

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
B.E - III SEMESTER

ANALOG AND DIGITAL ELECTRONICS LABORATORY
MANUAL

[AS PER CHOICE BASED CREDIT SYSTEM (CBCS)
SCHEME]

SUBJECT CODE: (18CSL37)

VERSION 1.0

Prepared By:

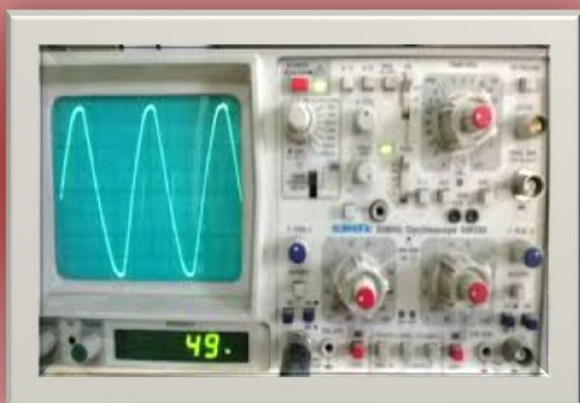
Mr. Chetan Balaji
Assistant Professor
Dept of CSE

*Reviewed and
Approved By:*

Dr. Shantala C P
Vice Principal &
Head
Dept of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR : 2018-19



CATHODE RAY OSCILLOSCOPE



DIGITAL IC TRAINER KIT

*Try to get whatever
U like, else u will be
forced to like whatever
U get.*

QMP 7.1 D/F



Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)
(**NAAC Accredited & ISO 9001:2015 Certified Institution**)
NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.



Department of Computer Science & Engineering

ANALOG AND DIGITAL ELECTRONICS LABORATORY WORK BOOK

[As per Choice Based Credit System (CBCS) scheme]

Subject Code: 18CSL37

B.E - III Semester

Academic Year: 2019-20

Name : _____

USN : _____

Batch : _____ Section : _____



Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)
(NAAC Accredited & ISO 9001:2015 Certified Institution)
NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.

QMP 7.1 D/F



SYLLABUS

Laboratory Code: 18CSL37

CIE Marks 40

Number of Contact Hours/Week 0:2:2

SEE Marks 60

Total Number of Lecture Hours 36

Exam Hours 03

CREDITS – 02

Course Learning Objectives: This course (18CSL37) will enable students to:

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):

- Simulation packages preferred: Multisim, Modelsim, PSpice or any other relevant.
- For Part A (Analog Electronic Circuits) students must trace the wave form on Tracing sheet / Graph sheet and label trace.
- Continuous evaluation by the faculty must be carried by including performance of a student in both hardware implementation and simulation (if any) for the given circuit.
- A batch not exceeding 4 must be formed for conducting the experiment. For simulation individual student must execute the program.

Laboratory Session-1:

Discuss-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description.

Laboratory Session-2:

Discuss-upon Logic design components, pin diagram (if any), Timing diagrams, etc.

Laboratory Programs:

PART A (Analog Electronic Circuits)

1. Design an astable multivibrator circuit for three cases of duty cycle (50%, <50% and >50%) using NE 555 timer IC. Simulate the same for any one duty cycle.
2. Using ua 741 Opamp, design a 1 kHz Relaxation Oscillator with 50% duty cycle. And simulate the same.
3. Using ua 741 opamp, design a window comparator for any given UTP and LTP. And simulate the same.

PART B (Digital Electronic Circuits)

4. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates. And implement the same in HDL.
5. 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.
6. Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table. And implement the same in HDL.
7. Design and implement code converter I) Binary to Gray (II) Gray to Binary Code using basic gates.
8. Design and implement a mod-n ($n < 8$) synchronous up counter using J-K Flip-Flop ICs and Demonstrate its working.
9. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ($n \leq 9$) and demonstrate on 7-segment display (using IC-7447)

Laboratory Outcomes: The student should be able to:

Use appropriate design equations / methods to design the given circuit.

- Examine and verify the design of both analog and digital circuits using simulators.
- Make use of electronic components, ICs, instruments and tools for design and testing of circuits for the given the appropriate inputs.
- Compile a laboratory journal which includes; aim, tool/instruments/software/components used, design equations used and designs, schematics, program listing, procedure followed, relevant theory, results as graphs and tables, interpreting and concluding the findings.

Conduct of Practical Examination:

- Experiment distribution
 - ✓ For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

For laboratories having PART A and PART B:

Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (Course to change in accordance with university regulations)
 - a) For laboratories having only one part – Procedure + Execution + Viva-Voce: $15+70+15 = 100$ Marks
 - b) For laboratories having PART A and PART B
 - i. Part A – Procedure + Execution + Viva = $6 + 28 + 6 = 40$ Marks
 - ii. Part B – Procedure + Execution + Viva = $9 + 42 + 9 = 60$ Marks

INDEX PAGE:

Sl.No	Name of the Experiment	Date			Manual Marks (Max Marks)	Record Marks	Signature (Student)	Signature (Faculty)
		Conduction	Repetition	Submission of Record				
1.	Laboratory Session 1							
2.	Laboratory Session 2							
3.	Astable Multivibrator							
4.	Relaxation Oscillator							
5.	Window Comparator							
6.	Adders and Subtractor							
7.	JK Flip Flop Using Nand gates							
8.	Code Converters							
9.	Parity Generator/Checker							
10.	Synchronous Counter							
11.	Asynchronous Counter							
Average								

Note:

If the student fails to attend the regular lab, the experiment has to be completed in the same week. Then the manual/observation and record will be evaluated for 50% of maximum marks.

Course Learning objectives:

- ❖ This laboratory course enable 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 Converter

Laboratory / Course outcomes:

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

Use appropriate design equations / methods to design the given circuit.

- Examine and verify the design of both analog and digital circuits using simulators.
- Make use of electronic components, ICs, instruments and tools for design and testing of circuits for the given the appropriate inputs.
- Compile a laboratory journal which includes; aim, tool/instruments/software/components used, design equations used and designs, schematics, program listing, procedure followed, relevant theory, results as graphs and tables, interpreting and concluding the findings.

General Instructions to Students:

1. Understand the theoretical concept for the respective experiment.
2. Draw the circuit diagram in the given space in the observation book.
3. Tabulate the readings in the observation book and plot the graphs if necessary.
4. Every Student must at least construct one circuit.
5. After the completion of the experiment, get signature in the observation book.
6. Before coming to next lab, Make sure that records will be updated and signed from the concerned faculty.

Guide Lines for Writing Manual:

Blank Page (USE PENCIL)	Ruled Page (USE PEN)
Practical No:	Practical No:
Aim:	Aim:
Circuit Diagram	Apparatus Required
Characteristics/Wave form/ Observation Table	Theory:
Result	Procedure:
	Calculation:
	Result:



Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)
(NAAC Accredited & ISO 9001:2015 Certified Institution)
 NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.



CONTENT SHEET

Sl. No	Name of the Experiment	Page NO.
1.	Introduction to Analog Electronics	1-5
2.	Introduction to Digital Electronics	6-12
3.	Design and implement an Astable multivibrator circuit using 555 timer	13-16
4.	Design and construct a rectangular waveform generator	17-18
5.	Using ua 741 opamp, design a window comparator for any given UTP and LTP. And simulate the same.	19-22
6.	Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor	23-30
7.	Simplify the given 4 variable logic expression using Entered Variable Map	31-34
8.	Realize a J-K Master / Slave Flip-Flop using NAND gates	35-38
9.	Design and implementation of Binary to Gray and Gray to Binary Code	39-44
10.	Design and implement a mod-n ($n < 8$) synchronous up counter using J-K Flip-Flop ICs	45-48
11.	Design and implement an asynchronous counter using decade counter IC	49-53
12.	Question Bank	54-55
13.	Reference Books	56
14.	Mini Project	57-59

Laboratory Session-1: Introduction to Analog Electronic components and Multi Sim.

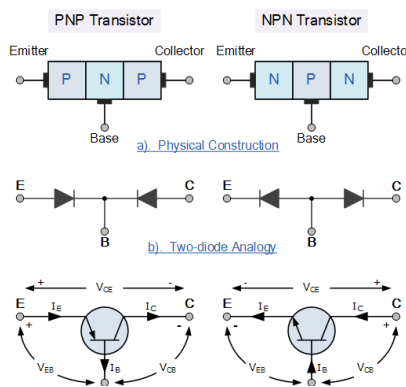
Analog Electronics is one of the fundamental courses found in all Electrical Engineering and most science programs. Analog circuit's process signals with continuous variation of voltage.

The different Components that are normally used in Analog Electronics are:

1. Bi polar Junction Transistors
2. MOSFET's
3. OP-AMP

A **transistor** is a semiconductor device used to amplify or switch electronic signals and electrical power. The word Transistor is an acronym, and is a combination of the words Transfer Varistor used to describe their mode of operation way back in their early days of development. It is composed of semiconductor material usually with at least three terminals known and labeled as the Emitter (E), the Base (B) and the Collector (C) respectively. for connection to an external circuit. There are two types of standard transistors, NPN and PNP, with different circuit symbols. The letters refer to the layers of semiconductor material used to make the transistor. Most transistors used today are NPN because this is the easiest type to make from silicon.

Bipolar Transistor Construction



Transistors are three terminal active devices made from different semiconductor materials that can act as either an insulator or a conductor by the application of a small signal voltage. The transistor's ability to change between these two states enables it to have two basic functions: "switching" (digital electronics) or "amplification" (analogue electronics).

The two most common types of transistors are:

- Field-Effect Transistors (FETs): voltage-controlled current flow.
- Bipolar Junction Transistors (BJTs): current-controlled current flow.

FET (Field Effect Transistors) can be classified as JFT and MOSFET.

MOSFETs differ from BJTs in that BJTs require that a current be applied to the base pin in order for current to flow between the collector and emitter pins. On the other hand, MOSFETs only require a voltage at the gate pin to allow current flow between the drain and source pins.

Operational Amplifiers are the heart and soul of all modern electronic instruments. Their flexibility, stability and ability to execute many functions make op-amps the ideal choice for analog circuits. Historically, op-amps evolved from the field of analog computation where circuits were designed to add, subtract, multiply, integrate, and differentiate etc. in order to solve differential equations found in many engineering applications. Today analog computers op-amps are found in countless electronic circuits and instruments.

Operational Amplifiers (OAs) are highly stable, high gain dc difference amplifiers. Since there is no capacitive coupling between their various amplifying stages, they can handle signals from zero frequency (dc signals) up to a few hundred kHz. Their name is derived by the fact that they are used for performing **mathematical operations** on their input signal(s).

Figure 1 shows the symbol for an OA. There are two inputs, the **inverting input** (-) and the **non-inverting input** (+). These symbols have nothing to do with the polarity of the applied input signals.

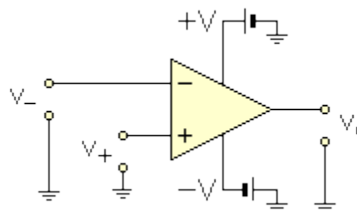


Figure 1. Symbol of the operational amplifier. Connections to power supplies are also shown.

The output signal (voltage), v_o , is given by: $v_o = A (v_+ - v_-)$

The Major Equipment that will be used in Analog Electronics lab is

1. CRO : Cathode Ray Oscilloscope
2. ASG: Amplitude Signal Generator

An **oscilloscope**, previously called an oscillograph, and informally known as a scope, **CRO** (for **cathode-ray oscilloscope**), or DSO (for the more modern digital storage **oscilloscope**), is a type of electronic test instrument that allows observation of constantly varying signal voltages. Actually **cathode ray oscilloscope** is very fast X-Y plotters that can display an input signal versus time or other signal as shown below. Oscilloscopes are used in the sciences, medicine, engineering, and the telecommunications industry. General-purpose instruments are used for maintenance of electronic equipment and laboratory work. Special-purpose oscilloscopes may be used for such purposes as analyzing an automotive ignition system or to display the waveform of the heartbeat as an [electrocardiogram](#).

Description:

The basic oscilloscope, as shown in the illustration, is typically divided into four sections: the display, vertical controls, horizontal controls and trigger controls. The display is usually a CRT or LCD panel which is laid out with both horizontal and vertical reference lines referred to as the graticule. In addition to the screen, most display sections are equipped with three basic controls: a focus knob, an intensity knob and a beam finder button.

The vertical section controls the amplitude of the displayed signal. This section carries a Volts-per-Division (Volts/Div) selector knob, an AC/DC/Ground selector switch and the vertical (primary) input for the instrument. Additionally, this section is typically equipped with the vertical beam position knob.

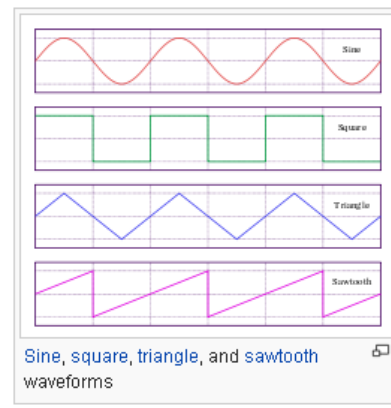
The horizontal section controls the time base or "sweep" of the instrument. The primary control is the Seconds-per-Division (Sec/Div) selector switch. Also included is a horizontal input for plotting dual X-Y axis signals. The horizontal beam position knob is generally located in this section.



2. ASG: Amplitude Signal Generator

A **function generator** is usually a piece of **electronic test equipment** or **software** used to generate different types of electrical **waveforms** over a wide range of frequencies. Some of the most common waveforms produced by the function generator are the sine, square, triangular and saw tooth shapes. These waveforms can be either repetitive or single-shot.

Function generators are used in the development, test and repair of electronic equipment. For example, they may be used as a signal source to test amplifiers or to introduce an error signal into a control loop.



Function generator basics:

Function generators, whether the old analog type or the newer digital type, have a few common features:

- A way to select a waveform type: sine, square, and triangle are most common, but some will give ramps, pulses, “noise”, or allow you to program a particular arbitrary shape.
- A way to select the waveform frequency. Typical frequency ranges are from 0.01 Hz to 10 MHz.
- A way to select the waveform amplitude.
- At least two outputs. The “main” output, which is where you find the desired waveform, typically has a maximum voltage of 20 volts peak-to-peak, or ± 10 volts range. The most common output impedance of the main output is 50 ohms, although lower output impedances can sometimes be found. A second output, sometimes called “sync”, “aux” or “TTL” produces a square wave with standard 0 and 5 volt digital signal levels. It is used for synchronizing another device (such as an oscilloscope) to the possibly variable main output signal. A wide variety of other features are available on most modern function generators, such as “frequency sweep”—the ability to automatically vary the frequency between a minimum and maximum value, “DC offset”—a knob that adds a specified amount of DC voltage to the time-varying waveform, and extra inputs or outputs that can be used to control these extra features by other instruments.

3. Introduction to Multisim:

NI Multisim (formerly MultiSIM) is an electronic schematic capture and simulation program which is part of a suite of circuit design programs, along with NI Ulti board. Multisim is one of the few circuit design programs to employ the original Berkeley SPICE based software simulation. Multisim was originally created by a company named Electronics Workbench, which is now a division of National Instruments. Multisim includes microcontroller simulation (formerly known as Multi MCU), as well as integrated import and export features to the Printed Circuit Board layout software in the suite, NI Ulti board.

Multisim is widely used in academia and industry for circuits' education, electronic schematic design and SPICE simulation.

Steps to Proceed:

[Step 1: Open Multisim](#)

[Step 2: Place Components](#)

[Step 3: Wire Components](#)

[Step 4: Place a Simulation Source](#)

[Step 5: Place Measurement Instruments](#)

[Step 6: Run a Simulation](#)

Laboratory Session-2. Logic design components

The digital logic design area covers the digital building blocks, tools, and techniques in the design of computers and other digital systems. Digital logic design is one of the topic areas that differentiate computer engineers from electrical engineers and computer scientists. It covers a variety of basic topics, including switching theory, combinational and sequential logic circuits, and memory elements.

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 and NOT are basic gates. NAND, NOR 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:

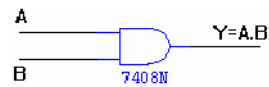
The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

X-OR GATE:

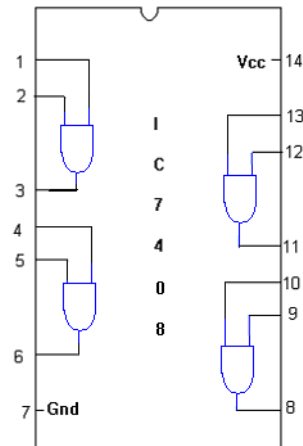
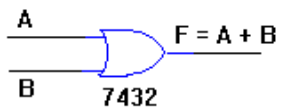
The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

PROCEDURE:

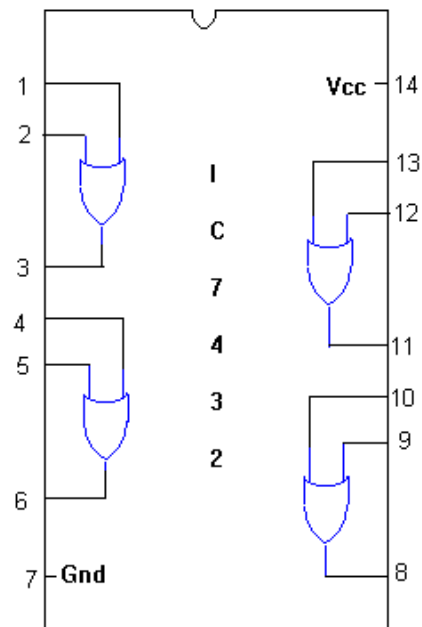
- (i) Connections are given as per circuit diagram.
- (ii) Logical inputs are given as per circuit diagram.
- (iii) Observe the output and verify the truth table.

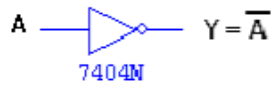
AND GATE:**SYMBOL:****TRUTH TABLE**

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

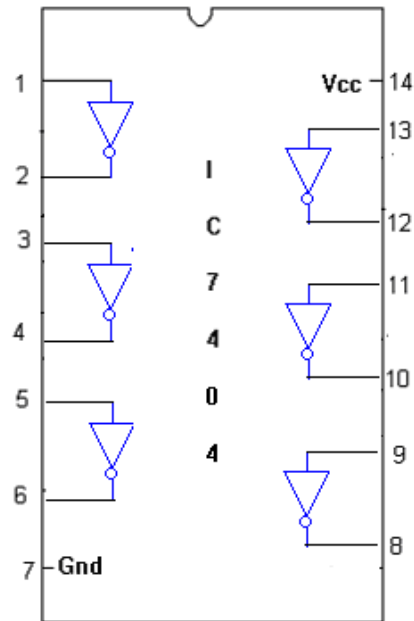
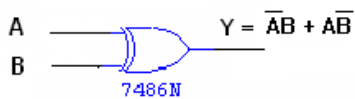
PIN DIAGRAM:**OR GATE:****SYMBOL :****TRUTH TABLE**

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

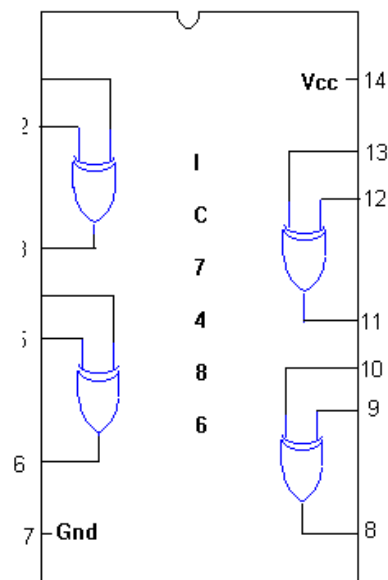
PIN DIAGRAM :

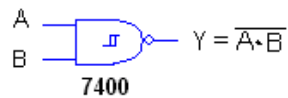
NOT GATE:**SYMBOL:****TRUTH TABLE :**

A	\overline{A}
0	1
1	0

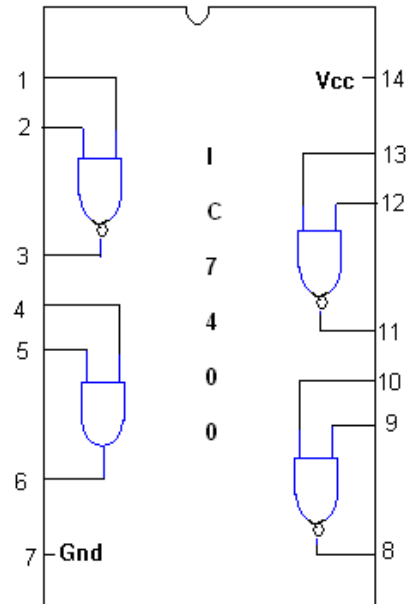
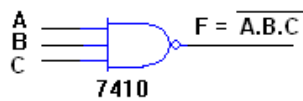
PIN DIAGRAM:**X-OR GATE:****SYMBOL:****TRUTH TABLE :**

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

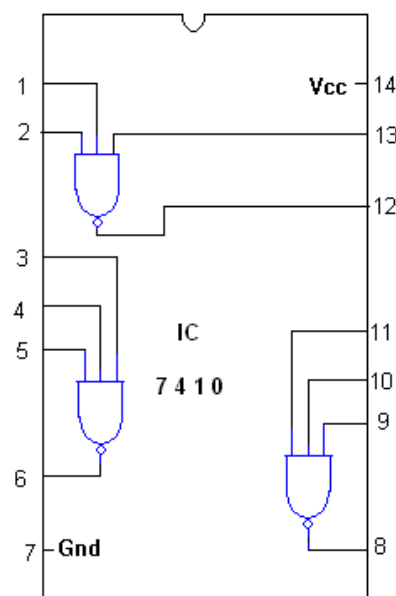
PIN DIAGRAM:

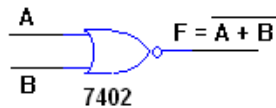
2-INPUT NAND GATE:**SYMBOL:****TRUTH TABLE**

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

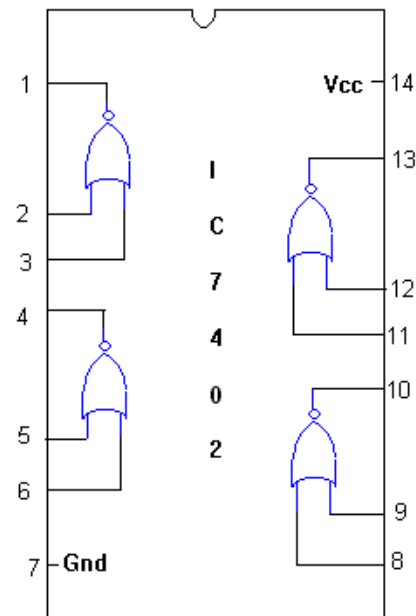
PIN DIAGRAM:**3-INPUT NAND GATE:****SYMBOL :****TRUTH TABLE**

A	B	C	$\overline{A \cdot B \cdot C}$
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

PIN DIAGRAM :

NOR GATE:**SYMBOL :****TRUTH TABLE**

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

PIN DIAGRAM:**INTRODUCTION TO XILINX**

Xilinx ISE is a software tool produced by Xilinx for synthesis and analysis of HDL designs, which enables the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

ModelSim is a verification and simulation tool for VHDL, Verilog, System Verilog, and mixed language designs.

INTRODUCTION TO VHDL

HDL (Hardware Description Language) based design has established itself as the modern approach to design of digital systems, with VHDL (VHSIC Hardware Description Language) and Verilog HDL being the two dominant HDLs. Numerous universities thus introduce their students to VHDL (or Verilog). The problem is that VHDL is complex due to its generality. Introducing students to the language first, and then showing them how to design digital systems with the language, tends to confuse students. The language issues tend to distract them from the understanding of digital components. And the synthesis subset issues of the language add to the confusion.

VHDL stands for **VHSIC** (Very High Speed Integrated Circuits) **Hardware Description Language**. In the mid-1980's the U.S. Department of Defense and the IEEE sponsored the development of this hardware description language with the goal to develop very high-speed integrated circuit. It has become now one of industry's standard languages used to describe digital systems. The other widely used hardware description language is Verilog. Both are powerful languages that allow you to describe and simulate complex digital systems. A third HDL language is ABEL (Advanced Boolean Equation Language) which was specifically designed for Programmable Logic Devices (PLD). ABEL is less powerful than the other two languages and is less popular in industry. This tutorial deals with VHDL, as described by the IEEE standard 1076-1993.

Although these languages look similar as conventional programming languages, there are some important differences. A hardware description language is inherently parallel, i.e. commands, which correspond to logic gates, are executed (computed) in parallel, as soon as a new input arrives. A HDL program mimics the behavior of a physical, usually digital, system. It also allows incorporation of timing specifications (gate delays) as well as to describe a system as an interconnection of different components

Sample Programs:

1. Write the Verilog /VHDL code for a 2 Input AND gate. Simulate and verify its working.

```
Entity    And1  is
    Port (A, B: IN STD_LOGIC;
          C: OUT STD_LOGIC);
End       And1;

Architecture Behavioral of And1 is
    Begin
        C <= A AND B;
End       Behavioral;
```

NOTE: write the VHDL code for remaining basic and universal gates. Simulate and realize the output.

2. Write the Verilog /VHDL code for a half adder. Simulate and verify its working.

Entity Half_adder is

Port (A, B: IN STD_LOGIC;

SUM, CARRY: OUT STD_LOGIC);

End Half_adder;

Architecture Behavioral of Half_adder is

Begin

SUM <= A XOR B;

CARRY <= A AND B;

End Behavioral;

CSE

Design:

1. Design an Astable Multivibrator for the given frequency of 1KHZ and duty cycle of 30%, 50%, 60%

Solution: For 60% since the given frequency is 1KHZ,

$$\text{Hence } T = 1/F = 1/1\text{KHZ} = 1\text{ms.}$$

$$\text{Duty cycle } D = 30\% = 30/100 = 0.3$$

$$\text{Capacitor charging time} = t_{\text{on}} = 0.69 R_1 C \text{ ----- (1)}$$

$$\text{Capacitor discharging time} = t_{\text{off}} = 0.69 R_2 C \text{ ----- (2)}$$

$$\text{The Total Time is } T = T_{\text{on}} + T_{\text{off}} \text{ ----- (3)}$$

The expression for Duty cycle is given by: **$D = T_{\text{on}}/T$** using this expression calculate for T_{on}

$$0.6 = T_{\text{on}}/1\text{ms} \quad : T_{\text{on}} = 0.6 * 1\text{ms} = 0.6\text{ms}$$

Now calculate T_{off} from expression (3)

$$T_{\text{off}} = T - T_{\text{on}} = 1\text{ms} - 0.6\text{ms} = 0.4\text{ms}$$

Hence $T_{\text{on}} = 0.6 \text{ ms}$ and $T_{\text{off}} = 0.4\text{ms}$

Now find the values of R_1 , R_2 and C_1 from expressions (1) and (2).

$$\text{From (1) } t_{\text{on}} = 0.69 (R_1) C \text{ ----- (4)}$$

$$\text{W.k.t } T_{\text{on}} = 0.6\text{ms Assume } C = 0.1\mu\text{f, } R_1 = 8.79\text{K}$$

$$t_{\text{off}} = 0.69 (R_2)C$$

$$\text{Choose } C = 0.1\mu\text{f, } T_{\text{off}} = 0.4\text{ms Therefore } R_2 = 5.77\text{K}$$

Apply these values in the circuit given in Fig 1. And check for the output.

****** Similarly design for Duty Cycle 40% and 50 %**

For a Duty cycle more than 50%, the Circuit in Fig 1 can be used.

But for duty cycle less than 50%, there should be a diode in parallel with R_2 so as to make the charging and discharging path separate. (By Using Diode, We can use for any Duty cycle.)

2. Design an Astable Multivibrator for the given frequency of 3KHZ and duty cycle of 70%.

Lab Experiment No: 01

Date:

ASTABLE MULTIVIBRATOR**Aim:**

Design an astable multivibrator circuit for three cases of duty cycle (50%, <50% and >50%) using NE 555 timer IC. Simulate the same for any one duty cycle.

Apparatus:

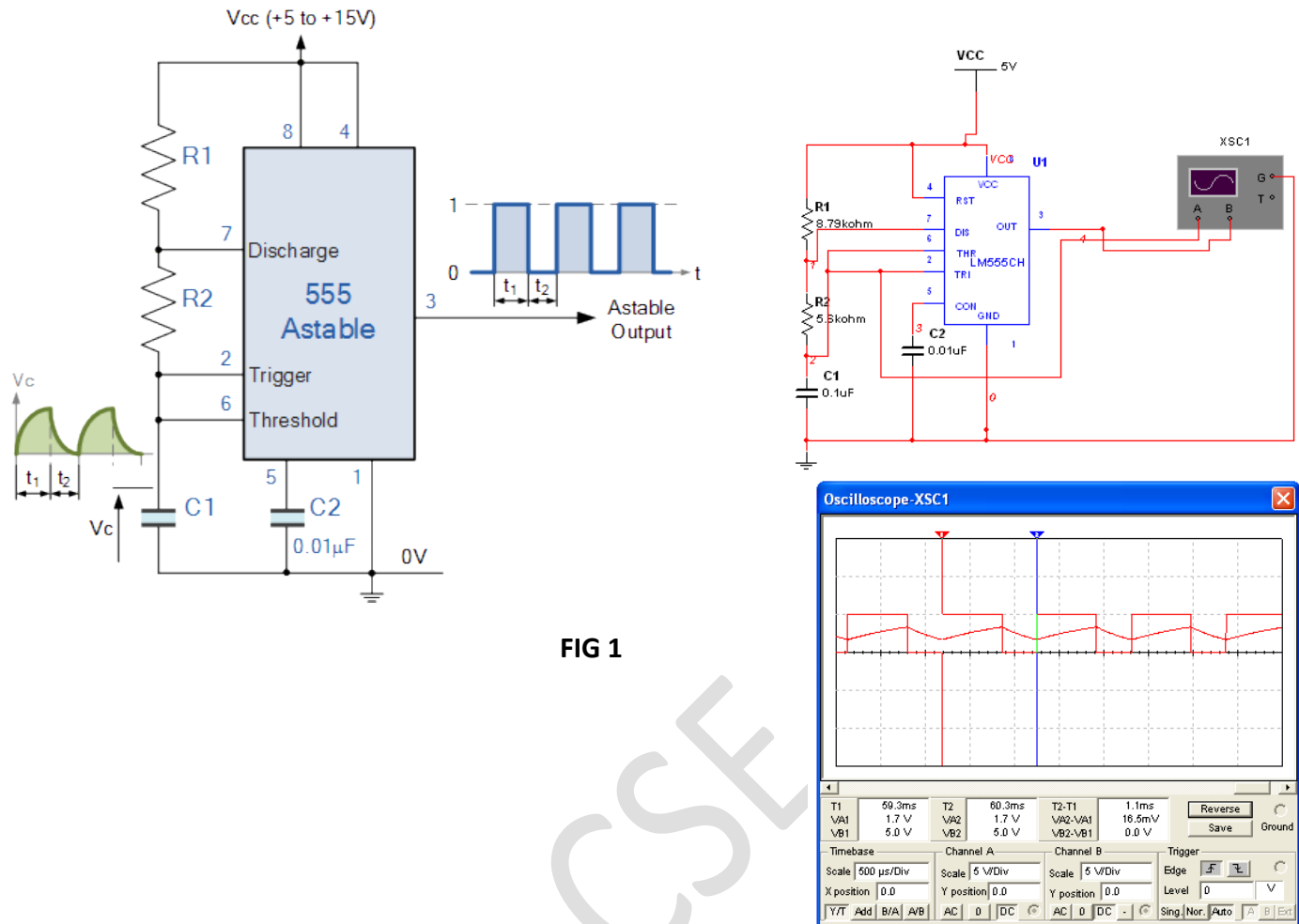
SI No	Particulars	Range	Qty
1	555 timer IC	--	1
2	Power Supply	--	-
3	Resistors	As per design	-
4	Capacitors	As per design	-
5	CRO, Patch cords and wires	--	1 set
6	Diode	1N4007	01

Theory:

A multivibrator is an electronic circuit used to implement a variety of simple multi-state systems such as oscillators, timers and flip-flops. It oscillates between a "HIGH" state and a "LOW" state producing a continuous output. There are three types of Multivibrator: Bistable, Monostable and Astable Multivibrator. An Astable Multivibrator or a Free Running Multivibrator is the multivibrator which has no stable states. The time period of each state are determined by Resistor Capacitor (RC) time constant. This circuit does not require any external trigger to change the state of the output, hence the name free-running.

The re-triggering is basically achieved by connecting the *trigger* input (pin 2) and the *threshold* input (pin 6) together, thereby allowing the device to act as an astable oscillator. Then the 555 Oscillator has no stable states as it continuously switches from one state to the other. Also the single timing resistor of the previous monostable multivibrator circuit has been split into two separate resistors, R1 and R2 with their junction connected to the *discharge* input (pin 7)

In the **555 Oscillator** circuit above, pin 2 and pin 6 are connected together allowing the circuit to re-trigger itself on each and every cycle allowing it to operate as a free running oscillator. During each cycle capacitor, C charges up through both timing resistors, R1 and R2 but discharges itself only through resistor, R2 as the other side of R2 is connected to the *discharge* terminal, pin 7.



Result:

Sl No	F(Theoretical)	D(Theoretical)	Time Period			F(Practical)	D(Practical)
			T on	T off	T		
1.	1 KHz	60%					
2.	1 K Hz	50%					
3.	1 KHz	40%					
4.	3 KHz	60%					

Staff Signature and date:

Student Signature and Date

Then the capacitor charges up to $2/3V_{cc}$ (the upper comparator limit) which is determined by the $0.693(R1+R2)C$ combination and discharges itself down to $1/3V_{cc}$ (the lower comparator limit) determined by the $0.693(R2 \cdot C)$ combination. This results in an output waveform whose voltage level is approximately equal to V_{cc} and whose output "ON" and "OFF" time periods are determined by the capacitor and resistors combinations.

OR

Initially, capacitor C is fully discharged, which forces the output to go to the HIGH state. An open discharge transistor allows capacitor C to charge from $+V_{cc}$ through resistors R1 and R2. When the voltage across C exceeds $+2V_{cc}/3$, the output enters the LOW state and the discharge transistor is switched ON at the same time. Capacitor C starts to discharge through R2 and the discharge transistor inside the IC. When the voltage across C falls below $+V_{cc}/3$, the output enters the HIGH state. The charge and discharge cycles repeat and the circuit behaves as a free running multivibrator. When connected as an astable multivibrator, the output from the **555 Oscillator** will continue indefinitely charging and discharging between $2/3V_{cc}$ and $1/3V_{cc}$ until the power supply is removed.

Applications:

555 Oscillator be used in a wide range of waveform generator circuits and applications that require very little output current such as in electronic test equipment for producing a whole range of different output test frequencies.

The 555 can also be used to produce very accurate sine, square and pulse waveforms or as LED or lamp flashers and dimmers to simple noise making circuits such as metronomes, tone and sound effects generators and even musical toys.

Procedure:

- a) Check the components / Equipment for their working condition.
- b) Rig up the circuit as shown in the diagram.
- c) Observe the output wave from CRO and Record the values of T_{on} and T_{off} .
- d) Calculate the T and duty cycle.
- e) Compare the theoretical value with practical value.

1. Design a Relaxation Oscillator for a given frequency of 1KHZ.

Relation between R_1 and R_2 is given $R_2 = 1.16 R_1$

Let $R_1 = 10 \text{ k}\Omega$. Then $R_2 = 11.6 \text{ k}\Omega$ To calculate R: Formula to be used:

$$f_0 = \frac{1}{2RC}$$

Given frequency is $f_0 = 1 \text{ KHz}$. Assume $C = 0.1 \mu\text{F}$ then $R = 5 \text{ k}\Omega$

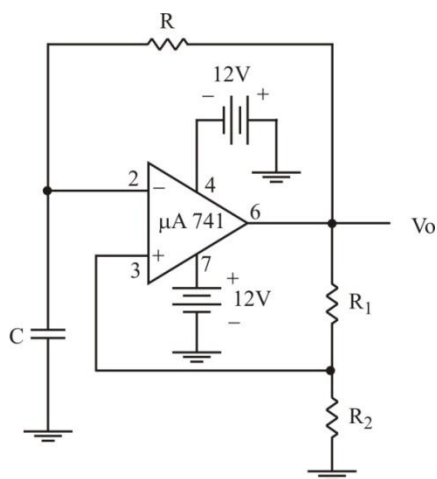
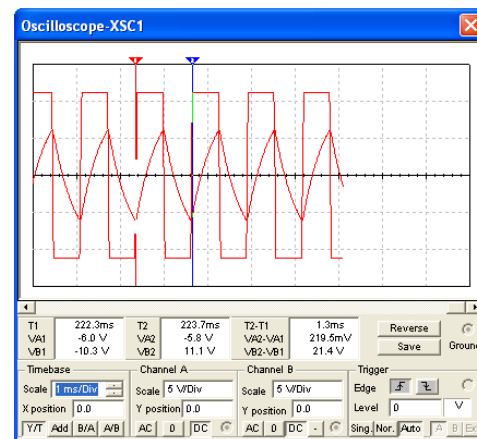
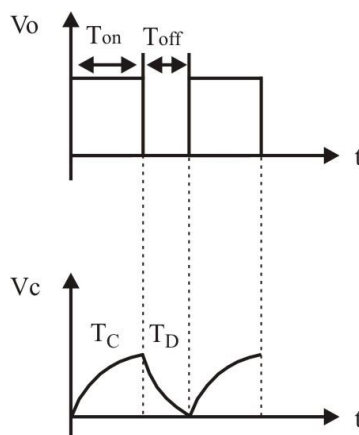
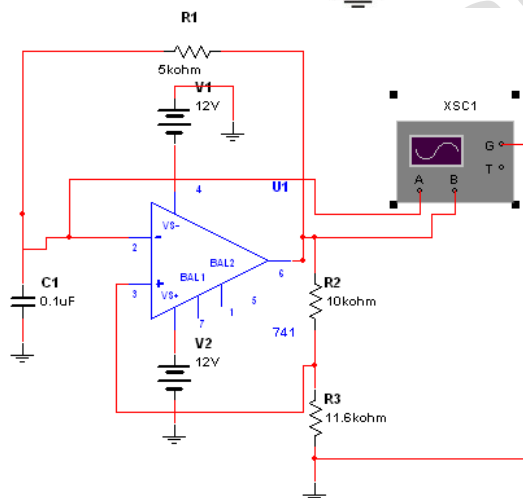
Circuit Diagram:Wave forms:

FIG 2.

**Tabular Column**

Sl No	f_0 (Theoretical)	f_0 (pract) H/W	f_0 (pract) simulation	Remarks
1.	1 KHz			
2.	3 KHz			

3. Design a Relaxation Oscillator for a given frequency of 3KHZ.

Result:

Staff Signature and date:

Student Signature and Date

Lab Experiment No: 02**Date:****RELAXATION OSCILLATOR****Aim:**

Using ua 741 Opamp, design a 1 kHz Relaxation Oscillator with 50% duty cycle. And simulate the same.

Apparatus:

Sl No	Particulars	Range	Qty
1	IC	μ A 741	01
2	Resisters	As per Design	-
3	Capacitors	As per Design	-
4	CRO with probes	-	1 set
5	Spring board and wires	-	1 set

Theory:

A *relaxation oscillator* is a circuit that repeatedly alternates between two states at with a period that depends on the charging of a capacitor. It generates a changing voltage at a particular frequency by charging and discharging a capacitor through a resistor, and is often built around an operational amplifier. The capacitor voltage may change exponentially when charged or discharged through a resistor from a constant voltage, or linearly when charged or discharged through a constant current source.

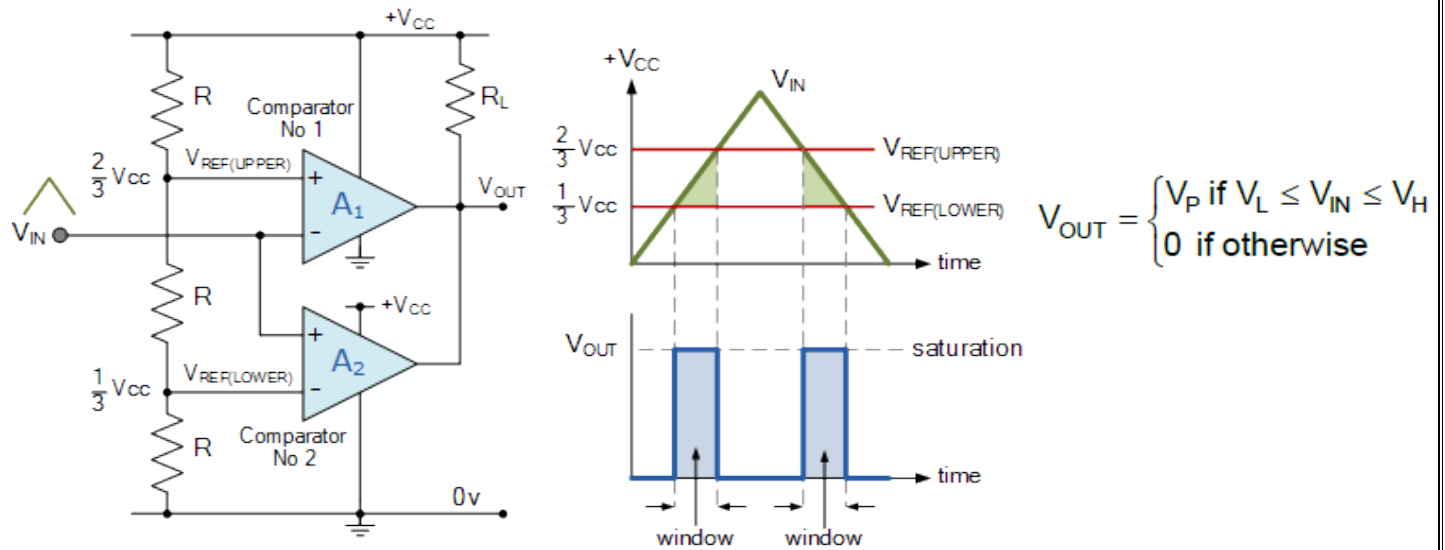
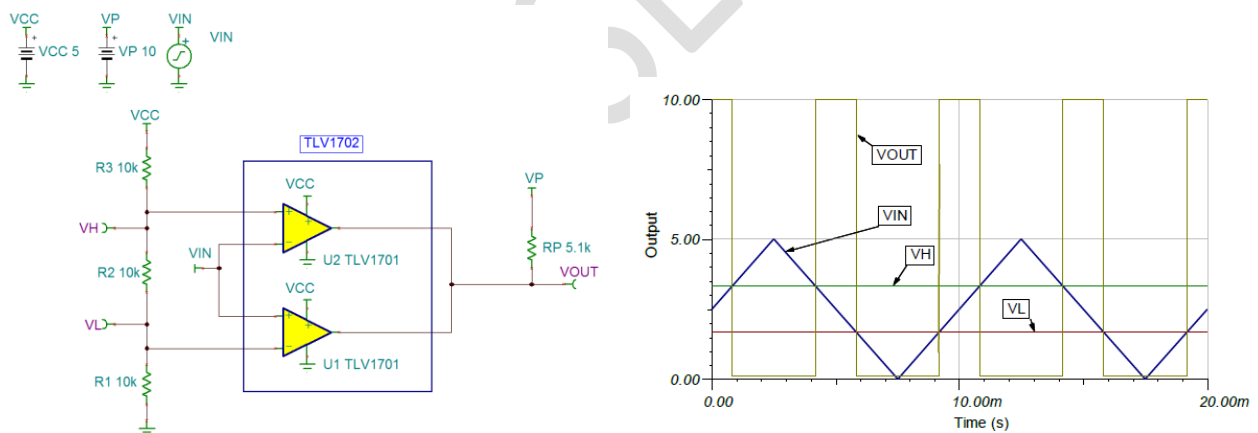
Applications:

Relaxation oscillators are generally used to

1. Produce low frequency signals for such applications as blinking lights, and electronic beepers, Clock signals in some digital circuits.

PROCEDURE:

1. Check the components / Equipment for their working condition.
2. Make connections as shown in circuit diagram.
3. Check the wave form at pin 6 of op Amp and make a note of Ton and Toff.
4. Calculate the frequency of the output and compare with the given frequency.

Circuit**Fig 3****Simulation Circuit:**

Note: To get Better Understanding, Use Two Diodes across output the two comparators so that
That at any point of time either the diode is FB.

SI No	UTP(D)	LTP(D)	UTP (P)	LTP (P)	Remarks
1.					
2.					

Lab Experiment No: 03**Date:****WINDOW COMPARATOR****Aim:**

Using ua 741 opamp, design a window comparator for any given UTP and LTP. And simulate the same.

Apparatus:

Sl No	Particulars	Range	Qty
1	IC	μA 741	02
2	Resisters	As per Design	-
3	CRO with probes	-	1 set
4	Spring board and wires	-	1 set

Theory:

A **Window Comparator** is basically the inverting and the non-inverting comparators above combined into a single comparator stage. The window comparator detects input voltage levels that are within a specific band or *window* of voltages, instead of indicating whether a voltage is greater or less than some preset or fixed voltage reference point.

Working of the Circuit:

When V_{IN} is below the lower voltage level, $V_{REF(LOWER)}$ which equates to $1/3V_{CC}$, the output will be LOW. When V_{IN} exceeds this $1/3V_{CC}$ lower voltage level, the first op-amp comparator detects this and switches the output HIGH to V_{CC} .

As V_{IN} continues to increase it passes the upper voltage level, $V_{REF(UPPER)}$ at $2/3V_{CC}$ and the second op-amp comparator detects this and switches the output back LOW. Then the difference between $V_{REF(UPPER)}$ and $V_{REF(LOWER)}$ (which is $2/3V_{CC} - 1/3V_{CC}$ in this example) creates the switching window for the positive going signal.

Design:

Design a window comparator for any given UTP/VH (Upper Triggering Pulse/Voltage High) and LTP/VL (Lower Triggering Pulse/Voltage Low). Assume **VH= 3.33 V and VL = 1.66V**

In the above **Fig 3**. A reference voltage, VCC, is divided down by resistors R1-R3. The two node voltages, VH and VL, define the upper window voltage and lower window voltage, respectively. When the input voltage is between VH and VL, the output is 'high', or VP. When outside the window voltage, the output is pulled down to 0 V.

Equations (1) and (2) define VH and VL, respectively. Assume VH= 3.33 V and VL = 1.66V

$$V_H = V_{CC} \times \frac{R_1 + R_2}{R_1 + R_2 + R_3} \quad .$$

Eqn 1

$$V_L = V_{CC} \times \frac{R_1}{R_1 + R_2 + R_3}$$

Eqn 2.

Solving Equations (1) and (2) for VCC, setting them equal to each other, then simplifying yields Equation (3).

$$V_H \left(\frac{R_1 + R_2 + R_3}{R_1 + R_2} \right) = V_L \left(\frac{R_1 + R_2 + R_3}{R_1} \right) \rightarrow \frac{V_H}{V_L} = \frac{R_1 + R_2}{R_1} \quad (3)$$

Given VH=3.33 V and VL=1.66 V, the ratio of R1 and R2 is calculated in Equation (4).

$$\frac{V_H}{V_L} = \frac{R_1 + R_2}{R_1} \rightarrow \frac{3.33}{1.66} = 2 = 1 + \frac{R_2}{R_1} \rightarrow 1 = \frac{R_2}{R_1}$$

R1 and R2 were selected to be 10 kΩ. While the values of R1 and R2 are related to the ratio of the window voltages, R3 determines the voltage value. R3 is calculated in Equation (5). **Given a single 5 V supply (VCC), the input voltage range is 0 V to 5 V.**

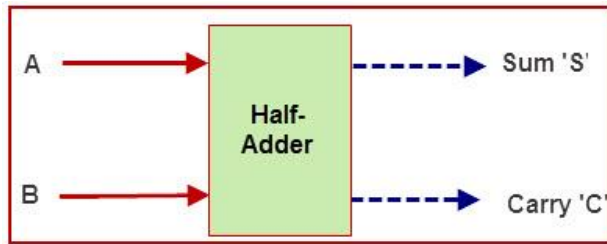
$$V_L = V_{CC} \times \frac{R_1}{R_1 + R_2 + R_3} \rightarrow R_3 = \frac{R_1 \times V_{CC}}{V_L} - (R_1 + R_2) = 10k\Omega$$

In summary, R1=R2=R3=10 kΩ.

Assignment:

Design a window comparator for any given UTP/VH (Upper Triggering Pulse/Voltage High) and LTP/VL (Lower Triggering Pulse/Voltage Low). Assume **VH= 5 V and VL = 2 V**

Intentionally Left Blank

Half Adder:**Half Adder Truth Table:**

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

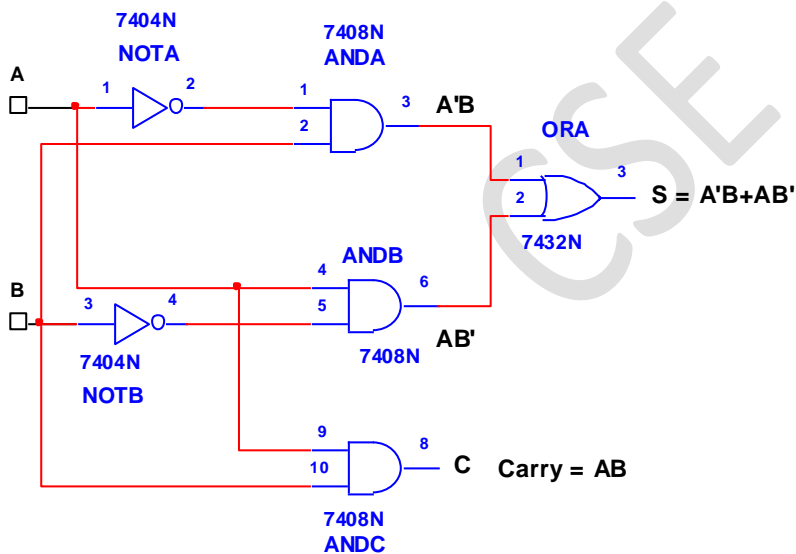
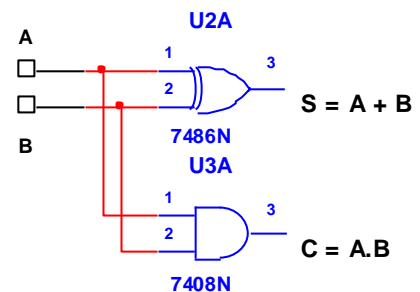
From the above truth table of Half adder:

The simplified Boolean function for Sum is: $S = A'B + AB'$ and Carry is $C = A.B$.

The circuit diagram is as shown in **Fig (A)** Using basic gates.

By simplifying the above expression of sum and carry using De Morgan's Law we get:

Sum = $A \oplus B$ and Carry = $A.B$. It is shown in **Fig (B)**.

**Fig A****Fig B**

Lab Experiment No: 04**Date:****ADDER AND SUBTRACTOR****Aim:**

Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.

Components:

Sl No	Particulars	IC Number	Qty
1	2 Input AND gate IC	7408	As Per requirements
2	2 Input OR gate IC	7432	As Per requirements
3	NOT gate	7404	As Per requirements
4	XOR gates	7486	As Per requirements
5	Patch Cords	----	As Per requirements
6	Digital IC Trainer Kit	----	01

Theory:

An **adder**, also called summer, is a digital circuit that performs addition of numbers.

1. Half Adder:

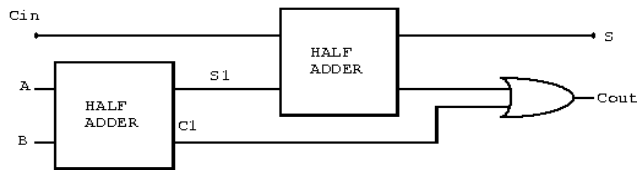
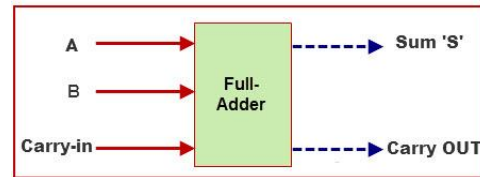
It is a combinational circuit that performs the addition of two bits; this circuit needs two binary inputs and two binary outputs, with one producing sum output and other produce carry output. The half-adder is useful to add one binary digit quantities.

2. Full adder:

This type of adder is a little more difficult to implement than a half-adder. The main difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as CIN. The output carry is designated as COUT and the normal output Sum is designated as S.

3. Half Subtractor:

A half Subtractor is a multiple output combinational logic network that does the subtraction of two bits of binary data. It has input variables and two output variables. Two inputs are corresponding to two input bits and two output variables correspond to the difference bit and borrow bit.

Full Adder:**Full Adder Using Two Half Adders:****Full Adder Block Diagram****Full Adder Truth Table:**

INPUTS			OUTPUT	
A	B	C-IN	C-OUT	S
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

From the above truth table:

The simplified Boolean function from the truth table using SoP method is: $S = A'B'C + A'BC' + ABC' + ABC$ **Fig (C)**

The simplified Boolean function from the truth table Using SoP is $C_{out} = AB + B C_{in} + C_{in} A$ **Fig (C)**

By simplifying the above expression using DeMorgan's Law we get: $Sum = A \oplus B \oplus C$ and

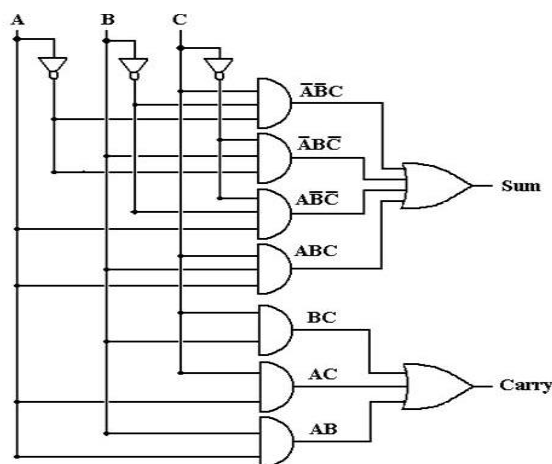


Fig (C)

$$Carry = C_{in}(A + B) + AB$$

Fig (D)

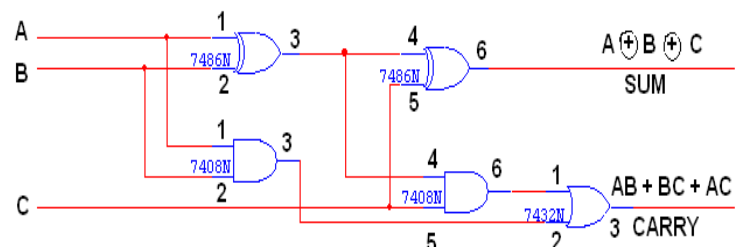


Fig (D)

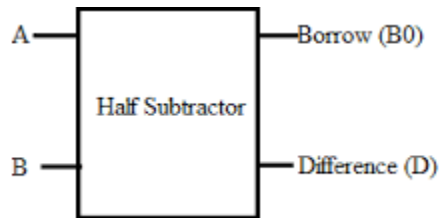
4. Full Subtractor:

A combinational logic circuit performs a subtraction between the two binary bits by considering borrow of the lower significant stage is called as the full Subtractor. In this, subtraction of the two digits is performed by taking into consideration whether a 1 has already borrowed by the previous adjacent lower minuend bit or not.

It has three input terminals in which two terminals corresponds to the two bits to be subtracted (minuend A and subtrahend B), and a borrow bit B_i corresponds to the borrow operation. There are two outputs, one corresponds to the difference D output and other borrow output B_o as shown in figure along with truth table.

Procedure:

1. Connections are made as shown in the logic diagrams.
2. Verify the truth table for all the logic circuits.

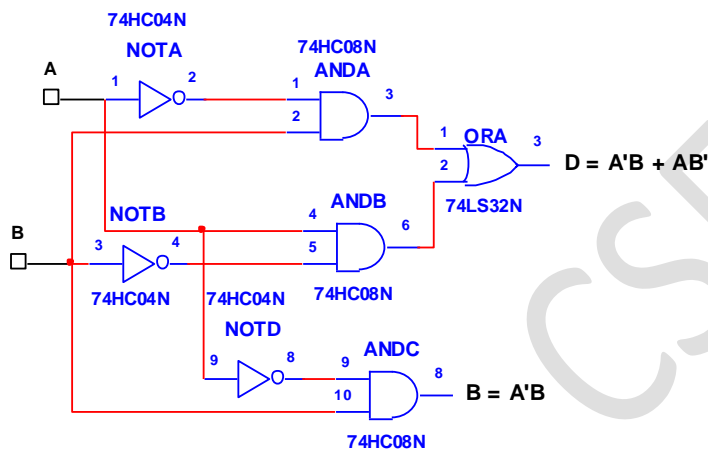
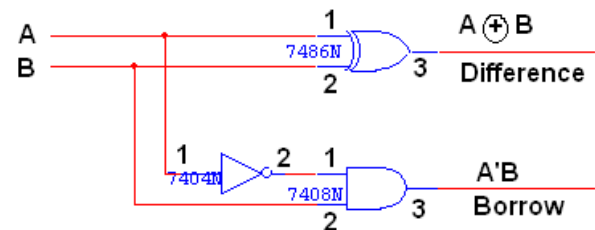
Half Subtractor:**Truth Table**

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

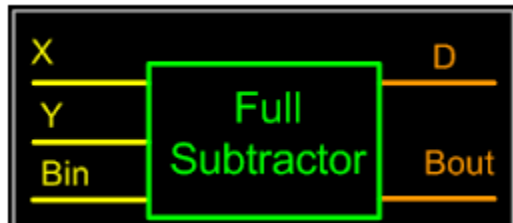
From the above truth table:

The expression for Difference using SOP is: $A'B + AB'$ and Borrow is: $A'B$ and the circuit **Using Basic Gates** is as shown in **Fig (E)**.

The simplified block diagram is as shown in **Fig (F)**.

**Fig (E)****Fig (F)****Assignment question:**

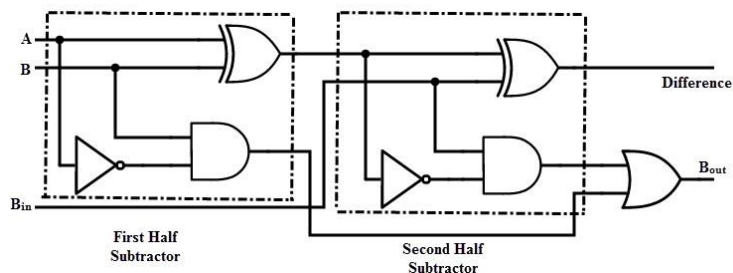
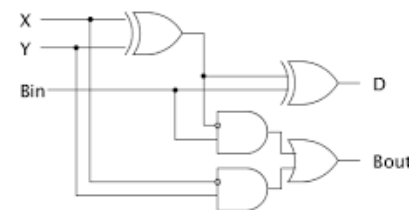
1. Realize the truth table for the above said circuits using K –Map. (SoP).

Full Subtractor:**Truth Table:**

X	Y	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

The expression for Full Subtractor is :

Difference: $X \oplus Y \oplus B_{in}$ Borrow: $X'Y + X B_{in} + Y \cdot B_{in}'$. The logic circuit is as shown in Fig (G) and Fig (H)

**Fig (G)****Fig (H)****Results:**

1. Truth table verified for :
2. Truth table verified for :
3. Truth table verified for :
4. Truth table verified for :

B. Design and develop the Verilog /VHDL code for an Half Adder, Full adder, Half Subtractor and Full Subtractor

1. Write the Verilog /VHDL code for a Half Adder. Simulate and verify its working.

```
Entity    HA is
    Port (A, B: IN STD_LOGIC;
          S, C: OUT STD_LOGIC);
End      HA;
Architecture Behavioral of HA is
    Begin
    S <= A XOR B;
    C <= A AND B;
End      Behavioral;
```

2. Write the Verilog /VHDL code for a Full Adder. Simulate and verify its working.

```
Entity    FA is
    Port (A, B, C : IN STD_LOGIC;
          S, C: OUT STD_LOGIC);
End      FA;
Architecture Behavioral of FA is
    Begin
    S <= A XOR B XOR C;
    C <= A AND B OR B AND C OR C AND A;
End      Behavioral;
```

OR

```
entity FA is
    Port ( a,b,c : in std_logic;
          s, c : out std_logic);
end FA;

architecture Behavioral of FA is
begin
    process(a,b,c)
    begin
        if(a='0' and b='0' and c='0')then
            s<='0';
            c<='0';
        elsif( a='0' and b='0' and c='1')then
            s<='1';
            c<='0';
        elsif( a='0' and b='1' and c='0')then
```

```
s<='1';
c<='0';
elseif( a='0' and b='1' and c='1')then
s<='0';
c<='1';
elseif( a='1' and b='0' and c='0')then
s<='1';
c<='0';
elseif( a='1' and b='0' and c='1')then
s<='0';
c<='1';
elseif( a='1' and b='1' and c='0')then
s<='0';
c<='1';
else
s<='1';
c<='1';
end if;
end process;
```

end Behavioral;

3. Write the Verilog /VHDL code for a Half Subtractor. Simulate and verify its working.

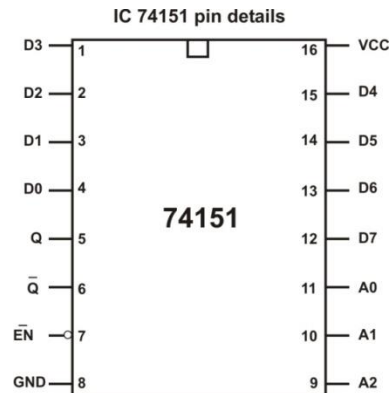
```
Entity    HS  is
    Port (A, B: IN STD_LOGIC;
          D, B: OUT STD_LOGIC);
End    HS;
Architecture Behavioral of HS is
    Begin
D <= ((NOT A) AND B) OR (A AND (NOT B) );
B <= ((NOT A) AND B);
End    Behavioral;
```

3. Write the Verilog /VHDL code for a Full Subtractor. Simulate and verify it's working.

Assignment: Try Your Self

Staff Signature and date:

Student Signature and Date

**Function Table:**

INPUTS				OUTPUT		Comments
A2	A1	A0	$\overline{\text{EN}}$	Q	$\overline{\text{Q}}$	
X	X	X	H	H	H	If En pin is at logical high value then irrespective of input, output will be 1.
L	L	L	L	D0	$\overline{\text{D0}}$	If En pin is at logical zero value then if input is 0 0 0 then output will be D0
L	L	H	L	D1	$\overline{\text{D1}}$	If En pin is at logical zero value then if input is 0 0 1 then output will be D1
L	H	L	L	D2	$\overline{\text{D2}}$	If En pin is at logical zero value then if input is 0 1 0 then output will be D2
L	H	H	L	D3	$\overline{\text{D3}}$	If En pin is at logical zero value then if input is 0 1 1 then output will be D3
H	L	L	L	D4	$\overline{\text{D4}}$	If En pin is at logical zero value then if input is 1 0 0 then output will be D4
H	L	H	L	D5	$\overline{\text{D5}}$	If En pin is at logical zero value then if input is 1 0 1 then output will be D5
H	H	L	L	D6	$\overline{\text{D6}}$	If En pin is at logical zero value then if input is 1 1 0 then output will be D6
H	H	H	L	D7	$\overline{\text{D7}}$	If En pin is at logical zero value then if input is 1 1 1 then output will be D7

Lab Experiment No: 05**Date:****MULTIPLEXER****Aim:**

5. a) Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the Simplified logic expression using 8:1 multiplexer IC.
- b) Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its Working.

Apparatus:

Sl. NO	Particulars	Specification	Quantity
1.	8:1 Multiplexer IC	IC 74151	1
2.	Digital IC Trainer Kit	----	1
3.	Patch cords	----	20

Theory:

A multiplexer or simply “mux” is a device that selects between a number of input signals. In its simplest form, a multiplexer will have two input signals, 1 control input, and 1 output. The number of inputs is generally a multiple of 2 (2, 4, 8, 16, etc), the number of outputs is 1, and n control inputs are used to select one of the data inputs. The multiplexer output value is same as that of the selected data input.

In other words, the multiplexer works like the input selector of a home music system. Only one input is selected at a time, and the selected input is transmitted to the single output. While on the music system, the selection of the input is made manually, the multiplexer chooses its input based on a binary number, the address input.

Multiplexers are used in building digital semiconductors such as CPUs and graphics controllers. They are also used in communications; the telephone network is an example of a very large virtual mux built from many smaller discrete ones.

Consider a 4 Variable expression as: $f(w, x, y, z) = \sum (2,4,5,7,10,11,14) + \sum' d (8,9,12,13,15)$

Let 'z' be map entered variable

Decimal Value	Inputs W X Y	MEV Z	Output F	Entry in MEV Map	Comments
0	0 0 0	0	0	0 (D0)	If function F equals 0 for both values of MEV, enter 0 in appropriate cell on MEV map
1	0 0 0	1	0		
2	0 0 1	0	1	\bar{Z} (D1)	If function F complements to the values of MEV then enter complement of MEV.
3	0 0 1	1	0		
4	0 1 0	0	1	1 (D2)	If function equals 1 for both values of MEV, enter 1.
5	0 1 0	1	1		
6	0 1 1	0	0	Z (D3)	If function equals to MEV value the enter MEV.
7	0 1 1	1	1		
8	1 0 0	0	X	X (D4)	If both function values are X then enter 0 or 1
9	1 0 0	1	X		
10	1 0 1	0	1	1 (D5)	If function equals 1 for both values of MEV, enter 1
11	1 0 1	1	1		
12	1 1 0	0	X	X (D6)	If both function values are X then enter 0 or 1
13	1 1 0	1	X		
14	1 1 1	0	1	1 (D7)	If function equals 1 for both values of MEV, enter 1
15	1 1 1	1	X		

Result:

Assignment: Simplify Entered Variable Map and realize the Simplified logic expression using 8:1 multiplexer IC. $f(w, x, y, z) = \sum (1,4,6,7,9,10,11,13) + \sum' d (2,3,12,14,15)$

Procedure:

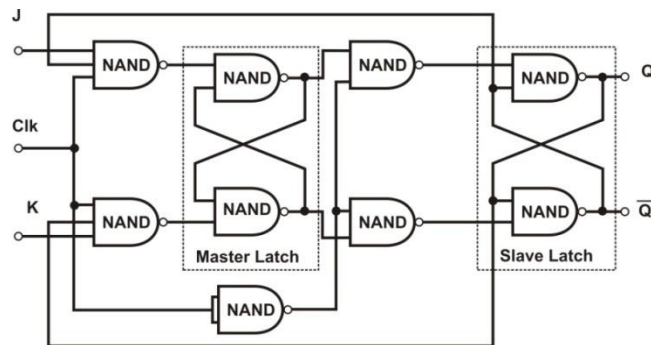
1. Verify all components and patch cords for their good working condition.
2. Make the connection as shown in the circuit diagram.
3. Give supply to the trainer kit.
4. Provide input data to circuit via switches and verify the function table and truth table.

b. Design and develop the Verilog /VHDL code for an 8:1 Multiplexer IC. Simulate and verify its working.

```
Entity MUX is
    Port( sel : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
          A, B, C, D, E, F, G, H: IN STD_LOGIC;
          Z: OUT STD_LOGIC);
End MUX;
Architecture Behavioral of MUX is
    Begin
        Process (sel, A, B, C, D, E, F, G, H)
            Begin
                Case sel is
                    When "000" => Z <= A;
                    When "001" => Z <= B;
                    When "010" => Z <= C;
                    When "011" => Z <= D;
                    When "100" => Z <= E;
                    When "101" => Z <= F;
                    When "110" => Z <= G;
                    When "111" => Z <= H;
                    When others => NULL;
                End Case;
            End Process;
        End Behavioral;
```

Staff Signature and date:**Student Signature and Date**

Circuit diagram:



JK FLIP-FLOP TRUTH TABLE:

J	K	Clk	Q	\bar{Q}
0	0	Pos- edge	No Change	
0	1	Pos- edge	0	1
1	0	Pos- edge	1	0
1	1	Pos- edge	Toggle	
X	X	Neg-edge	No Change	

b. Write the Verilog/ VHDL code for D FF with positive-edge triggering. Simulate and verify it's working.

```

Entity DFF is
    Port (D, CLK: IN STD_LOGIC;
          Q: OUT STD_LOGIC);
End DFF;
Architecture Behavioral of DFF is
    Begin
        Process (CLK)
            Begin
                If (CLK'event and CLK = '1') then
                    Q <= D;
                End if;
            End Process;
        End Behavioral;
    
```

Lab Experiment No: 06**Date:****J-K FLIP FLOPS****Aim:**

8. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
- b) Design and develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering.

Simulate and verify it's working.

Components:

Sl. NO	Particulars	Specification	Quantity
1.	Nand gates	7410,7400	As per Requirements
2.	Digital IC Trainer Kit	----	1
3.	Patch cords	----	20

Theory:

A **flip-flop** or latch is a circuit that has two stable states and can be used to store state information. A **flip-flop** is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. JK flip-flop provides the solution for SR flip-flop problem. Compared to SR flip-flop, JK flip-flop has two new connections from the Q and \bar{Q} outputs back to the original input gates. JK flip-flop behaves like the SR flip-flop except for input condition 1 and 1. Its output toggles for every clock pulse input unlike SR flip-flop. Although JK flip-flop circuit is an improvement on the clocked SR flip-flop it still suffers from timing problems called "race". This problem can be solved by Master-slave flip-flop.

The Master-slave JK flip-flop is basically two JK bitable flip-flops connected together in a series configuration with the outputs from Q and Q' from the slave flip-flop being fed back to the inputs of the Master with the outputs of the Master flip-flop being connected to the two inputs of the slave flip-flop. The circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal. In other words, the Master-Slave JK flip-flop is a "Synchronous" device as it only passes data with the timing of the clock signal.

Procedure:

1. Verify all the components and patch cords for their 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:

JK FLIP FLOP USING NAND GATES VHDL CODE

```
Entity JKFF is
  Port (J, K, CLK: IN STD_LOGIC;
        Q: OUT STD_LOGIC);
End JKFF;
Architecture Behavioral of JKFF is
  Begin
    Process (CLK)
      Variable X: STD_LOGIC;
    Begin
      If (CLK'event and CLK = '1') then
        If (J = '0' and K = '0') then
          X = X;
        elseif (J = '1' and K = '1') then
          X = Not X;
        elseif (J = '0' and K = '1') then
          X = 0;
        Else
          X = '1';
        End if;
      End if;
    End Process;
  End Behavioral;
```

Staff Signature and date:**Student Signature and Date**

INTENTIONALLY LEFT BLANK

CSE

Steps To Convert Grey to Binary Code:

Following steps can make your idea clear on this type of conversions.

- (1) The M.S.B of the binary number will be equal to the M.S.B of the given gray code.
- (2) Now if the second gray bit is 0 the second binary bit will be same as the previous or the first bit. If the gray bit is 1 the second binary bit will alter. If it was 1 it will be 0 and if it was 0 it will be 1.
- (3) This step is continued for all the bits to do **Gray code to binary conversion**.



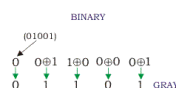
In the above example the M.S.B of the binary will be 0 as the M.S.B of gray is 0. Now move to the next gray bit. As it is 1 the previous binary bit will alter i.e. it will be 1, thus the second binary bit will be 1. Next look at the third bit of the gray code. It is again 1 thus the previous bit i.e the second binary bit will again alter and the third bit of the binary number will be 0. Now, 4th bit of the given gray is 0 so the previous binary bit will be unchanged, i.e. 4th binary bit will be 0. Now again the 5th grey bit is 1 thus the previous binary bit will alter, it will be 1 from 0. Therefore the equivalent Binary number in case of gray code to binary conversion will be (01001)

Steps To Convert Binary to Gray Code:

Steps given below elaborate on the idea on this type of conversion.

- (1) The M.S.B. of the gray code will be exactly equal to the first bit of the given binary number.
- (2) Now the second bit of the code will be exclusive-or of the first and second bit of the given binary Number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.
- (3) The third bit of gray code will be equal to the exclusive-or of the second and third bit of the given Binary number.

One example given below can make your idea clear on this type of conversion. Now concentrate on the example where the M.S.B. of the binary is 0 so for it will be 0 for the most significant gray bit. Next, the XOR of the first and the second bit is done. The bits are different so the resultant gray bit will be 1. Again move to the next step, XOR of second and third bit is again 1 as they are different. Next, XOR of third and fourth bit is 0 as both the bits are same. Lastly the XOR of fourth and fifth bit is 1 as they are different. That is how the result of binary to gray code conversion of **01001** is done whose equivalent gray code is **01101**.



Lab Experiment No: 07**Date:****CODE CONVERTERS****Aim:****Design and implement code converter I) Binary to Gray II) Gray to Binary Code using basic gates.****Components:**

Sl. NO	Particulars	Specification	Quantity
1.	EX-OR Gates	7486	As per Requirements
2.	Digital IC Trainer Kit	----	1
3.	Patch cords	----	20

Theory:

A symbolic representation of data/ information is called code. Ex: Binary Codes, Excess-3 Codes, Grey.

The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

GRAY CODES are non weighted codes that cannot be provided a weight to calculate their equivalent in decimal. Gray codes are often called reflected binary code. In this codes while traversing from one step to another step only one bit in the code group changes. In case of **Gray Code** two adjacent code numbers differs from each other by only one bit.

The base or radix of the binary number is 2. Hence, it has two independent symbols. The symbols used are 0 and 1. A binary digit is called as a bit. A binary number consists of sequence of bits, each of which is either a 0 or 1. Each bit carries a weight based on its position relative to the binary point. The weight of each bit position is one power of 2 greater than the weight of the position to its immediate right. e. g. of binary number is 100011 which is equivalent to decimal number 35.

Gray to Binary:

No	Grey				Binary			
	G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	0	0
7	0	1	1	1	0	1	0	1
8	1	0	0	0	1	1	1	1
9	1	0	0	1	1	1	1	0
10	1	0	1	0	1	1	0	0
11	1	0	1	1	1	1	0	1
12	1	1	0	0	1	0	0	0
13	1	1	0	1	1	0	0	1
14	1	1	1	0	1	0	1	1
15	1	1	1	1	1	0	1	0

Equations:

$$B_3 = G_3$$

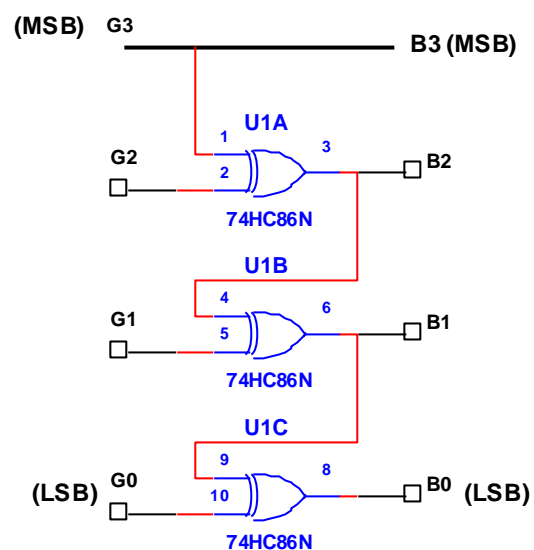
$$B_2 = G_3 \oplus G_2$$

$$B_1 = B_2 \oplus G_1$$

$$= (G_3 \oplus G_2) \oplus G_1$$

$$B_0 = B_1 \oplus G_0$$

$$= (G_3 \oplus G_2) \oplus (G_1 \oplus G_0)$$

**GREY TO BINARY CONVERTER**

Procedure:

1. Connections are made as shown in the block diagram.
2. Connect the circuit as per the Grey to Binary conversion.
3. Verify the truth table.
4. Connect the circuit as per the Binary to Grey conversion.
5. Verify the truth table.

CSE

Binary to Grey Converter:

No	Binary				Grey			
	B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

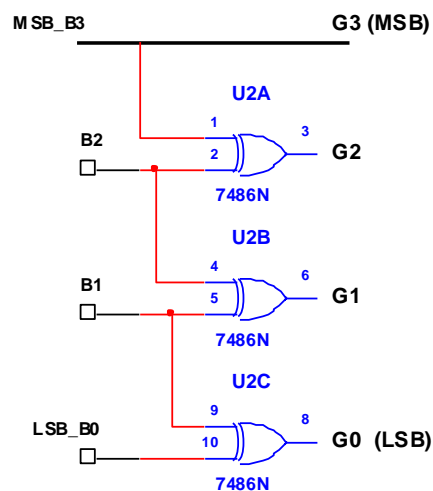
Equations:

$G3 = B3$

$G2 = B3 \oplus B2$

$G1 = B2 \oplus B1$

$G0 = B1 \oplus B0$

**BINARY TO GREY CONVERTER**

Results:

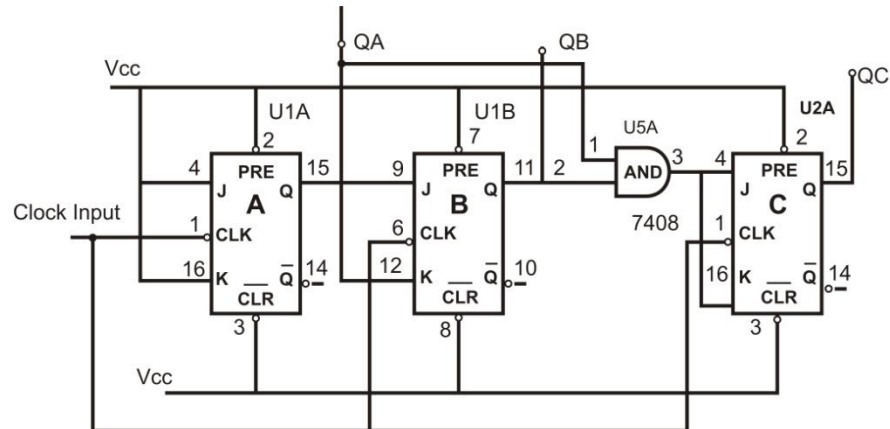
1. Truth Table verified for:
2. Truth Table verified for:

Assignment:

1. Realize the expression for the grey code and Binary code and simplify the same using K-Map.

Staff Signature and date:

Student Signature and Date

Circuit diagram of Mod – 8 counter: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

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

JA = 1

		KA			
		QA	QB	QC	QD
QB	00	X	1	1	X
	01	X	1	1	X
QC	11	X	1	1	X
	10	X	1	1	X

KA = 1

Lab Experiment No: 08**Date:****COUNTERS**

- 9 a) Design and implement a mod-n ($n < 8$) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
- b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.

Components:

Sl.No	Particulars	Range/Specification	Qty
1	JK flip-flop IC	7476	2
2	2 input Nand IC	7400	1
3	2 input And IC	7408	1
4	Digital IC trainer kit	---	1
5	Patch cords	---	20

Theory:

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

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.

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.

Procedure:

1. Verify all the components and patch cords for their 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.

JB

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

JB = QA

KB

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

KB = QA

JC

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

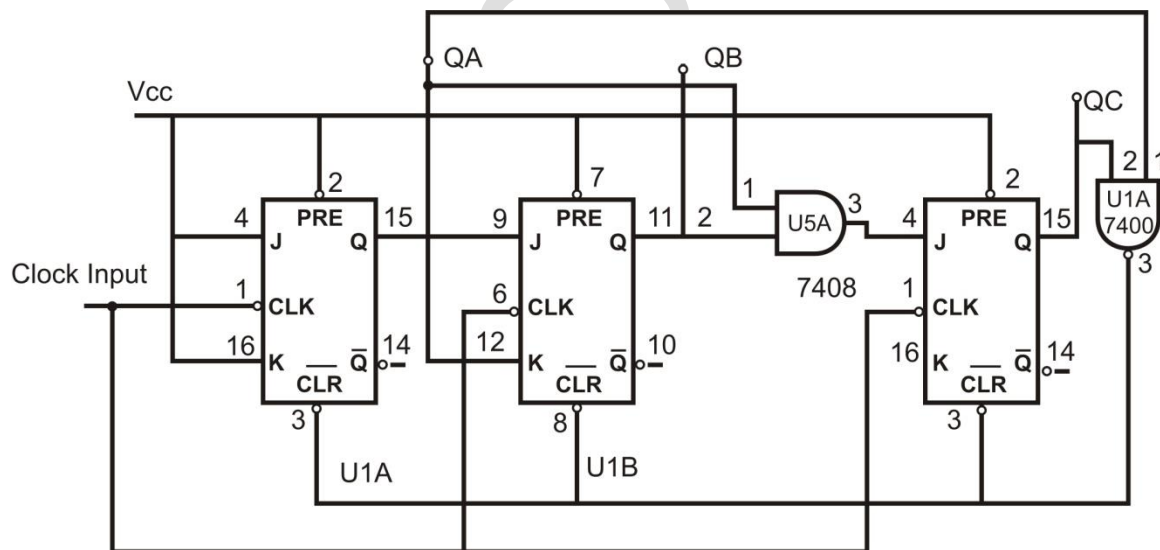
JC = QA QB

KC

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

KC = QA QB

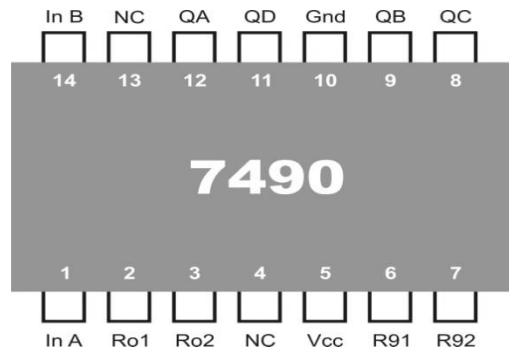
Mod-5 Circuit Diagram:



b. Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.

```
Entity Mod8    is
    Port (CLK, CLR: IN STD_LOGIC;
          Q: INOUT STD_LOGIC_VECTOR (2 DOWNTO 0):= "000");
End Mod8;
Architecture Behavioral of Mod8 is
    Begin
        Process (CLK)
            Begin
                If (CLK'event and CLK = '1') then
                    Q <= Q + 1;
                End if;
            End if;
        End Process;
    End Behavioral;
```

Result:**Staff Signature and date:****Student Signature and Date**

PIN DIAGRAM:FUNCTIONAL TABLE:

INPUTS				OUTPUTS			
R01	R02	R91	R92	QD	QC	QB	QA
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
L	X	H	H	H	L	L	H
X	L	H	H	H	L	L	H
L	X	L	X	COUNT			
X	L	X	L	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

Lab Experiment No: 09**Date:****DECADE COUNTER****Aim:**

- A. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ($n \leq 9$) and demonstrate on 7-segment display (using IC-7447).

Components:

Sl. NO	Particulars	Specification	Quantity
1.	Decade counter IC	7490	1
2.	Trainer Kit	----	1
3.	Patch cords	----	20

Theory:

A BCD counter is a special type of a digital counter which can count to ten on the application of a clock signal. In asynchronous counter a clock signal is provided for one flip-flop and its output is provided as clock source for next flip-flop. The output of asynchronous counter is not synchronized with clock signal.

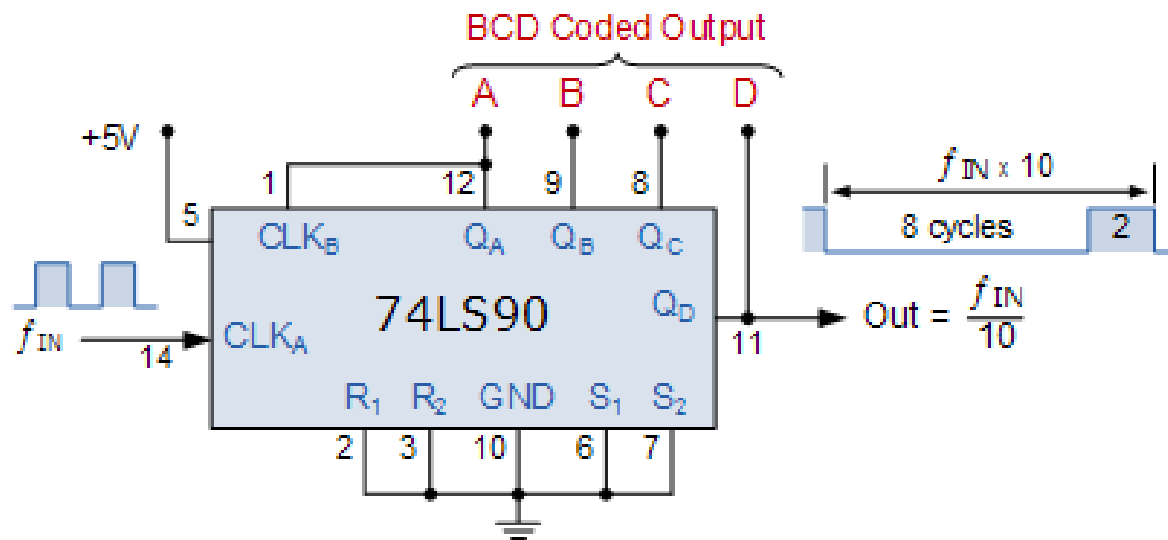
The 7490 is an asynchronous decade counter, able to count from 0 to 9 cyclically, and that is its natural mode. To make 7490 to work in normal mode the pin numbers 2, 3, 6, and 7 should hold at Low state. QA, QB, QC, QD are 4 output pins which gives the binary value of the decimal count. Pin 14 is Clock input.

Pin 2 and 3: Set inputs. They held to Low to activate 7490 IC as decade counter. At any instant of time, if they provide High signal then the output will hold at Low state until Pin 2 and 3 brought to Low voltage.

Pin 6 and 7: Clear inputs. At any instant of time, if they provide High signal then the output will hold at High state until Pin 6 and 7 brought to Low voltage.

Procedure:

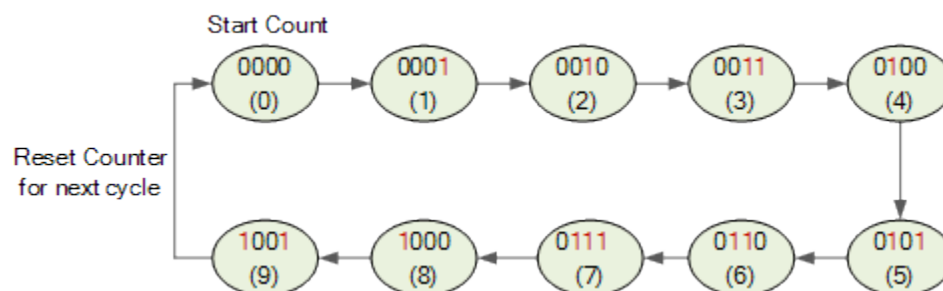
1. Verify all the components and patch cords for their good working condition.
2. Connect the reset terminals to high and set terminals to low and observe the output.
3. And now make connection as shown in the circuit diagram.
4. Give supply to the trainer kit and verify the truth table.

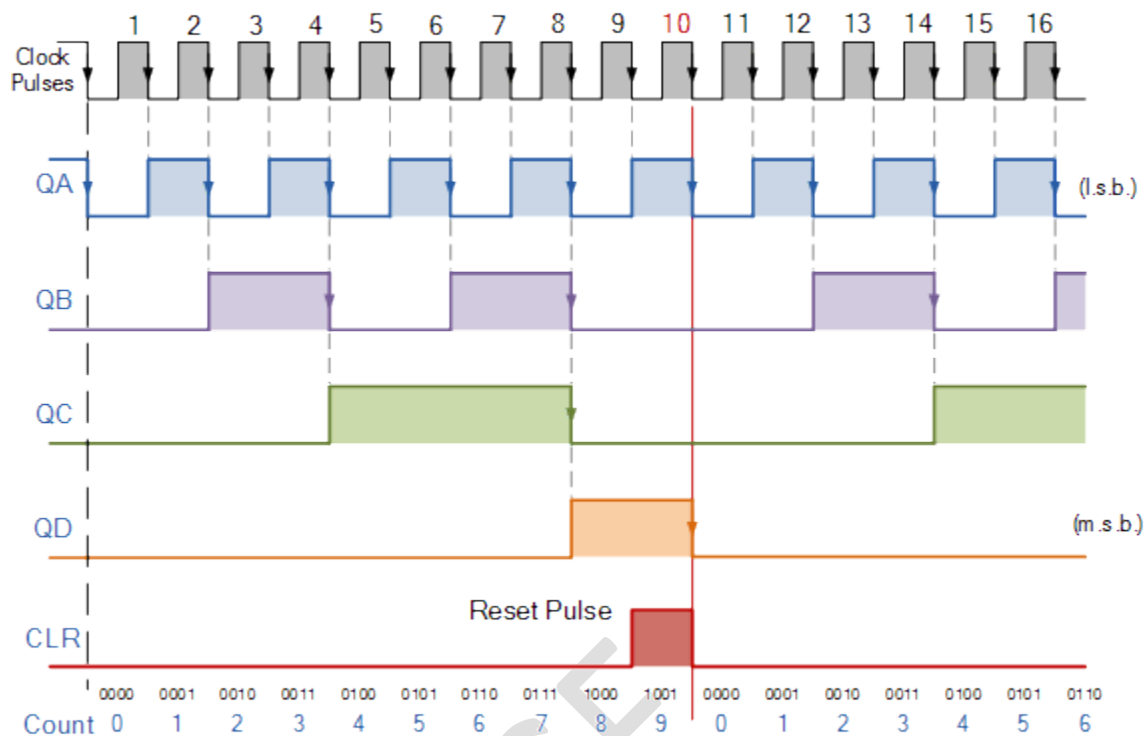


Truth table:

Clk	Q _D	Q _C	Q _B	Q _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
10	REPEATS			

BCD Counter State Diagram



Timing Diagram:

Then to summarize some of the advantages of Asynchronous Counters:

- **Asynchronous Counters** can easily be made from Toggle or D-type flip-flops.
- They are called “Asynchronous Counters” because the clock input of the flip-flops are not all driven by the same clock signal.
- Each output in the chain depends on a change in state from the previous flip-flops output.
- Asynchronous counters are sometimes called ripple counters because the data appears to “ripple” from the output of one flip-flop to the input of the next.
- They can be implemented using “divide-by-n” counter circuits.
- Truncated counters can produce any modulus number count.

Disadvantages of Asynchronous Counters:

- An extra “re-synchronizing” output flip-flop may be required.
- To count a truncated sequence not equal to 2^n , extra feedback logic is required.
- Counting a large number of bits, propagation delay by successive stages may become undesirably large.

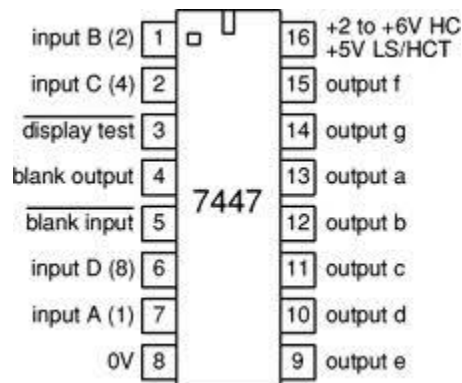
To demonstrate on 7-segment display (using IC-7447)

The 74LS47 display decoder receives the BCD code and generates the necessary signals to activate the appropriate LED segments responsible for displaying the number of pulses applied. As the 74LS47 decoder is designed for driving a common-anode display, a LOW (logic-0) output will illuminate an LED segment while a HIGH (logic-1) output will turn it "OFF". For normal operation, the LT (Lamp test), BI/RBO (Blanking Input/Ripple Blanking Output) and RBI (Ripple Blanking Input) must all be open or connected to logic-1 (HIGH).

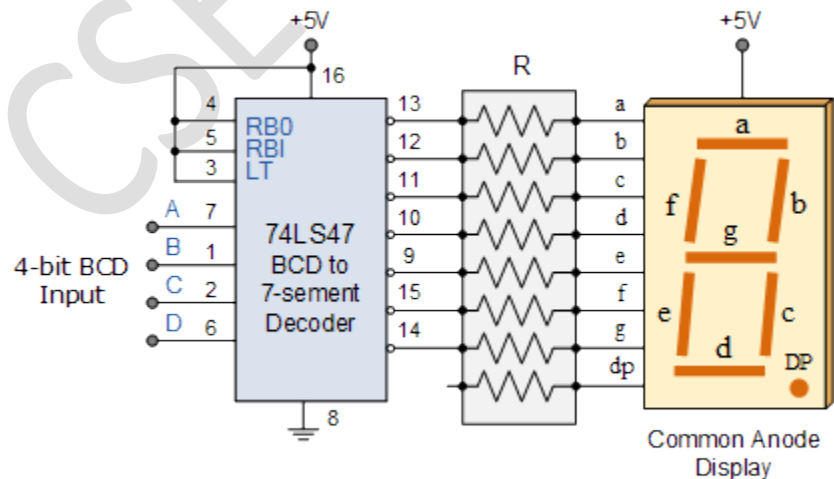
Note that while the 74LS47 has active LOW outputs and is designed to decode a common anode 7 segment LED display, the 74LS48 decoder/driver IC is exactly the same except that it has active HIGH outputs designed to decode a common cathode 7 segment displays. So depending upon the type of 7-segment LED display you have you may need a 74LS47 or a 74LS48 decoder IC.

The 74LS47 binary coded decimal inputs can be connected to the corresponding outputs of the 74LS90 BCD Counter to display the count sequence on the 7-segment display as shown each time the pushbutton SW1 is pressed. By changing the position of the pushbutton and 10k Ω resistor, the count can be made to change on the activation or release of the pushbutton switch, SW1.

PIN OUT OF 7447 BCD TO SEVEN SEGMENT CONVERTER



Final 4-bit BCD Counter Circuit:



Result:

Staff Signature and date:

Student Signature and Date

QUESTION BANK:**Lab Experiment 1. Schmitt Trigger**

1. What is a Schmitt trigger? Explain its operation.
2. Mention the advantages of Schmitt trigger and its area of application.
3. What is hysteresis curve? Explain
4. Why the Schmitt trigger is called as regenerative comparator?
5. Design a Schmitt trigger for the UTP = 8V and LTP = - 5V.

Lab Experiment 2. Relaxation Oscillator

1. What is an Oscillator? Mention the different types of oscillator.
2. Define a Relaxation Oscillator? Explain its operation.
3. Design a Relaxation Oscillator for a frequency of 4 KHz.

Lab Experiment 3. Astable Multivibrator

1. What is a multivibrator? What is the purpose of multivibrator? What are its types?
2. What is an astable multivibrator called so?
3. What is the disadvantage of an astable multivibrator?
4. Explain the working of Astable Multivibrators?

Lab Experiment 4. Adders

1. Give the basic rules for binary addition?
2. What is a half adder and what is its disadvantage?
3. Derive the expressions for Sum and carry of a half adder.
4. What is a full adder and mention its advantages?
5. Derive the expressions for Sum and carry of a full adder and realize it using NAND gates.
6. Write an expression for Half and full Subtractor by using K-Map and realize the expression using NAND gates.
7. Write the Verilog/VHDL code for Full adder, half adder, Half Subtractor and Full Subtractor.

Lab Experiment 5. Multiplexer

1. What is a multiplexer? Why it is called as Data Selector?
2. What is the importance of the enable Pin for any IC?
3. What is the role of Select line in a multiplexer? How do we decide the number of select lines?

4. What is the advantage of VEM technique? Explain with an example.
5. What are the different approaches for the realization of a Boolean Expression?

Lab Experiment 6. Code Converters

1. Explain the concept of Grey Code.
2. What are the different types of Codes that are normally used?
3. Use the weighting factors to convert the following BCD numbers to binary.

0101 0011 0010 0110 1000

4. Explain the concept of Excess-3 and Binary Codes?
5. What is the difference between weighted and Non Weighted Code? Explain with Example?

Lab Experiment 7. Parity generator and Parity Checker

1. Define Parity Generator and Parity Checker.
2. What is parity generation? What are its advantages?
3. Explain the different types of parity generators?
4. Implement the parity generator (a) Even (b) Odd for 4-bit message

Lab Experiment 8. Flip-Flops

1. What is the difference between a Latch and a Flip-Flop?
2. What are the different types of Flip Flips?
3. What is Race around condition in JK Flip Flop?
4. What is the drawback of RS Flip Flop? How it is over come?
5. Explain the working of MS JK Flip-Flop.

Lab Experiment 9 and 10. Counters

1. Define a counter?
2. What are the different types of counters? Explain
3. Design a Mod-5 Counter using JK Flip-Flops?
4. What is a BCD Counter?
5. What is the meaning of Modulus in a counter?
6. Why Asynchronous counter is known as ripple counter?
7. What is the importance of State diagrams?
8. What is a BCD counter?
9. Mention the difference between 7490 and 7492.

REFERENCE BOOKS:

1. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design With VHDL, 2nd edition, TATA McGraw Hill, 2005.
2. Fundamentals of Digital Logic with Verilog Design, Stephen Brown, Zvonko Vranesic, TMH, 2006.
3. Jacob Millman, Christos Halkias, Chetan D Parikh: Millman's Integrated Electronics Analog and Digital Circuits and Systems, 2nd Edition, Tata McGraw Hill, 2010.
4. R. D. Sudhaker Samuel: Electronic Circuits, Sanguine-Pearson, 2010.
5. Electronic Principles, Albert Malvino & David J Bates, 7th Edition.
6. Electronic Devices and Circuit Theory, Robert L. Boylestad, Louis Nashelsky, 9th Edition
7. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, 2009
8. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 7th Edition, Tata McGraw Hill, 2010.
9. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson Education
10. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine- Pearson, 2010.

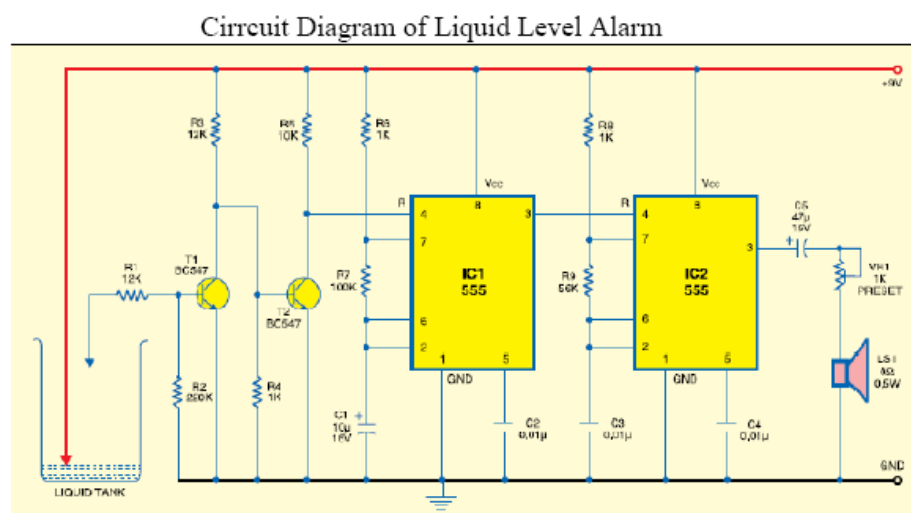
MINI PROJECT: Analog Electronics

1. Aim: Liquid Level Alarm:

APPARATUS REQUIRED: Components as shown in the circuit diagram(such as 555IC,, soldering iron and solder flux and PCB board.

BRIEF THEORY: Here is a simple circuit for (T1 and T2) and two timers 555 ICs (IC1 and IC2). Both IC1 and IC2 are wired in astable multivibrator mode. Timer IC1 produces low frequency, while timer IC2 produces high frequency. As a result, a beeping tone is generated when the liquid tank is full. Initially, when the tank is empty, transistor T1 does not conduct. Consequently, transistor T2 conducts and pin 4 of IC1 is low. This low voltage disables IC1 and it does not oscillate. The low output of IC1 disables IC2 and it does not oscillate. As a result, no sound is heard from the speaker. But when the tank gets filled up, transistor T1 conducts. Consequently, transistor T2 is cut off and pin 4 of IC1 becomes high. This high voltage enables IC1 and it oscillates to produce low frequencies at pin 3. This low-frequency output enables IC2 and it also oscillates to produce high frequencies. As a result, sound is produced from the speaker. Using preset VR1 you can control the volume of the sound from the speaker. The circuit can be powered from a 9V battery or from mains by using a 9V power adaptor.

CIRCUIT DIAGRAM:

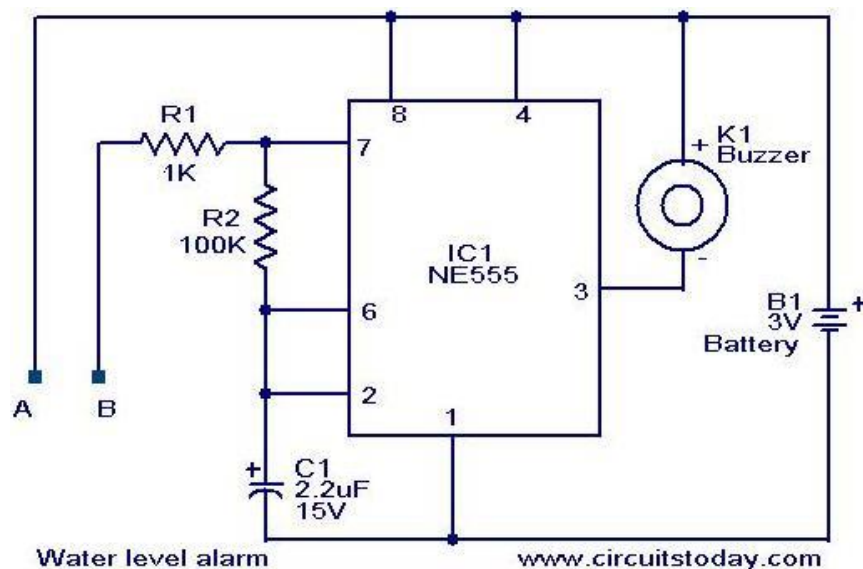


PROCEDURE: Assemble the circuit on a general purpose PCB and enclose in a suitable cabinet. Install two water-level probes using metal strips such that one touches the bottom of the tank and the other touches the maximum level of the water in the tank. Interconnect the sensor and the circuit using a flexible wire.

PRECAUTIONS:

- 1) Make the connections according to the Circuit diagram using soldering iron
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only

OR



Here is a simple water level alarm circuit using 555 timer that will produce an audible alarm when the water level reaches a preset level. The circuit can be powered of a 3V battery and is very handy to use.

The circuit is based on an astable multivibrator wired around IC1 (NE 555). The operating frequency of the astable multivibrator here will depend on capacitor C1, resistances R1, R2 and the resistance across the probes A & B. When there is no water up to the probes, they will be open and so the multivibrator will not produce oscillations and the buzzer will not beep. When there is water up to the level of probes, some current will pass through the water, the circuit will be closed to some extent, and the IC will start producing oscillations in a frequency proportional to the value of C1, R1, R2 and the resistance of water across the probes. The buzzer will beep to indicate the presence of water up to the level of the sensing probes.

Mini Project Digital Electronics:**1. NAND Gate circuit using Transistors.**

Let's make NAND Gate circuit using Transistors. In this Project NAND gate is made by combining AND and NOT Gate. Here first two transistor in series is used for AND gate and the last transistor makes NOT Gate. Both is connected by a 10K resistance. Output data of NOT gate is given to LED for indication.

Part List

Transistor = BC548 or any general purpose NPN Transistor will work.

LED (Light Emitting Diode) any Color except white.

Resistance = 10K-3Pcs, 270k and 220 ohms.

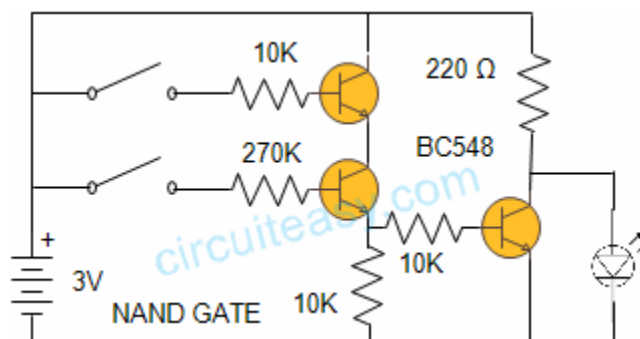
2 Pcs slide or SPST Switch

1 battery container for connecting 2 cell.

This circuit works well and consumes only 3 Volt.

There is no power switch in the circuit so battery should be removed when not in use.

Circuit Diagram of NAND GATE is given below:-



Other logic gate such as NOT, AND, OR can also be created by NAND Gates. So you can say NAND as Universal gate. This is also true of NOR gates.