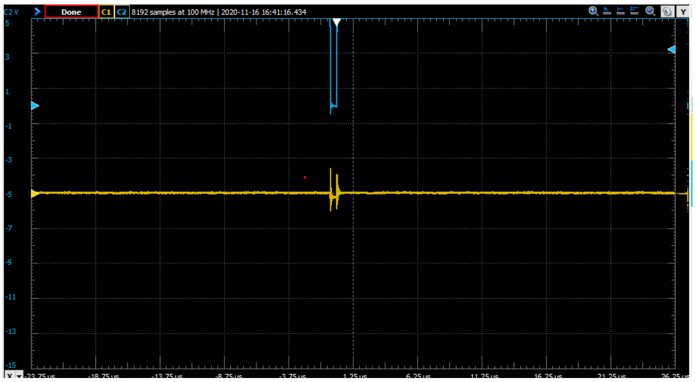


Figure 31 - Oscilloscope Single Shot of pulse.

Next, we constructed a Clock Pulse Generator as shown:

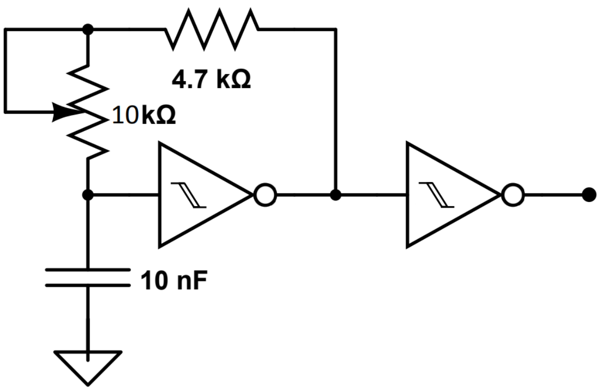


Figure 32 - Clock Pulse Generator Diagram

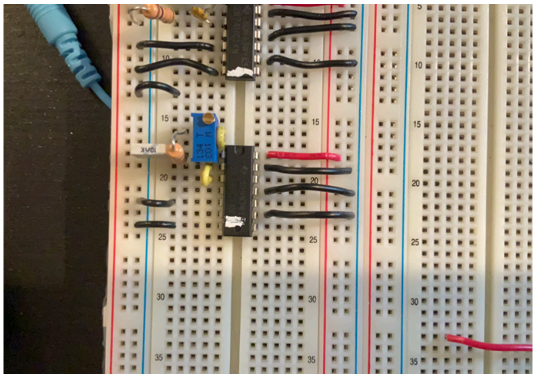


Figure 33 - Built Clock Pulse Generator

Hooking up an oscilloscope to the output of the CPG, we can verified its square wave output.

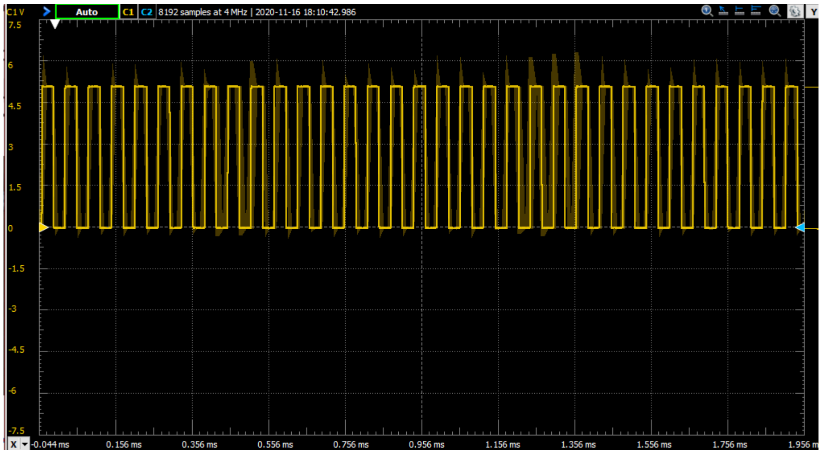


Figure 34 - Square wave output

By adjusting the potentiometer, we also verified that the frequency can be adjusted from $7.5kHz \pm 0.3kHz$ to $23.0kHz \pm 0.5kHz$. These values were measurements taken by the oscilloscope, and the uncertainty was taken to be the maximum variation of the measurements.

Next we created a divide-by-100 counter and hooked it up to a 7-segment display.

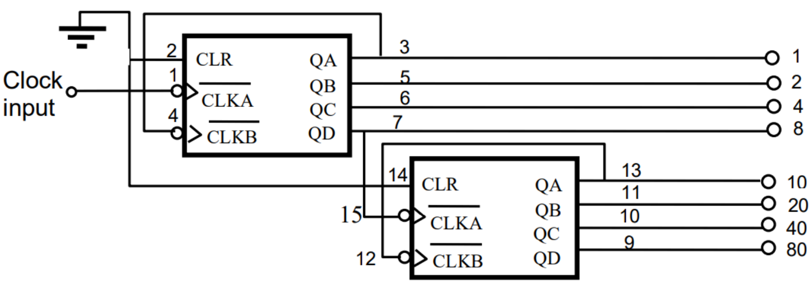


Figure 35 - Hundred Counter

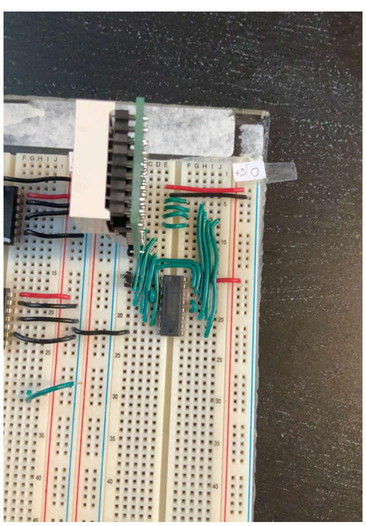


Figure 36 - Assembled counter.

It's operation was verified in the video: https://www.youtube.com/watch?v=edlV9Yeyc_A&feature=youtu.be

Next, we created an Analog to Time Convertor, which would generate a voltage ramp that increases linearly with time. This would be input into a comparator which would generate a stop pulse once the voltage exceeded the input voltage.

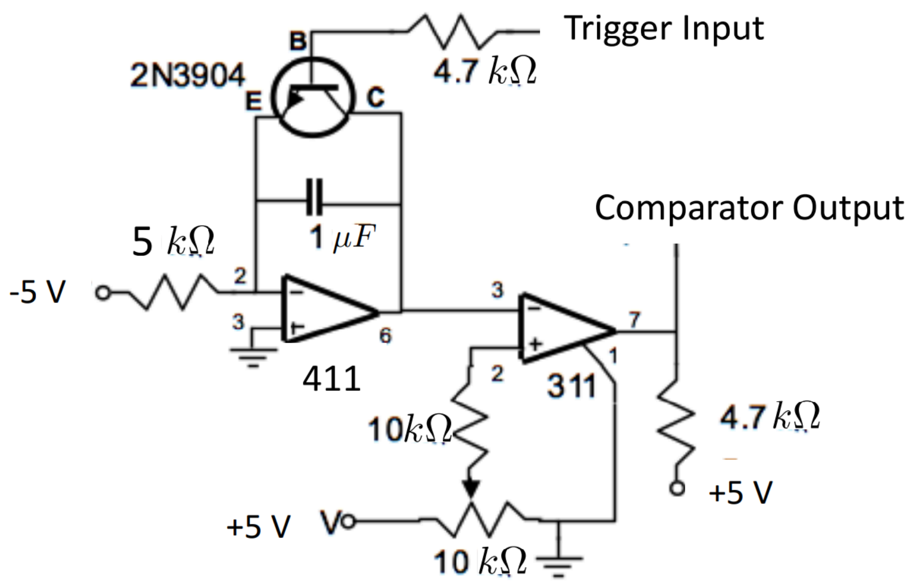


Figure 37 - ATC Circuit Diagram

Starting with the first Op-Amp circuit, assuming the capacitor starts with no charge, and the transistor is off, we can verify it produces a ramp as follows:

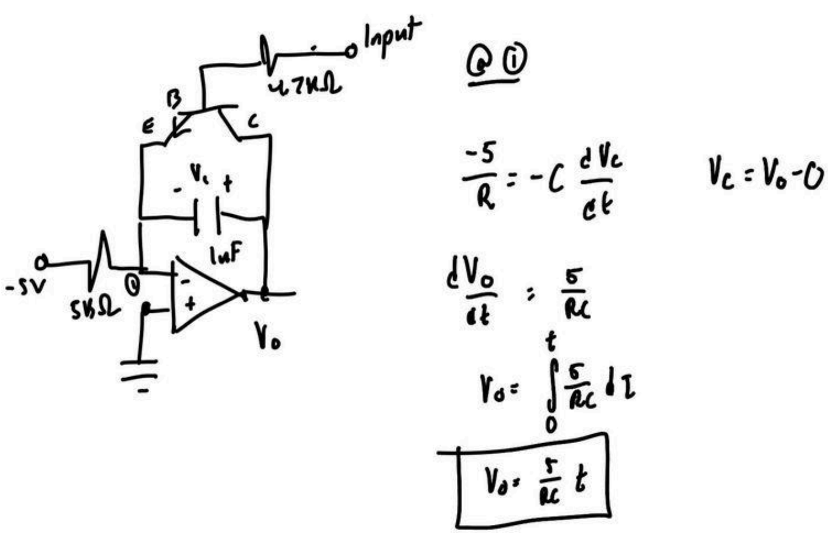


Figure 38 - Derivation of voltage ramp

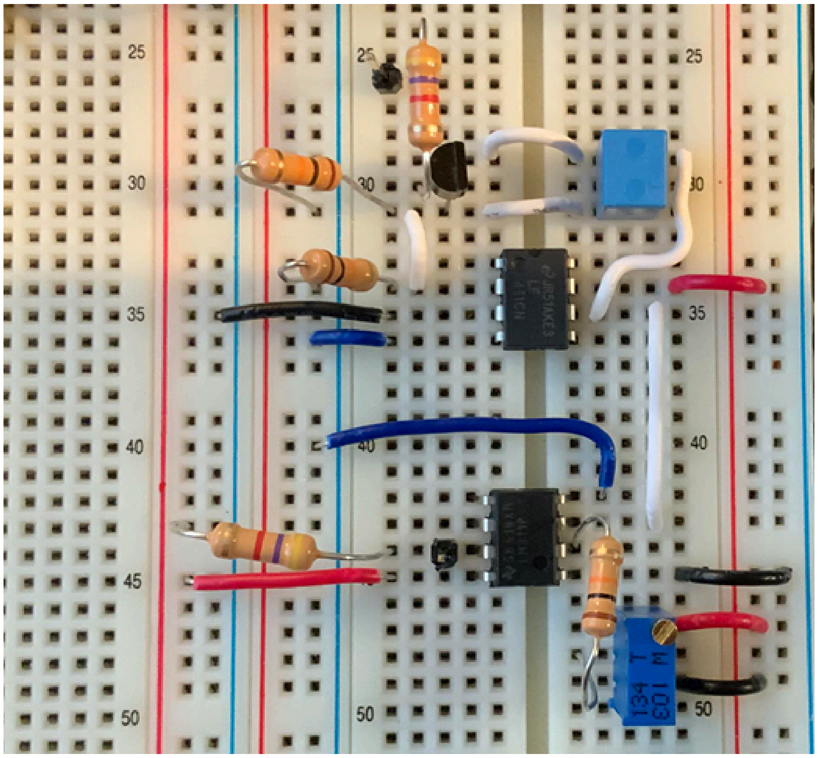


Figure 39 - Constructed ATC

We tested our constructed ATC by using two scopes to verify the ramp generation, and the comparator.

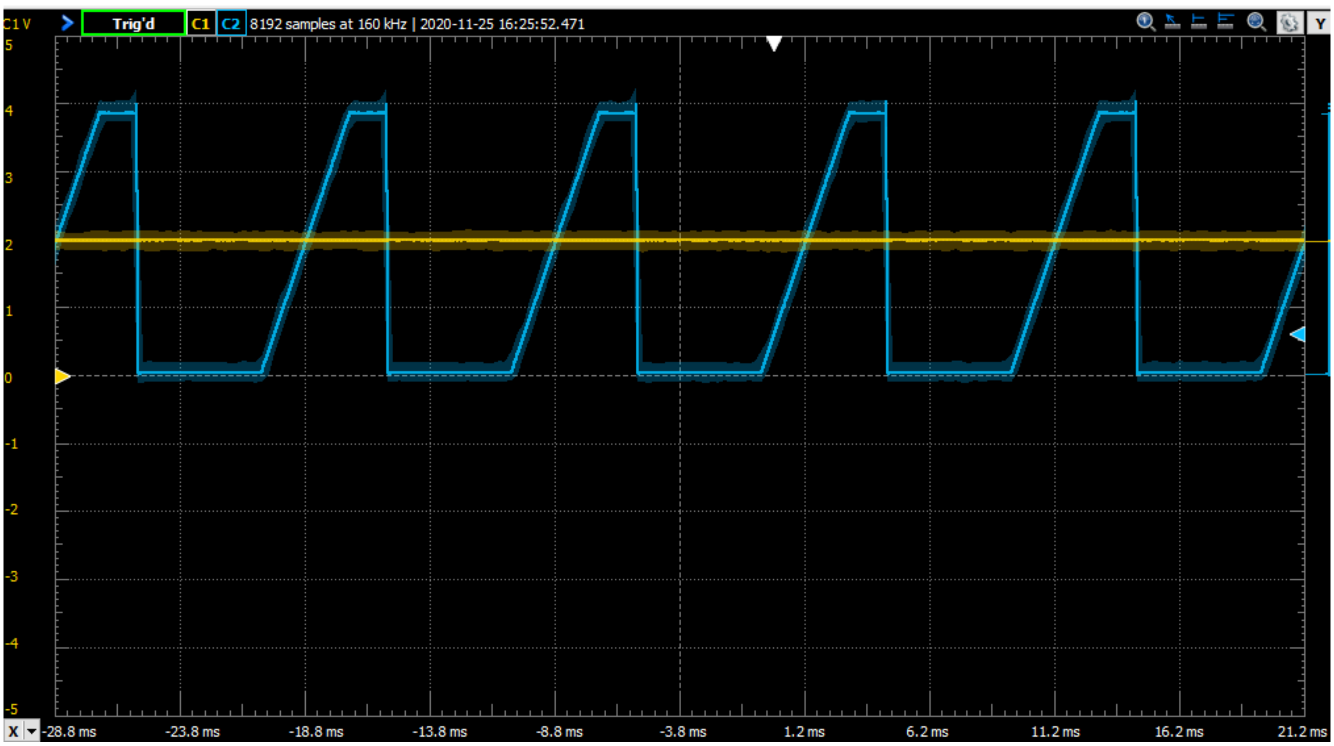


Figure 40 - Scope reading with yellow scope around input comparator voltage, and blue scope on the output of the voltage-time integrator.

Here we can see that the input comparator voltage is set at 2V. Now we move the yellow scope to the comparator output.

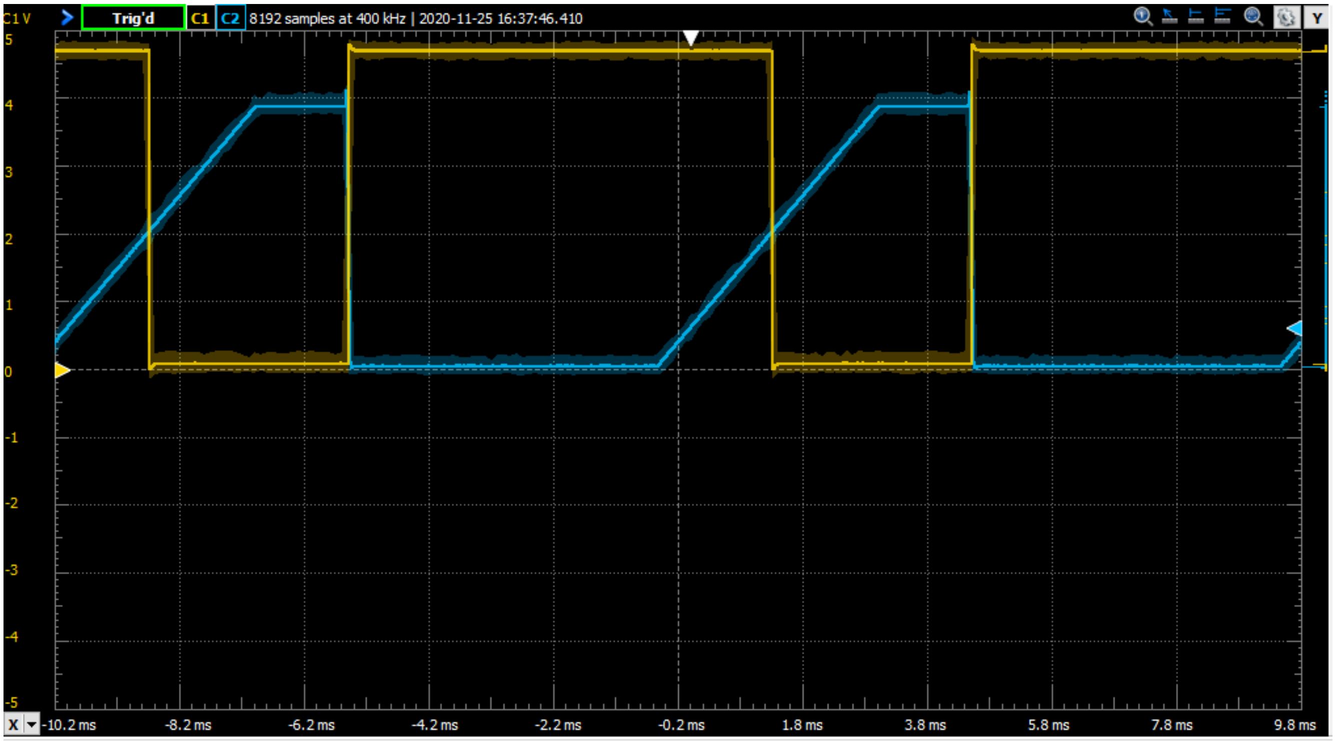


Figure 41 - Yellow scope is around comparator output, blue around output voltage ramp

And we see that the comparator turns off when the ramp exceeds 2V.

Lastly, we created a control logic circuit to coordinate the inputs/outputs of our components. This was created with two NAND gates.

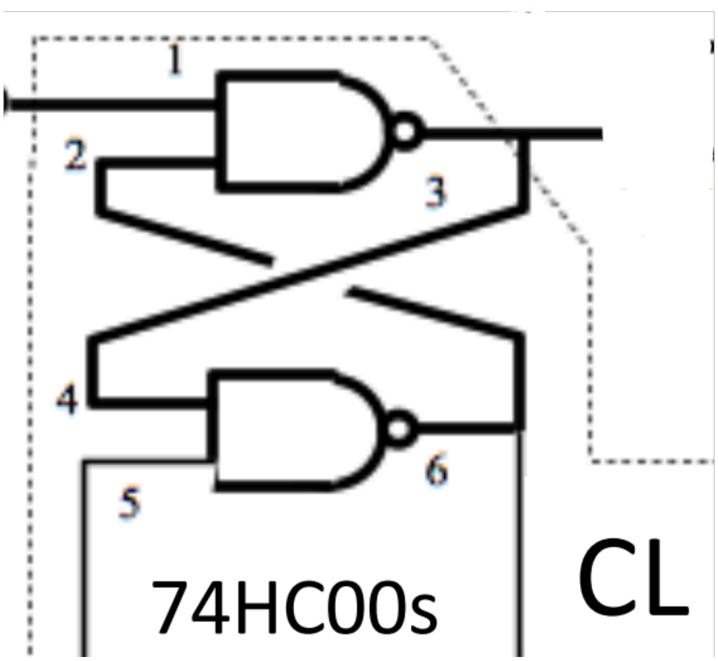


FIgure 42 - Control logic circuit diagram

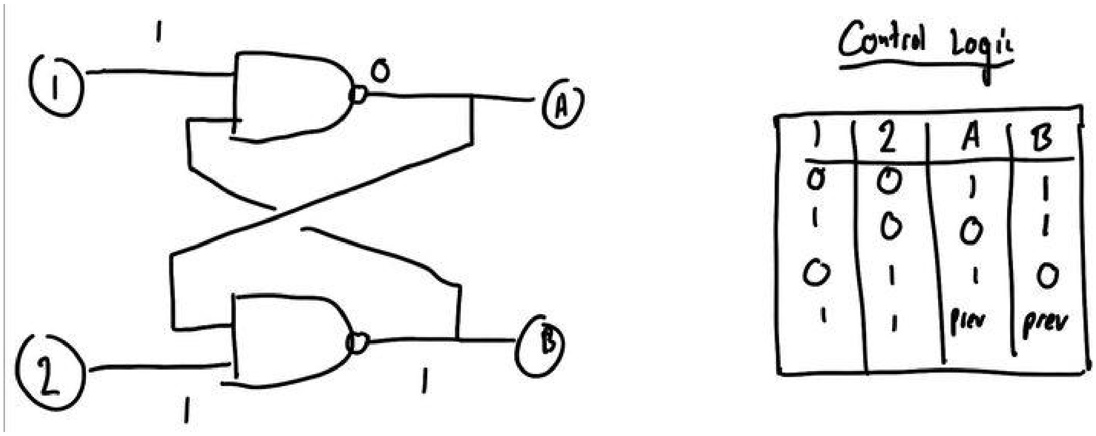


Figure 43 - Truth table

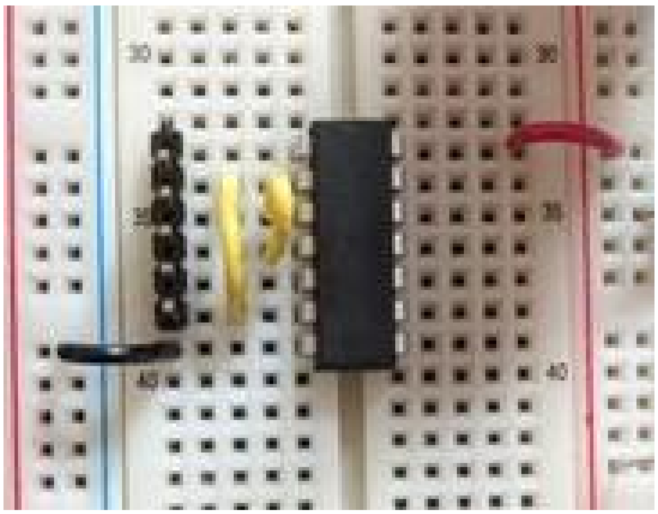


Figure 44 - Constructed Circuit

We tested our truth table by inputting the four different combinations.

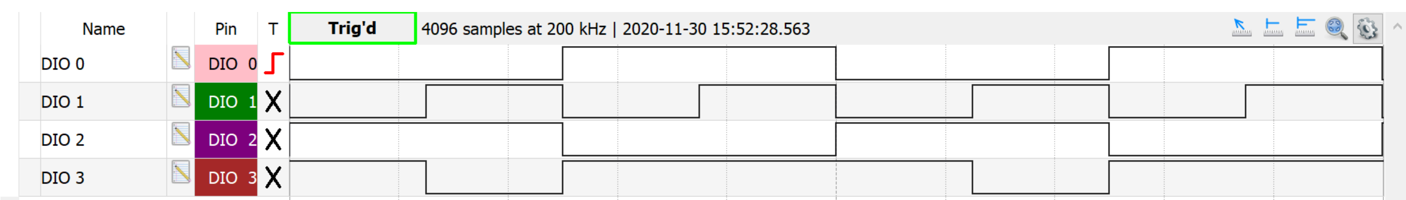


Figure 45 - Logic IO. DIO0 is the input to DIO3, and DIO1 is the input to DIO2.

Now we hooked up all the components together.

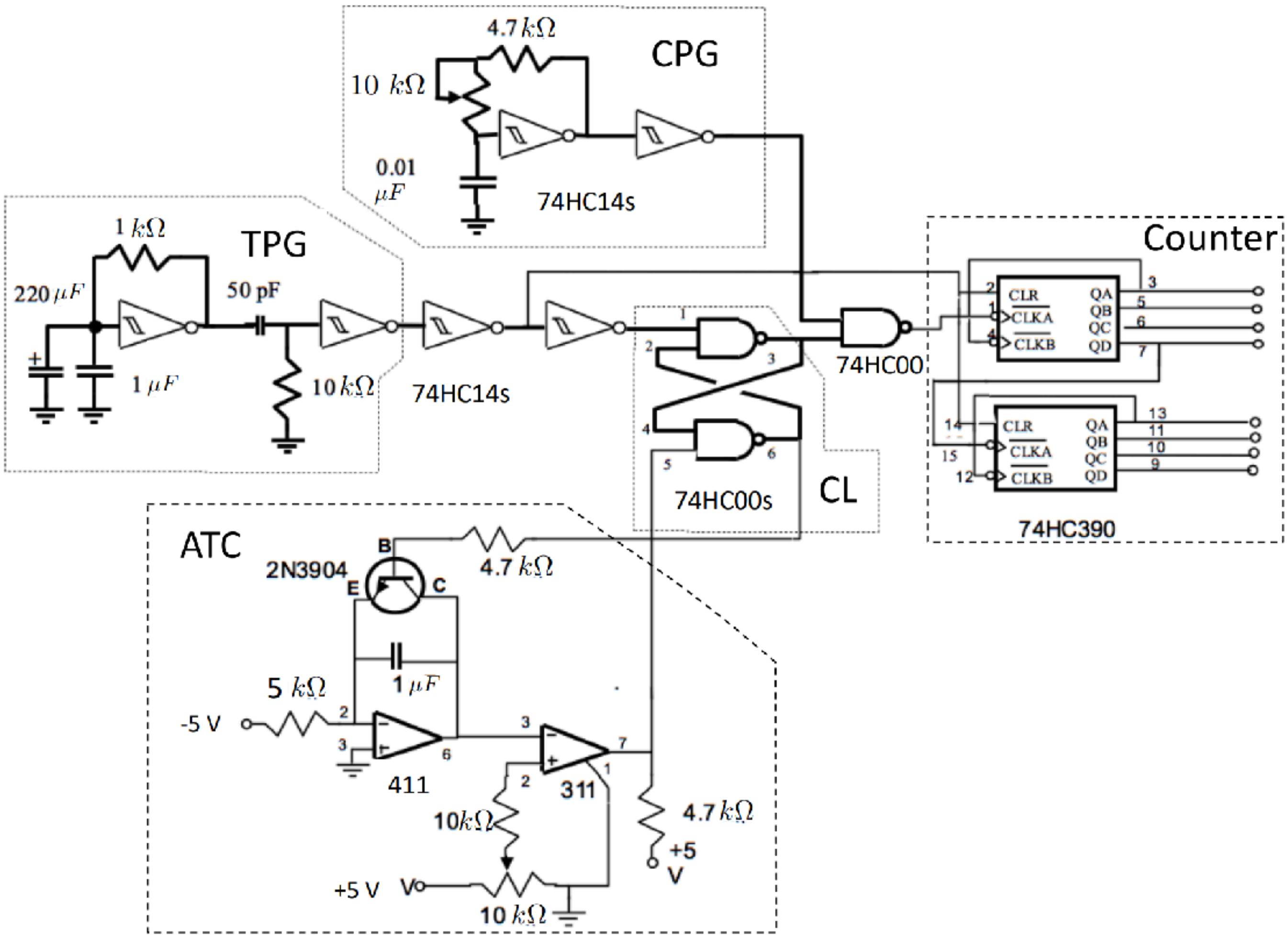


Figure 46 - Full ADC constructed

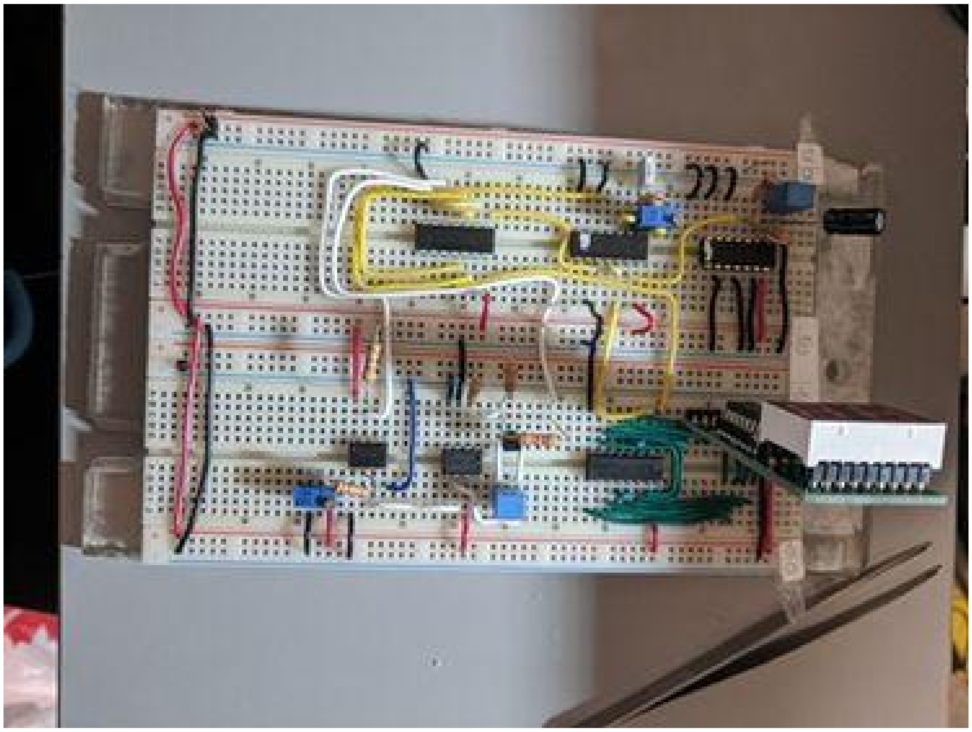


Figure 47 - Constructed ADC

This initial build did not work, and we had to troubleshoot. I first check the connections between components and luckily noticed that the wire connecting my TPG to the other Schmitt Trigger had been plugged into the input of the trigger, and not the output. After correcting the connection, the display turns on.

The display now turns on, and its value is changeable by the potentiometer in the ATC. Hooking up an oscilloscope to the output of the voltage potentiometer, and comparing it to the display we see the value is a bit off, so we calibrated by using an oscilloscope to make sure we set our input voltage to 2.5V, and adjusting the resistor in the TPG until the voltage and the display number were the same.

We then set the voltage to 1V, and hooked up digital input pins and the scope in the following places, and recorded data to generate a timing diagram. The Logic and Scope were both triggered on the falling edge of the control logic's timing window.

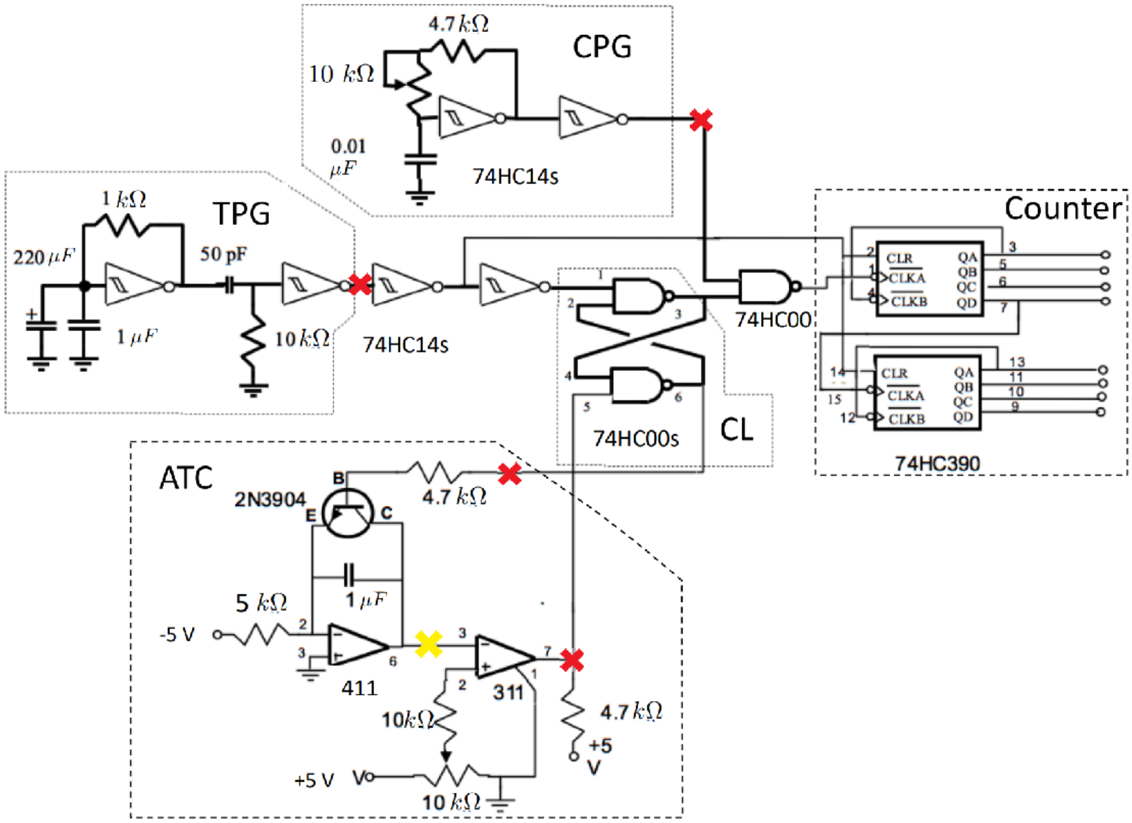


Figure 48 - Red cross signifies where we read digital input, yellow cross signifies where we read our scope.

4.1.2 Importing Waveform Data

Script for importing data from the following text file:

```
filename: C:\Evan\Eng\MatLab\Enph 259\Lab6\logic.csv
```

Auto-generated by MATLAB on 02-Dec-2020 15:09:59

Setup the Import Options and import the data

```
opts = delimitedTextImportOptions("NumVariables", 5);

% Specify range and delimiter
opts.DataLines = [1300, 6000];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["Times", "CPG", "TPG", "Comparator", "CL"];
opts.VariableTypes = ["double", "double", "double", "double", "double"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Import the data
tbl = readtable("C:\Evan\Eng\MatLab\Enph 259\Lab6\logic.csv", opts);
```

Convert to output type

```
Times = tbl.Times;
CPG = tbl.CPG;
TPG = tbl.TPG;
Comparator = tbl.Comparator;
CL = tbl.CL;
TPG = TPG + 1.02;
Comparator = Comparator + 2.02;
CL = CL + 3.06;
```

Clear temporary variables

```
clear opts tbl
```

Script for importing data from the following text file:

```
filename: C:\Evan\Eng\MatLab\Enph 259\Lab6\scope.csv
```

Auto-generated by MATLAB on 02-Dec-2020 15:13:59

Setup the Import Options and import the data

```
opts = delimitedTextImportOptions("NumVariables", 3);

% Specify range and delimiter
opts.DataLines = [3500, 8000];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["ScopeTimes", "Channel1V", "Var3"];
opts.SelectedVariableNames = ["ScopeTimes", "Channel1V"];
opts.VariableTypes = ["double", "double", "string"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, "Var3", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Var3", "EmptyFieldRule", "auto");

% Import the data
tbl = readtable("C:\Evan\Eng\MatLab\Enph 259\Lab6\scope.csv", opts);
```

Convert to output type

```
ScopeTimes = tbl.ScopeTimes;
Channel1V = tbl.Channel1V;
Channel1V = Channel1V + 4.02;
```

Clear temporary variables

```
clear opts tbl
plot(Times, CPG, Times, TPG, Times, Comparator, Times, CL, ScopeTimes,Channel1V)
ylim([0, 5.1])
xlim([-0.00005, 0.001])
legend("CPG", "TPG", "Comparator", "Control Logic", "Integrator Ramp", 'Location', 'bestoutside
set(gca, 'ytick',[]);
title('Timing Diagram for ATC')
xlabel('Time(s)')
ylabel('Voltage')
```

The TPG gives a very short pulse relatively infrequently, and starts the data colection, while the Comparator gives a very short pulse to end the data collection once the voltage ramp has exceeded the input voltage.

The control logic going to GND represents the timing window in which we want to count pulses, and the Integrator Ramp represents the climbing voltage.

The counter counts how many pulses are generated by the CPG within the timing window.

In this case, the time it takes our voltage ramp to reach 1V is around 0.9ms.

Finally, we compared 5 voltage values and created a voltage relationship table.

Voltage Set (V)	DMM Display Number
0.70 ± 0.01	06
1.40 ± 0.01	13
2.10 ± 0.01	20
2.80 ± 0.01	28
3.50 ± 0.01	35

Figure 49 - Voltage Conversion Table

We saw that by calibrating our ADC at 2.5V, only a small range around that value was precise, and going further would produce a slightly off number.