CHIP DESIGN FOR HIGH SCHOOL PROGRAM

Name: Bikash Kumar Mishra

B.Tech. Final Year

Branch: Biomedical Engineering

WEEK-1 SECTION-1

Energy is the capacity to do work. There are two types of energy:-
(i) Potential Energy:- Storage
(ii) Kinetic Energy:- Movement

Example:

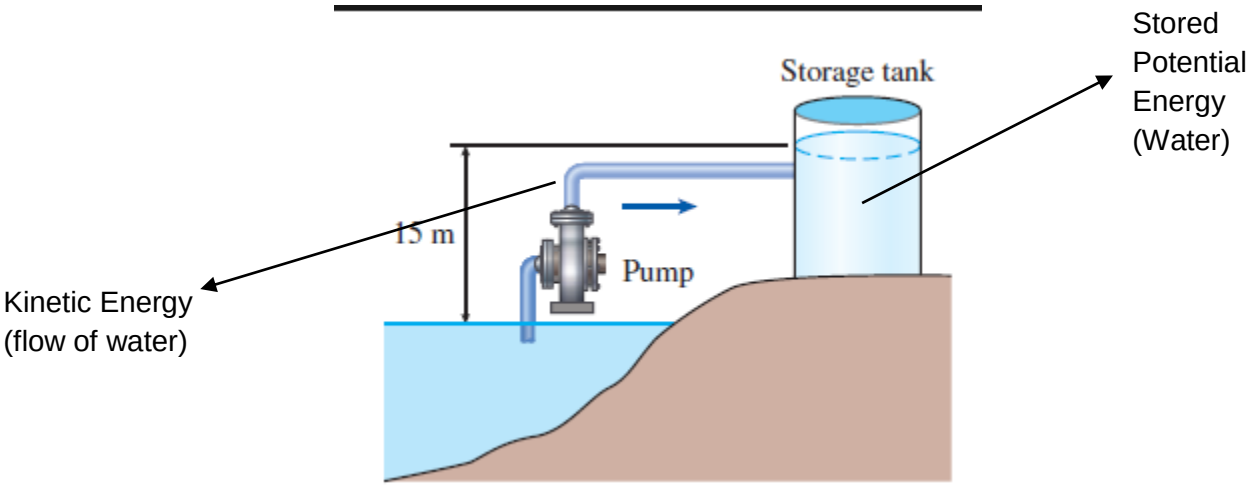


Fig. Stored Water and movement of water

Stored Water is potential Energy whereas the flow of water through the pump is kinetic energy.

Domain of Energy	Potential Energy	Kinetic Energy
Mechanical Energy	Water stored in a tank	Flowing of water from tank
Heat Energy	Heat stored inside an object	Flow of heat from hot body to cold body
Electrical Energy	Stored charges in a battery	Flow of charges (Current)

Cooking Analogy(Potential and Kinetic Energy)

Stored heat energy to cook food(Potential Energy)

Flow of heat from stove(Kinetic Energy)

The heat energy is flowing from stove to the pan is kinetic energy whereas the accumulation of heat energy in the pan for cooking is Potential Energy.

Mobile Charging Analogy(Potential and Kinetic Energy)

Storage of charges in mobile(Potential Energy)

Flow of charges from socket through charging wire(Kinetic Energy)

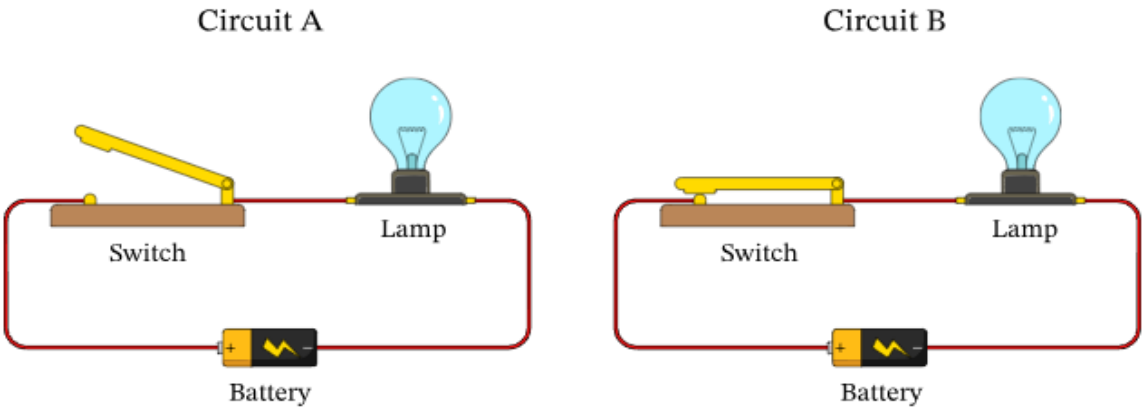
The charges are flowing from socket to the mobile is kinetic energy whereas the accumulation of charges in the mobile is Potential Energy.

In electrical parameters, Potential energy is Voltage and Unit is Volts. Kinetic energy is Current and Unit is Amperes.

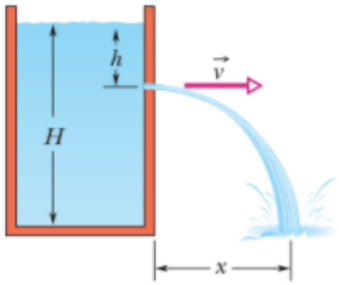
Tap to control flow

Stored Water in tank

Whether the tap is open or close, the tank has stored potential energy in form of water. But, there should be a path to flow of energy, which can be done by opening the tap which results in flow of water as form of kinetic energy. This tap in electrical parameter to allow flow of current.



The opened switch does not allow flow of current although battery has potential energy in circuit-A whereas the closed switch allow the current to flow in the circuit-B as there is path to flow.

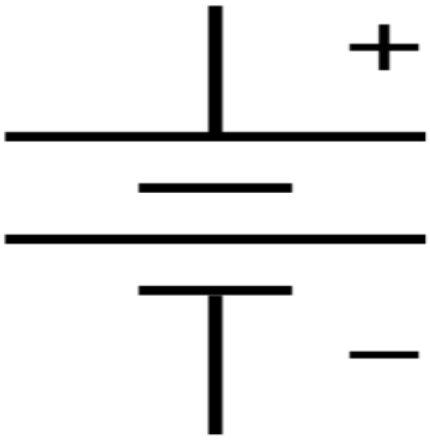


Water will flow from higher height(higher potential energy) to lower height (lower potential energy),i.e., there should be a potential enrgy difference.

Conclusion

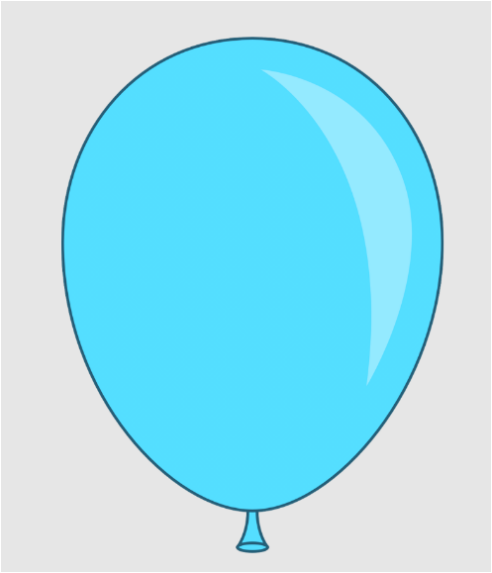
(1) Energy will flow if there is a path.
(2) Energy will flow from higher potential to lower potential energy.

WEEK-2 SECTION-1

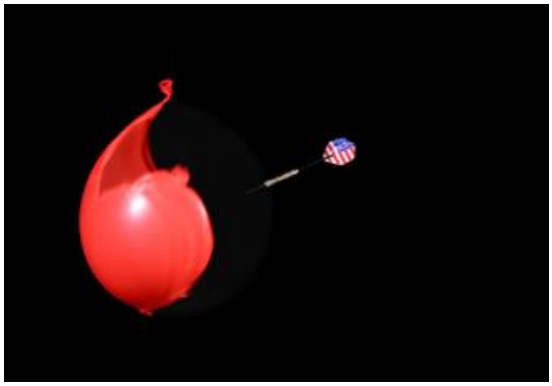


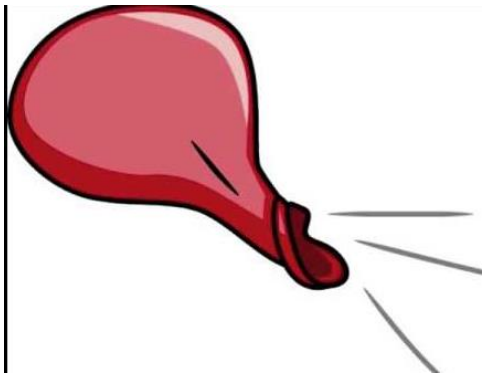
(Battery)

Stored charge inside battery is potential energy(Voltage) and flow of charge is Kinetic energy(Current).



Balloon is blown by air. The air inside balloon is stored as potential energy. In order to allow flow of air from balloon we have two ways:- (i)Abrupt flow, (ii) Controlled flow



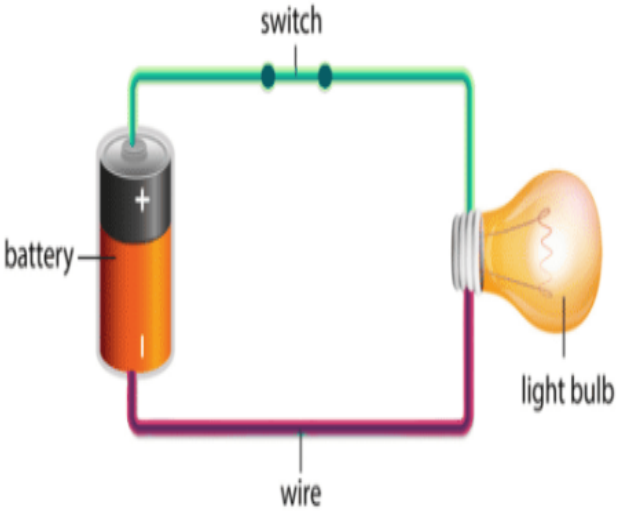


If we poke the balloon, the balloon will burst and air flow in a controller manner.

If balloon air is released from its mouth, it flows in controlled manner.

In both the cases air is flowing out, but in the bursting case air flow is abrupt due to which the balloon is damaged. However, if the balloon air is released from its mouth, then it flows in a controlled manner and does not allow balloon to damaged. If we replace the balloon with battery, then following thing happens:

If there will be abrupt release of stored charges, then battery will be damaged. If the stored charges release gradually in controlled manner, then battery will not be damaged.



Abrupt flow of energy(Battery damage)

Controlled flow of charge due to load(Bulb)

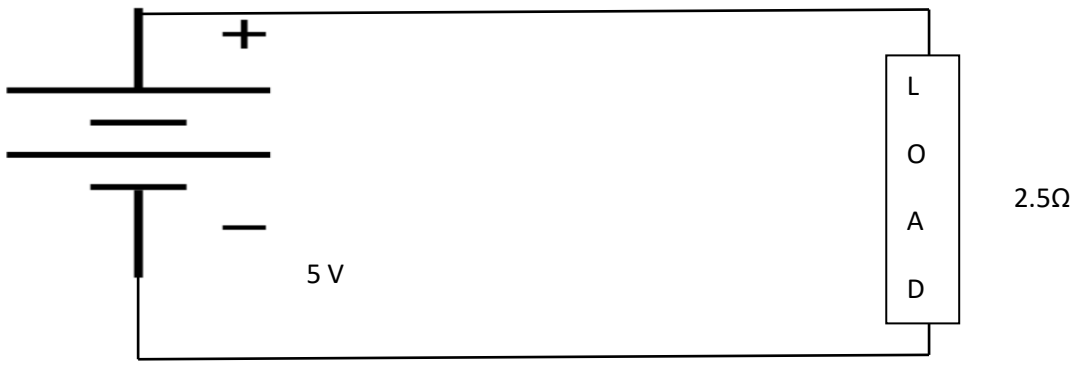
Load will limit the flow of current and allow controlled flow of current and does not damage the battery.

Ohm's Law:

"The voltage is directly proportional to current flow through a load."

V = I*R

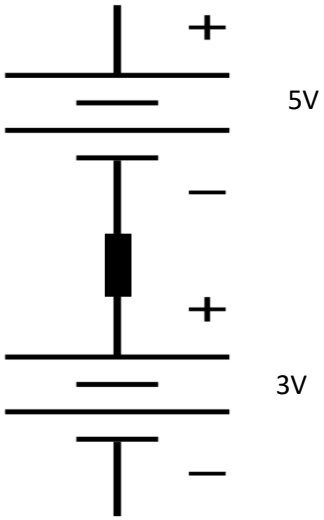
If R is small, for given V I will be large. Unit of resistance is ohms(Ω).



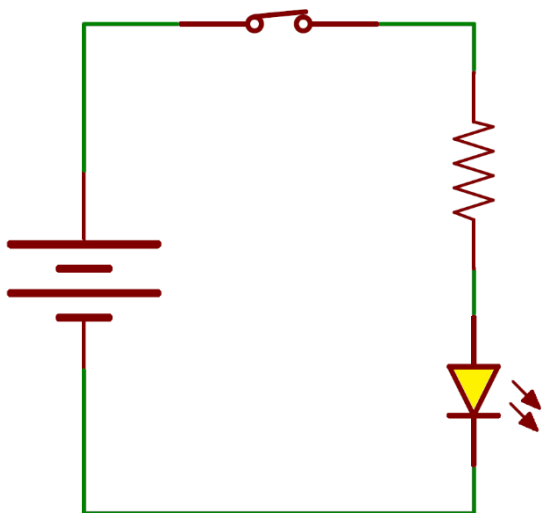
I = (V/R) = (5V/2.5Ω) = 2 A

In this circuit voltage of battery is 5V and load resistance is 2.5Ω, then ther current flow will be 2 A.

This voltage is analogus to floor. If it is 5V, then it is like moving from ground floor to 5th floor.



It is like moving from ground to 3rd floor(3V) and then from 3rd to 8th floor(5V).So, the total voltage will be 8V.



When switch is opened, there is no path for charges to flow,i.e., no current so led will not glow.
When switch is closed, there is a path to flow which allow flow of current and led will glow.

WEEK-2 (SECTION-2)

Switches are On and off device to make and break a contact in a circuit.

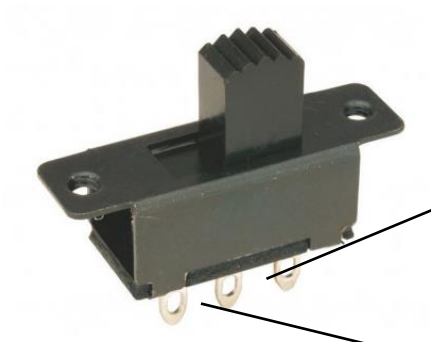
(Push Button)

Press Button-On state

Release Button-off state

(Throw Switch)

Once press and release, switch on. And once press and release off.

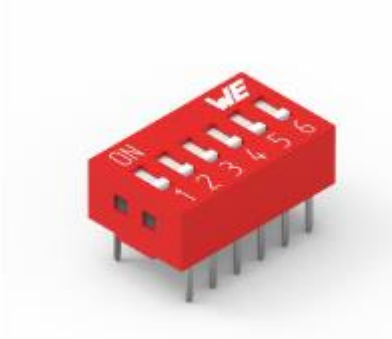


Normally closed contact

Normally open contact

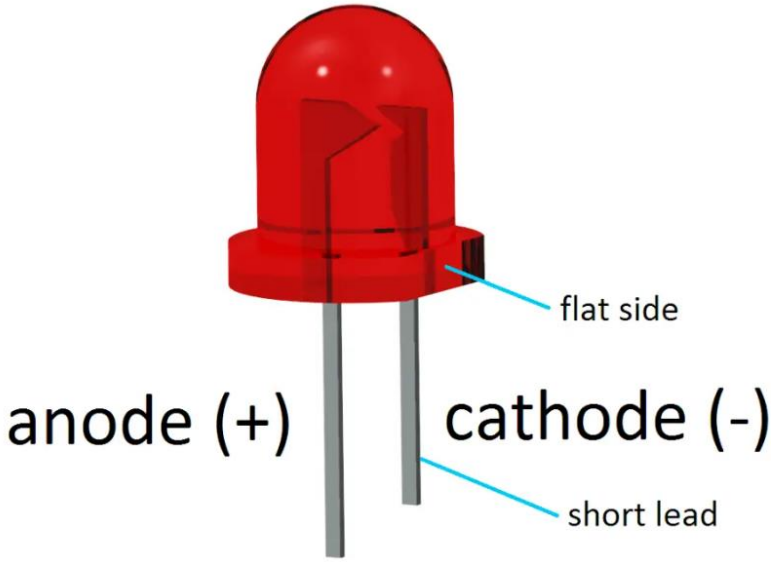
Slide Switch

(can control two circuits)



DIP Switch

(multiple throw switch)

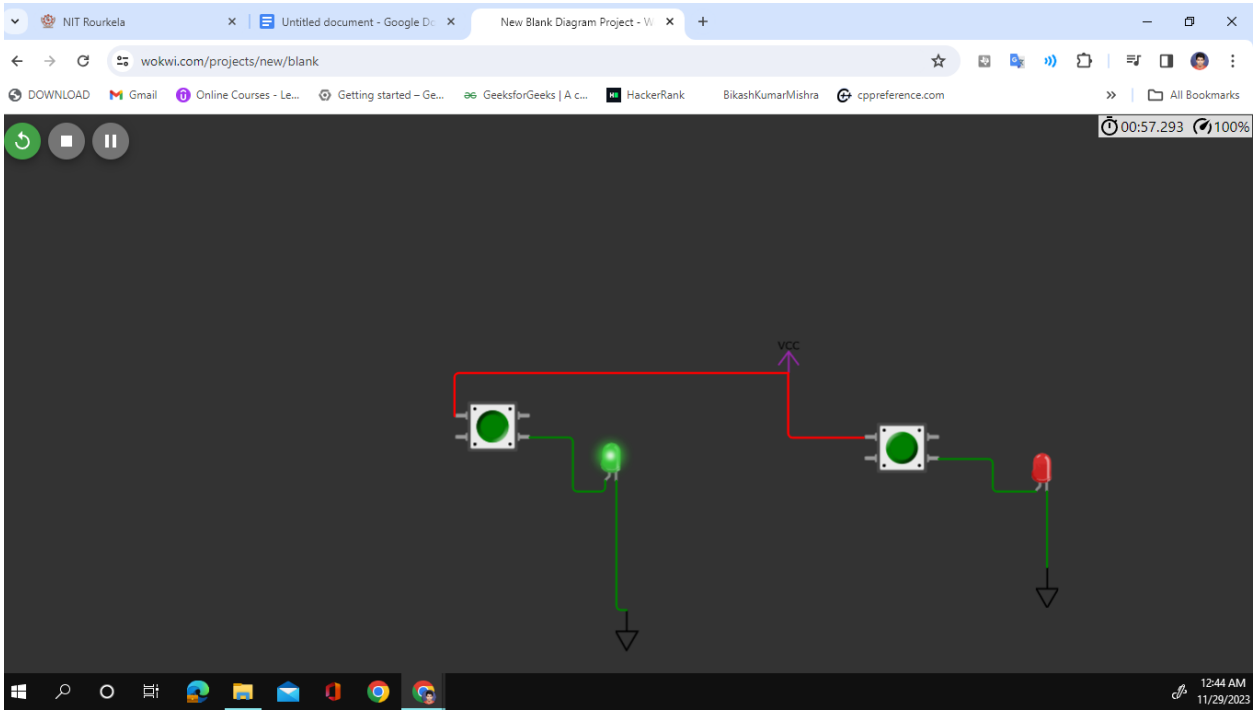


(LED-PN JUNCTION DIODE)

It is a unidirectional device only allows current in forward bias condition, where current will only flow from anode to cathode. So, positive end of the battery should be connected to the anode and negative end of the battery need to be connected to cathode.

WEEK-2 SECTION-2

PUSH BUTTON



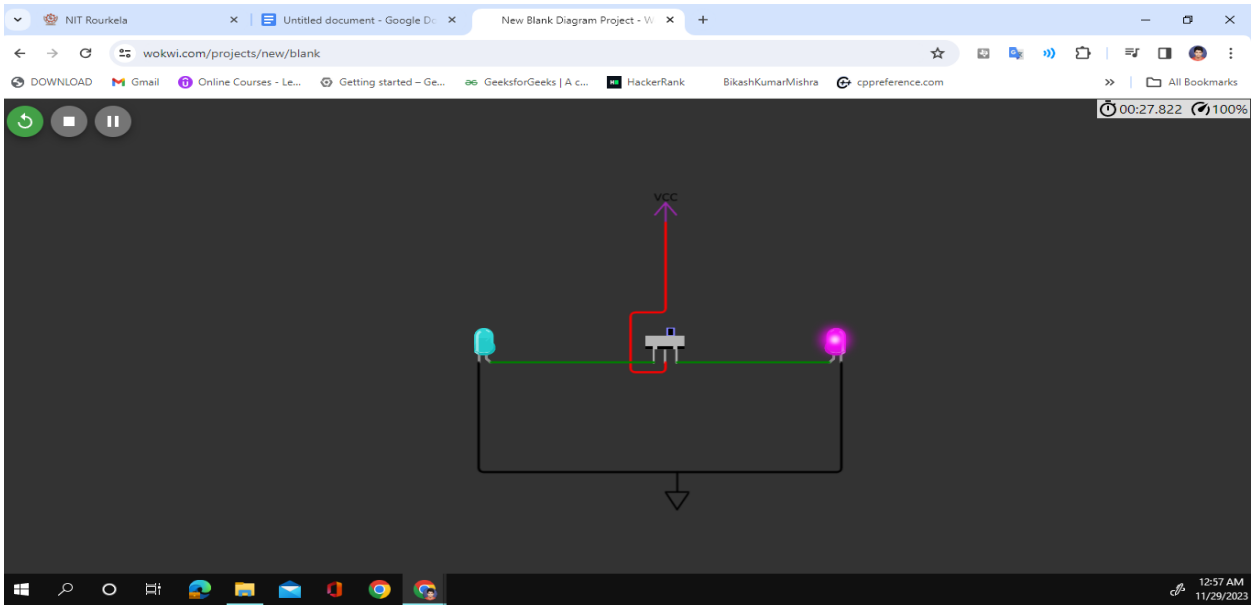
Wokwi Lab Link: https://wokwi.com/projects/382667989746475009

In this Wokwi lab, we are discussing about switches. When VCC is connected to one of the terminals of the switch, the other terminal is connected to the anode of the LED whereas the cathode terminal is connected to the ground. Whenever we push the button, the LED glows.
Note: We have to connect the positive terminal of the battery (or Vcc) to the anode and the negative terminal of the battery (or GND) to the cathode, otherwise, the LED will not glow as it is a unidirectional device.

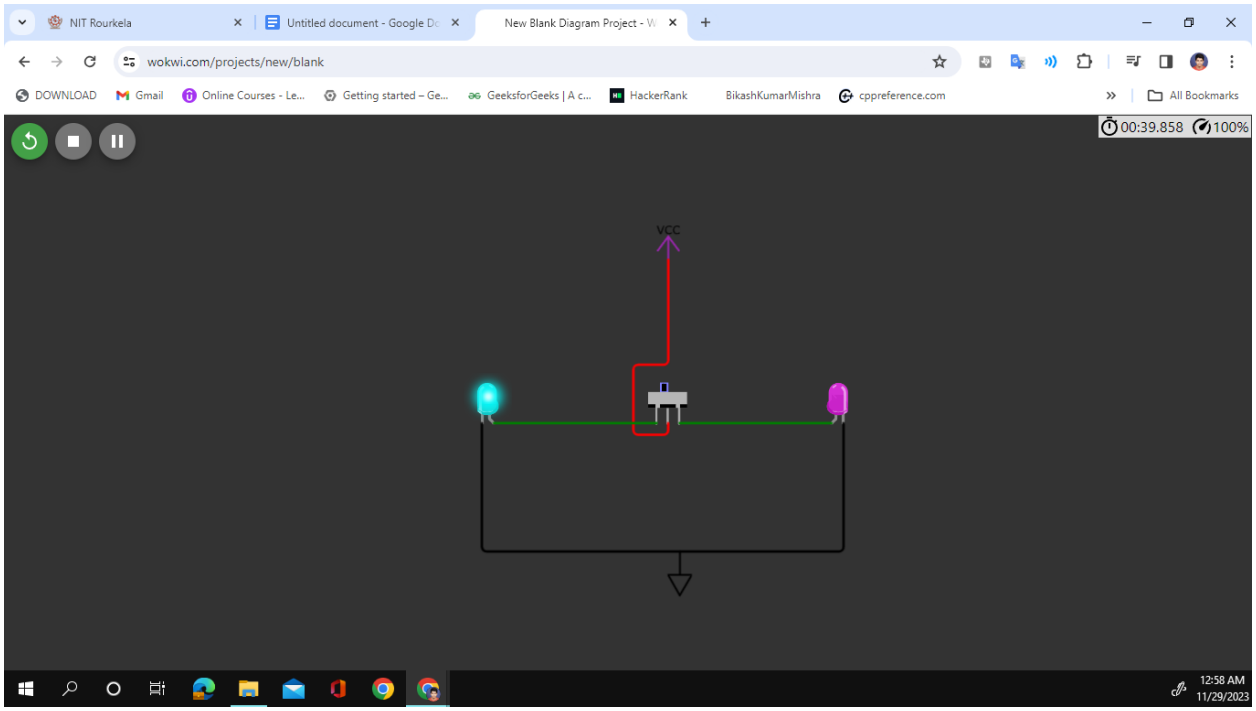
SLIDE SWITCH

In this Wokwi Lab, We are discussing about slide switch where we can control two circuits with a single switch. There are three terminals in the slide switch. The two end terminals 1 & 3 are connected to the anodes of the LED whereas the middle terminal is connected to the VCC(power supply). The cathode of both LEDs is connected to the ground.

When the slider switch is towards the right side, the VCC and anode of the pink LED are connected and it glows.



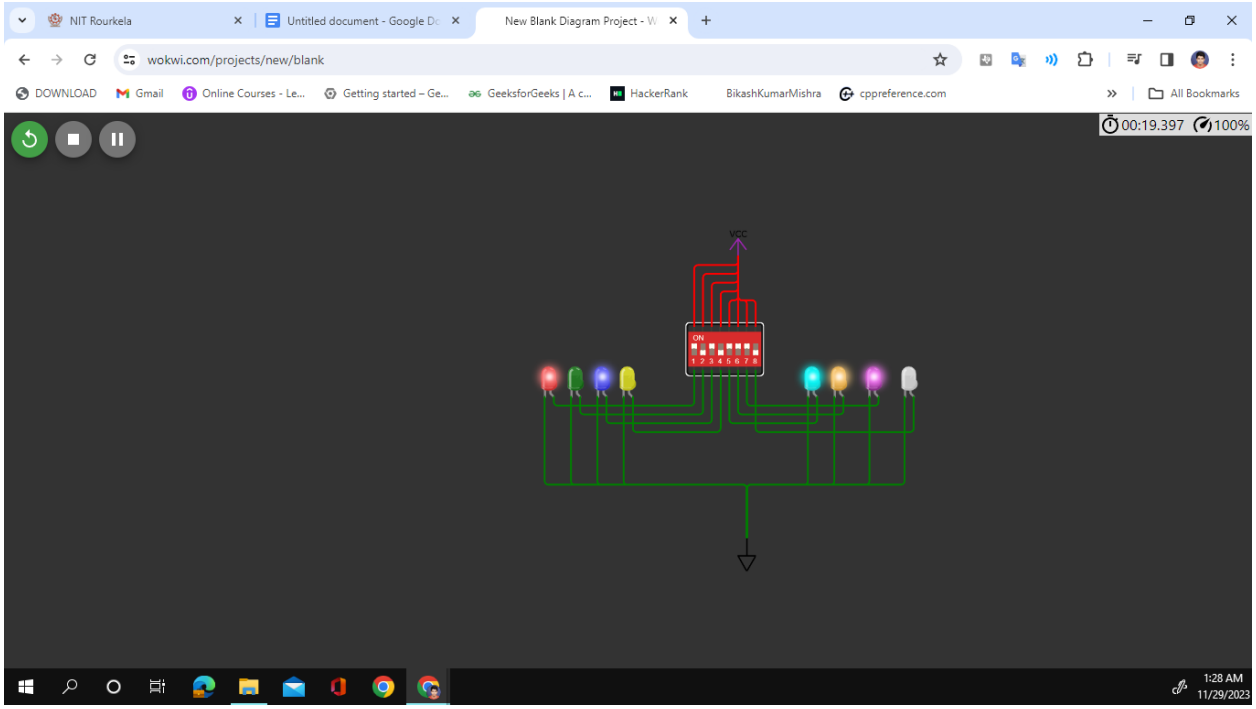
When the slider switch is towards the left side, the VCC and anode of the cyan LED are connected and it glows.



Wokwi Lab Link : https://wokwi.com/projects/382668789829944321

DIP SWITCH

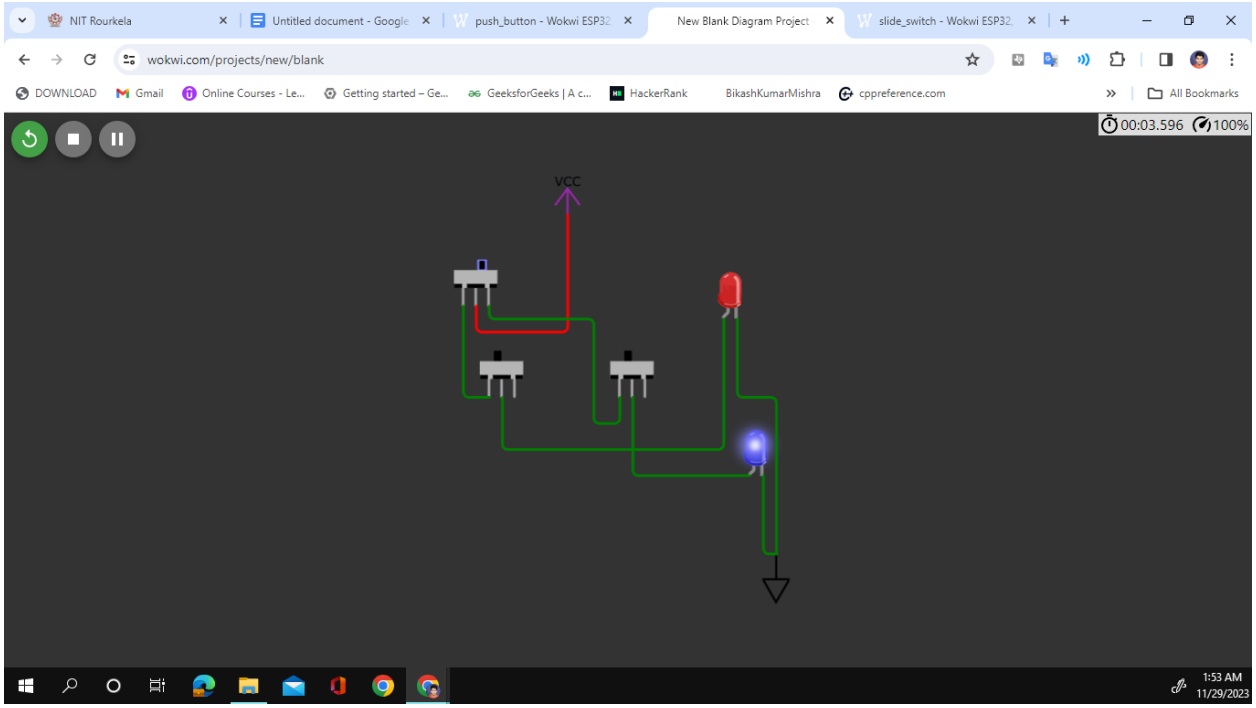
In a DIP switch, more than one switch is present which can be used to control more than one circuit to close the path. In this Wokwi Lab, We are using 8-Switch DIP.
In this DIP switch, switches no.1,3,5,6 & 7 are ON, which will glow the corresponding LEDs.



Wokwi LAB Link: https://wokwi.com/projects/382670196604829697

MULTIPLE SLIDE SWITCH

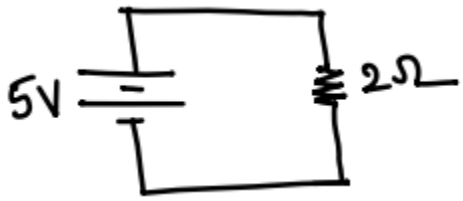
In this Wokwi lab, we are using three slide switches, to control the state of two LEDs. One slide switch's middle terminal is connected to the VCC and the other two terminals are connected to the two end terminals of the other two slide switches. The middle terminal of the two other slide switches is connected to the anode of the LEDs. The cathode of the two LEDs is connected to the ground.



Wokwi Lab Link: https://wokwi.com/projects/382672097334575105

Quiz 1

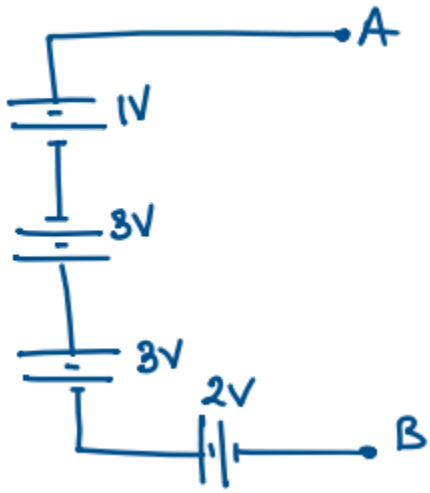
1) Find the current in the below circuit :



Solution:
In the given Circuit, Voltage(V) = 5 V and Resistance(R)=2Ω
Then, Current(I) = (V/R) = (5V/2Ω) = 2.5 A

Hence, the Current will be 2.5 A.

2) Find the net voltage across the terminal AB :

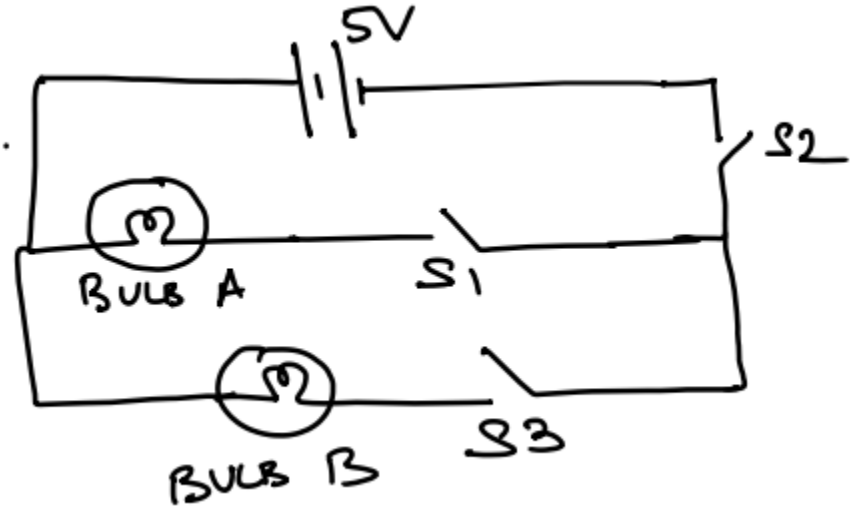


Solution:

The potential difference will be:
V = 1-3+3+2 = 3 V

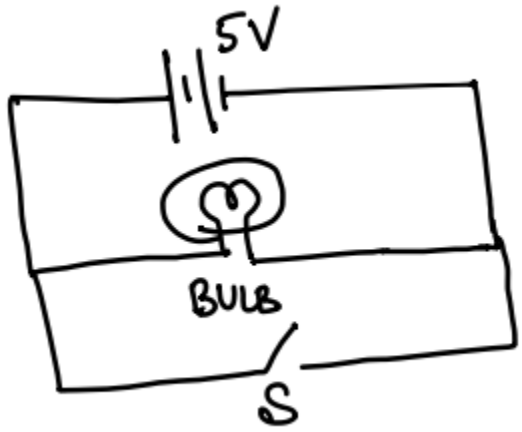
The voltage across terminal A and B is 3V.

3) Find the condition for the Bulb A and Bulb B to turn on



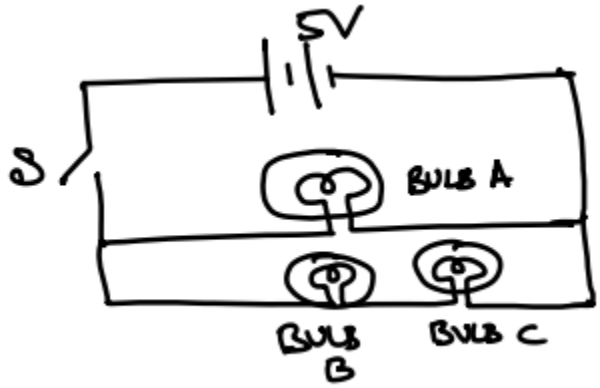
Solution:
For any of the bulb to turn on S2 must be closed. When only S1 is closed Bulb-A will glow whereas when only S3 is closed, Bulb-B will glow. If all the switches S1,S2,S3 are closed, then both Bulb-A and Bulb-B will glow.

4) What will happen when Switch S is operated in the below circuit



Solution:
If the switch is closed, the Current will try to flow through low resistance path and it will pass through the S-switch and the Bulb will not glow.

5) What happens when Switch S is operated in the below circuit



Solution:
When switch S is closed, if all the bulbs are 5V-power rated, then only Bulb-A will glow.

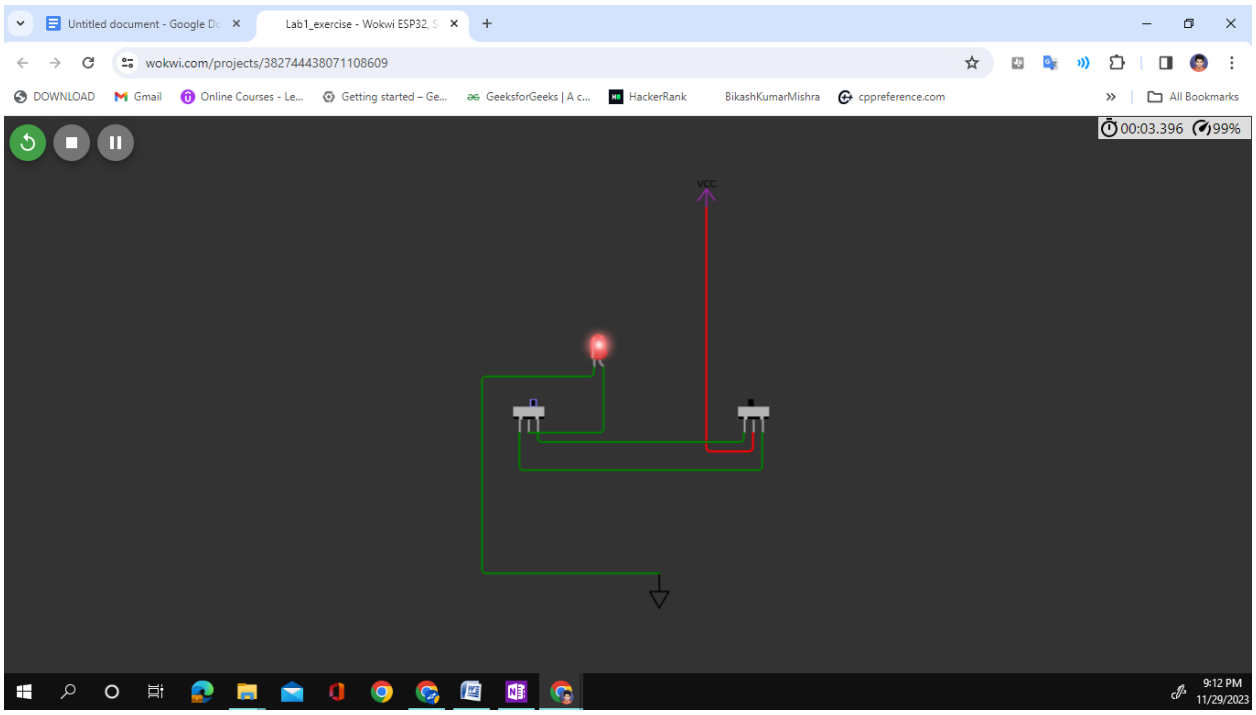
Lab1 Exercise

Staircase Wiring : Staircase wiring is a scheme of wiring where a single light can be turned ON or turned OFF from 2 different location. Imagine you have to climb up the staircase from ground floor to go first floor and its very dark in the night. You have a bulb which illuminates the path. You turn ON the bulb and start climbing. Once you reach the destination in the first floor , you need to turn off the bulb. But for this you should not be going to the ground floor again !! So , the bulb needs to be turned off from the first floor. Similarly while going from first floor to the ground floor , the bulb needs to be turned on from the first floor and turned off from the ground floor. Effectively the Bulb can be turned on or off from 2 locations. Create a circuit which will mimic this behavior.

Hint : Use a LED in place of a bulb and 2 Slide switches

Solution:

Wokwi Lab Link: https://wokwi.com/projects/382744438071108609



This is the possible solution for the staircase problem. We are using two slide switches, one power supply VCC, one ground, and one LED. The left end of both slide switches and the right end of both slide switches are connected to each other. One of the slide switch's middle terminals is connected to Vcc and the other middle terminal is connected to the anode of the LED. The cathode of the LED is connected to the ground. We can place any of the slide switches on the ground and other on the Vcc.

WEEK-3 (SECTION-1)

Switching Logics And Automation Basics

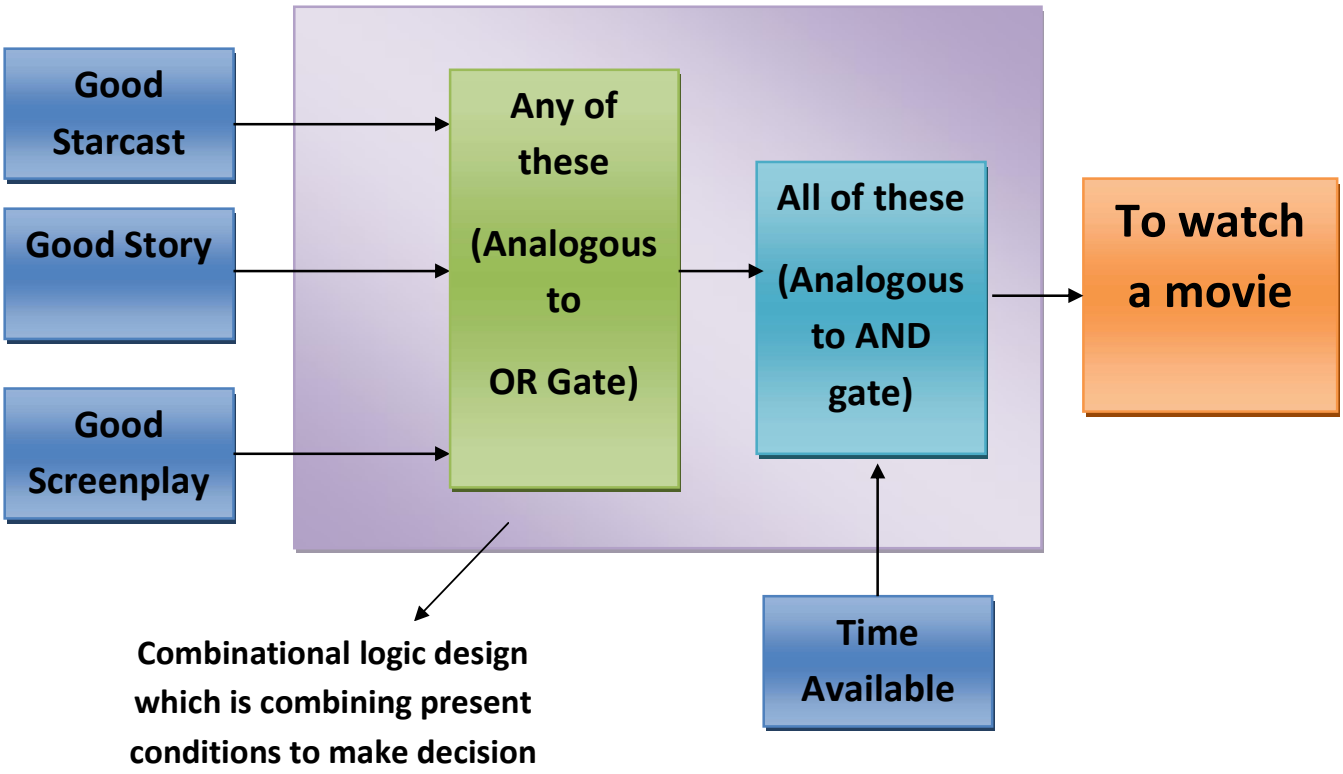
Logic is also called science of reasoning or decision making.

Example:

To watch a movie what is needed ?

Good Star cast, good story and Good screen play

Time available to watch the movie



Here. Logics are Binary(either yes or no),i.e., there are two possibilities: 1(ON/True) and 0(OFF/False). Because, all the signals present in the nature are analog signals. However, to analyze these signals using computer(for computation) we need to discretize these signals to digital signals. Hence, Binary logic is needed. The logic designs using binary logic to make decisions is the fundamental step of automation.

WEEK-3 (SECTION-2)

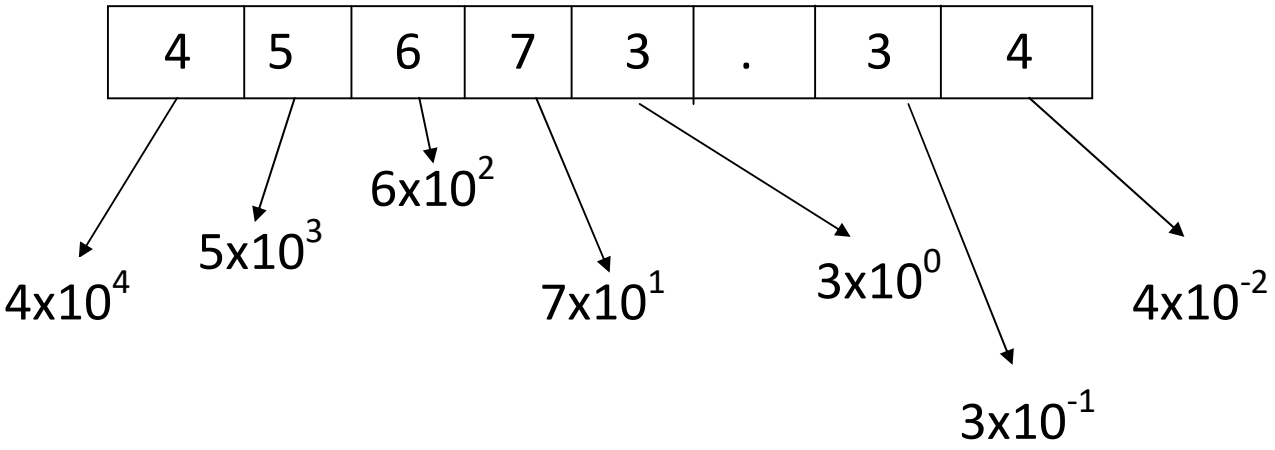
INTRODUCTION TO BOOLEAN ALGEBRA AND TRUTH TABLE

BINARY ARITHMETIC: Only 0 and 1 to represent numbers

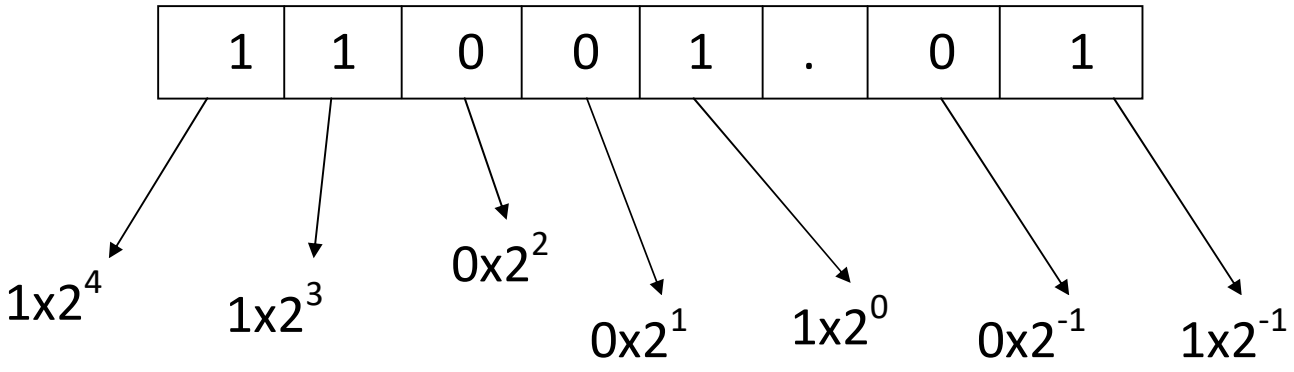
For example, 45 in decimal number system base is 10.

So, $25 = 2x10^1 + 5x10^0$. Digits are weighted by powers of base.

To represent 45673.34 in decimal(base-10) number system,



Similarly, to represent $(11001.01)_2$ in binary number system,



In decimal this binary number will be equivalent to,

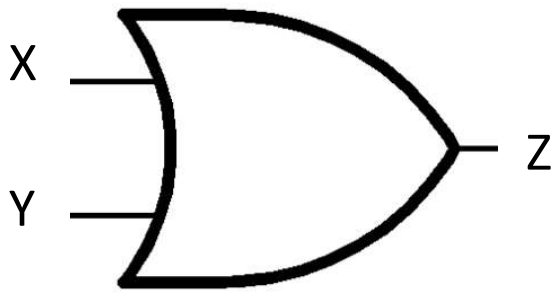
$(11001.01)_2 = (1x2^4)+ (1x2^3)+ (0x2^2)+ (0x2^1)+ (1x2^0)+ (0x2^{-1})+ (1x2^{-2})$

$= 16 + 8 + 0 + 0 + 1 + 0 + 0.25 = 25.25$

BOOLEAN ALGEBRA (for 2 inputs)

(1)OR operation

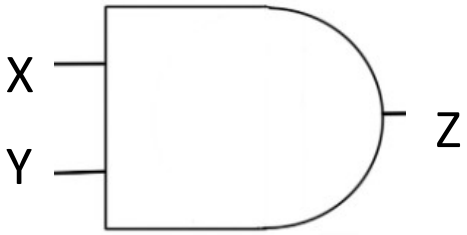
X(input)	Y(input)	Z(Output)
0	0	0
0	1	1
1	0	1
1	1	1



(Z=X.Y)

(2)AND operation

X(input)	Y(input)	Z(Output)
0	0	0
0	1	0
1	0	0
1	1	1



(Z=X+Y)

(3)OR operation (for 3 inputs)

X(input)	Y(input)	Z(input)	W(Output)
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

(W=X.Y.Z)

(4)AND operation

X(input)	Y(input)	Z(input)	W(Output)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(W=X+Y+Z)

For OR gate if all logics are 0, then the output will be zero(0), otherwise One(1). For AND gate if all logics are 1, then the output will be one(1), otherwise zero(0).

(5)NOT operation

X(input)	Y(output)
0	1
1	0

(Y=X') (where X' is complement of X)

The logical output of 1 will be zero whereas logical output of 0 will be 1. Hence, the output is complement of input.

Few important Boolean relations:

(1)X+X=X

(2)X.X=X

(3)X+1=1

(4)X.1=X

(5)X+0=X

(6)X.0=0

(7)X"=X

Truth Table: For a binary logic operation, all possible input combinations are listed and the output is evaluated.

WEEK-3(SECTION-2)

Introduction To Redundancy Theorem

Two Boolean operations are said to be equal if both of them give same output for the applied inputs.

Example:

Suppose, X and Y are two inputs and W and Z are two different Boolean experessions.

X	Y	Z=X+Y	X'	X'Y	W=X+X'Y
0	0	0	1	0	0
0	1	1	1	1	1
1	0	1	0	0	1
1	1	1	0	0	1

Here, for both Z=X+Y and Z=X+X'Y the output is same. So, both of the Boolean expressions are equal. Here, the reduced form of X+X'Y is X+Y. And this is called "**Redundancy theorem**".

$$X+X'Y=X+Y$$

Introduction To De-Morgan's Theorem

X	Y	X+Y	(X+Y)'	X'	Y'	X'.Y'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

From the Truth table, We can say, (X+Y)'=X'.Y'

X	Y	X.Y	(X.Y)'	X'	Y'	X'+Y'
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

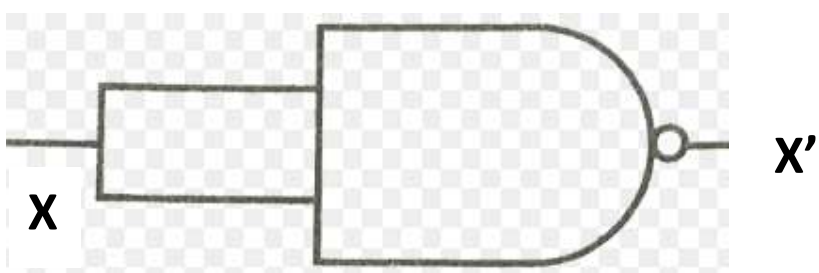
From the Truth table, We can say, (X+Y)'=X'.Y'
These two expressions in Boolean algebra are called De-Morgan's Theorem.

UNIVERSAL LOGIC GATES

The (X+Y)' is called NOR gate where as (X.Y)' is called NAND gate. NOR is inversion of OR gate whereas NAND is inversion of AND gate. The NAND and NOR gates can form all basic logic gates(AND,OR,NOT).So, these are called Universal Logic Gates.

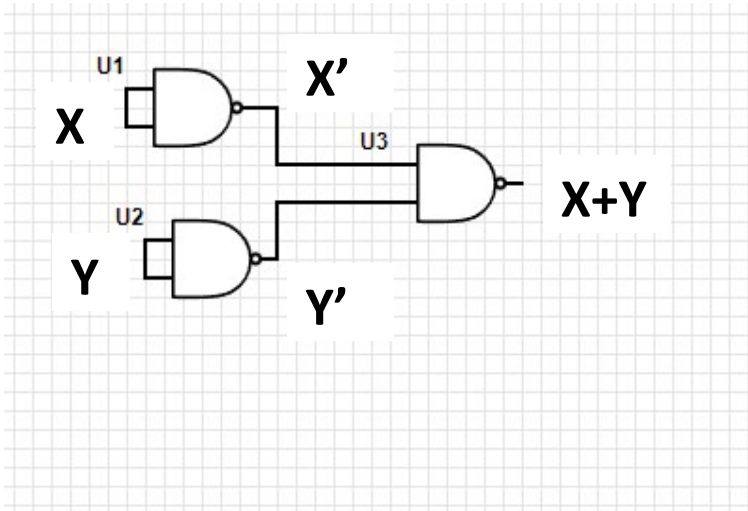
NAND as Universal Gate:
(1) NOT gate:

(X.X)' = X'



(2)OR gate:

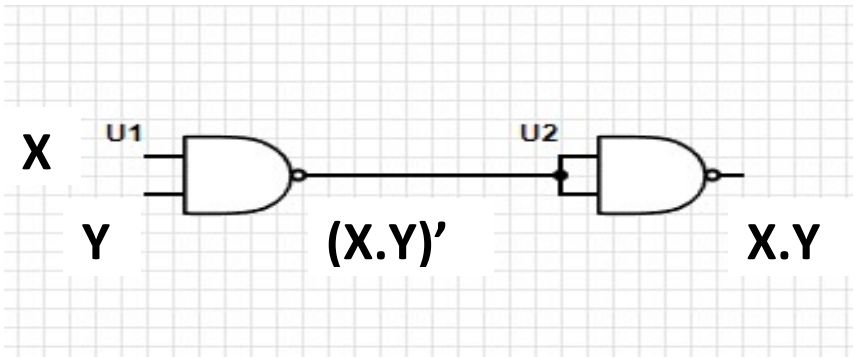
((X.X)'.(Y.Y)')' = (X'.Y')' = X"+Y"=X+Y



In this figure above if we apply X and Y inputs in U1 and U2, respectively, we will get the output at U3 as X+Y

(3)AND gate:

((X.Y)')'= (X.Y)" = X.Y

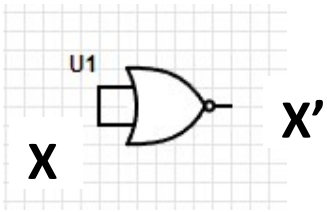


In this figure above if we apply X and Y inputs in U1, we will get the output at U2 as (X.Y)

NOR as Universal Gate:

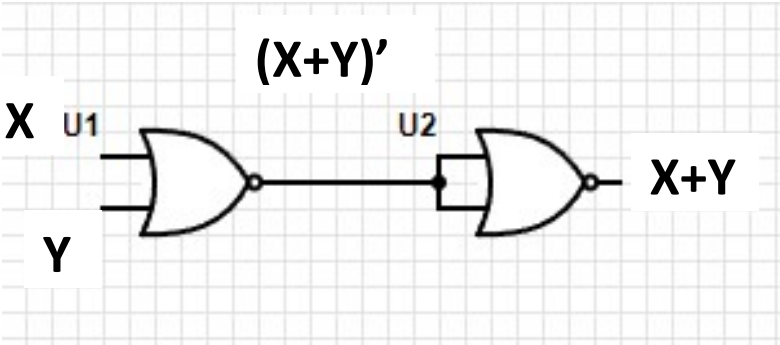
(4) NOT gate:

(X+X)’ = X’



(5)OR gate:

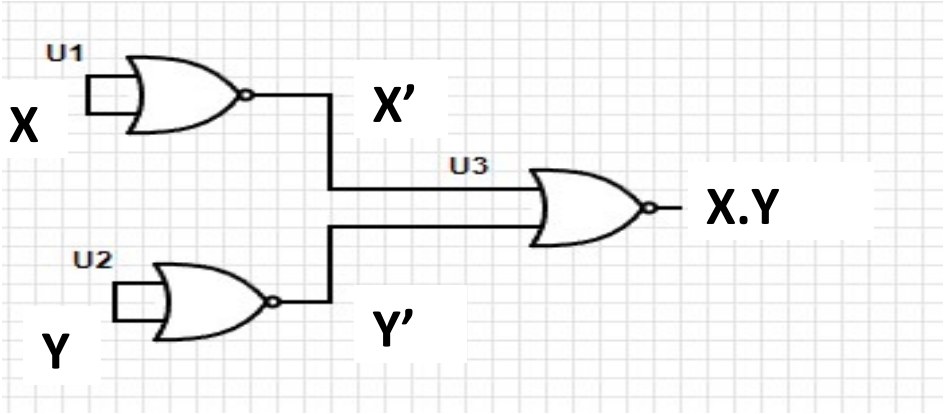
((X+X)’+(Y+Y)’)’ = (X’+Y’)’ = X”.Y”=X.Y



In this figure above if we apply X and Y inputs in U1, we will get the output at U2 as X+Y

(6)AND gate:

((X.Y)’)’= (X.Y)” = X.Y



In this figure above if we apply X and Y inputs in both U1 and U2, we will get the output at U3 as (X.Y).

WEEK-4(SECTION-1)

Introduction To Ex-OR and Ex-NOR OPERATIONS

Exclusive Or: This gate is basically Odd 1's detector.

Exclusive NOR: It is basically even 1's detector.

The Boolean expression for Ex-OR is $\oplus$. And for Ex-Nor, it is $\odot$.

For two inputs A and B ,

(1) Ex-OR:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

$$A \oplus B = AB' + A'B$$

(2) Ex-NOR:

A	B	$A \odot B$
0	0	0
0	1	1
1	0	1
1	1	0

$$A \odot B = AB + A'B'$$

Ex-NOR is inverse or complement of Ex-OR or we can say Ex-NOR is inverse of Ex-OR.

Proof:

$(A \oplus B) = (AB' + A'B)'$

$= (AB')'.(A'B)'$ (De-Morgan's Theorem)

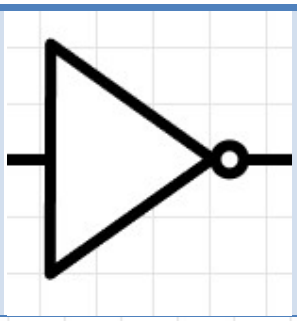
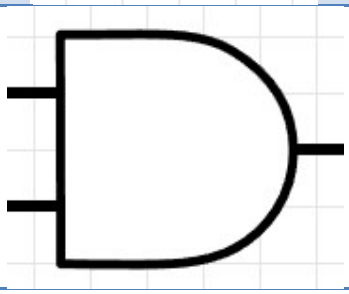
$= (A' + B'').(A'' + B')$ (De-Morgan's Theorem)

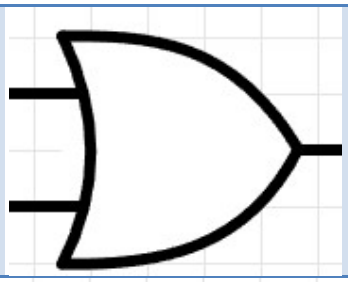
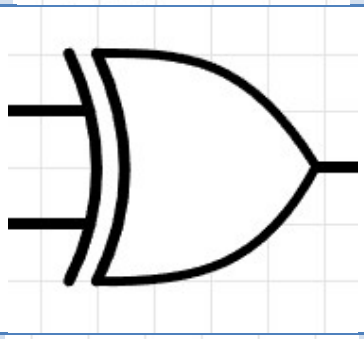
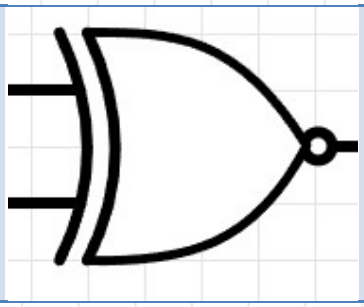
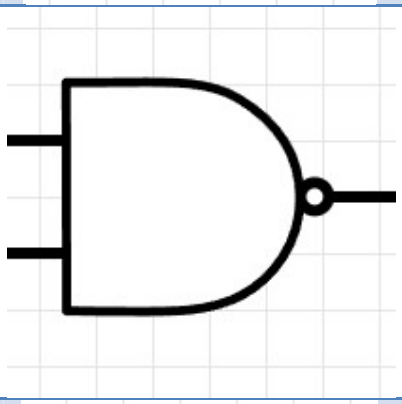
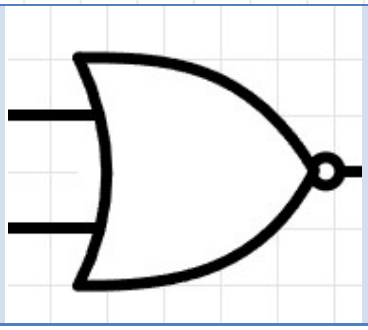
$= (A' + B).(A + B')$ ($A'' = A$ and $B'' = B$)

$= A'.A + A'B' + B.A + B.B'$ (Distributive law)

$= 0 + A'B' + B.A + 0$

$= AB + A'B' = A \odot B$

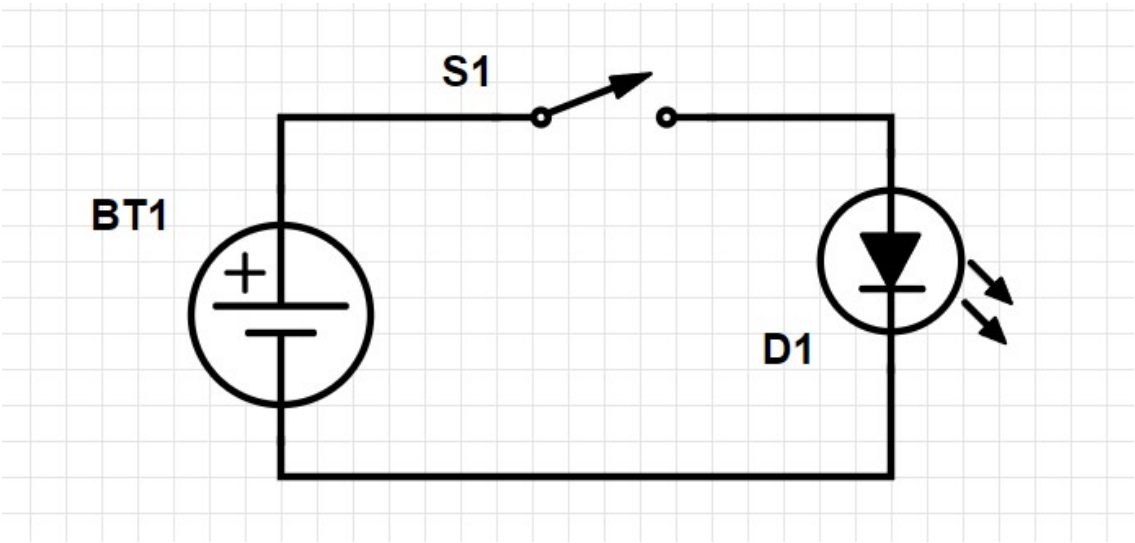
Gates	Circuit Diagram	Logical Expression(for two inputs,except NOT gate)
NOT		A'
AND		A.B

OR		A+B
Ex-OR		$A \oplus B$
Ex-NOR		$A \odot B$
NAND		(A.B)’
NOR		(A+B)’

WEEK-4(SECTION-2)

Introduction Basic Switch OPERATION

All digital inputs and outputs are either in the form of 0 or 1 as same as “ON” or “OFF” state. Hence, all Boolean functions are switching functions. So, using switches we can realize Boolean functions.



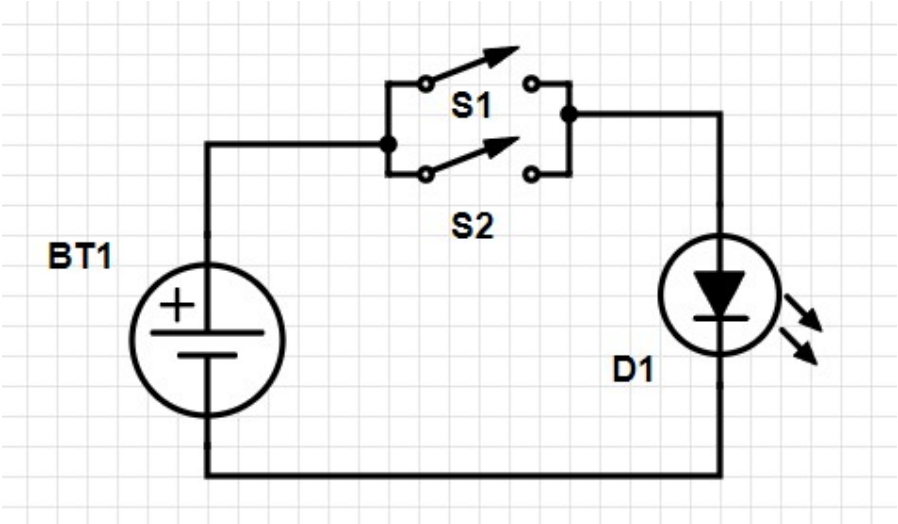
In this schematic when switch is open, led is off and if switch is closed led is on.

Now this acts as a buffer circuit, where the switch is buffer. If switch is opened(means 0), then led is off and when switch is closed(means 1), then led is on.

Similarly,

Realization of OR-gate:

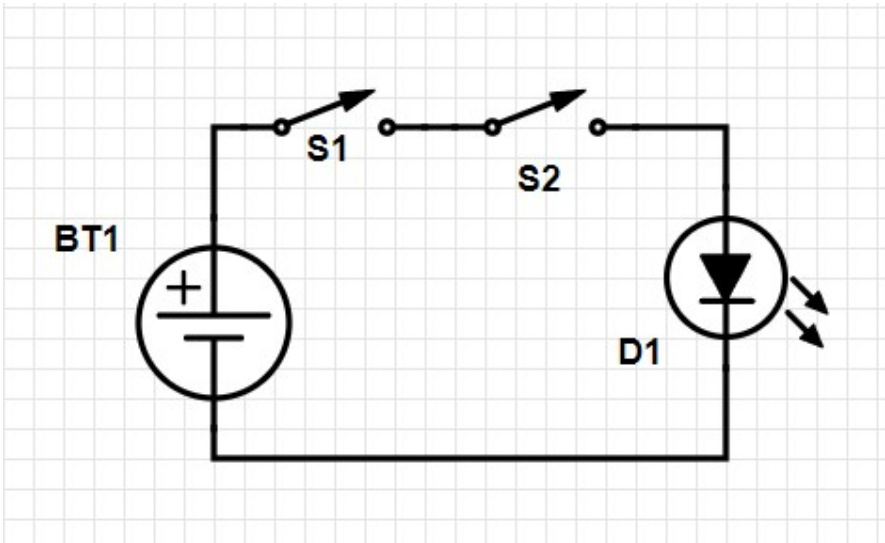
In the circuit below, if any of the switch is closed, then led will glow. And only when both the switches are opened, then led will be off. This logic is similar to the OR-gate. If the switch is open, then its state in digital is 0 and if the switch is closed, then its state in digital is 1.



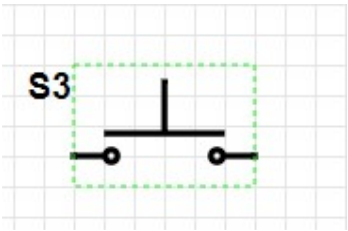
Switch S1	Switch S2	Output
Open(0)	Open(0)	OFF(0)
Open(0)	Closed(1)	ON(1)
Closed(1)	Open(0)	ON(1)
Closed(1)	Closed(1)	ON(1)

Realization of AND-gate:

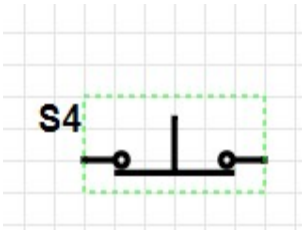
In the circuit below, if any of the switch is opened, then led will be OFF. And only when both the switches are closed, then led will glow. This logic is similar to the AND-gate.



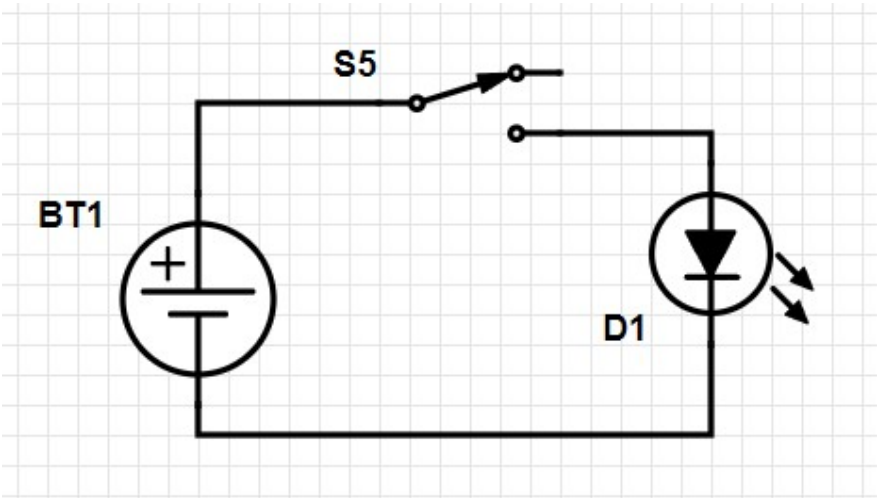
Switch S1	Switch S2	Output
Open(0)	Open(0)	OFF(0)
Open(0)	Closed(1)	OFF(0)
Closed(1)	Open(0)	OFF(0)
Closed(1)	Closed(1)	ON(1)



Normally Opened(NO)



Normally Closed(NC)



Realization of NOT gate:

In this circuit there are two states one is Normally opened(0) and other is normally closed state. This circuit will act as inverter (NOT gate), because when switch is opened, the led is glowing and when switch is closed, the led will be off.

Switch S5	Output
Open(0)	ON(1)
Close(1)	OFF(0)

`

WEEK-4(SECTION-3)

COMBINATIONAL LOGIC DESIGN

Combinational circuit is defined as the circuit where output depends on present input state.

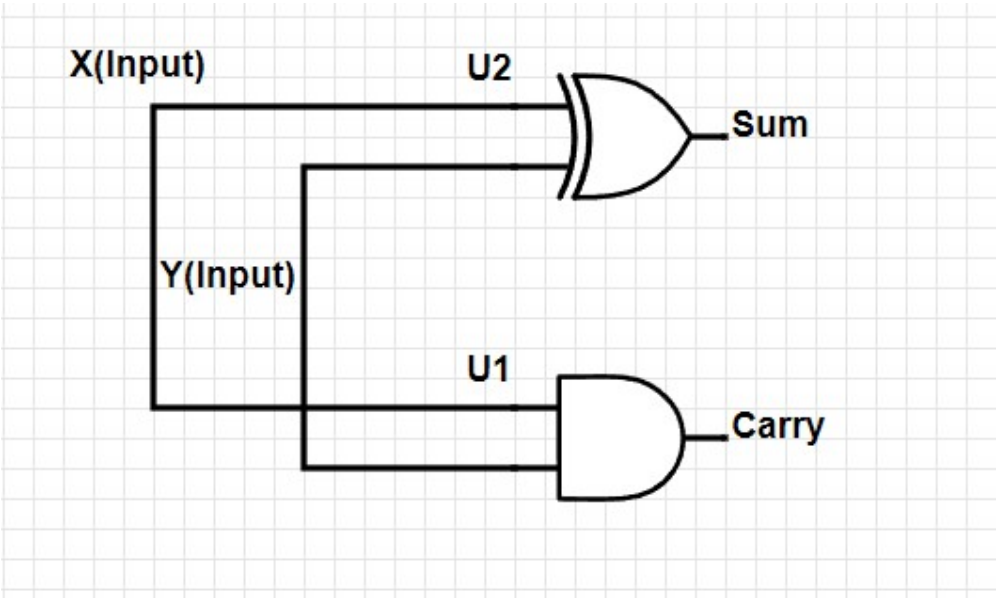
To design a combinational logic to add two 1-bit number:

Truth Table:

X(Input)	Y(Input)	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

If we analyze the sum as functions of X and Y, then sum will be Ex-OR gate of X and Y. And the Carry will be AND function of X and Y.

So, Sum = $X \oplus Y$ and Carry$=X.Y$



To design a 3-bit Number to detect whether it is a prime number or not:

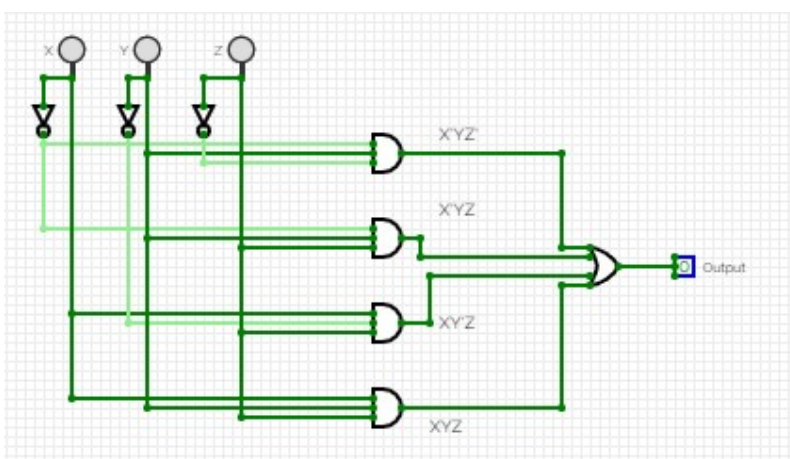
In a 3-bit number it is ranging from 000(0) to 111(7). So, whenever the 3-bit number is prime number (2,3,5,7), then the output will be 1. Otherwise, it will be zero.

X(MSB)	Y	Z(LSB)	Output(W)	Min-Terms
0	0	0	0	X'Y'Z'
0	0	1	0	X'Y'Z
0	1	0	1	X'YZ'
0	1	1	1	X'YZ
1	0	0	0	XY'Z'
1	0	1	1	XY'Z
1	1	0	0	XYZ'
1	1	1	1	XYZ

So, if we calculate the function of W in terms of X,Y,Z. Then,

W=∑m(2,3,5,7) (in SOP form Sum of min-terms

=X'YZ'+ X'YZ+ XY'Z+ XYZ (needs to be minimized)



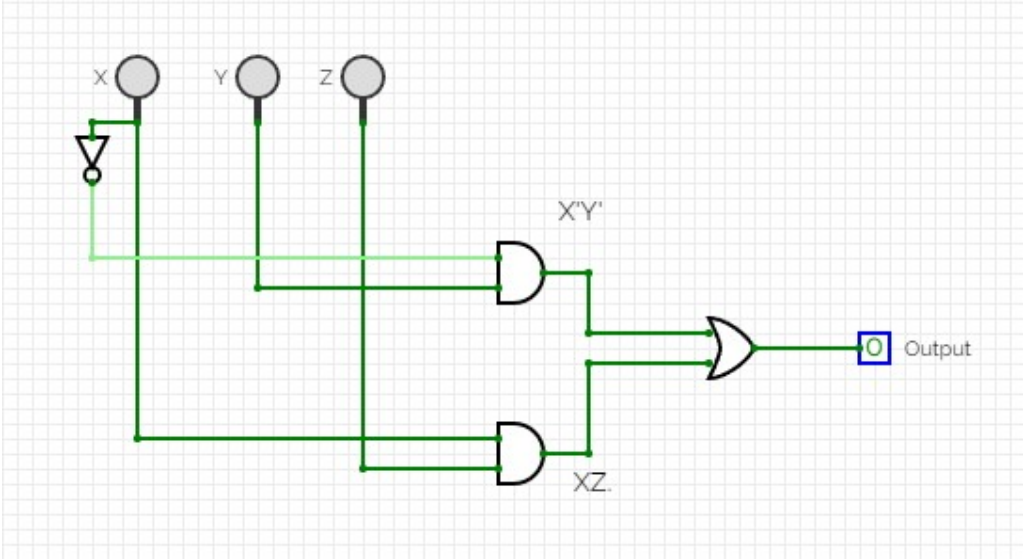
However, this is not a good technique to design a logic as it will increase the size and increase the cost. Hence, We need to minimize in order to reduce the number of gates which will eventually reduce the cost and size.

To minimize we can either do Boolean operation or we can do K-Map.

Using Boolean Operation,

W =X'YZ'+ X'YZ+ XY'Z+ XYZ =X'Y.[Z'+Z] + XZ.[Y'+Y] = X'Y+XZ

Hence, the logic design is minimized.



WEEK-4(SECTION-4)

MINIMIZATION USING K-MAP

The K-Map is based on the algorithm if for any of the two input combinations is giving output 1, then we can neglect the changing bit.

For grouping one bit in a K-Map,

X'YZ and X'YZ' – we can group it to form X'Y as for both the input Z and Z' the output is 1.

For grouping two bit in a K-Map,

X'YZ ,X'YZ', X'Y'Z' and X'Y'Z – we can group it to form X' as for inputs Y'Z',Y'Z,YZ' and YZ output is 1.

Similarly, for grouping 3-bits and 4-bits in a K-Map, we need 8 and 16. So, if number of bit is n, then number of terms we need to group will be 2^n.

The design for 2-variable K-Map will be,

X,Y	Y'	Y
X'	00	01
X	10	11

The design for 3-variable K-Map will be,

X,Y,Z	Y'Z'	Y'Z	YZ	YZ'
X'	000	001	011	010
X	100	101	111	110

The design for 4-variable K-Map will be,

X,Y,Z,W	Z'W'	Z'W	ZW	ZW'
X'Y'	0000	0001	0011	0010
X'Y	0100	0101	0111	0110
XY	1100	1101	1111	1110
XY'	1000	1001	1011	1010

Examples for grouping,

(1) 2-VARIABLE: Output =X'Y'+X'Y

As Y' and Y is common after grouping the output will be X'.

X,Y	Y'	Y
X'	1	1
X		

(2) 3-VARIABLE: Output =X'Y'Z+XY'Z'+X'Y'Z'+XY'Z

Here after grouping 4, the minimized expression will be

Output=Y'

X,Y,Z	Y'Z'	Y'Z	YZ	YZ'
X'	1	1		
X	1	1		

For Our previous example prime number detection,

Minimized expression will be

X'Y

W=∑m(2,3,5,7)

=X'YZ'+ X'YZ+ XY'Z+ XYZ

X,Y,Z	Y'Z'	Y'Z	YZ	YZ'
X'			1	1
X		1	1	

Minimized expression will be

XZ

Hence, the expression for W reduces to W =X'Y+XZ.

Some Other Examples,

(1) For this 3-variable K-Map

X,Y,Z	Y'Z'	Y'Z	YZ	YZ'
X'	1		1	
X				1

Here, further minimization is not possible as adjacent grouping is not valid. So, the output will be X'Y'Z'+X'YZ+X'YZ'.

(2) Let suppose for this truth table

X(Input)	Y(Input)	Z(Input)	W(Output)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

So, the K-Map for this truth table will be,

X,Y,Z	Y'Z'	Y'Z	YZ	YZ'
X'	1	1	1	1
X				

Minimized expression will be

X'

So, the output will be, W=X'.

WEEK-5(SECTION-1)

K-MAP OPTIMIZATION

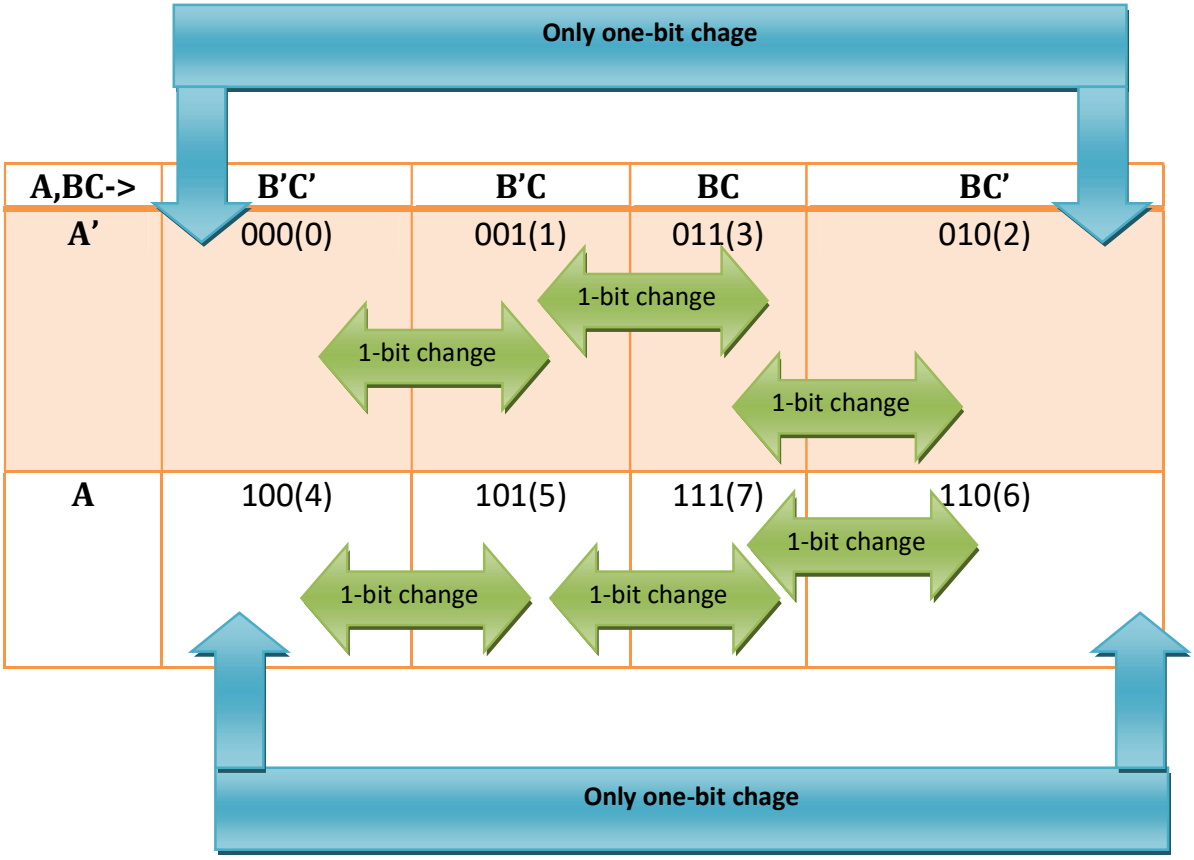
The basic principle of K-Map is that if only one-bit is changing then grouping can be done which is called minimization using K-Map.

For example,

ABC'+AB'C'=AC'.(B+B')=AC' simplified expression

However, ABC'+AB'C=A.(BC'+B'C)=A.(B$\oplus$C) which is not a simplified expression

Because, number of inputs are same.



In all of these cases, only one-bit is changing. So, grouping is possible. However, adjacent grouping is not possible.

Example of a Truth Table:

A	B	C	Y(OUTPUT)
0	0	0	0
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	0

A,BC->	B'C'	B'C	BC	BC'
A'		1	1	1
A		1		1

Y= B'C+BC'+A'BC. But, is this the most optimized Solution.

If we want to simplify using Boolean Algebra,

A'B'C+A'BC+AB'C+ABC'+A'BC' = B'C.(A'+A) + BC'.(A+A')+A'BC=B'C+BC'+A'BC

If I want to simplify using,

A'B'C+A'BC+AB'C+ABC'+A'BC'= A'B'C+A'BC+A'BC+AB'C+ABC'+A'BC' (as X+X=X)

=A'C.(B'+B)+BC'.(A+A')+AB'.(C+C')=A'C+BC'+AB'

So, we can group in the K-Map A'BC with A'BC' which will be simplified to A'B.

A'BC+A'BC'=A'B.(C+C')=A'B

A,BC->	B'C'	B'C	BC	BC'
A'		1	1	1
A		1		1

A,BC->	B'C'	B'C	BC	BC'
A'		1	1	1
A		1		1

However, this simplification is not possible grouping of 3. Only grouping of 2^n is possible. We can group either A'B'C and A'BC or A'BC and A'BC'. The result will be same.

For K-Map like these,

A,BC->	B'C'	B'C	BC	BC'
A'	1	1	1	1
A				

The Output will be,

A'B'+A'B=A'.(B'+B)=A'. So, here we can directly group all four variables as,

A,BC->	B'C'	B'C	BC	BC'
A'	1	1	1	1
A				

For example, in this K-Map,

A,BC->	B'C'	B'C	BC	BC'
A'	1	1		
A	1	1		

The grouping 4 will be simplified to B'.

In the following K-Map, groping of 4 is not possible. We can group A’B’C’ and AB’C’ and then A’B’C and A’BC.

A,BC->	B’C’	B’C	BC	BC’
A’	1	1	1	
A	1			

The simplified result will be,

B’C’+B’C+BC =B’C’ + C.(B’+B) = C+C’B’ =C+B’ (as A+A’B = A + B)

For example,

1	**1**	**1**	**1**

1	**1**		
1	1		

1	1	1	1

		1	**1**
		1	1

	1	**1**	
	1	1	

1			**1**
1			1

In these above cases grouping of 4 1's is possible. However in these below cases grouping of 4 1's is not possible.

1	**1**		
		1	1

		1	**1**
1	1		

1		**1**	
	1		1

1	**1**	**1**	
1			

	1	**1**	**1**
			1

			1
	1	1	1

1			
1	1	1	
	1	1	
1			1

1			**1**
	1	1	

1		1	1
			1

1	1		1
1			

WEEK-5(SECTION-2)

FOUR VARIABLES K-MAP

In K-map the adjacent cells arranged in such a way that adjacent cells are changing with one-bit variation.

0000(0)	0001(1)	0011(3)	0010(2)
0100(4)	0101(5)	0111(7)	0110(6)
1100(12)	1101(13)	1111(15)	1110(14)
1000(8)	1001(9)	1011(11)	1010(10)

A	B	C	D	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

For example, a Boolean function Y is output of 4-variable inputs(A,B,C,D).

Y=1 when ABCD=1,2,3,4,7,8,9,10 ,11 else it is 0.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

The minimized expression will be,

A'B'D'+AB'+A'B'C'+A'BC'D'

A,B,C,D	C'D'	C'D	CD	CD'
A'B'		1	1	1
A'B	1		1	
AB				
AB'	1	1	1	1

Minimized expression = A'B'D

Minimized expression = AB'

Minimized expression = A'B'C'

WEEK-6(SECTION-1)

LAB INTRO AND DEMO

DESIGNING COMBINATIONAL LOGIC

To design combinational logics, we need switches and gates. To design the circuit from truth table, we need K-Map as K-map will give me Boolean function.

However specifications can be given verbally or in wave form.

For example Specifications given verbally,

We need to design a 3-input combinational logic circuit, when the 3-bit input is **odd**, we want the logic to be high. Else, it is zero(low).

A(MSB)	B	C(LSB)	Decimal Equivalent	Y(OUTPUT)
0	0	0	0(even)	0
0	0	1	1(odd)	1
0	1	0	2(even)	0
0	1	1	3(odd)	1
1	0	0	4(even)	0
1	0	1	5(odd)	1
1	1	0	6(even)	0
1	1	1	7(odd)	1

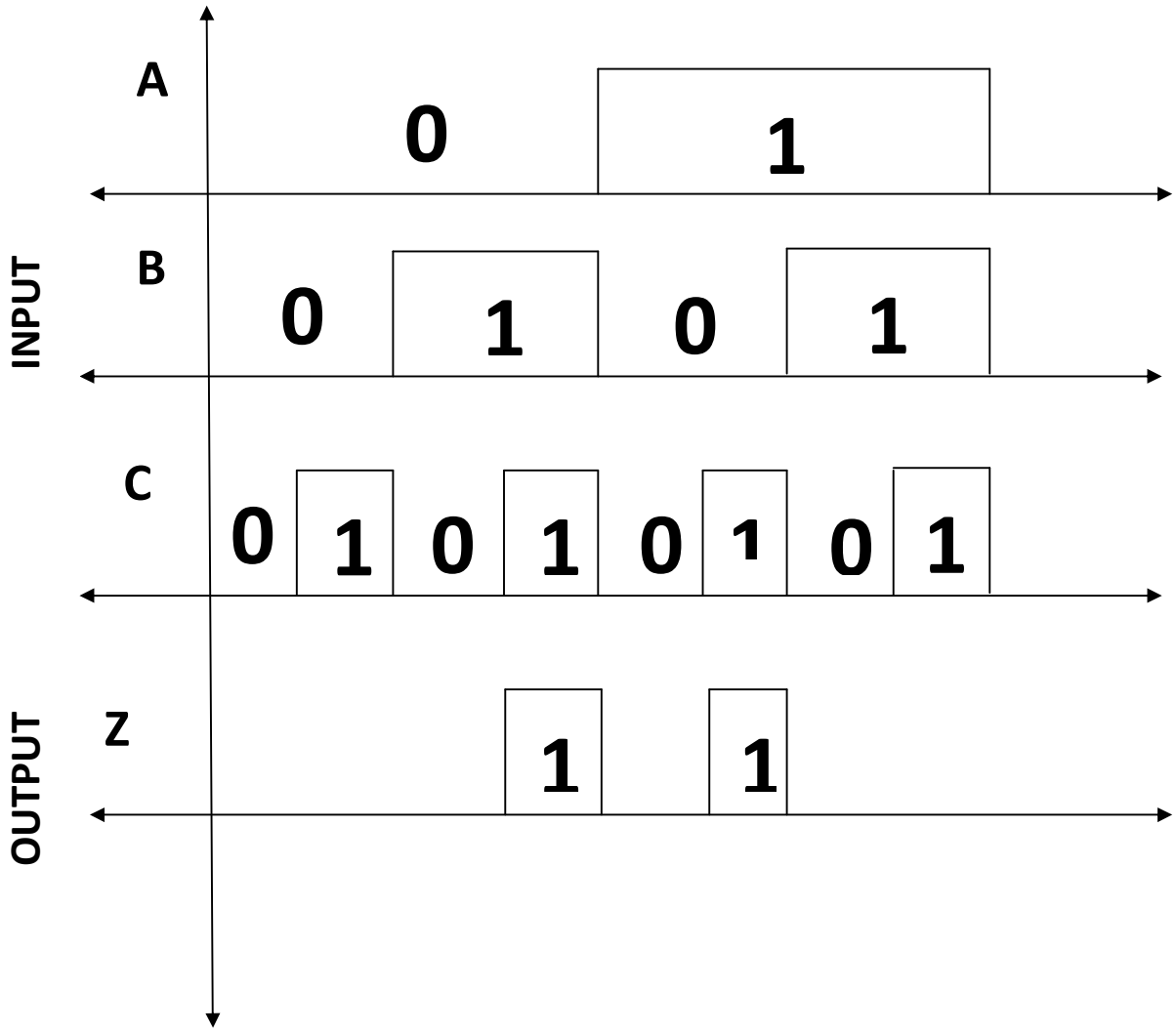
The 3-variable K-Map will be,

A,BC	B'C'	B'C	BC	BC'
A'		1	1	
A		1	1	

Minimized expression will be C

The 4 1's will be grouped and the corresponding Boolean function will be,

Y = C



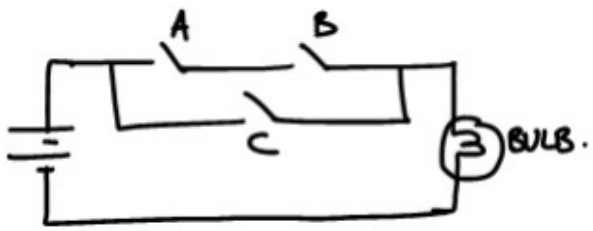
In this wave form, we have made 3-bit input combinations.[000 to 111]

OUTPUT(Z) = A'B'C+AB'C = C.(A⊕B). No Simplification is possible.

LAB-2

Implement the logical expression in each of the below mentioned cases for the bulb to turn on using Logic Gates.

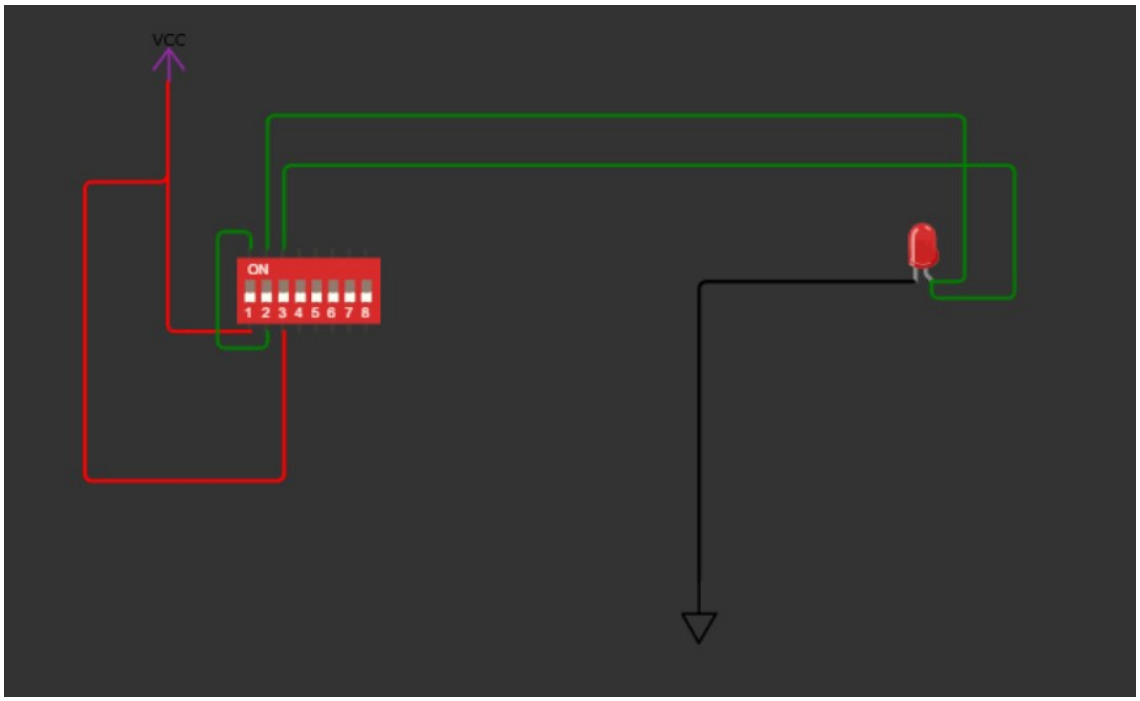
Validate the same in Wokwi tool by simulating the circuit using LED in place of Bulb.



1.

Solution:

Using DIP switch:



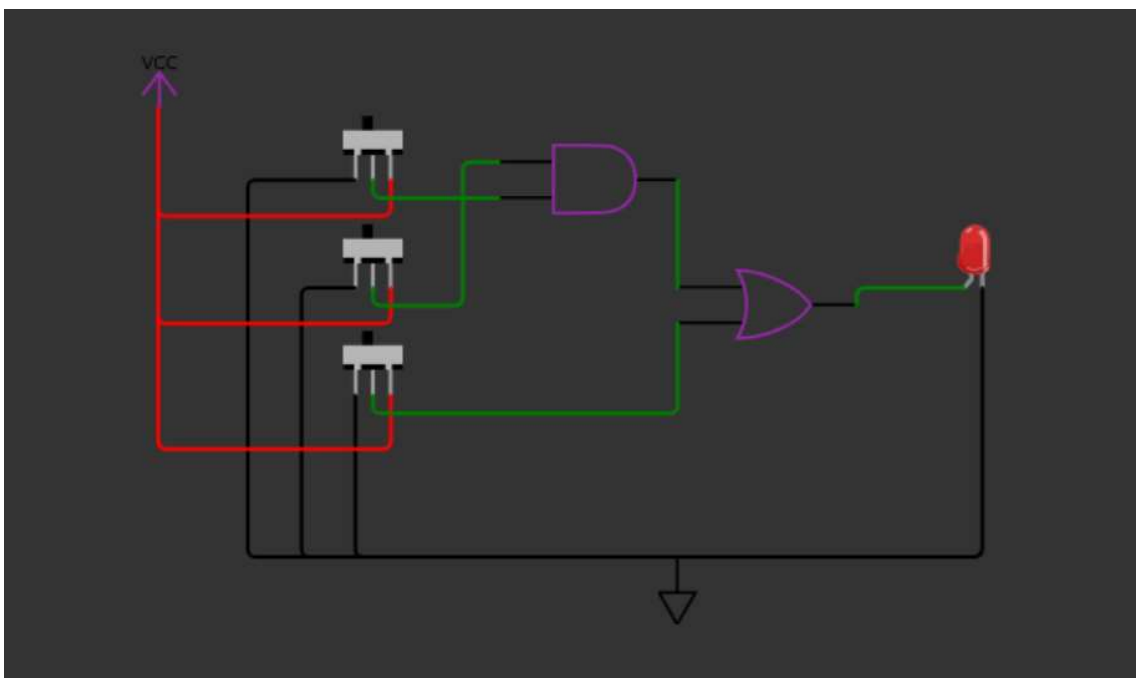
https://wokwi.com/projects/384971630474331137

In the above figure, it is indicating that either C is closed(1) or Both A and B are closed(1), then the led will be turned on.

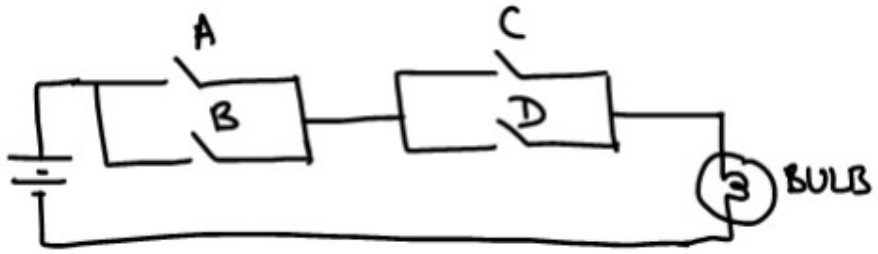
A(SWITCH)	B(SWITCH)	C(SWITCH)	Y(OUTPUT)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A,BC	B'C'	B'C	BC	BC'
A'		1	1	
A		1	1	1

Y(output) = AB+C



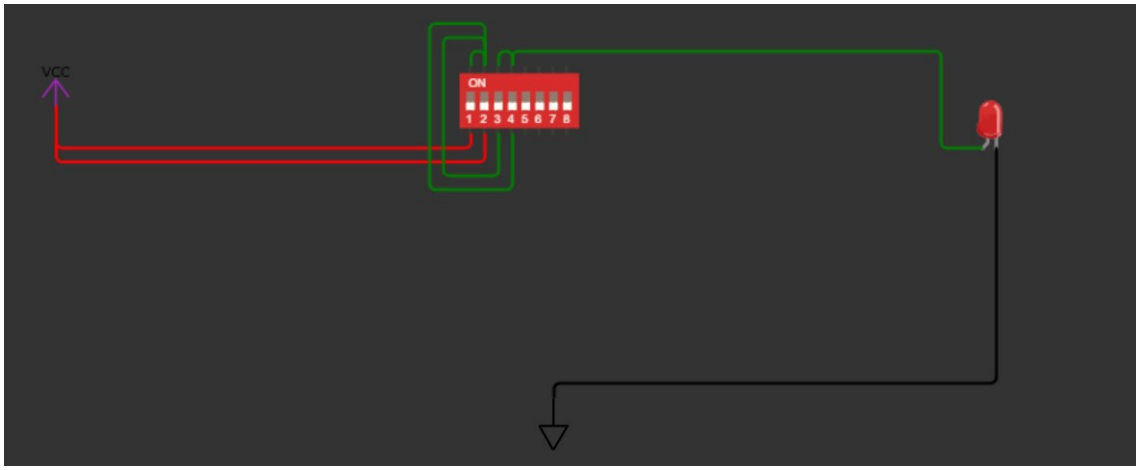
https://wokwi.com/projects/385053629378296833



2.

Solution:

Using DIP switch:



https://wokwi.com/projects/385054799358480385

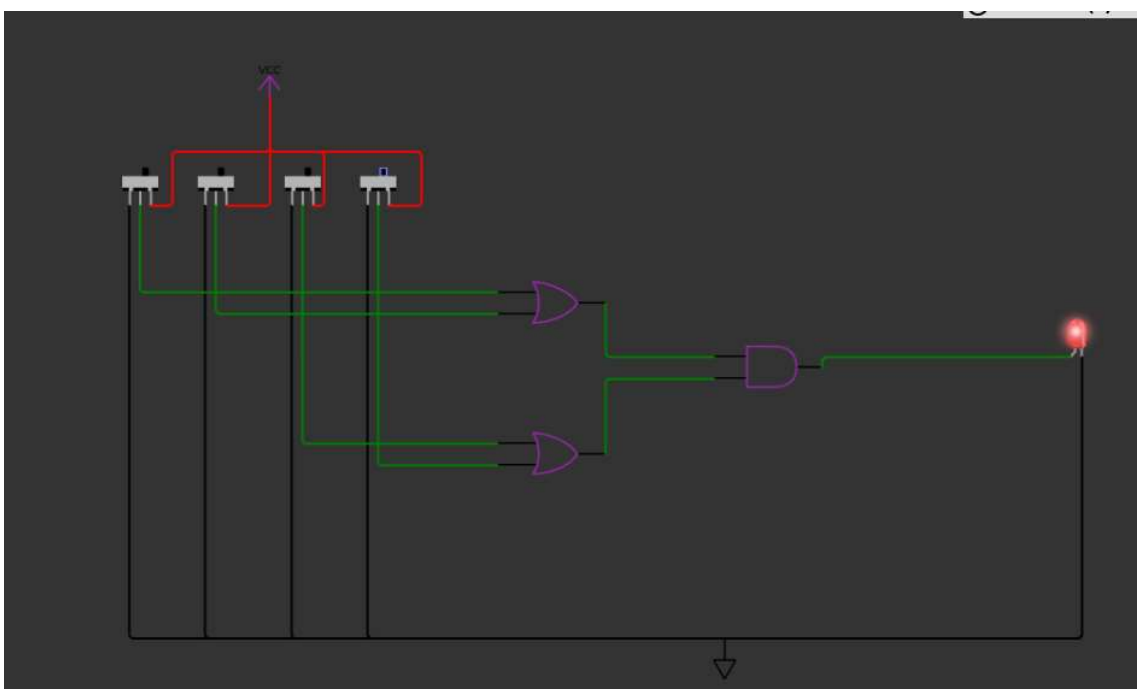
A(SWITCH)	B(SWITCH)	C(SWITCH)	D(SWITCH)	Y(OUTPUT)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1

1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

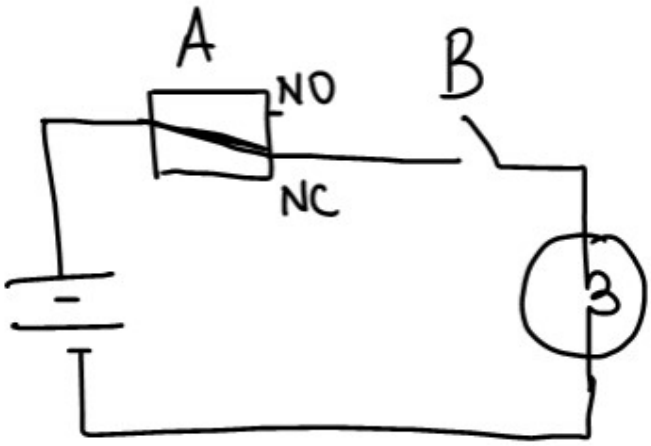
AB,CD	C'D'	C'D	CD	CD'
A'B'	0	0	0	0
A'B	0	1	1	1
AB	0	1	1	1
AB'	0	1	1	1

The minimized expression will be,

Y(OUTPUT)= BD + BC + AD + AC = B.(C+D)+A.(C+D) =(A+B).(C+D)



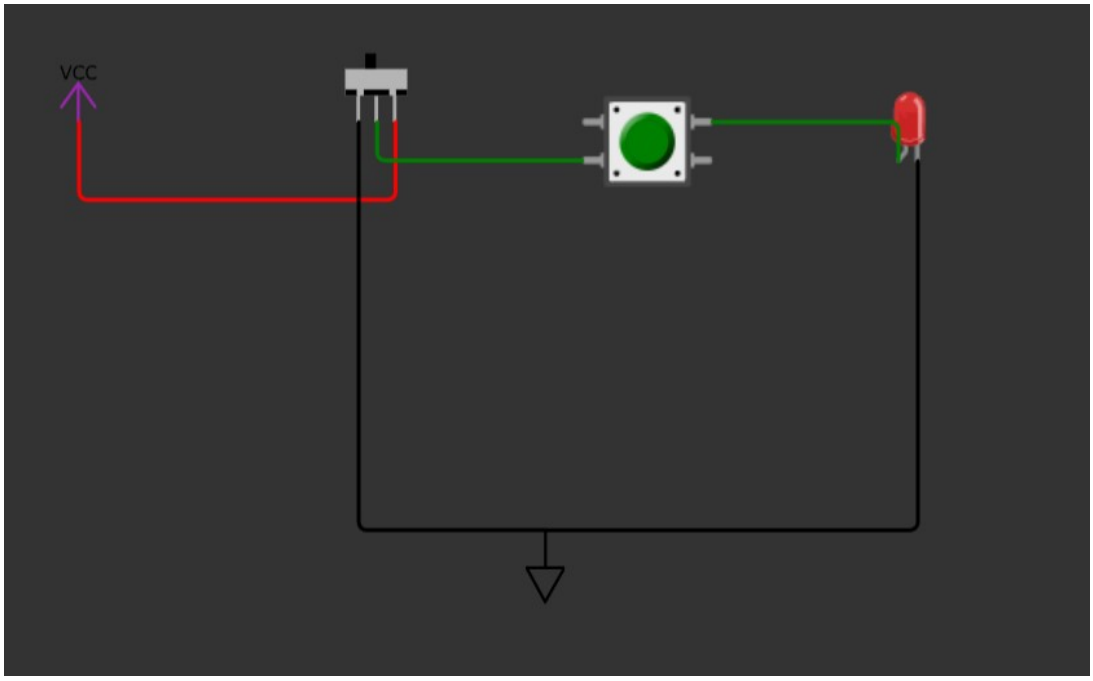
https://wokwi.com/projects/385248981790154753



3.

Solution:

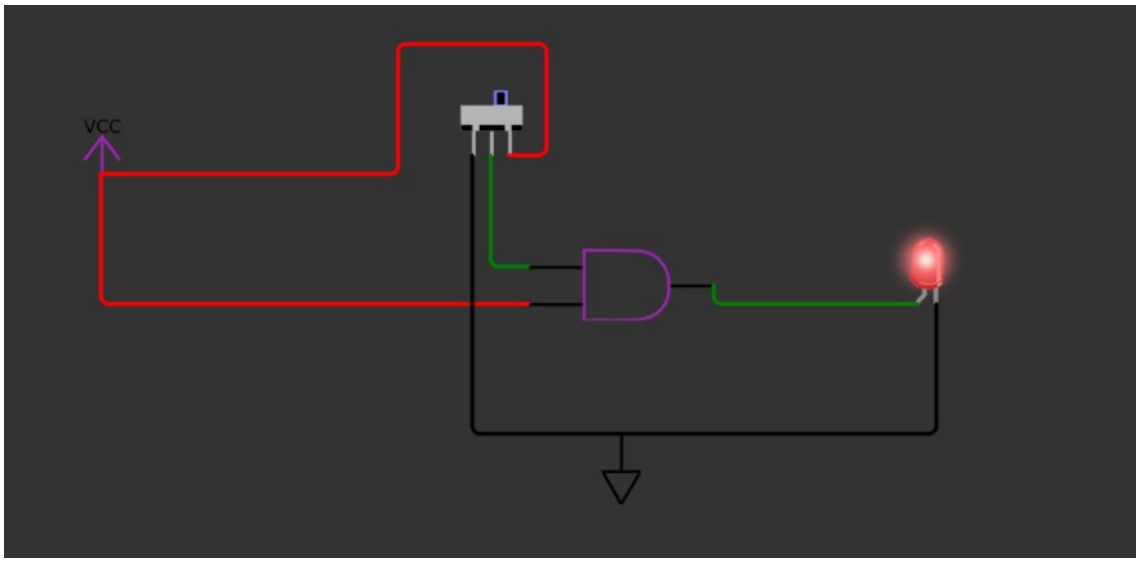
Using PUSH Button:



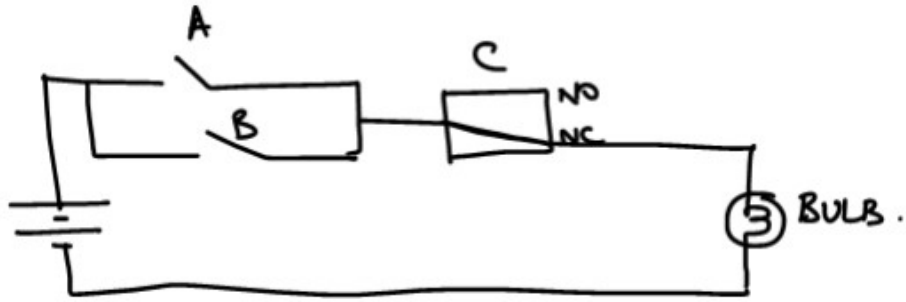
https://wokwi.com/projects/385055001421184001

A(Normally Closed)	B(Switch)	Y(OUTPUT)
1	0	0
1	1	1

From the table, we can directly say, Y(OUTPUT)=B



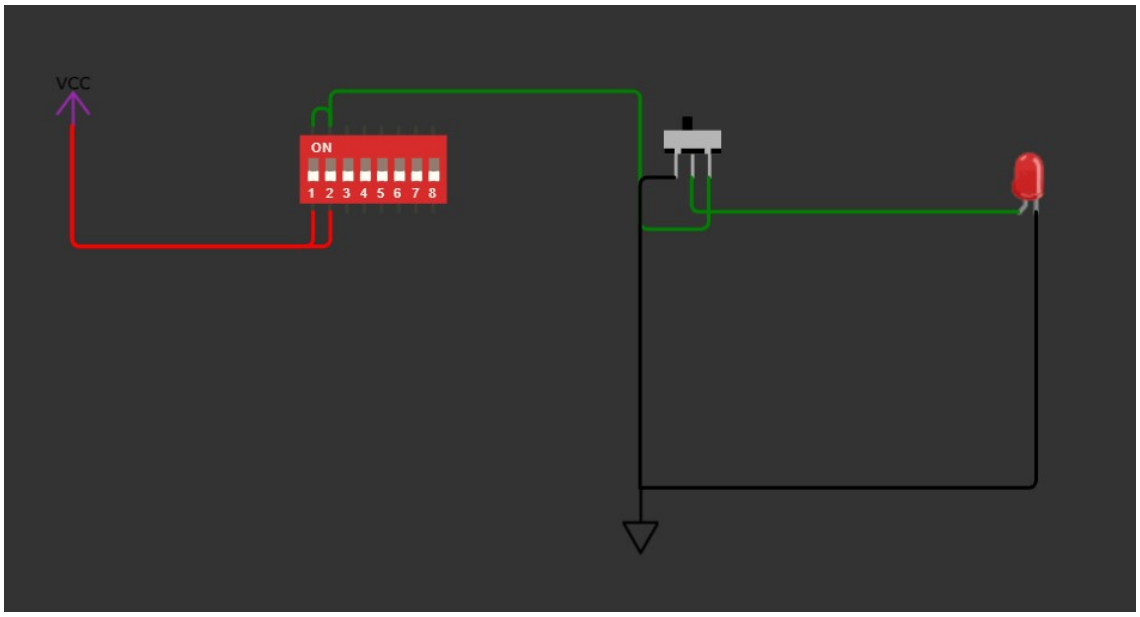
https://wokwi.com/projects/385249180502665217



4.

Solution:

Using DIP Switch:

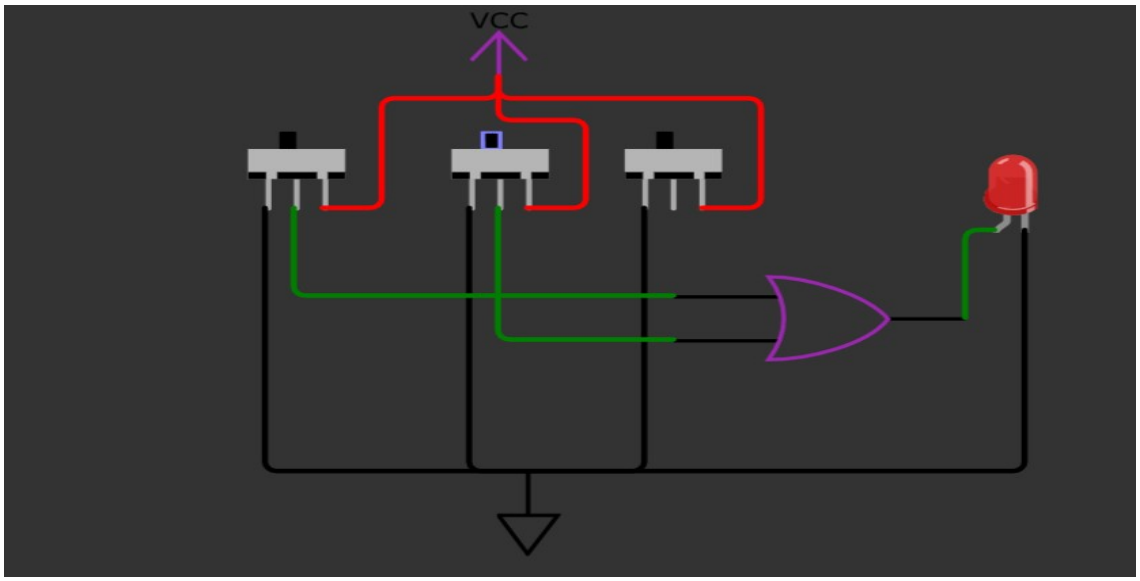


https://wokwi.com/projects/385055865452048385

A(SWITCH)	B(SWITCH)	C(Normally Close)	Y(OUTPUT)
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

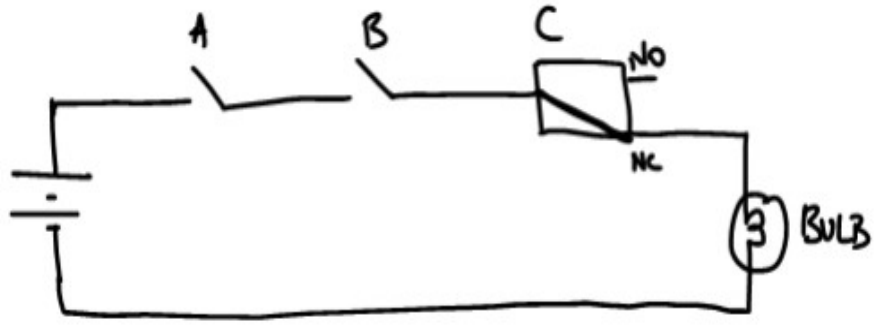
The minimized expression will be,

$$Y=A+B$$



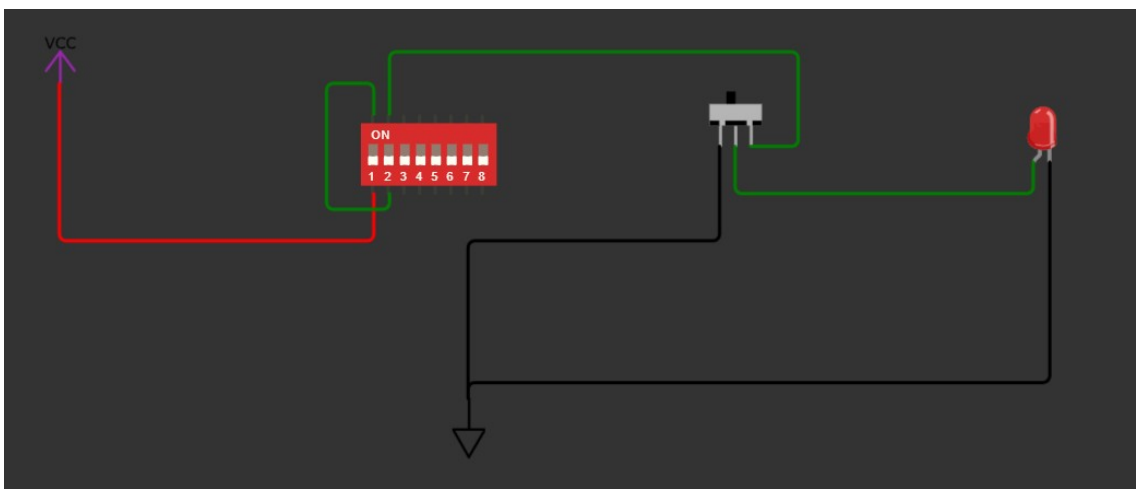
https://wokwi.com/projects/385249423451457537

5.



Solution:

Using DIP Switch:

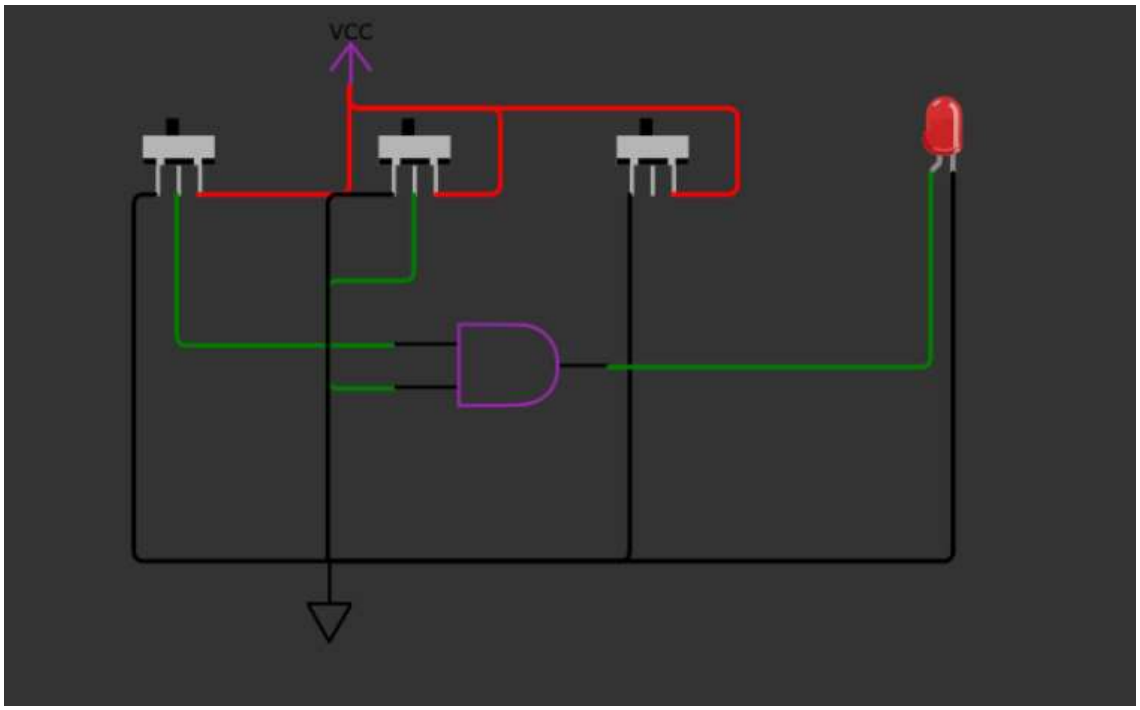


https://wokwi.com/projects/385056038126315521

A(SWITCH)	B(SWITCH)	C(Normally Close)	Y(OUTPUT)
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

The minimized expression will be,

$$Y = A.B$$



https://wokwi.com/projects/385249662378962945

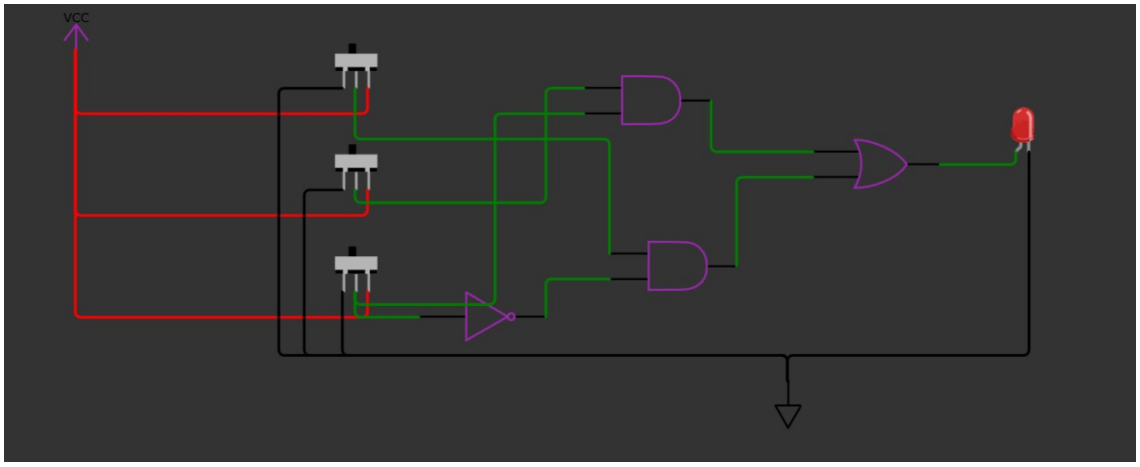
6. We have a 3 bit number ABC (A is the MSB) , whenever this number is a multiple of 3 or 4 or 7 output should be high , else the output should be low. Identify the Boolean expression for the same . Implement this Boolean expression and validate the truth table in Wokwi tool using logic gates.

A(MSB)	B	C(LSB)	Multiple of 3 or 4 or 7	Y(OUTPUT)
0	0	0	No	0
0	0	1	No	0
0	1	0	No	0
0	1	1	Yes	1
1	0	0	Yes	1
1	0	1	No	0
1	1	0	Yes	1
1	1	1	Yes	1

A,BC	B'C'	B'C	BC	BC'
A'			1	
A	1		1	1

The minimized expression will be,

Y(OUTPUT)= BC + AC’



https://wokwi.com/projects/385056703028390913

7. Given 3 inputs, design a logic circuit which decides the output as per the majority ?

(ie if majority of inputs are 1 the output should be 1 , if majority of input is 0 , the output should be 0, eg in 0 0 1 -> we have majority as 0 so output should be 0 , in 0 1 1 -> we have majority as 1 so the output should be 1). Show the K MAP identification of the logic, implement the same using logic gates in Wokwi tool.

Solution:

Using DIP Switches:

A(MSB)	B	C(LSB)	Majority 1	Y(OUTPUT)
0	0	0	No	0
0	0	1	No	0
0	1	0	No	0
0	1	1	Yes	1
1	0	0	No	0
1	0	1	Yes	1
1	1	0	Yes	1

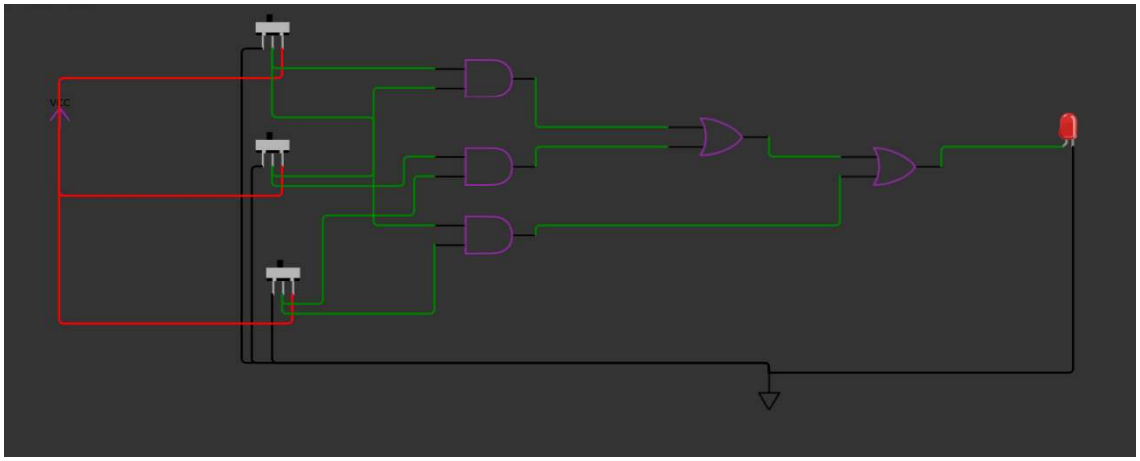
1	**1**	**1**	**Yes**	**1**

A,BC	B'C'	B'C	BC	BC'
A'			**1**	
A		**1**	**1**	**1**

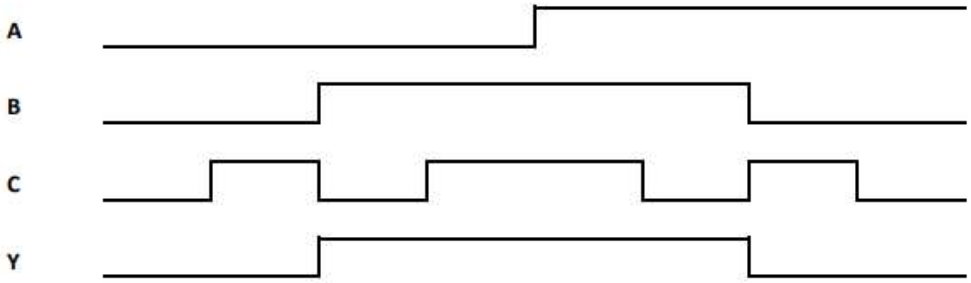
The minimized expression will be,

Y(OUTPUT) = AC+AB+BC

https://wokwi.com/projects/385057255130936321



8. Identify the logic for Y based on the inputs A, B , C given below.

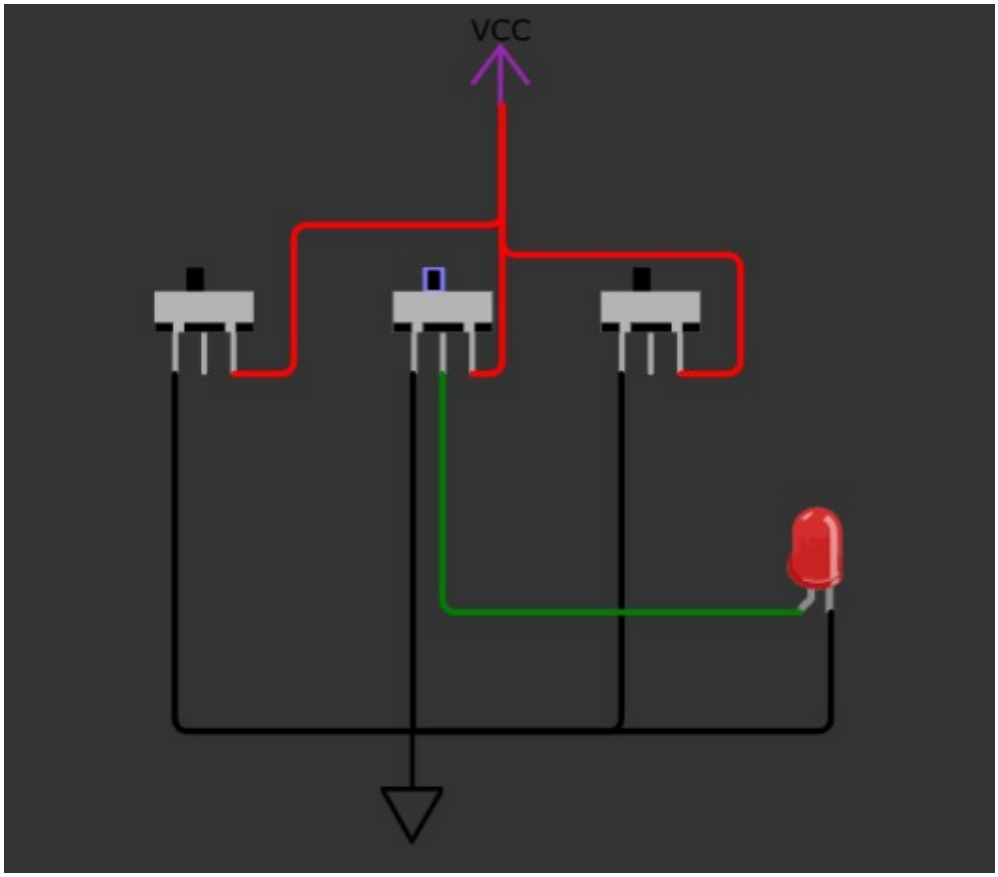


A	B	C	Y(OUTPUT)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

A,BC	B'C'	B'C	BC	BC'
A'			1	1
A			1	1

The minimized expression will be,

Y=B



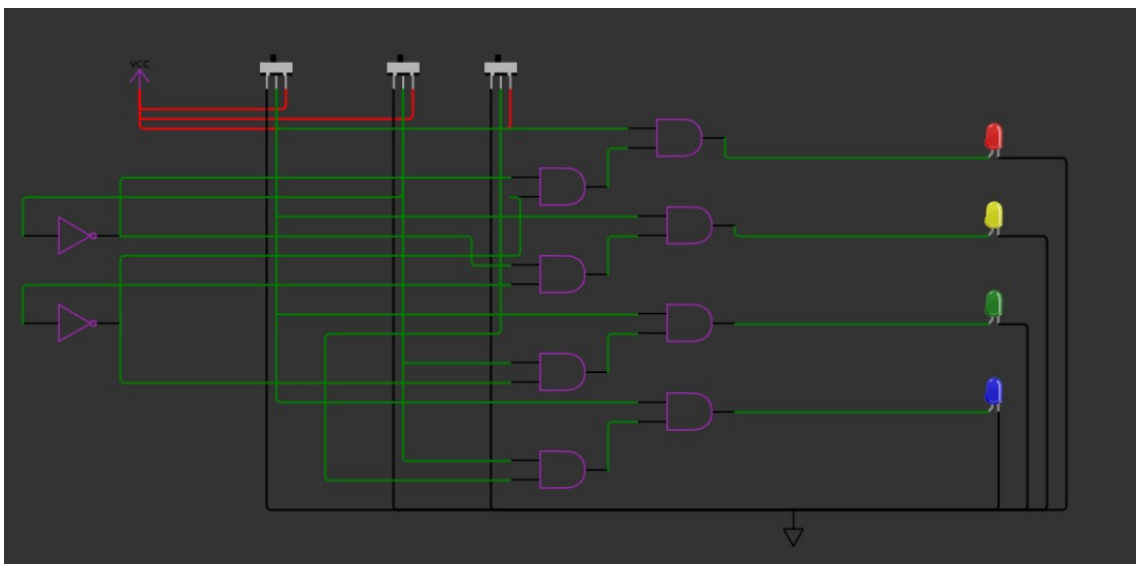
https://wokwi.com/projects/385248659991586817

9 . A,B,C are the inputs , Y0 ,Y1 , Y2 , Y3 are the outputs .

Identify the logical expression and implement the same using logic gates in Wokwi tool.

(hint : draw separate K Maps for each of the output in-terms of the inputs A,B,C).

A	B	C	Y0	Y1	Y2	Y3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

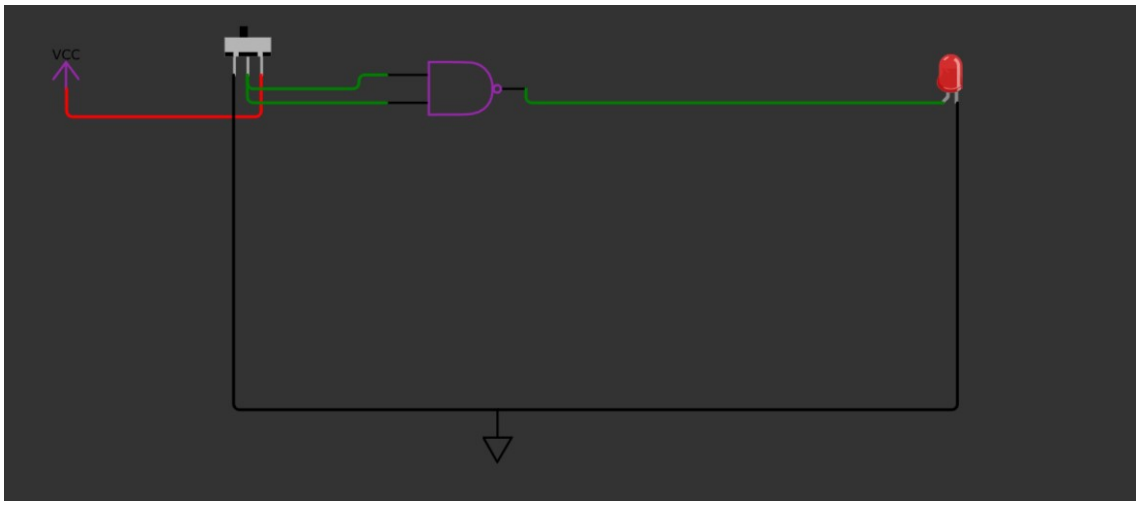


https://wokwi.com/projects/385058373201351681

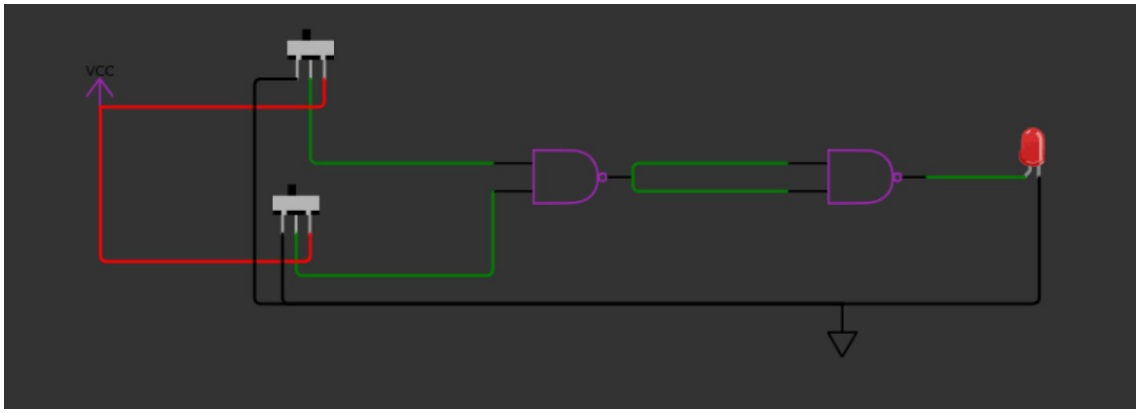
10 Realize the following Boolean expressions using 2 input NAND gate only.

a. Y = Not(A) .
b. Y = A.B
c. Y = A + B

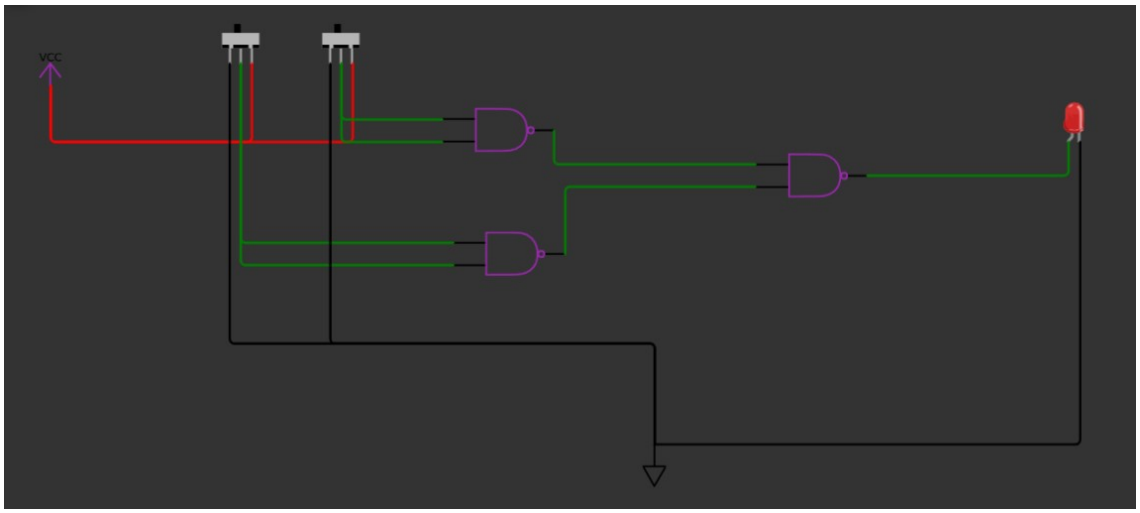
Solution:



https://wokwi.com/projects/385058545457228801



https://wokwi.com/projects/385058981242300417



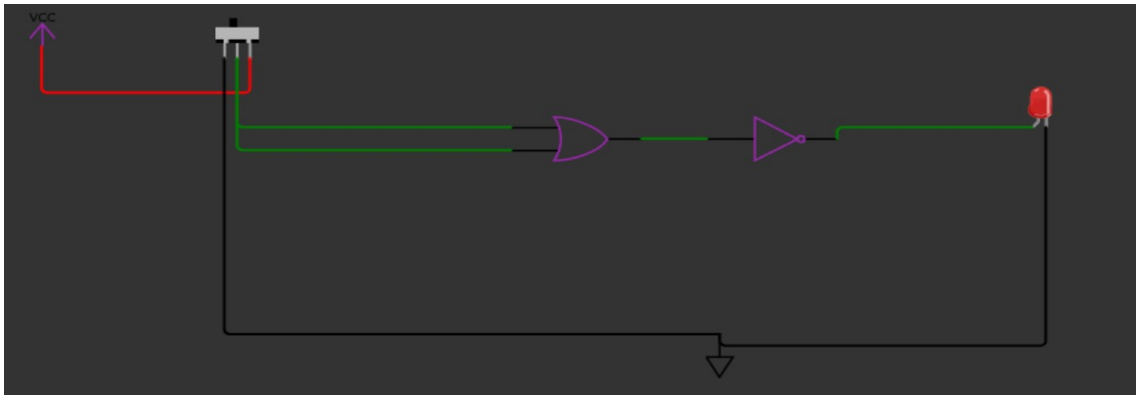
https://wokwi.com/projects/385059229999142913

11 Realize the following Boolean expressions using 2 input NOR gate only

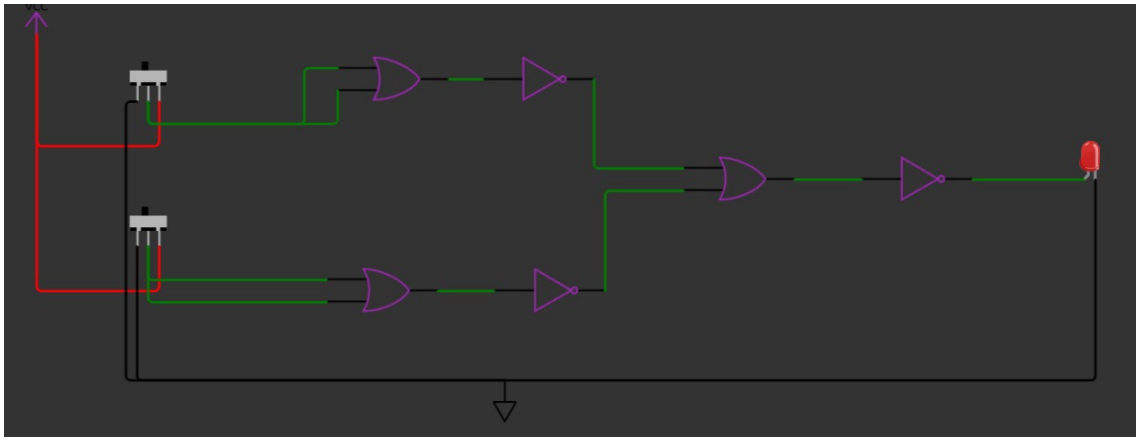
d. Y = Not(A) .

e. Y = A.B

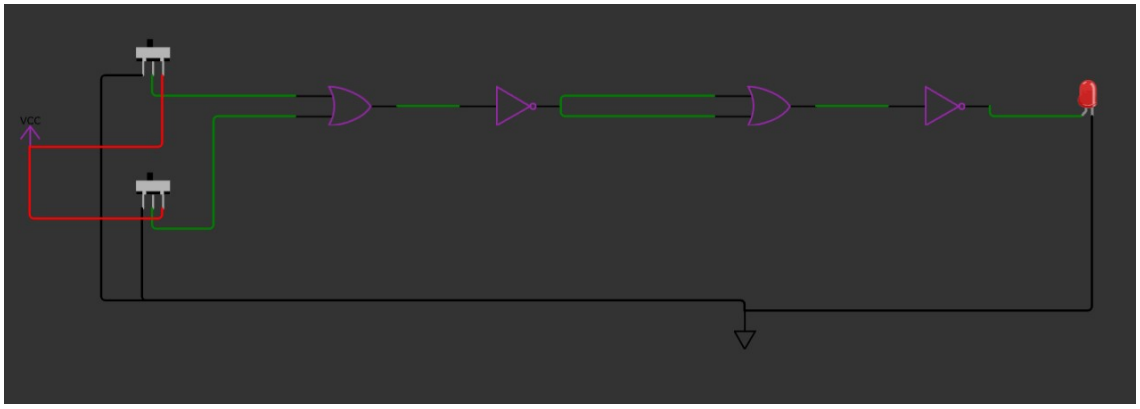
f. Y = A + B



https://wokwi.com/projects/385058784225332225



https://wokwi.com/projects/385059502880571393



https://wokwi.com/projects/385059710285766657

WEEK-6(SECTION-2)

LAB-3

LAB 3 : (Advanced K Map simplifications . Attempt this after Lecture 9 and 10.)

1) Y is 1 when A (MSB),B,C,D (LSB) = 1,2,4,5,6,7,8,9,11,12,13 else Y is 0.
Obtain the Boolean expression for Y by K Map reduction and implement the same using logic gates in wokwi tool.

Solution:

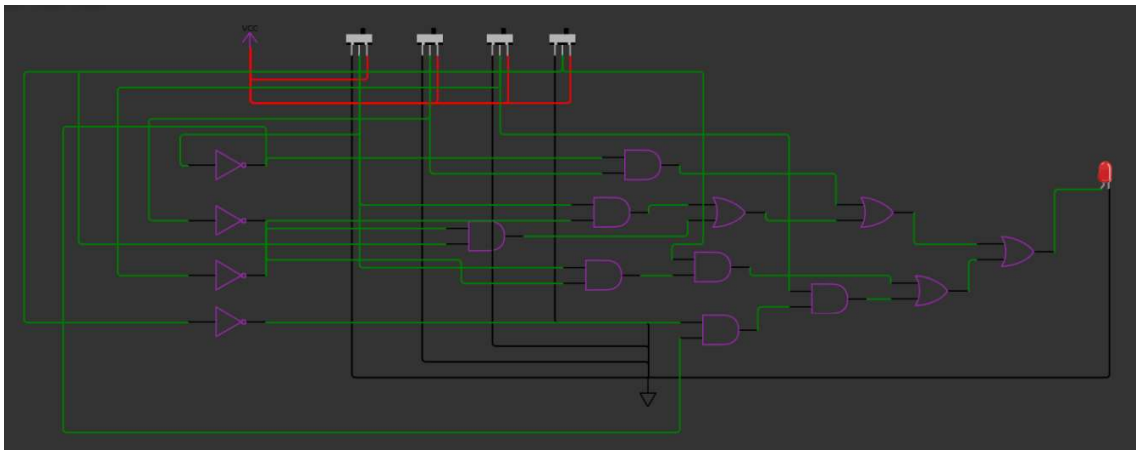
A(MSB)	B	C	D(LSB)	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

AB,CD	C'D'	C'D	CD	CD'
A'B'		1		1

A'B	1	1	1	1
AB	1	1		
AB'	1	1	1	

The minimized expression will be,

Y(OUTPUT)=A'B+C'D+AC'+AB'D+A'CD'



https://wokwi.com/projects/385247040519204865

2) Y is 1 when A(MSB),B,C,D(LSB) = 0,1,4,5,10,11,14 else Y is 0. Obtain the Boolean expression for Y by K Map reduction and implement the same using logic gates in Wokwi tool.

Solution:

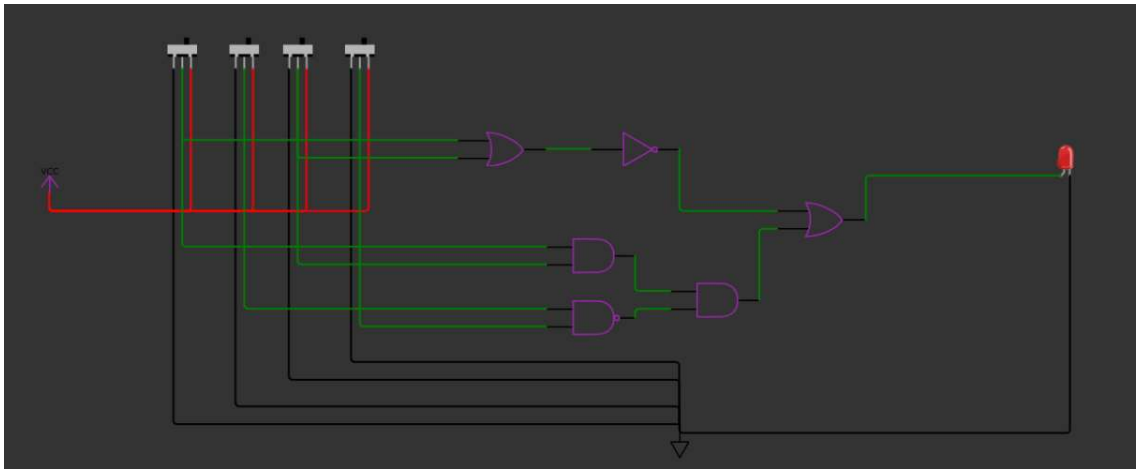
A(MSB)	B	C	D(LSB)	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0

1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

AB,CD	C'D'	C'D	CD	CD'
A'B'	1	1		
A'B	1	1		
AB				1
AB'			1	1

The minimized expression will be,

Y= A'C'+AB'C+ACD'=A'C'+AC.(B'+D')=A'C'+AC.((B.D)') = (A+C)'+AC.((B.D)')

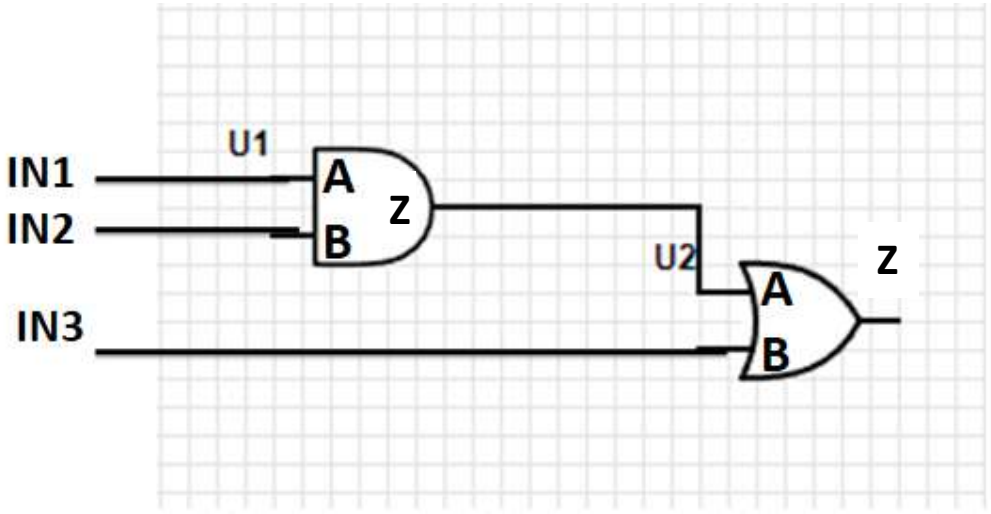


https://wokwi.com/projects/385247982387024897

WEEK-6(SECTION-3)

VERILOG-HDL(Hardware Description Language)

Verilog-HDL is a programming language to program a hardware to functionalize as per the specifications/requirements.

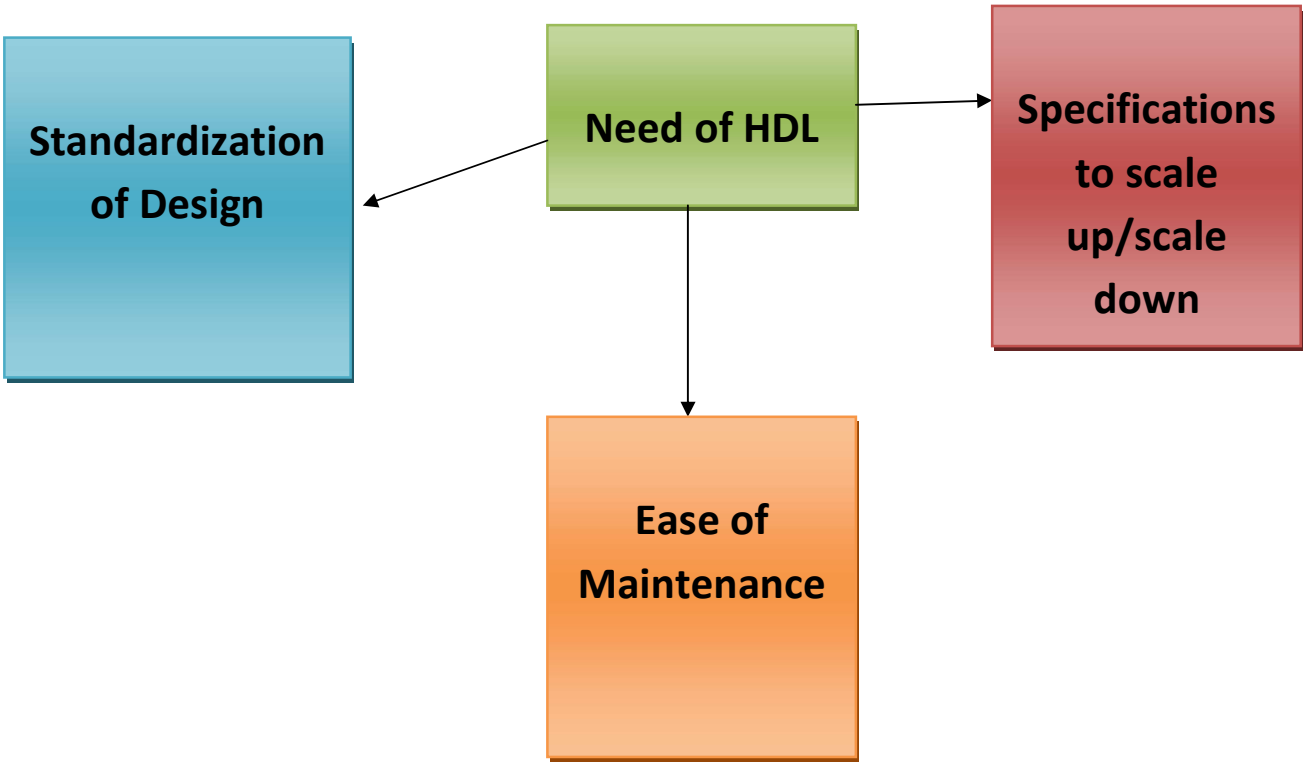


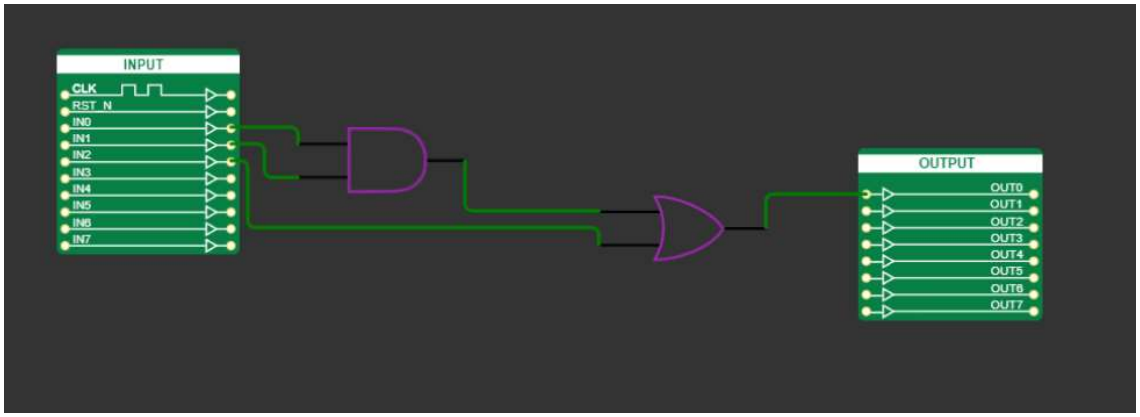
In the above schematic, the input1 and input2 is connected to A and B inputs of AND gate respectively. And input3 is connected to B input of OR gate. Output of AND gate is connected to A input of OR gate.

IN1----------A AND1
IN2----------B AND1
IN3----------B OR1
Z AND1----NET 1
A OR1------NET 1

Here, NET1 refers to the wire connecting the output of AND gate and input of OR gate. The specifications of AND1 and OR1 is necessary in order to identify specific gates as there are many no. of gates possible. The inputs of logic designs can't be left connected. The input of a logic gate can be connected from output of any other gate or input of primary inputs.

Netlist is a file which stores the specific wiring connections.





https://wokwi.com/projects/385148881045688321

```
/* Automatically generated from
https://wokwi.com/projects/385148881045688321 */

`default_nettype none

module tt_um_wokwi_385148881045688321(
```

```
  input  wire [7:0] ui_in,    // Dedicated inputs
  output wire [7:0] uo_out,    // Dedicated outputs
  input  wire [7:0] uio_in,    // IOs: Input path
  output wire [7:0] uio_out,    // IOs: Output path
  output wire [7:0] uio_oe,    // IOs: Enable path (active high: 0=input,
1=output)
  input ena,
  input clk,
  input rst_n
);
  wire net1 = ui_in[0];
  wire net2 = ui_in[1];
  wire net3 = ui_in[2];
  wire net4;
  wire net5;

  assign uo_out[0] = net4;
  assign uo_out[1] = 0;
  assign uo_out[2] = 0;
  assign uo_out[3] = 0;
  assign uo_out[4] = 0;
  assign uo_out[5] = 0;
  assign uo_out[6] = 0;
  assign uo_out[7] = 0;
  assign uio_out[0] = 0;
  assign uio_oe[0] = 0;
  assign uio_out[1] = 0;
  assign uio_oe[1] = 0;
  assign uio_out[2] = 0;
  assign uio_oe[2] = 0;
  assign uio_out[3] = 0;
  assign uio_oe[3] = 0;
  assign uio_out[4] = 0;
  assign uio_oe[4] = 0;
  assign uio_out[5] = 0;
  assign uio_oe[5] = 0;
  assign uio_out[6] = 0;
  assign uio_oe[6] = 0;
  assign uio_out[7] = 0;
  assign uio_oe[7] = 0;

  and_cell and1 (
    .a (net1),
    .b (net2),
    .out (net5)
  );
  or_cell or1 (
    .a (net5),
    .b (net3),
    .out (net4)
  );
endmodule
```

LAB-4

Boolean Logic Design Project (Attempt this after Lecture 10 and Lab 3)

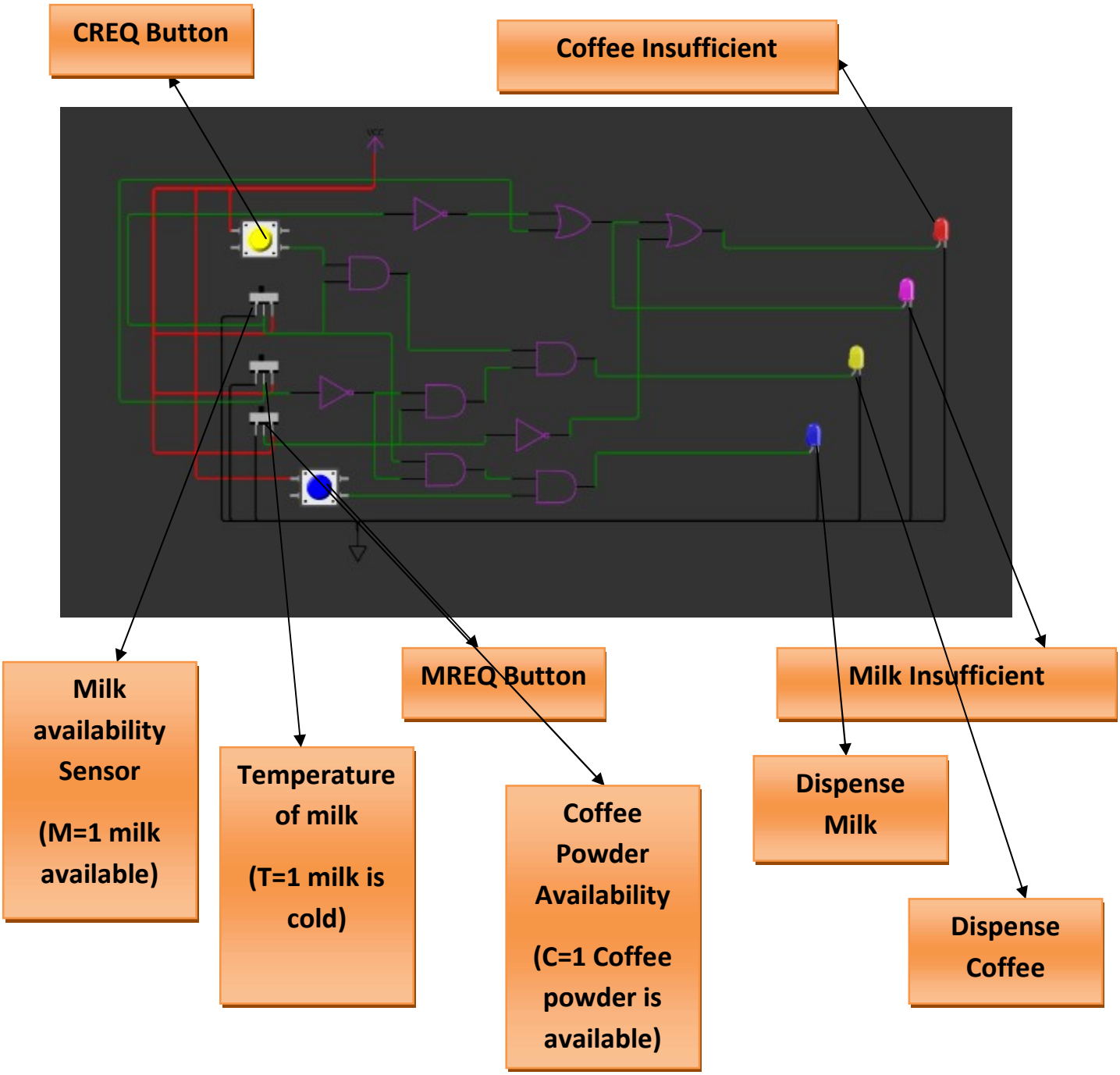
Identify the Boolean expression to implement the following requirement and implement the same using logic gates in Wokwi tool. Explain your solution

We need to design a logic for coffee / milk dispenser machine as follows.

1) We have the request to dispense Milk or Coffee from the user i.e User will press the button CREQ for Coffee , MREQ for Milk . (Note for both Coffee and Milk , we need milk to be available and also its hot.
2) Milk availability is known by a level sensor M . If M is high Milk is available else its not.
3) Similarly Milk is hot or not is known by temperature sensor T . If T is high milk is cold , if T is low , Milk is hot.
4) Coffee powder availability is known by sensor C . If C is high we have coffee powder else we don't have).
5) Coffee can be dispensed only when hot milk and coffee powder is available. Milk can be dispensed only when hot milk is available.
6) To dispense Coffee we need to generate a output Coff_Out as high (the output Coff_Out will be used in the system to open a electrically operated valve to dispense the coffee).
7) Similarly to dispense Milk , we need to generate an output Milk_Out high .
8) If Milk quantity or Coffee powder quantity is not available we need to generate an output Insuff_Qty as high.

Solution:

1. **CREQ** represents the yellow Push button. When it is pressed, coffee will be dispensed. At that time the M=1, T=0, C=1, which represnts that Milk is available (M=1)and it is hot(T=0) and coffee powder is available (C=1). All these sensors are replaced by the slide switches as indicated in the picture below.
2. **MREQ** represents the blue button. When it is pressed, milk will be dispensed. At that time, M=1 and T=0 which represents that Milk is available (M=1) and it is hot (T=0).
3. The red led is glowing when the milk is insufficient and pink led is glowing when the milk is insufficient.



Project Link:

https://wokwi.com/projects/385272659918413825

M	status	T	status	C	status	Output
0	Milk unavailable	0	Hot milk	0	Coffee Powder Unavailable	Red & Pink
0	Milk unavailable	0	Hot milk	1	Coffee Powder available	Red & Pink
0	Milk unavailable	1	Cold Milk	0	Coffee Powder Unavailable	Red & Pink
0	Milk unavailable	1	Cold Milk	1	Coffee Powder available	Red & Pink
1	Milk available	0	Hot milk	0	Coffee Powder Unavailable	Pink
1	Milk available	0	Hot milk	1	Coffee Powder available	Red and Pink off
1	Milk available	1	Cold Milk	0	Coffee Powder Unavailable	Red & Pink
1	Milk available	1	Cold Milk	1	Coffee Powder available	Red & Pink