

Ex: No: 1

Date: 5.10.2021

## VERIFICATION OF LOGIC GATES

Aim:

TO construct IC logic gates OR, AND, NOT, NOR, NAND gates and to verify the corresponding truth table.

Components Required :

IC 7432, 7408, 7404, 7400, 7402, chip, power supply, voltmeter.

Procedure :

## 1. OR Gate :

The IC chip 7432 is fixed on the bread board. The negative of the power supply is connected to pin '7' and the positive of the power supply is connected ~~to~~ pin '14' and is kept at 5 volts. we can constructed OR Gates with one IC chip for the first OR Gates 1, 2 are inputs and 3 will be output.

The input level are kept at (0,0) (0,1) (1,0) (1,1) levels and corresponding output voltage is measured on the truth table is verified. we can verify the output voltage with other OR Gates also.

## 2. AND Gate:

The IC chip 7408 is fixed on the bread board. The pair pin "7" is connected to negative and the "14" is connected to positive of the power supply which is kept at 5 volts. we can construct 4 AND gates with one IC chip. The input level are kept at (0,0) (0,1) (1,0) (1,1). The corresponding outputs are measured and they are found to be A B and the truth tables are verified.

## 3. NOT Gate:

The IC chip 7404 is fixed on the bread board. The pin "7" is connected to negative and the pin "14" is connected to the power supply which is kept at 5V. Here we can construct 6 NOT gates. When the input is "0" we get output and vice versa. The output is found to be  $\bar{A}$ .

## 4. NOR Gate:

The IC chip 7402 is fixed on the bread board. The pin "7" is connected to negative and the "14" is connected to positive of the power supply which is kept at 5V. Here we can 4 NOR gates. The input is kept at (0,0) (0,1) (1,0) (1,1). The output is measured and it is found to follow the relation  $\overline{A+B} = \bar{A} \cdot \bar{B}$ . The truth table is also verified (i.e) output when both the inputs are at "0" level.



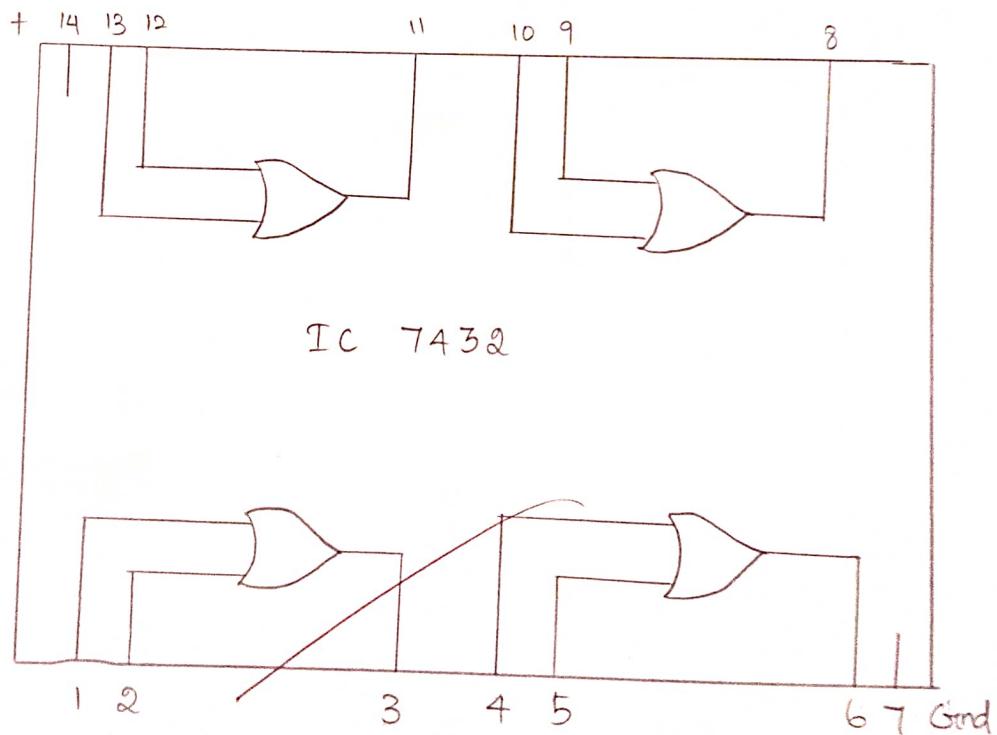
## 5. NAND Gate:

The IC chip 7400 is fixed on the bread board. The pin "7" is connected to negative and the pin "14" is connected to positive of the power supply which is kept at 5 volts. We can get 4 NAND gates. The input is kept at (0,0) (0,1) (1,0) (1,1) levels. The output is measured and it is found to follow the relation  $A \cdot B = \bar{A} + \bar{B}$ . The truth table is also verified (i.e) output is found to be "1" if any of the input is at "0" level.

voltage Table:

| Input |      | Output          |
|-------|------|-----------------|
| A     | B    | $y = A \cdot B$ |
| 0     | 0    | 0               |
| 0     | 5.08 | 0               |
| 5.08  | 0    | 0               |
| 5.08  | 5.08 | 4.70            |

OR Gate:



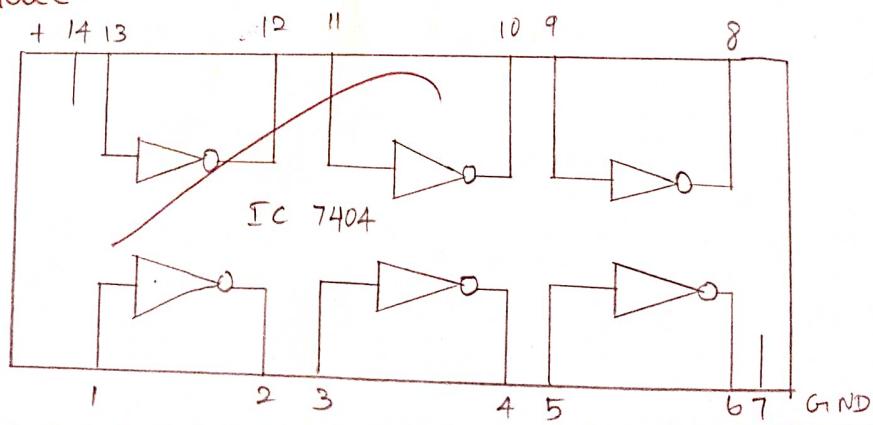
## Truth Table

| Input |   | Output      |
|-------|---|-------------|
| A     | B | $y = A + B$ |
| 0     | 0 | 0           |
| 0     | 1 | 1           |
| 1     | 0 | 1           |
| 1     | 1 | 1           |

## Voltage Table:

| Input |      | Output      |
|-------|------|-------------|
| A     | B    | $y = A + B$ |
| 0     | 0    | 0.11        |
| 0     | 0.08 | 3.43        |
| 0.08  | 0    | 3.43        |
| 5.08  | 5.08 | 3.43        |

## NOT Gate:

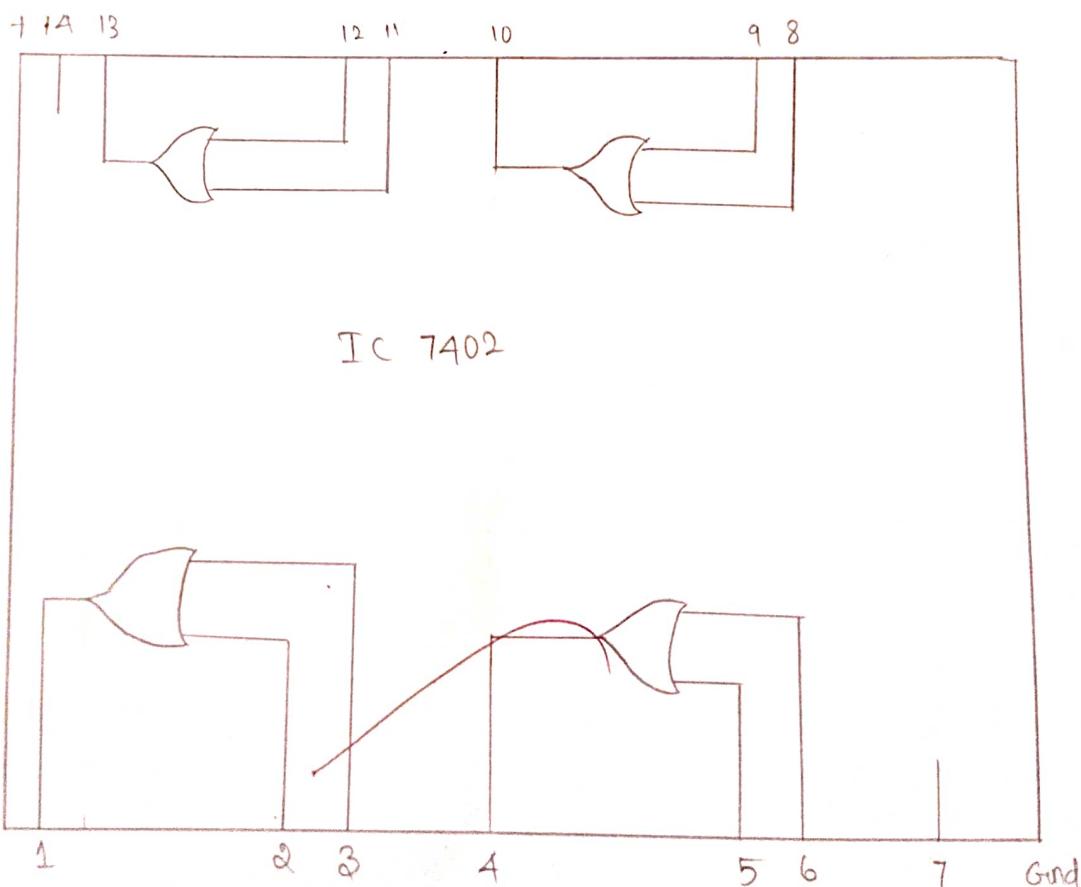


## Truth Table

| Input | Output        |
|-------|---------------|
| A     | $Y = \bar{A}$ |
| 0     | 1             |
| 1     | 0             |

| Input | Output        |
|-------|---------------|
| A     | $Y = \bar{A}$ |
| 0     | 3.43          |
| 5.08  | 0.11          |

## NOR Gate:



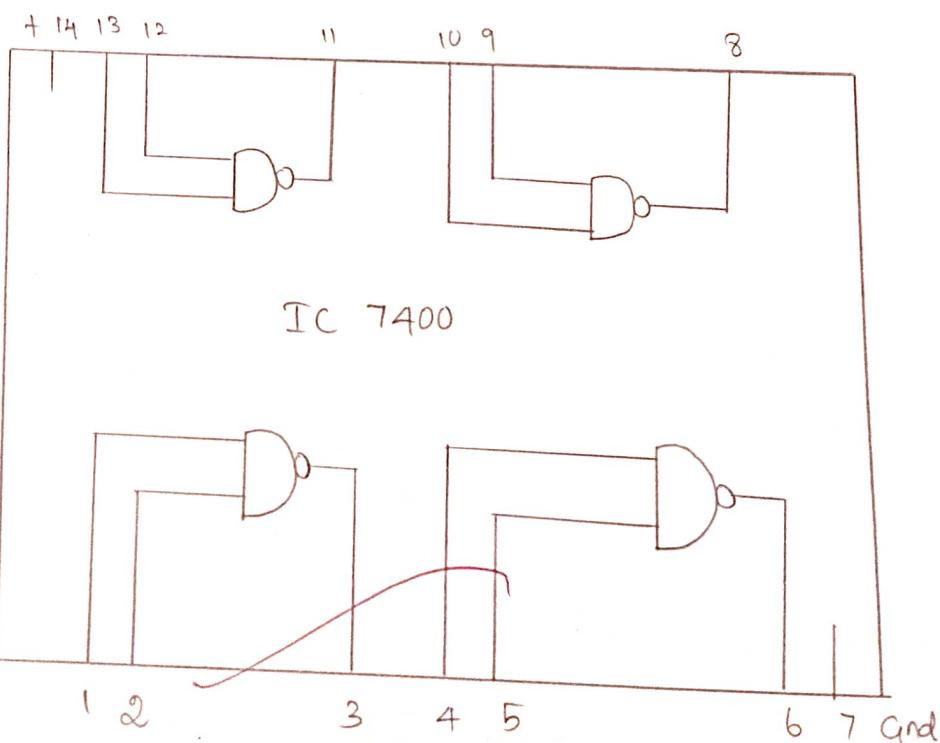
Truth Table

| Input |   | Output               |
|-------|---|----------------------|
| A     | B | $Y = \overline{A+B}$ |
| 0     | 0 | 1                    |
| 0     | 1 | 0                    |
| 1     | 0 | 0                    |
| 1     | 1 | 0                    |

Voltage Table

| Input |      | Output               |
|-------|------|----------------------|
| A     | B    | $Y = \overline{A+B}$ |
| 0     | 0    | 4.5V                 |
| 0     | 5.08 | 0                    |
| 5.08  | 5.08 | 0                    |
| 5.08  | 5.08 | 0                    |

NAND Gate:



Truth Table

| Input |   | output                     |
|-------|---|----------------------------|
| A     | B | $Y = \overline{A} \cdot B$ |
| 0     | 0 | 1                          |
| 0     | 1 | 1                          |
| 1     | 0 | 1                          |
| 1     | 1 | 0                          |

Voltage Table

| Input |      | output                     |
|-------|------|----------------------------|
| A     | B    | $Y = \overline{A} \cdot B$ |
| 0     | 0    | 3.41                       |
| 0     | 5.08 | 3.41                       |
| 5.08  | 0    | 3.41                       |
| 5.08  | 5.08 | 0.4                        |



### Result:

The basic logic gates are constructed using IC chip and the corresponding truth tables are also verified.

Ex: No: 8

Date: 21.10.21 construction of HALF And FULL ADDER

Aim:

To construct half adder and Full adder using logic gates and to verify their truth table.

Apparatus Required:

IC - 7400, 7408, 7432, 7486.

Bread board, 5 volt power supply.

Procedure:

Before using all the gates are checked using their tables.

1. Half Adder:

It is a combinational circuit which performs the addition of 2 bits. This circuit accepts two binary inputs and gives 2 binary outputs.

(i) The Half adder circuit is constructed as shown.

- (ii) It has two inputs A and B representing the bits to be added and two output sum and carry.
- (iii) To verify table is verified two binary inputs.

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$C = AB$$

The half adder circuit can also be constructed using IC - 7408 and IC - 7486.

## 2. Full Adder:

It is a combinational circuit that performs the arithmetic sum for three (3) input bits the inputs are: Augend addend and a carry from the previous lower significant position the outputs are: sum and carry. Full adder can be constructed using two half-adder circuits.

- (i) The full adder circuit is constructed using gates.

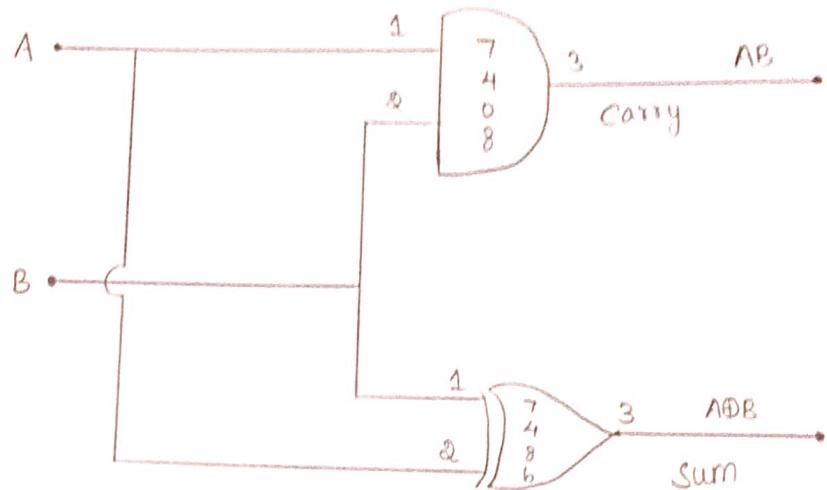
(ii) Two multi digit may be added serially and required a complete adder consisting of two half adder in cascade.

$$\begin{aligned}
 S &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC \\
 &= C(\bar{A}\bar{B} + AB) + \bar{C}(\bar{A}B + A\bar{B}) \\
 &= C(\overline{A \oplus B}) + \bar{C}(A \oplus B) \\
 &= C \oplus (A \oplus B)
 \end{aligned}$$

$$\begin{aligned}
 C &= \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\
 &= C(\bar{A}B + A\bar{B}) + AB(\bar{C} + C) \\
 &= C(A \oplus B) + AB.
 \end{aligned}$$



## HALF ADDER:



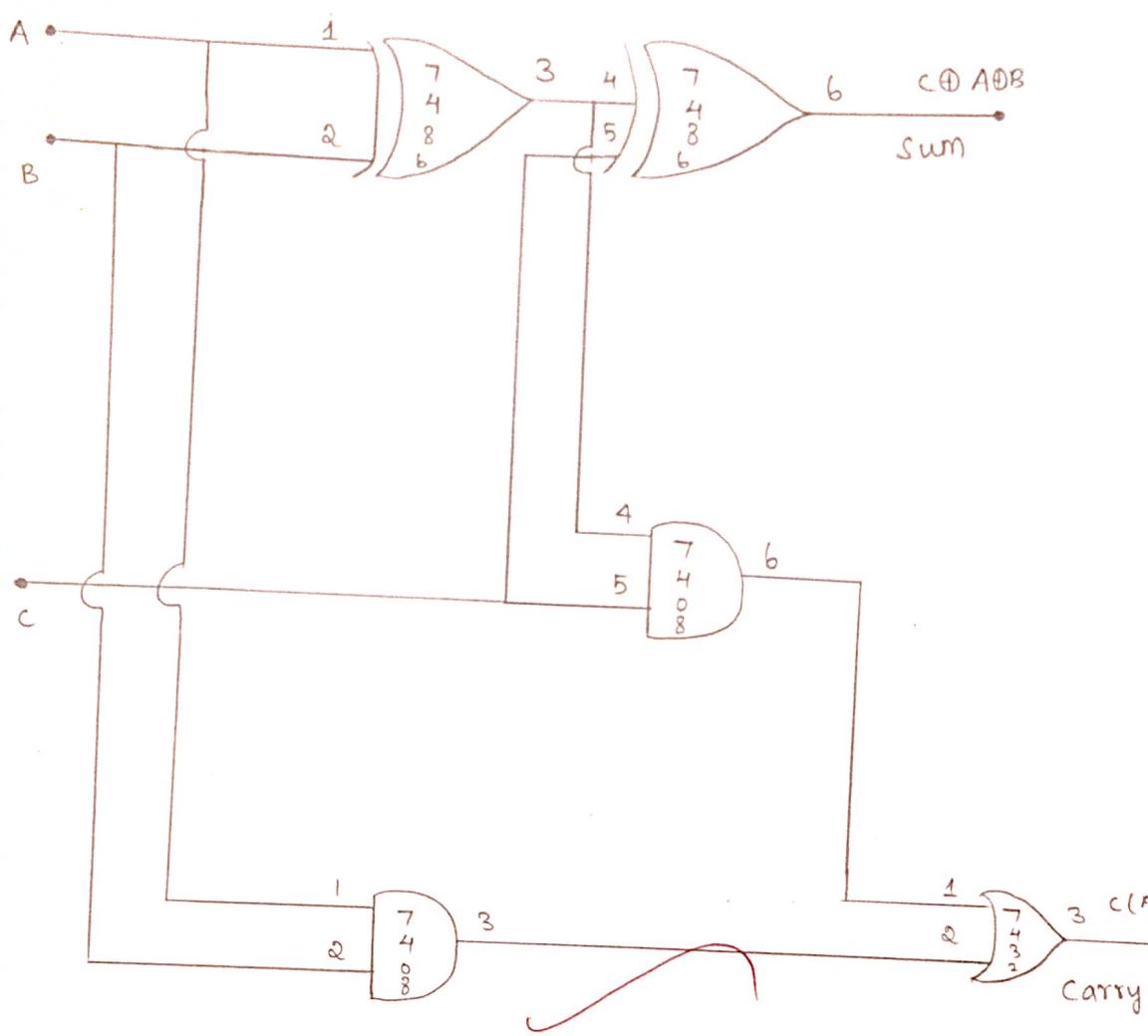
Truth Table:

| Input |   | Output |       |
|-------|---|--------|-------|
| A     | B | sum    | carry |
| 0     | 0 | 0      | 0     |
| 0     | 1 | 1      | 0     |
| 1     | 0 | 1      | 0     |
| 1     | 1 | 0      | 1     |

Voltage Table

| Input |      | Output |       |
|-------|------|--------|-------|
| A     | B    | sum    | carry |
| 0     | 0    | 0.10   | 0.11  |
| 0     | 4.73 | 3.29   | 0.11  |
| 4.81  | 0    | 3.29   | 0.11  |
| 4.86  | 4.89 | 0.13   | 3.42  |

## FULL ADDER :



Truth Table

| Input |   |   | Output |       |
|-------|---|---|--------|-------|
| 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     |

Voltage Table:

| Input |      |      | output |       |
|-------|------|------|--------|-------|
| A     | B    | C    | sum    | carry |
| 0     | 0    | 0    | 0.14   | 0.12  |
| 0     | 0    | 5.00 | 3.32   | 0.16  |
| 0     | 5.00 | 0    | 3.32   | 0.16  |
| 0     | 5.00 | 5.00 | 0.33   | 3.31  |
| 5.00  | 0    | 0    | 0.37   | 0.31  |
| 4.98  | 0    | 4.98 | 0.16   | 3.31  |
| 4.98  | 4.81 | 0    | 0      | 3.31  |
| 4.97  | 4.90 | 4.97 | 3.37   | 3.32  |



Result:

~~We~~ The half adder and full adder circuits are constructed using gates and their truth tables are verified.

Ex: No: 3

Date: 21.10.2021

## KARNAUGH MAP

## Aim:

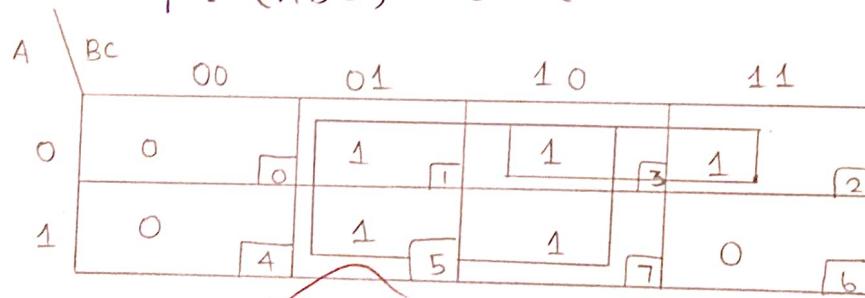
To simplify and optimize the logical function with Karnaugh map and to implement and verify that the operation of logical function using basic logic gates.

## Components Required:

Basic gates ( ), Power supply, module and multimeter.

## Problem:

$$F = (ABC) = \sum m(1, 2, 3, 5, 7)$$



## Procedure:

$$F = C + \bar{A}B$$

## Step 1:

The first step in designing a digital system is to have a clear idea of the variables involved in the process.

Further depending on the problem statement we have to arrive at the number of the output variable and their values for each and every combination of the input literals which can be conveniently represented in the form of a truth table.

Step 2:

In the given problem

No. of input variables : 3

which we will call A, B and C

No. of output variables : 1

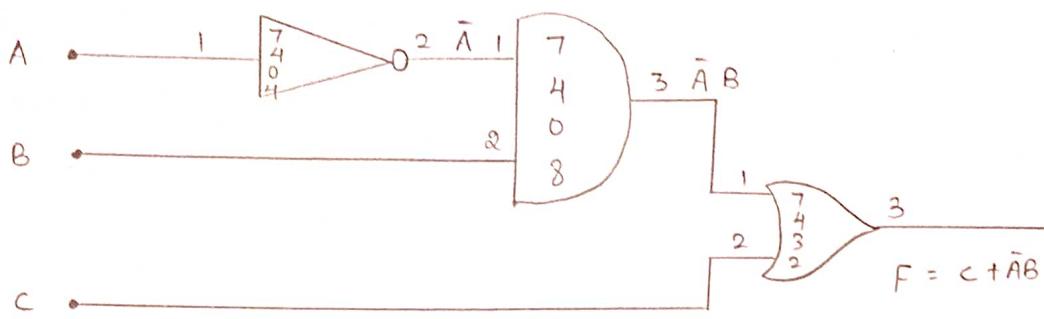
which we will call F.

Step 3:

~~Design process :~~

Having obtained the simplified logical expression, we can decide on the type and the number of gates required to realize the expected logic for every output bit for the given example can be designed using basic gates like AND, OR and NOT.

## Circuit Diagram:



## Truth Table:

| A | B | C | $\bar{A}$ | $\bar{A}B$ | $C + \bar{A}B$ |
|---|---|---|-----------|------------|----------------|
| 0 | 0 | 0 | 1         | 0          | 0              |
| 0 | 0 | 1 | 1         | 0          | 1              |
| 0 | 1 | 0 | 1         | 1          | 1              |
| 0 | 1 | 1 | 1         | 1          | 1              |
| 1 | 0 | 0 | 0         | 0          | 0              |
| 1 | 0 | 1 | 0         | 0          | 1              |
| 1 | 1 | 0 | 0         | 0          | 0              |
| 1 | 1 | 1 | 0         | 0          | 1              |

Result: The given expression is reduced by using K-map and the logical function is verified using basic gates.

Ex: No: 4

Date: 22.10.21

## SHIFT REGISTER

## Aim:

TO construct a four serial in serial out (SISO), serial In parallel out (SIP0), parallel In parallel out (PIPO) and parallel In serial out (PISO) shift register using D-Flip Flop IC and study its working.

## Apparatus Required:

D-Flip flop IC 7474, IC 7400, plus generator, Indicator Board.

## PRINCIPLE:

Register is a memory element capable of holding, one byte word in a eight bit up shift register is a special type of register which has the facility to shift its contents. Generally shift registers are constructed using flip-flop. So, a shift register is defined as a series of flip flop capable of shifting a binary number to either in the left or right.

According to the input and output there are four types of operations that are performed by a shift register. They are SISO, SIPO, PISO, PIPO.

### PROCEDURE :

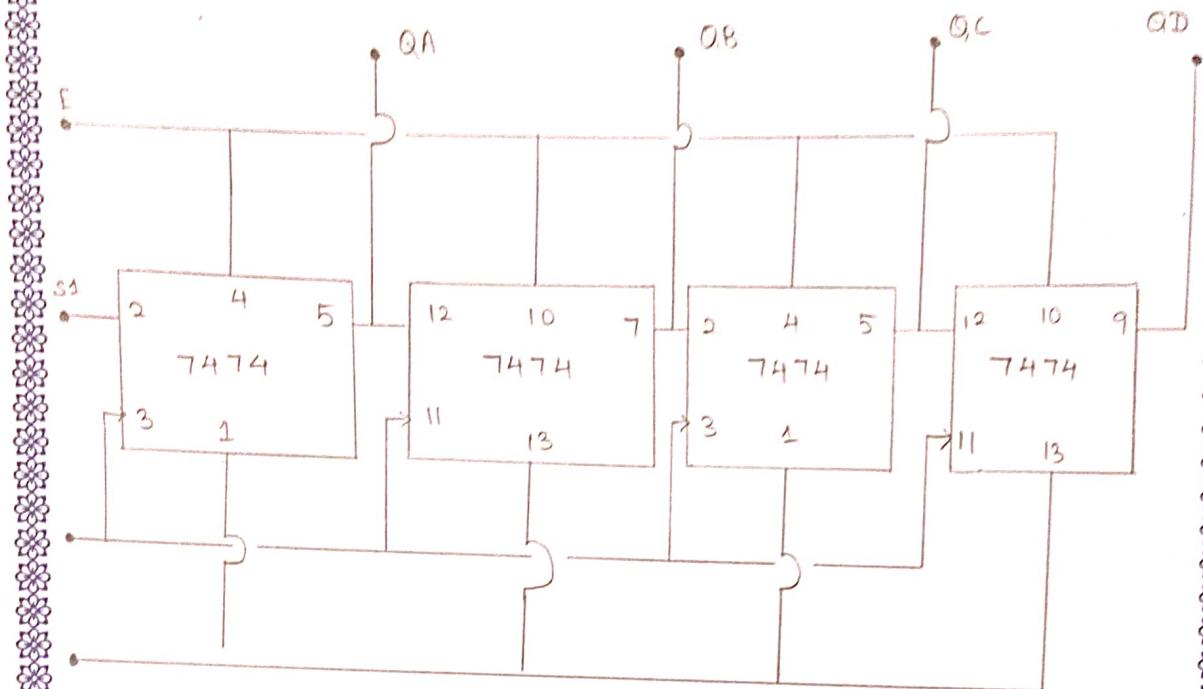
#### SISO and SIPO Shift Register [RIGHT] :

The connections are made as shown in figure where pin diagram given in figure. The serial input is to the given at input of the flip-flop. A one after another the flip-flop are cleared by making CLR as "0" and then it made as "1". The serial input is given and then clock pulse is applied. The output is available in parallel form after all the four serial input are given (i.e) after applying every clock pulse. The output are noted after Q0 from the 4th clock pulse onwards. Keep the serial input at "0" apply these none clock pulse and note the serial output at Q0.

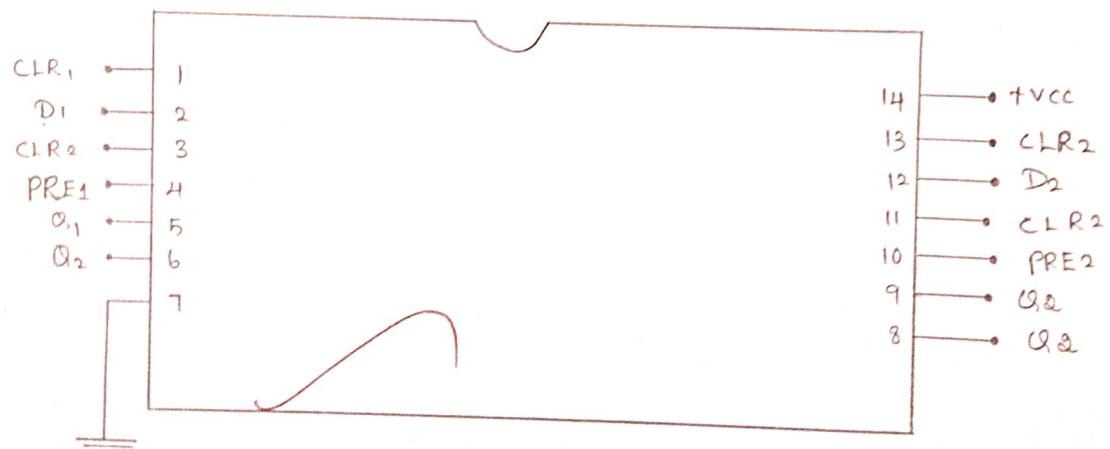
## SISO and SIPD shift Register [LEFT]:

The connection are made in shown in figure. The serial input is to be given at the SI of the first flip-flop one after another. Now the third flip-flop is D. First the flip-flop are cleared by making CLR as "0" and then it is made as 1. The serial input is given and then the clock pulse is applied. The output is available in parallel form after all the four input are given after four pulses. The outputs are noted after applying every clock pulse. A serial input is obtained at QD from the 4th clock pulse onwards. Keep the serial input at "0" and apply three more clock pulse and note the serial output (SO) at QA.

## SHIFT REGISTER [RIGHT] :



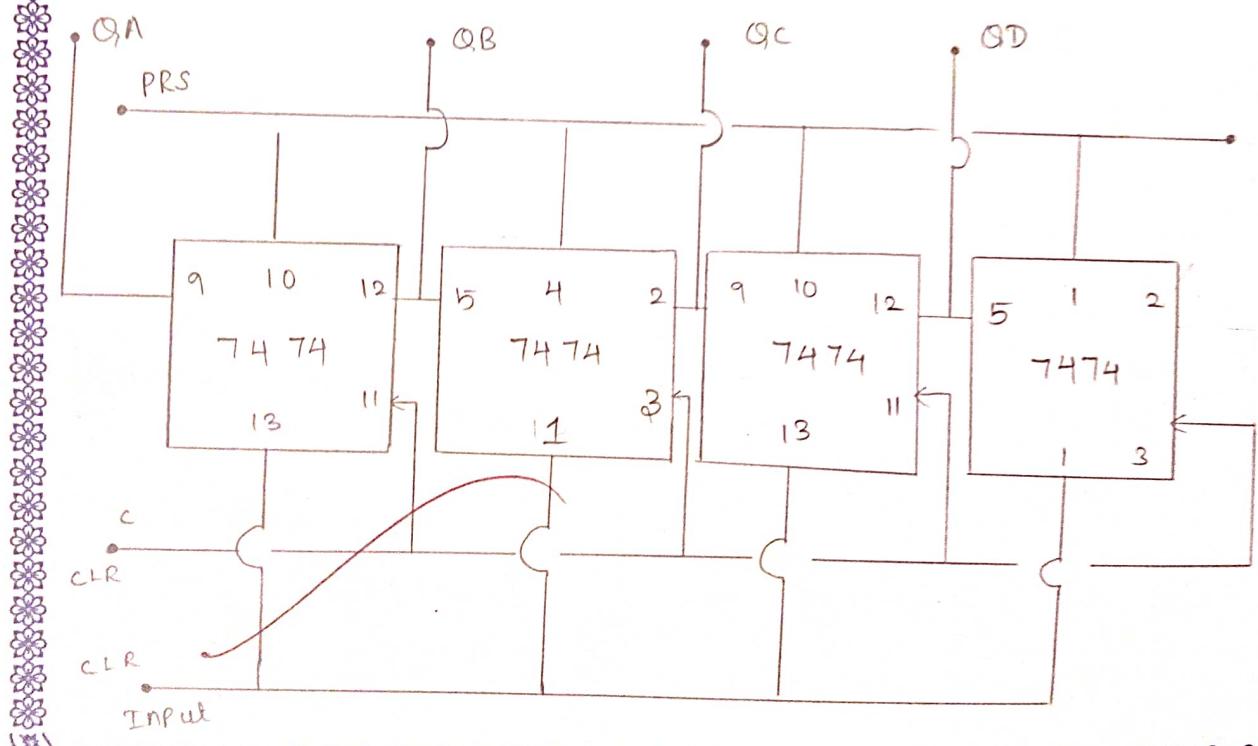
## PIN DIAGRAM :



RIGHT :

| PRE | CLR | SI | CLK | QA | QB | QC | QD |
|-----|-----|----|-----|----|----|----|----|
| 1   | 1   | 0  | -   | 0  | 0  | 0  | 0  |
| 1   | 1   | 1  | 1st | 0  | 0  | 0  | 1  |
| 1   | 1   | 1  | 2nd | 0  | 0  | 1  | 1  |
| 1   | 1   | 1  | 3rd | 0  | 1  | 1  | 1  |
| 1   | 1   | 1  | 4th | 1  | 1  | 1  | 1  |
| 1   | 1   | 0  | 5th | 1  | 1  | 1  | 0  |
| 1   | 1   | 0  | 6th | 1  | 1  | 0  | 0  |
| 1   | 1   | 0  | 7th | 1  | 0  | 0  | 0  |
| 1   | 1   | 0  | 8th | 0  | 0  | 0  | 0  |

SHIFT REGISTER [LEFT] :



LEFT:

| PRE | CLR | SI | CLK | QA | QB | QC | QD |
|-----|-----|----|-----|----|----|----|----|
| 1   | 0   | 0  | -   | 0  | 0  | 0  | 0  |
| 1   | 1   | 1  | 1st | 0  | 0  | 0  | 1  |
| 1   | 1   | 1  | 2nd | 0  | 0  | 1  | 1  |
| 1   | 1   | 1  | 3rd | 0  | 1  | 1  | 1  |
| 1   | 1   | 1  | 4th | 1  | 1  | 1  | 1  |
| 1   | 1   | 0  | 5th | 1  | 1  | 1  | 0  |
| 1   | 1   | 0  | 6th | 1  | 1  | 0  | 0  |
| 1   | 1   | 0  | 7th | 1  | 0  | 0  | 0  |
| 1   | 1   | 0  | 8th | 0  | 0  | 0  | 0  |

Result:

The SISO, SIPO, Shift register are constructed using D-flip flop and their truth tables are verified.

Ex: NO: 5

Date: 22.10.21

## UP - DOWN COUNTERS

## Aim:

TO construct and study the operation of up - counters Down - counters using JK flip flop and different asynchronous module counters using IC 7473.

## Apparatus Required:

IC 7473, JK... flip flop,  
IC 7420 NAND chips indicators board,  
logic power supply.

## DESCRIPTION :

A counter is a sequential logic made up of flip flops and is used to count the number of pulses applied to it. The output pulses changes the states of the flip flops in such a way that by observing the output levels the total number of input pulses applied can be determined. The ripple counter is one where each flip flop is triggered by the output of the proceeding flip flop.

## PROCEDURE:

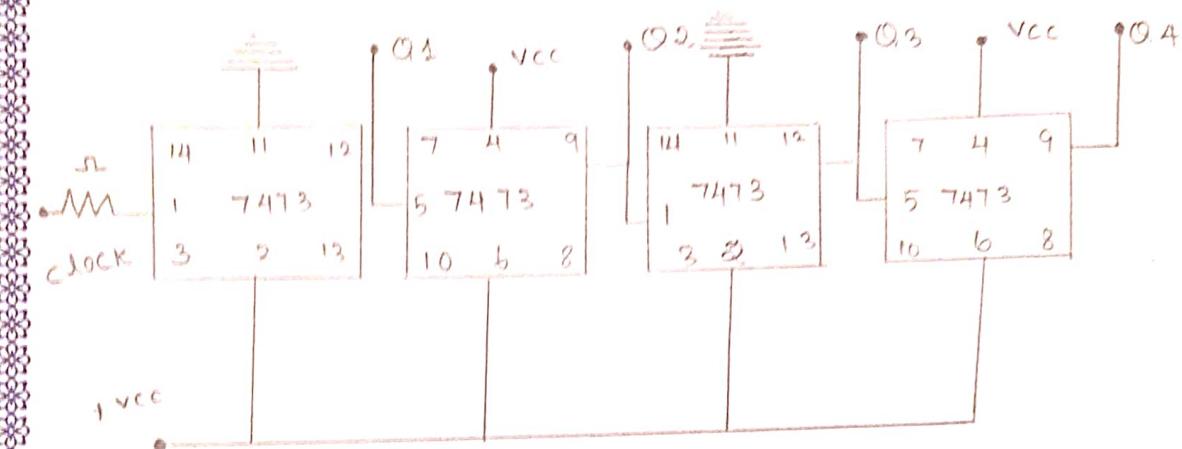
## UP - counter:

The circuit is wired as shown in figure. The clock pulse is applied to the clock output of the first flip flop. The output of the flip flop and so on. The clock pulses are applied one by one and the counters counts from zero to 15. The corresponding output is seen in the indicator board as well.

## Down - counter:

The circuit is wired as shown in figure. A pulses is connected to a pin 1 of a flip flop clock pulses / to B, C, D flip flops are given from the complements output of previous flip flop the clock is applied and the output is seen in the indicator board as well the counters count from 15 to zero.

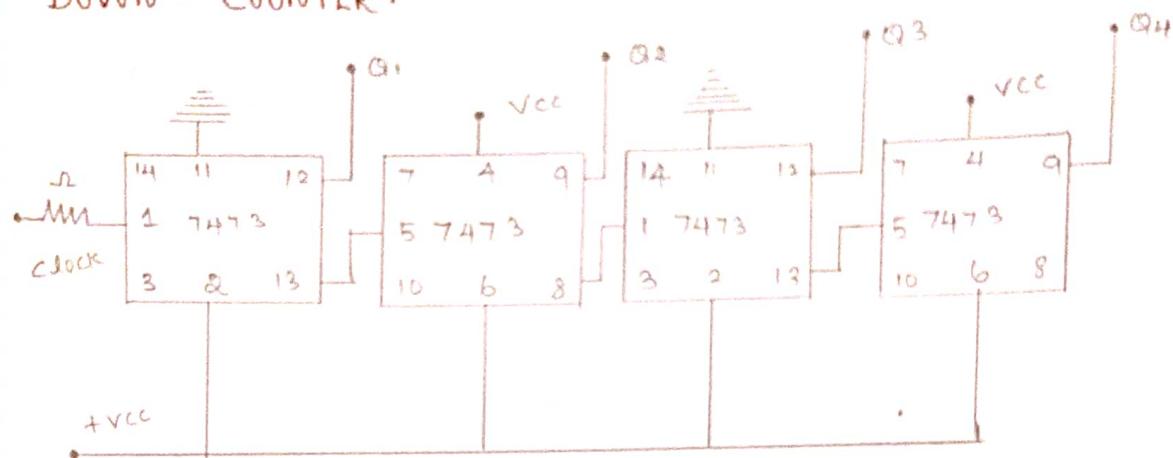
## UP - COUNTER:



## Tabulation:

| Count | D | C | B | A |
|-------|---|---|---|---|
| 0     | 0 | 0 | 0 | 0 |
| 1     | 0 | 0 | 0 | 1 |
| 2     | 0 | 0 | 1 | 0 |
| 3     | 0 | 0 | 1 | 1 |
| 4     | 0 | 1 | 0 | 0 |
| 5     | 0 | 1 | 0 | 1 |
| 6     | 0 | 1 | 1 | 0 |
| 7     | 0 | 1 | 1 | 1 |
| 8     | 1 | 0 | 0 | 0 |
| 9     | 1 | 0 | 0 | 1 |
| A     | 1 | 0 | 1 | 0 |
| B     | 1 | 0 | 1 | 1 |
| C     | 1 | 1 | 0 | 0 |
| D     | 1 | 1 | 0 | 1 |
| E     | 1 | 1 | 1 | 0 |
| F     | 1 | 1 | 1 | 1 |

## DOWN - COUNTER:



## Tabulation:

| count | D | C | B | A |
|-------|---|---|---|---|
| F     | 1 | 1 | 1 | 1 |
| E     | 1 | 1 | 1 | 0 |
| D     | 1 | 1 | 0 | 1 |
| C     | 1 | 1 | 0 | 0 |
| B     | 1 | 0 | 1 | 1 |
| A     | 1 | 0 | 1 | 0 |
| 9     | 1 | 0 | 0 | 1 |
| 8     | 1 | 0 | 0 | 0 |
| 7     | 0 | 1 | 1 | 1 |
| 6     | 0 | 1 | 1 | 0 |
| 5     | 0 | 1 | 0 | 1 |
| 4     | 0 | 1 | 0 | 0 |
| 3     | 0 | 0 | 1 | 1 |
| 2     | 0 | 0 | 1 | 0 |
| 1     | 0 | 0 | 0 | 1 |
| 0     | 0 | 0 | 0 | 0 |

### Result:

The up-counter, Down-counter using JK flip flop is constructed and different asynchronous module counters using IC 7473 is studied.

By

# MICROPROCESSOR EXPERIMENTS