

PRACTICAL NO. 1 Study of Logic gates and their ICs and universal gate

Verification and interpretation of truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates.

AIM: -To verify and interpret the logic and truth table for OR gate using Diode Resistance Logic (DRL), NOT gate using Transistor and AND, NAND, NOR, Ex-OR & Ex-NOR gates using Resistor Transistor Logic (RTL) in simulator 1 and verify the truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates in simulator 2.

THEORY: - Introduction

Logic gates are the basic building blocks of any digital system. Logic gates are electronic circuits having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as

1. AND gate
2. OR gate
3. NOT gate
4. NAND gate
5. NOR gate
6. Ex-OR gate
7. Ex-NOR gate

1) AND gate

The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot (.) is used to show the AND operation i.e. A.B or can be written as AB

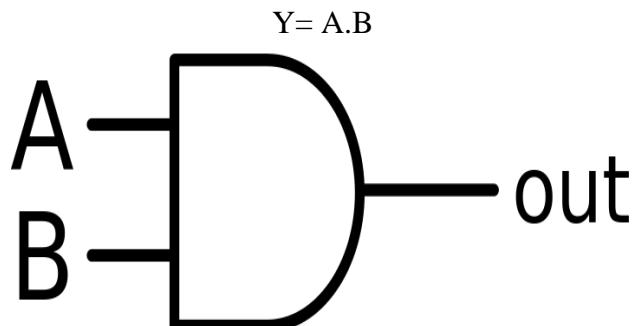


Figure-1:Logic Symbol of AND Gate

Input		Output
A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Figure-2:Truth Table of AND Gate

A simple 2-input logic AND gate can be constructed using RTL (Resistor-Transistor-Logic) switches connected together as shown below with the inputs connected directly to the transistor bases. Both transistors must be saturated “ON” for an output at Q.

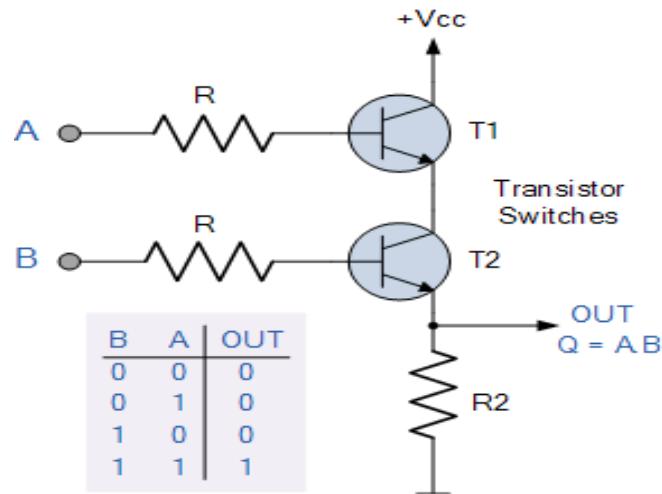


Figure-3:AND Gate through RTL logic

2) OR gate

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.

$$Y = A + B$$

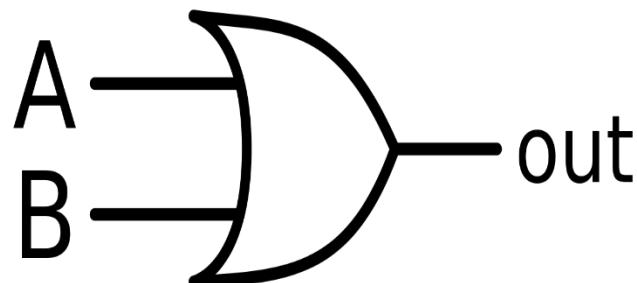


Figure-4:Logic Symbol of OR Gate

Input		Output
A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Figure-5:Truth Table of OR Gate

OR gate can be realized by DRL (Diode-Resistance-Logic) or by TTL (Transistor-Transistor-Logic). Presently, we will learn how to implement the OR gate using DRL (Diode-Resistance-Logic). To realise OR gate, we will use a diode at every input of the OR gate. The anode part of diode is connected with input while the cathode part is joined together and a resistor, connected with the cathode is grounded. In this case, we have taken two inputs which can be seen in the circuit below.

When both the inputs are at logic 0 or low state then the diodes D1 and D2 become reverse biased. Since the anode terminal of diode is at lower voltage level than the cathode terminal, so diode will act as open circuit so there is no voltage across resistor and hence output voltage is same as ground. When either of the diodes is at logic 1 or high state then the diode corresponding to that input is forward bias. Since this time anode is at high voltage than cathode therefore current will flow through forward biased diode and this current then appears on resistor causing high voltage at output terminal also. Hence at output we get high or logic 1 or +5V. So, if any or both inputs are high, the output will be high or “1”.

Diode OR Gate

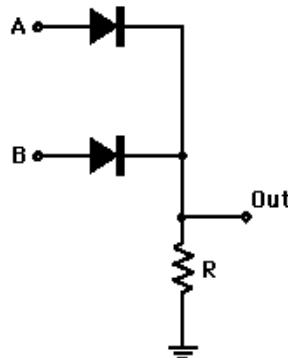


Figure-6:OR Gate through DRL logic

3) NOT gate

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A. This is also shown as A' or \bar{A} with a bar over the top, as shown at the outputs.

$$Y = A'$$

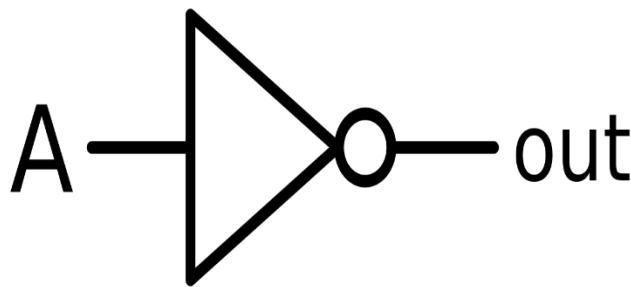


Figure-7:Logic Symbol of NOT Gate

Input	Output
A	Y
0	1
1	0

Figure-8:Truth Table of NOT Gate

NOT gate can be realized through transistor. The input is connected through resistor R2 to the transistor's base. When no voltage is present on the input, the transistor turns off. When the transistor is off, no current flows through the collector-emitter path. Thus, current from the supply voltage (V_{cc}) flows through resistor R1 to the output. In this way, the circuit's output is high when its input is low.

When voltage is present at the input, the transistor turns on, allowing current to flow through the collector-emitter circuit directly to ground. This ground path creates a shortcut that bypasses the output, which causes the output to go low.

In this way, the output is high when the input is low and low when the input is high.

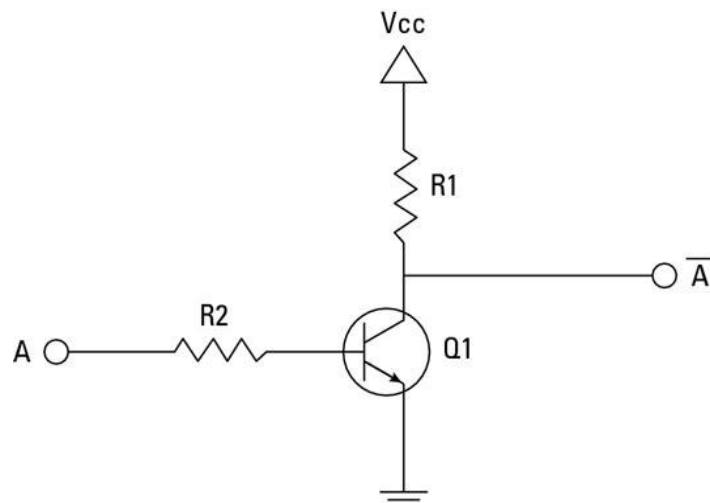


Figure-9:NOT Gate through Transistor

4) NAND gate

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

$$Y = AB$$

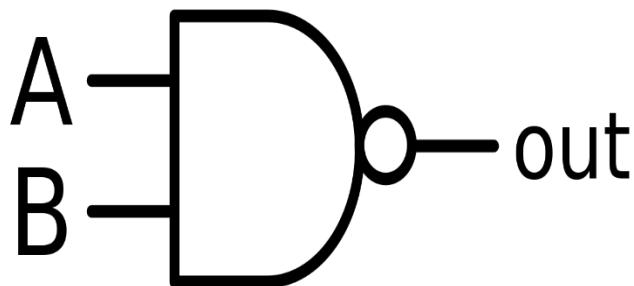


Figure-10:Logic Symbol of NAND Gate

Input	Input	Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Figure-11:Truth Table of NAND Gate

A simple 2-input logic NAND gate can be constructed using RTL (Resistor-transistor-logic) switches connected together as shown below with the inputs connected directly to the transistor bases. Either transistor must be cut-off or “OFF” for an output at Q.

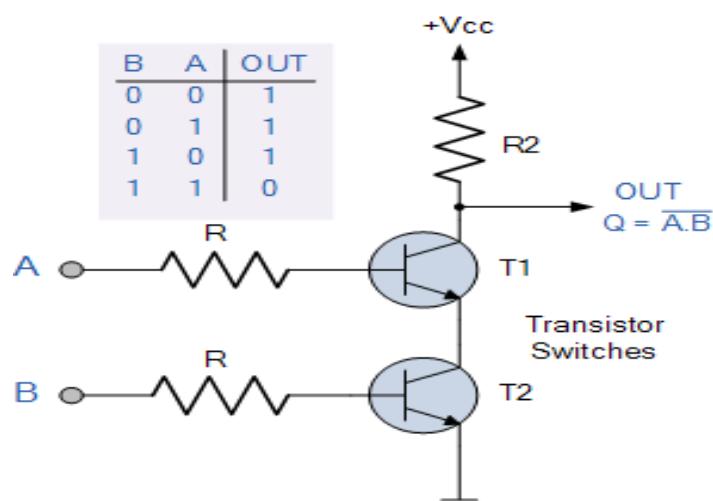


Figure-12:NAND gate through RTL Logic.

5) NOR gate

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

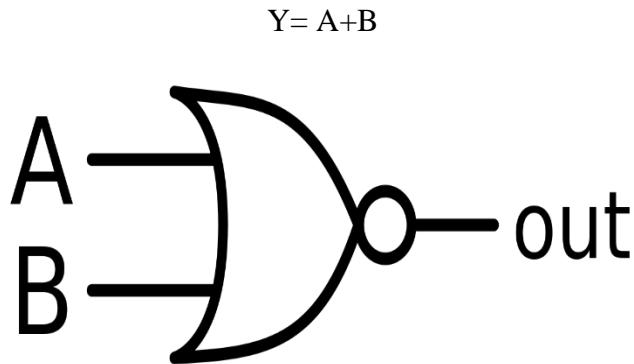


Figure-13:Logic Symbol of NOR gate

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Figure-14:Truth Table of NOR gate

A simple 2-input logic NOR gate can be constructed using RTL (Resistor-transistor-logic) switches connected together as shown below with the inputs connected directly to the transistor bases. Both transistors must be cut-off or “OFF” for an output at Q.

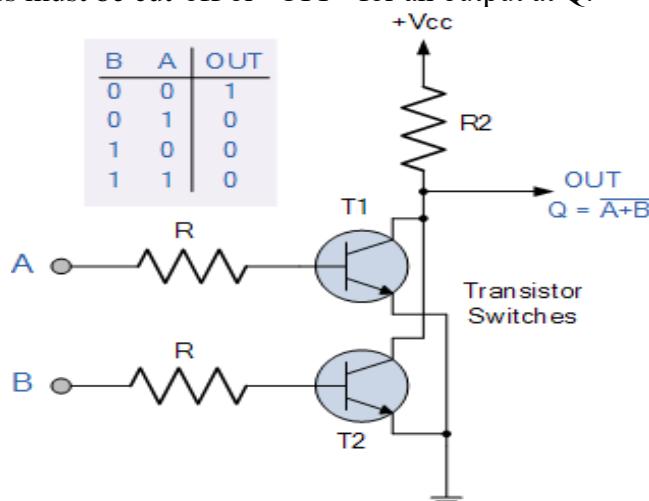


Figure-15:NOR gate through RTL Logic.

6) Ex-OR gate

The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both of its two inputs are high. An encircled plus sign (\oplus) is used to show the Ex-OR operation.

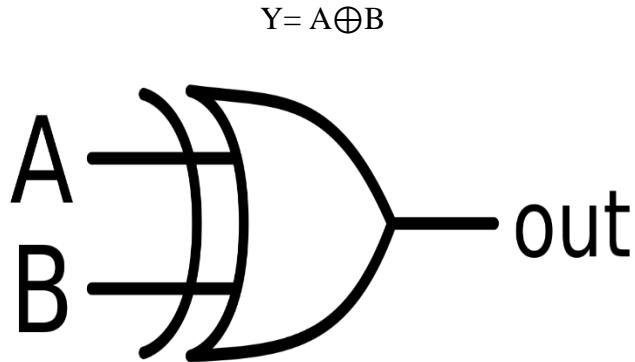


Figure-16:Logic Symbol of Ex-OR gate

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Figure-17:Truth Table of Ex-OR gate

Ex-OR gate is created from AND, NAND and OR gates. The output is high only when both the inputs are different.

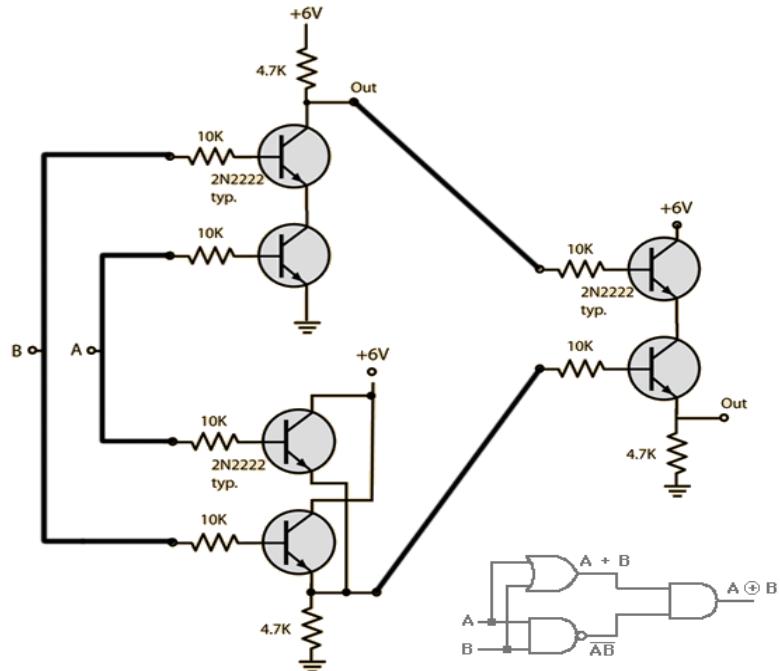


Figure-18:Ex-OR gate through RTL Logic.

7) Ex-NOR gate

The 'Exclusive-NOR' gate circuit does the opposite to the EX-OR gate. It will give a low output if either, but not both of its two inputs are high. The symbol is an EX-OR gate with a small circle on the output. The small circle represents inversion.

$$Y = A \oplus B$$

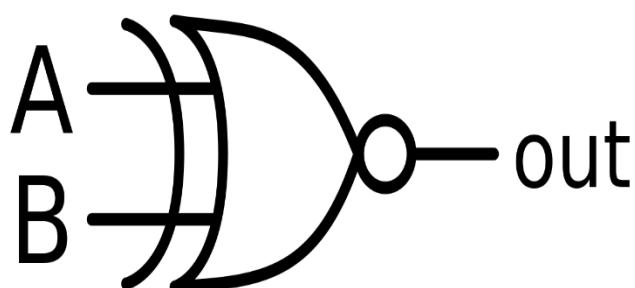


Figure-19:Logic Symbol of Ex-NOR gate

XNOR Truth Table		
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Figure-20:Truth Table of Ex-NOR gate

Ex-NOR gate is created from AND, NOT and OR gates.The output is high only when both the inputs are same.

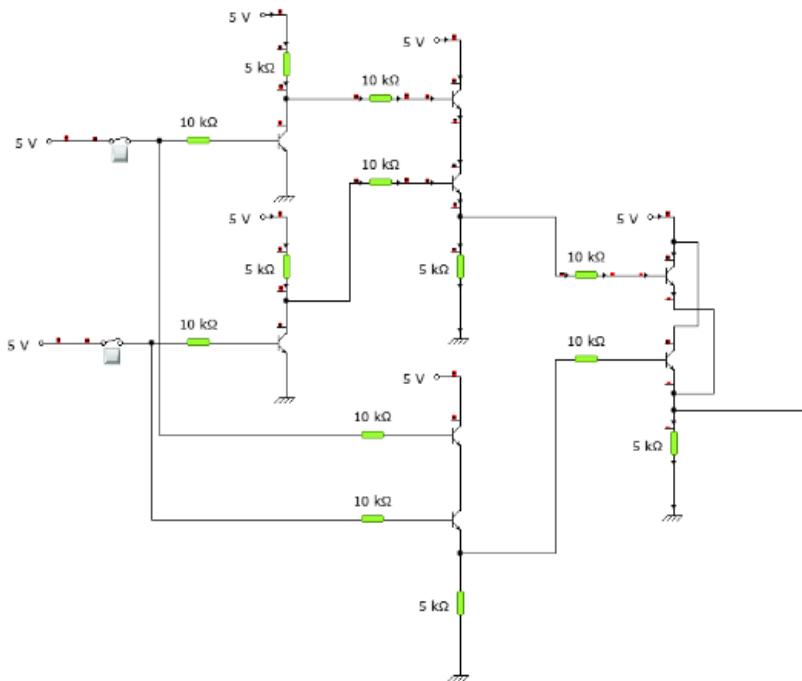


Figure-21: Ex-NOR gate through RTL Logic.

PRETEST: -

Screenshot of a web browser showing a quiz interface for Boolean Algebra. The URL is de-iitr.vlabs.ac.in/exp/truth-table-gates/pretest.html.

Aim: Verification and interpretation of truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates.

Theory: Electronic circuits that operate on one or more input signals to produce standard output _____

Pretest:

- a: Series circuits
- b: Parallel circuits
- c: Logic signals
- d: Logic gates

Procedure: A _____ gate gives the output as 1 only if all the inputs signals are 1.

a: AND

- b: OR
- c: NOR
- d: Ex-OR

Simulation: The boolean expression of an OR gate is _____

a: A.B

b: AB+AB'

c: A+B

d: AB'

Posttest:

References:

Feedback:

The boolean expression of an OR gate is _____

a: A.B

b: AB+AB'

c: A+B

d: AB'

The gate which is used to reverse the output obtained is _____

a: NOR

b: NAND

c: EX-OR

d: NOT

Which of the following gate will returns a 1 only if both the inputs are 0?

a: AND

b: OR

c: NAND

d: EX-OR

Submit Quiz

5 out of 5

PROCEDURE: -

1)AND Gate

Simulator 2:

- Step-1) Enter the Boolean input "A" and "B".
- Step-2) Enter the Boolean output for your corresponding inputs.
- Step-3) Click on "Check" Button to verify your output.
- Step-4) Click "Print" if you want to get print out of Truth Table.

2)OR Gate

Simulator 2:

- Step-1) Enter the Boolean input "A" and "B".
- Step-2) Enter the Boolean output for your corresponding inputs.

Step-3) Click on "Check" Button to verify your output.

Step-4) Click "Print" if you want to get print out of Truth Table.

3)NOT gate

Simulator 2:

Step-1) Enter the Boolean input "A".

Step-2) Enter the Boolean output for your corresponding input.

Step-3) Click on "Check" Button to verify your output.

Step-4) Click "Print" if you want to get print out of Truth Table.

4)NAND gate

Simulator 2:

Step-1) Enter the Boolean input "A" and "B".

Step-2) Enter the Boolean output for your corresponding inputs.

Step-3) Click on "Check" Button to verify your output.

Step-4) Click "Print" if you want to get print out of Truth Table.

5)NOR gate

Simulator 2:

Step-1) Enter the Boolean input "A" and "B".

Step-2) Enter the Boolean output for your corresponding inputs.

Step-3) Click on "Check" Button to verify your output.

Step-4) Click "Print" if you want to get print out of Truth Table.

6)Ex-OR gate

Simulator 2:

Step-1) Enter the Boolean input "A" and "B".

Step-2) Enter the Boolean output for your corresponding inputs.

Step-3) Click on "Check" Button to verify your output.

Step-4) Click "Print" if you want to get print out of Truth Table.

7)Ex-NOR gate

Simulator 2:

Step-1) Enter the Boolean input "A" and "B".

Step-2) Enter the Boolean output for your corresponding inputs.

Step-3) Click on "Check" Button to verify your output.

Step-4) Click "Print" if you want to get print out of Truth Table.

STIMULATION: -

AND GATE:

boolean Algebra presentation

Virtual Labs

Verification and interpretation of truth table for AND, OR, NOT, NAND...

Verification of truth table for AND gate

TRUTH TABLE				
Serial No.	A	B	Output	Remarks
1	1	1	1	Correct
2	1	0	1	Incorrect
3	1	0	0	Correct
4	0	0	0	Correct
5	0	1	0	Correct
6	0	1	0	Correct

Type here to search

Print

27°C Haze 21:41 05-12-2021

OR GATE:

boolean Algebra presentation

Virtual Labs

Verification and interpretation of truth table for AND, OR, NOT, NAND...

Verification of truth table for OR gate

TRUTH TABLE				
Serial No.	A	B	Output	Remarks
1	1	1	1	Correct
2	1	0	1	Correct
3	0	1	1	Correct
4	0	0	1	Incorrect
5	0	0	0	Correct

Type here to search

Print

27°C Haze 21:43 05-12-2021

NOT:

boolean Algebra presentation Virtual Labs

de-iitr.vlabs.ac.in/exp/truth-table-gates/simulation.html

Apps Gmail YouTube Maps

Verification and interpretation of truth table for AND, OR, NOT, NAND...

Verification of truth table for NOT gate

A —————— out

1 1

TRUTH TABLE Print

Serial No.	A	Output	Remarks
1	1	0	Correct
2	1	1	Incorrect

Type here to search 27°C Haze 21:44 05-12-2021

NAND:

boolean Algebra presentation Virtual Labs

de-iitr.vlabs.ac.in/exp/truth-table-gates/simulation.html

Apps Gmail YouTube Maps

Verification and interpretation of truth table for AND, OR, NOT, NAND...

Verification of truth table for NAND gate

A —————— out

0 1 1

TRUTH TABLE Print

Serial No.	A	B	Output	Remarks
1	1	1	0	Correct
2	1	1	1	Incorrect
3	1	0	1	Correct
4	0	0	1	Correct
5	0	1	1	Correct

Type here to search 27°C Haze 21:45 05-12-2021

NOR:

DE Journal Virtual Labs

<de-iitr.vlabs.ac.in/exp/truth-table-gates/simulation.html>

Apps Gmail YouTube Maps

Verification and interpretation of truth table for AND, OR, NOT, NAND...

Verification of truth table for NOR gate

TRUTH TABLE

Serial No.	A	B	Output	Remarks
1	1	0	1	Incorrect
2	1	0	1	Incorrect
3	0	0	1	Correct
4	1	1	1	Incorrect
5	1	1	0	Correct

Print

Type here to search

32°C Smoke 16:16 07-12-2021

EX-OR:

DE Journal Virtual Labs

<de-iitr.vlabs.ac.in/exp/truth-table-gates/simulation.html>

Apps Gmail YouTube Maps

Verification and interpretation of truth table for AND, OR, NOT, NAND...

Verification of truth table for XOR gate

TRUTH TABLE

Serial No.	A	B	Output	Remarks
1	0	0	0	Correct
2	0	1	0	Incorrect
3	0	1	1	Correct
4	1	1	1	Incorrect
5	1	1	0	Correct
6	1	0	1	Correct

Print

Type here to search

32°C Smoke 16:17 07-12-2021

EX-NOR:

The screenshot shows a web browser window titled "DE JOURNAL" with the URL "de-iitr.vlabs.ac.in/exp/truth-table-gates/simulation.html". The page displays a logic circuit diagram for an XNOR gate with inputs A and B, and an output labeled "out". Below the circuit are two input boxes: one containing "0" and another containing "1". To the right of the output is a box with "0" and a green "Check" button. At the bottom, there is a "TRUTH TABLE" section with a "Print" button. The truth table is as follows:

Serial No.	A	B	Output	Remarks
1	1	1	0	Incorrect
2	1	1	1	Correct
3	1	0	1	Incorrect
4	1	0	0	Correct
5	0	0	1	Correct
6	0	1	0	Correct

POSTTEST:

The NOR gate is OR gate followed by _____

a: AND gate
 b: NAND gate
 c: NOT gate
 d: None of the above

The NAND gate is AND gate followed by _____

a: NOT gate
 b: OR gate
 c: AND gate
 d: None of the above

In boolean algebra, the bar sign (-) indicates _____

a: OR operation
 b: AND operation
 c: NOT operation
 d: None of the above

Submit Quiz

5 out of 5

CONCLUSION:

NOT Gate: When logic 1 is applied to one of NOT gate, then output becomes zero.

OR Gate: The output of an OR gate is a 1 if one or the other or both of the inputs are 1, but a 0 if both inputs are 0.

AND Gate: The output of an AND gate is only 1 if both its inputs are 1. For all other possible inputs, the output is 0.

NOR Gate: The output of the NOR gate is a 1 if both inputs are 0 but a 0 if one or the other or both the inputs are 1. NAND Gate: The output of the NAND gate is a 0 if both inputs are 1 but a 1 if one or the other or both the inputs are 0. EXOR gate: The output of the XOR gate is a 1 if either but not both inputs are 1 and a 0 if the inputs are both 0 and both 1.

Any Boolean expression can be realized using NOT, AND, OR, NAND, NOR, EXOR gates.

Practical no. 2 Implement the given Boolean expressions using minimum number of gates.

Realization of logic functions with the help of universal gates NAND and NOR Gate

AIM: - To implement the logic functions i.e. AND, OR, NOT, Ex-OR, Ex- NOR and a logical expression with the help of NAND and NOR universal gates respectively.

THEORY: - Introduction

Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/output combination is called Truth Table.

1)Nand gate as Universal gate

NAND gate is actually a combination of two logic gates i.e. AND gate followed by NOT gate. So its output is complement of the output of an AND gate. This gate can have minimum two inputs. By using only NAND gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NOR. So this gate is also called as universal gate.

1.1)NAND gates as OR gate

$$\begin{array}{lll} \text{From} & \text{DeMorgan's} & \text{theorems:} \\ (A \cdot B)' & = & A' + B' \\ (A' \cdot B')' & = & A'' + B'' = A + B \end{array}$$

So, give the inverted inputs to a NAND gate, obtain OR operation at output.

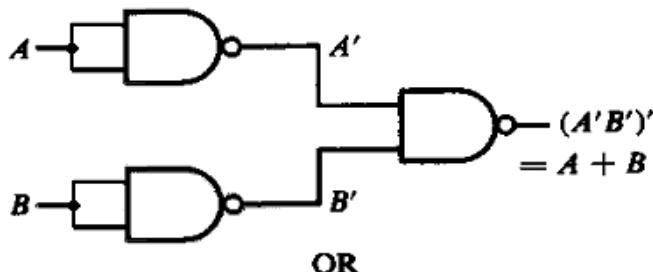


Figure-1:NAND gates as OR gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure-2:Truth table of OR

1.2)NAND gates as AND gate

A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.

$$Y = ((A \cdot B)')'$$

$$Y = A \cdot B$$

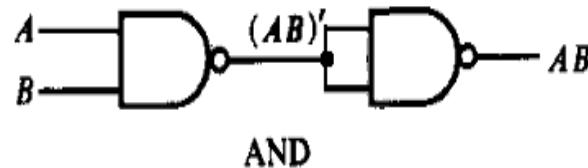


Figure-3:NAND gates as AND gate

Input		Output
A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Figure-4:Truth table of AND

1.3)NAND gates as Ex-OR gate

The output of a two input Ex-OR gate is shown by: $Y = A'B + AB'$. This can be achieved with the logic diagram shown in the left side.

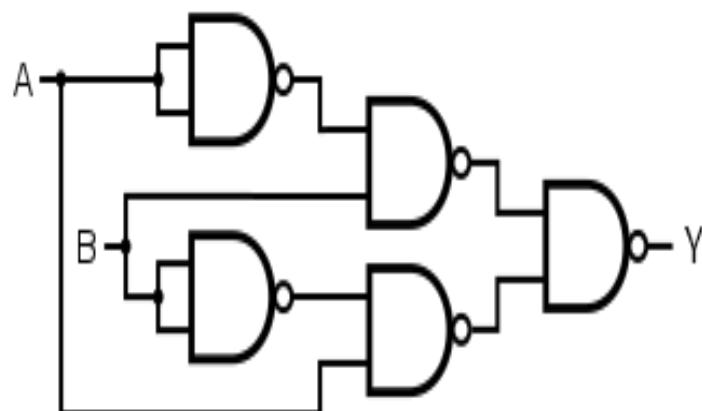


Figure-5:NAND gate as Ex-OR gate

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Figure-6:Truth table of Ex-OR

1.4)NAND gates as Ex-NOR gate

Ex-NOR gate is actually Ex-OR gate followed by NOT gate. So give the output of Ex-OR gate to a NOT gate, overall output is that of an Ex-NOR gate.

$$Y = AB + A'B'$$

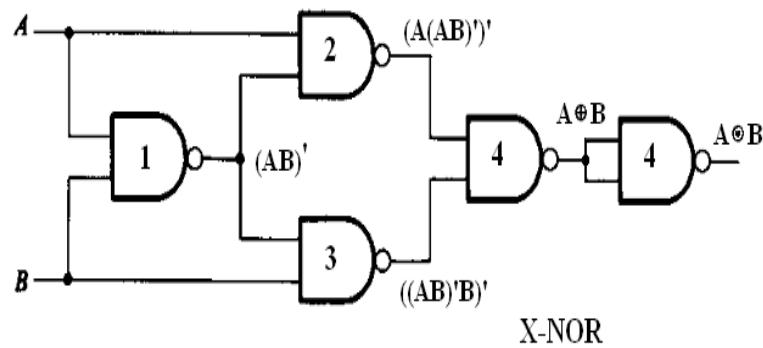


Figure-7:NAND gates as Ex-NOR gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Figure-8:Truth table of Ex-NOR

1.5) Implementing the simplified function with NAND gates only

We can now start constructing the circuit. First note that the entire expression is inverted and we have three terms ANDed. This means that we must use a 3-input NAND gate. Each of the three terms is, itself, a NAND expression. Finally, negated single terms can be generated with a 2-input NAND gate acting as an inverted. The expression illustrates a circuit using NAND gates only.

$$F = ((C'.B.A)'(D'.C.A)'(C.B'.A)')'$$

The stepwise simplification of this expression is done on the basis of this logic diagram in Figure 9:

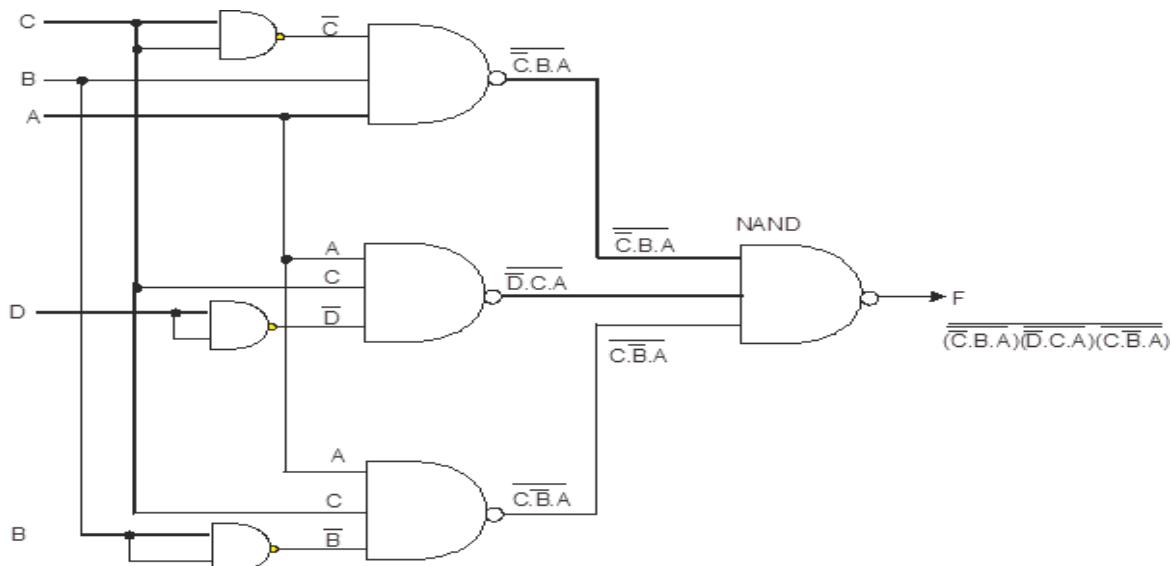


Figure-9:Implementing the simplified function with NAND gates only

2)Nor gate as Universal Gate

NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate. So its output is complement of the output of an OR gate. This gate can have minimum two inputs, output is always one. By using only NOR gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NAND. So this gate is also called universal gate.

2.1)NOR gates as OR gate

A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.

$$\begin{aligned} Y &= ((A+B)')' \\ Y &= (A+B) \end{aligned}$$

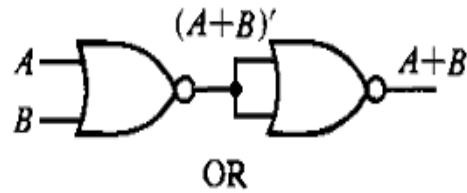


Figure-10:NOR gates as OR gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure-11:Truth table of OR

2.2)NOR gates as AND gate

From DeMorgan's theorems:
 $(A+B)' = A'B'$
 $(A'+B')' = A''B'' = AB$

So, give the inverted inputs to a NOR gate, obtain AND operation at output.

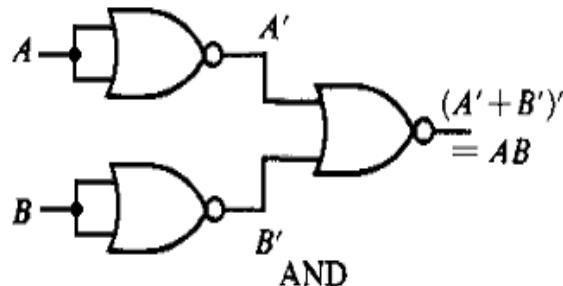


Figure-12:NOR gates as AND gate

Input		Output
A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Figure-13:Truth table of AND

2.3)NOR gates as Ex-OR gate

Ex-OR gate is actually Ex-NOR gate followed by NOT gate. So give the output of Ex-NOR gate to a NOT gate, overall output is that of an Ex-OR gate.
 $Y = A'B + AB'$

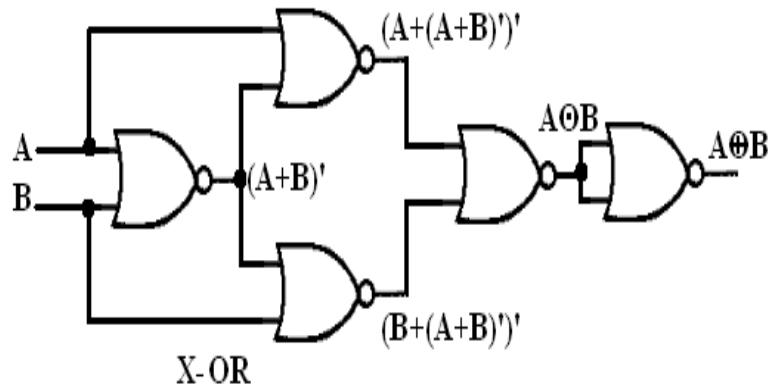


Figure-14:NOR gates as Ex-OR gate

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Figure-15:Truth table of Ex-OR

2.4)NOR gates as Ex-NOR gate

The output of a two input Ex-NOR gate is shown by: $Y = AB + A'B'$. This can be achieved with the logic diagram shown in the left side.

Gate No.	Inputs	Output
1	A, B	$(A + B)'$
2	A, $(A + B)'$	$(A + (A+B)')'$
3	$(A + B)', B$	$(B + (A+B))'$
4	$(A + (A+B)')', (B + (A+B))'$	$AB + A'B'$

Now the output from gate no. 4 is the overall output of the configuration.

$$\begin{aligned}
 Y &= ((A + (A+B)')' (B + (A+B))')' \\
 &= (A + (A+B)')' . (B + (A+B))'' \\
 &= (A + (A+B)') . (B + (A+B))' \\
 &= (A + A'B') . (B + A'B') \\
 &= (A + A') . (A + B') . (B + A') . (B + B') \\
 &= 1 . (A + B') . (B + A') . 1 \\
 &= (A + B') . (B + A') \\
 &= A . (B + A') + B' . (B + A') \\
 &= AB + AA' + B'B + B'A' \\
 &= AB + 0 + 0 + B'A' \\
 &= AB + B'A' \\
 \Rightarrow Y &= AB + A'B'
 \end{aligned}$$

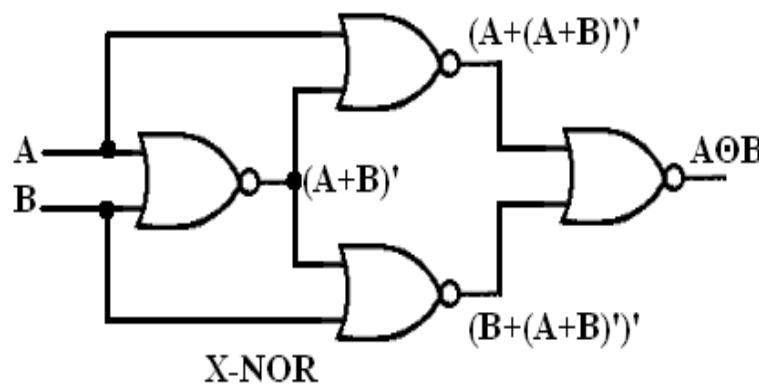


Figure-16:NOR gates as Ex-NOR gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Figure-17:Truth table of Ex-NOR

2.5)Constructing a circuit with NOR gates only

Designing a circuit with NOR gates only uses the same basic techniques as designing a circuit with NAND gates; that is, the application of deMorgan's theorem. The only difference between NOR gate design and NAND gate design is that the former must eliminate product terms and the latter must eliminate sum terms.

$$F = (((C \cdot B' \cdot A) + (D \cdot C' \cdot A) + (C \cdot B' \cdot A))')$$

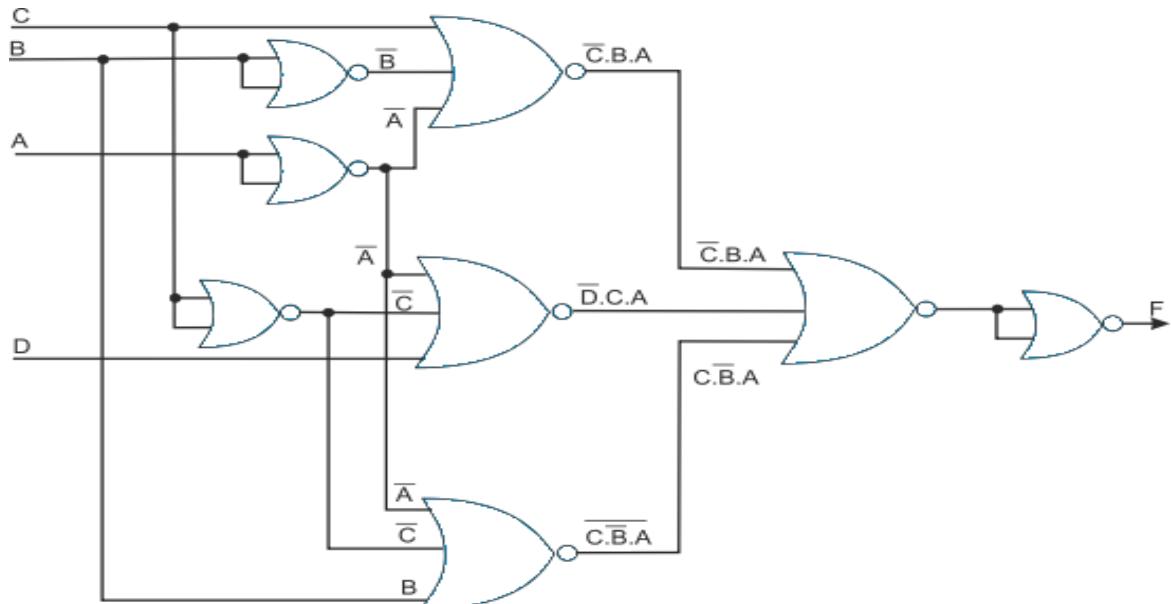


Figure-18 : Implementing the simplified function with NOR gates only

PRETEST: -

DE JOURNAL Virtual Labs

de-iitr.vlabs.ac.in/exp/realization-of-logic-functions/pretest.html

HOME PARTNERS CONTACT

Virtual Labs An MoE Govt of India Initiative

Electronics and Communication Engineering > Digital Electronics IITR > Experiments

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Realization of logic functions with the help of universal gates NAND and NOR Gate

The universal gate is _____

- a: AND gate
- b: OR gate
- c: NAND gate
- d: None of the above

The inputs of a NAND gate are connected together. The resulting circuit is _____

- a: OR gate
- b: AND gate
- c: NOT gate
- d: None of the above

The NOR gate is OR gate followed by _____

- a: AND gate
- b: NAND gate
- c: NOT gate

Type here to search

33°C Smoke 16:40 07-12-2021

DE Journal Virtual Labs

The NOR gate is OR gate followed by _____

- a: AND gate
- b: NAND gate
- c: NOT gate
- d: None of the above

Digital circuit can be made by the repeated use of _____

- a: OR gates
- b: NOT gates
- c: NAND gates
- d: None of the above

A single transistor can be used to build _____ gates .

- a: OR Gate
- b: NOT Gate
- c: AND Gate
- d: NAND Gate

Submit Quiz

5 out of 5

Type here to search

33°C Smoke ENG 07-12-2021

PROCEDURE: -

- Step-1) Select and drag " " for generating wire of the circuit.
- Step-2) Join the wire to perform the required logic.
- Step-3) Click on check button to check the connections.
- Step-4) If connections are wrong click on reset button to reset connections.

Note: Follow these steps for all experiments.

STIMULATION: -

NAND AS UNIVERSAL:

DE Journal Virtual Labs

Realization of

de-iitr.vlabs.ac.in says
RIGHT CONNECTION

OK

Lgates NAND..

INSTRUCTIONS

Experiment to perform logic of Or Using Nand on kit

Y=A+B

Check Reset Print Next

Type here to search

33°C Smoke ENG 07-12-2021

DE Journal x Virtual Labs x +

<de-iitr.vlabs.ac.in/exp/realization-of-logic-functions/simulation.html>

Apps Gmail YouTube Maps

de-iitr.vlabs.ac.in says
RIGHT CONNECTION

OK

Realization of Logic Functions Using NAND Gates

INSTRUCTIONS

Experiment to perform logic of AND Using NAND on kit

Y = A · B

```

graph LR
    A((A)) --> N1[NAND2]
    B((B)) --> N1
    N1 --> N2[NAND2]
    B --> N2
    N2 --> Y["Y = A · B"]
    
```

Check Reset PRINT Next

de-iitr.vlabs.ac.in says
RIGHT CONNECTION

OK

Realization of Logic Functions Using NAND Gates

INSTRUCTIONS

Experiment to perform logic of Ex-OR Using NAND on kit

Y = AB + ĀB

```

graph LR
    A((A)) --> N1[NAND2]
    B((B)) --> N1
    A --> N3[NAND2]
    B --> N3
    N1 --> N2[NAND2]
    N3 --> N2
    N2 --> N4[NAND2]
    N4 --> N5[NAND2]
    N5 --> Y["Y = AB + ĀB"]
    
```

Check Reset PRINT Next

de-iitr.vlabs.ac.in says
RIGHT CONNECTION

OK

Realization of Logic Functions Using NAND Gates

INSTRUCTIONS

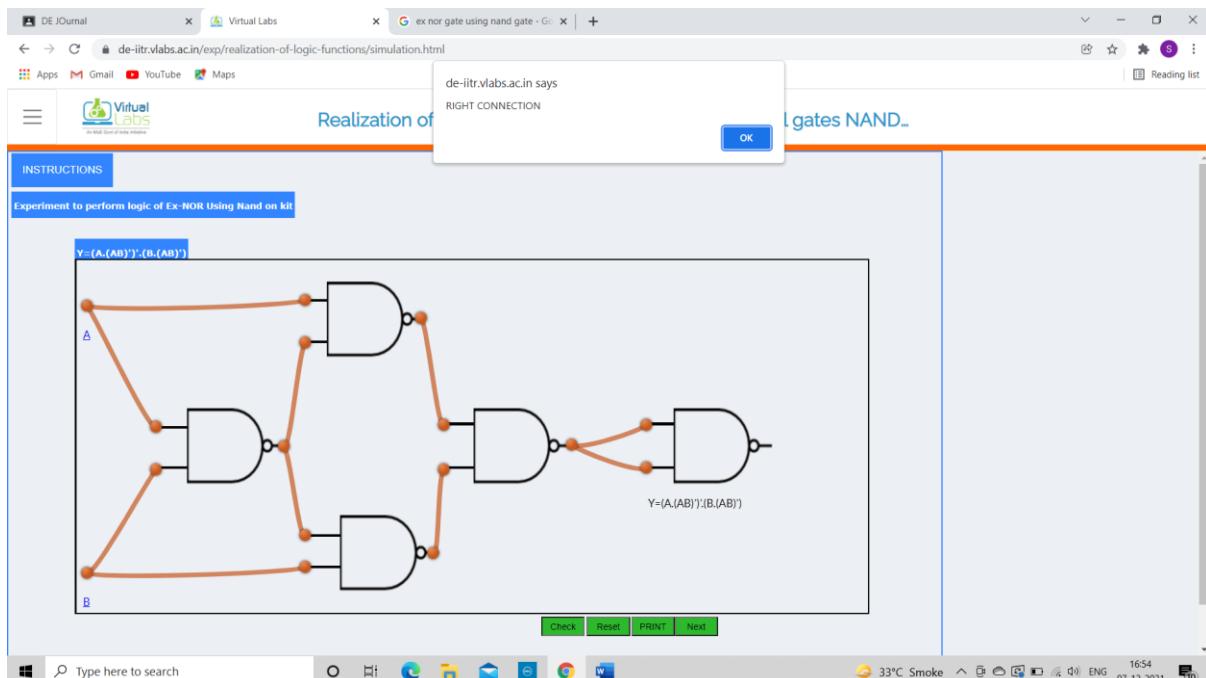
Experiment to perform logic of Ex-OR Using NAND on kit

Y = AB + ĀB

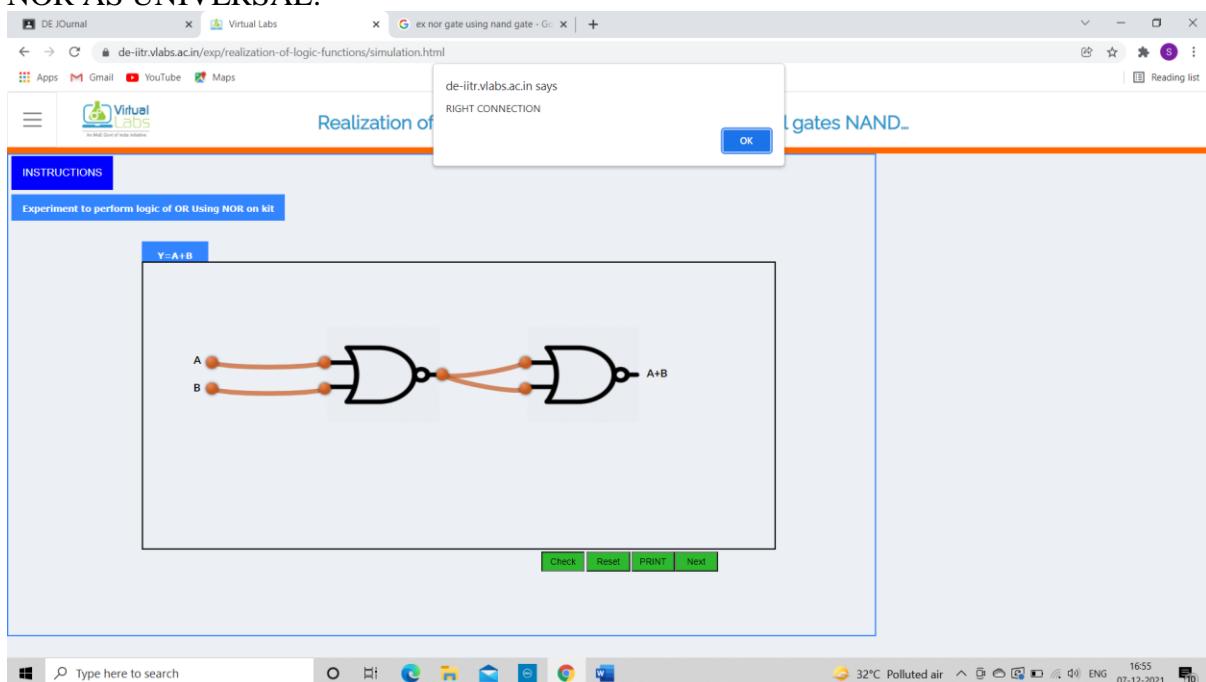
```

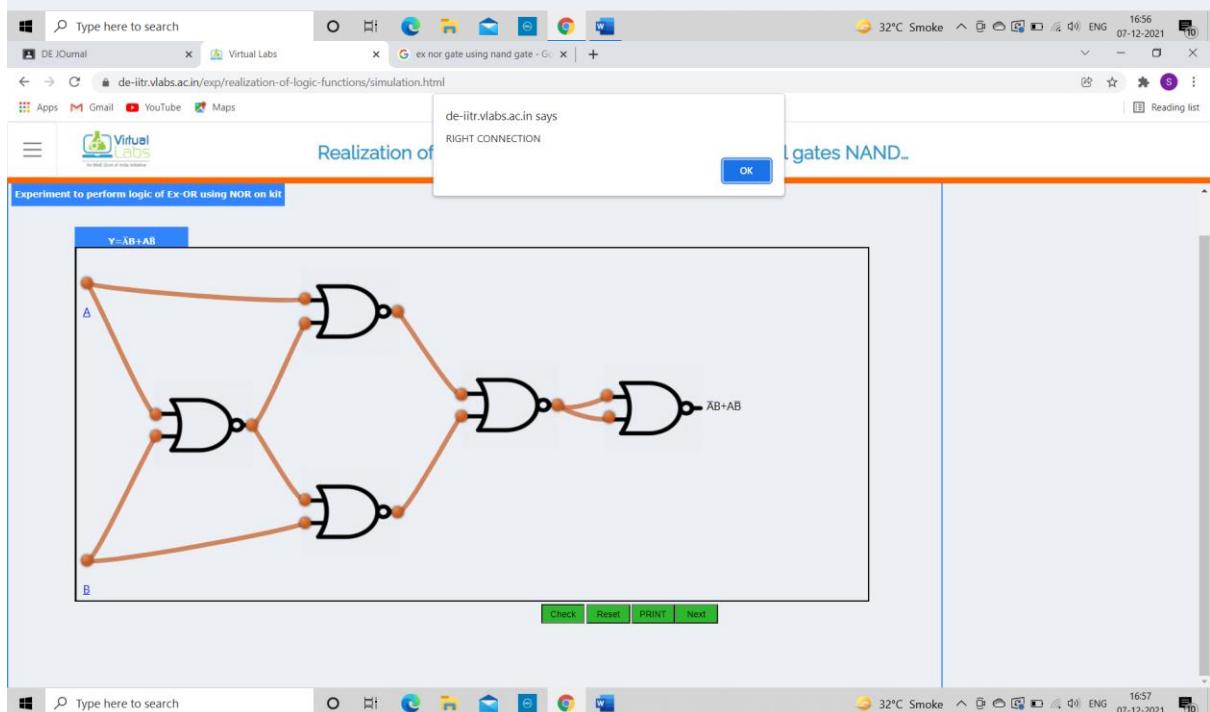
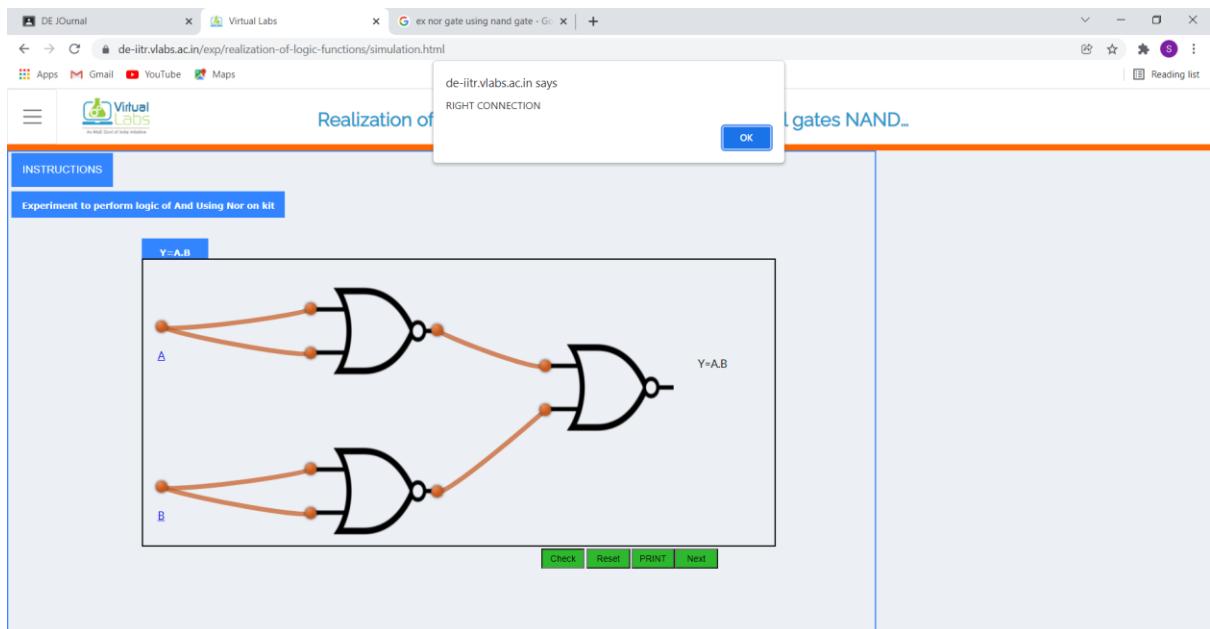
graph LR
    A((A)) --> N1[NAND2]
    B((B)) --> N1
    A --> N3[NAND2]
    B --> N3
    N1 --> N2[NAND2]
    N3 --> N2
    N2 --> N4[NAND2]
    N4 --> N5[NAND2]
    N5 --> Y["Y = AB + ĀB"]
    
```

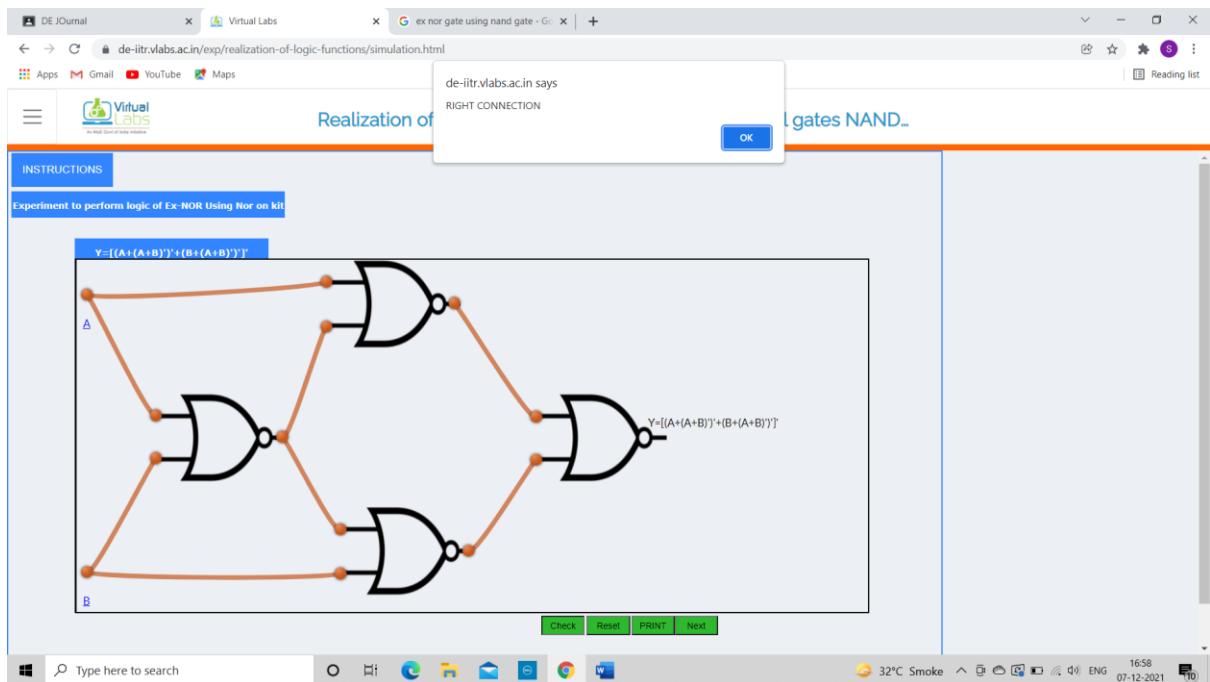
Check Reset PRINT Next



NOR AS UNIVERSAL:







POSTTEST: -

Realization of logic functions with the help of universal gates NAND and NOR Gate

In Boolean algebra, the bar sign ($\bar{\cdot}$) indicates _____.

- a: OR operation
- b: AND operation
- c: NOT operation
- d: None of the above

The output of an AND gate with three inputs, A, B, and C, is HIGH when _____.

- a: A = 1, B = 1, C = 0
- b: A = 0, B = 0, C = 0
- c: A = 1, B = 1, C = 1
- d: A = 1, B = 0, C = 1

If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

- a: 1
- b: 2
- c: 7

DE Journal Virtual Labs

de-itrlabs.ac.in/exp/realization-of-logic-functions/posttest.html

App Gmail YouTube Maps

HOME PARTNERS CONTACT

If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

a: 1
 b: 2
 c: 7
 d: 8

Which of the following logical operations is represented by the + sign in Boolean algebra?

a: inversion
 b: AND
 c: OR
 d: complementation

Which of the following equations would accurately describe a four-input OR gate when A = 1, B = 1, C = 0, and D = 0?

a: $1 \cdot 1 \cdot 0 \cdot 0 = 0$
 b: $1 \cdot 1 \cdot 0 \cdot 0 = 0$
 c: $1 \cdot 1 \cdot 0 \cdot 0 = 1$
 d: $1 \cdot 1 \cdot 0 \cdot 0 = 0$

Submit Quiz

5 out of 5

Type here to search

32°C Smoke 17:03 07-12-2021

CONCLUSION:

All the gates are realized using NAND and NOR gates and truth tables are verified.

PRACTICAL NO. 3 Implement combinational circuits.

A] Implementation of Boolean functions using MUX.

AIM: - Implementation of Boolean functions using MUX. It is a digital system experiment where a user can test circuits

THEORY: - Multiplexer: In electronics, a multiplexer (or MUX) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a data selector.

PRETEST: -

The screenshot shows a web browser window with the URL de-iitg.vlabs.ac.in/exp/boolean-functions-using-mux/pretest.html. The page title is "Implementation of Boolean functions using MUX". On the left, there is a sidebar with navigation links: Aim, Theory, Pretest (which is currently selected), Procedure, Simulation, Posttest, References, and Feedback. The main content area contains two questions. Question 1 asks: "It is possible for an enable or strobe input to undergo an expansion of two or more MUX ICs to the digital multiplexer with the proficiency of large number of:". The options are: a: Inputs, b: Outputs, c: Selection lines, and d: Enable lines. Question 2 asks: "A digital multiplexer is a combinational circuit that selects". The options are: a: One digital information from several sources and transmits the selected one, b: Many digital information and convert them into one, c: Many decimal inputs and transmits the selected information, and d: Many decimal outputs and accepts the selected information. At the bottom of the content area is a blue "Submit Quiz" button. Below the content area, it says "2 out of 2". The browser's address bar shows "DE JOURNAL" and the tabs bar shows "Virtual Labs". The taskbar at the bottom includes icons for File, Home, Task View, Edge, Mail, Google Chrome, Settings, and File Explorer. The system tray shows the date as 07-12-2021, the time as 21:59, and the weather as 28°C Haze.

PROCEDURE: -

Components used:

We used the following components for this experiment-

IC 74153(MULTIPLEXER)

IC 7404(NOT gate)

How to make connection:

After Starting the experiment first click on the Components button to get component list. Now you can Drag and Drop any component in the circuit designing area. To make connection between components , just click on the Blue bubble of any components and Drag it to another Blue bubble of the same or any other components. To delete connection or to remove any component use Double click on that component or connection.

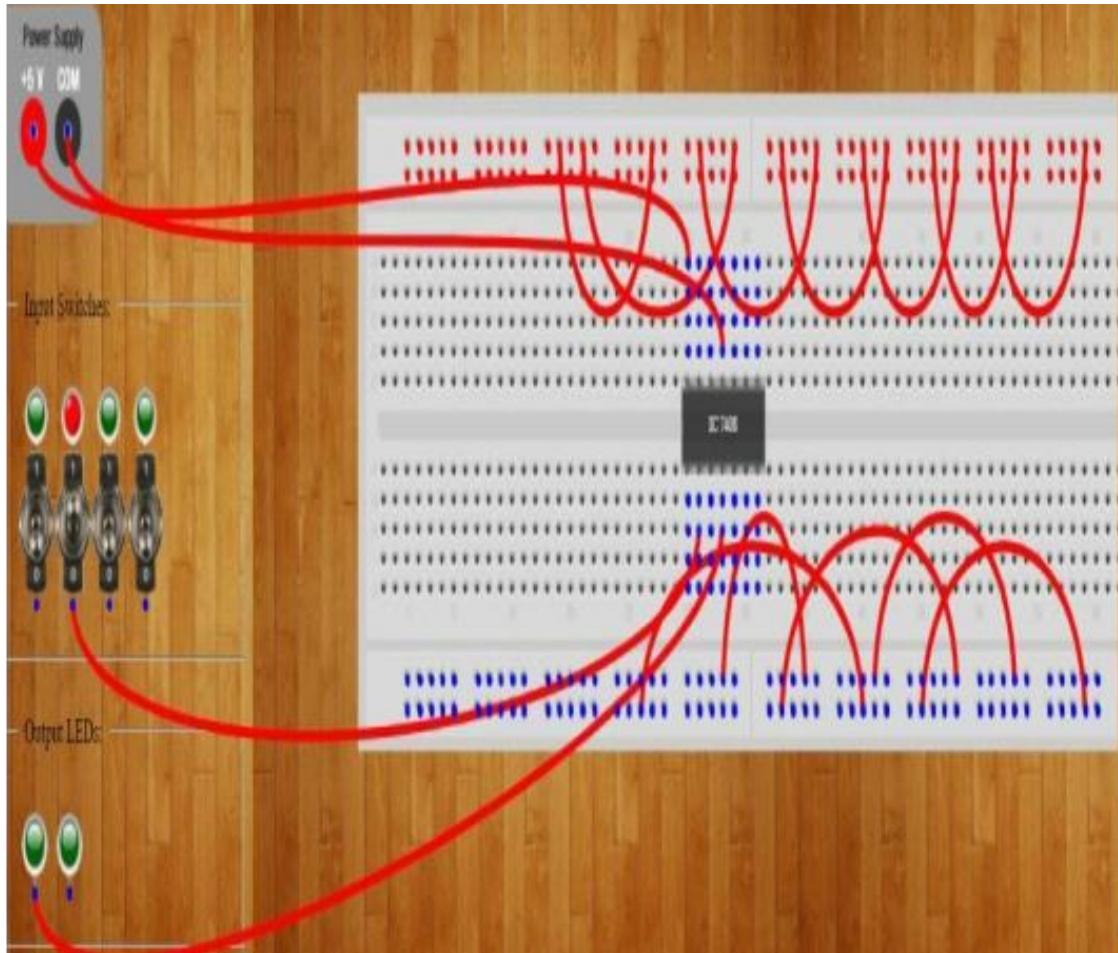
How to run:

Connect the Vcc and Ground pins of the ICs with the power supply. Now connect the input pins of the ICs with the Input Switches. Connect the output pins with output LEDs. Only pins with Blue bubbles can be used.

Green LEDs are used for indicating logic 0 and Red LEDs are used for logic 1.

After connecting all the required components, click on the Start button.

STIMULATION: -



POSTTEST: -

The screenshot shows a web browser window with the following details:

- Title Bar:** DE Journal, Virtual Labs
- Address Bar:** de-iitg.vlabs.ac.in/exp/boolean-functions-using-mux/posttest.html
- Header:** Virtual Labs, An MoU Govt of India Initiative, HOME, PARTNERS, CONTACT
- Content:** Electronics and Communication Engineering > Digital Electronics IITG > Experiments
- Section:** Implementation of Boolean functions using MUX.
- Questions:**
 - In a multiplexer, the selection of a particular input line is controlled by:
 - a: Data controller
 - b: Selected lines
 - c: Logic gates
 - d: Both data controller and selected lines
 - How many select lines would be required for an 8-line-to-1-line multiplexer?
 - a: 2
 - b: 4
 - c: 8
 - d: 3
 - Two input multiplexer would have?
 - a: 1 select line
 - b: 2 select lines
 - c: 4 select lines

The screenshot shows a web browser window with the following details:

- Title Bar:** DE Journal, Virtual Labs
- Address Bar:** de-iitg.vlabs.ac.in/exp/boolean-functions-using-mux/posttest.html
- Header:** Virtual Labs, An MoU Govt of India Initiative, HOME, PARTNERS, CONTACT
- Content:** Theory
- Section:** Implementation of Boolean functions using MUX.
- Questions:**
 - In a multiplexer, the selection of a particular input line is controlled by:
 - a: Data controller
 - b: Selected lines
 - c: Logic gates
 - d: Both data controller and selected lines
 - How many select lines would be required for an 8-line-to-1-line multiplexer?
 - a: 2
 - b: 4
 - c: 8
 - d: 3
 - Two input multiplexer would have?
 - a: 1 select line
 - b: 2 select lines
 - c: 4 select lines
 - d: 3 select lines
- Buttons:** Submit Quiz
- Status:** 3 out of 3

CONCLUSION:

A multiplexer, or MUX, is a circuit that selects a single output from multiple inputs. It has multiple uses. As seen above, Boolean function can be implemented using Boolean function.

B] Applications of Multiplexers

AIM: -The students will be able to understand the concept of applications of multiplexers and its use as universal circuits.

Students will be able:

1. To construct higher order multiplexers (4:1) using 2:1 Mux.

THEORY: -

1.1 Introduction

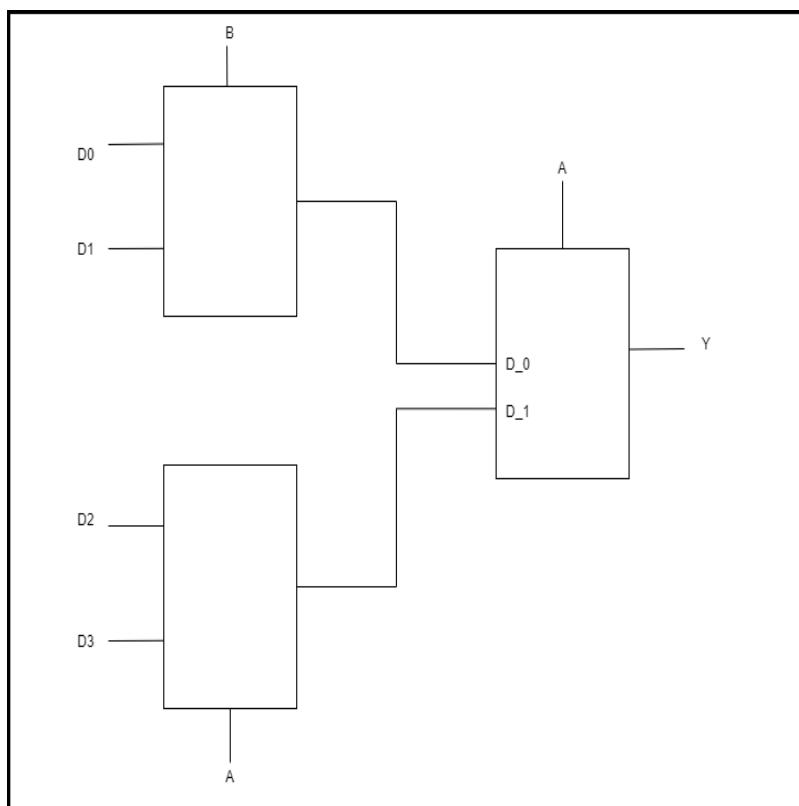
Whenever there is a need to connect multiple input devices, one at a time, to a system, then a digital combinational circuit called multiplexer is useful. 1.2. Example 1: Implementation of 4:1 Mux The expression for the output Y in case of a 4:1 multiplexer is given as:

$$Y = A' \cdot B' \cdot D_0 + A' \cdot B \cdot D_1 + A \cdot B' \cdot D_2 + A \cdot B \cdot D_3$$

Or

$$Y = A' \cdot (B' \cdot D_0 + B \cdot D_1) + A \cdot (B' \cdot D_2 + B \cdot D_3)$$

Construct the 4:1 mux as shown in figure Using simulator the circuit for 4:1 multiplexer can be constructed and verified.



Assignment: Realize the equation $Y = A' \cdot B + B' \cdot C' + A \cdot B \cdot C$ using a 4-to-1 multiplexer.

Solution: There are three variables in the given equation. Let variables A & B be used as select lines and variable C as the data input D. Express Y as a function of minterms of three variables as shown:

$$Y = A' \cdot B + B' \cdot C' + A \cdot B \cdot C$$

$$Y = A' \cdot B + B' \cdot C'(A' + A) + A \cdot B \cdot C$$

$$Y = A' \cdot B' \cdot C' + A' \cdot B \cdot 1 + A \cdot B' \cdot C' + A \cdot B \cdot C \quad \text{--- (iv)}$$

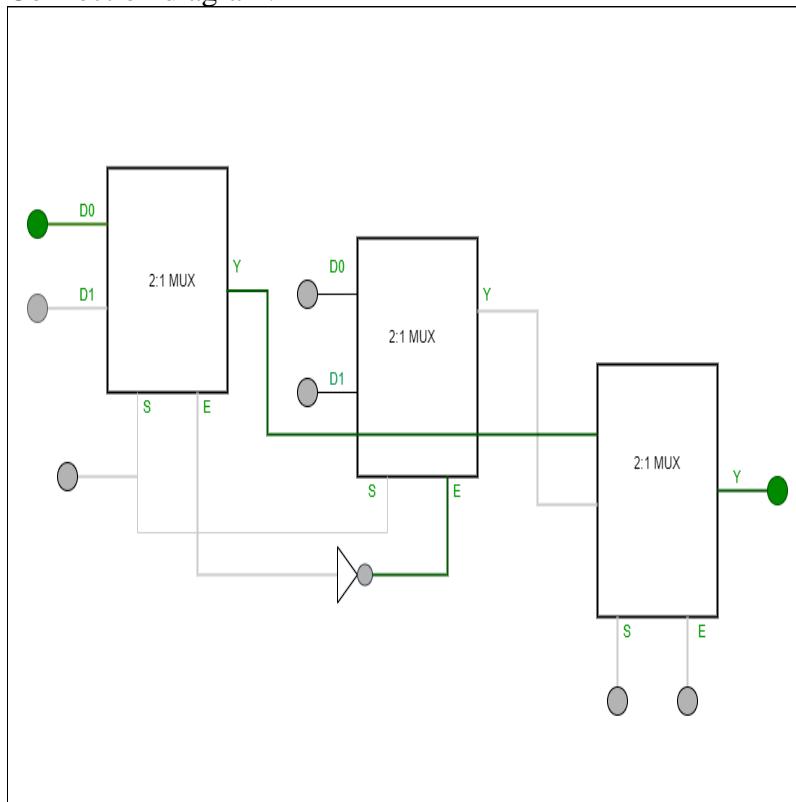
The logic equation for 4:1 Mux is:

$$Y = A' \cdot B' \cdot D_0 + A' \cdot B \cdot D_1 + A \cdot B' \cdot D_2 + A \cdot B \cdot D_3 \quad \text{--- (v)}$$

Comparing equations (iv) & (v),

$$D_0 = C', D_1 = 1, D_2 = C' \text{ and } D_3 = C$$

Connection diagram:



Function Table:

CONTROL I/Ps		OUTPUT	
Enable I/P	Select	I/P	O/P
E'	S0	S1	Y
1	X	X	0
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3

Using simulator, the circuit can be build and verified.

PRETEST: -

Atif Aslam: Pehli Dafa Song | DE JOURNAL | Virtual Labs

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Aim Applications of Multiplexers

Theory Multiplexers can also be referred to as universal circuit builders. State True or False.

Pretest a : True b : False

Procedure

Simulation A 2:1 Multiplexer is connected as shown in figure image000. What output Boolean expression is implemented by the circuit?

Posttest

References

Feedback

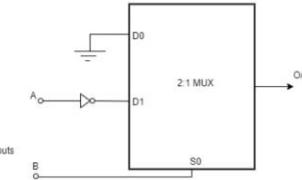
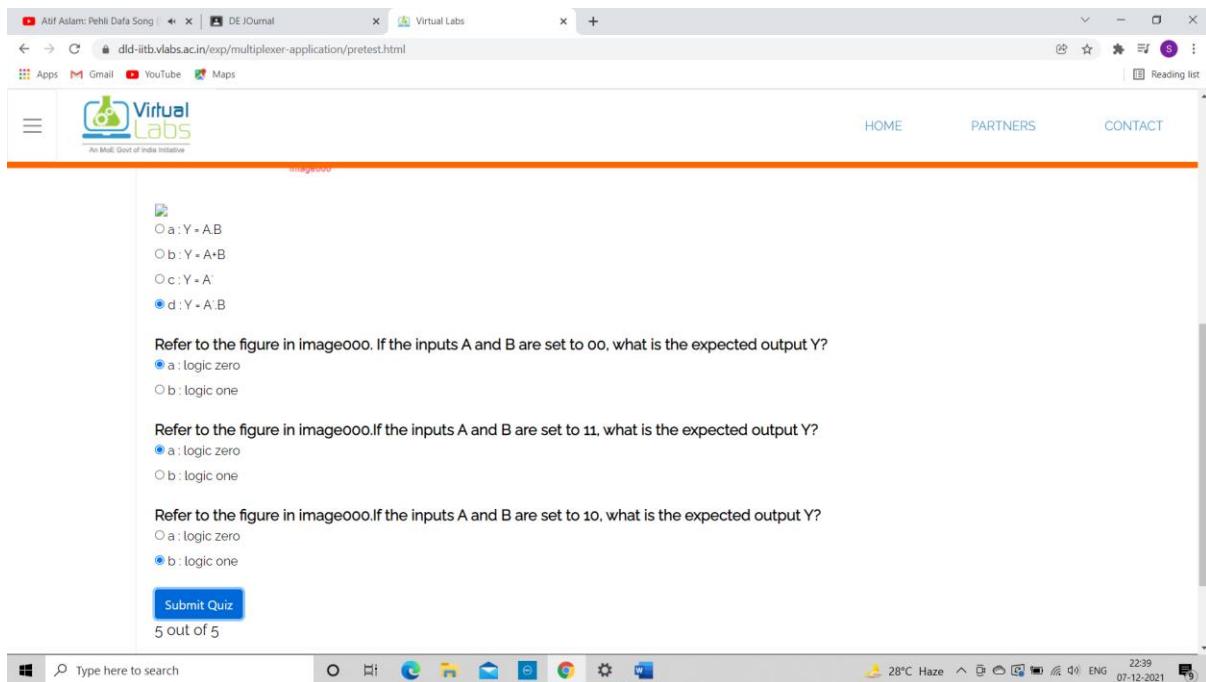
inputs 

image000

Type here to search 28°C Haze 22:39 07-12-2021



PROCEDURE: -

Simulation Screen 1

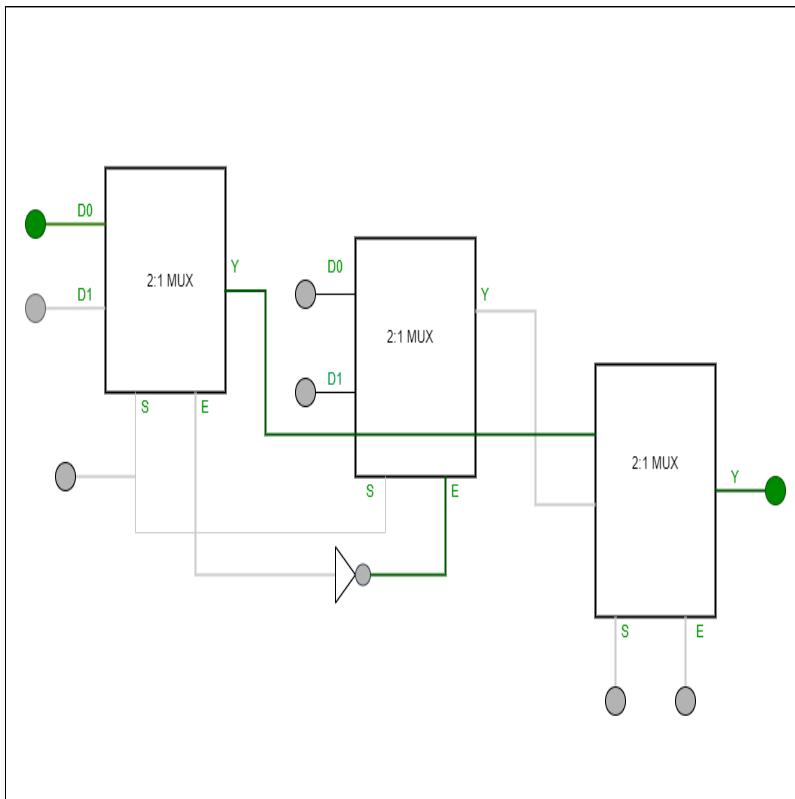
1. Click on the "Click to begin" button.

Simulation Screen 2: To formulate the Function table and Boolean Expression for 4:1 multiplexer

1. Select appropriate options from the drop down menu and formulate the function table.
2. Click submit.
3. If incorrect try again.
4. If correct, select the correct expression defining the multiplexer.
5. Try again if answer is incorrect.
6. Proceed to the next step. Click Next.

Simulation Screen 3: Simulator Scene - To design, construct and verify the operation of a 4:1 multiplexer using 2:1 multiplexers

Design of 4:1 Mux using 2:1 muxes (refer figure).



General instructions:

1. Click on the digital switch and click on the canvas.
2. Repeat step 1 for all required three inputs.
3. Click on the 2:1 mux symbol and then on the canvas at an appropriate position on the canvas.
4. Repeat for the placement of 2nd 2:1 mux.
5. Click on the 3rd 2:1 mux and then on the canvas to place it at an appropriate position on the canvas.
6. Use the connection tabs and make the necessary connections.
7. Connect the output to the LED.
8. Simulate and verify the circuit.

STIMULATION: -

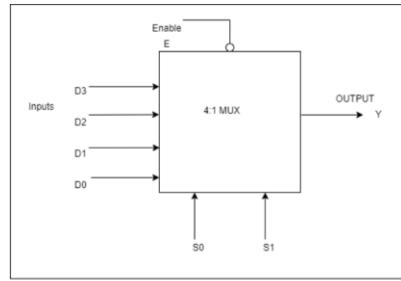


Applications of Multiplexers

Q1.Enter the outputs for a 4:1 MUX

CONTROL I/Ps		OUTPUT
Enable I/P	Select I/P	O/P
E'	S ₁	S ₀
1	X	X
0	0	0
0	0	1
0	1	0
0	1	1

SUBMIT TRY AGAIN



Q2.Select the correct Boolean expression for the above truth table.

E' S₁' S₀' D₀ + E' S₁' S₀ D₁ + E' S₁ S₀

NEXT

SCOREKEEPER

Question	Status
Q1.	Correct answer.Your score is 5
Q2.	Correct answer.Your score is 3



Available Gates

- MUX
- NOT
- Y-connector
- Green LED

Digital Switch

Connections

- Connect op1 to ip1
- Connect op1 to ip2
- Connect op2 to ip2
- Connect op2 to ip1
- Connect op1 to S
- Connect op1 to E
- Connect op2 to S
- Connect op2 to E

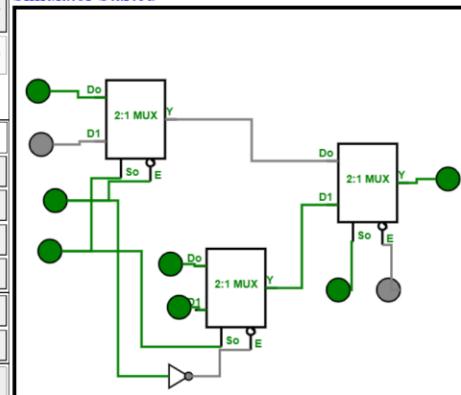
UNDO GATE **UNDO LINK**

REDO GATE **REDO LINK**

Start Simulator

Stop Simulator Reset

Simulator Started



CONTROL I/Ps		OUTPUT
Enable I/P	Select I/P	O/P
E'	S ₁	S ₀
1	X	X
0	0	0
0	0	1
0	1	0
0	1	1



POSTTEST:

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Virtual Labs

HOME **PARTNERS** **CONTACT**

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Applications of Multiplexers

A 2:1 Multiplexer is connected as shown in figure image002. What output Boolean expression is implemented by the circuit?

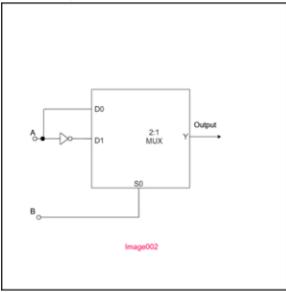


image002

a: $Y = A \cdot B$
 b: $Y = A + B$

Type here to search

29°C Polluted air 20:14 02-01-2022

Virtual Labs

HOME **PARTNERS** **CONTACT**

c: $Y = A \oplus B$
 d: $Y = A \text{XNOR } B$

Refer to the figure in image002. If the inputs A and B are set to 00, what is the expected output Y?
 a: logic zero
 b: logic one

Refer to the figure in image002. If the inputs A and B are set to 01, what is the expected output Y?
 a: logic zero
 b: logic one

Refer to the figure in image002. If the inputs A and B are set to 10, what is the expected output Y?
 a: logic zero
 b: logic one

Refer to the figure in image002. If the inputs A and B are set to 11, what is the expected output Y?
 a: logic zero
 b: logic one

Submit Quiz

5 out of 5

Type here to search

29°C AQI 205 20:14 02-01-2022

CONCLUSION:

A multiplexer is constructed using inputs and select lines and a 4:1 multiplexer can be created using three 2:1 multiplexer.

C] Design of four variable function using MSI ICs

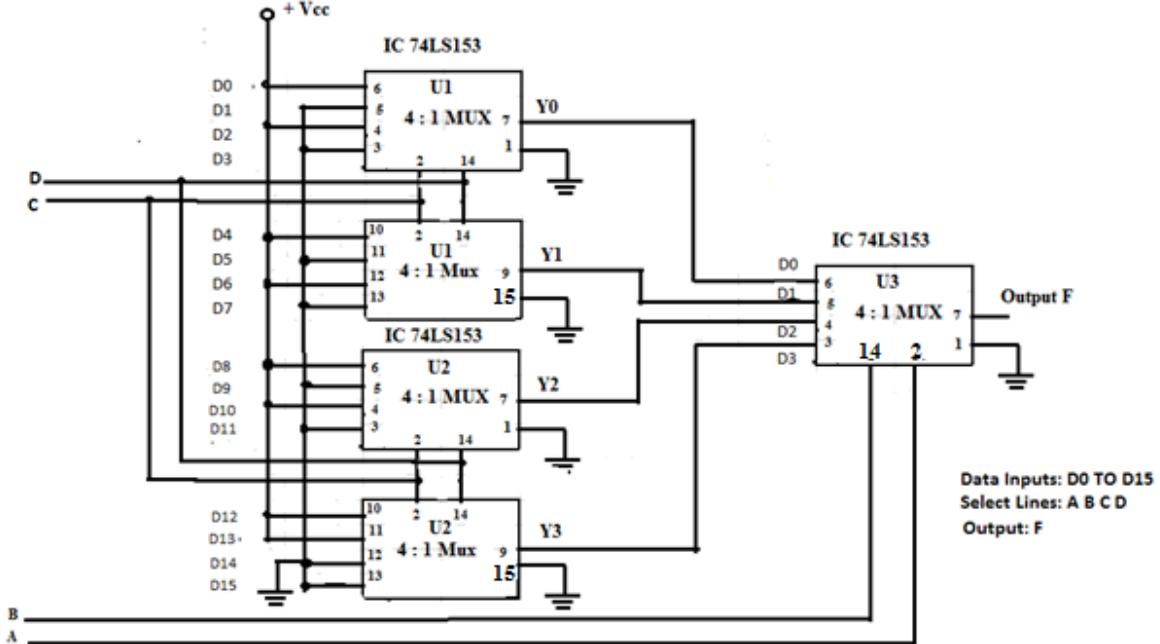
AIM: To design and implement the given 4 variable function using IC74LS153. Verify its Truth-Table.

THEORY: In Experiment # 1 we studied the use of IC 74LS153 as a Dual 4:1 MUX. We required only one IC 74LS153 to design an 8:1 MUX. Now in Experiment # 2 we shall study the application of MUX as a Universal Logic Generator for a four variable system. To implement a four variable Universal Function Generator using IC 74LS153, let us follow these steps:

1. To begin with we shall design a 16:1 MUX using multiple 74LS153 ICs.
2. IC 74LS153 has two 4:1 MUXes within it. Each MUX has two select lines and four data input lines. So in all eight data inputs D0 to D7 are available from first IC # U1.
3. Similarly second IC # U2 provides another eight data inputs D8 to D15. Thus in all 16 input lines are now available.
4. To select one of these 16 data inputs, we need four select lines; let us label them as A B C & D; where A is the most significant bit (MSB) and D is the least significant bit (LSB).
5. Each IC has 4 select lines.
6. All eight select lines of U1 and U2 are connected together and labelled as select lines C & D.
7. To get additional two select lines we introduce the third IC U3. Only one 4:1 MUX of U3 is required. The two select lines of U3 are labelled as A & B.
8. The strobe inputs 1G' & 2G' of all the ICs is connected to ground so that all the IC's are enabled for operation.
9. Thus in all we need three IC's for the design of 4-variable Universal Logic Generator as shown in Fig.1.

The output function F can be written as:

$$F = A'.B'.Y_0 + A'.B.Y_1 + A.B'.Y_2 + A.B.Y_3$$



The function table for Fig.1 is given below. Any four variable logic function can be derived using this circuit.

Decimal	Select Lines				Output
Equivalent	A	B	C	D	Y
0	0	0	0	0	D0
1	0	0	0	1	D1
2	0	0	1	0	D2
3	0	0	1	1	D3
4	0	1	0	0	D4
5	0	1	0	1	D5
6	0	1	1	0	D6
7	0	1	1	1	D7
8	1	0	0	0	D8
9	1	0	0	1	D9
10	1	0	1	0	D10
11	1	0	1	1	D11
12	1	1	0	0	D12
13	1	1	0	1	D13
14	1	1	1	0	D14
15	1	1	1	1	D15

For the logic diagram & pin diagram, refer Fig.2. The function table for IC 74LS153 is also provided for reference.

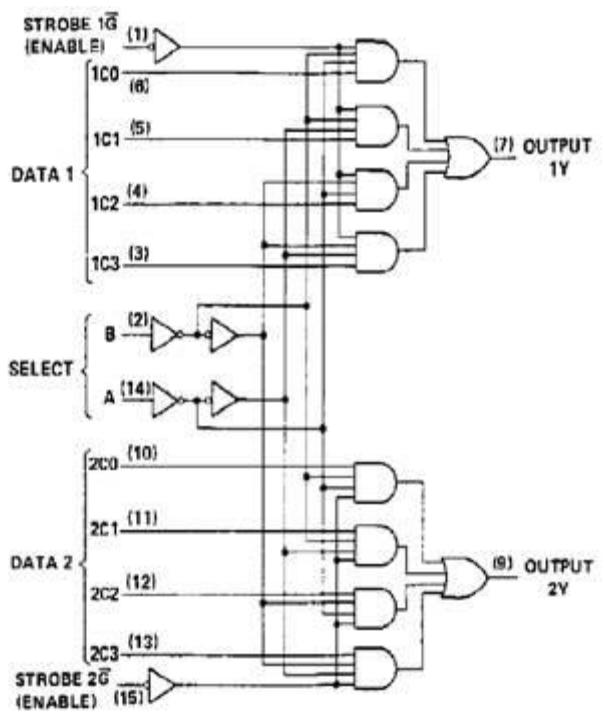
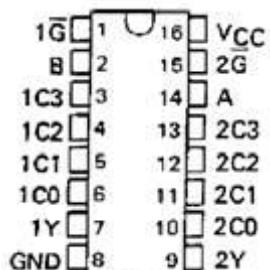


Fig. 2. Internal Structure & Pin Diagram of IC74LS153



1	0	1	1	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	1
1	1	0	0	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	0
1	1	0	0	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	1
1	1	0	1	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	0
1	1	0	1	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	1
1	1	1	0	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	0
1	1	1	0	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	1
1	1	1	1	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	0
1	1	1	1	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	1

(Where : ‘1’ indicate VCC/+5V, ‘0’ indicate 0V, ‘X’ indicate “don’t care“)

Here we have implemented $\sum m(0,2,4,6,8,10,12,13)$:-
 D0, D2, D4, D6, D8, D10, D12, D13 to GND(Ground)
 and D1, D3, D5, D7, D9, D11, D14, D15 to VCC(+5V)

So the resultant truth table is :

Inputs				Output
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

PRETEST:

Design of four variable function using MSI ICs

Multiplexer has

a : Many input one output.
 b : Many input many output.
 c : One input many output
 d : One input one output.

How many 4:1 MUX required to design 16:1 MUX?

a : 2
 b : 3
 c : 4
 d : 5

Which IC is required to design 16:1 MUX?

a : 74LS191
 b : 74LS153
 c : 74LS00

Which IC is required to design 16:1 MUX?

a : 74LS191
 b : 74LS153
 c : 74LS00
 d : 74LS93

How many inputs and outputs are there in 16:1 Multiplexer?

a : 16 inputs 1 output.
 b : 8 inputs 16 output.
 c : 4 inputs 1 output.
 d : 1 inputs 16 output.

Applications of Multiplexer are -----

a : in Communication System
 b : in Computer memory
 c : in Telephone Network
 d : All of the above

Submit Quiz

5 out of 5

PROCEDURE:

Steps:

1. Switch ON the circuit, by pressing Main Switch.
2. Set the appropriate Inputs.
3. After giving the inputs, observe the corresponding outputs.
4. Verify the results.

Inputs:

1. Select Line A : I/P15
2. Select Line B : I/P14
3. Select Line C : I/P13
4. Select Line D : I/P12

Outputs:

1. Output (O/P15)

STIMULATION:

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Lecture 19" and displays a truth table titled "Design of four variable function using MSI ICs". The table has columns for I/P15 through O/P15. The data is as follows:

I/P15	I/P14	I/P13	I/P12	O/P15
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

POSTTEST:

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Lecture 19" and displays a posttest for "Design of four variable function using MSI ICs". The posttest consists of several questions with multiple choice answers.

Aim: Design of four variable function using MSI ICs

Theory: 16:1 MUX have how many select lines?

a:1
 b:2
 c:3
 d:4

Pretest: To design 16:1 MUX using IC 74LS153 how many such IC's are required?

a:1
 b:2
 c:3
 d:4

Procedure: If select lines are $S_3 S_2 S_1 S_0 = 1111$ then which input is selected by 16:1 multiplexer?

a: D₁₅
 b: D₅
 c: D₈

If select lines are $S_3 S_2 S_1 S_0 = 1111$ then which input is selected by 16:1 multiplexer?

a : D₁₅
 b : D₅
 c : D₈
 d : D₄

A combinational circuit that selects one from many inputs is _____.

a : Encoder
 b : Decoder
 c : Demux
 d : Mux

The number of select lines for 32:1 multiplexer is _____.

a : 4
 b : 5
 c : 16
 d : 3

Submit Quiz

5 out of 5

Conclusion:

As seen in the above stimulation, 4 variable function can be designed and implemented using IC74LS153. And the its truth table was also verified.

PRACTICAL NO. 4 Implement code converters Implement combinational circuits.

A] Design of Gray to Binary code converter using MSI ICs

AIM: To design and implement 3-bit Gray to Binary code converter using IC-74LS138.

THEORY: Grey code

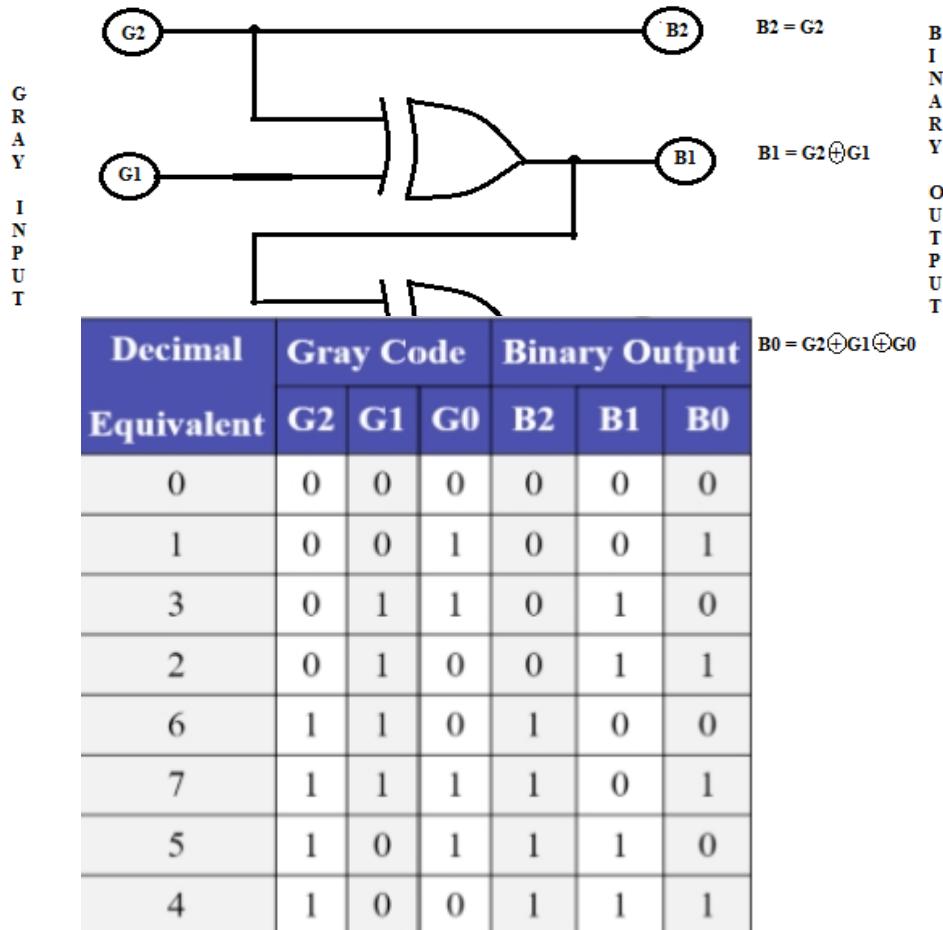
The Gray code is unweighted code. Gray code exhibits only a single bit change from one code word to the next in sequence. Due to this specific feature Gray code is important in applications such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

Let us design a 3-bit Gray to Binary code converter and implement using IC-74LS138.

The relation between these two codes can be understood from the following diagram and equations:

Fig. 1. Gray Code to Binary Conversion

From the truth table it is observed that the output B2 is HIGH for minterms m4, m5, m6 and m7. Hence equation defining B2 output can be written as: $B2 = \Sigma m(4, 5, 6, 7)$. The output B1 is HIGH for minterms m2 and m3 m4 & m5. Hence equation defining B1 output can be



written as: $B1 = \Sigma m(2, 3, 4, 5)$. The output B0 is HIGH for minterms m1, m2 m4 m7. Hence equation defining B0 output can be written as: $B0 = \Sigma m(1, 2, 4, 7)$. The Binary outputs for given 3 bit Gray code are summarized as follows:

$$B2 = \Sigma m(4, 5, 6, 7)$$

$$B1 = \Sigma m(2, 3, 4, 5)$$

$$B0 = \Sigma m(1, 2, 4, 7)$$

These equations can be implemented by using IC74LS138 as a 3:8 decoder as shown in Fig.2. Every Binary output has 4-minterms each. Hence IC74LS20- four input NAND gate ICs are required to produce the final Binary equivalent.

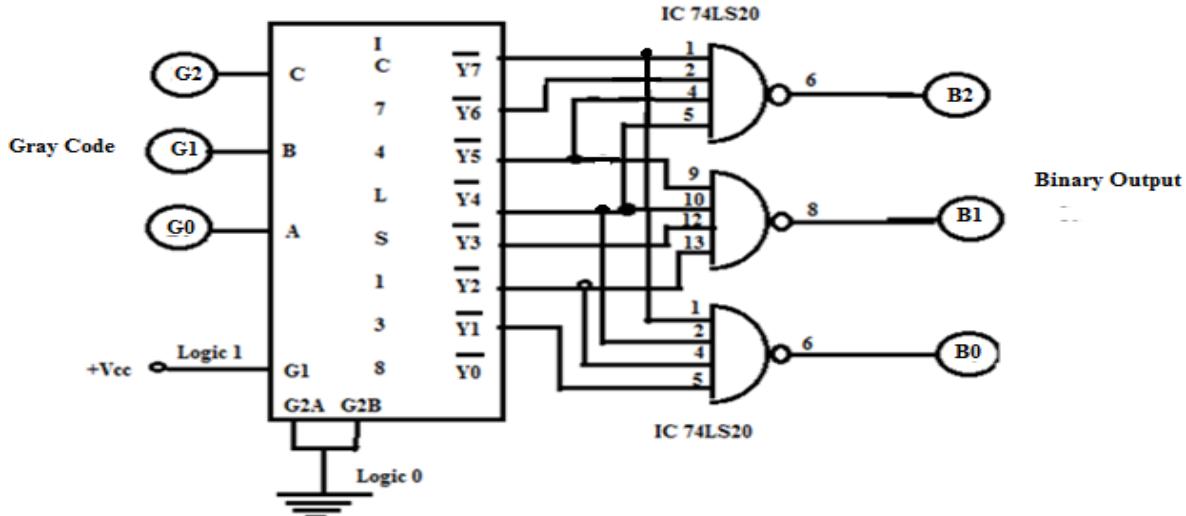


Fig.2. Implementation of Gray code to Binary Converter using IC 74LS138 Decoder
The logic diagram, connection diagram and function table for IC 74LS138 are given below:

Function Table
LS138

Inputs			Outputs							
Enable	Select		Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
G1	G2*	C	B	A						
X	H	X	X	X	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	L

* $G2 = G2A + G2B$

H = High Level, L = Low Level, X = Don't Care

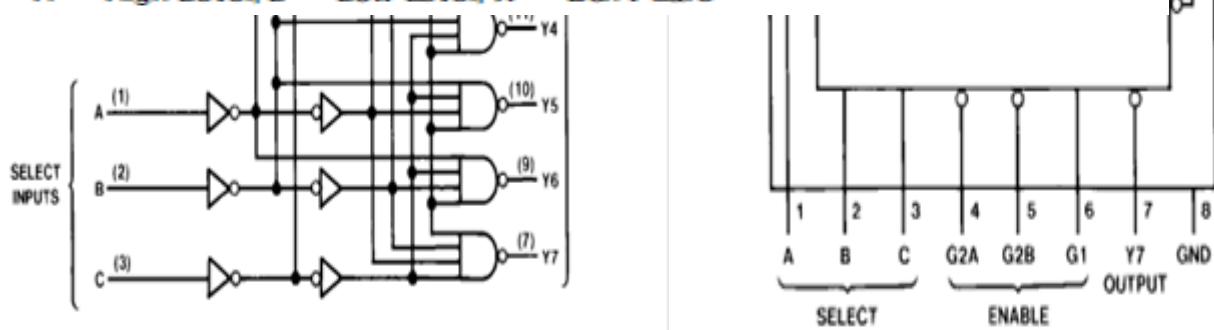


Fig.3. Logic Diagram, Connection Diagram & Function Table of IC 74LS138.

Numerical :

G2, G1, G0

Outputs=> B2,B1,B0

B2=G2

$$B_0 = G_2 \oplus G_1 \oplus G_0 \quad G_1 \\ (\oplus \Rightarrow Ex - Or Operation)$$

Ex- OR operation table:

Truth table :

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Examples:

- Input “011”

$$G_2 = B_2 = G_2 G_1 \oplus G_0 = 1, \quad G_0 = 1 \\ B_1 = G_2 \oplus G_1 = 0 \oplus 1 = 1 \\ B_0 = G_2 \oplus G_1 \oplus G_0 = 0 \oplus 1 \oplus 1 = 1 \oplus 1 = 0 \\ \text{Output “010”} \\ \text{• Input “110”}$$

$$G_2 = 1, \quad G_1 = 1, \quad G_0 = 0 \\ B_1 = G_2 \oplus G_1 = 1 \oplus 1 = 0 \\ B_0 = G_2 \oplus G_1 \oplus G_0 = 1 \oplus 1 \oplus 0 = 0 \oplus 0 = 0 \\ \text{Output “100”}$$

- Input “111”

$$G_2 = B_2 = G_2 \oplus G_1 \oplus G_0 = 1 \oplus 1 \oplus 1 = 0 \oplus 1 = 1 \\ B_1 = G_2 \oplus G_1 = 1 \oplus 1 = 0 \\ B_0 = G_2 \oplus G_1 \oplus G_0 = 1 \oplus 1 \oplus 1 = 0 \oplus 1 = 1 \\ \text{Output “101”}$$

PRETEST:

video1185401918.mp | DE Journal | Virtual Labs | Virtual Labs | Lecture 19 | (1) WhatsApp | DE assignment yashu | +

HOME PARTNERS CONTACT

References

Feedback

3-bit Gray to Binary code converter can be easily implemented using which of the following IC's ?

a : 74LS138 and Additional gates
 b : 74LS93 and Additional gates
 c : 74LS90 and Additional gates
 d : 74LS76

Gray code = 111. Binary Number = 101 State that above conversion is correct or not?

a : correct.
 b : incorrect.

Gray code = 011. Binary Number = 001 State that above conversion is correct or not?

a : correct.
 b : incorrect.

Submit Quiz

5 out of 5

Type here to search 28°C Smoke 21:47 02-01-2022

video1185401918.mp | DE Journal | Virtual Labs | Virtual Labs | Lecture 19 | (1) WhatsApp | DE assignment yashu | +

HOME PARTNERS CONTACT

Aim

Theory

Pretest

Which of the following is binary number?

a : 10100.
 b : 30
 c : 2E6
 d : None of the above

Procedure

Simulation

Posttest

References

Feedback

The Gray code is ___ code?

a : Unweighted.
 b : Unit Distance.
 c : Both of the above.
 d : None of the above

3-bit Gray to Binary code converter can be easily implemented using which of the following IC's ?

a : 74LS138 and Additional gates
 b : 74LS93 and Additional gates
 c : 74LS90 and Additional gates

Type here to search 28°C Smoke 21:47 02-01-2022

PROCEDURE:

Steps:

1. Switch ON the circuit, by pressing Main Switch.
2. Set the appropriate Inputs.
3. After giving the inputs, observe the corresponding outputs.
4. Verify the results.

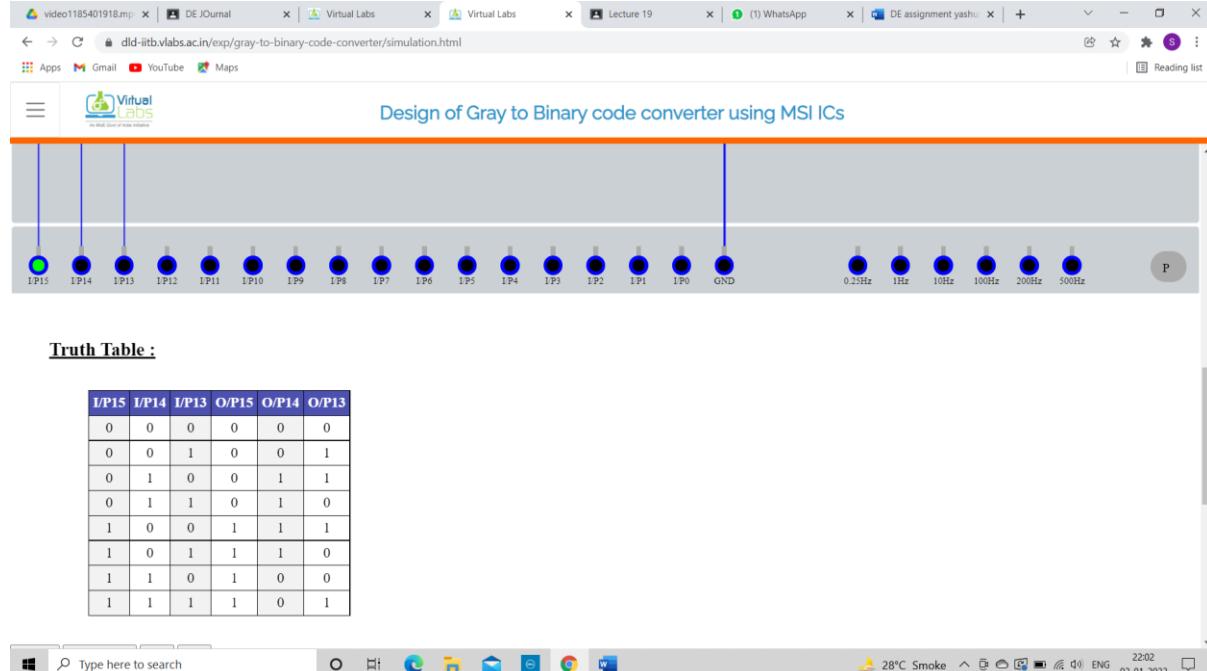
Inputs:

1. Gray Code (I/P15 to I/P13)

Outputs:

1. Binary Code (O/P15 to O/P13)

STIMULATION:



Truth Table :

I/P15	I/P14	I/P13	O/P15	O/P14	O/P13
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	1

POSTTEST:

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Design of Gray to Binary code converter using MSI ICs

A binary code that progresses such that only one bit changes between two successive codes is _____.

a : Nine's-complement code.
 b : 8421 code
 c : Excess-3 code
 d : Gray code

Why is the Gray code more practical to use when coding the position of a rotating shaft?

a : All digits change between counts.
 b : Two digits change between counts.
 c : Only one digit changes between counts.
 d : None of the above

The primary use for Gray code is_____.

a : Coded representation of a shaft's mechanical position.
 b : Turning on/off software switches.
 c : To represent the correct ASCII code to indicate the angular position of a shaft on rotating machinery.

FEEDBACK

c : Only one digit changes between
 d : None of the above

The primary use for Gray code is_____1

a : Coded representation of a shaft's mechanical position.
 b : Turning on/off software switches.
 c : To represent the correct ASCII code to indicate the angular position of a shaft on rotating machinery.
 d : to convert the angular position of a shaft on rotating machinery into hexadecimal code.

In a Gray code, each number is 3 greater than the binary representation of that number.

a : True
 b : False

What is the equivalent Binary value of Gray code 100?

a : 010
 b : 111
 c : 101
 d : 000

Submit Quiz

5 out of 5

Type here to search

28°C Smoke 22:04 02-01-2022

CONCLUSION:

Thus, a 3-bit gray to binary code converter can be designed and implemented using IC-74LS138.

B] Design of Binary to Gray code converter using MSI ICs

AIM: To design and implement 3-bit Binary to Gray code converter using IC-74LS138

THEORY: Binary Code: It is weighted code i.e. it is a code in which weight is assigned to every symbol position in the code word. The positional weights in binary code are shown below:

Binary Code :----> 2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4} ...

Decimal -----> 16 8 4 2 1 $1/2$ $1/4$ $1/8$ $1/16$...

Gray Code: It is a non-weighted code i.e. it does not have any specific/fixed weight assigned to each symbol position in the code word. The unique feature of Gray code is that at a time only "one" bit changes In other words, in Gray code every new code differs from the previous in terms of one single bit.

Use of Gray Code: For correct measurement of angular position of shaft.

Design: The Binary and their equivalent Gray Codes are related as shown in Fig.1

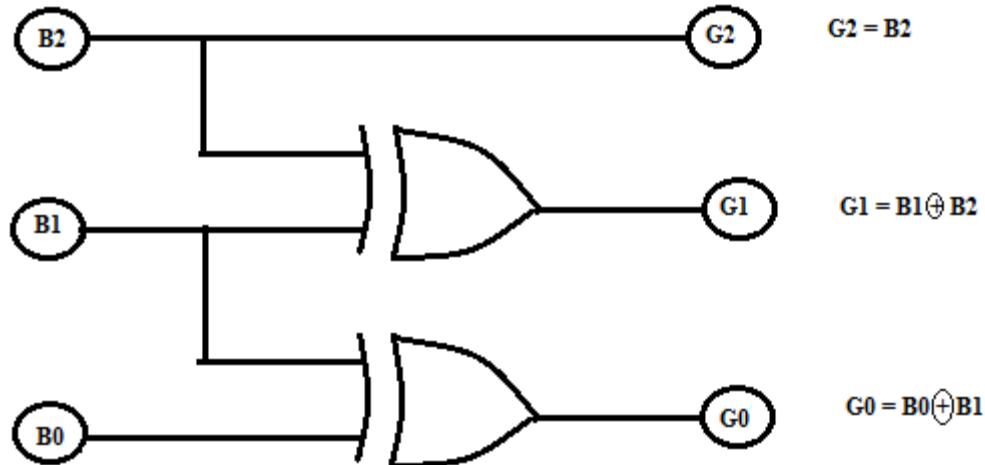


Fig.1. Logic Diagram showing relation between Binary and Gray Code.

Let us now prepare the 3 bit- Binary to 3 bit Gray code truth table:

Decimal Equivalent	Binary			Gray Code		
	B2	B1	B0	G2	G1	G0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

From the truth table it is observed that the output G2 is HIGH for minterms m4, m5, m6 and m7. Hence equation defining G2 output can be written as: $G2 = \Sigma m(4, 5, 6, 7)$. The output G1 is HIGH for minterms m2 and m3 m4 & m5. Hence equation defining G1 output can be written as: $G1 = \Sigma m(2, 3, 4, 5)$. The output G0 is HIGH for minterms m1, m2 m5 m6. Hence equation defining G1 output can be written as: $G0 = \Sigma m(1, 2, 5, 6)$. The Gray outputs for given 3 bit Binary are summarized as follows:

$$G2 = \Sigma m(4, 5, 6, 7)$$

$$G1 = \Sigma m(2, 3, 4, 5)$$

$$G0 = \Sigma m(1, 2, 5, 6)$$

The above Boolean expressions can be implemented using IC 74LS138 as a 3:8 decoder by just connecting its relevant outputs to 4- input NAND gates as shown in Fig. 1. Four input IC74LS20 is used to produce the final Gray code equivalent.

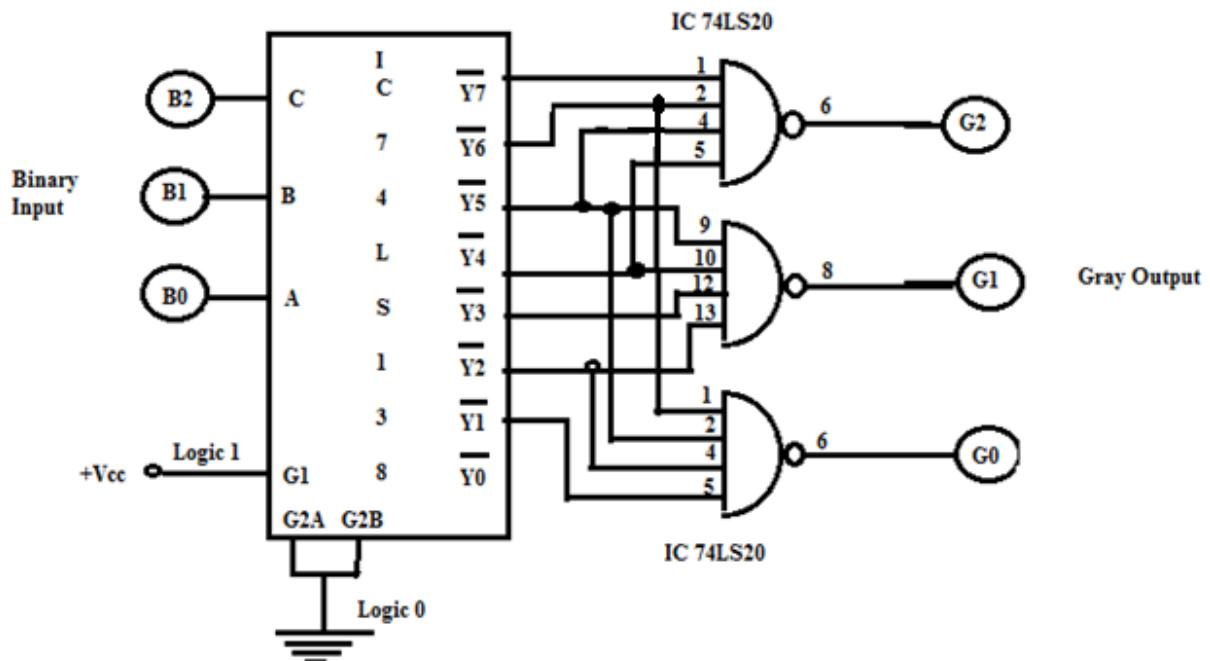


Fig. 2. Binary to Gray Code Converter using IC 74LS138

The logic diagram, connection diagram and function table for IC 74LS138 are given below:

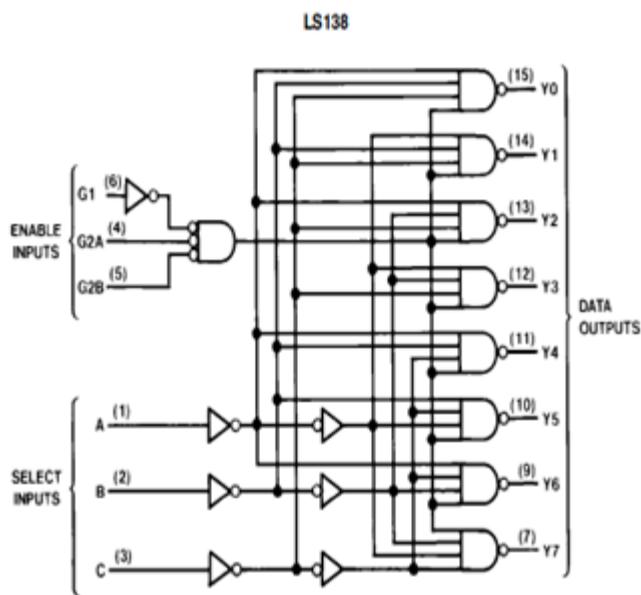


Fig.3. Logic Diagram, Connection Diagram & Function Table of IC 74LS138.

Numerical:

Formulas: Inputs $\Rightarrow B_2, B_1, B_0$

Outputs $\Rightarrow G_2, G_1, G_0$

$$G_2 = B_2$$

$$G_1 = B_2 \oplus B_1$$

$$G_0 = B_1 \oplus B_0$$

($\oplus \Rightarrow$ Ex-Or Operation)

Ex-Or Operation Table:

$$Y = A \oplus B$$

Truth table:

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Examples:

Input "011"

$$B_2 = 0, \quad B_1 = 1, \quad B_0 = 1$$

$$G_2 = B_2 = 0$$

$$G_1 = B_2 \oplus B_1 = 0 \oplus 1 = 1$$

$$G_0 = B_1 \oplus B_0 = 1 \oplus 1 = 0$$

Output = "010"

Input "110"

$$B_2 = 1, \quad B_1 = 1, \quad B_0 = 0$$

$$G_2 = B_2 = 1$$

$$G_1 = B_2 \oplus B_1 = 1 \oplus 1 = 0$$

$$G_0 = B_1 \oplus B_0 = 1 \oplus 0 = 1$$

Output = "101"

Input "111"

$$B_2 = 1, \quad B_1 = 1, \quad B_0 = 1$$

$$G_2 = B_2 = 1$$

$$G_1 = B_2 \oplus B_1 = 1 \oplus 1 = 0$$

$$G_0 = B_1 \oplus B_0 = 1 \oplus 1 = 0$$

Output = "100"

PRETEST:

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Design of Binary to Gray code converter using MSI ICs

Q1: The weight of the LSB as a binary number is:
 a : 2^0 .
 b : 2^1 .
 c : 10^1 .
 d : 10^0 .

Q2: A binary number's value changes most drastically when the _____ is changed.
 a : LSB.
 b : Duty cycle.
 c : MSB.
 d : Frequency.

Q3: The _____ code is a non-weighted binary code.
 a : Excess-3
 b : Gray
 c : Multiple bit

References

Feedback

Q3: The _____ code is a non-weighted binary code.
 a : LSB.
 b : Duty cycle.
 c : MSB.
 d : Frequency

Q4: Binary Number = 101 Gray code = 111 State that above conversion is correct or not?
 a : Correct
 b : In-correct

Q5: Binary Number = 001 Gray code = 011 State that above conversion is correct or not?
 a : Correct
 b : In-correct

Submit Quiz

5 out of 5

PROCEDURE:

Steps:

Switch ON the circuit, by pressing Main Switch.

Set the appropriate Inputs.

After giving the inputs, observe the corresponding outputs.

Verify the results.

Inputs:

Binary Code (I/P15 to I/P13)

Outputs:

Gray Code (O/P15 to O/P13)

STIMULATION:

The screenshot shows a web browser window titled "Design of Binary to Gray code converter using MSI ICs". The page includes a header with "Virtual Labs" and a navigation bar with "DE Journal" and "Virtual Labs". Below the header, there are sections for "Inputs" and "Outputs".

Inputs:

- B2 : I/P15
- B1 : I/P14
- B0 : I/P13

Outputs:

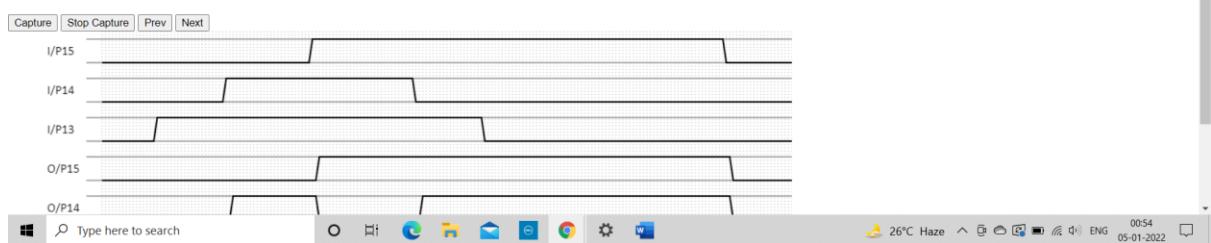
- G2 : O/P15
- G1 : O/P14
- G0 : O/P13

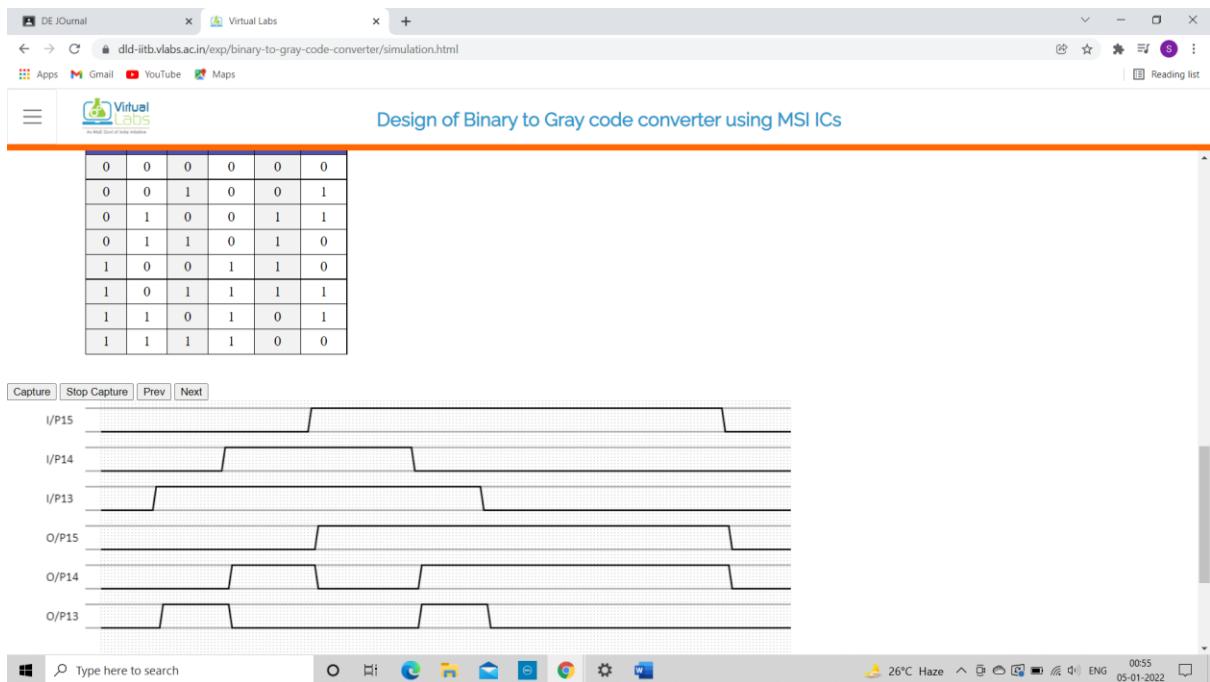
Below the input and output lists are several buttons: "Show Datasheet", "Cancel", "Generate Truth Table", and "Debug Mode".

The main area displays a logic diagram with 16 pins labeled I/P15 through O/P13 and VCC. A digital display shows "0 0" with a green "Main Switch" button. The Windows taskbar at the bottom shows the browser window, a search bar, and various system icons.

Truth Table :

I/P15	I/P14	I/P13	O/P15	O/P14	O/P13
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0





POSTTEST:

The screenshot shows a 'Posttest' section for the experiment. On the left is a sidebar with links: Aim, Theory, Pretest, Procedure, Simulation, Posttest (which is currently selected), References, and Feedback. The main content area has a title 'Design of Binary to Gray code converter using MSI ICs'.

Q1. Convert the binary number 1100 to Gray code.
 a : 0011
 b : 1010
 c : 1100
 d : 1001

Q2. Gray code is used in devices which convert analog quantities to digital signal because it is
 a : Error Free.
 b : much simpler than binary code.
 c : superior to Excess-3 code.
 d : None of the above

Q3. The code where all successive numbers differ from their preceding number by single bit is
 a : Binary code.
 b : BCD
 c : Excess - 3

At the bottom is a Windows taskbar with icons for Start, Search, Task View, Edge, File Explorer, Mail, Google Chrome, Settings, and File Explorer. The system tray shows the date (05-01-2022), time (00:59), and battery level (26°C Haze).

DE Journal × **Virtual Labs** × +

ddl-iitb.vlabs.ac.in/exp/binary-to-gray-code-converter/posttest.html

App Gmail YouTube Maps

Virtual Labs
An MoU Govt of India Initiative

HOME PARTNERS CONTACT

Q3. The code where all successive numbers differ from their preceding number by single bit is
 a : Binary code.
 b : BCD
 c : Excess - 3
 d : Gray.

Q4. What is the equivalent Gray code of Binary number 011?
 a : 010
 b : 111
 c : 101
 d : 000

Q5. What is the equivalent Gray code of Binary number 111?
 a : 011
 b : 101
 c : 100
 d : 100

Submit Quiz

5 out of 5

Type here to search

26°C Haze 00:59 05-01-2022

CONCLUSION:

Thus, a 3-bit binary to gray code converter can be designed and implemented using IC-74LS138.

C] Verify Binary to Gray and Gray to Binary conversion using NAND gates only.

AIM: To analyse the truth table of binary to gray and gray to binary converter using combination of NAND gates and to understand the working of binary to gray and gray to binary converter with the help of LEDs display

THEORY:

Introduction

Binary Numbers is default way to store numbers, but in many applications binary numbers are difficult to use and a variation of binary numbers is needed. Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit (binary digit). Gray codes are very useful in the normal sequence of binary numbers generated by the hardware that may cause an error or ambiguity during the transition from one number to the next. So, the Gray code can eliminate this problem easily since only one bit changes its value during any transition between two numbers.

Gray code has property that two successive numbers differ in only one bit because of this property gray code does the cycling through various states with minimal effort and used in K-maps, error correction, communication etc.

In computer science many a times we need to convert binary code to gray code and vice versa. This conversion can be done by applying following rules :

1) Binary to Gray conversion :

1. The Most Significant Bit (MSB) of the gray code is always equal to the MSB of the given binary code.
2. Other bits of the output gray code can be obtained by Ex-ORing binary code bit at that index and previous index.

There are four inputs and four outputs. The input variable are defined as B_3, B_2, B_1, B_0 and the output variables are defined as G_3, G_2, G_1, G_0 . From the truth table, combinational circuit is designed. The logical expressions are defined as :

$$B_3 = G_3$$

$$B_2 \oplus B_3 = G_2$$

$$B_1 \oplus B_2 = G_1$$

$$B_0 \oplus B_1 = G_0$$

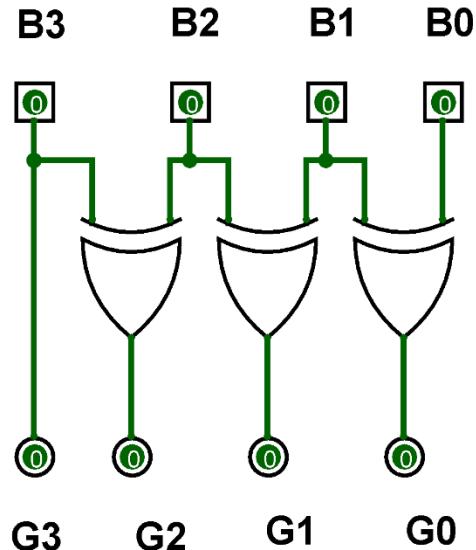


Figure-1: Binary to Gray Code Converter Circuit

Natural-binary code				Gray code			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Figure-2: Binary to Gray Code Converter Truth Table

2) Gray to binary conversion :

- 1.The Most Significant Bit (MSB) of the binary code is always equal to the MSB of the given binary number.
- 2.Other bits of the output binary code can be obtained by checking gray code bit at that index. If current gray code bit is 0, then copy previous binary code bit, else copy invert of previous binary code bit.

There are four inputs and four outputs. The input variable are defined as G_3, G_2, G_1, G_0 and the output variables are defined as B_3, B_2, B_1, B_0 . From the truth table, combinational circuit is designed. The logical expressions are defined as :

$$G_0 \oplus G_1 \oplus G_2 \oplus G_3 = B_0$$

$$G_1 \oplus G_2 \oplus G_3 = B_1$$

$$G_2 \oplus G_3 = B_2$$

$G_3 =$

B_3

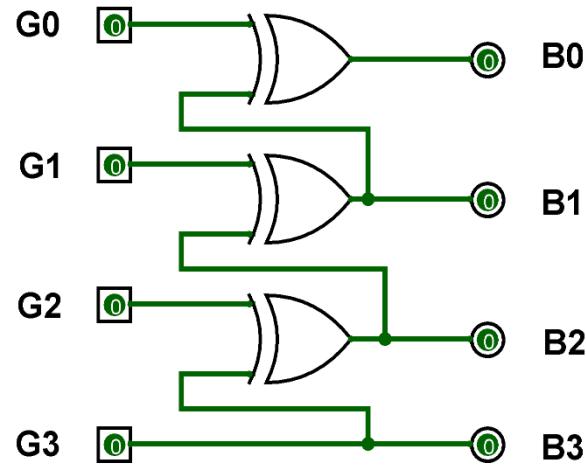


Figure-3: Gray to Binary Code Converter Circuit

Gray code				Natural-binary code			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

Figure-4: Gray to Binary Code Converter Truth Table

PRETEST:

DE Journal Virtual Labs

de-iitr.labs.ac.in/exp/binary-conversion/pretest.html

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Electronics IITR > Experiments

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Verify Binary to Gray and Gray to Binary conversion using NAND gates only.

A code converter is a logic circuit that _____.

a: Inverts the given input
 b: Converts into decimal number
 c: Converts data of one type into another type
 d: Converts to octal

The primary use for Gray code is _____.

a: Coded representation of a shaft's mechanical position
 b: Turning on/off software switches
 c: To represent the correct ASCII code to indicate the angular position of a shaft on rotating machinery
 d: To convert the angular position of a shaft on rotating machinery into hexadecimal code

4-bit gray code can be converted into _____.

a: 1-bit binary
 b: 2-bit binary
 c: 3-bit binary

Type here to search

26°C Haze 01:02 05-01-2022

DE Journal Virtual Labs

de-iitr.labs.ac.in/exp/binary-conversion/pretest.html

HOME PARTNERS CONTACT

4-bit gray code can be converted into _____.

a: 1-bit binary
 b: 2-bit binary
 c: 3-bit binary
 d: 4-bit binary

Gray to binary conversion can be implemented with _____.

a: AND
 b: Ex-OR
 c: NAND
 d: NOR

A binary code that progresses such that only one bit changes between two successive codes is :

a: Nine's-complement code
 b: 8421 code
 c: Excess-3 code
 d: Gray code

Submit Quiz

5 out of 5

Type here to search

26°C Haze 01:02 05-01-2022

PROCEDURE:

Gray to Binary Conversion

Step-1) Connect battery to supply 5V to the circuit.
 Step-2) Press Switches for different inputs.



The switch in ON state is and the switch in OFF state is .

Step-3) The corresponding combination of input and output LEDs lit up for different combination of inputs.

The input gray code LEDs are G₃,G₂,G₁ and G₀ and the output binary code LEDs B₃,B₂,B₁ and B₀ glow accordingly.

The input LED in OFF state is  and in ON state is .

The output LED in OFF state is  and in ON state is .

Step-4) Click "Add" to add the values to the Truth Table.

Binary to Gray Code Conversion

Step-1) Connect battery to supply 5V to the circuit.
Step-2) Press Switches for different inputs.

The switch in ON state is  and the switch in OFF state is .

Step-3) The corresponding combination of input and output LEDs lit up for different combination of inputs.

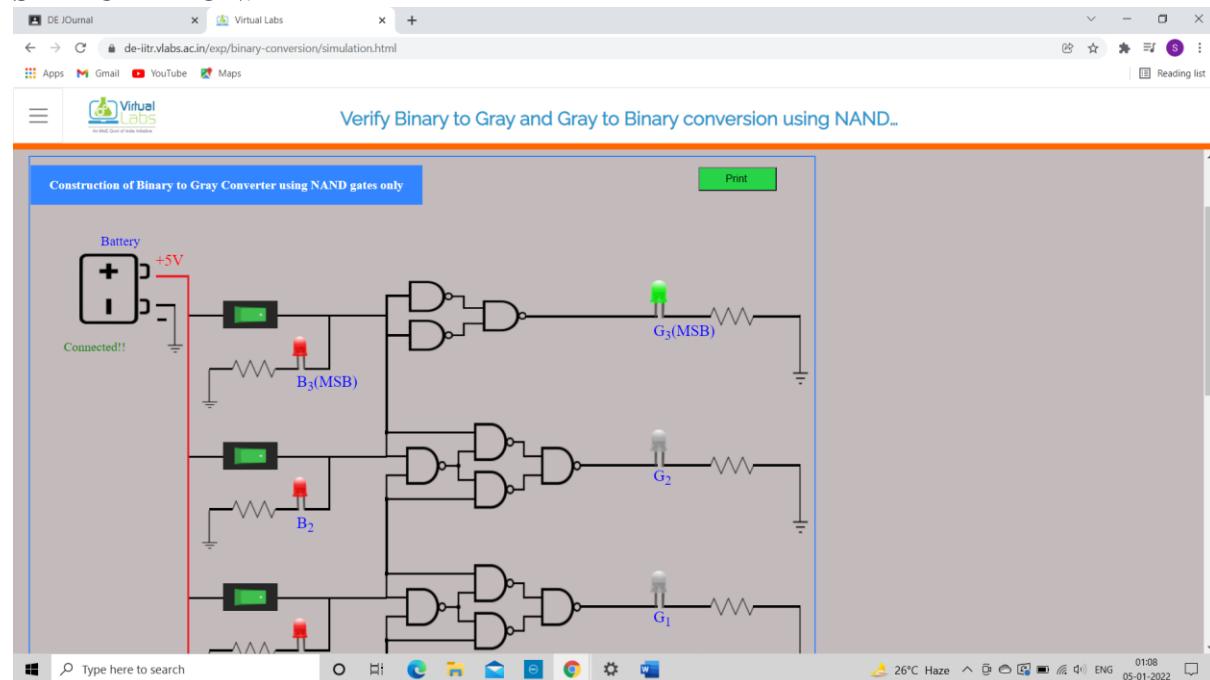
The input binary code LEDs are B_3, B_2, B_1 and B_0 and the output gray code LEDs G_3, G_2, G_1 and G_0 glow accordingly.

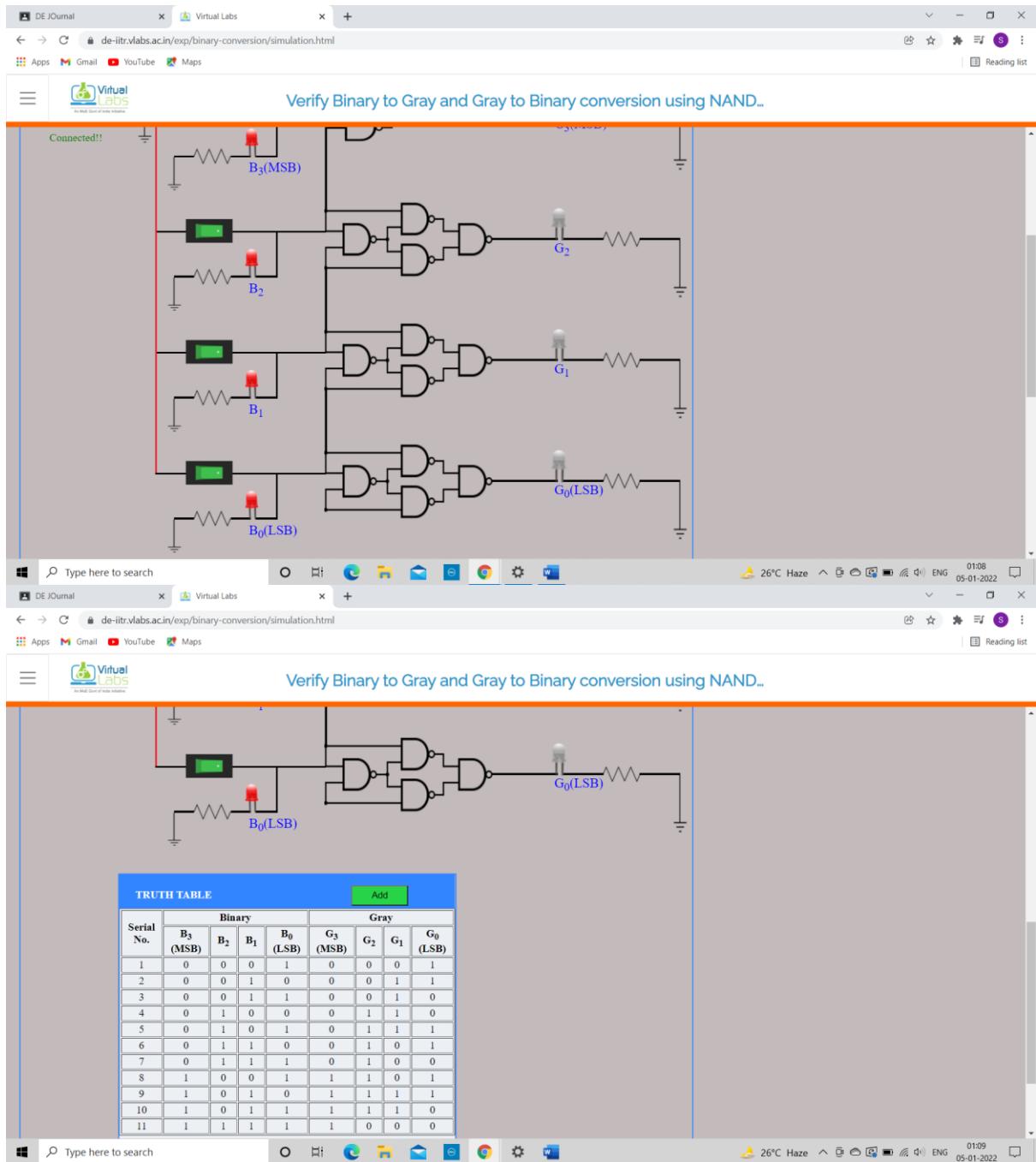
The input LED in OFF state is  and in ON state is .

The output LED in OFF state is  and in ON state is .

Step-5) Click "Print" to get the print out of the Truth Table.

STIMULATION:





POSTTEST:

Verify Binary to Gray and Gray to Binary conversion using NAND gates only.

Convert binary number into gray code: 100101

a: 101101
 b: 001110
 c: 110111
 d: 111001

If two systems have different codes then circuit inserted between them is _____.

a: Combinational circuit
 b: Sequential circuit
 c: Combinational sequence circuit
 d: Conversion circuit

Reflected binary code is also known as _____.

a: BCD code
 b: Gray Code
 c: ASCII code

Reflected binary code is also known as _____.

a: BCD code
 b: Gray Code
 c: ASCII code
 d: Binary code

Code is a symbolic representation of _____.

a: Discrete information
 b: Continuous information
 c: Decimal information into binary
 d: Binary information into decimal

Convert gray code into binary code: 0101.

a: 1001
 b: 0110
 c: 1101
 d: 0111

Submit Quiz

5 out of 5

CONCLUSION:

Thus, through the stimulation we can analyse the truth table of binary to gray and gray to binary converter using combination of NAND gates.

Practical no 5 Implement Adder and Subtractor Arithmetic circuits

A] Construction of half/ full adder using XOR and NAND gates and verification of its operation

AIM: To verify the truth table of half adder and full adder by using XOR and NAND gates respectively and analyse the working of half adder and full adder circuit with the help of LEDs in simulator 1 and verify the truth table only of half adder and full adder in simulator 2.

THEORY: Introduction

Adders are digital circuits that carry out addition of numbers. Adders are a key component of arithmetic logic unit. Adders can be constructed for most of the numerical representations like Binary Coded Decimal (BCD), Excess – 3, Gray code, Binary etc. out of these, binary addition is the most frequently performed task by most common adders. Apart from addition, adders are also used in certain digital applications like table index calculation, address decoding etc.

Binary addition is similar to that of decimal addition. Some basic binary additions are shown below.

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 0 \\ \text{(carry) } 1 \end{array}$$

Figure 1. Schematic representation of half adder

1) Half Adder

Half adder is a combinational circuit that performs simple addition of two binary numbers. If we assume A and B as the two bits whose addition is to be performed, the block diagram and a truth table for half adder with A, B as inputs and Sum, Carry as outputs can be tabulated as follows.

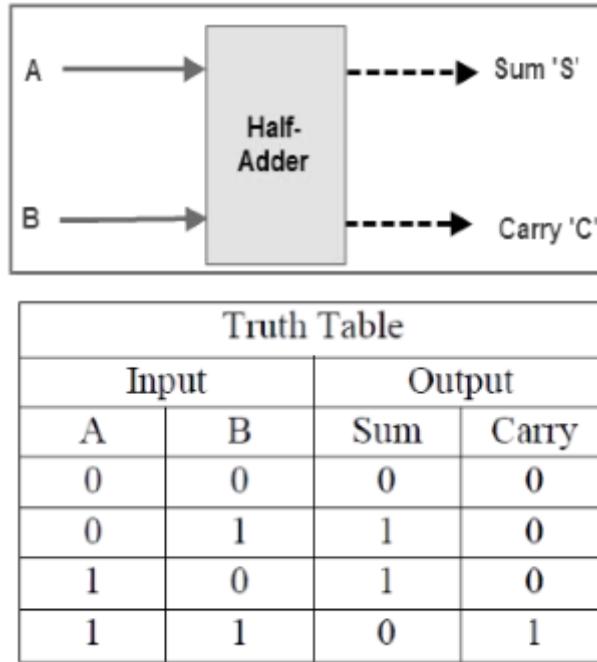


Figure 2. Block diagram and truth table of half adder

The sum output of the binary addition carried out above is similar to that of an Ex-OR operation while the carry output is similar to that of an AND operation. The same can be verified with help of Karnaugh Map.

The truth table and K Map simplification and logic diagram for sum output is shown below.

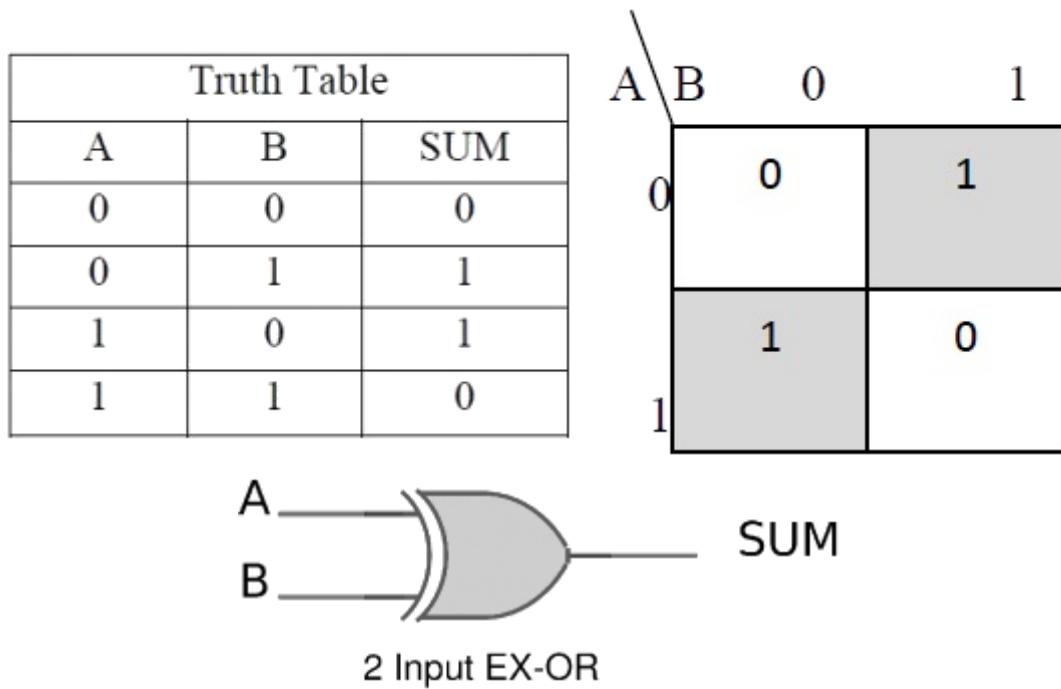


Figure 3. Truth table, K Map simplification and Logic diagram for sum output of half adder

$$\text{Sum} = A B' + A' B$$

The truth table and K Map simplification and logic diagram for carry is shown below.

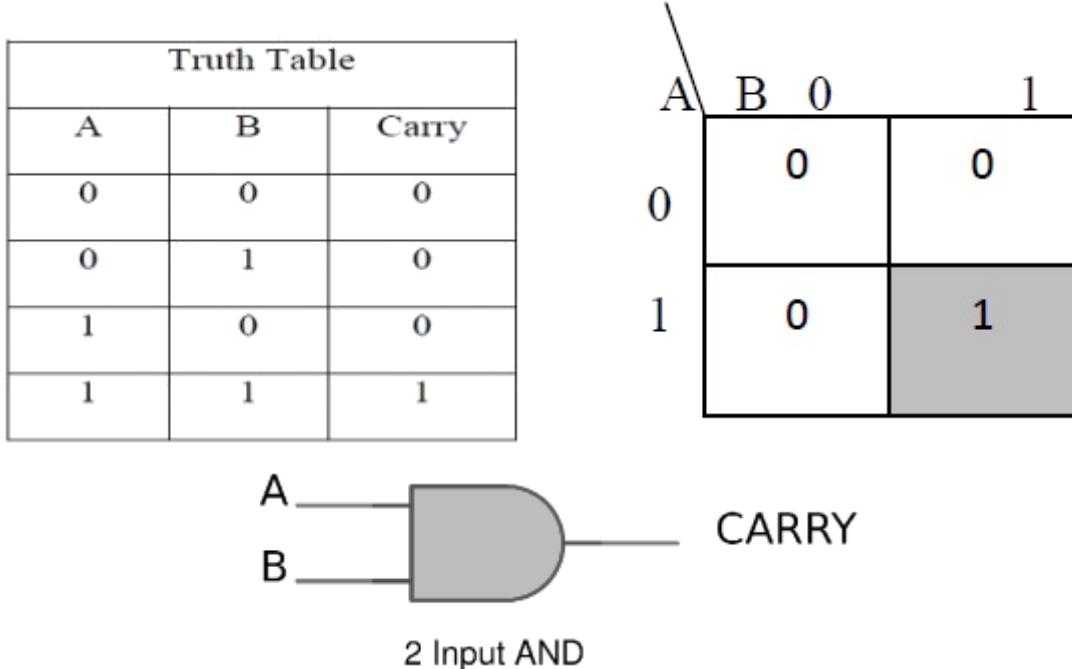


Figure 4. Truth table, K Map simplification and Logic diagram for sum output of half adder

$$\text{Carry} = AB$$

If A and B are binary inputs to the half adder, then the logic function to calculate sum S is Ex – OR of A and B and logic function to calculate carry C is AND of A and B. Combining these two, the logical circuit to implement the combinational circuit of half adder is shown below.

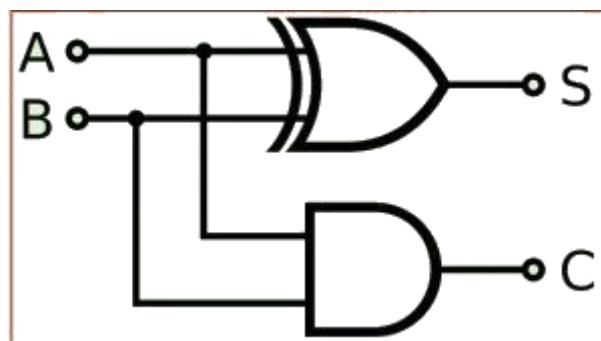


Figure 5. Half Adder Logic Diagram

As we know that NAND and NOR are called universal gates as any logic system can be implemented using these two, the half adder circuit can also be implemented using them. We know that a half adder circuit has one Ex – OR gate and one AND gate.

1.1)Half Adder using NAND gates

Five NAND gates are required in order to design a half adder. The circuit to realize half adder using NAND gates is shown below.

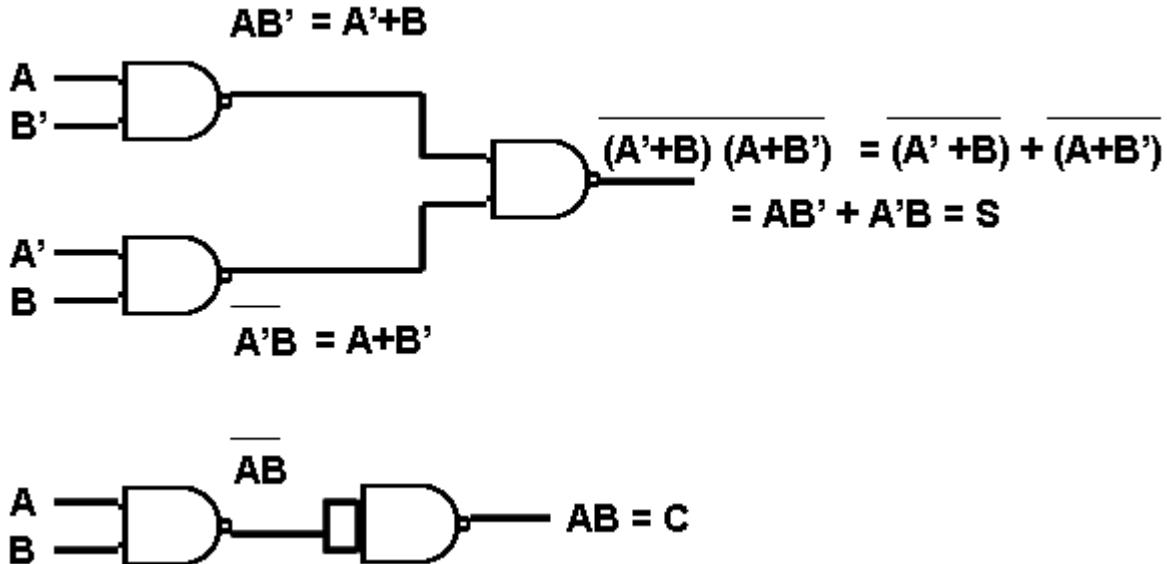


Figure 6. Realization of half adder using NAND gates

1.2)Half Adder using NOR gates

Five NOR gates are required in order to design a half adder. The circuit to realize half adder using NOR gates is shown below.

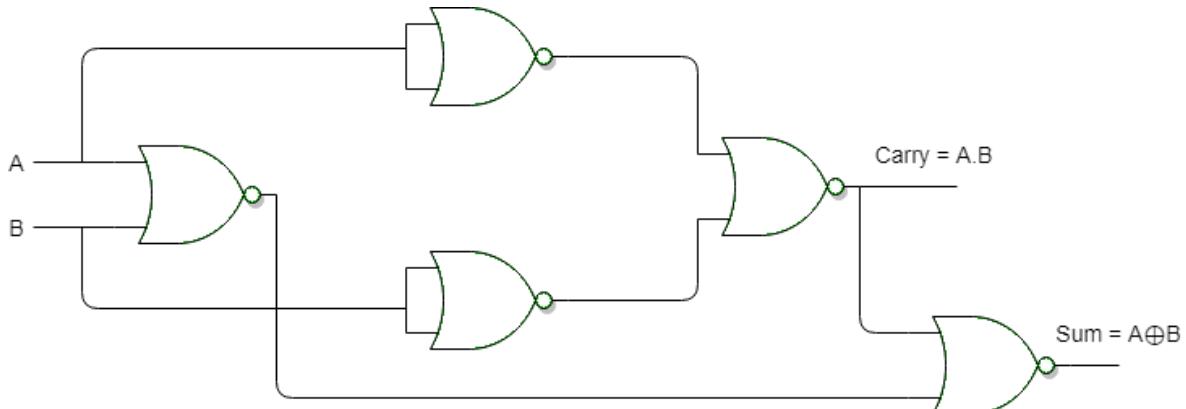


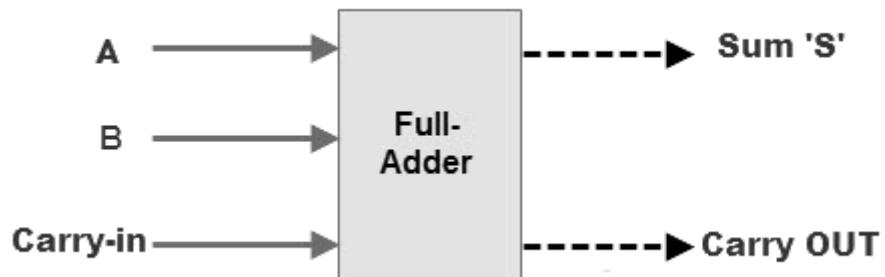
Figure 7. Realization of half adder using NOR Gates

2)Full Adder

Full adder is a digital circuit used to calculate the sum of three binary bits. Full adders are complex and difficult to implement when compared to half adders. Two of the three bits are same as before which are A, the augend bit and B, the addend bit. The additional third bit is carry bit from the previous stage and is called 'Carry' – in generally represented by CIN. It calculates the sum of three bits along with the carry. The output carry is called Carry – out

and is represented by Carry OUT.

The block diagram of a full adder with A, B and CIN as inputs and S, Carry OUT as outputs is shown below.



Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 8. Full Adder Block Diagram and Truth Table

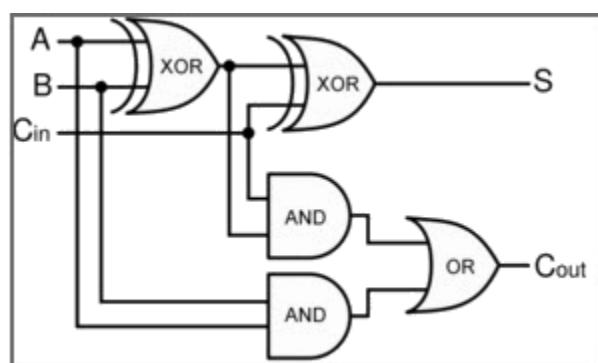


Figure 9. Full Adder Logic Diagram

Based on the truth table, the Boolean functions for Sum (S) and Carry – out (COUT) can be derived using K – Map.

		BC _{IN}	00	01	11	10
		0	0	1	0	1
		1	1	0	1	0
A						

Figure 10. The K-Map simplified equation for sum is $S = A'B'Cin + A'BCin' + ABCin$

		BC _{IN}	00	01	11	10
		0	0	0	1	0
		1	0	1	1	1
A						

Figure 11. The K-Map simplified equation for COUT is $COUT = AB + ACin + BCin$

In order to implement a combinational circuit for full adder, it is clear from the equations derived above, that we need four 3-input AND gates and one 4-input OR gates for Sum and three 2-input AND gates and one 3-input OR gate for Carry – out.

2.1)Full Adder using NAND gates

As mentioned earlier, a NAND gate is one of the universal gates and can be used to implement any logic design. The circuit of full adder using only NAND gates is shown below.

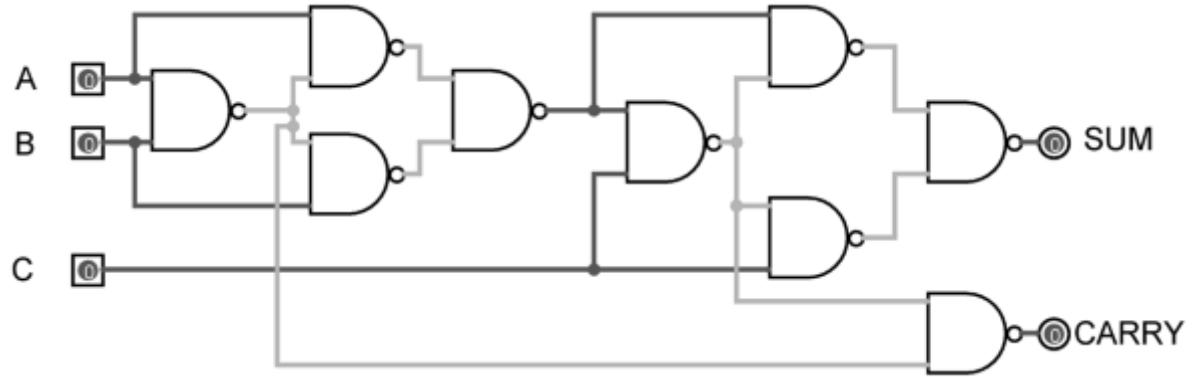


Figure 12. Full Adder using NAND gates

2.2)Full Adder using NOR gates

As mentioned earlier, a NOR gate is one of the universal gates and can be used to implement any logic design. The circuit of full adder using only NOR gates is shown below.

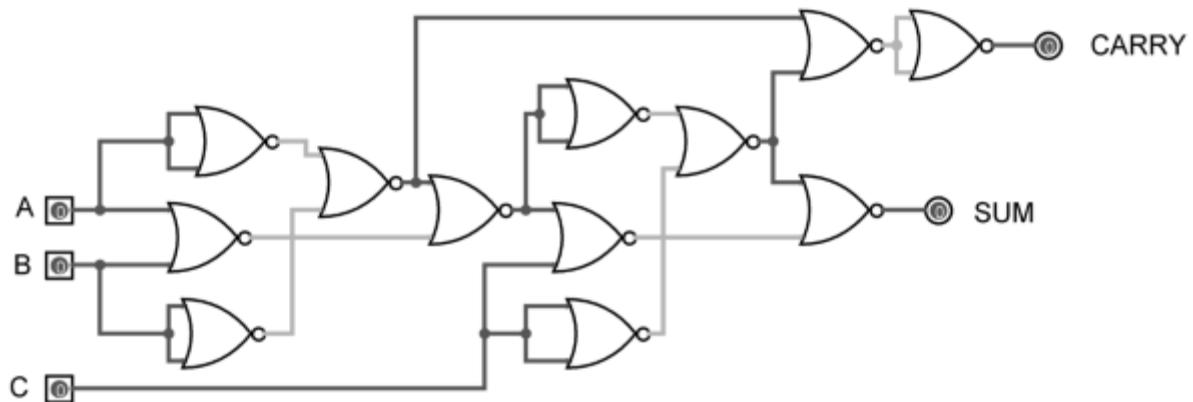


Figure 13. Full Adder using NOR gates

PRETEST:

DE Journal Virtual Labs

de-iitr.labs.ac.in/exp/half-full-adder/pretest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Electronics IITR > Experiments

Aim
Theory
Pretest
Procedure
Simulation
Posttest
References
Feedback

Construction of half/ full adder using XOR and NAND gates and verification of its operation

In parts of the processor, adders are used to calculate _____
 a: Addresses
 b: Table indices
 c: Increment and decrement operators
 d: All of the Mentioned

Total number of inputs in a half adder is _____
 a: 2
 b: 3
 c: 4
 d: 1

In which operation carry is obtained?
 a: Subtraction
 b: Addition
 c: Multiplication

26°C Haze 01:15 05-01-2022

DE Journal Virtual Labs

de-iitr.labs.ac.in/exp/half-full-adder/pretest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

In which operation carry is obtained?
 a: Subtraction
 b: Addition
 c: Multiplication
 d: Both addition and subtraction

If A and B are the inputs of a half adder, the sum is given by _____
 a: A AND B
 b: A OR B
 c: A Ex-OR B
 d: A EX-NOR B

If A and B are the inputs of a half adder, the carry is given by _____
 a: A AND B
 b: A OR B
 c: A Ex-OR B
 d: A Ex-NOR B

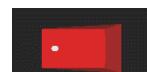
Submit Quiz
5 out of 5

26°C Haze 01:15 05-01-2022

PROCEDURE:

1)HALF ADDER

Simulator 1:



- Step-1) Connect the supply(+5V) to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switches to select the required inputs "A" and "B".
- Step-4) Press "ADD" button to add your inputs and outputs in the given table.
- Step-5) Repeat steps 3 & 4 for next state of inputs and their corresponding outputs.
- Step-6) Press the "PRINT" button after completing your simulation to get your results.

Step-7) Press the "RESET" button whenever you want to refresh your simulator.

Simulator 2:

- Step-1) Enter the Boolean input "A" and "B".
- Step-2) Enter the Boolean output for your corresponding inputs.
- Step-3) Click on "Circuit" button to check the circuit diagram for half adder.
- Step-4) Click on "Check" Button to verify your output.
- Step-5) Click "Print" if you want to get print out of Truth Table.
- Step-6) Click on "Reset" button if you want to reset input and outputs.

2) FULL ADDER

Simulator 1:

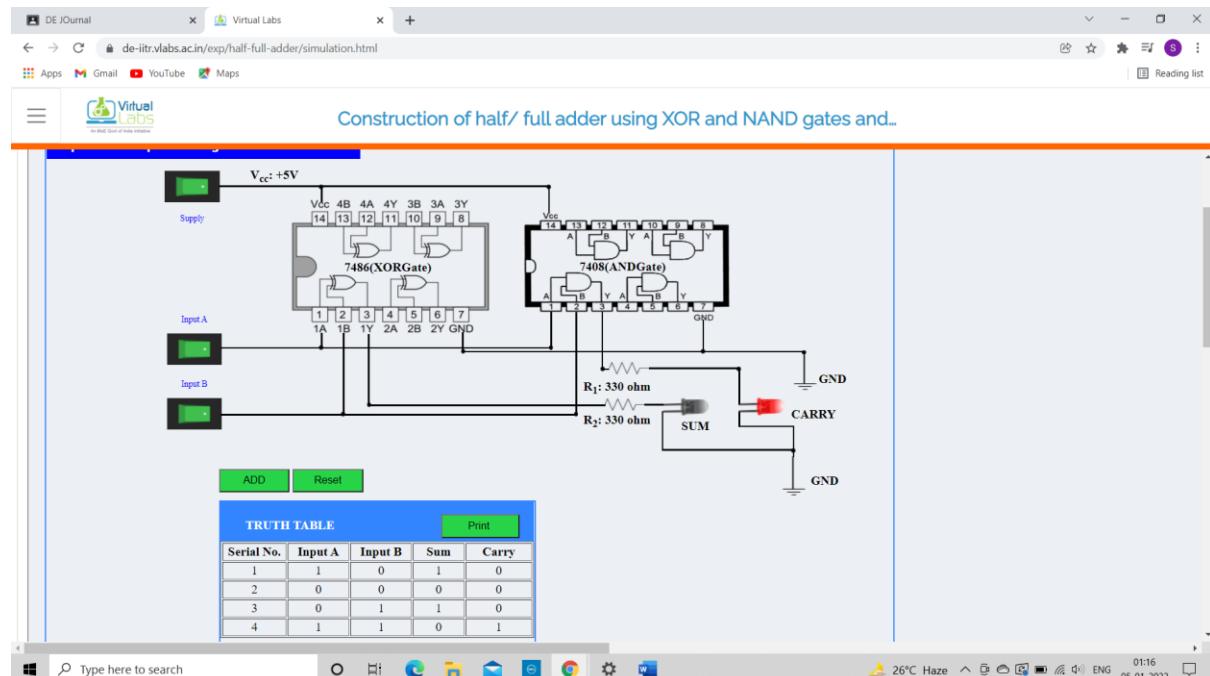


- Step-1) Connect the supply(+5V) to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switches to select the required inputs "A" and "B" and "Cin".
- Step-4) Press "ADD" button to add your inputs and outputs in the given table.
- Step-5) Repeat steps 3 & 4 for next state of inputs and their corresponding outputs.
- Step-6) Press the "PRINT" button after completing your simulation to get your results.
- Step-7) Press the "RESET" button whenever you want to refresh your simulator.

Simulator 2:

- Step-1) Enter the Boolean input "A" and "B" and "Cin".
- Step-2) Enter the Boolean output for your corresponding inputs.
- Step-3) Click on "Circuit" button to check the circuit diagram for full adder.
- Step-4) Click on "Check" Button to verify your output.
- Step-5) Click "Print" if you want to get print out of Truth Table.
- Step-6) Click on "Reset" button if you want to reset input and outputs.

STIMULATION:



DE Journal × Virtual Labs × +

de-iitr.vlabs.ac.in/exp/half-full-adder/simulation.html

Apps Gmail YouTube Maps

☰ Virtual Labs An MoE Govt of India Initiative

Construction of half/ full adder using XOR and NAND gates and...

Verification of truth table for HALF ADDER

HALF ADDER

INPUTS: A, B

OUTPUTS: S (SUM), C (CARRY)

Circuit Diagram:

```

    graph LR
        A((A)) --> X1(( ))
        B((B)) --> X1
        X1 --> S((S))
        A --> X2(( ))
        B --> X2
        X2 --> C((C))
    
```

TRUTH TABLE

Serial No.	A	B	Sum	Carry	Remarks
1	0	0	0	0	Correct
2	0	1	1	0	Correct
3	1	0	1	0	Correct
4	1	1	0	1	Correct

Print

Type here to search

26°C Haze ENG 01:17 05-01-2022

POSTTEST:

DE Journal × Virtual Labs × +

de-iitr.vlabs.ac.in/exp/half-full-adder/posttest.html

Apps Gmail YouTube Maps

☰ Virtual Labs An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Electronics IITR > Experiments

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Construction of half/ full adder using XOR and NAND gates and verification of its operation

Half-adders have a major limitation in that they cannot _____

- a: Accept a carry bit from a present stage
- b: Accept a carry bit from a next stage
- c: Accept a carry bit from a previous stage
- d: Accept a carry bit from the following stages

The difference between half adder and full adder is _____

- a: Half adder has two inputs while full adder has four inputs
- b: Half adder has one output while full adder has two outputs
- c: Half adder has two inputs while full adder has three inputs
- d: All of the Mentioned

If A, B and C are the inputs of a full adder then the sum is given by _____

- a: A AND B AND C
- b: A OR B AND C
- c: A Ex-OR B XOR C

Type here to search

26°C Haze ENG 01:19 05-01-2022

If A, B and C are the inputs of a full adder then the sum is given by _____

a: A AND B AND C
 b: A OR B AND C
 c: A Ex-OR B XOR C
 d: A OR B OR C

If A, B and C are the inputs of a full adder then the carry is given by _____

a: A AND B OR (A OR B) AND C
 b: A OR B OR (A AND B) C
 c: (A AND B) OR (A AND B) C
 d: A Ex-OR B Ex-OR (A Ex-OR B) AND C

How many AND, OR and Ex-OR gates are required for the configuration of full adder?

a: 1, 2, 2
 b: 2, 1, 2
 c: 3, 1, 2
 d: 4, 0, 1

Submit Quiz

5 out of 5

CONCLUSION:

Thus, we the truth table of half adder and full adder can be verified using XOR and NAND gates respectively.

B] To Study & Verify Half and Full Subtractor

AIM: To verify the truth table of half subtractor by using the ICs of XOR, NOT and AND gates and of full subtractor by using the ICs of XOR, AND, NOT and OR gates respectively and analyse the working of half subtractor and full subtractor circuit with the help of LEDs in simulator 1 and verify the truth table only of half subtractor and full subtractor in simulator 2.

THEORY: Introduction

Subtractor circuits take two binary numbers as input and subtract one binary number input from the other binary number input. Similar to adders, it gives out two outputs, difference and borrow (carry-in the case of Adder). There are two types of subtractors.

1. Half Subtractor
2. Full Subtractor

1) Half Subtractor

The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, A (minuend) and B (subtrahend) and two outputs Difference and Borrow. The logic symbol and truth table are shown below.

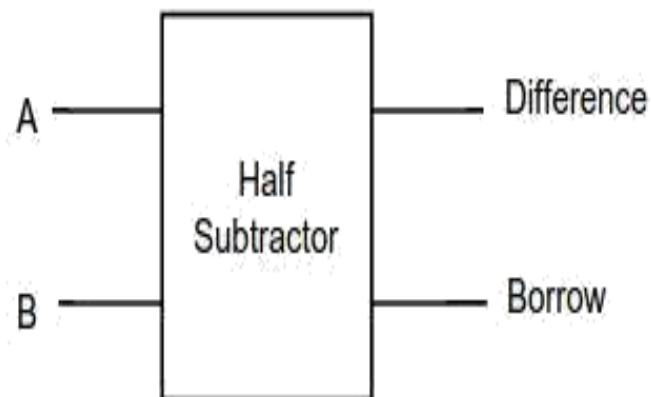


Figure-1:Logic Symbol of Half subtractor

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Figure-2:Truth Table of Half subtractor

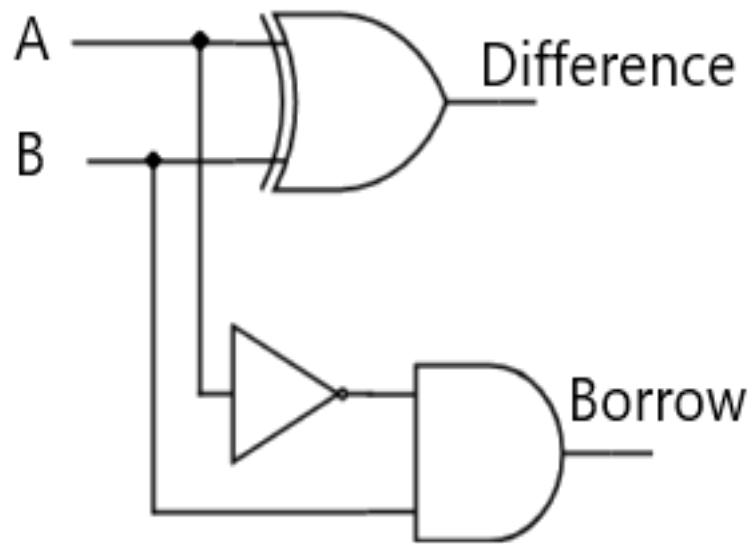


Figure-3:Circuit Diagram of Half subtractor

From the above truth table we can find the boolean expression.

$$\text{Difference} = A \oplus B$$

$$\text{Borrow} = A' B$$

From the equation we can draw the half-subtractor circuit as shown in the figure 3.

2) Full Subtractor

A full subtractor is a combinational circuit that performs subtraction involving three bits, namely A (minuend), B (subtrahend), and Bin (borrow-in) . It accepts three inputs: A (minuend), B (subtrahend) and a Bin (borrow bit) and it produces two outputs: D (difference) and Bout (borrow out). The logic symbol and truth table are shown below.

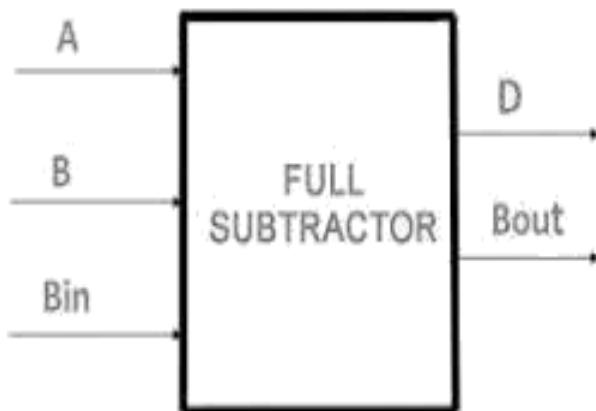


Figure-4:Logic Symbol of Full subtractor

A	B	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Figure-5:Truth Table of Full subtractor

From the above truth table we can find the boolean expression.

$$D = A \oplus B \oplus B_{in}$$

$$B_{out} = A' B_{in} + A' B + B B_{in}$$

From the equation we can draw the Full-subtractor circuit as shown in the figure 6.

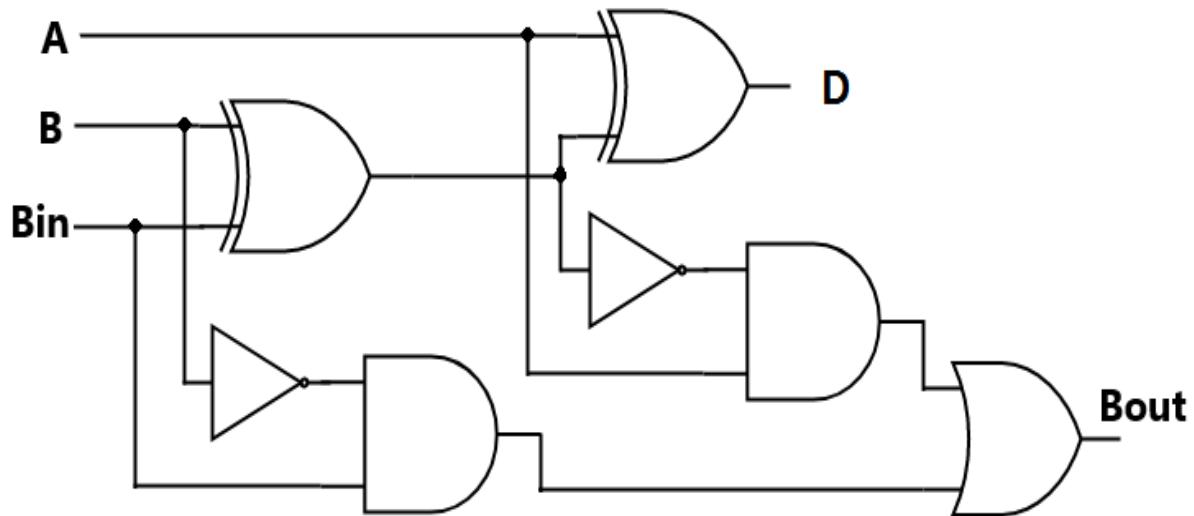


Figure-6:Circuit Diagram of Full subtractor

PRETEST:

Half subtractor is used to perform subtraction of _____

a: 2 bits
 b: 3 bits
 c: 4 bits
 d: 5 bits

For subtracting 1 from 0, we use to take a _____ from neighbouring bits.

a: Carry
 b: Borrow
 c: Input
 d: Output

How many outputs are required for the implementation of a subtractor?

a: 1
 b: 2
 c: 3

Let the input of a subtractor is A and B then what the output will be if $A = B$?

a: 0
 b: 1
 c: A
 d: B

Let A and B is the input of a subtractor then the difference output will be _____

a: A Ex-OR B
 b: A AND B
 c: A OR B
 d: A Ex-NOR B

Submit Quiz

5 out of 5

PROCEDURE:

1)HALF SUBTRACTOR

Simulator 1:



- Step-1) Connect the Supply(+5V) to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switches to select the required inputs "A" and "B".
- Step-4) Press "ADD" button to add your inputs and outputs in the given table.
- Step-5) Repeat steps 3&4 for next state of inputs and their corresponding outputs.
- Step-6) Press the "PRINT" button after completing your simulation to get your results.

Simulator 2:

- Step-1) Enter the Boolean input "A" and "B".
- Step-2) Enter the Boolean output for your corresponding inputs.
- Step-3) Click on "Check" Button to verify your output.
- Step-4) Click "Print" if you want to get print out of Truth Table.
- Step-5) Click "Reset" if you want to reset inputs and outputs.

2) FULL SUBTRACTOR

Simulator 1:

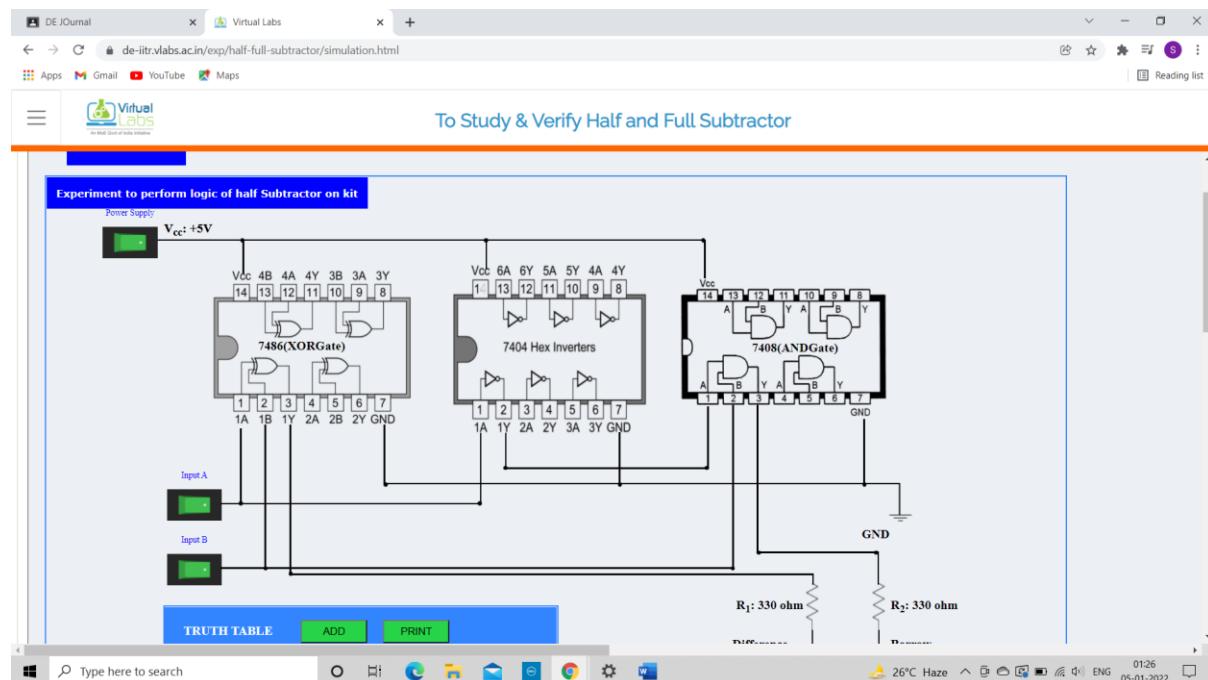


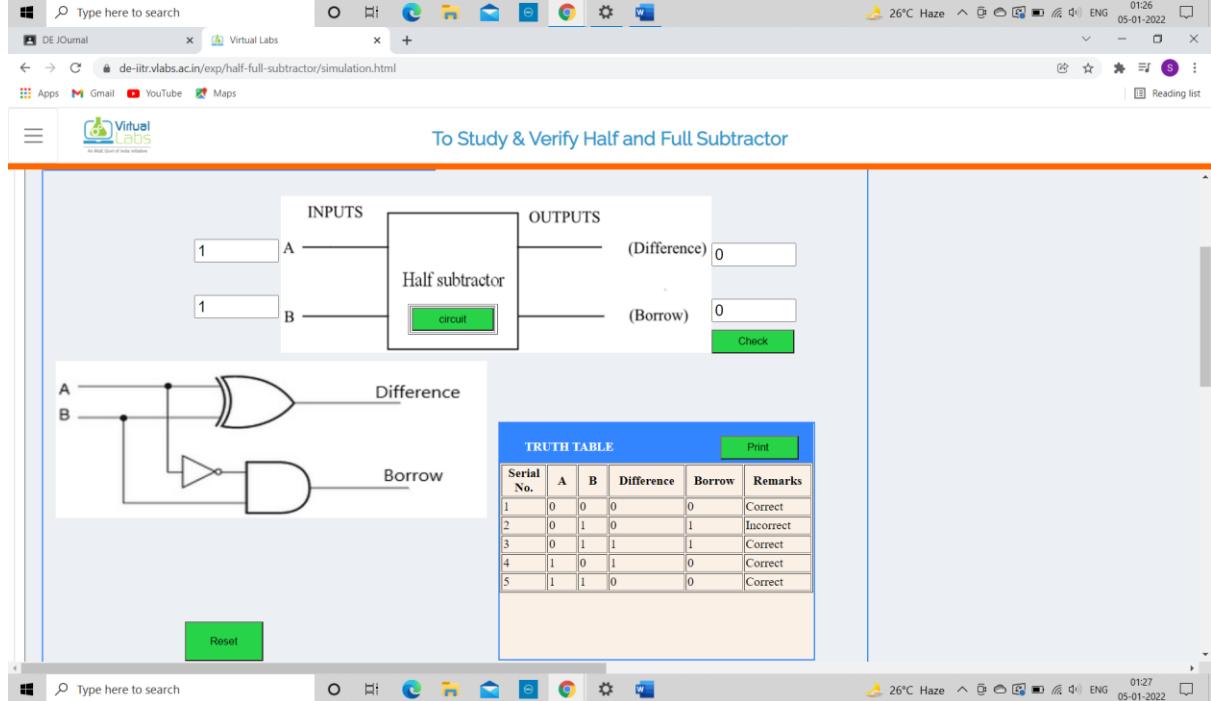
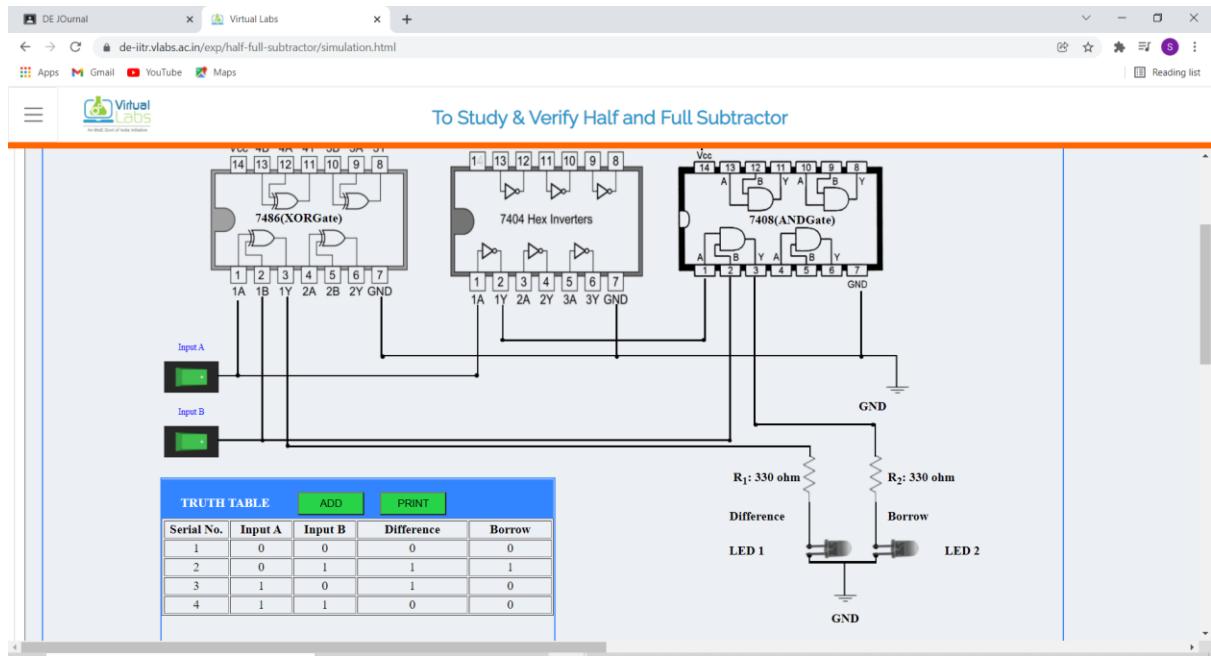
- Step-1) Connect the Supply(+5V)  to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switches to select the required inputs "A" and "B" and "Bin".
- Step-4) Press "ADD" button to add your inputs and outputs in the given table.
- Step-5) Repeat steps 3&4 for next state of inputs and their corresponding outputs.
- Step-6) Press the "PRINT" button after completing your simulation to get your results.

Simulator 2:

- Step-1) Enter the Boolean inputs "A" and "B" and "Bin".
- Step-2) Enter the Boolean output for your corresponding inputs.
- Step-3) Click on "Check" Button to verify your output.
- Step-4) Click "Print" if you want to get print out of Truth Table.
- Step-5) Click "Reset" if you want to reset inputs and outputs.

STIMULATION:





POSTTEST:

DE Journal × **Virtual Labs** × + de-itrl.vlabs.ac.in/exp/half-full-subtractor/posttest.html

App Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Electronics IITR > Experiments

Aim
Theory
Pretest
Procedure
Simulation
Posttest
References
Feedback

To Study & Verify Half and Full Subtractor

Full subtractor is used to perform subtraction of _____
 a: 2 bits
 b: 3 bits
 c: 4 bits
 d: 5 bits

The output of a full subtractor is given by (if A, B and Bin are the inputs).
 a: A AND B Ex-OR Bin
 b: A Ex-OR B Ex-OR Bin
 c: A OR B NOR Bin
 d: A NOR B Ex-OR Bin

The output of a full subtractor is same as _____
 a: Half adder
 b: Full adder
 c: Half subtractor

The output of a full subtractor is same as _____
 a: Half adder
 b: Full adder
 c: Half subtractor
 d: Decoder

What does minuend and subtrahend denotes in a subtractor?
 a: Their corresponding bits of input
 b: Its outputs
 c: Its inputs
 d: Borrow bits

Let A and B is the input of a subtractor then the borrow will be _____
 a: A AND B'
 b: A' AND B
 c: A OR B
 d: A AND B

Submit Quiz

5 out of 5

26°C Haze 01:30 05-01-2022

CONCLUSION:

Thus, the truth table of half subtractor can be verified by using the ICs of XOR, NOT and AND gates and of full subtractor by using the ICs of XOR, AND, NOT and OR gates respectively.

C] Implement of Half adder

AIM: To implement Half adder by using basic and universal gates

THEORY: Adders

Digital computers perform a variety of information processing tasks. Among the basic tasks encountered are the various arithmetic operations. The most basic arithmetic operation is the addition of two binary digits.

A combinational circuit that performs the addition of two bits is called a half adder. Again the combinational circuit that performs addition of three bits (Two significant bits and a previous carry) is called Full adder.

Half Adder

A half adder is a combinational circuit with two binary inputs (augend and addend bits) and two binary outputs (sum and carry bits). It adds two inputs (A and B) and produces the sum (S) and the carry (C) bits. It is an arithmetic circuit used to perform the arithmetic operation of addition of two single bit words.

Truth Table

Inputs		
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

The characteristic equation of a Half Adder is expressed as:

$$\text{Sum} = A\bar{B} + \bar{A}B = A \oplus B$$

$$\text{Cout} = AB$$

When any of the inputs A and B is equal to 1, the Sum is 1. Otherwise, it is 0. Carry Cout is 1 only when both the inputs are 1.

Full Adder

The combinational circuit that performs addition of three bits (Two significant bits and a previous carry) is called Full adder.

Truth Table

Inputs			Outputs	
A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1

Truth Table

Inputs			Outputs	
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

The characteristic equation of a Full Adder is expressed as:

$$\text{Sum} = \bar{A}\bar{B}C_{in} + \bar{A}BC_{in} + A\bar{B}\bar{C}_{in} + ABC_{in} = (A\bar{B} + \bar{A}B)\bar{C}_{in} + (AB + \bar{A}\bar{B})C_{in} = (A \oplus B)\bar{C}_{in} + (\neg(A \oplus B))C_{in} = A \oplus B \oplus C_{in}$$

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} = AB + (A \oplus B)C_{in} = AB + AC_{in} + BC_{in}$$

The Sum is High i.e. 1 only when odd number of the inputs are High. Carry Cout is High i.e. 1 only when more than one input are High.

PRETEST:

PROCEDURE:

Components used:

We used the following components for this experiment-

IC 7408 (AND gate) Datasheet

IC 7486 (XOR gate) Datasheet

How to make connection:

After Starting the experiment first click on the Components button to get component list. Now you can Drag and Drop any component in the circuit designing area. To make connection between components, just click on the Blue bubble of any components and Drag it to another Blue bubble of the same or any other components. To delete connection or to remove any component use Double click on that component or connection.

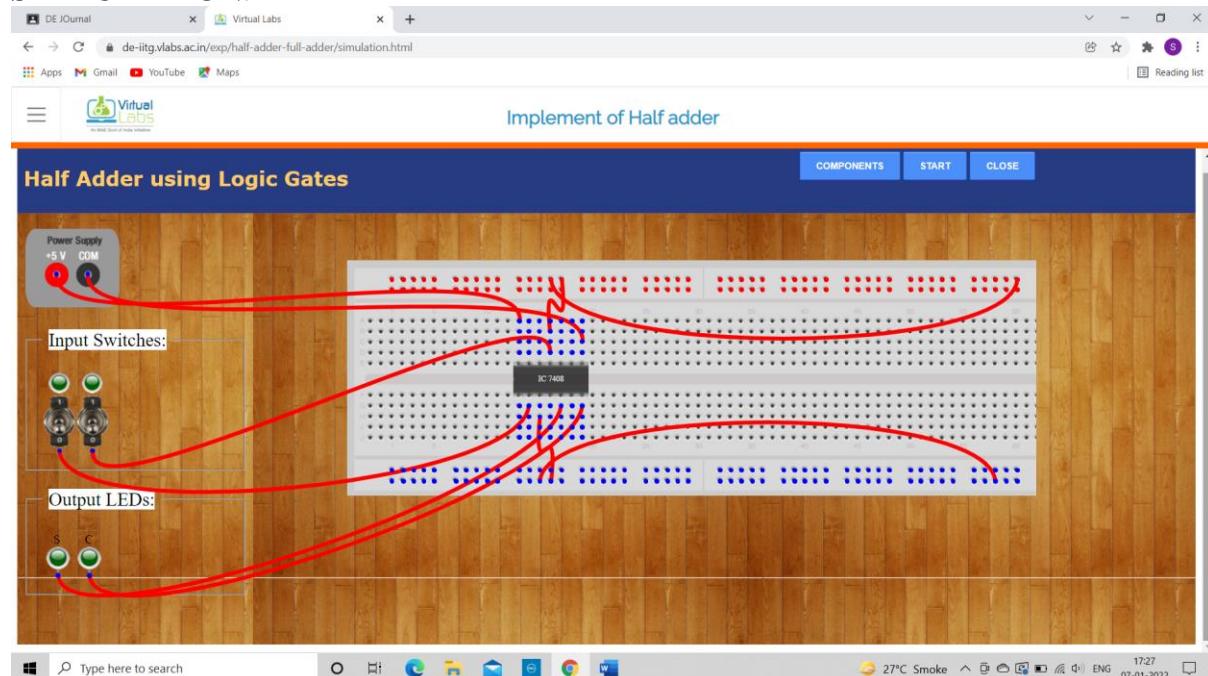
How to run:

Connect the Vcc and Ground pins of the ICs with the power supply. Now connect the input pins of the ICs with the Input Switches. Connect the output pins with output LEDs. Only pins with Blue bubbles can be used.

Green LEDs are used for indicating logic 0 and Red LEDs are used for logic 1.

After connecting all the required components, click on the Start button.

STIMULATION:



POSTTEST:

The screenshot shows a posttest quiz page. The top navigation bar includes 'DE Journal', 'Virtual Labs', and other links. The main content area has a header 'Implement of Half adder'. On the left, a sidebar lists categories: Aim, Theory, Pretest, Procedure, Simulation, Posttest, References, and Feedback. The 'Posttest' section is currently selected. It contains two questions with multiple-choice answers. Question 1 asks about the difference between a half adder and a full adder, with options a, b, c, and d. Option d is selected. Question 2 asks about two-bit addition, with options a, b, c, and d. Option c is selected. Below the questions is a 'Submit Quiz' button. At the bottom, it says '2 out of 2'. The footer includes 'Community Links', 'Contact Us', 'Follow Us', and standard system status icons.

CONCLUSION:

Thus, we can implement half adder by using basic and universal gates.

Practical no. 6 Implement Arithmetic circuits

A] Design of binary subtractor using MSI ICs

AIM: To design and implement 4-bit Binary subtractor using IC-74LS83 and using 2's complement technique

THEORY:

IC 74LS83 is a 4-bit parallel binary adder chip. It adds/subtracts a four-bit number (nibble) with another 4-bit number. The block symbol for the IC is shown in Fig.1. This IC has two sets of 4-bit inputs along with a carry input C0. It performs binary subtraction on the A & B inputs and the carry input C0. It generates a 4-bit Difference and a Borrow out C4.

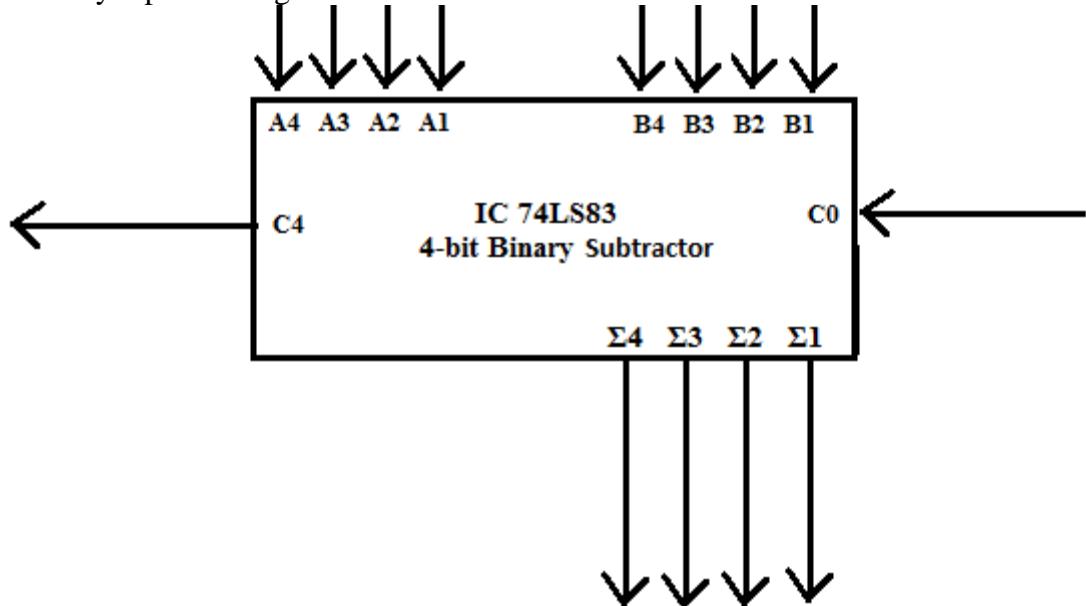


Fig. 1 Block Symbol of four-bit Subtractor IC

A circuit that can subtract 4-bit numbers can be designed using a control input and additional EX-OR IC 74LS86. For this we use the EX-OR gate as a “Controlled Inverter”. The explanation for this concept can be easily understood from Fig.2. The four bit input B4, B3, B2 & B1 can be passed through the controlled inverter IC74LS83 and the A4, A3, A2 & A1 are connected directly to A inputs of IC 74LS83 as shown in Fig 3.

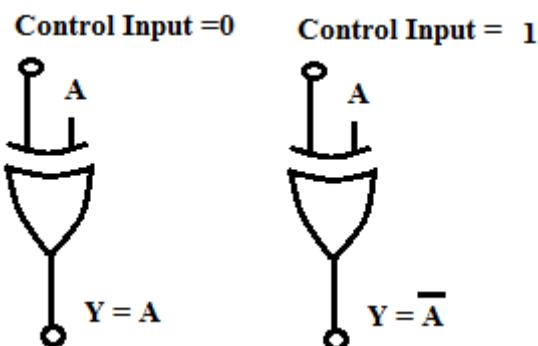


Fig.2. Exclusive-OR gate used as a Controlled Inverter

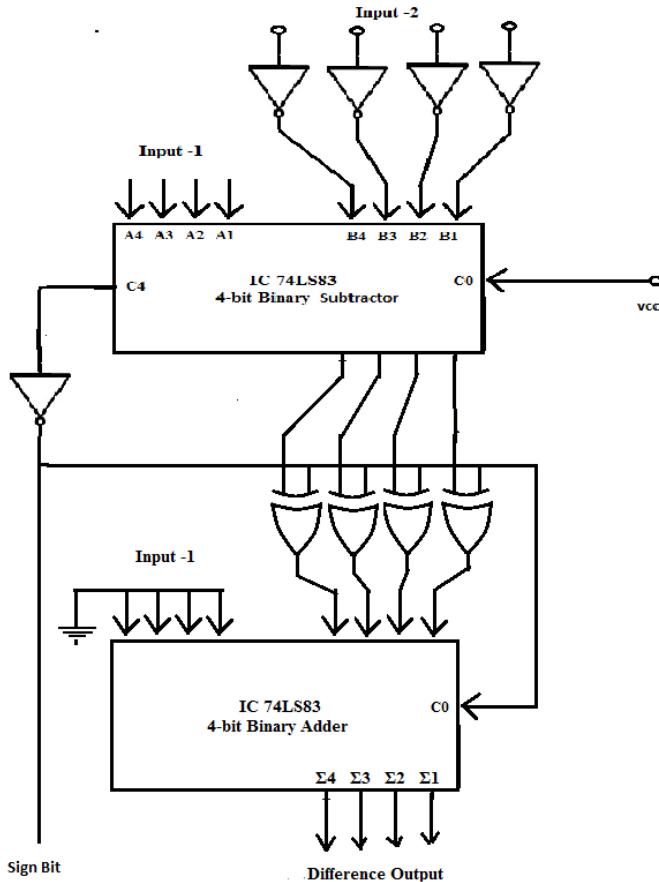


Fig.3. 4-bit Binary Subtractor

Four-bit Subtraction: When Control input is set = 1, the Carry –in input C0 = 1. In this situation, the Ex-OR gates will provide 1.'s complement of the Input-2 to the B-inputs of Adder IC. Moreover as C0 = 1, the addition of 1 to the 1's complement of B gives 2's complement of B. Now the IC74LS83 adds Input-1 i.e. A4,A3, A2,A1 to the 2's complement of B and produces the Carry and Sum output on C4 & the lines Σ4, Σ3, Σ2, & Σ1.

Example1: Let Input-1 = A4 A3 A2 A1 = 1001 & Input-2 = 0111 & control input be set to 1. 1's complement of 0111 = 1000. Since carry input C0 = 1, the input B becomes, B4 B3 B2 B1 = 1000 + 1 = 1001. Now IC 74LS83 performs addition of A & 2,s complement of B and produces the output.

$$\begin{array}{r}
 1001 \\
 +1001 \\
 \hline
 \text{Carry} \rightarrow 10010
 \end{array}$$

↙ Sum

Since a Carry is generated discard the Carry and the Sum is the final output of subtraction operation.

The result is Σ4 Σ3 Σ2 & Σ1 = 0 0 1 0.

Example2: Let Input-1 = A4 A3 A2 A1 = 0111 & Input-2 = 1001 & control input be set to 1. 1's complement of Input-2 i.e. 1001 = 0110 Since carry input C0 = 1, the input B becomes,

$$B_4 \quad B_3 \quad B_2 \quad B_1 = 0110 + 1 = 0111.$$

Now IC 74LS83 performs addition of A & 2's complement of B and produces the output.

$$\begin{array}{r}
 \textcolor{blue}{0} \textcolor{blue}{1} \textcolor{blue}{1} \textcolor{blue}{1} \\
 + \textcolor{blue}{0} \textcolor{blue}{1} \textcolor{blue}{1} \textcolor{blue}{1} \\
 \hline
 \textcolor{blue}{1} \textcolor{blue}{1} \textcolor{blue}{1} \textcolor{blue}{0}
 \end{array}$$

No Carry **Sum**

In this case no Carry is generated during addition. Hence the answer can be obtained by taking the 2's complement of the Sum output and attaching a negative sign. So the 2's complement of 1110 = 0010 and the final result of subtraction is $\Sigma 4 \Sigma 3 \Sigma 2 \& \Sigma 1 = -0010$.

Numerical:

For subtracting two number, the 2's complement of one number is calculated and then added to other number

To calculate 2's complement of a number, the 1's complement of number is calculated i.e. number is represented in binary format, and then inverted. And then add 1 to it.

For eg.: Complement of 2 is:

$$\begin{aligned}
 2 &\Rightarrow 0010 \\
 1\text{'s complement} &\Rightarrow 1101 \\
 2\text{'s complement} &\Rightarrow 1101 + 1 \Rightarrow 1110
 \end{aligned}$$

Hence 2's complement of 2 is 1110.

Examples:

$$1. \quad 1. \quad 4 - 0$$

$$\text{2's Complement of } 0 = 10000$$

$$\begin{array}{r}
 4 \\
 + 2\text{'s complement of } 0 \\
 4
 \end{array}
 \quad
 \begin{array}{r}
 0 \ 1 \ 0 \ 0 \\
 1 \ 0 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 1 \ 0 \ 0
 \end{array}$$

(The MSB bit is complemented to achieve the Sign-bit)

Sign Bit = 0 (Positive Number)

$$2. \quad 2. \quad 4 - 2$$

$$\text{2's Complement of } 2 = 01110$$

$$\begin{array}{r}
 4 \\
 + 2\text{'s complement of } 2 \\
 2
 \end{array}
 \quad
 \begin{array}{r}
 0 \ 1 \ 0 \ 0 \\
 0 \ 1 \ 1 \ 1 \ 0 \\
 1 \ 0 \ 0 \ 1 \ 0
 \end{array}$$

Sign Bit = 0 (Positive Number)

$$3. \quad 3. \quad 2 - 4$$

$$\text{2's Complement of } 4 = 1100$$

$$\begin{array}{r}
 2 \\
 + 2\text{'s complement of } 4 \\
 \hline
 14
 \end{array}
 \quad
 \begin{array}{r}
 0 \ 0 \ 1 \ 0 \\
 0 \ 1 \ 1 \ 0 \ 0 \\
 0 \ 1 \ 1 \ 1 \ 0
 \end{array}$$

Sign Bit = 1 (Negative Number)

As the output is negative number, it is in the 2's complement format. Hence the 2's complement of Answer need to be taken, to achieve the right answer:

Answer => 1110

1's Complement => 0001

2's Complement => 0001 + 1 = 0010

Hence, final answer = 0010 with Sign bit = 1

PRETEST:

Design of binary subtractor using MSI ICs

Pretest

2's complement of binary number 0101 is

a: 1011
b: 1111
c: 1101
d: 1110

The binary subtraction 0000 - 0001 =

a: -0001
b: +0001
c: -1111
d: +1111

What is 2's Complement of Binary number 0000 1000 ?

a: 1110 1000
b: 1111 1000
c: 1010 1100

What is 2's Complement of Binary number 0000 1000 ?

a : 1110 1000
 b : 1111 1000
 c : 1010 1100
 d : 1110 0011

When performing subtraction using 2's-complement method -----

a : the minuend and the subtrahend are both changed to the 2's-complement.
 b : the minuend is changed to 2's-complement and the subtrahend is left in its original form.
 c : the minuend is left in its original form and the subtrahend is changed to its 2's-complement.
 d : the minuend and subtrahend are both left in their original form.

One way to use 4 bit adder circuit to perform 4 bit subtraction is by -----

a : inverting the output.
 b : inverting the carry-in..
 c : inverting the B inputs.
 d : grounding the B inputs.

Submit Quiz

5 out of 5

PROCEDURE:

Steps:

1. Switch ON the circuit, by pressing Main Switch.
2. Set the appropriate Inputs.
3. After giving the inputs, observe the corresponding outputs.
4. Verify the results.

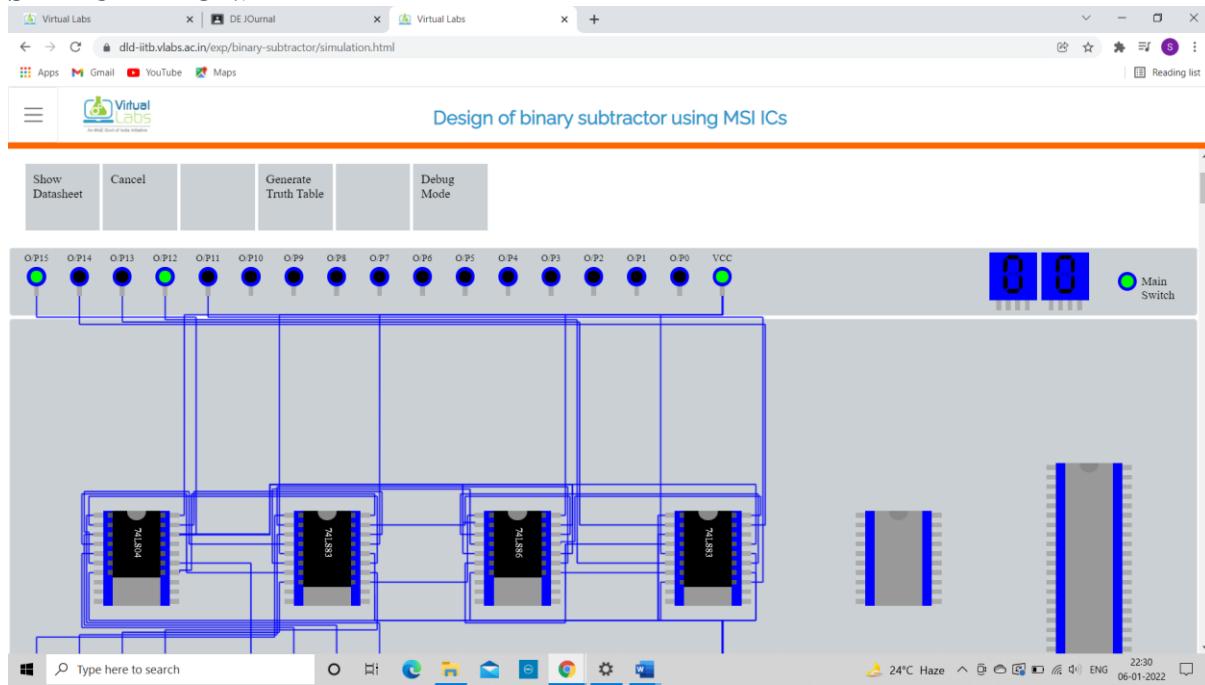
Inputs:

1. Input A (I/P15 to I/P12)
2. Input B (I/P10 to I/P7)

Outputs:

1. Difference (O/P14 to O/P11)
2. Sign Bit(O/P15)

STIMULATION:



Design of binary subtractor using MSI ICs

I/P15	I/P14	I/P13	I/P12	I/P10	I/P9	I/P8	I/P7	O/P15	O/P14	O/P13	O/P12	O/P11
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	1
0	0	0	0	0	0	1	0	1	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	1	1
0	0	0	0	0	1	0	0	1	0	1	0	0
0	0	0	0	0	1	0	1	1	0	1	0	1
0	0	0	0	0	1	1	0	1	0	1	1	0
0	0	0	0	0	1	1	1	1	0	1	1	1
0	0	0	0	1	0	0	0	1	1	0	0	0
0	0	0	0	1	0	0	1	1	1	0	0	1
0	0	0	0	1	0	1	0	1	1	0	1	0
0	0	0	0	1	0	1	1	1	1	0	1	1
0	0	0	0	1	1	0	0	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	1	0	1	0	0	0	1

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

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

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

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

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

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

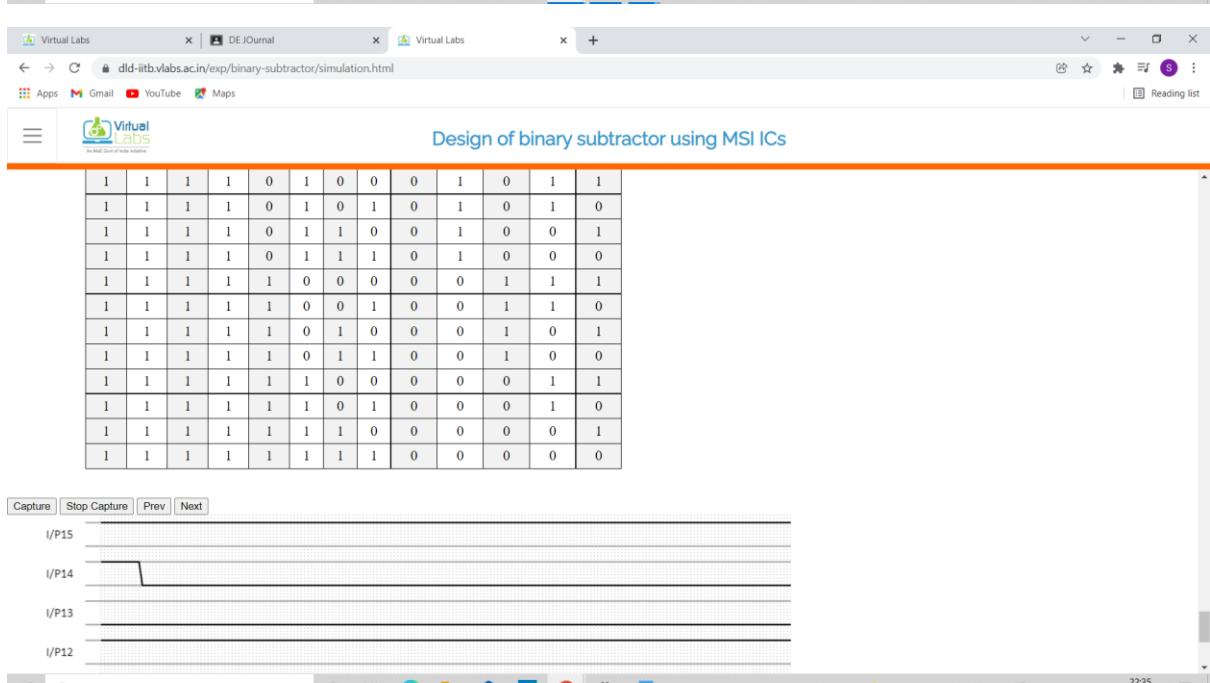
1	0	0	0	1	1	0	0	1	0	1	0	0
1	0	0	0	1	1	0	1	1	0	1	0	1
1	0	0	0	1	1	1	0	1	0	1	1	0
1	0	0	0	1	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	0	0	1
1	0	0	1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1	0	1	0	0	0
1	0	0	1	0	0	1	0	0	0	1	1	1
1	0	0	1	0	0	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0	1	0	0
1	0	0	1	0	1	1	0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

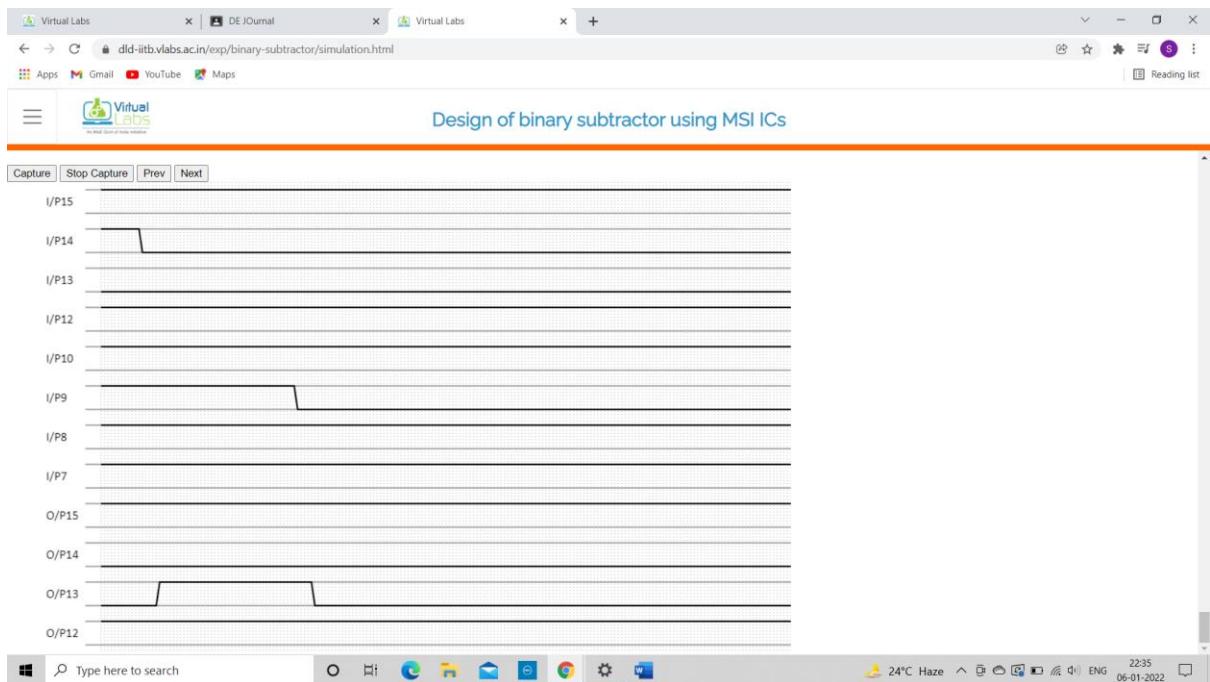
1	0	1	0	0	0	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1	0	1	0	0	1
1	0	1	0	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	1	1	0	0	1	1	1
1	0	1	0	0	0	1	1	0	0	0	1	0
1	0	1	0	0	1	0	0	0	0	1	1	0
1	0	1	0	0	1	0	1	0	0	1	0	1
1	0	1	0	0	1	1	0	0	0	1	0	0
1	0	1	0	0	1	1	1	0	0	0	1	0
1	0	1	0	0	1	1	1	1	0	0	1	0
1	0	1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	1	0	0	0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

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

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

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





POSTTEST:

[DE Journal](#) [Virtual Labs](#)

← → ⌛ [dld-iitb.vlabs.ac.in/exp/binary-subtractor/posttest.html](#)

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Design of binary subtractor using MSI ICs

Aim

Theory

Pretest After performing $1010 - 1100$ using 2's-complement method, What will be the value of sign-bit?
 a : 0
 b : 1

Procedure

Simulation After performing $10001010 - 10001010$ using 2's-complement method, What will be the value of sign-bit?
 a : 0
 b : 1

Posttest Determine the two's-complement of each binary number. 00110 , 00011 , 11101
 a : 11010 , 11100 , 00001
 b : 00111 , 00010 , 00010
 c : 00110 , 00011 , 11101
 d : 11010 , 11101 , 00011

Solve this binary problem: 01110010 - 01001000 = _____
 a : 00011010
 b : 00011010

24°C Haze 22:35 06-01-2022

DE Journal x Virtual Labs x +

ddl-iitb.vlabs.ac.in/exp/binary-subtractor/posttest.html

App Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

b : 00111 , 00010 , 00010.
 c : 00110 , 00011 , 11101.
 d : 11010 , 11101 , 00011.

Solve this binary problem: 0110010 - 01001000 = -----
 a : 00011010
 b : 00101010
 c : 01110010
 d : 00111100

Solve this binary problem: 10100001 - 00010110 = -----
 a : 10001011
 b : 00010011
 c : 11010100
 d : 01011100

Submit Quiz

5 out of 5

Community Links Contact Us Follow Us

Sakshat Portal

Type here to search

Phone: General Information: 011-26520200

30°C Haze 11:57 05-01-2022

CONCLUSION:

Thus, a 4-bit Binary subtractor is designed and implemented using IC-74LS83 and using 2's complement technique.

B] Implementation of binary adder using MSI ICs

AIM: To design and implement 4-bit Binary adder using IC-74LS83

THEORY: IC 74LS83 is a 4-bit parallel binary adder chip. It adds a four-bit number (nibble) with another 4-bit number. The block symbol for the IC is shown in Fig.1. This IC has two sets of 4-bit inputs along with a carry input C0. It performs binary addition on the A & B inputs and the carry input C0. It generates a 4-bit sum and a carry out C4.

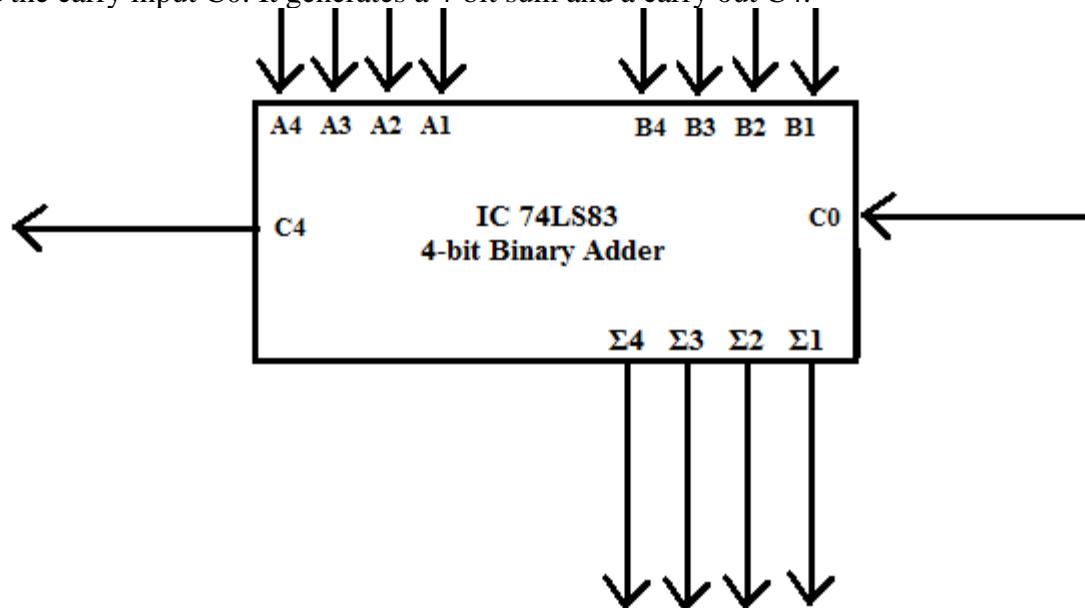


Fig. 1 Block Symbol of four-bit adder IC

A circuit that can add or subtract 4-bit numbers can be designed using a control input and additional EX-OR IC 74LS86. For this we use the EX-OR gate as a “Controlled Inverter”. The explanation for this concept can be easily understood from Fig.2. The four bit input B4, B3, B2 & B1 can be passed through the controlled inverter IC74LS83 and the A4, A3, A2 & A1 are connected directly to A inputs of IC 74LS83 as shown in Fig 3.

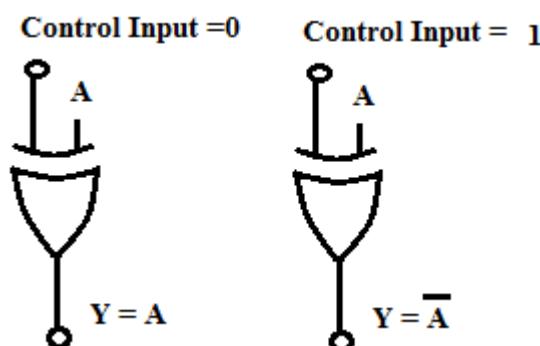


Fig.2. Exclusive-OR gate used as a Controlled Inverter

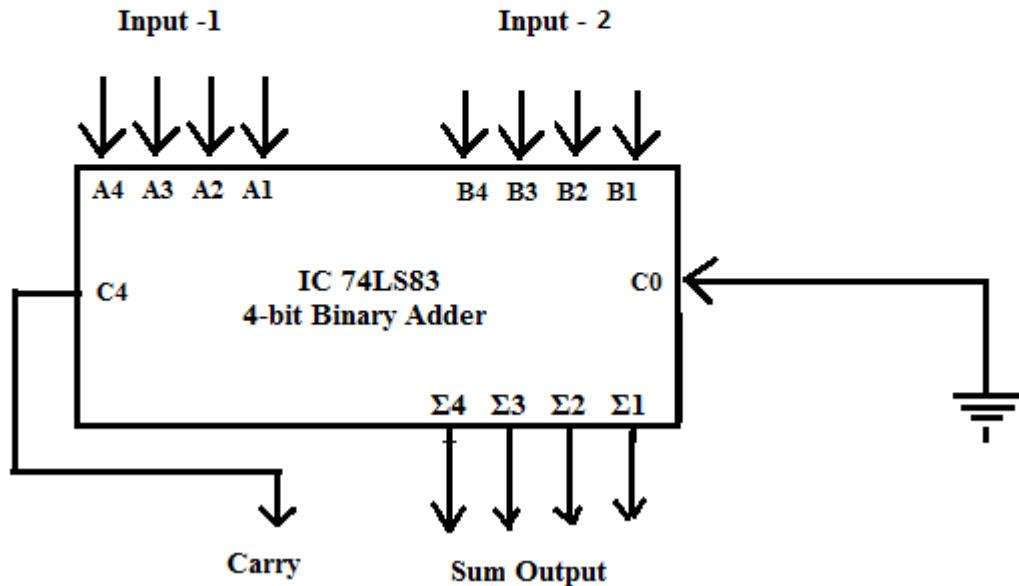


Fig.3. 4-bit Binary Adder/Subtractor

Working:

Case 1: Four-bit Addition: When Control input is set = 0, the Carry –in input C0 = 0. In this situation, the EX-OR gate will simply pass on the Input -2 to B-inputs of Adder IC. It behaves to be transparent. IC 74LS83 will perform simple four-bit Binary addition. It produces the Carry and Sum output on C4 & the lines Σ_4 , Σ_3 , Σ_2 , & Σ_1 .

Example: Let Input-1 = A4 A3 A2 A1 = 1001 & Input-2 = B4 B3 B2 B1 = 0111 & control input be set to 0.

$$\begin{array}{r}
 1001 \\
 + 0111 \\
 \hline
 \text{Carry} \rightarrow 1000 \\
 \text{Sum}
 \end{array}$$

Case 2: Four-bit Subtraction: When Control input is set = 1, the Carry –in input C0 = 1. In this situation, the Ex-OR gates will provide 1,’s complement of the Input-2 to the B-inputs of Adder IC. Moreover as C0 = 1, the addition of 1 to the 1’s complement of B gives 2’s complement of B.

Now the IC74LS83 adds Input-1 i.e. A4,A3, A2,A1 to the 2’s complemet of B and produces the Carry and Sum output on C4 & the lines Σ_4 , Σ_3 , Σ_2 , & Σ_1 .

Example1: Let Input-1 = A4 A3 A2 A1 = 1001 & Input-2 = 0111 & control input be set to 1. 1’s complement of 0111 = 1000. Since carry input C0 = 1, the input B becomes, B4 B3 B2 B1 = 1000 + 1 = 1001.

Now IC 74LS83 performs addition of A & 2,s complement of B and produces the output.

$$\begin{array}{r}
 1001 \\
 +1001 \\
 \hline
 10010
 \end{array}$$

Sum

Since a Carry is generated discard the Carry and the Sum is the final output of subtraction operation.

The result is $\Sigma_4 \Sigma_3 \Sigma_2 \& \Sigma_1 = 0010$.

Example2: Let Input-1 = A4 A3 A2 A1 = 0111 & Input-2 = 1001 & control input be set to 1. 1's complement of Input-2 i.e. 1001 = 0110 Since carry input C0 = 1, the input B becomes, B4 B3 B2 B1 = 0110 + 1 = 0111.

Now IC 74LS83 performs addition of A & 2's complement of B and produces the output.

$$\begin{array}{r}
 0111 \\
 +0111 \\
 \hline
 1110
 \end{array}$$

Sum

In this case no Carry is generated during addition. Hence the answer can be obtained by taking the 2's complement of the Sum output and attaching a negative sign.

So the 2's complement of 1110 = 0010 and the final result of subtraction is $\Sigma_4 \Sigma_3 \Sigma_2 \& \Sigma_1 = -0010$.

Numerical:

1+2

$$\begin{array}{r}
 1 \\
 + 2 \\
 \hline
 3
 \end{array}
 \quad
 \begin{array}{r}
 0 0 0 1 \\
 0 0 1 0 \\
 \hline
 0 0 1 1
 \end{array}$$

Carry (Out) = 0

3+2

$$\begin{array}{r}
 & 3 & & 0 & 0 & 1 & 1 \\
 + & 2 & & 0 & 0 & 1 & 0 \\
 \hline
 & 5 & & \boxed{0} & 0 & 1 & 0 & 1
 \end{array}$$

Carry (Out) = 0

3+7

$$\begin{array}{r}
 & 3 & & 0 & 0 & 1 & 1 \\
 + & 7 & & 1 & 0 & 1 & 0 \\
 \hline
 & 10 & & \boxed{0} & 1 & 1 & 0 & 1
 \end{array}$$

Carry (Out) = 0

11+10

$$\begin{array}{r}
 & 11 & & 1 & 0 & 1 & 1 \\
 + & 10 & & 1 & 0 & 1 & 0 \\
 \hline
 & 21 & & \boxed{1} & 0 & 1 & 0 & 1
 \end{array}$$

Carry (Out) = 1

PRETEST:

Implementation of binary adder using MSI ICs

Q1. 74LS83 is a binary adder IC designed to add two _____ numbers.
 a : 4-bit.
 b : 8-bit.
 c : 16-bit.
 d : 32-bit.

Q2. The output of a Binary Adder is
 a : Sum
 b : Borrow
 c : Difference
 d : Sum and Carry

Q3. A carry look ahead adder is frequently used for addition because
 a : It costs less.
 b : It is faster.
 c : It is more accurate.

Q3. A carry look ahead adder is frequently used for addition because
 a : It costs less.
 b : It is faster
 c : It is more accurate
 d : It uses fewer gates

Q4. IC 74LS83 is a _____.
 a : 4-bit parallel Binary Adder
 b : 4-bit Comparator
 c : 4-bit parallel Binary Subtractor
 d : Decoder

Q5. The Output of $0001 + 1110 = \text{_____}$.
 a : 1111
 b : 0101
 c : 1100
 d : 0010

Submit Quiz

5 out of 5

PROCEDURE:

Steps:

1. Switch ON the circuit, by pressing Main Switch.
2. Set the appropriate Inputs.
3. After giving the inputs, observe the corresponding outputs.
4. Verify the results.

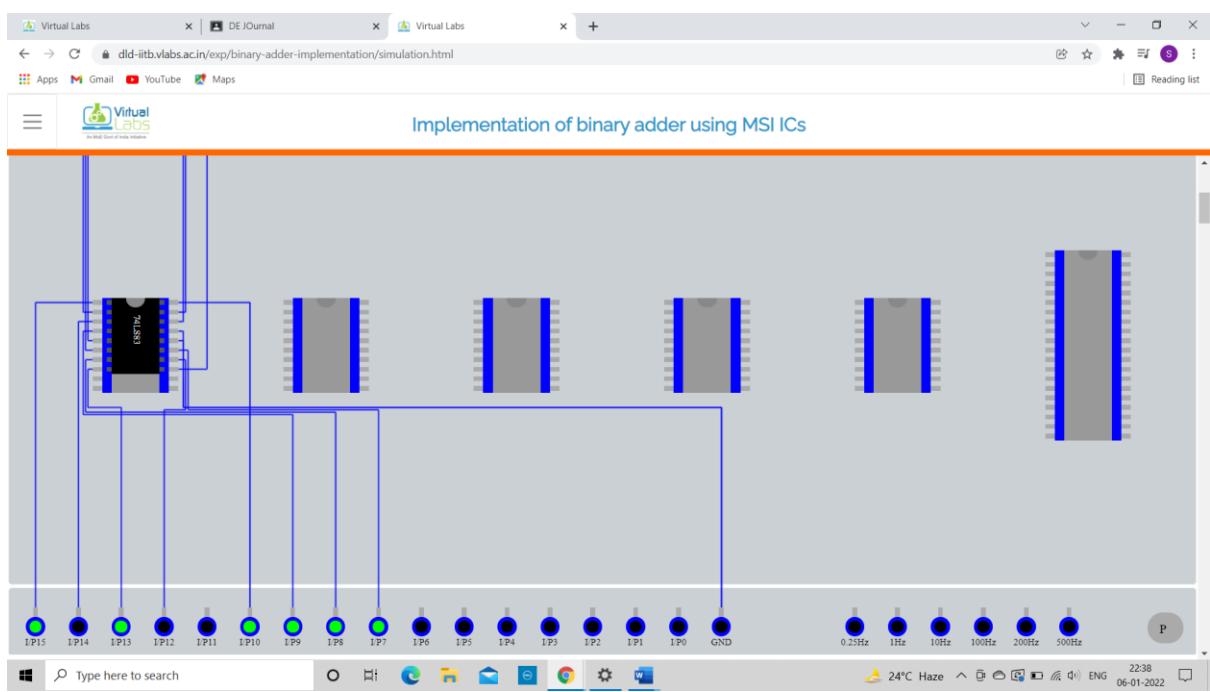
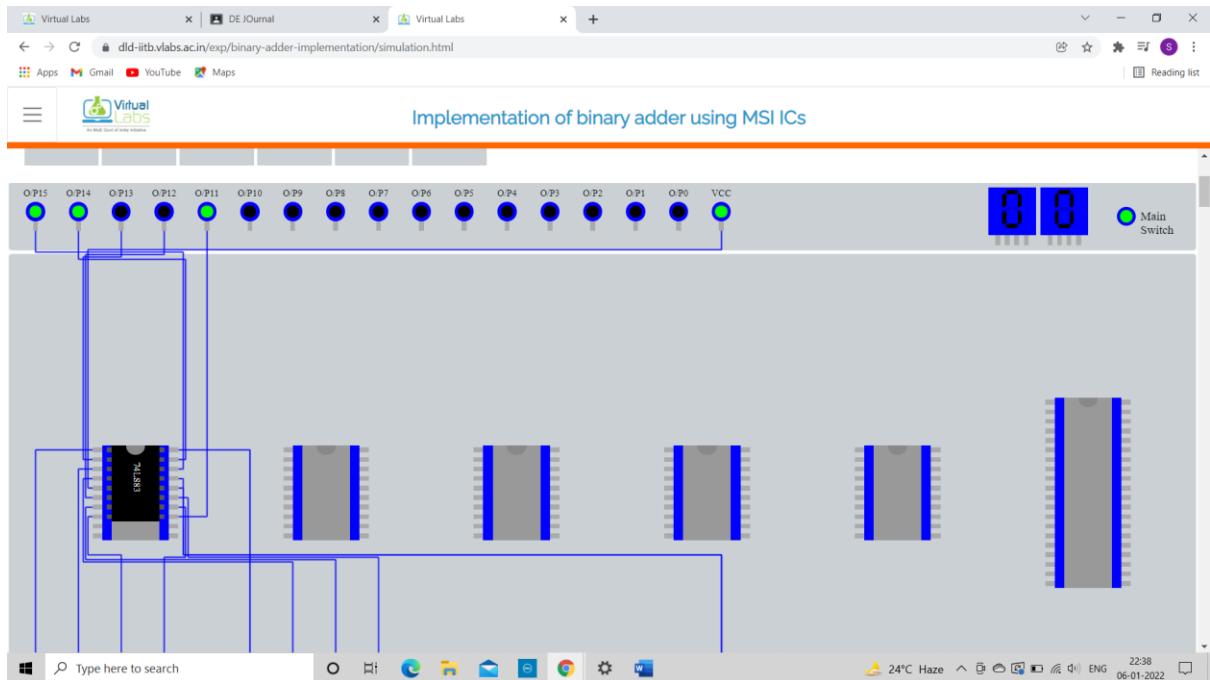
Inputs:

1. Input A (I/P15 to I/P12)
2. Input B (I/P10 to I/P7)

Outputs:

1. Sum (O/P14 to O/P11)
2. Carry Output (O/P15)

STIMULATION:



Virtual Labs DE Journal Virtual Labs

dd-iitb.vlabs.ac.in/exp/binary-adder-implementation/simulation.html

☰  Implementation of binary adder using MSI ICs

Truth Table :

I/P15	I/P14	I/P13	I/P12	I/P10	I/P9	I/P8	I/P7	O/P15	O/P14	O/P13	O/P12	O/P11
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1	0	0	1	0	1
0	0	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	1	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	1	0	1	0	0	1
0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	0	1	0	1	1	0	1	0	1	1
0	0	0	0	1	1	0	0	0	1	1	0	0
0	0	0	0	1	1	1	0	0	1	1	0	0
0	0	0	0	1	1	1	0	0	1	1	0	1
0	0	0	0	1	1	1	1	0	1	1	0	1
0	0	0	0	1	1	1	1	0	1	1	1	0
0	0	0	0	1	1	1	1	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	0	1

Virtual Labs DE Journal Virtual Labs

dd-iitb.vlabs.ac.in/exp/binary-adder-implementation/simulation.html

☰  Implementation of binary adder using MSI ICs

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

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

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

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

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

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

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

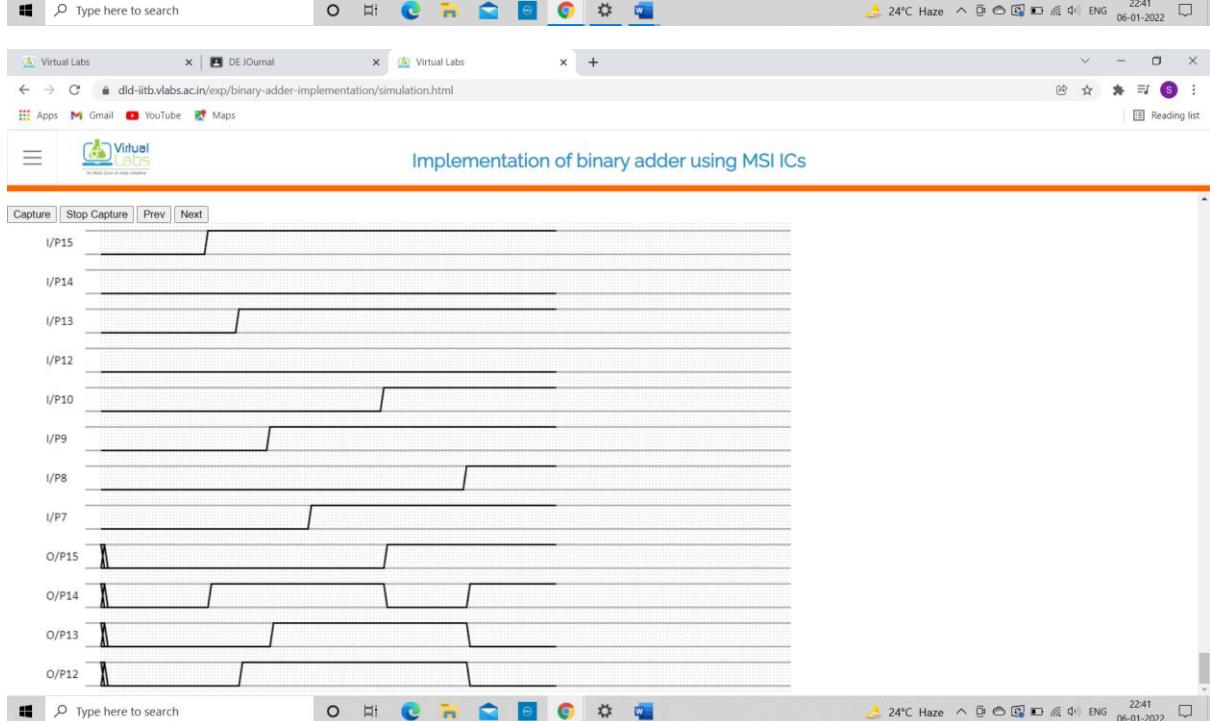
1	0	0	1	1	1	0	1	1	0	1	1	1
1	0	0	1	1	1	1	0	1	0	1	1	1
1	0	0	1	1	1	1	1	1	1	0	0	0
1	0	1	0	0	0	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1	0	1	0	1	1
1	0	1	0	0	0	1	0	0	1	1	0	0
1	0	1	0	0	0	1	1	0	1	1	0	1
1	0	1	0	0	1	0	0	0	1	1	1	0
1	0	1	0	0	1	0	1	0	1	1	1	1
1	0	1	0	0	1	1	0	1	0	0	0	0
1	0	1	0	0	1	1	1	1	0	0	0	1
1	0	1	0	1	0	0	0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

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

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

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

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



POSTTEST:

Implementation of binary adder using MSI ICs

Q1. Two 4-bit binary numbers A = 1011 and B = 1111 are applied to a 4-bit parallel adder. The carry input is 1. What are the values for the sum and carry output?

a : S₄ S₃ S₂ S₁ = 0111, Cout = 0.
 b : S₄ S₃ S₂ S₁ = 1111, Cout = 1.
 c : S₄ S₃ S₂ S₁ = 1011, Cout = 1.
 d : S₄ S₃ S₂ S₁ = 1100, Cout = 1.

Q2. Two 4-bit binary numbers A = 1110 and B = 1010 are applied to a 4-bit parallel adder. The carry input is 0. What are the values for the sum and carry output?

a : S₄ S₃ S₂ S₁ = 1000, Cout = 0.
 b : S₄ S₃ S₂ S₁ = 1000, Cout = 1.
 c : S₄ S₃ S₂ S₁ = 1011, Cout = 0.
 d : S₄ S₃ S₂ S₁ = 1100, Cout = 1.

Q3. How many IC's are required for 8 bit Binary Adder?

a : 1
 b : 2
 c : 3
 d : 4

Q3. How many IC's are required for 8 bit Binary Adder?

a : 1
 b : 2
 c : 3
 d : 4

Q4. Solve 1010 + 0011 = -----

a : Sum -0010 Carry=1
 b : Sum -1101 Carry=0
 c : Sum -1101 Carry=1
 d : Sum -0001 Carry=0

Q5. Solve 1111 + 0110 = -----

a : Sum -0101 Carry=1
 b : Sum -0101 Carry=0
 c : Sum -1110 Carry=0
 d : Sum -1011 Carry=1

Submit Quiz

5 out of 5

CONCLUSION:

Thus, a 4-bit binary adder can be designed and implemented using IC-74LS83 and its truth table was generated.

C] Verify the truth table of one bit and two bit comparator using logic gates.

AIM: To analyse the truth table of 1-bit comparator by using NOT, AND and NOR logic gate ICs and 2-bit comparator by using 1-input NOT, 3-input AND, 2-input AND, 3-input OR and 2-input Ex-NOR logic gate ICs and to understand the working of 1-bit comparator and 2-bit comparator with the help of LEDs display.

THEORY: Introduction

A magnitude digital comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for $A > B$ condition, one for $A = B$ condition and one for $A < B$ condition.

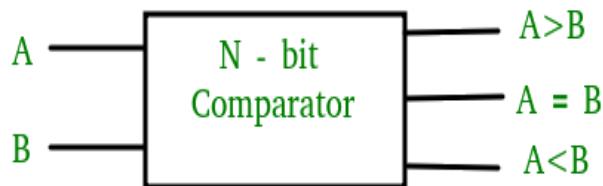


Figure-1: Block Diagram of Comparator

1) 1-Bit Magnitude Comparator :

A comparator used to compare two bits is called a single bit comparator. It consists of two inputs each for two single bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers. The truth table for a 1-bit comparator is given below :

A	B	$A < B$	$A = B$	$A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Figure-2: Truth Table of 1-Bit Comparator

From the above truth table logical expressions for each output can be expressed as follows:

$$\begin{array}{lll} A & > & B \\ A & < & B \\ A & = & B \end{array} : \quad \begin{array}{l} AB' \\ A'B \\ A'B' + AB \end{array}$$

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below :

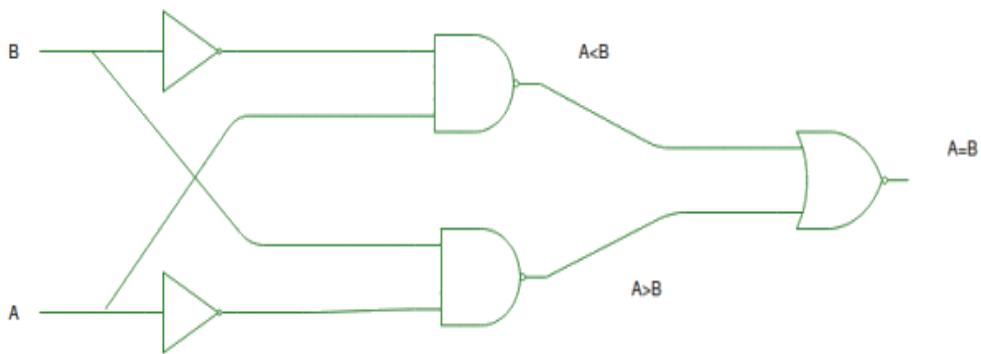


Figure-3: Logic Circuit of 1-Bit Comparator

2) 2-Bit Magnitude Comparator :

A comparator used to compare two binary numbers each of two bits is called a 2-bit magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers.

The truth table for a 2-bit comparator is given below:

INPUT				OUTPUT		
A ₁	A ₀	B ₁	B ₀	A < B	A = B	A > B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Figure-4: Truth Table of 2-Bit Comparator

From the above truth table logical expressions for each output can be expressed as follows:

$$\begin{aligned}
 A > B & : A_1 B_1' + A_0 B_1' B_0' + A_1 A_0 B_0' \\
 A = B & : A_1' A_0' B_1' B_0' + A_1' A_0 B_1' B_0 + A_1 A_0 B_1 B_0 + A_1 A_0' B_1 B_0' \\
 & : A_1' B_1' (A_0' B_0' + A_0 B_0) + A_1 B_1 (A_0 B_0 + A_0' B_0') \\
 & : (A_0 B_0 + A_0' B_0') (A_1 B_1 + A_1' B_1') \\
 & : (A_0 \text{ Ex-Nor } B_0) (A_1 \text{ Ex-Nor } B_1)
 \end{aligned}$$

$$A < B : A_1'B_1 + A_0'B_1B_0 + A_1'A_0'B_0$$

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below :

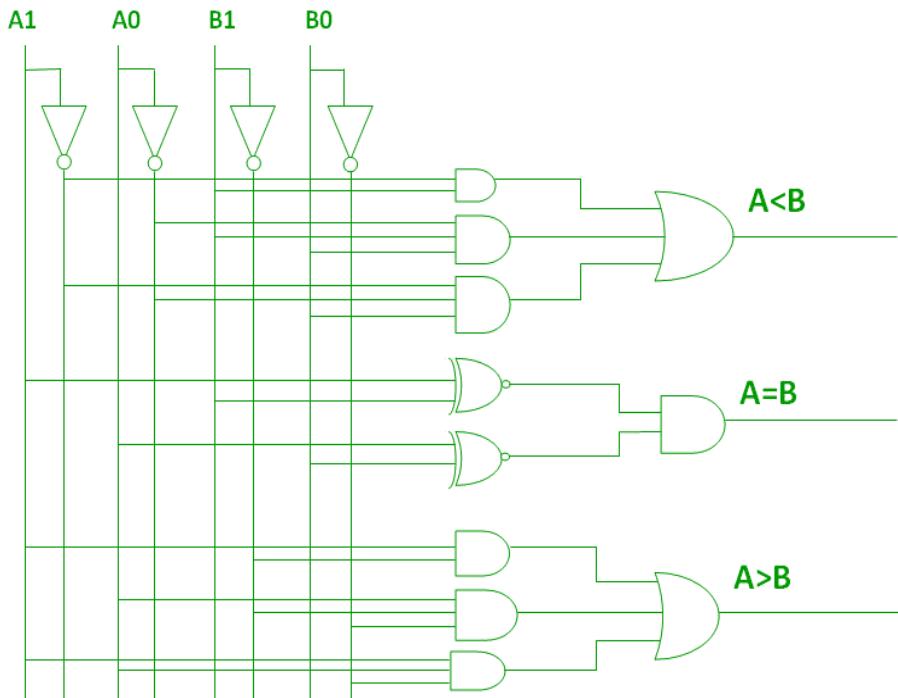


Figure-5: Logic Circuit of 2-Bit Comparator

- ### Applications of Comparators :
1. Comparators are used in central processing units (CPUs) and microcontrollers (MCUs).
 2. These are used in control applications in which the binary numbers representing physical variables such as temperature, position, etc. are compared with a reference value.
 3. Comparators are also used as process controllers and for Servo motor control.
 4. Used in password verification and biometric applications.

PRETEST:

DE Journal Virtual Labs

de-iitr.vlabs.ac.in/exp/comparator-using-logic-gates/pretest.html

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Electronics IITR > Experiments

Aim

Theory

Pretest

All the comparisons made by comparator is done using _____.

a: 1 circuit
 b: 2 circuits
 c: 3 circuits
 d: 4 circuits

One that is not the outcome of magnitude comparator is _____.

a: $a > b$
 b: $a - b$
 c: $a < b$
 d: $a + b$

If two numbers are not equal then binary variable will be _____.

a: 0
 b: 1
 c: A

Type here to search

30°C Haze 12:06 05-01-2022

DE Journal Virtual Labs

HOME PARTNERS CONTACT

If two numbers are not equal then binary variable will be _____.

a: 0
 b: 1
 c: A
 d: B

Which one is a basic comparator?

a: AND
 b: Ex-OR
 c: NAND
 d: Ex-NOR

Comparators are used in _____.

a: Memory
 b: Motherboard
 c: CPU
 d: Hard drive

Submit Quiz

5 out of 5

Type here to search

30°C Haze 12:06 05-01-2022

PROCEDURE:

1- Bit Comparator

Simulator 1:

- Step-1) Switch ON the power supply button to supply 5V to the circuit.
 Step-2) Press Switch 1 for input A and Switch 2 for input B.



The switch in ON state is and the switch in OFF state is
 Step-3) i)When the input A is greater than the input B,LED 1 lits up.
 ii)When the input A is lesser than the input B,LED 2 lits up.
 iii)When the input A is equal to the input B,LED 3 lits up.

The LED in OFF state is  and the LED in ON state is 
Step-4) Click on "Add" Button to add data to the Truth Table.
Step-5) Repeat Steps 2 to 4 for another set of data.
Step-6) Click "Print" to get the print out of the Truth Table.

Simulator 2:

- Step-1) Enter the Boolean input "A" and "B".
Step-2) Enter the Boolean output for your corresponding inputs.
Step-3) Click on "Check" Button to verify your output.
Step-4) Click "Print" if you want to get print out of Truth Table.

2- Bit Comparator

Simulator 1:

- Step-1) Switch ON the power supply button to supply 5V to the circuit.
Step-2) Press Switch 1 for input A₁,Switch 2 for input A₀,Switch 3 for input B₁and Switch 4 for input B₀.

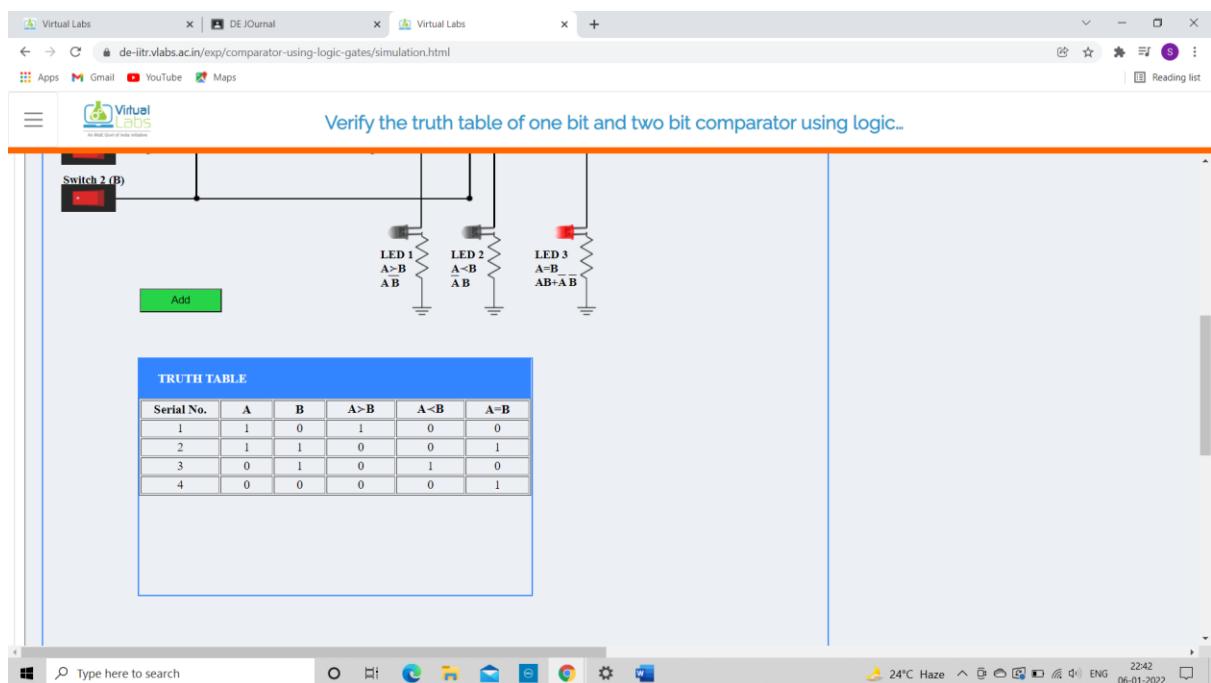
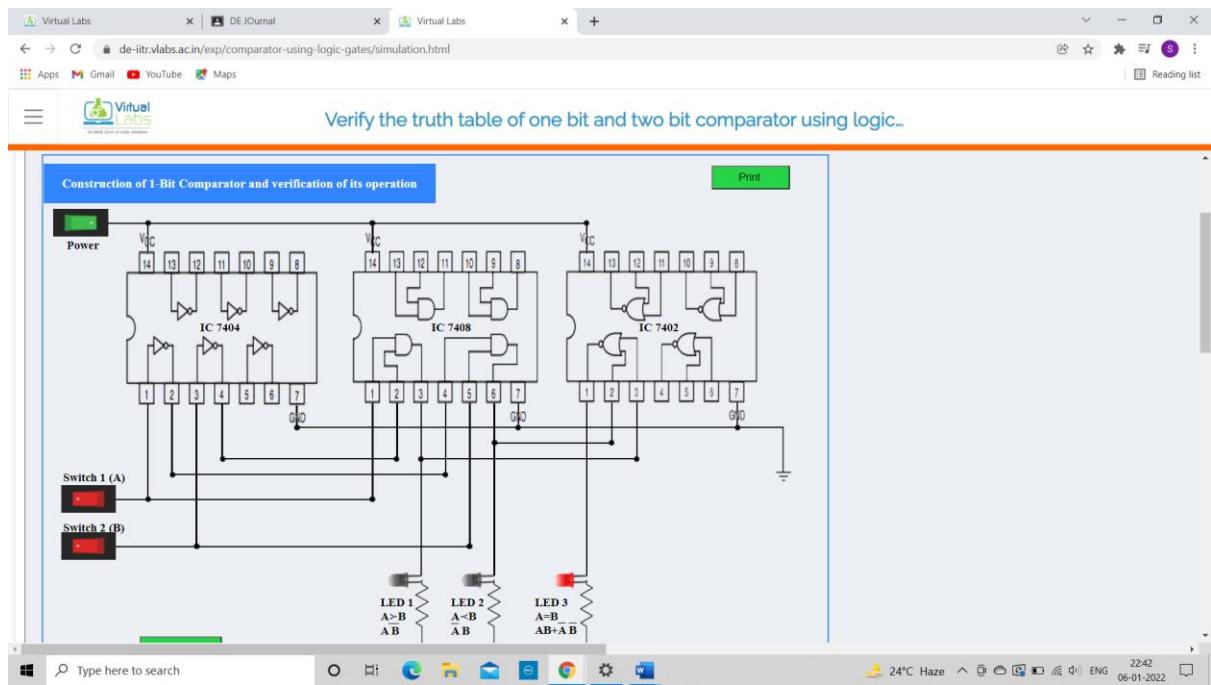
The switch in ON state is  and the switch in OFF state is 
Step-3) i)When the input A₁A₀ is greater than the input B₁B₀,LED 1 lits up.
ii)When the input A₁A₀ is lesser than the input B₁B₀,LED 2 lits up.
iii)When the input XA₁A₀ is equal to the input B₁B₀,LED 3 lits up.

The LED in OFF state is  and the LED in ON state is 
Step-4) Click on "Add" Button to add data to the Truth Table.
Step-5) Repeat Steps 2 to 4 for another set of data.
Step-6) Click "Print" to get the print out of the Truth Table.

Simulator 2:

- Step-1) Enter the two bit Boolean input "A" and "B".
Step-2) Inputs should be written such that for 'A'="A1A0" and for 'B'="B1B0"
Step-3) Enter the Boolean output for your corresponding inputs.
Step-4) Click on "Check" Button to verify your output.
Step-5) Click "Print" if you want to get print out of Truth Table.

STIMULATION:



Virtual Labs DE Journal Virtual Labs

Verify the truth table of one bit and two bit comparator using logic...

Verification of truth table of one bit Comparator

$C = \bar{A}\bar{B}$ $\Rightarrow A < B$

$D = \bar{A}\bar{B} + \bar{A}\bar{B}\bar{B}$ $\Rightarrow A = B$

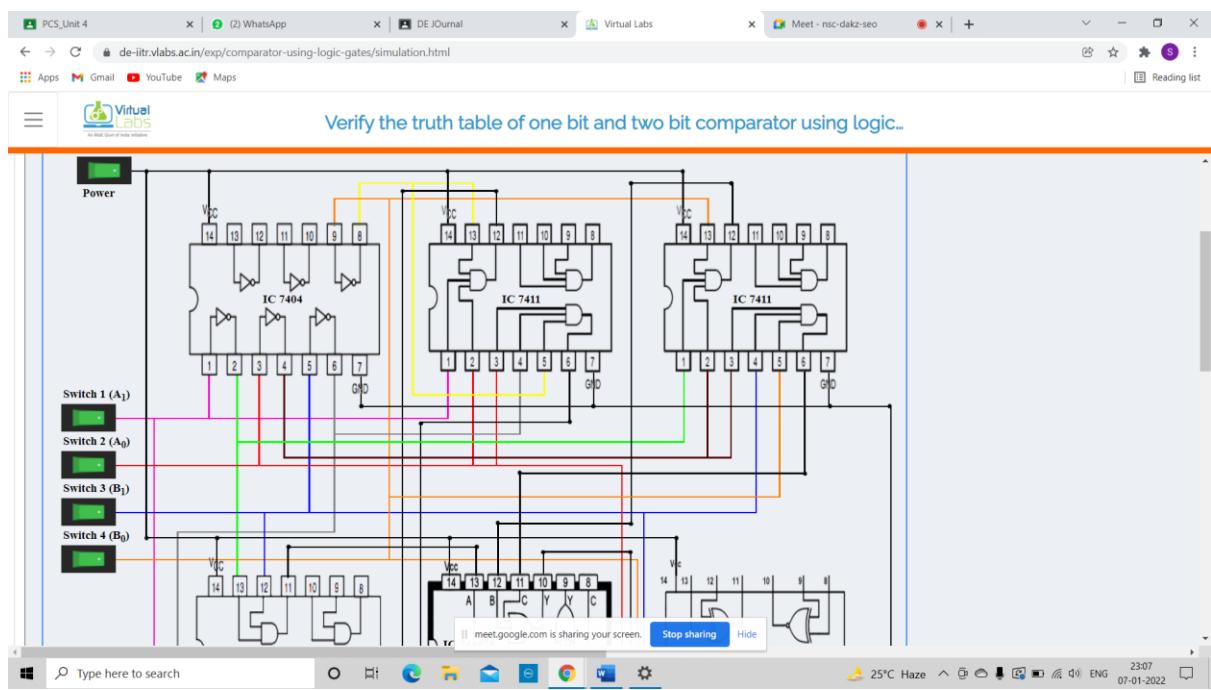
$E = \bar{A}\bar{B}\bar{B}$ $\Rightarrow A > B$

Check

TRUTH TABLE

Serial No.	A	B	$A < B$	$A = B$	$A > B$	Remarks
1	0	1	1	0	0	Correct
2	1	1	0	1	0	Correct
3	1	0	0	0	1	Correct
4	0	0	0	1	0	Correct

Type here to search 24°C Haze 22:44 06-01-2022



Verify the truth table of one bit and two bit comparator using logic...

LED 3

$$A=B \\ AB+\bar{A}B$$

LED 2

$$A>B \\ A \\ B$$

LED 1

$$A < B \\ A \\ B$$

TRUTH TABLE

Serial No.	A ₁	A ₀	B ₁	B ₀	A>B	A<B	A=B
1	0	0	0	0	0	0	1
2	0	0	0	1	0	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	0	1	0
5	0	1	0	0	1	0	0
6	0	1	0	1	0	0	1
7	0	1	1	0	0	1	0
8	0	1	1	1	0	1	0
9	1	0	0	0	1	0	0
10	1	0	0	1	1	0	0
11	1	0	1	0	0	0	1
12	1	0	1	1	0	1	0
13	1	1	0	0	1	0	0
14	1	1	0	1	1	0	0
15	1	1	1	0	1	0	0
16	1	1	1	1	0	0	1

COLOR OF INPUTS

Color	Inputs
pink	A ₁
green	A ₁ '
red	A ₀
dark blue	A ₀ '
blue	B ₁
grey	B ₁ '
orange	B ₀
yellow	B ₀ '

POSTTEST:

Verify the truth table of one bit and two bit comparator using logic gates,

Theory

The purpose of a Digital Comparator is _____.

a: To convert analog input into digital
 b: To create different outputs
 c: To add a set of different numbers
 d: To compare a set of variables or unknown numbers

Procedure

A magnitude comparator is defined as a digital comparator which has _____.

a: Only one output terminal
 b: Two output terminals
 c: Three output terminals
 d: No output terminal

Posttest

In a comparator, if we get input as A>B then the output will be _____.

a: B
 b: 1
 c: A
 d: o

In a comparator, if we get input as $A > B$ then the output will be _____.

a: B
 b: 1
 c: A
 d: o

How many inputs are required for a digital comparator ?

a: 1
 b: 3
 c: 2
 d: 4

An identify comparator is defined as a digital comparator which has _____.

a: Only one output terminal
 b: Two output terminals
 c: Three output terminals
 d: No output terminal

Submit Quiz

5 out of 5

CONCLUSION:

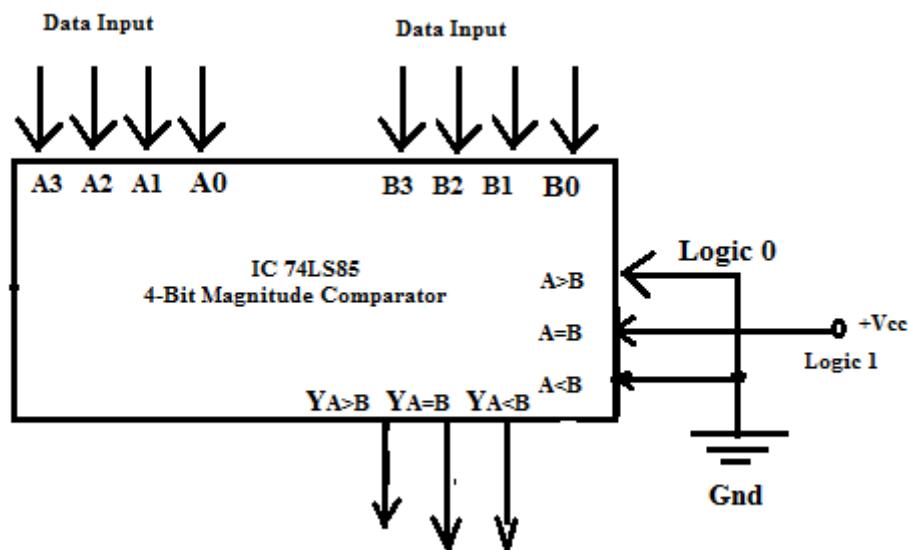
Thus, the truth table of 1-bit comparator was verified by using NOT, AND and NOR logic gate ICs and 2-bit comparator by using 1-input NOT, 3-input AND , 2-input AND, 3-input OR and 2-input Ex-NOR logic gate ICs.

D] Implementation of 4-bit digital comparator using MSI ICs

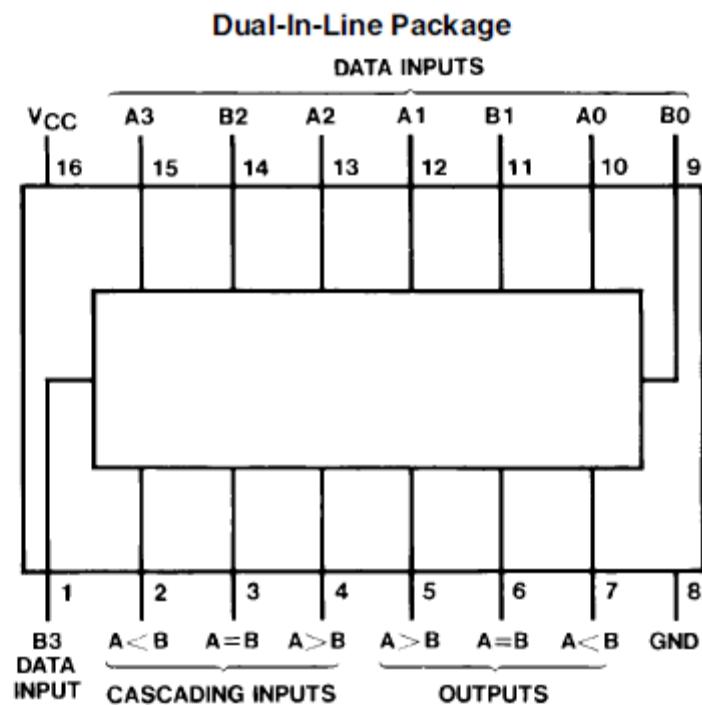
AIM: To design and implement 4-bit Comparator using IC 74LS85

THEORY: IC 74LS85 is a four-bit magnitude comparator that compares Binary Numbers. It checks whether a 4-bit binary number $A_3 A_2 A_1 A_0$ is greater than, less than or equal to another 4-bit number $B_3 B_2 B_1 B_0$. The IC has two sets of four bit inputs to be compared and also has three cascade inputs $A > B$, $A = B$ and $A < B$ and produces three outputs $A > B$, $A = B$ and $A < B$. These devices are fully expandable to any number of bits without external gates.

Diagram:



Connection Diagram

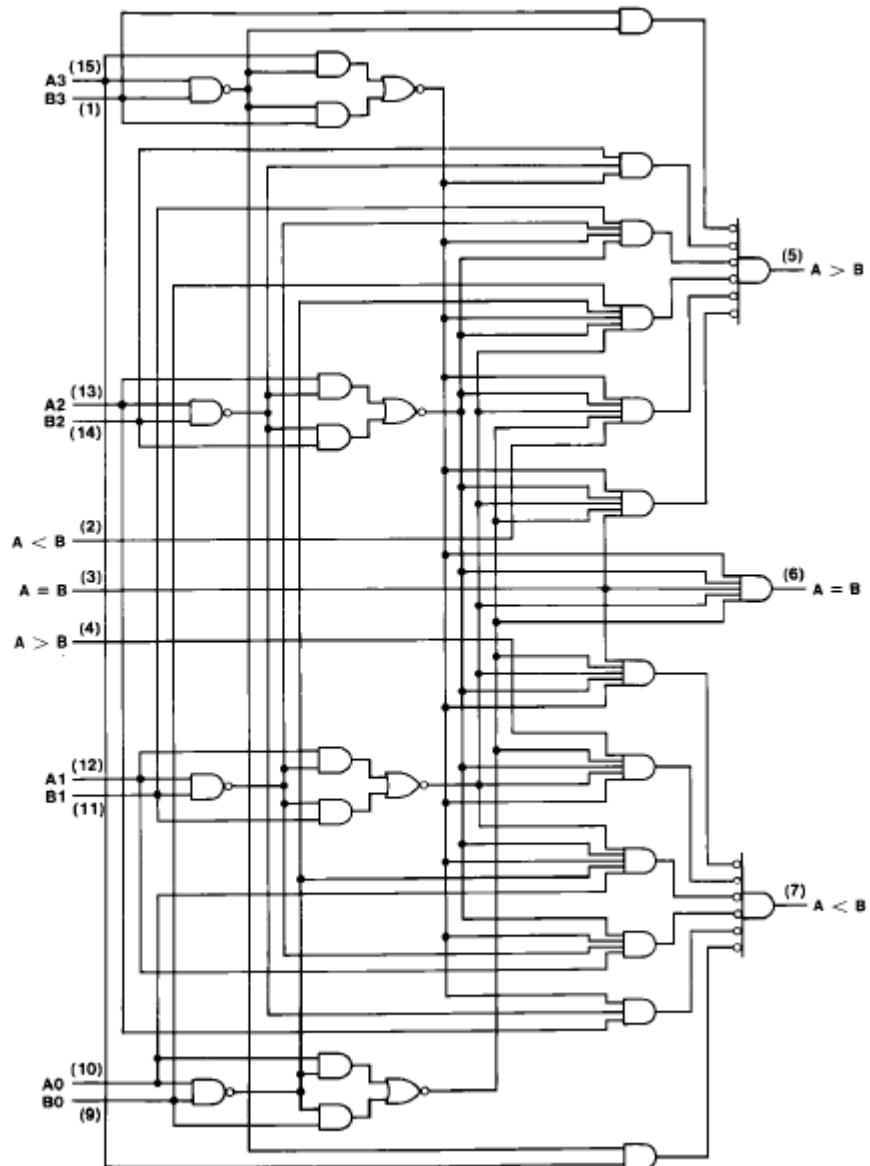


Function Table

Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

H = High Level, L = Low Level, X = Don't Care

Logic Diagram



Numerical:

The Comparator compare the number bit by bit from MSB to LSB.

The function table of 4-bit comparator is:

Comparing Input		Output		
A ₃ , B ₃	A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	A > B A = B A < B
A ₃ > B ₃ X	X	X	1	0 0
A ₃ < B ₃ X	X	X	0	0 1
A ₃ = B ₃ A ₂ > B ₂ X	X	1	0	0
A ₃ = B ₃ A ₂ < B ₂ X	X	0	0	1
A ₃ = B ₃ A ₂ = B ₂ A ₁ > B ₁ X	1	0	0	0
A ₃ = B ₃ A ₂ = B ₂ A ₁ < B ₁ X	0	0	1	0
A ₃ = B ₃ A ₂ = B ₂ A ₁ = B ₁ A ₀ > B ₀ 1	0	0	0	0
A ₃ = B ₃ A ₂ = B ₂ A ₁ = B ₁ A ₀ < B ₀ 0	0	1	0	1
A ₃ = B ₃ A ₂ = B ₂ A ₁ = B ₁ A ₀ = B ₀ 0	1	0	0	0

Example:

1. A = 0, B = 0

A = 0 0000

B = 0 0000

Ans:

A>BA=BA<B

Low High Low

- 2.

3. A = 0, B = 2

A = 0 0000

B = 2 0010

Ans:

A>BA=BA<B

Low Low High

- 4.

5. A = 4, B = 2

A = 4 0100

B = 2 0010

Ans:

A>BA=BA<B

High Low Low

PRETEST:

The screenshot shows a web browser window with the URL dld-iitb.vlabs.ac.in/exp/four-bit-digital-comparator/pretest.html. The page is titled "Implementation of 4-bit digital comparator using MSI ICs". On the left, there is a sidebar with navigation links: Aim, Theory, Pretest (which is highlighted in orange), Procedure, Simulation, Posttest, References, and Feedback. The main content area contains several questions with multiple-choice answers. At the bottom, there is a search bar and a taskbar with various icons.

Pretest

IC 74LS85 compares _____.

a : Binary numbers.
 b : BCD codes
 c : Decimal numbers
 d : Hex numbers

The IC 74LS85 has ___ Outputs.

a : 1
 b : 2
 c : 3
 d : 4

The IC 74LS85 has ___ cascading inputs.

a : 1
 b : 2
 c : 3

PROCEDURE:

Steps:

1. Switch ON the circuit, by pressing Main Switch.
2. Set the appropriate Inputs.
3. After giving the inputs, observe the corresponding outputs.
4. Verify the results.

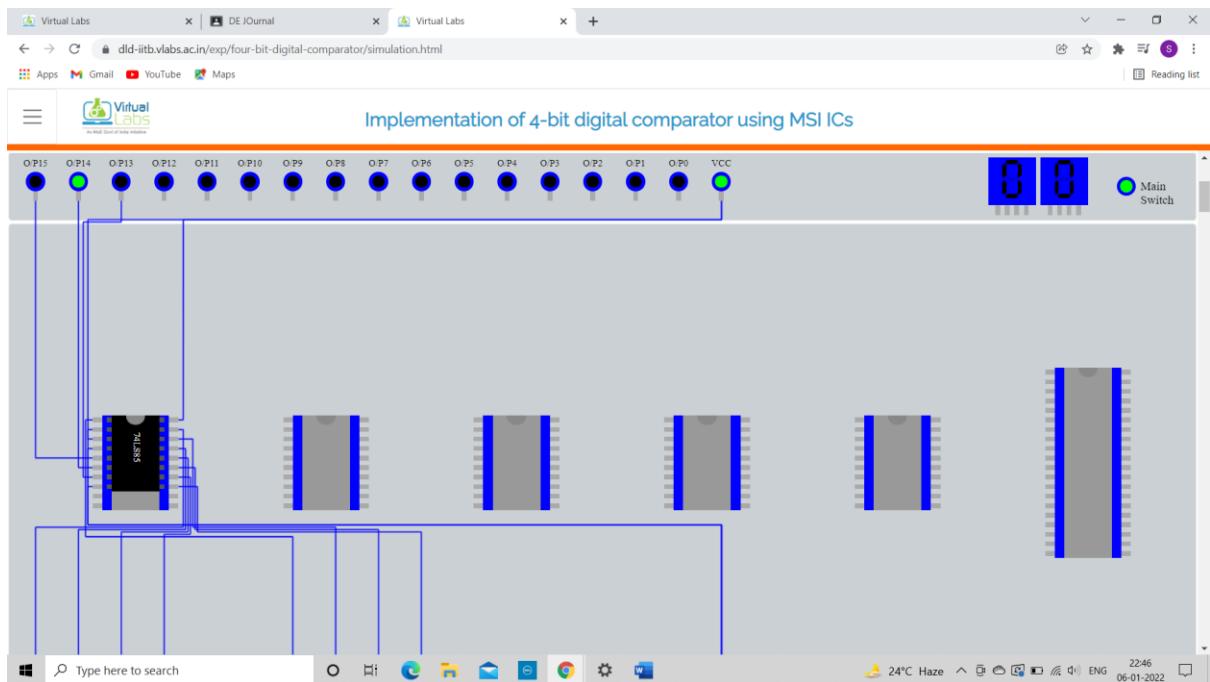
Inputs:

1. Input A (I/P15 to I/P12)
2. Input B (I/P9 to I/P6)

Outputs:

1. A>B (O/P15)
2. A=B (O/P14)
3. A<B (O/P13)

STIMULATION:



Truth Table :

I/P15	I/P14	I/P13	I/P12	I/P9	I/P8	I/P7	I/P6	O/P15	O/P14	O/P13
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	1	1	0	0	0	1
0	0	0	0	0	1	1	1	0	0	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	1
0	0	0	0	1	0	1	0	0	0	1
0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0	0

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

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

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

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

Virtual Labs DE Journal Virtual Labs

dld-iitb.vlabs.ac.in/exp/four-bit-digital-comparator/simulation.html

Apps Gmail YouTube Maps

Reading list

Implementation of 4-bit digital comparator using MSI ICs

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

Type here to search 24°C Haze 22:48 06-01-2022

Virtual Labs DE Journal Virtual Labs

dld-iitb.vlabs.ac.in/exp/four-bit-digital-comparator/simulation.html

Apps Gmail YouTube Maps

Reading list

Implementation of 4-bit digital comparator using MSI ICs

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

Type here to search 24°C Haze 22:48 06-01-2022

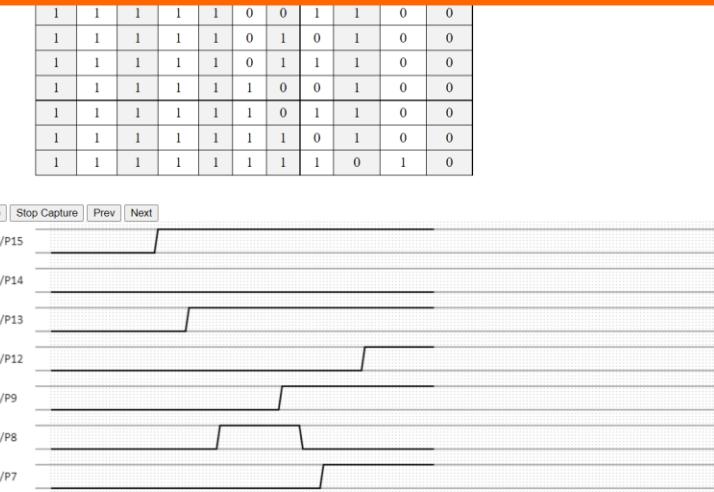
Implementation of 4-bit digital comparator using MSI ICs									
1	0	0	0	1	1	0	1	0	0
1	0	0	0	1	1	1	0	0	0
1	0	0	0	1	1	1	1	0	0
1	0	0	1	0	0	0	0	1	0
1	0	0	1	0	0	0	1	1	0
1	0	0	1	0	0	1	0	1	0
1	0	0	1	0	0	1	0	1	0
1	0	0	1	0	0	1	1	1	0
1	0	0	1	0	1	0	0	1	0
1	0	0	1	0	1	0	1	1	0
1	0	0	1	0	1	1	0	1	0
1	0	0	1	0	1	1	1	1	0
1	0	0	1	1	0	0	0	1	0
1	0	0	1	1	0	1	0	0	1
1	0	0	1	1	1	0	0	0	1
1	0	0	1	1	1	0	0	0	1
1	0	0	1	1	1	1	0	0	1
1	0	0	1	1	1	1	1	0	1
1	0	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	1	0	0

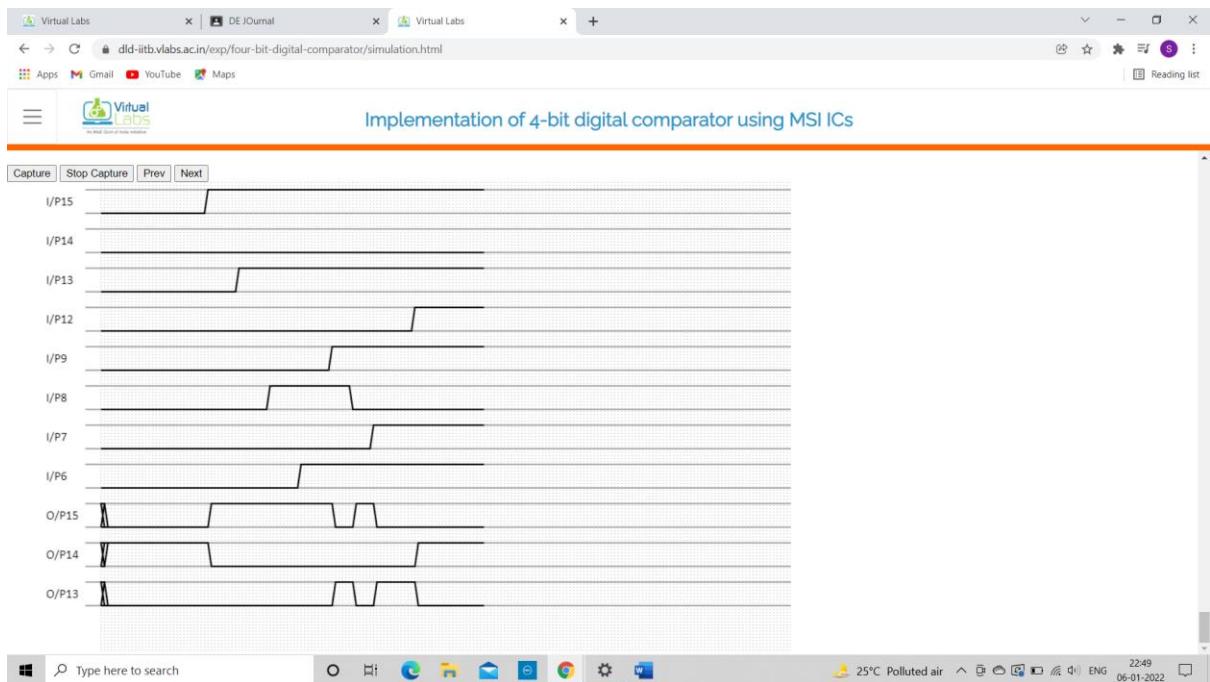
Implementation of 4-bit digital comparator using MSI ICs									
1	0	1	0	0	0	0	1	1	0
1	0	1	0	0	0	1	0	1	0
1	0	1	0	0	0	1	1	1	0
1	0	1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	1	1	0
1	0	1	0	1	0	0	0	1	0
1	0	1	0	1	0	0	1	0	0
1	0	1	0	1	0	0	1	1	0
1	0	1	0	1	0	1	0	0	1
1	0	1	0	1	0	1	1	0	0
1	0	1	0	1	1	0	0	0	1
1	0	1	0	1	1	1	0	0	1
1	0	1	0	1	1	1	1	0	1
1	0	1	1	0	0	0	0	1	0
1	0	1	1	0	0	0	1	1	0
1	0	1	1	0	0	1	1	1	0
1	0	1	1	0	1	0	0	1	0

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

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

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





POSTTEST:

[DE JOURNAL](#) [Virtual Labs](#)

← → ⌛ 🔍 did-iitb.vlabs.ac.in/exp/four-bit-digital-comparator/posttest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Aim

Theory

Pretest

IC 74LS85 is a _____.
 a : Four-bit magnitude comparator.
 b : Adder
 c : Subtractor
 d : Multiplexer

Procedure

Simulation

Posttest

The IC 74LS85 has ___ sets of four bit inputs to be compared
 a : 1
 b : 2.
 c : 3
 d : 4

One that is not outcome of magnitude comparator is
 a : a>b
 b : a=b
 c : a<b

Type here to search

25°C Polluted air 22:49 06-01-2022

25°C Haze 12:11 05-01-2022

The screenshot shows a Microsoft Edge browser window with two tabs open: "DE Journal" and "Virtual Labs". The "Virtual Labs" tab is active and displays a quiz from "dld-iitb.vlabs.ac.in/exp/four-bit-digital-comparator/postest.html".

The quiz consists of three questions:

- One that is not outcome of magnitude comparator is**
 - a: a**a**
 - b: a**-**
 - c: a**b**
 - d: a**+b**
- Output of comparator when A = 1010 and B = 1011 is _____.**
 - a: A > B = 1, A < B = 0, A - B = 1
 - b: A > B = 0, A < B = 1, A - B = 0
 - c: A > B = 1, A < B = 0, A - B = 0
 - d: A > B = 0, A < B = 1, A - B = 1
- Output of comparator when A = 1100 and B = 1010 is _____.**
 - a: A > B = 1, A < B = 0, A - B = 1
 - b: A > B = 0, A < B = 1, A - B = 0
 - c: A > B = 1, A < B = 0, A - B = 0
 - d: A > B = 0, A < B = 1, A - B = 1

A blue "Submit Quiz" button is visible at the bottom left of the quiz area. Below the quiz, it says "5 out of 5".

The browser's taskbar at the bottom shows various pinned icons and the system tray on the right indicating "30°C Haze", "12:12", "05-01-2022", and battery status.

CONCLUSION:

Thus, a 4-bit Comparator was designed and implemented using IC 74LS85.

Practical no. 7 Implement Encode and Decoder and Multiplexer and Demultiplexers

A] Multiplexers using Basic Logic Gates

AIM: The students will be able to understand the concept of multiplexers and its implementation using logic gates. The students will be able to understand the concept of encoder.

Students will be able:

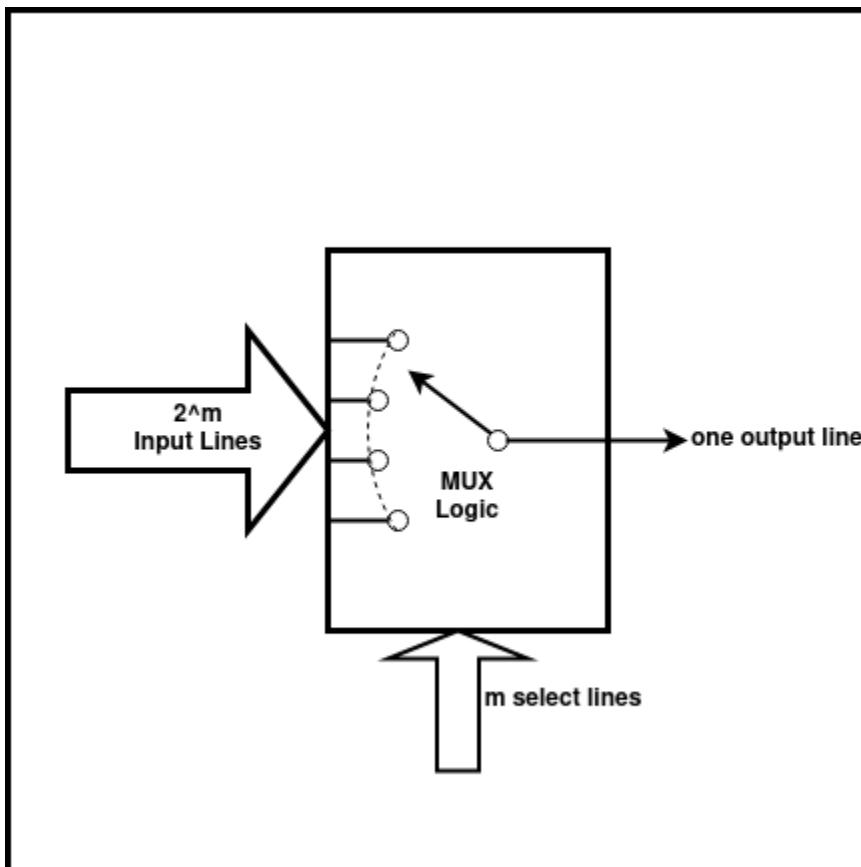
1. To select appropriate gates to build a multiplexer circuit.
2. To identify the role of a multiplexer circuit.
3. To design, construct and verify the operation of a 2:1 multiplexer using basic logic gates

THEORY:

1.1

Introduction

A digital multiplexer is similar to a multi-position switch with many inputs and only one output. It has control inputs to select a particular input. It is also called as a channel selector and abbreviated as Mux.



A multiplexer has m select lines, 2^m inputs and only one output as shown in the figure. Commercially, multiplexers ICs(integrated circuits) are available in powers of 2, e.g. 2:1, 4:1, 8:1 and 16:1 Mux.

No.of select lines (m)	No.of inputs (2^m)	No. of Outputs Y	Multiplexer Type
1	2	1	2:1

No.of select lines (m)	No.of inputs (2^m)	No. of Outputs Y	Multiplexer Type
2	4	1	4:1
3	8	1	8:1
4	16	1	16:1

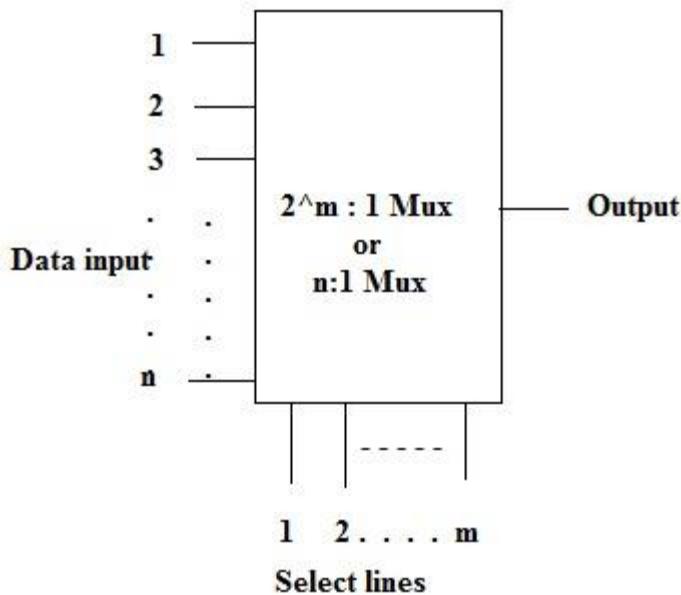
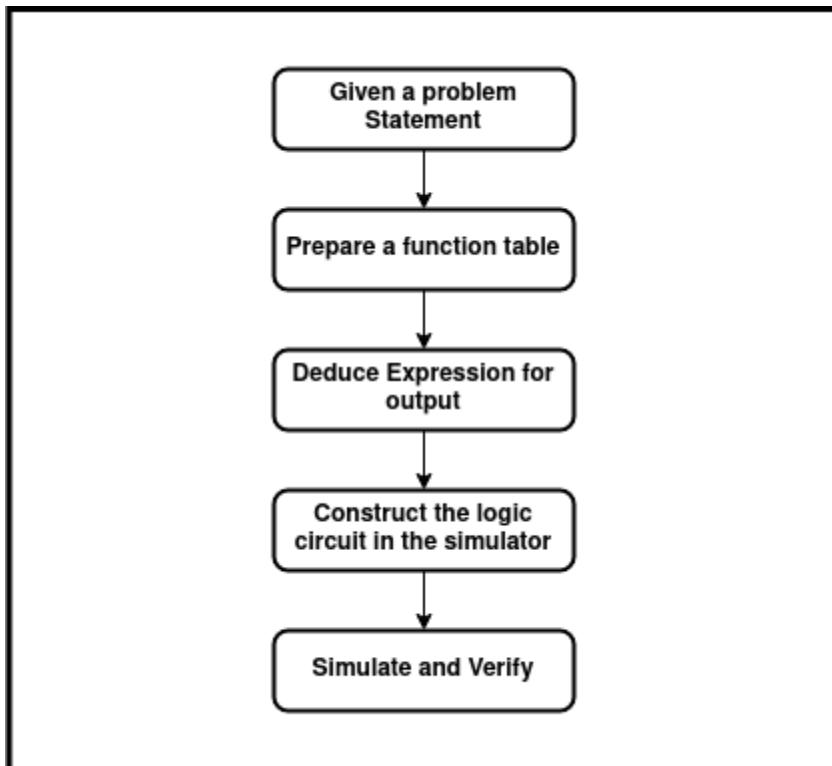


Fig.1 Block Diagram of multiplexer

It is easier to build multiplexers using gates (small scale integration -SSI Ics) for a few select lines. But as the number of select lines increases, use of medium scale integration MSI Ics becomes the best choice. TTL IC 74LS150 is a 16:1 Mux. In this experiment design of smaller size Muxes are considered. A 2:1(read as 2 as to 1) multiplexer can be designed using: a. basic logic gates b. universal gates (NAND & NOR).

1.2.Example 1: Design of 2:1 Mux using basic logic gates A simple 2:1 Mux will have 2 input lines D0 & D1 and one select line S0 and a single output Y. The select line can take a value either 0 or 1: a. If S0 takes a value 0, the input D0 is selected and the output Y = D0. b. If S0 takes a value 1, the input D1 is selected and the output Y = D1.

1.2.1 Theoretical Design: The flow diagram explains the steps involved in the design of the circuit:



Apply these steps to design a 2:1 Mux using logic gates:

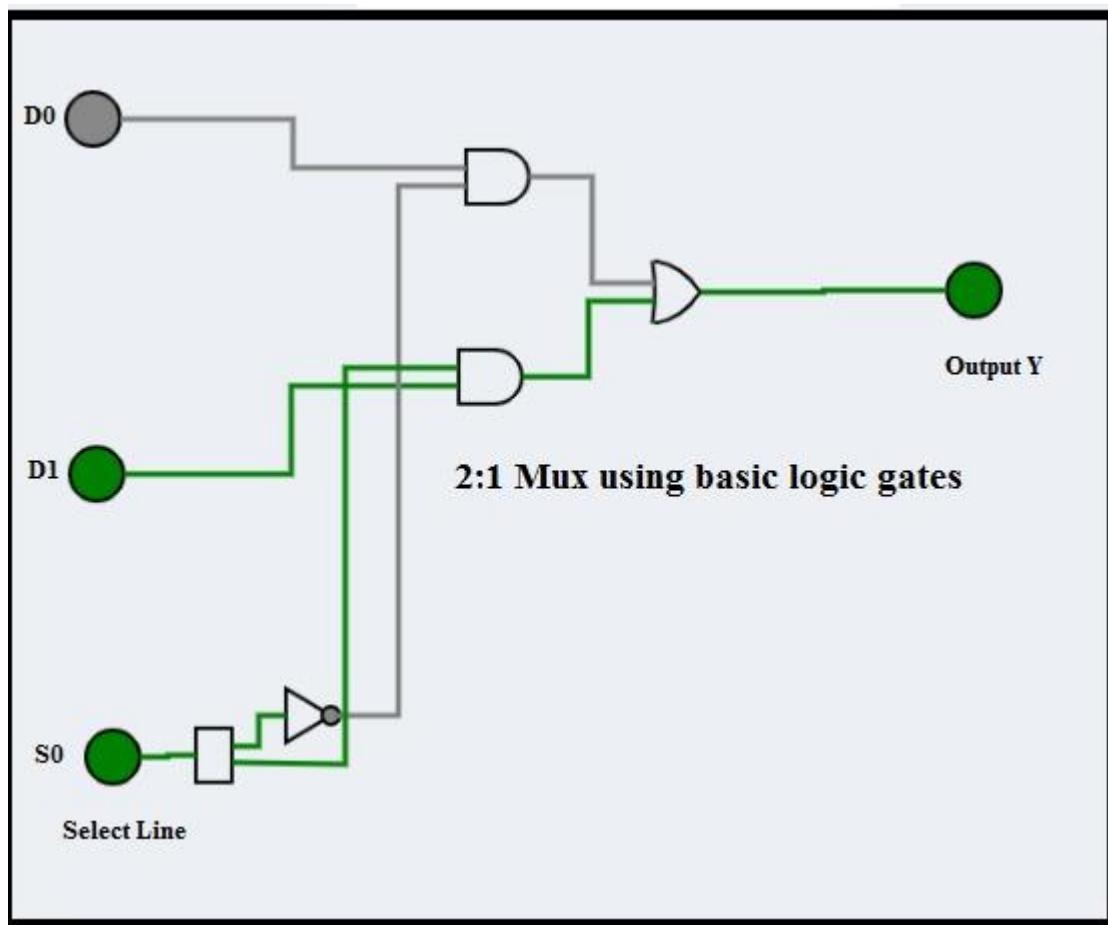
a. Prepare the function table of the 2:1 Mux:

Select I/p	Inputs		Fundamental Product FP	Output Y	Output Y (in terms of input)
S0	D1	D0			
0	X	0	S0'. D0'	0	$Y = D0$
0	X	1	S0'. D0	1	
1	0	X	S0 . D1'	0	$Y = D1$
1	1	X	S0 . D1	1	

b. Formulate the expression for output Y by considering only those FPs for which the output is 1. $Y = S0'. D0 + S0 . D1$ The simplified function can be tabulated as:

Select Input S0	Output Y
0	$Y = D_0$
1	$Y = D_1$

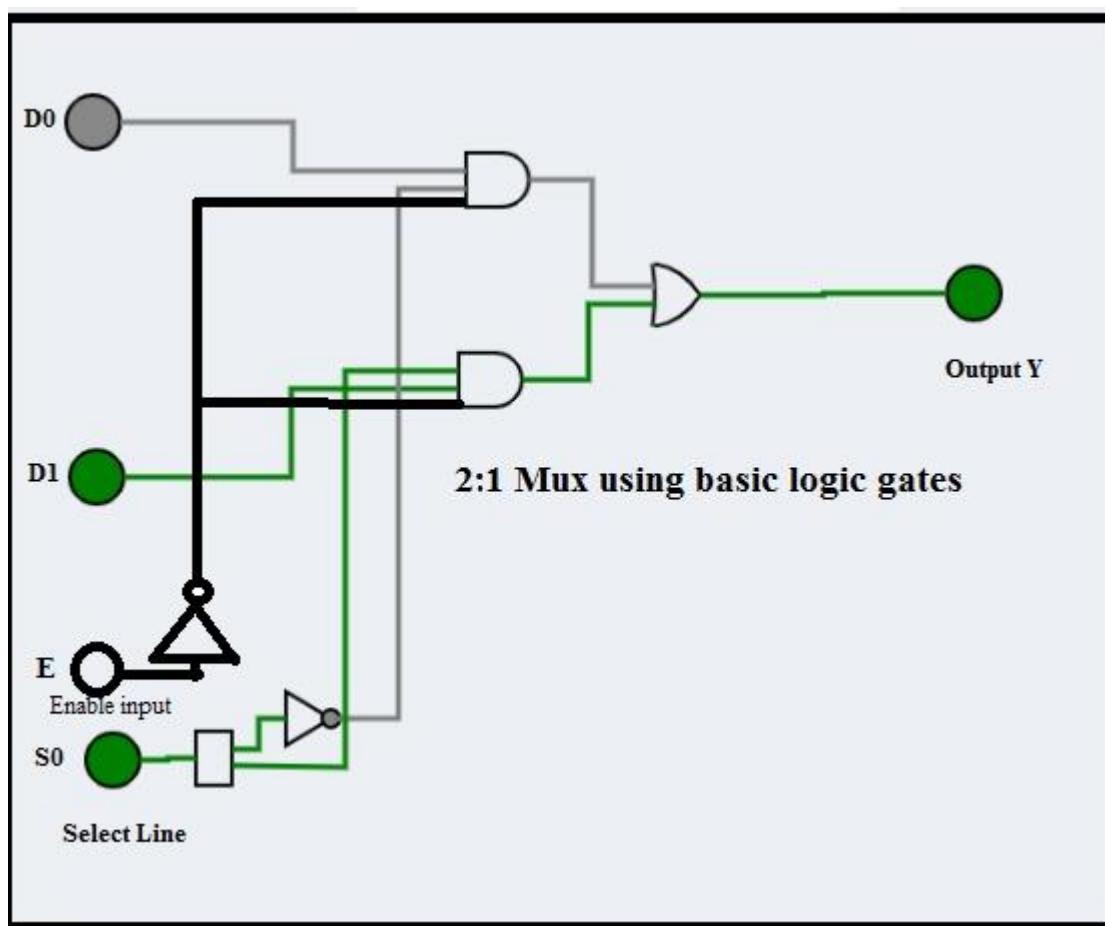
c. Draw the logic diagram for the expression:



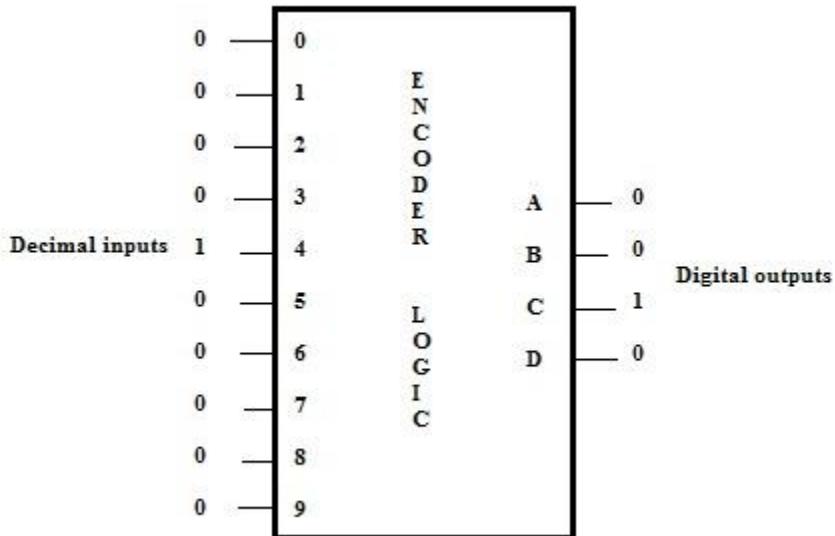
d. Using simulator construct the circuit and verify its operation.

e. The same circuit can also be designed by having an active low enable input as shown in image. The simplified function can be tabulated as:

Select Input S0	Enable Input E	Output Y
X	1	0
0	0	$Y = D_0$
1	0	$Y = D_1$



1.3. Encoder Logic The figure shows the concept of encoder wherein input line 4 is high and all other inputs are low. The output of the encoder is a decimal 4, whose equivalent binary is 0100. The concept of decimal to binary(10 to 4 binary) encoder is shown in the figure given below. On similar lines an octal-binary encoder will have eight inputs and produce 3-bit binary output.



PRETEST:

DE Journal x Virtual Labs x +

dld-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/pretest.html

Apps Gmail YouTube Maps

 An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Aim	Multiplexers using Basic Logic Gates
Theory	A multiplexer is a _____.
Pretest	<input type="radio"/> a : data selector/multi-position switch. <input type="radio"/> b : parallel to serial converter. <input type="radio"/> c : design alternative to sum-of-product solutions. <input checked="" type="radio"/> d : All of these
Procedure	How many select lines are required for a 32:1 Mux?
Simulation	<input type="radio"/> a: Two <input type="radio"/> b: Four <input checked="" type="radio"/> c: Five <input type="radio"/> d: Eight
Posttest	Any n-variable truth table can be implemented using a universal logic circuit referred to as multiplexer. State True or False.
References	<input checked="" type="radio"/> a: True <input type="radio"/> b: False
Feedback	

Type here to search

30°C Haze 12:15 05-01-2022

DE Journal x Virtual Labs x +

[ddl-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/pretest.html](#)

App Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

d : Eight

Any n-variable truth table can be implemented using a universal logic circuit referred to as multiplexer. State True or False.

a : True

b : False

The enable input in a multiplexer is also referred to as_____

a : decode input

b : strobe input

c : Select input

d : All of these

For the circuit shown below, input $S_1 = 1, S_0 = 0$, then the output is_____

00

01

Y

Type here to search

DE Journal x Virtual Labs x +

[ddl-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/pretest.html](#)

App Gmail YouTube Maps

30°C Haze 12:16 05-01-2022

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

00

01

10

11

Y

S1 S0

4:1 Mux

a : $Y = 0$

b : $Y = 1$

c : $Y = X$

d : None of these

Submit Quiz

5 out of 5

Type here to search

DE Journal x Virtual Labs x +

[ddl-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/pretest.html](#)

App Gmail YouTube Maps

30°C Haze 12:16 05-01-2022

PROCEDURE:

Simulation Screen 1: Quiz – To select appropriate gates to build a multiplexer circuit.

1. Click on “Start Quiz”.
2. Click on the appropriate gate.
3. Click on “Check answer”.
4. If answer is correct, click “Go to next question”.
5. If answer is incorrect, “Wrong gate selected” pops up, click ok to try again and proceed.
6. Repeat steps 2 to 5 and complete the quiz.

7. In first attempt if all 3 questions are answered correctly, you get an overall score of 5 points. For any incorrect answers the points are deducted.
9. For overall score lesser than 2 , you get a prompt “Please try the **THEORY** again”. Click ok. Click on **THEORY** tab and revise the concepts and repeat steps 1 to 8.
10. In case of score greater than equal to 2, proceed to click on “Next stage”.

Simulation Screen 2: Juice box – To identify the role of a multiplexer circuit.

1. For the given question, click on select inputs S2, S1 and S0 appropriately.
2. Click on submit.
3. For incorrect answer, click ok to try again.
4. If answer is correct, Bingo! Click ok. Repeat steps 1 to 4 for next question.
5. Select appropriate answer from the drop down box for the next 2 questions.
6. For incorrect answer, click ok to try again.
7. If answer is correct, Bingo! Click ok. Repeat steps 5 to 7 for next question.
8. Click on Next.

Simulation Screen 3: To formulate the Function table and Boolean Expression for 2:1 multiplexer

1. Select appropriate options from the drop down menu and formulate the function table.
2. Click submit.
3. If incorrect try again.
4. If correct, select the correct expression defining the multiplexer.
5. Try again if answer is incorrect.
6. Proceed to the next step. Click Next.

Simulation Screen 4: Simulator Scene – To design, construct and verify the operation of a 2:1 multiplexer using basic logic gates.

Assume the enable input is activated (i.e. it is already set to zero)

General instructions:

1. Apply the function table/Boolean expression in Screen 3 to design a 2:1 mux.
2. Click on the required object (AND, OR, NOT gate, digital switch, LED, connection wires) and click on canvas to place the component on canvas one by one.
3. Every component is treated as an object with one/two inputs and one output.
4. Digital switch output is connected as input. Click on ‘output 1 to input 1’ wire to connect the output to the first input of next gate, whereas ‘output 1 to input 2’ wire to connect the output to second input of gate.
5. Y connector has one input and two outputs. When same input is to be connected at two places, use a y connector accordingly.
6. Do connections as per the design.
7. Use undo link/undo gate or redo link/gate to make corrections if any.
8. Start simulator.
9. Set select input S0 and data inputs D0 & D1 and verify the corresponding outputs.

10. Compare the output of the design against the corresponding output in the function table of 2:1 mux given on right hand side and validate your design.

STIMULATION:

DE Journal Virtual Labs

ddl-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/simulation.html

Apps Gmail YouTube Maps

Multiplexers using Basic Logic Gates

1. Which basic logic gate will you choose on inputs of 2:1 Multiplexer?

OR XOR AND NOR NOT NAND

Go to Next Question

Type here to search DE Journal Virtual Labs

ddl-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/simulation.html

Apps Gmail YouTube Maps

Multiplexers using Basic Logic Gates

2. Which basic logic gate will you choose on select line of 2:1 Multiplexer?

OR XOR AND NOR NOT NAND

Go to Next Question



DE JOURNAL Virtual Labs

dd-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/simulation.html

Virtual Labs

did-iitb.vlabs.ac.in says
Your score is : 3

OK

3. Which basic logic gate will you choose on output of 2:1 Multiplexer?

OR XOR AND NOR NOT NAND

Go to Next Question

Type here to search

DE JOURNAL Virtual Labs

dd-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/simulation.html

Virtual Labs

did-iitb.vlabs.ac.in says
BINGO!!!

OK

JUICE BOX

Question No.1 : What will be the value of select inputs S2,S1,S0 to get orange juice ?

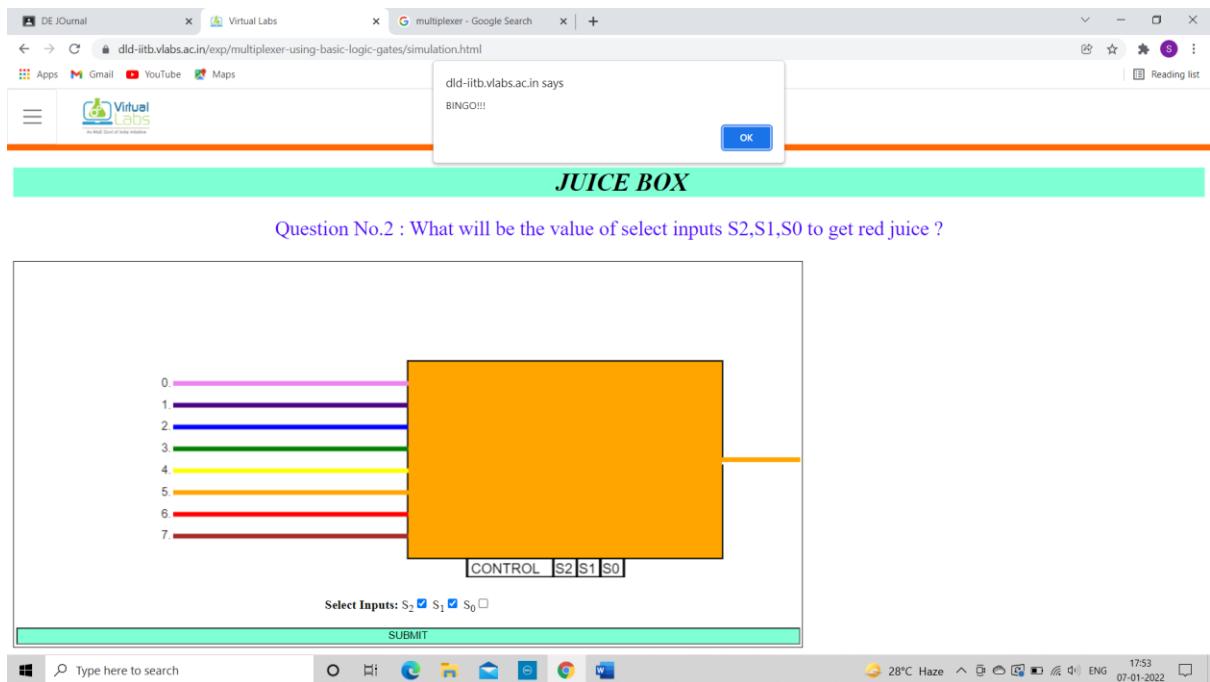
0 1 2 3 4 5 6 7

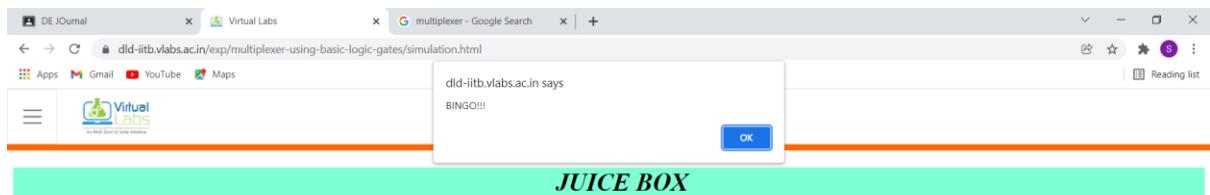
CONTROL S2 S1 S0

Select Inputs: S₂ S₁ S₀

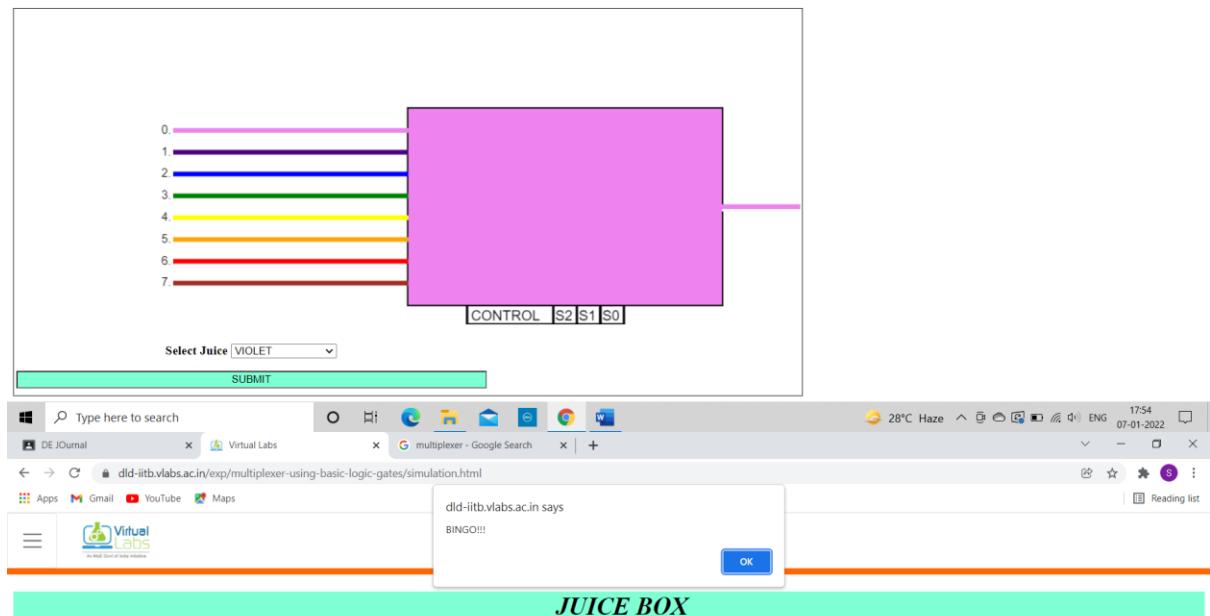
SUBMIT

28°C Haze 17:34 07-01-2022

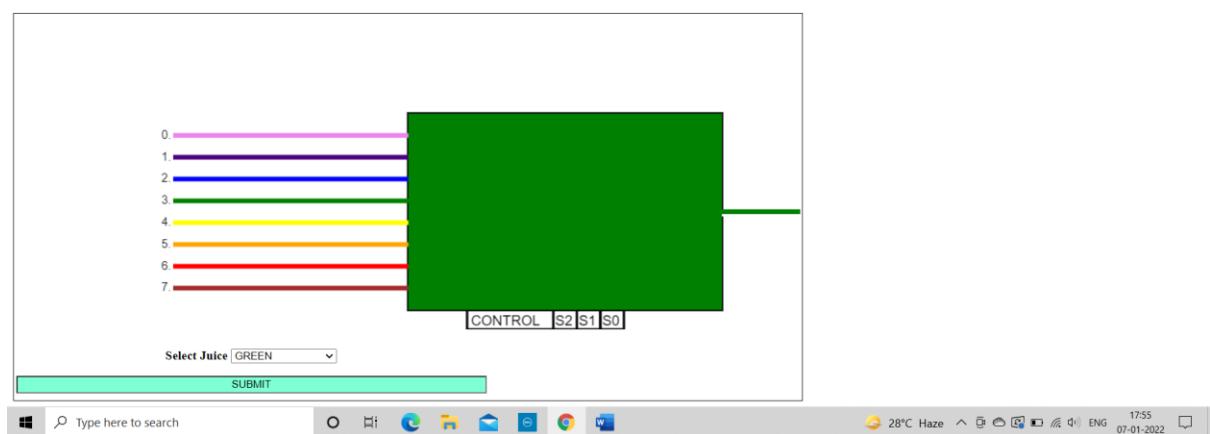




Question No.3 : Which juice would you get, if the values of select inputs are S2=0 S1=0 S0=0 ?



Question No.4 : Which juice would you get, if the values of select inputs are S2=0 S1=1 S0=1 ?



Multiplexers using Basic Logic Gates

Q1. Enter the corresponding outputs for the given 2:1 MUX

CONTROL I/Ps		OUTPUT
Enable I/P	Select I/P	O/P
E ¹	S ₀	Y
1	X	D ₀
0	0	D ₀
0	1	D ₁

2:1 Mux Representation

SCOREKEEPER

Question	Status
Q1.	Correct answer. Your score is 5
Q2.	Correct answer. Your score is 2

DE JOURNAL

Multiplexers using Basic Logic Gates

Simulator Started

Available Gates

- AND
- OR
- NOT
- Y-connector
- Green LED
- Digital Switch

Connections

Connect op1 to ip1	Connect op1 to ip2
Connect op2 to ip2	Connect op2 to ip1
UNDO GATE	UNDO LINK
REDO GATE	REDO LINK

Start Simulator

Stop Simulator

Reset

CONTROL I/Ps		OUTPUT
Enable I/P	Select I/P	O/P
E ¹	S ₀	Y
0	0	D ₀
0	1	D ₁

POSTTEST:

DE Journal x Virtual Labs x +

did-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/posttest.html

App Gmail YouTube Maps

 An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

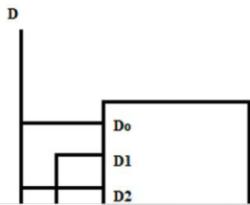
Aim
Theory
Pretest
Procedure
Simulation
Posttest
References
Feedback

Multiplexers using Basic Logic Gates

The 2:1 multiplexer using basic logic gates with active low enable input can be build using:

- a : Two AND, Two NOT, One OR gates
- b : Two AND, One NOT, One OR gates
- c : Two AND, Two NOT, Two OR gates
- d : One AND, Three NOT, One OR gates

Select the output when $S_2 = S_1 = 1$ and $S_0 = 0$.



Type here to search

DE Journal x Virtual Labs x +

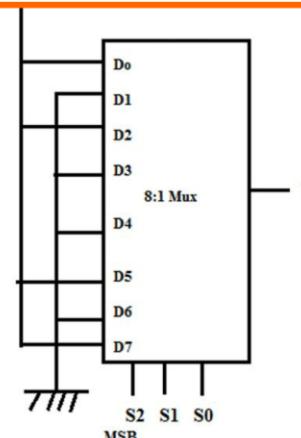
did-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/posttest.html

App Gmail YouTube Maps

30°C Haze 12:17 05-01-2022

 An MoE Govt of India Initiative

HOME PARTNERS CONTACT



a : $Y = 0$
 b : $Y = 1$

Type here to search

30°C Haze 12:18 05-01-2022

DE Journal Virtual Labs

dld-iitb.vlabs.ac.in/exp/multiplexer-using-basic-logic-gates/posttest.html

App Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

a: $Y = 0$
 b: $Y = 1$
 c: $Y = D$
 d: None of these

Identify the circuit shown below.

a: 2:1 Mux
 b: 4:1 Mux

Type here to search

DE Journal Virtual Labs

30°C Haze 12:18 05-01-2022

b: 4:1 Mux
 c: 6:1 Mux
 d: 8:1 Mux

Design, construct and verify a 2:1 Mux using basic logic gates. Get the simulation results evaluated from your faculty.

Differentiate between multiplexer and encoder?

5 out of 5

Type here to search

30°C Haze 12:18 05-01-2022

CONCLUSION:

Thus, 2:1 multiplexer was designed, constructed and verified by using basic logic gates.

B] Multiplexer using Universal Gates.

AIM: The students will be able to understand the concept of multiplexers and its implementation using logic gates. The students will be able to understand the concept of encoder.

Students will be able:

1. To design, construct and verify the operation of a 2:1 multiplexer using universal gates.

THEORY: 1.1 Introduction

Whenever there is a need to connect multiple input devices, one at a time, to a system, then a digital combinational circuit called multiplexer is useful. A digital multiplexer is similar to a multi-position switch with many inputs and only one output. It has control inputs to select a particular input. A multiplexer has m select lines, 2^m inputs and only one output as shown in the figure.

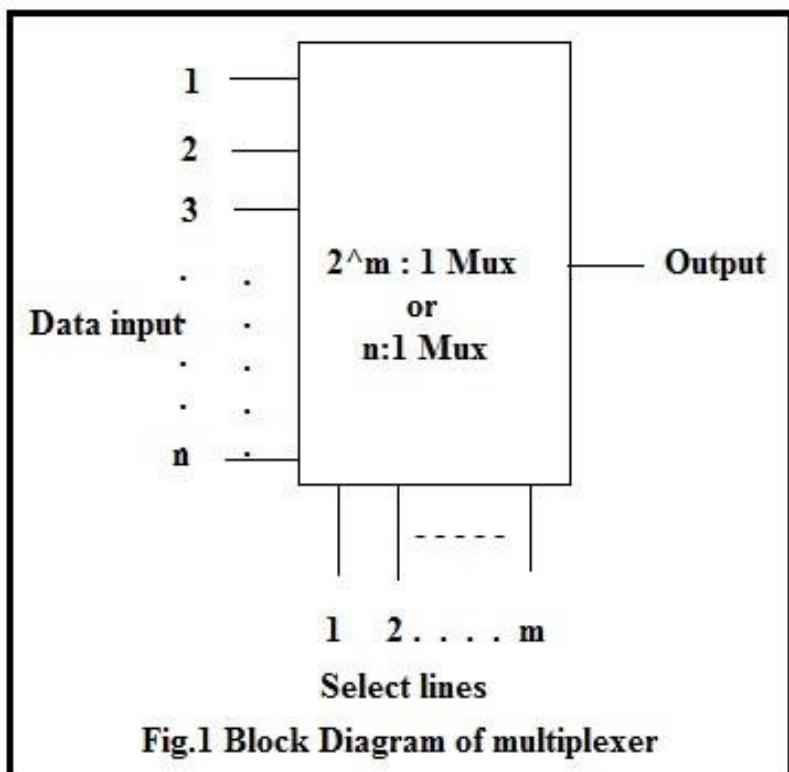


Fig.1 Block Diagram of multiplexer

It is easier to build multiplexers using gates (small scale integration -SSI ICs) for a few select lines. A 2:1(read as 2 as to 1) multiplexer can be designed using:

- a. Basic logic gates
- b. Universal gates (NAND & NOR).

A simple 2:1 Mux will have 2 input lines D0 & D1 and one select line S0 and a single output Y. The select line can take a value either 0 or 1:

- a. If S0 takes a value 0, the input D0 is selected and the output Y = D0.
- b. If S0 takes a value 1, the input D1 is selected and the output Y = D1.

1.2.1 Theoretical Design:

- a. Prepare the function table of the 2:1 Mux:

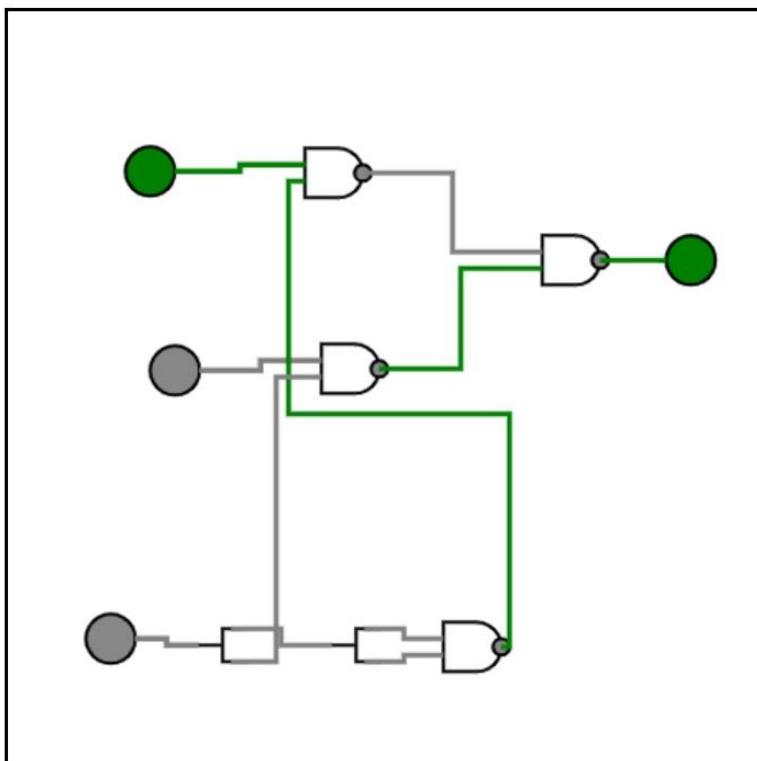
Select L/p	Inputs		Fundamental Product FP	Output Y	Output Y (in terms of input)
S0	D1	D0			
0	X	0	$S0' \cdot D0'$	0	$Y = D0$
0	X	1	$S0' \cdot D0$	1	
1	0	X	$S0 \cdot D1'$	0	$Y = D1$
1	1	X	$S0 \cdot D1$	1	

Function table

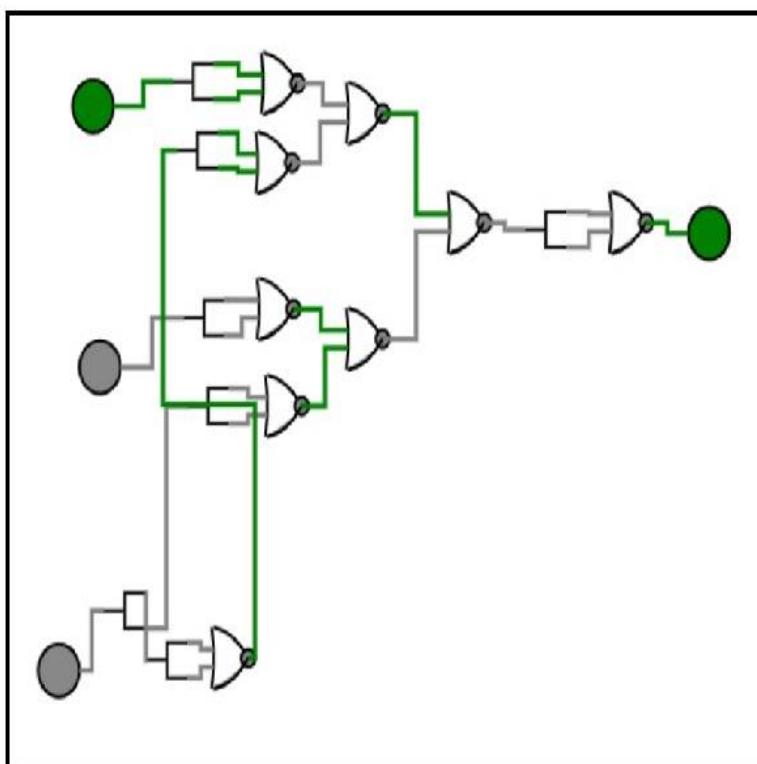
b. Formulate the expression for output Y by considering only those FPs for which the output is 1. $Y = S0' \cdot D0 + S0 \cdot D1$ The simplified function can be tabulated as:

Select Input S0	Output Y
0	$Y = D0$
1	$Y = D1$

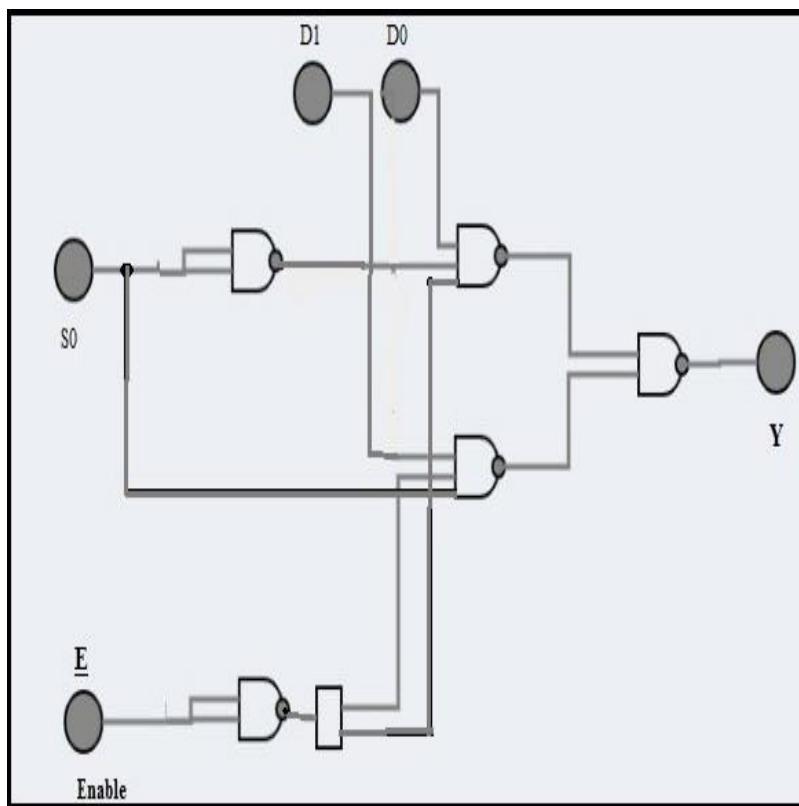
c. Draw the logic diagram for the expression using only NAND gates:



Draw the logic diagram for the expression using only NOR gates:



d. Using simulator construct the circuit and verify its operation. e. The same circuit can also be designed by having an active low enable input as shown below (Example for NAND gate Mux with enable):



The simplified function with active low enable input can be tabulated as:

Select Input S0	Enable input <u>E</u>	Output Y
X	1	0
0	0	$Y = D0$
1	0	$Y = D1$

PRETEST:

DE Journal Virtual Labs

dld-iitb.vlabs.ac.in/exp/multiplexer-using-universal-logic-gates/pretest.html

App Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Pretest

Procedure Simulation Posttest References Feedback

A multiplexer using basic gates is shown in the figure given image002. If $A = B = 11$ and $D_0 D_1 D_2 D_3 = 1001$, then the output of the circuit is:

image002

a : zero
 b : one

What if the select inputs in image002 are set to $A = B = 01$ and the data inputs are set to $D_0 D_1 D_2 D_3 = 0100$? The output Y is:

a : zero
 b : one

Type here to search

DE Journal Virtual Labs

dld-iitb.vlabs.ac.in/exp/multiplexer-using-universal-logic-gates/pretest.html

App Gmail YouTube Maps

30°C Haze 12:19 05-01-2022

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

image002

a : zero
 b : one

What if the select inputs in image002 are set to $A = B = 01$ and the data inputs are set to $D_0 D_1 D_2 D_3 = 0100$? The output Y is:

a : zero
 b : one

How many select lines are required for a 4:1 Mux?

a : Two
 b : Four
 c : Five
 d : Eight

Any multiplexer can be designed using universal logic gates. State True or False.

a : True
 b : False

Submit Quiz

4 out of 4

Type here to search

PROCEDURE:

Simulation Screen 1: Option - To select universal gate to build 2:1 multiplexer

1. Choose between the NAND and NOR gate for the experiment.
2. Click on the gate of your choice.
3. Click on "NEXT STAGE".

Simulation Screen 2: To fill the Truth table for chosen gate

1. Select appropriate options from the drop down menu and fill the truth table.
2. Click submit.
3. If incorrect try again.

4. If correct, proceed to the next step. Click Next.

Simulation Screen 3: Simulator Scene - To design, construct and verify the operation of a 2:1 multiplexer using universal logic gate.

Assume the enable input is activated (i.e. it is already set to zero)

General instructions:

1. Click on the digital switch and click on the canvas.
2. Repeat step 1 for three inputs.
3. Click on the NAND/NOR symbol and then on the canvas at an appropriate position on the canvas.
4. Repeat for the placement of all NAND/NOR gates.
5. Place the Y connector on canvas wherever one to many connections are expected.
6. Use the connection tabs and make the necessary connections.
7. Connect the outputs to the LED.
8. Simulate and verify the circuit.

Once the simulation is completed to satisfaction, reload the page and follow same instructions for the other universal gate.

STIMULATION:

The screenshot shows a web browser window with the title "Virtual Labs" and the URL "ddi-iitb.vlabs.ac.in/exp/multiplexer-using-universal-logic-gates/simulation.html". The main content area is titled "Multiplexer using Universal Gates." It displays a logic circuit diagram with two inputs and one output. Below the diagram are icons for NOR and NAND gates. A green button labeled "NEXT STAGE" is visible at the bottom.



DE Journal × Virtual Labs × G multiplexer using nor gates - Go | +

ddl-iitb.vlabs.ac.in/exp/multiplexer-using-universal-logic-gates/simulation.html

Apps Gmail YouTube Maps

Multiplexer using Universal Gates.

Enter the outputs for NOR Gate:

INPUTS		OUTPUT
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

SUBMIT Try Again

Correct answer.Your score is 5

NEXT

Type here to search

DE Journal × Virtual Labs × G 2 to 1 multiplexer using nor gates - Go | +

27°C Haze ENG 18:19 07-01-2022

Available Gates

- NOR
- Y-connector
- Green LED
- Digital Switch

Connections

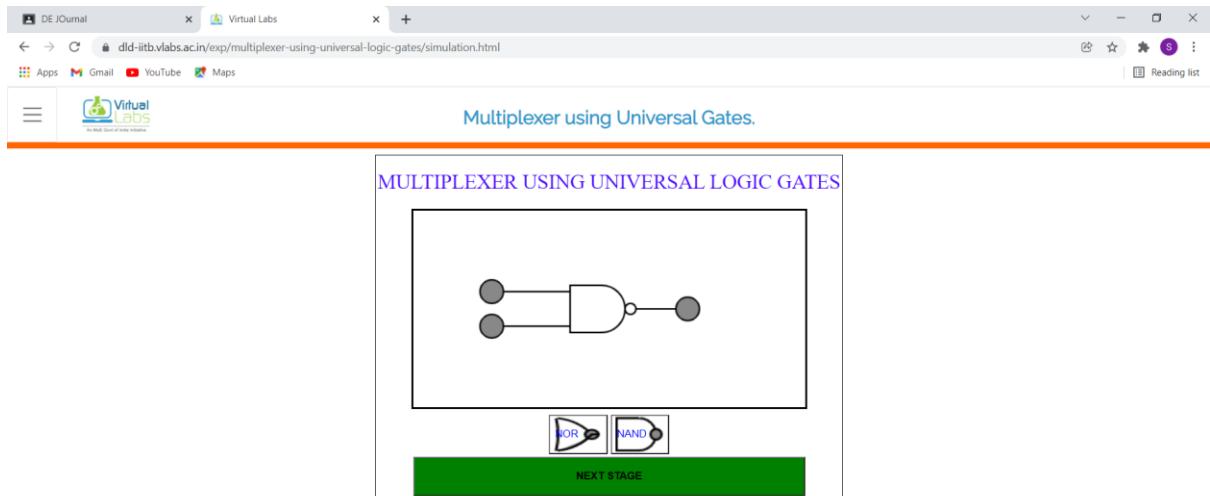
- Connect op1 to ip1
- Connect op2 to ip2
- Connect op2 to ip1
- Connect op1 to ip2
- UNDO GATE
- UNDO LINK
- REDO GATE
- REDO LINK

Simulator Started

CONTROL I/Ps

Enable I/P	Select I/P	O/P
E'	S ₀	Y
0	0	D ₀
0	1	D ₁





Type here to search

DE Journal Virtual Labs

27°C Haze 1851 07-01-2022

Apps Gmail YouTube Maps

Reading list

Multiplexer using Universal Gates.

Enter the outputs for NAND Gate:

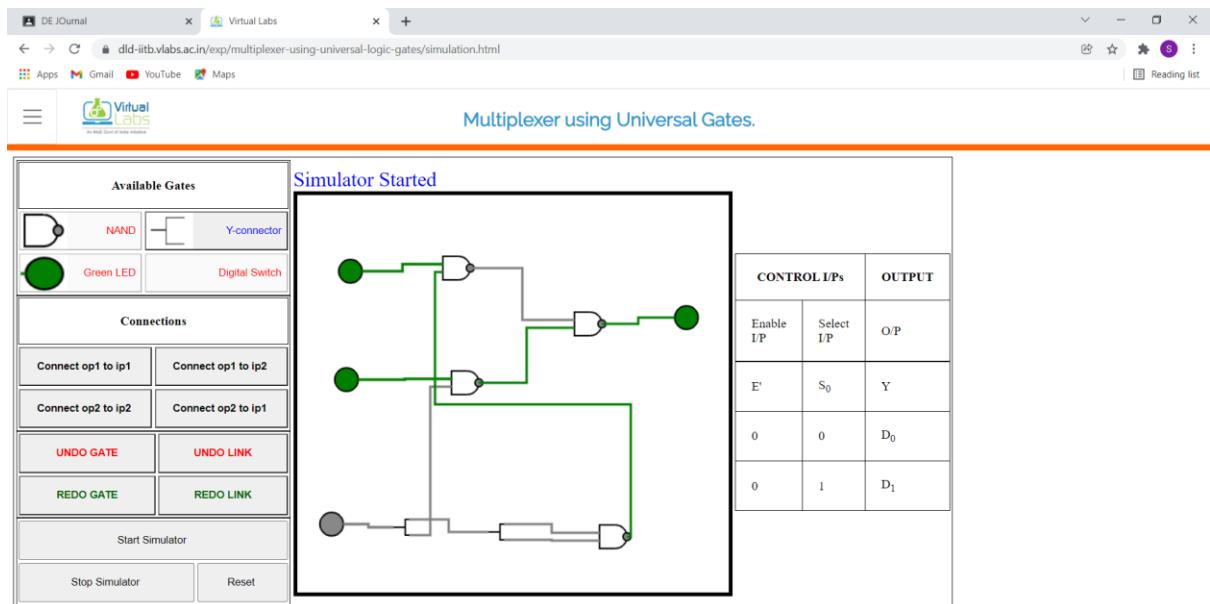
INPUTS		OUTPUT
A	B	Y
0	0	<input type="text" value="1"/>
0	1	<input type="text" value="1"/>
1	0	<input type="text" value="1"/>
1	1	<input type="text" value="0"/>

SUBMIT Try Again

Correct answer. Your score is 5

NEXT





POSTTEST:

DE Journal Virtual Labs

dld-iitb.vlabs.ac.in/exp/multiplexer-using-universal-logic-gates/posttest.html

Virtual Labs An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Aim: Multiplexer using Universal Gates.

Theory: A 2:1 Mux using basic logic gates can be converted into a 2:1 Mux using only NAND gates, if:

- a : AND gates are replaced by NAND gates.
- b : OR gate is replaced by NAND gate.
- c : Both AND & OR gates are replaced by NAND gates.
- d : AND & OR gates replaced by NAND gates & NOT gate replaced by a NAND gate with both inputs tied together as single input.

Procedure: To build a 2:1 using NAND gates (without enable), how many NAND gates are required?

- a : 3
- b : 4
- c : 5
- d : 6

Simulation: To build a 2:1 using NOR gates (without enable), how many NOR gates are required?

- a : 5
- b : 7
- c : 9

Feedback: 30°C Haze 12:21 05-01-2022



DE Journal Virtual Labs

ddl-iitb.vlabs.ac.in/exp/multiplexer-using-universal-logic-gates/posttest.html

App Gmail YouTube Maps

Reading list

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Posttest

To build a 2:1 using NAND gates (without enable), how many NAND gates are required?

a : 3
 b : 4
 c : 5
 d : 6

To build a 2:1 using NOR gates (without enable), how many NOR gates are required?

a : 5
 b : 7
 c : 9
 d : 11

Design, construct and verify a 2:1 Mux using NAND gates only. Get the simulation results evaluated from your faculty.

Implement 2:1 mux using only NOR gates. Use simulator to verify the result.

Submit Quiz

4 out of 5

Type here to search

30°C Haze 12:21 05-01-2022

CONCLUSION:

Thus, 2:1 multiplexer was designed, constructed and verified by using universal logic gates.

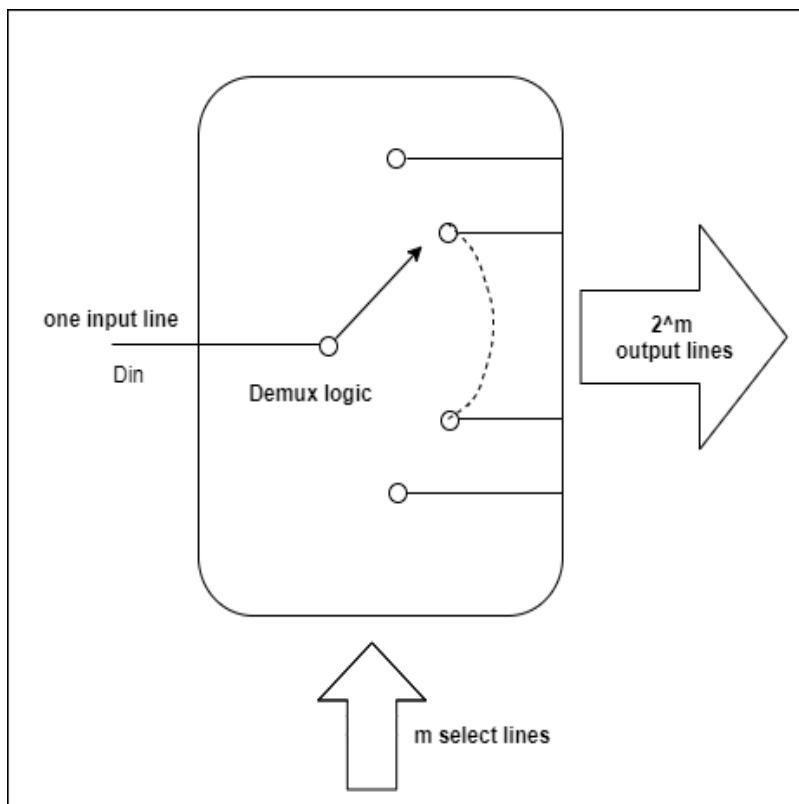
C] Demultiplexer using basic logic gates.

AIM: The students will be able to understand the concept of demultiplexers/decoders and its implementation using logic gates. Students will be able:

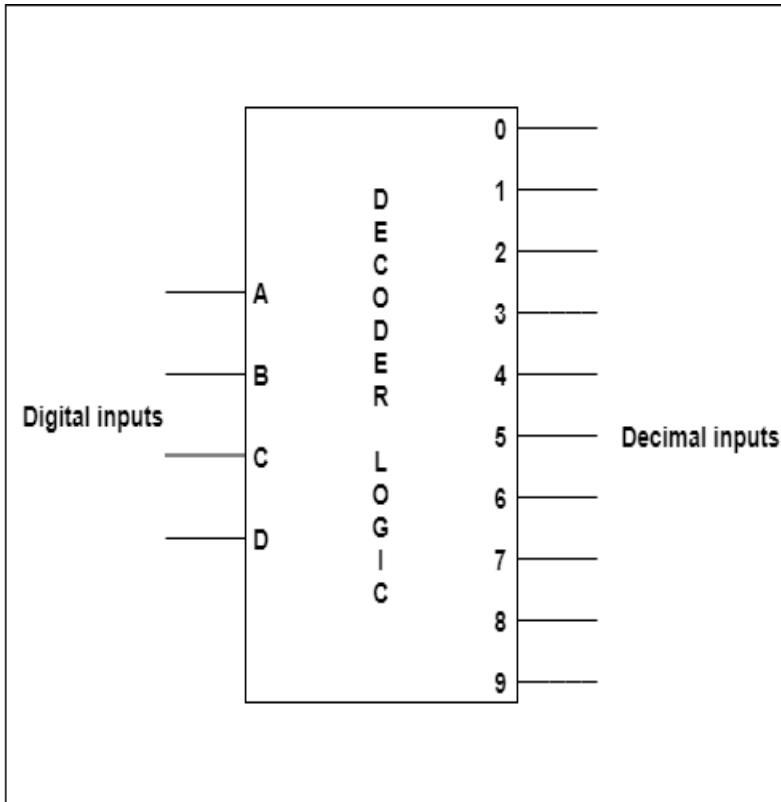
1. To identify the role of a demultiplexer circuit.
2. To design, construct and verify the operation of a 1:2 demultiplexer using basic logic gates.

THEORY: 1.1. Introduction

In certain applications, there is a need to connect the output of a system to a number of different destinations, but one at a time. In such cases, we need a digital combinational circuit called a demultiplexer. A demultiplexer is a logic circuit with one input and multiple outputs. It connects a single data input line to one of the 2^m output lines based on the status of the (m) select lines as shown in the figure.



On the other hand, a decoder is a logic circuit that accepts a 4-bit binary number and indicates its decimal equivalent(between 0 & 9) at the output as shown in the figure.



Commercially, demultiplexers ICs(integrated circuits) are available in form 1-to-2, 1-to-4, 1-to-8 or 1-to-16 where the number of select lines is 1,2,3,4 respectively. The same ICs, when used as decoders, are referred to as 1:2, 2:4, 3:8, 4:16(m:2^m). The demux/decoder circuits can be designed easily using gates-small scale integration SSI ICs for a small number of select lines. If the size of the demux/decoder increases, using medium-scale integration (MSI) ICs is the right choice. TTL IC 74154 is a 1-to-16 demultiplexer IC. In this experiment, we focus our attention on designing demux using gates.

1.2. Design of 1-to-2 demux using basic logic gates

A simple 1-to-2 demux will have single data input line D_{in} and one select line A and two outputs Y_0 & Y_1 . The select line can take a value either 0 or 1:

- If A takes a value 0, the input D_{in} is passed to the selected output Y_0 . i.e $Y_0 = D_{in}$.
- If A takes a value 1, the input D_{in} is connected to the selected output Y_1 . i.e. $Y_1 = D_{in}$.

Select I/p	Data Input	Fundamental Product FP	Output Y
A	D_{in}		
0	0	$A'.D_{in}'$	0
0	1	$A'.D_{in}'$	1
1	0	$A.D_{in}'$	0
1	1	$A.D_{in}'$	1

Function table

The expressions for outputs Y_0 & Y_1 can be formulated by considering only those FPs for which the output is 1.

Y_0

=

A' .

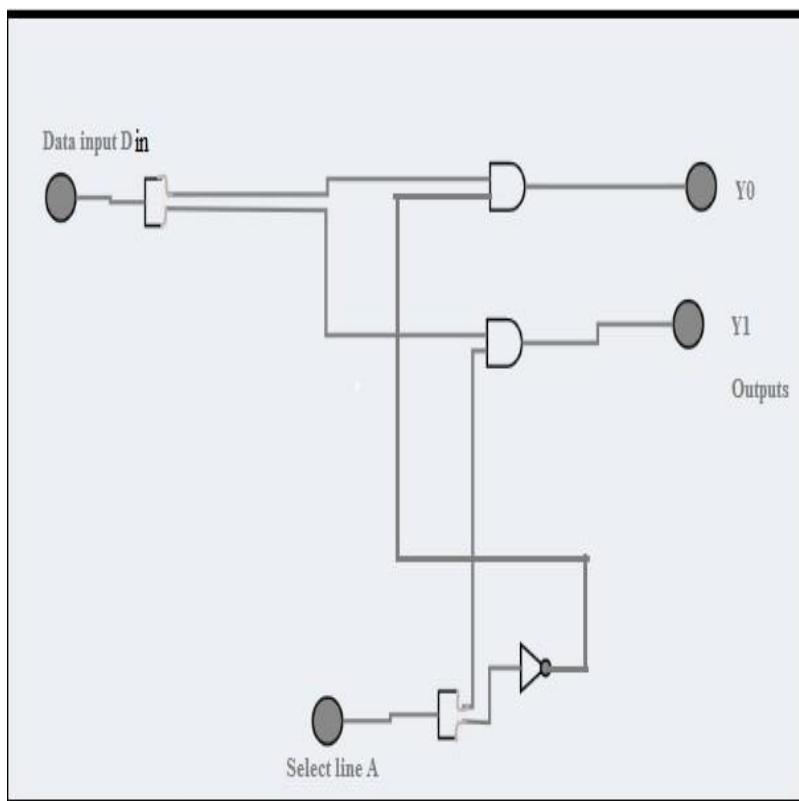
D_{in}

$Y_1 = A \cdot D_{in}$

The simplified function can be tabulated as:

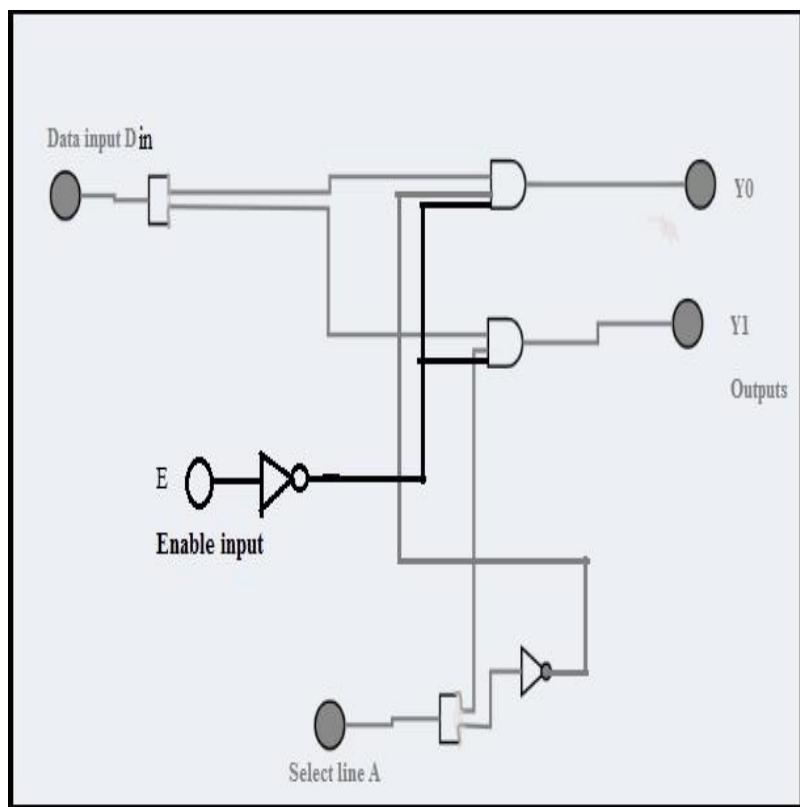
Select Input	Outputs	
A	Y_0	Y_1
0	$Y_0 = D_{in}$	0
1	0	$Y_1 = D_{in}$

Using basic logic gates a 1:2 demux can be constructed as shown in the figure.



An active low enable input may also be applied to the circuit to enable/disable it. The simplified function can be tabulated as:

Select I nput	Enable I/P	Outputs	
A	<u>E</u>	Y0	Y1
X	1	0	0
0	0	Y0 =din	0
1	0	0	Y1 = Din



PRETEST:

The screenshot shows a Microsoft Edge browser window with the URL dld-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/pretest.html. The page is titled "Demultiplexer using basic logic gates". On the left, there is a sidebar with navigation links: Aim, Theory, Pretest (which is currently selected), Procedure, Simulation, Posttest, References, and Feedback. The main content area contains several multiple-choice questions:

- A demultiplexer is a _____.
 - a : data distributor/multi-position switch.
 - b : combinational logic circuit.
 - c : design alternative to sum-of-product solutions.
 - d : All of these
- How many select lines are required for a 1:16 demux?
 - a : Two
 - b : Four
 - c : Five
 - d : Eight
- Any n-variable truth table can be implemented using a universal logic circuit referred to a demultiplexer. State True or False.
 - a : True
 - b : False

Below the questions, there is a "FEEDBACK" section with the following text and options:

O c : Five
O d : Eight

Any n-variable truth table can be implemented using a universal logic circuit referred to a demultiplexer. State True or False.
 a : True
 b : False

The enable input in a demultiplexer is also referred to as_____.

a : decode input
 b : strobe input
 c : Select input
 d : All of these

Decoders can be referred to as:
 a : 3:8 convertor
 b : 1 of 8 demux
 c : binary to octal decoder
 d : All of these

Submit Quiz
5 out of 5

PROCEDURE:

Simulation Screen 1: Quiz - To select appropriate gates to build a demultiplexer circuit.

1. Click on "Start Quiz".
2. Click on the appropriate gate.
3. Click on "Check answer".
4. If answer is correct, click "Go to next question".
5. If answer is incorrect, "Wrong gate selected" pops up, click ok to try again and proceed.
6. Repeat steps 2 to 5 and complete the quiz.

7. In first attempt if all 3 questions are answered correctly, you get an overall score of 5 points. For any incorrect answers the points are deducted.
9. For overall score lesser than 2 , you get a prompt "Please try the **THEORY** again". Click ok. Click on **THEORY** tab and revise the concepts and repeat steps 1 to 8.
10. In case of score greater than equal to 2, proceed to click on "Next stage".

Simulation Screen 2: Water tank - To identify the role of a demultiplexer circuit.

1. For the given question, click on select inputs S2, S1 and S0 appropriately.
2. Click on submit.
3. For incorrect answer, click ok to try again.
4. If answer is correct, Bingo! Click ok. Repeat steps 1 to 4 for next question.
5. Select appropriate answer from the drop down box for the next 2 questions.
6. For incorrect answer, click ok to try again.
7. If answer is correct, Bingo! Click ok. Repeat steps 5 to 7 for next question.
8. Click on Next.

Simulation Screen 3: To formulate the Function table and Boolean Expression for 1:2 demultiplexer

1. Select appropriate options from the drop down menu and formulate the function table.
2. Click submit.
3. If incorrect try again.
4. If correct, select the correct expression defining the demultiplexer.
5. Try again if answer is incorrect.
6. Proceed to the next step. Click Next.

Simulation Screen 4: Simulator Scene - To design, construct and verify the operation of a 1:2 demultiplexer using basic logic gates.

Assume the enable input is activated (i.e. it is already set to zero)

General instructions:

1. Apply the function table/Boolean expression in Screen 3 to design a 1:2 demux.
2. Click on the required object (AND, OR, NOT gate, digital switch, LED, connection wires) and click on canvas to place the component on canvas one by one.
3. Every component is treated as an object with one/two inputs and one output.
4. Digital switch output is connected as input. Click on 'output 1 to input 1' wire to connect the output to the first input of next gate, whereas 'output 1 to input 2' wire to connect the output to second input of gate.
5. Y connector has one input and two outputs. When same input is to be connected at two places, use a y connector accordingly.
6. Do connections as per the design.
7. Use undo link/undo gate or redo link/gate to make corrections if any.
8. Start simulator.
9. Set select input S0 and data inputs D0 & D1 and verify the corresponding outputs.

10. Compare the output of the design against the corresponding output in the function table of 1:2 demux given on right hand side and validate your design.

STIMULATION:

Virtual Labs DE Journal Virtual Labs

dd-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/simulation.html

Apps Gmail YouTube Maps

Demultiplexer using basic logic gates.

1. Which gate will you choose on input of 2:1 Demultiplexer

The diagram shows a 2:1 Demultiplexer circuit. It has two inputs, labeled A and B, each connected to one of the two inputs of an AND gate. The outputs of these two AND gates are connected to the two inputs of a third AND gate, which produces the output Y.

OR XOR AND NOR NOT NAND

Go to Next Question

Type here to search 23:07 06-01-2022

Virtual Labs DE Journal Virtual Labs

dd-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/simulation.html

Apps Gmail YouTube Maps

Demultiplexer using basic logic gates.

2. Which gate will you choose on select line of 2:1 Demultiplexer

The diagram shows a 2:1 Demultiplexer circuit. It has one input, labeled S, which is connected to the select line of an inverter. The output of the inverter is connected to one of the two inputs of an AND gate. The other input of this AND gate is connected to the data input D. The output of this AND gate is the output Y.

OR XOR AND NOR NOT NAND

Go to Next Question



Virtual Labs DE Journal Virtual Labs

dd-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/simulation.html

Virtual Labs

did-iitb.vlabs.ac.in says
Your score is : 4

OK

3. Which gate will you choose on output of 2:1 Demultiplexer

OR XOR AND NOR NOT NAND

Go to Next Question

Type here to search

Virtual Labs DE Journal Virtual Labs

dd-iitb.vlabs.ac.in says
BINGO!!!

OK

WATER TANK

Question No.1 : What will be the value of select inputs S2,S1,S0 to get water from Tap 2 ?

WATER TANK

Tap 0, Tap 1, Tap 2, Tap 3, Tap 4, Tap 5, Tap 6, Tap 7

CONTROL S2 S1 S0

Select Inputs: S₂ S₁ S₀

SUBMIT



Question No.2 : What will be the value of select inputs S2,S1,S0 to get water from Tap 0?

WATER TANK

Select Inputs: S₂ S₁ S₀

SUBMIT



Question No.3 : Which tap will you get water from, If the values of select inputs are S2=1 S1=0 S0=1 ?

WATER TANK

Select Tap:

SUBMIT



Virtual Labs x DE Journa x Virtual Labs x

ddl-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/simulation.html

OK

WATER TANK

Question No.4 : Which tap will you get water from, If the values of select inputs are S2=1 S1=1 S0=1 ?

Select Tap: Tap7

SUBMIT

Type here to search 25°C Haze 23:10 06-01-2022 ENG

DE Journa x Virtual Labs x G 2 to 1 multiplexer using nor gate x demultiplexer truth table - Google x

Reading list

Demultiplexer using basic logic gates.

Q1. Enter the corresponding outputs for the given 1:2 DEMUX

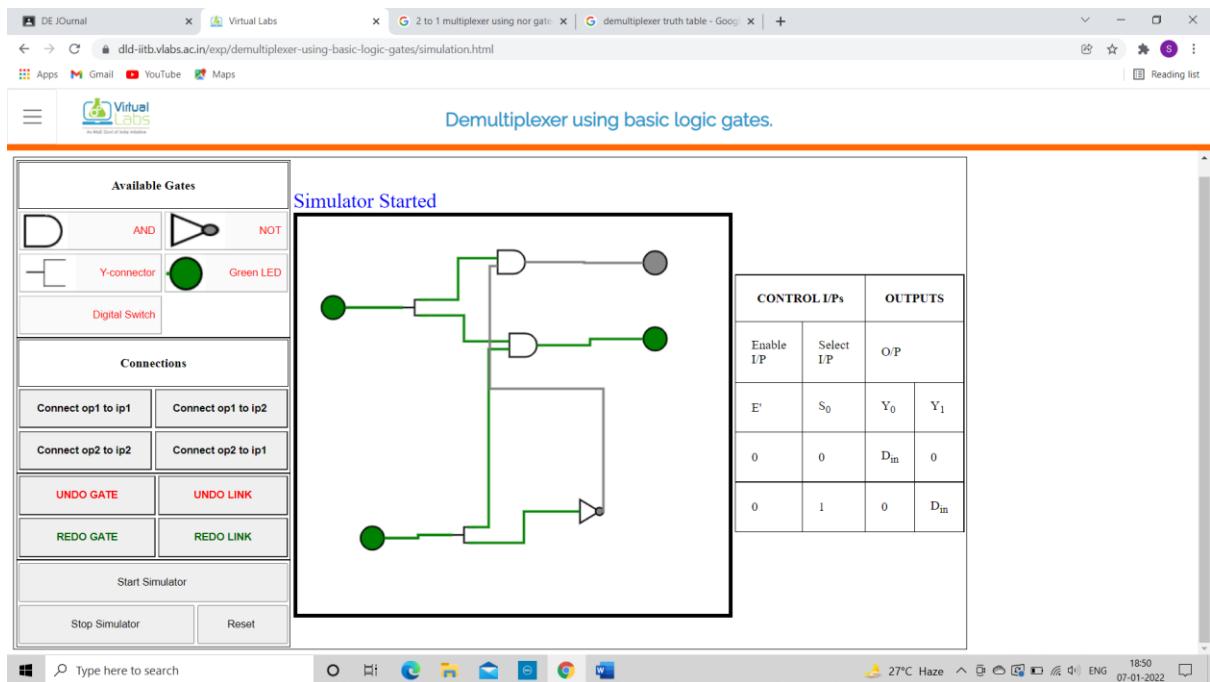
CONTROL I/Ps		OUTPUTS	
Enable I/P	Select I/P	O/P	
E	S ₀	Y ₀	Y ₁
1	X	<input type="text"/> 0	<input type="text"/> 0
0	0	<input type="text"/> Din	<input type="text"/> 0
0	1	<input type="text"/> 0	<input type="text"/> Din

Q2. Select the correct Boolean expression for the above truth table.

Y₀ = Din S₀, Y₁ = Din S₀

SCOREKEEPER

Question	Status
Q1.	Correct answer. Your score is 5
Q2.	Correct answer. Your score is 2



POSTTEST:

DE Journal Virtual Labs Google Search

Virtual Labs An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Logic Design (Logic Gates & Mux-Demux) > Experiments

Aim Theory Pretest Procedure Simulation Posttest References Feedback

Demultiplexer using basic logic gates.

To implement multiple output functions (sum of product type-SOP) using demultiplexer, additional _____type of gates are required.

a : AND
 b : OR
 c : NOT
 d : EX-OR

Design, construct and verify a 1:2 demux using NOR gates only. Get the simulation results evaluated from your faculty.

Can 1:4 demux chips be cascaded to build 1:8 demux?

a : Yes
 b : No

To implement a full adder, which of the following demux would be preferable?

a : 1:4
 b : 1:8
 c : 1:16
 d : 1:128

Windows Taskbar: Type here to search, Start, Task View, Edge, Mail, File Explorer, Google Chrome, Microsoft Word, 30°C AQI 225, ENG, 12:24, 05-01-2022

DE JOURNAL Virtual Labs

dld-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/posttest.html

App Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

b : No

To implement a full adder, which of the following demux would be preferable?

a : 1:4
 b : 1:8
 c : 1:16
 d : 1:32

The figure shown below represents_____.

a : half adder

Type here to search

DE JOURNAL Virtual Labs

dld-iitb.vlabs.ac.in/exp/demultiplexer-using-basic-logic-gates/posttest.html

App Gmail YouTube Maps

30°C AQI 225 12:24 05-01-2022

HOME PARTNERS CONTACT

The figure shown below represents_____.

a : half adder
 b : full adder
 c : half subtractor
 d : full subtractor

Submit Quiz

5 out of 5

Type here to search

30°C AQI 225 12:24 05-01-2022

CONCLUSION:

Thus, 1:2 demultiplexer was designed, constructed and verified by using basic logic gates.

D] Demultiplexer using universal logic gates

AIM: The students will be able to understand the concept of demultiplexers/decoders and its implementation using universal logic gates. Students will be able:

1. To identify the role of a decoder.
2. To design, construct and verify the operation of a 1:2 demultiplexer using universal logic gates.

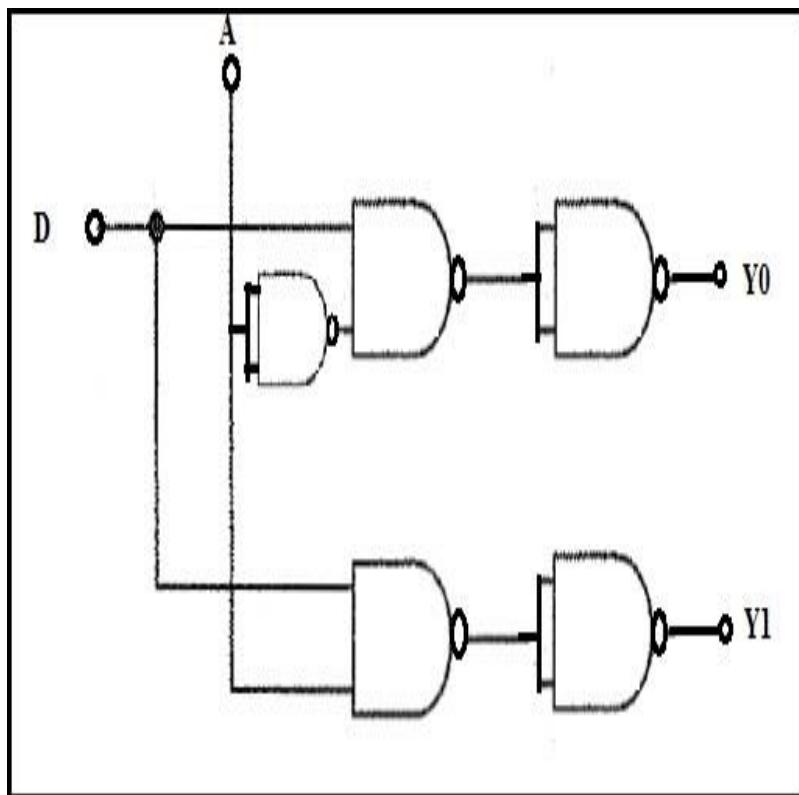
THEORY: 1.1. Introduction

A demultiplexer is a logic circuit with one input and multiple outputs. Whereas a 2:4 decoder is a logic circuit that accepts a 2-bit binary number and indicates its decimal equivalent (between 0 & 3) at the output. TTL ICs such as 74LS155, 74LS138 and 74LS154 are 2:4, 3:8 and 4:16 type decoder ICs respectively. They are also referred to as 1-to-4, 1-to-8 and 1-to-16 type demultiplexers. In case of a decoder there is no data input. The select lines/control bits are the only inputs. IC 74LS154 can be used as a decoder-demultiplexer IC as it performs both operations.

1.2 Design a 1:2 demux using basic universal logic gates. NAND & NOR gates are called as universal logic gates. The 1:2 demux logic can be implemented using only NAND gates.

CONCEPT:

Whenever both the inputs of NAND gate are tied together as single input, it works as a NOT gate. Whereas a NAND gate followed by a NOT gate gives an AND logic. Using these concepts a 1:2 demux can be designed as shown in figure



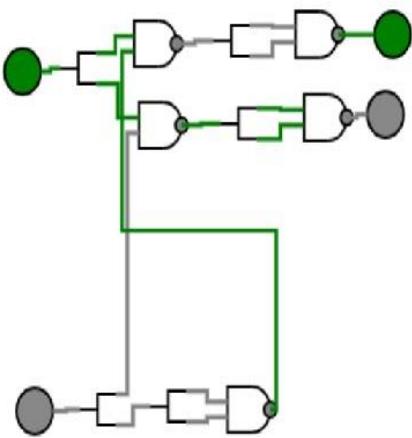
The function table of a 1:2 demux is tabulated. Function table

Select I/p	Data Input	Fundamental Product FP	Output Y	Output Y (in terms of input)
A	Din			
0	0	$A' \cdot \text{Din}'$	0	$Y_0 = \text{Din}$
0	1	$A' \cdot \text{Din}$	1	
1	0	$A \cdot \text{Din}'$	0	$Y_1 = \text{Din}$
1	1	$A \cdot \text{Din}$	1	

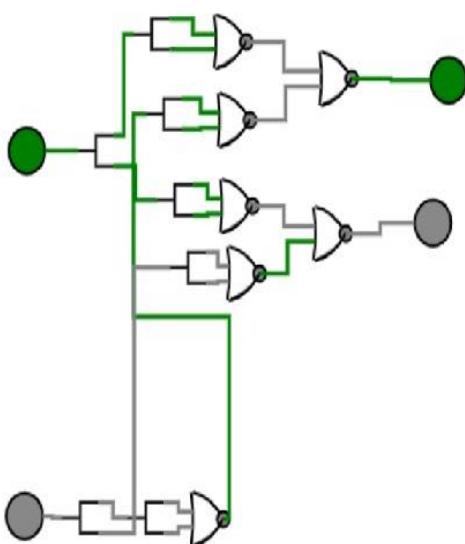
The expressions for outputs Y_0 & Y_1 can be formulated by considering only those FPs for which the output is 1. $Y_0 = A' \cdot \text{Din}$ $Y_1 = A \cdot \text{Din}$ The simplified function can be tabulated as:

Select Input	Outputs	
A	Y_0	Y_1
0	$Y_0 = \text{Din}$	0
1	0	$Y_1 = \text{Din}$

Using universal logic gates a 1:2 demux can be constructed as shown in the figures
1.Using NAND gates



2.Using NOR gates



PRETEST:

Theory
Pretest
Procedure
Simulation
Posttest
References
Feedback

1. In the figure shown in image002, a _____ is shown using basic logic gates.

Image002

a : 1:2 mux
 b : 2:1 mux
 c : 2:1 demux
 d : 1:2 demux

In figure image002, when control input A = 0 and D=1 ?

a : $Y_0 = 0, Y_1 = 0$
 b : $Y_0 = 1, Y_1 = 0$
 c : $Y_0 = 0, Y_1 = 1$

In figure image002, when control input A = 0 and D=0 ?

a : $Y_0 = 0, Y_1 = 0$
 b : $Y_0 = 0, Y_1 = 1$
 c : $Y_0 = 0, Y_1 = 0$
 d : $Y_0 = 1, Y_1 = 0$

In figure image002, what is the output if A = 1 and D = 0?

a : $Y_0 = 0, Y_1 = 1$
 b : $Y_0 = 1, Y_1 = 1$
 c : $Y_0 = 0, Y_1 = 0$
 d : $Y_0 = 1, Y_1 = 0$

In figure image002, what is the output if A = 1 and D = 1?

a : $Y_0 = 0, Y_1 = 1$
 b : $Y_0 = 1, Y_1 = 0$
 c : $Y_0 = 0, Y_1 = 0$
 d : $Y_0 = 1, Y_1 = 1$

Submit Quiz

5 out of 5

PROCEDURE:

Simulation Screen 1: Option - To select universal gate to build 1:2 demultiplexer

1. Choose between the NAND and NOR gate for the experiment.
2. Click on the gate of your choice.
3. Click on "NEXT STAGE".

Simulation Screen 2: To fill the Truth table for chosen gate

1. Select appropriate options from the drop down menu and fill the truth table.
2. Click submit.

3. If incorrect try again.

4. If correct, proceed to the next step. Click Next.

Simulation Screen 3: Simulator Scene - To design, construct and verify the operation of a 2:1 multiplexer using universal logic gate.

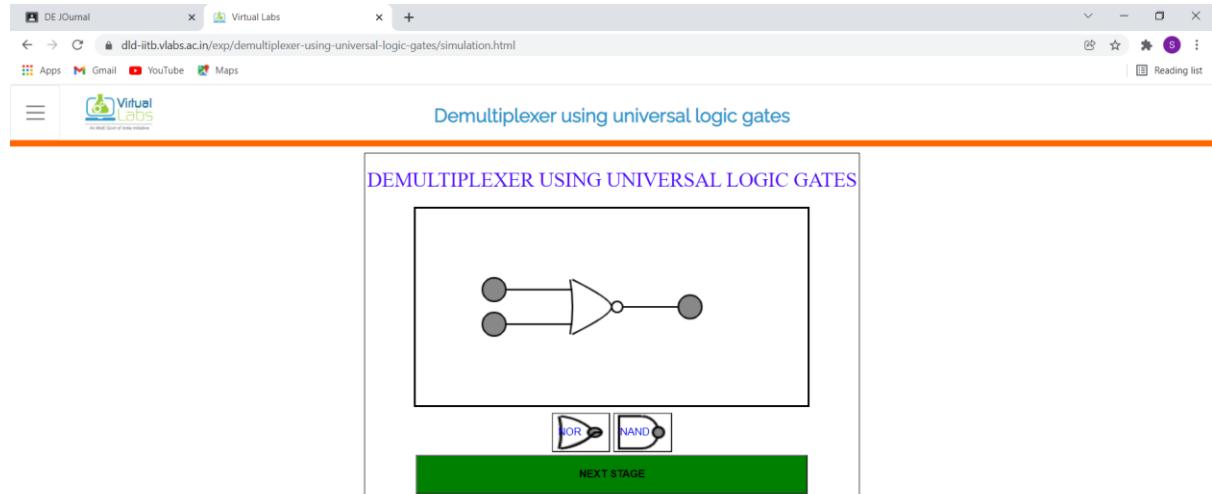
Assume the enable input is activated (i.e. it is already set to zero)

General instructions:

1. Click on the digital switch and click on the canvas.
2. Repeat step 1 for three inputs.
3. Click on the NAND/NOR symbol and then on the canvas at an appropriate position on the canvas.
4. Repeat for the placement of all NAND/NOR gates.
5. Place the Y connector on canvas wherever one to many connections are expected.
6. Use the connection tabs and make the necessary connections.
7. Connect the outputs to the LED.
8. Simulate and verify the circuit.

Once the simulation is completed to satisfaction, reload the page and follow same instructions for the other universal gate.

STIMULATION:



DE Journal Virtual Labs Google 2:1 demultiplexer using nor gate

Virtual Labs

Enter the outputs for NOR Gate:

INPUTS		OUTPUT
A	B	Y
0	0	<input type="text" value="1"/>
0	1	<input type="text" value="0"/>
1	0	<input type="text" value="0"/>
1	1	<input type="text" value="0"/>

SUBMIT Try Again(1)

Correct answer. Your score is 4

NEXT

Type here to search

DE Journal Virtual Labs Google 2:1 demultiplexer using universal logic gates

Available Gates

- NOR
- Y-connector
- Green LED
- Digital Switch

Connections

- Connect op1 to ip1
- Connect op2 to ip2
- Connect op2 to ip1
- Connect op1 to ip2
- UNDO GATE
- UNDO LINK
- REDO GATE
- REDO LINK

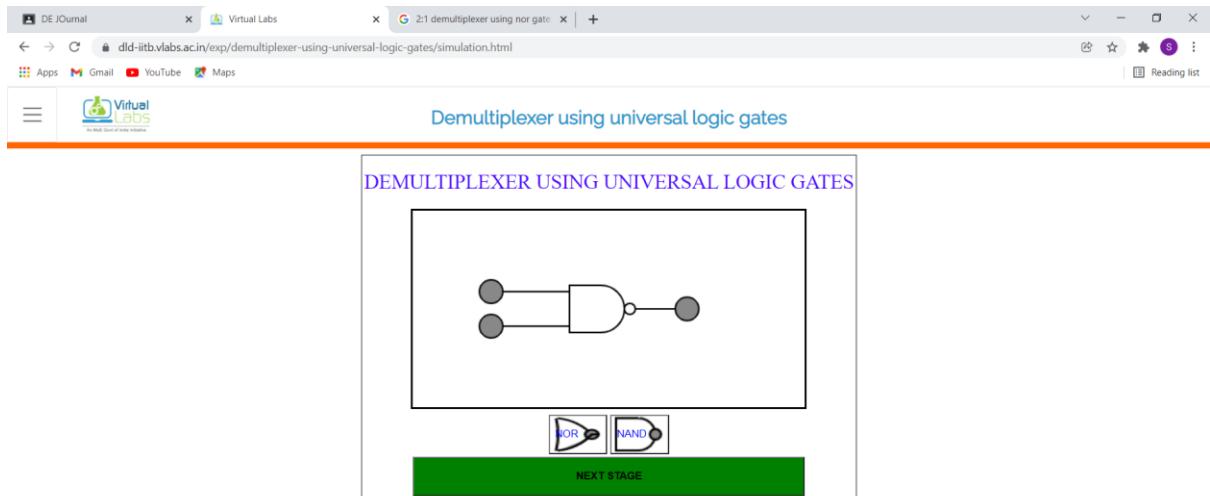
Start Simulator

Stop Simulator Reset

Simulator Started

CONTROL I/Ps		OUTPUTS	
Enable I/P	Select I/P	O/P	
E'	S_0	Y_0	Y_1
0	0	D_in	0
0	1	0	D_in





Type here to search

DE Journal Virtual Labs

27°C Haze ENG 19:07 07-01-2022

Apps Gmail YouTube Maps

Reading list

Demultiplexer using universal logic gates

Enter the outputs for NAND Gate:

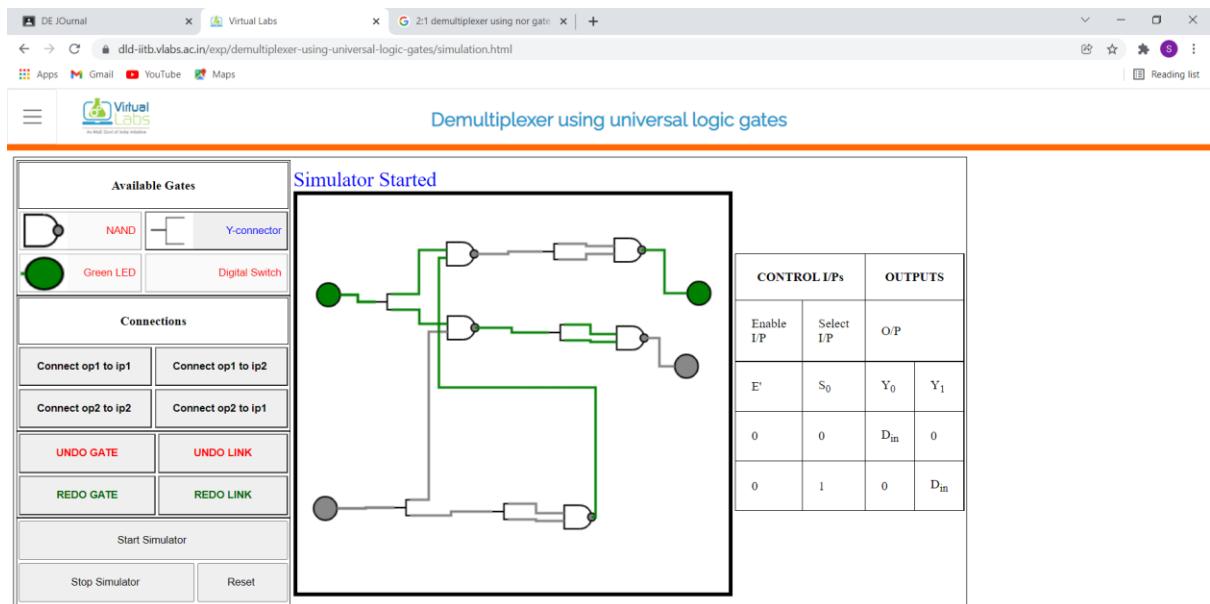
INPUTS		OUTPUT
A	B	Y
0	0	<input type="text" value="1"/>
0	1	<input type="text" value="1"/>
1	0	<input type="text" value="1"/>
1	1	<input type="text" value="0"/>

SUBMIT Try Again

Correct answer. Your score is 5

NEXT

25°C Haze ENG 23:23 06-01-2022



POSTTEST:

DE Journal x Virtual Labs x +

Pretest Procedure Simulation Posttest References Feedback

Posttest

Image003

○ a : 1:2 mux
○ b : 1:2 demux
c : 1:2 decoder
○ d : none of these

In figure image003, when control input A = 0,
 ○ a : Y0 = 0, Y1 = 0
b : Y0 = 0, Y1 = 1
 ○ c : Y0 = 1, Y1 = 0
 ○ d : Y0 = 1, Y1 = 1

HOME PARTNERS CONTACT

Type here to search 30°C Smoke 12:28
05-01-2022

DE Journal Virtual Labs

ddi-iitb.vlabs.ac.in/exp/demultiplexer-using-universal-logic-gates/posttest.html

App Gmail YouTube Maps

Reading list

Virtual Labs
An MoU Govt of India Initiative

HOME PARTNERS CONTACT

b : $Y_0 = 0, Y_1 = 1$
 c : $Y_0 = 1, Y_1 = 0$
 d : $Y_0 = 1, Y_1 = 1$

In figure image003, when control input A = 1,
 a : $Y_0 = 0, Y_1 = 0$
 b : $Y_0 = 0, Y_1 = 1$
 c : $Y_0 = 1, Y_1 = 0$
 d : $Y_0 = 1, Y_1 = 1$

In figure image003, what value of A will result in $Y_0 = Y_1 = 0$?
 a : A = 0
 b : A = 1
 c : A = x
 d : Not possible

Higher order demux/decoders can be built using lower order demux/decoders. State True or False.
 a : True
 b : False

Submit Quiz

5 out of 5

Type here to search

30°C Smoke 12:28 05-01-2022

CONCLUSION:

Thus, 1:2 demultiplexer was designed, constructed and verified by using universal logic gates.

E] Implementation of 8:1 MUX using MSI ICs

AIM: To design and implement 8:1 MUX using IC-74LS153 and Verify its Truth Table.

THEORY: A modern home stereo system may have a switch that selects music from one or four sources: a radio tuner, a cassette tape, a compact disc (CD) or an auxiliary input. The auxiliary input could be an audio from VCR or DVD or a smart phone. The switch selects input from one of these four sources and connects it to the power amplifier and speakers as shown in Fig.1. In simple terms, this is what a multiplexer (MUX) does. A MUX is a combinational logic block that selects one-of-N inputs and directs the information to a single output. It acts like a multi-position switch. Mux is a well-known MSI (medium-scale integration) IC.

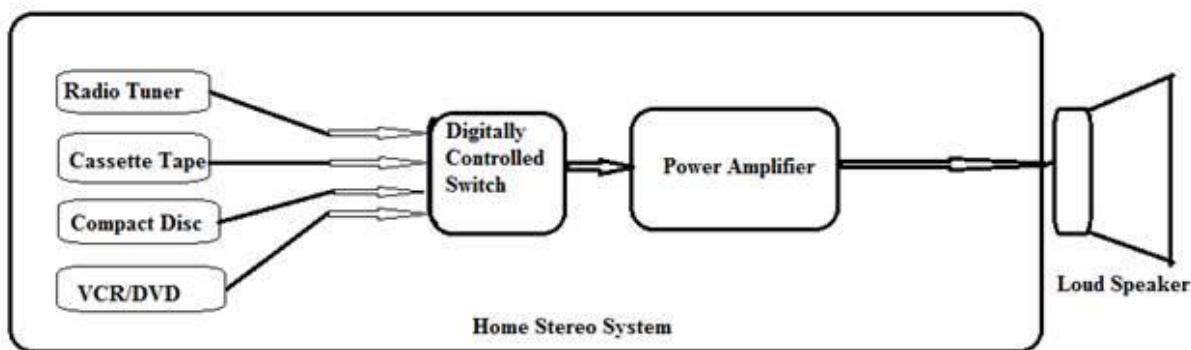


Fig.1. Block schematic: Home Stereo System

Multiplexer or Data Selector is a very widely used combinational circuit. It has multiple inputs and one output. It accepts several data inputs and allows only one of them at a time to get through to the output. The routing of the desired input to the output is controlled by the select lines. M select lines can select one of the 2^M input channels. The generalized block schematic of a multiplexer is shown in Fig. 2. Mechanical rotary switch is a good analogy to explain the MUX concept.

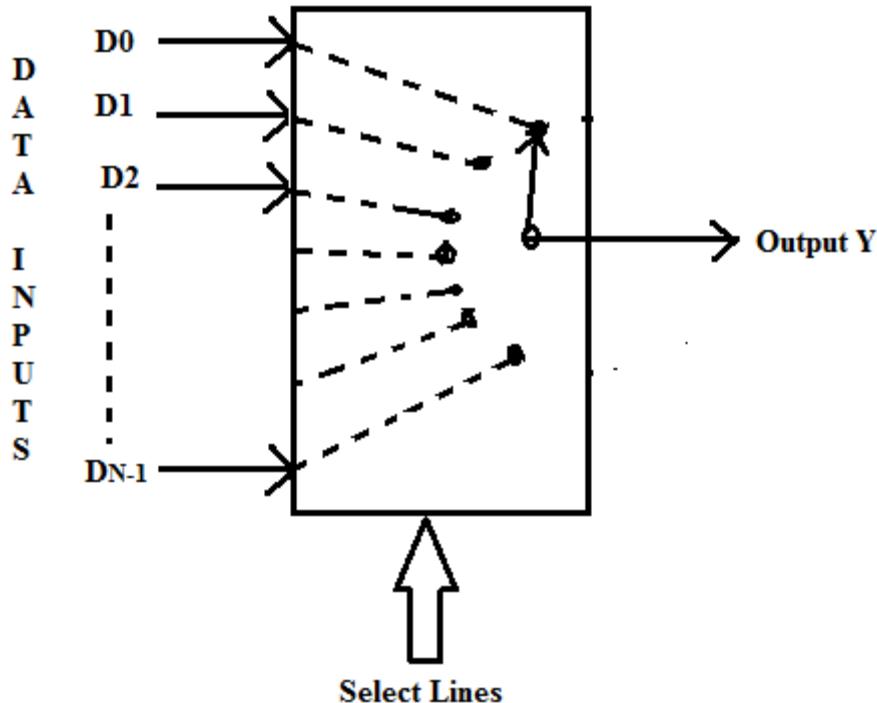


Fig.2. Multiplexer Schematic: As a digitally controlled multiposition switch

The multiplexer selects 1 out of N input data sources and sends selected data to single output channel. This Many as to One function is called as Multiplexing. A data selector is a sort of one-package-solution to a complicated logic problem. It consists of large number of gates packaged inside a single integrated circuit (IC). It can be considered to belong to medium scale integration

MSI

technology.

MUX Applications: Multiplexers basically can be used as universal logic elements. It provides a low-cost reliable and compact solution to many logic problems with three to five input variables.

The

MUX

applications

include:

1. Data Selection
2. Data Routing
3. Operation Sequencing
4. Parallel -to-Serial Conversion
5. Waveform Generation
6. Logic Function Generation.

Design:

The functional name of IC 74LS153 is Dual 4 line to 1 Line Data Selector/Multiplexer. It implies that there are two 4:1 Multiplexers (Mux) inside the IC. It is a 16 pin Dual-In-Line Package (DIP) IC. Every 4:1 Mux comprises of four input lines, two select lines, one active low strobe line and a single output line. An 8:1 Mux should have eight input lines and three select lines and single output. To design an 8:1 Mux using IC 74LS153, we need to obtain three select lines using Strobe input and the select lines B and A. The strobe input can be treated as the third select line C, which is directly connected to upper 4:1 Mux and through a NOT gate to the lower 4:1 Mux. This will ensure that when C = 0, upper Mux is enabled and depending on select inputs B and A, one of the inputs from D0 to D3 will be passed/switched to its output. Similarly when C = 1, the lower Mux is enabled and depending on select inputs B and A, one of the inputs from D4 to D7 will be passed/switched to its output. The outputs of the two Muxes then can be given to an OR gate to produce the single final output Y as shown in Fig.3. The function table of the 8:1 MUX is given in Table 1. The Boolean expression defining the

output is given by:
 $Y = C'.[B'.A'.D_0 + B'.A.D_1 + B.A'.D_2 + B.A.D_3] + C.[B'.A'.D_4 + B'.A.D_5 + B.A'.D_6 + B.A.D_7]$

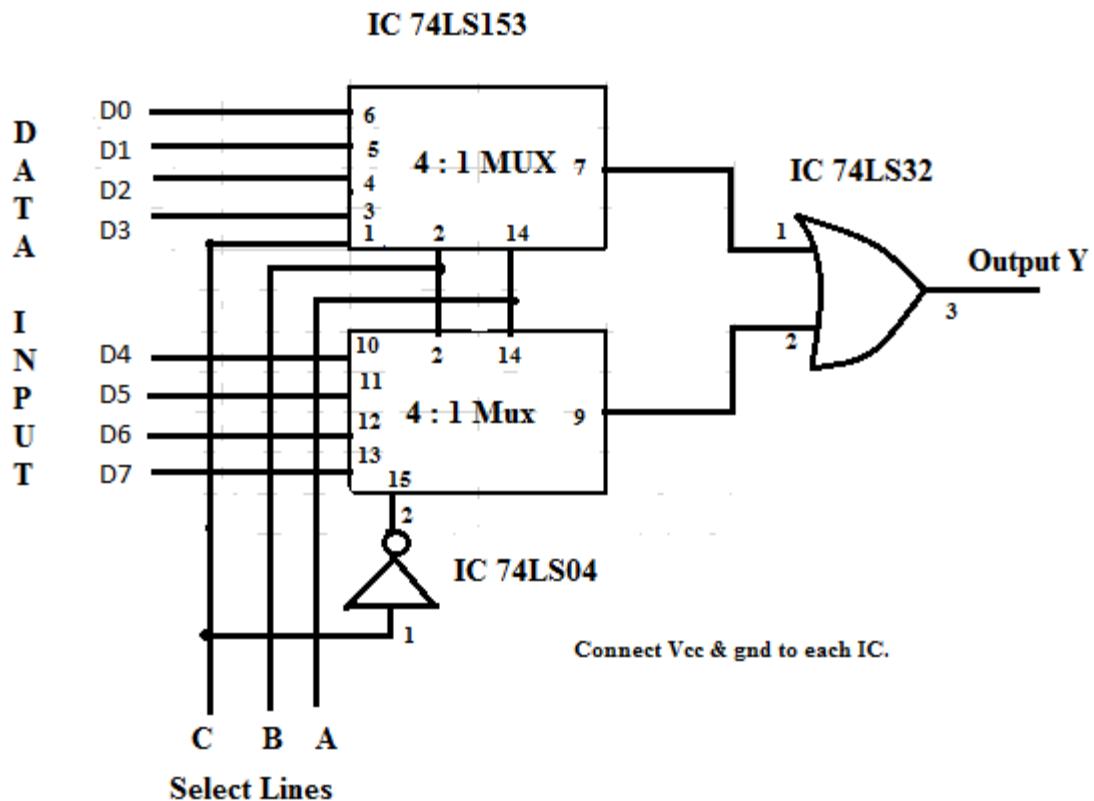


Fig. 3. Eight to One Multiplexer using IC 74LS153
The logic diagram, for 8:1 MUX in IC 74LS153 is:

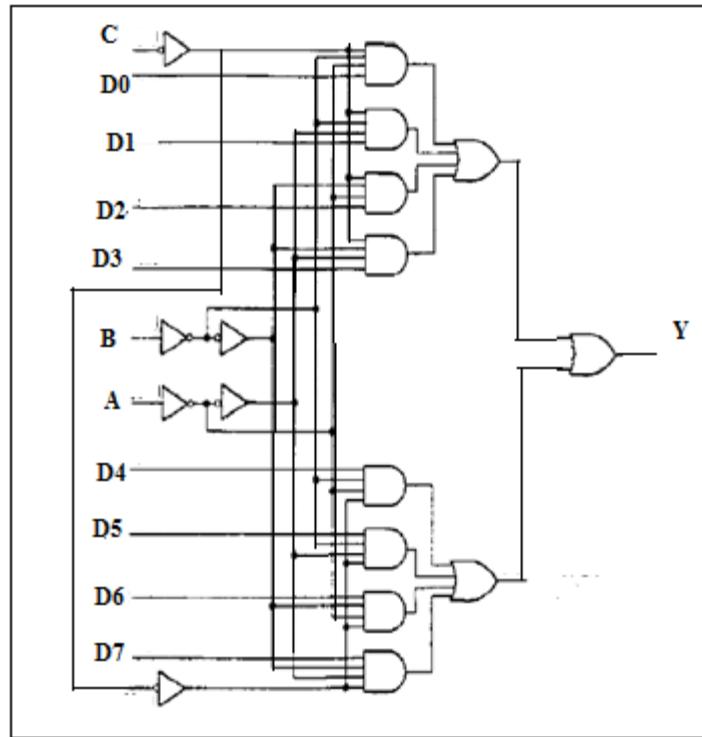


Fig. 4. Logic Diagram

Numerical:

For 8:1 MUX, there are 8 inputs and 1 output.

For those 8 inputs the 3 select lines will be needed. (Since $2^3 = 8$)

So the truth table of 8:1 MUX will be:

Select Lines			Inputs								Output	MUX selected
C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	Y	
0	0	0	0	X	X	X	X	X	X	X	0	Upper 4:1 MUX
0	0	0	1	X	X	X	X	X	X	X	1	
0	0	1	X	0	X	X	X	X	X	X	0	
0	0	1	X	1	X	X	X	X	X	X	1	
0	1	0	X	X	0	X	X	X	X	X	0	
0	1	0	X	X	1	X	X	X	X	X	1	
0	1	1	X	X	X	0	X	X	X	X	0	
1	0	0	X	X	X	X	0	X	X	X	0	

1	0	0	X	X	X	X	1	X	X	X	1	
1	0	1	X	X	X	X	X	0	X	X	0	
1	0	1	X	X	X	X	X	1	X	X	1	
1	1	0	X	X	X	X	X	X	0	X	0	
1	1	0	X	X	X	X	X	X	1	X	1	
1	1	1	X	X	X	X	X	X	X	0	0	
1	1	1	X	X	X	X	X	X	X	1	1	

Lower
4:1
MUX

(Where : ‘1’ indicate VCC/+5V, ‘0’ indicate 0V, ‘X’ indicate “don’t care“)

PRETEST:

The screenshot shows a web browser window with the following details:

- Address Bar:** dd-iitb.vlabs.ac.in/exp/implementation-multiplexer-using-msi/pretest.html
- Page Title:** Implementation of 8:1 MUX using MSI ICs
- Left Sidebar (AIMI):**
 - Theory
 - Pretest** (selected)
 - Procedure
 - Simulation
 - Posttest
 - References
 - Feedback
- Main Content Area:**

Figure out the number of inputs and outputs in 8:1 Multiplexer.

a : 1 input 8 outputs.
 b : 2 inputs 4 outputs.
 c : 8 inputs 1 output.
 d : 4 inputs 1 output.

Multiplexer has

a : Many input one output.
 b : Many input many output.
 c : One input many output.
 d : One input one output.

In 4:1 multiplexer state the number of select lines required?

a : 1
 b : 2.
 c : 3.
 d : 4

Select lines are used to select
- Bottom Status Bar:** Type here to search, system icons, and system status (30°C, 12:30, 05-01-2022).

In 4:1 multiplexer state the number of select lines required?
 a : 1.
 b : 2.
 c : 3.
 d : 4.

Select lines are used to select
 a : Input.
 b : Output.
 c : Both
 d : None of the above.

The abbreviation of multiplexer is _____.
 a : MPX.
 b : MUX.
 c : Both of the above.
 d : None of the above.

Submit Quiz

5 out of 5

PROCEDURE:

Steps:

1. Switch ON the circuit, by pressing Main Switch.
2. Set the appropriate Inputs.
3. After giving the inputs, observe the corresponding outputs.
4. Verify the results.

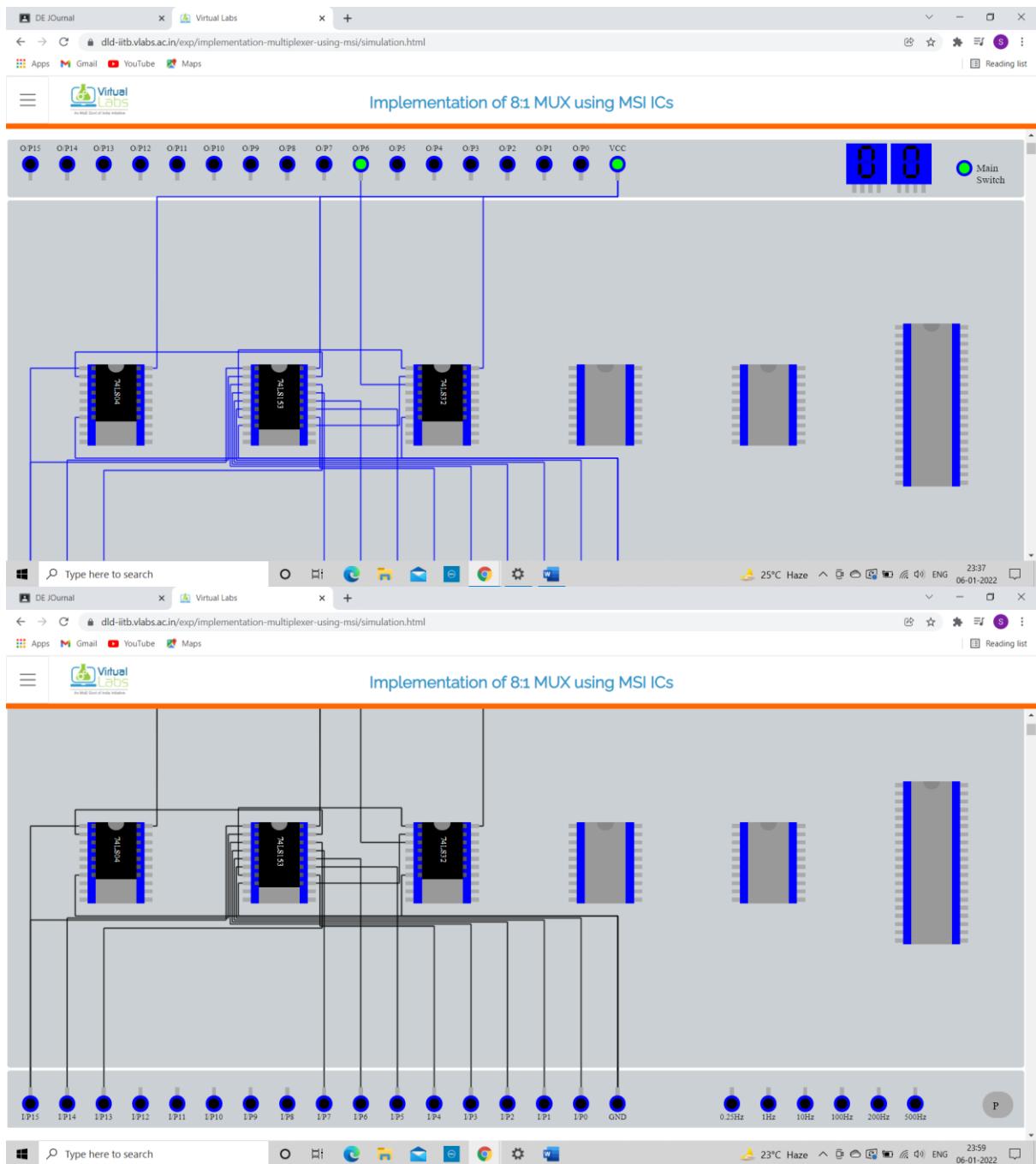
Inputs:

1. Select Line C : I/P15
2. Select Line B : I/P14
3. Select Line A : I/P13
4. Input Line D0 : I/P0
5. Input Line D1 : I/P1
6. Input Line D2 : I/P2
7. Input Line D3 : I/P3
8. Input Line D4 : I/P4
9. Input Line D5 : I/P5
10. Input Line D6 : I/P6
11. Input Line D7 : I/P7

Outputs:

1. Output (O/P6)

STIMULATION:



POSTTEST:

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** DE Journal - did-iitb.vlabs.ac.in/exp/implementation-multiplexer-using-msi/posttest.html
- Address Bar:** did-iitb.vlabs.ac.in/exp/implementation-multiplexer-using-msi/posttest.html
- Top Navigation:** HOME, PARTNERS, CONTACT
- Section Headers:** Theory, Pretest, Procedure, Simulation, Posttest, References, Feedback.
- Main Content:**
 - Posttest:** Which is the major functioning responsibility of the multiplexing combinational circuit?
 - a : Decoding the binary information.
 - b : Encoding the information.
 - c : Generation of selected path between multiple sources and a single destination.
 - d : All of the above.
 - Feedback:** If there are N numbers of select lines then state the numbers of input lines in the multiplexer?
 - a : 2^N
 - b : 2^N
 - c : $2/N$
 - d : $2N+1$
 - Posttest:** Which digital circuit acts as switch.?
 - a : Comparator
 - b : Counter
 - c : De-Multiplexer
 - d : Multiplexer
 - Feedback:** If select lines are $S_1 S_0 = 11$ then which of the input is selected by multiplexer?
 - a : D1
 - Posttest:** Which digital circuit acts as switch.?
 - a : Comparator
 - b : Counter
 - c : De-Multiplexer
 - d : Multiplexer
 - Feedback:** If select lines are $S_1 S_0 = 11$ then which of the input is selected by multiplexer?
 - a : D1
 - b : D2
 - c : D3
 - d : D0
 - Posttest:** One application of a digital multiplexer is to facilitate:
 - a : Data generation.
 - b : Serial-to-parallel conversion.
 - c : Parity checking.
 - d : Data selector.
 - Buttons:** Submit Quiz, 5 out of 5.

CONCLUSION:

Thus, 8:1 multiplexer was designed and implemented using IC-74LS153 and its truth table was verified.

F] Implementation and verification of decoder/de-multiplexer and encoder using logic gates.

AIM: To analyse the truth table of $4 * 2$ decoder/de-multiplexer using NOT (7404) and AND (7408) logic gate ICs and $2 * 4$ encoder using OR (7432) logic gate IC and to understand the working of $4 * 2$ decoder and $2 * 4$ encoder circuit with the help of LEDs display.

THEORY: Introduction

Binary code of N digits can be used to store 2^N distinct elements of coded information. This is what encoders and decoders are used for. Encoders convert 2^N lines of input into a code of N bits and Decoders decode the N bits into 2^N lines.

1) 2x4 Decoder / De-multiplexer

The name “Decoder” means to translate or decode coded information from one format into another, so a digital decoder transforms a set of digital input signals into an equivalent decimal code at its output

A decoder is a combinational circuit that converts binary information from n input lines to a maximum of $m=2^n$ unique output lines.



Figure 1. Logic Diagram of Decoder

1.1) 2-to-4 Binary Decoder

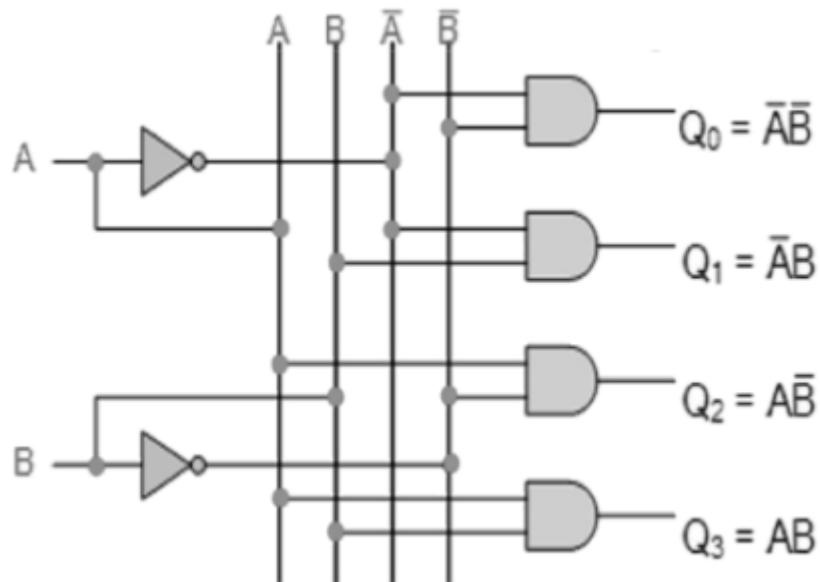


Figure 2. Circuit Diagram of 2-to-4 Decoder

The 2-to-4 line binary decoder depicted above consists of an array of four AND gates. The 2 binary inputs labelled A and B are decoded into one of 4 outputs, hence the description of 2-to-4 binary decoder. Each output represents one of the minterms of the 2 input variables, (each output = a minterm).

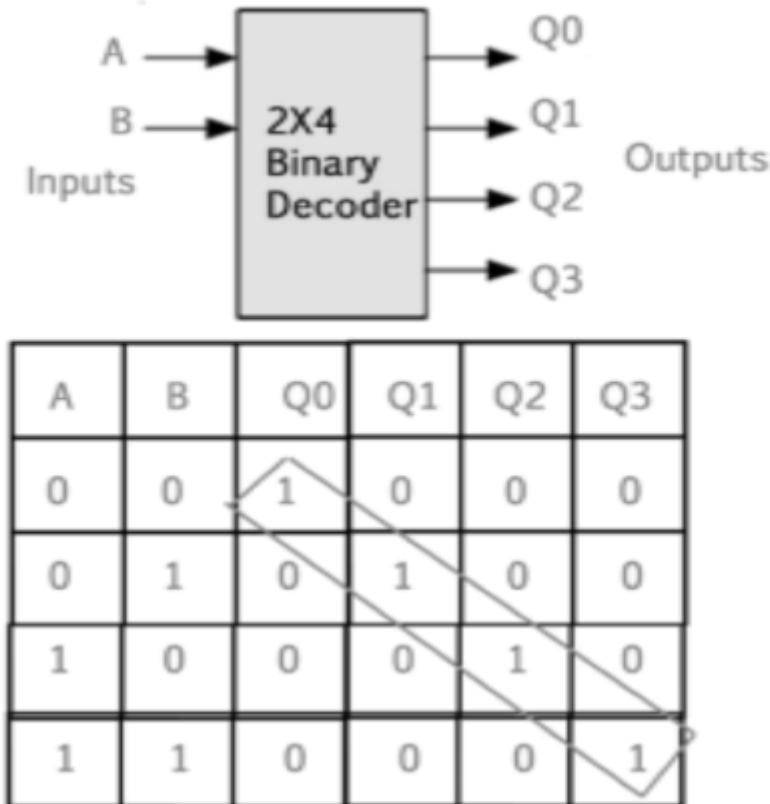


Figure 3. Logic Diagram and Truth table of 2-to-4 Decoder

The binary inputs A and B determine which output line from Q0 to Q3 is “HIGH” at logic level “1” while the remaining outputs are held “LOW” at logic “0” so only one output can be active (HIGH) at any one time.

Therefore, whichever output line is “HIGH” identifies the binary code present at the input, in other words it “decodes” the binary input. Some binary decoders have an additional input pin labelled “Enable” that controls the outputs from the device.

This extra input allows the decoders outputs to be turned “ON” or “OFF” as required. Output is only generated when the Enable input has value 1; otherwise, all outputs are 0. Only a small change in the implementation is required: the Enable input is fed into the AND gates which produce the outputs.

If Enable is 0, all AND gates are supplied with one of the inputs as 0 and hence no output is produced. When Enable is 1, the AND gates get one of the inputs as 1, and now the output depends upon the remaining inputs. Hence the output of the decoder is dependent on whether the Enable is high or low.

2) Encoder

An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2^n input lines and ‘n’ output lines, hence it encodes the information from 2^n inputs into an n-bit code. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^n input lines with ‘n’ bits.

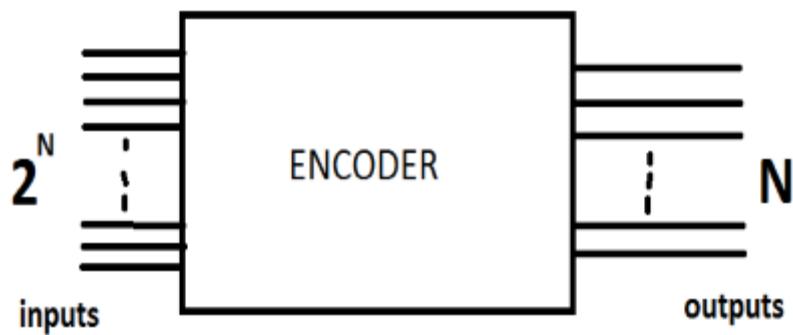


Figure 4. Logic Diagram of ENCODER

2.1)4 : 2 Encoder

The 4 to 2 Encoder consists of four inputs $Y_3, Y_2, Y_1 & Y_0$ and two outputs $A_1 & A_0$. At any time, only one of these 4 inputs can be ‘1’ in order to get the respective binary code at the output.

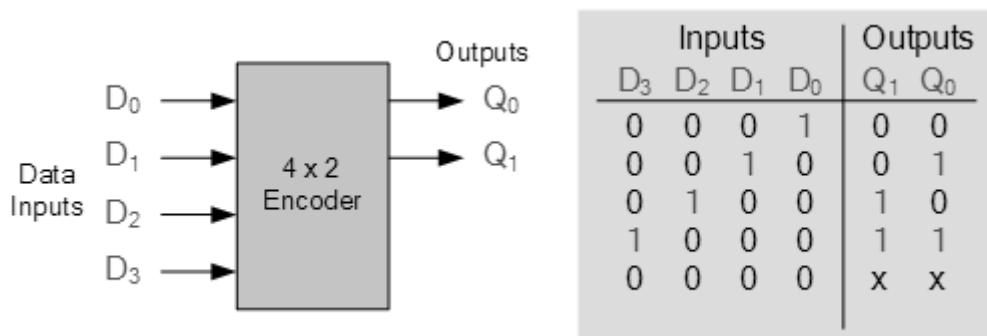


Figure 5. Logic symbol and truth table of 4 to 2 encoder

PRETEST:

DE Journal Virtual Labs

de-iitr.vlabs.ac.in/exp/decoder-demultiplexer-encoder/pretest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Theory

Pretest

BCD to seven segment conversion is a _____
 a: Decoding process
 b: Encoding process
 c: Comparing process
 d: None of the mentioned

Procedure

Simulation

Posttest

References

Feedback

Invalid BCD can be made to valid BCD by adding with _____
 a: 0101
 b: 0110
 c: 0111
 d: 1001

Device which converts an input device state into a binary representation of ones or zeros is termed as
 a: encoder
 b: decoder
 c: multiplexer
 d: data selector

A circuit that changes a code into a set of signals is called
 a: encoder
 b: decoder

Device which converts an input device state into a binary representation of ones or zeros is termed as
 a: encoder
 b: decoder
 c: multiplexer
 d: data selector

A circuit that changes a code into a set of signals is called
 a: encoder
 b: decoder
 c: multiplexer
 d: data selector

Modulo 6 counter can be built using a three-element
 a: shift register
 b: bus
 c: flip flop
 d: trigger

Submit Quiz

5 out of 5

30°C Smoke 12:33 05-01-2022

PROCEDURE:

1)2X4 DECODER / De-multiplexer



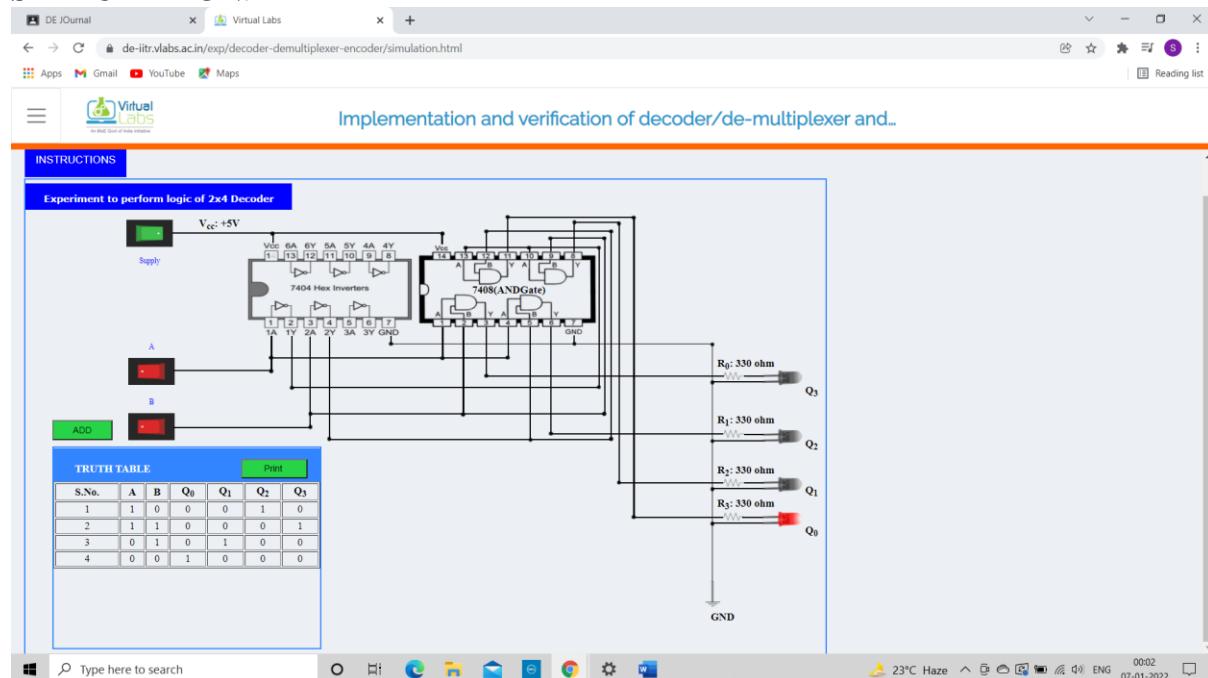
- Step-1) Connect the supply(+5V) to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switches to select the required inputs named "A" and "B". Also check their corresponding outputs named " Q_0 " and " Q_1 " and " Q_2 " and " Q_3 ".
- Step-4) Press "ADD" button to add your inputs and outputs in the given table.
- Step-5) Repeat step 3 & step 4 for next state of inputs and their corresponding outputs.
- Step-6) Press the "PRINT" button after completing your simulation to get your results.

2)4X2 ENCODER



- Step-1) Connect the supply(+5V) to the circuit.
Step-2) First press "ADD" button to add basic state of your output in the given table.
Step-3) Press the switches to select the required inputs "D₀" and "D₁" and "D₂" and "D₃". Also check their corresponding outputs named "Q₁" and "Q₀".
Step-4) Press "ADD" button to add your inputs and outputs in the given table.
Step-5) Repeat step 3 & step 4 for next state of inputs and their corresponding outputs.
Step-6) Press the "PRINT" button after completing your simulation to get your results.

STIMULATION:



POSTTEST:

A decoder converts n inputs to ____ outputs

- a: n^n
- b: n
- c: n^2
- d: 2^n

BCD to 7 segment conversion is a _____

- a: Comparing process
- b: None of the answers
- c: Encoding process
- d: Decoding process

Decoders and Encoders are doing reverse operation.

- a: True
- b: False
- c: may be
- d: may not be

Decoders and Encoders are doing reverse operation.

- a: True
- b: False
- c: may be
- d: may not be

Which of the following are building block of the Encoders?

- a: OR Gate
- b: AND Gate
- c: NOT Gate
- d: NAND Gate

Which of the following is the Decoder IC?

- a: 7890
- b: 8870
- c: 4047
- d: 4041

Submit Quiz

5 out of 5

CONCLUSION:

Thus, 4:2 decoder was implemented using NOT and AND logic gate ICs and 2:4 encoder was implemented using OR logic gate IC.

G] Implementation of 4x1 multiplexer and 1x4 demultiplexer using logic gates.

AIM: To analyse the truth table and working of 1x4 De-Multiplexer by using 3-input NAND and 1-input NOT logic gate ICs and 4x1 Multiplexer by using 3-input AND, 3-input OR, and 1-input NOT logic gate ICs.

THEORY: Introduction

The function of a multiplexer is to select the input of any 'n' input lines and feed that to one output line. The function of a de-multiplexer is to inverse the function of the multiplexer and the shortcut forms of the multiplexer. The de-multiplexers are mux and demux. Some multiplexers perform both multiplexing and de-multiplexing operations.

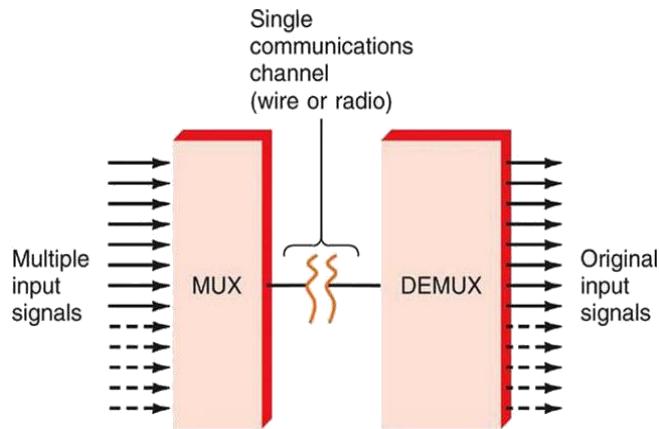


Figure-1: Block diagram of Multiplexer and De-multiplexer

1) Multiplexer
Multiplexer is a device that has multiple inputs and a single line output. The select lines determine which input is connected to the output, and also to increase the amount of data that can be sent over a network within certain time. It is also called a data selector.

Multiplexers are classified into four types:

- a) 2-1 multiplexer (1 select line)
- b) 4-1 multiplexer (2 select lines)
- c) 8-1 multiplexer(3 select lines)
- d) 16-1 multiplexer (4 select lines)

1.1) 4x1 Multiplexer

4x1 Multiplexer has four data inputs D0, D1, D2 & D3, two selection lines S0 & S1 and one output Y. The block diagram of 4x1 Multiplexer is shown in the following figure. One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. Truth table of 4x1 Multiplexer is shown below.

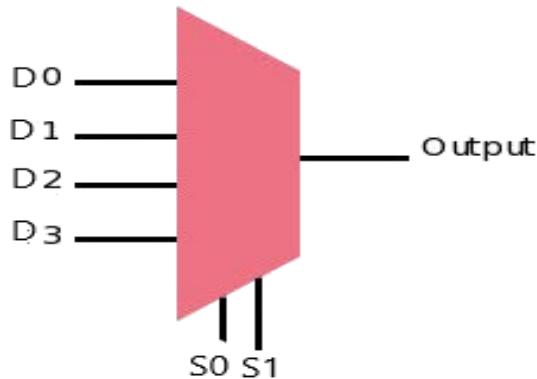


Figure-2:Block diagram of 4x1 Multiplexer

Selection Lines	Output	
S0	S1	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

Figure-3:Truth table of 4x1 Multiplexer

2) De-multiplexer De-multiplexer is also a device with one input and multiple output lines. It is used to send a signal to one of the many devices. The main difference between a multiplexer and a de-multiplexer is that a multiplexer takes two or more signals and encodes them on a wire, whereas a de-multiplexer does reverse to what the multiplexer does.

De-multiplexer are classified into four types:

- a)1-2 demultiplexer (1 select line)
 b)1-4 demultiplexer (2 select lines)
 c)1-8 demultiplexer (3 select lines)
 d)1-16 demultiplexer (4 select lines)

2.2)

1x4

De-multiplexer

1x4 De-Multiplexer has one input Data(D), two selection lines, S0 & S1 and four outputs Y0, Y1, Y2 & Y3. The block diagram of 1x4 De-Multiplexer is shown in the following figure.

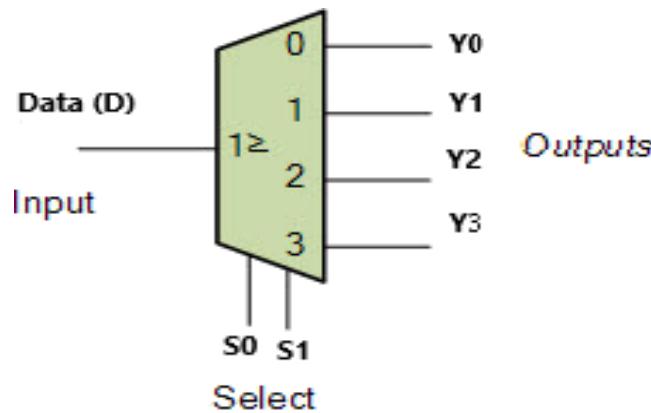


Figure-4: Block diagram of 1x4 De-Multiplexer

Selection Inputs		Outputs			
S0	S1	Y3	Y2	Y1	Y0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

Figure-5: Truth table of 1x4 De-Multiplexer

PRETEST:

DE Journal Virtual Labs

de-iitr.labs.ac.in/exp/multiplexer-demultiplexer/pretest.html

HOME PARTNERS CONTACT

Aim Implementation of 4x1 multiplexer and 1x4 demultiplexer using logic gates.

Theory

Pretest

A demultiplexer accepts inputs.
 a: Single
 b: Multiple
 c: Two
 d: Three

Procedure

In a multiplexer, the selection of a particular input line is controlled by _____
 a: Data controller
 b: Selected lines
 c: Logic gates
 d: Both data controller and selected lines

Simulation

If the number of n selected input lines is equal to 2^m then it requires _____ select lines.
 a: 2
 b: m
 c: n
 d: z^n

Posttest

References

Feedback

Which of the following circuit can be used as parallel to serial converter?

Type here to search

30°C Smoke 12:36 05-01-2022

PROCEDURE:

1) 4x1 Multiplexer



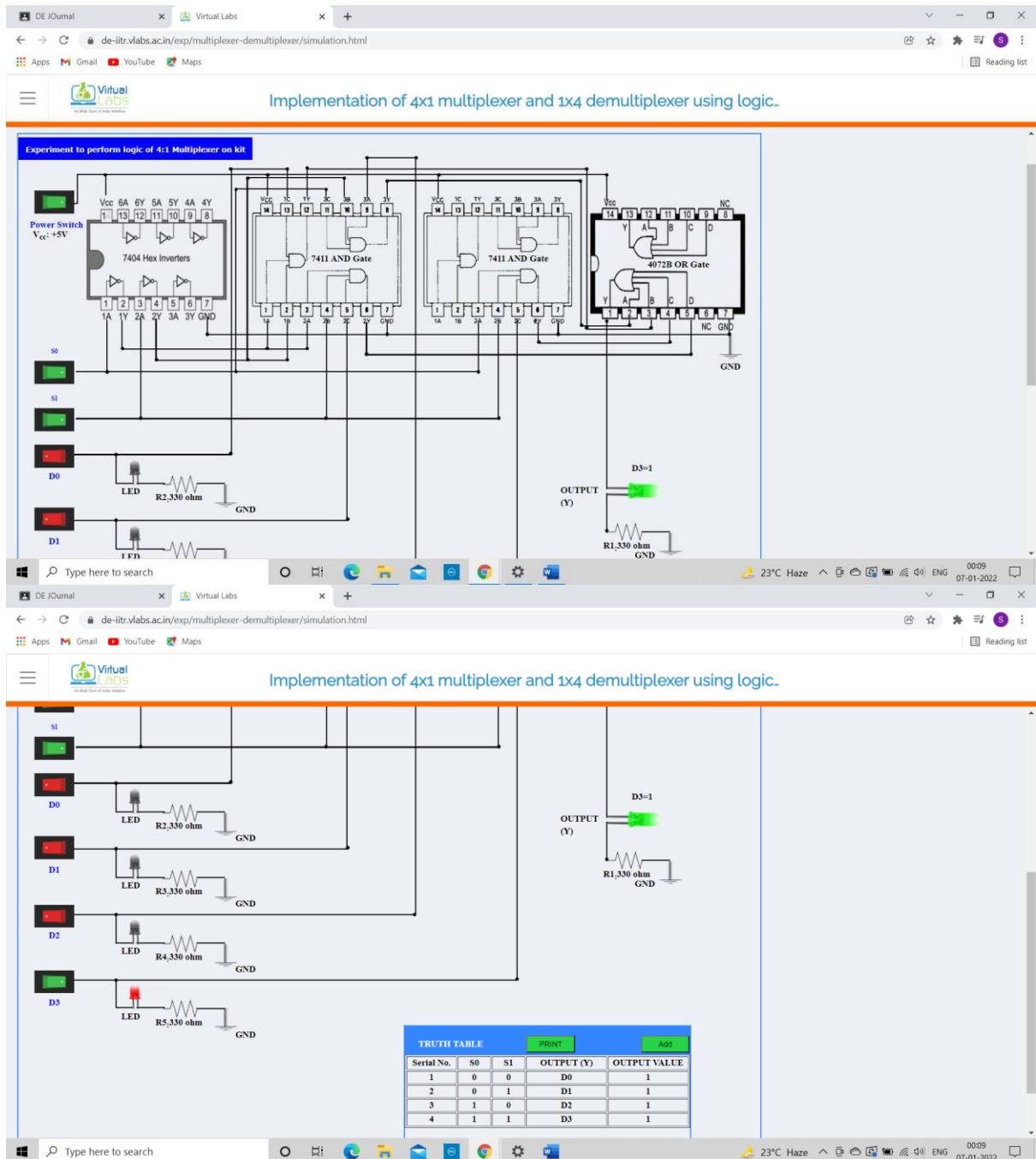
- Step-1) Connect the supply(+5V) to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switches "S0" and "S1" to select the desired input line.
- Step-4) Press "D0"/"D1"/"D2"/"D3" any one button to add your inputs.
- Step-5) Press "ADD" button to add your inputs and outputs in the given table.
- Step-6) Repeat step 3, 4 and step 5 for next state of inputs and their corresponding outputs.
- Step-7) Press the "PRINT" button after completing your simulation to get your results.

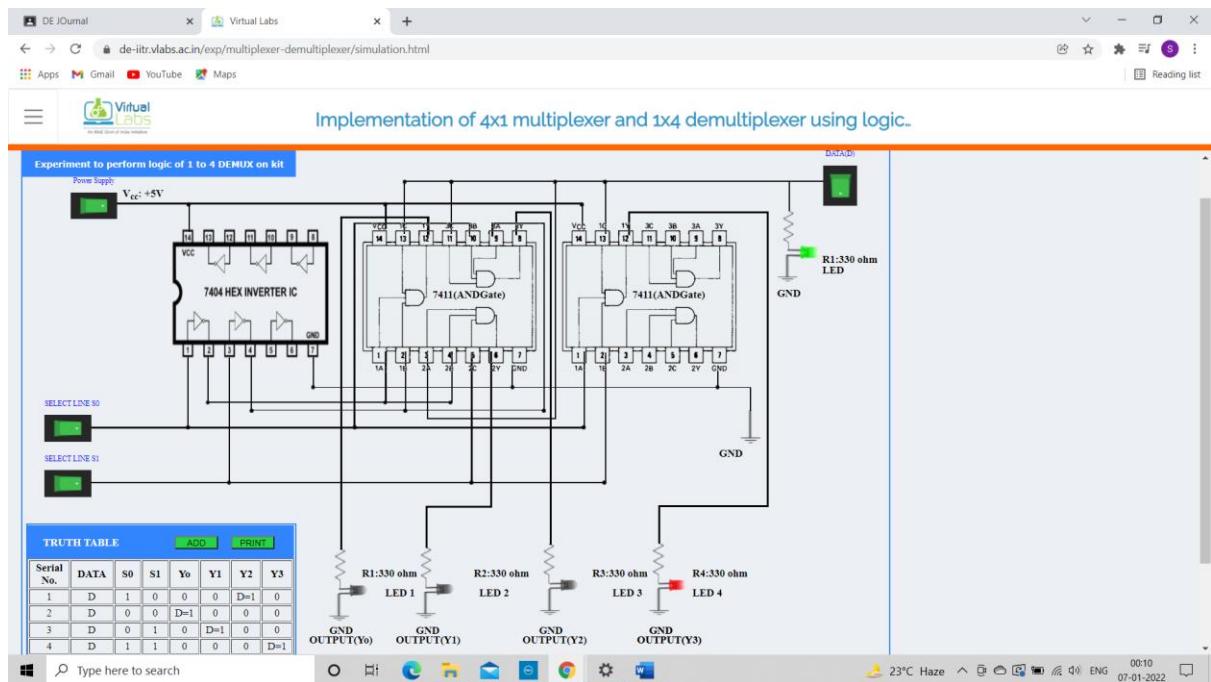
2) 1x4 De-Multiplexer



- Step-1) Connect the supply(+5V) to the circuit.
- Step-2) First press "ADD" button to add basic state of your output in the given table.
- Step-3) Press the switch Data(D) for Input.
- Step-4) Press switches "S0" and "S1" to select the desired input line.
- Step-5) Press "ADD" button to add your inputs and outputs in the given table.
- Step-6) Repeat step 4 and step 5 for next state of inputs and their corresponding outputs.
- Step-7) Press the "PRINT" button after completing your simulation to get your results.

STIMULATION:





POSTTEST:

Implementation of 4x1 multiplexer and 1x4 demultiplexer using logic gates.

Theory

How many NOT gates are required for the construction of a 4-to-1 multiplexer?

- a: 3
- b: 4
- c: 2
- d: 5

In 1-to-4 demultiplexer, how many select lines are required?

- a: 2
- b: 3
- c: 4
- d: 5

How many select lines are required for a 1-to-8 demultiplexer?

- a: 2
- b: 3
- c: 4
- d: 5

Which IC is used for the implementation of 1-to-16 DEMUX?

- a: 74138
- b: 74151
- c: 74152
- d: 74153

DE Journal × Virtual Labs × +

de-itrlabs.ac.in/exp/multiplexer-demultiplexer/posttest.html

App Gmail YouTube Maps

Virtual Labs
An MoU Govt of India Initiative

HOME PARTNERS CONTACT

How many select lines are required for a 1-to-8 demultiplexer?

a: 2
 b: 3
 c: 4
 d: 5

Which IC is used for the implementation of 1-to-16 DEMUX?

a: IC 74154
 b: IC 74155
 c: IC 74139
 d: IC 74138

The enable input is also known as _____

a: Select input
 b: Decoded input
 c: Strobe
 d: Sink

Submit Quiz

5 out of 5

Type here to search 30°C Smoke 12:37 05-01-2022

CONCLUSION:

Thus, the 1:4 demultiplexer was implemented by using 3 input NAND and 1 input NOT logic gate ICs and 4:1 multiplexer by using basic gates(AND, OR and NOT) ICs.

Practical no 8 Study of flip-flops and counters

A] Implementation of JK Flip-Flop

AIM: Implementation of JK Flip-Flop

THEORY: Flip-flops are synchronous bistable devices, also known as bistable multivibrators. Here synchronous means that the output changes state only at a specific point on a triggering input called the clock (CLK), which is designated as a control input C; that is, changes in the output occur in synchronization with the clock. An edge-triggered flip-flop changes state either at the positive edge (riging edge) or at the negative edge (falling edge) of the clock pulse and is sensitive to its inputs only at this transition of the clock.

Edge-Triggered J-K Flip-Flop

The J-K flip-flop is versatile and is a widely used type of flip-flop. The J and K designations for the inputs have no known significance except that they are adjacent letters in the alphabet. The function of J-K flip-flop is identical to that of the S-R flip-flop in the SET, RESET and no-change conditions of operation. The difference is that the J-K flip-flop has no invalid state as does the S-R flip-flop. The input mark J is for set and the input mark K is for reset. When both inputs J and K are equal to 1, the flip flop switches to its complement state, that is, if $Q = 1$, it switches to $Q = 0$ and vice versa. A J-K flip-flop constructed with two crossed coupled NOR gates and two AND gates. Output Q is ANDed with K and CP inputs so that the flip-flop is cleared during a clock pulse only if Q was previously 1. Similarly output Q' is ANDed with J and CP inputs so that the flip is set with a clock pulse only when Q' was previously 1. When both J and K are 1, the input pulse is transmitted through one AND gate only; the one whose input is connected to the flip-flop output that is presently equal to 1. Thus if $Q = 1$, the output of the upper AND gate becomes 1 upon application of the clock pulse and the flip-flop is cleared. If $Q' = 1$, the output of the lower of the lower AND gate becomes 1 and the flip-flop is set. In either case the output of the flip-flop is complemented. It is very important to realize that because of the feedback connection in the JK flip-flop, a CP pulse that remains in the 1 state while both J and K are equal to 1 will cause the output to complement again and repeat complementing until the pulse goes back to 0. To avoid this undesirable operation, the clock pulse must have a time duration that is shorter than the propagation delay time of the flip-flop.

PRETEST:

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Implementation of JK Flip-Flop

1. When both inputs of a J-K flip-flop cycle, the output will

a: Be invalid

b: Change

c: Not change

d: Toggle

2. What does the triangle on the clock input of a J-K flip-flop mean?

a: Level enabled

b: Level triggered

c: Both a & b

d: Edge triggered

Submit Quiz

2 out of 2

Community Links Contact Us Follow Us

Type here to search

23°C Haze 00:14 07-01-2022

PROCEDURE:

Components used:

We used the following components for this experiment-

IC 7476 (J-K flip-flop) Datasheet

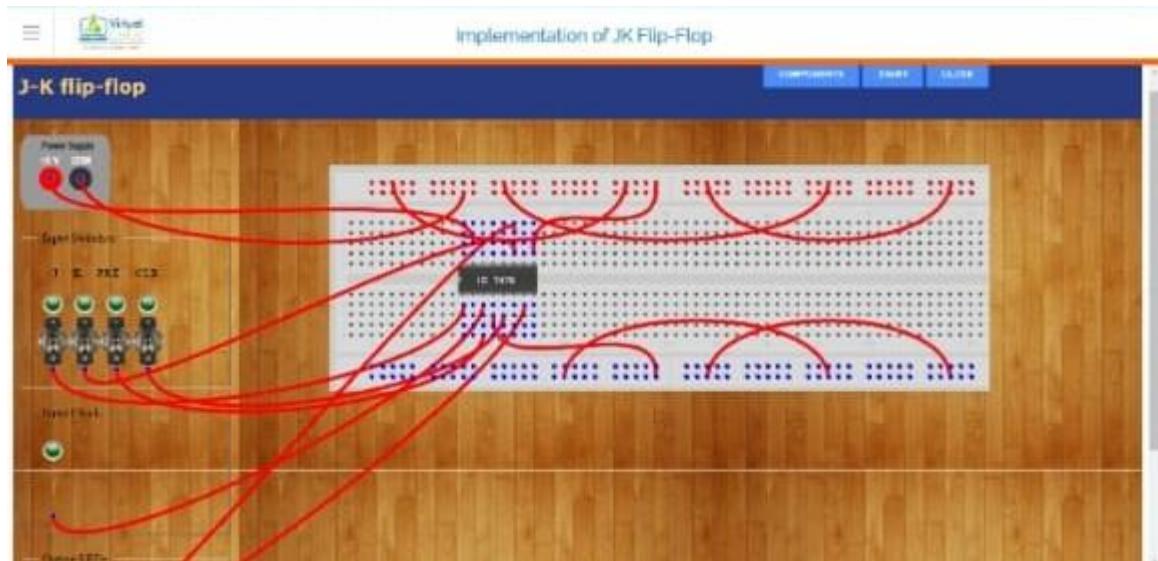
How to make connection:

After Starting the experiment first click on the Components button to get component list. Now you can Drag and Drop any component in the circuit designing area. To make connection between components, just click on the blue bubble of any components and drag it to another Blue bubble of the same or any other components. To delete connection or to remove any component use Double click on that component or connection.

How to run:

After connecting all the required components, click on the Start button and you will get a new start window, where you can give the inputs. After this, you click the run button and finally the outputs are shown.

STIMULATION:



POSTTEST:

DE Journal Virtual Labs

de-iitg.vlabs.ac.in/exp/truth-tables-flip-flops/posttest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Electronics IITG > Experiments

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Implementation of JK Flip-Flop

1.What does the half circle on the clock input of a J-K flip-flop mean?
 a: Level enabled
 b: Level triggered
 c: negative edge triggered
 d: Positive edge triggered

2.How is a J-K flip-flop made to toggle?
 a: J = 0, K = 0
 b: J = 1, K = 0
 c: J = 0, K = 1
 d: J = 1, K = 1

Submit Quiz

2 out of 2

Community Links Contact Us Follow Us

Type here to search

23°C Haze 00:18 07-01-2022

Conclusion:

Thus, JK flip flop was implemented using IC-7476.

B] Synchronous up down counter

AIM: To implement synchronous up down counter

THEORY: A counter is a sequential circuit that goes through a prescribed sequence of states upon the application of input pulses. The input pulses called count pulses, may be clock pulses, or they may originate from an external source and may occur at prescribed intervals of time or at random.

Synchronous Counter

Synchronous counters are designed in such a way that the clock pulses are applied to the CP inputs of all the flip-flops. The common pulse triggers all the flip-flops simultaneously, rather than one at a time in succession.

In the 3-bit synchronous counter, we have used three J-K flip-flops. As in the diagram, The J and K inputs of FF0 are connected to HIGH. The inputs J and K of FF1 are connected to the output of FF0, and the J and K inputs of FF2 are connected to the output of an AND gate, which is fed by the outputs of FF0 and FF1.

PRETEST:

The screenshot shows a web browser window with the URL de-iitg.vrlabs.ac.in/exp/synchronous-up-down-counter/pretest.html. The page title is "Virtual Labs". The main content area is titled "Theory" and contains a series of multiple-choice questions related to counters:

1. UP-DOWN counter is a combination of:
 - a: Latches
 - b: Flip-flops
 - c: UP counter
 - d: Up counter & down counter
2. Binary counter that counts incrementally and decrementally is called
 - a: Up-down counter
 - b: LSI counters
 - c: Down counter
 - d: Up counter
3. UP-DOWN counter is also known as
 - a: Dual counter
 - b: Multi counter
 - c: Multimode counter
 - d: None of the mentioned

At the bottom of the page, there is a "Submit Quiz" button and a status message "3 out of 3". The browser interface includes a search bar, taskbar icons, and system status indicators like temperature and date.

PROCEDURE: Components used:

We used the following components for this experiment-

IC 7476 (J-K flip-flop) Datasheet

IC 7408 (AND gate) Datasheet

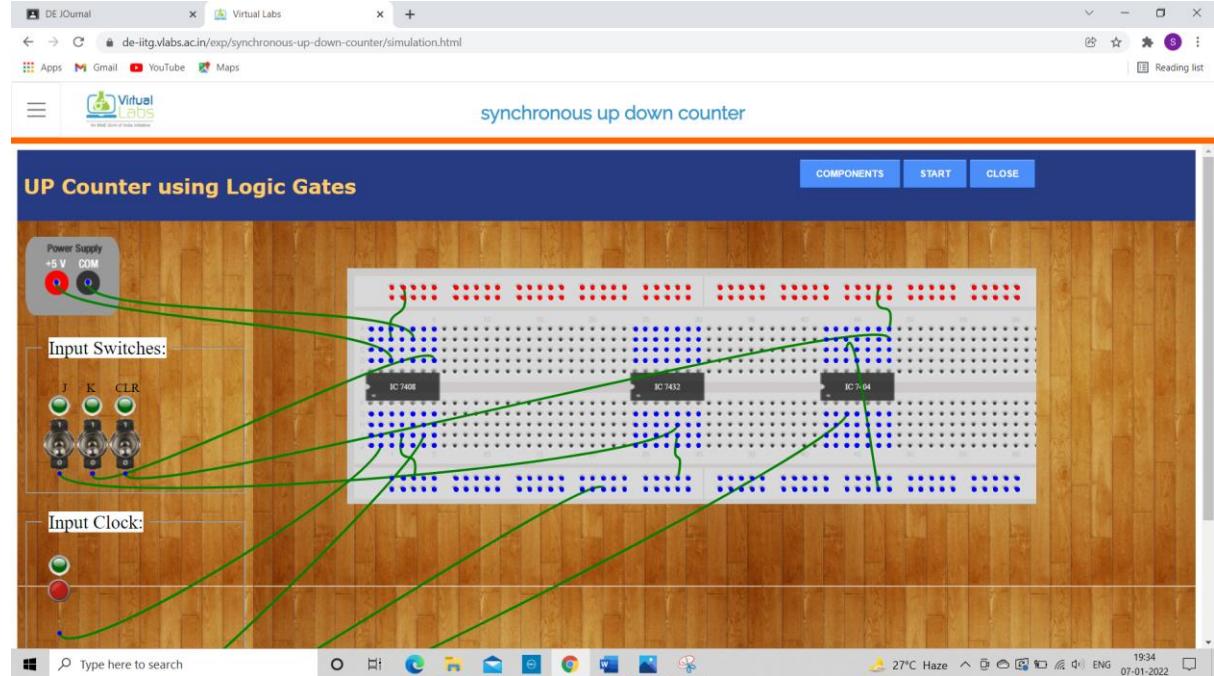
How to make connection:

After Starting the experiment first click on the Components button to get component list. Now you can Drag and Drop any component in the circuit designing area. To make connection between components, just click on the Blue bubble of any components and Drag it to another Blue bubble of the same or any other components. To delete connection or to remove any component use Double click on that component or connection.

How to run:

After connecting all the required components, click on the Start button and you will get a new start window, where you can give the inputs. After this, you click the run button and finally the outputs are shown.

STIMULATION:



POSTTEST:

The screenshot shows a posttest quiz page. The left sidebar lists categories: Aim, Theory, Pretest, Procedure, Simulation, Posttest (which is currently selected), References, and Feedback. The main content area has a title 'synchronous up down counter' and two questions. Question 1 asks about the trigger for each flip-flop, with options a, b, c, and d. Option b is selected. Question 2 asks what a counter is, with options a, b, c, and d. Option b is selected. At the bottom, there's a 'Submit Quiz' button and a status message '2 out of 2'.

CONCLUSION:

Thus, synchronous up down counter was implemented by using IC-7476(JK flip flop) and IC-7408(AND gate)

C] Construction of NOR gate latch and verification of its operation

AIM: To verify the truth table and timing diagram of NOR gate latch using NOR gate IC and analyse the circuit of NOR gate latch with the help of LEDs display.

THEORY: Introduction

Latches are basic storage elements that operate with signal levels (rather than signal transitions). Latches controlled by a clock transition are flip-flops. Latches are edge-sensitive devices. Latches are useful for the design of the asynchronous sequential circuit.

SR (Set-Reset) Latch – SR Latch is a circuit with:

- (i) 2 cross-coupled NOR gates or 2 cross-coupled NAND gates.
- (ii) 2 inputs S for SET and R for RESET.
- (iii) 2 outputs Q, Q.

The SR Latch using NOR gate is shown below with its truth table:

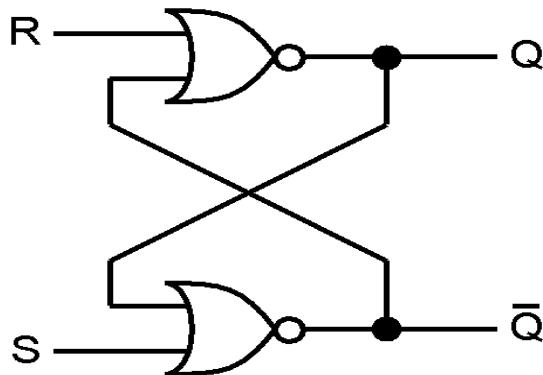


Figure-1:Logic Symbol of NOR gate latch

R	S	Q(n)	Q'(n)
0	0	Q(n-1)	Q'(n-1)
0	1	1	0
1	0	0	1
1	1	Invalid outputs	

Figure-2:Truth Table of NOR gate latch

While the R and S inputs are both low, feedback maintains the Q and \bar{Q} outputs in a constant state, with Q the complement of \bar{Q} . If S (Set) is pulsed high while R (Reset) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low. The R = S = 1 combination is called a restricted combination or a forbidden state because, as both NOR gates then output zeros, it breaks the logical equation Q = \bar{Q} . The combination is also inappropriate in circuits where both inputs may go low simultaneously (i.e. a transition from restricted to keep). The output would lock at either 1 or 0 depending on the propagation time relations between the gates (a race condition).

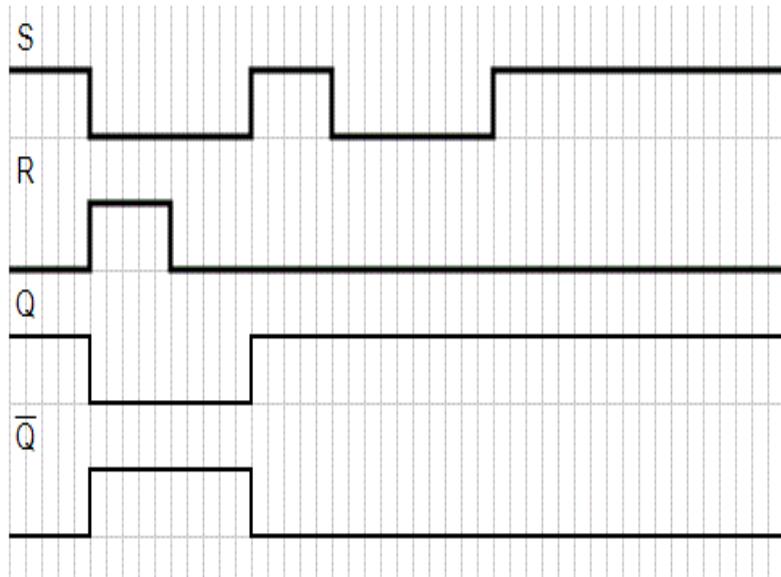


Figure-3:Timing Diagram of NOR gate latch

PRETEST:

Screenshot of a web-based pretest for a NOR gate latch experiment. The page title is "Construction of NOR gate latch and verification of its operation". The left sidebar contains links for Aim, Theory, Pretest (which is currently selected), Procedure, Simulation, Posttest, References, and Feedback. The main content area displays questions and answer options. The browser address bar shows the URL: de-iitr.labs.ac.in/exp/nor-gate-latch/pretest.html. The system status bar at the bottom indicates the date as 07-01-2022, the time as 00:24, the temperature as 23°C, and the weather as Haze.

Pretest

The full form of SR is _____

a: System Rated
 b: Set Reset
 c: Set Ready
 d: Set Rated

A latch is an example of a _____

a: Monostable multivibrator
 b: Astable multivibrator
 c: Bistable multivibrator
 d: 555 timer

Why latches are called a memory devices?

a: It has capability to store 8 bits of data
 b: It has internal memory of 4 bit
 c: It can store one bit of data

DE Journal x Virtual Labs x +

de-itrl.vlabs.ac.in/exp/nor-gate-latch/pretest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Why latches are called a memory devices?

a: It has capability to store 8 bits of data
 b: It has internal memory of 4 bit
 c: It can store one bit of data
 d: It can store infinite amount of data

There are _____ types of latches.

a: 2
 b: 5
 c: 4
 d: 3

The SR latch consists of _____ inputs

a: 1
 b: 2
 c: 3
 d: 4

Submit Quiz

5 out of 5

Type here to search

23°C Haze ENG 00:24 07-01-2022

PROCEDURE:



Step-1) Connect the supply(+5V)  to the circuit.

Step-2) Press switch 1  for input R(Reset) and switch 2  for input S(Set).

Step-3) Then press "ADD" button to add data in the given truth table and simultaneously generate the timing diagram.

Step-4) Repeat step 2 and step 3 for another set of data.

Step-5) Press the "Print" button after completing your simulation to get your results.

STIMULATION:

DE Journal x Virtual Labs x +

de-itrl.vlabs.ac.in/exp/nor-gate-latch/simulation.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

Construction of NOR gate latch and verification of its operation

Construction of NOR Gate Latch and verification of its operation

Battery

V_{CC}

IC 7402

Switch 1 (R)

Switch 2 (S)

Q

Q̄

Add

Connected!!

23°C Haze ENG 00:24 07-01-2022

DE Journal × Virtual Labs × +

de-iitr.vlabs.ac.in/exp/nor-gate-latch/simulation.html

Apps Gmail YouTube Maps

☰ Construction of NOR gate latch and verification of its operation

Add

TRUTH TABLE

Serial No.	R	S	Q_{n-1}	\bar{Q}_{n-1}	Q_n	\bar{Q}_n	Remarks
1	1	0	x	x	0	1	Reset
2	0	1	0	1	1	0	Set
3	1	1	1	0	0	0	Invalid
4	0	0	0	0	0	1	No change

TIMING DIAGRAM

23°C Haze 00:25 07-01-2022

POSTTEST:

DE Journal × Virtual Labs × +

de-iitr.vlabs.ac.in/exp/nor-gate-latch/posttest.html

Apps Gmail YouTube Maps

☰

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Theory

Pretest

When a high is applied to the set line of an SR latch, then _____

a: Q output goes high
 b: Q output goes high
 c: Both Q and Q' goes high
 d: Q output goes low

Procedure

Latch is a device with _____

a: One stable state
 b: Two stable states
 c: Three stable states
 d: Infinite stable states

Simulation

When both inputs of SR latches are low, the latch _____

a: Q output goes high
 b: Q output goes high
 c: It remains in its previously set or reset state
 d: It goes to its next set or reset state

Posttest

In an SR latch built from NOR gates, which condition is not allowed?

a: S=0, R=0
 b: S=1, R=1
 c: S=1, R=0
 d: S=0, R=1

23°C Haze 00:26 07-01-2022

When both inputs of SR latches are low, the latch _____

a: Q output goes high
 b: Q output goes high
 c: It remains in its previously set or reset state
 d: It goes to its next set or reset state

In an SR latch built from NOR gates, which condition is not allowed?

a: S=0, R=0
 b: S=0, R=1
 c: S=1, R=0
 d: S=1, R=1

A latch is _____ sensitive.

a: Edge
 b: Level
 c: Both level and edge
 d: None

Submit Quiz

5 out of 5

CONCLUSION:

Thus, NOR gate latch was verified using NOR gate IC and truth table and timing diagram was generated.

D] Verify the truth table of RS, JK, T and D flip-flops using NAND & NOR gates

AIM: To verify the truth table and timing diagram of RS, JK, T and D flip-flops by using NAND & NOR gates ICs and analyse the circuit of RS, JK, T and D flip-flops with the help of LEDs display.

THEORY:

Introduction

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems.

1. R-S flip flop
2. D flip flop
3. J-K flip flop
4. T flip flop

1) RS flip flop

The basic NAND gate RS flip flop circuit is used to store the data and thus provides feedback from both of its outputs again back to its inputs. The RS flip flop actually has three inputs, SET, RESET and clock pulse.

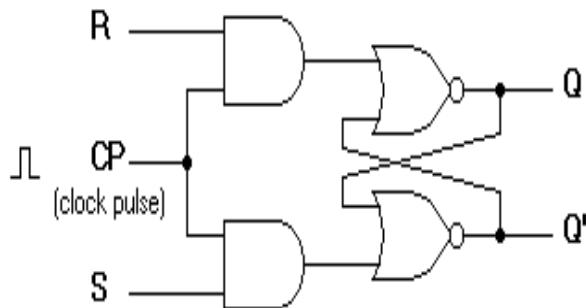


Figure-1:R-S flip flop circuit diagram

INPUTS			OUTPUT T	STATE
CLK	S	R	Q	
X	0	0	No Change	Previous
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	-	Forbidden

Figure-2:Characteristics table of R-S flip flop

2) D flip flop

A D flip flop has a single data input. This type of flip flop is obtained from the SR flip flop by connecting the R input through an inverter, and the S input is connected directly to data input. The modified clocked SR flip-flop is known as D-flip-flop and is shown below. From the truth table of SR flip-flop we see that the output of the SR flip-flop is in unpredictable state when the inputs are same and high. In many practical applications, these input conditions are not required. These input conditions can be avoided by making them complement of each other.

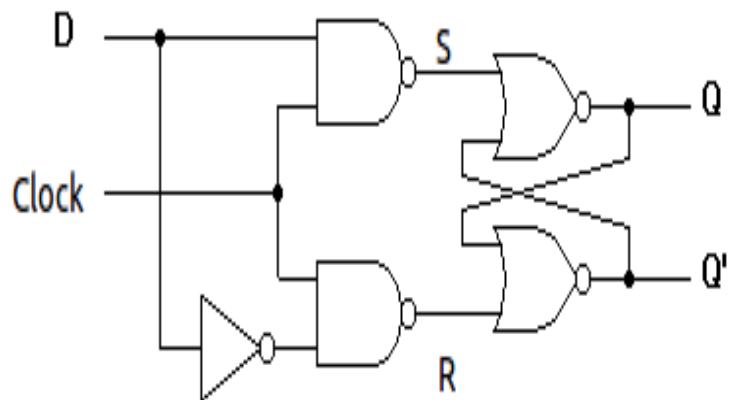


Figure-3:Circuit diagram of D flip flop

Input			Output	
D	reset	clock	Q	Q'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

Figure-4:Characteristics table of D flip flop

3) J-K flip flop

In a RS flip-flop the input $R=S=1$ leads to an indeterminate output. The RS flip-flop circuit may be re-joined if both inputs are 1 than also the outputs are complement of each other as shown in characteristics table below.

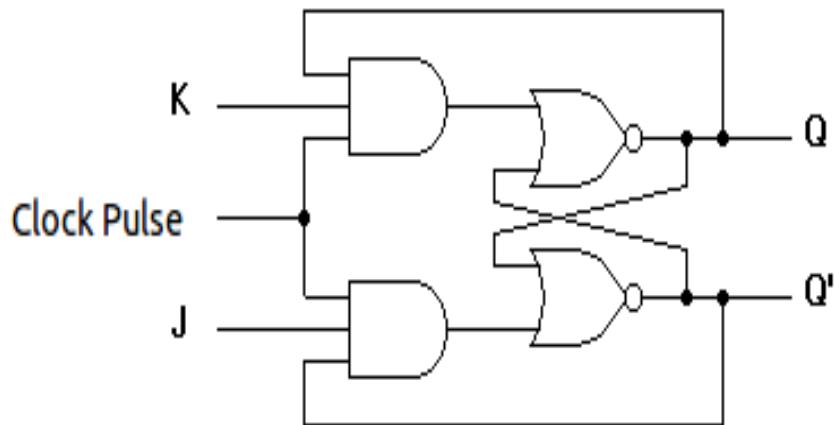


Figure-5:Circuit diagram of J-K flip flop

Trigger	Inputs		Output				Inference
			Present State		Next State		
CLK	J	K	Q	Q'	Q	Q'	
	x	x	-	-	-	-	Latched
	0	0	0	1	0	1	No Change
			1	0	1	0	
	0	1	0	1	0	1	Reset
			1	0	0	1	
	1	0	0	1	1	0	Set
			1	0	1	0	
	1	1	0	1	1	0	Toggles
			1	0	0	1	

Figure-6: Characteristics table of J-K flip flop

4) T flip flop

T flip-flop is known as toggle flip-flop. The T flip-flop is modification of the J-K flip-flop. Both the JK inputs of the JK flip – flop are held at logic 1 and the clock signal continuous to change as shown in table below.

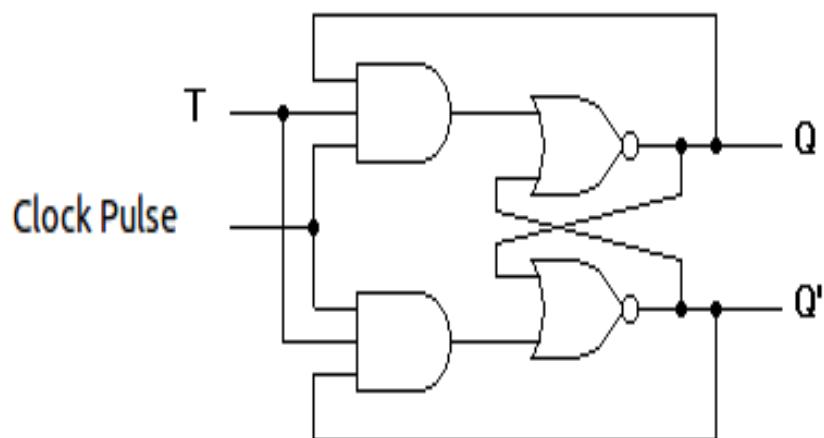


Figure-7: Circuit diagram of T flip flop

T flip-flop

T	Clock	Q	Q'
0	↑	Q	Q'
1	↑	Q'	Q
x	↓	Q	Q'

Figure-8:Characteristics table of T flip flop

PRETEST:

The output of latches will remain in set/reset until _____

a: The trigger pulse is given to change the state
 b: Any pulse given to go into previous state
 c: They don't get any pulse more
 d: The pulse is edge-triggered

What is a trigger pulse?

a: A pulse that starts a cycle of operation
 b: A pulse that reverses the cycle of operation
 c: A pulse that prevents a cycle of operation
 d: A pulse that enhances a cycle of operation

If Q = 1, the output is said to be _____

a: Set
 b: Reset
 c: Previous state
 d: Current state

The sequential circuit is also called _____

a: Flip-flop
 b: Latch

The screenshot shows a web browser window with the following details:

- Tab Bar:** DE Journal, Virtual Labs
- Address Bar:** de-iitr.vlabs.ac.in/exp/truth-tables-flip-flops/pretest.html
- Content Area:**
 - A logo for "Virtual Labs" with the text "An MoE Govt of India Initiative".
 - Navigation links: HOME, PARTNERS, CONTACT.
 - Quiz questions:
 - "d: A pulse that enhances a cycle of operation" (radio button c: Previous state selected)
 - "If Q = 1, the output is said to be _____" (radio button a: Set selected)
 - "The sequential circuit is also called _____" (radio button b: Latch selected)
 - "The basic latch consists of _____" (radio button a: Two inverters selected)
 - A "Submit Quiz" button.
- Taskbar:** Shows icons for File, Home, Task View, Start, Taskbar settings, and a search bar. The search bar contains "Type here to search".
- System Tray:** Displays weather (23°C Haze), battery level, network status, and system time (00:28 07-01-2022).

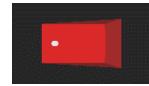
PROCEDURE:

1) SR flip flop



- Step-1) Connect the supply(+5V) to the circuit.
 Step-2) First press "ADD" button to add basic state of your output in the given table.
 Step-3) Press the switches to select the required inputs "S" and "R" and apply the clock pulse.
 Step-4) Press "ADD" button to add your inputs and outputs in the given table and their corresponding graph.
 Step-5) Repeat steps 3&4 for next state of inputs and their corresponding outputs.
 Step-6) Press the "Print" button after completing your simulation to get your results.

2) D flip flop



- Step-1) Connect the supply(+5V) to the circuit.
 Step-2) First press "ADD" button to add basic state of your output in the given table.
 Step-3) Press the switches to select the required inputs "D" and "Clock" .
 Step-4) Press "ADD" button to add your inputs and outputs in the given table and their corresponding graph.
 Step-5) Repeat step 3 & step 4 for next state of inputs and their corresponding outputs.
 Step-6) Press the "Print" button after completing your simulation to get your results.

3) J-K flip flop



- Step-1) Connect the supply(+5V) to the circuit.
 Step-2) First press "ADD" button to add basic state of your output in the given table.
 Step-3) Press the switches to select the required inputs "J" and "K" and apply the clock pulse.
 Step-4) Press "ADD" button to add your inputs and outputs in the given table and their corresponding graph.
 Step-5) Repeat step 3 & step 4 for next state of inputs and their corresponding outputs.
 Step-6) Press the "Print" button after completing your simulation to get your results.

4) T flip flop



- Step-1) Connect the supply(+5V) to the circuit.
Step-2) First press "ADD" button to add basic state of your output in the given table.
Step-3) Press the switches to select the required inputs "T" and apply the clock pulse.
Step-4) Press "ADD" button to add your inputs and outputs in the given table and their corresponding graph.
Step-6) Press the "Print" button after completing your simulation to get your results.

STIMULATION:

Verify the truth table of RS, JK, T and D flip-flops using NAND & NOR.

Experiment to perform SR Flip Flop on kit

CLOCK DIAGRAM

TRUTH TABLE

Serial No.	clock	S	R	Q(n-1)	Q(n)	Q	Remark
1	1	1	0	0	1	1	0
2	1	0	1	1	0	0	1
3	1	1	1	0	1	0	0
4	1	0	0	0	0	0	0

PRINT

Add

CLOCK DIAGRAM

Y-axis

X axis

DE Journal × Virtual Labs × +

de-iitr.vlabs.ac.in/exp/truth-tables-flip-flops/simulation.html

☰ Apps Gmail YouTube Maps

☰ Reading list

Verify the truth table of RS, JK, T and D flip-flops using NAND & NOR.

Experiment to perform logic of D - Flipflop on kit

TRUTH TABLE **PRINT** **Add**

Serial No.	clock	D	Q(n-1)	Q'(n-1)	Q	Q'	Remark
1	1	1	0	1	1	0	Set
2	1	0	1	0	0	1	Reset

CLOCK DIAGRAM

Verify the truth table of RS, JK, T and D flip-flops using NAND & NOR.

Experiment to perform logic of JK FLIP FLOP on kit

TRUTH TABLE

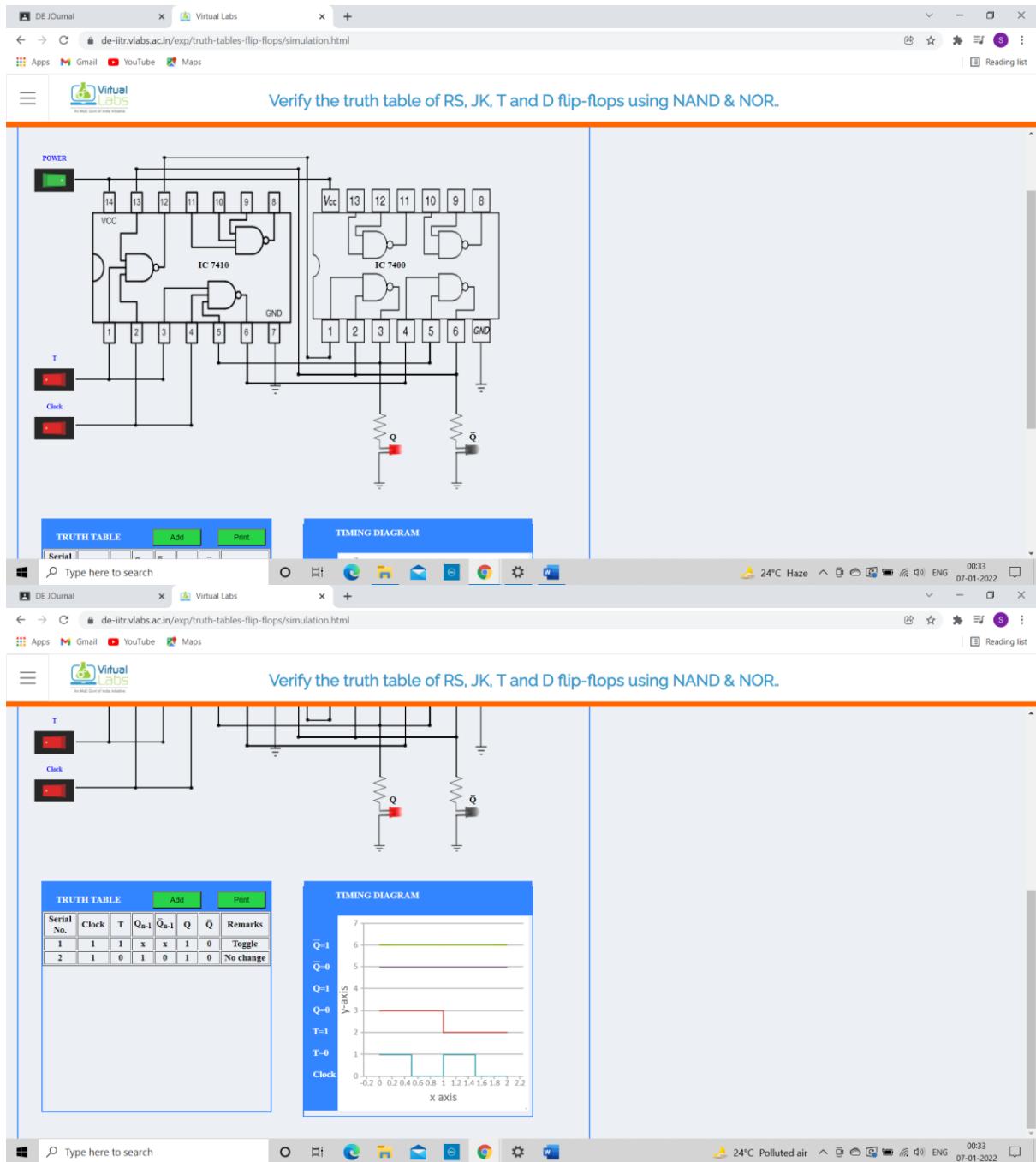
Serial No.	clock	J	K	Q(n-1)	$\bar{Q}(n-1)$	Q	\bar{Q}	Remark
1	1	1	0	0	1	1	0	set
2	1	1	0	1	1	0	0	Reset

CLOCK DIAGRAM

TRUTH TABLE

Serial No.	clock	J	K	Q(n-1)	$\bar{Q}(n-1)$	Q	\bar{Q}	Remark
1	1	1	0	0	1	1	0	set
2	1	0	1	1	0	0	1	Reset
3	1	0	0	0	1	0	1	No change
4	1	1	1	0	1	1	0	toggle

CLOCK DIAGRAM



POSTTEST:

DE Journal Virtual Labs

de-itrl.vlabs.ac.in/exp/truth-tables-flip-flops/posttest.html

App Gmail YouTube Maps

S Reading list

 An MoE Govt of India Initiative

[HOME](#) [PARTNERS](#) [CONTACT](#)

Aim Theory Pretest Procedure Simulation Posttest References Feedback

Verify the truth table of RS, JK, T and D flip-flops using NAND & NOR gates

Which of the following is correct for a gated D-type flip-flop?

- a: The Q output is either SET or RESET as soon as the D input goes HIGH or LOW
- b: The output complement follows the input when enabled
- c: Only one of the inputs can be HIGH at a time
- d: The output toggles if one of the inputs is held HIGH

A basic S-R flip-flop can be constructed by cross-coupling of which basic logic gates?

- a: AND or OR gates
- b: Ex-OR or Ex-NOR gates
- c: NOR or NAND gates
- d: AND or NOR gate

The truth table for an S-R flip-flop has how many valid entries?

- a: 1
- b: 2
- c: 3
- d: 4

The flip-flops which has not any invalid states are _____

Type here to search

DE Journal Virtual Labs

23°C Haze ENG 00:30 07-01-2022

App Gmail YouTube Maps

S Reading list

 An MoE Govt of India Initiative

[HOME](#) [PARTNERS](#) [CONTACT](#)

The truth table for an S-R flip-flop has how many valid entries?

- a: 1
- b: 2
- c: 3
- d: 4

The flip-flops which has not any invalid states are _____

- a: S-R, J-K, D
- b: S-R, J-K, T
- c: J-K, D, S-R
- d: J-K, D, T

Both the J-K & the T flip-flop are derived from the basic _____

- a: S-R flip-flop
- b: S-R latch
- c: D latch
- d: D flip-flop

[Submit Quiz](#)

5 out of 5

Community Links Contact Us Follow Us

24°C AQI 146 ENG 00:34 07-01-2022

CONCLUSION:

Thus, RS, JK, T and D flip flops were verified using NAND and NOR gates ICs and truth table and timing diagram was generated.

E] Design and verify the 4- Bit Synchronous/ Asynchronous Counter using JK flip flop

AIM: To verify the truth table and timing diagram of 4-bit synchronous parallel counter and 4-bit asynchronous parallel counter by using JK flip flop ICs and analyse the circuit of 4-bit synchronous parallel counter and 4-bit asynchronous parallel counter with the help of LEDs display.

THEORY:

Introduction

A counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit. For example, in UP counter a counter increases count for every rising edge of clock.

Classification of Counters

Counters are broadly divided into two categories

1. Asynchronous counter
2. Synchronous counter

1) Asynchronous Counter

In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following counters is driven by output of previous flip flops. We can understand it by following diagram-

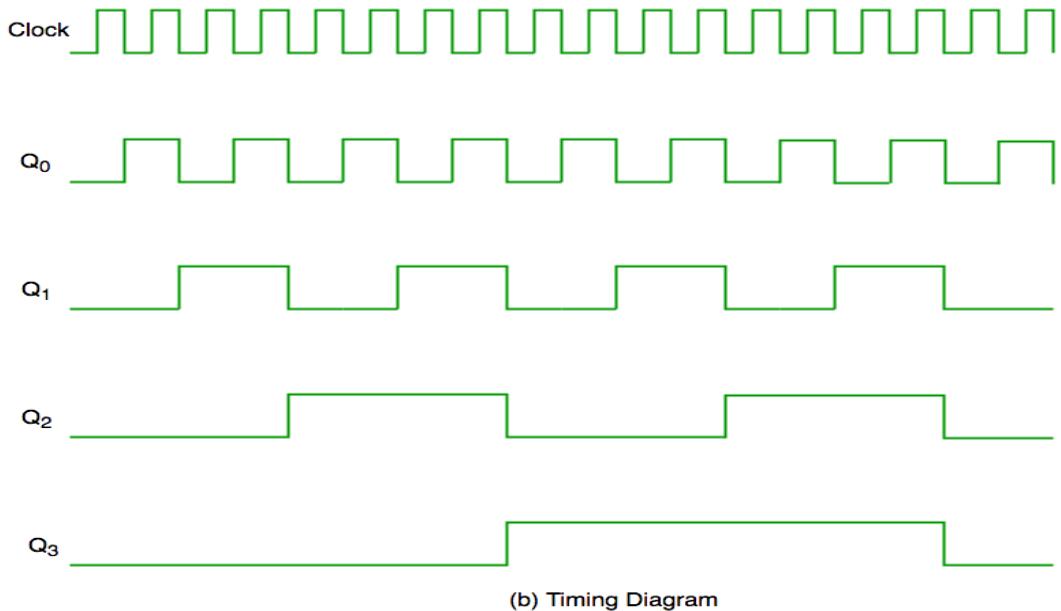
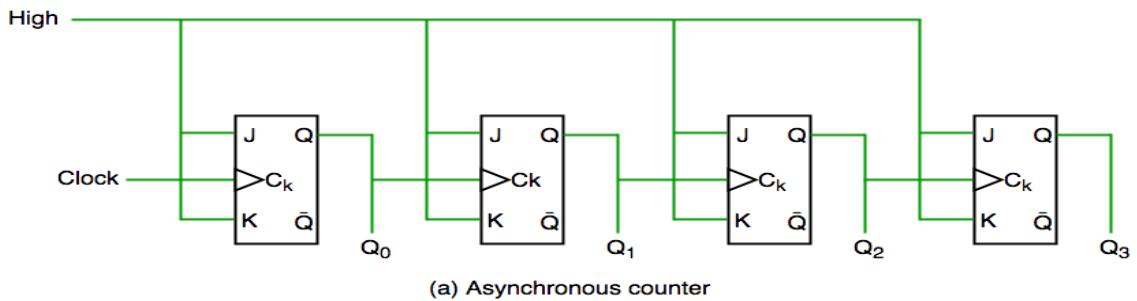


Figure-1: Asynchronous Counter Circuit and Timing Diagram

It is evident from timing diagram that Q0 is changing as soon as the rising edge of clock pulse is encountered, Q1 is changing when rising edge of Q0 is encountered(because Q0 is like clock pulse for second flip flop) and so on. In this way ripples are generated through Q0,Q1,Q2,Q3 hence it is also called RIPPLE counter.

2) Synchronous Counter

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop.

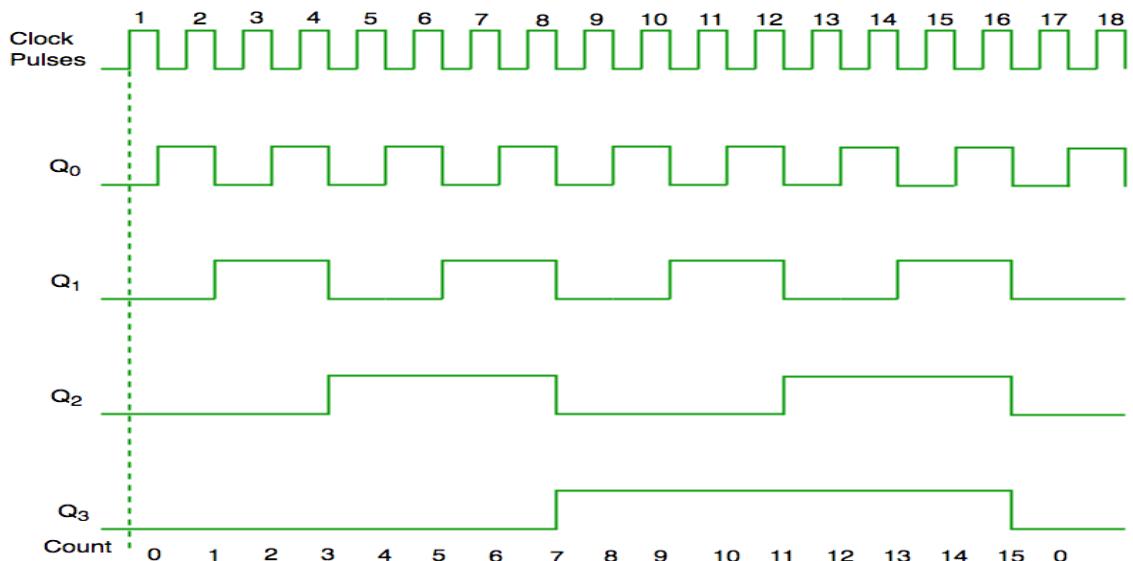
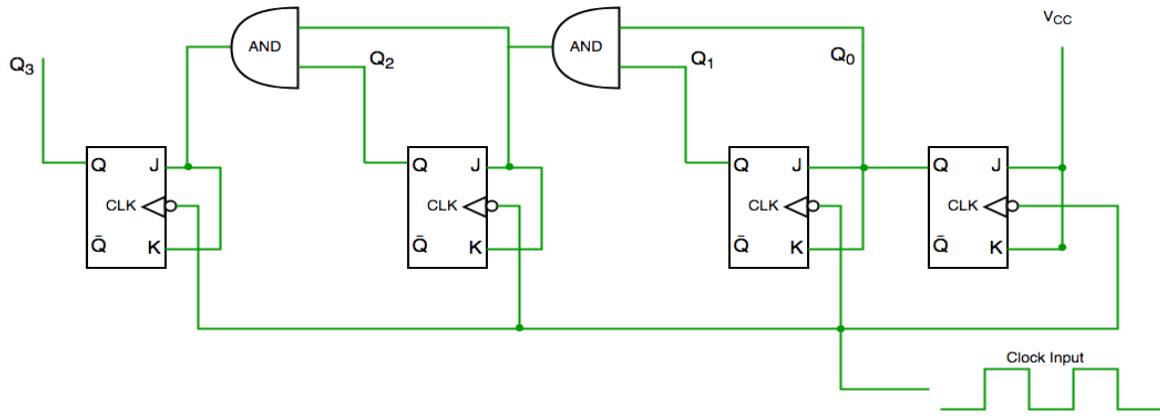


Figure-2: Synchronous Counter Circuit and Timing Diagram

From circuit diagram we see that Q0 bit gives response to each falling edge of clock while Q1 is dependent on Q0, Q2 is dependent on Q1 and Q0 , Q3 is dependent on Q2,Q1 and Q0.

PRETEST:

DE Journal Virtual Labs

In digital logic, a counter is a device which _____.

- a: Counts the number of outputs
- b: Stores the number of times a particular event or process has occurred
- c: Stores the number of times a clock pulse rises and falls
- d: Counts the number of inputs

A counter circuit is usually constructed of _____.

- a: A number of latches connected in cascade form
- b: A number of NAND gates connected in cascade form
- c: A number of flip-flops connected in cascade
- d: A number of NOR gates connected in cascade form

Ripple counters are also called _____.

- a: SSI counters
- b: Synchronous counters
- c: VLSI counters
- d: Asynchronous counters

The parallel outputs of a counter circuit represent the _____.

- a: Parallel data word
- b: Clock frequency
- c: Clock count
- d: Counter modulus

What is the maximum possible range of bit-count specifically in n-bit binary counter consisting of 'n' number of flip-flops?

- a: 0 to 2^n
- b: 0 to $2^n + 1$
- c: 0 to $2^n - 1$
- d: 0 to $2^n + 1/2$

Submit Quiz

5 out of 5

PROCEDURE:

Synchronous Counter

Step-1) Press Switch to supply 5V to the circuit.



The switch in ON state is  and the switch in OFF state is .

Step-2) Press Counter button to start the counter.



Step-3) Different combination of LEDs lit up for different combination of inputs.



The LED in OFF state is  and the LED in ON state is .

The data is simultaneously added to the Truth Table.

Step-4) Repeat Steps 2 to 3 for another set of data.

Step-5) Click on "Generate Waveform" Button to generate the Timing Diagram .

Step-6) Click "Print" to get the print out of the Truth Table and the Timing Diagram.

Asynchronous Counter

Step-1) Press Switch to supply 5V to the circuit.



The switch in ON state is and the switch in OFF state is .

Step-2) Press Counter button to start the counter.



Step-3) Different combination of LEDs lit up for different combination of inputs.



The LED in OFF state is and the LED in ON state is .
The data is simultaneously added to the Truth Table.

Step-4) Repeat Steps 2 to 3 for another set of data.

Step-5) Click on "Generate Waveform" Button to generate the Timing Diagram .

Step-6) Click "Print" to get the print out of the Truth Table and the Timing Diagram.

STIMULATION:

The screenshot shows a virtual lab simulation for a 4-bit asynchronous parallel counter using J-K flip-flops. The circuit diagram at the top features two IC 7476 chips. Each chip has its inputs (TCK, TMR, TCL, CLEAR, CLOCK) connected to a common clock source labeled "Negatively Triggered Clock". The outputs Q1 and Q0 of each chip are connected to the inputs of the next stage. A green switch labeled "Switch" is shown connected to the first stage's clock input. Below the circuit is a truth table titled "TRUTH TABLE" and a timing diagram titled "TIMING DIAGRAM". The truth table lists four rows of data with columns for Serial No., Clock, Q3, Q2, Q1, and Q0. The timing diagram shows the digital waveforms for the clock and the four output bits over 15 clock cycles.

Serial No.	Clock	Q3	Q2	Q1	Q0
1	1	0	0	0	0
2	2	0	0	0	1
3	3	0	0	1	0
4	4	0	0	1	1

TIMING DIAGRAM

Timing Diagram

Generate Waveform

24°C Haze 00:38 07-01-2022

DE Journal x Virtual Labs x +

de-iitr.vlabs.ac.in/exp/4bit-synchronous-asynchronous-counter/simulation.html

Apps Gmail YouTube Maps

Reading list

Design and verify the 4- Bit Synchronous/ Asynchronous Counter..

TRUTH TABLE

Serial No.	Clock	Q3	Q2	Q1	Q0
1	1	0	0	0	0
2	2	0	0	0	1
3	3	0	0	1	0
4	4	0	0	1	1
5	5	0	1	0	0
6	6	0	1	0	1
7	7	0	1	1	0
8	8	0	1	1	1
9	9	1	0	0	0
10	10	1	0	0	1
11	11	1	0	1	0
12	12	1	0	1	1
13	13	1	1	0	0
14	14	1	1	0	1
15	15	1	1	1	0
16	16	1	1	1	1

TIMING DIAGRAM Generate Waveform

Type here to search DE Journal Virtual Labs +

de-iitr.vlabs.ac.in/exp/4bit-synchronous-asynchronous-counter/simulation.html

Apps Gmail YouTube Maps

Reading list

Design and verify the 4- Bit Synchronous/ Asynchronous Counter..

4-BIT Synchronous Parallel Counter using J-K Flip-Flop Print

TRUTH TABLE

Serial No.	Clock	Q3	Q2	Q1	Q0
1	1	0	0	0	0
2	2	0	0	0	1
3	3	0	0	1	0
4	4	0	0	1	1

TIMING DIAGRAM Generate Waveform

DE Journal Virtual Labs

de-iitr.vlabs.ac.in/exp/4bit-synchronous-asynchronous-counter/simulation.html

Apps Gmail YouTube Maps

Design and verify the 4- Bit Synchronous/ Asynchronous Counter..

Switch

Negatively Triggered Clock

TRUTH TABLE					
Serial No.	Clock	Q3	Q2	Q1	Q0
1	1	0	0	0	0
2	2	0	0	0	1
3	3	0	0	1	0
4	4	0	0	1	1
5	5	0	1	0	0
6	6	0	1	0	1
7	7	0	1	1	0
8	8	0	1	1	1
9	9	1	0	0	0
10	10	1	0	0	1
11	11	1	0	1	0
12	12	1	0	1	1
13	13	1	1	0	0
14	14	1	1	0	1
15	15	1	1	1	0
16	16	1	1	1	1

TIMING DIAGRAM

Generate Waveform

POSTTEST:

DE Journal Virtual Labs

de-iitr.vlabs.ac.in/exp/4bit-synchronous-asynchronous-counter/posttest.html

Apps Gmail YouTube Maps

Virtual Labs

An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Design and verify the 4- Bit Synchronous/ Asynchronous Counter using JK flip flop

One of the major drawbacks to the use of asynchronous counters is that _____

- a: Low-frequency applications are limited because of internal propagation delays
- b: High-frequency applications are limited because of internal propagation delays
- c: Asynchronous counters do not have major drawbacks and are suitable for use in high- and low-frequency counting applications
- d: Asynchronous counters do not have major drawbacks and are suitable for use in high- and low-frequency counting applications

A counter is implemented using three (3) flip-flops, possibly it will have _____ maximum output status.

- a: 3
- b: 7
- c: 8
- d: 15

Which sequential circuits are applicable for counting pulses?

- a: Counters
- b: Flip Flops
- c: Registers
- d: Latches

Type here to search

24°C Haze 00:41 07-01-2022

DE Journal Virtual Labs

de-itrlabs.ac.in/exp/4bit-synchronous-asynchronous-counter/posttest.html

HOME PARTNERS CONTACT

Which sequential circuits are applicable for counting pulses?

a: Counters
 b: Flip Flops
 c: Registers
 d: Latches

A decimal counter has _____ states.

a: 5
 b: 10
 c: 15
 d: 20

Counter is a _____ .

a: Combinational circuit
 b: Sequential circuit
 c: Both
 d: None

Submit Quiz

5 out of 5

CONCLUSION:

Thus, 4-bit synchronous parallel counter and 4-bit asynchronous counter was verified using JK flip flop ICs and truth table and timing diagram were generated.

Practical no 9 Design of shift registers and shift register counters.

A] Design and Verify the 4-Bit Serial In - Parallel Out Shift Registers.

AIM: To analyse the circuit and truth table of 4-bit SIPO (serial input parallel output) shift register by using IC 7474 (D flip flop).

THEORY:

Introduction

In Serial In Parallel Out (SIPO) shift registers, the data is stored into the register serially while it is retrieved from it in parallel-fashion. Figure 1 shows an n -bit synchronous SIPO shift register sensitive to positive edge of the clock pulse. Here the data word which is to be stored (Data in) is fed serially at the input of the first flip-flop (D1 of FF1). It is also seen that the inputs of all other flip-flops (except the first flip-flop FF1) are driven by the outputs of the preceding ones like the input of FF2 is driven by the output of FF1. In this kind of shift register, the data stored within the register is obtained as a parallel-output data word (Data out) at the individual output pins of the flip-flops (Q1 to Qn).

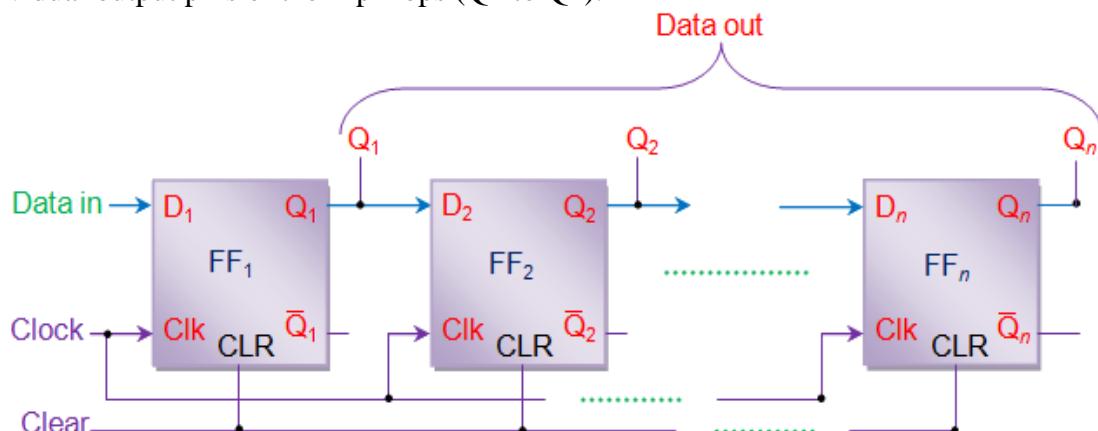


Figure 1 n -bit Serial-In Parallel-Out Right-Shift Shift Register

In general, the register contents are cleared by applying high on the clear pins of all the flip-flops at the initial stage. After this, the first bit, B1 of the input data word is fed at the D1 pin of FF1. This bit (B1) will enter into FF1, get stored and thereby appears at its output Q1 on the appearance of first leading edge of the clock. Further at the second clock pulse, the bit B1 right-shifts and gets stored into FF2 while appearing at its output pin Q2 while a new bit, B2 enters into FF1. Similarly at each clock pulse the data within the register moves towards right by a single bit while a new bit of the input word enters into the register. Meanwhile one can extract the bits stored within the register in parallel-fashion at the individual flip-flop outputs.

Analyzing on the same grounds, one can note that the n -bit input data word is obtained as an n -bit output data word from the shift register at the rising edge of the n th clock pulse. This working of the shift-register can be summarized as in Table I and the corresponding waveforms are given by figure 2.

Table I Data Movement in Right-Shift SIPO Shift Register

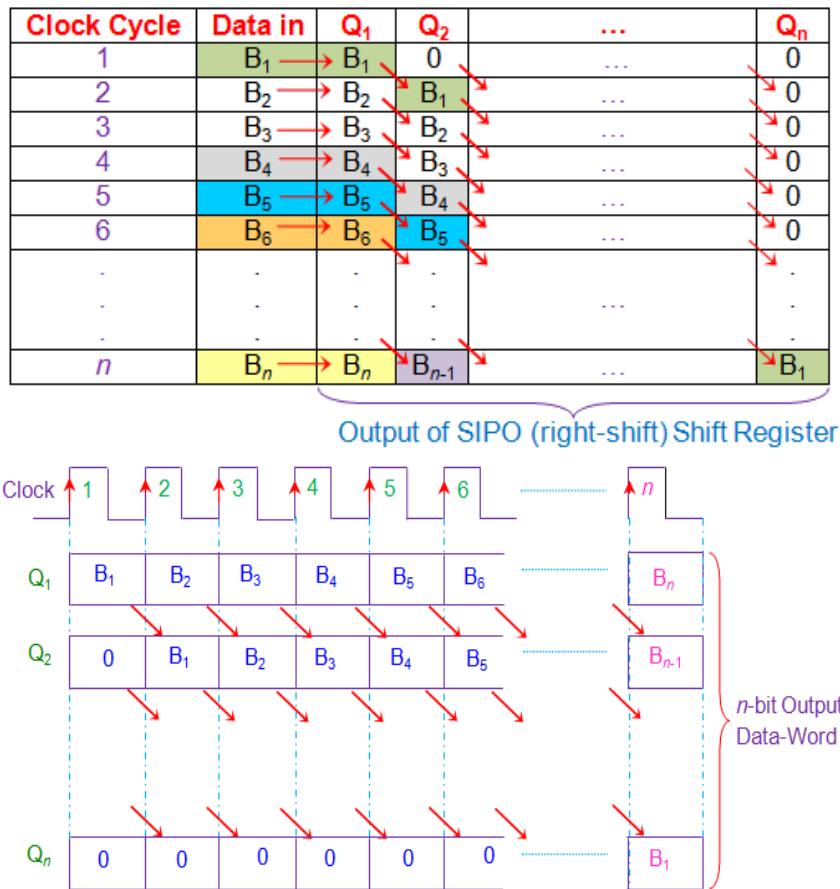


Figure 2 Output Waveform of n -bit Right-Shift SIPO Shift Register

In the right-shift SIPO shift-register, data bits shift from left to right for each clock pulse. However if the data bits are made to shift from right to left in the same design, one gets a left-shift SIPO shift-register as shown by figure 3. Nevertheless the basic working principle remains the same except the fact that now B_n down to B_1 is stored in Q_n down to Q_1 i.e. $Q_1 = B_1, Q_2 = B_2 \dots Q_n = B_n$ at the n th clock pulse.

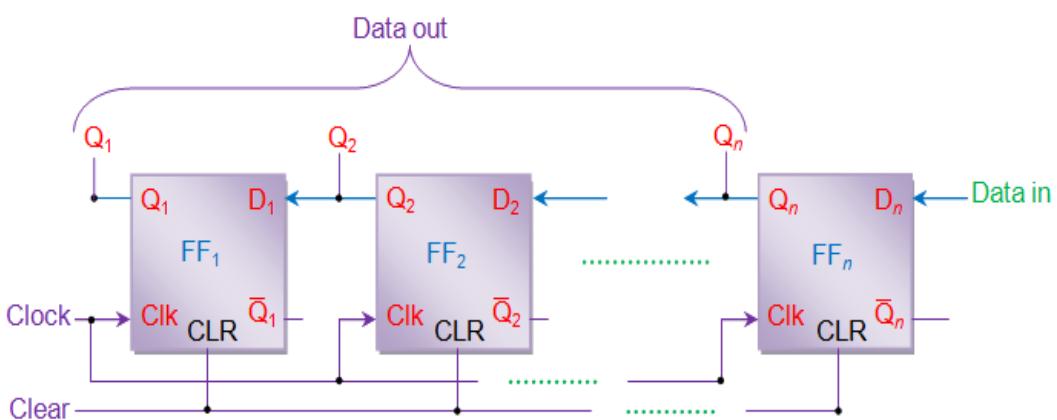


Figure 3 n -bit Serial-In Parallel-Out Left-Shift Shift Register

PRETEST:

Screenshot of a web browser showing a pretest quiz for a 4-bit serial-in parallel-out shift register. The quiz consists of several multiple-choice questions with radio buttons. The browser interface includes a search bar, pinned tabs for DE Journal and Virtual Labs, and a taskbar at the bottom.

Design and Verify the 4-Bit Serial In - Parallel Out Shift Registers.

A register that can be used to provide data movements _____

a: Parallel Register
 b: Simple Register
 c: Serial Register
 d: Shift Register

A n-bit register has a group of _____ flip-flops and some logic gates.

a: n
 b: p
 c: 10
 d: 01

Shifting a register content to left by one position is equivalent to _____

a: division by 2
 b: addition by 2
 c: multiplication by 2
 d: subtraction by 2

Shifting a register content to left by one position is equivalent to _____

a: division by 2
 b: addition by 2
 c: multiplication by 2
 d: subtraction by 2

To serially shift a nibble(4 bits) of data into a shift register, there must be _____

a: 1 clock pulse
 b: 8 clock pulses
 c: 4 clock pulses
 d: 1 clock pulse for each 1 in the data

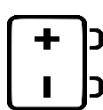
What is meant by parallel load of a shift register?

a: Parallel shifting of data
 b: Each flip-flop is loaded with data one at a time
 c: All flip-flops are preset with data
 d: None of the above

Submit Quiz

5 out of 5

PROCEDURE:



Step-1) Connect the supply(+5V) to the circuit.



Step-2) Keep the Reset and Preset as active-high signals



Step-3) Apply the data at data input

Step-4) Press clock pulse and observe this data at LED Q₃.

Step-5) Then press "ADD" button to add data in the given truth table.

Step-6) Apply the next data at data input.

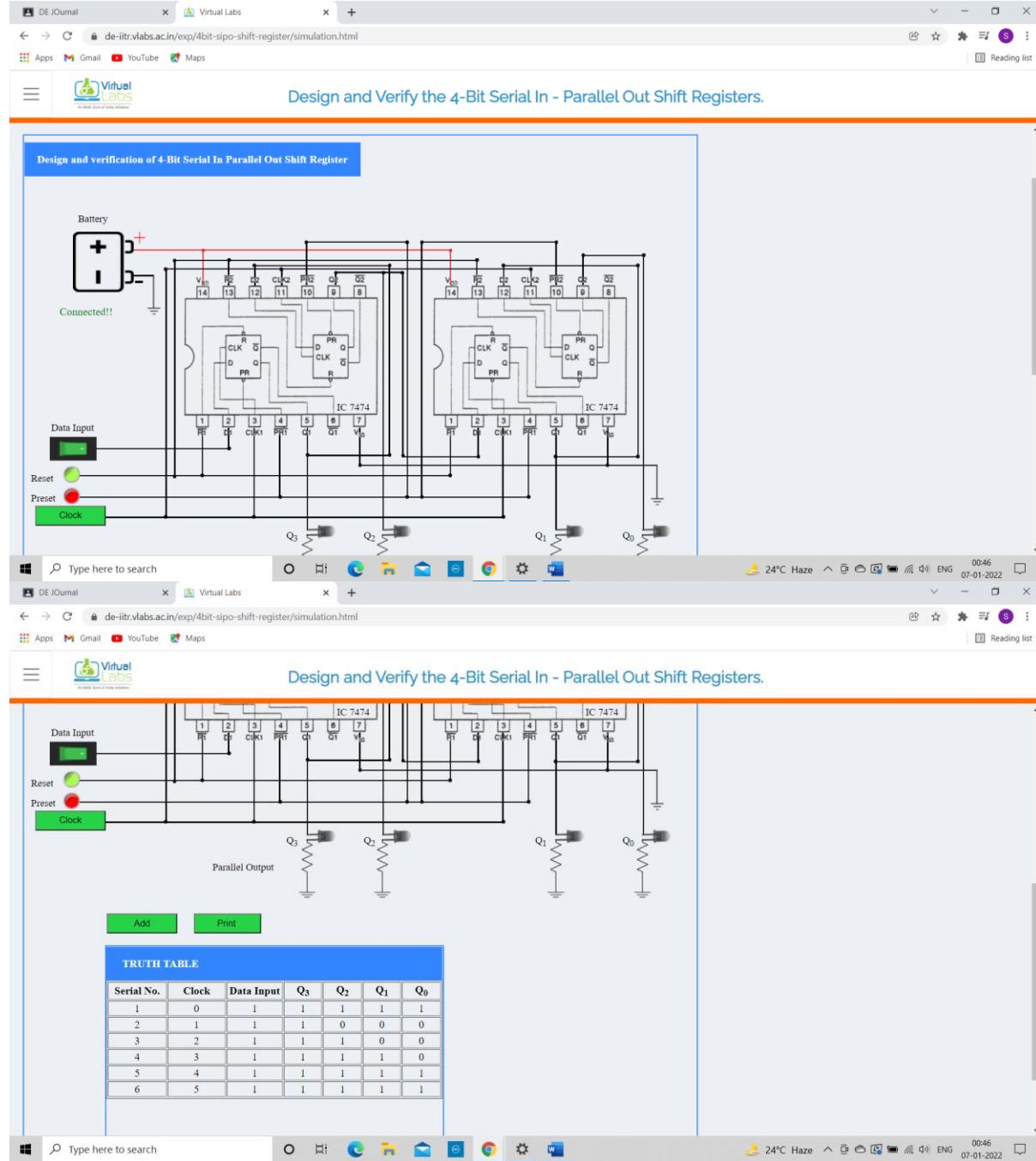
Step-7) Press clock pulse and observe that the data at LED Q₃ will shift to LED Q₂ and the

new data applied will appear at Q_3 .

Step-8) Repeat steps 3 to 5 till all the 4 bits appear at the output of shift register.

Step-9) Press the "Print" button after completing your simulation to get your results.

STIMULATION:



POSTTEST:

DE Journal **Virtual Labs**

de-itrlabs.ac.in/exp/4bit-sipo-shift-register/posttest.html

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Aim

Theory

Pretest

Procedure

Simulation

Posttest

References

Feedback

Design and Verify the 4-Bit Serial In - Parallel Out Shift Registers.

Based on how binary information is entered or shifted out, shift registers are classified into _____ categories.

a: 2
 b: 3
 c: 4
 d: 5

How can parallel data be taken out of a shift register simultaneously?

a: Use the Q output of the first FF
 b: Use the Q output of the last FF
 c: Tie all of the Q outputs together
 d: Use the Q output of each FF

The group of bits 11001 is serially shifted (right-most bit first) into a 5-bit parallel output shift register with an initial state 01110. After three clock pulses, the register contains _____

a: 01110
 b: 00001
 c: 00101
 d: 00101

DE Journal **Virtual Labs**

24°C Haze 00:47 07-01-2022

Type here to search

Apps Gmail YouTube Maps

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

The group of bits 11001 is serially shifted (right-most bit first) into a 5-bit parallel output shift register with an initial state 01110. After three clock pulses, the register contains _____

a: 01110
 b: 00001
 c: 00101
 d: 00101

The full form of SIPO is _____

a: Serial-in Parallel-out
 b: Parallel-in Serial-out
 c: Serial-in Serial-out
 d: Serial-In Peripheral-Out

A shift register that will accept a parallel input or a bidirectional serial load and internal shift features is called as?

a: Tristate
 b: End around
 c: Universal
 d: Conversion

Submit Quiz

5 out of 5

DE Journal **Virtual Labs**

24°C Haze 00:48 07-01-2022

Type here to search

Apps Gmail YouTube Maps

CONCLUSION:

Thus, 4-bit SIPO (serial in parallel out) shift register was implemented using IC-7474(D flip flop) and truth table was generated.