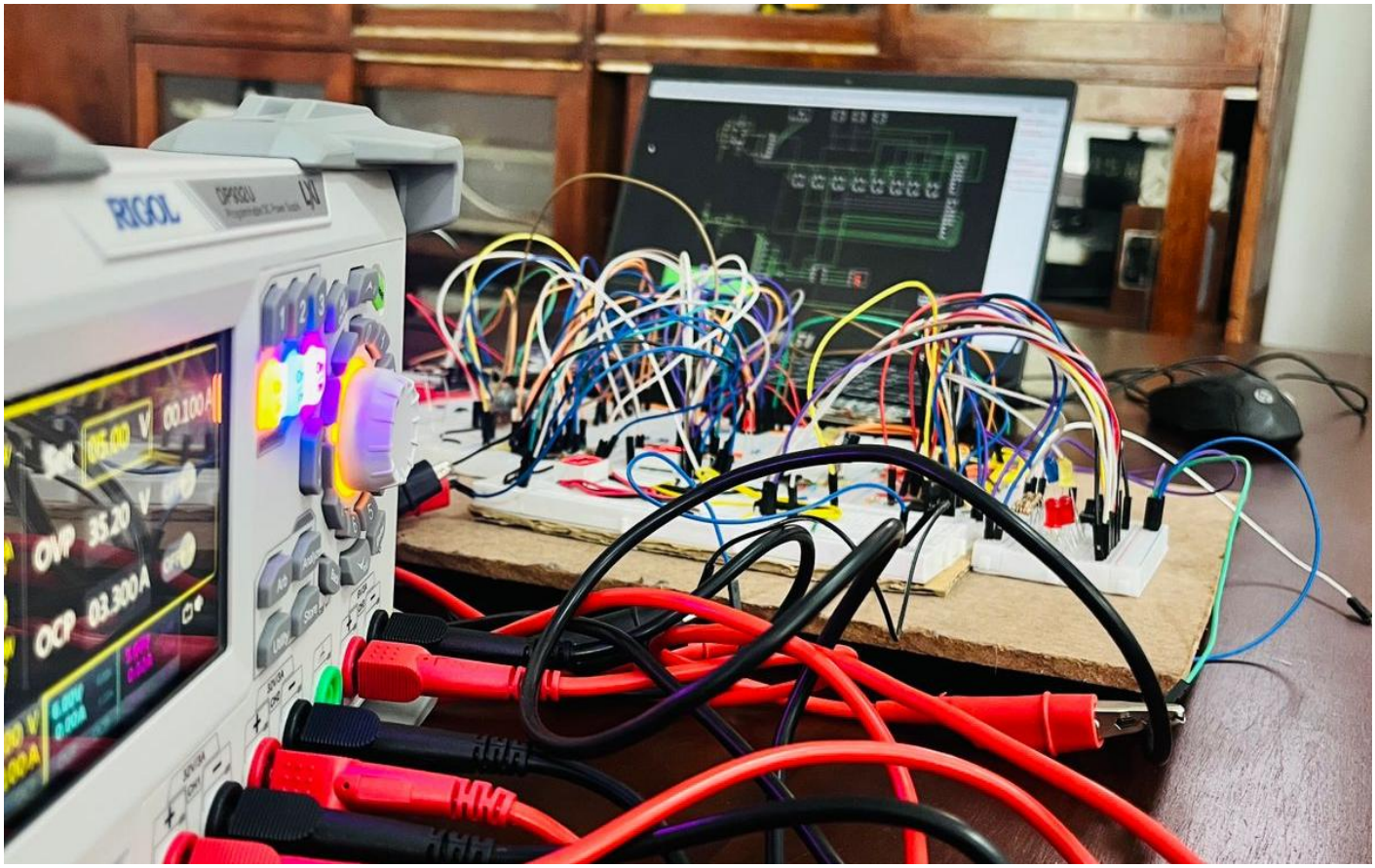
EE3024 - Digital Signal Processing

Project - Advanced Light Intensity Indicator (ALII)



Group 09 :

230409T - Mendis T.A.S.D.

230416L - Muhamath M.A.A.A.

230420T - Nanayakkara G.A.A.

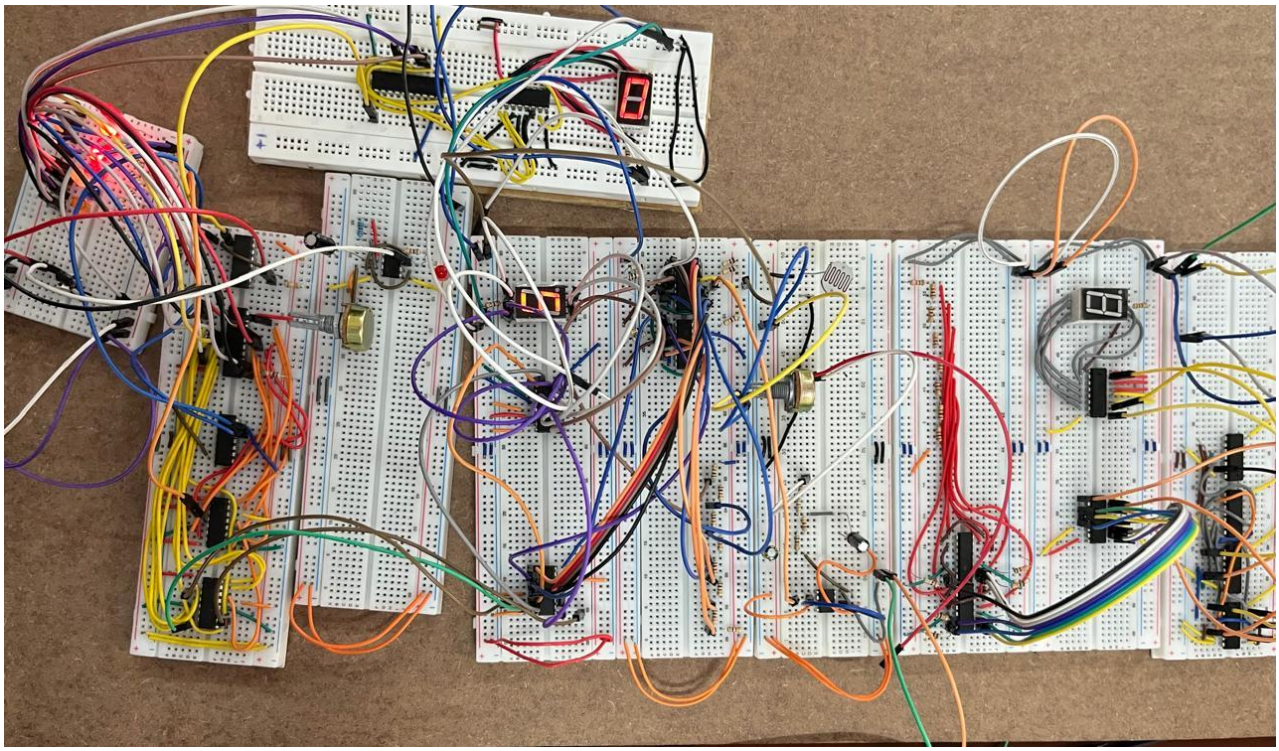
230426R - Nawarathna R.M.D.P.

230441H - Nilushika B.A.V.

The Advanced Light Intensity Indicator (ALII) is a smart monitoring module designed to provide accurate, real-time feedback on ambient light levels, the system allows for the efficient management of artificial lighting.

This report details the design and implementation of the ALII module, which is built entirely using discrete hardware components (Logic Gates, Op-Amps, and Timers) without the use of programmable microcontrollers. The system architecture is divided into four key stages:

1. Input Filtering: To eliminate electrical noise from the power line.
2. Analog-to-Digital Conversion: To quantize light intensity into levels "0" through "7".
3. Stability Control: To prevent display flickering caused by temporary light fluctuations.
4. Average Intensity Indicator: To calculate and display the long-term average light level over a 5 to 15-minute period.



1. Analog filtering stage

Objective: The primary objective of this stage is to eliminate 50–100 Hz electrical noise from the power line before the signal is processed. This ensures the 7-segment display remains stable and does not flicker.
While a Band Notch Filter can specifically target 50-100Hz noise, it is often to design in a practical scenario. Hence, We implemented a Second-Order Active Low-Pass Filter utilizing the Sallen-Key topology as a simplified alternative.

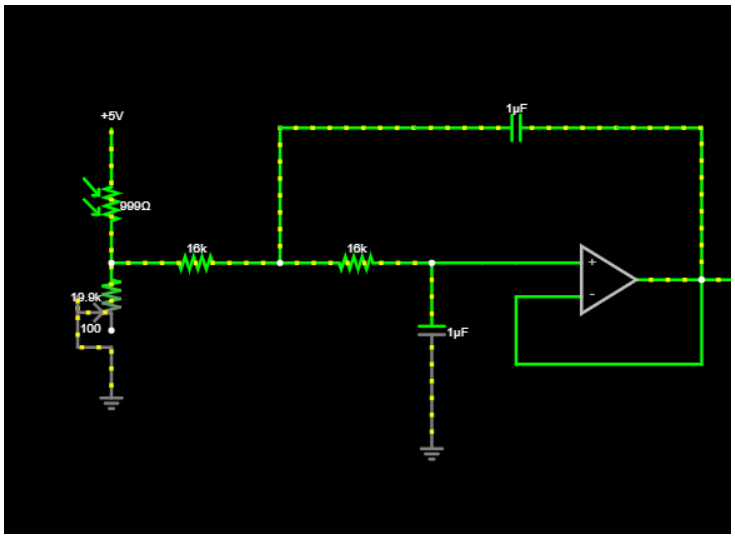


Figure 1 - 2nd order Sallen key filter

Why Sallen-Key?
This topology was chosen because it provides a steeper attenuation slope compared to a first-order passive filter. This allows for a sharper separation between the useful DC light signal and the unwanted high-frequency noise.

Component Selection:
The circuit utilizes a unity-gain configuration with R = 16 kΩ and C = 1 µF.

The cut-off frequency (f_c) for a unity gain Sallen Key low pass filter with equal component values is calculated as:

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

$$f_c = \frac{1}{2\pi(16000)(0.000001)}$$

$$f_c = 9.95\ Hz$$

The calculated cut-off frequency is approximately 10 Hz. The 50 Hz main noise is significantly higher than the cut-off frequency. Since this is a second-order filter, the noise is significantly attenuated (filtered out), resulting in a clean voltage signal for the comparators.

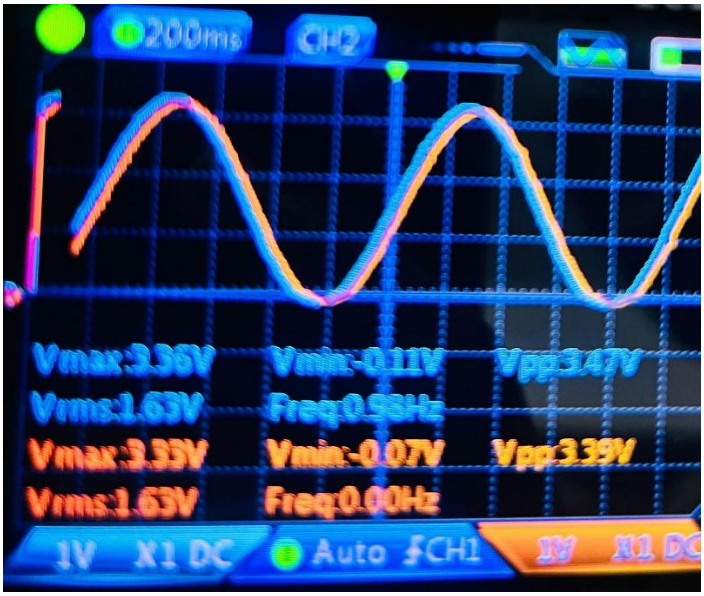


Figure 2 - filter output of 1Hz input signal

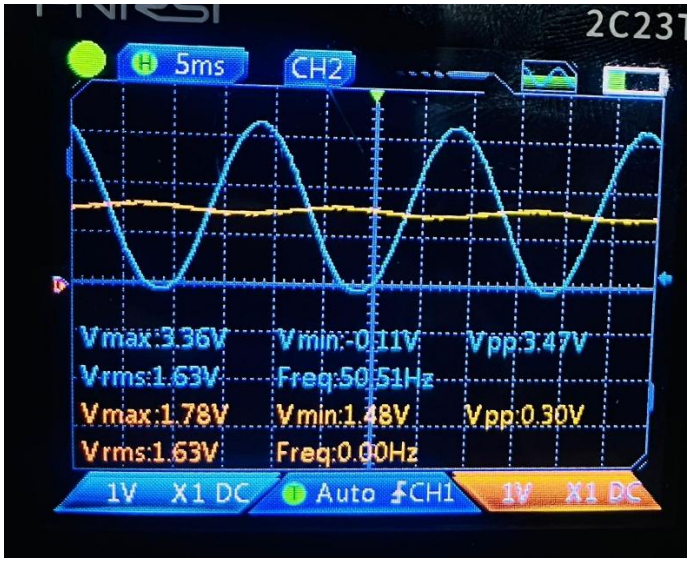


Figure 3 - filter output of 50Hz input signal

2. Analog-to-Digital Conversion (Flash ADC)

Objective: To satisfy the requirement of displaying light levels from "0" to "7" without using a pre-built ADC chip.

We designed a 3-bit Flash ADC using discrete components. This stage converts the continuous analog voltage from the Sallen-Key filter into discrete digital logic levels.

The Flash ADC consists of two main sub-circuits,
Resistor Ladder Network: A series chain of equal-value resistors acts as a voltage divider. This divides the supply voltage (Vcc) into 7 distinct reference voltage levels, creating equal steps for detection.
Comparator Bank: We utilized 7 Operational Amplifiers configured as comparators. Each comparator monitors the filtered LDR signal against one of the reference voltages from the ladder.

To ensure the display covers the full range of light intensity linearly, the step size (resolution) of the ADC was calculated

Supply Voltage (Vcc) = 5V Resistor Value (R) = 1kΩ

$$V_{step} = \frac{V_{CC}}{R_{total}} \times R$$

$$V_{step} = \frac{5V}{8R} \times R$$

$$V_{step} = 0.625V$$

So the system quantizes the light intensity into 0.625V intervals.
The direct output of this stage is a Thermometer Code.
For example, if the light level is 3, the bottom three comparators will be HIGH, and the top four will be LOW.
This intermediate code is then passed for encoding into binary and stabilization.
To display this as a decimal number, the signal must be converted into a standard 3-bit binary code We implemented a custom combinational logic circuit to achieve this without using a priority encoder chip.

The resulting 3-bit binary signal is fed into a BCD-to-7-Segment Decoder. This driver translates the binary code into the specific signals required to illuminate the correct number on the display.

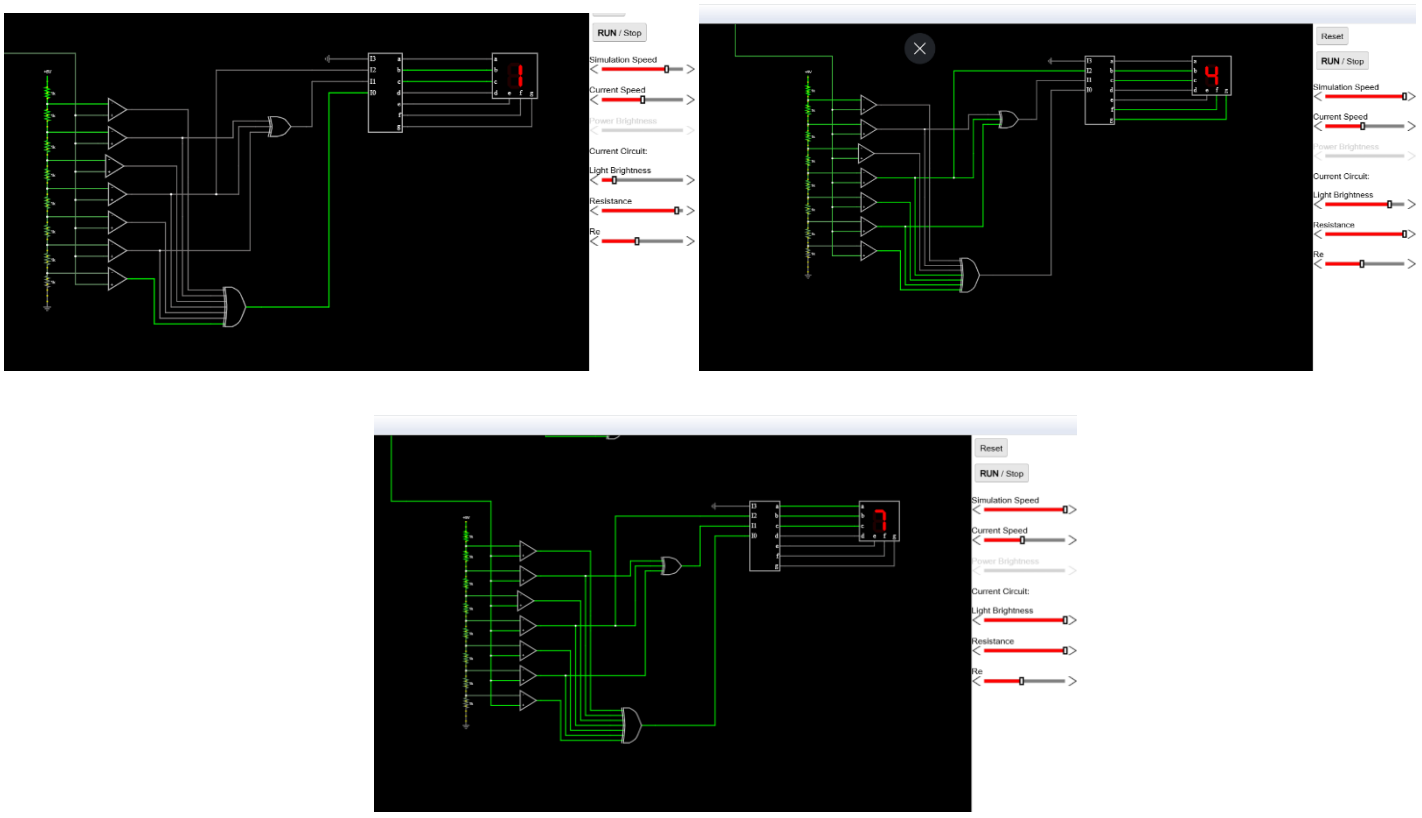


Figure 4 - ADC displaying the change of the light intensity

- We added pull-up resistors to the comparator outputs to ensure a strong and stable logic 'High' (5V) whenever a voltage threshold is crossed.

Without these resistors, the comparator outputs could 'float' in an undefined state, causing the subsequent digital logic to misread the signal. This addition guarantees good transitions between 0 and 1, preventing errors in the display.

- During practical implementation, we observed that the system displayed a maximum light level of '6' rather than '7'.

  This occurred because the LDR retains a small internal resistance even under very bright light, preventing the input voltage from reaching the highest reference threshold. This behavior is a common characteristic of real-world sensors compared to ideal simulations, resulting in a slightly reduced detection range.

3. Stability & Time Delay Feature

Objective : To prevent the display from flickering due to temporary light fluctuations (e.g., passing headlights or shadows), the system includes a "Stability Filter." This feature ensures the output only changes if the new light level stays for a set duration, adjustable between 30 and 300 seconds.

- The Change Detector

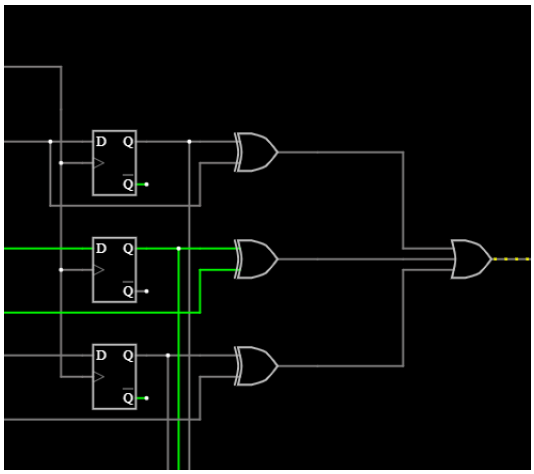


Figure 5 - Change Detection Logic

This circuit uses a feedback loop to actively detect changes before triggering the timer.

It uses three XOR gates to compare the current real-time signal from the ADC (Inputs) against the currently displayed value held in the D-Flip-Flops (Outputs Q).

Mechanism:
If the Input matches the Output (stable light), the XOR output is low.
If the light changes (input ≠ output), the corresponding XOR gate goes high.
The outputs of these XOR gates are summed by a 3-input OR gate. This creates a single "Change Detected" signal that initiates the timing sequence

- Timing Circuit

The "Change Detected" signal drives the charging of an RC network, which determines the delay duration.

Once triggered, the 555 output goes HIGH for a specific duration determined by the RC network connected to pins.
The falling edge of this timer pulse (after the delay finishes) acts as the clock signal for the D-Flip-Flops, latching the new data to the display.

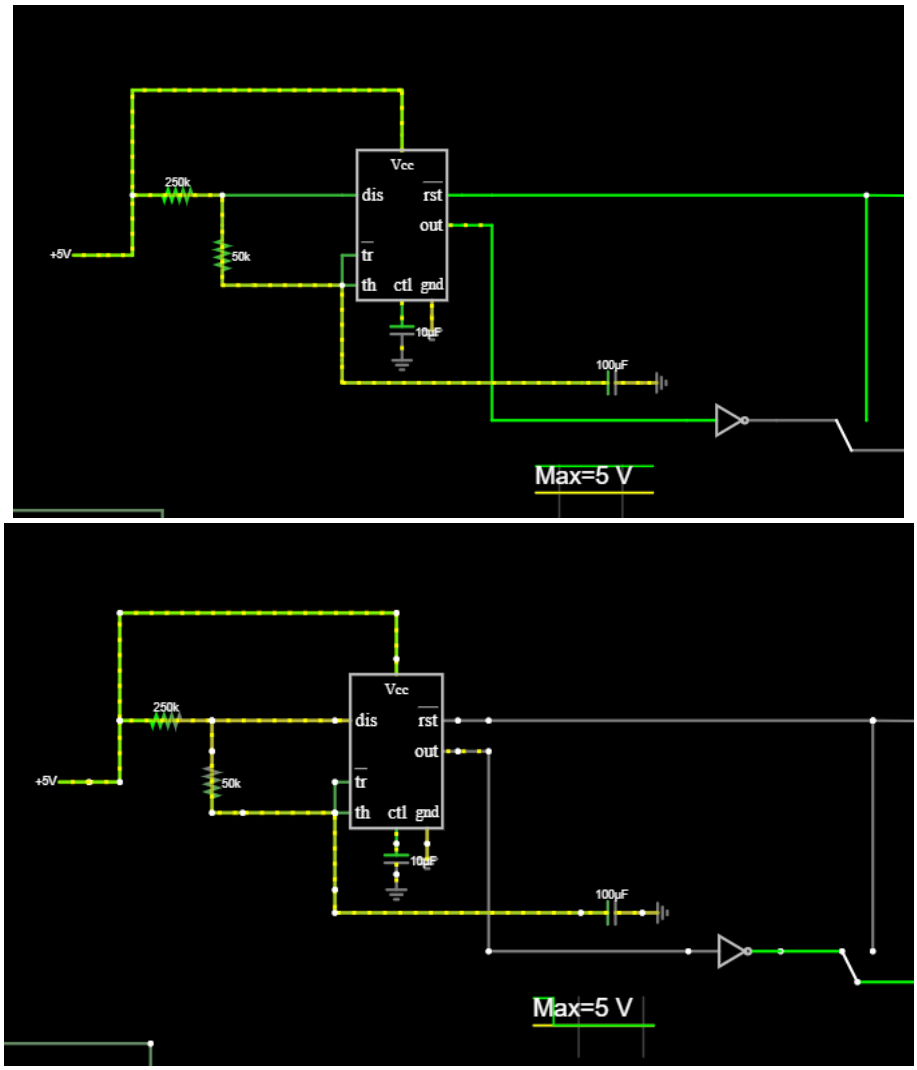


Figure 6 - 555 Monostable Timer Delay Circuit

- .Timing Design & Component Selection

To achieve the specific adjustable range of 30s –300s, we designed a series resistor network consisting of a Fixed Resistor and a Variable Resistor.

The duration of a 555 monostable pulse is given by the formula:

$$T = 1.1 \times R \times C$$

Capacitor Selection (C) : 100µF

Fixed Resistor (R_{fixed})= 250 kΩ (This resistor sets the minimum delay ensuring the system maintains a minimum stability window of approximately 30 seconds.)

$$T_{min} = 1.1 \times 250k\Omega \times 100\mu F$$

$$T_{min} = 27.5\ s$$

(In the actual circuit simulation, this value yielded an observed delay of 33 seconds, which safely satisfies the "minimum 30 seconds" requirement.)

To find the total resistance required for the 300 s,

$$300s = 1.1 \times (R_{pot} + 250k\Omega) \times 100\mu F$$

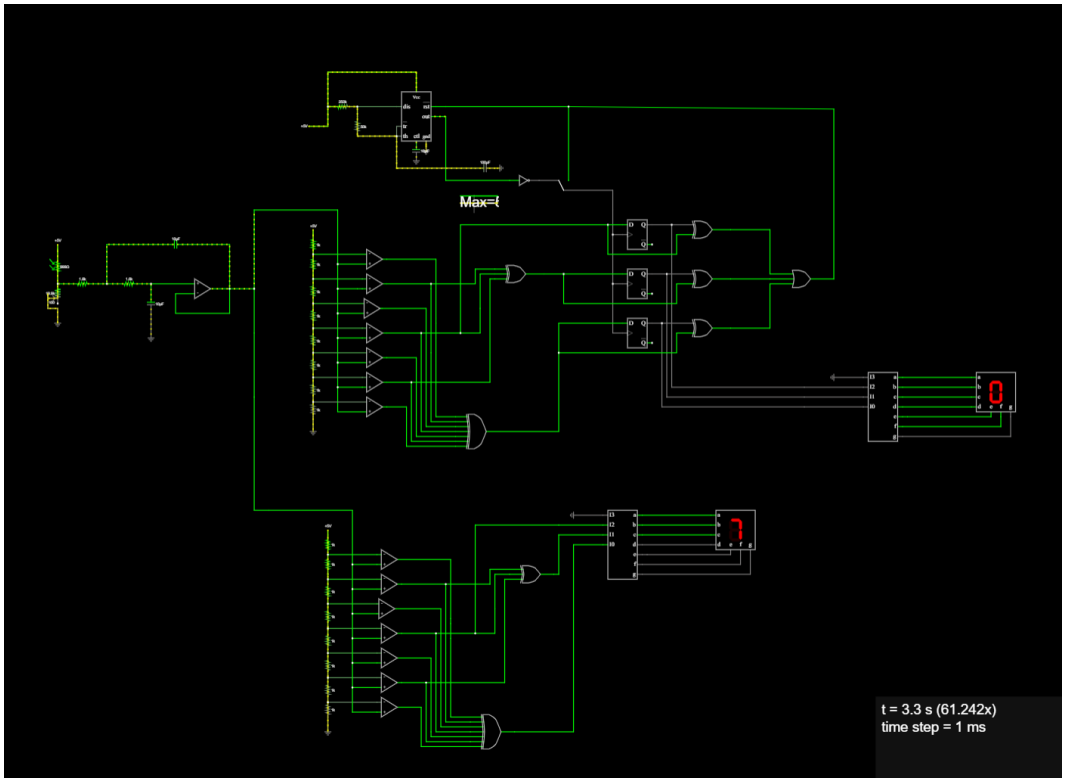
$$R_{pot} = 2.47M\Omega$$

To cover the full 30s to 300s range, the final design implements a 250kΩ fixed resistor in series with a 2.5 MΩ Potentiometer.

- Manual Override Switch

As shown in the circuit diagram, a switch is integrated at the output stage.

Function: This switch allows the user to bypass the 555 timer's output and connect the Flip-Flop clock line directly to the Trigger signal.



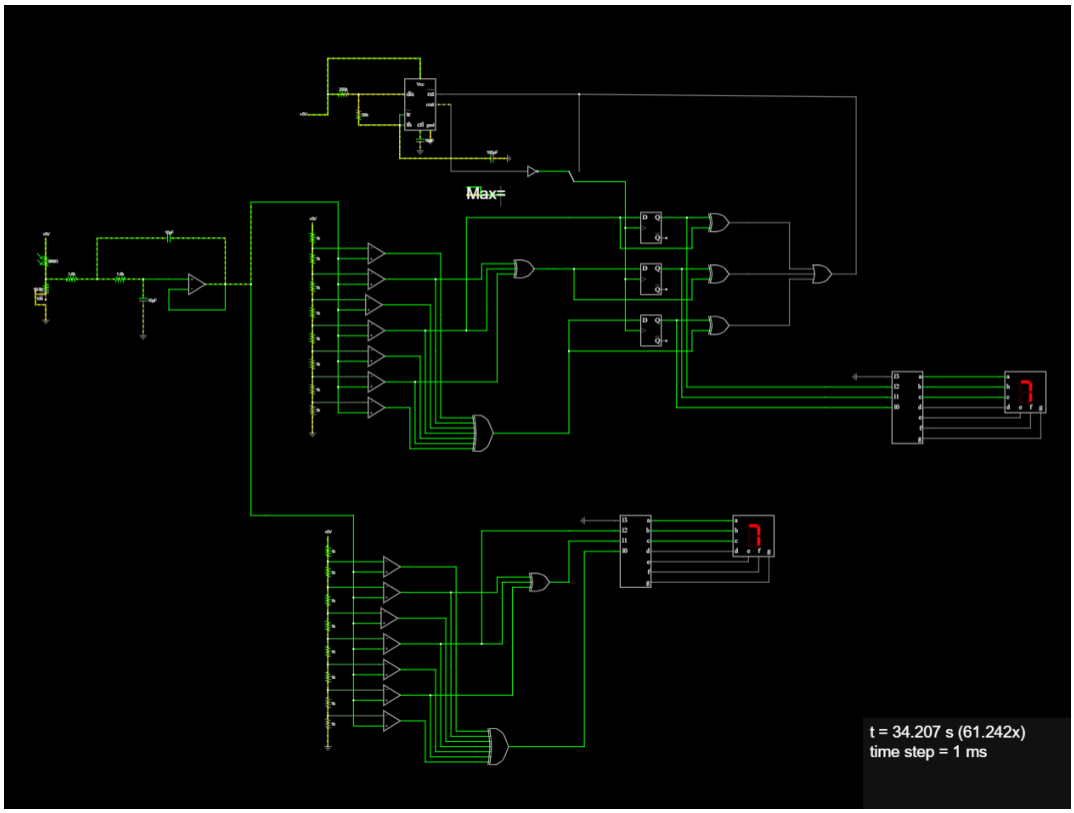


Figure 7 - completed stability circuit

Practical Design Optimizations :

- Dedicated Discharge Path

To ensure the timing circuit resets constantly after every cycle, we added a 50kΩ discharge resistor in parallel with the discharge pin of the 555 timer. Without this path, residual charge could remain in the timing capacitor, causing time delays to be shorter than calculated. The 50kΩ resistor forces the capacitor to discharge fully to 0V once the pulse ends, ensuring accurate and repeatable timing for the stability delay.

- To reduce the component count and avoid using an extra IC for a single inverter, we configured a spare NOR gate to function as a NOT gate.

By grounding one input of the NOR gate, and feeding the signal into the other allowed us to invert the 555 timer output.

4. Average Light Intensity

Objective :

To provide a stable, long-term system of measurement of the lighting environment, the system calculates the average light intensity over a period adjustable between 300 and 900 seconds. Instead of relying on capacitor charging curves (which can drift), this design utilizes a Digital Sampling and Accumulation method for high precision.

Circuit Topology:
The averaging engine is composed of four distinct stages:

1. The Sampling Clock: A 555 Timer configured in Astable mode, generating the base clock pulses.
2. The counter
3. The Arithmetic Unit: Two 4-bit Full Adders cascaded together to form an 8-bit adding.
4. The Memory Register: A bank of 8D-Flip-Flops.

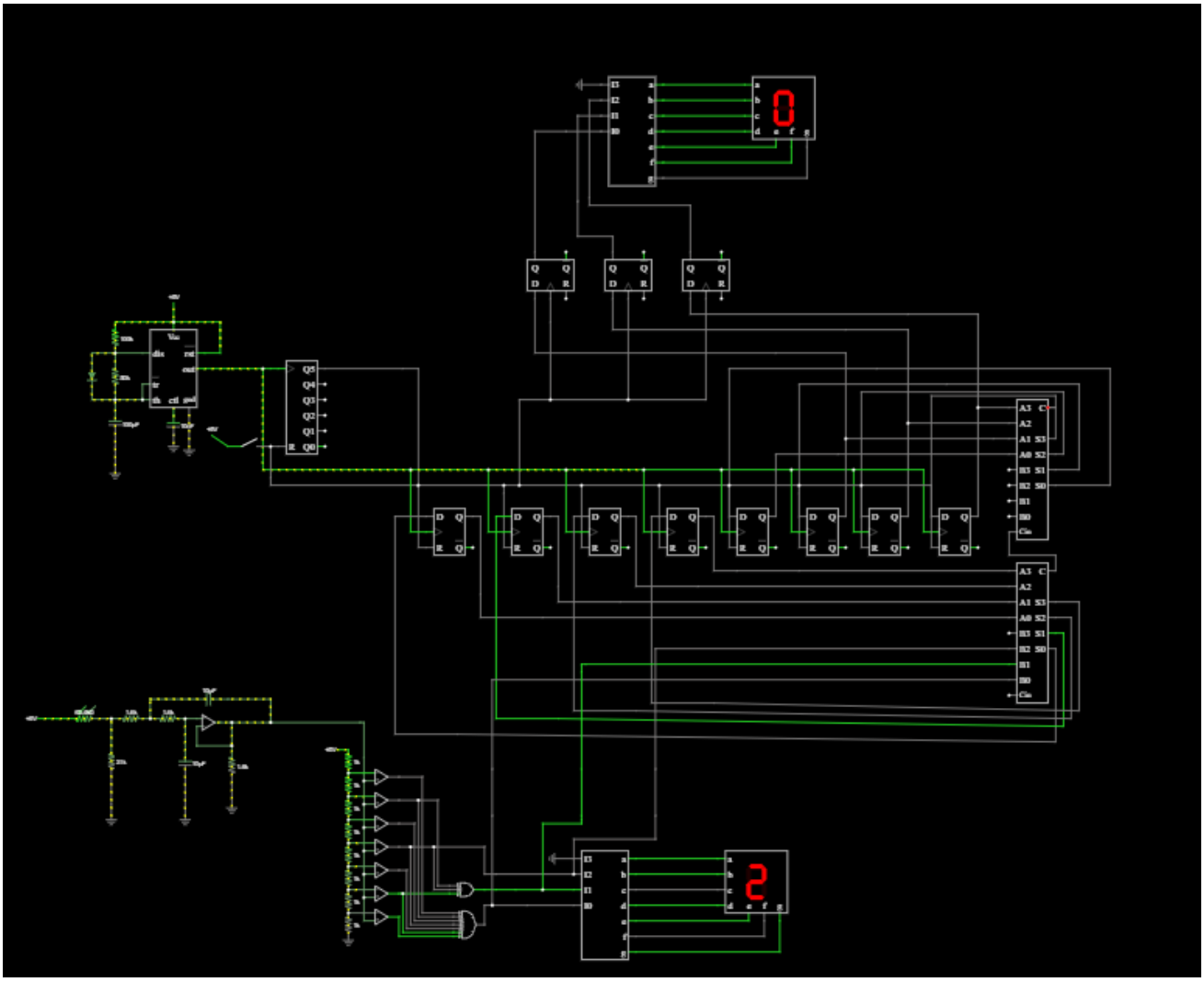


Figure 8 - completed average circuit

- Why 2 Adders and 8 Flip-Flops?

This specific component selection is driven by the mathematical requirements of the "Summing" operation.

The input light signal is a 3-bit number (Values 0–7). To calculate an average, we need to sum this number many times (e.g., 16 times).

Max possible single reading = 7

Max possible sum after 32 samples = 32 x 7 = 224

A single 4-bit adder can only count up to 15 (1111_2). If the sum exceeds 15, it overflows and resets, destroying the calculation. By cascading two 4-bit Full Adders (connecting the Carry-Out of the first to the Carry-In of the second), we create an 8-bit Adder

An 8-bit adder can count up to 255. Since our maximum expected sum is 224, the 8-bit capacity provided by the two adders is sufficient to handle the calculation without error.

The adder calculates *Current Input + Previous Sum*. However, the adder has no memory, it forgets the "Previous Sum" instantly. We need a component to store the result and feed it back into the adder for the next cycle.

Bit-Width**:** Since our maximum expected sum is 224, we need enough binary bits to represent the number 224

$2^8 = 226$ (Sufficient)

We implemented 8 D-Flip-Flops to act as a 8-bit Register. This register stores the cumulative sum (up to 224) and feeds it back to the adder inputs, allowing the system to accumulate the total value over time

- counter

The Role of the CD4040BE Counter: Instead of acting as a simple frequency divider, the CD4040BE Counter functions as a Cycle Controller to define the integration window.

Operation: The counter monitors the number of samples processed by the accumulator.

The Reset Cycle: It counts up to 32 clock pulses. Upon reaching the 32nd sample (Bit 5 / Q5 High), the counter triggers a feedback loop that sends a RESET signal to both itself and the 8-bit Accumulator Flip-Flops.

Purpose: This automatically clears the memory after every 32-sample batch, preventing register overflow and ensuring that each "Average Intensity" reading is calculated from a fresh, distinct time window of 5 to 15 minutes.

- The Averaging Logic (Bit-Shift Division)

To display the average, the accumulated sum must be divided by the number of samples

The system is designed to sum 32 samples. In binary, dividing by 32 (2^5) is achieved by ignoring the bottom 5 bits (Right Bit-Shift).

As shown in the schematic, the 7-segment display is connected only to the top 3 Flip-Flops of the register.

This effectively drops the lower 5 bits, performing an automatic mathematical division by 32. This converts the "Total Sum" back into the "Average Level" (0-7) for the display.

555 Timer: Controls the sampling rate. By adjusting the variable resistor on the 555 timer, we control how long it takes to collect the 16 samples, effectively adjusting the averaging window from 300s to 900s.

the 555 Timer operates in Astable Mode to function as a continuous clock oscillator. Consequently, the timing period is given by the Astable formula:

$$T = 0.693 \times (R_1 + 2R_2) \times C$$

This differs fundamentally from the Monostable formula utilized in the Stability Stage (Stage 3), where the timer functioned as a "One-Shot" delay. The coefficient 0.693 represents the capacitor oscillating between $\frac{1}{3}$ V_{cc} and $\frac{2}{3}$ V_{cc} whereas 1.1 (ln 3) represents a full charge from 0V to $\frac{2}{3}$ V_{cc} in the monostable configuration.

- Reset: A manual button clears all 8 Flip-Flops to zero, restarting the calculation.