



TECHNO INDIA UNIVERSITY
WEST BENGAL

DIGITAL ELECTRONICS LABORATORY FILE

NAME:

ID:



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

NAME (IN CAPITAL): _____

DEPARTMENT: _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SUBJECT NAME: _____

SUBJECT CODE: _____

INDEX

FINAL SIGNATURE OF TEACHER WITH REMARKS



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.1

NAME OF THE EXPERIMENT: Familiarization with logic gates such as Basic, Universal and Exclusive Gates.

AIM & OBJECTIVE: To Verify The Truth Tables of Basic Gates (NOT, AND, OR) & Universal Gates (NAND, NOR) & Exclusive Gates (XOR).

Apparatus/ Components required: Digital trainer kit, IC- 7404, 7408, 7432, 7400, 7402, 7486 and connecting wires.

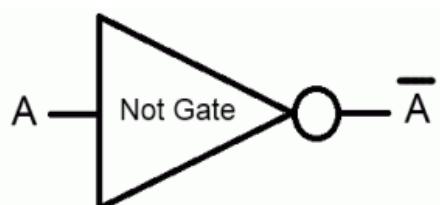
THEORY:

I. THE INVERTER /NOT GATE (IC- 7404):

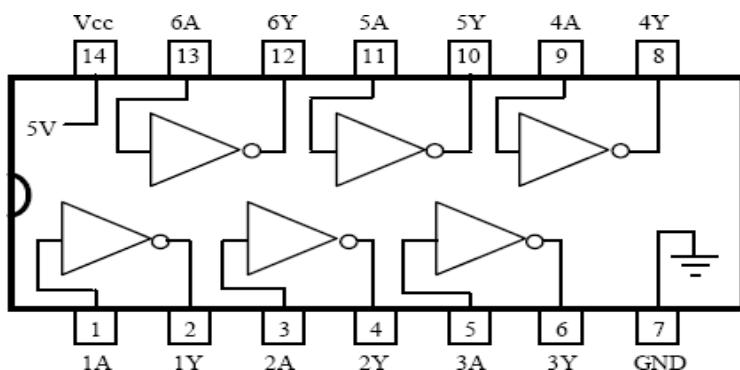
The Inverter or the NOT gate performs the logic function of inversion/complementation. It changes one logic level to opposite logic level, in terms of bits, it changes 1 to 0 and 0 to 1.

LOGIC- $Y = \bar{A}$, when Y is the output and A is the input.

LOGIC SYMBOL:



PIN DIAGRAM:



IC 7404



CIRCUIT CONNECTION:

Connect Pin 14 to $V_{CC} = +5V$, pin 7 to Ground. Inputs should be feed to pins 1, 3, 5, 9, 11 & 13 and outputs should be obtained from 2, 4, 6, 8, 10 & 12 respectively.

TRUTH TABLE:

Input A	Output Y
0	1
1	0

II. AND GATE (IC-7408):

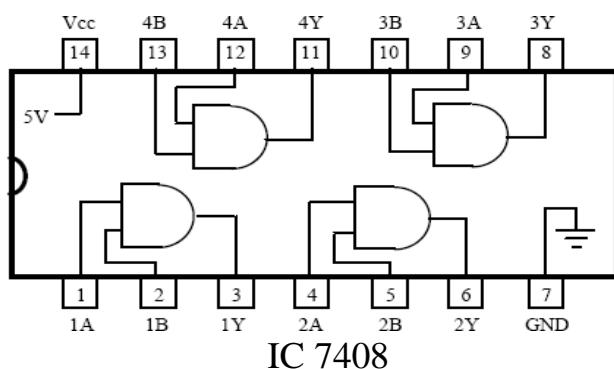
The AND gate performs logical multiplication, more commonly known as AND operation. It is composed of two or more inputs and a single output.

LOGIC- $Y = A \cdot B$ where Y is the output and A and B are the inputs (for two input AND gate).

LOGIC SYMBOL:



PIN DIAGRAM:





CIRCUIT CONNECTION:

Connect Pin 14 to $V_{CC} = +5V$, pin 7 to Ground. Inputs should be feed to pins (1 & 2) or (4 & 5) or(10 & 9) or (12 & 13) and Outputs should be obtained from 3, 6, 8 & 11 correspondingly.

TRUTH TABLE:

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

III. OR GATE (IC-7432):

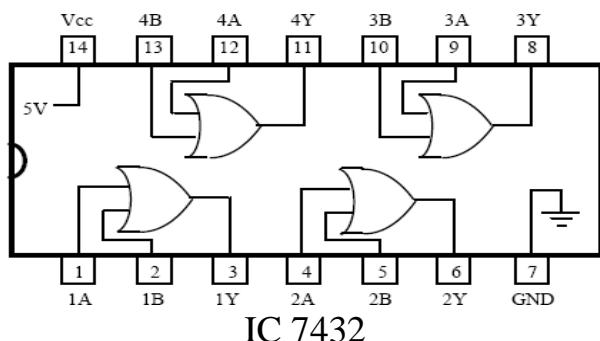
The OR gate performs logical addition, more commonly known as OR operation in digitalelectronics. It is composed of two or more inputs and a single output.

LOGIC- $Y = A + B$ where Y is the output and A and B are the inputs (for two input OR gate).

LOGIC SYMBOL:



PIN DIAGRAM:





CIRCUIT CONNECTION:

Connect Pin 14 to $V_{CC} = +5V$, pin 7 to Ground. Inputs should be feed to pins (1 & 2) or (4 & 5) or (10 & 9) or (12 & 13) and Outputs should be obtained from 3, 6, 8 & 11 correspondingly.

TRUTH TABLE:

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

IV. NAND GATE (IC-7400):

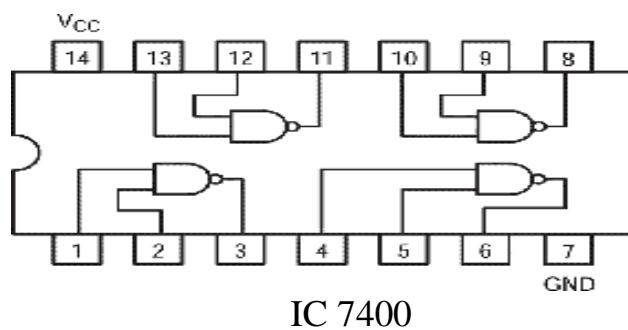
The term NAND is a contraction of NOT-AND and implies an AND function with a complemented (inverted) output. The NAND gate is a very popular logic function because it is "Universal" function; that is it can be used to construct an AND gate, OR gate and NOT gate or a combination of these gates.

LOGIC- $Y = \overline{A \cdot B}$ where Y is the output and A and B are the inputs (for two input NAND gate).

LOGIC SYMBOL:



PIN DIAGRAM:



IC 7400



CIRCUIT CONNECTION:

Connect Pin 14 to $V_{CC} = +5V$, pin 7 to Ground. Inputs should be feed to pins (1 & 2) or (4 & 5) or (10 & 9) or (12 & 13) and Outputs should be obtained from 3, 6, 8 & 11 correspondingly.

TRUTH TABLE:

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

V. NOR GATE (IC-7402):

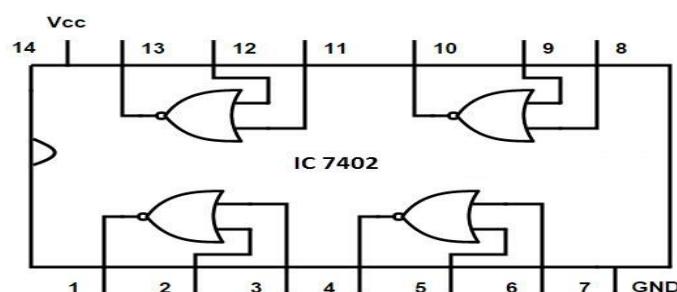
The term NOR is a contraction of NOT-OR and implies an OR function with a complemented (inverted) output. The NOR gate is a very popular logic function because it is "Universal" function like NAND; that is it can be used to construct an AND gate, OR gate and NOT gate or a combination of these gates.

LOGIC- $Y = \overline{A + B}$, where Y is the output and A and B are the inputs (for two input NOR gate).

LOGIC SYMBOL:



PIN DIAGRAM:





CIRCUIT CONNECTION:

Connect Pin 14 to $V_{CC} = +5V$, pin 7 to Ground. Inputs should be feed to pins (2 & 3) or (5 & 6) or (8 & 9) or (12 & 11) and Outputs should be obtained from 1, 4, 10 & 13 correspondingly.

TRUTH TABLE:

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

VI. XOR GATE (IC-7486):

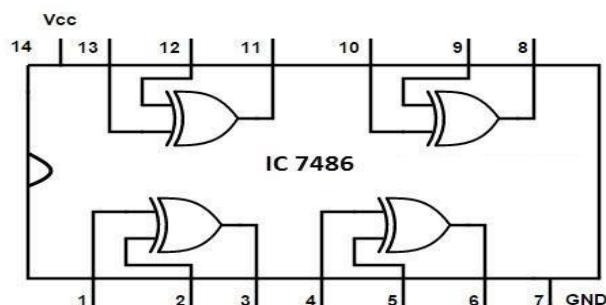
The XOR gates give a HIGH output when the inputs are at opposite logic levels. This is a widely used gate because of its special arithmetic property.

LOGIC- $Y = \bar{A}\bar{B} + A\bar{B} = A \oplus B$ where Y is the output and A, B is the inputs of XOR gate.

LOGIC SYMBOL:



PIN DIAGRAM:





CIRCUIT CONNECTION:

Connect Pin 14 to $V_{CC} = +5V$, pin 7 to Ground. Inputs should be feed to pins (1 & 2) or (4 & 5) or (10 & 9) or (12 & 13) and Outputs should be obtained from 3, 6, 8 & 11 correspondingly.

TRUTH TABLE:

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



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXERCISES:

Draw the Circuit Diagram using any Basic, Universal and Exclusive gates and write down the truth table and verify the truth table in the Digital Trainer Kit.

1. $W = \bar{A} + \bar{B} + AB$
2. $X = \bar{A}B + A\bar{B} + AB$
3. $Y = A\bar{B} + \bar{A}\bar{B}$
4. $Z = (\overline{A + B}).C + \overline{AB}$

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.2

NAME OF THE EXPERIMENT: Realization of basic logic gates and exclusive gates using Universal gates.

AIM & OBJECTIVE: To Verify the Truth Tables of Basic Gates (NOT, AND, OR) & Exclusive Gates (XOR, XNOR) using NAND and NOR gates.

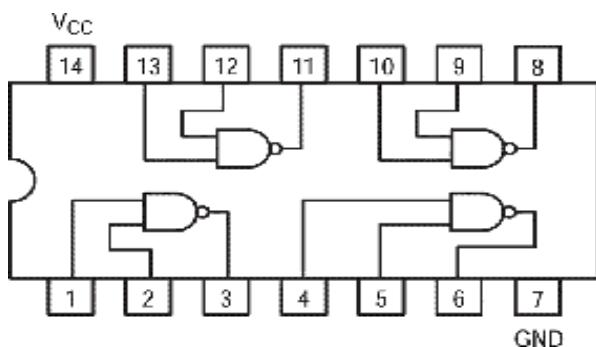
Apparatus/ Components required: Digital trainer kit, IC- 7400, 7402 and connecting wires.

THEORY:

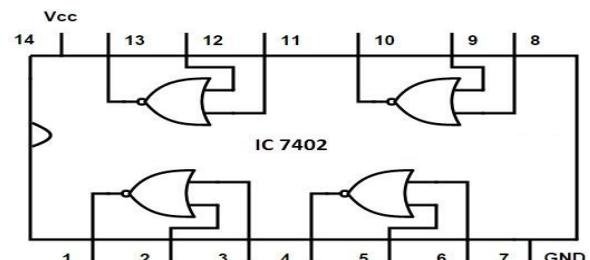
The NAND and the NOR gates can be used to produce any logic function, that is why they are referred as "universal gates".

All Boolean expressions consists of various combination of basic function like NOT, AND & OR, however rather than using separate gates, all operation can be executed using only NAND or NOR gate.

PIN DIAGRAM:



IC 7400



IC 7402

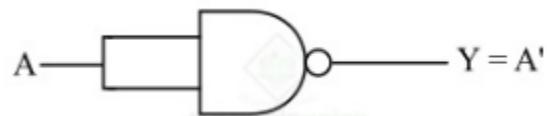


LOGIC DIAGRAMS:

Use of NAND Gate:

i) NOT using NAND:

All NAND input pins connect to the input signal A gives an output \bar{A} .

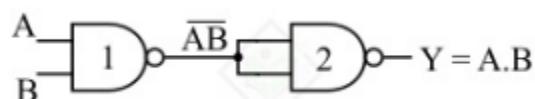


TRUTH TABLE:

Input A	Output Y
0	1
1	0

ii) AND using NAND:

An AND gate can be replaced by NAND gates as shown in the figure (The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter).

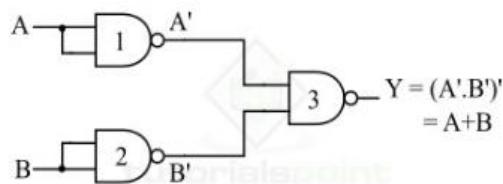


TRUTH TABLE:

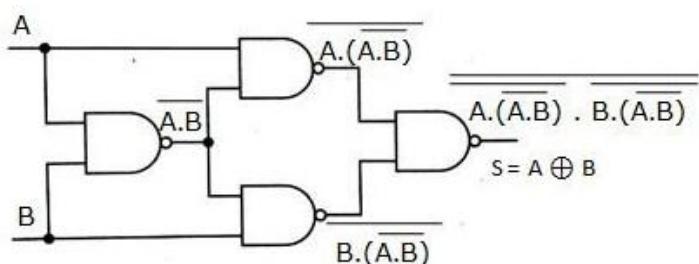
Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

**iii) OR using NAND:**

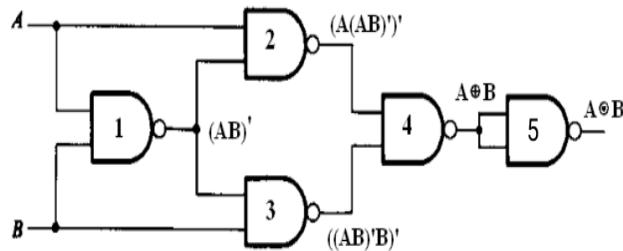
An OR gate can be replaced by NAND gates as shown in the figure (The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters).

**TRUTH TABLE:**

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

iv) XOR using NAND:**TRUTH TABLE:**

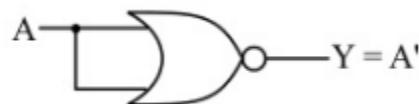
Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

**v) XNOR using NAND:****TRUTH TABLE:**

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

Use of NOR Gate:**i) NOT using NOR:**

All NOR input pins connect to the input signal A gives an output \bar{A} .

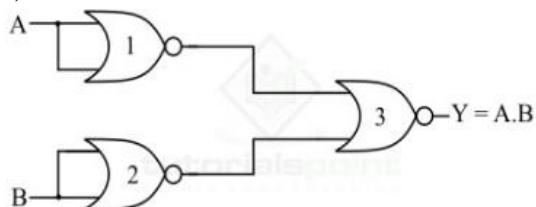
**TRUTH TABLE:**

Input	Output
A	Y
0	1
1	0



ii) AND using NOR:

An AND gate can be replaced by NOR gates as shown in the figure (The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters).

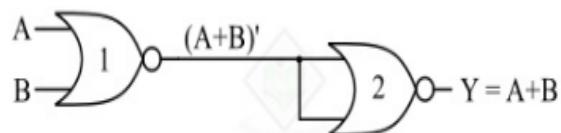


TRUTH TABLE:

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

iii) OR using NOR:

An OR gate can be replaced by NOR gates as shown in the figure (The OR gate is replaced by a NOR gate with its output complemented by a NOR gate inverter).

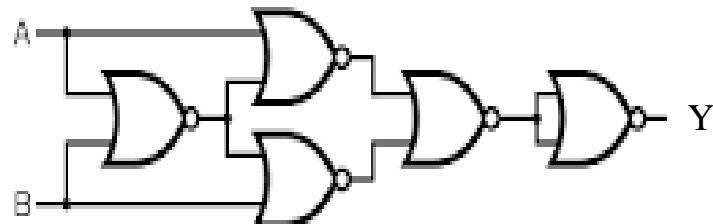


TRUTH TABLE:

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



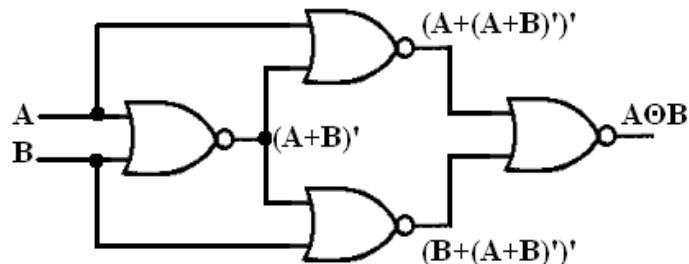
iv) XOR using NOR:



TRUTH TABLE:

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

v) XNOR using NOR:



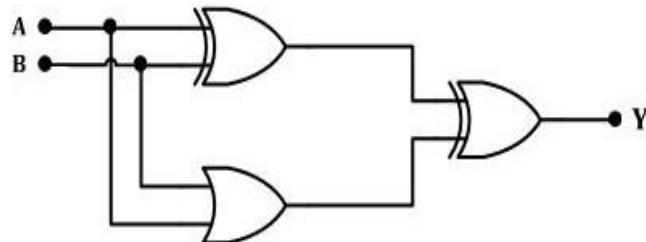
TRUTH TABLE:

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

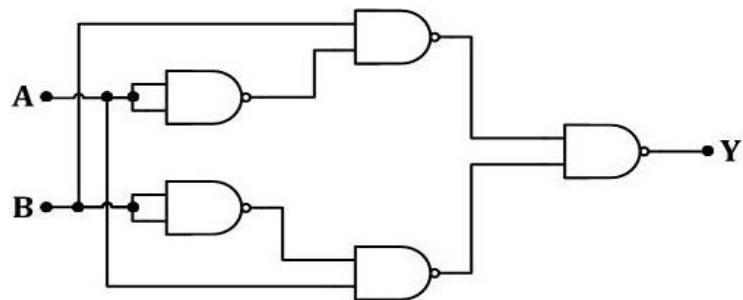


EXERCISES:

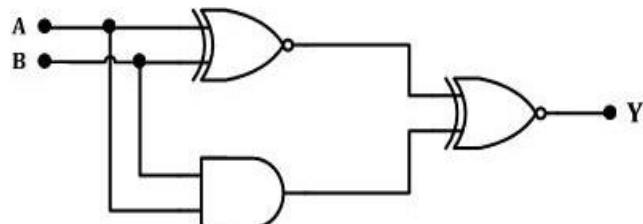
1. Simplify the $\Sigma_m(1,3,4,6,9,11,12,14)$ and implement this using NAND gate only.
2. Find the $\Pi_m(1,3,4,6,9,11,12,14)$ and Implement this using NOR Gate Only.
3. Implement the simplified logic circuit from the following figure.(NAND Gate only)



4. Draw the simplest possible logic diagram that implements the output of the logic diagram shown below (Using NAND/NOR Gate only).



5. Implement the simplified logic circuit from the following figure.(NOR Gate only)



CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.3

NAME OF THE EXPERIMENT: Realization of even and odd parity generation and checking.

AIM & OBJECTIVE: To implement a 3-bit logic circuit for parity generator and checker (even and odd) and verify the truth table.

Apparatus/ Components required: Digital trainer kit, IC- 7486, IC-7404 and connecting wires.

THEORY:

Parity Generator:

A parity bit is an extra bit that is attached to a code group that is being transmitted from one location to another for detecting errors. The parity bit is made either 0 or 1, depending on the number of 1s that are contained in the code group. Two different methods are used for attaching an extra bit ---

- (I) Even parity
- (II) Odd Parity

(I) **Even Parity Generator:** In this method, the value of the parity bit is chosen so that the total number of 1s in the code group (including the parity bit) is an even number.

(II) **Odd Parity Generator:** In this method, the value of the parity bit is chosen so that the total number of 1s in the code group (including the parity bit) is an odd number.



TRUTH TABLE:

For Even Parity generator:

Inputs			Output P_{even}
A	B	C	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

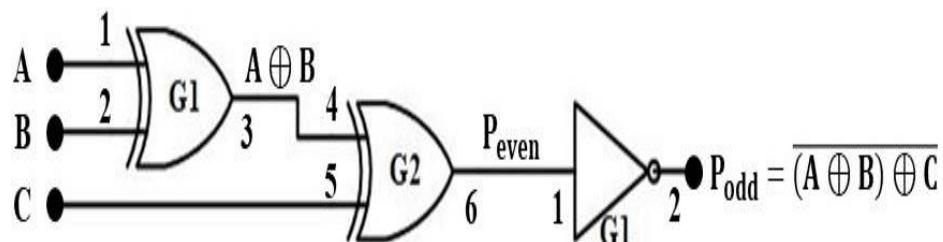
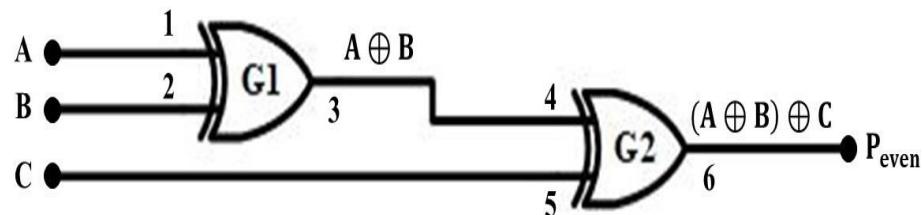
$$P_{\text{even}} = (A \oplus B) \oplus C$$

For Odd Parity generator:

Inputs			Output P_{odd}
A	B	C	
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$P_{\text{odd}} = \overline{(A \oplus B) \oplus C}$$

LOGIC DIAGRAMS:



Parity Checker:

It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the transmission end. When the digital data is received, a parity checking circuit generates an error signal if the total number of 1s is odd in an even parity system or even in an odd parity system.



This parity check can always detect a single-bit error but unable to detect two or more-bit errors within the same code group or word.

When a parity error occurs, the ‘sum even’ output goes low and ‘sum odd’ output goes high. If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the ‘sum odd’ output goes low and ‘sum even’ output goes high.

- (I) Even Parity Checker: Consider that three input messages along with even parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit which checks the possibility of error on data. Since the data is transmitted with even parity, 4 bits received at circuit must have an even number of 1s. If any error occurs, the received message consists of odd number of 1s. The output of the parity checker will be

$$P_{ch_{even}} = \begin{cases} 1 & ; \text{ when error occurs} \\ 0 & ; \text{ when no error} \end{cases}$$

- (II) Odd Parity Checker: Consider that three input messages along with odd parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit which checks the possibility of error on data. Since the data is transmitted with odd parity, 4 bits received at circuit must have an odd number of 1s. If any error occurs, the received message consists of even number of 1s. The output of the parity checker will be

$$P_{ch_{odd}} = \begin{cases} 0 & ; \text{ when no error} \\ 1 & ; \text{ when error occurs} \end{cases}$$



TRUTH TABLE:

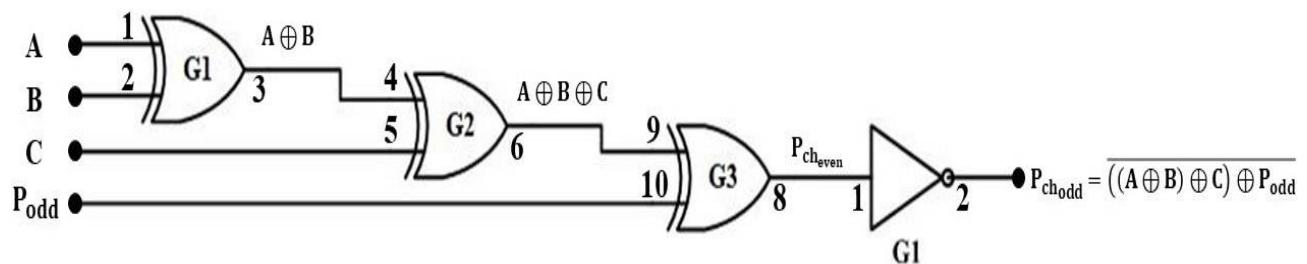
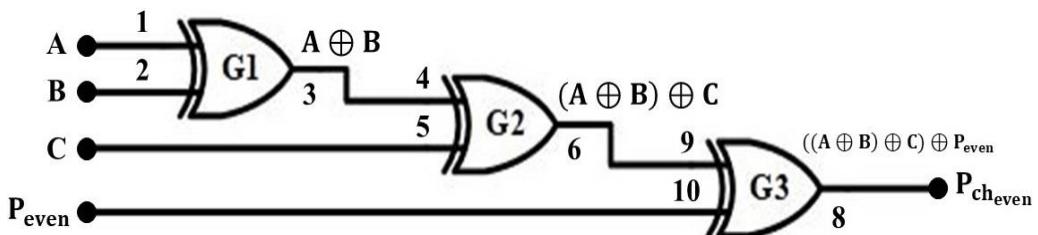
For Even Parity Checker:

Inputs				Output
A	B	C	P _{even}	P _{ch_{even}}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

For Odd Parity Checker:

Inputs				Output
A	B	C	P _{odd}	P _{ch_{odd}}
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
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	1

LOGIC DIAGRAMS:





TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXERCISES:

1. Implement the simplified circuit in SOP form that has three inputs and one output. The output is:
 - a) Equal to 1 when all the inputs are equal to 1 and odd number of inputs are equal to 1,
 - b) Equal to 0 for other cases.
2. Implement the simplified circuit in SOP form that has three inputs and one output. The output is:
 - a) Equal to 1 when none of the inputs are equal to 1 and even number of inputs are equal to 1,
 - b) Equal to 0 for other cases.

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.4

NAME OF THE EXPERIMENT: Binary to Gray and Gray to Binary Conversion.

AIM & OBJECTIVE: To Verify The Truth Tables of Binary to Gray and Gray to Binary Conversion.

Apparatus/ Components required: Digital trainer kit, IC- 7486 and connecting wires.

THEORY:

I. Binary to Gray Code Converter:

The logical circuit which converts Binary code to equivalent Gray code is known as Binary to Gray code converter. If an ‘n’-bit Binary number is represented by B_n, B_{n-1}, \dots, B_1 and its equivalent Gray code will be G_n, G_{n-1}, \dots, G_1 , where B_n and G_n are the MSBs, then the Gray code bits are obtained from the Binary code as follows.

$$G_n = B_n,$$

$$G_{n-1} = B_n \oplus B_{n-1},$$

$$G_{n-2} = B_{n-1} \oplus B_{n-2},$$

⋮

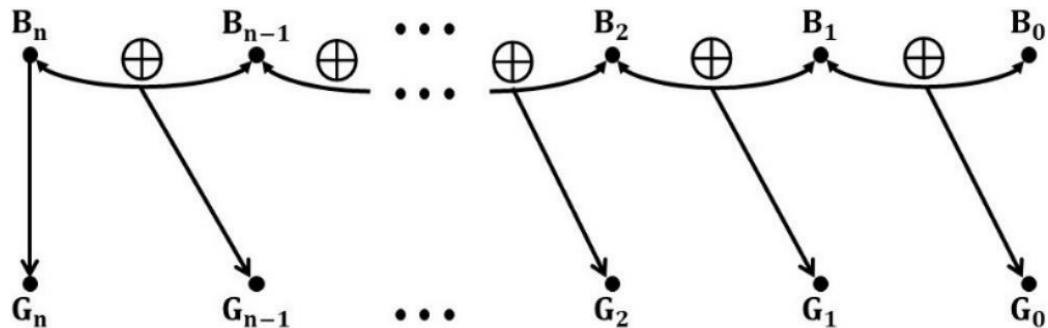
$$\text{and } G_1 = B_2 \oplus B_1$$

The conversion steps for Binary to Gray code are:

- (i) Record the Binary most significant bit (MSB) as MSB for Gray code.
- (ii) XOR the Binary MSB with the next position (Binary) bit and record the resultant bit.
- (iii) Record successive XOR-ed bits until completed.



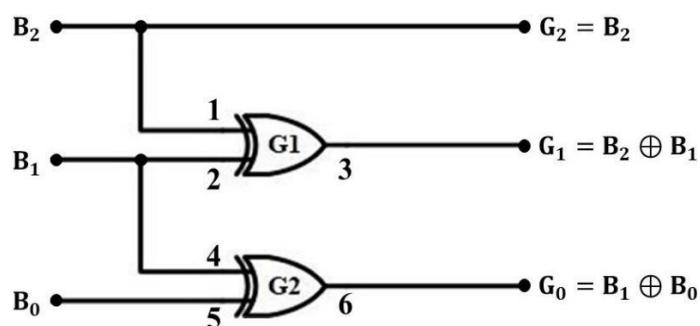
The conversion process is shown in below.



TRUTH TABLE:

Inputs			Outputs		
B ₂	B ₁	B ₀	G ₂	G ₁	G ₀
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

LOGIC DIAGRAMS:





II. Gray to Binary Code Converter:

The logical circuit which converts Gray code to equivalent Binary code is known as Gray to Binary code converter. If an ‘n’-bit Gray number is represented by G_n, G_{n-1}, \dots, G_1 and its equivalent Binary code will be B_n, B_{n-1}, \dots, B_1 , where G_n and B_n are the MSBs, then the Binary code bits are obtained from the Gray code as follows.

$$B_n = G_n,$$

$$B_{n-1} = B_n \oplus G_{n-1} = G_n \oplus G_{n-1},$$

$$B_{n-2} = B_{n-1} \oplus G_{n-2} = G_n \oplus G_{n-1} \oplus G_{n-2},$$

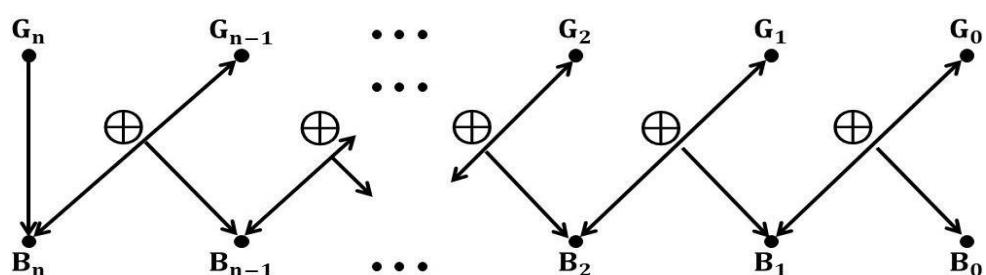
⋮

$$\text{and } B_1 = B_2 \oplus G_1 = G_n \oplus G_{n-1} \oplus G_{n-2} \oplus \dots \oplus G_2 \oplus G_1$$

The conversion steps for Gray to Binary code are:

- (i) Record the most significant bit (MSB) of Gray code as MSB for Binary code.
- (ii) XOR the MSB of Binary code with the next position bit of Gray code and record the resultant bit.
- (iii) Continue the XOR for present Binary bit and next Gray code bit until the LSB is recorded for Binary code.

The conversion process is shown in below.

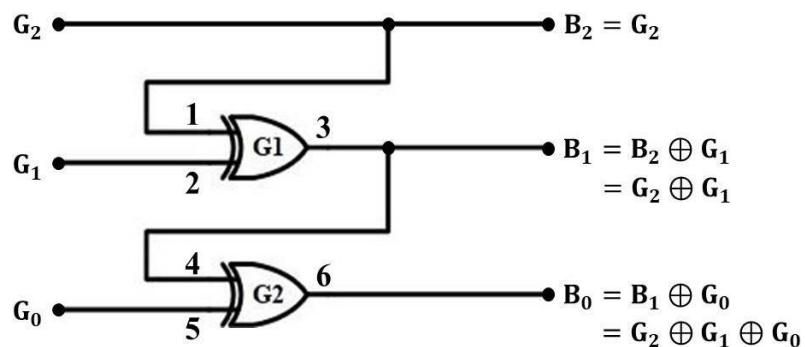




TRUTH TABLE:

Inputs			Outputs		
G_2	G_1	G_0	B_2	B_1	B_0
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

LOGIC DIAGRAMS:





EXERCISES:

1. Design a Circuit with 3 inputs A, B, C and 3 outputs X, Y, Z.
 - a) The output Z is logic 1
 - I. When A=0 and there is odd number of input is at logic 1.
 - II. When A=1 and there is even number of inputs is at logic 1.
 - b) The output $Y = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C$
 - c) The output X is logic 1 when A = 1.
2. Design a Circuit with 3 inputs A, B, C and 3 outputs X, Y, Z.
 - a) The output $Z = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C}$
 - b) The output $Y = \sum_m(2,3,4,5)$
 - c) The output X is logic 1 when A=1.
3. Design a Circuit with 3 inputs A, B, C and 3 outputs X, Y, Z.
 - a) The output $Z = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C}$
 - b) The output $Y = \sum_m(2,3,4,5)$
 - c) The output X is logic 1 when A=1.
4. Design a Circuit with 3 inputs A, B, C and 3 outputs X, Y, Z.
 - a) The output Z is logic 1 when there is odd number of inputs is at logic 1.
 - b) The output $Y = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C$
 - c) The output X is logic 1 when A = 1.

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.5

NAME OF THE EXPERIMENT: Study of Half Adder and Full Adder circuits.

AIM & OBJECTIVE: To design a Half Adder and Full Adder circuits using basic gates and to verify its truth table.

Apparatus/ Components required: Digital trainer kit, IC- 7486, 7408, 7432 and connecting wires.

THEORY:

In digital electronics adder is a circuit that adds binary numbers. Adders can be constructed to carry out many numerical representations.

Half Adder: A Half Adder is a logical circuit that performs an addition operation on two binary digits. The Half adder produces sum and carry values which are both binary digits.

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

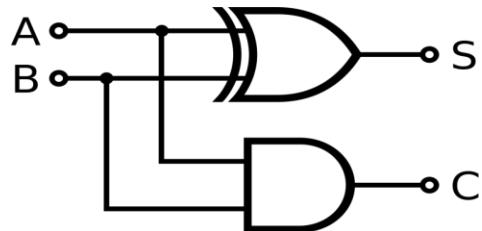
$$C_{\text{OUT}} = A \cdot B$$

TRUTH TABLE:

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



LOGIC DIAGRAMS:



Full Adder: A Full Adder is a logical circuit that performs an addition operation on three binary digits. A Full Adder produces a Sum and Carry Values, which are both binary digits. It can be combined with other Full adders or work on its own.

$$\text{Sum} = (A \oplus B) \oplus C_{IN}$$

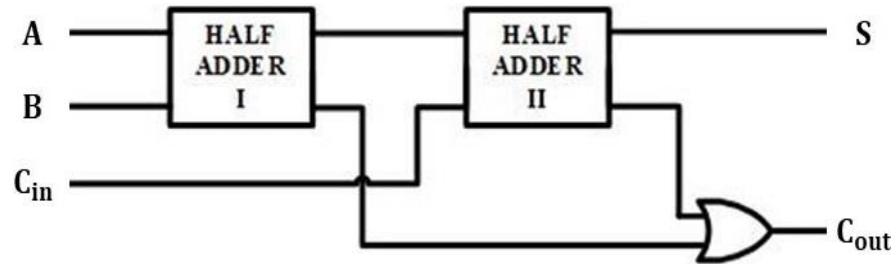
$$C_{OUT} = (AB) + C_{IN} \cdot (A \oplus B)$$

TRUTH TABLE:

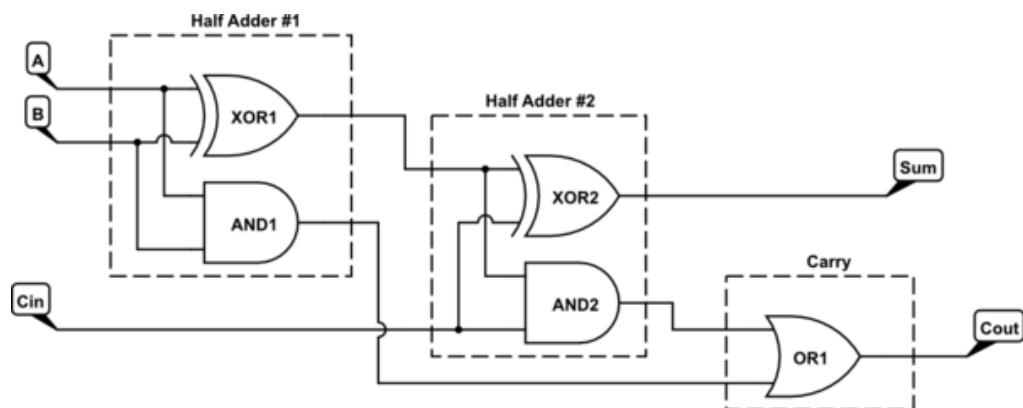
Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



BLOCK DIAGRAM:



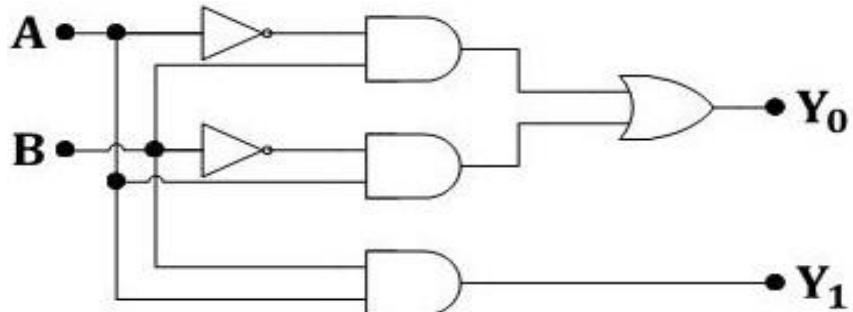
LOGIC DIAGRAMS:





EXERCISES:

1. Design a 2 input 1 bit Adder Circuit by which we get two output simultaneously.
2. Implement the simplified logic circuit from the following figure.



3. Implement the simplified circuit that is inside the block diagram having two outputs with respect to three inputs.



The output equations are: $F_1 = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}\bar{Z} + XYZ$
 $F_2 = \bar{X}YZ + X\bar{Y}Z + XY\bar{Z} + XYZ$

4. Design a Circuit with 3 inputs A, B, C and 2 outputs X, Y.
 - a) The output X is logic 1 when even number of inputs is at logic 1 or all the inputs at logic 1.
 - b) The output Y is logic 1 when odd number of inputs is at logic 1.

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.6

NAME OF THE EXPERIMENT: Study of Half Subtractor and Full Subtractor circuits.

AIM & OBJECTIVE: To design a Half Subtractor and Full Subtractor circuits using basic gates and to verify its truth table.

Apparatus/ Components required: Digital trainer kit, IC- 7486, 7408, 7404, 7432 and connecting wires.

THEORY:

Half Subtractor: Single-bit value is subtracted from another single bit value to produce difference and borrow bits. Any borrow from second bit into first bit is known as borrow taken. Half Subtractor circuit performs subtraction of single bit value from another single bit value and produces single bit difference and borrows.

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

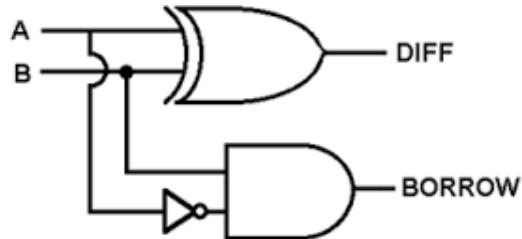
$$\text{Borrow} = \bar{A}B$$

TRUTH TABLE:

Inputs		Outputs	
A	B	D _i (Difference)	B _o (Borrow)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



LOGIC DIAGRAMS:



Full Subtractor: Single bit value is subtracted from second single bit value considering previous borrow to produce difference and borrow bits. Any borrow from second bit into first bit is known as borrow. Full Subtractor circuit performs subtraction of single bit from another single bit value considering borrow and produces single bit difference and borrow.

$$\text{Difference} = (A \oplus B) \oplus Bi;$$

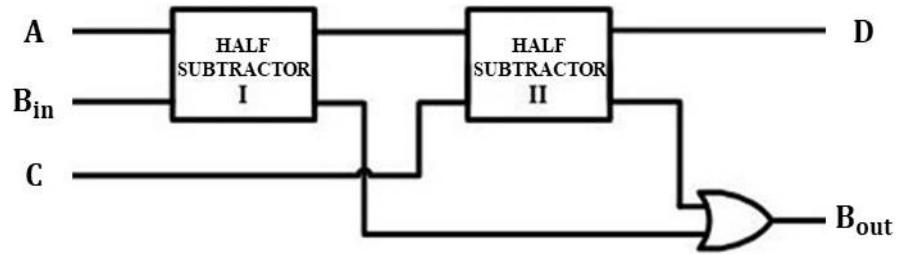
$$\text{Borrow} = \bar{A}B + \overline{(A \oplus B)} \cdot Bi;$$

TRUTH TABLE:

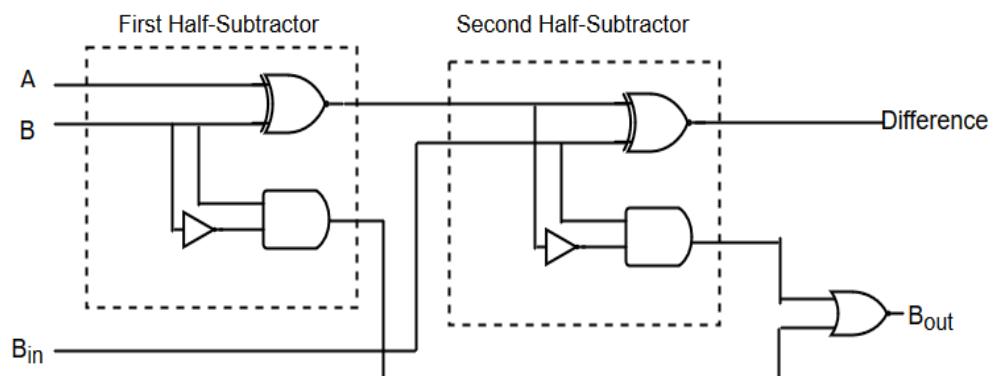
Input			Output	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



BLOCK DIAGRAM:



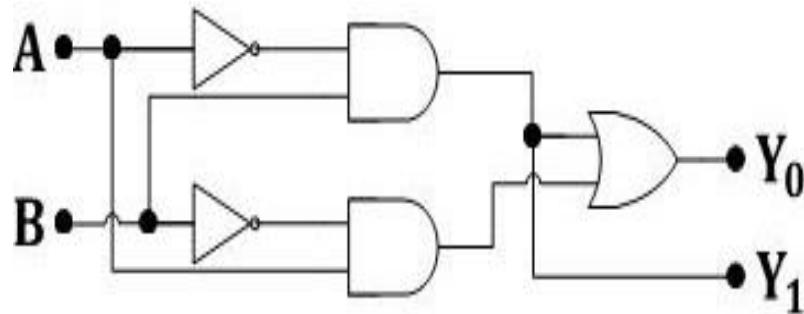
LOGIC DIAGRAMS:





EXERCISES:

1. Design a 2 input 1 bit Subtractor Circuit by which we get two output simultaneously.
2. Implement the simplified logic circuit from the following figure.



3. Implement the simplified circuit that is inside the block diagram having two outputs with respect to three inputs.



$$\begin{aligned}F_1 &= \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}\bar{Z} + XYZ \\F_2 &= \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + XY\bar{Z}\end{aligned}$$

4. Design a Circuit with 3 inputs A, B, C and 2 outputs X, Y.
 - a) The output $X = \sum_m(1,2,3,7)$
 - b) The output Y is logic 1 when odd number of inputs is at logic 1.

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.7

NAME OF THE EXPERIMENT: Design of Combinational Circuits using Multiplexer.

AIM & OBJECTIVE:

- To Design and verify the truth table of 8×1 Multiplexer circuit.
- To Design of Combinational Circuits using Multiplexer.

Apparatus/ Components required: Digital trainer kit, IC- 74151, IC-7432 and connecting wires.

THEORY:

The term multiplex means “**many into one.**” Multiplexing 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 transmits it to a single output line. That is why the multiplexers are also called “**data selectors.**”

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 combinations determine which input is selected.

The size of a multiplexer is specified by the number $2n$ of its input lines and the single output line referred to as “ **2^n to 1**” multiplexer or “ **$2^n \times 1$** ” multiplexer or “ **$2^n:1$** ” multiplexer. It is then implied that it also contains n selection lines. A multiplexer is often abbreviated as MUX.

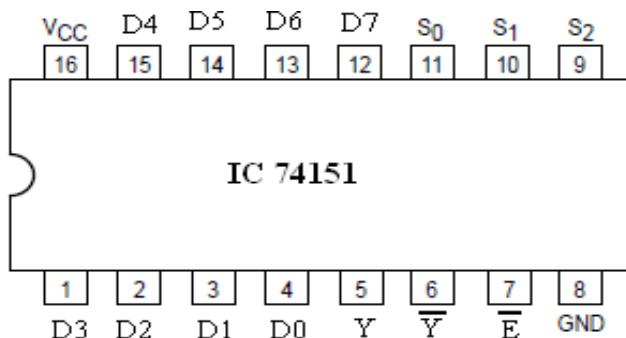
Multiplexers usually have an ENABLE input that can be used to control the multiplexing function. When this input is enabled, that is, when it is in logic ‘1’ or logic ‘0’ state, depending upon whether the ENABLE input is active HIGH or active LOW respectively, the output is enabled. The multiplexer functions normally. When the ENABLE input is inactive, the output is disabled and permanently goes to either logic ‘0’ or logic ‘1’ state, depending upon whether the output is uncomplemented or complemented.



8x1 Multiplexer:

An 8-to-1 Multiplexer (MUX) is a combinational circuit with 8 input lines (D0, D1, D2, D3, D4, D5, D6 and D7) and one output line (Y) and three selection lines (S2, S1 and S0). On receiving active low enable input, Mux is able to perform the intended function.

PIN DIAGRAMS:



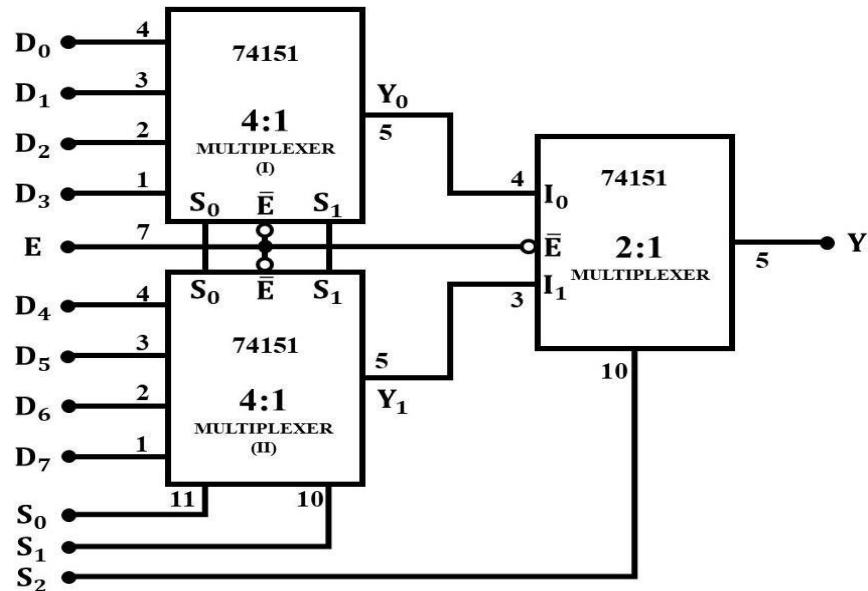
TRUTH TABLE:

INPUT				OUTPUT
E	S ₂	S ₁	S ₀	Y
0	X	X	X	0
1	0	0	0	D ₀
1	0	0	1	D ₁
1	0	1	0	D ₂
1	0	1	1	D ₃
1	1	0	0	D ₄
1	1	0	1	D ₅
1	1	1	0	D ₆
1	1	1	1	D ₇



8:1 Multiplexer using two 4:1 Multiplexer and one 2:1 Multiplexer:

LOGIC DIAGRAMS:



TRUTH TABLE:

Inputs				Output
\bar{E}	S_2	S_1	S_0	Y
0	X	X	X	0
1	0	0	0	D_0
1	0	0	1	D_1
1	0	1	0	D_2
1	0	1	1	D_3
1	1	0	0	D_4
1	1	0	1	D_5
1	1	1	0	D_6
1	1	1	1	D_7

$X \rightarrow 0$ or 1 (don't care)

OUTPUT EQUATION:

$$Y = \bar{E} \bar{S}_2 \bar{S}_1 \bar{S}_0 D_0 + \bar{E} \bar{S}_2 \bar{S}_1 S_0 D_1 + \bar{E} \bar{S}_2 S_1 \bar{S}_0 D_2 + \bar{E} S_1 \bar{S}_0 D_3 + \bar{E} S_2 \bar{S}_1 \bar{S}_0 D_4 \\ + \bar{E} S_2 \bar{S}_1 S_0 D_5 + \bar{E} S_2 S_1 \bar{S}_0 D_6 + \bar{E} S_2 S_1 S_0 D_7$$



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXERCISES:

1. Implement the 8:1 multiplexer using 4:1 multiplexer and OR gate.
2. Implement the full adder circuits using 8:1 multiplexer.
3. Implement the full Subtractor circuits using 8:1 multiplexer.
4. Implement the Boolean functions by selecting appropriate multiplexer.

$$Y(A, B, C, D) = \sum m(1, 3, 4, 11, 12, 13, 14, 15)$$

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.8

NAME OF THE EXPERIMENT: Realization of decoder circuit using logic gates.

AIM & OBJECTIVE: To design a 2 to 4 decoder using logic gates and verify the truth table.

Apparatus/ Components required: Digital trainer kit, IC 7486, IC 7408, IC 7432, IC 7404 and connecting wires.

THEORY:

In a digital system, discrete quantities of information are represented with binary codes. A binary code of ' n ' bits can represent up to ' 2^n ' distinct elements of the coded information.

A decoder is a combinational circuit that converts ' n '-bits of binary information of input lines to a maximum of ' 2^n ' unique output lines. If the ' n '-bit decoded information has unused or don't-care combinations, the decoder output will have less than ' 2^n ' outputs. Usually, decoders are designated as an ' n ' to ' m ' lines decoder, where ' n ' is the number of input lines and ' $m \leq 2^n$ ' is the number of output lines. The purpose is to generate the ' 2^n ' (or less) minterms of ' n ' input variables.

Decoders have a wide variety of applications in digital systems such as data demultiplexing, digital display, digital to analog converting, memory addressing, etc.

2-to-4 Decoder:

A 2-to-4 Decoder is a combinational circuit has two inputs (A and B) and four outputs (D0-D3). Based on the 2 inputs, one of the four outputs is selected.



TRUTH TABLE:

INPUTS		OUTPUTS			
A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

LOGICAL EXPRESSION:

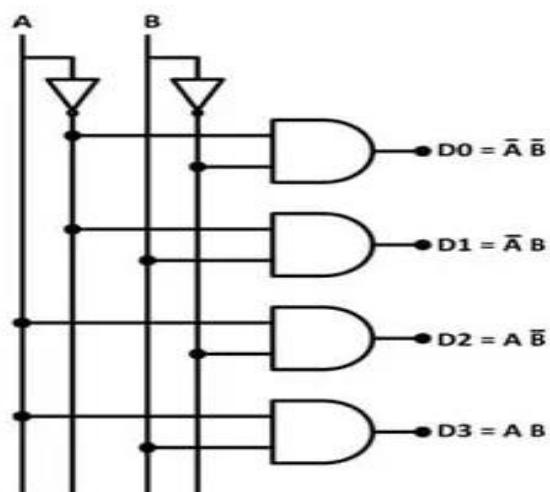
$$D_0 = \bar{A} \cdot \bar{B}$$

$$D_1 = \bar{A} \cdot B$$

$$D_2 = A \cdot \bar{B}$$

$$D_3 = A \cdot B$$

LOGIC DIAGRAMS:





TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXERCISES:

1. Design 2 x 4 decoder using suitable logic gate, and also implement the Boolean function $\Sigma_m(1, 2)$ using it.
2. Design 2 x 4 decoder using suitable logic gate, and also implement the Boolean function $\Sigma_m(0, 3)$ using it.

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.9

NAME OF THE EXPERIMENT: Realization of Flip-Flops.

AIM & OBJECTIVE: To verify the truth table of S-R and J-K Flip-Flop.

Apparatus/ Components required: Digital Trainer Kit, IC- 7400, IC- 7476 and connecting wires.

THEORY:

The logic circuits whose outputs at any instant of time depends only on the input signals present at that time are known as combinational circuits. The output in combinational circuits does not depend upon any past inputs or outputs.

On the other hand, the logic circuits whose outputs at any instant of time depends on the present inputs as well as on the past outputs are called sequential circuits. In sequential circuits, the output signals are fed back to the input side.

Sequential circuits are broadly classified into two main categories, known as synchronous or clocked and asynchronous or unclocked sequential circuits, depending on the timing of their signals.

The basic 1-bit digital memory circuit is known as a flip-flop. It can have only two states, either the 1 state or the 0 state. A flip-flop is also known as a bistable multivibrator. It has one or more inputs and two outputs. The two outputs are complementary to each other.

If Q is 1 i.e., Set, then \bar{Q} is 0; if Q is 0 i.e., Reset, then \bar{Q} is 1. That means Q and \bar{Q} cannot be at the same state simultaneously. If it happens by any chance, it violates the definition of a flip-flop and hence is called an undefined condition. Normally, the state of Q is called the state of the flip-flop, whereas the state of \bar{Q} is called the complementary state of the flip-flop. When the output Q is either 1 or 0, it remains in that state unless one or more inputs are excited to effect a change in the output. Since the output of the flip-flop remains in the same state until the trigger pulse is applied to change the state, it can be regarded as a memory device to store one binary bit.



Generally, synchronous circuits change their states only when clock pulses are present. The operation of the basic flip-flop can be modified by including an additional input to control the behaviour of the circuit.

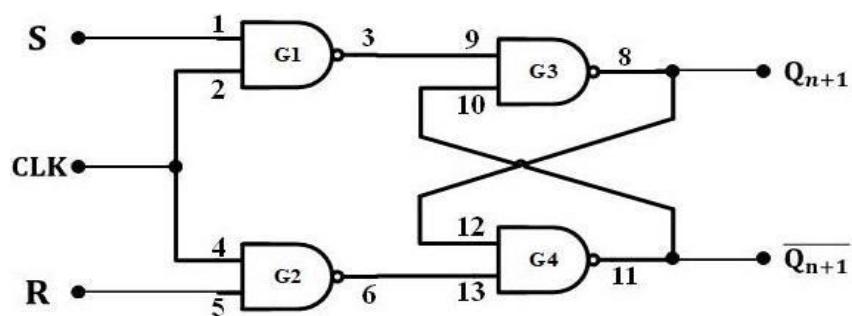
S-R Flip-Flop: It is known as “Set” and “Reset” Flip-Flop The characteristic is defined as

- If $R=0, S=0$, i.e. in the absence of any trigger input, the output retains its state obtained in immediately preceding operation i.e. output Y is either 0 or 1. This state is known as No Change state.
- If $R=0, S=1$, i.e. if any trigger input is applied, output Y is 1. This condition is known as SET.
- If $R=1, S=0$, i.e. if any trigger input is applied, output Y is 0. This condition is known as RESET.
- If $R=1, S=1$, i.e. If a trigger is applied, a problem arises since the output Y is forced simultaneously to 0 and 1. Thus this state is known as Forbidden State.

TRUTH TABLE:

Inputs			Outputs		Operation performed
CLK	S	R	Q_{n+1}	\bar{Q}_{n+1}	
0	X	X	Q_n	\bar{Q}_n	Memory or latch
1	0	0	Q_n	\bar{Q}_n	Memory or latch
1	0	1	0	1	Reset to 0
1	1	0	1	0	Set to 1
1	1	1	1	1	Invalid

LOGIC DIAGRAM:



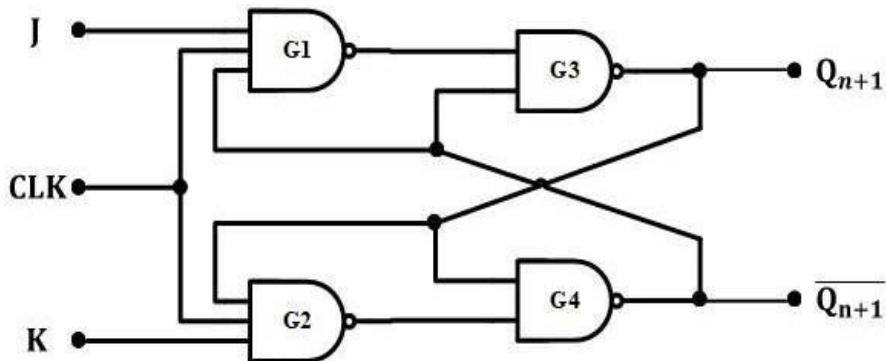


J-K Flip-Flop: The main disadvantage of S-R Flip-Flop is its forbidden state but J-K Flip-Flop overcomes this operation for both $J=K=1$, The Output Toggles its previous states. Rest of the states are same as S-R FF. For purpose of counting, the JK flip-flop is the ideal element to use. The variable J and K are called control I/Ps because they determine what the flip-flop does when a positive edge arrives.

TRUTH TABLE:

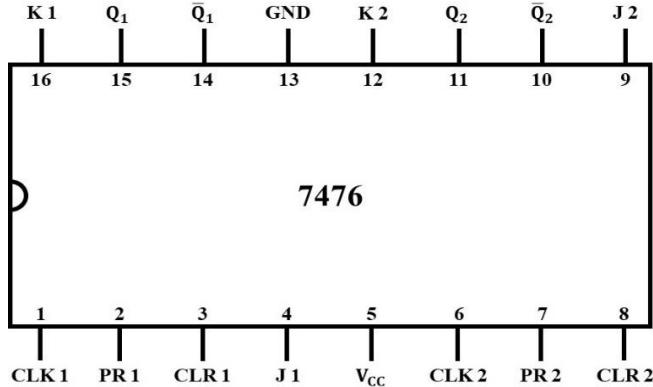
Inputs			Outputs		Operation performed
CLK	J	K	Q_{n+1}	\bar{Q}_{n+1}	
0	X	X	Q_n	\bar{Q}_n	Memory or latch
1	0	0	Q_n	\bar{Q}_n	Memory or latch
1	0	1	0	1	Reset to 0
1	1	0	1	0	Set to 1
1	1	1	\bar{Q}_n	Q_n	Toggle

LOGIC DIAGRAM:





PIN CONFIGURATION OF IC 7476 (MASTER SLAVE DUAL J-K FLIP-FLOP):



Procedure to connect the pins of IC 7476 for J-K flip-flop:

Pin 1 → Connect to manual pulse

Pin 2, Pin 3 and Pin 5 → Connect with +5V power supply for normal CLK Operation

Pin 4 and Pin 16 → Connect with input lines

Pin 15 and Pin 14 → Connect to output lines

Pin 13 → Connect with GND (0V)



EXERCISES:

1. A circuit has three inputs and one output. Show that the output is invalid when all the inputs are HIGH under any external signal. Implement the circuit using NAND gates only.
2. A circuit has three inputs and one output. Show that the output is toggling when all the inputs are HIGH under any external signal. Implement the circuit using proper components.
3. Verify the truth table of D-Flip Flop using NAND Gates.
4. Verify the truth table of T-Flip Flop using IC 7476.

CONCLUSION:



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXPERIMENT NO.: _____

EXPERIMENT NAME: _____

NAME (IN CAPITAL): _____

ID: _____ **YEAR:** _____ **SEMESTER:** _____

SL. NO.	GROUP MEMBER'S NAME (IN CAPITAL)	ID
1.		
2.		
3.		
4.		
5.		

DATE OF EXPERIMENT: _____ **DATE OF SUBMISSION:** _____

SIGNATURE OF TEACHER



EXPERIMENT NO.10

NAME OF THE EXPERIMENT: Realization of asynchronous and synchronous counters.

AIM & OBJECTIVE:

- a) To implement the asynchronous 2-bit UP counter using IC 7476 and verify the truth table.
- b) To implement the asynchronous 2-bit DOWN counter using IC 7476 and verify the truth table.
- c) To implement the synchronous 2-bit UP counter using IC 7476 and verify the truth table.
- d) To implement the synchronous 2-bit DOWN counter using IC 7476 and verify the truth table.

Apparatus/ Components required: Digital trainer kit, IC 7476 (Master slave dual JK flip-flop), and connecting wires.

THEORY:

Counters are one of the simplest types of sequential networks. A counter is usually constructed from one or more flip-flops that change state in a prescribed sequence when input pulses are received. A counter driven by a clock can be used to count the number of clock cycles. Counters can be broadly classified into three categories:

- (i) Asynchronous and Synchronous counters
- (ii) Single and multimode counters
- (iii) Modulus counters

In asynchronous counters, each flip flop is triggered by the previous flip-flop. This type of counters is also called ripple or serial counter.

In synchronous counters, each flip flop is triggered by the same clock pulses simultaneously. This type of counter is also known as a parallel counter.



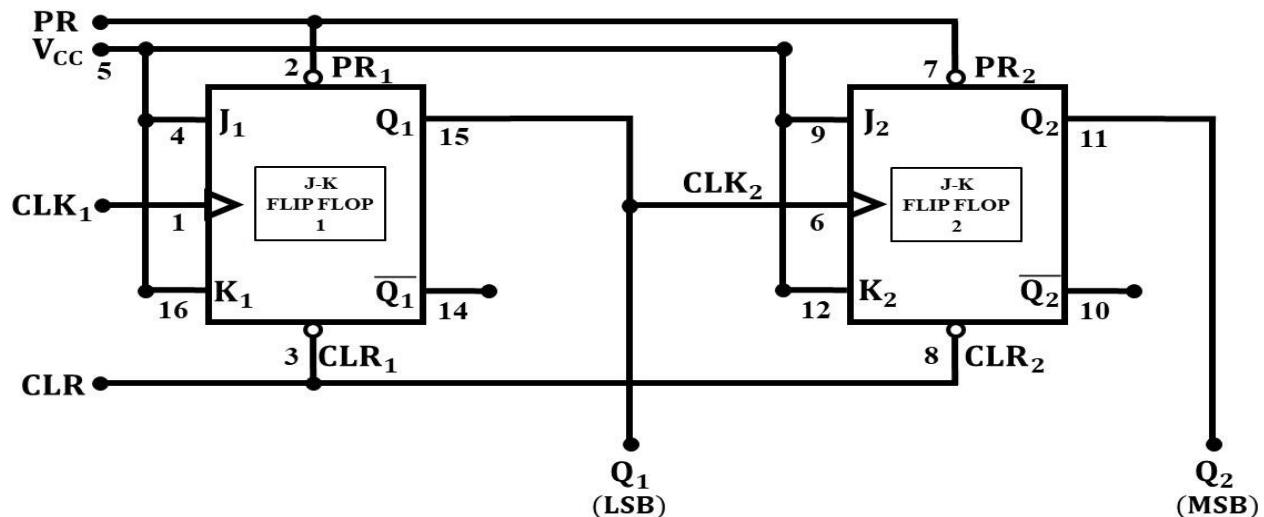
In single mode counters, the count operation can be done either in count-up mode or count-down mode.

In multimode counters, the count operation can be done in both count-up mode and count-down mode simultaneously.

Modulus counters are defined based on the number of states they are capable of counting. This type of counter can again be classified into two types: Mod N and MOD < N.

(I) Asynchronous 2-bit UP counter:

Block Diagram (with IC 7476 pin numbers)



Procedure to connect the pins of IC 7476 for Asynchronous 2-bit UP counter:

Pin 1 → Connect to manual pulse

Pin 2, Pin 3, Pin 4, Pin 5, Pin 7, Pin 8, Pin 9, Pin 12, and Pin 16 → Connect with +5V power supply

Pin 11 and Pin 15 → Connect with output lines

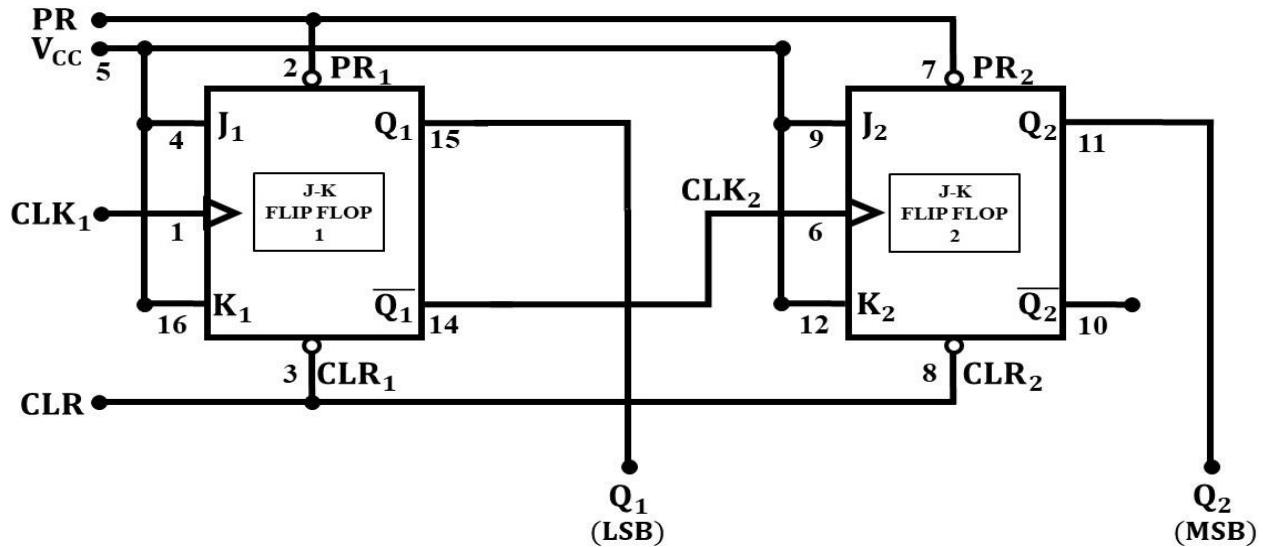
Pin 15 and Pin 6 → Make shorted for next stage clock pulse

Pin 13 → Connect with GND (0V)



TRUTH TABLE:

After pulses	Count	
	Q ₂	Q ₁
0	0	0
1	0	1
2	1	0
3	1	1

(II) Asynchronous 2-bit DOWN counter:Block Diagram (with IC 7476 pin numbers)

Procedure to connect the pins of IC 7476 for Asynchronous 2-bit UP counter:

Pin 1 → Connect to manual pulse

Pin 2, Pin 3, Pin 4, Pin 5, Pin 7, Pin 8, Pin 9, Pin 12, and Pin 16 → Connect with +5V power supply

Pin 11 and Pin 15 → Connect with output lines

Pin 14 and Pin 6 → Make shorted for next stage clock pulse

Pin 13 → Connect with GND (0V)

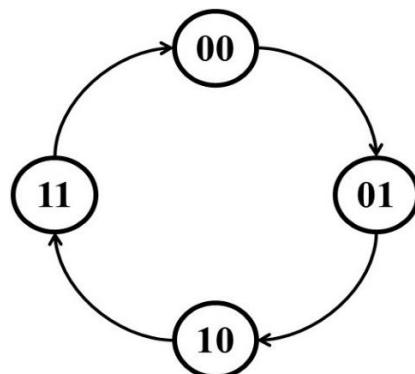


TRUTH TABLE:

After pulses	Count	
	Q_2	Q_1
0	0	0
3	1	1
2	1	0
1	0	1

(III) Synchronous 2-bit UP counter:

State Diagram:



Procedure to connect the pins of IC 7476 for Synchronous 2-bit UP counter:

Pin 1 and Pin 6 → Connect to manual pulse

Pin 2, Pin 3, Pin 4, Pin 5, Pin 7, Pin 8, and Pin 16 → Connect with +5V power supply

Pin 9, Pin 12, and Pin 15 → Make shorted for next stage input signals

Pin 11 and Pin 15 → Connect with output lines

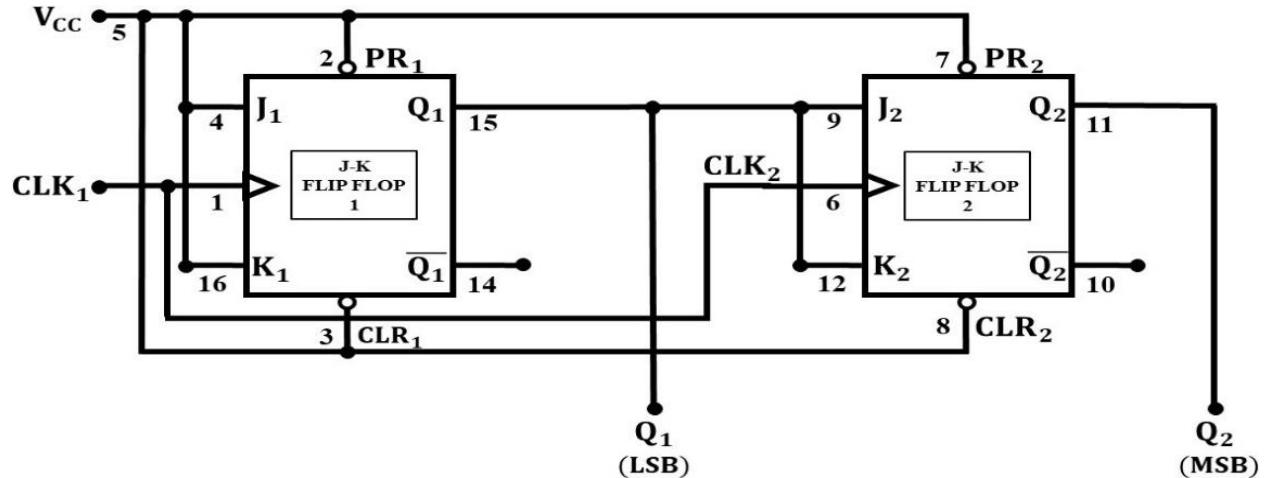
Pin 13 → Connect with GND (0V)

Excitation Table:

Present State		Next State		Excitation Inputs			
Q_2	Q_1	Q_2	Q_1	J_2	K_2	J_1	K_1
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

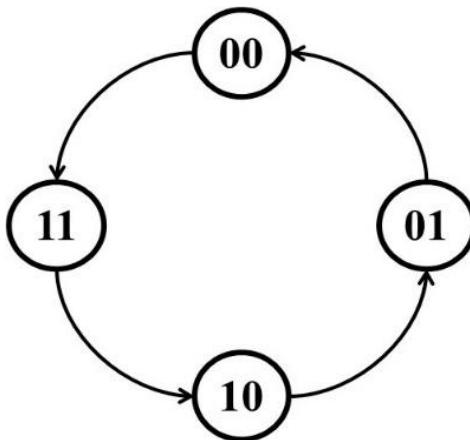


Block Diagram (with IC 7476 pin numbers)



(IV) Synchronous 2-bit DOWN counter:

State Diagram:



Procedure to connect the pins of IC 7476 for Synchronous 2-bit DOWN counter:

Pin 1 and Pin 6 → Connect to manual pulse

Pin 2, Pin 3, Pin 4, Pin 5, Pin 7, Pin 8, and Pin 16 → Connect with +5V power supply

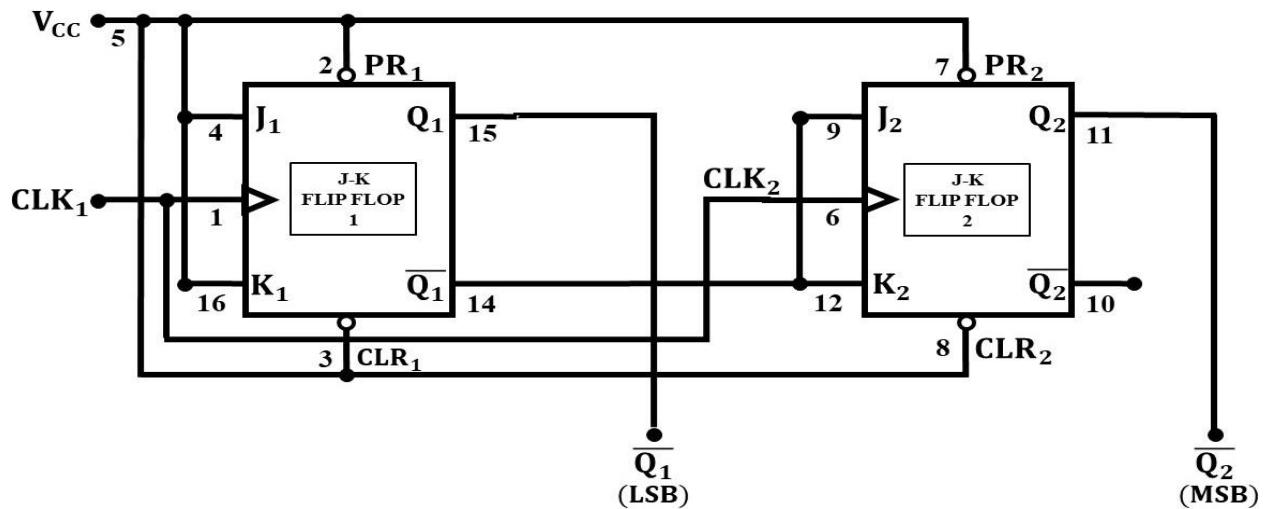
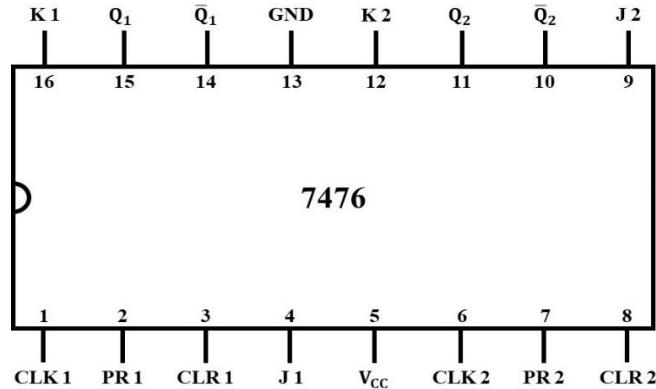
Pin 9, Pin 12, and Pin 14 → Make shorted for next stage input signals

Pin 11 and Pin 15 → Connect with output lines

Pin 13 → Connect with GND (0V)

**Excitation Table:**

Present State		Next State		Excitation Inputs			
Q_2	Q_1	Q_2	Q_1	J_2	K_2	J_1	K_1
0	0	1	1	1	X	1	X
1	1	1	0	X	0	X	1
1	0	0	1	X	1	1	X
0	1	0	0	0	X	X	1

Block Diagram (with IC 7476 pin numbers)**PIN CONFIGURATION OF IC 7476 (MASTER SLAVE DUAL J-K FLIP-FLOP):**



TECHNO INDIA UNIVERSITY

W E S T B E N G A L

EXERCISES:

1. Implement a circuit using IC-7476 that counts $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$ when enable input will apply simultaneously to flip-flop(s).
2. Implement a circuit using IC-7476 that counts $00 \rightarrow 11 \rightarrow 10 \rightarrow 01 \rightarrow 00$ when enable input will apply simultaneously to flip-flop(s).
3. Implement a circuit using IC-7476 that counts $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ when enable input will not apply at the same time to flip-flop(s).
4. Implement a circuit using IC-7476 that counts $0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$ when enable input will not apply at the same time to flip-flop(s).

CONCLUSION: