S

Estd.2001

Sri Indu

College of Engineering & Technology

UGC Autonomous Institution

Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

NAAC

NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

DIGITAL ELECTRONICS LABORATORY (R22ECE2126)

LAB MANUAL

II Year I Semester

DEPARTMENT OF INFORMATION TECHNOLOGY

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

B. TECH –INFORMATION TECHNOLOGY

INSTITUTION VISION

To be a premier Institution in Engineering & Technology and Management with competency, values and social consciousness.

INSTITUTION MISSION

IM$_1$ Provide high quality academic programs, training activities and research facilities.

IM$_2$ Promote Continuous Industry-Institute Interaction for Employability, Entrepreneurship, Leadership and Research aptitude among stakeholders.

IM$_3$ Contribute to the Economical and technological development of the region, state and nation.

DEPARTMENT VISION

To be a recognized knowledge centre in the field of Information Technology with self - motivated, employable engineers to society.

DEPARTMENT MISSION

The Department has following Missions:

DM$_1$ To offer high quality student centric education in Information Technology.

DM$_2$ To provide a conducive environment towards innovation and skills.

DM$_3$ To involve in activities that provide social and professional solutions.

DM$_4$ To impart training on emerging technologies namely cloud computing and IOT with involvement of stake holders.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: **Higher Studies:** Graduates with an ability to apply knowledge of Basic sciences and programming skills in their career and higher education.

PEO2: **Lifelong Learning:** Graduates with an ability to adopt new technologies for ever changing IT industry needs through Self-Study, Critical thinking and Problem solving skills.

PEO3: **Professional skills:** Graduates will be ready to work in projects related to complex problems involving multi-disciplinary projects with effective analytical skills.

PEO4: **Engineering Citizenship:** Graduates with an ability to communicate well and exhibitsocial, technical and ethical responsibility in process or product.

PROGRAM OUTCOMES (POs) & PROGRAM SPECIFIC OUTCOMES (PSOs)

PO	Description
PO 1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO 2	**Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO 3	**Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO 4	**Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO 5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO 6	**The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO 7	**Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO 8	**Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO 10	**Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO 11	**Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)
	Program Specific Outcomes
PSO 1	**Software Development:** To apply the knowledge of Software Engineering, Data Communication, Web Technology and Operating Systems for building IOT and Cloud Computing applications.
PSO 2	**Industrial Skills Ability:** Design, develop and test software systems for world-wide network of computers toprovide solutions to real world problems.
PSO 3	**Project implementation:** Analyze and recommend the appropriate IT Infrastructure required for the implementationof a project.

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time),those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with thesynopsis / program / experiment details.
3. Student should enter into the laboratory with:
   a) Laboratory observation notes with all the details (Problem statement, Aim, Algorithm,Procedure, Program, Expected Output, etc.,) filled in for the lab session.
   b) Laboratory Record updated up to the last session experiments and other utensils (if any)needed in the lab.
   c) Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation notebook, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain thediscipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the labafter completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Head of the Department **Principal**

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

B.Tech. - II Year – I Semester

L	T	P	C
0	0	2	1

(R22ECE2126) DIGITAL ELECTRONICS LABORATORY

Course Outcomes: Upon completing this course, the students will be able to

1. Acquire the knowledge on numerical information in different forms and Boolean Algebratheorems.
2. Define Postulates of Boolean algebra and to minimize combinational functions, and designthe combinational circuits.
3. Design and analyze sequential circuits for various cyclic functions.
4. Characterize logic families and analyze them for the purpose of AC and DC parameters.

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	1	2	1	-	-	1	-	-	2
CO2	3	2	2	1	2	1	-	-	1	-	-	2
CO3	2	3	3	2	2	1	-	-	1	-	-	1
CO4	3	2	1	1	1	-	-	-	-	-	-	-

List of Experiments

1. Realization of Boolean Expressions using Gates
2. Design and realization logic gates using universal gates
3. Design a 4 – bit Adder
4. Design a 4 – bit Subtractor
5. Design and realization a 4 – bit gray to Binary and Binary to Gray Converter
6. Design and realization of a 4-bit pseudo random sequence generator using logic gates.
7. Design and realization of an 8-bit parallel load and serial out shift register using flip-flops.
8. Design and realization Asynchronous and Synchronous counters using flip-flops
9. Design and realization 8x1 using 2x1 mux
10. Design and realization 4-bit comparator
11. Verification of truth tables and excitation tables
12. Realization of logic gates using DTL, TTL, ECL, etc.,

1. **Major Equipment required for Laboratories:**
    1. 5 V Fixed Regulated Power Supply/ 0-5V or more Regulated Power Supply.
    2. 20 MHz Oscilloscope with Dual Channel.
    3. Bread board and components/ Trainer Kit.
    4. Multimeter.

LIST OF EXPERIMENTS

1. Realization of Boolean Expressions using Gates
2. Design and realization logic gates using universal gates(NAND/NOR)
3. Design a 4 – bit Adder
4. Design a 4 – bit Subtractor
5. Design and realization of a 4 – bit gray to Binary and Binary to Gray Converter
6. Design and realization of 4-bit Pseudo Random sequence generator using logic gates
7. Design and realization of 8-bit parallel load and serial out shift register using flip-flops
8. Design and realization of a Synchronous and Asynchronous counter using flip-flops
9. Design and realization of 8x1 MUX using 2x1 MUX
10. Design and realization of 4-bit comparator
11. Verification of truth tables and excitation tables
12. Realization of logic gates using DTL, TTL, ECL, etc.

EXPERIMENT NO:1

Realization of Boolean Expressions using Gates

Aim: Implementation of the given Boolean function using logic gates in both sop and pos forms.

Apparatus Required: Logic gates trainer kit, logic gates / ICs, patch cords.

Circuit Diagram:

SOP FORM:

SOP: - It is the sum of the Products form in which the terms are taken as 1. It is denoted in the K-Map expression by the Sign summation (∑)

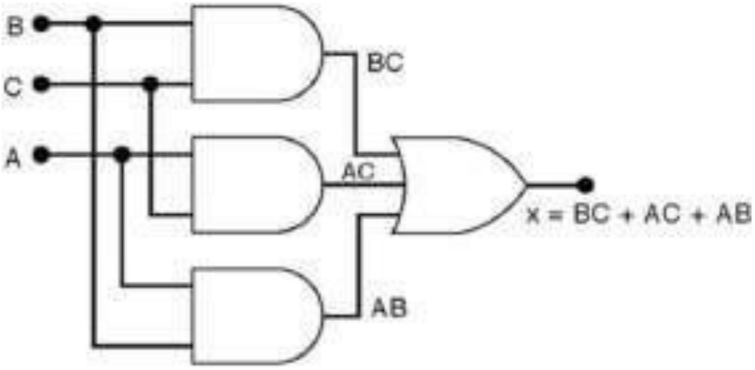


Fig:SOP Form

Truth Table:

A	B	C	AB	AC	BC	X
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

POS FORM:

POS: - It is the product of the sums form in which the terms are taken as 0. It is denoted in the K-Map expression by the Sign pie (π)

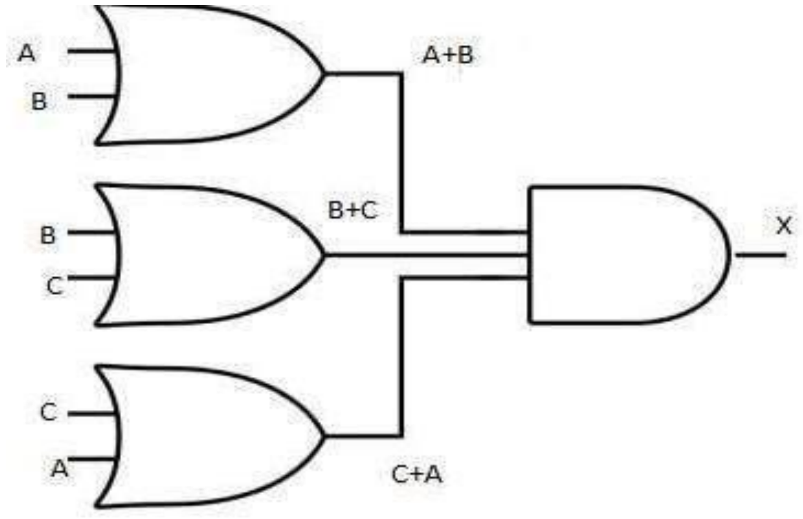


Fig: POS form

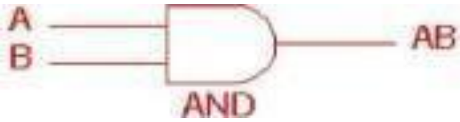
Truth Table:

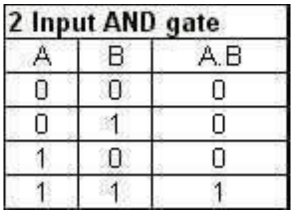
A	B	C	A+B	B+C	C+A	X
0	0	0	0	0	0	0
0	0	1	0	1	1	0
0	1	0	1	1	0	0
0	1	1	1	1	1	1
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Theory: Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/ output combination is called **Truth Table**. Various gates and their working is explained here.

AND Gate

AND gate produces an output as 1, when all its inputs are 1; otherwise the output is 0. This gate can have minimum 2 inputs but output is always one. Its output is 0 when any input is 0.





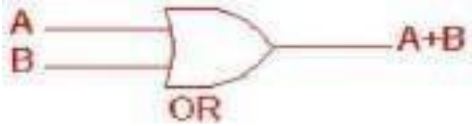
2 Input AND gate

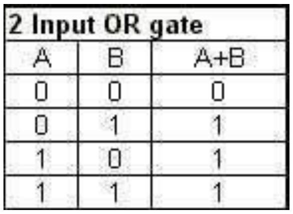
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

IC 7408

OR Gate

OR gate produces an output as 1, when any or all its inputs are 1; otherwise the output is 0. This gate can have minimum 2 inputs but output is always one. Its output is 0 when all input are 0.





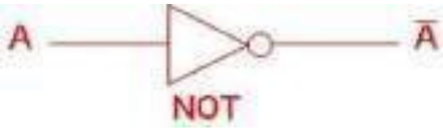
2 Input OR gate

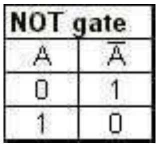
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

IC 7432

NOT Gate

NOT gate produces the complement of its input. This gate is also called an INVERTER. It always has one input and one output. Its output is 0 when input is 1 and output is 1 when input is 0.





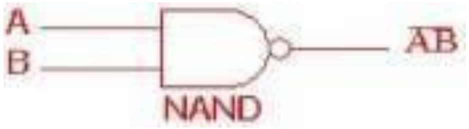
NOT gate

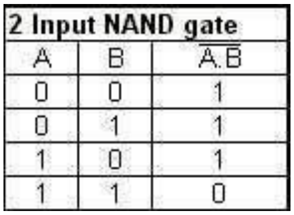
A	$\overline{A}$
0	1
1	0

IC 7404

NAND Gate

NAND gate is actually a series of AND gate with NOT gate. If we connect the output of an AND gate to the input of a NOT gate, this combination will work as NOT-AND or NAND gate. Its output is 1 when any or all inputs are 0, otherwise output is 1.



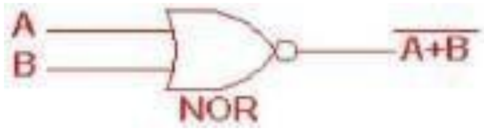


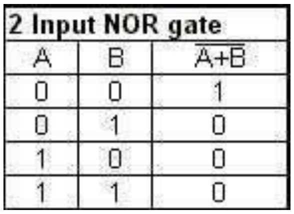
2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

IC 7400

NOR Gate

NOR gate is actually a series of OR gate with NOT gate. If we connect the output of an OR gate to the input of a NOT gate, this combination will work as NOT-OR or NOR gate. Its output is 0 when any or all inputs are 1, otherwise output is 1.



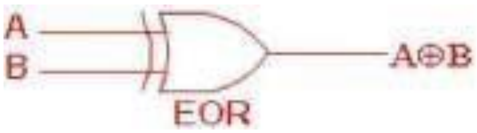


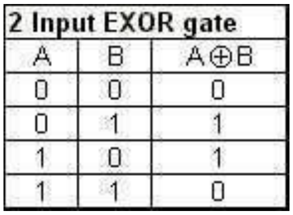
2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

IC 7402

Exclusive OR (X-OR) Gate

X-OR gate produces an output as 1, when number of 1's at its inputs is **odd**, otherwise output is 0. It has two inputs and one output.



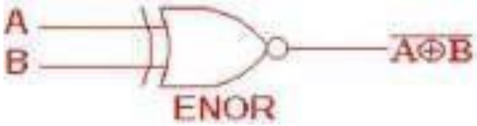


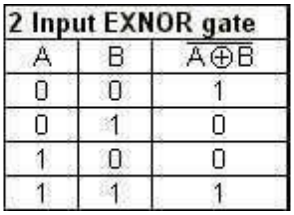
2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

IC 7486

Exclusive NOR (X-NOR) Gate

X-NOR gate produces an output as 1, when number of 1's at its inputs is **not odd**, otherwise output is 0. It has two inputs and one output.





2 Input EXNOR gate		
A	B	$\overline{A\oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the inputs of any one logic gate to the logic sources and its output to the logic indicator.
3. Apply various input combinations and observe output for each one.
4. Verify the truth table for each input/ output combination.
5. Repeat the process for all other logic gates.
6. Switch off the ac power supply.

RESULT: Realization of Boolean expression s by using logic gates are verified

VIVA:

1. What is the use of gates?
2. Compare all the gates?
3. Why we need ICs?
4. Define the term digital?
5. Which gate is equal to AND-inverter Gate?
6. What are the logic gates?
7. What is a truth table?
8. What is the maximum number of outputs a logic gate can have?
9. What is an AND gate?
10. What is a NOT gate?

Experiment No: 2

Design and realization of logic gates using universal gates

Aim:-To study the realization of basic gates using universal gates. Understanding how to construct any combinational logic function using NAND or NOR gates only.

Aparatus Required:

IC 7402(NOR), IC 7400(NAND), 7404(NOT), 7408(AND), 7432(OR), KL 33002, Power supply, connecting wires and Breadboard etc. or Trainer kit, patch cards.

Theory:

AND, OR, NOT are called basic gates as their logical operation cannot be simplified further.

NAND and NOR are called universal gates as using only NAND or only NOR, any logic function can be implemented. Using NAND and NOR gates and **De Morgan's Theorems** different basic gates & EX-OR gates are realized.

De Morgan`s Law:

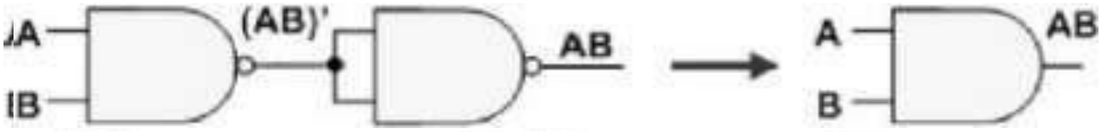
In formal logic, De Morgan's laws are rules relating the logical operators "AND" and "OR" in terms of each other via negation. With two operands A and B:

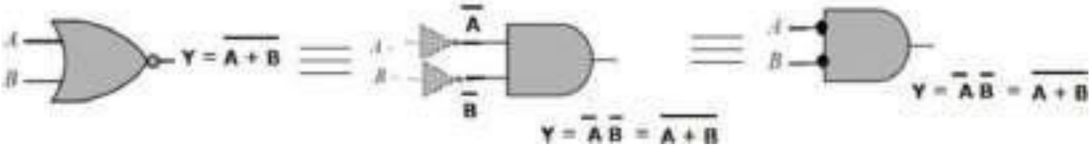
1. $A.B = A + B$

2. $A+B = A . B$

The NAND gate is equivalent to an OR gate with the bubble at its inputs which are as shown.

The NOR gate is equivalent to an AND gate with the bubble at its inputs which are as shown.



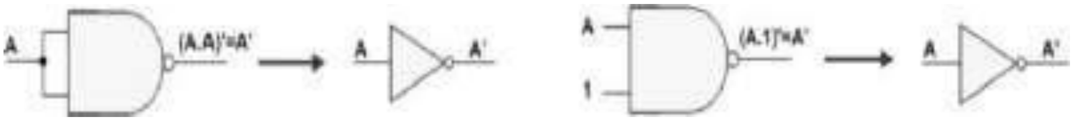

A
B
Y = $\overline{A+B}$
A
B
$\overline{A}$
$\overline{B}$
Y = $\overline{A}\,\overline{B}$ = $\overline{A+B}$
A
B
Y = $\overline{A}\,\overline{B}$ = $\overline{A+B}$

IMPLEMENTING INVERTER USING NAND GATE :

The figure shows two ways in which a NAND gate can be used as an inverter (NOT gate).

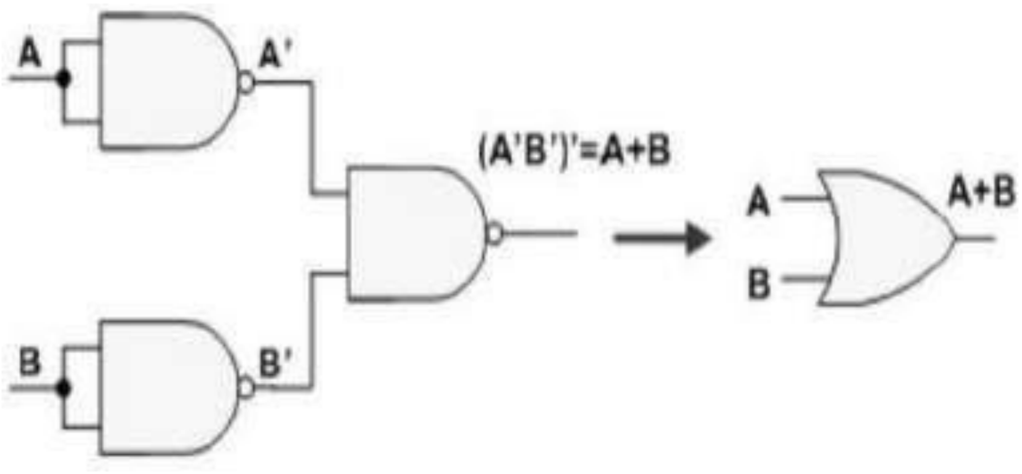
1. All NAND input pins connect to the input signal A gives an output A'.
2. One NAND input pin is connected to the input signal A while all other input pins are connected to logic 1. The output will be A'.

An AND gate can be replaced by NAND gates as shown in the figure (The AND is replaced by a NAND gate with its output complemented bya NAND gate inverter).



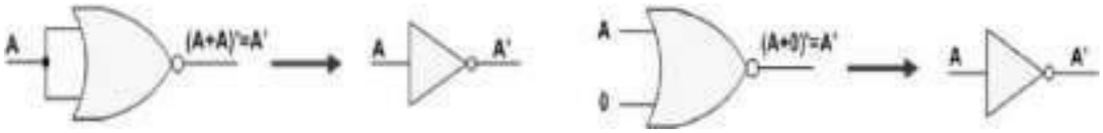
IMPLEMENTING OR USING NAND GATE :

An OR gate can be replaced by NAND gates as shown in the figure (The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters).



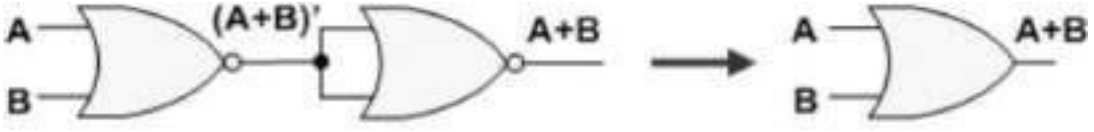
IMPLEMENTING INVERTER USING NOR GATE:

- The figure shows two ways in which a NOR gate can be used as an inverter (NOT gate).
- All NOR input pins connect to the input signal A gives an output A'.
- One NOR input pin is connected to the input signal A while all other input pins are connected to logic 0. The output will be A'.



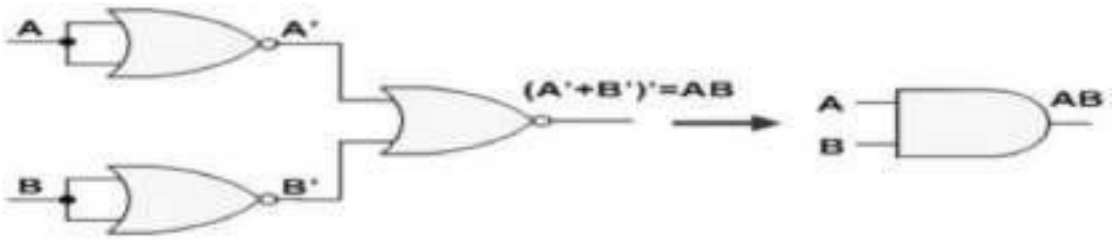
IMPLEMENTING OR USING NOR GATE:

- An OR gate can be replaced by NOR gates as shown in the figure (The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter)



IMPLEMENTING AND USING NOR GATE:

- An AND gate can be replaced by NOR gates as shown in the figure (The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters)



Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the NAND gates/NOR gates for any of the logic functions to be realized.
3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply various input combinations and observe output for each one.
5. Verify the truth table for each input/ output combination.
6. Repeat the process for all logic functions.
7. Switch off the ac power supply.

RESULT: Realization of logic gates with the help of Universal Gates-NAND AND NOR Gates are verified.

VIVA:

1. What do u mean by universal gate?

2. What is truth table?

3. How many AND gates are required to realize the following expression

Y=AB+BC?

4. What is the difference between X-OR and X-Nor Gates?

5.Write properties of X-OR gate?

6. What is universal gates explain?

7. Draw the truth table for NAND?
8. Which is not universal gate?
9.Why it's called universal gates?
10. Draw the truth table for NOR?

Experiment No: 3&4

Design a 4-bit Adder / Subtractor

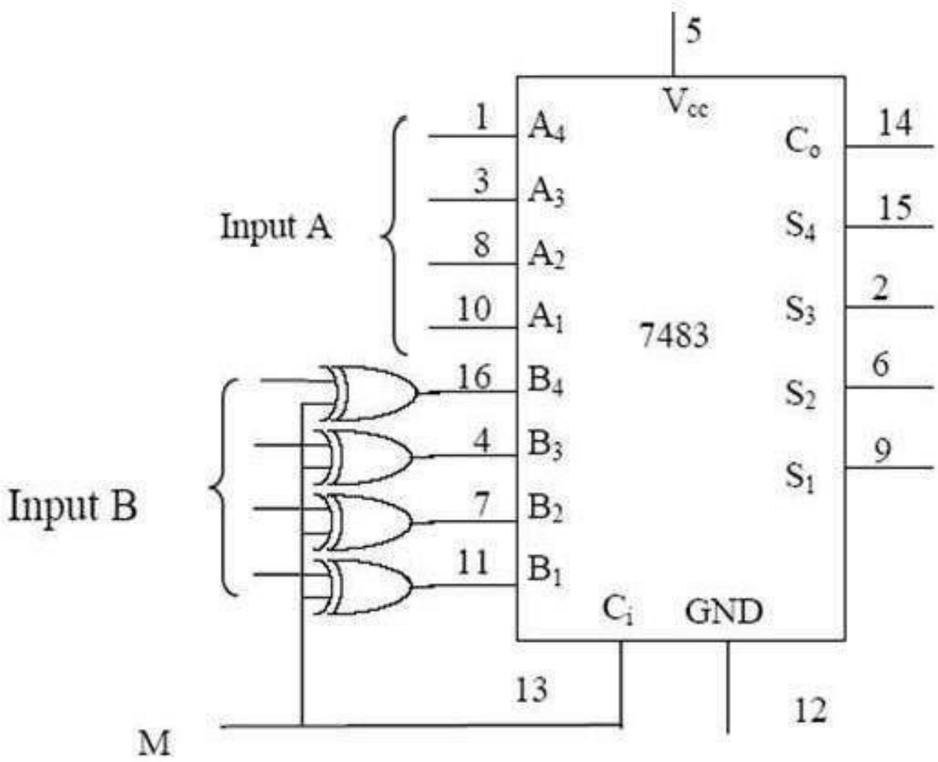
Aim: To verify a 4-bit Adder / Subtractor

Apparatus: Adder/subtractor logic trainer kit, patch cords

Theory:

In Digital Circuits, A **Binary Adder-Subtractor** is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds. It is one of the components of the ALU (Arithmetic Logic Unit).This Circuit Requires prerequisite knowledge of Exor Gate, Binary Addition and Subtraction, Full Adder.

Circuit diagram:



When M=0 addition is performed, when M=1 subtraction is performed

Truth table:

ADDITION:

Inputs								Outputs				
A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	C_0	S_4	S_3	S_2	S_1
1	0	0	0	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	1	0	1	1	0	1	1	1

SUBTRACTION:

Inputs								Outputs				
A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	B_0	S_4	S_3	S_2	S_1
1	0	0	0	0	0	1	0	0	0	1	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	1	0	1	1	1	1	0	1

Procedure:

1. Connect the circuit as per circuit diagram.
2. Connect the inputs to the switches and outputs to the logic indicators.
3. Apply the different combinations of inputs and observe the outputs.
4. Apply M=0 for addition operation and M=1 for subtraction operation.
5. Note down the values of c_{out} and sum in addition operation and difference and borrow in subtraction operation.

Result: 4-bit Adder / Subtractor is verified.

VIVA:

1. What is Adder?
2. What is Subtractor?
3. Define Overflow?
4. What is major difference between Half-Adder and Full-Adder?
5. What is the use of 4-bit adder?
6. Which IC is used as 4-bit binary subtractor?
7. How many inputs does a 4 -bit adder have?
8. Why adder is used?
9. What is full subtractor formula?
10. What is BCD adder?

Experiment No: 5

Design and realization a 4-bit gray to Binary and Binary to Gray Converter

Aim: To Design and realization of 4-bit gray to Binary and Binary to Gray Converter

Apparatus: binary to gray logic trainer kit, Patch cords

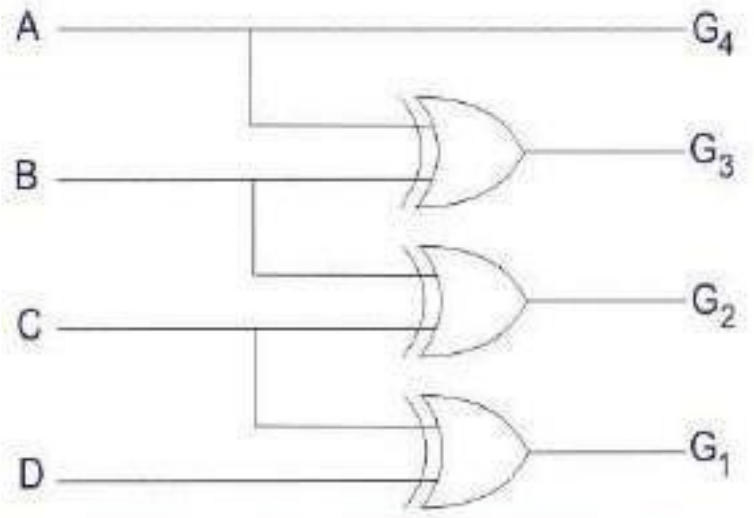
Theory:

Binary to Gray Code Converter

The logical circuit which converts binary code to equivalent gray code is known as **binary to gray code converter**. The gray code is a non weighted code. The successive gray code differs in one bit position only that means it is a unit distance code. It is also referred as cyclic code. It is not suitable for arithmetic operations. It is the most popular of the unit distance codes. It is also a reflective code. An n-bit Gray code can be obtained by reflecting an n-1 bit code about an axis after 2^{n-1} rows, and putting the MSB of 0 above the axis and the MSB of 1 below the axis. Reflection of Gray codes is shown below.

The 4 bits binary to gray code conversion table is given below

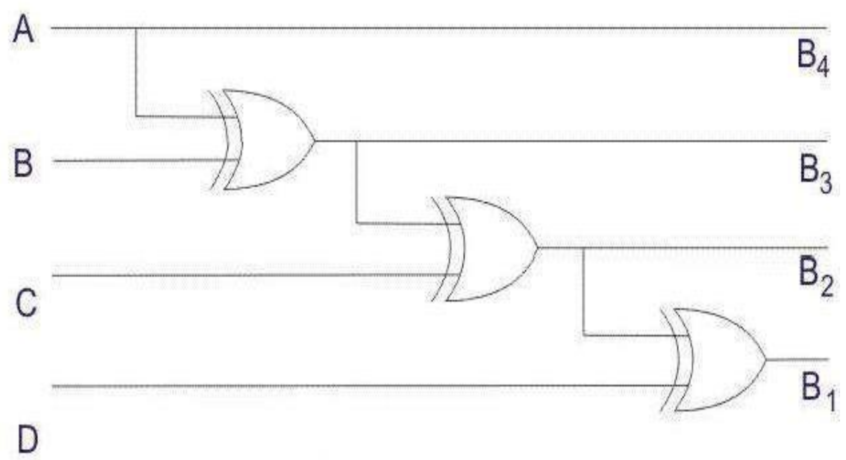
INPUTS				OUTPUTS			
BINARY CODE				GRAY CODE			
A	B	C	D	G1	G2	G3	G4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



Logic Circuit for Binary to Gray Code Converter

Grey to Binary conversion:

INPUTS				OUTPUTS			
BINARY CODE				GRAY CODE			
A	B	C	D	G1	G2	G3	G4
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



Logic Circuit for Gray to Binary Code Converter

Procedure:

1. Connect the circuit as per the circuit diagram.
2. Connect the inputs to the switch and outputs to the logic indicators.
3. Apply the different combinations of inputs and observe the outputs.

Result: 4-bit gray to Binary and Binary to Gray Converter is verified.

VIVA:

1. What is a code converter?
2. Give the primary use for Gray code
3. Convert binary number into gray code : 100101
4. Convert gray code into binary :101010

5.What is GRAY in binary?

6.Which gate converts binary to gray code?

7.How do u convert BCD to GRAY?

8.Why do we use gray code?

9.Which IC is used in gray to binary?

10. Why do we convert gray code to binary?

Experiment No: 6

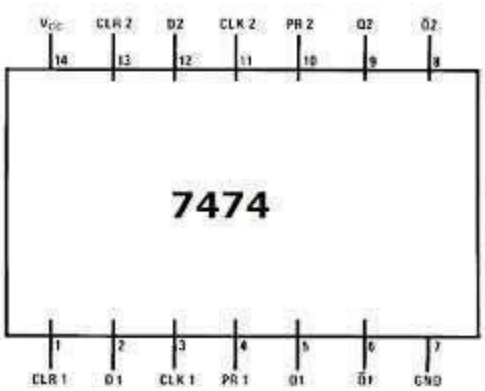
Design and realization of a 4-bit pseudo random sequence generator using logic gates.

Aim: To Design and realization of a 4-bit pseudo random sequence generator using logic gates.

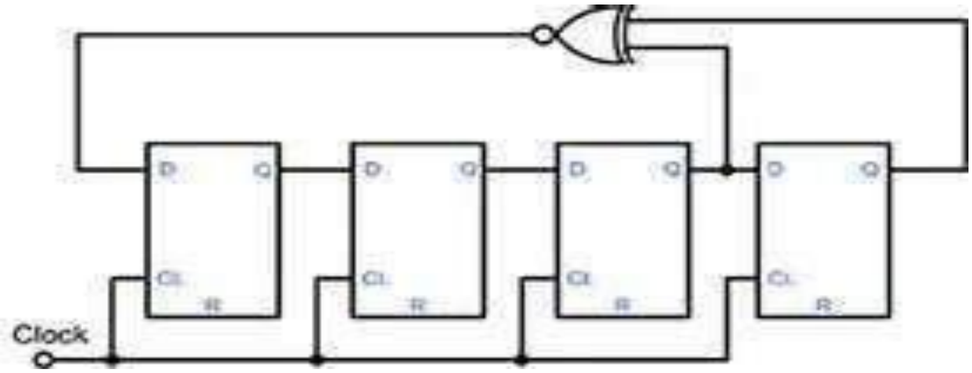
Apparatus: logic trainer kit, patch cords

Circuit diagram:

Pin Configuration



Circuit diagram:



Truth Table:

D	Q_A	Q_B	Q_C	Q_D
1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
0	1	1	1	0
1	0	1	1	1
1	1	0	1	1
0	1	1	0	1
0	0	1	1	0
1	0	0	1	1
0	1	0	0	1
1	0	1	0	0
0	1	0	1	0
0	0	1	0	1
0	0	0	1	0
0	0	0	0	1
1	0	0	0	0

Procedure:

1. Connect the circuit as per the circuit diagram.
2. Connect the XNOR gate output to the D input and inputs to XNOR gate are from third and fourth flip flop.
3. Connect outputs of all the flip flops to LED's.
4. Connect the clock and clear inputs as shown in diagram.
5. Observe the output on the LED's.

Result: Designed and realized a 4-bit pseudo random sequence generator using logic gates.

VIVA:

1.What are the 4 types of flip flops?
2. Sequence generator used for?
3.What's the sequence generator?
4.What is the logic gate theory?
5.What is pseudo random sequence?
6.Where is sequence used?
7.Is a sequence generator a type of circuit?
8.What are the 4 types of sequence?
9.What is the sequence formula?
10.Why is sequence important?

Experiment No: 7

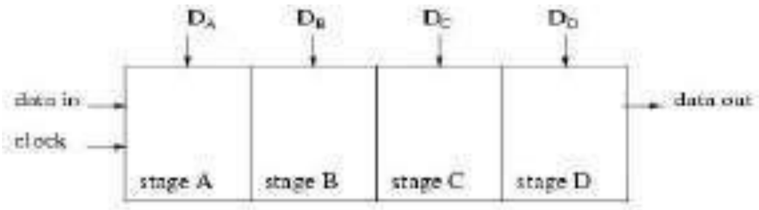
8-bit parallel load and serial out shift register using flip-flops.

Aim: To Design and realize an 8-bit parallel load and serial out shift register using flip-flops.

Apparatus: PISO trainer kit, patch cords

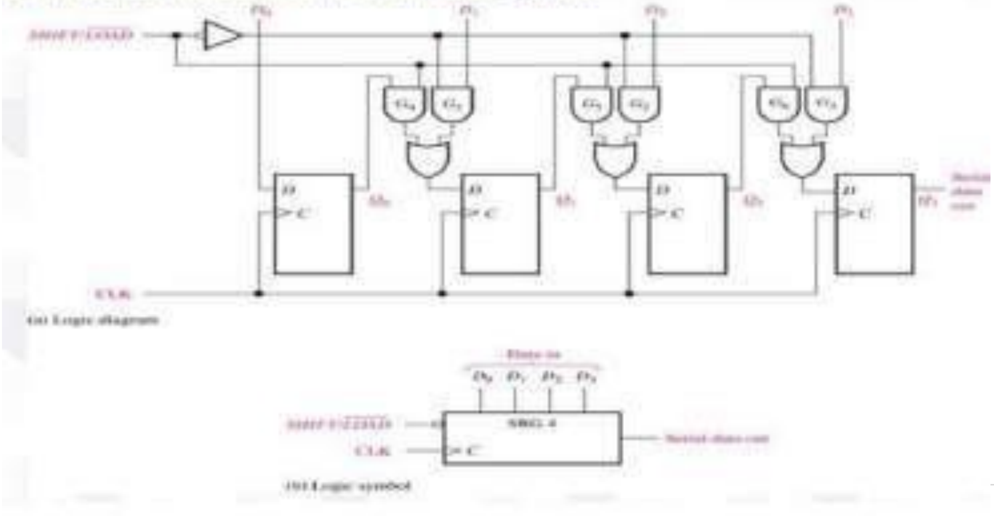
Theory:

Parallel-in/ serial-out shift registers do everything that the previous serial-in/ serial-out shift registers do plus input data to all stages simultaneously. The parallel-in/ serial-out shift register stores data, shifts it on a clock by clock basis, and delays it by the number of stages times the clock period. In addition, parallel-in/ serial-out really means that we can load data in parallel into all stages before any shifting ever begins. This is a way to convert data from a *parallel* format to a *serial* format. By parallel format we mean that the data bits are present simultaneously on individual wires, one for each data bit as shown below. By serial format we mean that the data bits are presented sequentially in time on a single wire or circuit as in the case of the "data out" on the block diagram below.

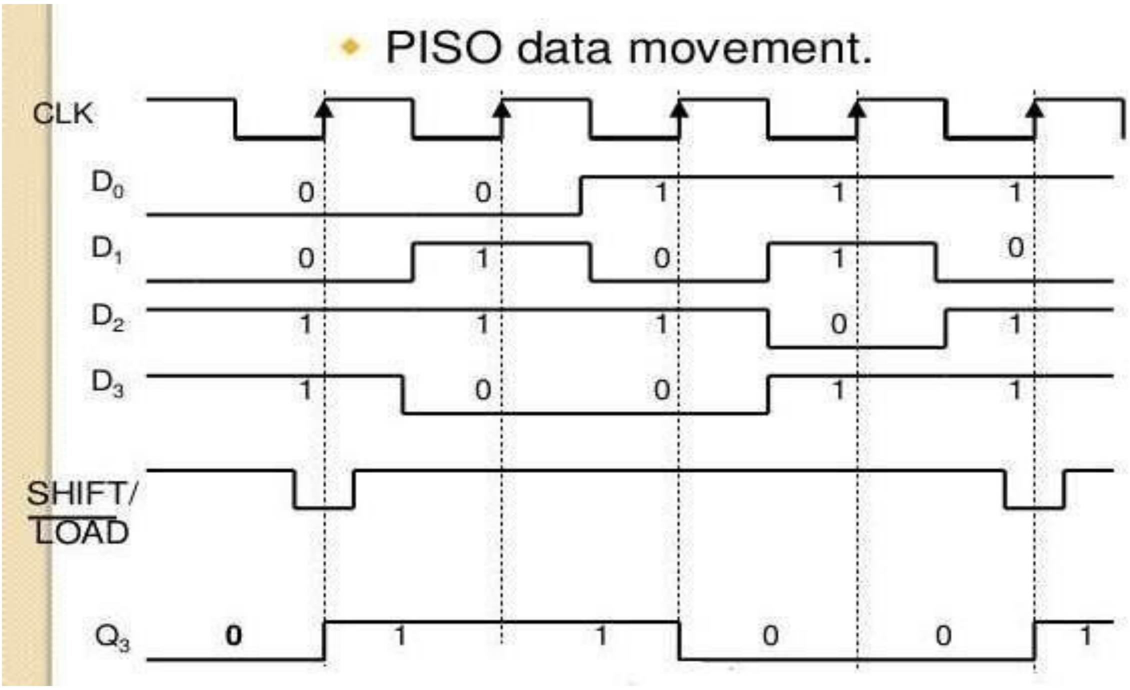


Parallel-in, serial-out shift register with 4-stages

A 4-bit parallel in/serial out shift register.



Timing Diagram:



Procedure:

1. Connect the circuit as per circuit diagram.
2. Connect the CLK to CLK I/P as shown above.
3. Connect D0,D1,D2,D3 and Shift/Load to input switches.
4. If shift/load =0 inputs are loaded into flipflops and shift/load = 1 shift operation is performed.
5. Observe the serial output at the last flipflop.

Result: 8-bit parallel load and serial out shift register using flip-flops is verified.

VIVA: 1.What is parallel and serial out shift register?
2. What is an 8-bit shift register?
3.What is the principle of Sipo?
4.What is meant by the parallel load of shift register?
5.What are the types of shift register?
6.What is the use of shift register?
7.What is parallel register?
8.What are two functions of a shift register?
9.What are the applications of register?
10.what is the difference b/w register and counter?

Experiment No: 8

Synchronous and Asynchronous counter using flip-flops

Aim: To Design and realization of a Synchronous and Asynchronous counters using flip-flops

Apparatus:

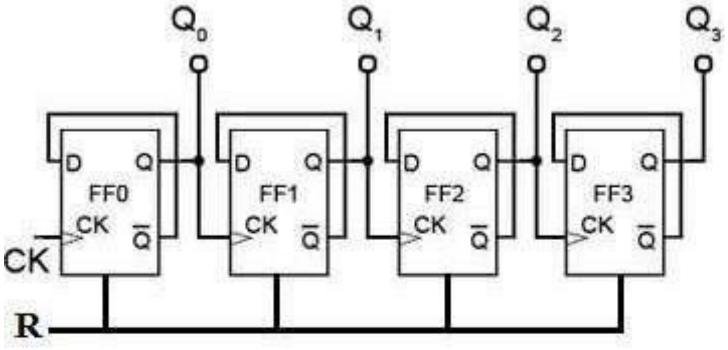
Counters trainer kit, patch cords

Theory:

Since we know that binary count sequences follow a pattern of octave (factor of 2) frequency division, and that J-K flip-flop multivibrators set up for the "toggle"mode are capable of performing this type of frequency division, we can envision a circuit made up of several J-K flip-flops, cascaded to produce four bits of output. The main problem facing us is to determine *how* to connect these flip-flops together so that they toggle at the right times to produce the proper binary sequence.

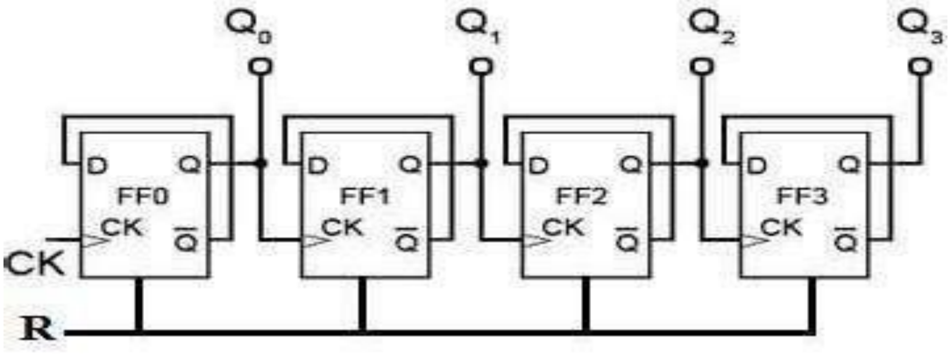
The **Synchronous Counter**, the external clock signal is connected to the clock input of EVERY individual flip-flop within the counter so that all of the flip-flops are clocked together simultaneously (in parallel) at the same time giving a fixed time relationship. In other words, changes in the output occur in "synchronisation" with the clock signal.

Asynchronous up counter:

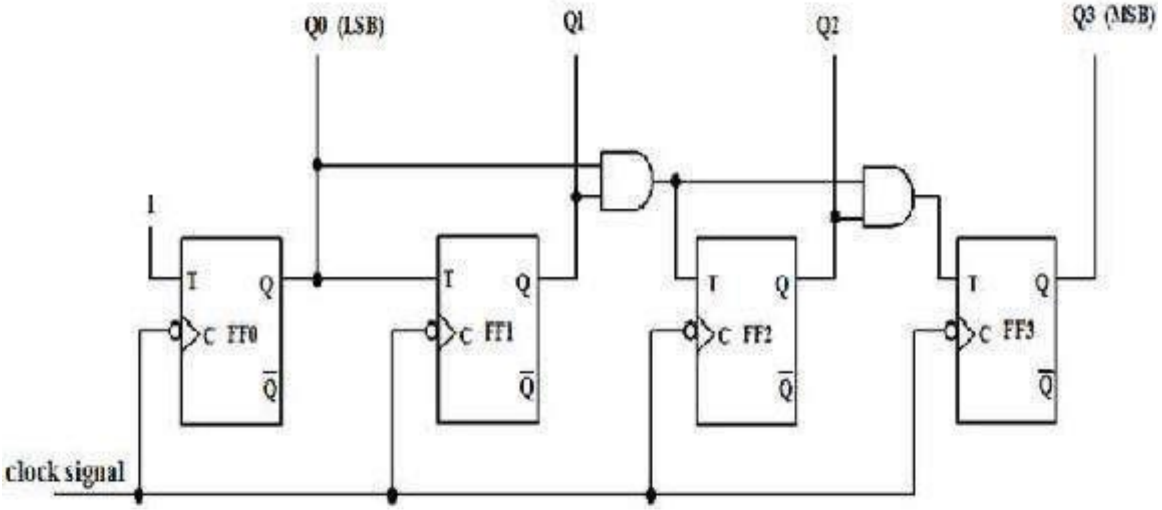


Asynchronous up counter

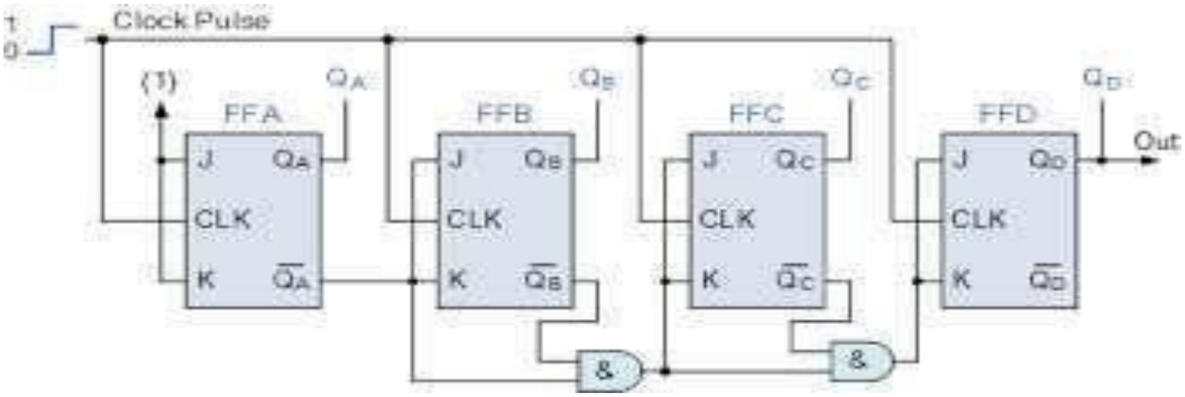
Asynchronous Down counter:

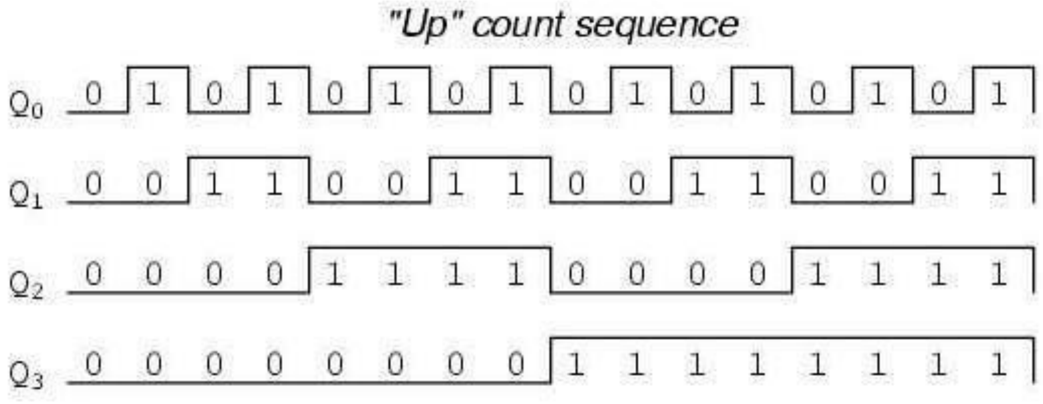


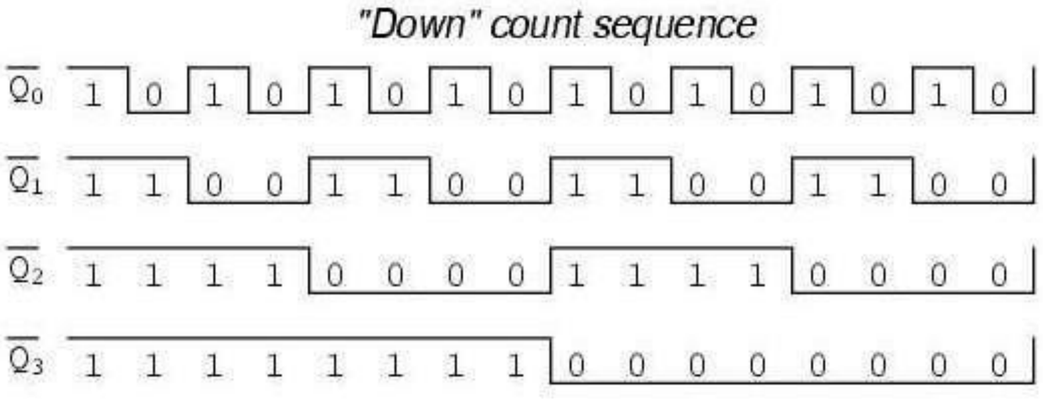
Synchronous up counter:



Synchronous Down counter:







Truth table:

Up Counter:

Clock Pulse	Decimal No.	Q_3	Q_2	Q_1	Q_0
1st	0	0	0	0	0
2nd	1	0	0	0	1
3rd	2	0	0	1	0
4th	3	0	0	1	1
5th	4	0	1	0	0
6th	5	0	1	0	1
7th	6	0	1	1	0
8th	7	0	1	1	1
9th	8	1	0	0	0
10th	9	1	0	0	1
11th	10	1	0	1	0
12th	11	1	0	1	1
13th	12	1	1	0	0

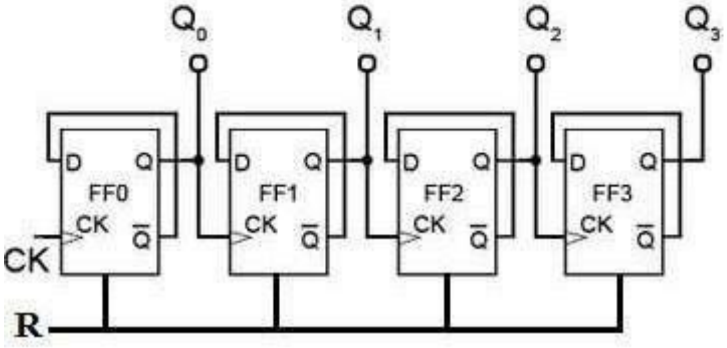
14th	13	1	1	0	1
15th	14	1	1	1	0
16th	15	1	1	1	1

Down counter:

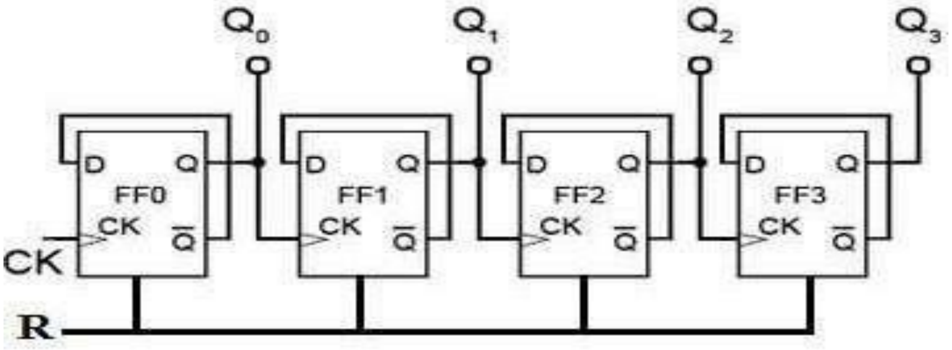
Clock Pulse	Decimal No.	Q_3	Q_2	Q_1	Q_0
1st	15	1	1	1	1
2nd	14	1	1	1	0
3rd	13	1	1	0	1
4th	12	1	1	0	0
5th	11	1	0	1	1
6th	10	1	0	1	0
7th	9	1	0	0	1
8th	8	1	0	0	0
9th	7	0	1	1	1
10th	6	0	1	1	0
11th	5	0	1	0	1
12th	4	0	1	0	0
13th	3	0	0	1	1
14th	2	0	0	1	0
15th	1	0	0	0	1
16th	0	0	0	0	0

Circuit Diagram:

Asynchronous up counter:



Asynchronous down counter:



Truth table:

Up Counter:

Clock Pulse	Decimal No.	Q_3	Q_2	Q_1	Q_0
1st	0	0	0	0	0
2nd	1	0	0	0	1
3rd	2	0	0	1	0
4th	3	0	0	1	1
5th	4	0	1	0	0
6th	5	0	1	0	1
7th	6	0	1	1	0
8th	7	0	1	1	1
9th	8	1	0	0	0
10th	9	1	0	0	1
11th	10	1	0	1	0
12th	11	1	0	1	1
13th	12	1	1	0	0
14th	13	1	1	0	1
15th	14	1	1	1	0
16th	15	1	1	1	1

Down counter:

Clock Pulse	Decimal No.	Q_3	Q_2	Q_1	Q_0
1st	15	1	1	1	1
2nd	14	1	1	1	0
3rd	13	1	1	0	1
4th	12	1	1	0	0
5th	11	1	0	1	1
6th	10	1	0	1	0
7th	9	1	0	0	1
8th	8	1	0	0	0
9th	7	0	1	1	1
10th	6	0	1	1	0
11th	5	0	1	0	1
12th	4	0	1	0	0
13th	3	0	0	1	1
14th	2	0	0	1	0
15th	1	0	0	0	1
16th	0	0	0	0	0

Procedure:

1. Connect the circuit as per circuit diagram.
2. Connect the CLK to CLK I/P as shown in fig.
3. Verify the truth tables.

Result: Design and realization of a Synchronous and Asynchronous counters using flip-flops are verified.

VIVA:

1.Differenciate asynchronous and synchronous counters?
2.Why do we use asynchronous counter?
3.What is another name for asynchronous counter?
4.What are the types of synchronous counter?
5.Draw the 2-bit synchronous counter?
6.What is the principle of Asynchronous counter?
7.Where is asynchronous used?
8.Which is faster synchronous or asynchronous?
9.Draw 2-bit asynchronous counter?
10.Why is it called asynchronous?

Experiment No: 9

8x1 Multiplexer using 2x1 Multiplexer

Aim:

To Design and realization of 8x1 using 2x1 mux

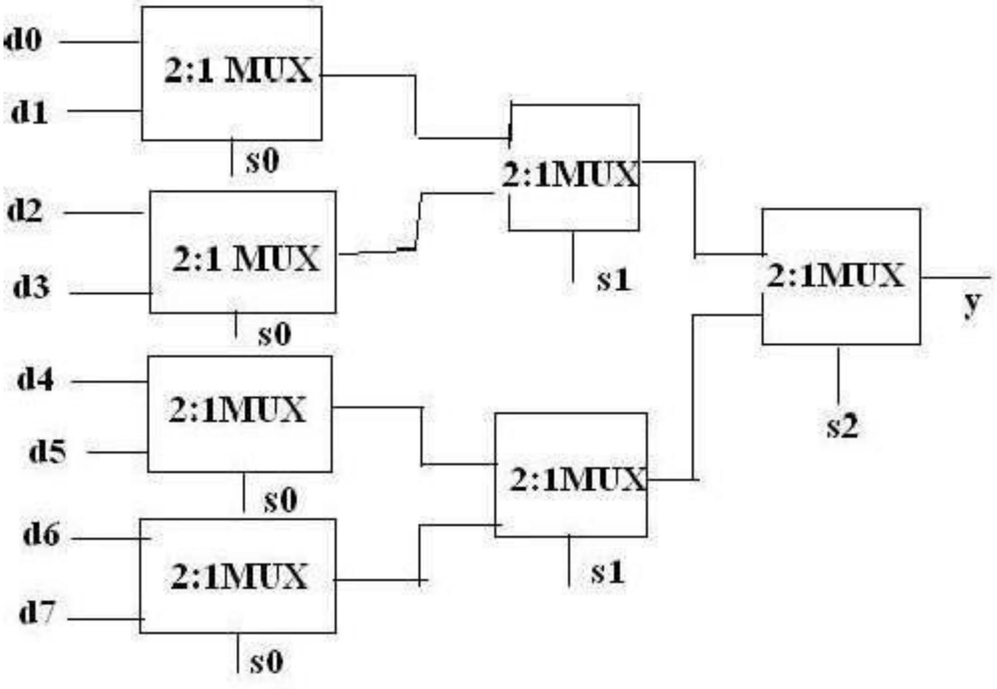
Apparatus:

Trainer kit, Patch cords

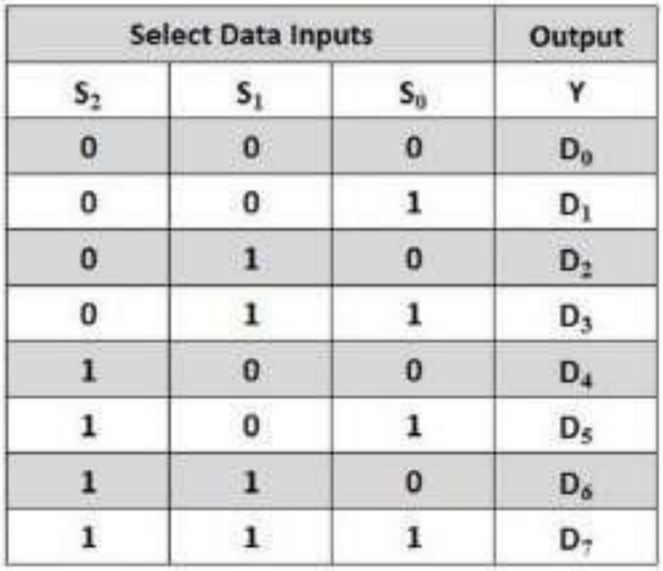
Theory:

This is an 8X1 MUX with inputs d0,d1,d2,d3,d4,d5,d6,d7 , Y as output and S2, S1, S0 as selection lines. The output will depend upon the combination of S2,S1 & S0 as shown in the truth table.

Circuit Diagram:



Truth table:



Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Procedure:

1. Connect the circuit as per circuit diagram.
2. Connect D0 to D7 to the input switches.
3. Connect selection line inputs to the input switches.
4. Connect the output terminal Y to the output indicator (LED).
5. Verify the truth table given above.

Result: Design and realization of 8x1 using 2x1 mux is done.

VIVA:

1.write the algorithm to implement 8 x 1 multiplexer using 2 x1 multiplexer?
2.How does a 8X1 multiplexer work?
3.Can we design a 4x1 mux with two 2x1 multiplexers?
4.What is the Boolean function of 8 to 1 multiplexer?
5.What is the Boolean function of 2 to 1 multiplexer?
6.What are the applications of MUX?
7.Why is multiplexer called universal gate?
8. What is the logic function of MUX?
9.What is the full form of MUX?
10Which IC is used for MUX?

Experiment No: 10

Design and realization 4-bit comparator

Aim: To verify the truth table of 4-bit Comparator.

Apparatus: MTS-COMPARATOR KIT,

PATCH CORDS

Theory:

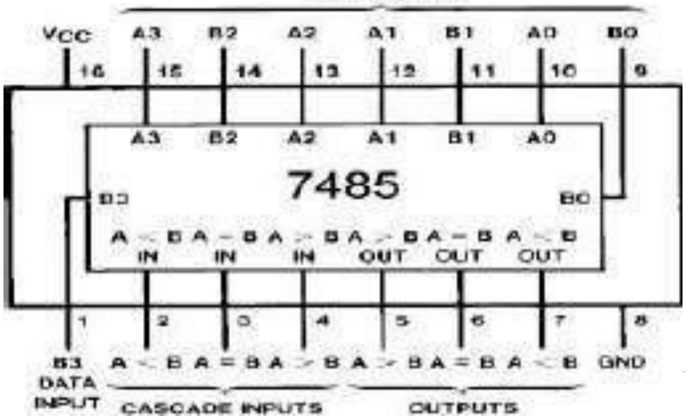
A magnitude digital Comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for A > B condition, one for A = B condition and one for A < B condition.

A comparator used to compare two bits is called a single bit comparator. It consists of two inputs each for two single bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers.

A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers.

A comparator used to compare two binary numbers each of four bits is called a 4-bit magnitude comparator. It consists of eight inputs each for two four bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers.

Pin diagram:



Truth Table:

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	H	L	L
A3<B3	X	X	X	X	X	X	L	H	L
A3=B3	A2>B2	X	X	X	X	X	H	L	L
A3=B3	A2<B2	X	X	X	X	X	L	H	L
A3=B3	A2=B2	A1>B1	X	X	X	X	H	L	L
A3=B3	A2=B2	A1<B1	X	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	X	X	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	L	L	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	L	H	H	L

Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect Inputs (A0,A1,A2,A3,B0,B1,B2,B3) to Input Switches
3. Connect Cascade Inputs (A<B,A>B,A=B) to Input Switches
4. Connect Outputs (A<B,A>B,A=B) to Output Switches.
5. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
6. Apply various input combinations and observe output for each one.
7. Verify the truth table for each input/ output combination.
8. Repeat the process for all logic functions.
9. Switch off the ac power supply.

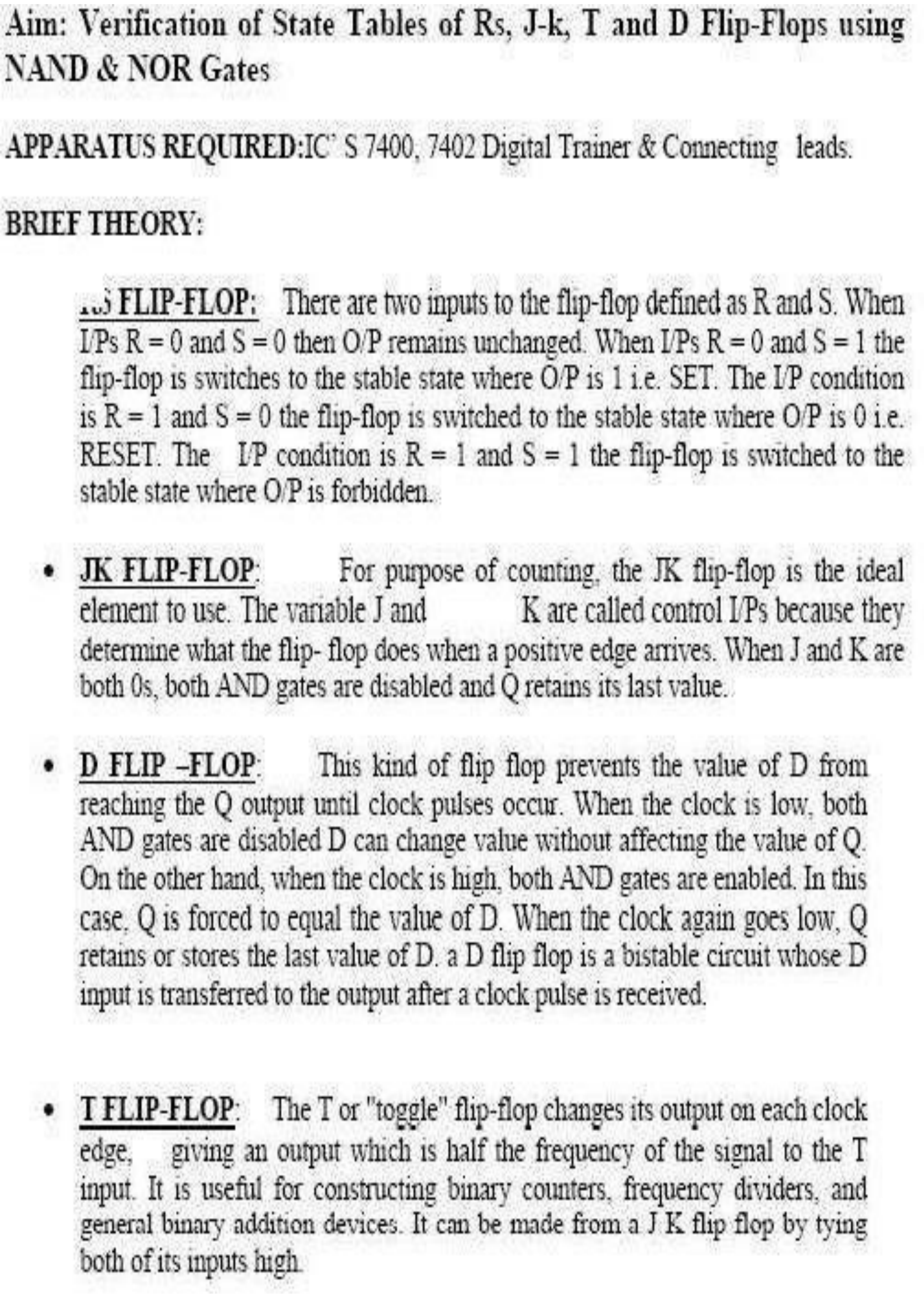
Result: Truth table of 4-bit Comparator is verified.

VIVA:

1.What is bit comparator?
2.What is 4-bit comparator ?
3.What is the function of comparator?
4.Write an example of a 4-bit comparator?
5. Write 4-bit comparator truth table?
6.Design a 1-bit comparator?
7.Design a 2-bit comparator?
8.Design a 3-bit comparator?
9.Design a 4-bit comparator?
10. Write the Boolean expression of 2-bit comparator?

Experiment No: 11

Verification of truth tables and excitation tables



Aim: Verification of State Tables of Rs, J-k, T and D Flip-Flops using NAND & NOR Gates

APPARATUS REQUIRED:IC' S 7400, 7402 Digital Trainer & Connecting leads.

BRIEF THEORY:

RS FLIP-FLOP: There are two inputs to the flip-flop defined as R and S. When I/Ps R = 0 and S = 0 then O/P remains unchanged. When I/Ps R = 0 and S = 1 the flip-flop is switches to the stable state where O/P is 1 i.e. SET. The I/P condition is R = 1 and S = 0 the flip-flop is switched to the stable state where O/P is 0 i.e. RESET. The I/P condition is R = 1 and S = 1 the flip-flop is switched to the stable state where O/P is forbidden.

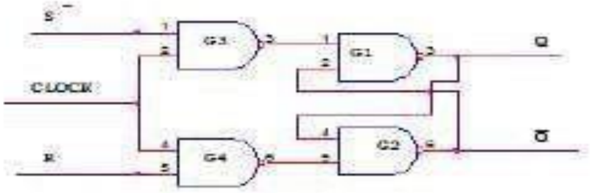
- **JK FLIP-FLOP**: For purpose of counting, the JK flip-flop is the ideal element to use. The variable J and K are called control I/Ps because they determine what the flip- flop does when a positive edge arrives. When J and K are both 0s, both AND gates are disabled and Q retains its last value.

- **D FLIP –FLOP**: This kind of flip flop prevents the value of D from reaching the Q output until clock pulses occur. When the clock is low, both AND gates are disabled D can change value without affecting the value of Q. On the other hand, when the clock is high, both AND gates are enabled. In this case, Q is forced to equal the value of D. When the clock again goes low, Q retains or stores the last value of D. a D flip flop is a bistable circuit whose D input is transferred to the output after a clock pulse is received.

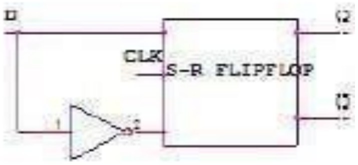
- **T FLIP-FLOP**: The T or "toggle" flip-flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. It is useful for constructing binary counters, frequency dividers, and general binary addition devices. It can be made from a J K flip flop by tying both of its inputs high.

CIRCUIT DIAGRAM:

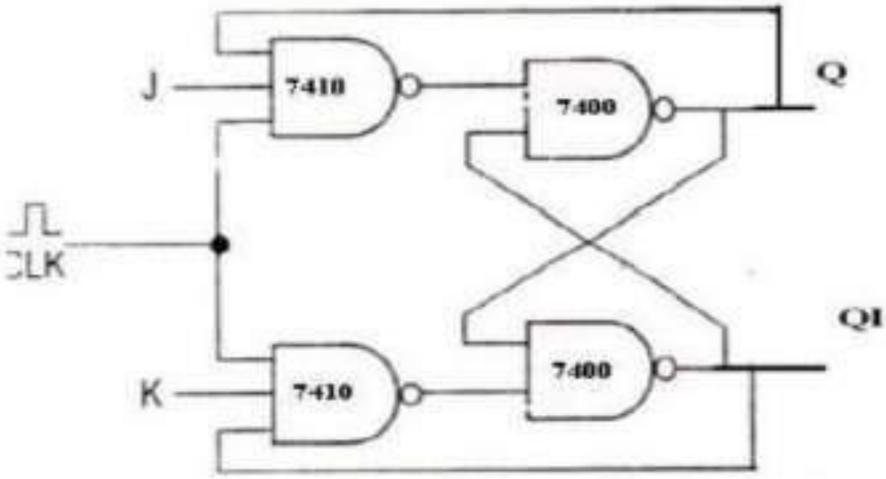
SR Flip Flop

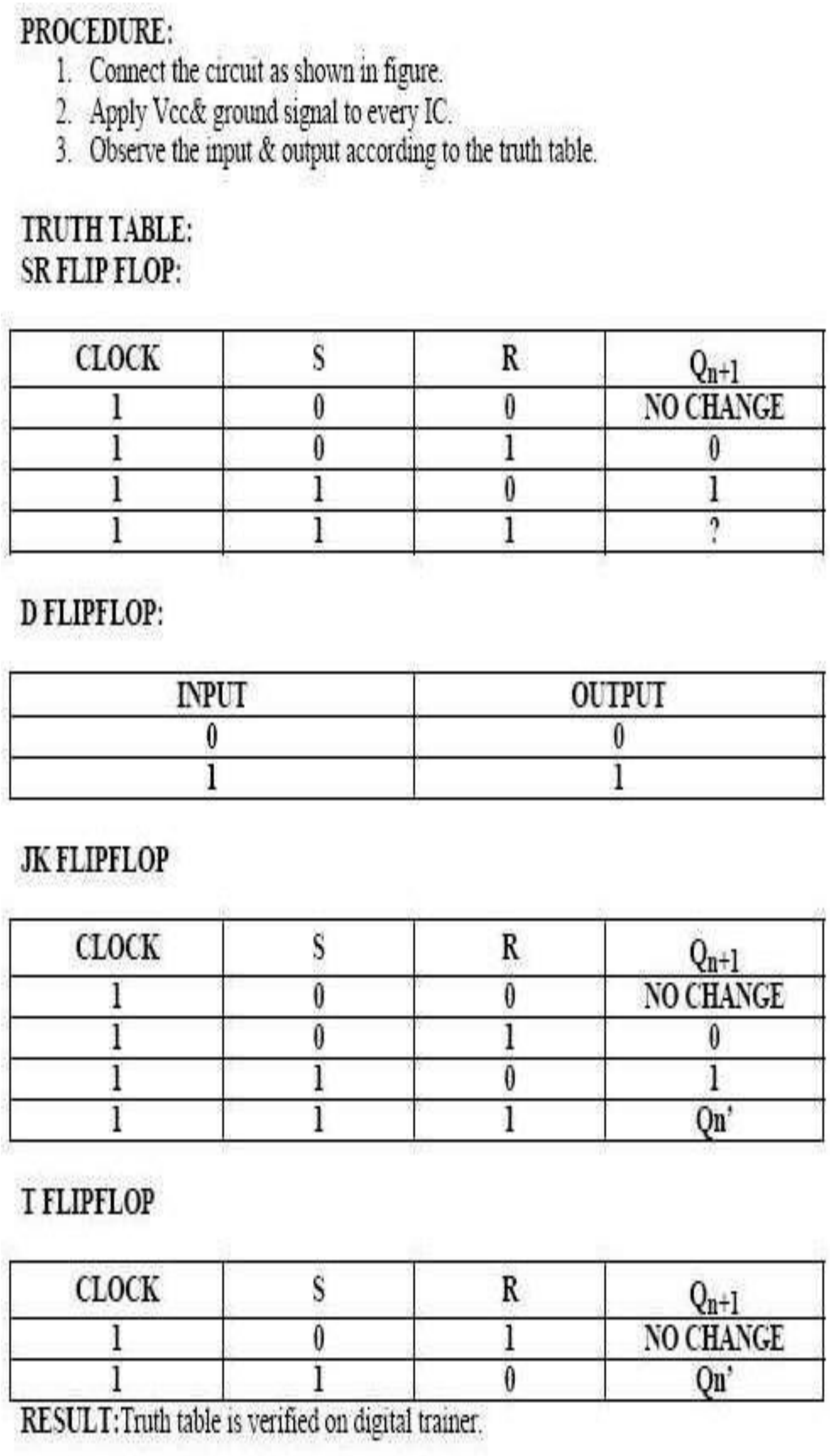


D Flip Flop



J-k flip-flop using NAND gates





PROCEDURE:

1. Connect the circuit as shown in figure.
2. Apply Vcc& ground signal to every IC.
3. Observe the input & output according to the truth table.

TRUTH TABLE:

SR FLIP FLOP:

CLOCK	S	R	Q_{n+1}
1	0	0	NO CHANGE
1	0	1	0
1	1	0	1
1	1	1	?

D FLIPFLOP:

INPUT	OUTPUT
0	0
1	1

JK FLIPFLOP

CLOCK	S	R	Q_{n+1}
1	0	0	NO CHANGE
1	0	1	0
1	1	0	1
1	1	1	Qn'

T FLIPFLOP

CLOCK	S	R	Q_{n+1}
1	0	1	NO CHANGE
1	1	0	Qn'

RESULT: Truth table is verified on digital trainer.

VIVA:

1. What is the main use of flip-flop?
2. What is the aim of flip-flop?
3. What is flip-flop RS?
4. Why is RS flip-flop used?
5. Why is it called flip-flop?
6. Which flip-flop is most important?
7. What is T flip-flop?
8. What is full form of D flip-flop?
9. What is flip-flop clock?
10. What is the full form of JK flip flop?

Experiment No: 12

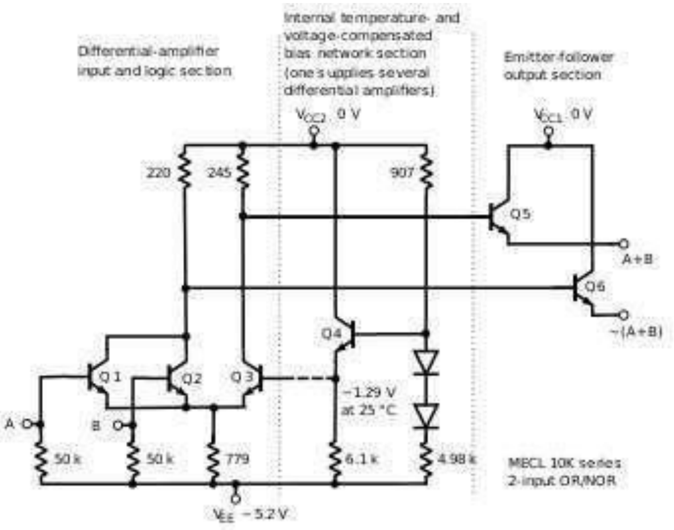
Realization of logic gates using DTL, TTL, ECL, etc.,

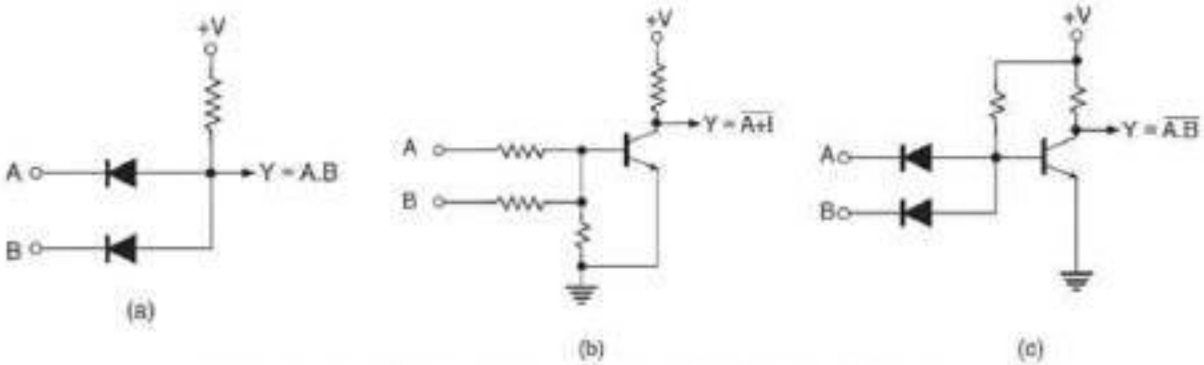
Aim: Realization of logic gates using DTL, TTL, ECL, etc.,

Apparatus: Logic trainer kit, Connecting wires

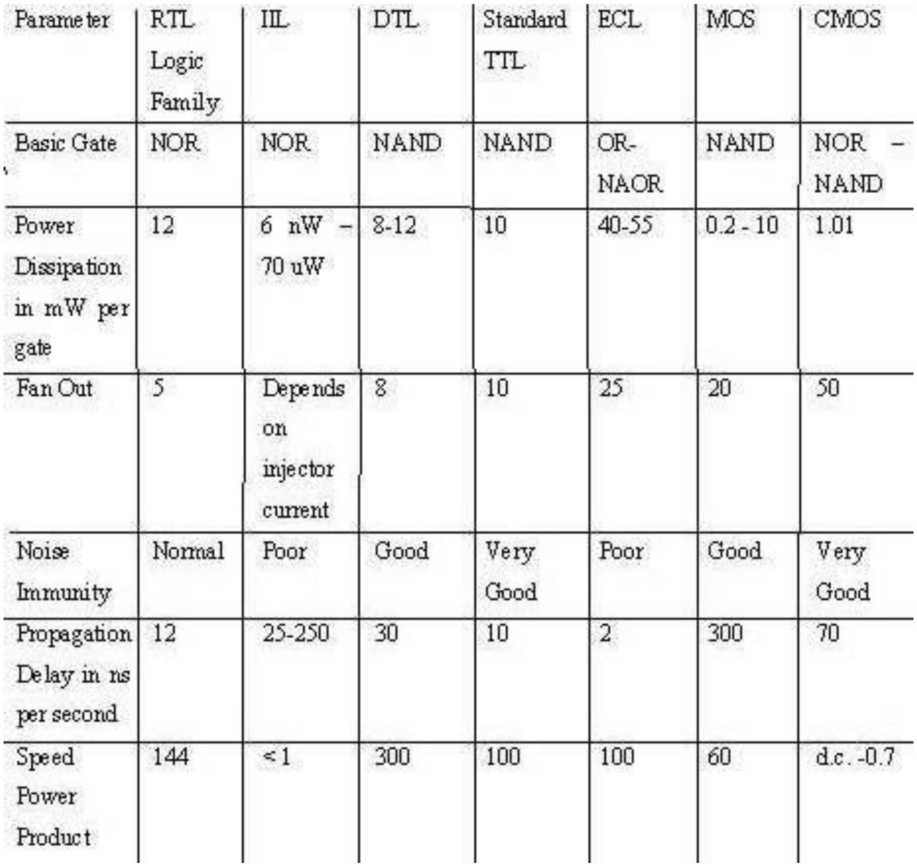
Theory: Diode- Transistor **Logic**, or **DTL**, refers to the technology for designing and fabricating digital **circuits** wherein **logic gates** employ diodes **in** the input stage and bipolar junction transistors at the output stage. The output BJT switches between its cut-off and saturation regions to create **logic** 1 and 0, respectively.

Circuit Diagram:


Differential-amplifier input and logic section
Internal temperature- and voltage-compensated bias network section (one s upplies several differential amplifiers)
Emitter-follower output section
V_{CC2} 0 V
V_{CC1} 0 V
220
245
907
Q5
A+B
Q6
~(A+B)
Q4
Q1
Q2
Q3
−1.29 V at 25 °C
A
B
50 k
50 k
779
6.1 k
4.98 k
MECL 10K series 2-input OR/NOR
V_{EE} −5.2 V



(a) Diode logic (b) resistor transistor logic and (c) diode transistor logic.



Parameter	RTL Logic Family	IIL	DTL	Standard TTL	ECL	MOS	CMOS
Basic Gate	NOR	NOR	NAND	NAND	OR-NAOR	NAND	NOR – NAND
Power Dissipation in mW per gate	12	6 nW – 70 uW	8-12	10	40-55	0.2 - 10	1.01
Fan Out	5	Depends on injector current	8	10	25	20	50
Noise Immunity	Normal	Poor	Good	Very Good	Poor	Good	Very Good
Propagation Delay in ns per second	12	25-250	30	10	2	300	70
Speed Power Product	144	< 1	300	100	100	60	d.c. -0.7

Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the NOR gates for any of the logic functions to be realised.
3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply varous input combinations and observe output for each one.
5. Verify the tructh table for each input/ output combination.
6. Repeat the process for all logic functions.
7. Switch off the ac power supply.

Result: Realization of logic gates using DTL, TTL, ECL truth tables are verified.

VIVA:

1. What is TTL also known as?
2. How many stages does DTL consist?
3. What is a disadvantage of DTL?
4. What TTL is used for?
5. Why is TTL faster than DTL?
6. What is DTL used for?
7. What is the concept of DTL?
8. What are the uses of DTL?
9. How many stages does TTL consist?
10. What are the disadvantage of TTL, ECL?

BEYOND THE SYLLBUS EXPERIMENT 1

DESIGN 450KHZ CLOCK USING NAND/NOR GATES

AIM: To generate 450KHZ clock using nand/nor gates.

APPARATUS : clock generator trainer kit, NAND & NOR IC's, Patch cords

THEORY:

Pulse generator using NAND gate

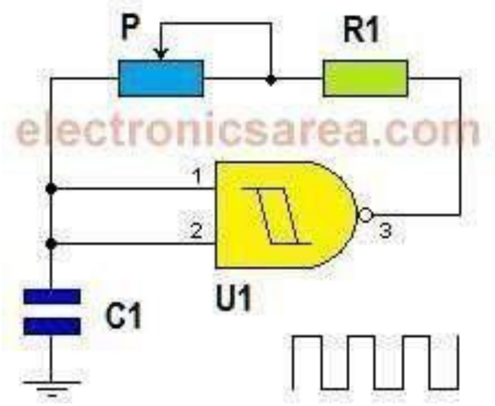
Pulse generator using NAND gate:

This simple **pulse generator using NAND gate** is very useful and can be used as clock generator for other circuits. The circuit's main element is a 2 input (Schmitt Trigger) NAND gate connected as NOT gate.

Analyzing the circuit diagram and remembering the truth table for a NAND gate, we know that when the two inputs are logic "1", the output will be at logic "0" and when the two inputs are at "0", the output will be a logical "1". Exactly the same behavior that would have a NOT gate.

How the pulse generator using NAND gate works?

The NAND gate has at its output a square wave whose frequency depends on the values of capacitorC1 and the series combination of the potentiometer P and resistor R1.



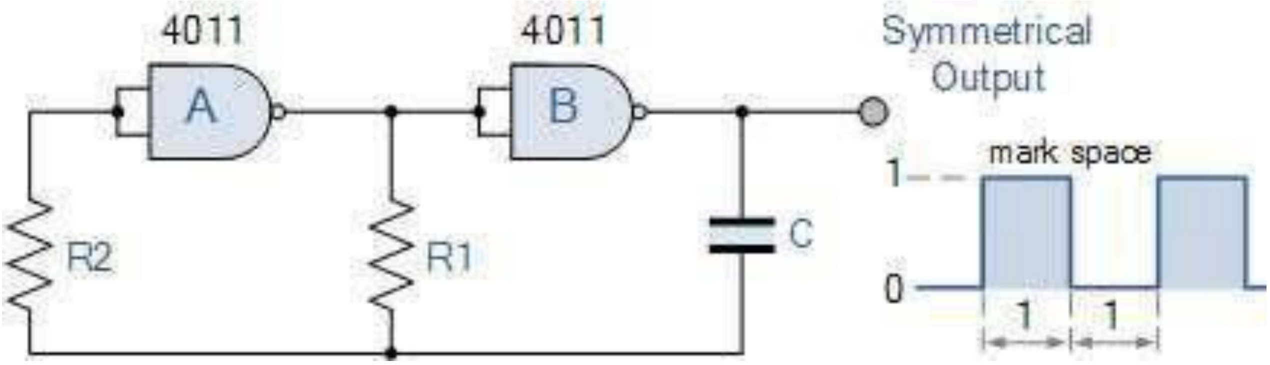
When the output of NAND gate is High (logical 1), the capacitor C1 (which has no charge) sends a logical zero (0) at the inputs of the NAND gate. The capacitor begins to charge through the resistorand potentiometer assembly.

Once the capacitor is charged, both inputs of the gate are high (logical 1), causing its output to go low (logic zero). With the low output, capacitor discharge begins through the same set of resistorand potentiometer in series, returning to its original state. The process is repeated continously.You can use a 3.0V to 15V power suppy.

SQUARE WAVE GENERATOR USING NAND GATE:

Square waves are extensively used in many digital circuits. Many combinational logic circuits require this wave to operate. Here are few methods you can generate simple square wave using NAND, Inverter and Schmitt Trigger gates. These kind of square wave generator fit perfectly for simple oscillator applications with minimum effort required to build.

CIRCUIT DIAGRAM:



Procedure :

1. Connect the circuit as per the circuit diagram .
2. Apply the different values of resistors and capacitors and the observe the frequency .
3. Repeat the process for different combinations of resistors and capacitors.
4. Compare the theoretical and practical values of frequencies.

Result: Clock generation using NAND/NOR gates is verified.

BEYOND THE SYLLBUS EXPERIMENT

Design and Realization of a sequence detector-a finite state machine

Aim: To design and realize sequence detector-a finite state machine.

Apparatus: logic gates / ICs

PATCH CORDS

Theory:

A sequence detector is a sequential state machine which takes an input string of bits and generates an output 1 whenever the target sequence has been detected. In a Mealy machine, output depends on the present state and the external input (x). Hence in the diagram, the output is written outside the states, along with inputs. Sequence detector is of two types:

Overlapping

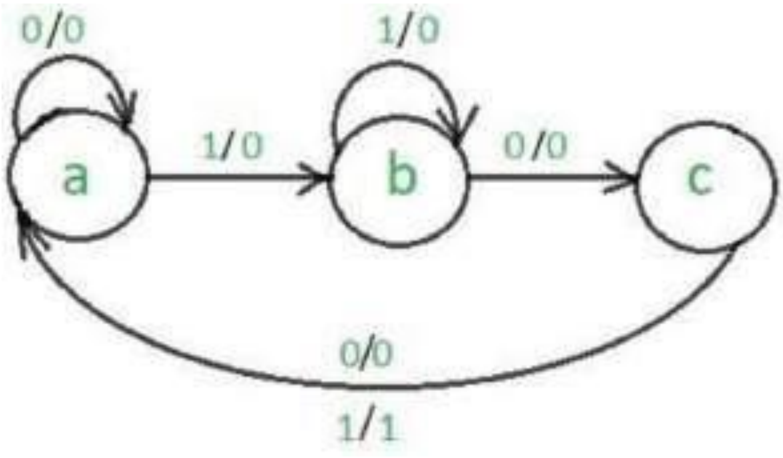
Non-Overlapping

In an overlapping sequence detector the last bit of one sequence becomes the first bit of next sequence. However, in non-overlapping sequence detector the last bit of one sequence does not become the first bit of next sequence. In this post, we'll discuss the design procedure for non -overlapping 101 Mealy sequence detector.

Procedure:

Step 1: Develop the state diagram –

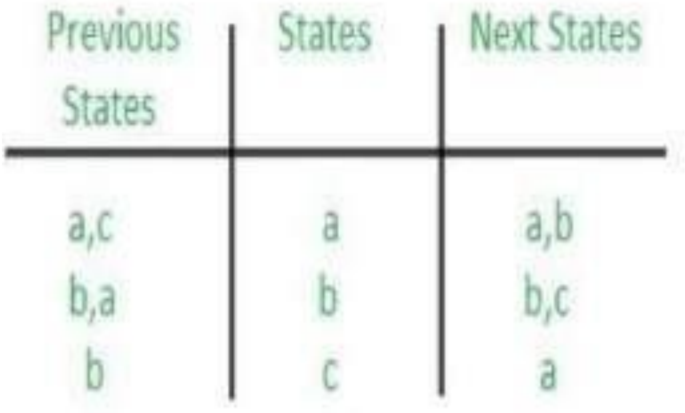
The state diagram of a Mealy machine for a 101 sequence detector is:



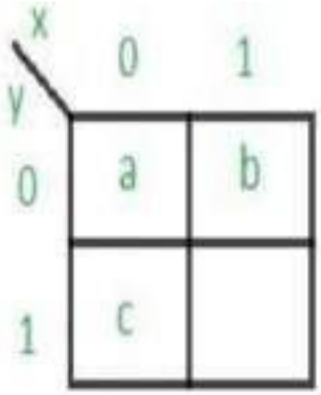
Step 2: Code Assignment –

Rule 1 : States having the same next states for a given input condition should have adjacent assignments.

Rule 2: States that are the next states to a single state must be given adjacent assignments. Rule 1 given preference over Rule 2

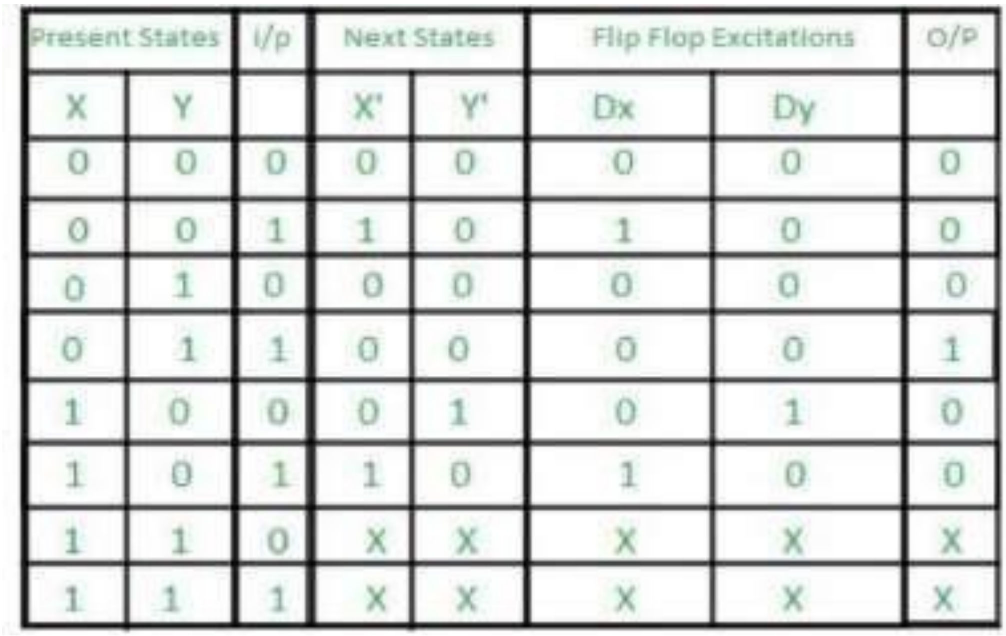


Previous States	States	Next States
a,c	a	a,b
b,a	b	b,c
b	c	a



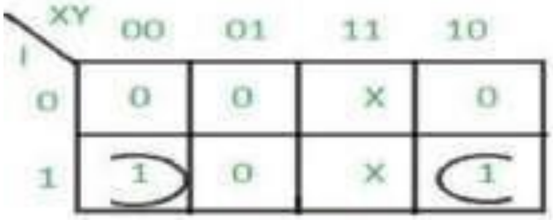
y \ x	0	1
0	a	b
1	c	

Step3:MakePresent State/Next Statetable – We’lluse D-Flip Flopsfordesignpurpose.



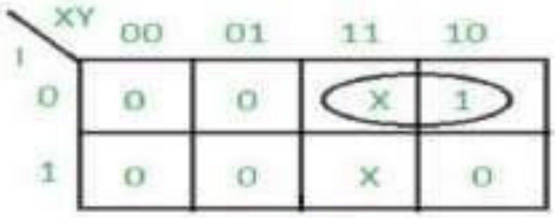
Present States		I/p	Next States		Flip Flop Excitations		O/P
X	Y		X'	Y'	Dx	Dy	
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X

Step 4: Draw K-maps for Dx, Dy and output (Z)



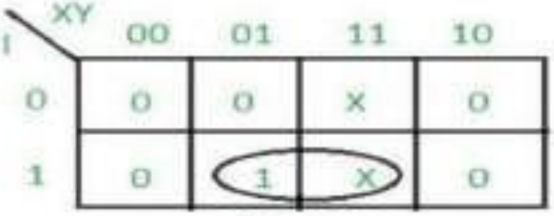
I \ XY	00	01	11	10
0	0	0	X	0
1	1	0	X	1

Dx=$\overline{Y}$.I



I \ XY	00	01	11	10
0	0	0	X	1
1	0	0	X	0

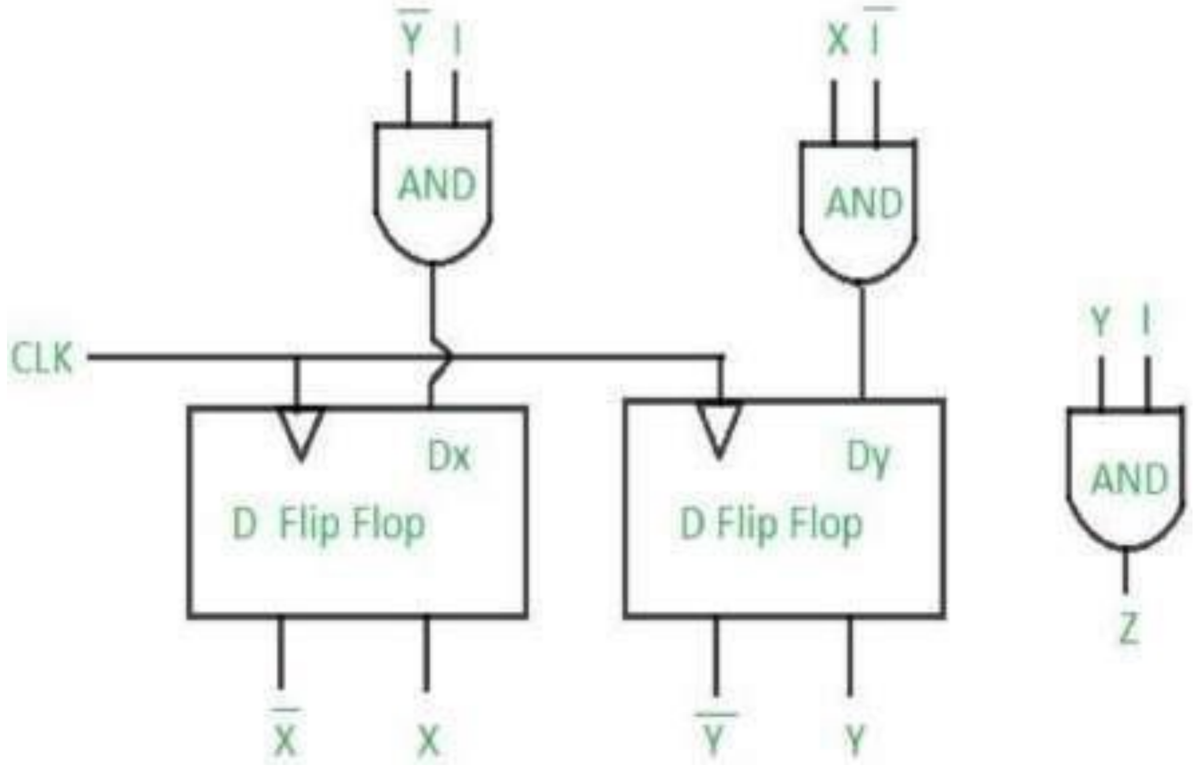
Dy= X.$\overline{I}$



I \ XY	00	01	11	10
0	0	0	X	0
1	0	1	X	0

Z=Y.I

Step 5: Finally implement the circuit



Result: Verified the sequence detector of sequence 101 theoretically and practically.