

EXPERIMENT - 1

AIM:-

Verification of the truth tables of TTL gates.

APPARATUS REQUIRED:-

Digital lab kit, single strand wires, breadboard, TTL IC's AND (IC-7408), OR (IC-7432), NAND (IC-7400), NOR (IC-7402), NOT (IC-7404) and NOR (IC-7486).

THEORY:-

Logic gates are idealized or physical devices implementing a Boolean function, which it performs a logical operation on one or more logical inputs and produce a single output. Depending on the context, the term may refer to an ideal logic gate, one that has for instance zero rise time and unlimited fan out or it may refer to an non-ideal physical device.

The main hierarchy is as follows:-

1. Basic Gates
2. Universal Gates
3. Advanced Gates

BASIC GATES

AND GATE:

The AND gate performs a logical multiplication commonly known as AND function. The output is high when both the inputs are high. The output is low level when any one of the inputs is low.

OR GATE:

The OR gate performs a logical addition commonly known as OR function. The output is high when any one of the inputs is high. The output is low level when both the inputs are low.

NOT GATE:

The NOT gate is called an inverter. The output is high when the input is low. The output is low when the input is high.

UNIVERSAL GATES

NAND GATE:

The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and any one of the input is low. The output is low level when both inputs are high.

NOR GATE:

The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

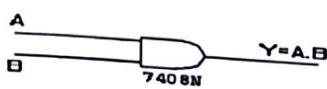
ADVANCED GATES

X-OR GATE:

The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

AND GATE:

SYMBOL:

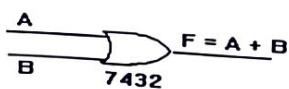


TRUTH TABLE

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE:

SYMBOL:



TRUTH TABLE

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE:

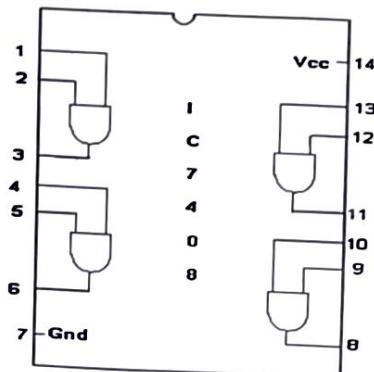
SYMBOL:



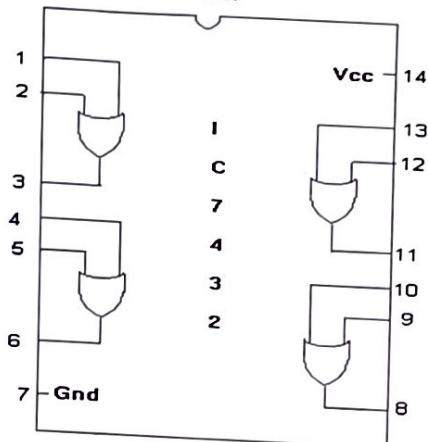
TRUTH TABLE :

A	\bar{A}
0	1
1	0

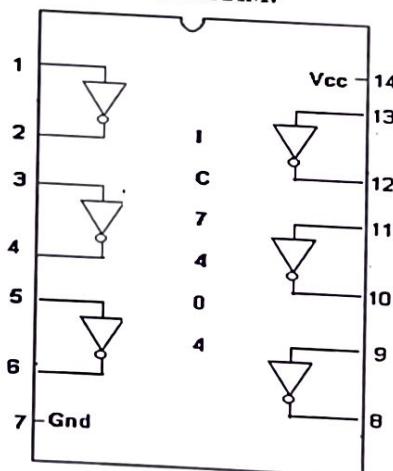
PIN DIAGRAM:



PIN DIAGRAM:



PIN DIAGRAM:



SYMBOL:



TRUTH TABLE

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE:

SYMBOL:



TRUTH TABLE

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE:

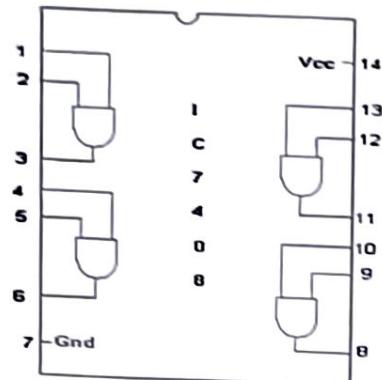
SYMBOL:



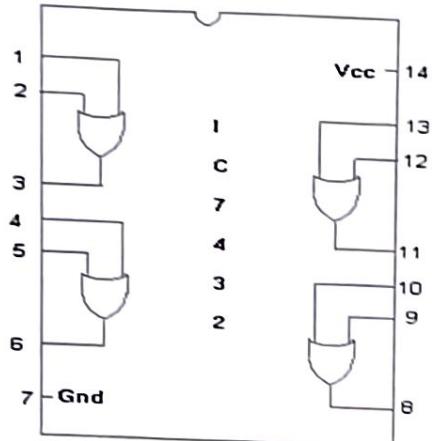
TRUTH TABLE :

A	A-bar
0	1
1	0

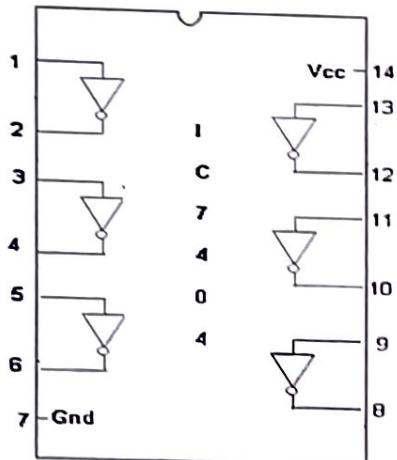
PIN DIAGRAM:



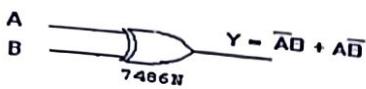
PIN DIAGRAM:



PIN DIAGRAM:



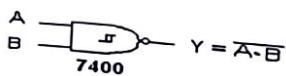
**XOR GATE:
SYMBOL:**



TRUTH TABLE :

A	B	$\overline{AB} + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

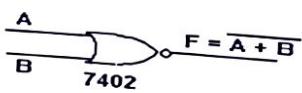
**2-INPUT NAND GATE:
SYMBOL:**



TRUTH TABLE

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

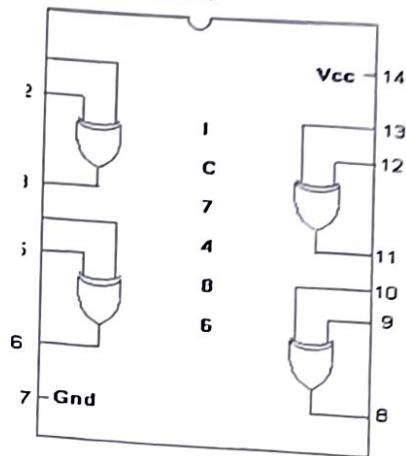
**NOR GATE:
SYMBOL:**



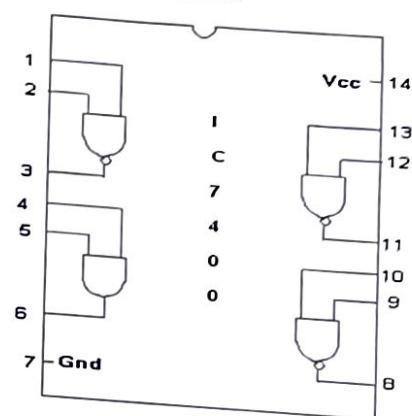
TRUTH TABLE

A	B	$\overline{A+B}$
0	0	1
0	1	1
1	0	1
1	1	0

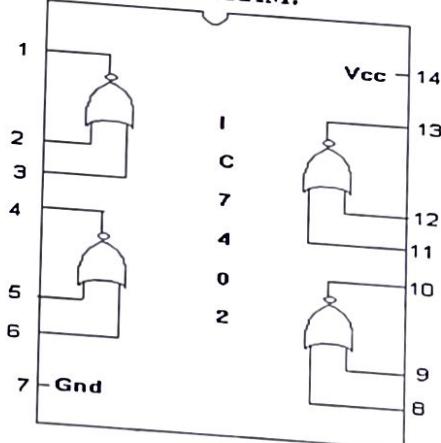
PIN DIAGRAM:



PIN DIAGRAM:



PIN DIAGRAM:



PROCEDURE:

- I. Place the breadboard gently on the observation table.
- II. Fix the IC which is under observation between the half shadow line of breadboard, so there is no shortage of voltage.
- III. Connect the wire to the main voltage source (V_{cc}) whose other end is connected to last pin of the IC (14 place from the notch).
- IV. Connect the ground of IC (7th place from the notch) to the ground terminal provided on the digital lab kit.
- V. Give the input at any one of the gate of the ICs i.e. 1st, 2nd, 3rd, 4th gate by using connecting wires. (In accordance to IC provided).
- VI. Connect output pins to the led on digital lab kit.
- VII. Switch on the power supply.
- VIII. If LED glows red then output is true, if it glows green output is false, which is numerically denoted as 1 and 0 respectively. The Color can change based on the IC manufacturer it's just verification of the Truth Table not the color change.

RESULT:

Thus the logic gates are studied and their truth tables were verified.

PRECAUTIONS:-

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

AIM:-**EXPERIMENT - 2**

Verify the NAND and NOR gates as universal logic gates.

APPARATUS REQUIRED:-

logic trainer kit, NAND gates (IC 7400), NOR gates (IC 7402), wires.

THEORY:-

NAND gate is actually a combination of two logic gates: 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; output is always one. By using only NAND gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NOR. So this gate is also called universal gate.

NAND gates as NOT gate:

A NOT produces complement of the input. It can have only one input, tie the inputs of a NAND gate together. Now it will work as a NOT gate. Its output is

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

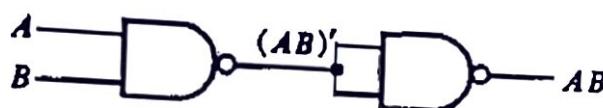


NOT (inverter)

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.

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



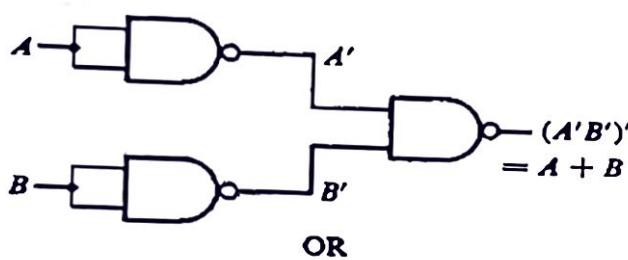
AND

NAND gates as OR gate:

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

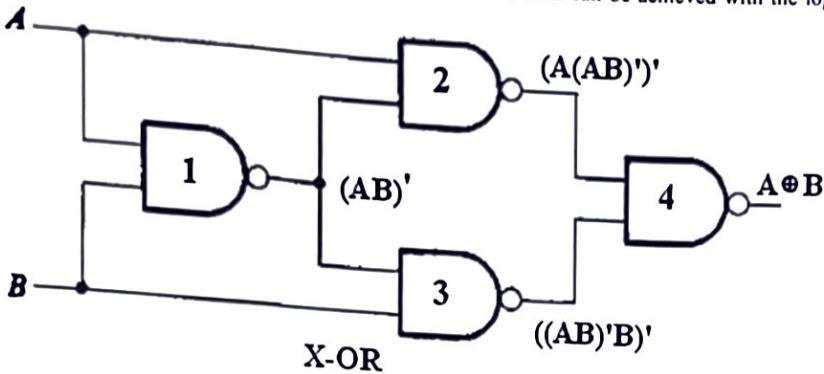
$$(A' \cdot B')' = A'' + B'' = A + B$$

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



NAND gates as X-OR gate:

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



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

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

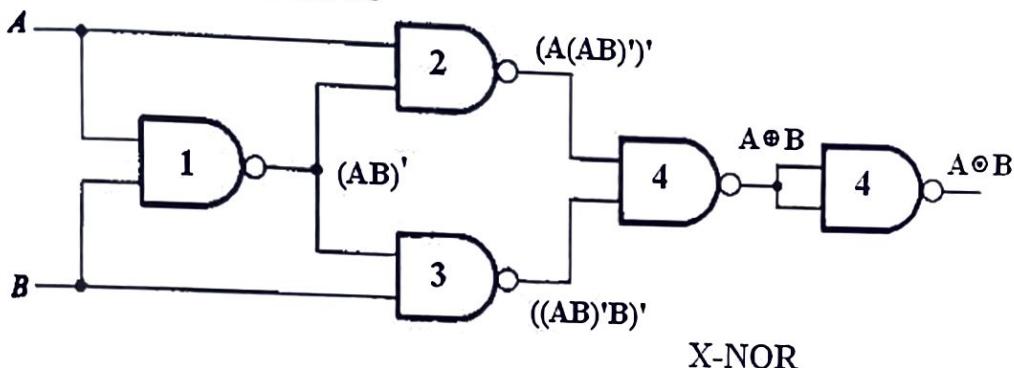
$$\begin{aligned}
 Y &= ((A(AB))'((AB)'))' \\
 &= (A(AB))'' + (B(AB))'' \\
 &= (A(AB))' + (B(AB))' \\
 &= (A(A' + B))' + (B(A' + B))' \\
 &= (AA' + AB') + (BA' + BB') \\
 &= (0 + AB' + BA' + 0)
 \end{aligned}$$

$$Y = AB' + BA'$$

NAND gates as X-NOR gate:

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

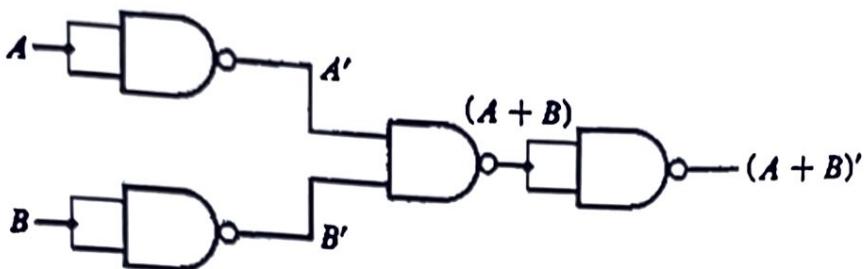
$$Y = AB + A'B'$$



NAND gates as NOR gate

A NOR gate is an OR gate followed by NOT gate. So connect the output of OR gate to a NOT gate, overall output is that of a NOR gate.

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



NOR

PROCEDURE:

1. Connect the trainer kit to ac power supply.
2. Connect the NAND gates for any of the logic functions to be realized.
3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply various input combinations and observe output for each one.
5. Verify the truth table for each input/ output combination.
6. Repeat the process for all logic functions.
7. Switch off the power supply.

THEORY:

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, X-OR, X-NOR, NAND. So this gate is also called universal gate.

NOR gates as NOT gate:

A NOT produces complement of the input. It can have only one input, tie the inputs of a NOR gate together. Now it will work as a NOT gate. Its output is

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

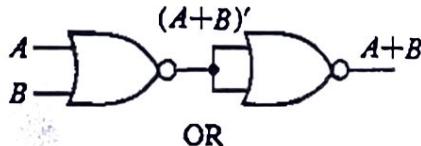


NOT

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.

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



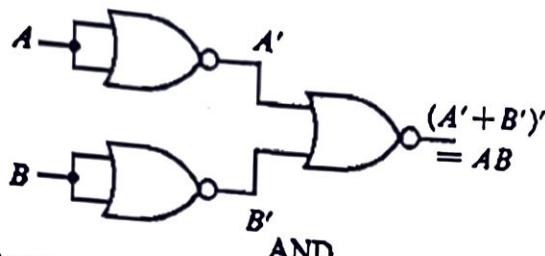
OR

NOR gates as AND gate:

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

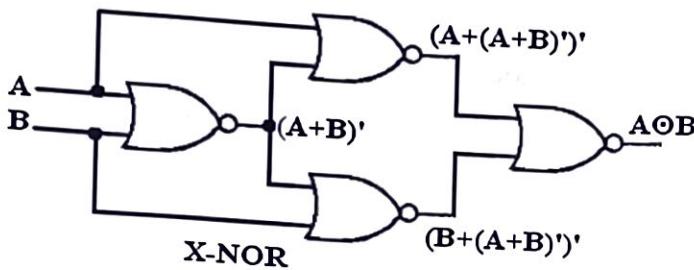
$$(A'+B')' = A''B'' = AB$$

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



NOR gates as X-NOR gate:

The output of a two input X-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

	Inputs	Output
1	A, B	$(A + B)'$
2	$A, (A + B)'$	$(A + (A+B))'$
3	$(A + B)', B$	$(B + (A+B))'$
4	$(A + (A + B))', (B + (A+B))'$	$AB + A'B'$

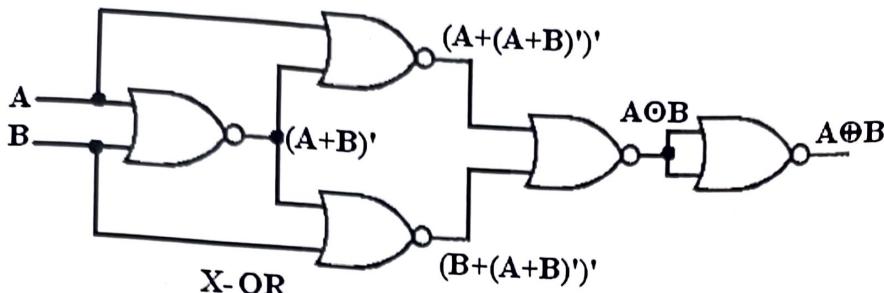
Now the output from gate no. 4 is the overall output of the configuration.

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

NOR gates as X-OR gate:

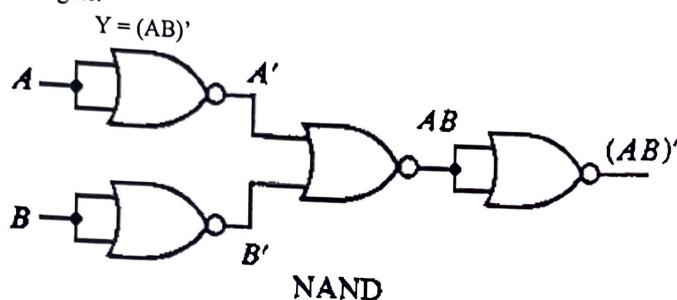
X-OR gate is actually X-NOR gate followed by NOT gate. So give the output of X-NOR gate to a NOT gate, overall output is that of an X-OR gate.

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



NOR gates as NAND gate:

A NAND gate is an AND gate followed by NOT gate. So connect the output of AND gate to a NOT gate, overall output is that of a NAND gate.



PROCEDURE:

1. Connect the trainer kit to ac power supply.
2. Connect the NOR gates for any of the logic functions to be realised.
3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply various input combinations and observe output for each one.
5. Verify the truth table for each input/ output combination.
6. Repeat the process for all logic functions.
7. Switch off the ac power supply.

RESULT:

NAND & NOR are verified as universal gates successfully.

PRECAUTIONS:-

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

EXPERIMENT - 3

AIM:-

Design and verification of the truth tables of Half and Full adder circuits.

APPARATUS REQUIRED:-

logic trainer kit, NAND gates (IC 7400), XOR gates (IC 7486), AND gates (IC 7408), wires.

THEORY:

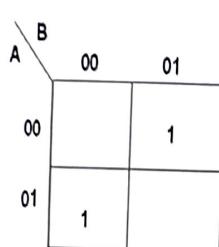
half adder:

A half adder has two inputs for the two bits to be added and two outputs one from the sum 'S' and other from the carry 'C' into the higher adder position. Above circuit is called as a carry signal from the addition of the less significant bits sum from the X-OR Gate the carry out from the AND gate.

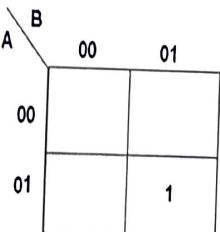
Truth Table for Half Adder

INPUT		OUTPUT	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

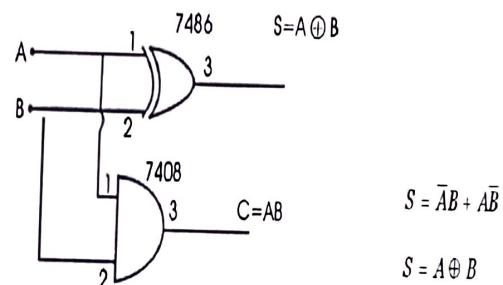
K-Map for SUM:



$$\text{SUM} = A'B + AB'$$



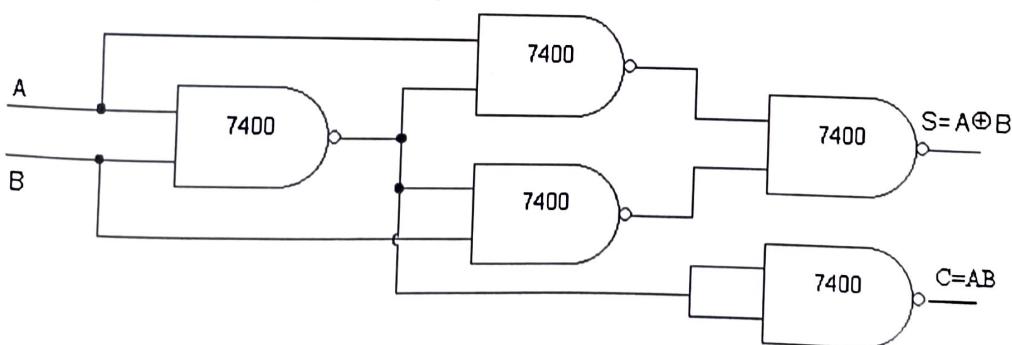
$$\text{CARRY} = AB$$



$$S = A \oplus B$$

$$C = AB$$

Half Adder using NAND gates only:-

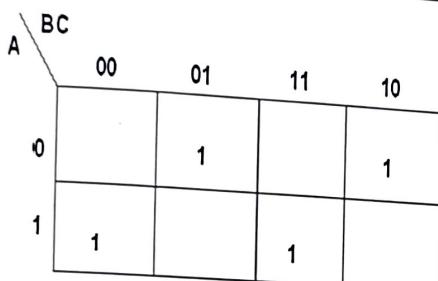


Full Adder:

A full adder is a combinational circuit that forms the arithmetic sum of input; it consists of three inputs and two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR Gate, carry output will be taken from OR Gate.

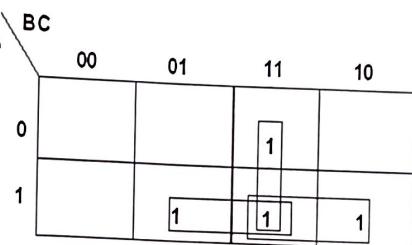
INPUT			OUTPUT	
A	B	C	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

K-Map for SUM:-



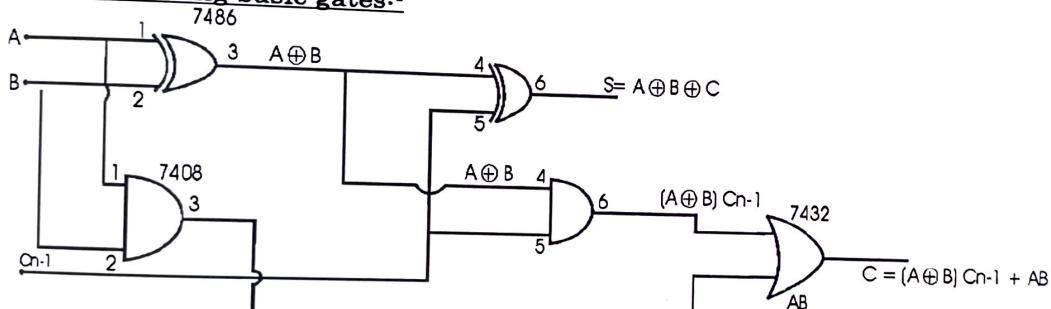
$$\text{SUM} = A'B'C + A'BC' + ABC' + ABC$$

K-Map for CARRY:-

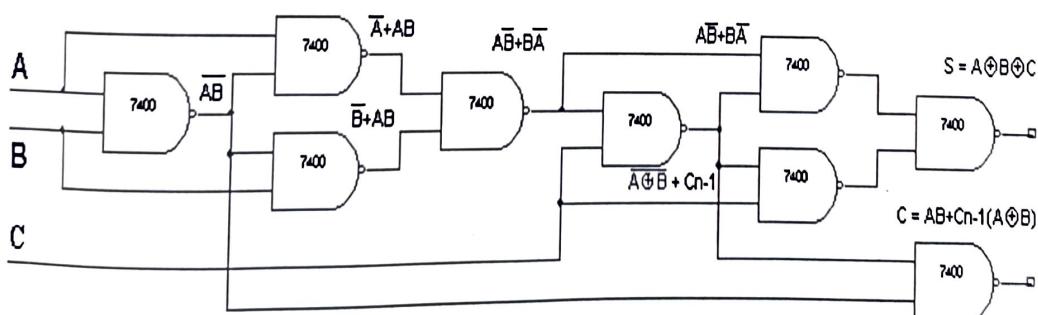


$$\text{CARRY} = AB + BC + AC$$

Full Adder using basic gates:-



Full Adder using NAND gates only:-



PROCEDURE:

- (i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

RESULT:

Thus the half adder & full adder was designed and their truth table is verified.

PRECAUTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

EXPERIMENT - 4

AIM:-

Design and verification of the truth tables of Half and Full Subtractor circuits.

APPARATUS REQUIRED:-

logic trainer kit, NAND gates (IC 7400), XOR gates (IC 7486), AND gates (IC 7408), NOT gates (IC 7404), connecting wires.

THEORY:

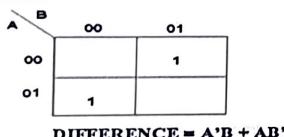
Half Subtractor:

The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

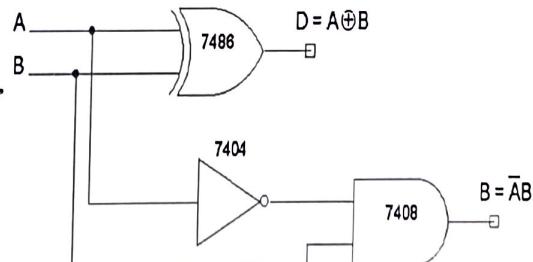
Truth Table for Half Subtractor

INPUT		OUTPUT	
A	B	Bo	Diff
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

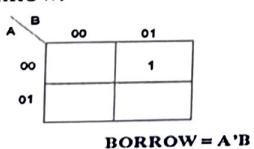
K-Map for DIFFERENCE:



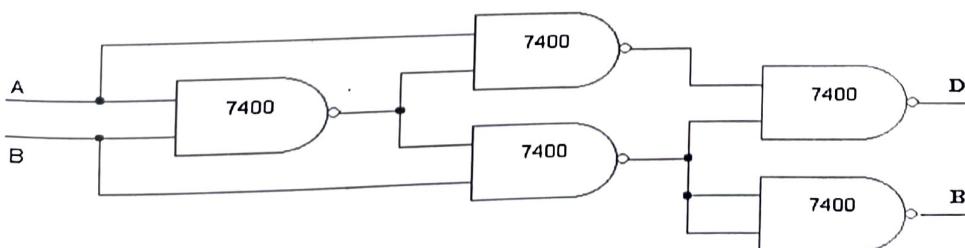
Using X – OR and Basic Gates (a) Half Subtractor



K-Map for BORROW:



(ii) Using only NAND gates (a) Half subtractor



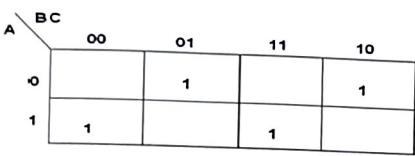
Full Subtractor:

The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs. The two half subtractor put together gives a full subtractor. The first half subtractor will be C and A-B. The output will be difference output of full subtractor. The expression AB assembles the borrow output of the half subtractor and the second term is the inverted difference output of first X-OR.

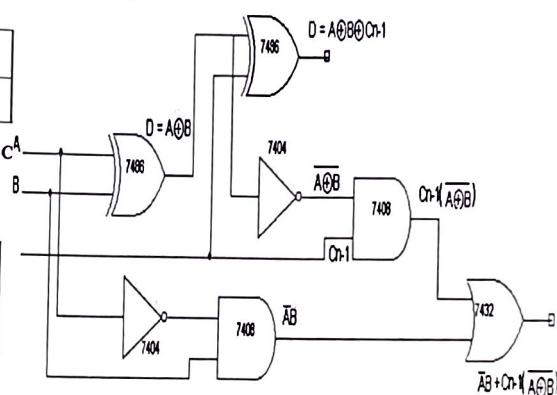
Truth Table for Full Subtractor

INPUT			OUTPUT	
A	B	C	Bo	Diff
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

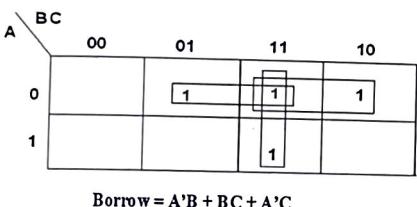
K-Map for Difference:



Full Subtractor

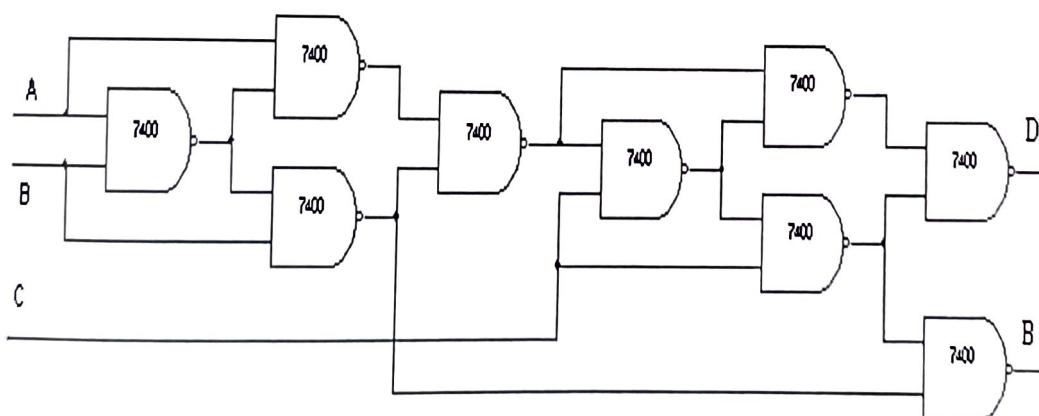


K-Map for Borrow:



$$\text{Borrow} = A'B + BC + A'C$$

Using Only NAND Gates (Full Subtractor)



PROCEDURE:

- (i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

RESULT:

Thus the half subtractor and full subtractor was designed and their truth table is verified.

PRECAUTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

AIM:-

EXPERIMENT - 5

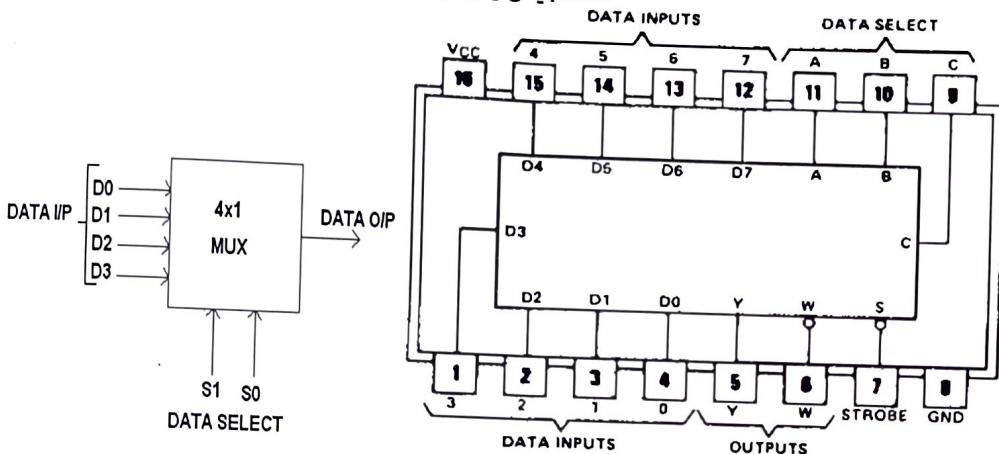
Verification of the truth table of the Multiplexer 74150.

APPARATUS REQUIRED:-

logic trainer kit, IC- 74150, wires.

THEORY:

A Multiplexer (or a data selector) is a logic circuit that accepts several data inputs and allows only one of them at a time to get through to the output. The selection of the desired data input is controlled by the SELECT (or ADDRESS) INPUTS. Multiplexer means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input lines and n selection lines whose bit combination determine which input is selected. Figure below shows the block diagram of a Multiplexer.



In this diagram the inputs and outputs are indicated by means of broad arrows to indicate that there may be one or more lines. Depending upon the digital code applied at the SELECT inputs, one out of the data sources is selected and transmitted to the single output channel. The Multiplexer becomes enabled when the strobe signal is active LOW. The pin out of a 8:1 multiplexer IC 74150 is shown above. The output of this circuit is the inverted input. This is a 16-pin DIP.

Function Table

INPUT		OUTPUT
S1	S0	Y
0	0	D0 = D0 S1' S0'
0	1	D1 = D1 S1' S0
1	0	D2 = D2 S1 S0'
1	1	D3 = D3 S1 S0
Y = D0 S1' S0' + D1 S1' S0 + D2 S1 S0' + D3 S1 S0		

EXPERIMENT - 5

AIM:-

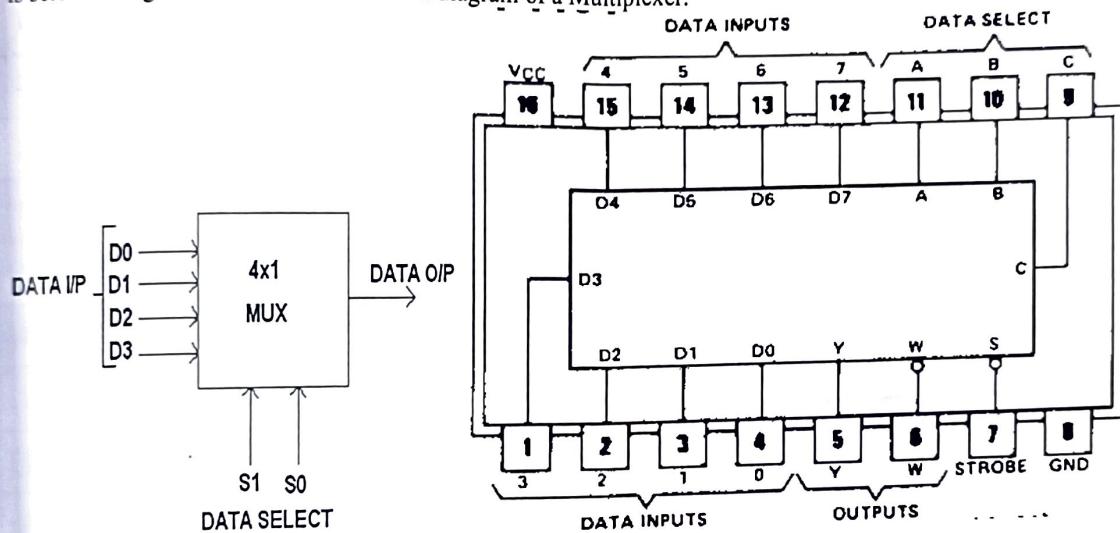
Verification of the truth table of the Multiplexer 74150.

APPARATUS REQUIRED:-

logic trainer kit, IC- 74150, wires.

THEORY:

A Multiplexer (or a data selector) is a logic circuit that accepts several data inputs and allows only one of them at a time to get through to the output. The selection of the desired data input is controlled by the SELECT (or ADDRESS) INPUTS. Multiplexer means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input lines and n selection lines whose bit combination determine which input is selected. Figure below shows the block diagram of a Multiplexer.



In this diagram the inputs and outputs are indicated by means of broad arrows to indicate that there may be one or more lines. Depending upon the digital code applied at the SELECT inputs, one out of the data sources is selected and transmitted to the single output channel. The Multiplexer becomes enabled when the strobe signal is active LOW. The pin out of a 8:1 multiplexer IC 74150 is shown above. The output of this circuit is the inverted input. This is a 16-pin DIP.

Function Table

INPUT		OUTPUT
S1	S0	Y
0	0	$D_0 = D_0 S_1' S_0'$
0	1	$D_1 = D_1 S_1' S_0$
1	0	$D_2 = D_2 S_1 S_0'$
1	1	$D_3 = D_3 S_1 S_0$

$Y = D_0 S_1' S_0' + D_1 S_1' S_0 + D_2 S_1 S_0' + D_3 S_1 S_0$

C314

30

PIN DIAGRAM:

E7	1
E6	2
E5	3
E4	4
E3	5
E2	6
E1	7
E0	8
ST	9
Q	10
D	11
GND	12

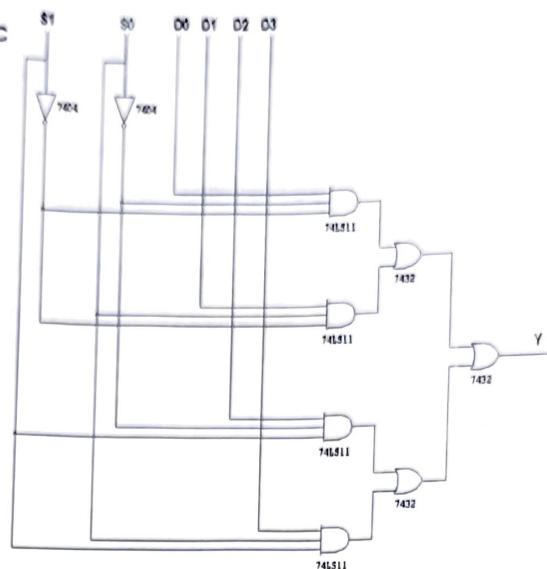
C

1 5 0

2 4 1

3

CIRCUIT DIAGRAM FOR MULTIPLEXER:



Truth Table

INPUT	OUTPUT
S1 S0	Y
0 0	D0
0 1	D1
1 0	D2
1 1	D3

PROCEDURE:-

- 1) Assemble the circuit on bread board, as per above diagram.
- 2) Give the logical inputs and check for the proper output, as per the truth table.

CONCLUSION:

Hence verified the Multiplexer (8:1) operation using IC-74150

PRECAUTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

~~C O D E~~
~~Z~~

EXPERIMENT - 6

AIM:-

Verification of the truth table of the De-Multiplexer 74154.

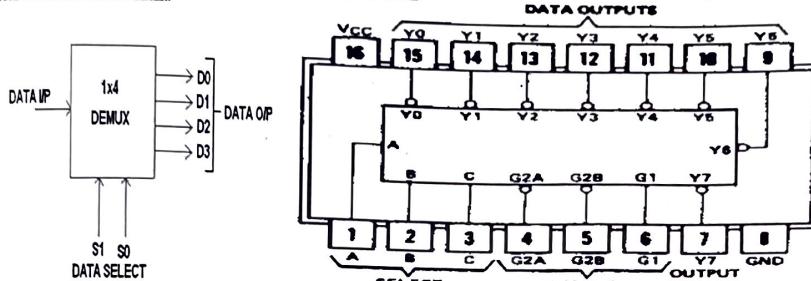
APPARATUS REQUIRED:-

Logic trainer kit, IC- 74154, wires.

THEORY:

A Demultiplexer performs the reverse operation of a Multiplexer. It accepts a single input and distributes it over several outputs. The SELECT input code determines to which output the data input will be transmitted. The Demultiplexer becomes enabled when the strobe signal is active LOW. This circuit can also be used as binary-to-decimal decoder with binary inputs applied at the select input lines and the output will be obtained on the corresponding line. These devices are available as 2-line-to-4-line decoder, 3-line-to-8-line decoder, 4-line-to-16-line decoder. The output of these devices is active LOW. Also there is an active low enable/data input terminal available. Figure below shows the block diagram of a Demultiplexer.

BLOCK DIAGRAM FOR 1x4 DEMULTIPLEXER:



PIN CONFIGURATION:

FUNCTION TABLE:

S1	S0	INPUT
0	0	D0 = X S1' S0
0	1	D1 = X S1' S0
1	0	D2 = X S1 S0'
1	1	D3 = X S1 S0

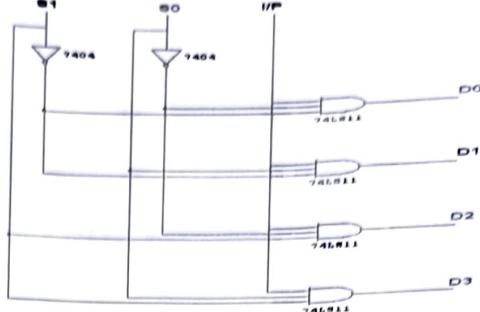
$$Y = X S1' S0 + X S1' S0 + X S1 S0' + X S1 S0$$

In this diagram the inputs and outputs are indicated by means of broad arrows to indicate that there may be one or more lines. Depending upon the digital code applied at the SELECT inputs, one data is transmitted to the single output channel out of many. The pin out of a 16:1 Demultiplexer IC 74154 is shown above. The output of this circuit is active low. This is a 24-pin DIP.

PIN DIAGRAM:



CIRCUIT DIAGRAM:



TRUTH TABLE

INPUT			OUTPUT			
S ₁	S ₀	I/P	D ₀	D ₁	D ₂	D ₃
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

PROCEDURE:-

- 1) Assemble the circuit on bread board, as per above Pin diagram.
- 2) Give the logical inputs and check for the proper output, as per the truth table.

CONCLUSION:

Hence verified the Demultiplexer (16:1) operation using IC-74154.

PRECAUTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

EXPERIMENT - 7

AIM:-

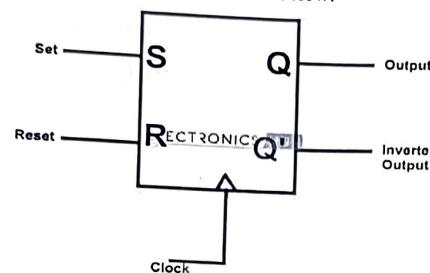
Design and test of an S-R flip-flop using NOR/NAND gates.

APPARATUS REQUIRED:-

Logic trainer kit, NAND gates ICs- 7400, NOR gates ICs-7402, wires.

THEORY:

The SR flip-flop is one of the fundamental parts of the sequential circuit logic. SR flip-flop is a memory device and a binary data of 1-bit can be stored in it. SR flip-flop has two stable states in which it can store data in the form of either binary zero or binary one. Like all flip-flops, an SR flip-flop is also an edge sensitive device. SR flip-flop is one of the most vital components in digital logic and it is also the most basic sequential circuit that is possible. The S and R in SR flip-flop means 'SET' and 'RESET' respectively. Hence it is also called Set-Reset flip-flop. The symbolic representation of the SR Flip Flop is shown below.



Working Principle:

SR flip-flop works during the transition of clock pulse either from low to high or from high to low (depending on the design) i.e. it can be either positive edge triggered or negative edge triggered.

For a positive edge triggered SR flip-flop, suppose, if S input is at high level (logic 1) and R input is at low level (logic 0) during a low to high transition on clock pulse, then the SR flip-flop is said to be in SET state and the output of the SR flip-flop is SET to 1.

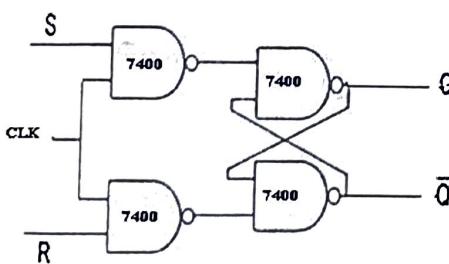
1. For the same clock situation, if the R input is at high level (logic 1) and S input is at low level (logic 0), then the SR flip-flop is said to be in RESET state and the output of the SR flip-flop is RESET to 0.

The SR flip-flops can be designed by using logic gates like NOR gates and NAND gates.

S-R Flip-Flop Using NAND Gate

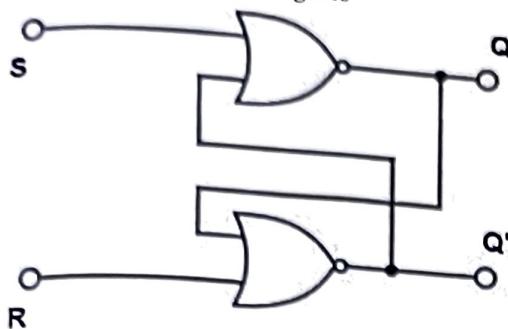
SR flip flop can be designed by cross coupling of two NAND gates. It is an active low input SR flip-flop. The circuit of SR flip-flop using NAND gates is shown in below figure:

R-S flip-flop using NAND gates



\bar{S}	\bar{R}	Q	State
1	1	Previous State	No change
1	0	0	Reset
0	1	1	Set
0	0	?	Forbidden

R - S flip-flop using NOR gates



S	R	Q	State
0	0	Previous State	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden

PROCEDURE:

- Connections are given as per circuit diagram.
- Logical inputs are given as per circuit diagram.
- Observe the output and verify the truth table.

RESULT:

Design of S-R Flip flop using NAND & NOR gates was verified successfully.

PRECAUTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

EXPERIMENT - 8

AIM:-

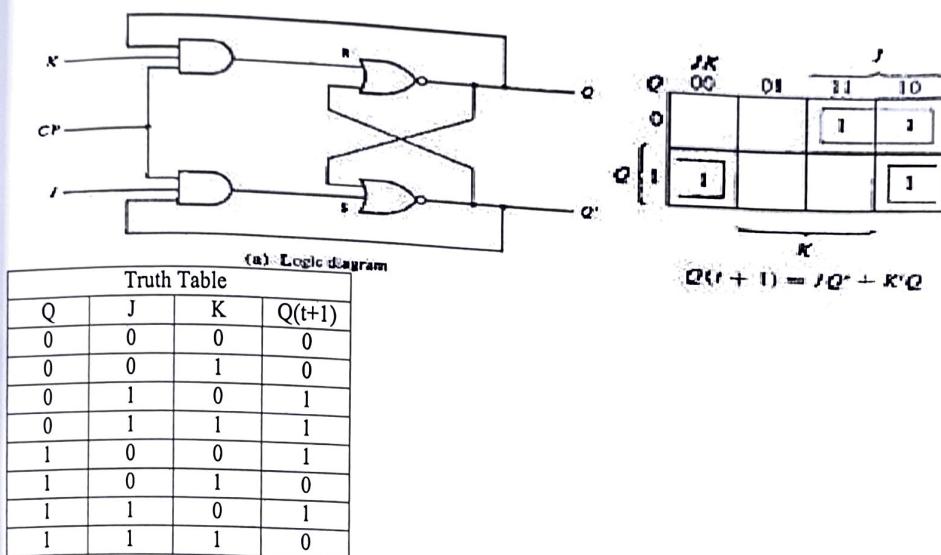
Verify the truth table of a J-K flip-flop (7476)

APPARATUS REQUIRED:-

Logic trainer kit, Flip-flop ICs- 7476, wires.

THEORY:

The JK flip-flop is the modified version of SR flip-flop with no invalid state; i.e. the state $J=K=1$ is not forbidden. It works such that J serves as set input and K serves as reset. The only difference is that for the combination $J=K=1$ this flip-flop now performs an action: it inverts its state.



The flip-flop is constructed in such a way that the output Q is ANDed with K and CP. This arrangement is made so that the flip-flop is cleared during a clock pulse only if Q was previously 1. Similarly, Q' is ANDed with J and CP, so that the flip-flop is cleared during a clock pulse only if Q' was previously 1.

When $J=K=0$

When both J and K are 0, the clock pulse has no effect on the output and the output of the flip-flop is the same as its previous value. This is because when both the J and K are 0, the output of their respective AND gate becomes 0.

When $J=0, K=1$

When J=0, the output of the AND gate corresponding to J becomes 0 (i.e.) S=0 and R=1. Therefore, Q' becomes 0. This condition will reset the flip-flop. This represents the RESET state of Flip-flop.

When $J=1, K=0$

In this case, the AND gate corresponding to K becomes 0 (i.e.) S=1 and R=0. Therefore, Q becomes 0. This condition will set the Flip-flop. This represents the SET state of Flip-flop.

When $J=K=1$

EXPERIMENT - 8

AIM:-

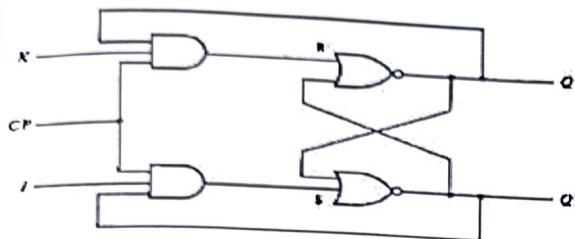
Verify the truth table of a J-K flip-flop (7476)

APPARATUS REQUIRED:-

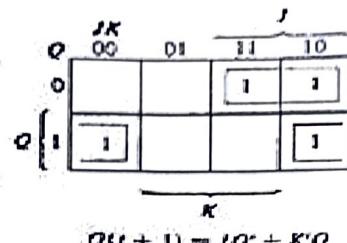
Logic trainer kit, Flip-flop ICs- 7476, wires.

THEORY:

The JK flip-flop is the modified version of SR flip-flop with no invalid state; i.e. the state $J=K=1$ is not forbidden. It works such that J serves as set input and K serves as reset. The only difference is that for the combination $J=K=1$ this flip-flop; now performs an action: it inverts its state.



(a) Logic diagram



Truth Table			
Q	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

The flip-flop is constructed in such a way that the output Q is ANDed with K and CP. This arrangement is made so that the flip-flop is cleared during a clock pulse only if Q was previously 1. Similarly, Q' is ANDed with J and CP, so that the flip-flop is cleared during a clock pulse only if Q' was previously 1.

When $J = K = 0$

When both J and K are 0, the clock pulse has no effect on the output and the output of the flip-flop is the same as its previous value. This is because when both the J and K are 0, the output of their respective AND gate becomes 0.

When $J=0, K=1$

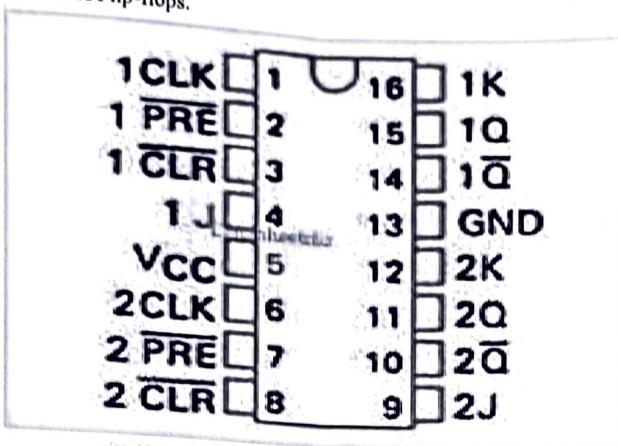
When $J=0$, the output of the AND gate corresponding to J becomes 0 (i.e.) S=0 and R=1. Therefore, Q' becomes 0. This condition will reset the flip-flop. This represents the RESET state of Flip-flop.

When $J=1, K=0$

In this case, the AND gate corresponding to K becomes 0 (i.e.) S=1 and R=0. Therefore, Q becomes 0. This condition will set the Flip-flop. This represents the SET state of Flip-flop.

When $J=K=1$

Consider the condition of $CP=1$ and $J=K=1$. This will cause the output to complement again and again. This complement operation continues until the Clock pulse goes back to 0. Since this condition is undesirable, we have to find a way to eliminate this condition. This undesirable behavior can be eliminated by Edge triggering of JK flip-flop or by using master slave JK Flip-flops.



Pin diagram of IC 7476

OBSERVATION TABLE:

Inputs					Theoretical Outputs	Experimental Output
J	K	CLK	PRE	CLR	Q & Condition	Q & Condition
X	X	L	0	0		
X	X	L	0	1		
X	X	L	1	0		
X	X	L	1	1		
0	0	L	1	1		
0	1	L	1	1		
1	0	L	1	1		
1	1	L	1	1		

PROCEDURE:

- (i) Connections are given as per circuit diagram.
- (ii) Logical inputs are given as per circuit diagram.
- (iii) Observe the output and verify the truth table.

RESULT:

Thus the J-K Flip flop was designed and truth table is verified.

PRECAUTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

AIM:-

Verify the truth table of a D flip-flop (7474)

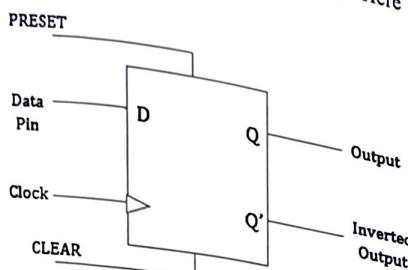
EXPERIMENT - 9

APPARATUS REQUIRED:-

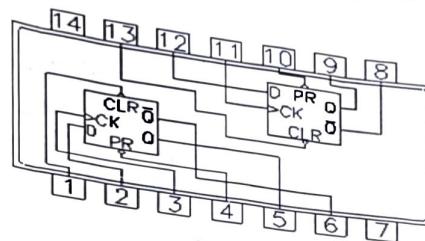
Logic trainer kit, Flip-flop ICs- 7474, wires.

THEORY:

D flip flop also called as delay flip flop where it can be used to introduce a delay in the digital circuit by changing the propagation delay of the flip flop. Here the input data bit at D will reflects at the output after a certain propagation delay.



Symbol: D Flip-flop



7474
Dual D Flip-Flop
with Preset and Clear

Truth Table of D Flip-Flop			Characteristic Table of D Flip-Flop		
Clock	INPUT	OUTPUT	D	Q	Q'
	D	Q	0	0	0
LOW	X	0	0	0	0
HIGH	0	0	1	0	0
HIGH	1	1	1	1	1

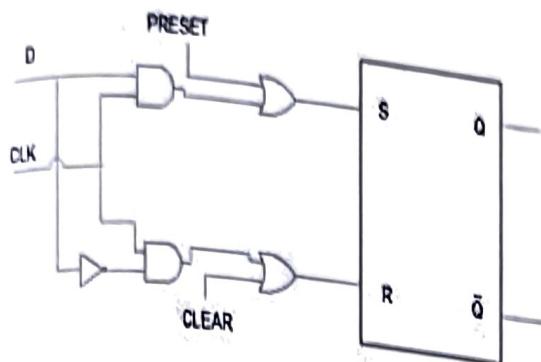
Characteristic Equation:

$$\begin{aligned} Q' &= D \bar{Q}' + \bar{D} Q \\ Q &= D \end{aligned}$$

D Flip Flop with PRESET and CLEAR

PRESET is the input to the D flip flop which sets the output data to High i.e. 1. and CLEAR is also an input which clears the output data or output state. A high PRESET forces Q to 1; a high CLEAR resets Q to 0. Figure shows clocked flip flop with PRESET and CLEAR inputs.

Clocked D Flip-Flop with PRESET and CLEAR



In the above circuit irrespective of the AND gates out, if the PRESET input is high the OR gate out directly sets the S input which makes Q to 1 and in the same way if CLEAR input is high it resets the Q to 0.

PROCEDURE:

- (i) Connections are given as per circuit diagram.
 - (ii) Logical inputs are given as per circuit diagram.
 - (iii) Observe the output and verify the truth table.

RESULT:

Thus the D Flip flop was designed and their truth table is verified.

PRECAUTIONS:

- All connections should be made neat and tight.
 - Digital lab kits and ICs should be handled with utmost care.
 - While making connections main voltage should be kept switched off.
 - Never touch live and naked wires.

EXPERIMENT - 10

AIM:-

Operate the counters 7490, 7493.

APPARATUS REQUIRED:-

Logic trainer kit, Counter ICs- 7490, IC - 7493 wires.

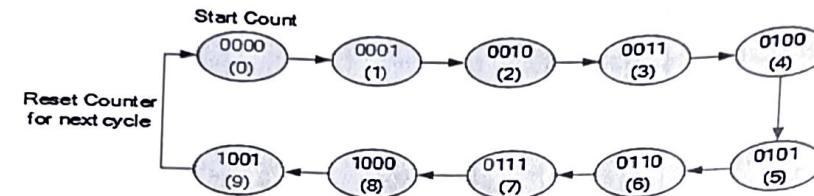
THEORY:

Asynchronous 74LS90 Decade Counter

Digital counters count upwards from zero to some pre-determined count value on the application of a clock signal. Once the count value is reached, resetting them returns the counter back to zero to start again.

A decade counter counts in a sequence of ten and then returns back to zero after the count of nine. Obviously to count up to a binary value of nine, the counter must have at least four flip-flops within its chain to represent each decimal digit as shown.

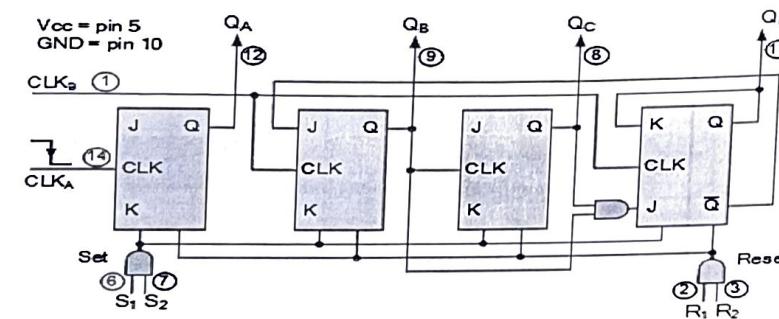
BCD Counter State Diagram



The 74LS90 BCD Counter

The 74LS90 integrated circuit is basically a MOD-10 decade counter that produces a BCD output code. The 74LS90 consists of four master-slave JK flip-flops internally connected to provide a MOD-2 (count-to-2) counter and a MOD-5 (count-to-5) counter. The 74LS90 has one independent toggle JK flip-flop driven by the CLK A input and three toggle JK flip-flops that form an asynchronous counter driven by the CLK B input as shown.

74LS90 BCD Counter



The counter's four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD counter circuit's code. So for example, Q_A, Q_B, Q_C and Q_D. The 74LS90 counting sequence is triggered on the negative going edge of the clock signal, that is when the clock signal CLK goes from logic 1 (HIGH) to logic 0 (LOW).

The additional input pins R_1 and R_2 are counter "reset" pins while inputs S_1 and S_2 are "set" pins. When connected to logic 1, the Reset inputs R_1 and R_2 reset the counter back to zero, 0 (0000), and when the Set inputs S_1 and S_2 are connected to logic 1, they Set the counter to maximum, or 9 (1001) regardless of the actual count number or position.

PROCEDURE:

- (i) Connections are given as per circuit diagram.
- (ii) Logical inputs are given as per circuit diagram.
- (iii) Observe the output and verify the truth table.

RESULT:

Thus the Counters were designed and their truth table is verified.

PREGATIOMS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

AIM:-**EXPERIMENT - 11**

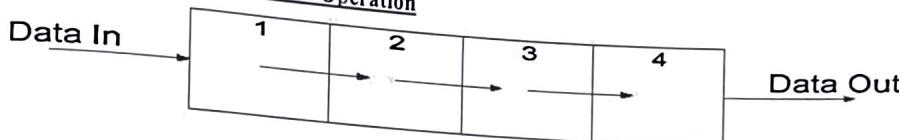
Design of 4-bit shift register (shift right).

APPARATUS REQUIRED:-

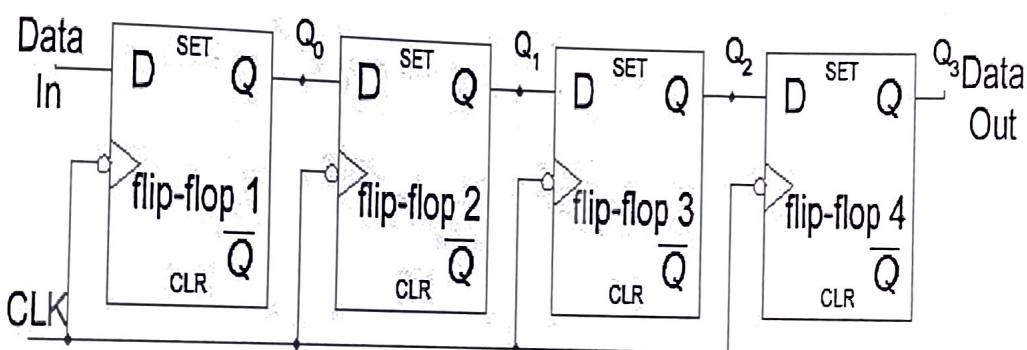
Logic trainer kit, D Flip-flop IC - 7474 wires.

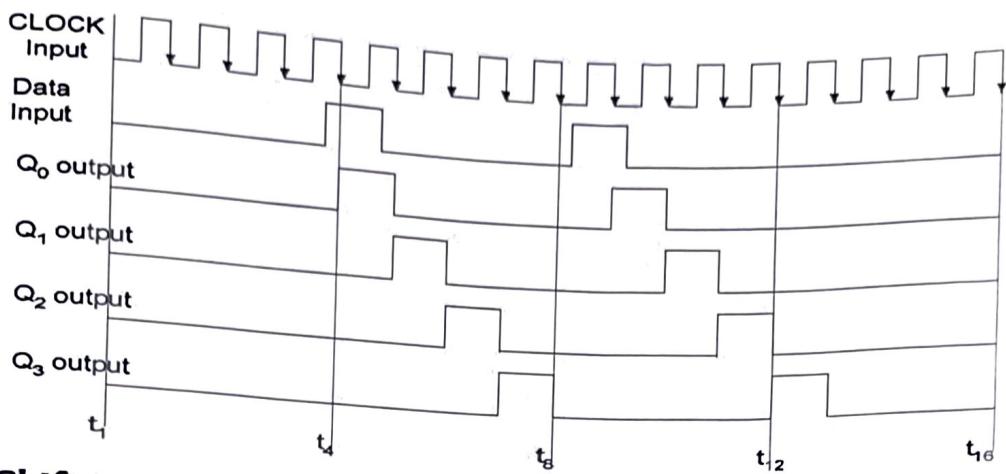
THEORY:**Serial In/Shift Right/Serial Out Operation**

Data is shifted in the right hand direction one bit at a time with each transition of the clock signal. The data enters the shift register serially from the left hand side and after four clock transitions the 4-bit registers has 4-bits of data. The data is shifted out serially one bit at a time from the right hand side of the register if clock signals are continuously applied. Thus after 8 clock signals the 4-bit data is completely shifted out of the shift register.

Serial In/Serial Right/Serial Out Operation

Serial shift registers can be implemented using any type of flip-flops. A serial shift register implemented using D flip-flops with the serial data applied at the D input of the first flip-flop and serial data out obtained at the Q output of the last flip-flop is shown in figure. At each clock transition 1 bit of serial data is shifted in and at the same instant 1-bit of serial data is shifted out. For a 4-bit shift register, 8 clock transitions are required to shift in 4-bit data and completely shift out the 4-bit data. As the data shifted out 1-bit at a time, a logic 0 value is usually shifted in to fill up the vacant bits in the shift register.

Serial In/Shift Right/Serial Out Register**Timing diagram of a Serial In/Shift Right/Serial Out Register**



Shift Register Truth Table

Outputs	Q_0	Q_1	Q_2	Q_3
Reset	0	0	0	0
CK Pulse 1	1	0	0	0
CK Pulse 2	0	1	0	0
CK Pulse 3	0	0	1	0
CK Pulse 4	0	0	0	1

PROCEDURE:

- (i) Connections are given as per circuit diagram.
- (ii) Logical inputs are given as per circuit diagram.
- (iii) Observe the output and verify the truth table.

RESULT:

Thus the Shift register was designed and their truth table is verified.

PREGACTIONS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.

Describe
Discuss in
Compartis
While sho
What do I
What do I

AIM:-

EXPERIMENT - 12

Design of modulo-4 counter using J K flip flop.

APPARATUS REQUIRED:-

Logic trainer kit, J-K Flip-flop IC - 7476 wires.

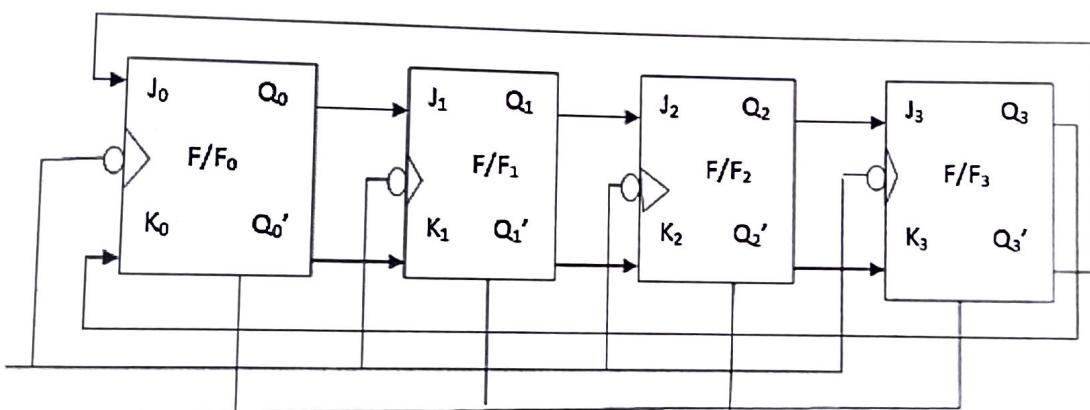
THEORY:

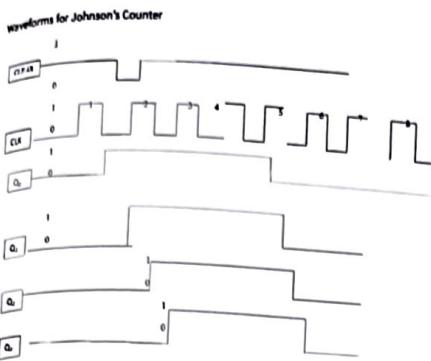
The Johnson counter is a modification of ring counter. In this the inverted output of the last stage flip flop is connected to the input of first flip flop. If we use n flip flops to design the Johnson counter, it is known as $2n$ bit Johnson counter or Mod $2n$ Johnson counter. This is an advantage of the Johnson counter that it requires only half number of flip flops that of a ring counter uses, to design the same Mod. The main difference between the 4 bit ring counter and the Johnson counter is that in ring counter, we connect the output of last flip flop directly to the input of first flip flop. But in Johnson counter, we connect the inverted output of last stage to the first stage input. The flip flop is also known as Twisted Ring Counter, with a feedback. In Johnson counter the input of the first flip flop is connected from the inverted output of the last flip flop. The Johnson counter or switch trail ring counter is designed in such a way that it overcomes the limitations of ring counter. Mainly it reduces the number of flip flops required for designing the circuit.

Operation

- Initially, a short negative going pulse is applied to the clear input of all flip-flops. This will reset all the flip-flops. Hence, initially the o/p's are $Q_3Q_2Q_1Q_0 = 0000$.
- But $Q'_3 = 1$ and since it is copied to J_0 it is also equal to 1.
- $J_0 = 1$ and $K = 1$ initially.
- On the first negative edge of clock arrives at first f/f. o/p of $Q_0 = 1$.
- after 1st -ve edge clock the o/p's of f/f's will be, $Q_3Q_2Q_1Q_0 = 0001$
- On second -ve clock o/p of 2nd f/f will be 1 i.e $Q_1 = 1$.
 $Q_3Q_2Q_1Q_0 = 0011$
- Similarly for 3rd -ve edge clock, $Q_3Q_2Q_1Q_0 = 0111$
- For 4th -ve edge clock, $Q_3Q_2Q_1Q_0 = 1111$
- Now as soon as 5th -ve edge is arrived o/p of 1st f/f becomes 0 i.e $Q_0 = 1$ i.e $Q_3Q_2Q_1Q_0 = 1110$ - This operation continues till the o/p is reached to zero o/p state. i.e $Q_3Q_2Q_1Q_0 = 0000$

Logic diagram:-





Truth Table for a 4-bit Johnson Ring Counter

Clock Pulse No.	FFA	FFB	FFC	FFD
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

PROCEDURE:

- (i) Connections are given as per circuit diagram.
- (ii) Logical inputs are given as per circuit diagram.
- (iii) Observe the output and verify the truth table.

RESULT:

Thus the Johnson counter was designed and their truth table is verified.

PREGATIÖNS:

- All connections should be made neat and tight.
- Digital lab kits and ICs should be handled with utmost care.
- While making connections main voltage should be kept switched off.
- Never touch live and naked wires.