Project Report for

Digital Design Assignment

BY

Names of the Students	*ID Numbers*
Akshay Valsaraj	***2018B1A70608G***
Dev Churiwala	***2018B1A70602G***
Pranav Ballaney	***2018B1A70635G***
Sonakshi Gupta	***2018B1A70614G***
Srijan Mahajan	***2018B1A70606G***
Taher Lilywala	***2018B1A70609G***

Prepared in partial fulfilment of the requirements of
Digital Design
CS F215

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
November 2020

Table of Contents

1. Problem Statement **3**

2. Top Level Block Diagram **3**
- 2.1. Basic Implementation 3
- 2.2. Circuit with Additional Functionality 4

3. Assumptions **5**

4. Notes on the Design Process **5**

5. State Diagram **5**

6. Diagram **6**
- 6.1. Circuit diagram 6
- 6.2. Pin-out diagram 6

7. Input/Output Combinations **7**

8. Going Beyond the Requirements **8**
- 8.1. Top-level Block Diagram 9
- 8.2. Circuit diagram 9
- 8.3. State Diagram 10

9. Bill of Materials **11**

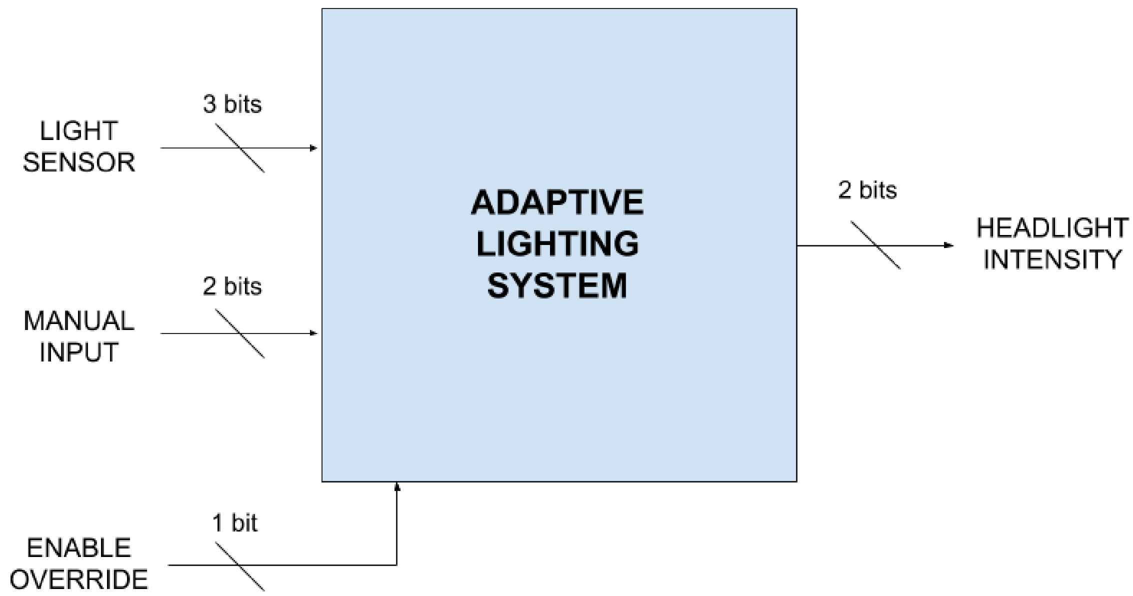
10. Appendix **12**
- 10.1. Datasheets 12
- 10.2. Simplification and K-Maps 12

1. Problem Statement

Design an adaptive lighting system for your automobile, whose headlights turn on automatically as the ambient light reduces. Similarly the reverse should happen at day break. Feel free to pick on how many intensity settings you want for the headlight. Assume that the light sensor measures light to dark in 8 levels.

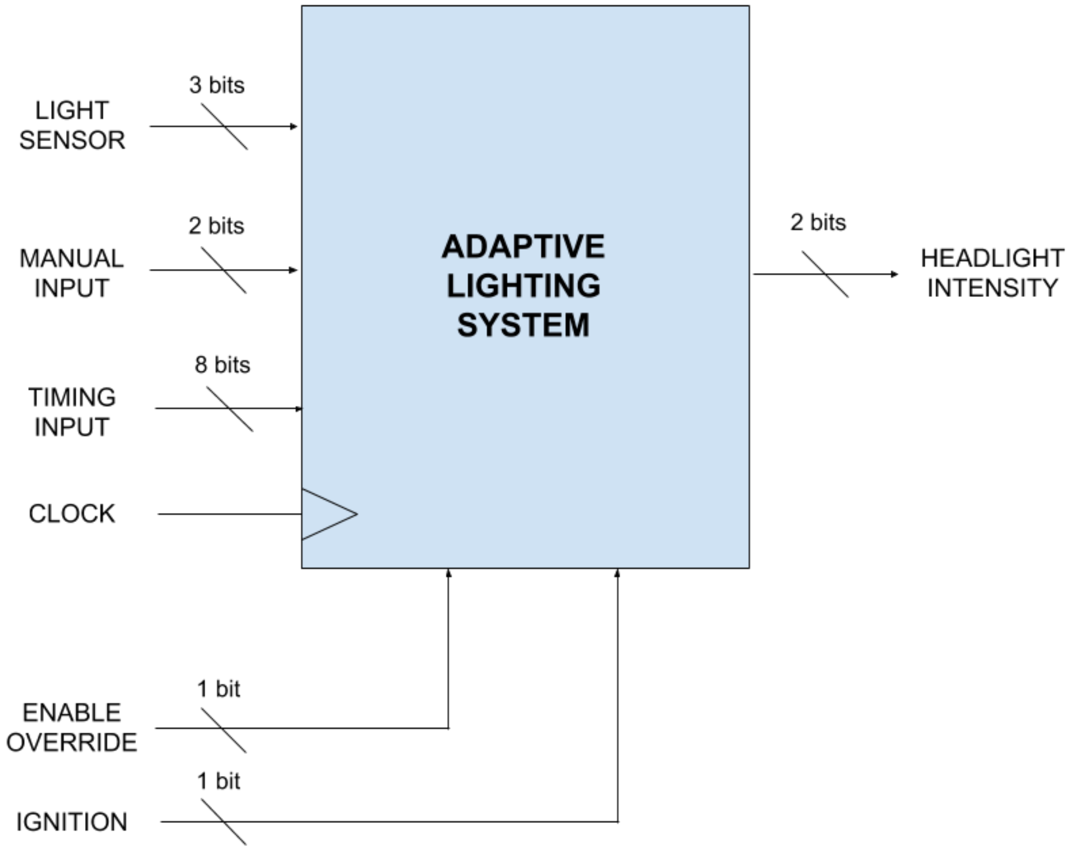
2. Top Level Block Diagram

2.1. Basic Implementation



The **light sensor** measures ambient light in eight levels (0 to 7) and that value is provided as an input to our adaptive lighting system. Our system outputs the adapted **headlight intensity** which has four levels (0 to 3) and depends on either the sensor input or manual input. The **manual input** can be used to control the headlight intensity only when the **override** signal is on, that is, it is set to 1.

2.2. Circuit with Additional Functionality



For additional functionality, a few more inputs are required. The **ignition** input is 1 when ignition is ON and 0 when it is OFF. The headlights will stay on for a few extra seconds after the ignition is turned off. This extra time is decided by the **timing input**. Since this is a synchronous design, an external **clock** signal is also required.

3. Assumptions

1. We are given a light sensor module which outputs the ambient light in 8 levels, with 0 denoting the darkest and 7 denoting the brightest ambient lights. We consider them to be 3 bit unsigned binary values that follow the Gray code from 000 to 100.
2. We assume that the output of the light sensor is never faulty and is an accurate representation of the ambient lighting conditions.
3. We assume that we have a headlight module which can be adjusted to 4 different intensities, with 0 denoting the OFF state and 3 representing the brightest state. We consider them to be 2-bit unsigned binary values that follow the Gray code from 00 to 10.
4. We have assumed that we have an external clock signal of 1Hz frequency.

4. Notes on the Design Process

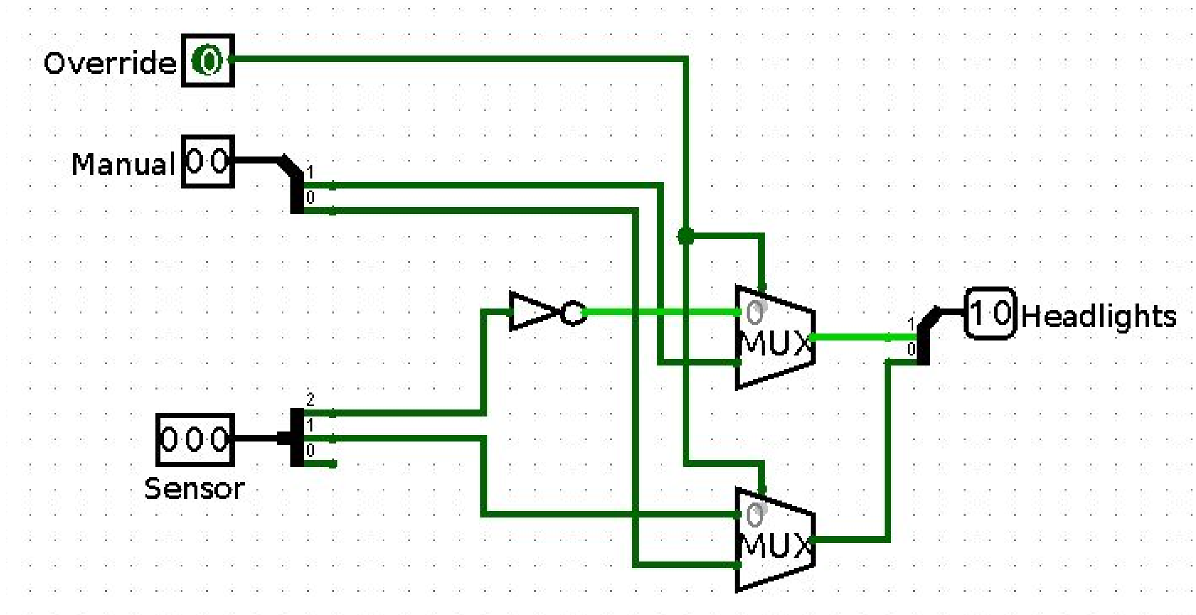
1. In case of a malfunction in the sensor, the system would be rendered useless. To get around this, we added a manual override, where the user would be able to manually control the intensity of lights.
2. While designing the circuit, we considered a sequential design, but we found that a combinatorial design was best suited for such a problem. A combinatorial circuit consumes a low amount of power, while fulfilling all requirements of the solution.
3. A potential difficulty in such a design would occur in case of sensor malfunction. For instance, if the sensor fluctuates between two values, the output would also fluctuate. This could damage the headlights. However, in our design, the input does not depend on the LSB of the input. This itself offers some level of protection against input fluctuations.
4. A more comprehensive mechanism to protect against fluctuations has been explored in an additional design, where we implement some additional features to go beyond the assignment's requirements.

5. State Diagram

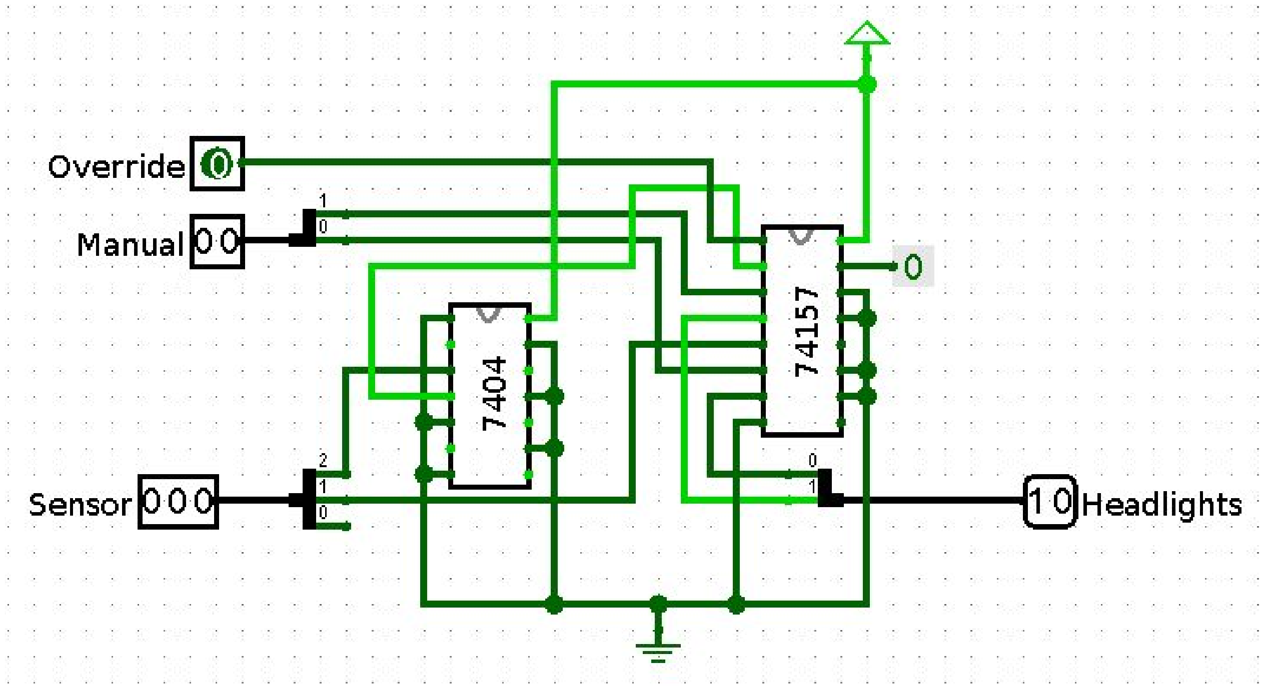
Since our primary circuit is combinatorial, it has no state diagram. A state diagram for the second circuit has been included along with the circuit, in Section 8.3

6. Diagram

6.1. Circuit diagram



6.2. Pin-out diagram



7. Input/Output Combinations

The primary circuit follows the following input/output table, where all inputs and outputs are assumed to follow the Gray code.

Override	Sensor			Manual Input		Headlights	
O	**S2**	**S1**	**S0**	**M1**	**M0**	**H1**	**H0**
0	0	0	x	x	x	1	0
0	0	1	x	x	x	1	1
0	1	0	x	x	x	0	0
0	1	1	x	x	x	0	1
1	x	x	x	0	0	0	0
1	x	x	x	0	1	0	1
1	x	x	x	1	0	1	0
1	x	x	x	1	1	1	1

8. Going Beyond the Requirements

In order to deliver beyond the requirements of the problem, we added the following features:

1. **Protection from fluctuating inputs:** The circuit decides the output based on the average of the previous four inputs. This confers protection to headlights in case the sensor exhibits oscillatory behaviour. Besides, this also offers an aesthetic feature that headlight intensity changes gradually.

   The implementation involves two shift registers to hold previous inputs and a block made of adders and NOT gates that effectively averages the inputs. Here as well, we ignore the LSB of the input, so only two shift registers are required, and the addition is simplified. The states have been coded in such a way that just two NOT gates are required to calculate the effective average of the previous states from their sum. This implementation follows the regular binary code instead of Gray code to make implementation easier, since addition is involved. An input/output table and K maps have been attached in the appendix.

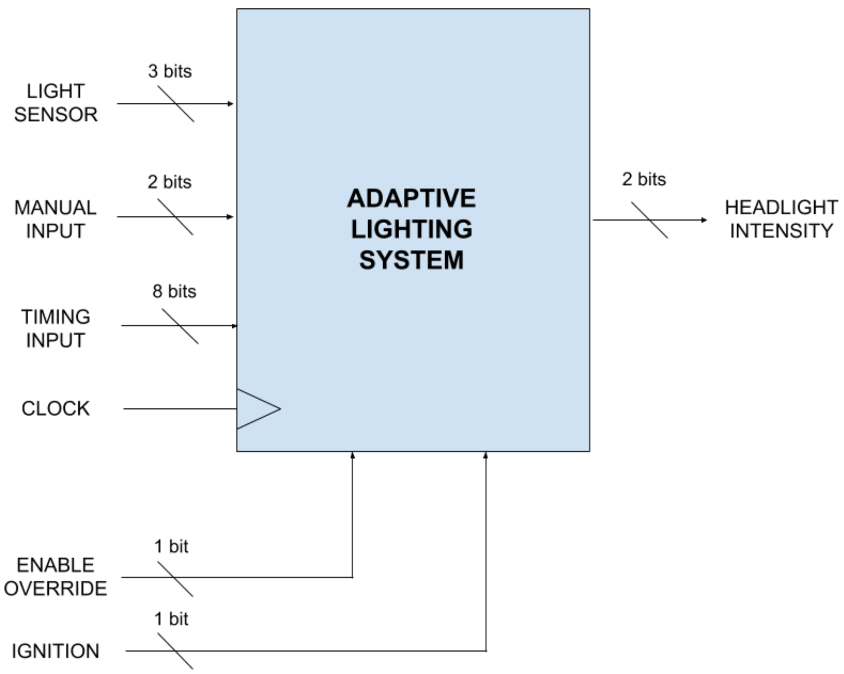
2. **Lights stay on to allow the passengers to exit safely:** After the ignition is turned off, the circuit counts down from a user-specified value. During this time, the automatic headlight system remains functional to allow the passengers to navigate in case the car has been parked in the dark and avoid potholes, puddles and the like.

   We have assumed an additional 1-bit ignition input, along with a 8-bit duration control. This allows the user to set the amount of time from 0 to 255 seconds (around 4 and a half minutes), as per convenience.

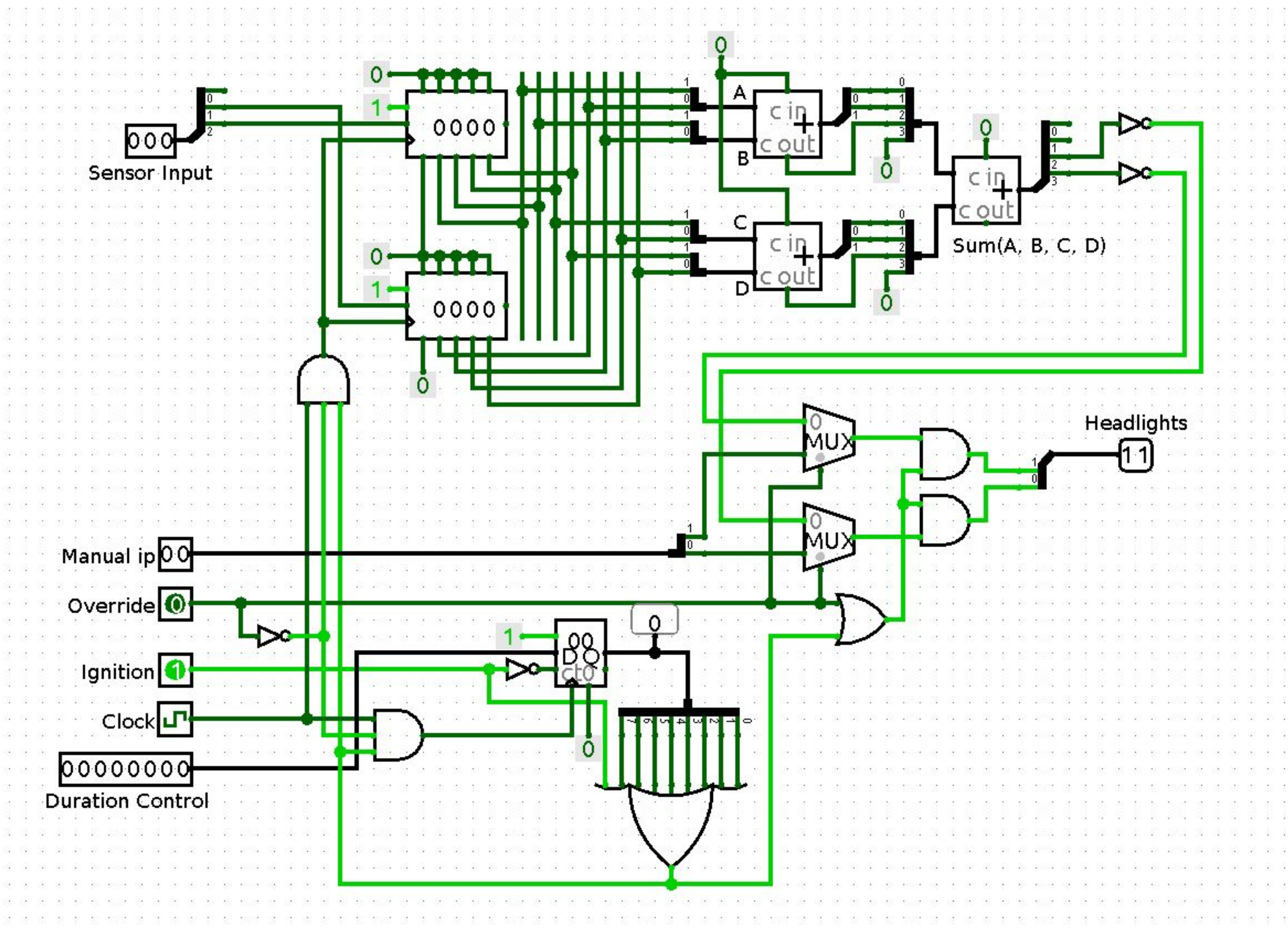
3. **Manual override:** A manual override in case the sensor malfunctions. This bypasses the exit feature as well, and the lights maintain a manually specified state as long as they are supplied with power.

We have also taken care to shut off the components that are not required in a particular state to save power. For example, when the override is ON, power to all registers and counters is turned off.

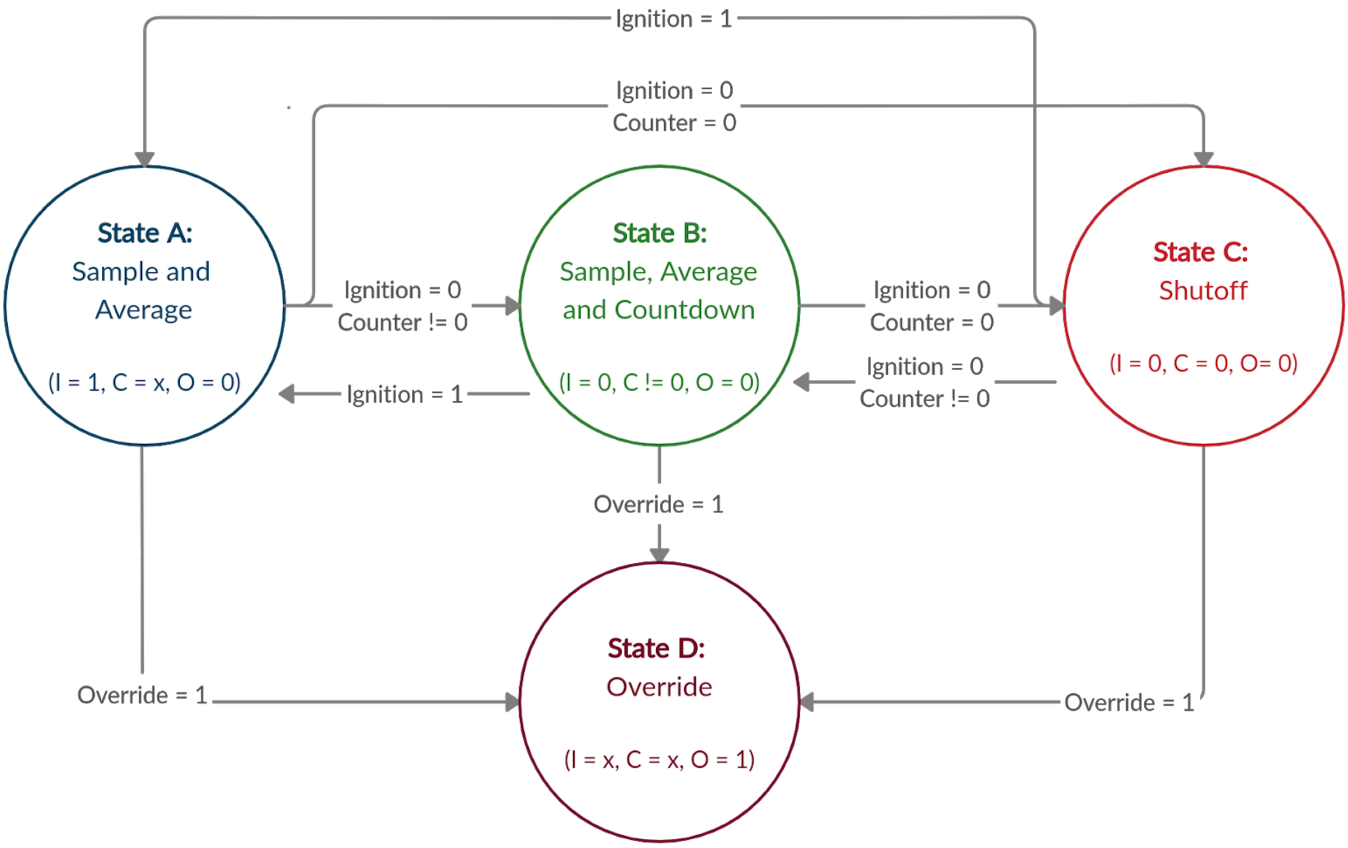
8.1. Top-level Block Diagram



8.2. Circuit diagram



8.3. State Diagram



Since the circuit had so many inputs, the number of states was too high to make a complete diagram. A simplified diagram is shown above.

The circuit has four major "states":

1. **Sample and Average:** In this state, the circuit samples the sensor at clock edge and averages the previous four sensor inputs to determine the intensity of the headlights, The duration specified by the user is loaded into the counter.

2. **Sample, Average and Countdown:** Turning off the ignition sends the circuit into this state. Here, the circuit starts counting down from the user specified value, but still keeps sampling the sensor.

3. **Shutoff:** This state is reached when the ignition is off and the counter has reached 0 after counting down from the user specified value. The sensor is no longer sampled and the counter is stopped to maintain this state. The circuit can get out of this state if the ignition is turned back on, or if the override is turned on.

4. **Override:** From any state, if the override is turned ON, the circuit stops sampling the input, along with the counter. The manual input is directly sent as an output to the headlights.

9. Bill of Materials

IC number	Description	Manufacturer	Price
7404	Hex inverter NOT gate IC	Texas Instruments	Rs. 18
74157	Quad 2 i/p Multiplexer IC	Texas Instruments	Rs. 20

These are the ICs required for the basic circuit implementation. Datasheets for these ICs have been included in the appendix.

Additional ICs required to go beyond the requirements :

- 74LS95: 4-bit bidirectional universal shift registers
- 74LS83: 4-bit binary adder IC; has 2-bit functionality
- SN74ALS869: Synchronous 8-bit up/down counter
- SN74LVC1G08: Single 2 i/p AND gate IC
- SN74LVC1G32: Single 2 i/p OR gate IC

10. Appendix

10.1. Datasheets

1. Datasheet for 7404:
   https://www.electronicscomp.com/datasheet/74ls04-ic-datasheet.pdf

2. Datasheet for 74157:
   https://www.electronicscomp.com/datasheet/74hc157-ic-datasheet.pdf

3. Datasheet for 74LS95:
   https://www.electronicscomp.com/datasheet/74ls95-ic-datasheet.pdf

4. Datasheet for 74LS83:
   https://datasheetspdf.com/pdf-file/375717/Motorola/74LS83/1

5. Datasheet for SN74ALS869:
   https://www.ti.com/lit/ds/sdas115c/sdas115c.pdf?ts=1605943231348

6. Datasheet for 74LS11:
   https://www.ti.com/lit/ds/symlink/sn74lvc1g08-q1.pdf?ts=1605953898624

7. Datasheet for SN74LVC1G32:
   https://www.ti.com/lit/ds/symlink/sn74lvc1g3208-q1.pdf?ts=1605970872182
   &ref_url=https%253A%252F%252Fwww.google.com%252F

10.2. Simplification and K-Maps

The following table was used to average the last four inputs. S3, S2, S1, S0 represent the sum of the last four inputs with S0 being the LSB. X and Y are outputs to the headlights, with X being the MSB. The inputs and outputs here follow the regular binary code.

Sum				Output		
S3	**S2**	**S1**	**S0**	**X**	**Y**	**Decimal**
0	0	0	x	1	1	3
0	0	1	x	1	1	3
0	1	0	x	1	0	2
0	1	1	x	1	0	2
1	0	0	x	0	1	1
1	0	1	x	0	1	1
1	1	0	x	0	0	0
1	1	1	x	x	x	x

K-map for X:

S3 / S2 S1	00	01	11	10
0	*1*	*1*	*1*	*1*
1	0	0	x	0

X = S3’

K-map for Y:

S3 / S2 S1	00	01	11	10
0	*1*	*1*	0	0
1	*1*	*1*	x	0

Y = S2’