



INDRAPRASTHA INSTITUTE *of*  
INFORMATION TECHNOLOGY  
DELHI

Department  
of  
Electronics & Communication Engineering

ECE111|Digital Circuits  
Section: A

**Dr. G.S. Visweswaran**

Lab 2:

Name: Arnav Goel  
Roll No: 2021519  
Date: 27th Jan 2022

## Aim:

- A) To prove De Morgan's Laws i.e.  $(A+B)' = A'B'$  and  $(A \cdot B)' = A' + B'$  using Combinational Circuits.
- B) To make a Binary Half Adder using Gates.
- C) To make a Binary Full Subtractor using Gates.

## Components/ICs Used:

### A) De Morgan's Laws:

- 1) Power Supply
- 2) Wires
- 3) Slideswitch
- 4) NOR Gate IC
- 5) NAND Gate IC
- 6) OR Gate IC
- 7) AND Gate IC
- 8) Resistor
- 9) LED

### B) Binary Half Adder:

- 1) AND Gate IC
- 2) XOR Gate IC
- 3) Power Supply
- 4) Slideswitch
- 5) Resistor
- 6) LEDs
- 7) Wires

### C) Binary Full Subtractor:

- 1) OR Gate IC
- 2) AND Gate IC
- 3) XOR Gate IC
- 4) Slideswitch
- 5) Hex Invertor
- 6) Power Supply
- 7) Wires
- 8) LEDs
- 9) Resistors

Link of TinkerCAD Workspace:

A) De Morgan's Laws:

- a) <https://www.tinkercad.com/things/3XfIpY1ZVAc-neat-trug/editel?sharecode=tOGdfcE6LfYkafb6dEWLdnhmKsjV2j01NSxmmHqPDYg>
- b) [https://www.tinkercad.com/things/5oBWIsZaFay-fantabulous-bojo/editel?sharecode=Z650ZGSTwAOZX\\_CkLwVgQ7txkU3POUfrD6Sev5PtRas](https://www.tinkercad.com/things/5oBWIsZaFay-fantabulous-bojo/editel?sharecode=Z650ZGSTwAOZX_CkLwVgQ7txkU3POUfrD6Sev5PtRas)

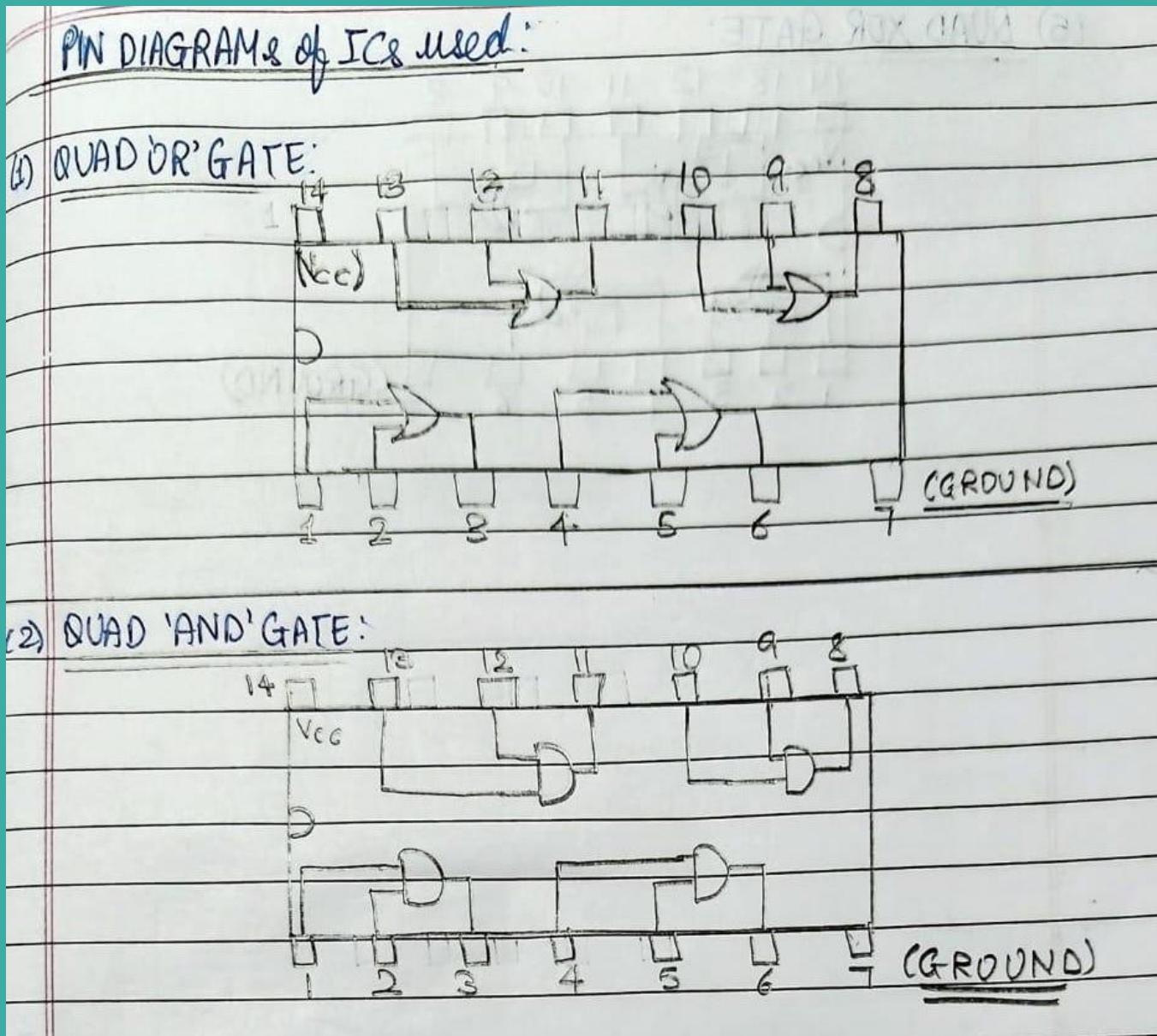
B) Binary Half Adder:

- [https://www.tinkercad.com/things/3000BG970A0-frantic-densor-jaiks/editel?sharecode=PWwThqat6R6Buzwrd3dwKFkbJfbLEV5eVr-617g\\_fU](https://www.tinkercad.com/things/3000BG970A0-frantic-densor-jaiks/editel?sharecode=PWwThqat6R6Buzwrd3dwKFkbJfbLEV5eVr-617g_fU)

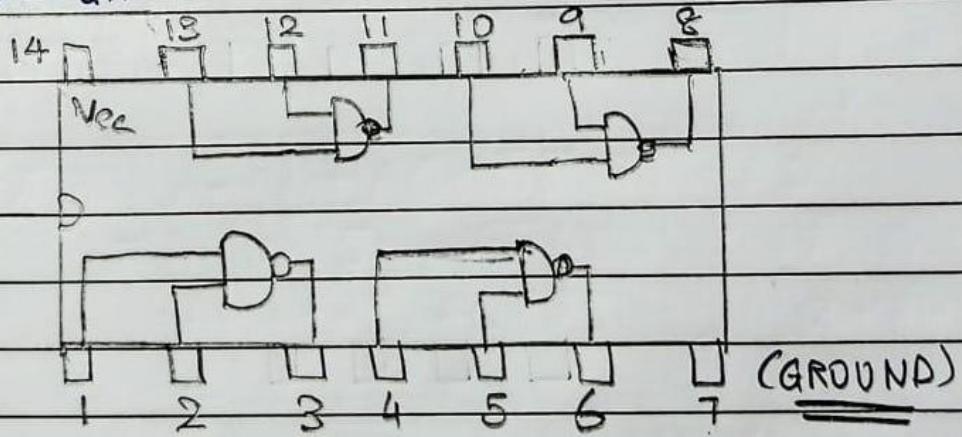
C) Binary Full Subtractor:

- [https://www.tinkercad.com/things/dTm4rX6Guv9-cool-kieran-hillar/editel?sharecode=VzxkLsvgGYL1juhjlC\\_N-gpraviCJEwNzy9NpBdHtdg](https://www.tinkercad.com/things/dTm4rX6Guv9-cool-kieran-hillar/editel?sharecode=VzxkLsvgGYL1juhjlC_N-gpraviCJEwNzy9NpBdHtdg)

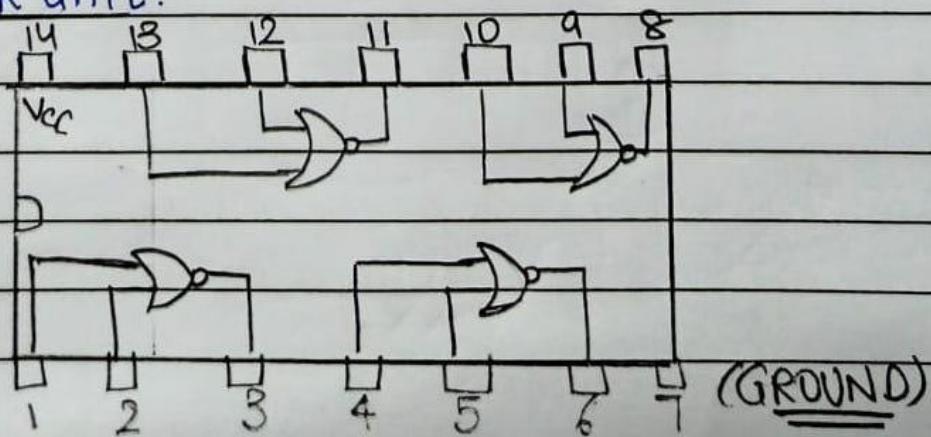
Pin Diagram of the IC (If Applicable):



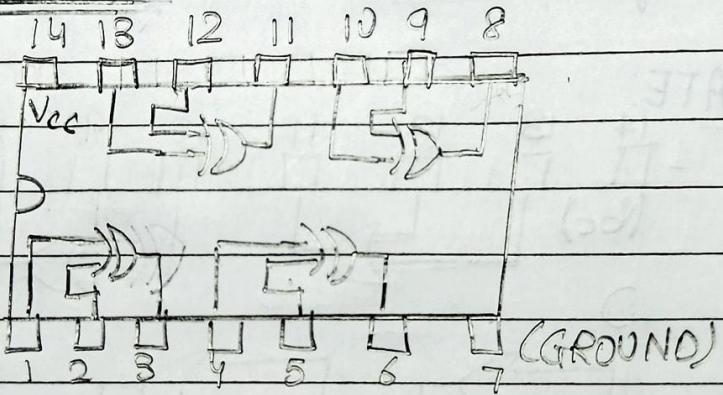
3) QUAD 'NAND' GATE:



4) QUAD 'NOR' GATE:



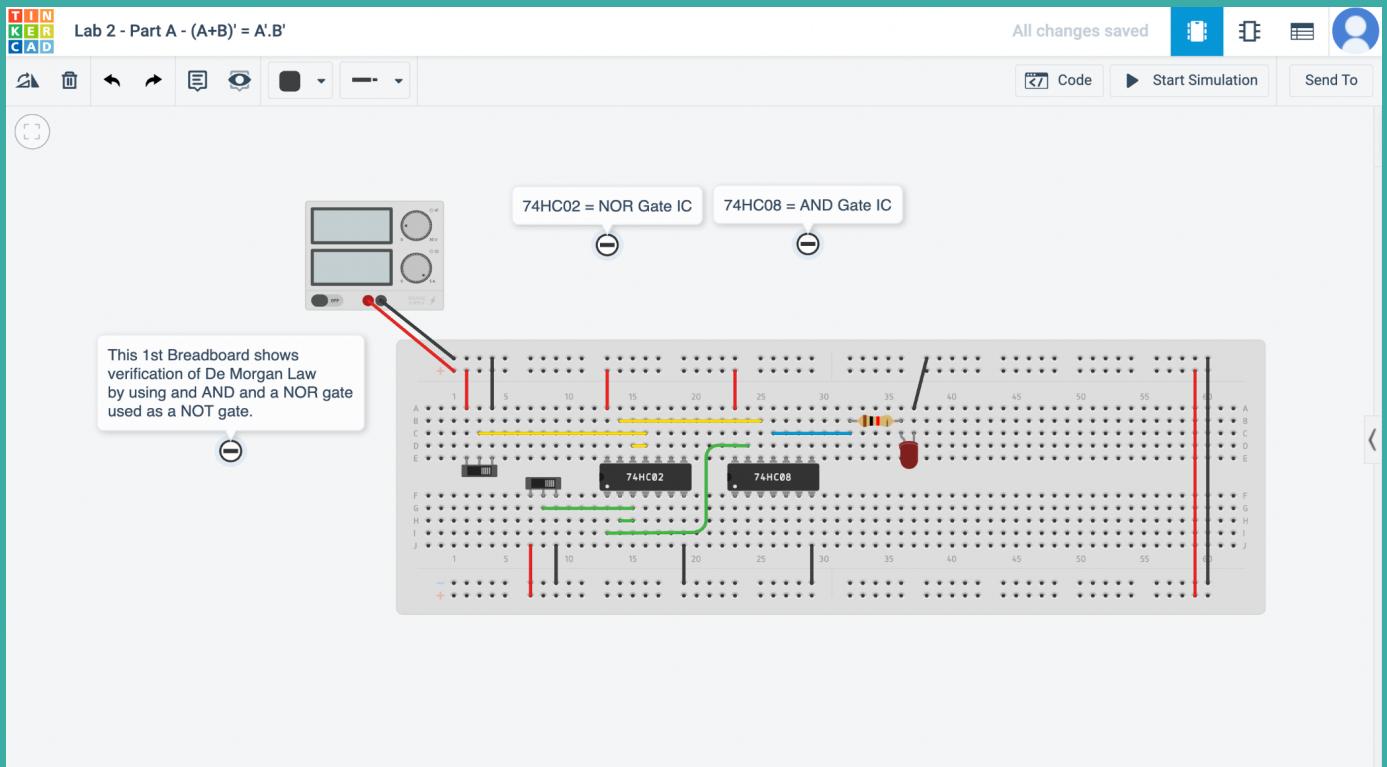
5) QUAD XOR GATE:



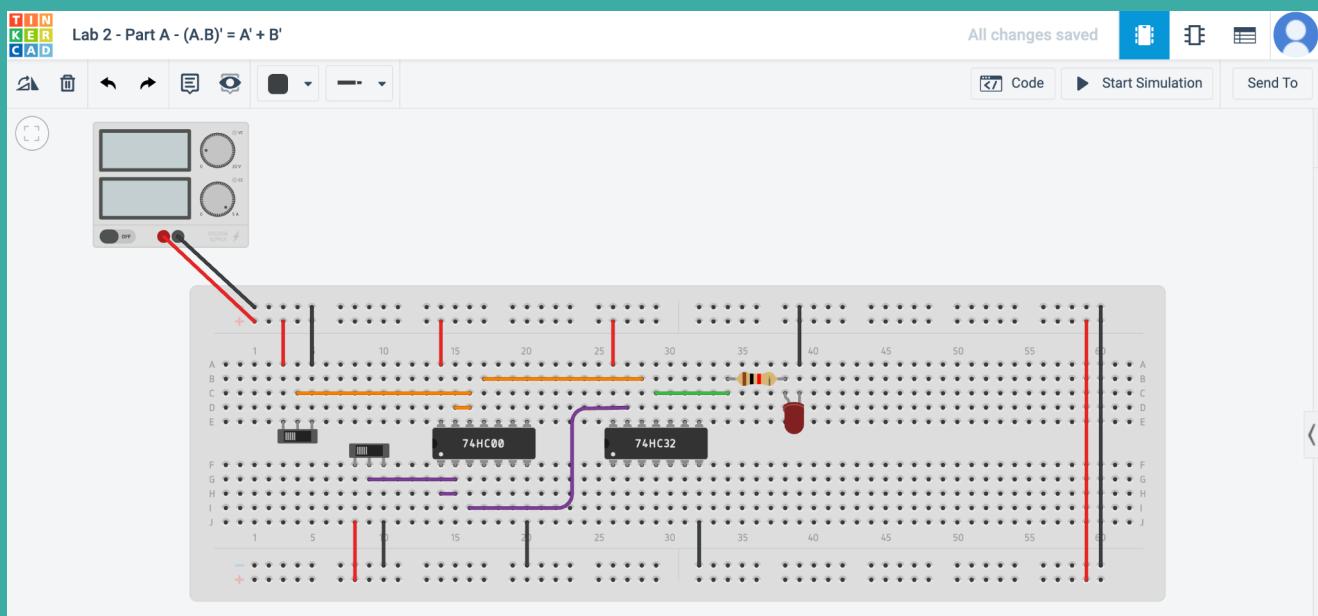
## Circuit Diagram: (Screenshot of Circuitverse/TinkerCAD workspace):

### A) De Morgan's Laws:

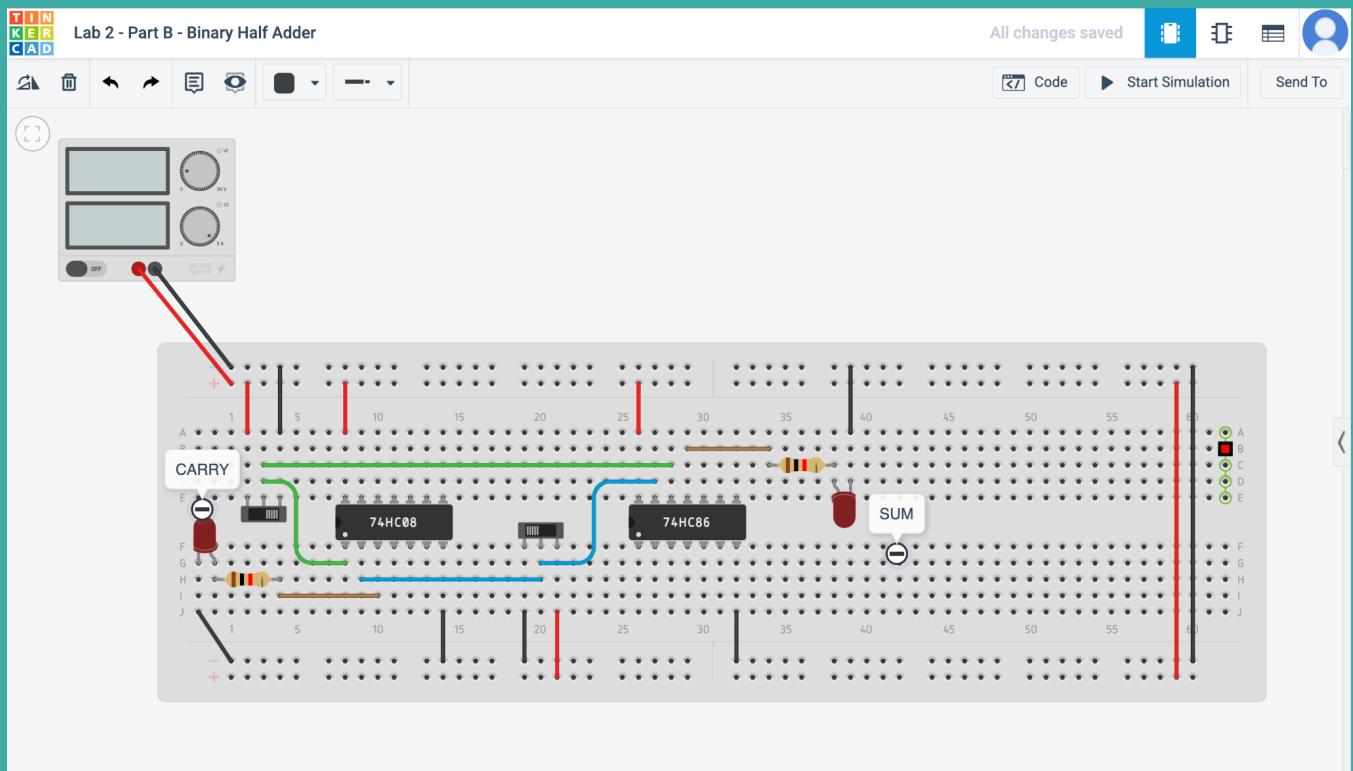
#### a) 1st Theorem - $((A+B)') = A'.B'$ :



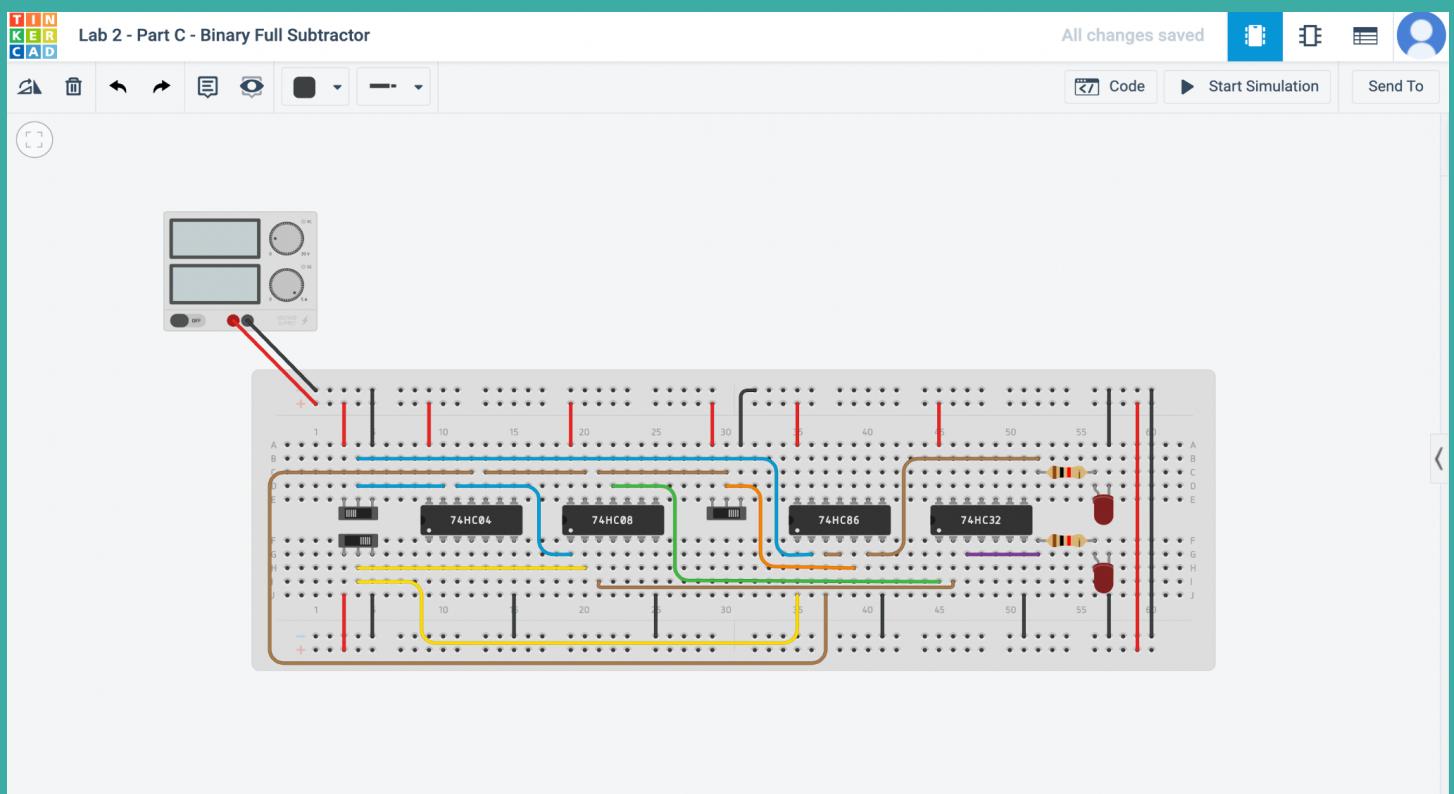
#### b) 2nd Theorem - $((A.B)') = A' + B'$ :



## B) Binary Half Adder:



## C) Binary Full Subtractor:



Truth Tables:

A) De Morgan's Laws:

$$1) (A+B)' = A'B'$$

A	B	$(A+B)'$	$A'B'$
1	1	0	0
1	0	0	0
0	1	0	0
0	0	1	1

NOR Gate:

A	B	NOR Output
1	1	0
1	0	0
0	1	0
0	0	1

---

$$2) (A \cdot B)' = A' + B'$$

A	B	$(A \cdot B)'$	$A' + B'$
1	1	0	0
1	0	1	1
0	1	1	1

0	0	1	1
---	---	---	---

NAND Gate:

A	B	NAND Output
1	1	0
1	0	1
0	1	1
0	0	1

B) Binary Half Adder:

A	B	CARRY	SUM
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0

C) Binary Full Subtractor:

A	B	BIN	BORROW	DIFFERENCE
1	1	1	1	1
1	1	0	0	0

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

---

### Observations/Results:

#### A) De Morgan's Laws:

We proved De Morgan's Law that is:

- 1)  $(A+B)' = A'.B'$
- 2)  $(A.B)' = A' + B'$

#### B) Binary Half-Adder:

We verified the operation of a Binary Half-Adder and found the logical expressions for SUM and CARRY to be the following:

- 1)  $SUM = A'B + AB' = A \oplus B \longrightarrow (XOR)$
- 2)  $CARRY = A.B$

#### C) Binary Full Subtractor:

We experimentally verified the truth table for a Binary Full Subtractor and found the logical expressions for the DIFFERENCE AND BORROW outputs as:

- 1)  $DIFFERENCE = A \oplus B \oplus BIN$
- 2)  $BORROW (Bout) = A'.B + (A \oplus B).BIN$

### Application:

#### A) De Morgan's Laws

De Morgan's Laws are extensively used to simplify Boolean Expressions and also used for NAND and NOR Gates. Thus they also find their use in Programming.

**B) Binary Half Adder:**

It is used to perform addition on binary bits in the Arithmetic and Logic Unit (ALU) in the computer. It is also used in digital calculators.

**C) Binary Full Subtractor:**

It is used in the Arithmetic and Logic Unit (ALU) of the computer to perform subtraction on Binary Bits. It is also used in digital calculators.

**Problem Statement:**

A private collector has received a valuable gemstone which they wish to put on display. Due to its value, the collector has proposed an idea to prevent thieves from stealing the gemstone or escaping after attempting to steal it. The gemstone rests on top of a pressure plate on a pedestal, surrounded by a glass case. If glass is removed with weight intact or weight is removed from the pressure plate with glass not removed a warning is issued. If the glass is broken and the gemstone's weight is removed from the pressure plate, the alarm is set on.

Design the logic circuit required for this trap, modelling the glass case, pressure plate and steel barrier as follows:

- Glass case (input J): 0 – Glass not removed 1 – Glass removed
- Pressure plate (input K): 1 – Weight removed 0 – Weight applied
- Alarm (output X): 0 – Alarm off 1 – Alarm On
- Warning (output Y): 0 – Warning off 1 - Warning On

**Problem Statement Solution:**

From the problem statement it is clear that the Warning is issued either when glass is removed with weight intact (i.e.  $J = 1, K = 0$ ) or weight is removed and glass isn't (i.e.  $J = 0, K = 1$ ). This is the example of a XOR

Gate.

**Therefore,  $J \oplus K = Y \longrightarrow (1)$**

Alarm is set on (i.e.  $X = 1$ ) only when the glass is broken AND the gemstone's weight is removed (i.e.  $J = 1, K = 1$ ). This is clearly an example of an AND gate. **Therefore,  $J \cdot K = X \longrightarrow (2)$**

The truth table is as follows:

J	K	Y	X
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

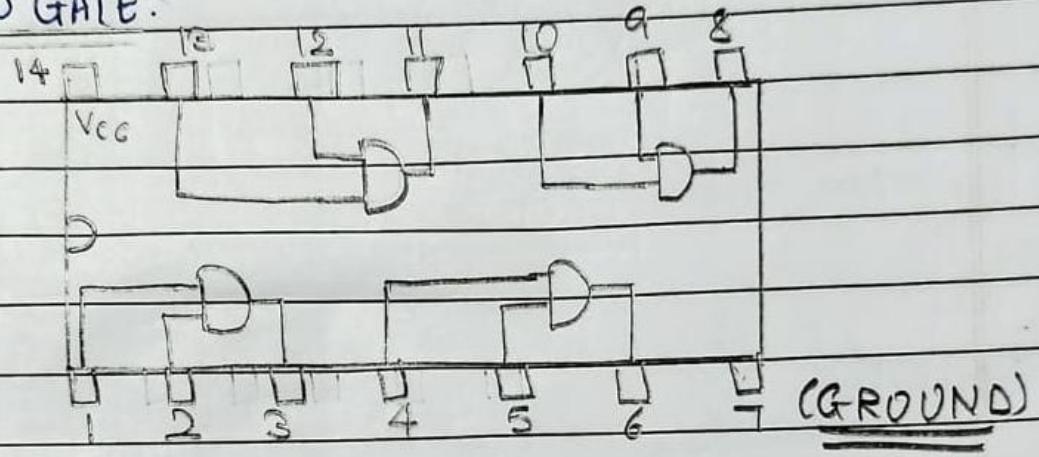
**Thus, this setup is clearly an example of a Binary Half Adder with the Warning (Y) as SUM and Alarm (X) as CARRY.**

**Components Used:**

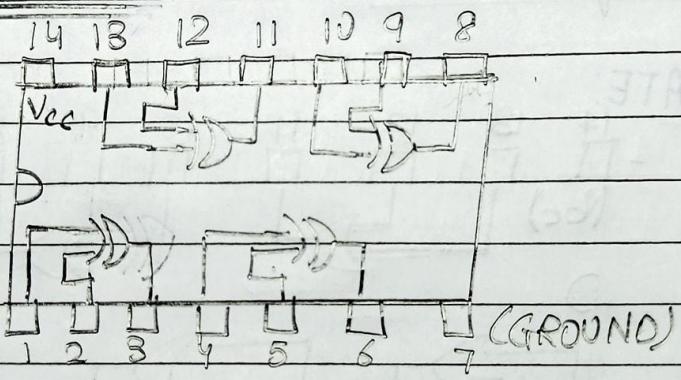
- 1) AND Gate IC
- 2) XOR Gate IC
- 3) Resistors
- 4) LEDs
- 5) Resistors
- 6) Wires
- 7) Power Supply
- 8) Slideswitch

**Pin Diagrams of ICs Used:**

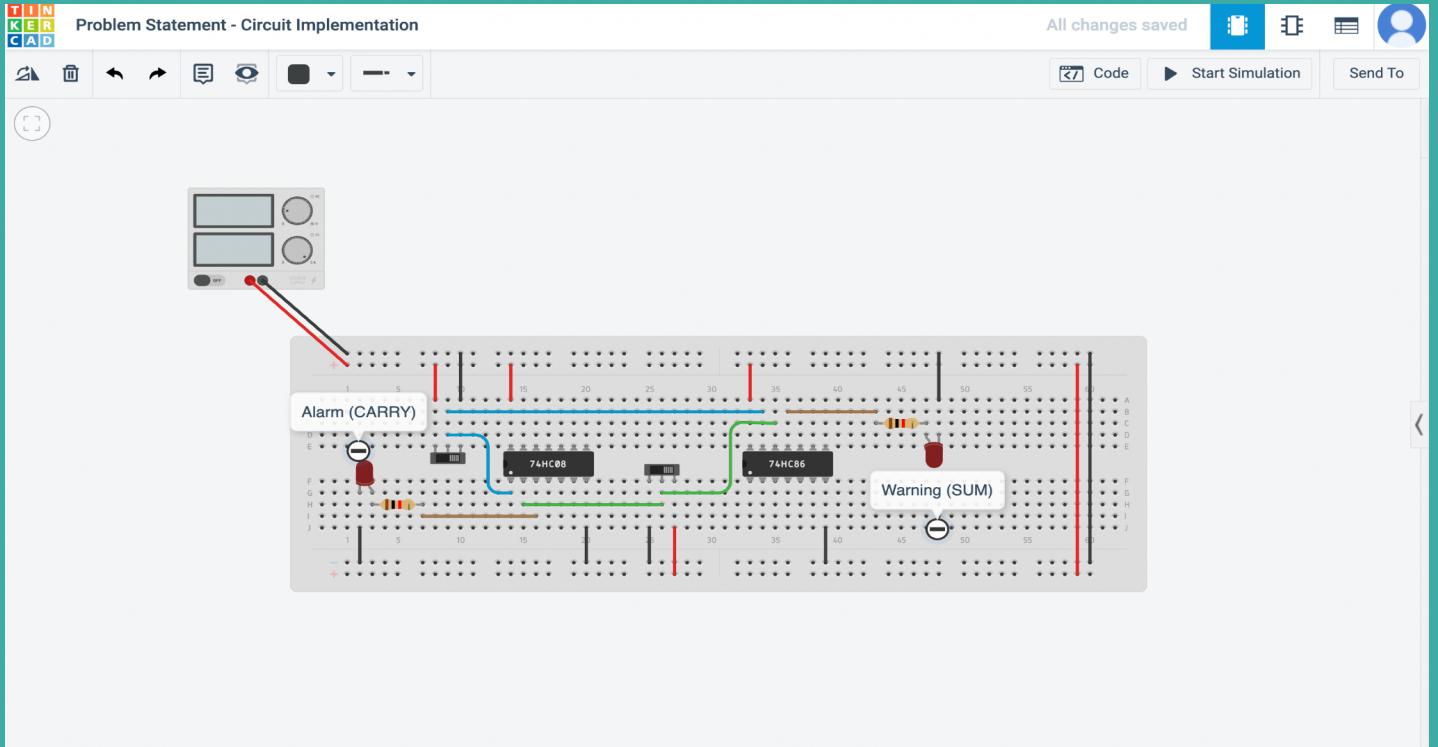
→ QUAD 'AND' GATE:



(5) QUAD XOR GATE:



## Circuit Implementation:



## TinkerCad Link for the Problem Statement Solution:

- [https://www.tinkercad.com/things/ey8KhucDN4H-sizzling-allis/editel?sharecode=m9updi\\_GI6FYmMvlWYF8fKh99A7fqWpXPI\\_y-xOjEjq](https://www.tinkercad.com/things/ey8KhucDN4H-sizzling-allis/editel?sharecode=m9updi_GI6FYmMvlWYF8fKh99A7fqWpXPI_y-xOjEjq)



INDRAPRASTHA INSTITUTE *of*  
INFORMATION TECHNOLOGY  
DELHI

Department  
of  
Electronics & Communication Engineering

ECE111|Digital Circuits  
Section: A

**Dr. G.S. Visweswaran**

Lab\_3:

Student Name: Arnav Goel  
Roll No: 2021519  
Date: 03/02/2022

## Aim:

In this experiment, we will design 2x1, 4x1 and 8x1 Multiplexers having Enable pin, using basic logic gates in TinkerCad and test its performance for various conditions.

### A) 2x1 MUX:

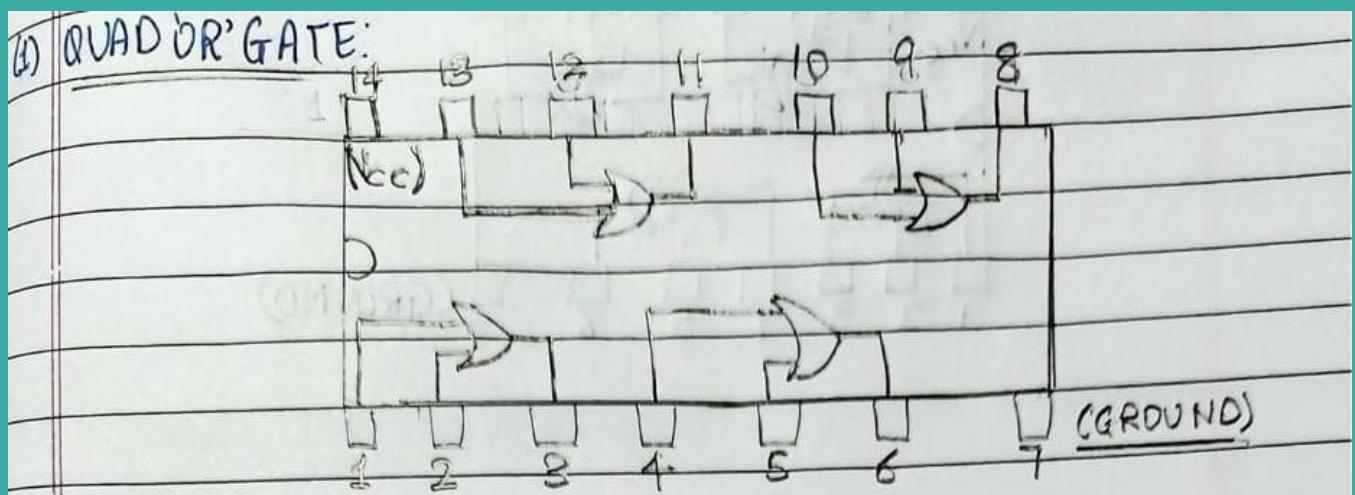
#### 1) Components/ICs Used:

- a) OR Gate IC
- b) AND Gate IC
- c) HEX Inverter
- d) Resistors
- e) LEDs
- f) Power Supply
- g) Wires
- h) Breadboard
- i) Slideswitch

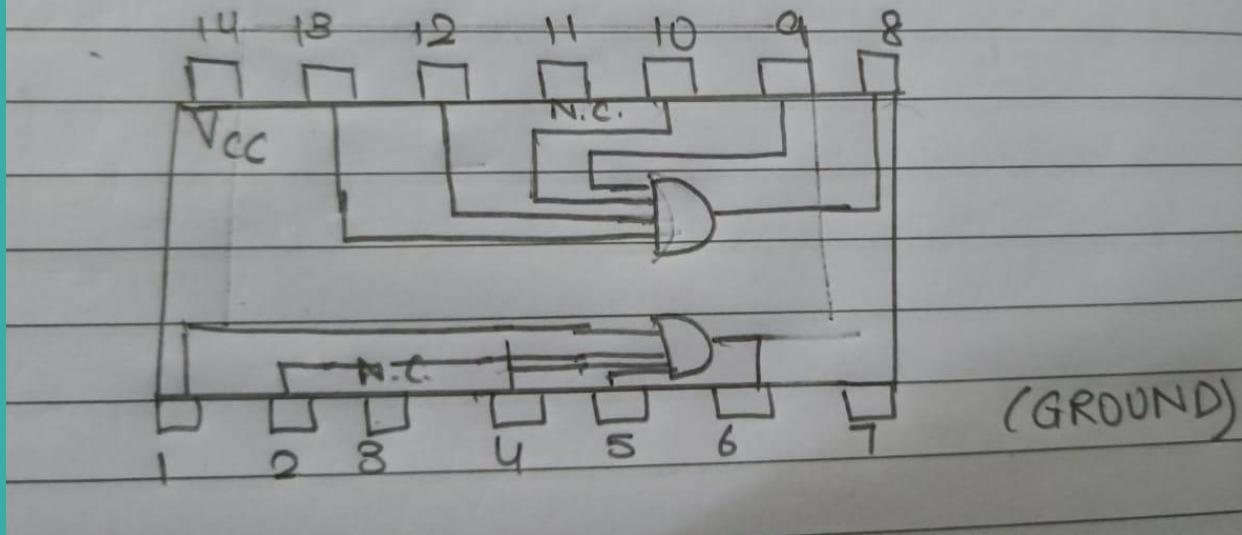
#### 2) Link of TinkerCAD Workspace :

- <https://www.tinkercad.com/things/gcrHls9CXS9-lab-3-2x1/edit#sharecode=C4vAsSRCYCoMbZD-nGdJWIVz5tKB6232vpkGYze cdwQ>

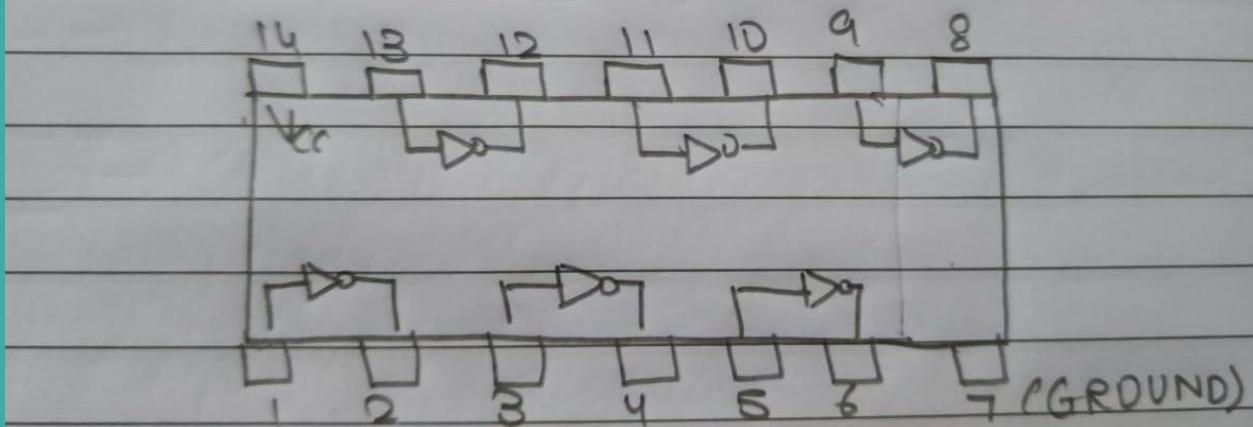
#### 3) Pin Diagram of the IC :



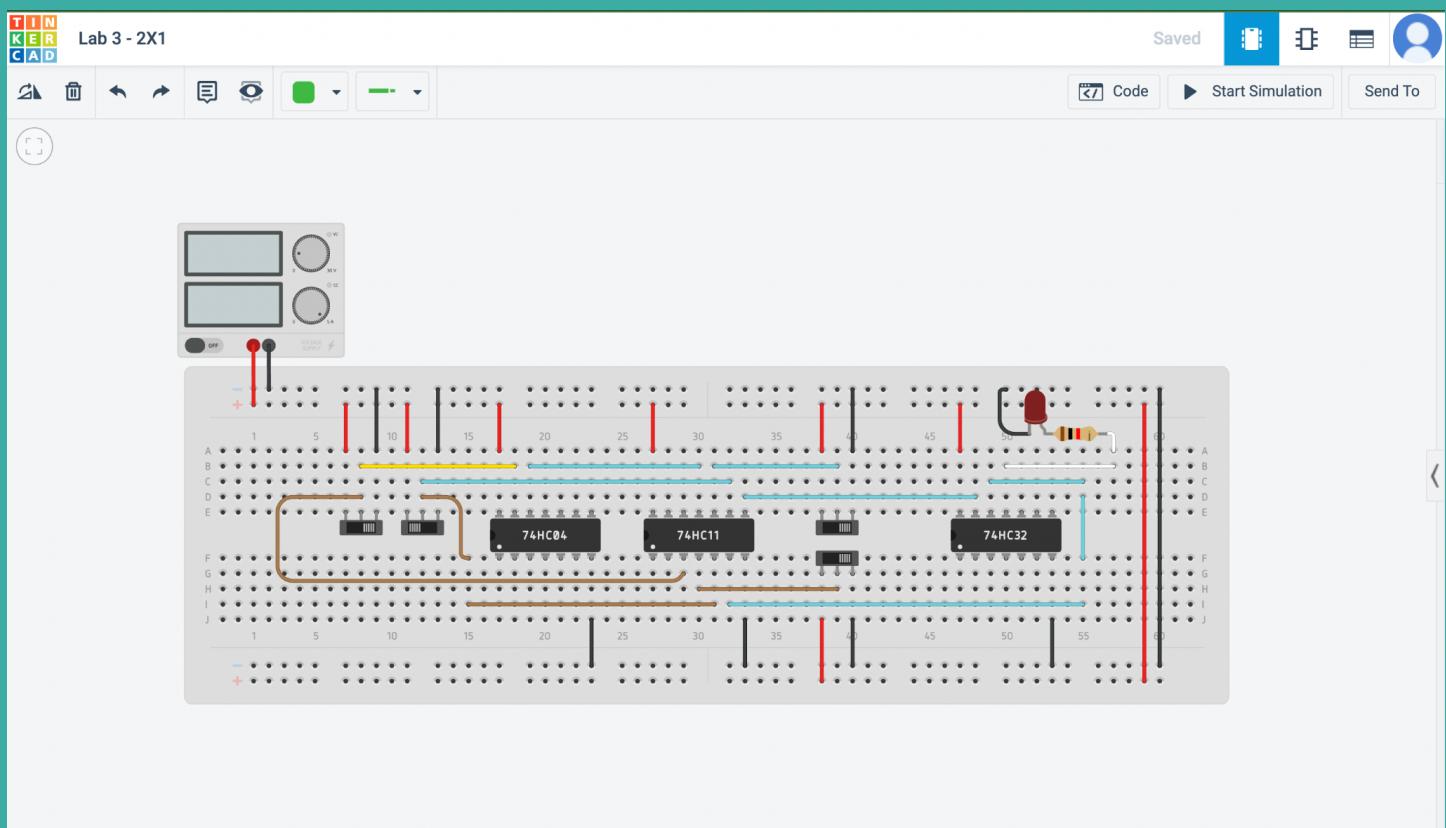
(2) DUAL 4-INPUT AND GATE:



(3) HEX INVERTER:



4) Circuit Diagram: (Screenshot of TinkerCAD workspace):



### 5) Truth Table:

S	EN	Y
X	0	0
0	1	I0
1	1	I1

### 6) Observations/Results:

We made a 2x1 MUX using AND, OR and NOT Gates with EN and Signal Line.

## **B) 4x1 MUX:**

### **1) Components/ICs Used:**

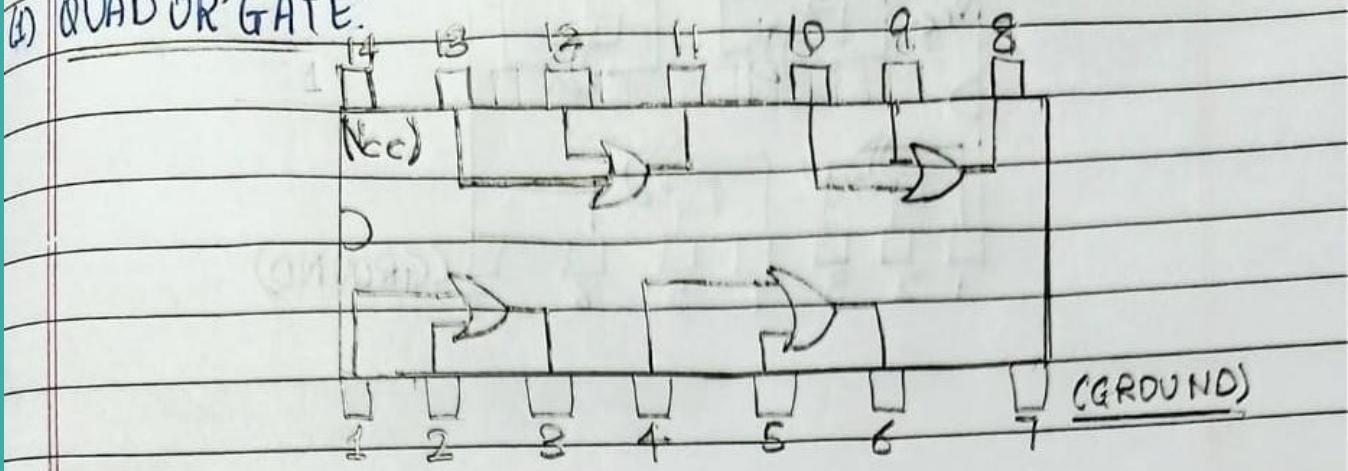
- a) OR Gate IC
- b) AND Gate IC
- c) HEX Inverter
- d) Resistors
- e) LEDs
- f) Power Supply
- g) Wires
- h) Breadboard
- i) Slideswitch

### **2) Link of TinkerCAD Workspace:**

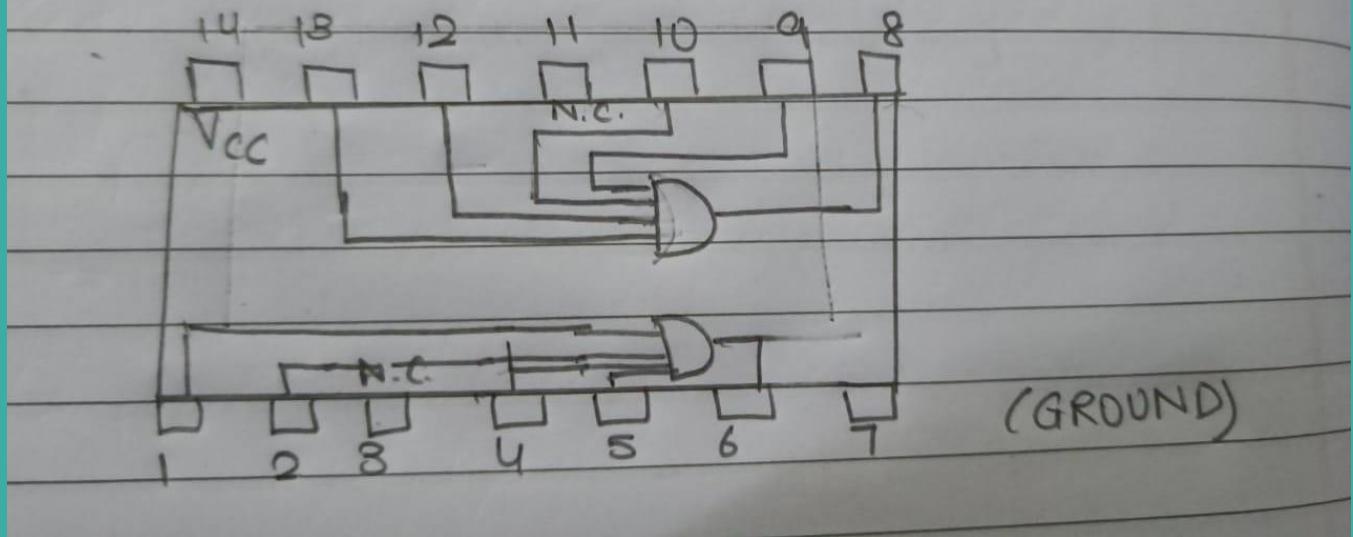
- [https://www.tinkercad.com/things/3SrjAGvdb4u-4x1-mux-lab-3/  
editel?sharecode=qs3iCdzSOHK9AU\\_KDzBQyKSSkfmA8wxtV9a  
r3ss8Us8](https://www.tinkercad.com/things/3SrjAGvdb4u-4x1-mux-lab-3/editel?sharecode=qs3iCdzSOHK9AU_KDzBQyKSSkfmA8wxtV9ar3ss8Us8)

### **3) Pin Diagram of the ICs:**

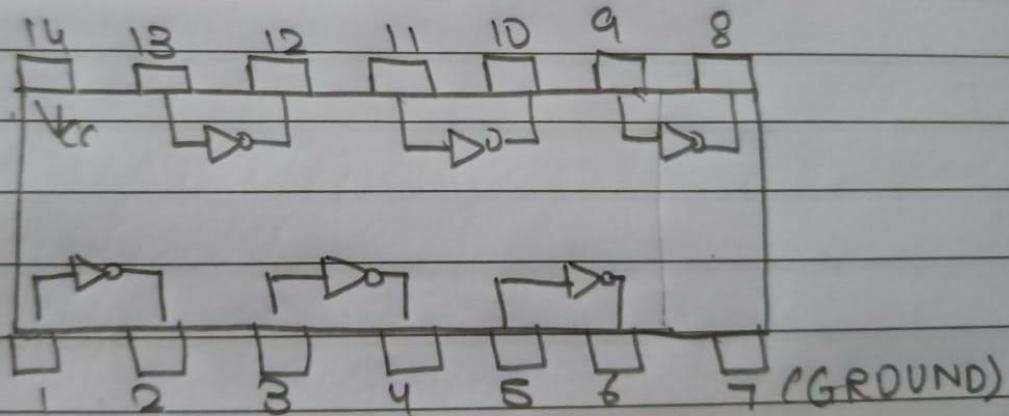
(1) QUAD 'OR' GATE:



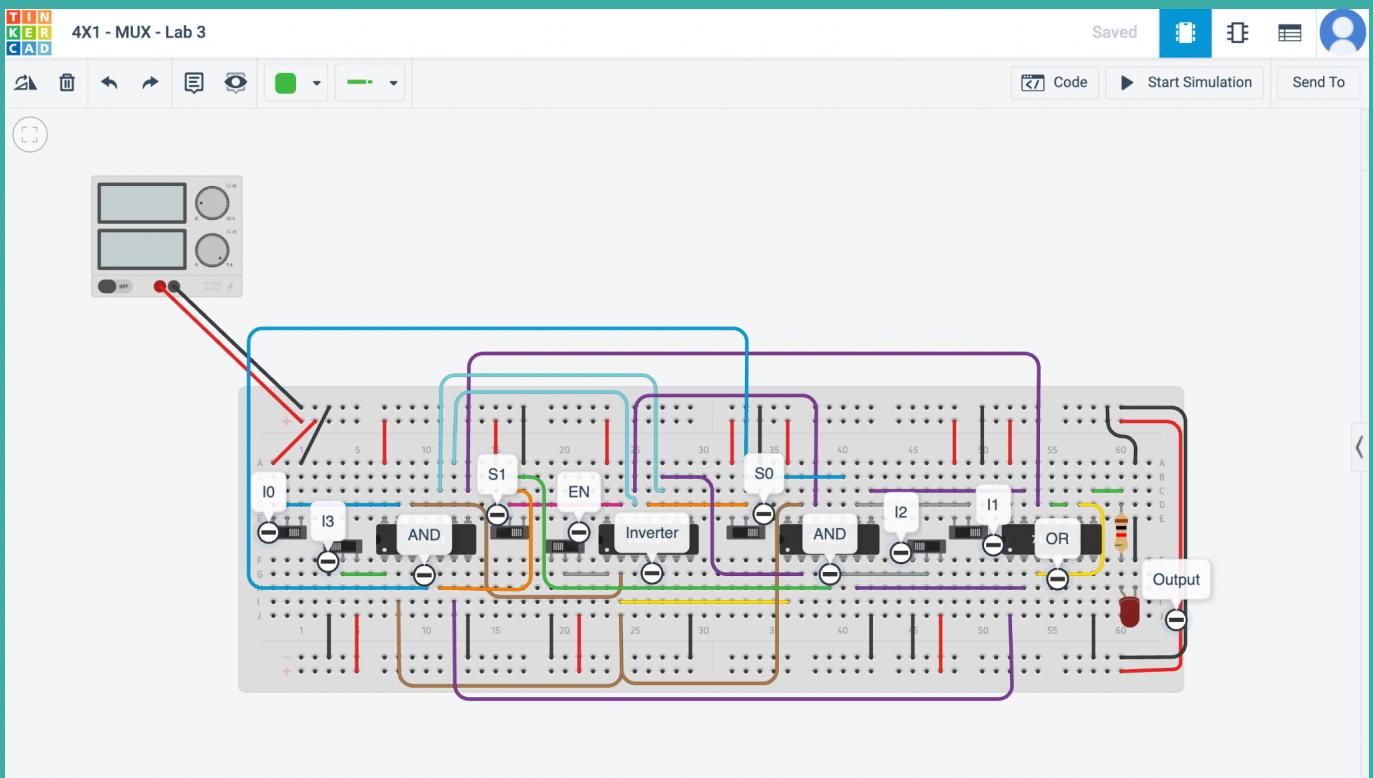
(2) DUAL 4-INPUT AND GATE:



### (8) HEX INVERTOR:



#### 4) Circuit Diagram: (Screenshot of TinkerCAD workspace):



### 5) Truth Table:

S1	S0	EN	Y
X	X	1	0
0	0	0	I0
0	1	0	I1
1	0	0	I2
1	1	0	I3

### 6) Observations/Results:

We made a 4x1 MUX using AND, OR and NOT Gates with EN and Signal Line.

## C) 8x1 MUX:

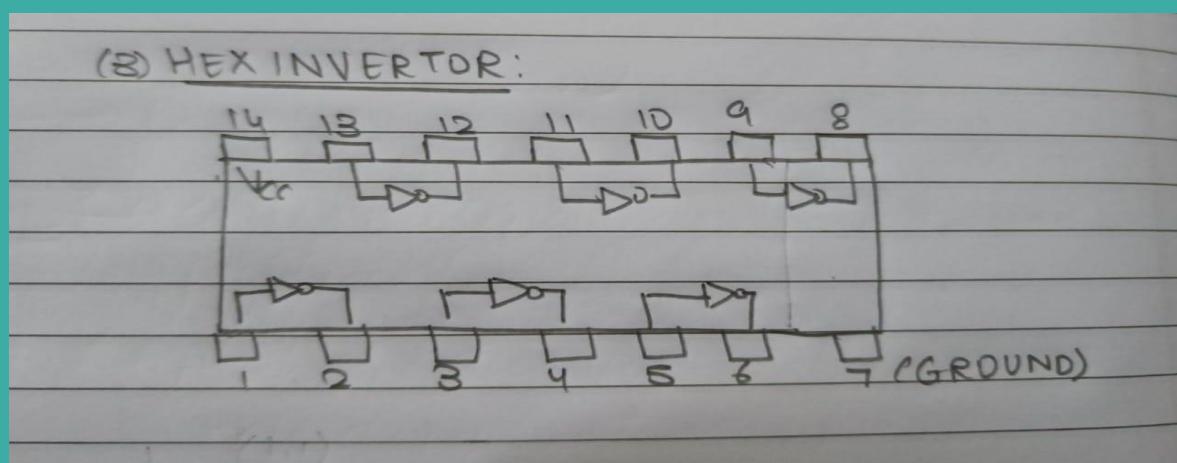
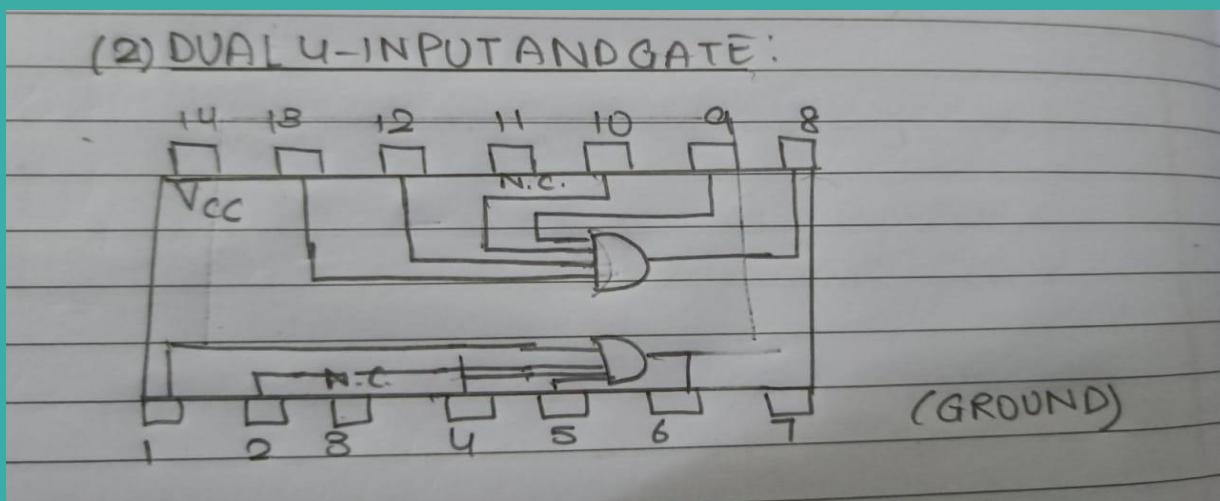
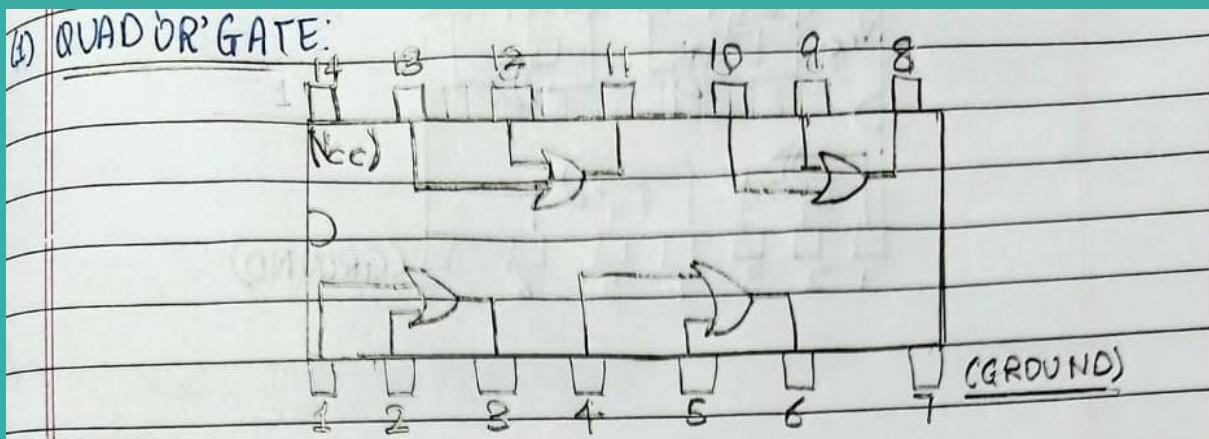
### 1) Components/ICs Used:

- a) OR Gate IC
- b) AND Gate IC
- c) HEX Inverter
- d) Resistors
- e) LEDs
- f) Power Supply
- g) Wires
- h) Breadboard
- i) Slideswitch

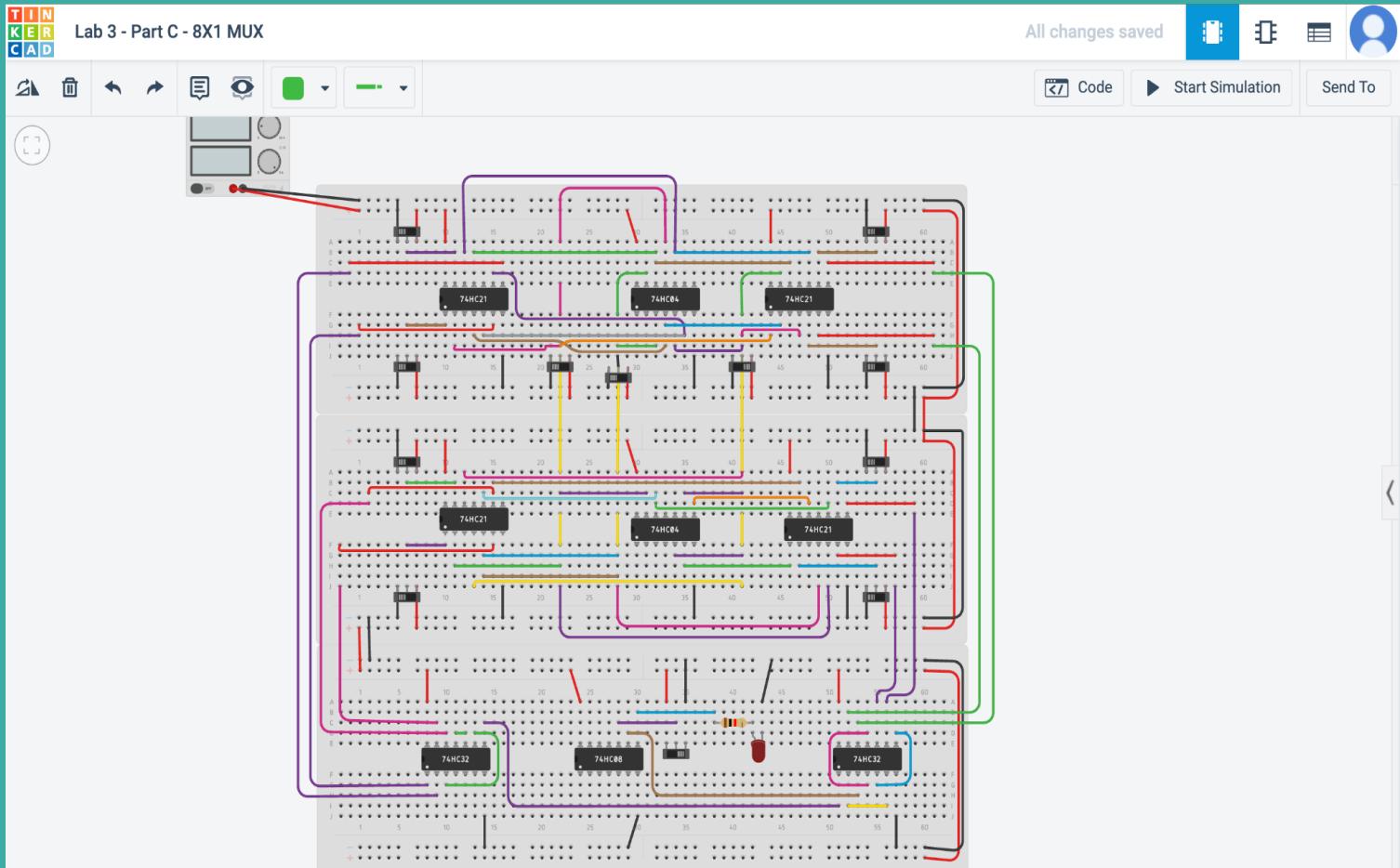
### 2) Link of TinkerCAD Workspace:

- <https://www.tinkercad.com/things/9uatzuvUBel-lab-3-part-c-8x1-mux/editel?sharecode=X0IGfIDrKMv25eCyfRtWLdkSEBekAcHdS5R3eVpA44E>

### 3) Pin Diagram of the IC:



#### 4) Circuit Diagram: (Screenshot of TinkerCAD workspace):



#### 5) Truth Table:

Select Data Inputs			Enabler	Output
S2	S1	S0	EN	Y
X	X	X	0	0
0	0	0	1	D0
0	0	1	1	D1
0	1	0	1	D2

0	1	1	1	D3
1	0	0	1	D4
1	0	1	1	D5
1	1	0	1	D6
1	1	1	1	D7

#### 6) Observations/Results:

We made a 8x1 MUX using AND, OR and NOT Gates with EN and Signal Line.

#### D) Application:

- 1) Multiplexers are used in telephone networks to merge audio signals into one output signal.
- 2) Multiplexers are used in making ALUs in computers to perform operations.
- 3) Multiplexers are used to store large amounts of Computer Memory.

#### E) Problem Statement:

A pound lock is used to raise and lower boats on rivers and canals. They consist of a pair of gates creating a chamber (lock) in which the water level is varied using valves.

To go upstream, the boat enters the drained lock through the bottom gate and the gate is closed behind it. The lock is then filled with water by opening the filling valve. Once full, the top gate is opened and the boat leaves.

To go downstream, the boat enters the filled lock through the top gate and the gate is closed behind it. The lock is then drained by opening the drain valve. Once drained, the bottom gate is opened and the boat leaves.

In both cases, the gates are controlled manually by a single lever. When the lever is not pulled, both gates should be closed. When pulled, it must open only the correct gate for the current water level. Hence, the top gate must only open if the lock is full, and the bottom gate must only open if the lock is drained.

Design the logic circuit required for the lock controller, modelling the gate lever, valve control and gates as follows:

→ Gate Lever(Input J):

0 – Closes both gates

1 – Opens one gate depending on water level

→ Valve Control (Input K):

0 – Lock drained (drain valve open)

1 – Lock filled (filler valve open)

→ Gates: (Outputs: Bottom - X and Top - Y)

0 – Gate closed

1 – Gate opened

### Problem Statement Solution:

The Truth Table which we can derive from the following problem statement is:

J	K	X	Y
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

So as per the problem statement, when both gates' lever is closed i.e. J = 0, irrespective of K, both Bottom and Top gate will be closed and hence X = 0 and Y = 0.

So the expressions we get from X and Y from this are as follows:

$$X = J \cdot K'$$

$$Y = J \cdot K$$

### Link to TinkerCad File for the Circuit:

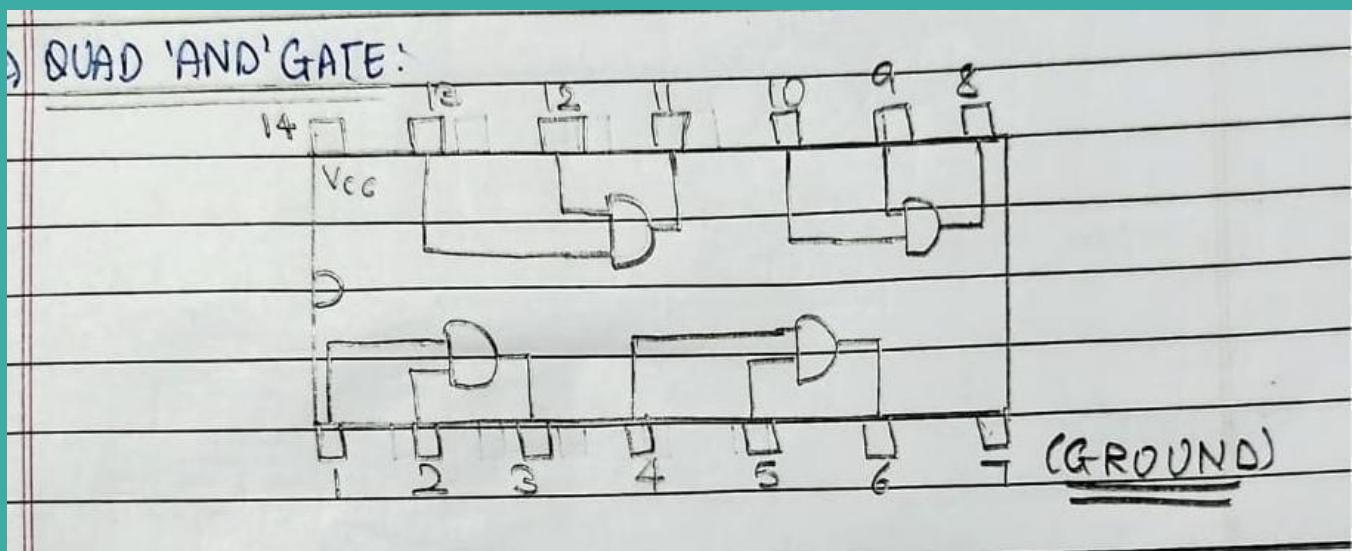
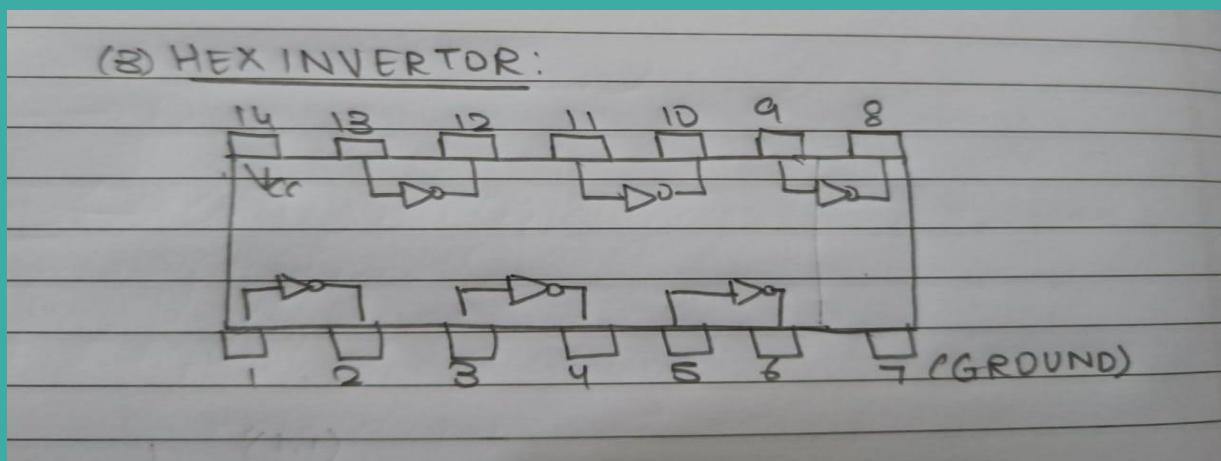
- [https://www.tinkercad.com/things/44YBfuiuvrp-lab-3-problem-statement/editel?sharecode=kvvCEryJQh3NIL62rGuQE02rd5Dp2M3KZ\\_ipdEqOrWg](https://www.tinkercad.com/things/44YBfuiuvrp-lab-3-problem-statement/editel?sharecode=kvvCEryJQh3NIL62rGuQE02rd5Dp2M3KZ_ipdEqOrWg)

### Components Used:

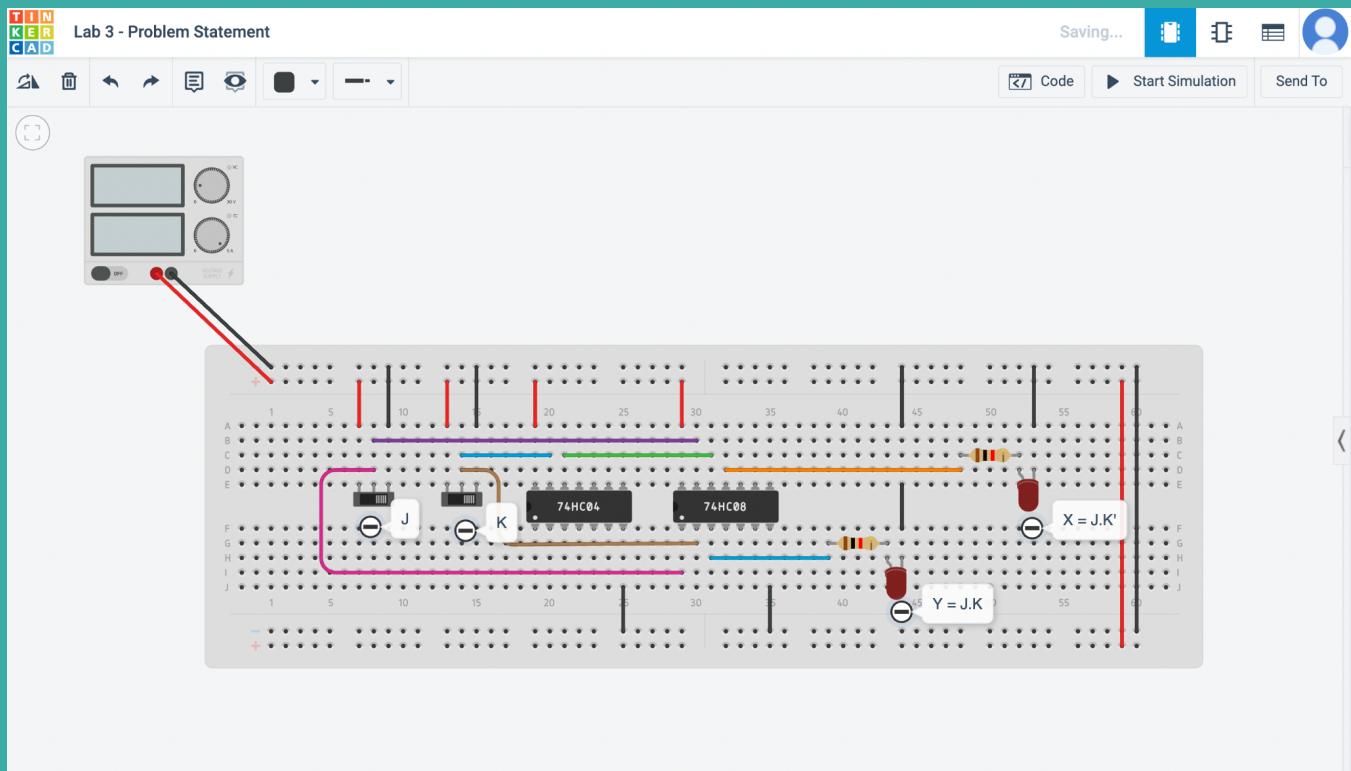
- 1) Hex Inverter
- 2) AND Gate IC

- 3) Slideswitch
- 4) Wires
- 5) LEDs
- 6) Breadboard
- 7) Resistor
- 8) Power Supply

**Pin Diagrams of ICs Used:**



## Circuit Diagram:





INDRAPRASTHA INSTITUTE *of*  
INFORMATION TECHNOLOGY  
DELHI

Department  
of  
Electronics & Communication Engineering

ECE111|Digital Circuits  
Section: A

**Dr. G.S. Visweswaran**

Lab\_4:

Student Name: Arnav Goel  
Roll No: 2021519  
Date: 10th Feb 2022

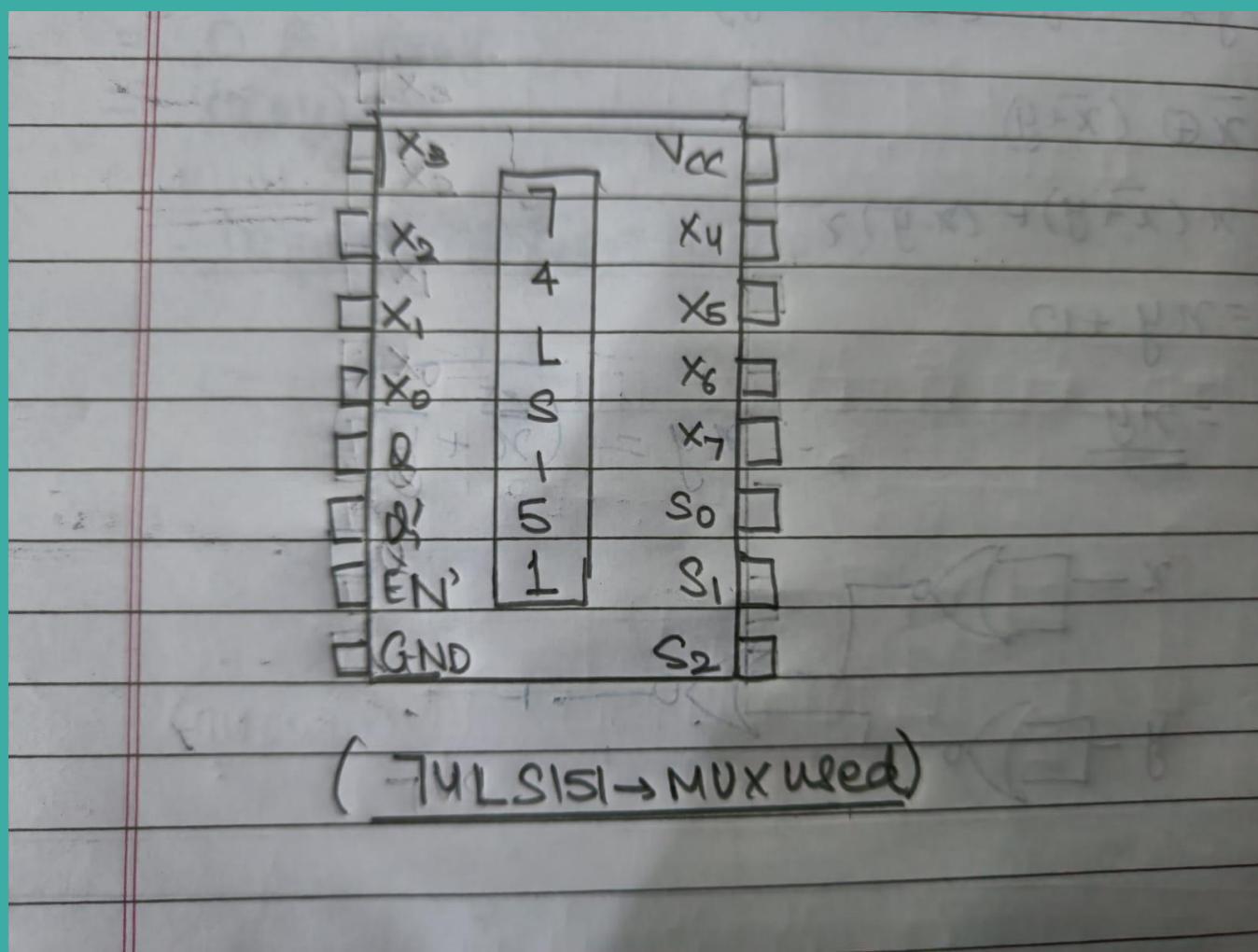
**Aim:**

To aim, design and test an ALU using 8x1 and 2x1 Multiplexers.

**Components/ICs Used:**

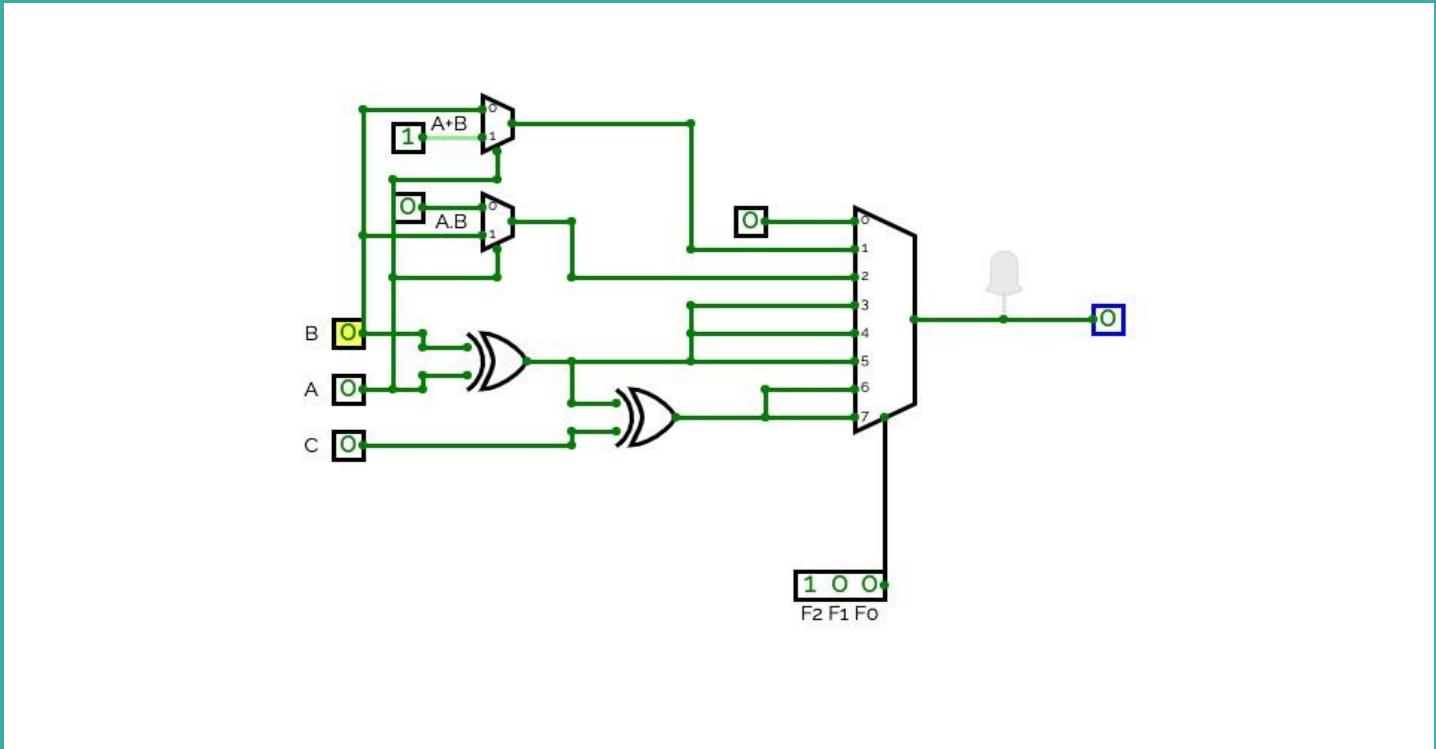
1. 8x1 Multiplexer
2. 2x1 Multiplexer
3. XOR Gate
4. 3 Bit Variable Input
5. 1 Bit Variable Input
6. Wires
7. LED
8. AND Gate

**Pin Diagram of the IC (If Applicable):**



## Circuit Diagram: (Screenshot of Circuitverse):

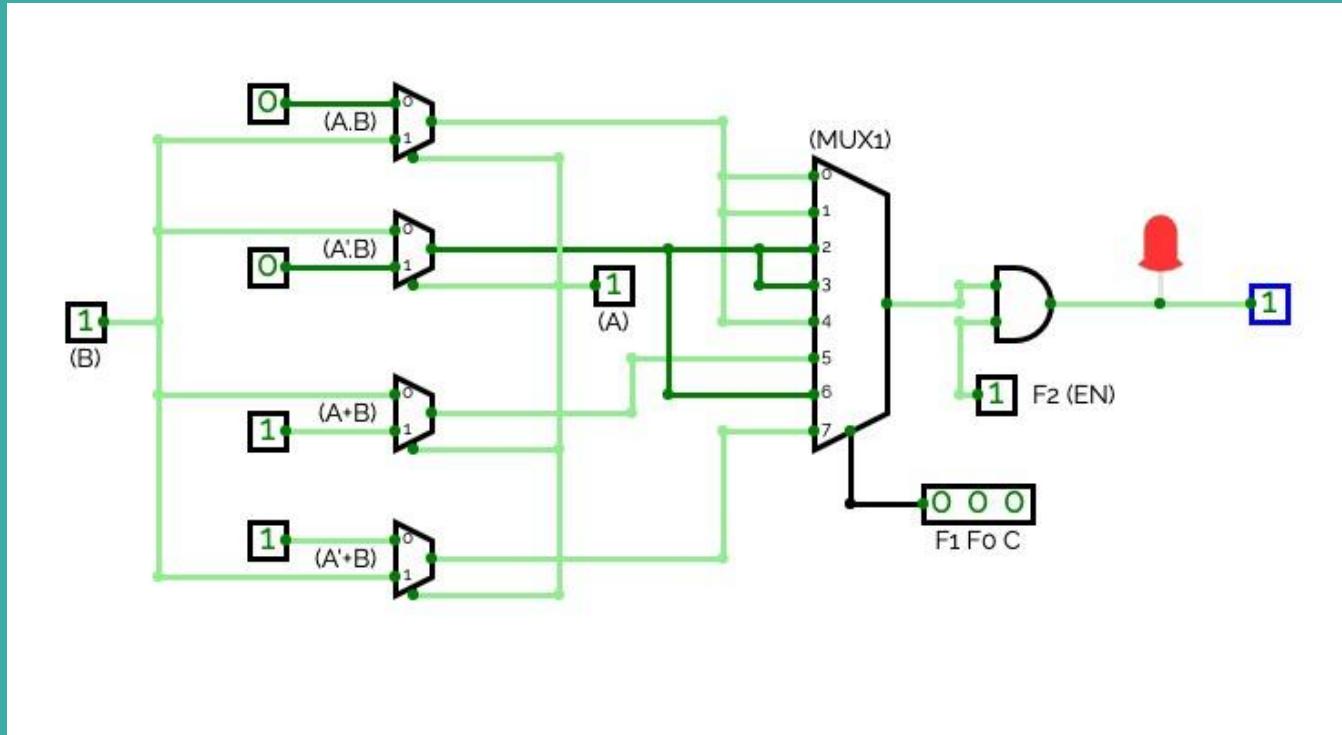
### 1) MUX0 or Y0:



### Link to the Circuitverse File:

- <https://circuitverse.org/users/116189/projects/673956>

### 2) MUX1 or Y1:



## Link to Circuitverse File:

- <https://circuitverse.org/users/116189/projects/673813>

## Truth Table:

### 1) MUX0 or Y0:

F2F1F0	ALU Function	Y0
000	0	0
001	A OR B	A+B
010	A AND B	A.B
011	A XOR B	A $\oplus$ B
100	A PLUS B	SUM
101	A MINUS B	DIFFERENCE
110	A PLUS B PLUS C	SUM
111	A MINUS B MINUS C	DIFFERENCE

In cases of 100 and 110 it behaves like a HALF and like a FULL ADDER giving its **SUM** whereas when input is 110 and 111, it behaves like a HALF and FULL SUBTRACTOR giving its **DIFFERENCE**.

### 2) MUX1 or Y1:

F2F1F0	ALU Function	Y1
000	0	-
001	A OR B	-
010	A AND B	-
011	A XOR B	-
100	A PLUS B	CARRY

101	A MINUS B	BORROW
110	A PLUS B PLUS C	CARRY
111	A MINUS B MINUS C	BORROW

In cases of 100 and 110 it behaves like a HALF and like a FULL ADDER giving its **CARRY** whereas when input is 110 and 111, it behaves like a HALF and FULL SUBTRACTOR giving its **BORROW**.

### 3) Truth Table for Individual Functions:

i)

A	B	A+B	A.B	$A \oplus B$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

ii) Half Adder:

A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

**iii) Full Adder:**

A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

**iv) Half Subtractor:**

A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

**v) Full Subtractor:**

A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1

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

### Observations/Results:

Applied various combinations of inputs and checked outputs corresponding to each. Tabulated them and verified the functionality of an ALU using an 8x1 Multiplexer and 2x1 Multiplexer to make our various logic gates.

### Application:

- 1) Multiplexers are used to implement a huge quantity of memory in a computer.
- 2) MUXs are used for data transfers from computers on satellites.
- 3) MUXs bring multiple audio signals into a single line for transmission in Telephone Networks.

### Problem Statement:

Spent fuel from nuclear reactors remains dangerously radioactive for long periods of time. This radiation must be contained. Spent fuel pools are often used to store the spent fuel underwater, because water cools the fuel down and blocks the radiation. Several conditions must be monitored and controlled to ensure that radioactive materials are not released into the environment.

One facility monitors the radioactive concentrations of both the water in the pool and the air surrounding the pool. If both concentrations differ from normal levels, then the main alarm turns on, indicating a major problem. The water level of the pool is also monitored. If the water level drops below a certain threshold, or if only one of the concentrations differ from normal levels, then the minor alarm turns on.

Hint: Design the logic circuit required to turn the alarms on and off, modeling the radioactive concentrations, water level and alarms as follows:

→ Radioactive concentrations (inputs J (water) and K (air))

- 0 – Same as normal level
- 1 – Differs from normal level
- Water level (input L):
- 0 – Above or at safe threshold
- 1 – Below safe threshold
- Alarms (outputs X (main) and Y (minor)):
- 0 – Alarm on
- 1 – Alarm off

### **Problem Statement Solution:**

We have been given two situations as to when would the Main Alarm go off and when would the Minor Alarm go off.

For the minor alarm, it depends on Water Level OR if only one of Water or Air level being different from normal (XOR), we can derive the following expression for Y (Minor Alarm):

$$Y = (L + (J \oplus K))' \longrightarrow (1) \quad (\text{Inverse because Alarm On is taken as 0})$$

Similarly for the Major Alarm, since it only depends on the difference between normal conc and conc of water AND air, we can derive the following expression:

$$X = (J \cdot K)' \longrightarrow (2) \quad (\text{Inverse because Alarm On is taken as 0})$$

We can derive the following Truth Table from the following expressions:

J(Water)	K(Air)	L(Water Level)	X(Main)	Y(Minor)
0	0	0	1	1
0	1	0	1	0
0	0	1	1	0
0	1	1	1	0
1	0	0	1	0
1	1	0	0	1
1	0	1	1	0
1	1	1	0	0

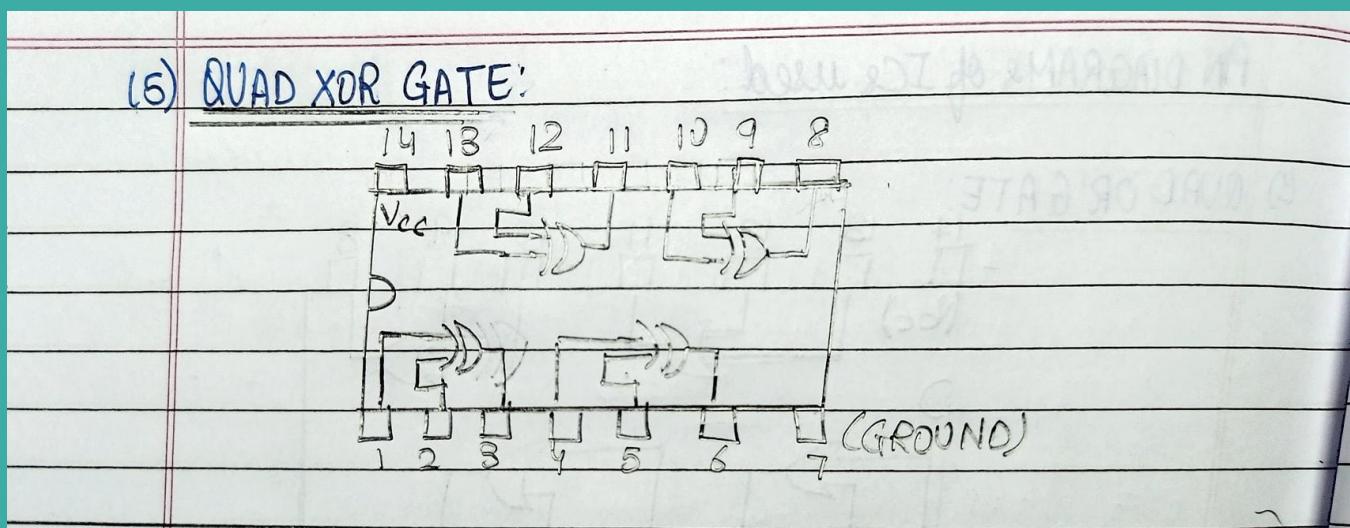
## TinkerCad Link for Circuit Implementation:

- [https://www.tinkercad.com/things/jTeO24RGPQJ-lab-4-problem-statement/editel?sharecode=ipQFF1javiNWMBDBvmfXRYm3\\_eISwNU4qpOwAdPEQvw](https://www.tinkercad.com/things/jTeO24RGPQJ-lab-4-problem-statement/editel?sharecode=ipQFF1javiNWMBDBvmfXRYm3_eISwNU4qpOwAdPEQvw)

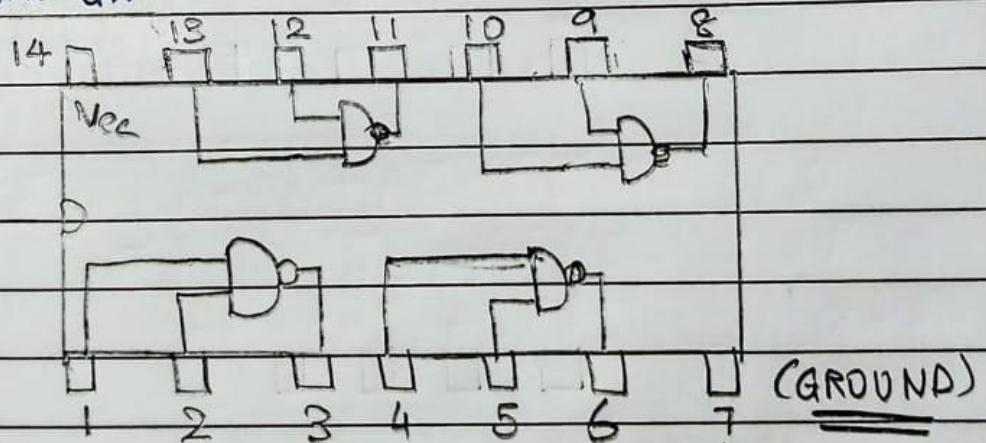
## Components Used:

- 1) Breadboard
- 2) Power
- 3) Resistor
- 4) XOR Gate IC
- 5) NAND Gate IC
- 6) NOR Gate IC
- 7) LEDs
- 8) Wires

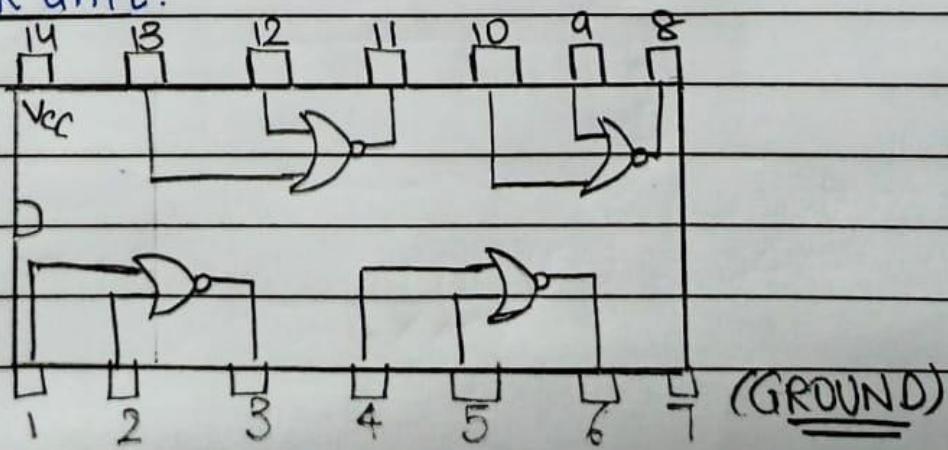
## Pin Diagrams of ICs used:



(3) QUAD 'NAND' GATE:



(4) QUAD 'NOR' GATE:



Circuit Diagram:

