AMRITA SAI

INSTITUTE OF SCIENCE & TECHNOLOGY

Accredited by NAAC with 'A' Grade

(Approved by AICTE, New Delhi, Affiliated to JNTUK Kakinada, SBTET)

DLD Lab Manual

www.SureshQ.blogspot.in

Department of E.C.E.

LIST OF EXPERIMENTS

DIGITAL LOGIC DESIGN LABORATORY

STUDENT NAME: ***REG NO:***

BRANCH: YEAR:

S.NO	EXERCISE NAME	DATE	STAFF SIGNATURE	REMARKS / GRADE
1	Verification of Basic Logic Gates.			
2	Implementing all individual gates with universal gates NAND & NOR.			
3	Design a circuit for the given canonical form, draw the circuit diagram and verify the De-Morgan laws.			
4	Design a combinational logic circuit for 4X 1 MUX and verify the truth table.			
5	Design a combinational logic circuit for 4X 1 De -MUX and verify the truth table.			
6	Construct Half Adder and Full Adder using Half Adder and verify the truth table.			
7	Verification of truth tables of the basic Flip -Flops with synchronous and asynchronous modes.			
8	Implementation of Master Slave Flip-Flop with J-K Flip- Flop and verify the truth table for Race Around condition.			
9	Design a Decade Counter and verify the truth table.			
10	Design the Mod 6 counter using D – Flip-Flop.			
11	Construct 4-Bit Ring counter with T– Flip -Flop and verify the truth table.			
12	Design a 8- Bit Right Shift Register using D – Flip -Flop and verify the truth table.			
13	Verify the Data Read and Data Write operations for the IC 74189.			
14	Design a Gray code encoder and interface it to SRAM IC 74189 for write operation display on 7-segment.			
15	Design a Gray code De-coder and interface it to SRAM IC 74189 for write operation display on 7-segment.			

Signature of Staff Member

Experiment No:1

Verification of Basic Logic Gates

AIM: To study the operations and verifying truth tables of Basic Logic Gates AND, OR, NOT and EXOR gates using IC's.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7408,7432 ,7404,7400,7402 and 7486

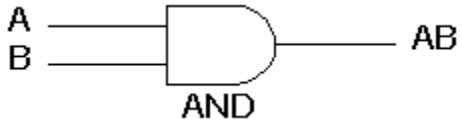
EQUIPMENT:

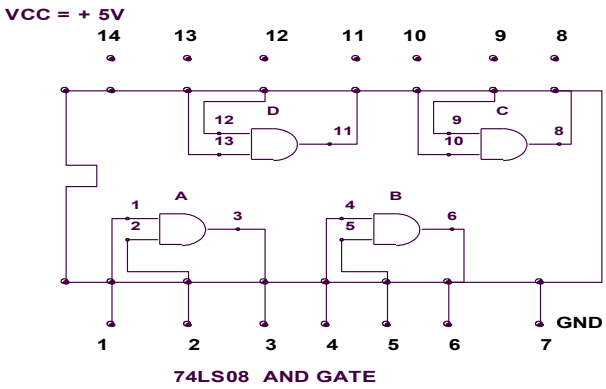
1. Power supply
2. Bread Board.

Theory:

AND Gate:

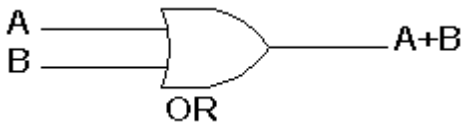
The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. The logic symbol and its expression of AND Gate is shown in Fig. The IC 74LS08 is a two-input AND Gate IC it consists of 4-AND gates. The IC has 14 pins as shown in Fig. The truth table of AND Gate is as shown in table.





OR Gate:

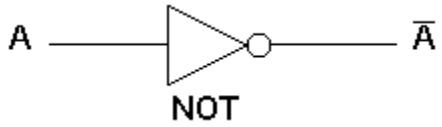
The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high. A plus (+) is used to show the OR operation. The logic symbol and its expression of OR Gate are shown in Fig. The IC 74LS32 is a two in put OR Gate IC it consists of 4-OR gates. The IC has 14 pins as shown in Fig. The truth table of OR Gate is as shown in table.



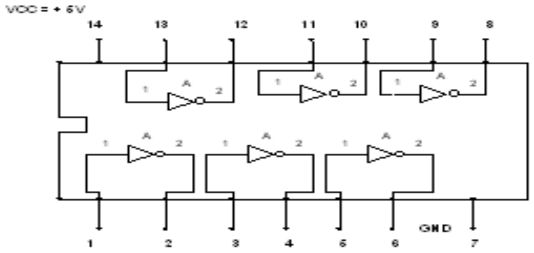
OR - Gate Symbol

NOT Gate:

When in put variable A is low the output of a NOT gate is High. The logic symbol of NOT Gate is shown in Fig .The IC 74LS04 is a single in put NOT Gate IC and it consists of 6-NOT gates. The IC has 14 pins constructed in Dual in Line package(DIP) as shown in Fig. The truth table of NOT Gate is as shown in Fig.



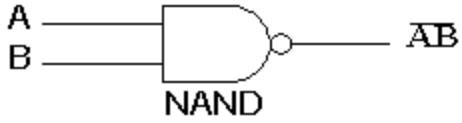
NOT - Gate Symbol



IC 74LS 04 – NOT GATE

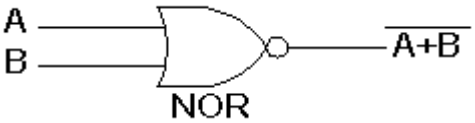
NAND Gate:

The outputs of all NAND gates are high if **any** of the inputs are low. The logic symbol of NAND Gate is shown in Fig .The IC 74 LS00 is a two input NAND Gate IC. It consists 4 NAND gates built in. The truth table of NAND Gate is as shown in Table.



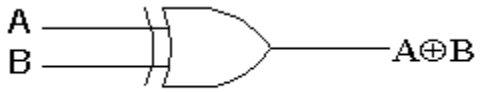
NOR Gate:

The outputs of all NOR gates are low if **any** of the inputs are high. The logic symbol of NOR Gate is shown in Fig. The IC 74 LS02 is two in-put NOR Gate IC it consists of 4 NOR gates. The truth table of NOR Gate is as shown



EX-OR Gate:

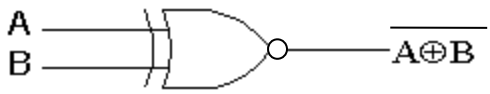
The '**Exclusive-OR**' gate is a circuit which will give a high output if **either, but not both**, of its two inputs are high. . The logic symbol of **EXOR** Gate is shown in Fig .The IC 74LS86 is a single in put **EXOR** Gate IC and it consists of 4- **EXOR** gates. The truth table of **EXOR** Gate is as shown in Table.



EXOR

EX-NOR Gate:

The '**Exclusive-NOR**' gate is a circuit which will give a high output if **either, but not both**, of its two inputs are high. . The logic symbol of **EXNOR** Gate is shown in Fig . The truth table of **EXOR** Gate is as shown in Table.



EXNOR

BOOLEAN Algebra

George Boole introduced the fundamental concepts of a two-values (binary) system called Boolean Algebra. Boolean algebra is mathematics of logic. It is basic tool available to the logic designer and thus can be used for simplification of complex logic expressions. Postulates and theorems of Boolean algebra and their applications are used in minimizing Boolean expressions. *Variables* are the different symbols in a Boolean expression. They may take on the value '0' or '1'.(www.SureshQ.Blogspot.in)

Postulates of Boolean Algebra

The following are the important postulates of Boolean algebra:

$$\bar{0} = 1$$

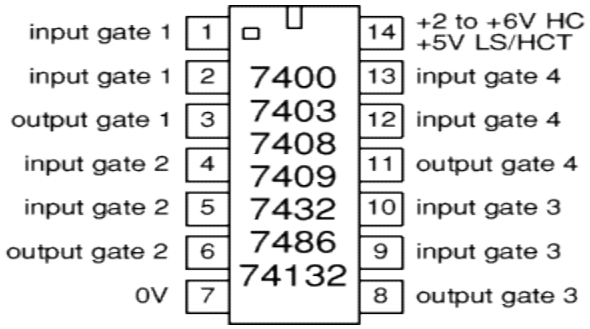
$$\bar{1} = 0$$

Boolean Theorem

1. (*a*) A + A = A	(*b*) A.A = A Tautology Law
2. (*a*) A + 1 = 1	(*b*) A.0 = 0 Union Law
3. (A′)′ = A Involution Law	
4. (*a*) A + (B + C) = (A + B) + C	(*b*) A.(B.C) = (A.B).C Associative Law
5. (*a*) (A + B)′ = A′B′	(*b*) (A.B)′ = A′ + B′ De Morgan's Law
6. (*a*) A + AB = A	(*b*) A(A + B) = A Absorption Law
7. (*a*) A + A′B = A + B	(*b*) A(A′ + B) = AB
8. (*a*) AB + AB′ = A	(*b*) (A + B) (A + B′) = A Logical adjancy
9. (*a*) AB + A′C + BC = AB + A′C	(*b*) (A + B) (A′ + C) (B + C) = (A+B) Consensus Law

PROCEDURE:

1. Construct the circuit on breadboard for each Gate as shown in figures by inserting the appropriate IC.
2. Check the combinations of various inputs as shown in truth tables for each Gate.
3. If the input is low connect input to Ground, which indicates logic 0.
4. If input is high or logic 1 then connect the input to the power supply.
5. When output is high the LED will glow which indicates output as high, if the LED is not glowing then the output is low.



PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.(www.SureshQ.blogspot.in)
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

OBSERVATIONS:

Truth Table for AND Gate		
A	B	X = A*B
0	0	
0	1	
1	0	
1	1	

Truth Table for NOT Gate	
A	X = $\bar{A}$
0	
1	

Truth Table for EXOR Gate		
A	B	X =
0	0	
0	1	
1	0	
1	1	

Truth Table for NAND Gate

Truth Table for EXNOR Gate

		X =
1	1	

RESULT:

Signature of Staff Member

Experiment No:2

Implementing all individual gates with Universal gates NAND & NOR

AIM: To implement all individual gates with universal gates NAND and NOR**.**

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7400 & 7402

EQUIPMENT:

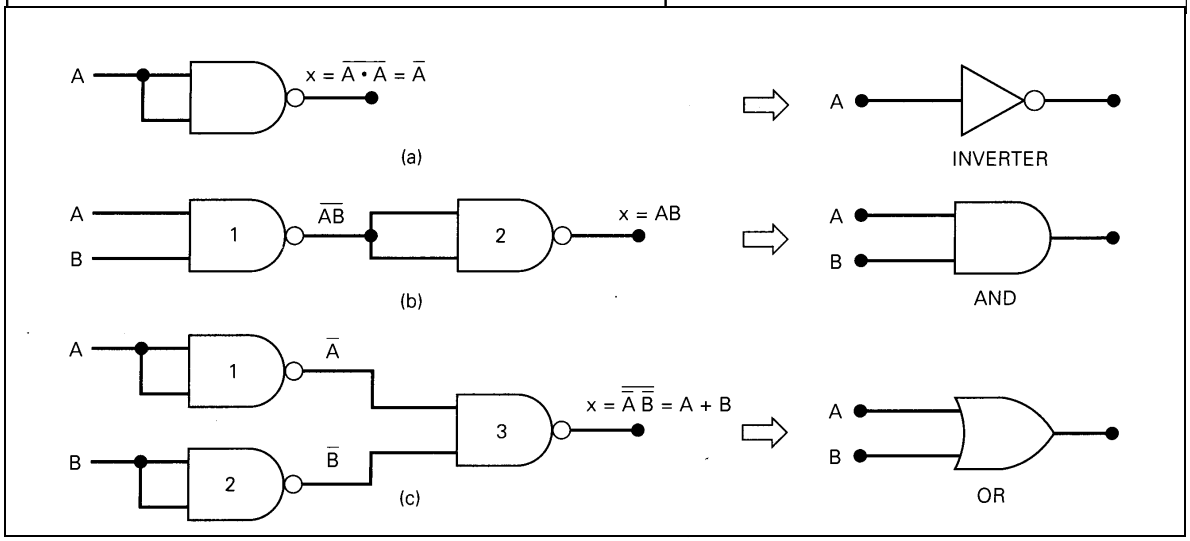
1. Power supply
2. Bread Board.

Theory:

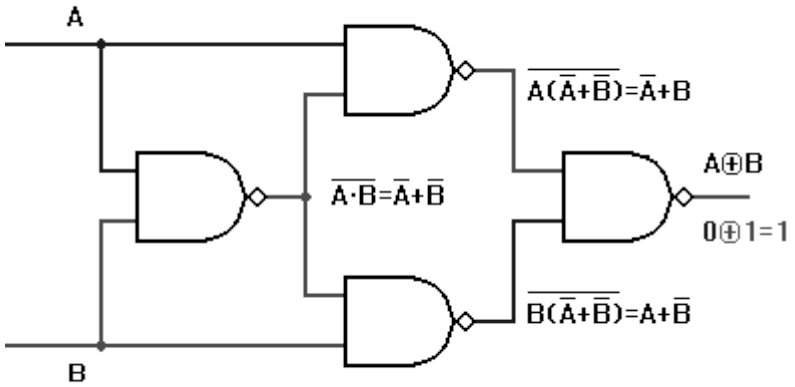
To realize the Basic Logic Gates using NAND Gate:

The NAND and NOR gates are called universal functions since with either one the AND and OR functions and NOT can be generated. A function in sum of products form can be implemented using NAND gates by replacing all AND and OR gates by NAND gates. The outputs of all NAND gates are high if **any** of the inputs are low. The logic symbol of NAND Gate is shown in Fig.

Logic Function Using NAND Gate	Logic Gate

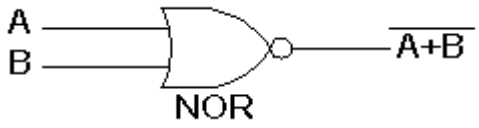


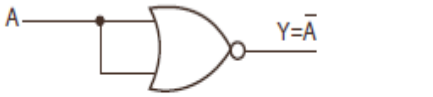
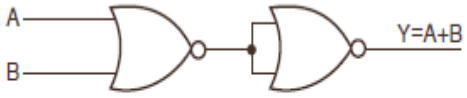
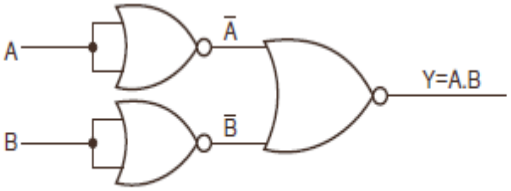
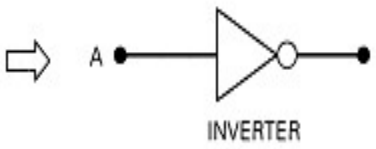
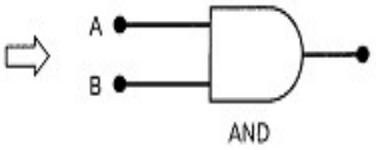
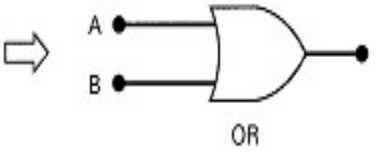
Realize the EX-OR gate using NAND Gate:



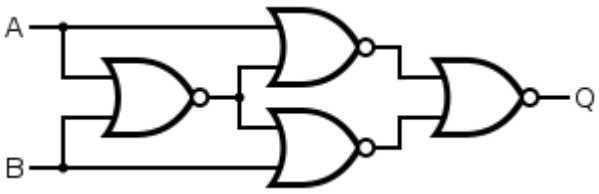
To realize the Basic Logic Gates using NOR Gate:

The NOR gate is called universal gate since with this gate AND, OR & NOT functions and can be generated. A function in product of sums form can be implemented using NOR gates by replacing all AND and OR gates by NOR gates. The outputs of all NOR gates are high if **any** of the inputs are HIGH. The logic symbol of NOR Gate is shown in Fig.



Logic Function Using NOR Gate	Logic Gate
 (a)  (b)  (c)	  

Realizing the EX OR gate using NOR Gate:



Y = (AB + B'A')'
= (AB)'.(B'.A')'
= (A'+B').((B')'+(A')')
= (A'+B').(B+A)
= A'B+A'A+BB'+B'A
= A'B+AB' (Boolean expression for XOR Gate)

PROCEDURE:

1. Construct the circuit on breadboard for each Gate as shown in figures by inserting the appropriate IC.
2. Check the combinations of various inputs as shown in truth tables for each Gate.
3. If the input is low connect input to Ground, which indicates logic 0.
4. If input is high or logic 1 then connect the input to the power supply.
5. When output is high the LED will glow which indicates output as high, if the LED is not glowing then the output is low.

OBSERVATIONS:

Truth Table for NAND Gate

A	B	X = AXB
0	0	
0	1	
1	0	
1	1	

Truth Table for NOR Gate

A	B	X = A+B
0	0	
0	1	
1	0	
1		

RESULT:

Signature of Staff Member

Experiment No:3

Design a circuit for the given canonical form, Draw the circuit diagram and verify the De-Morgan laws

AIM: To construct a circuit for the given canonical form, draw the circuit diagram and verify the De-Morgan laws.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7408, 7432 and 7404.

EQUIPMENT:

1. Power supply
2. Bread Board.

/

THEORY:

Canonical Forms

Boolean functions are commonly expressed using the following forms:

- Canonical forms:
  - Sum of minterms
  - Product of maxterms
- Standard forms:
  - Sum of products **(SOP)**
  - Product of sums **(POS)**

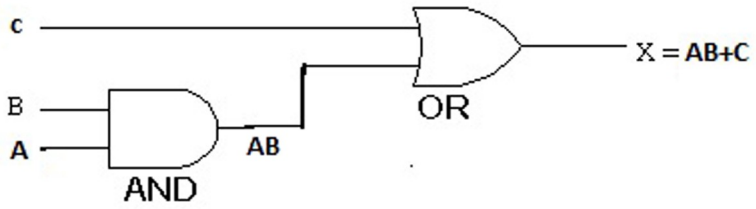
Represented as a **sum of minterms** only : f **= Σ(minterms)**
Represented as a **product of maxterms** only : f**= π(maxterms)**

• A minterm is a product of all variables taken either in their direct or complemented form
• A maxterm is a sum of all variables taken either in their direct or complemented form

$$y = f(x_{n-1}, \ldots, x_0) = \sum_{i=0}^{2^n-1} y_i \cdot m_i$$

Boolean function can be specified using a sum (logical OR) of minterms.
X = AB+C the equivalent canonical form is ABC+A'BC+AB'C+A'B'C+ABC'.
Algebraically it can be shown that these two are equivalent as follows:
ABC+A'BC+AB'C+A'B'C+ABC'= BC(A+A') + B'C(A+A') + ABC'
= BC1 +B'C1 + ABC'
= C(B+B') + ABC'
= C + ABC'
= C + AB

Realization of Boolean Expression:X=AB+C



Substitute a one in the truth table for each entry below.
Creating a Truth Table from Minterms Step One
Finally put zeros in all the entries that you did not fill with ones in the first step above:

Minterms for 3 Input Variables

C	B	A	Minterm
0	0	0	A'B'C'
0	0	1	AB'C'
0	1	0	A'BC'
0	1	1	ABC'
1	0	0	A'B'C
1	0	1	AB'C
1	1	0	A'BC
1	1	1	ABC

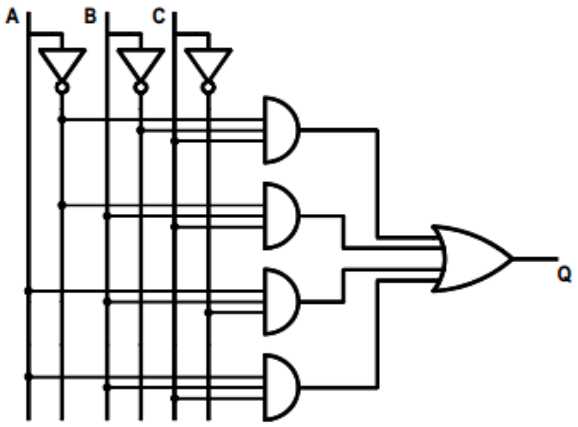
Y	C	B	A	Minterm	m_j
0	0	0	0		
1	0	0	1		
2	0	1	0		
3	0	1	1	1	m_3
4	1	0	0	1	m_4
5	1	0	1	1	m_5
6	1	1	0	1	M_6
7	1	1	1	1	m_7

C	B	A	Minterm
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$y = 0 \cdot m_0 + 0 \cdot m_1 + 0 \cdot m_2 + 1 \cdot m_3 + 1 \cdot m_4 + 1 \cdot m_5 + 1 \cdot m_6 + 1 \cdot m_7$$
$$= m_3 + m_4 + m_5 + m_6 + m_7$$

$$Q = \overline{A}.\overline{B}.C + \overline{A}.B.C + A.B.\overline{C} + A.B.C$$

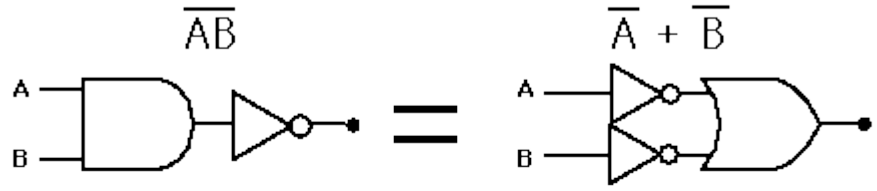
Sum of Products circuit in canonical form



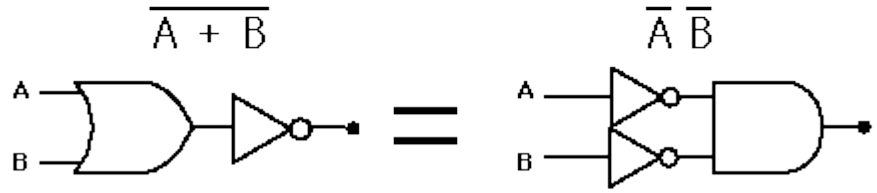
A mathematician named De Morgan developed a pair of important rules regarding group complementation in Boolean algebra.

De Morgan's Law

1) (A + B)′ = A′B′
2) (A.B)′ = A′ + B′



A NAND gate is equivalent to an inversion followed by an OR



A NOR gate is equivalent to an inversion followed by an AND

PROCEDURE:

1. Construct the circuit on breadboard for each Gate as shown in figures by inserting the appropriate IC.
2. Check the combinations of various inputs as shown in truth tables for each Gate.
3. If the input is low connect input to Ground, which indicates logic 0.
4. If input is high or logic 1 then connect the input to the power supply.
5. When output is high the LED will glow which indicates output as high, if the LED is not glowing then the output is low.

OBSERVATIONS:

De Morgan's Law:1

A	B	A* B	A*B
0	0		
0	1		
1	0		
1	1		

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

De Morgan's Law:2

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

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

RESULT:

Signature of Staff Member

Experiment No:4

4 X 1 MULTIPLEXER

AIM:

To construct and verify the 4X1 Multiplexer and verify the truth table.

APPARATUS REQUIRED:

COMPONENTS:

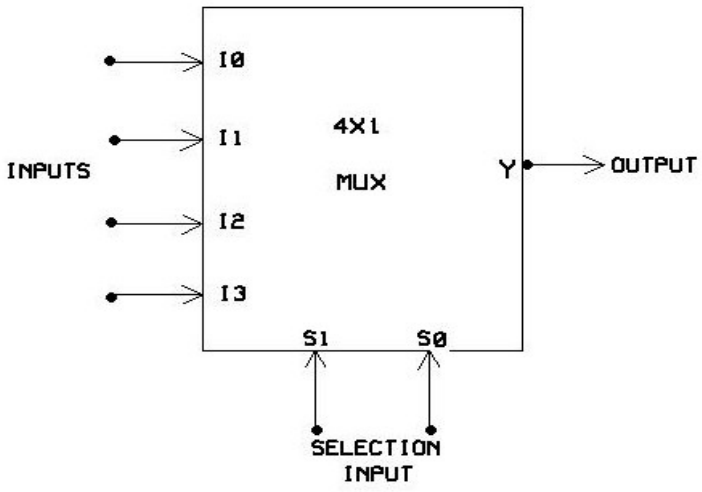
1. IC 7411
2. IC 7432
3. IC 7404
4. Resistors - 330 Ω -- 1 no's.
5. LED's --1 no's
6. Single lead probes.

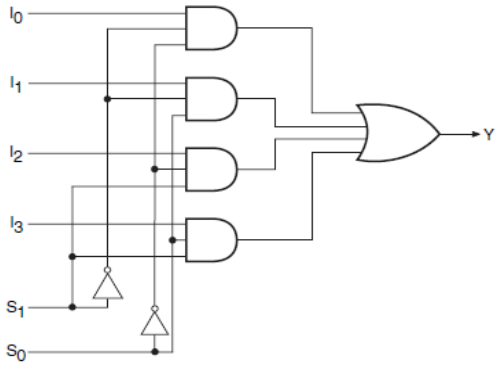
EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

A **Multiplexer** or mux is a combinational circuits that selects several analog or digital input signals and forwards the selected input into a single output line. A multiplexer of 2^n inputs has n selected lines, are used to select which input line to send to the output. A 4 * 1 line MUX, it is possible to implement any 2-variable function directly, simply by connecting I_0, I_1, I_2, I_3 to logic 1 in logic 0, as dictated by a truth table.



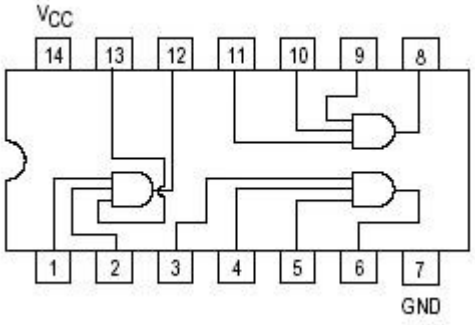


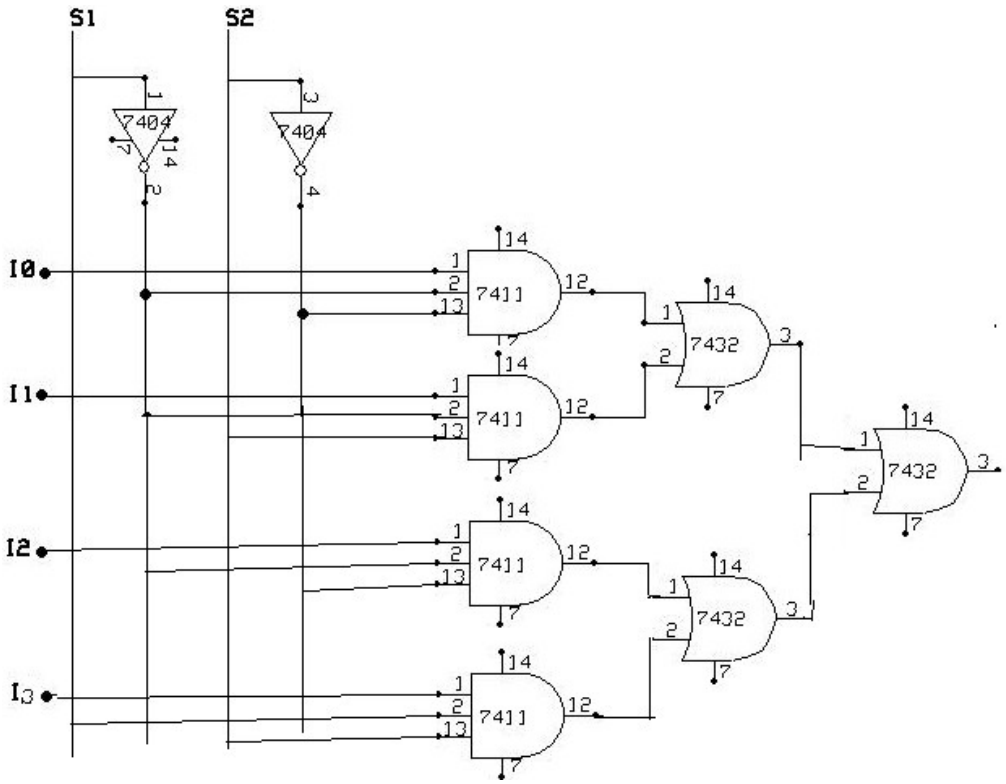
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

The standard ICs available for multiplexers are 2:1, 4:1, 8:1 and 16:1. The different digital ICs are given in Table. The IC 7411 a three input AND gate as shown in fig.

PROCEDURE:

1. Construct the circuit on breadboard for each Gate as shown in figures by inserting the appropriate IC.
2. Check the combinations of various inputs as shown in truth tables for each Gate.
3. If the input is low connect input to Ground, which indicates logic 0.
4. If input is high or logic 1 then connect the input to the power supply.
5. When output is high the LED will glow which indicates output as high, if the LED is not glowing then the output is low.





4*1 MUX Using IC 7411(3 Input AND), IC 7432(2 Input OR Gate) & IC 7404 NOT Gate

OBSERVATIONS: -

Function Table for Multiplexer

S_1	S_0	Output

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.(www.SureshQ.Blogspot.in)
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

RESULT:

Signature of Staff Member

Experiment No:5

1X4 DEMULTIPLEXER

AIM:

To construct and verify the 1X4 De Multiplexer and verify the truth table.

APPARATUS REQUIRED:

COMPONENTS:

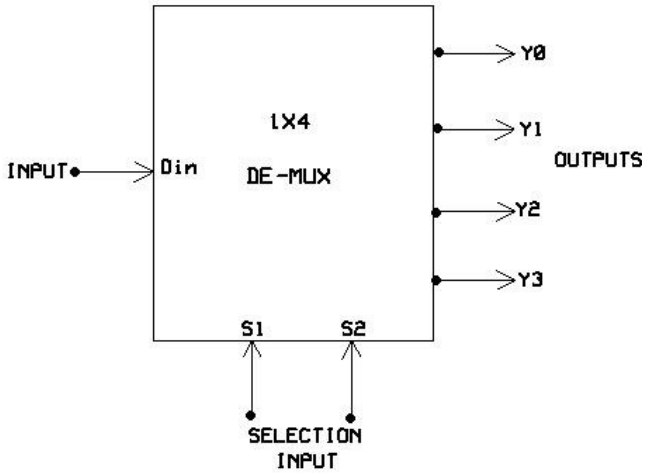
1. IC 7411
2. IC 7404
3. Resistors - 330 Ω -- 4 no's.
4. LED's --4 no's
5. Single lead probes.

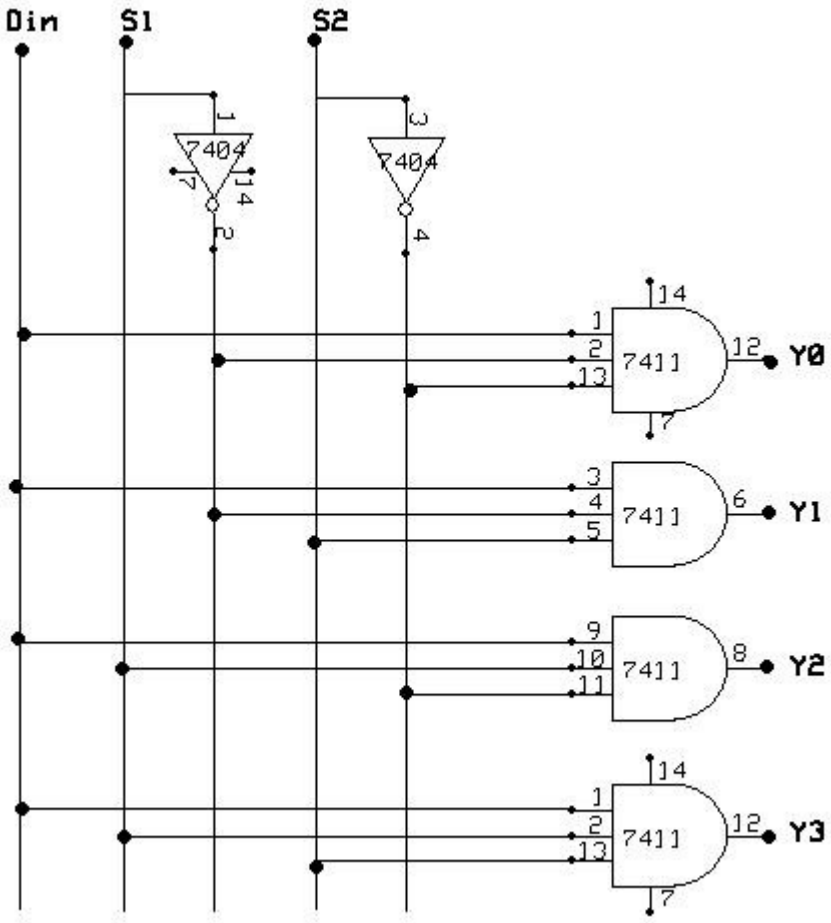
EQUIPMENT:

3. Power supply
4. Bread Board.

DEMULTIPLEXER

A *de-multiplexer* is a combinational logic circuit with an input line, 2^n output lines and n select lines. It routes the information present on the input line to any of the output lines. The output line that gets the information present on the input line is decided by the bit status of the selection lines.





1*4 DE MUX Using IC 7411(3 Input AND) & IC 7404 NOT Gate

PROCEDURE:

1. Connect the logic circuit diagram of 1*4 Demux on bread board as shown in figures.
1. Give power supply to pin no. 14 of each IC.
2. Ground the pin no 7 of each IC.
3. Connect LED's as an out put for Y_0, Y_1, Y_2 and Y_3.
4. Connect a 330 Ω resistor in series with each LED.
5. Check the truth table for all combinations.
6. When output is high the LED will glow, indicates logic 1.
7. When output is low the LED will not glow, indicates logic 0

OBSERVATIONS: -

Function Table for 4x1 Demultiplexer

S_1	S_0	Output Y_3	Output Y_2	Output Y_1	Output Y_0

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

RESULT:

Signature of Staff Member

Experiment No: 6

Half Adder & Full Adder

AIM:

To construct and check the truth tables for Half-Adder, Full-Adder. To construct a Full Adder using two Half Adders.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7408
2. IC 7432
3. IC 7486
4. LED - 2 no's
5. 330 Ω resistor-2 no's

EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

HALF ADDER

When two binary digits are added the output will be sum and carry. The two input variables are A and B, the outputs are taken as sum - S and carry -C. The logic symbol of Half- adder is shown in fig 1.

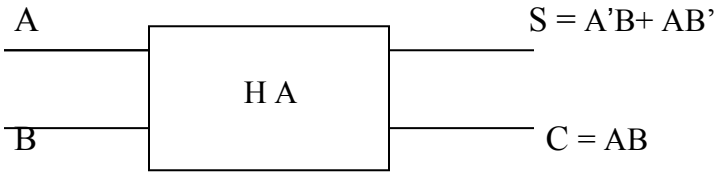


Fig 1. Half- adder Logic Symbol

The combinations for the sum and carry are written by the formulae are given as.

$$S = A'B + AB'$$

$$C = AB$$

The Half- adder can be constructed by using two logic gates named as EX-OR Gate and AND Gate. The EX-OR output is sum –S and output of AND Gate is a carry - C. The combinational circuit is as shown in fig-2.

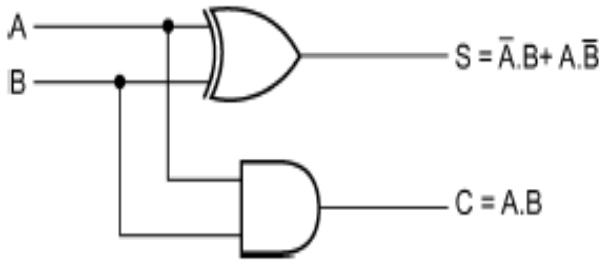


Fig 2. Half- adder

FULL ADDER

The Full –Adder circuit is used to add three binary digits. The two outputs are sum- S and carry-C. The three inputs are input A, input B and carry input C. The outputs are sum S and carry out X. The truth table is as shown in table-2.
The combinations for sum and carry output is given as

$$S = A'B'C + A'BC' + AB'C' + ABC$$

$$X = AB + BC + AC$$

The construction of Full adder using two EX-OR Gates, two AND Gates and one OR Gate is as shown in fig-4. The IC numbers are 7486, 7408 and 7432.

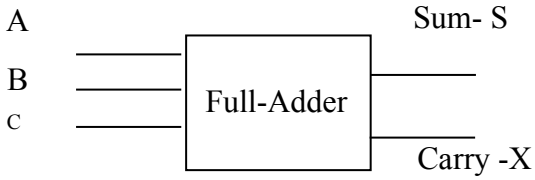


Fig 3. Full- adder Logic symbol

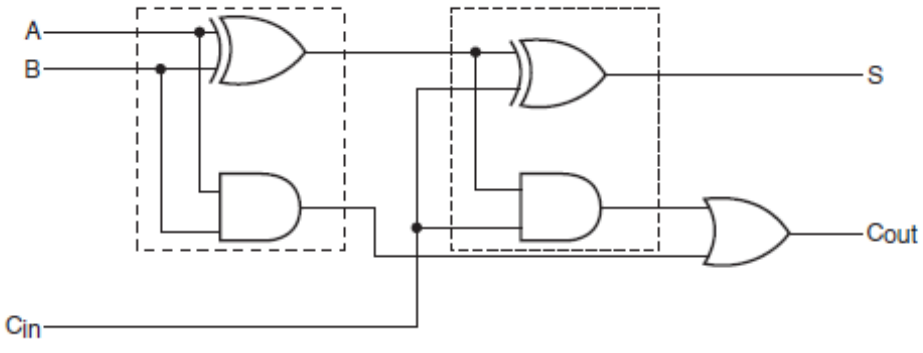


Fig 4. Full Adder

The construction of Full adder using two Half Adders is as shown in fig-5.

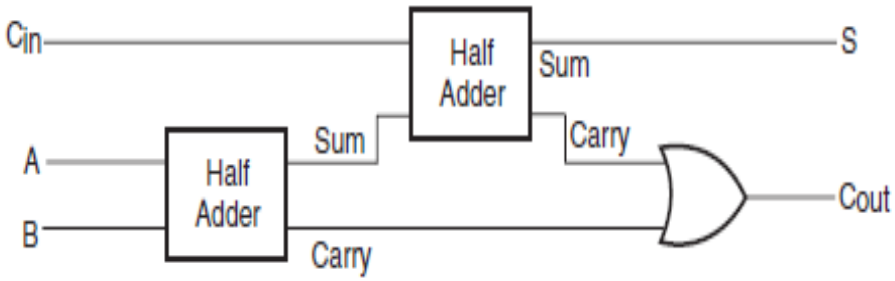


Fig 5. Full Adder Using Two Half Adders

PROCEDURE:

1. Connect the logic circuit diagram of Half adder, Full adder on bread board as shown in figures.
2. Give power supply to pin no. 14 of each IC.
3. Ground the pin no 7 of each IC.
4. Connect LED's as an out put for sum, carry, difference and barrow.
5. Connect a 330 Ω resistor in series with each LED.
6. Check the truth table for all combinations.
7. When output is high the LED will glow, indicates logic 1.
8. When output is low the LED will not glow, indicates logic 0

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.(www.SureshQ.Blogspot.in)
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+ 5V for TTL) or 0'level voltage to the inputs of the IC.
10. Don't connect the outputs of the gates to ground.

OBSERVATIONS:

Truth table for Half -Adder

A	B	S C	C
0	0		
0	1		
1	0		
1	1		

Truth table for Full Adder

A	B	C_{in}	S	C
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Karnaugh maps for the sum and carry-out of a full adder

Sum

Carry-out

S =

C =

RESULT:

Signature of Staff Member

Experiment No: 7

BASIC FLIP -FLOPS

AIM: To Study about Flip –Flop's with Synchronous & Asynchronous Modes.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7400
2. IC 7411
3. LED – 2 no.'s
4. Single lead probes.

EQUIPMENT:

1. POWER SUPPLY
2. BREAD BOARD.

THEORY:

A flip-flop has two stable states. One of the stable states is known as SET or LOGIC 1 where as the other stable is called RESET, CLEAR or LOGIC 0. It stores a binary digit either 0 or 1. There are two input terminals know as data inputs and two output terminals called Q and $\bar{Q}$. These two outputs are complementary at any instant of time Q=0 the $\bar{Q}$ =1 and vice versa.

The flip-flop has a clock indicated with CLK. A train of clock pulse is applied at this CLK input terminal of flip-flop, the state of output at any instance of time depends upon the past outputs and the present data inputs at that time. The state can be changed only after applying clock pulse.

Types of flip-flops:

1) SR flip flop
2) JK flip flop
3) D flip- flop
4) T flip-flop

Synchronous and Asynchronous Inputs

Most flip-flops have both synchronous and asynchronous inputs. Synchronous inputs are those whose effect on the flip-flop output is synchronized with the clock input. R, S, J, K and D inputs are all synchronous inputs. They force the flip-flop output to go to a predefined state irrespective of the logic status of the synchronous inputs.

Asynchronous inputs are those that operate independently of the synchronous inputs and the input clock signal. These are in fact override inputs as their status overrides the status of all synchronous inputs and also the clock input. PRESET and CLEAR inputs are examples of asynchronous inputs. When active, the PRESET and CLEAR inputs place the flip-flop Q output in the '1' and '0' state respectively.

S-R Flip-flop

The construction of a S-R flip-flop using only NAND gates is shown in fig 1. The clock terminal connected as short between two NAND gates. The inputs are applied according to the table shown in table no.1. When S=1 and R=0 after applying of clock pulse Q=1 and $\bar{Q}$ = 0,this state is know as set condition. When S=0 and R=1 when clock pulse is high Q=0 and $\bar{Q}$ = 1,

this is reset condition. For the combination S=0 and R=0 the outputs will be the previous outputs only. The combination S=1 and R=1 are not applied because the outputs will be toggling mode i.e. Outputs are not stable.

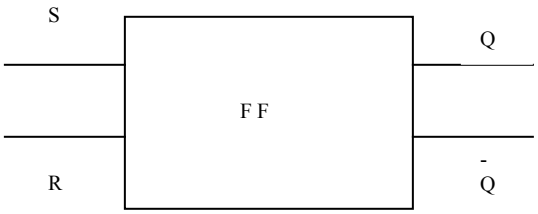


FIG 1. S-R LATCH

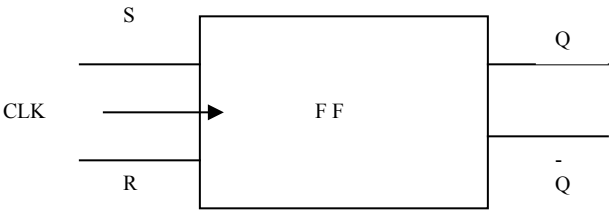
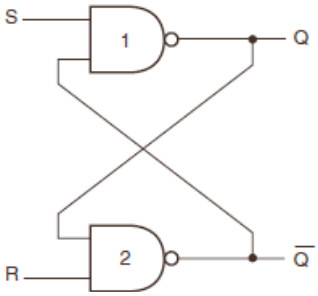


FIG 2. CLOCKED S-R FLIP-FLOP



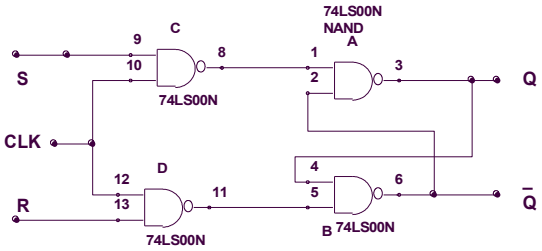


FIG 4. CLOCKED RS FLIP-FLOP

JK Flip-flop

The construction of JK flip-flop using NAND gates is shown in fig-.
When clock is applied if J=1 and K=0 then the outputs Q=1 and $\bar{Q}$ =0 this condition is called as SET. If J=0 and K=1 then the outputs Q=0 and $\bar{Q}$ =1 this is called as reset condition. The truth table for the J-K flip flop is shown below.

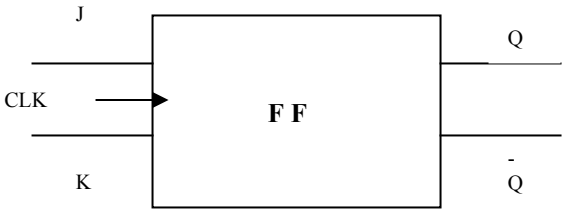


FIG 5 .J-K FLIP-FLOP

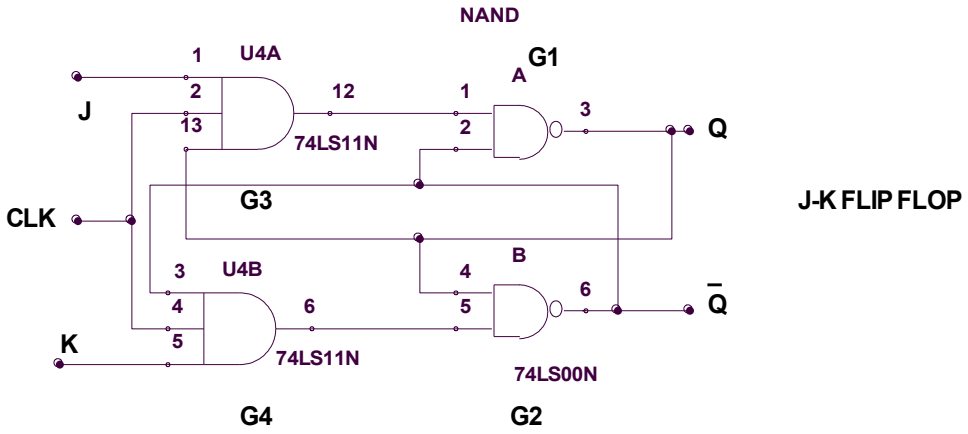


FIG 6. JK- FLIP-FLOP

D Flip-flop

The Construction of D- Flip-flop using JK Flip-flop as shown in fig 7. The D input is obtained by connecting a NOT Gate between J and K inputs. The Logic symbol of D flip-flop is shown in fig 8. When clock is applied if D=1 then the outputs Q=1 and $\bar{Q}$ =0. If D=0 then the outputs Q=0 and $\bar{Q}$ =1. The truth table for the D flip flop is shown below.

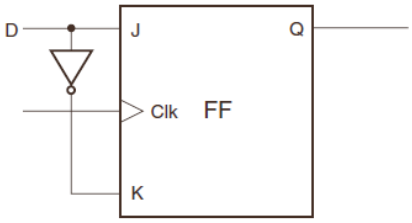


FIG 7 .J-K as D FLIP-FLOP

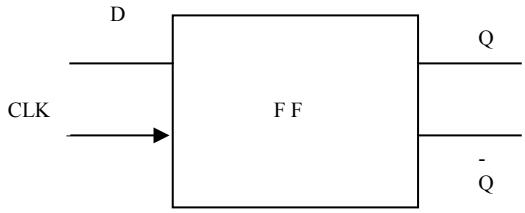


FIG 8. D- FLIP-FLOP

T Flip – Flop:

Connect J and K inputs of the flip -flop as a short as shown in fig 9. So that it acts like a T- flip flop. Observe the outputs according to the truth table shown in fig-.

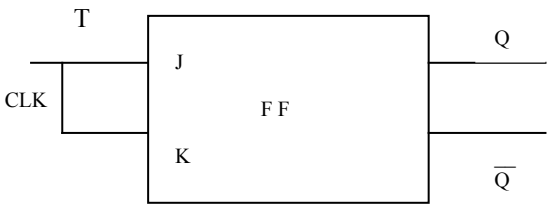


FIG 9 .J-K as T FLIP-FLOP

PROCEDURE :

SR Flip- Flop

1. Connect +5v to pin no-14 and pin no-7 is grounded for IC 7400 and IC 7411.
2. Connect the inputs S and R according to the characteristic table.
3. Observe the outputs Q and $\bar{Q}$.
4. In each case observe the $\bar{Q}$, that will be opposite to Q.

JK Flip- Flop

1. Connect +5v to pin no-14 and pin no-7 is grounded for IC 7400 and IC 7411.
2. Connect the inputs J and K according to the characteristic table.
3. Apply clock
4. Observe the outputs Q and $\bar{Q}$.
5. In each case observe the $\bar{Q}$, that will be opposite to Q.

D Flip- Flop

1. Connect +5v to pin no-14 and pin no-7 is grounded for IC 7404, IC 7400 and IC 7411.
2. The D input is obtained by connecting a NOT Gate between J and K inputs.
3. Apply the input D according to the characteristic table.
4. Apply clock
5. Observe the outputs Q and $\bar{Q}$.
6. In each case observe the $\bar{Q}$, that will be opposite to Q.

T Flip- Flop

1. Connect +5v to pin no-14 and pin no-7 is grounded for IC 7400 and IC 7411.
2. The T input is obtained by connecting J and K inputs.
3. Apply the input T according to the characteristic table.
4. Apply clock
5. Observe the outputs Q and $\bar{Q}$.
6. In each case observe the $\bar{Q}$, that will be opposite to Q.

OBSERVATIONS:

CHARACTERISTIC TABLE FOR SR FLIP FLOP			
S	**R**	Q_n	Q_{n+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

EXCITATION TABLE FOR SR FLIP FLOP				Comment
Q_n	Q_{n+1}	S_n	R_n	**STATE**
0	0			No Change
0	1			Reset
1	0			Set
1	1			Indeterminate/Forbidden

STATE TABLE FOR SR FLIP FLOP			Comment
S	**R**	Q_{n+1}	**STATE**
0	0		
0	1		
1	0		
1	1		

CHARACTERISTIC TABLE FOR JK FLIP FLOP			
J	**K**	Q_n	Q_{n+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

EXCITATION TABLE FOR JK FLIP FLOP				Comment
Q_n	Q_{n+1}	J_n	K_n	**STATE**
0	0			
0	1			
1	0			
1	1			

STATE TABLE FOR JK FLIP FLOP			Comment
J	**K**	Q_{n+1}	**STATE**
0	0		
0	1		
1	0		
1	1		

CHARACTARISTIC TABLE FOR D FLIP FLOP

D	Q_{n+1}	Comment
0	0	
0	1	
1	0	
1	1	

EXCITATION TABLE FOR D FLIP FLOP

Q_n	Q_{n+1}	D_n	Comment STATE

TRUTH TABLE FOR D FLIP FLOP

D	Q_n	Q_{n+1}	Comment STATE
0	0	0	
0	1	0	
1	0	1	
1	1	1	

CHARACTARISTIC TABLE FOR T FLIP FLOP

T	Q_n	Q_{n+1}	Comment STATE
0	0	0	
0	1	1	
1	0	1	
1	1	0	

EXCITATION TABLE FOR T FLIP FLOP

Q_n	Q_{n+1}	T_n	STATE

RESULT:

Signature of Staff Member

Experiment No:8

Implementation of Master Slave Flip-Flop with J-K Flip- Flop and verify the truth table for Race around condition

AIM: **Implementation of Master Slave Flip-Flop with J-K Flip- Flop and verify the truth table for Race around condition**.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7476
2. LED – 2 no.'s
3. Single lead probes.

EQUIPMENT:

1. POWER SUPPLY
2. BREAD BOARD.

THEORY:

Master Slave JK Flip-flop:

Whenever the width of the pulse clocking the flip-flop is greater than the propagation delay of the flip-flop, the change in state at the output is not reliable. This phenomenon is referred to as the *race problem*. As the propagation delays are normally very small, the occurrence of a race condition is reasonably high. One way to get over this problem is to use a *master–slave* configuration. Figure 1 shows a master–slave flip-flop. Figure 2 shows a master–slave flip-flop constructed with two J-K flip-flops.
The first flip-flop is called the master flip-flop and the second is called the slave. The clock to the slave flip-flop is the complement of the clock to the master flip-flop. When the clock pulse is present, the master flip-flop is enabled while the slave flip-flop is disabled. As a result, the master flip-flop can change state while the slave flip-flop cannot. When the clock goes LOW, the master flip-flop gets disabled while the slave flip-flop is enabled. Therefore, the slave J-K flip-flop changes state as per the logic states at its J and K inputs

The construction of Master Slave JK flip-flop using NAND gates is shown in fig-3 When clock is applied if J=1 and K=0 then the outputs Q=1 and $\bar{Q}$ =0 this condition is called as SET. If J=0 and K=1 then the outputs Q=0 and $\bar{Q}$ =1 this is called as reset condition. The truth table for the J-K flip flop is shown below. The JK flip flop can also be constructed by using NAND Gates.

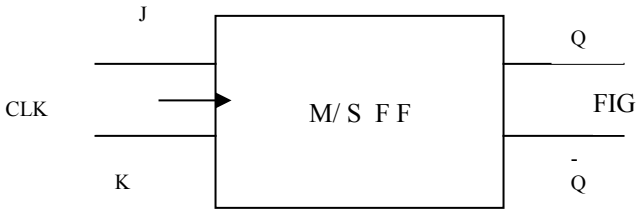


FIG 1 .Master Slave J-K Flip-Flop Symbol

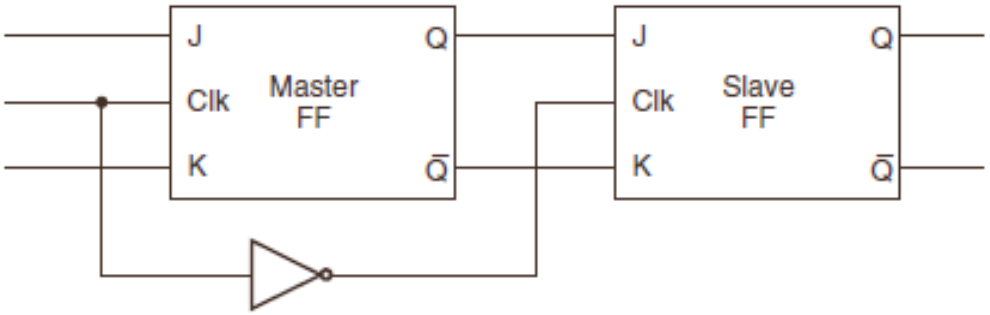


FIG 2 .Master Slave J-K Flip-Flop Using 2 J-K Flip-Flops

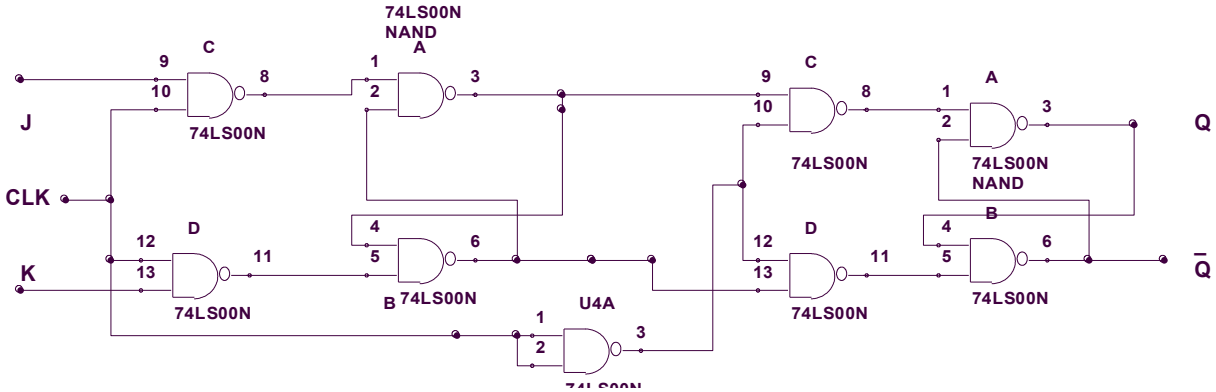
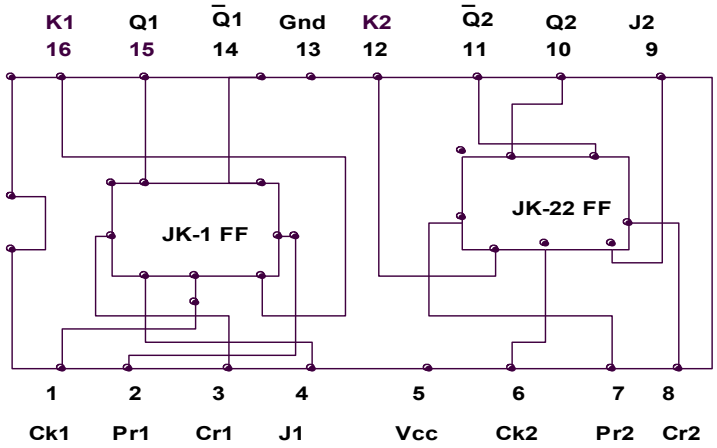


FIG 3 .Master Slave J-K Flip-Flop Construction using Only NAND Gates



IC 74 LS 76

FIG 4 . Master Slave J-K Flip-Flop IC

74 LS 76 PIN DETAILS		
1.CLK 1	7.Preset 2	13.GND
2.Preset 1	8.Clear 2	14.$\bar{Q}_1$
3.Clear 1	9.J_2	15.Q_1
4.J_1	10.$\bar{Q}_2$	16.K_1
5.Vcc.	11.Q_2	
6.Clk 2	12.K_2	

The Master slave JK flip flop IC –7476 is shown in fig-4 The IC 7476 is a dual JK master-salve flip- flop having preset and clear terminals, the Pin diagram is shown in table.

WORK PROCEDURE:

1. Connect +5v to pin no-5 and pin no-13 is grounded.

2. Connect the preset terminal to the ground for a short time and observe the output Q at that instant. It is logic 1.
3. Connect pin no-8 to ground for a short time an Observe the output Q that will be 0(reset).
4. In each case observe the $\bar{Q}$, that will be opposite to Q.
5. Verify the preset and clearing operation independent of JK inputs.
6. Connect the inputs J and K according to the characteristic table.
7. Observe the outputs Q and $\bar{Q}$.

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

OBSERVATIONS:

TRUTH TABLE FOR JK FLIP FLOP			
J	**K**	Q_n	Q_{n+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

CHARACTARISTIC TABLE FOR JK FLIP FLOP				Comment
Q_n	Q_{n+1}	J_n	K_n	
0	0			
0	1			
1	0			
1	1			

STATE TABLE FOR JK FLIP FLOP			Comment
J	**K**	Q_{n+1}	
0	0		
0	1		
1	0		
1	1		

RESULT:

Signature of Staff Member

Experiment No:9

Decade Counter

AIM:

To design and construct the Decade counter and verify the truth table.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7490
2. IC 7493
3. Resistors - 330 Ω -- 4 no's.
4. LED's -- 4 no's
5. Single lead probes.

EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

7490 decade (0-9) Counter

These are **ripple** counters. The Counter outputs respond to a clock pulse. The count advances as

the **clock** input becomes low (on the falling-edge), this is indicated by the bar over the clock label. This is the usual clock behavior of ripple counters and it means a counter output can directly drive the clock input of the next counter in a chain.

The counter is in two sections: clock A-Q_3 and clock B- Q_2-Q_1-Q_0. For normal use connect Q_A to clock B to link the two sections, and connect the external clock signal to clock A. For normal operation at least one reset 0 input should be low, making both high resets the counter to zero (0000, Q_3-Q_0 low). The 7490 has a pair of reset 9 inputs on pins 6 and 7, these reset the counter to nine (1001) so at least one of them must be low for counting to occur.

The counter consists of four flip-flops. It counts from 0000 to 1001 and the next clock pulse resets the counter to 0000. It skips six discrete states. The output waveform is shown in fig-. In 74LS90 IC there are four JK master slave flip-flops are employed. The arrangement is such that

J_0 and K_o of FF_o are always maintained at 1 level.
$J_1 = K_1 = T_1 = Q_0$
$J_2 = K_2 = T_2 = Q_0\ Q_1$
$J_3 = T_3 = Q_0\ Q_1\ Q_2$
$K_3 = Q_0$.

The counter works as follows:

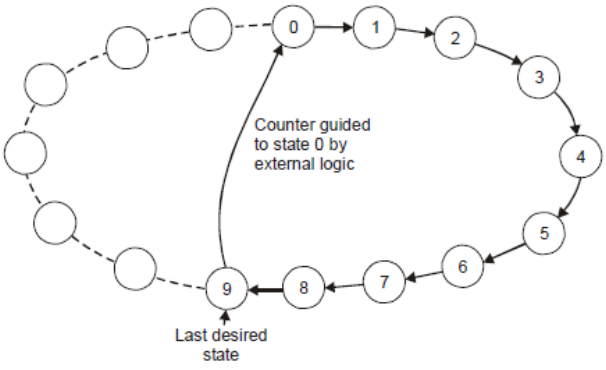
1. Let the counter be reset initially i.e., $Q_3Q_2Q_1Q_0$=0000
2. The first pulse causes FF_0 to toggle making $Q_3Q_2Q_1Q_0$= 0001. This makes T_1 = 1.
3. The second pulse makes both FF_0 and FF_1 to toggle. This makes $Q_3Q_2Q_1Q_0$ = 0010.

4. The third pulse makes FF_0 to toggle. This makes $Q_3Q_2Q_1Q_0$= 0011. But $T_2 = Q_1Q_0$.
5. Therefore T_2 goes to 1 level.
6. The fourth pulse makes FF_0, FF_1 and FF_2 to toggle making $Q_3Q_2Q_1Q_0$ = 0100.
7. The condition progresses in similar fashion till the seventh pulse J_3 has remained in 0 level.
8. $\bar{Q}_3$ was in 1 level. At the end of the second pulse Q_3 goes low. But J_3 also low. Hence $\bar{Q}_3$ remains in 1 level.
9. At end of the seventh pulse $Q_3Q_2Q_1Q_0$) = 0111.Now $T_1 = T_2 = T_3 = J_3 = K_3$= 1.
10. The eighth pulse makes all the flip flops as $J_3 = T_3$ = 1. hence $Q_3Q_2Q_1Q_0$=1000.
11. The ninth pulse makes FF_0 to toggle. Now $Q_3Q_2Q_1Q_0$ = 1001.It remains such that Q_3 = 1 and $\bar{Q}_3$ =0.
12. The tenth pulse makes FF_3 to toggle. FF_3 acts like a JK flip flop. AS J3 = 0 and K3 = 1, the input pulse makes Q_3 = 0 and $\bar{Q}_3$ = 1. Hence $Q_3Q_2Q_1Q_0$ reach 0000 at the end of tenth pulse.

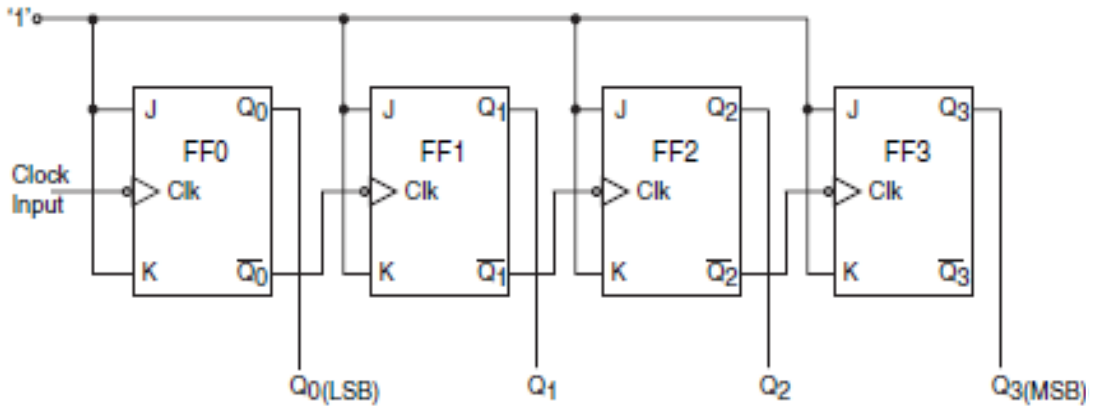
IC 74LS 90 is used to count up to 9, ie.0 to 9 or 10 numbers. By applying input pulses manually the output will be a number in the form of glowing LED's. The sequence of counting is as shown in below -.

IC 7490 **PIN DIAGRAM:**

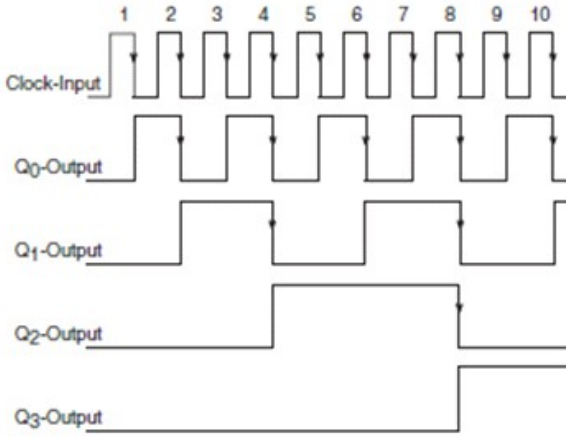
⇨

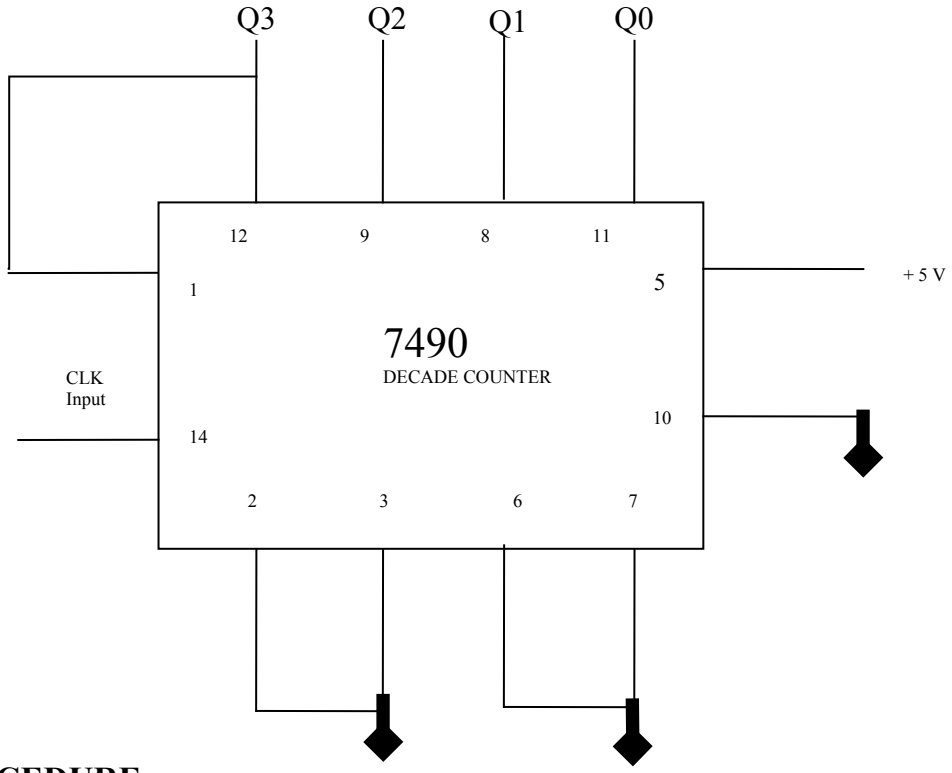


LOGIC CIRCUIT DIAGRAM:



Timing Diagram for Decade Counter:



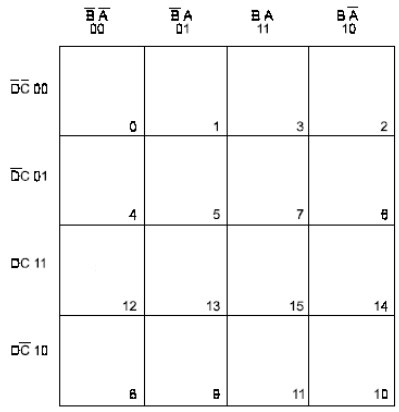


PROCEDURE:

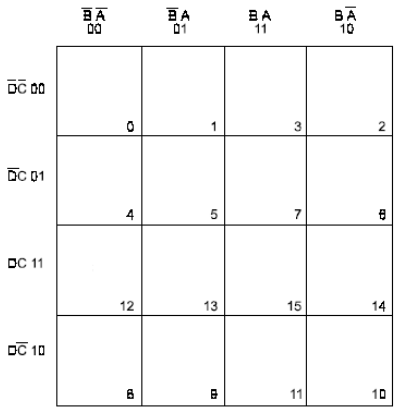
1. Setup the circuit as shown in fig-.
2. Now apply input signals at point A input manually (trigger pulses) and note sequence of counting and check the outputs as given in the table.

Counting											
Clk I/P	Present State $Q_3Q_2Q_1Q_0$ D C B A	Next State $Q_3Q_2Q_1Q_0$ D C B A	Decimal Number	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

J_3



K_3



J_2 K_2

J_1 K_1

J_0 K_0

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

RESULT:

Signature of Staff Member

Experiment No: 10

Mod - 6 Counter Using D – Flip-Flop

AIM:

To construct and check the Modulo - 6 Counter using D – Flip-Flop.

APPARATUS REQUIRED:

COMPONENTS:

1. IC 7474
2. IC 7400
3. Resistors - 330 Ω -- 8 no's.
4. LED's -- 4 no's
5. Single lead probes.

EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

In digital logic and computing, a **counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. In synchronous counters, the clock inputs of all the flip-flops are connected together and are triggered by the input pulses. Thus, all the flip-flops change state simultaneously (in parallel). Synchronous counters can also be implemented with hardware finite state machines, which are more complex but allow for smoother, more stable transitions.

The counter will therefore count from 000 to 101, and for a very short period of time, be in state 110 before the counter is cleared. This state is called the temporary state and the counter usually only remains in a temporary state for a few nanoseconds. We can essentially say that the counter skips 110 and 111 so that it goes only six different states; thus, it is a MOD-6 counter. The 111 state is the unused state here. In a state machine with unused states, we need to make sure that the unused states do not cause the system to hang, ie. no way to get out of the state. We don't have to worry about this here because even if the system does go to the 111 state, it will go to state 000, a valid state) on the next clock pulse.

PRINCIPLE:

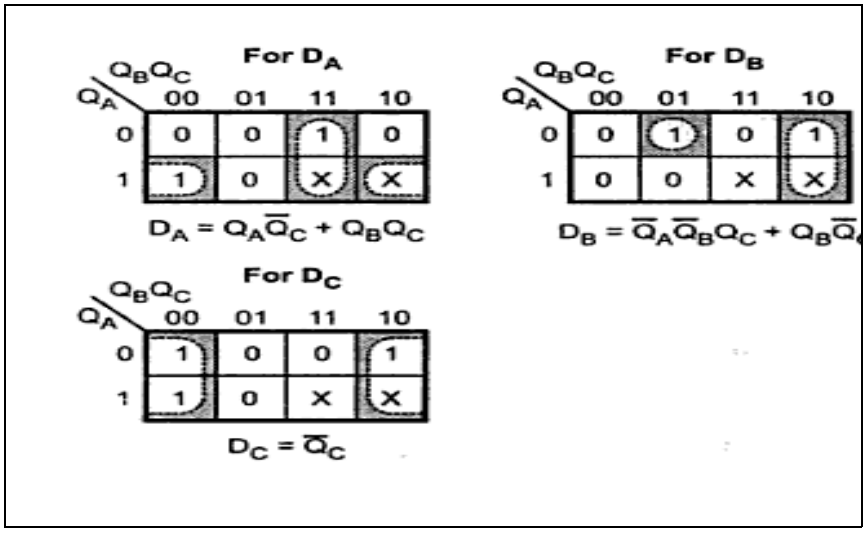
Three flip-flops are required to construct a Modulo –6 counter and decimal state 6 must be decoded and used to reset all flip-flops to give a repeated count from 0 to 5 (000 to 101). State 6 is given byQ_2Q_1Qo (101).

PROCEDURE:

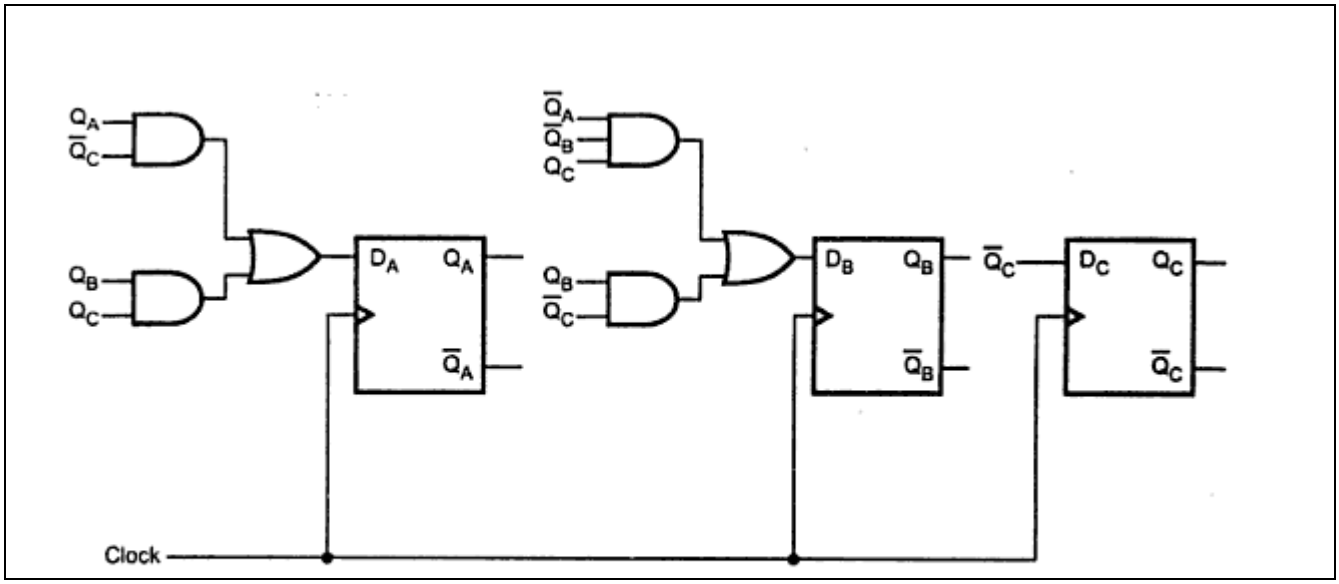
1. Setup the circuit as shown in fig-.
2. Now apply input signals at point A input manually (trigger pulses) and note sequence of counting and check the outputs as given in the table.

Transition table for Mod-6 counter

Present state			Next state		
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	x	x	x
1	1	1	x	x	x



K-map for MOD-6 Counter using D-Flip Flop



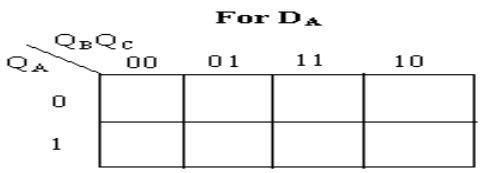
Implementation of Mod-6 Counter using D flip-flop

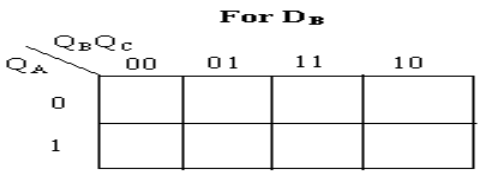
PRECAUTIONS:

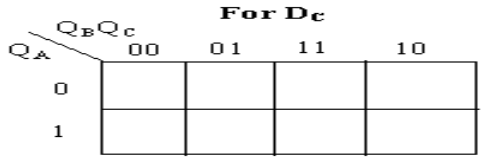
1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

OBSERVATIONS:

Counting						
Clk I/P	Present State $Q_CQ_BQ_A$ C B A	Next State $Q_CQ_BQ_A$ C B A	Decimal Number	D_2	D_1	D_0
1						
2						
3						
4						
5						
6						
7						
8						







k-map for MOD-6 COUNTER using D-Flip Flop

RESULT

-

Signature of Staff Member

Experiment No: 11

4-Bit Ring counter with T– Flip -Flop

AIM: To Construct 4-Bit Ring counter with T– Flip -Flop and verify the truth table.

APPARATUS REQUIRED:

COMPONENTS:

1. IC-7476
2. LED's ---- 4 No's
3. Single lead probes.

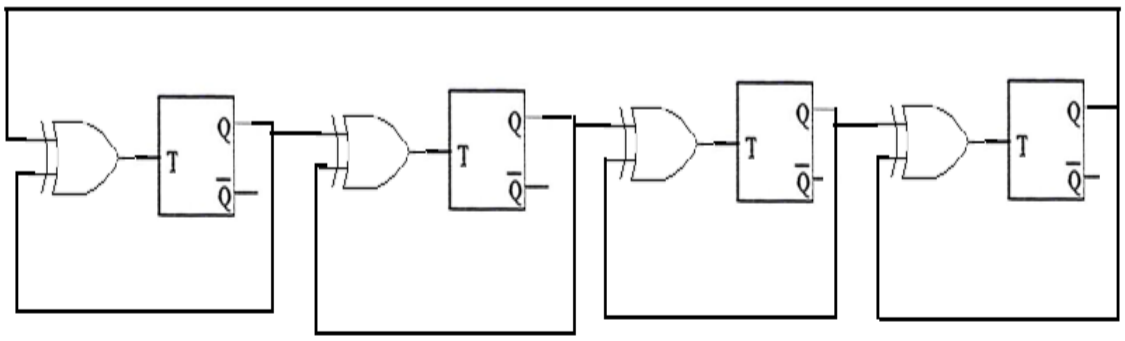
EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

Ring (Johnson) counters:

Johnson counters are a variation of standard ring counters, with the inverted output of the last stage fed back to the input of the first stage. They are also known as twisted ring counters. An n-stage Johnson counter yields a count sequence of length 2n, so it may be considered to be a mod-2^n counter. Figure shows a 4-bit Johnson counter.



PROCEDURE:

1. Setup the circuit as shown in fig-above.
2. Now apply input signals at point A input manually (trigger pulses) and note sequence of counting and check the outputs as given in the table.

TRUTH TABLE:

CLK PULSE	Q1	Q2	Q3	Q4
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

OBSERVATIONS: -

CLK PULSE	Q1	Q2	Q3	Q4
0				
1				
2				
3				

RESULT:

Signature of Staff Member

Experiment No: 12

Design a 8-Bit Right Shift Register using D–Flip -Flop and verify the truth table

AIM: To design a 8-Bit Right Shift Register using D–Flip -Flop and verify the truth table.

APPARATUS REQUIRED:

COMPONENTS:

1. IC-74164.
2. LED's ---- 4 No's
3. Single lead probes.

EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

SHIFT REGISTERS

A common form of register used in many types of logic circuits is a shift register. Shift register can present data to a device in a serial or parallel form, irrespective of the manner in which it is fed to a shift register. Shift register is simply a set of flip flops (usually D latches or RS flip flops) connected together so that the output of one becomes the input of the next, and so on in series. It is called a shift register because the data is shifted through the register by one bit position on each clock pulse.

IC 74164 is a 8 – bit Shift register is a serial-in parallel-out shift register is architecturally identical to a serial-in serial-out shift register except that in the case of the former all flip-flop outputs are also brought out on the IC terminals. Figure shows the logic diagram of a typical serial-in parallel-out shift register. In fact, the logic diagram shown in Fig. is that of IC 74164, a popular eight-bit serial-in parallel-out shift register. The gated serial inputs A and B control the incoming serial data, as a logic LOW at either of the inputs inhibits entry of new data and also resets the first flip-flop to the logic LOW level at the next clock pulse. Logic HIGH at either of the inputs enables the other input, which then determines the state of the first flip-flop. Data at the serial inputs may be changed while the clock input is HIGH or LOW, and the register responds to LOW-to-HIGH transition of the clock. Figure shows the relevant timing waveforms.

PRINCIPLE:

Registers are the group of flip-flops. Registers are sequential circuits and as they employ flip flops they possess memory. The simplest type of register is a data register, which is used for the temporary storage of data. The data register is a synchronous device, because all the flip flops change state at the same time. The other type is a Shift Register.

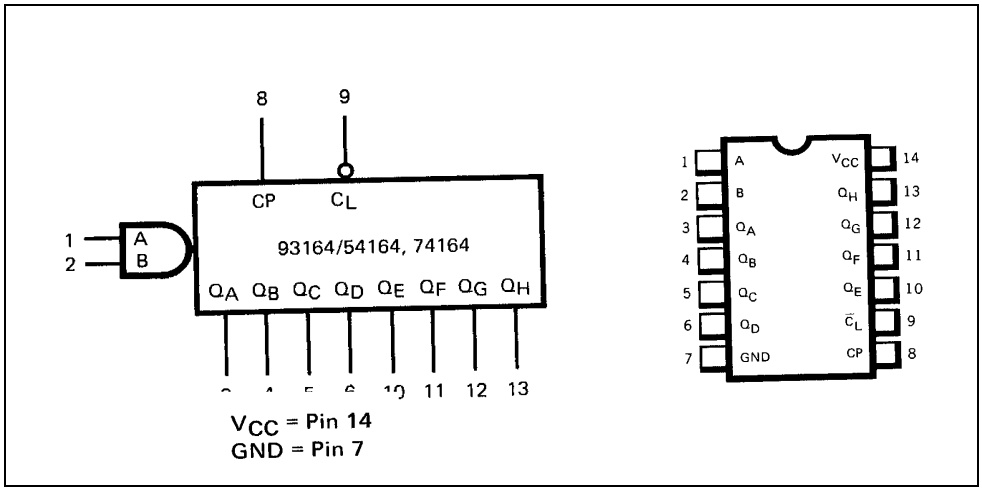
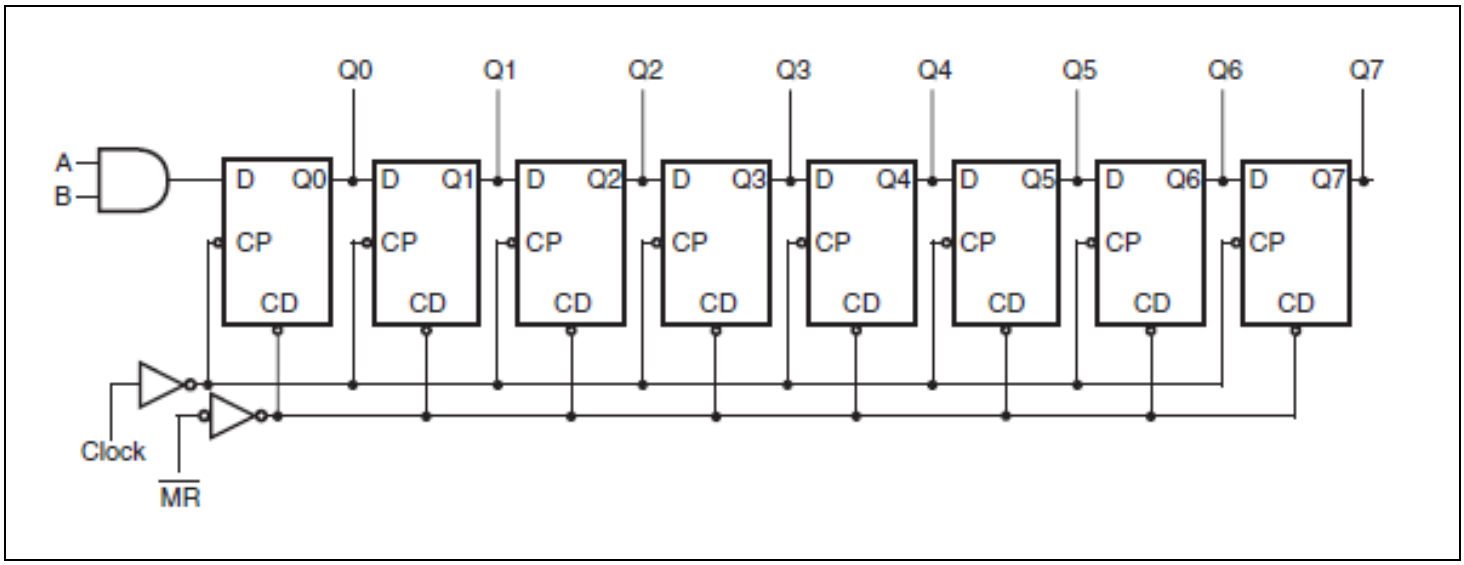


Fig: Pin diagram for IC 74164.



PROCEDURE:

1. Setup the circuit as shown in fig-above.
2. Now apply input signals at point A input manually (trigger pulses) and note sequence of counting and check the outputs as given in the table.

Let Input="11011010"

CLK	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	0	1	1	0	0	0	0	0
4	1	0	1	1	0	0	0	0
5	1	1	0	1	1	0	0	0
6	0	1	1	0	1	1	0	0
7	1	0	1	1	0	1	1	0
8	0	1	0	1	1	0	1	1

OBSERVATIONS: -

Let Input="10101100"

CLK	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0								
1								
2								
3								
4								
5								
6								
7								
8								

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

RESULT:

Signature of Staff Member

Experiment No: 13

Verify the Data Read and Data Write operations for the IC 74189

AIM: **Verify the Data Read and Data Write operations for the IC 74189.**.

APPARATUS REQUIRED:

COMPONENTS:

1. RAM trainer kit
2. Single lead probes.

THEORY:

PROCEDURE:

1. Setup the circuit as shown in fig-above.
2. Now apply input signals at point A input manually (trigger pulses) and note sequence of counting and check the outputs as given in the table.

OBSERVATIONS: -

PRECAUTIONS:

1. Always use a straight lead probe to insert into the breadboard.
2. Apply proper grounding for IC's.
3. Check the starting pin number for each IC indicated with a dot as starting pin.
4. Use IC remover to remove IC from breadboard to avoid damage of pins of IC.
5. Don't touch the pins of IC's while power on.
6. Don't bend the pins of IC's.
7. Insert the components into the breadboard firmly.
8. Loose contact may result in error at output.
9. Give 1 level Voltage (+5V for TTL) or 0 level voltage to the inputs of the IC.

RESULT:

Signature of Staff Member

Experiment No: 14

Design a Gray code encoder and interface it to SRAM IC 74189 for write operation display on 7- segment

AIM: **Verify the Data Read and Data Write operations for the IC 74189.**.

APPARATUS REQUIRED:

COMPONENTS:

1. IC-74189
2. Single lead probes.

EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

PROCEDURE:

OBSERVATIONS: -

PRECAUTIONS:

RESULT:

Signature of Staff Member

Experiment No: 15

Design a Gray code De-coder and interface it to SRAM IC 74189 for write operation display on 7- segment

AIM: **Verify the Data Read and Data Write operations for the IC 74189.**.

APPARATUS REQUIRED:

COMPONENTS:

1. IC-74189
2. Single lead probes.

EQUIPMENT:

1. Power supply
2. Bread Board.

THEORY:

OBSERVATIONS: -

PRECAUTIONS:

RESULT:

Signature of Staff Member