****

**School of Electrical Engineering and Computer Sciences, NUST**

**Department of Computer Science**

**EE-221 Digital Logic Design**

**Course Instructor:** Dr. Nasir Mahmood

**Course/Section:** BSCS6-B

**Group Members**

|  |  |
| --- | --- |
| **Name** | **Reg No.** |
| Sania Saeed | 177074 |
| Sana Naz | 197631 |
| Maliha Sameen | 174057 |
| Maham Nadeem | 189424 |
| Hunain Hussain Khan | 177089 |

# MODULE 3:

# OVERVIEW:

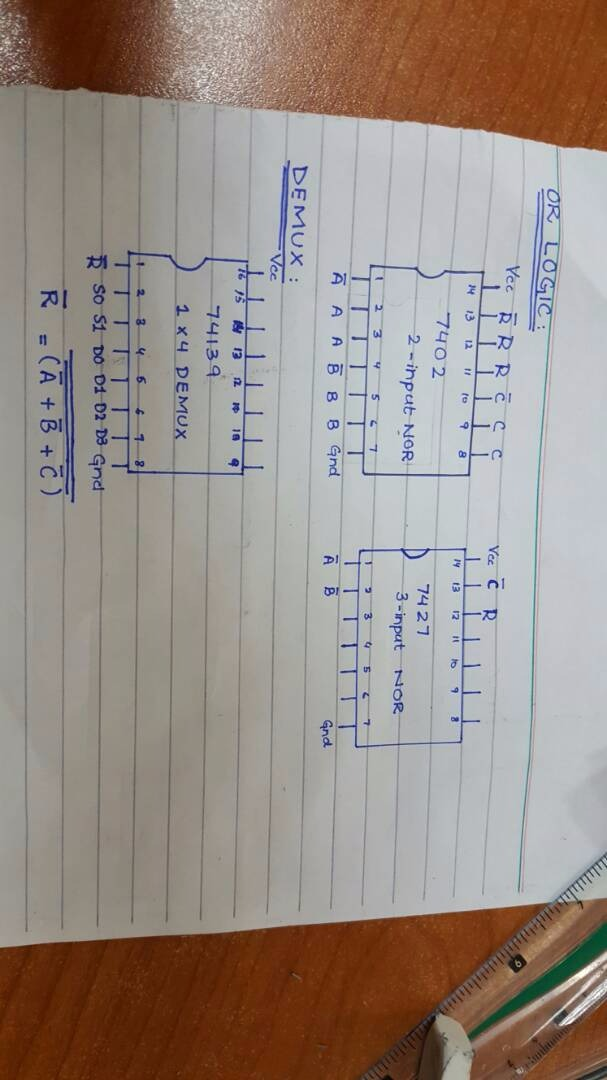
Water is a necessity for plants; they need a supply of water at different times and in different amount. Our watering system is based on probes which will be installed in the soil. We would install three probes at three different locations as we are implementing our system for three types of plant. These probes will give a logic zero if the water in the soil is below a desired level, otherwise a logic high would be the output of the probe. The location of the probes would be used, in order to identify the type of plants which are to be watered as there are four types of plants at four different locations. Each type of plant is to be watered for different time interval and at different timings and that is also considered by us. There are three locations hence there are three valve each valve is responsible for watering plants which are located at different places. The input from the probe, location of the probe and timing would select the appropriate valve which is to be switched on.

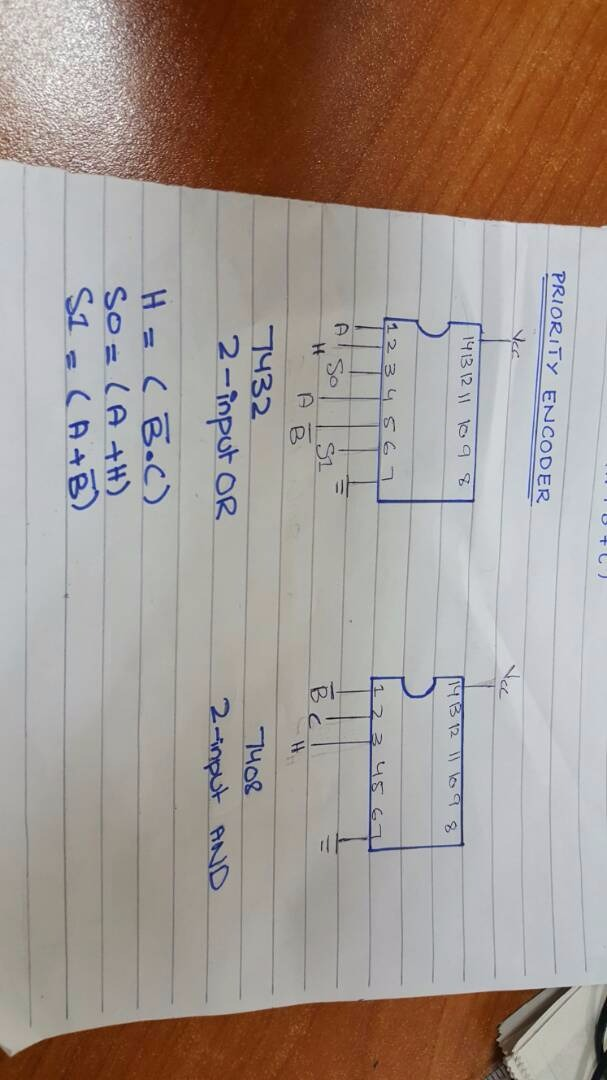
We have implemented a mealy machine, As the next state is dependent on the present state and the current input. In this circuit, we are using three probes located at three different locations, which contain three different types of plants. These probes will give a logic zero if the water in the soil is below a desired level, otherwise a logic high would be the output of the probe. Input from these probes will be inverted by the use of a **NOR** gate and then the inputs from all three probes will go through two (three-input) **NOR** gate which will be active if at least one of the location needs water in such case the probe will give a zero input, which will be inverted into a ONE logic. The output form the two **NOR** gates, will go through a **1X4 DEMUX (74153).** At the select lines of the DEMUX the location of the probe will be provided which would be taken from the output of the internal circuit of a priority encoder. The output from the demux will trigger the **555 timer ic** the output of this ic will be inverted and will be **OR** with the input from the water sensor which will indicate whether the water has reached the location or not, the output from this **OR** will be given as the clock to the D flip flop. The flip flop would allow the water pump to run for the set time.

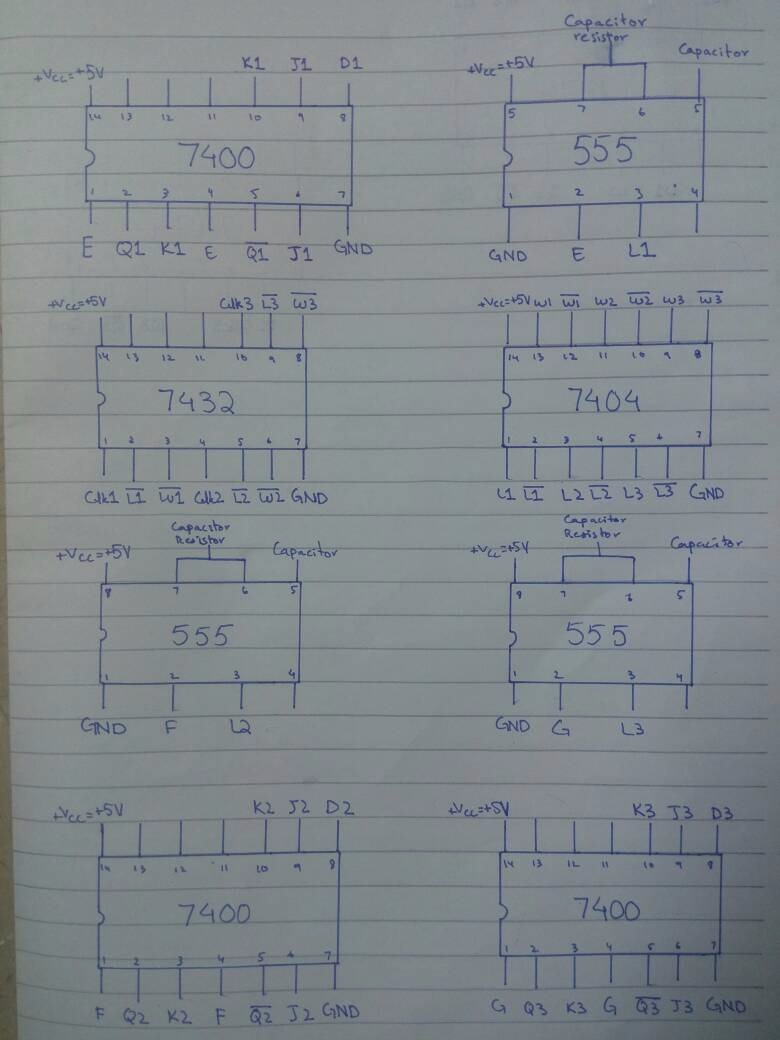
# 

Figure 1: Logic Diagram

# Schematics:







# 

# COMPONENTS:

The components used in this module are as follows:

* The OR LOGIC
* 4 x 2 Priority Encoder
* 1 x 4 Demultiplexer
* NE 555 Timer

## The OR Logic:

As in this project we’re catering three types of plants i.e. Hydrophilic, Hydrophobic and Mesophytes so, we’re implementing this OR logic through a 2-input NOR gate IC and a 3-input NOR gate IC. The input A is a probe for Hydrophilic plants, B for Mesophytes and C for Hydrophobic plants. Probes give Logic 1 when the water is full and Logic 0 when the water is empty. The equation for this OR Logic becomes

R’ = A’ + B’ + C’

R’ is the output we get which tells us that either of the three types of plants needs water. The output R’ then serves as an input to the Demultiplexer.

***Truth Table:***

|  |  |  |  |
| --- | --- | --- | --- |
| A’ | B’ | C’ | R’ |
| **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** |
| **0** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** |

## 4 x 2 Priority Encoder:

Priority Encoder gives the location of the minterm that is high. We used 4 x 2 Priority Encoder in this module to find out the location of the plant that needs water. We’re catering three types of plants and Hydrophilic plants get the higher priority. The minterm D0 goes to GND and C’ goes to D1 minterm, B’ goes to D2 minterm and A’ goes to D3 minterm. Two output lines Y0 and Y1 comes from the priority encoder and gives the binary value of the minterm that is high. These output lines serves as select lines for the Demultiplexer.

***Truth Table:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C’ | B’ | A’ | S0 | S1 |
| 0 | 0 | 0 | Never Occur | Never Occur |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

## 1 x 4 Demultiplexer:

Demultiplexer is a device that takes a single input line and sends it to one of the output lines based on the select lines input. We used demultiplexer to send R’ (output of OR logic) to one of the output lines based on the location of the plant taken from the priority encoder.

***Truth Table:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S0 | S1 | A | B | C |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

## NE 555 Timer:

The output from the **DEMUX** is given to the mono-stable multi –vibrator. This ic generates a pulse for a fixed time. For the actual circuit we need the time to be 15 minutes, 10 minutes and 5 minutes but for the sake of representation we have selected the time to be 1 minute, 11 seconds and 110 seconds.

SR-Flip Flop is present in the 555 ic:

**Characteristic table:**

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Q | Q` |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |

The formula:

**T = 1.1\*R\*C**

Where, C = 100 µF

60 = 1.1\*R\*100 µF

R = 550 kΩ

Second:

11 = 1.1\*R\*100 µF

R = 100 kΩ

Third:

110 = 1.1\*R\*100 µF

R = 1000 kΩ

D – Flip flop:

The **“D flip flop”** store and output whatever logic level is applied to its data terminal so long as the clock input is HIGH. Once the clock input goes LOW then it will not change state and store whatever data was present on its output before the clock transition occurred.

**D D FLIP-FLOP**

**D**  **Q**

**CLK** **Q**`

The input from the DEMUX (E) will go through a combination circuit, thus the input at the D flip flop will be:

**D = IQ + IQ’**

The clock will be the sum of the inverted output of the 555 timer ic and the inverted input from the water sensor which is at the location of the plants.

Characteristic table:

|  |  |
| --- | --- |
| D | Q(n+1) |
| 0 | 0 |
| 1 | 1 |

STATE TABLE:

State table will be same for all the 3 outputs from the DEMUX because same operation is being performed.

|  |  |  |
| --- | --- | --- |
| Qt (present state) | A(Input) | Q(t+1)(Next state) |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**0**

**1**

**0**

**1**

**State 1**

State Diagram:

State table will be same for all the 3 outputs from the DEMUX because same operation is being performed.

**State 1**

**State2**

**01**

**00**

**01**

**11**

**10**

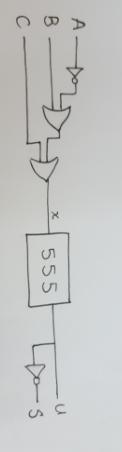
**00**

**11**

**10**

**State2**

Theoretical Design of Automatic Plant Shade and UV Light System

In the shelter control, firstly three sensors are to detect the three conditions (windy, rainy and not sunny) if one of these conditions is true then the shelter is to be closed and the UV lights are to be lit up to let the plants have all the proper wavelength of light they need. All of this system is to be controlled by the NE555 timer which has a clock of 12 hours, which means the shelter is to be opened for half a day and closed in the night. For NE555 timer capacitors and resistors are used to control the clock.

Block Diagram:

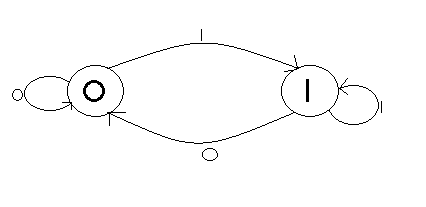
Stepwise Circuit Formation:

The six steps to solve any problem of sequential design are given as follows:

Step#1:

**Understanding the problem**, the main problem was to control the shelter upon plants to let them have all the light they need with considerations to some problems, which are named as conditions.

Step#2:

Making a **state diagram**, let the UV lights are on at 1 and off at 0. So, if the input from the conditions is 0 the UV lights are to be switched off (opened shelter) and if the input is 1 the UV lights are to be switched on (closed shelter).

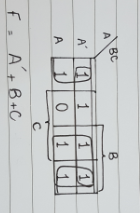
Step#3:

The **State Table**, before that the input is according to the conditions as follows and next is the state table is as follows:

A= light sensor ,B=water sensor ,C=wind sensor

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | x |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Present State | Next State | |
| x | Q(t) | Q(t+1) | |
| 0 | 0 | 0 | |
| 0 | 1 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 1 |

Step#4:

The **K-Map** for the input is as follows:

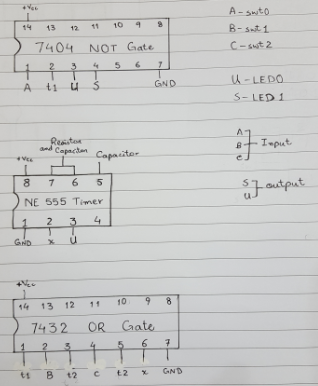
Step#5:

The **Expressions** for the inputs and outputs is:

**x=A`+B+C**

UV lights = **U = Q(t)**

Shelter = **S = Q`(t)**



Schematic Diagram:

Theoretical Design of Water Control System

A plant sanctuary consists of two tanks; an overhead tank and an underground tank. After an appropriate level is reached in the overhead tank the motor has to be turned on. For this purpose we will install a water sensor in the overhead tank at a level below which we don’t want the water level to decrease. At this point a 0 logic signal will be produced which will pass through the **NAND gate.** Another water sensor will be installed in the underground tank below which if the water level falls then the water pump won’t be able to work properly because there won’t be sufficient water for it to operate. Now, both the logic will pass through an **AND** gate then through a flip flop which will allow the water tank to stay on for 15 minutes. So basically the inputs from the underground tank and the overhead tank are used to activate the water pump. The clock of the flip flop will be given by inverting the output from the **555 timer ic** and then taking its **OR** with the input from the third water sensor which is located at water entering pipe in the overhead tank.

**Step 1 Water Sensor:**

Water Sensor is an electronic device that is designed to detect the presence of water. There are three water sensors in circuit. The first one is being installed in overhead tank which will detect presence of water in overhead tank while second sensor is being installed in rooftop tank which will detect presence of water in rooftop tank.

The last sensor is being installed in the pipe connecting the rooftop tank and the overhead tank due to which clock of D flip-flop goes LOW if water passes through it.

**Timer 555:**

The output from the combinational circuit is given to the mono-stable multi –vibrator. This ic generates a pulse for a fixed time. For the actual circuit we need the time to be 15 minutes but for the sake of representation we have selected the time to be 1 minute.

SR-Flip Flop is present in the 555 ic:

**Characteristic table:**

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Q | Q` |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |

The formula:

**T = 1.1\*R\*C**

Where, C = 100 µF

60 = 1.1\*R\*100 µF

R = 550 kΩ

**Step 2: STATE DIAGRAM:**

**0**

**10**

**11**

**00**

**01**

**11**

**10**

**00**

**01**

**State2**

**State 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Present State | Input | Input | Next State |
| Q (t) | A | B | Q (t+1) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**Step 3 State Table:**

**Step 4 D FLIP-FLOP**

The **“D flip flop”** store and output whatever logic level is applied to its data terminal so long as the clock input is HIGH. Once the clock input goes LOW then it will not change state and store whatever data was present on its output before the clock transition occurred.

**D**  **Q**

**D D FLIP-FLOP**

**CLK** **Q**`

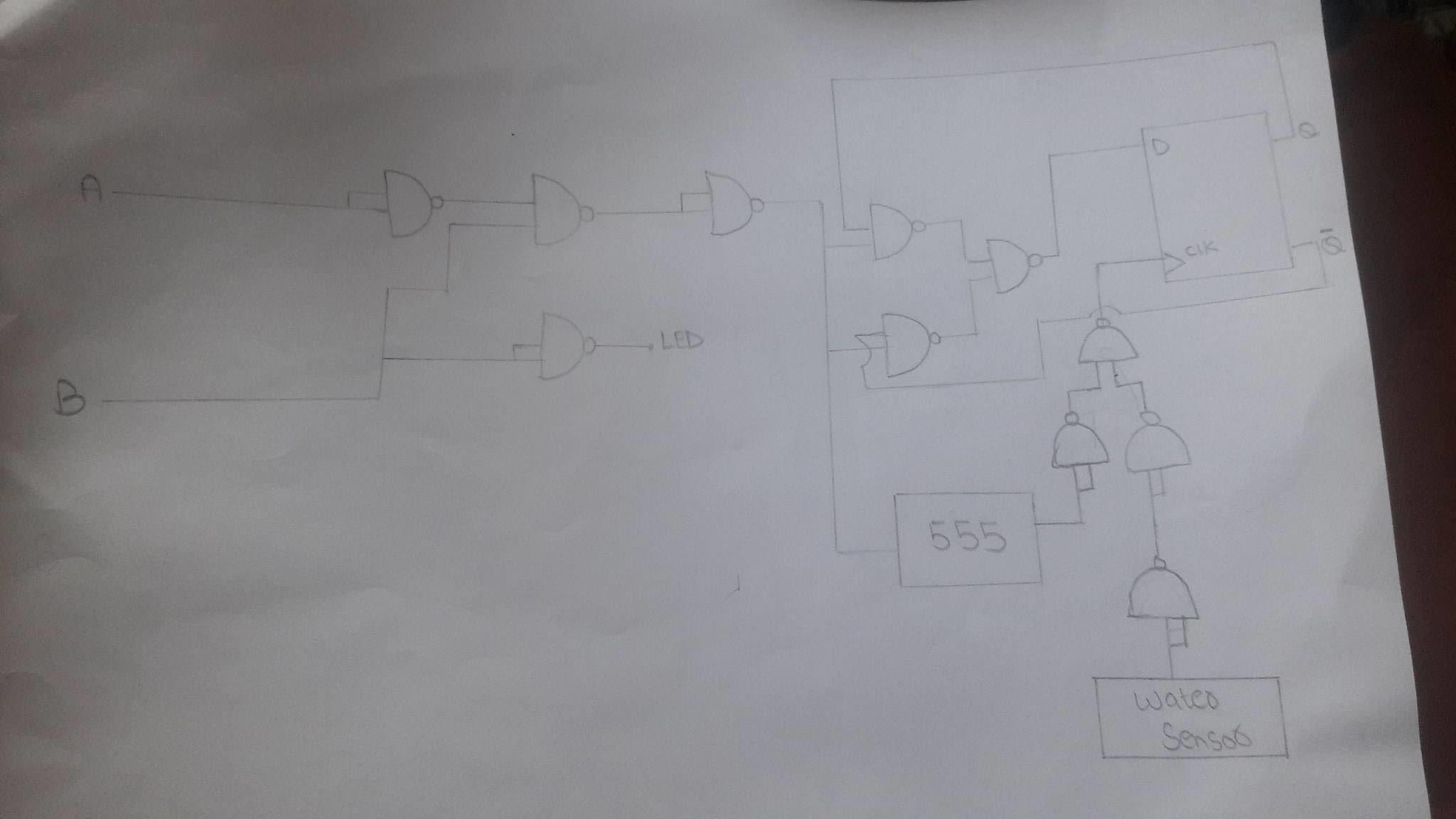
In the circuit, D flip flop will turn the motor on and off depending on the input at its data terminal.

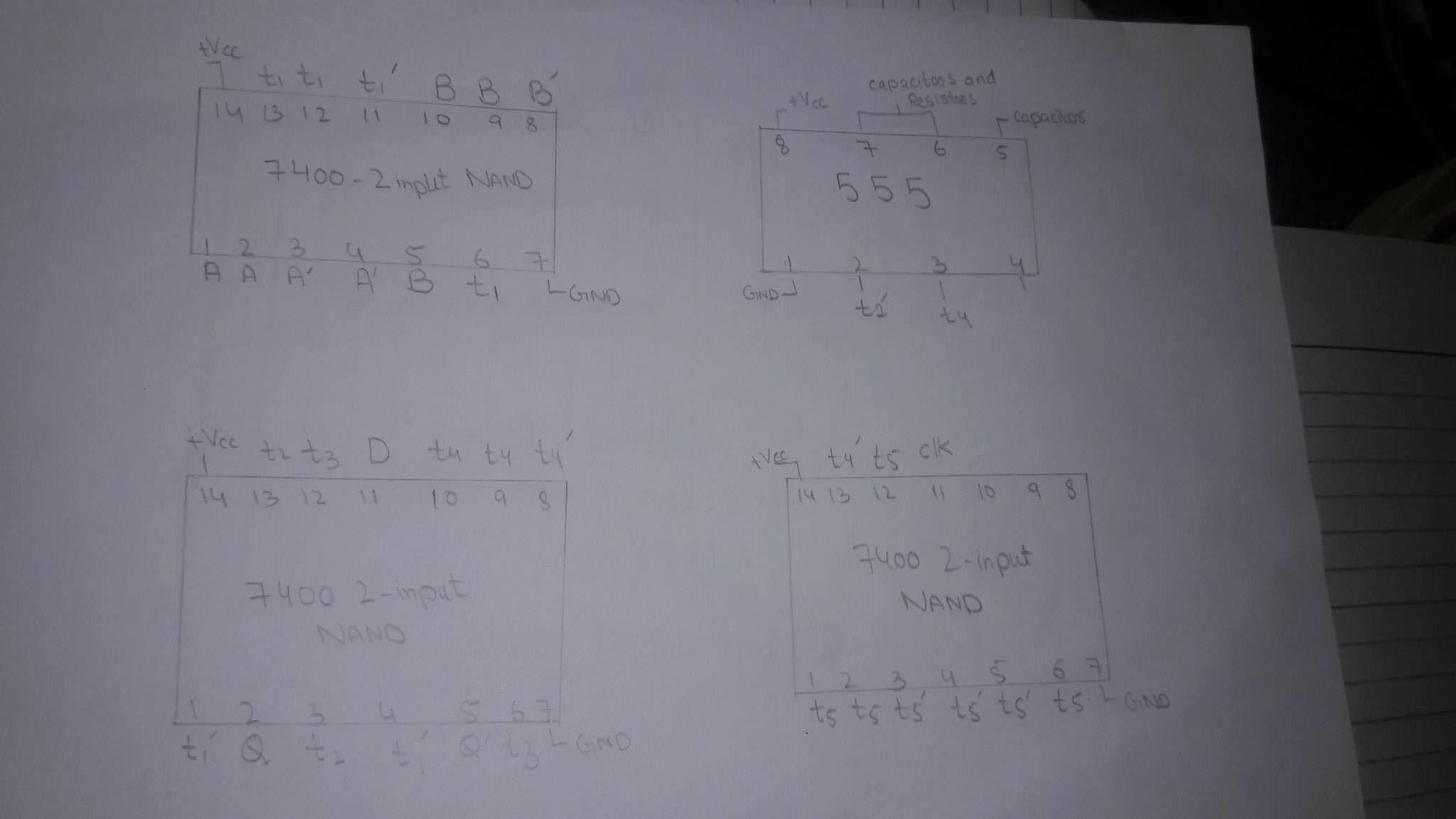
**Characteristic table of D Flip-Flop:**

|  |  |  |  |
| --- | --- | --- | --- |
| D | Q(t) | Q(t+!) | Operation |
| 0 | 0 | 0 | Reset |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | 1 | Set |

**Step 5 State Equation:**

A`(t) B(t) Q(t) + A`(t) B(t) Q`(t) = A`B Q + A`B Q` = A`B

**Logic Diagram:**

**Hardware Schematic:**

