Advanced Light Intensity Indicator

EE3024 - Digital Signal Processing

(G-16)

230605P - SHAHITHYAN R.

230617E - SIRIBADDANA S.P.P.

230620G - SNEKAN S.

230624X - S. SUBEETSHAN

230627J - SURAWEERA S.C

❖ Introduction and system overview

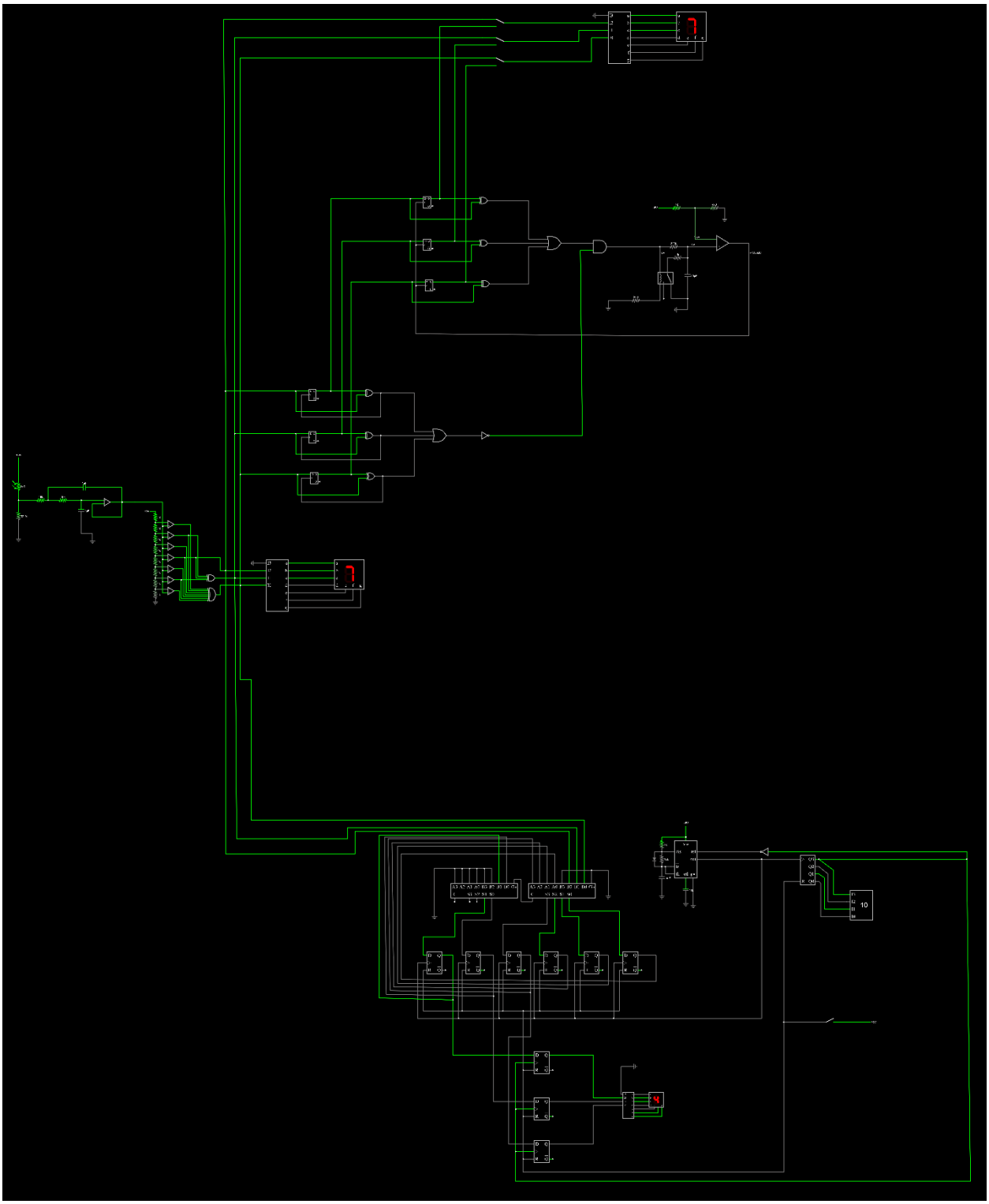
- Lighting is a huge part of our daily energy use. To save electricity, it makes sense to dim or turn off artificial lights when there is enough natural sunlight. However, building a system to do this reliably is difficult because sensors can be tricked by passing shadows or electrical noise. This project, the Advanced Light Intensity Indicator (ALII), solves these problems to help save energy and keep public spaces safely lit.

- The ALII is a "smart" light sensor system. It measures how bright the room is and shows the level on a digital display (0 to 7). What makes this project unique is that it is built entirely without computers or microcontrollers. Instead, it uses pure electronic components, such as logic gates, transistors, and amplifiers, to "think" and make decisions.

- During this project, we had to overcome these main challenges,
    1. **Electrical Noise:** The wiring in buildings creates a 50 Hz "hum" that can mess up sensor readings. We had to filter this out so the display wouldn't flicker.
    2. **False Alarms:** A person walking or a vehicle moving past a sensor can block the light for just a second. We didn't want the system to react to these quick changes, so we needed a way to ignore them and show the intensity of light that lasts at least 30-300 seconds.
    3. **Seeing the average light intensity for a long period of time:** Sometimes, we may want to know how bright it has been in the environment during the past 10 minutes; we had to make a circuit to calculate and display the average light intensity over the 10 minutes.

- Therefore, we designed our ALII module to perform these four specific tasks,
    1. **Filter the Signal:** Use an analog filter to remove electrical noise above 50Hz to obtain a clear electrical signal.
    2. **Show the Level of light intensity:** Convert the analog light signal into a simple digital signal and represent it on a seven-segment display by using seven intensity levels.
    3. **Wait for Stability:** Ensure the light level stays the same for 30–300 seconds before updating the screen, to prevent recognizing rapid changes of light due to sudden interruptions.
    4. **Calculate the Average:** Build a digital averaging circuit that remembers the light level over 300-900 seconds and shows the average value on a second screen.

❖ Complete circuit simulation diagram



❖ Feature-by-feature explanation

a) Analog filtering stage

- The purpose of the analog filtering stage is mainly to filter up these three types of noise.
  1. Power line interference (typically 50-60 Hz)
  2. Artificial light flickering in artificial lighted environments (typically 50-100 Hz)
  3. High frequency switching noise by nearby power supplies (Phone chargers, Laptop bricks, Wi-Fi routers, etc.

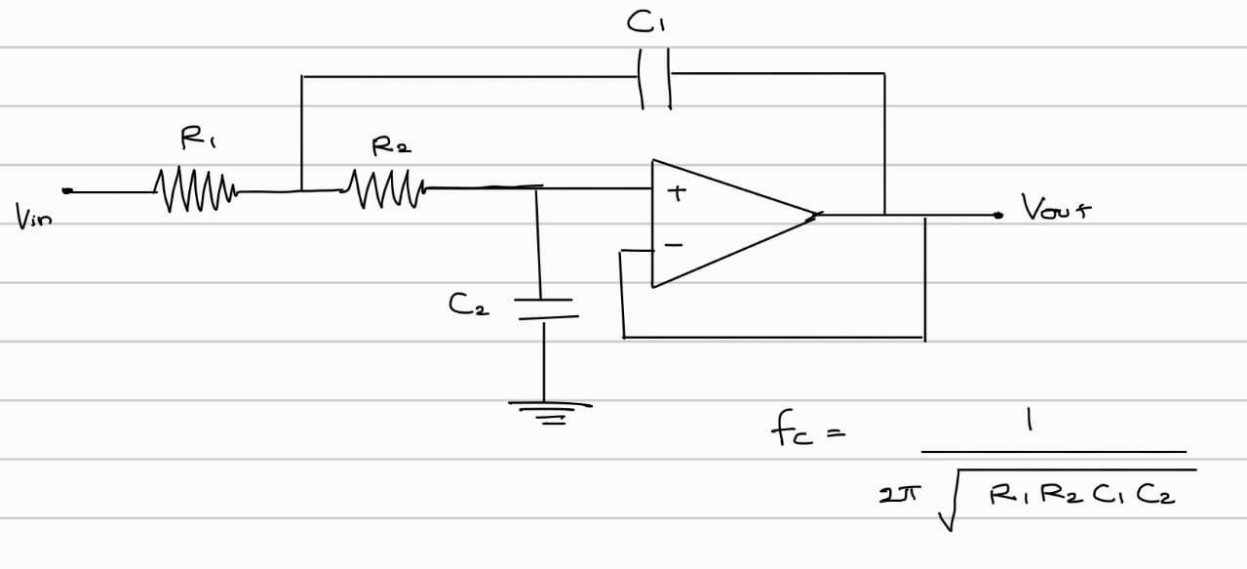
- To filter these noises, we must use a low-pass filter. We decided to use a low-pass active filter instead of a passive filter to eliminate the loading effect caused by a passive filter of the same order, due to the availability of more components, such as inductors, in the passive filter. But when using an Active filter with an operational amplifier, it acts as a Buffer circuit, which ensures that the voltage representing light intensity is transferred to the quantization stage (ADC circuit) without attenuation or distortion.
- And, this 2^{nd} order active filter provides a 40dB attenuation per decade, ensuring the sharp knee that passes 10Hz signals while attenuating the 50Hz noise.

➢ 2^{nd} Order Sallen-Key Low-pass filter

- Cutoff frequency(fc) = 10Hz
- Stopband frequency = 50Hz

- Attenuation = 40dB/decade

○ Filter Calculations



- Let's choose R1=R2=R and C1=C2=C to be easy in the physical implementation of the filter. Therefore, the equation is changed as,

$$f_c = \frac{1}{2\pi RC}$$

- Let C=1uF according to the market value,

$$R = \frac{1}{2\pi f C} = \frac{1}{2\pi \times 10 \times 1\times10^{-6}}$$

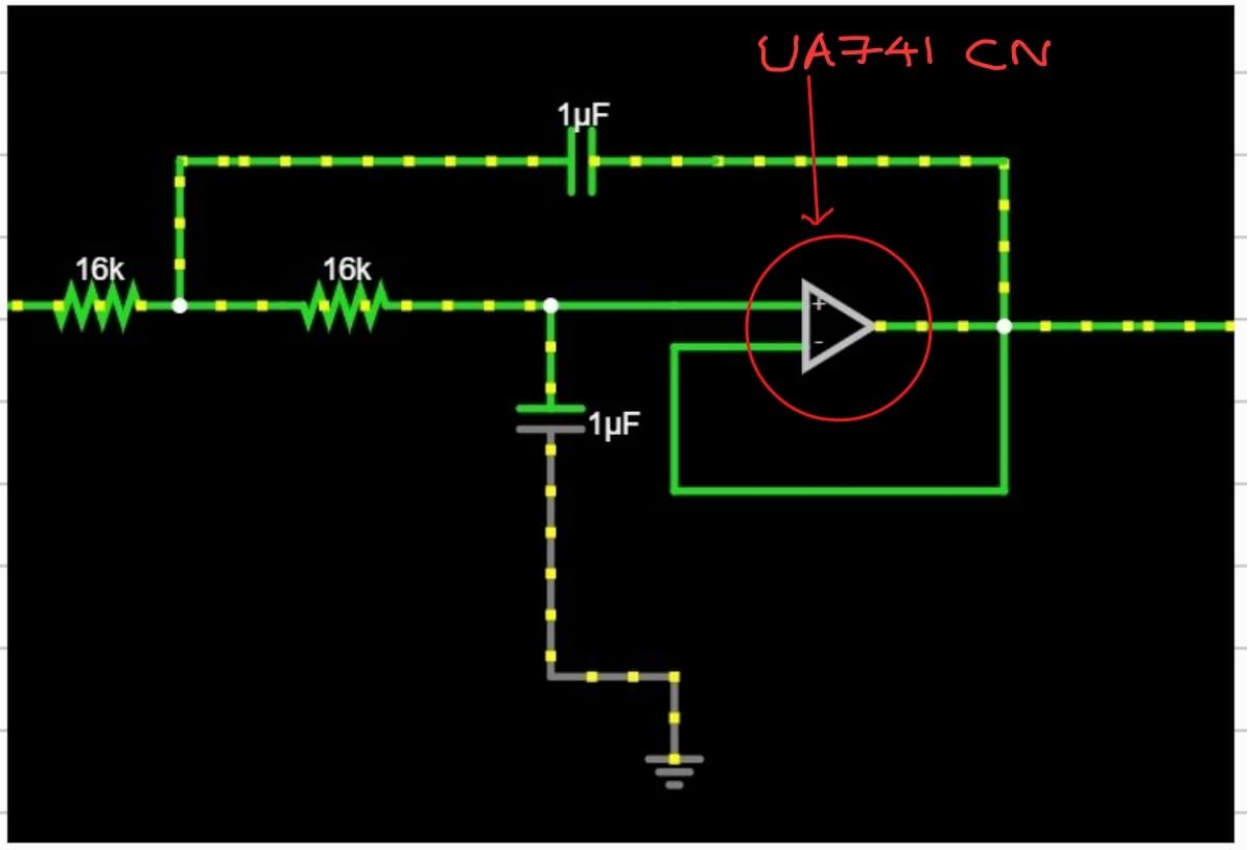
$$R = 15915.45\ \Omega$$

$$R = 15.915\ K\Omega \approx 16\ K\Omega$$

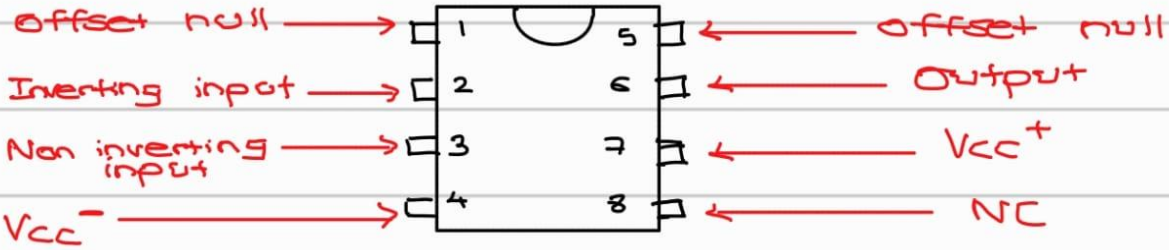
- We had to use a 16KΩ capacitor in the market instead of the 15.915 KΩ value. Therefore, the cut-off frequency slightly changed from the original 10Hz value.

$$\therefore f_{c(actual)} = \frac{1}{2\pi (16\times10^{3})(1\times10^{-6})} = 9.947 \text{ Hz}$$

- Here is the simulation of the filter with calculated resistor and capacitor values

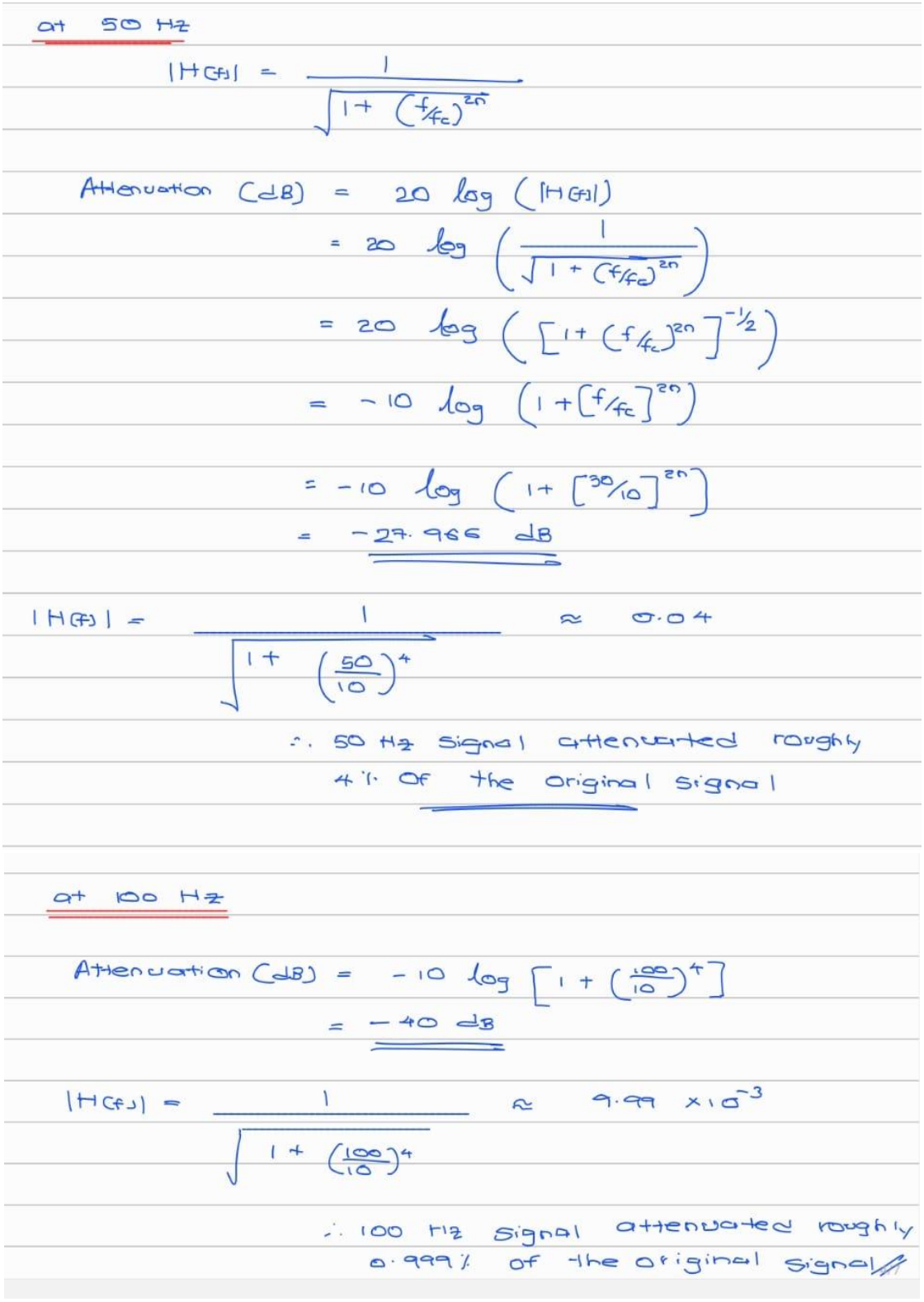


- We've used the UA741CN single OP-AMP IC for this filter, and given below is the pin configuration of UA741CN.

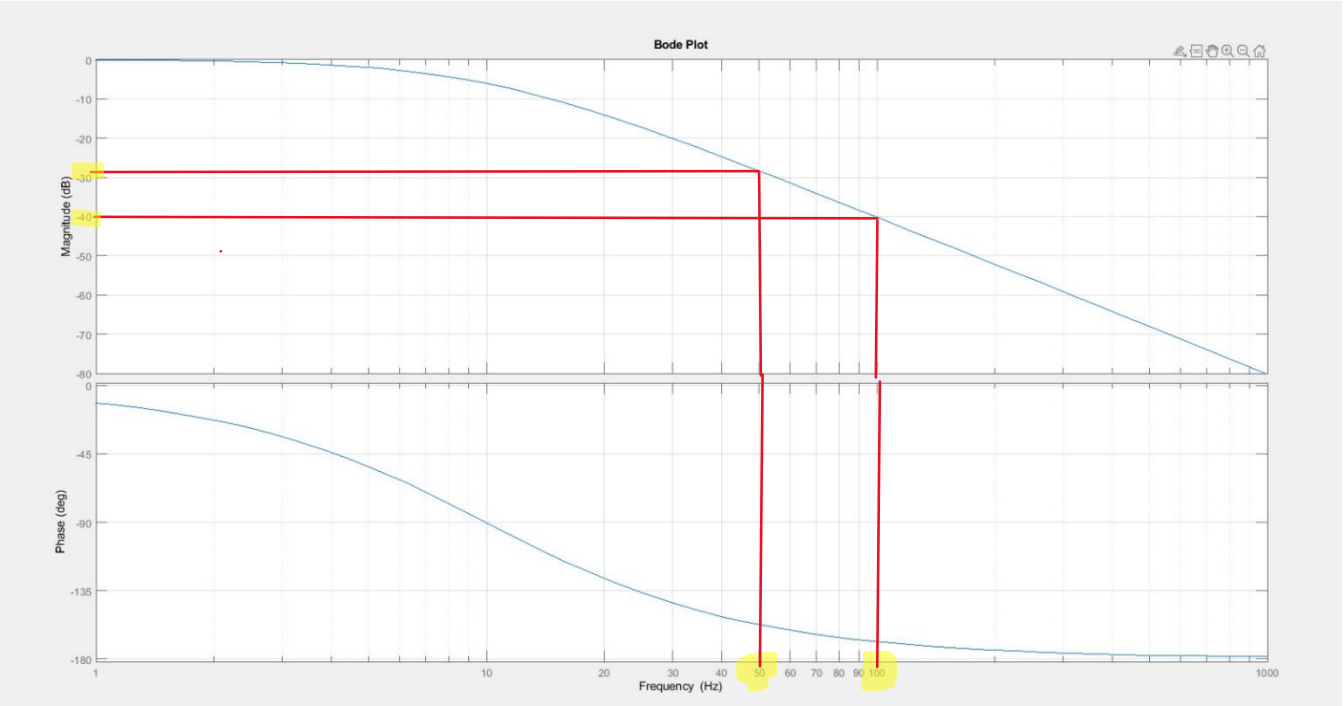


○ Filter Performance Analysis

- Let's calculate the stopband attenuation of the filter at 50Hz and 100Hz.



- Therefore, we can see this second-order low-pass Sallen-Key filter gives 96% attenuation at 50Hz and 99% attenuation at 100Hz.
- Given below is the Bode plot of the second-order low-pass Sallen-Key filter.



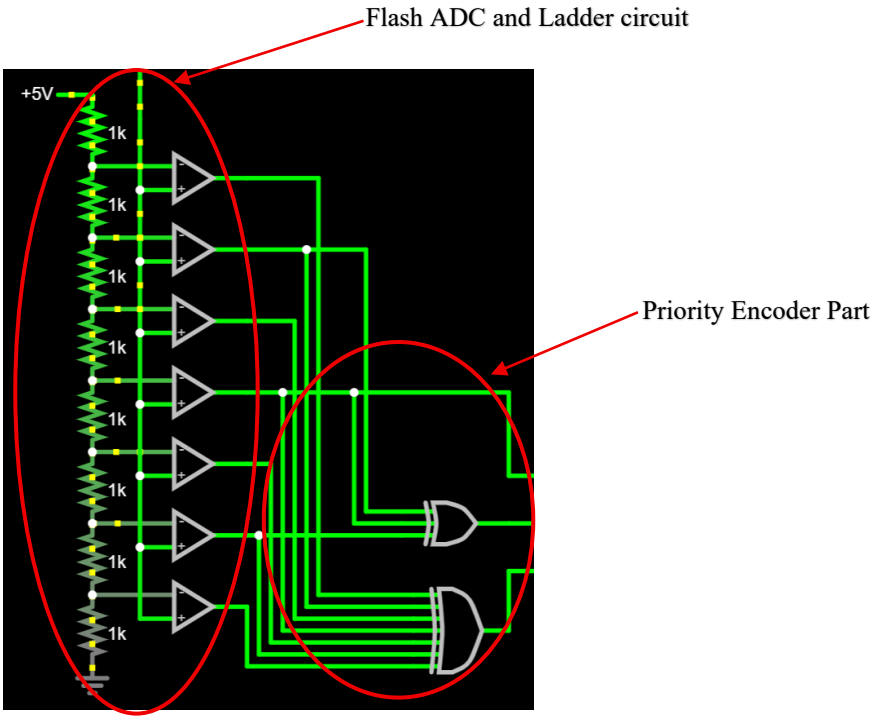
b) ADC part

The main idea behind this design is simple. We take the changing light level detected by the LDR, and we convert that into a digital value using the flash ADC.

This part consists of four main stages.

1. LDR Sensing Stage
2. Voltage Divider
3. Flash ADC
4. Priority Encoder

Each stage helps signal step by step until we finally obtain a clean digital output.



LDR

Changes in Resistance corresponding to Light Intensity: The Higher the Light Intensity Lower the resistance. We combine LDR with a variable resistor to form a voltage divider. This gives us an analog signal smoothly varying with the amount of light falling on the LDR this varying voltage becomes the input to our ADC.

Low-pass filter

Remove unwanted fluctuations or electrical noise.

Flash ADC and resistor ladder

The next major stage is flash ADC. This works by using many comparators in parallel. The first part is a resistor ladder. The ladder divides the reference voltage into several evenly spaced threshold levels. The reference voltages are fed into the comparator's negative point. If the input voltage is higher than the reference voltage, the comparator output shows high, or else it remains low. So, based on the light level, different numbers of comparators will turn on.

Priority Encoder

We turn the 7 comparators' output into a 3-bit binary value using a Priority Encoder, specifically the 74LS148N model.

Challenges

1. Reference Voltage instability, Use a Voltage Regulator

Use Precision Resistor

2.LM339N Model output was Pulled Low When Active, Use Pull-Up Resistor

3. Challenges in Power Consumption: Use Lower Power Comparators

4. Wiring Complexity, Use Flat Wires as Much as Possible

Alternatives

Here, we can use the LM393N model instead of the LM339N model.

Also, we can use 74LS148N instead of 74HC148N.

c) Stability Circuit.

In this stability circuit, the three output bits from the ADC are sent to the D terminals of the first three D flip-flops. There are already existing values in the Q terminals of these three D flip-flops. The XOR gates compare the current D value with the value at the respective Q terminal (Previous D value), and if they are different, the XOR output will be 1. These outputs of the XOR gates act as clock

signals to the D flip-flops. When these D flip-flops receive the clock signal, they update their D and Q values.

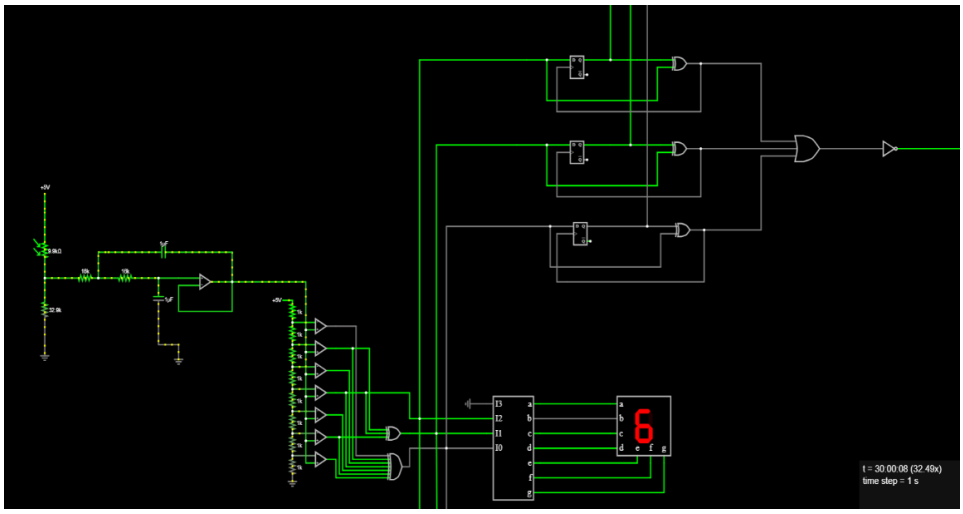


Figure (1)

Then the XOR outputs are sent through an OR gate. If at least one of the above shown XOR gates is activated, then it will trigger the connected OR gate. Then the OR gate output is negated using a NOT gate. The Q outputs of the D flip-flops are then sent to the D inputs of another three D flip-flops.

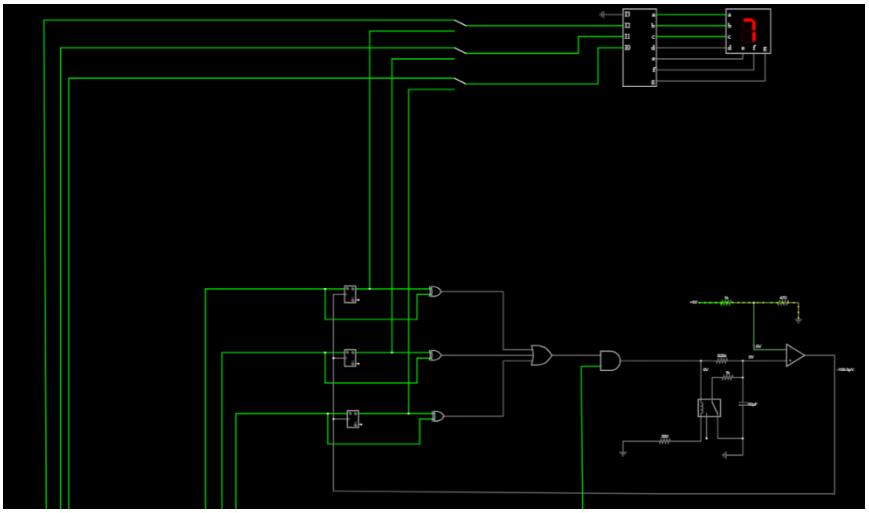


Figure (2)

The outputs of these D flip-flops and their respective D inputs are sent through three other XOR gates. These XOR gates compare the previously stored value with the current values and output 1 if they are different. The previously stored values of the D flip-flops correspond to the value shown in the seven-segment display. If at least one of the XOR outputs is 1, which means the D and Q values of the respective D flip-flop are different, it will activate the connected OR gate output, which then passes through an AND gate. The other input to the AND gate is the negated OR

output shown in Figure 1. When both are at high voltage, the AND gate output will be 1.

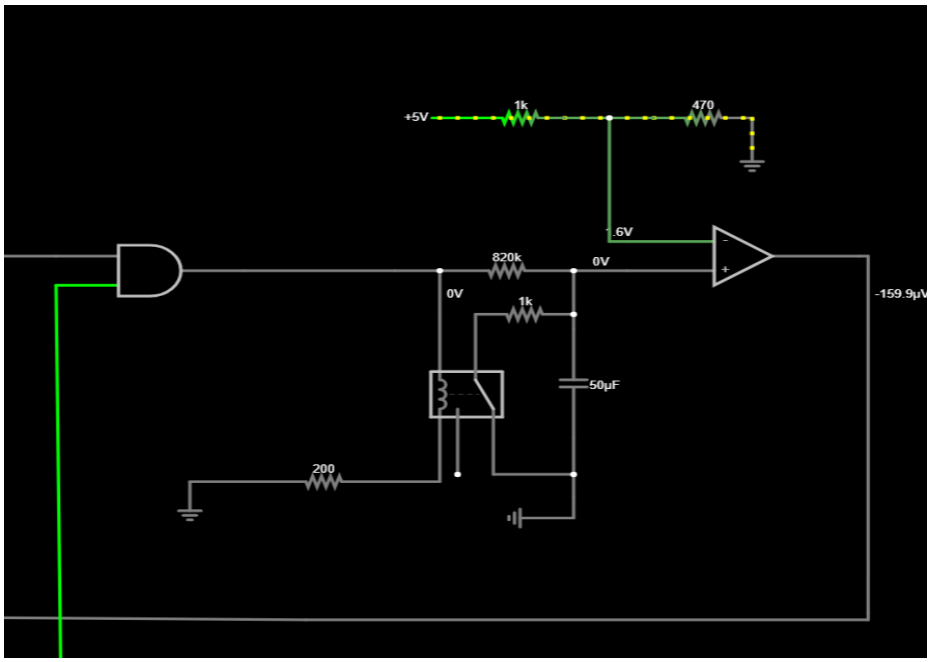


Figure (3)

The output of the AND gate is then sent to the circuitry shown in Figure 3. When the AND output is high, a current is passed through the inductor of the relay switch shown in Figure 3.

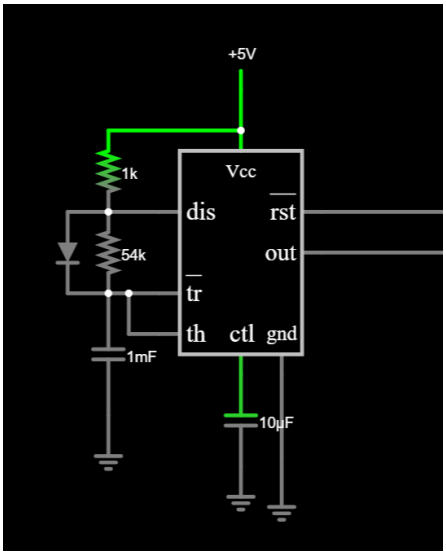
The inductor will create a magnetic field due to the current, and the switch shown will move towards the terminal closer to the inductor. As a result, the capacitor will get charged. When the voltage across the charged capacitor increases above the voltage of the inverting terminal of the op-amp shown, which is adjusted by the voltage divider circuit shown, then the output of the op-amp will be high. This will be used as a clock signal to update the D flip-flops in Figure 2.

When the values of the inputs and the outputs of the D flip-flops shown in Figure 2 are the same, then the segment will update to the new value as the stable value. The switches shown in Figure 2 serve the purpose of mode switching between the stability mode, which shows the stable value maintained for more than a given time, and the instantaneous mode, which shows the instantaneous light intensity.

d) Averaging Circuit

555 Timer and the Counter.

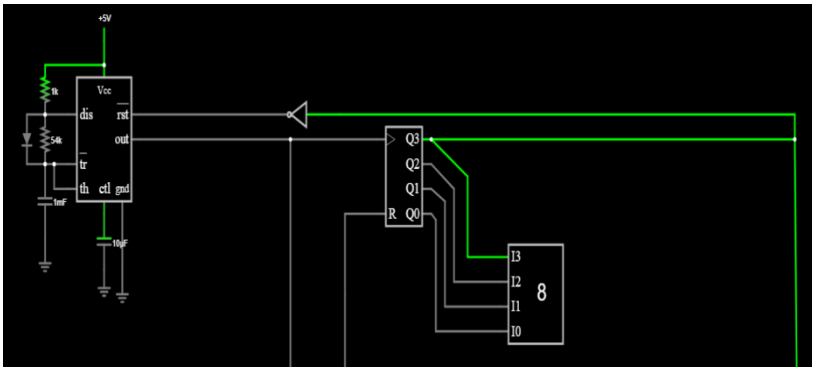
- The timer in this circuit provides clock signals to the D flip-flops. The timer used here is NE 555N.
- The main functionality of the counter is to count the number of pulses generated by the timer.



Here, the timer was used in the astable configuration. Here, the values 1kΩ, 54kΩ(Variable),1mF are calculated based on the equation $\boldsymbol{T = 0.693 \times (R1 + R2) \times C}$ which gives the width of a high voltage pulse generated by the timer. 1mF was a market value, and the others were based on the calculation from the above equation.

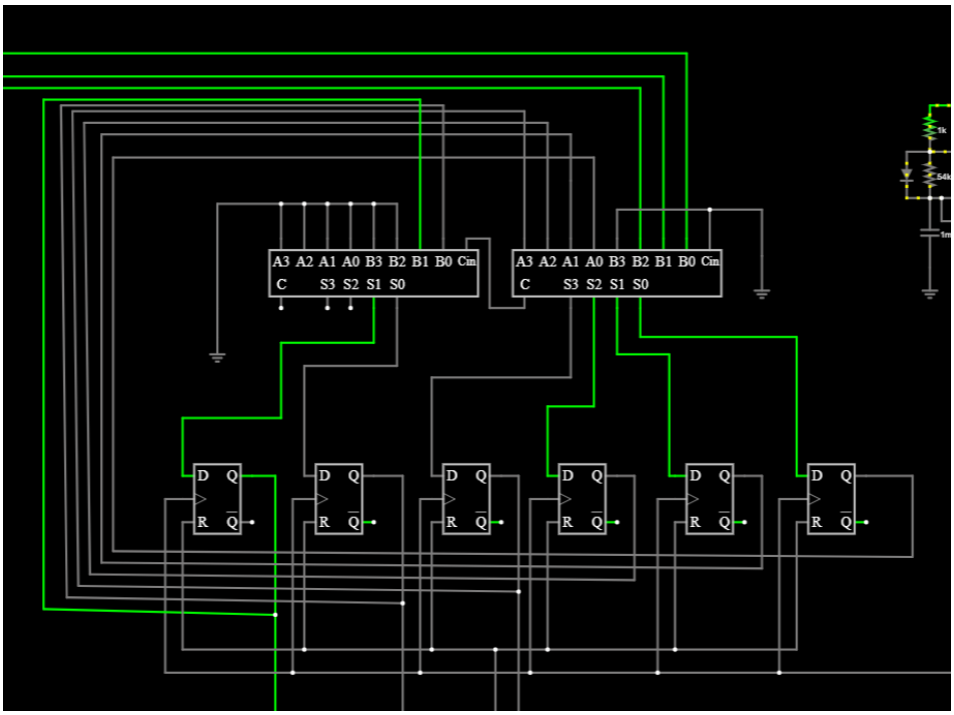
The control pin was grounded through a 10nF capacitor. This enables us to provide a reference voltage without any noise interruptions; there may be noise due to the power line interferences and other circuitry. Capacitors of small capacitance values are better for noise filtering.

The counter counts high voltage pulses from the timer and outputs the binary value corresponding to the decimal count. Here we are getting 8 samples, so we are going to stop the counter after 8 pulses. When the counter counts 8, the line that indicates the bit at the Q3 position is activated, and then this line is given to the reset pin of the timer through a NOT gate. After this, the timer is reset, and it will stop generating the pulses. To activate the timer, once again, we must enable the line that is connected to the reset pin.



Adder and Averaging Logic.

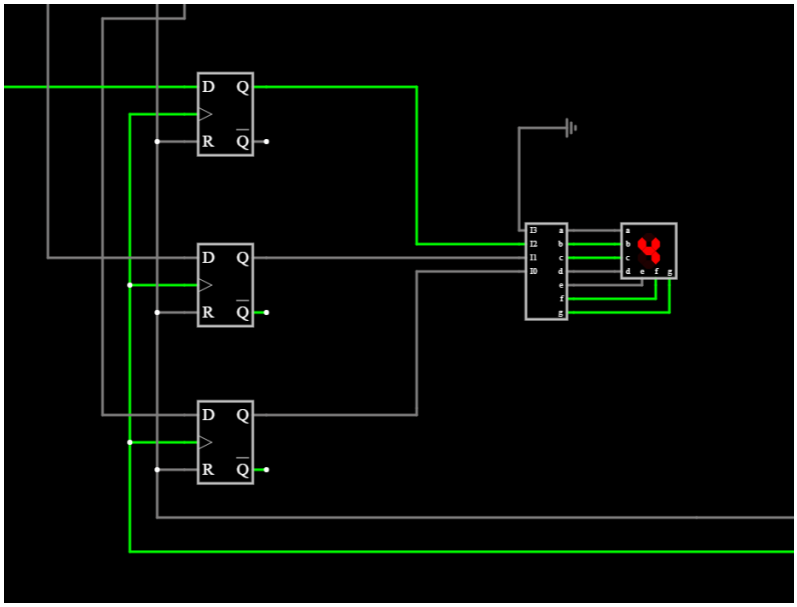
Initially, all the Q values of D flip-flops should be 0. To do that, the reset push button is implemented. At the reset condition, pins on the right side of the first adder (B0, B1) and the left side (A0, A1, A2, A3) will be at 0. These 6 bits represent the previous sum value. Here to get the maximum sum, 6 bits are sufficient. So, all other pins to the left side of the second adder (A0, A1, A2, A3, B2, B3) are grounded.



Here, two 4-bit adders are used. The carry out of the adder to the right side is sent to the adder to the left of the figure as carry in.

In the six D flip-flops, the D values will be the current sum, and the Q values will be the previous sum. Initially, (in the reset condition), Q values are 0. D values will be the sum of 0 and ADC output. When the pulse comes from the 555-timer as a clock signal to the D flip-flops, Q will take the D values, and these Q values are sent to the adders as the input. Then that transferred value and the current ADC value will be added and saved as the D value (it can vary instantly). When the next pulse comes, it will be transferred to Q and will be sent to the adder as the input. This addition operation will happen 8 times as a loop.

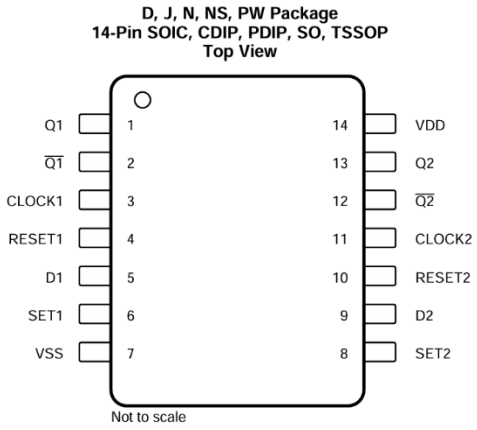
To get the average, we should divide the sum by 8. This is equivalent to a 3-bit shift in the binary domain. So, the first 3 bits from the left corner are taken to the 3 D flip-flops as the D values.

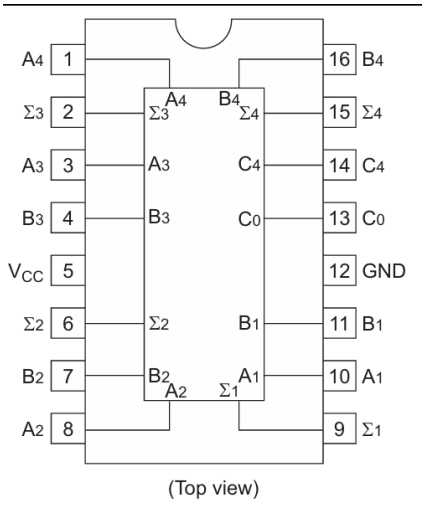


When the 8th pulse comes, the counter's Q3 value will be 1. That will be sent through a NOT gate and sent to the 555-timer as a reset. When the reset pin of the 555-timer becomes 0, it will stop generating pulses. At the same time, this Q3 will be sent to the 3 D flip-flops shown in the figure as a clock signal. When the flip-flops receive the clock signal, they will update their D and Q values. Then it will be sent to the decoder, and the decoder output will be sent to the 7-segment display. Then it will display the average value.

Here, the codes of the D flip-flops and adders used are respectively CD4013BE and SN74HC83.

Here are the pin configurations, respectively.





****