ADICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Chikkamagaluru-577102

LAB MANUAL

(2019-20)

18CSL37

ANALOG AND DIGITAL ELECTRONICS LABORATORY

III Semester

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka –590018

III SEMESTER

18CSL37-ANALOG AND DIGITAL ELECTRONICS LABORATORY MANUAL

Mr. Suman M.H B.E M.Tech.
Asst. Professor
Dept. of IS&E
A.I.T Chikkamagaluru

Dr. Sampath S M.Tech., PhD
Professor & HOD
Dept. of IS&E
A.I.T Chikkamagaluru

ANALOG AND DIGITAL ELECTRONICS LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018 -2019)
SEMESTER – III

Subject Code: 18CSL37 **I.A. Marks: 40**
Hours/Week: 01I + 02P **Exam Hours: 03**
Total Hours: 40 **Exam Marks : 60**

Course objectives: This laboratory course enables students to get practical experience in design, assembly and evaluation/testing of :

- Analog components and circuits including Operational Amplifier, Timer, etc.
- Combinational logic circuits.
- Flip - Flops and their operations
- Counters and registers using flip-flops.
- Synchronous and Asynchronous sequential circuits.
- A/D and D/A converters

Descriptions (if any)

Any simulation package like MultiSim / P-spice /Equivalent software may be used.

Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-upon Logic design components, pin diagram (if any), Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 20 marks as lab experiments.

CONTENTS

Sl. No.	Program	Page No.
1	Introduction to ADE laboratory	4 – 9
2	Study of LOGIC GATES	10 – 14
3	Design an astable multivibrator ciruit for three cases of duty cycle (50%, 50%) using NE 555 timer IC. Simulate the same for any one duty cycle.	15 – 17
4	Using ua 741 Opamp, design a 1 kHz Relaxation Oscillator with 50% duty cycle. And simulate the same.	18 – 20
5	Using ua 741 opamap, design a window comparate for any given UTP and LTP. And simulate the same.	21 - 24
6	Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates. And implement the same in HDL.	25 – 30
7	Given a 4-variable logic expression, simplify it using appropriate technique and realize the simplified logic expression using 8:1 multiplexer IC. And implement the same in HDL.	31 – 34
8	Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table. And implement the same in HDL.	35 – 40
9	Design and implement code converter I)Binary to Gray (II) Gray to Binary Code using basic gates.	41 – 43
10	Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.	44 – 46
11	Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447)	47 – 51
12	Viva – Voice questions	56 - 58

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- Design and demonstrate various combinational logic circuits.
- Design and demonstrate various types of counters and Registers using Flip-flops
  Use simulation package to design circuits.
- Understand the working and implementation of ALU.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (1 to 11 nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:

   a) For questions having part a only-

   Procedure + Conduction + Viva: **20 + 50+10 =80**

   Marks b) For questions having part a and b

   Part a- Procedure + Conduction + Viva: 10 **+ 35 +05= 50 Marks**

   Part b- Procedure + Conduction + Viva: 10 **+ 15 +05= 30 Marks**

5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

INTRODUCTION TO ADE LAB

ELECTRONIC COMPONENTS

1. RESISTORS

A resistor is a component of an electrical circuit that resists the flow of electrical current. A resistor has two terminals across which electricity must pass, and is designed to drop the voltage of the current as it flows from one terminal to the next. A resistor is primarily used to create and maintain a known safe current within an electrical component.

Resistance is measured in ohms, after Ohm's law. A 1000 Ohm resistor is typically shown as 1K-Ohm (kilo Ohm), and 1000 K-Ohms is written as 1M-Ohm (mega ohm).

Figure 1: A Resistor

Resistor Colour Code:

Colour	1st Band	2nd Band	3rd Band (multiplier)	4th Band (tolerance)
Black	0	0	x 10^0	
Brown	1	1	x 10^1	± 1%
Red	2	2	x 10^2	± 2%
Orange	3	3	x 10^3	
Yellow	4	4	x 10^4	
Green	5	5	x 10^5	± 0.5%
Blue	6	6	x 10^6	± 0.25%
Violet	7	7	x 10^7	± 0.1%
Gray	8	8	x 10^8	± 0.05%
White	9	9	x 10^9	
Gold			x 0.1	± 5%
Silver			x 0.01	± 10%

2. Capacitors

The capacitor's capacitance (*C*) is a measure of the amount of charge (*Q*) stored on each plate for a given potential difference or *voltage* (*V*) which appears across the plates. In SI units, a capacitor has a capacitance is measured in farad (F).

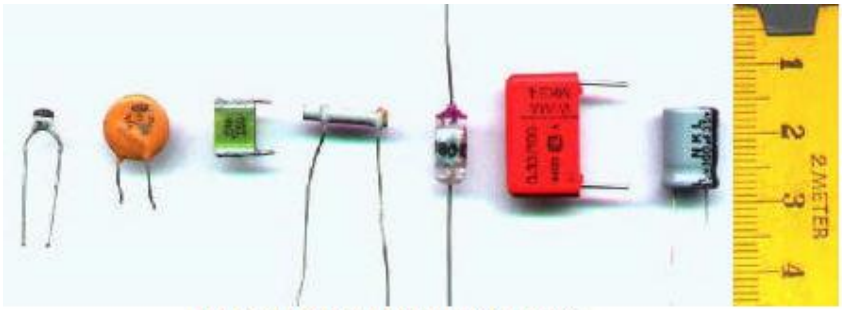


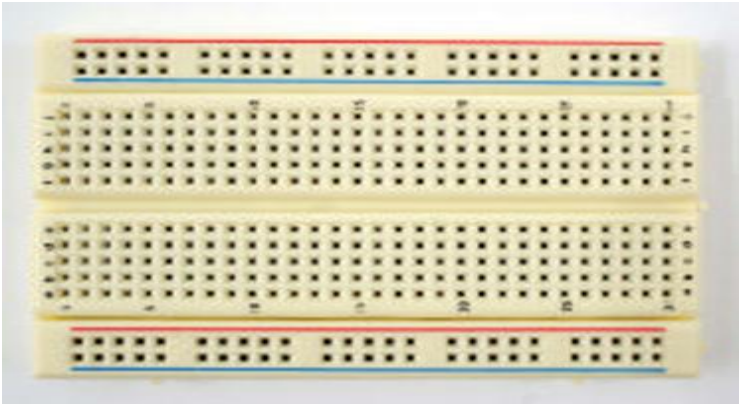
Figure 2: different capacitor models

Picofarad (pF)	Nanofarad (nF)	Microfarad (uF)	Code	Picofarad (pF)	Nanofarad (nF)	Microfarad (uF)	Code
10	0.01	0.00001	100	4700	4.7	0.0047	472
15	0.015	0.000015	150	5000	5.0	0.005	502
22	0.022	0.000022	220	5600	5.6	0.0056	562
33	0.033	0.000033	330	6800	6.8	0.0068	682
47	0.047	0.000047	470	10000	10	0.01	103
100	0.1	0.0001	101	15000	15	0.015	153
120	0.12	0.00012	121	22000	22	0.022	223
130	0.13	0.00013	131	33000	33	0.033	333
150	0.15	0.00015	151	47000	47	0.047	473
180	0.18	0.00018	181	68000	68	0.068	683
220	0.22	0.00022	221	100000	100	0.1	104
330	0.33	0.00033	331	150000	150	0.15	154
470	0.47	0.00047	471	200000	200	0.2	254
560	0.56	0.00056	561	220000	220	0.22	224
680	0.68	0.00068	681	330000	330	0.33	334
750	0.75	0.00075	751	470000	470	0.47	474
820	0.82	0.00082	821	680000	680	0.68	684
1000	1.0	0.001	102	1000000	1000	1.0	105
1500	1.5	0.0015	152	1500000	1500	1.5	155
2000	2.0	0.002	202	2000000	2000	2.0	205
2200	2.2	0.0022	222	2200000	2200	2.2	225
3300	3.3	0.0033	332	3300000	3300	3.3	335

3. Breadboard

A breadboard is a material or a d evice used to build a prototype of an electronic circuit.

The breadboard has many strip s of metal (copper usually) which run underne aththe board. The metal strips are laid out as s hown These strips connect the holes on the top of the board. This makes it easy to connect co mponents together to build circuits. To use the bread board, the legs ofcomponents are place d in the holes (the sockets). The holes are made so that they will hold the component in place. Each hole is connected to one of the metal strips running underneath the board. The long top and bottom row of holes are usually used for power supply connections.

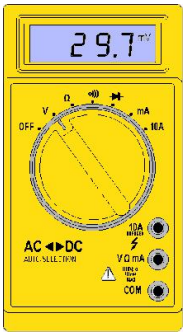


4. Power Supply

A power supply is a separate u nit or part of a circuit that supplies power to the rest of the circuit or to a system. The power supply takes the current from your wall electrical socket and converts it into the various voltages your circuit needs.

5. Multimeter

A meter is a measuring instrument. An ammeter measures current, a voltmeter measures the potential difference (voltage) between two points, and an ohmmeter measures resistance. A multimeter combines these functions and possibly some additional ones as well, into a single instrument.



6. Signal/Function Generator

A function generator is a device that can produce various patterns of voltage at a variety of frequencies and amplitudes. It is used to test the response of circuits to common input signals. The electrical leads from the de vice are attached to the ground and signal input terminals of the device under test.

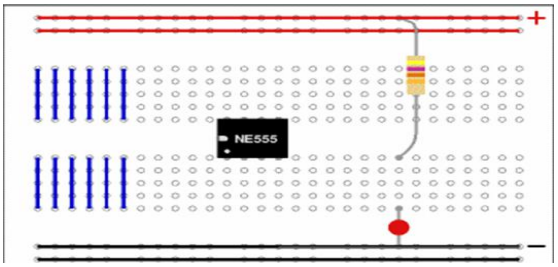
Most function generators allow the user to choose the shape of the output from a small number of options.

- Square wave - The signal goes directly from high to low voltage.
- Sine wave - The signal curves like a sinusoid from high to low voltage.
- Triangle wave - The signal goes from high to low voltage at a fixed rate.

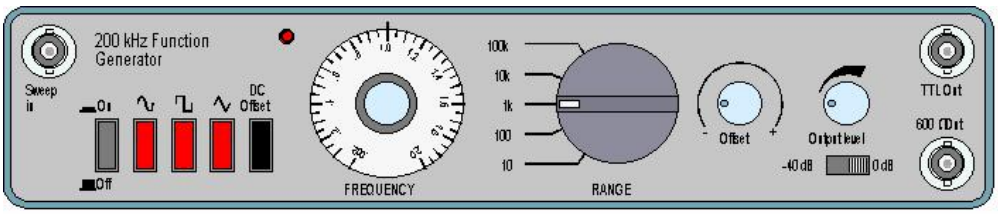
The amplitude control on a function generator varies the voltage difference between the high and low voltage of the output signal. The frequency control of a function generator controls the rate at which output signal oscillates.

Most function generators allow the user to choose the shape of the output from a small number of options.

- Square wave - The signal goes directly from high to low voltage.
- Sine wave - The signal curves like a sinusoid from high to low voltage.
- Triangle wave - The signal goes from high to low voltage at a fixed rate.



The amplitude control on a function generator varies the voltage difference between the high and low voltage of the output signal. The frequency control of a function generator controls the rate at which output signal oscillates.

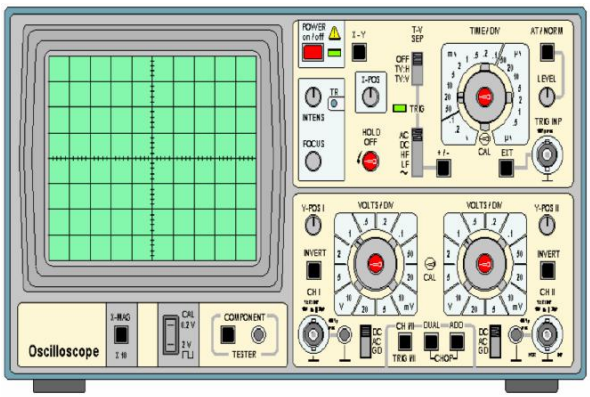


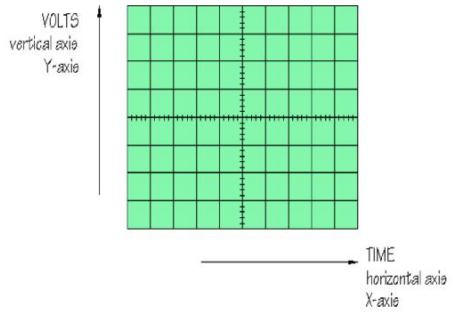
Switch on the function generator and adjust the output level to produce a visible signal on the oscilloscope screen. Adjust TIME/DIV and VOLTS/DIV to obtain a clear display and investigate the effects of pressing the waveform shape buttons.

The rotating FREQUENCY control and the RANGE switch are used together to determine the frequency of the output signal.

7. Oscilloscope

An oscilloscope is easily the most useful instrument available for testing circuits because it allows you to see the signals at different points in the circuit. The best way of investigating an electronic system is to monitor signals at the input and output of each system block, checking that each block is operating as expected and is correctly linked to the next.





The **screen** of this oscilloscope has 8 squares or divisions on the vertical axis, and 10 squares or divisions on the horizontal axis. Usually, these squares are 1 cm in each direction:

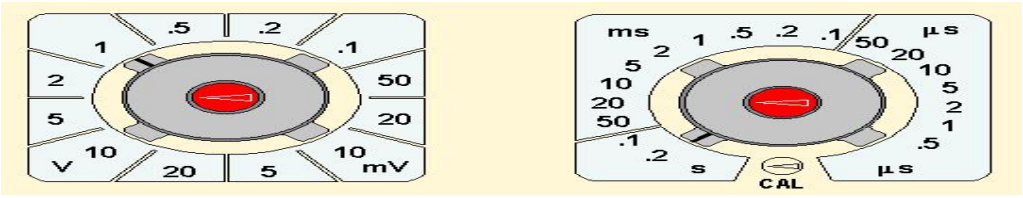
Setting up the CRO

i. **Before you switch the o scilloscope on, check that all the controls are in their 'normal' positions.**

a) all push button switches are in the OUT position
b) all slide switches are in t he UP position
c) all rotating controls are C ENTRED

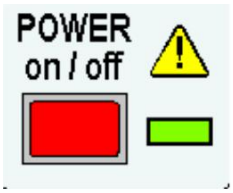
ii. **Check through all the controls and put them in these positions:**

iii. Set both VOLTS/DIV controls to 1 V/DIV and the TIME/DIV contr ol to 2 s/DIV, its slowest setting:



VOLTS/DIV TIME/DIV

iv. Switch ON, red button, top center:



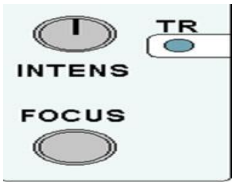
The green LED illuminates and, after a few moments, you should see a small bright spot, or trace, moving fairly slowly across the screen.

v. Find the Y-POS 1 control:

The Y-POS 1 allows you to move the spot up and down the screen. For the present, adjust the trace so that it runs horizontally across the center of the screen.

vi. Now investigate the INT ENSITY and FOCUS controls:

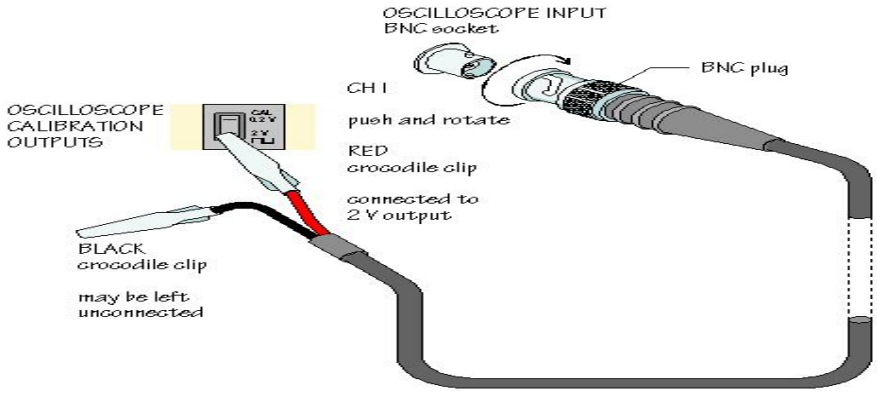
When these are correctly set, the spot will be reasonably bright but not glaring, and as sharply focused as possible. (The TR control is screwdriver adjusted. It is only needed if the spot moves at an angle rather than horizontally across the screen with no signal connected.)



Adjusting the INTENSITY control changes the brightness of the oscilloscope display. The FOCUS should be set to produce a bright clear trace. If required, TR can be adjusted using a small screwdriver so that the oscilloscope trace is exactly horizontal when no signal is connected.

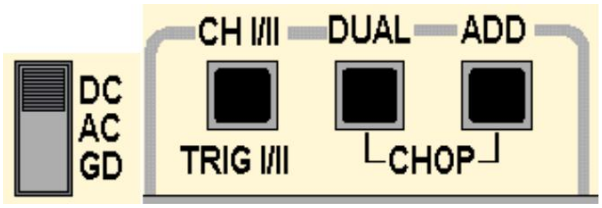
vii. The TIME/DIV control determines the horizontal scale of the graph which appears on the oscilloscope screen.

viii. The VOLTS/DIV controls determine the vertical scale of the graph drawn on the oscilloscope screen.



The diagram shows a lead with a BNC plug at one end and crocodile clips at the other. Adjust VOLTS/DIV and TIME/DIV until you obtain a clear picture of the signal, which should look like this:

DC/AC/GND slide switches: In the DC position, the signal input is connected directly to the Y-amplifier of the corresponding channel, CH I or CH II. In the AC position, a capacitor is connected into the signal pathway so that DC voltages are blocked and only changing AC signals are displayed.



In the GND position, the input of the Y-amplifier is connected to 0 V. This allows you to check the position of 0 V on the oscilloscope screen.

Trace selection switches: The settings of these switches control which traces appear on the oscilloscope screen.

STUDY OF LOGIC GATES

AIM:

To study about logic gates and verify their truth tables.

COMPONENTS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	NAND GATE 2 I/P	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1
7.	NAND GATE 3 I/P	IC 7410	1
8.	IC TRAINER KIT	-	1
9.	PATCH CORD	-	14

THEORY:

Circuit that takes the logical decision and the process are called logic gates. Each gate has one or more input and only one output. OR, AND, NOT are basic gates. NAND, NOR and X-OR are known as universal gates. Basic gates form these gates.

AND GATE:

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

OR GATE:

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

NOT GATE:

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

NAND GATE:

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

NOR GATE:

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

X-OR GATE:

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

PROCEDURE:

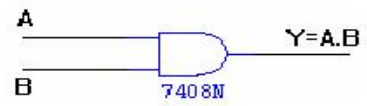
(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

AND GATE:

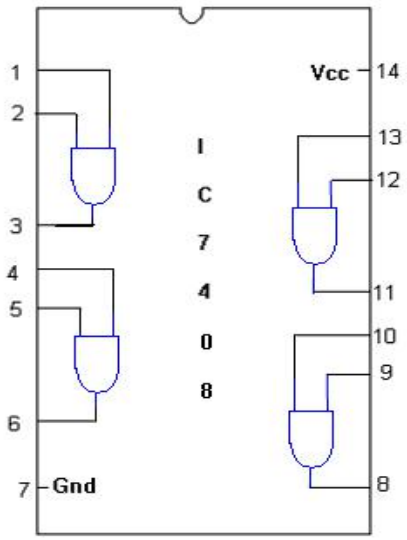
SYMBOL:



TRUTH TABLE

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

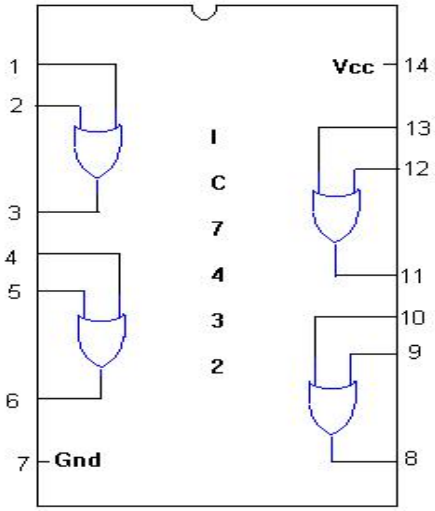
PIN DIAGRAM:



OR GATE:

SYMBOL :

PIN DIAGRAM :

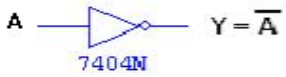


TRUTH TABLE

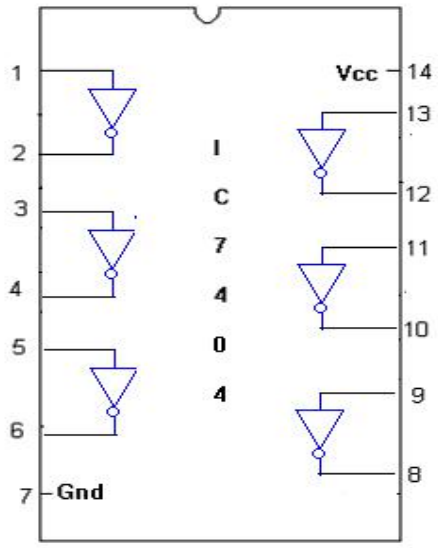
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE:

SYMBOL:



PIN DIAGRAM:



TRUTH TABLE :

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

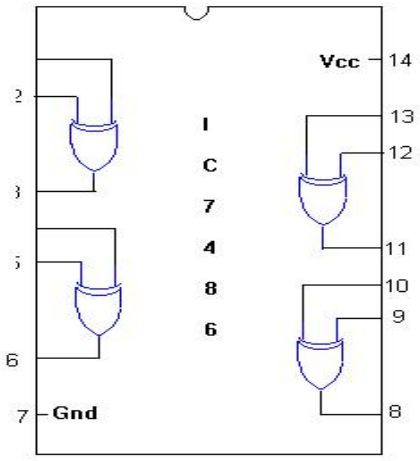
X-OR GATE:

SYMBOL:

TRUTH TABLE :

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

PIN DIAGRAM:



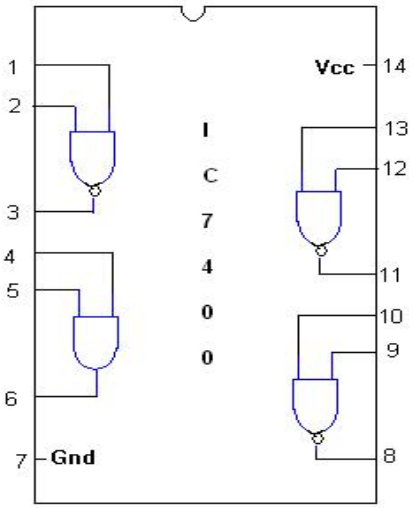
2-INPUT NAND GATE:

SYMBOL:

TRUTH TABLE

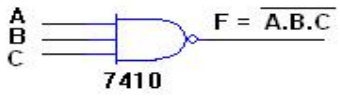
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

PIN DIAGRAM:

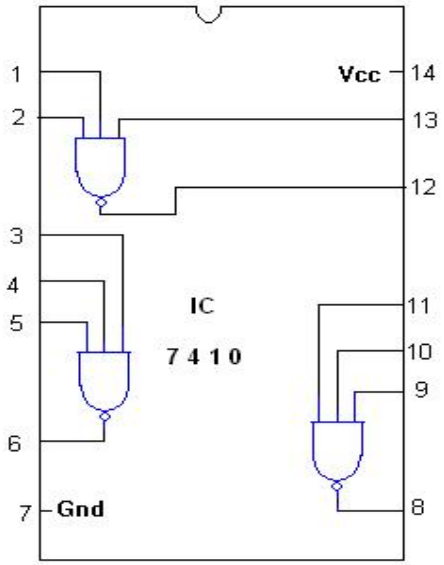


3-INPUT NAND GATE:

SYMBOL :



PIN DIAGRAM :

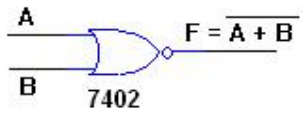


TRUTH TABLE

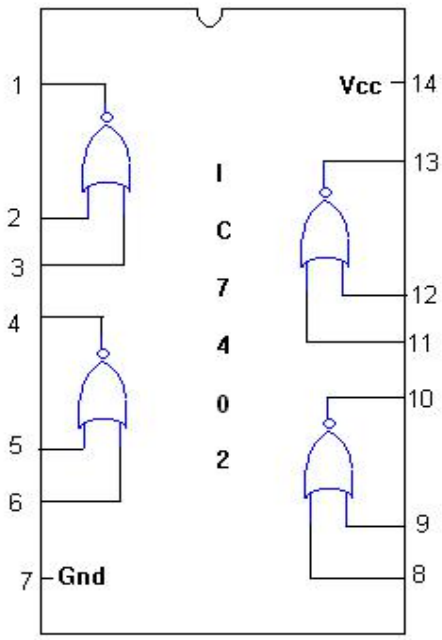
A	B	C	$\overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NOR GATE:

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

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

EXPERIMENT 1

ASTABLE MULTIVIBRATOR USING 555 TIMER

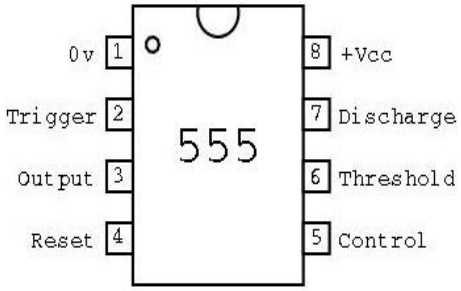
AIM: Design and implement an astable multivibrator using 555 Timer for a given frequency and duty cycle.

a) Duty cycle = 75 % and f = 1 KHz

b) Duty cycle = 50% and f = 1 KHz

COMPONENTS REQUIRED: 555 Timer IC, Resistors of 3.3KΩ, 6.8KΩ, Capacitors of 0.1 µF, 0.01 µF, Regulated power supply, CRO.

Pin Diagram Of Ic 555



DESIGN:

Case i) For Duty Cycle > 50%

Given frequency (f) = 1 KHz and duty cycle = 75% (=0.75)

Therefore T=1/f=1ms=Ton+Toff=1ms

Capacitor charges through RA and RB there for charging time is given by

Ton=0.693(RA+RB) C

during this time the output is High, so

Ton=0.693(RA+RB) C = 0.75 ms----------(1)

Capacitor discharges through R_2 only there for discharging time is given by

Toff=0.693 RB C,

During this time the output is Low, so

Toff=0.693 RB C = 0.25 ms----------(2)

$T = T_{on} + T_{off} = 0.693 (R_A + 2 R_B) C = 1ms$

Duty Cycle is calculated by the following manner=Ton/Ton+Toff=0.75

Duty cycle = $T_{on} / T = 0.75$. Hence $T_{on} = 0.75T = 0.75$ ms and $T_{off} = T - T_C = 0.25$ms.

Let C=0.1µF and substituting in the above equations,

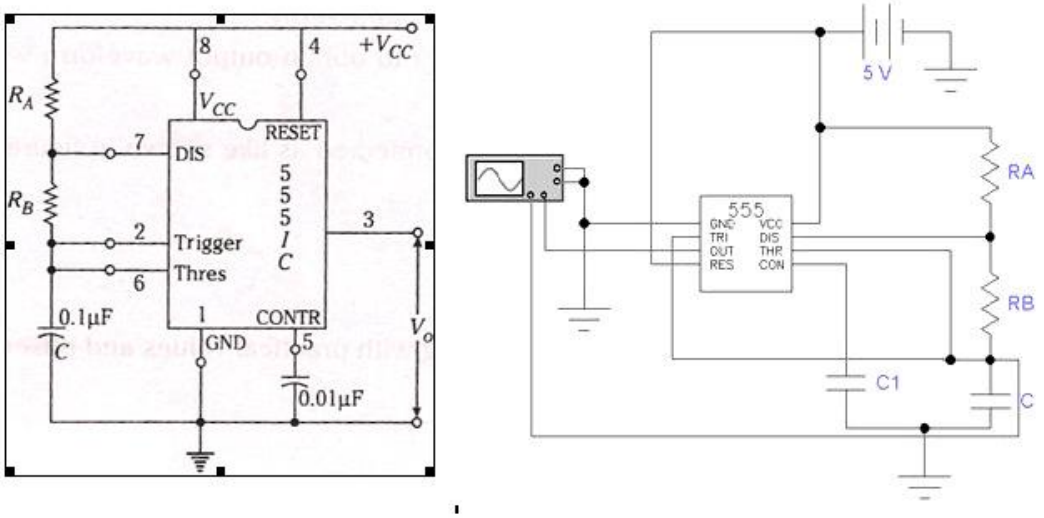
R_B = 3.6 KΩ (from equation 2) and

R_A = 7.2KΩ (from equation 1 & R_B values).

Choose RA = 6.8kΩ and RB = 3.3kΩ.

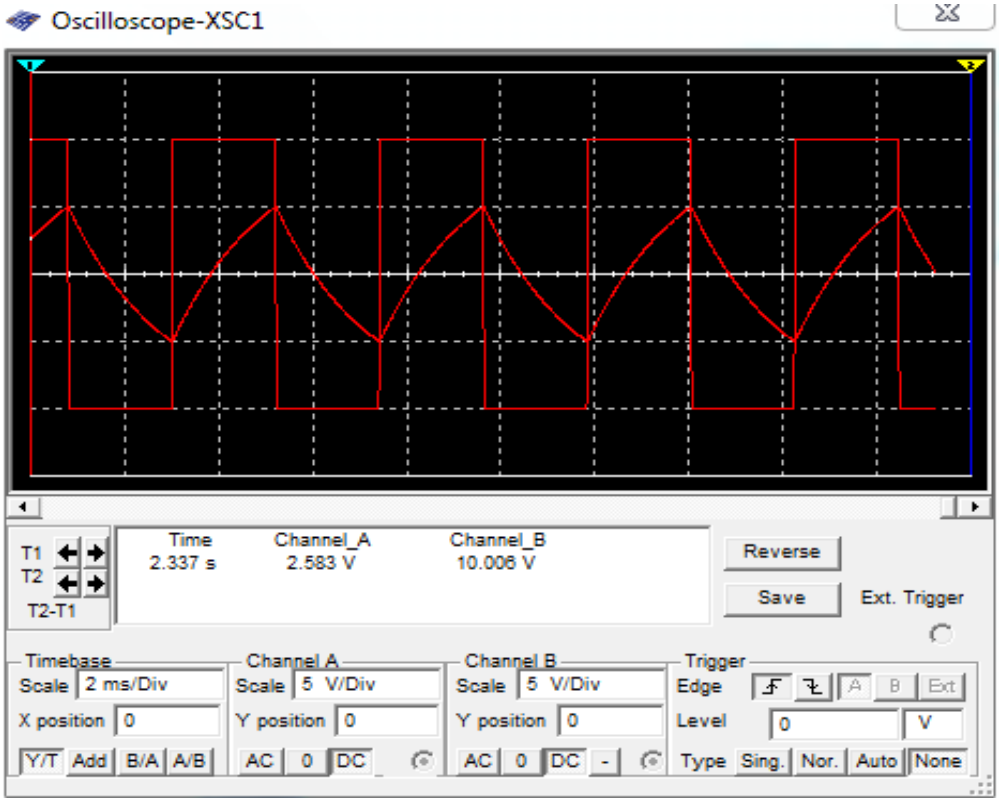
The Vcc determines the upper and lower threshold voltages (observed from the capacitor voltage waveform) as $V_{UT} = \frac{2}{3}V_{CC}$ & $V_{LT} = \frac{1}{3}V_{CC}$.

Note: The duty cycle determined by R_A& R_B can vary only between 50 & 100%. If R_A is much smaller than R_B, the duty cycle approaches 50%.



Circuit Diagram and actual connections

Astable Multivibrator : Output



DESIGN:

Case ii) For Duty Cycle = 50%

Therefore T=1/f=1ms=Ton+Toff =1ms

Given frequency (f) = 1 KHz an d duty cycle = 50% (=0.5)

To achieve a duty cycle of less t han or equal to 50% is to **connect a diode D across resistor RB**

The time for which the output is high is given by

$$t_{ON} = t_C = 0.693\ (RA)*C$$

$$t_{OFF} = t_d = 0.693\ (RB)*C$$

Thus the total period of the output waveform is

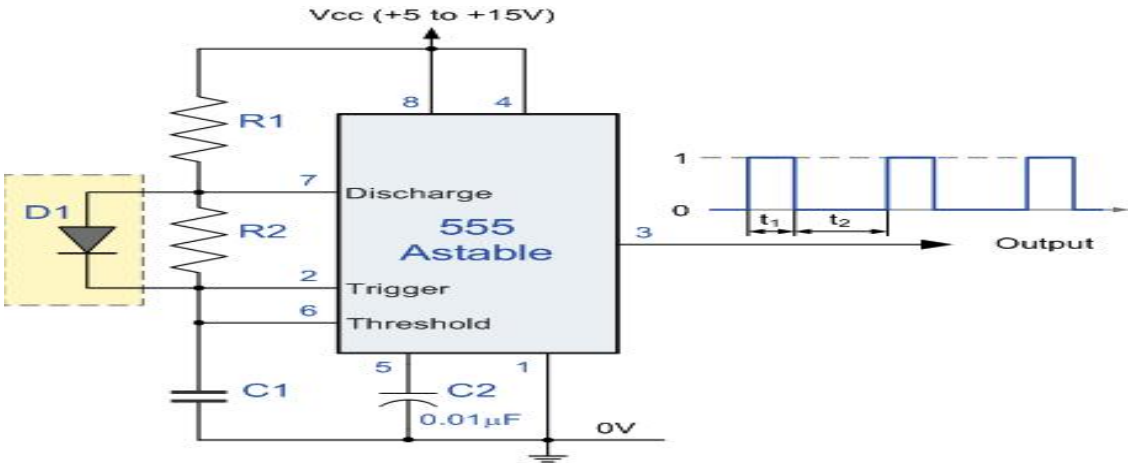
$$T = t_{ON} + t_{off} = 0.693\ (RA + RB)*C = 1ms$$

For duty cycle equal to 50% RA needs to be equal to resistor RB.

For RA=R B=R,
T=0.693*2RA*C=1ms

Therefore 2RA=14.4 kΩ.

Hence RA=7.2 kΩ =RB



Astable Multivibrator using 555 for 50% Duty Cycle

For RA <RB , the duty cycle is less than 50%.

In this case, the capacitor C charges through RA and diode D to approximately 2/3 Vcc and discharges through RB until the capacitor voltage equals approximately 1/3 Vcc, after which the cycle repeats.

THEORY:

Multivibrator is a form of oscillator, which has a non-sinusoidal output. The output waveform is rectangular. The multivibrators are classified as: **Astable or free running multivibrator**: It alternates automatically between two states (low and high for a rectangular output) and remains in each state for a time dependent upon the circuit constants. It is just an oscillator as it requires no external pulse for its operation. **Monostable or one shot multivibrator**: It has one stable state and one quasi stable. The application of an input pulse triggers the circuit time constants. After a period of time determined by the time constant, the circuit returns to its initial stable state. The process is repeated upon the

application of each trigger pulse. **Bistable Multivibrators**: It has both stable states. It requires the application of an external triggering pulse to change the output from one state to other. After the output has changed its state, it remains in that state until the application of next trigger pulse. Flip flop is an example.

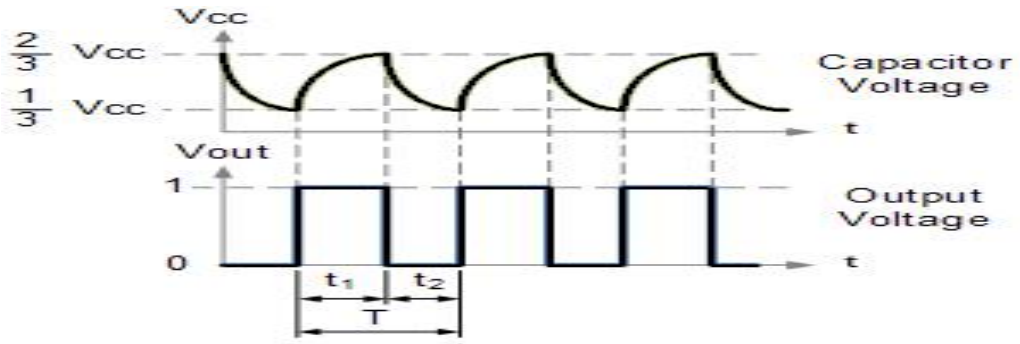
PROCEDURE:

1. Before making the connections, check the components using multimeter.
2. Make the connections as shown in figure and switch on the power supply.
3. Observe the capacitor voltage waveform at 6th pin of 555 timer on CRO.
4. Observe the output waveform at 3rd pin of 555 timer on CRO (shown below).
5. Note down the amplitude levels, time period and hence calculate duty cycle.

RESULT:

The frequency of the oscillations = ……Hz.

WAVEFORMS



Results: An astablemultivibrator of given duty cycle and frequency is designed. A comparison of the experimental values with the given ones is represented below:

Quantity measured	Theoretical value	Experimental value
Frequency		
Duty Cycle		

EXPERIMENT 2

OP-AMP AS A RELAXATION OSCILLATOR

AIM: Design and construct a rectangular waveform generator (op-amp relaxation oscillator) for a given frequency and demonstrate its working.

COMPONENTS REQUIRED:

Op-amp µA 741, Resistor of 1KΩ, 10KΩ, 20 kΩ Potentiometer, Capacitor of 0.1 µF, Regulated DC power supply, CRO

DESIGN:

The period of the output rectangular wave is given as $T = 2RC \ln \frac{1+\beta}{1-\beta}$ -------(1)

Where, $\beta = \frac{R_1}{R_1 + R_2}$ is the feedback fraction

Example: Design for a frequency of 1kHz (implies $T = \frac{1}{f} = \frac{1}{10^3} = 10^{-3} = 1ms$)

Use R_2=1.16 R_1,

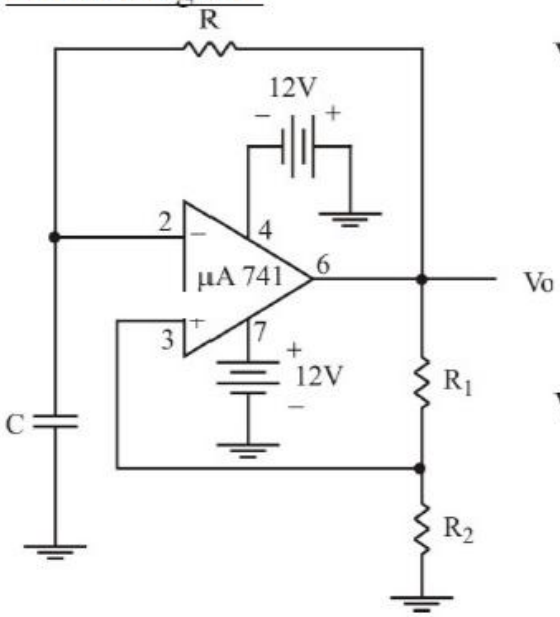
Let R_1 = 10kΩ, then R_2 = 10 kΩ (use 20kΩ potentiometer)

Choose next a value of C and then calculate value of R from equation (2).

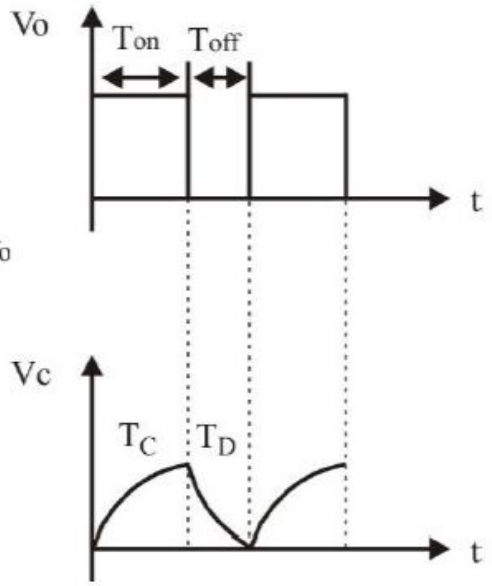
Let C=0.1µF (i.e., 10^{-7}), then $R = \frac{T}{2C} = \frac{10^{-3}}{2 \times 10^{-7}} = 5K\Omega$

The voltage across the capacitor has a peak voltage of $V_c = \frac{R_1}{R_1 + R_2} V_{sat}$

Circuit Diagram:



Wave forms:

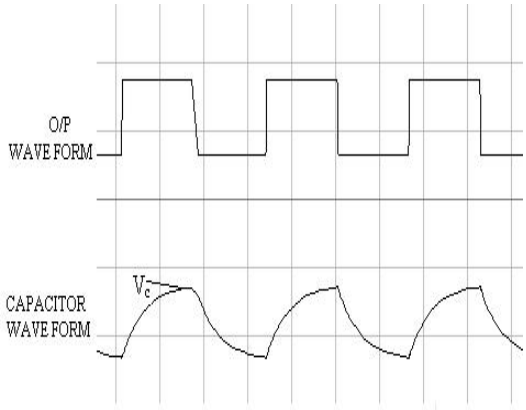


Values

C=0.1µF

R_1 = 10kΩ, R_2 = 11.6 kΩ,R = 4.7k/5.1kΩ

WAVEFORMS



THEORY:

Op-Amp Relaxation Oscillator is a simple Square wave generator which is also called as a Free running oscillator or Astable multivibrator or Relaxation oscillator. In this figure the op-amp operates in the saturation region. Here, a fraction (R2/(R1+R2)) of output is fed back to the non-inverting input terminal. Thus reference voltage is (R2/(R1+R2)) Vo and may take values as +(R2/(R1+R2)) Vsat or - (R2/(R1+R2)) Vsat. The output is also fed back to the inverting input terminal after integrating by means of a low-pass RC combination. Thus whenever the voltage at inverting input terminal just exceeds reference voltage, switching takes place resulting in a square wave output.

If R1 = R_2, we have T = 2RC ln (3)

Another example, if R_2=1.16 R_1, then T = 2RC ----------(2)

PROCEDURE:

1. Before making the connections check all the components using multimeter.
2. Make the connections as shown in figure and switch on the power supply.
3. Observe the voltage waveform across the capacitor on CRO.
4. Also observe the output waveform on CRO. Measure its amplitude and frequency.

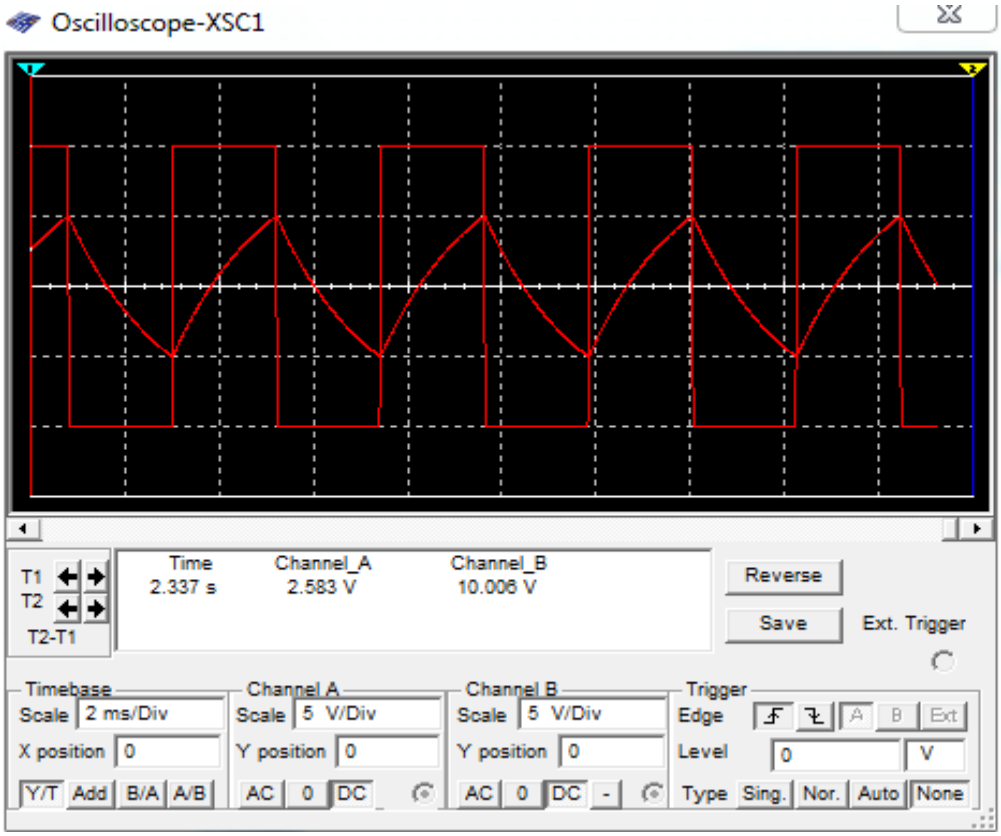
RESULT:

The frequency of the oscillations = ……Hz.

Experiment No. 2 (b)

Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.

Output:



EXPERIMENT 3

SCHMITT TRIGGER

AIM: Design and construct a Schmitt trigger circuit using op-amp for the given UTP and LTP values and demonstrate its working..

COMPONENTS REQUIRED: IC µA741, Resistor of 10KΩ, 100KΩ, DC regulated power supply, Signal generator, CRO

DESIGN:

1)Let given UTP = 2.5 V, LTP= 1V. Assume Vsat = 12 V

$UTP = \frac{R_1 V_{ref}}{R_1 + R_2} + \frac{R_2 V_{sat}}{R_1 + R_2}$ where V_{sat} is the positive saturation of the opamp V_{cc}

$\& \; LTP = \frac{R_1 V_{ref}}{R_1 + R_2} - \frac{R_2 V_{sat}}{R_1 + R_2}$

Hence given the LTP & UTP values to find the R_1 , R_2 & V_{ref} values, the following design is used.

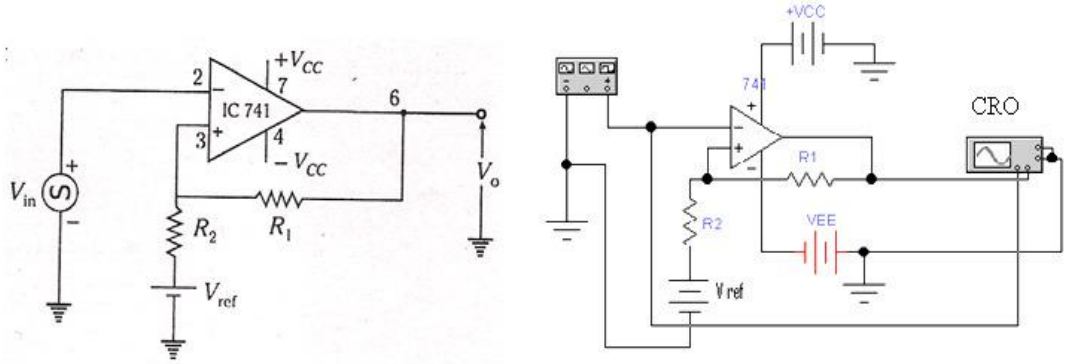
$UTP + LTP = \frac{2R_1 V_{ref}}{R_1 + R_2}$ ------(1)

$UTP - LTP = \frac{2R_1 V_{sat}}{R_1 + R_2}$ ------(2)

Let V_{sat} = 12V, UTP = 2.5 V & L TP = 1V, then equation (2) yields R_1 = $15R_2$ Let $R_2 = 1K\Omega$, then $R_1 = 15K\Omega$

From equation (1) we have $V_{ref} = \frac{(UTP + LTP)(R1 + R2)}{2R_1} = 1.88V$

Circuit Diagram of Schmitt Trigger Circuit



DESIGN 2)

Let given UTP = 2 V, LTP= -1V. Assume Vsat = 12 V

$UTP = \frac{R_1 V_{ref}}{R_1 + R_2} + \frac{R_2 V_{sat}}{R_1 + R_2}$ where V_{sat} is the positive saturation of the opamp V_{cc}

$\& \; LTP = \frac{R_1 V_{ref}}{R_1 + R_2} - \frac{R_2 V_{sat}}{R_1 + R_2}$

Hence given the LTP & UTP values to find the R_1 , R_2 & V_{ref} values, the following design is used.

$UTP + LTP = \frac{2R_1 V_{ref}}{R_1 + R_2}$ ------(1)

$UTP - LTP = \frac{2R_1 V_{sat}}{R_1 + R_2}$ ------(2)

Let V_{sat} = 12V, UTP = 2 V & LTP = -1V, then equation (2) yields $R_1 = 7R_2$

Let $R_2 = 1K\Omega$, then $R_1 = 7K\Omega$

From equation (1) we have $V_{ref} = \frac{(UTP + LTP)(R_1 + R_2)}{2R_1} = 0.57V$

Design 3) Let given UTP = 4V, LTP= 2V. Assume Vsat = 12 V

$UTP = \frac{R_1 V_{ref}}{R_1 + R_2} + \frac{R_2 V_{sat}}{R_1 + R_2}$ where V_{sat} is the positive saturation of the opamp

$\& \; LTP = \frac{R_1 V_{ref}}{R_1 + R_2} - \frac{R_2 V_{sat}}{R_1 + R_2}$

Hence given the LTP & UTP values to find the R_1 , R_2 & V_{ref} values, the following design is used.

$UTP + LTP = \frac{2R_1 V_{ref}}{R_1 + R_2}$ ------(1)

$UTP - LTP = \frac{2R_1 V_{sat}}{R_1 + R_2}$ ------(2)

Let V_{sat} = 12V, UTP = 4 V & LTP = 2V, then equation (2) yields $R_1 =$
$9R_2$ $R_2 = 10K\Omega$, then $R_1 = 90K\Omega$

From equation (1) we have $V_{ref} = \frac{(UTP + LTP)(R_1 + R_2)}{2R_1} = 3.33V$

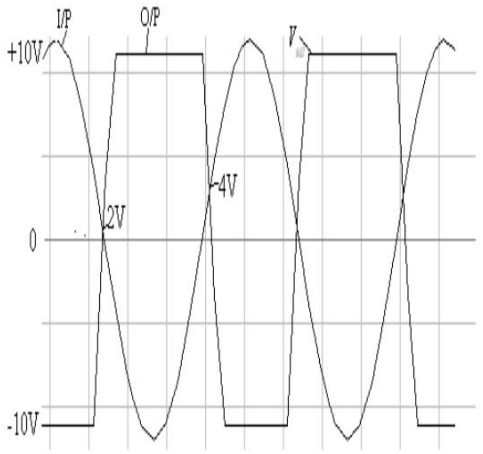
THEORY:

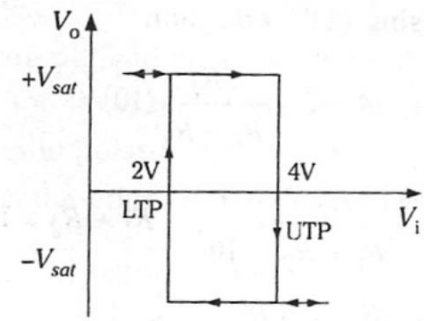
Schmitt Trigger converts s an irregular shaped waveform to a square wave or pulse. Here, the input voltage triggers the output voltage every time it exceeds certain voltage levels called the upper threshold voltage VUTP and lower threshold voltage VLTP. The input voltage is applied to the inverting input. Because the feedback voltage is aiding the input voltage, the feedback is positive. A comparator using positive feedback is usually called a Schmitt Trigger. Schmitt Trigger is used as a squaring circuit, in digital circuitry, amplitude comparator, etc.

PROCEDURE:

1. Before doing the connections, check all the components using multimeter.
2. Make the connection as shown in circuit diagram.
3. Using a signal generator apply the sinusoidal input waveform of peak-to-peak amplitude of 10V, frequency 1kHz.
4. Keep the CRO in dual mode; apply input (Vin) signal to the channel 1 an d observe the output (Vo) on channel 2 which is as shown in the waveform below. Note the amplitude levels from the waveforms.
5. Now keep CRO in X-Y mode and observe the hysteresis curve.

Waveforms:



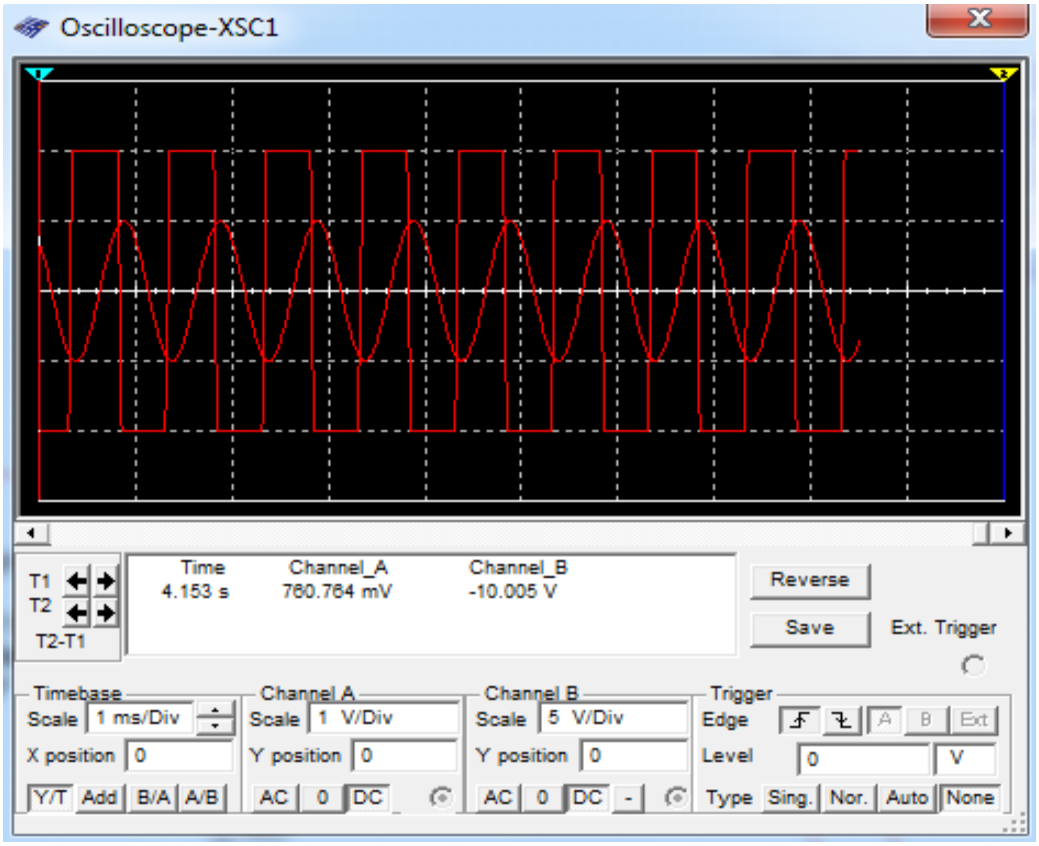


CRO in DUAL mode CRO in X-Y mode showing the Hysteresis curve.

Experiment No. 1(b)

Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.

Output:



EXPERIMENT NO.4

DESIGN OF ADDER AND SUBTRACTOR

AIM:

To design and construct half adder, full adder, half subtractor and full subtractor circuits and verify the truth table using logic gates.

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	23

THEORY:

HALF ADDER:

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

FULL ADDER:

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

HALF SUBTRACTOR:

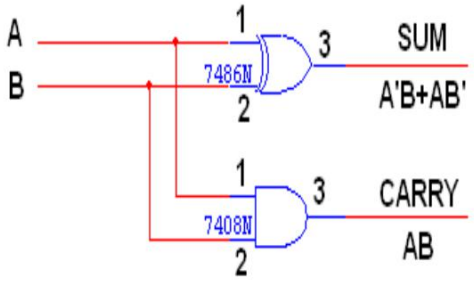
The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

FULL SUBTRACTOR:

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

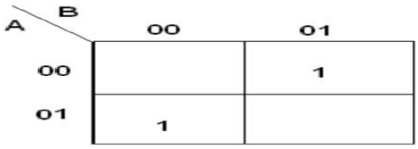
LOGIC DIAGRAM:

HALF ADDER

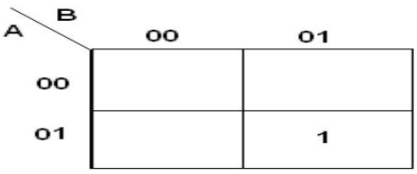


TRUTH TABLE:

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



K-Map for SUM:



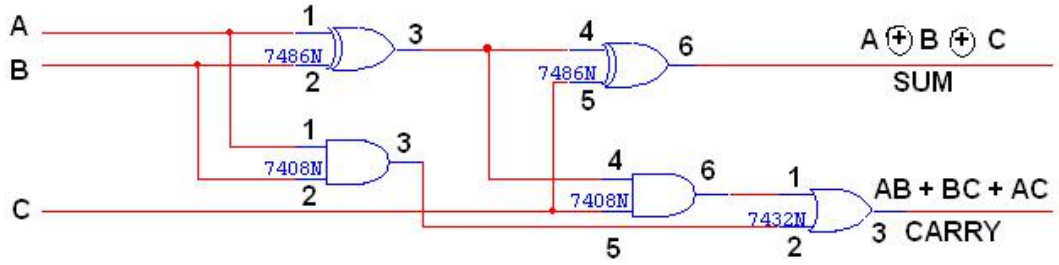
K-Map for CARRY:

SUM = A'B + AB' **CARRY = AB**

FULL ADDER

LOGIC DIAGRAM:

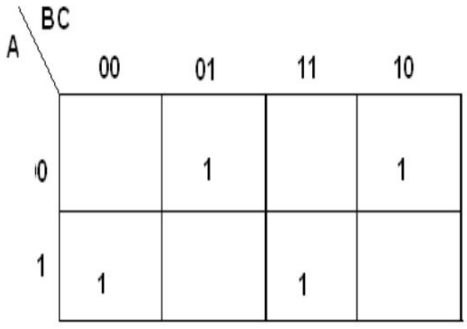
FULL ADDER USING TWO HALF ADDER



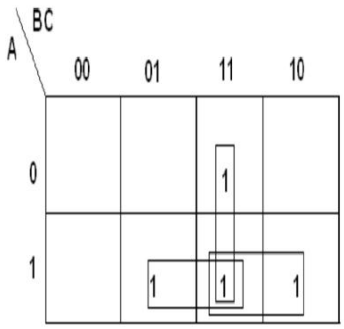
TRUTH TABLE:

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

K-Map for SUM:



K-Map for CARRY:

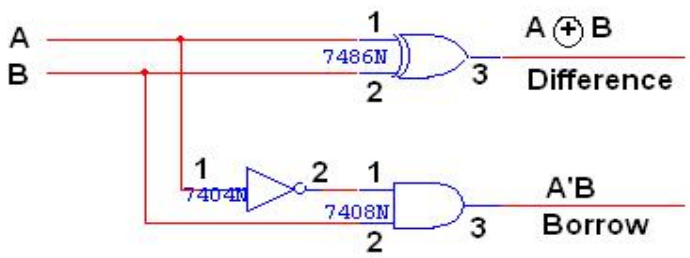


SUM = A'B'C + A'BC' + ABC' + ABC

CARRY = AB + BC + AC

LOGIC DIAGRAM:

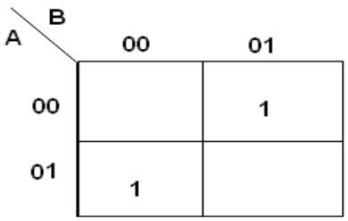
HALF SUBTRACTOR



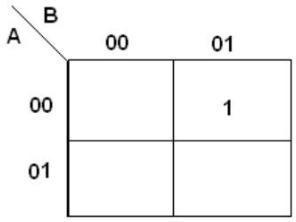
TRUTH TABLE:

A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

K-Map for DIFFERENCE:



K-Map for BORROW:

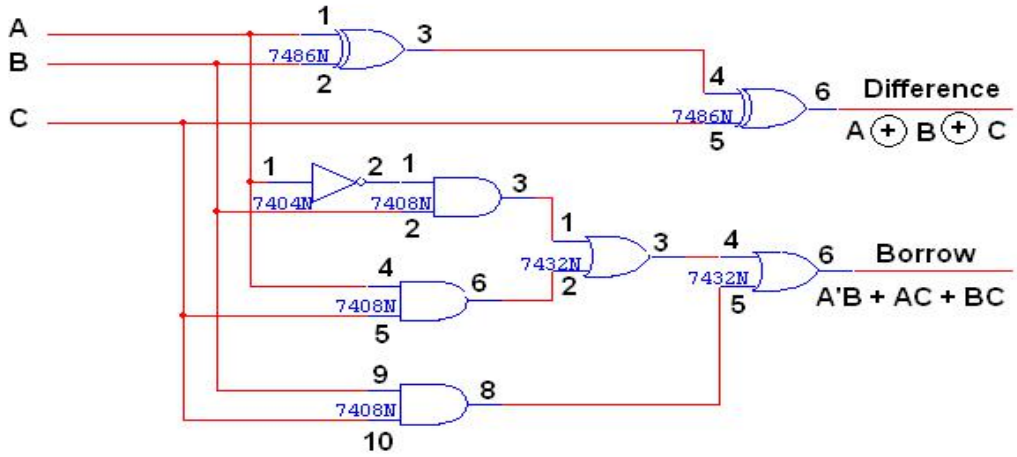


DIFFERENCE = A'B + AB'

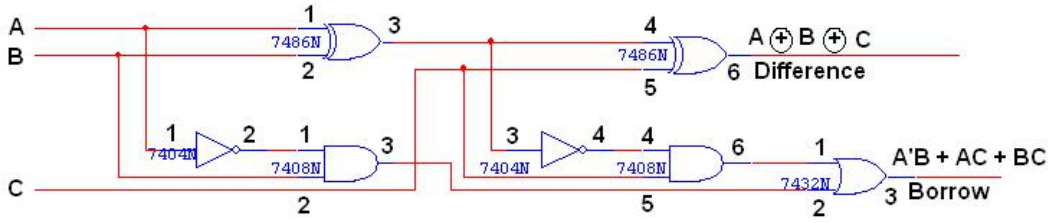
BORROW = A'B

LOGIC DIAGRAM:

FULL SUBTRACTOR



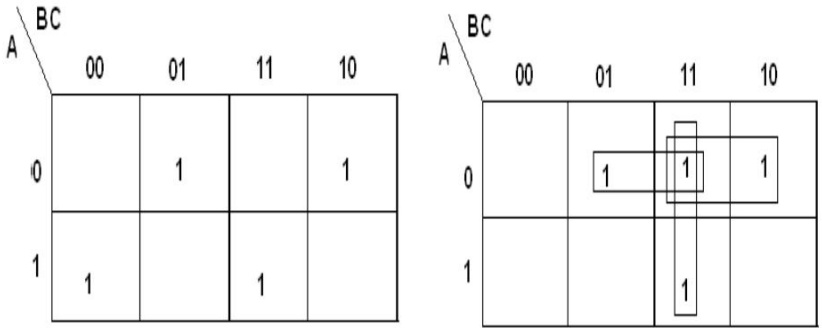
FULL SUBTRACTOR USING TWO HALF SUBTRACTOR:



TRUTH TABLE:

A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-Map for Borrow



Difference = A'B'C + A'BC' + AB'C' + ABC Borrow = A'B + BC + A'C

PROCEDURE:

(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

RESULT:

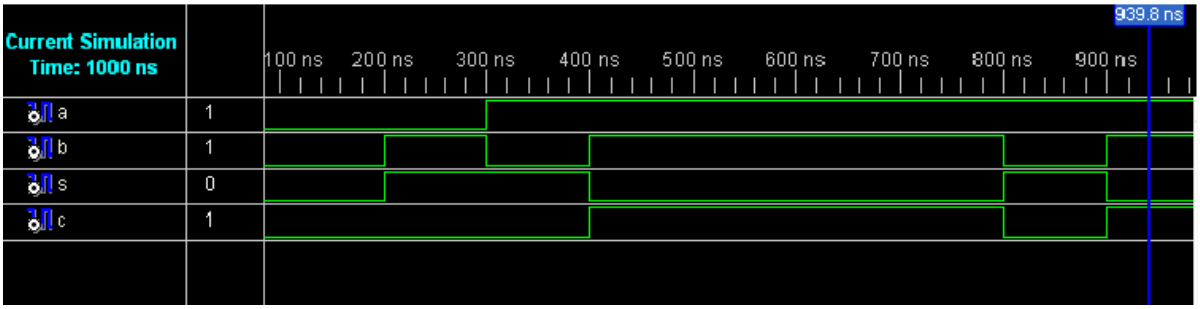
Truth table is verified.

EXPERIMENT NO.4 (b)

HALF ADDER :

```
entity half_adder is
     part (A,B: in std_logic);
     SUM,CARRY:out std_logic;
end half_adder;

architecture equation of half_addet is begin
     SUM <= A xor B;
     CARRY <= A and B;
end equation;
```

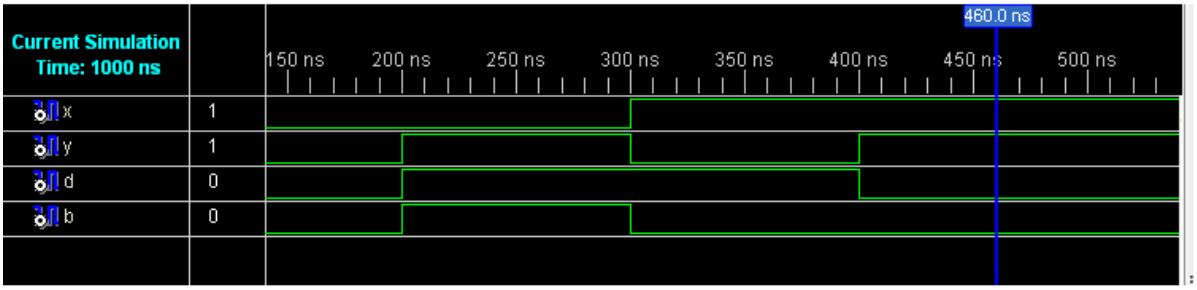
Half adder : Output



HALF SUBTRACTOR :

```
entity half_subtractor is
     part (A,B: in std_logic);
     Dout, Bout :out std_logic;
end half_ subtractor;

architecture equation of half_ subtractor is
begin
     Dout <= A xor B;
     Bout <= (not A) and B;
end equation;
```

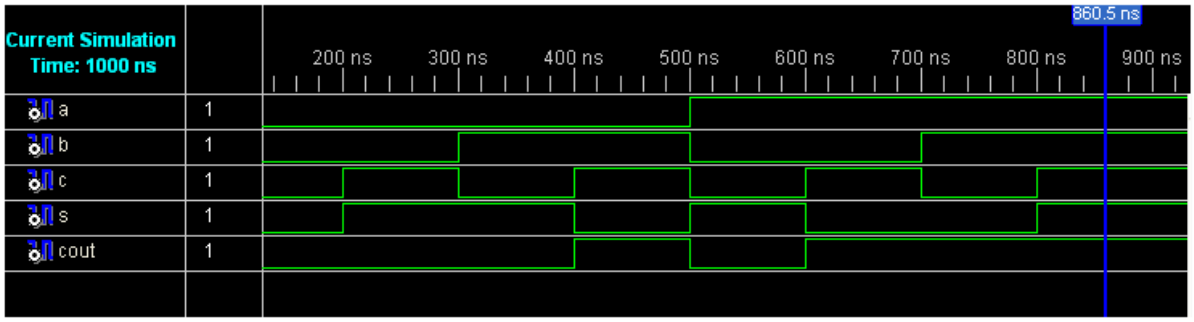
HALF SUBTRACTOR : OUTPUT



FULL ADDER:

```
entity full_adder is
     part (A,B,C: in std_logic);
     SUM,CARRY:out std_logic;
end full_adder;

architecture equation of full_addet is begin
     SUM <= A xor B xor C;
     CARRY <= (A and B) or (B and C) or (A and C);
end equation;
```

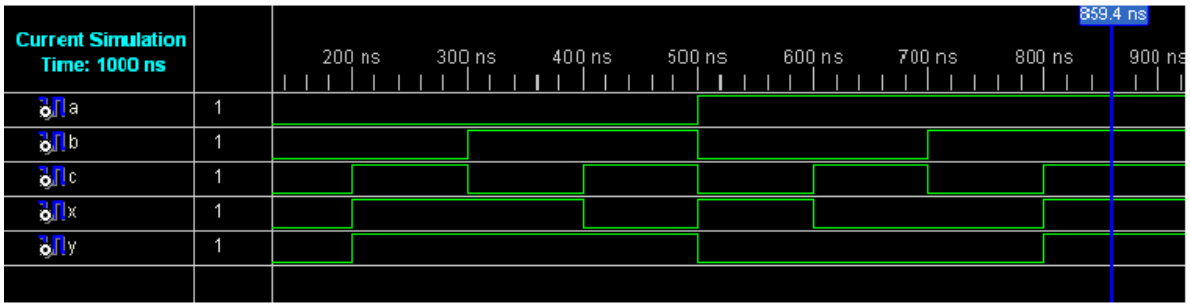
Full adder : Output



FULL SUBTRACTOR :

```
entity full_subtractor is
     part (A,B: in std_logic);
     Dout, Bout :out std_logic;
     end full_ subtractor;

architecture equation of full_ subtractor is
begin
     Dout <= A xor B xor C;
     Bout <= (not A) and B) or (not A) and C)
or (B and C);
end equation;
```

FULL SUBTRACTOR : OUTPUT



EXPERIMENT 5

REALISATION OF LOGIC EXPRESSION USING 8:1 MULTIPLEXER

AIM: To simplify a given 4-variable logic expression using Entered Variable Map and to realize the simplify logic expression using 8:1 multiplexer IC.

COMPONENTS REQUIRED:

Sl.No	Components	Quantity
1.	Trainer Kit	01
2.	Patch Chords	20
3.	IC 74151	01
4.	IC 7404	01

THEORY:

Multiplex means many into one. A multiplexer is a circuit with many inputs but only one output. The inputs of the multiplexer are divided into two categories namely, data inputs and select inputs. A multiplexer having 'n' data inputs have 'm' control signals such that $n \leq 2m$. Depending on the value of the select inputs, data on one of the 'n' inputs is steered to the output. The figure shows the block diagram of a multiplexer.

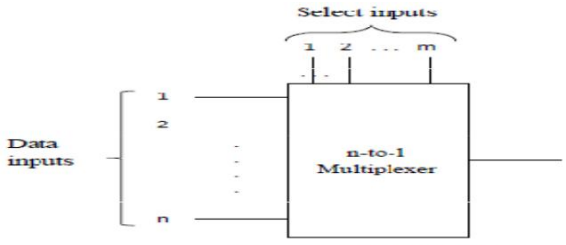
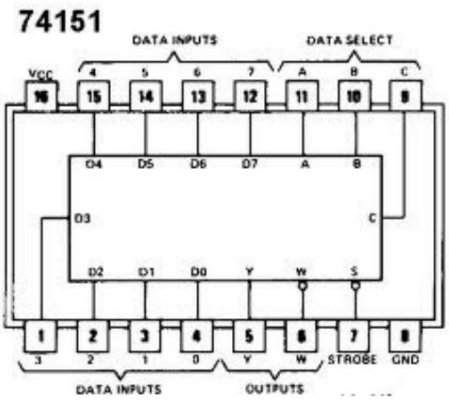


Fig 7a.n-to-1 Multiplexer

Multiplexer can be used to implement any logic expression. Commercial multiplexer ICs come in integer power of 2, e.g. 2-to-1, 4-to-1, 8-to-1, 16-to-1 multiplexers. Hence to implement a logic expression with 'n' variables, a multiplexer with 'n' select inputs is needed i.e. 2n – to-1 multiplexer. Hence it is called as universal logic circuit.

Theory:
Map Entered
Rules for enteri

Rule No.	M
1.	0 1
2	0 1
3.	0 1
4.	0 1
5.	0 1
6.	0 1
7.	0 1
8.	0 1
9.	0 1



PIN DIAGRAM

74151

DESIGN:

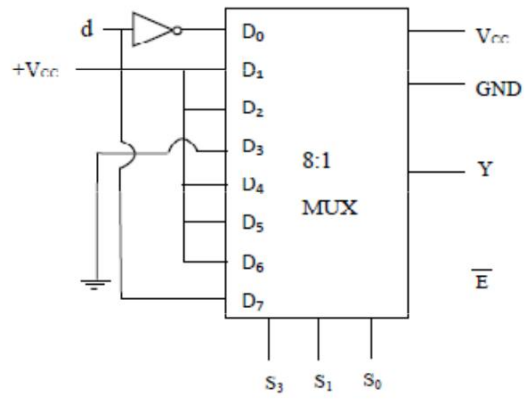
Consider the 4-variable logic expression

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

Example 1) Y= F (A, B,C,D) = Σ m(0, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 15)

This expression is equivalent to the following truth table.

CIRCUIT DIAGRAM:



To realize this expression using as 8-to-1 multiplexer:

Step 1: Any 3 variables are fed as select inputs. The fourth variable is then the data input. In the example, variables A, B and C are selected as the select inputs and D the data input.

Step 2:

i. Write all the combinations of 3 select inputs (row 1)

ii. For each value of the 4th variable D, write the corresponding output. (Row 2 & 3)

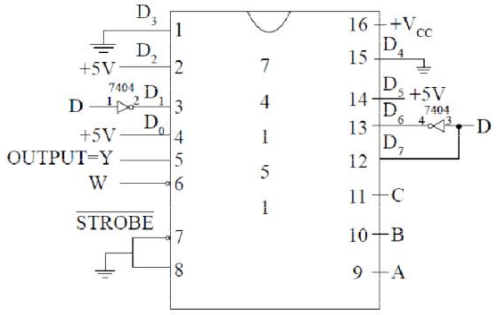
iii. Write Y as a function of d (row 4)
iv. Assign data input values for 8-to-1 MUX by copying Y values obtained in row 4.

Note: The data inputs of a 8:1 MUX are labelled D0 to D7

Example 2)

Simplify the function using MEV technique

f(A,B,C,D)=Σm (2,4,5,12,15) + d(0,1,10,11).



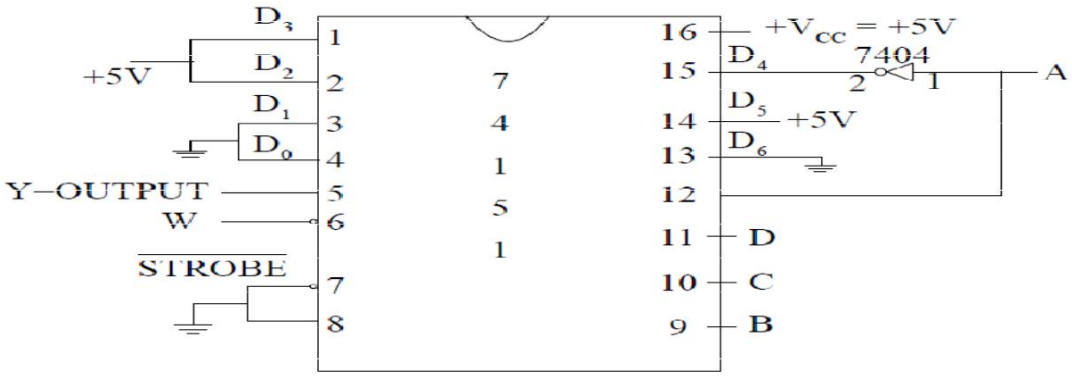
Method 2)
f (A,B,C,D) = Σm(2,3,4,5,13,15) + d(8,9,10,11).

	D_0	D_1	D_2
$\bar{A}$	0	1	(2)
A	8	9	10
	0	0	1=+V_{CC} =+5V

ABC	000	001	010	011	100	101	110	111
D=0	1	1	1	0	1	1	1	0
D=1	0	1	1	0	1	1	1	1
Y	D'	1	1	0	1	1	1	D
8-to-1 MUX data input	D_0=D'	D_1=1	D_2=1	D_3=0	D_4=1	D_5=1	D_6=1	D_7=D

Rules to obtain implementation table:
If both A' and A are not marked (ie. circled), then the output of D0=0.
If both A' and A are marked then the output of D1=1.

If 2 is marked and 10 is not circled , then take the value of that row which is circled. In this example, 2 is circled consider its corresponding row =A' as value of D2. D2 = A'
11 is marked and 3 is not marked. Hence take 11's value = A . D3 =A.



PROCEDURE:

1. Simplify the given logic expression using Map entered variable map.
2. Check all the IC components using digital IC tester.
3. Make connections as per the circuit diagram.
4. Give supply to the trainer kit.
5. Provide input data to the circuit via switches
6. Verify the truth table sequence. Observe the outputs.

RESULT:

Truth table is verified.

Experiment No: 5 (b)

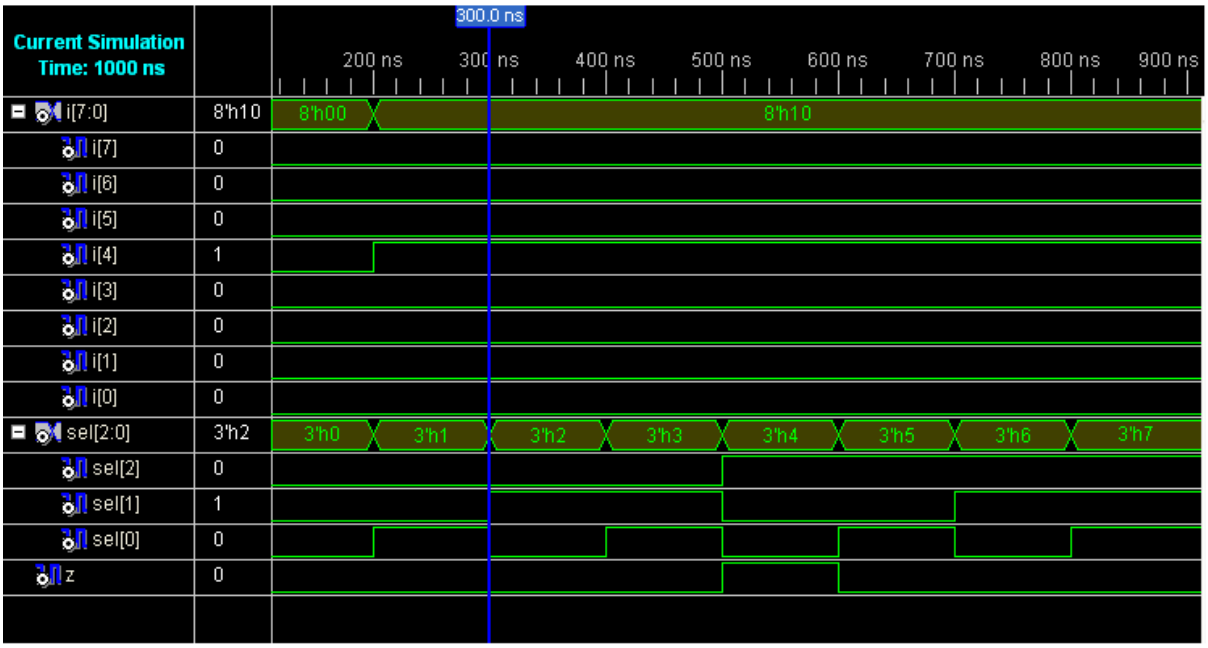
Design and develop the verilog / VHDL code for an 8:1 multiplexer. Simulate and verify its working.

VHDL Code:

```
entity mux8to1 is
    Port (sel : in  STD_LOGIC_VECTOR (2 downto 0);
          I : in  STD_LOGIC_ VECTOR (7 downto O);
          Zout : out  STD_LOGIC);
end mux8to1;

architecture Behavioral of mux8to1 is

begin
     Zout <=I(0) when  sel ="000" else
            I(1) when  sel ="001" else
            I(2) when  sel ="010" else
            I(3) when  sel ="011" else
            I(4) when  sel ="100" else
            I(5) when  sel ="101" else
            I(6) when  sel ="110" else
            I(7);

end Behavioral;
```

8:1 MUX : Output

8:1 Mux : 0utput



EXPERIMENT NO 8
J-K MASTER/SLAVE FF USING NAND GATES

AIM: To realize a J-K Master/Slave flip flop using NAND gates and verify its truth table.

COMPONENTS USED:

Sl.No	Components	Quantity
1.	Trainer kit.	01
2.	Patch chords	30
3.	IC 74LS00	01
4.	IC 74LS10	02

THEORY:

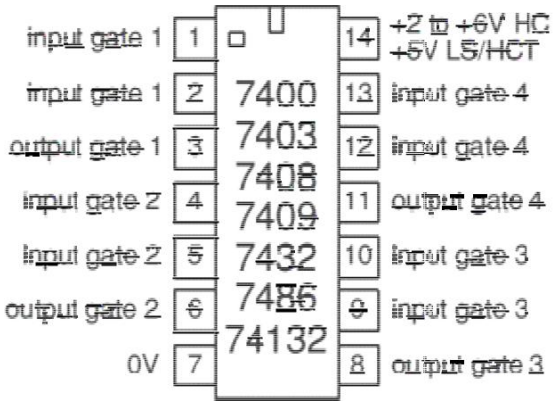
The Q and Q' outputs will only change state on the falling edge of the CLK signal, and the J and K inputs will control the future output

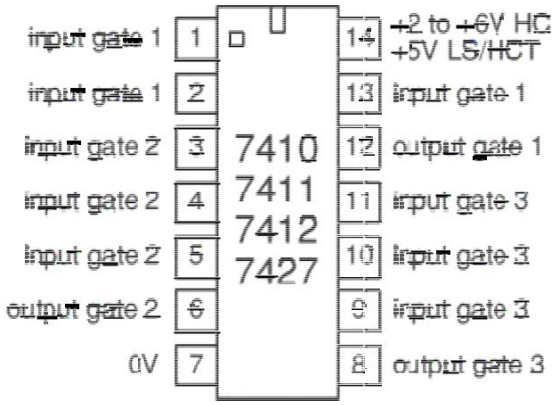
If both the J and K inputs are held at logic 1 and the CLK signal continues to change, the Q and Q' outputs will simply change state with each falling edge of the CLK signal. (The master latch circuit will change state with each *rising* edge of CLK.)

Master Slave Flip Flop:

A JK master flip flop is positive edge triggered, whereas slave is negative edge triggered. Therefore master first responds to J and K inputs and then slave. If J=0 and K=1, master resets on arrival of positive clock edge. High output of the master drives the K input of the slave. For the trailing edge of the clock pulse the slave is forced to reset. If both the inputs are high, it changes the state or toggles on the arrival of the positive clock edge and the slave toggles on the negative clock edge. The slave does exactly what the master does.

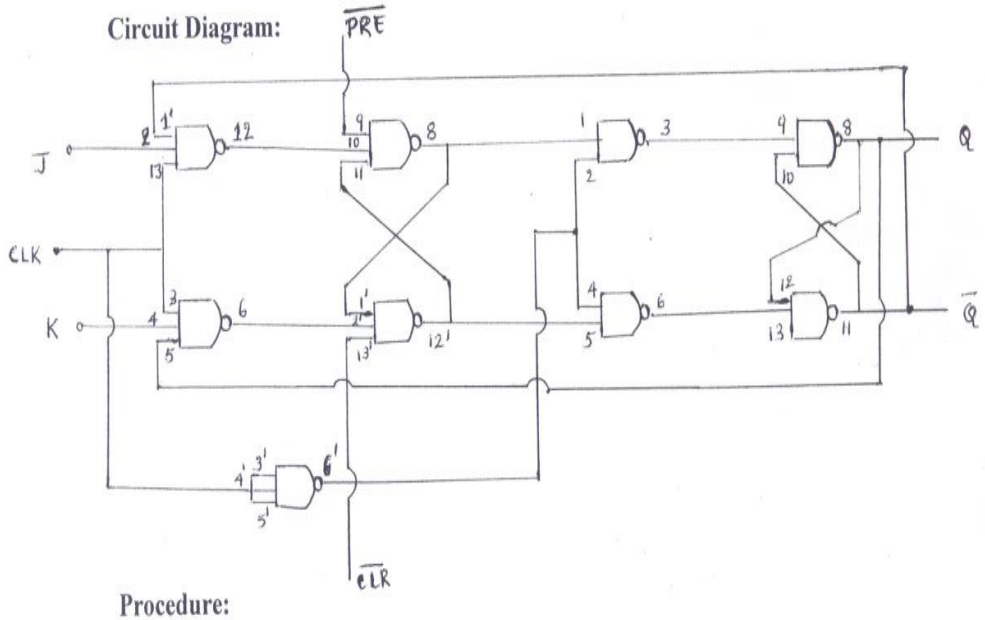
Pin Details of the ICs:





Clk	J	K	Q	--- Q	Comment
⎍⎍	0	0	Q_0	---- Q_0	No change
⎍⎍	0	1	0	1	Reset
⎍⎍	1	0	1	0	Set
⎍⎍	1	1	Q_0	Q_0	toggle

Circuit Diagram:



Procedure:

(1) Verify all components and patch chords whether they are in good condition or not.
(2) Make connection as shown in the circuit diagram.
(3) Give supply to the trainer kit.
(4) Provide input data to circuit via switches.
(5) Verify truth table sequence and observe outputs.

RESULT:

Truth table is verified.

Experiment No. 8 (b)

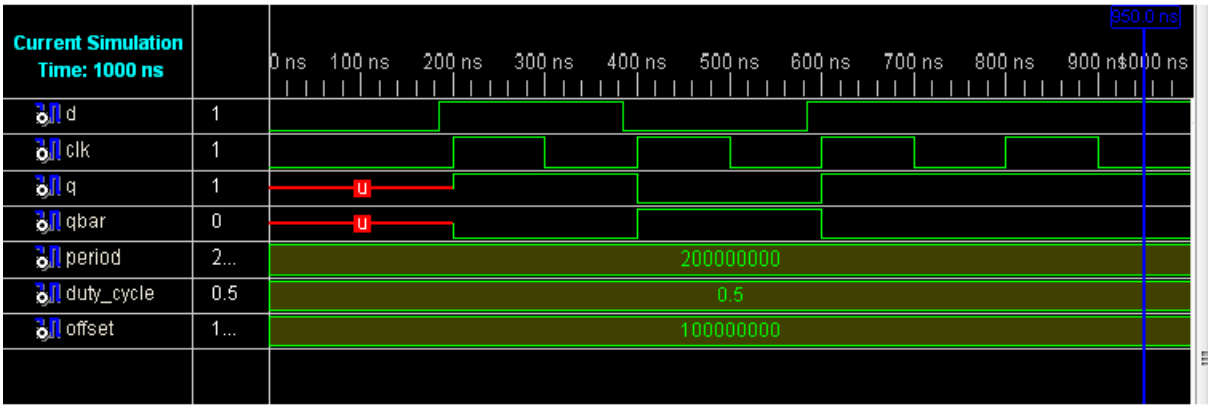
Design and develop the verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.

VHDL Code:

```
entity ddflipflop is
    Port (clk, D : in  STD_LOGIC;
          Q : inout  STD_LOGIC);
          Qbar : out  STD_LOGIC);
end ddflipflop;

architecture Behavioral of ddflipflop is

begin
process(clk)
  begin
     if(rising_edge (clk)) then
     Q<=D;
     end if;
  end process;
Qbar<=not Q;
end Behavioral;
```

D Flip flop : Output

D Flip flop : Output



EXPERIMENT NO.: 7

DESIGN AND IMPLEMENTATION OF CODE CONVERTOR

AIM:

To design and implement 4-bit

(i) Binary to Gray code converter

(ii) Gray to binary code converter

COMPONENTS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
4.	NOT GATE	IC 7404	1
5.	IC TRAINER KIT	-	1
6.	PATCH CORDS	-	35

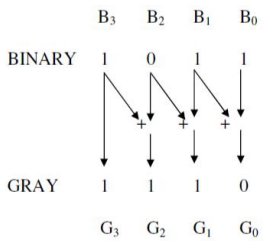
THEORY:

A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables. A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps.

The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.

BINARY TO GRAY CODE CONVERSION

Steps:The example shows the steps involved in conversion of a binary code to its gray code.

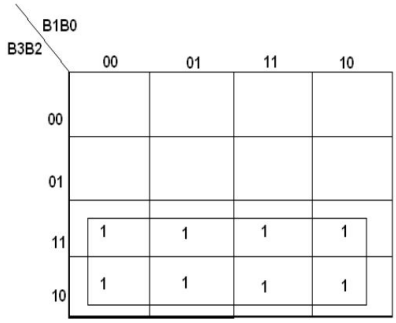


Binary code taken for the example is 1011.In the conversion process the most significant bit (MSB) of the binary code is taken as the MSB of the Gray code. The bit positions G2, G1 and G0 is obtained by adding (B3, B2),(B2, B1) and (B1, B0) respectively, ignoring the carry generated. From the K-Map simplification

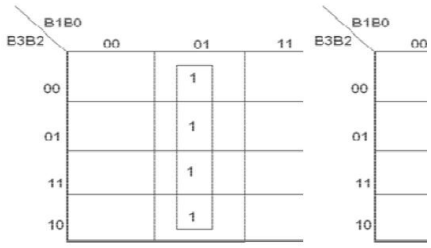
TRUTH TABLE:

Binary input				Gray code output			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

K-Map for G_3:

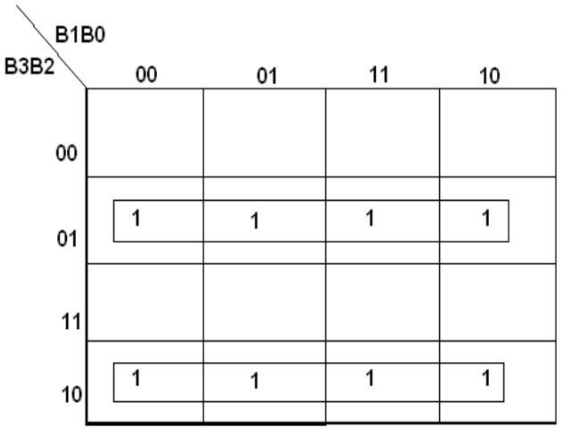


K-Map for G_2:

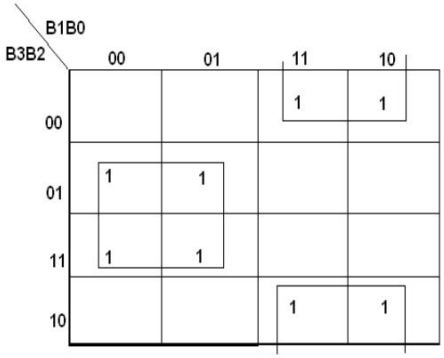


$G2 = B3 \oplus B2$

K-Map for G_1:



K-Map for G_0:



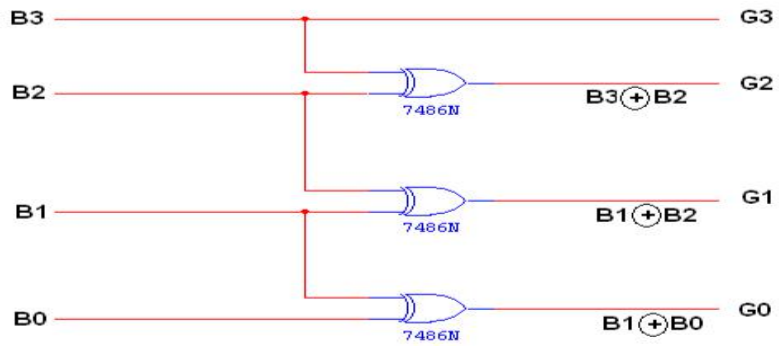
G0 = B1⊕B0 G1 = B1⊕B2

For binary to Gray code convers ion the following Boolean expressions are obtained,

$$G_3 = B3$$
$$G_2 = B_3 \oplus B_2$$
$$G_1 = B2 \oplus B_1$$
$$G_0 = B1 \oplus B_0$$

LOGIC DIAGRAM:

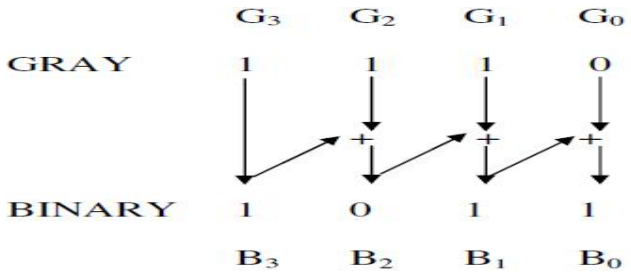
BINARY TO GRAY CODE C ONVERTOR



Gray to Binary Code Conversion Steps:

The example shows the steps inv olved in conversion of a Gray code to binary co de.

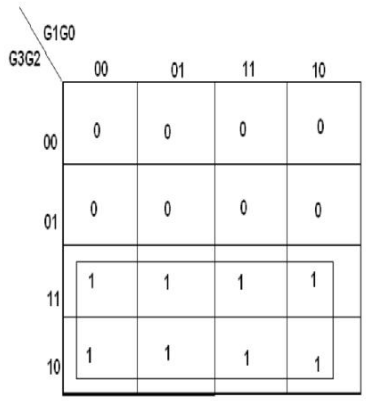
Gray code taken for the example is 1110



TRUTH TABLE:

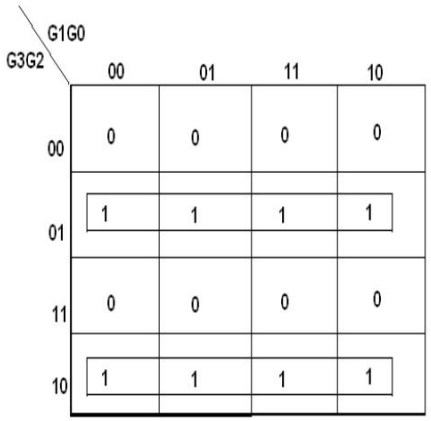
Gray code input				Binary output			
G3	**G2**	**G1**	**G0**	**B3**	**B2**	**B1**	**B0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**1**	**0**	**0**	**0**	**1**
0	**0**	**1**	**1**	**0**	**0**	**1**	**0**
0	**0**	**1**	**0**	**0**	**0**	**1**	**1**
0	**1**	**1**	**0**	**0**	**1**	**0**	**0**
0	**1**	**1**	**1**	**0**	**1**	**0**	**1**
0	**1**	**0**	**1**	**0**	**1**	**1**	**0**
0	**1**	**0**	**0**	**0**	**1**	**1**	**1**
1	**1**	**0**	**0**	**1**	**0**	**0**	**0**
1	**1**	**0**	**1**	**1**	**0**	**0**	**1**
1	**1**	**1**	**1**	**1**	**0**	**1**	**0**
1	**1**	**1**	**0**	**1**	**0**	**1**	**1**
1	**0**	**1**	**0**	**1**	**1**	**0**	**0**
1	**0**	**1**	**1**	**1**	**1**	**0**	**1**
1	**0**	**0**	**1**	**1**	**1**	**1**	**0**
1	**0**	**0**	**0**	**1**	**1**	**1**	**1**

K-Map for B3:



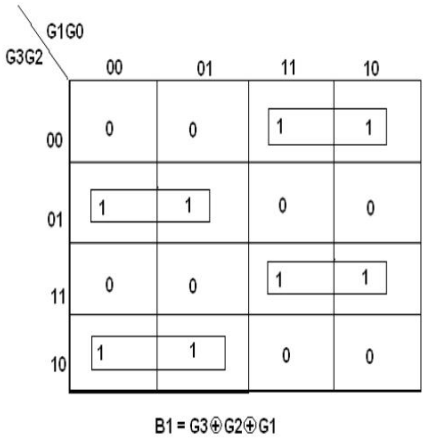
B3 = G3

K-Map for B2:



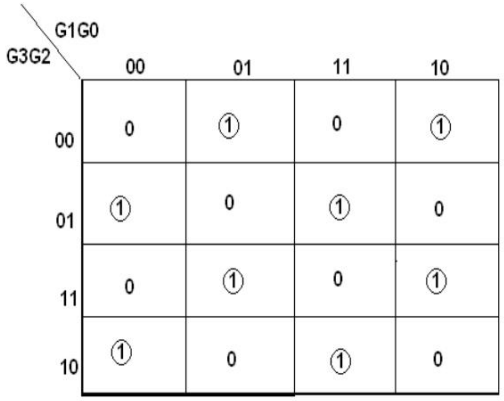
B2 = G3⊕G2

K-Map for B_1:



B1 = G3⊕G2⊕G1

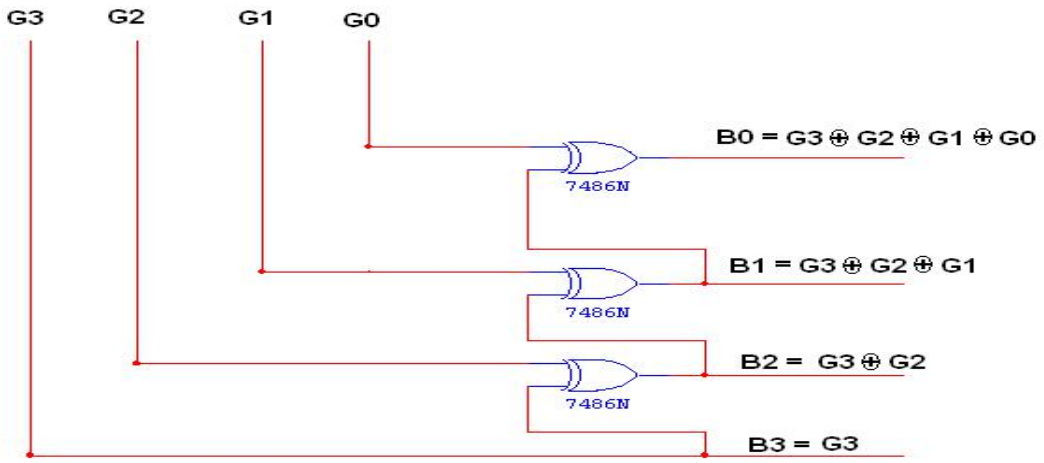
K-Map for B_0:



B0 = G3⊕G2⊕G1⊕G0

LOGIC DIAGRAM:

GRAY CODE TO BINARY CONVERTOR



In the conversion process the most significant bit (MSB) of the Gray code is taken as the MSB of the binary code. The bit positions B2, B1 and B0 is obtained by adding (B3, G2), (B2, G1) and (B1, G0) respectively, ignoring the carry generated. From the K-Map simplification for Gray code to binary code conversion the following Boolean expressions are obtained,

$$B3 = G_3$$
$$B_2 = G_3 \oplus G_2$$
$$B_1 = G_3 \oplus G_2 \oplus G_1$$
$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

PROCEDURE:

(i) Connections were given as per circuit diagram.

(ii) Logical inputs were given as per truth table

(iii) Observe the logical output and verify with the truth tables.

RESULT:

Binary to Gray and Gray to Binary converters are designed, constructed using logic gates and their truth table was verified.

EXPERIMENT NO: 8
SYNCHRONOUS UP COUNTER

AIM: Design and implement a mod n (n<8) synchronous up counter using JK FF IC's and demonstrate its working.

COMPONENTS USED: IC 74LS76, IC 74LS08, Patch chords, power chords, and Trainer kit.

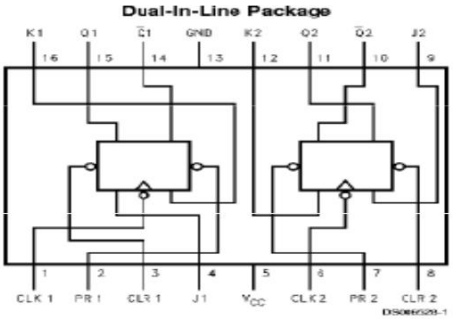
THEORY

In digital logic and computing, a counter is a device which stores and displays the number of times a particular event or process has occurred, often in relationship to a clock signal.

A synchronous counter is one whose output bits change state simultaneously. Such a counter circuit can be built from JK flip-flop by connecting all the clock inputs together, so that each and every flip-flop receives the exact same clock pulse at the exact same time. This results in all the individual output bits changing state at exactly the same time in response to the common clock signal with no ripple effect i.e. with no propagation delay.

By examining the four-bit binary count sequence, it noticed that just before a bit toggles, all preceding bits are "high". That is a synchronous up-counter can be implemented by toggling the bit when all of the less significant bits are at a logic high state. For example, bit 1 toggles when bit 0 is logic high; bit 2 toggles when both bit 1 and bit 0 are logic high; bit 3 toggles when bit 2, bit 1 and bit 0 are all high; and so on.

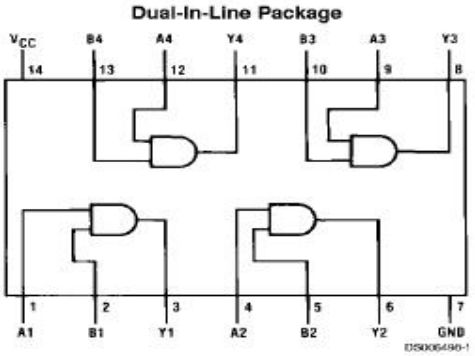
Pin diagram of 7476



IC 7476 contains 2 JK flip-flops with pre-set and clear signals.

PR	CLR	CLK	J	K	Q	Q’
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
H	H	H	L	L	Q0	Q0’
H	H		H	L	H	L
H	H		L	H	L	H
H	H		H	H	Toggle	

IC: 7408



Transition Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

DESIGN FOR MOD 8 UP COUNTER:

Present State			Next state			Flip flop inputs					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	K_C	J_C	K_B	J_B	K_A	J_A
0	0	0	0	0	1	X	0	X	0	X	1
0	0	1	0	1	0	X	0	X	1	1	X
0	1	0	0	1	1	X	0	0	X	X	1
0	1	1	1	0	0	X	1	1	X	1	X
1	0	0	1	0	1	0	X	X	0	X	1
1	0	1	1	1	0	0	X	X	1	1	X
1	1	0	1	1	1	0	X	0	X	X	1
1	1	1	0	0	0	1	X	1	X	1	X

Design:

JA

QC \ QB QA	00	01	11	10
0	1	X	X	1
1	1	X	X	1

JA = 1

KA

QC \ QB QA	00	01	11	10
0	X	1	1	X
1	X	1	1	X

KA = 1

JB

QC \ QB QA	00	01	11	10
0	0	1	X	X
1	0	1	X	X

JB = QA

KB

QC \ QB QA	00	01	11	10
0	X	X	1	0
1	X	X	1	0

KB = QA

JC

QC \ QB QA	00	01	11	10
0	0	X	1	0
1	X	X	X	X

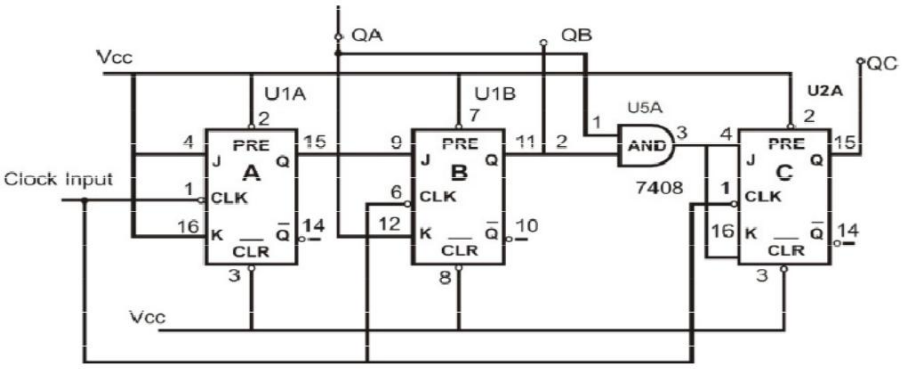
JC = QA QB

KC

QC \ QB QA	00	01	11	10
0	X	X	X	X
1	0	0	1	0

KC = QA QB

Circuit diagram of Mod – 8 counter:

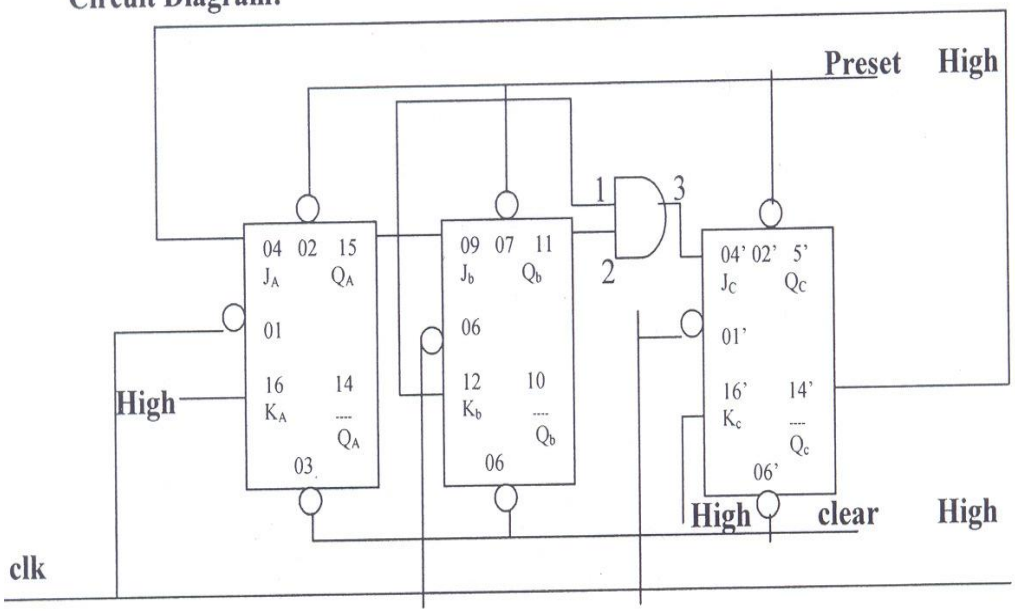


Mod-5 Counter Synchronous Counter: This have five counter states. The counter design table for such counter shows the three flip-flop and their states also (0 to 4 states). 6 inputs needed for the three flip-flops.

Present State	Next State	JcKc	Jb Kb	JaKa
Qc QbQa	Qc+1 Qb+1 Qa+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	0 0 0	x 1	0 x	0 x
1 0 1	x xx	x x	x x	x x
1 1 0	x xx	x x	x x	x x
1 1 1	x xx	x x	x x	x x

Jc = QbQa, Kc=1, Jb=Qa,Kb=Qa, Ja=Qc' ,Ka=1

Circuit Diagram:



PROCEDURE:

1. Verify all the components and patch cords for good working condition.

2. Make connection as shown in the circuit diagram.

3. Give supply to the trainer kit

4. Provide input data to circuit via switches and verify the truth table.

RESULT:

Truth table is verified.

EXPERIMENT NO 9

ASYNCHRONOUS COUNTER USING DECADE COUNTER IC

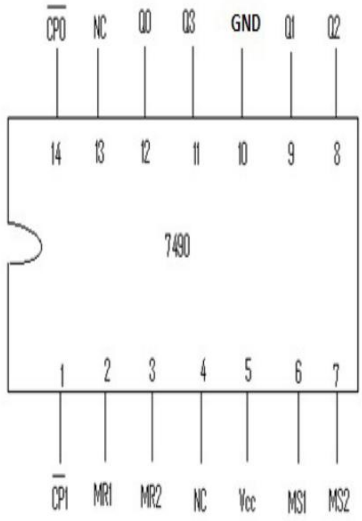
AIM: Design and implement asynchronous counter using decade counter IC to count up from 0 to n (n≤9) and demonstrate on 7-segment display.

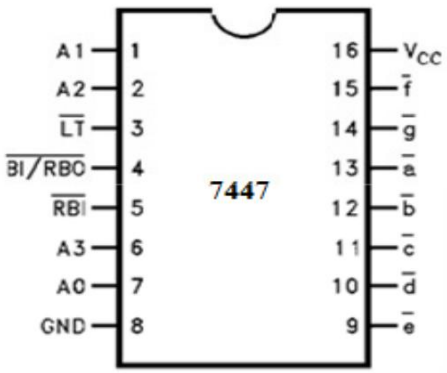
COMPONENTS USED: IC 74LS90, Patch chords, Power chords and Trainer kit.

THEORY:

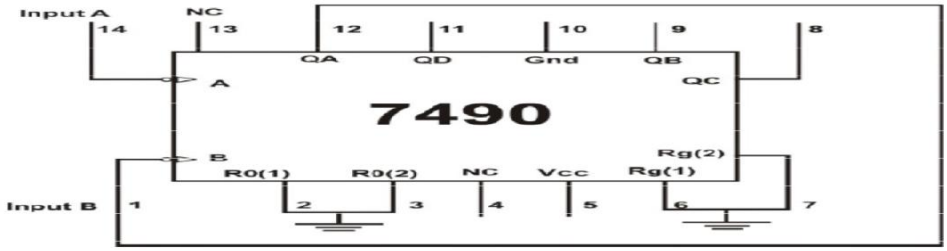
Asynchronous counter is a counter in which the clock signal is connected to the clock input of only first stage flip flop. The clock input of the second stage flip flop is triggered by the output of the first stage flip flop and so on. This introduces an inherent propagation delay time through a flip flop. A transition of input clock pulse and a transition of the output of a flip flop can never occur exactly at the same time. Therefore, the two flip flops are never simultaneously triggered, which results in asynchronous counter operation.

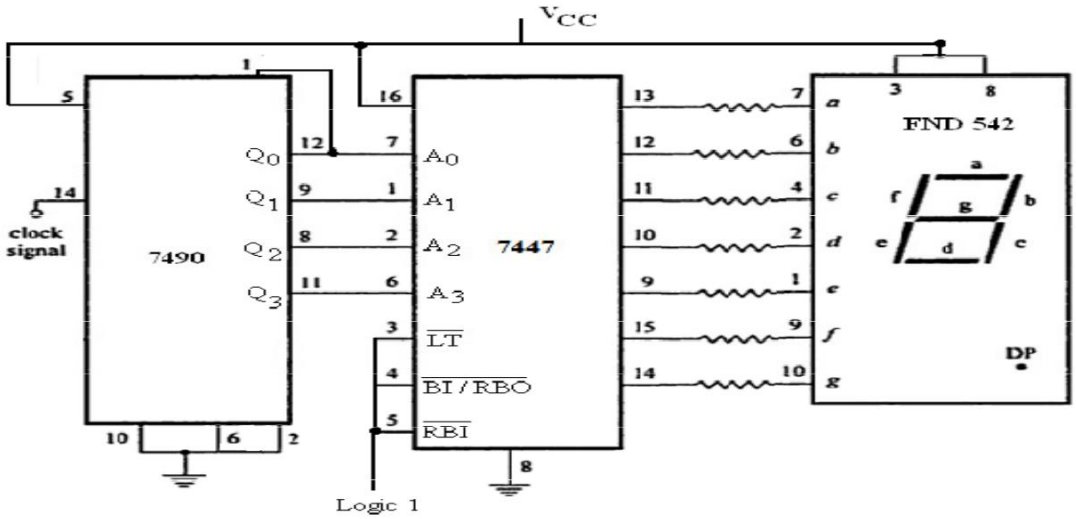
Pin Diagram





Circuit Diagram:





Function Table:

Clock	Q_a	Q_b	Q_c	Q_d
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
8	0	1	1	1
9	1	0	0	0

RESULT:

Truth table is verified.

VIVA QUESTIONS ON ANALOG

1. Draw the basic structure of an N channel junction field effect transistor.
2. Why is FET known as a unipolar device?
3. What are the advantages and disadvantages of JFET over BJT?
4. What is a channel?
5. Distinguish between JFET and MOSFET.
6. What is an effect of cascading?
7. What are all the factors affecting the bandwidth of the RC Coupled amplifier?
8. Explain bypass capacitor?
9. What is meant by coupling capacitor?
10. Why does amplifier gain reduce?
11. Explain the different regions in frequency response?
12. State the types of distortions in amplifier?
13. What is cross over distortion? How it can be eliminated?
14. Define noise?
15. Draw the symbol of JFET and MOSFET.
16. What are the two modes of MOSFET?
17. Define pinch-off voltage
18. What is feedback and what are feedback amplifiers?
19. What is meant by positive and negative feedback?
20. What are the advantages and disadvantages of negative feedback?
21. Differentiate between voltage and current feedback in amplifiers?
22. What is the type of feedback used in an op- amp Schmitt trigger?
23. Give the expression for the frequency of oscillations in an op-amp sine wave oscillator?
24. What are the conditions for sustained oscillations or or what is Barkhausen criterion
25. What are the classifications of Oscillators?
26. What are the types of feedback oscillators?
27. Define Piezo-electric effect?
28. Draw the equivalent circuit of crystal oscillator?
29. How does an oscillator differ from an amplifier?

VIVA QUESTIONS ON LOGIC DESIGN

1. Why NAND & NOR gates are called universal gates?
2. Realize the EX – OR gates using minimum number of NAND gates.
3. Give the truth table for EX-NOR and realize using NAND gates?
4. Compare TTL logic family with CMOS family?
5. Which logic family is fastest and which has low power dissipation?
6. What are the different methods to obtain minimal expression?
7. What is a Min term and Max term
8. State the difference between SOP and POS.
9. What is meant by canonical representation?
10. What is K-map? Why is it used?
11. What is a multiplexer?
12. What is a de-multiplexer?
13. What are the applications of multiplexer and de-multiplexer?
14. Derive the Boolean expression for multiplexer and de-multiplexer.
15. In a 2n to 1 multiplexer how many selection lines are there?
16. Implement an 8:1 mux using 4:1 mux?
17. What is a comparator?
18. What are the applications of comparator?
19. Derive the Boolean expressions of one bit comparator and two bit comparators.
20. How do you realize a higher magnitude comparator using lower bit comparator?
21. Design a 2 bit comparator using a single Logic gates?
22. Design an 8 bit comparator using a two numbers of IC 7485?
23. What are the applications of decoder?
24. What is the difference between decoder & encoder?
25. For n- 2n decoder how many i/p lines & how many o/p lines?
26. Using 3:8 decoder and associated logic, implement a full adder
27. What is the difference between decoder and de-mux?
28. What are the different types of LEDs?
29. What are the applications of LEDs?
30. What is a priority encoder?
31. What is the difference between Flip-Flop & latch?
32. Give examples for synchronous & asynchronous inputs?
33. What are the applications of different Flip-Flops?
34. What is the advantage of Edge triggering over level triggering?
35. What is the relation between propagation delay & clock frequency of flip-flop?
36. What is race around in flip-flop & how to overcome it?

37. Convert the J K Flip-Flop into D flip-flop and T flip-flop
38. List the functions of asynchronous inputs?
39. What is the necessity for sequence generation?
40. What are PISO, SIPO, and SISO with respect to shift register?
41. Differentiate between serial data & parallel data
42. What is the significance of Mode control bit?
43. What is a ring counter?
44. What is a Johnson counter?
45. How many Flip-flops are present in IC 7495?
46. What is an asynchronous counter?
47. How is it different from a synchronous counter?
48. Realize asynchronous counter using T flip-flop
49. What are synchronous counters?
50. What are the advantages of synchronous counters?
51. What is an excitation table?
52. Write the excitation table for D, T FF
53. Design mod-5 synchronous counter using T FF
54. What is a presettable counter?
55. What are the applications of presettable counters?
56. Write the circuit for preset value of 0100 and N=5 (up counter)
57. What is a decade counter?
58. What do you mean by a ripple counter?