

CONTENTS

EXPT NO.	EXPERIMENT NAME	PAGE	DATE
1.	VERIFICATION OF UNIVERSAL GATES	3-10	3-08-15
2.	REALISATION OF A BOOLEAN EXPRESSION	11-14	17-08-15
3.	REALISATION OF BASIC GATES USING UNIVERSAL GATES	15-19	10-08-15
4.	DEMORGANS THEOREM	20-21	31-08-15
5.	REALISATION OF HALF ADDER AND FULL ADDER USING NAND GATE	22-23	24-08-15
6.	ADDER SUBTRACTOR USING IC7483	24-25	28-09-15
7.	BCD TO EXCESS 3 AND EXCESS 3 TO BCD	26-27	28-09-15
8.	SIMPLIFICATION OF LOGIC EXPRESSION USING 8:1 MUX	28-31	14-09-15
9.	FULL ADDER-SUBTRACTOR USING 3:8 DECODER	32-35	14-09-15
10.	1-BIT 2-BIT COMPARATOR	36-38	21-09-15
11.	7 SEGMENT DISPLAY	39-40	21-09-15
12.	RING COUNTER AND JOHNSON COUNTER	41-42	12-10-15
13.	DESIGN AND IMPLEMENTATION OF RS-FLIP FLOP D- FLIPFLOP AND T-FLIP FLOP	43-47	5-10-15
14.	DESIGN AND IMPLEMENTATION OF JK-FLIP FLOP AND JKMS FLIP FLOP	48-51	5-10-15
15.	VERILOG/HDL SIMULATION	52-55	12-10-15
16.	MOD -6 COUNTER	56-58	8-12-15

EXPERIMENT NO.1

Date : 3/8/2015

VERIFICATION OF BASIC GATES AND UNIVERSAL GATES

AIM: To verify the basic gates using IC's and universal gates.

APPARATUS REQUIRED:

1. IC trainer kit
2. IC 7408
3. IC 7432
4. IC 7404
5. IC 7400
6. IC 7402
7. Patch chords

THEORY:

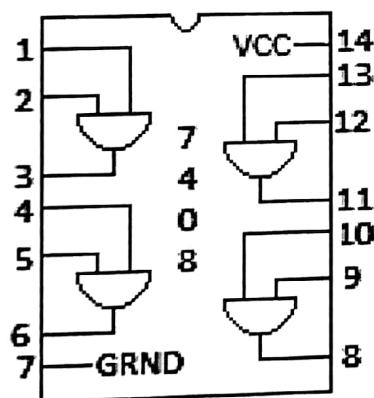
AND GATE

Design: 2 input AND gate – IC 7408

AND gate has 2 or more input and single output. It is an electronic device in which all the inputs must be high to get a high output. Its logical variable A and B are combined using AND multiplication. The resultant Y can be expressed as $Y=A \cdot B$.

Boolean Expression: $Y=A \cdot B$

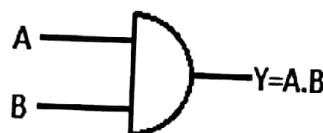
Pin Diagram:



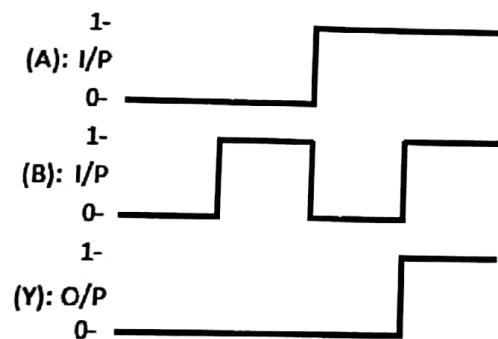
Truth Table:

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

Logic Diagram:



Timing diagram



OR GATE:

Design: 2 input OR gates - IC 7432.

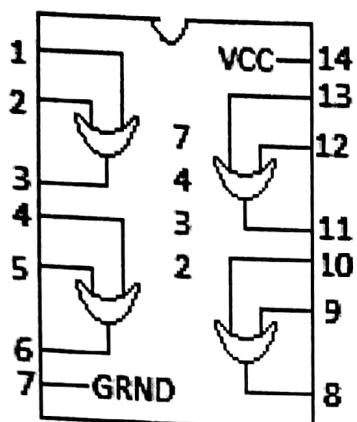
The circuit has 2 or more inputs and a single output. It is an electric circuit in which output will be high if any of the input variable is high and it will display the output.

The symbol for OR gate is '+'. The OR operation by relation $Y=A+B$ where A and B are the inputs and Y is the output.

Boolean expression:

$$Y=A+B$$

Pin diagram:



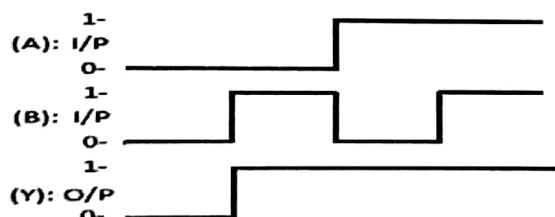
Truth table:

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

Logic diagram:



Timing diagram:



NOT GATE:

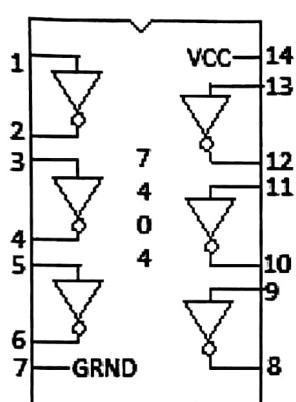
Design: NOT Gate - IC 7404

The gate that performs mathematical operation by taking the complement is called NOT gate is called the inveter and the operation of taking the complement is called inversion.

A NOT gate circuit has a single input and an output. The expression of NOT gate is given by $Y=A'$ where A is the input.

Boolean Expression: $Y=A'$

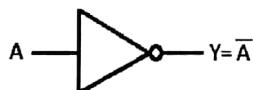
Pin Diagram:



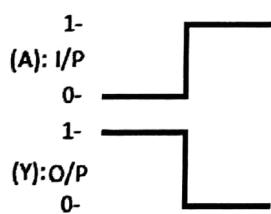
Truth Table:

A	Y
0	1
1	0

Logical Diagram:



Timing diagram:



NAND GATE:

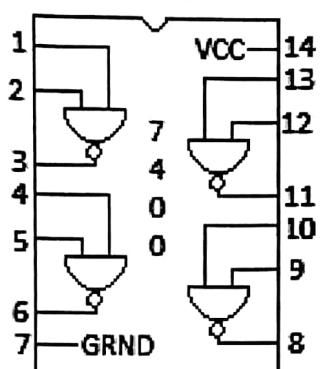
Boolean Expression: $Y = \neg(A \cdot B)$

Design: NAND gate - IC 7400

The NAND is contradiction of NOT-AND gate. The negative following an NAND gate. It is cascaded combination of NAND gate and NOT gate A and B are the 2 inputs, Y is the output.

NAND is represented by $Y = \neg(A \cdot B)$

Pin Diagram:



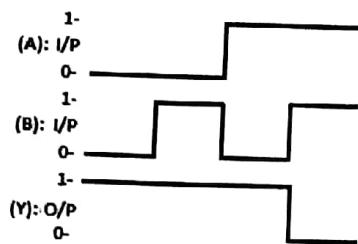
Truth Table:

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

Logical Diagram:



Timing diagram



NOR GATE:

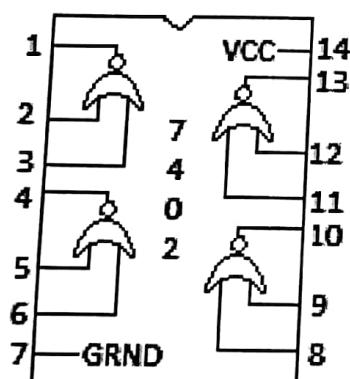
Design: NOR gate - IC 7402

The NOR gate is the combination of NOT-OR gate. The negation of OR gate is called NOR gate. A NOR gate is the cascaded combination of NOT gate and OR gate.

NOR gate is separated by $Y = \sim(A+B)$, A and B are the two inputs, Y is the output. In NOR gate output will be high if both the inputs are lower else Output is low.

Boolean Expression: $Y = \sim(A+B)$

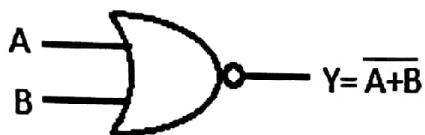
Pin diagram:



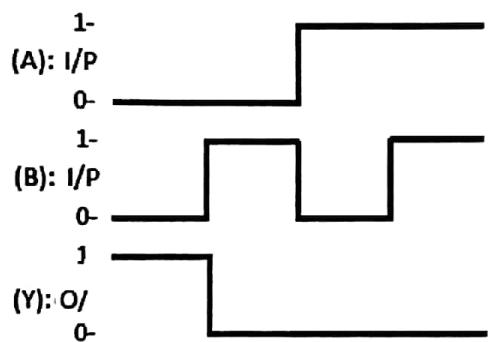
Truth Table:

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

Logic Diagram:

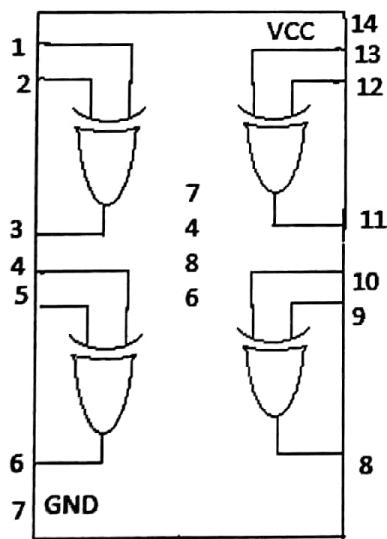


Timing diagram:



XOR GATE:

Pin diagram:



Truth table:

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

Logic diagram:



Design: The exclusive-OR gate has a high output only when an odd number of inputs is high. It checks whether it consists of even number of paratives .

Boolean expression: $Y = A'B + AB'$

PROCEDURE :

1. Insert the IC in one of the pin sockets in the IC trainer and lock the socket. identify the pin numbers.
2. Using patch chords connect +VCC and GND as per the pin diagram.
3. Rigup the circuit as per the logic diagram realized.
4. Connect the power supply to the IC trainer kit.
5. Using the HIGH_LOW input switches give the different input combinations.
6. For each input combination monitor the output and verify the truth table.

RESULT: All gates are verified and hence the truth table is verified.

EXPERIMENT NO:2

Date: 17/8/2015

SIMPLIFICATION OF BOOLEAN EXPRESSION USING LOGIC GATES

AIM: To simplify the given boolean expressions using Logic gates.

APPARATUS REQUIRED:

1. IC trainer kit
2. IC 7408
3. IC 7404
4. IC 7432
5. IC 7400
6. IC 7402
7. Patch chords

THEORY:

Simplification using boolean algebra:

$$\begin{aligned} 1) F(A,B,C,D) &= A'B'C'D + AB'C'D + BD + BCD' \\ &= B'C'D + BD (1) + BCD' \\ &= B'C'D + CBD + C'BD + BCD' \\ &= BC (D + D') + C'BD + B'C'D \\ &= BC + C'D (B' + B) \\ &= BC + C'D \end{aligned}$$

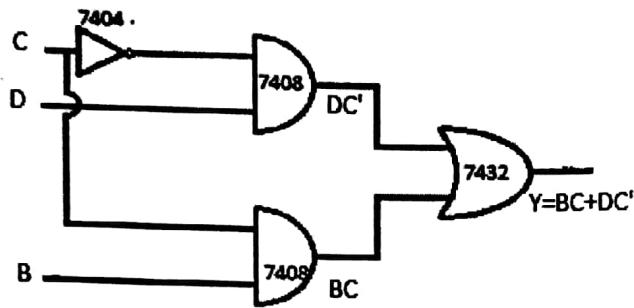
Truth table for Y1:

A	B	C	D	A'	B'	C'	D'	A'B'C'	AB'C'D	BCD'	BD	Y1
0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	1	1	1	1	0	1	0	0	0	1
0	0	1	0	1	1	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0
0	1	0	0	1	0	1	1	0	0	0	0	0
0	1	0	1	1	0	1	0	0	0	0	1	1
0	1	1	0	1	0	0	1	0	0	1	0	1
0	1	1	1	1	0	0	0	0	0	0	1	1
1	0	0	0	0	1	1	1	0	0	0	0	0
1	0	0	1	0	1	1	0	0	1	0	0	1
1	0	1	0	0	1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	1	1
1	1	1	0	0	0	0	1	0	0	1	0	1
1	1	1	1	0	0	0	0	0	0	0	1	1

TRUTH TABLE FOR Y2:

A	B	C	D	C'	BC	C'D	Y2=BC+C'D
0	0	0	0	1	0	0	0
0	0	0	1	1	0	1	1
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	1	1
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	1	1
1	1	1	0	0	1	0	1
1	1	1	1	0	1	0	1

Logical diagram:



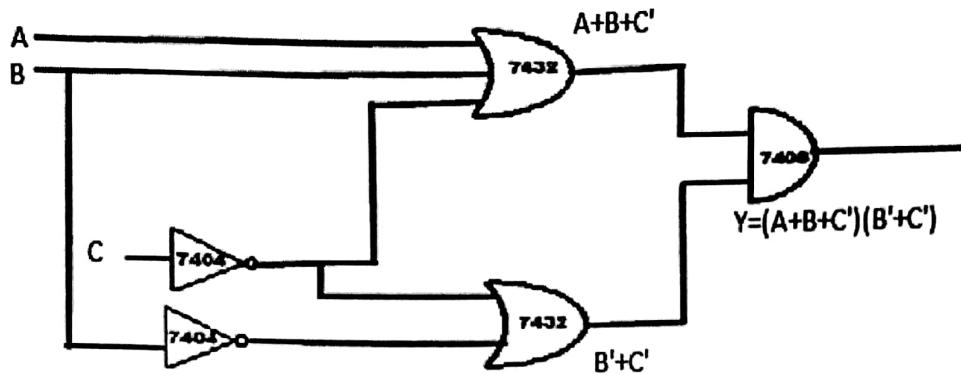
$$2) F(A,B,C,D) = \pi m(2,3,6,7,14,15)$$

$$\begin{aligned}
 &= (A+B+C'+D) \quad (A+B+C'+D') \quad (A+B'+C'+D) \quad (A+B'+C'+D') \quad (A'+B'+C'+D) \quad (\\
 &\quad A'+B'+C'+D') \\
 &= (A+B+C'+D \cdot D') \quad (A+B'+C'+D \cdot D') \quad (A'+B'+C'+D \cdot D') \\
 &= (A+B+C') \quad (A+B'+C') \quad (A'+B'+C') \\
 &= (A+B+C') \quad (A \cdot A' + B' + C') \\
 &= (A+B+C') \quad (B'+C')
 \end{aligned}$$

Truth table :

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

Logic diagram:



PROCEDURE:

1. Insert the IC in one of the pin sockets in the IC trainer and lock the socket. Identify the pin numbers.
2. Using patch chords connect +VCC and GND as per the pin diagram.
3. Rig up the circuit as per the logic diagram realized.
4. Connect the power supply to the IC trainer kit.
5. Using the HIGH_LOW input switches give the different input combinations.
6. For each input combination monitor the output and verify the truth table.

RESULT: Hence the expressions are verified using the truth table.

EXPERIMENTNO:3

Date:10/8/2015

REALIZATION OF BASIC GATES USING UNIVERSAL GATES

AIM: To realize the basic gates using universal gates.

APPARATUS REQUIRED:

- 1.IC trainer kit
- 2.IC 7400
- 3.IC 74002
- 4.Patch chords

THEORY:

➤ OR GATE USING NOR - IC 7402

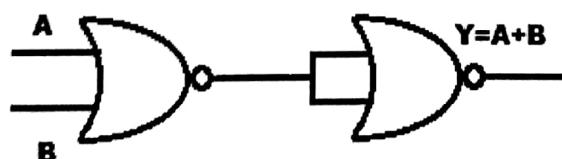
Design: Consider two inputs A and B. When we complement a NOR we get an OR gate.

Boolean expression: $Y=A+B$

Truth table:

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

Logic Diagram



OR GATE USING NAND 7400

Design: Consider two inputs A and B.

By shorting 2 NAND gates we get a NOT gate. The combination of the NOT gate and the NAND gate we get OR gate.

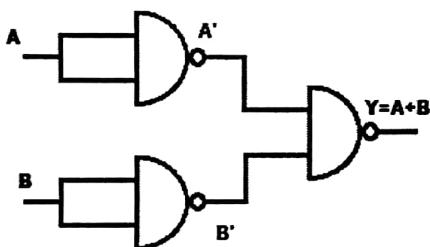
Boolean expression: $Y=A+B$

➤ Truth table:

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

➤ Here $Y=A+B$

➤ Logic Diagram:



NOT GATE using NOR IC 7402

➤ Truth Table:

A	Y
0	1
1	0

➤ Here $Y=A'$

➤ Logic Diagram:



Design: Consider A as the only input.

By shorting NOR gate we get NOT gate.

Boolean expression: $Y = A'$

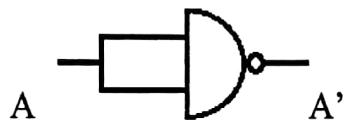
➤ **NOT GATE USING NAND - IC 7400**

➤ **Truth Table:**

A	Y
0	1
1	0

Here $Y = A'$

➤ **Logic Diagram:**



Design: Consider A as the only input.

By shorting NAND gate we get NOT gate.

Boolean Expression: $Y = A'$

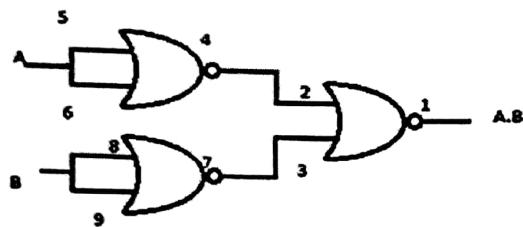
AND GATE USING NOR - IC 7402

Truth Table:

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

Here $Y = A \cdot B$

Logical Diagram



Design: Consider A and B as the inputs.

By shorting to 2 NOR gates we get a NOT gate. The combination of the NOT gate and the NOR gate we get AND gate.

Boolean Expression: $Y=A \cdot B$

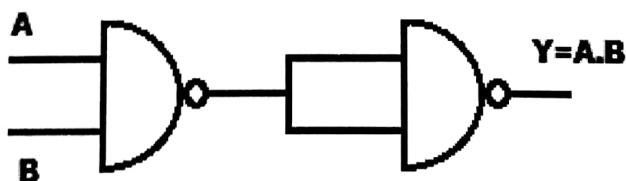
AND GATE USING NAND - IC 7400

Truth Table:

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

Here $Y=A \cdot B$

Logical diagram:



Design: Consider two inputs A and B.

By complementing the NAND Gate we get AND gate.

Boolean expression: $Y = A \cdot B$

PROCEDURE:

1. Insert the IC in one of the pin sockets provided in the IC trainer and lock the socket. Identify the pin numbers.
2. Using patch chords connect +VCC and GND as per the pin diagram.
3. Rig up the circuit as per the logic diagram realized.
4. Connect the power supply to the IC trainer kit.
5. Using the HIGH_LOW input switches give the different input combinations.
6. For each input combination monitor the output and verify the truth table.

RESULT: All the truth tables are verified , hence the logic circuit is also verified.

VERIFICATION OF DE MORGAN'S LAWS

AIM: To verify de- morgan laws

1. Bubbled OR gate = NAND gate ($\overline{A \cdot B} = \bar{A} \cdot \bar{B}$)
- 2 .Bubbled AND gate = NOR gate ($\overline{A + B} = \bar{A} + \bar{B}$)

APPARATUS REQUIRED:

1. IC 7400
2. IC 7402
3. IC 7404
4. IC 7408
5. IC 7432
6. Patch chords
7. Digital IC trainer kit

THEORY:

➤ DE MORGANS 1ST LAW:

Complement of the product is equal to sum of the complements.

If we complement the output of NAND gates and bubbled OR gate i.e.inputs of OR gate are inverted ,the results are same.Therefore output of NAND gate and bubbled OR gate from the truth table also remains the same. This is de-morgans first law i.e. $\overline{A \cdot B} = \bar{A} + \bar{B}$

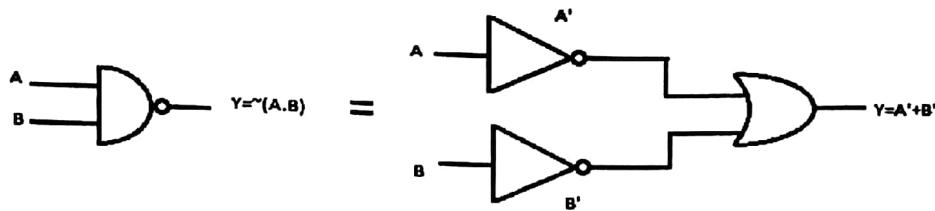
Boolean expression:

$$\overline{(A + B)} = \bar{A} \cdot \bar{B}$$

Truth table:

A	B	$A \cdot B$	$\sim(A \cdot B)$	$(\bar{A} + \bar{B})$
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

Logic circuit:



➤ DE-MORGANS 2ND LAW:

Complement of sum is equal to product of the complements.

If we complement the output of NOR gate and bubbled AND gate, the results will be same. Therefore output of NOR gate and bubbled AND gate from the truth table also remains same. The complement of sum is equal to the product of complement. This is de-morgans second law i.e. $A + B = \bar{A} \cdot \bar{B}$.

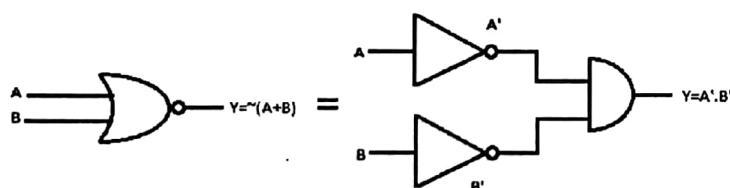
Boolean expression:

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

Truth table:-

A	B	A+B	$\sim(A+B)$	$(\bar{A} \cdot \bar{B})$
0	0	0	1	1
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

Logic circuit



PROCEDURE:

1. Insert the IC in one of the pin sockets provided in the IC trainer and lock the socket. Identify the pin numbers.
2. Using patch chords connect +VCC and GND as per the pin diagram.
3. Rig up the circuit as per the logic diagram realized.
4. Connect the power supply to the IC trainer kit.
5. Using the HIGH_LOW input switches give the different input combinations.
6. For each input combination monitor the output and verify the truth table

RESULT: Thus the de-morgan laws are verified using the truth table.

EXPERIMENT 05

DATE 24-08-15

HALF ADDER AND FULL ADDER USING NAND AND NOR GATES.

AIM: IMPLEMENTATION OF HALF ADDER AND FULL ADDER USING NAND GATE

COMPONENTS:

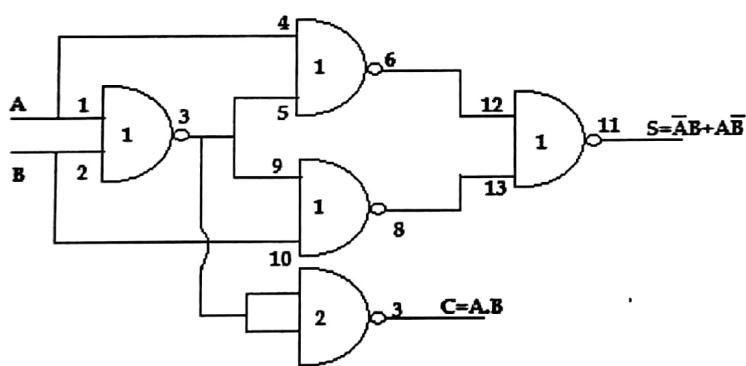
: IC transfer kit

: IC 7400

: PATCH CORDS

TRUTH TABLE:

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

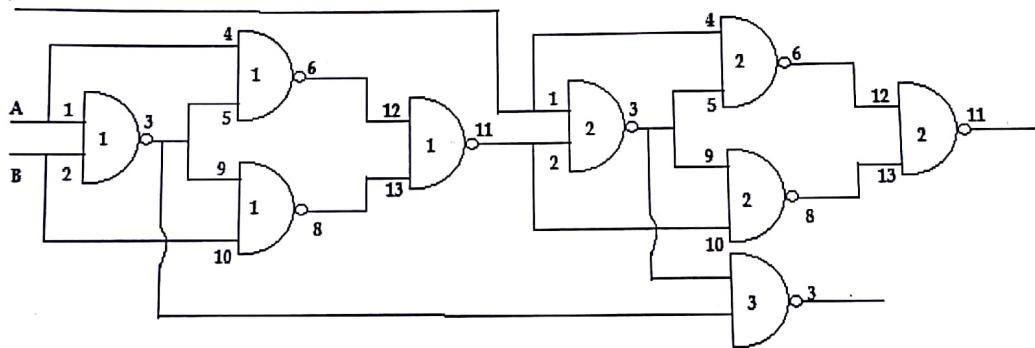


HALF ADDER: half adder adds the 2 one bit binary numbers A and B. It has two outputs sum and carry. The logic circuit used for this is called half adder.

Procedure: Place the IC 7400 into the digital integrated circuit and connect wires into the terminals as shown in the figure .Different carry and sum value are generated at different input combination.

FULL ADDER: It has 3 inputs where as half adder has 2 inputs. For a full adder there is 3 inputs A,B,C . The output is carry and sum output is XOR between input &A& the half adder sum output , with B and C as inputs. Carry will be high if anyone of 2 inputs out of 3 inputs are high.

A	B	C	S	C
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



RESULT: SUM is generated only when A is zero and B is one or B is zero and A is one. Carry is high when both the inputs are high.

IMPLEMENTATION OF ADDER / SUBTRACTOR
CIRCUIT USING IC7483

AIM:-Design and implementation of adder/subtractor using IC7483 .

APPARATUS REQUIRED:-

- 1).IC trainer kit
- 2.)IC 7483
- 3.)IC 7486
- 4.)Patch chords

THEORY:-

- i. **Adder**:-The circuit is laid out from left to right, similar to the way we add binary numbers . Therefore the least significant bit is on the right and the most significant bit on the left .The full adder circuits adds three bit binary numbers, sum and carry. When the select line is 0 it acts as a adder.
- ii. **Subtractor**:-The full subtractor is combinational circuit which is led to perform subtraction of these bits. It has three inputs and two outputs namely D(difference) and B(borrow). When the select line is 1 it acts as a subtractor.

Truth table:-

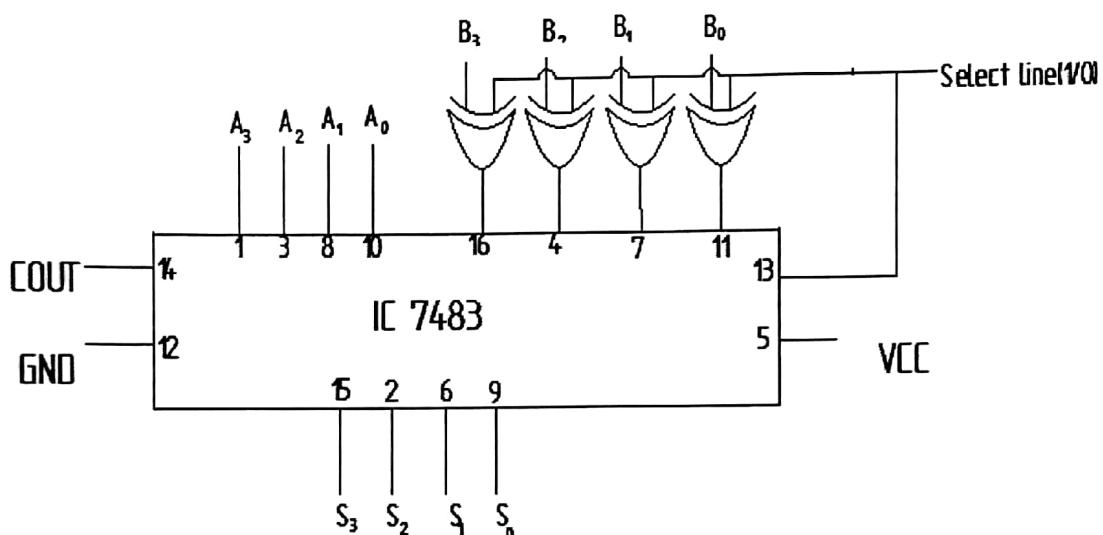
A	B	C	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Truth table for adder circuit

A	B	C	DIFFERENCE	BORROW
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

Truth table for subtractor circuit

Logic diagram :-



PROCEDURE:

- 1 Insert the IC in one of the pin sockets in the IC trainer and lock the socket.
- Identify the pin numbers.
- 2 Using patch chords connect +VCC and GND as per the pin diagram.
- 3 Rig up the circuit as per the logic diagram realized.
- 4 Connect the power supply to the IC trainer kit.
- 5 Using the HIGH_LOW input switches give the different input combinations.
- 6 .For each input combination monitor the output and verify the truth table.

RESULT:-The full adder and full subtractor using IC 7483 is implemented and verified.

IMPLEMENTATION OF BCD TO EXCESS-3CODE AND EXCESS-3 TO BCD

AIM:

To design and realize the following using IC 7483.

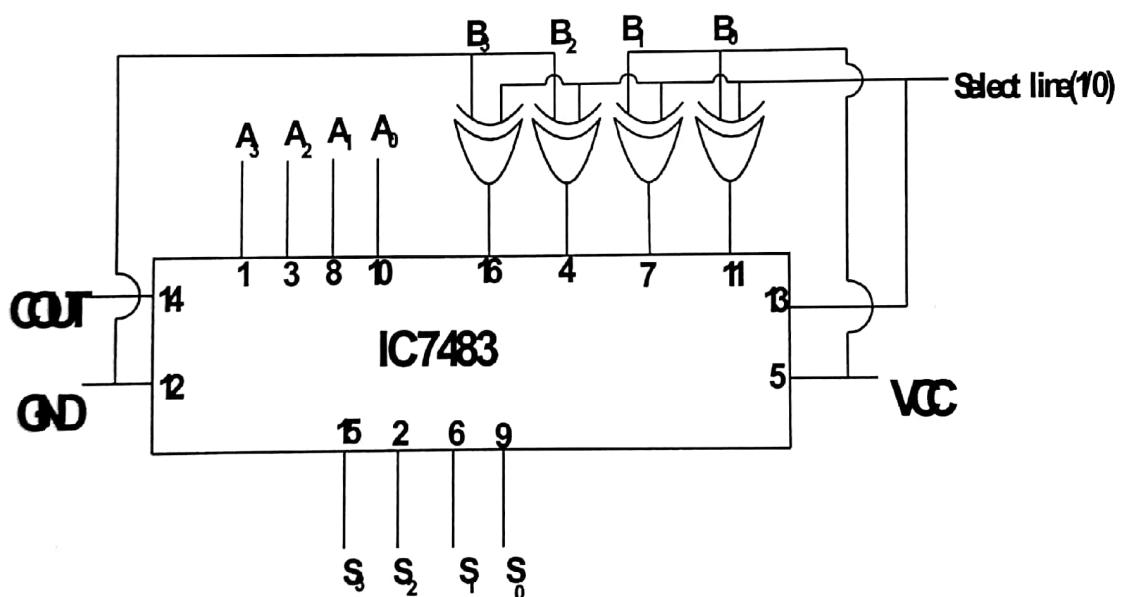
- BCD to Excess- 3 Code
- Excess-3 to BCD Code.

APPARATUS REQUIRED:-

1. IC trainer kit
2. IC 7483
3. Patch chords

THEORY: -

Pin Diagram using IC7483:



PROCEDURE:

- 1 . Insert the IC in one of the pin sockets in the IC trainer and lock the socket.
- Identify the pin numbers.
2. Using patch chords connect +VCC and GND as per the pin diagram.
3. Rig up the circuit as per the logic diagram realized.
4. Connect the power supply to the IC trainer kit.
5. Using the HIGH_LOW input switches give the different input combinations.
6. For each input combination monitor the output and verify the truth table.

Code converter is a combinational circuit that translates the input code word into a new corresponding word. The excess-3 code digit is obtained by adding three to the corresponding BCD digit. To Construct a BCD-to-excess-3-code converter with a 4-bit adder feed BCD- code to the 4-bit adder as the first operand and then feed constant 3 as the second operand. The output is the corresponding excess-3 code.

To make it work as a excess-3 to BCD converter, we feed excess-3 code as the first operand and then feed 2's complement of 3 as the second operand. The output is the BCD code

- Apply BCD as first operand(A) and binary 3 as second operand(B) and $\text{cin}=0$ for Realizing BCD-to-Excess-3-code:
- Apply Excess-3-code code as first operand(A) and binary 3 as second operand(B) and $\text{Cin}=1$ for realizing Excess-3-code to BCD.

RESULT: Realized BCD code to Excess-3 code conversion and vice versa using IC7483.

EXPERIMENT NO. 6

SIMPLIFICATION OF LOGIC EXPRESSION USING 8:1 MUX

Aim:

Given any four variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.

Components required: IC trainer kit, IC 74151[8:1 MUX], IC 7420 (Four input NAND gate), IC7404, Patch cords

Theory:

Multiplexer is a device which takes more than one input and gives a simple output. Multiplexers are referred to as universal logic circuit this is because a $2^n : 1$ MUX can be used to design solution for any n variable truth table. Map entered variable is similar to K-map method.

In IC 74151, pin 16 is connected to VCC, pin 8 is connected to ground.

Control inputs A,B,C are connected to pin 9,10,11 respectively. Pin 4 and 5 are outputs. Pin 7 is STROBE and pin 1 to 4 and 12 to 15 are inputs.

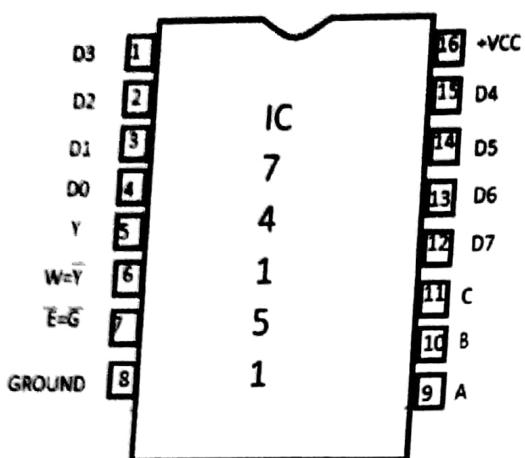
Procedure:

Map Entered Variable method can handle number of variables beyond four. MEV method makes it possible to use smaller maps to handle a larger number of variables. So we can use this method to reduce the 16:1MUX to 8:1MUX.

Construct the MEV truth table to identify the min terms and don't care terms. Then according to the list of obtained min terms and don't cares give the inputs to the 8:1 MUX. Construct the circuit according to the block diagram and check the outputs for the given number of inputs as in the truth table and verify the working of the MUX which gives the output as the given 16:1MUX

PIN DIAGRAM OF 8:1 MULTIPLEXER

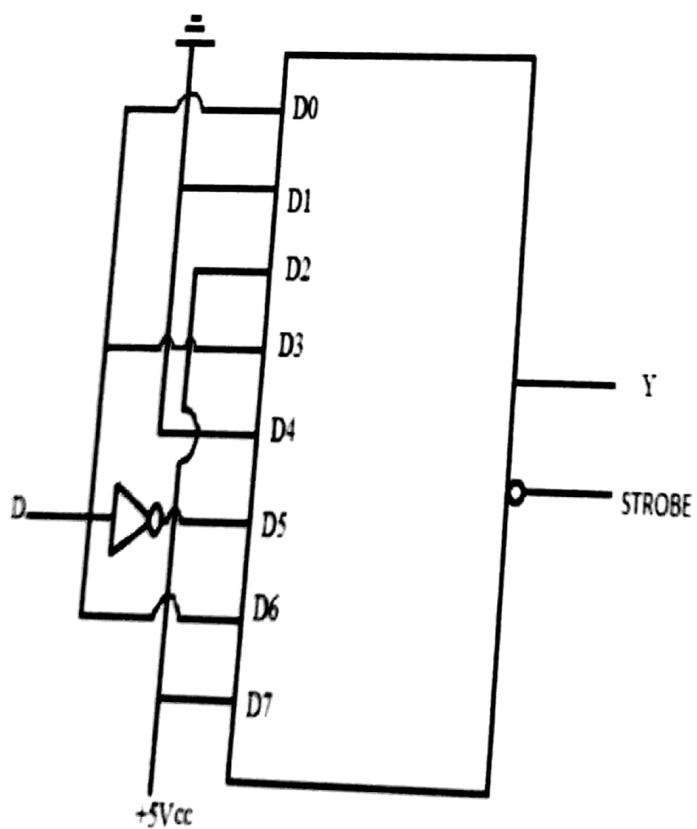
PIN DIAGRAM OF 8:1



Circuit diagram:

$$F(A,B,C,D) = \sum m(1,4,5,7,10,13,14,15)$$

Taking D as Entered variable we have the following



Truth table:

	A	B	C	D	Y	
D0	0	0	0	0	0	D
	0	0	0	1	1	
D1	0	0	1	0	0	0
	0	0	1	1	0	
D2	0	1	0	0	1	1
	0	1	0	1	1	
D3	0	1	1	0	0	D
	0	1	1	1	1	
D4	1	0	0	0	0	0
	1	0	0	1	0	
D5	1	0	1	0	1	D'
	1	0	1	1	0	
D6	1	1	0	0	0	D
	1	1	0	1	1	
D7	1	1	1	0	1	1
	1	1	1	1	1	

RESULT:

The given four variable logical expression was simplified using EVM method and realized

EXPERIMENTNO.9

Date14/9/2015

IMPLEMENTATION OF FULL ADDER AND FULL SUBTRACTOR USING 3:8 DECODER

AIM:To setup a full adder and full subtractor using 3:8 decoders.

APPARATUS REQUIRED:

1. IC trainer kit
2. IC 74138
3. IC 7420
4. Patch chords

THEORY:-

From the truth table of full adder we get $F(a,b,c)$ sum= $\Sigma m(1,2,4,7)$

$F(a,b,c)$ carry= $\Sigma m(3,5,6,7)$

From the truth table of full subtractor we get $F(a,b,c)$ difference= $\Sigma m(1,2,4,7)$

$F(a,b,c)$ carry= $\Sigma m(1,2,3,7)$

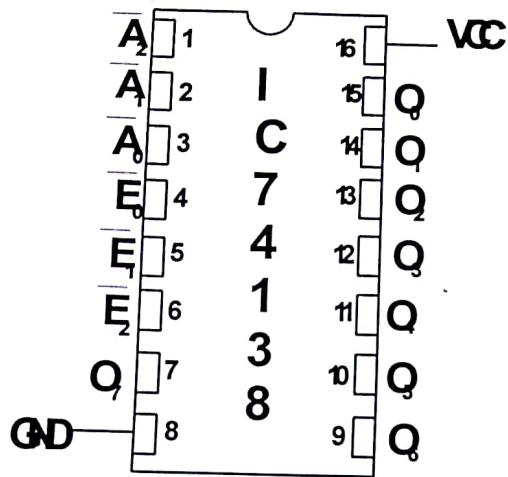
➤ Full adder

➤ Truth Table:

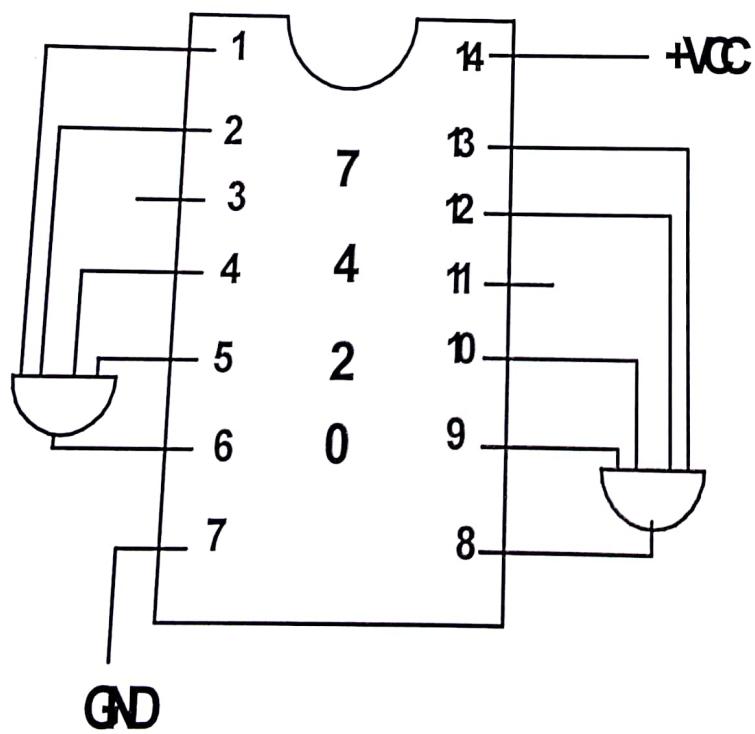
A	B	C	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

❖ Truth table for adder circuit

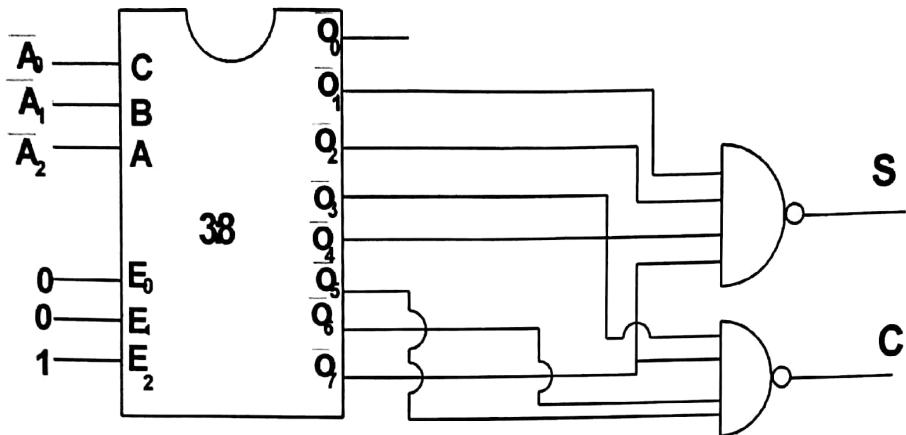
Pin Diagram of IC74138:



Pin Diagram of IC 7420:



Logic circuit 3:8 decoder:



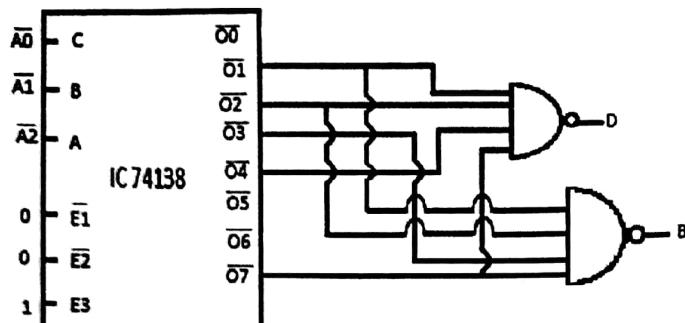
➤ Full subtractor

Truth Table:

A	B	C	DIFFERENCE	BORROW
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

❖ Truth table for subtractor circuit

Logic Diagram:



3:8 Decoder as full subtractor

PROCEDURE:

1. 3:8 decoder IC i.e., IC74138 is placed in one slot and IC7420 is placed in another slot provided in trainer kit.
2. The connections are made as shown in fig(a) using patch cords to design a full adder. The connections are made as shown in fig (b) using patch cords to design a full subtractor.
3. Active low (0) is sent across the pin 4 and pin 2 i.e., E_1, E_2 respectively and active high(1) is sent across the pin 6 i.e., E_3 to enable the circuit.
4. All possible combinations of inputs are given across the pins 3,2 and 1 i.e., A_1 and A_2 respectively and the corresponding sum and carry are noted and verified. Borrow are noted and verified from the truth table.

RESULT:

Full adder and Full subtractor Circuits are realized and verified using 3:8 decoders and truth table.

Date : 21/9/2015

EXPERIMENT NO.10

REALIZATION OF ONE AND TWO BIT COMPARATOR

AIM:- Verification of one bit-comparator using basic gates.

APPARATUS REQUIRED:-

1. IC trainer kit
2. IC 7408
3. IC 7404
4. IC 7402
5. IC 7485
6. Patch chords

THEORY:-

One-bit comparator compares magnitude of two 1-bit binary number. Say X and Y activates one of the three outputs $X=Y$, $X>Y$ and $X<Y$. The logic equations for the output can be written as follows where G, L and D stands for $X>Y$, $X<Y$, and $X=Y$ respectively.

$$(X>Y)=G=X\bar{Y};$$

$$(X<Y)=L=\bar{X}Y;$$

$$(X=Y)=E=\bar{X}\bar{Y}+XY=\bar{X}\oplus Y \quad (X=Y) = E = \overline{G + L}$$

Two-bit comparator compares magnitude of 2-bit binary number.

$$\text{The expression for } (A_1 A_0 = B_1 B_0) = (\overline{A_1 \oplus B_1}) \cdot (\overline{A_0 \oplus B_0})$$

$$\text{The expression for } (A_1 A_0 < B_1 B_0) = A_1' B_1 + A_0' B_0 \quad (\overline{A_1 \oplus B_1})$$

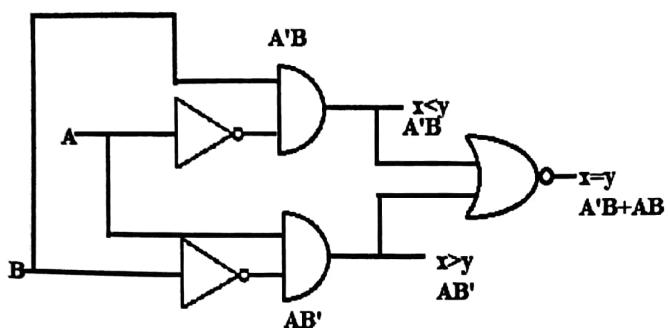
$$\text{The expression for } (A_1 A_0 > B_1 B_0) = A_1 B_1' + A_0 B_0' \quad (\overline{A_1 \oplus B_1})$$

ONE-BIT COMPARATOR

Truth Table:

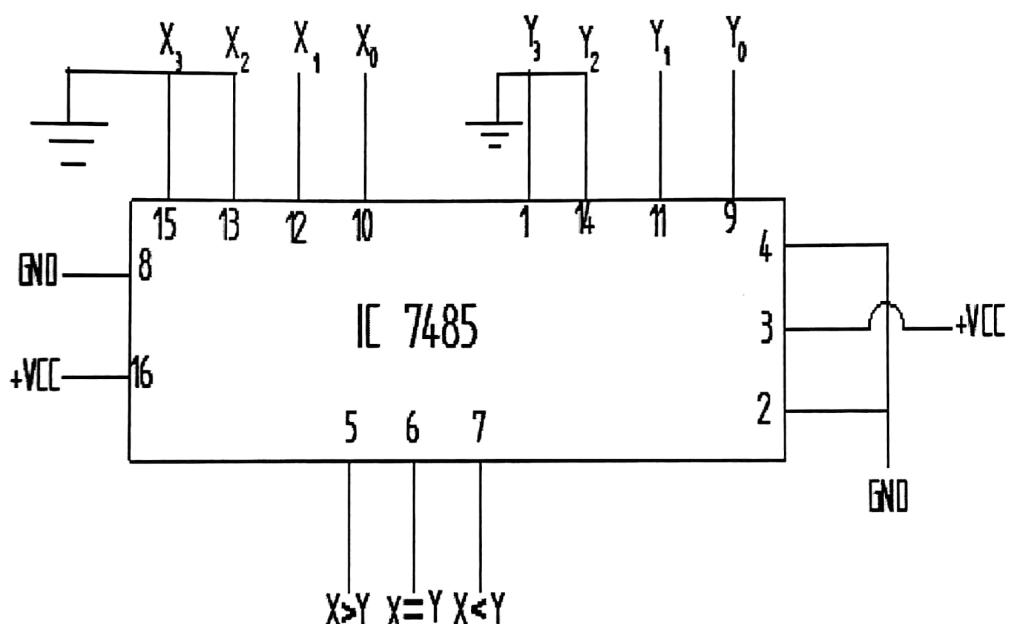
X	Y	$X>Y$ (G)	$X=Y$ (E)	$X<Y$ (L)
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Logic Diagram:



TWO-BIT COMPARATOR USING IC7485

Pin Diagram using 7485:



Truth Table:

A1	A0	B1	B0	A1A0 B1B0	=	A1A0 B1B0	<	A1A0 B1B0	>
0	0	0	0	1		0		0	
0	0	0	1	0		1		0	
0	0	1	0	0		1		0	
0	0	1	1	0		1		0	
0	1	0	0	0		0		1	
0	1	0	1	1		0		0	
0	1	1	0	0		1		0	
0	1	1	1	0		1		0	
1	0	0	0	0		0		1	
1	0	0	1	0		0		1	
1	0	1	0	1		0		0	
1	0	1	1	0		1		0	
1	1	0	0	0		0		1	
1	1	0	1	0		0		1	
1	1	1	0	0		0		1	
1	1	1	1	1		0		0	

PROCEDURE:-

1. Insert the IC in one of the pin sockets in the IC trainer and lock the socket. Identify the pin numbers.
2. Using patch chords connect +VCC and GND as per the pin diagram.
3. Rig up the circuit as per the logic diagram realized.
4. Connect the power supply to the IC trainer kit.
5. Using the HIGH_LOW input switches give the different input combinations.
6. For each input combination monitor the output and verify the truth table

RESULT:-

One-bit comparator is verified using basic gates and truth table.

Two-bit comparator is verified using IC7485 and truth table.

EXPERIMENT NO.11

Date : 21/9/2015

7-SEGMENT STATIC DISPLAY SYSTEM

AIM:-To set up and test a 7-segment static display system to display numbers 0-9.

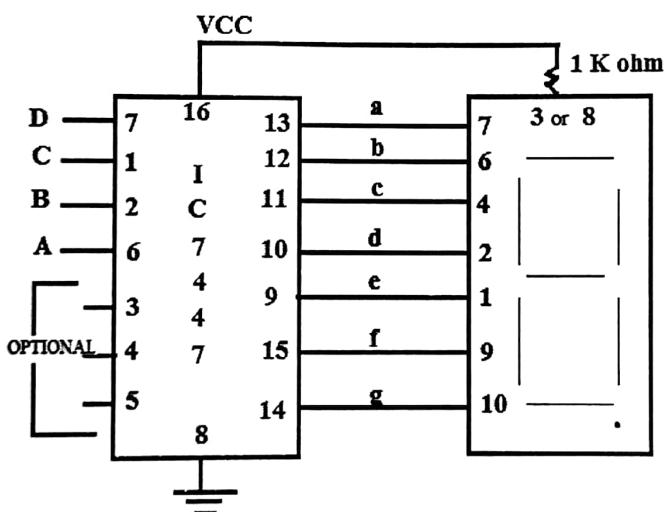
APPARATUS REQUIRED:-

1. IC trainer kit
2. IC 7447
3. 7-segment display
4. Patch chords

THEORY:-

The BCD code expresses each digit in a decimal number by its nibble equivalent. In 7-segment indicator, by forward biasing different LED's we can display the digits 0 to 9. For instance to display 0 we need to light up segments a, b, c, d, e, f. To light up 5, we need to light up segments a, c, d, f, g. In IC7447 pin number 8 is grounded, pin numbers 1, 2, 6, and 7 are connected as control inputs. Pin number 3 is LT, pin number 4 is BI/RBI and pin number 5 is RBD in 7 segment display, pin number 5 is open and pins 7, 6, 4, 2, 1, 9, 10 are connected to pins 13, 12, 11, 10, 9, 15, 14 of IC447 respectively.

Pin Diagram:



Truth Table:

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	✓	✓	✓	✓	✓	✓	
0	0	0	1		✓	✓				
0	0	1	0	✓	✓		✓	✓		✓
0	0	1	1	✓	✓	✓	✓			✓
0	1	0	0		✓	✓			✓	✓
0	1	0	1	✓		✓	✓		✓	✓
0	1	1	0			✓	✓	✓	✓	✓
0	1	1	1	✓	✓	✓				✓
1	0	0	0	✓	✓	✓	✓	✓	✓	✓
1	0	0	1	✓	✓	✓			✓	✓

PROCEDURE:-

- 1 Insert the IC in one of the pin sockets in the IC trainer and lock the socket and also connect the 7-segment indicator using the patch chords with the knob towards the common anode type. Identify the pin numbers.
- 2 Using patch chords connect +VCC and GND as per the pin diagram.
- 3 Rig up the circuit as per the logic diagram realized.
- 4 Connect the power supply to the IC trainer kit.
- 5 Using the HIGH_LOW input switches give the different input combinations.
- 6 For each input combination monitor the output and verify the truth table.

RESULT:-

BCD to 7-segment display is verified through truth table using IC 7447.

EXPERIMENT NO: 14

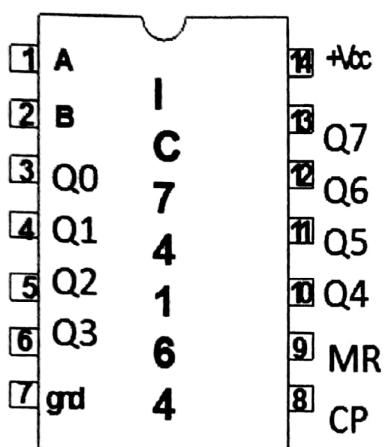
RING COUNTER AND JOHNSON COUNTER

Aim:

To verify ring counter and Johnson counter.

Components required: IC trainer kit, Patch chords, IC 74164

Pin Diagram:



Ring counter:

Theory:

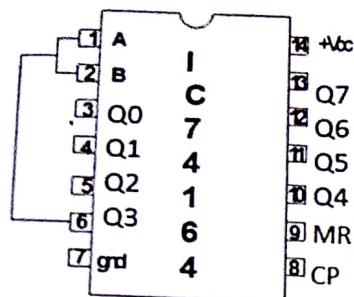
One of the application of the register were in the value of the last flip-flop is fed as input to the first flip-flop and for every clock cycle, the value is incremented by one bit position and when reaches the last flip-flop, again it start rotating from the first flip-flop forming a circular moment. Hence it is called as ring "counter".

To make it operate in this fashion, it is very much essential to present the first flip-flop with value "1" and then connect the output of last flip-flop as the input of the first flip-flop

Procedure:

1. IC 74164 is placed in the trainer kit.
2. Pin no. 7& 14 are given as ground and VCC respectively.
3. Pin no.8 will be clock pulse. A mono pulse should be given to obtain the output.
4. Pin no. 6,5,4,3 are connected to the output as Q₃,Q₂,Q₁ and Q₀ respectively.
5. To obtain ring counter short A & B and connect it to pin no.6 (Q₃).

DIAGRAM:



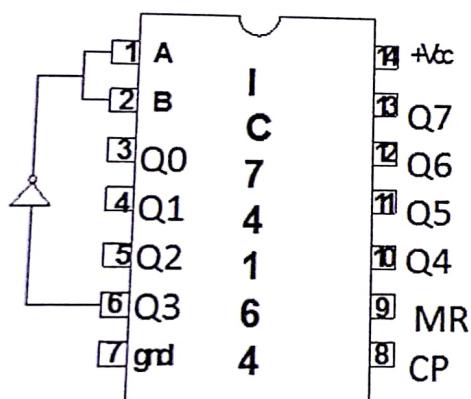
Johnson Counter:

Theory:

This is another kind of counter which is very much similar to ring counter with two differences.

1. No need to present the counter value
2. Negation of the output of last counter is fed as input to the first counter

Circuit:



Procedure:

1. IC 74164 is placed in the trainer kit.
2. Pin no. 7 & 14 are given as ground and VCC respectively.
3. Pin no. 8 will be clock pulse. A mono pulse should be given to obtain the output.
4. Pin no. 6, 5, 4, 3 are connected to the output as Q₃, Q₂, Q₁ and Q₀ respectively.
5. To obtain johnson counter, output of pin no. 1 and 2 are complemented by NOT gate and then short A & B and connect it to pin no.6 (Q₃).

Result: Desired output is obtained.

EXP NO: 13

DATE: 05.10.2015

RS, D and T FLIP FLOPS

Aim:

To Design and implement RS flip flops (Reset-Set Flip flop) and D flip flops (Data flip flops) using logic gates.

Apparatus Required:

1. IC 7400
2. IC 7402
3. IC 7404
4. IC 7408
5. Patch Cords

OBSERVATIONS:

RS FLIP FLOP:

Theory:

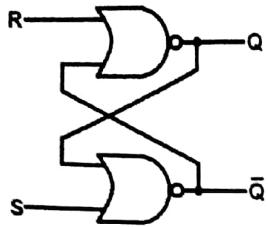
An S-R flip flop consists of two AND gates and two cross-coupled NOR gates. A clocked S-R flip-flop has an additional clock input so that the S and R inputs are active only when the clock is high. When the clock goes low, the state of flip-flop is latched and cannot change until the clock goes high again. Therefore, the clocked S-R flip-flop is also called "enabled" S-R flip-flop.

The truth tables of the circuits are shown below.

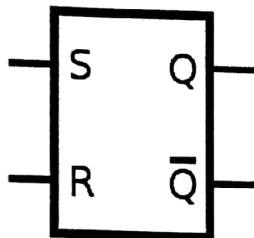
TRUTH TABLE:

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

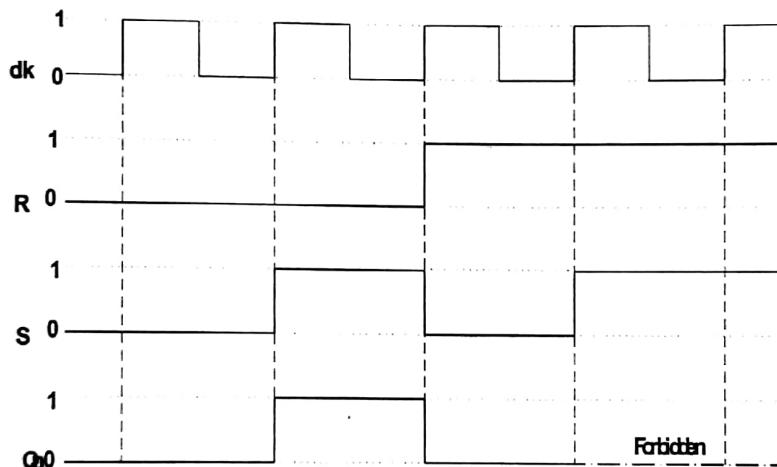
CIRCUIT DIAGRAM:



BLOCK DIAGRAM:



Timing table:



Procedure:

- Check all the components for their working.
- Insert the appropriate IC into the IC base and lock the socket.
- Make connections as shown in the circuit diagram.
- Using high-low input combinations give the inputs.
- Check the outputs for given input using truth table.

- Verify the results and observe the outputs.

D FLIP FLOP

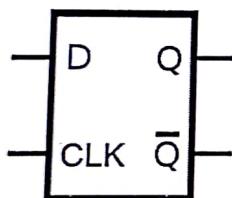
Theory:

A D flip flop combines the S and R inputs of an S-R flip flop into one input by adding an Inverter. When the clock is high, the output follows the D input, and when the clock Goes low, the state is latched.

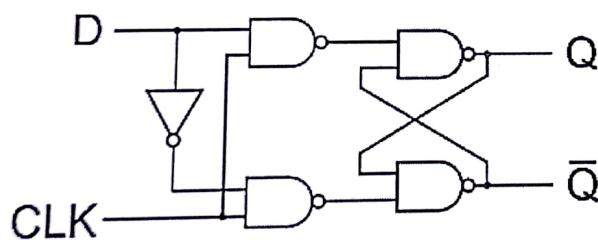
TRUTH TABLE:

INPUT				OUTPUT
PRE	CLR	CLK	D	Q
L	H	X	X	H
H	L	X	X	L
H	H	P.T	H	H
H	H	P.T	L	L

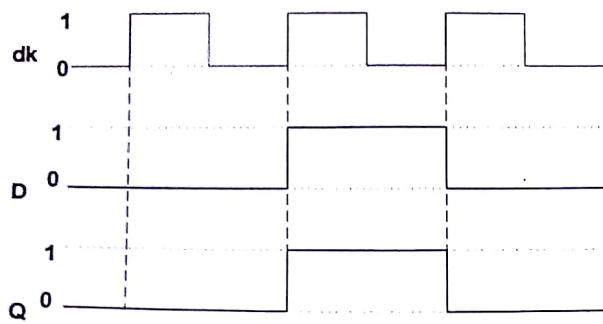
BLOCK DIAGRAM:



CIRCUIT DIAGRAM:

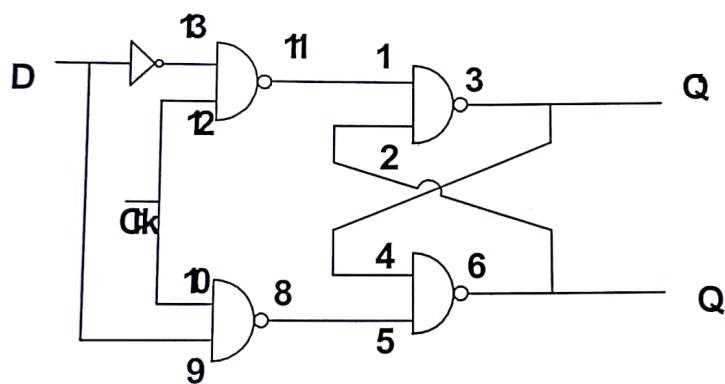


Timing Table:

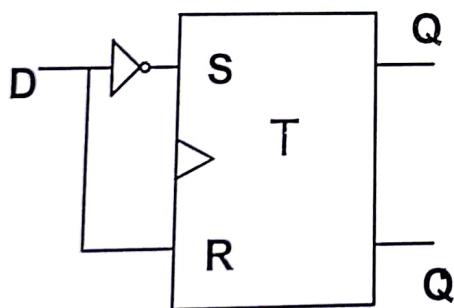


T-flip flop:

Diagram:



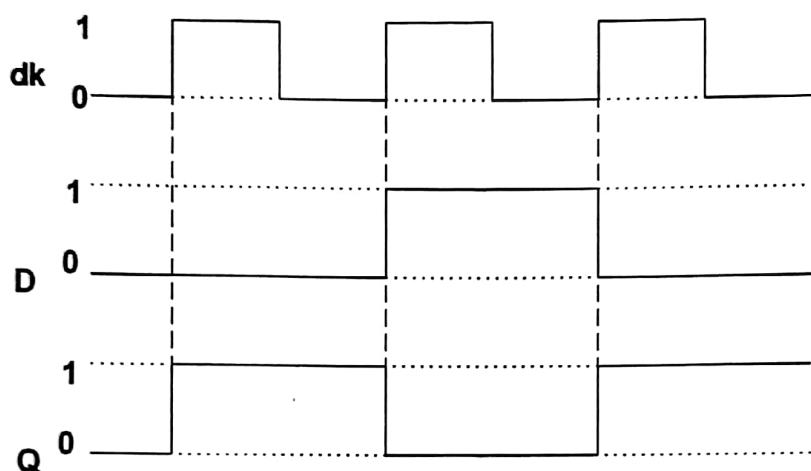
BLOCK DIAGRAM:



Truth Table:

CLK	T	Q	\bar{Q}
0	X	X	X
1	0	1	0
1	1	0	1

Timing Table:



Procedure:

1. Place the IC 7402 and IC 7408 in the trainer kit such that semicircle of the IC is at the top.
2. Pin 7 and pin 14 of both the IC's are connected to ground and VCC respectively.
3. Connections are made as shown in the figure.
4. The input clock is connected to a clock of 100 Hz.
5. Output changes as the input changes with respect to clock.

Result:

Truth table of T flip flop is verified.

EXPERIMENT NO.-14

DATE-5/10/15

DESIGN AND IMPLEMENTATION OF JK-FLIP FLOP AND JK MASTER SLAVE-FLIP FLOP

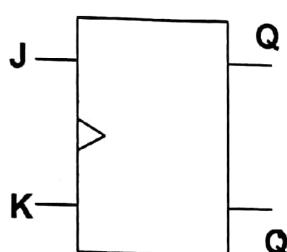
Aim:

To verify the truth table of JK-flip flop and JK Master slave-flip flop.

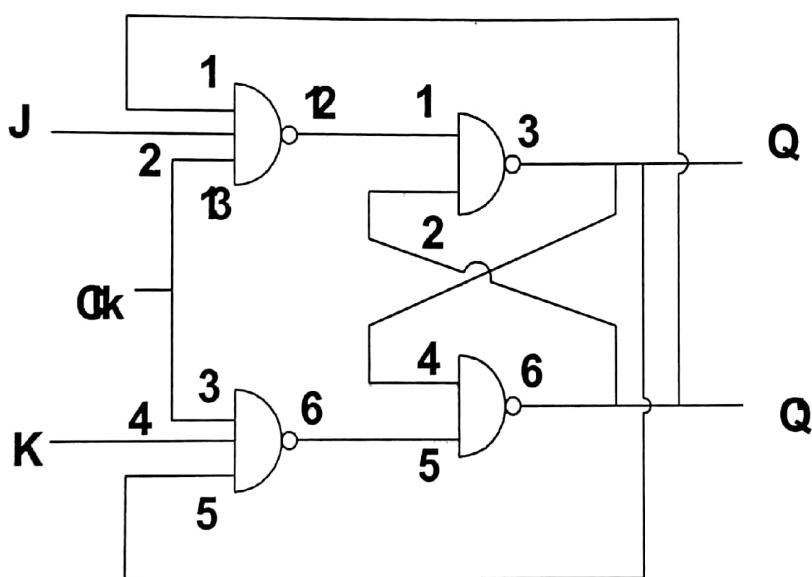
JK-flip flop:

Components required: Trainers kit, Patchchords, IC 7400, IC 7402, IC 7404, IC 7408

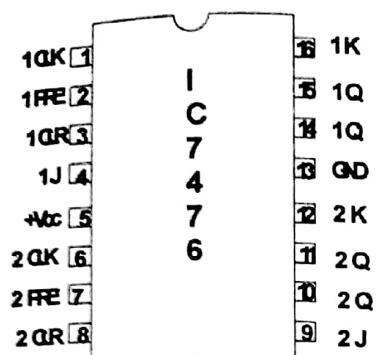
Diagram:



CIRCUIT DIAGRAM:



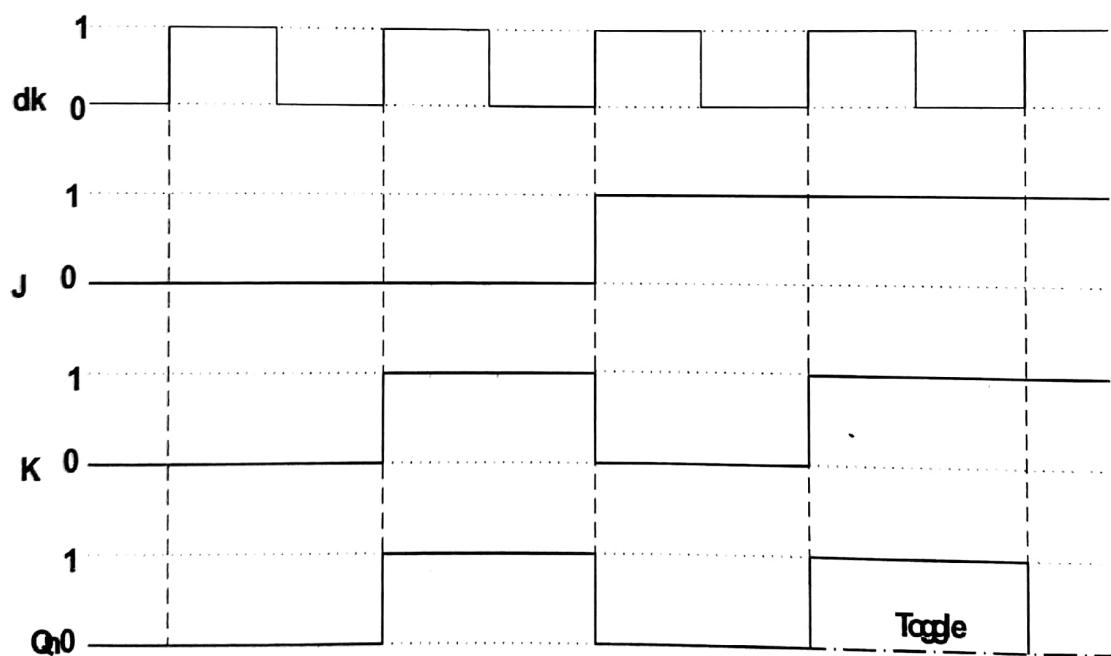
PIN DIAGRAM:



Truth Table:

J	K	Q
0	0	Q
0	1	0
1	0	1
1	1	Toggle

Timing table:



Procedure:

1. Place the IC's IC 7408 and IC 7402 in the trainer kit such that the semicircle of IC is at the top.
2. Pin 7 of IC and pin 14 of IC is connected to ground and VCC respectively.
3. Connections are made as shown in the circuit diagram.
4. The inputs are maintained at different logic states and the corresponding output is sorted and truth table is verified.

Result:

The truth table of clocked JK flip flop is verified and truth table of JK flip flop is verified.

JK master slave-flip flop:

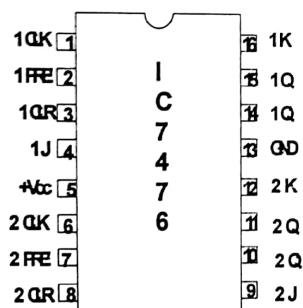
Components required:

1. Trainer kit
2. Patch chords
3. IC 7476

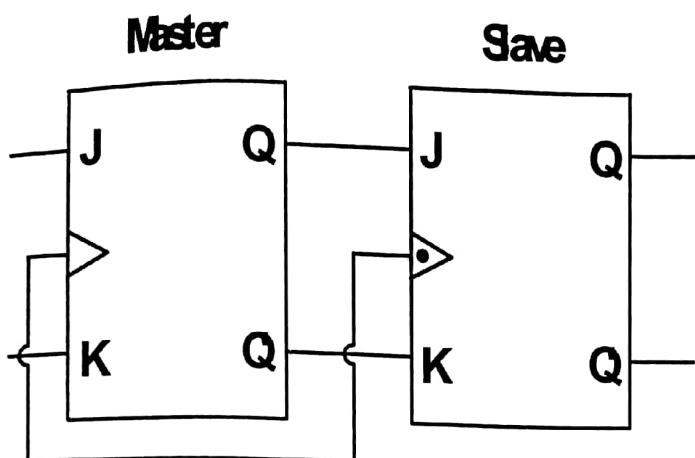
Theory:

1. Master is positive edge triggered and slave is negative edge triggered.
2. If $J=1$ and $K=0$, i.e., master sets during positive transition. The high output Q of the master drives the J input of the slave during NT and it sets slave to set.
3. If $J=0$ and $K=1$, i.e., master resets during PT of the clock. The high output \bar{Q} drives the K input of the slave during NT and it forces slave to reset.
4. If $J=1$ and $K=1$, master toggles during PT and slave toggles during NT.
5. If $J=0$ and $K=0$, flip flop is disabled and Q remains unchanged,

Pin diagram:



BLOCK DIAGRAM:



Procedure:

1. Place the IC 7476 in the trainer kit.
2. Pin no.5 is connected to VCC and pin no. 13 is connected to ground.
3. Connections are made as shown in the pin diagram.
4. The inputs are maintained at different logic states and the corresponding output is sorted and truth table is verified.

Result:

The truth table of clocked JK master slave-flip flop is verified and truth table of JK master slave-flip flop is verified.

EXPERIMENT NO.15

Date : 12/10/2015

HDL IMPLEMENTATION OF BASIC GATES AND UNIVERSAL GATES USING VERLOG

AIM: To implement the basic gates(AND,NOT&NOT)and universal gates(NAND&NOR) using Verilog code.

PROCEDURE:

- Create a new project (for each gate).
- Process ->implement the top module.
- Create a test fixture (Behavioral stimulation).
- Give different possible combination to the input.
- Stimulate behavioral design.

CODES:

➤ AND GATE:

```
Moduleandgate(A,B,Y);
input A,B;
output Y;
and g1(Y,A,B);
endmodule
```

➤ OR GATE:

```
Moduleorgate(A,B,Y);
input A,B;
output Y;
or g1(Y,A,B);
endmodule
```

➤ NOT GATE:

```
Modulenotgate(A,Y);
input A;
output Y;
not g1(Y,A);
endmodule
```

➤ NOR GATE:

```
Modulenorgate(A,B,Y);
input A,B;
output Y;
nor g1(Y,A,B);
endmodule
```

➤ NAND GATE:

```
Modulenandgate(A,B,Y);
input A,B;
output Y;
nand g1(Y,A,B);
endmodule
```

RESULT: Hence the basic gate and universal gate are implemented using VHDL

IMPLEMENTATION OF 8:1 MULTIPLEXER USING VERILOG

AIM: To implement 8:1 multiplexer using Verilog code.

CODE:

```
module 8to1mux(A,B,C,D0,D1,D2,D3,D4,D5,D6,D7,Y)
    input A,B,C,D0,D1,D2,D3,D4,D5,D6,D7;
    output Y;
    reg Y;

    always @((A or B or C) or D0 or D1 or D2 or D3 or D4 or D5 or D6 or D7)
        case({A,B,C})
            0:Y=D0;
            1:Y=D1;
            2:Y=D2;
            3:Y=D3;
            4:Y=D4;
            5:Y=D5;
            6:Y=D6;
            7:Y=D7;

        Endcase
    endmodule
```

RESULT: Hence 8:1 multiplexer implemented using VHDL.

HDL IMPLEMENTATION OF FULL ADDER USING VERILOG

AIM: To implement full adder using Verilog code.

CODE:

```
Modulefulladder(A,B,C,SUM,CARRY);
```

```

input A,B,C;
output SUM,CARRY;
assign SUM=A^B^C;
assign CARRY=(A&B)|(B&C)|(C&A);
end module

```

RESULT: Hence the full adder is implemented using VHDL.

HDL OF MULTIPLEXER AND DEMULTIPLEXER

AIM: To implement multiplexer and de-multiplexer using Verilog code.

CODES:

➤ **MUX:**

```

module 4to1mux(S,A,B,D0,D1,D2,D3,Y);
input S,A,B,D0,D1,D2,D3;
output Y;
reg Y;
always @((A or B or D0 or D1 or D2 or D3))
case{(A,B)}
  0:Y=D0;
  1:Y=D1;
  2:Y=D2;
  3:Y=D3;
Endcase
end module

```

DEMUX:

```

module 1to4demux(S,B,D0,D1,D2,D3,Y)
input [1:0]S,D0,D1,D2,D3;
output [3:0]Y;
reg Y;
always @((s[1,0] or D))
case (s[1,0],D)
  0 0:Y
  0 1:Y
  1 0:Y
  1 1:Y
Endcase
end module

```

RESULT: Hence the multiplexer and de-multiplexer is implemented using VHDL.

EXPERIMENT NO.-16

MOD-N(N<8) SYNCHRONOUS COUNTER

Aim:

Design and implementation of a mod-6 synchronous up counter using J-K flip flop ICs.

Components required:

- IC 7476
- IC 7408
- Patch cords.

Theory:

Present			Next			J _C	K _C	J _B	K _B	J _A	K _A
C _n	B _n	A _n	C _{n+1}	B _{n+1}	A _{n+1}						
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	0	0	0	x	1	0	x	x	1

K-Map simplification:

A _n \B _n C _n	00	01	11	10
0	0	0	1	0
1	X	x	x	x

$$J_A = B_n C_n$$

A _n \B _n C _n	00	01	11	10
0	X	x	x	X
1	0	1	x	X

$$K_A = C_n$$

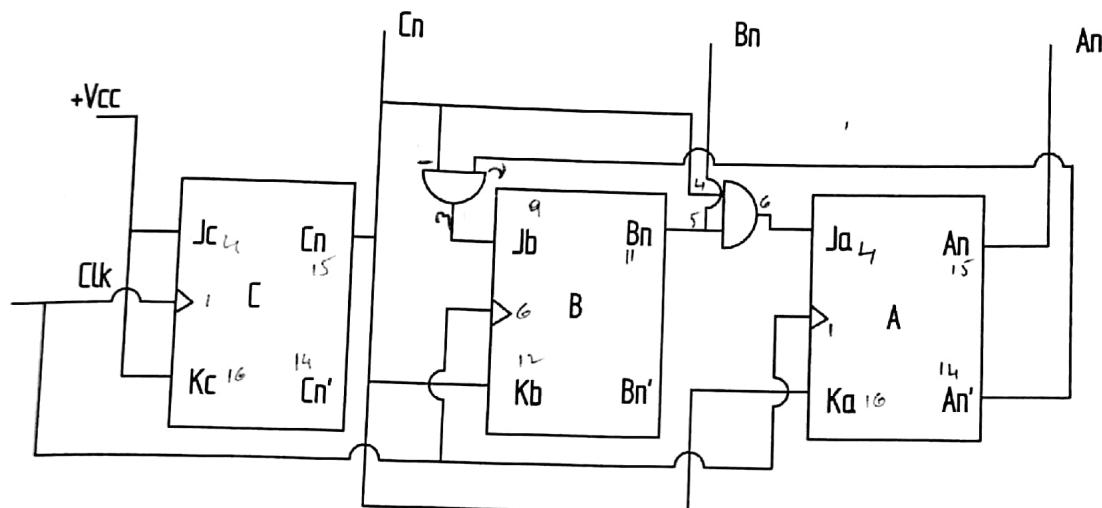
A _n \B _n C _n	00	01	11	10
0	0	1	x	X
1	0	0	x	X

$$J_B = \overline{An} \cdot Cn$$

$A_n \setminus B_n \setminus C_n$	00	01	11	10
0	x	x	1	0
1	x	x	x	x

K_B-C_n

Circuit Diagram:



Procedure:

- 1) Write the present state , next state and the corresponding excitation values.
 - 2)Plot the K-Map from the values of the excitation table. Write the equations for J_A , K_A , J_B , K_B , J_C , K_C and draw the circuit diagram.
 - 3)IC is latched to the trainer kit and connections are done as shown in the figure.
 - 4)Turn on the power supply and verify the mod 6 counter.

Result:

Mod-6 counter is thus verified.