P.E.S. Education Trust (R), Mandya

P.E.S. COLLEGE OF ENGINEERING

(An Autonomous Institution Affiliated to VTU, Belagavi, Grant -in- Aid Institution (Government of Karnataka), World Bank Funded College (TEQIP), Accredited by NBA & NAAC and Approved by AICTE, New Delhi.)

Mandya - 571401, Karnataka

DIGITAL LOGIC DESIGN
[P21IS304] Manual

Department of Information Science and Engineering

P.E.S. College of Engineering

Mandya - 571401, Karnataka

DIGITAL LOGIC DESIGN [As per Choice Based Credit System (CBCS) & OBE Scheme] **SEMESTER – III**			
Course Code:	**P21IS304**	**Credits:**	**04**
Teaching Hours/Week (L: T: P):	**3:0:2**	**CIE Marks:**	**50**
Total Theory Teaching Hours: **Total Laboratory Hours:**	**40** **24**	**SEE Marks:**	**50**

Course Learning Objectives: This course will enable the students to:

- Understand Boolean laws and minimization techniques and fundamental gates
- Design of combinational logic circuits using minimum number of gates, Decoders and Multiplexers
- Understand the Sequential logic components and Design of sequential circuits
- Understand and use high-level hardware description languages (VHDL) to design combinational / sequential circuits
- Conduct and Simulate practical experiments of combinational and sequential circuit

UNIT – I	**Boolean Algebra and minimization of switching functions**	**8 Hours**

Boolean Algebra: Introduction, Logic gates, Boolean Laws, Duality, Boolean expression in standard SOP and POS, Realization using basic gates and universal gates.

Minimization Of Switching Functions: Introduction, K-Map: Two-variable, Three-variable and Four-variable K-map, Don't care combinations**,** Map entered variable (VEM), Limitation of K-map, Code converters: Binary to gray**,** BCD to Excess 3**,** Quine-Mc-Clusky method- 3 variable

Self-study component:	Quine-Mc-Clusky method- 4,5 variable
Practical Topics: **(6 Hours)**	Verify the truth table for different logic gates using IC's 1. A committee of three individuals decides issues for an organization. Each individual votes either yes or no for each proposal that arises. A proposal is passed if it receives at least two yes votes. Design a circuit using minimum number of NAND gates only that determines whether a proposal passes. 2. Design Logic circuit to convert 3 bit binary to gray code using basic gates.

UNIT – II	**Combinational Logic Design**	**8 Hours**

Introduction to combinational circuits, Adders, Subtractors, ripple carry adder, Look ahead carry adder, Comparators:1-bit and 2bit magnitude comparator, Encoders: octal to Binary and Decimal to BCD encoder, Priority encoders, Decoders: 2 to 4, 3 to 8 line decoder, Multiplexers: 2:1,4:1, 8:1,16:1 , Design combinational circuits using Decoders and Multiplexers

Self-study component:	7 Segment Decoder, Demultiplexer
Practical Topics: **(6Hours)**	1. Design Full adder using suitable Decoder 2. A lawn sprinkling system is controlled automatically by certain combinations of the following variables. Season (S=1, if summer; 0, otherwise) Moisture content of soil (M=1, if high;0 if low) Outside temperature (T=1, if high;0 if low) Outside humidity (H=1, if high;0 if low) The sprinkler is turned on under any of the following circumstances:

	i. The moisture content is low in winter. ii. The temperature is high and the moisture content is low in summer. iii. The temperature is high and the humidity is high in summer. iv. The temperature is low and the moisture content is low in summer. v. The temperature is high and the humidity is low. Implement using suitable multiplexer (use 8x1 mux)

UNIT – III	**Flip flops**	**8 Hours**

Introduction, Classification of sequential circuits: Asynchronous and Synchronous, NAND and NOR latches and flip flops: Excitation tables, State diagram and Characteristic equation of SR, JK, Race around condition, Master slave JK flip flops, Excitation tables, State diagram and Characteristic equation of D and T flip flops, Conversion of SR to JK, JK to D, T to D Flip flops

Self-study component:	Conversion of JK to SR, D to JK and D to T Flip flops
Practical Topics: **(4 Hours)**	Verify the truth table of JK and D Flip Flops 1. Implement Master slave D Flip Flop using only NAND Gates 2. Design and demonstrate the conversion of JK flip flop to T Flip Flop

UNIT – IV	**Shift Registers and Counters**	**8 Hours**

Introduction, Data Transmission In Shift Registers, Serial In Serial Out Shift Register, Serial In Parallel Out Shift Register, Parallel In Serial Out Shift Register, Parallel In Parallel Out Shift Register, **Design of** shift registers using JK and D flip Flop's, Application Of Shift Registers: Ring Counter, Johnson Counter, Up/Down Synchronous and Asynchronous Introduction, Design counters using JK and T Flip flop

Self-study component:	Effects of propagation delay in ripple counters, Sequence detector design
Practical Topics: **(4 Hours)**	1. Design and demonstrate 3-bit serial in serial out shift register using D Flip Flop's 2. Design and demonstrate 2-bit synchronous counter for the given sequence using JK Flip Flop.

UNIT – V	**Introduction to VHDL**	**8 Hours**

Hardware description languages, VHDL description of combinational circuits, VHDL models for multiplexers, VHDL modules, Sequential statements and VHDL processes, Modelling Flip-flops using VHDL Processes, VHDL Modelling registers and counters using VHDL processes

Self-study component:	Compilation, simulation and synthesis of VHDL code, Simple synthesis examples.
Practical Topics: **(4 Hours)**	Write the VHDL code for basic gates and verify its working 1. Write the VHDL code for 8:1 Mux. Simulate and verify it's working. 2. Write the VHDL code for JK and D flip-flop. Simulate and verify it's working. 3. Write the VHDL code for 3- bit synchronous down counter. Simulate and verify it's working.
NOTE	These experiments are for Practice **Practical Topics** will be changed every academic year

Course Outcomes: On completion of this course, students are able to:	
COs	**Course Outcomes** with ***Action verbs*** for the Course topics
CO1	**Apply** Boolean Algebra/ K Map and knowledge of fundamental gates in minimizing Logic function
CO2	**Analyze** Combinational and Sequential circuits
CO3	**Design** Combinational /Sequential logic circuit for the given problem
CO4	**Develop** VHDL code for Combinational / Sequential logic circuit
CO5	**Conduct** and Simulate practical experiments for demonstrating the working of Combinational and Sequential circuit both with component realization and VHDL code

Text Book(s):

1. A. Anand Kumar, Fundamentals of Digital Circuits,4th Edition, PHI Learning, ISBN: 9788120352681, Nov- 2016
2. Charles H.Roth, Jr., Lizy Kurian John, Digital Systems Design using VHDL,2nd Edition, CENGAGE Learning,2012

Reference Book(s):

1. M.Morris Mano, Michael D.Ciletti, Digital Design with an introduction to the verilog HDL, VHDL and systemverilog,6th edition, Pearson Publication,2020
2. Donald P Leach, Albert Paul Malvino, Goutam Saha, Digital Principles and applications,8th edition, McGraw-Hill Education,2017

Web and Video link(s):

1. https://nesoacademy.org/ec/05-digital-electronics

E-Books/Resources:

1. https://dvikan.no/ntnu-studentserver/kompendier/digital-systems-design.pdf
2. https://drive.google.com/file/d/1lw9LhePHIhwBljiWSXrmEJgXj5RE05j4/view?usp=sharing

CO-PO-PSO Mapping

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	**Apply** Boolean Algebra / K Map and knowledge of fundamental gates in minimizing Logic function	3												2	
CO2	**Analyze** Combinational and Sequential circuits	1	3	1										1	
CO3	**Design** combinational /sequential logic circuit for the given problem	1	2	3										1	
CO4	**Implement** Combinational/ Sequential logic circuit using VHDL code	1	1	2										2	
CO5	**Conduct** and Simulate practical experiments for demonstrating the working of combinational and sequential circuit both with component realization and VHDL code	1	1	2	1	2				2				2	

EXPERIMENT: 0 LOGIC GATES

AIM: To study and verify the truth table of logic gates.

LEARNING OBJECTIVE:
Identify various ICs and their specification.

COMPONENTS REQUIRED:
Logic gates (IC) trainer kit, Connecting patch chords, IC 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486.

THEORY:
The basic logic gates are the building blocks of more complex logic circuits. These logic gates perform the basic Boolean functions, such as AND, OR, NAND, NOR, Inversion, Exclusive-OR, Exclusive-NOR. Fig. below shows the circuit symbol, Boolean function, and truth. It is seen from the Fig that each gate has one or two binary inputs, A and B, and one binary output, C. The small circle on the output of the circuit symbols designates the logic complement. The AND, OR, NAND, and NOR gates can be extended to have more than two inputs. A gate can be extended to have multiple inputs if the binary operation it represents is commutative and associative.

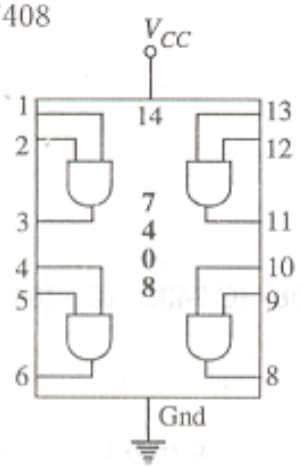
These basic logic gates are implemented as small-scale integrated circuits (SSICs) or as part of more complex medium scale (MSI) or very large-scale (VLSI) integrated circuits. Digital IC gates are classified not only by their logic operation, but also the specific logic-circuit family to which they belong. Each logic family has its own basic electronic circuit upon which more complex digital circuits and functions are developed. The following logic families are the most frequently used.

TTL-Transistor-transistor logic
ECL-Emitter-coupled logic
MOS-Metal-oxide semiconductor
CMOS-Complementary metal-oxide semiconductor

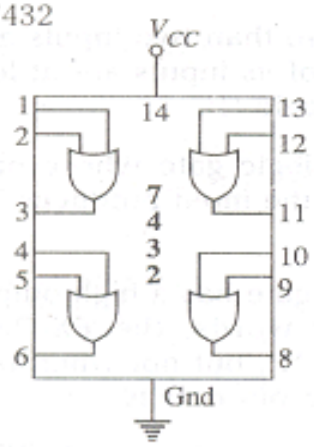
TTL and ECL are based upon bipolar transistors. TTL has a well-established popularity among logic families. ECL is used only in systems requiring high-speed operation. MOS and CMOS are based on field effect transistors. They are widely used in large scale integrated circuits because of their high component density and relatively low power consumption. CMOS logic consumes far less power than MOS logic. There are various commercial integrated circuit chips available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

PIN DIAGRAMS:

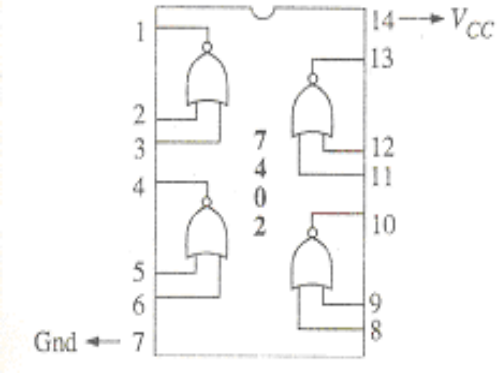
IC 7408



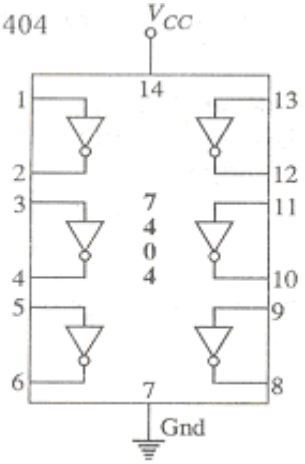
IC 7432



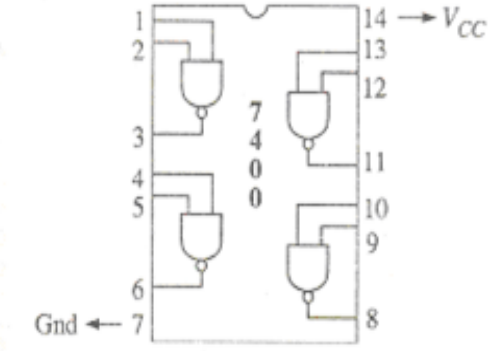
IC 7402



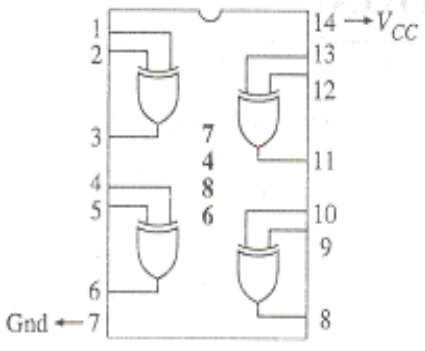
IC 7404

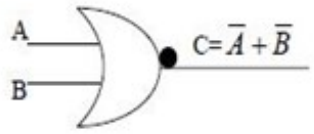
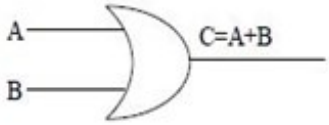
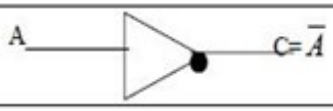
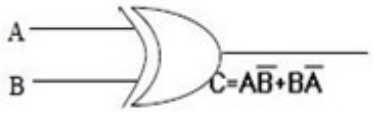


IC 7400



IC 7486



S.NO	GATE	SYMBOL	INPUTS		OUTPUT
			A	B	C
1.	NAND IC 7400		0	0	1
			0	1	1
			1	0	1
			1	1	0
2.	NOR IC 7402		0	0	1
			0	1	0
			1	0	0
			1	1	0
3.	AND IC 7408		0	0	0
			0	1	0
			1	0	0
			1	1	1
4.	OR IC 7432		0	0	0
			0	1	1
			1	0	1
			1	1	1
5.	NOT IC 7404		1	-	0
			0	-	1
6.	EX-OR IC 7486		0	0	0
			0	1	1
			1	0	1
			1	1	0

PROCEDURE:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Provide the input data via the input switches and observe the output through output LEDs.

RESULT: The above truth tables are verified.

VIVA QUESTIONS:

1. Why NAND & NOR gates are called universal gates?
2. Realize the EX – OR gates using minimum number of NAND gates.
3. Give the truth table for EX-NOR and realize using NAND gates?
4. What is the logic low and High levels of TTL IC's and CMOS IC's?
5. Compare TTL logic family with CMOS family?
6. Which logic family is fastest and which has low power dissipation?

EXPERIMENT: 1 DESIGN FOR SCENARIO

SCENARIO:
A committee of three individuals decides issues for an organization. Each individual votes either yes or no for each proposal that arises. A proposal is passed if it receives at least two yes votes. Design a circuit using minimum number of NAND gates only that determines whether a proposal passes.

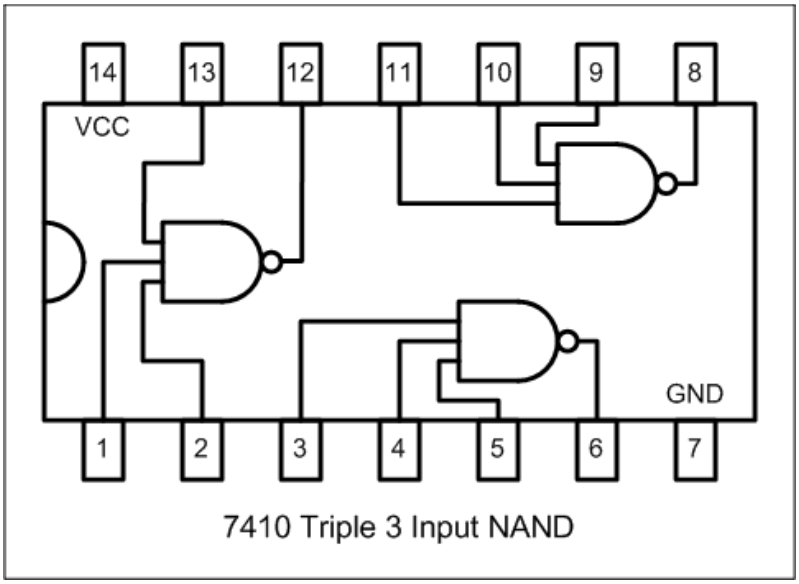
AIM:
To design a circuit for the given scenario using minimum number of NAND gates only.

LEARNING OBJECTIVE:
To learn to realize designing a circuit for a given scenario.

COMPONENTS REQUIRED:
IC 7400, IC 7410, Patch Cords & IC Trainer Kit.

THEORY:
NAND Gate: In digital electronics, a NAND gate (NOT-AND) is a logic gate which produces an output which is false only if all its inputs are true; thus its output is complement to that of an AND gate. A LOW (0) output results only if all the inputs to the gate are HIGH (1); if any input is LOW (0), a HIGH (1) output results. The NAND gate is significant because any Boolean function can be implemented by using a combination of NAND gates. This property is called functional completeness. It shares this property with the NOR gate.

PIN DIAGRAM:



7410 Triple 3 Input NAND

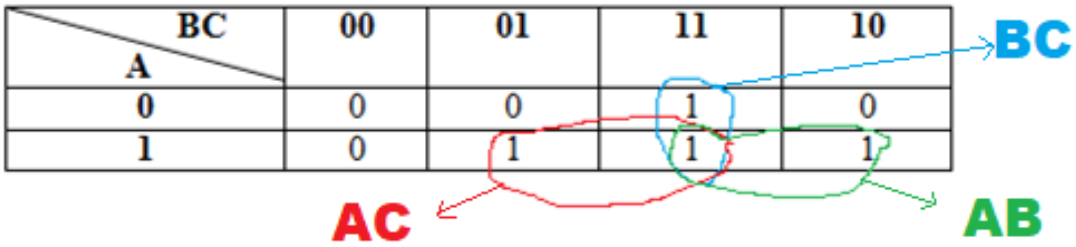
DESIGN SOLUTION:

Let us consider 3 individuals as A, B, C (inputs) and proposal as F (output). Also, Yes as '1' and No as '0'.

Step 1: Construct the Truth Table for the given scenario.

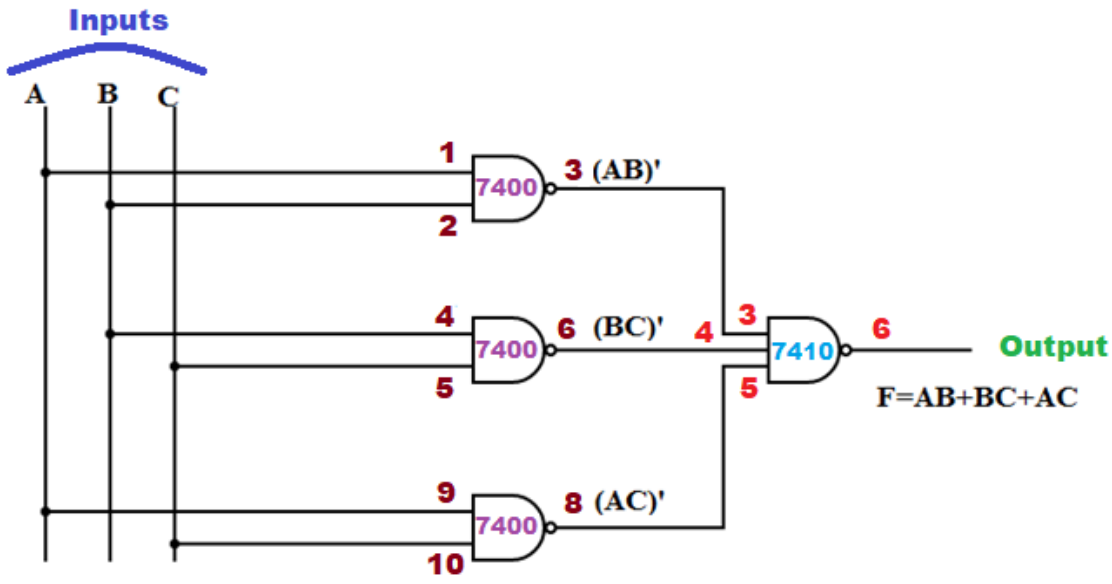
(Indiviual1) A	(Indiviual2) B	(Indiviual3) C	(Proposal) F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Step 2: Use Karnaugh Map (K-Map) to get the simplified expression for the proposal (F)



From the above figure, we can say that the simplified Boolean expression for the proposal (F) = AB + BC + AC.

Step 3: Construct the Logic circuit for the Boolean expression, F = AB + BC + AC using only NAND gates.



PROCEDURE:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT:

The truth table of the above circuit is verified.

VIVA QUESTIONS:

1. What is the Boolean expression for 3-input NAND gate?
2. The function NAND (a_1, a_2, ..., a_n) is logically equivalent to _____
3. What are the applications of NAND gates?

EXPERIMENT: 2 DESIGN OF BINARY TO GRAY CODE CONVERTER

AIM: To design 3-bit binary to gray code converter circuit using basic gates.

LEARNING OBJECTIVE:
To learn to realize 3-bit binary to gray code converter using basic gates.

COMPONENTS REQUIRED:
IC 7486, Patch Cords & IC Trainer Kit.

THEORY:
Binary Code: It is weighted code. i.e., it is a code in which weight is assigned to every symbol position in the code word. The positional weights in binary code are shown below:

Binary Code: -----> ….. 2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4} …

Decimal ----------> ….. 16 8 4 2 1 1/2 1/4 1/8 1/16 …

Gray Code: It is a non-weighted code i.e., it does not have any specific/fixed weight assigned to each symbol position in the code word. The unique feature of Gray code is that at a time only "one" bit changes In other words, in Gray code every new code differs from the previous in terms of one single bit.

Gray Code uses: Used in analog to digital converters, as well as being used for error correction in digital communication.

DESIGN SOLUTION:

Step 1: Construct the Truth Table for the given problem

Decimal Equivalent	Binary			Gray Code		
	B2	B1	B0	G2	G1	G0
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**0**	**0**	**1**
2	0	1	0	**0**	**1**	**1**
3	0	1	1	**0**	**1**	**0**
4	1	0	0	**1**	**1**	**0**
5	1	0	1	**1**	**1**	**1**
6	1	1	0	**1**	**0**	**1**
7	1	1	1	**1**	**0**	**0**

Step 2: Use Karnaugh Map (K-Map) to get the simplified expression for G2, G1 & G0.

For G2:

B2 \ B1B0	00	01	11	10
0	0	0	0	0
1	1	1	1	1

$G2 = B2$

For G1:

B2 \ B1B0	00	01	11	10
0	0	0	1	1
1	1	1	0	0

$\overline{B2}\ B1$

$B2\ \overline{B1}$

$G1 = \overline{B2}\ B1 + B2\ \overline{B1}$

$G1 = B2 \oplus B1$

For G0:

B2 \ B1B0	00	01	11	10
0	0	1	0	1
1	0	1	0	1

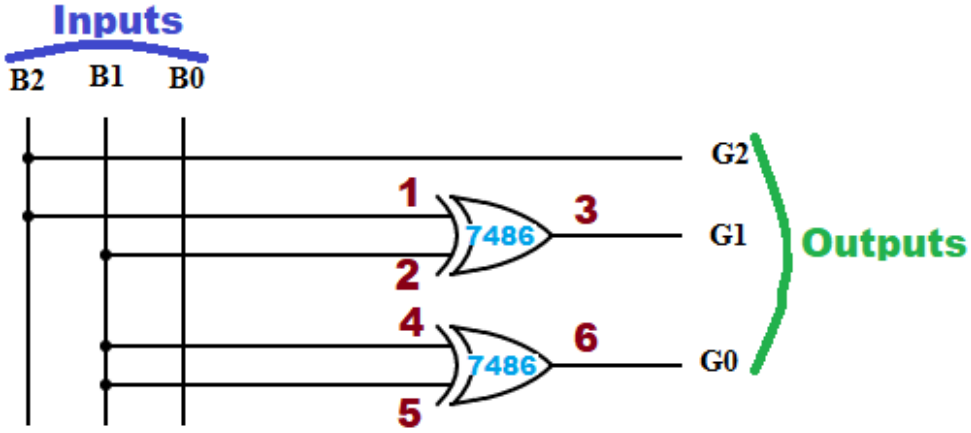
$B1\ \overline{B0}$

$\overline{B1}\ B0$

$G0 = B1\ \overline{B0} + \overline{B1}\ B0$

$G0 = B1 \oplus B0$

Step 3: Construct the Logic circuit using basic gates based on the simplified Boolean expressions obtained for G2, G1 & G0.



PROCEDURE:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT: The truth table of the above circuit is verified.

VIVA QUESTIONS:

1. Gray code is also known as _____
2. List any two disadvantages of gray code.

EXPERIMENT: 3 DESIGN FULL ADDER USING SUITABLE DECODER

AIM: To realize Full Adder using 3-to-8 decoder IC and 4 input NAND gates.

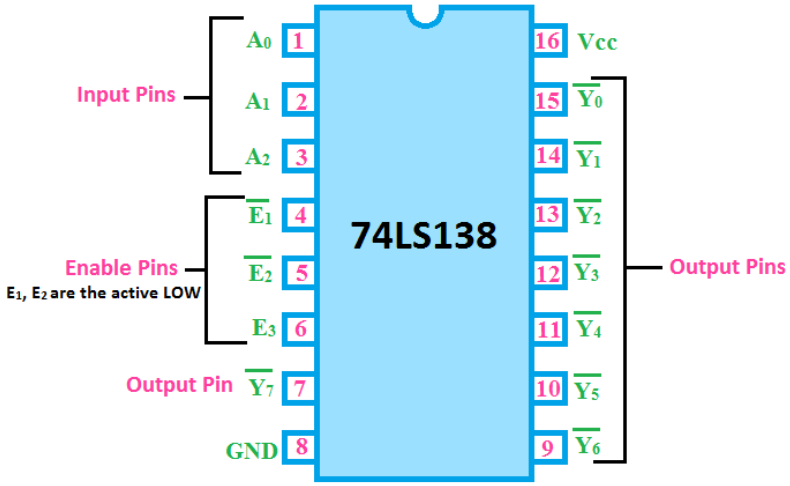
LEARNING OBJECTIVE:
To learn about applications of decoder.

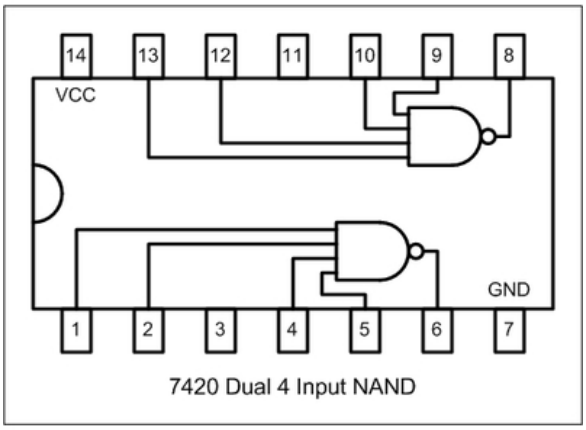
COMPONENTS REQUIRED:
IC74138, IC7420, Patch chords & IC Trainer Kit

THEORY:
Decoder: The de-multiplexer or decoder performs the reverse operation of a Multiplexer. It accepts a single input and distributes it over several outputs. The input code determines to which output the data input will be transmitted.

IC 74138: The IC 74138 is a high-speed dual input Decoder Multiplexer. This device has two independent decoders, each accepting two binary weights inputs (S_0 & S_1) and providing four mutually exclusive active – low outputs (O_0 – O_3). Each decoder has an active- low enable (E). When E=1, every output is forced high.

PIN DIAGRAMS:





DESIGN SOLUTION:

Step 1: Construct the Truth Table for the given problem

Inputs			Outputs	
A	**B**	**C**	**Sum**	**Carry**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Step 2: Represent Sum & Carry in Boolean expression by using the above truth table.

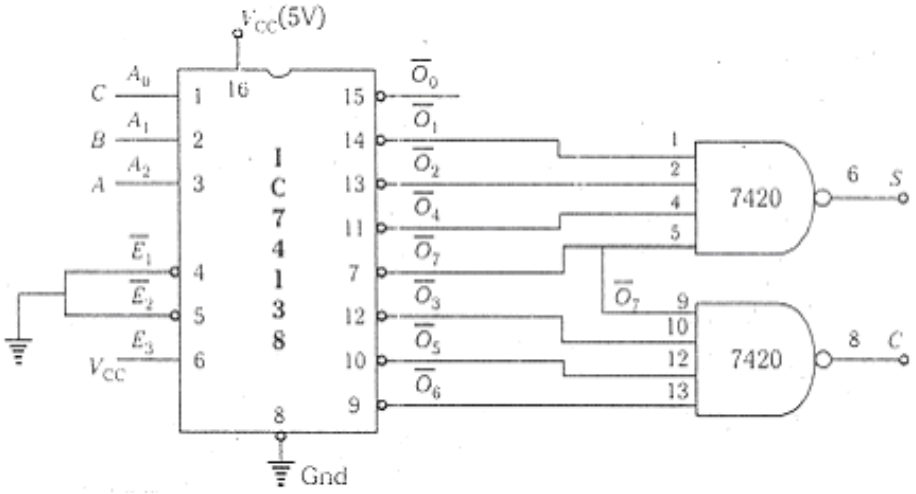
Sum = $\sum$ (m1, m2, m4, m7)

= $\sum m$ (1, 2, 4, 7)

Carry = $\sum$ (m3, m5, m6, m7)

= $\sum m$ (3, 5, 6, 7)

Step 3: Construct the Logic circuit of Full adder using 3-to-8 decoder and 4-input bit NAND gate.



PROCEDURE:

1. Verify the components and patch chords.
2. Make connections as shown in circuit diagram
3. Give supply to the trainer kit.
4. Provide the input data to the circuits via switches.
5. Record and verify the output for each combination and verify the truth table.

RESULT: Truth table is verified.

VIVA QUESTIONS:

1. What are the applications of adders?
2. What are the applications of decoders?
3. What are the applications of IC7420?

EXPERIMENT: 4 APPLICATION of MUX

AIM: A lawn sprinkling system is controlled automatically by certain combinations of the following variables:
Season (S = 1, if summer; 0 otherwise)
Moisture content of soil (M = 1, if high; 0 if low)
Outside temperature (T = 1, if high; 0 if low)
Outside humidity (H = 1, if high; 0 if low)
The sprinkler is turned on under any of the following circumstances:

(i) The moisture content is low in winter.
(ii) The temperature is high and the moisture content is low in summer.
(iii) The temperature is high and the humidity is high in summer.
(iv) The temperature is low and the moisture content is low in summer.
(v) The temperature is high and the humidity is low.

Implement using suitable multiplexer (Use 8x1 MUX).

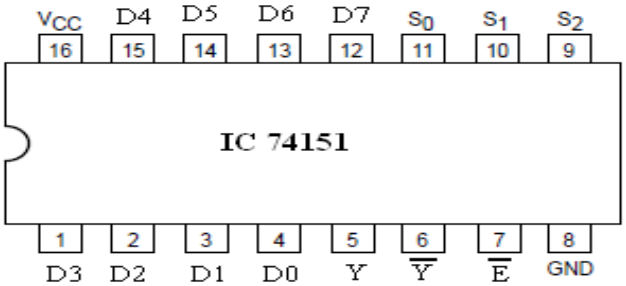
LEARNING OBJECTIVE:
To learn about various applications of multiplexer.

COMPONENTS REQUIRED:
IC 74151, IC 7404, Patch Cords & IC Trainer Kit.

THEORY:
Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. By using control signals (select lines) we can select any input to the output. Multiplexer is also called as data selector because the output bit depends on the input data bit that is selected. The general multiplexer circuit has 2^n input signals, n control/select signals and 1 output signal.

De-multiplexers perform the opposite function of multiplexers. They transfer a small number of information units (usually one unit) over a larger number of channels under the control of selection signals. The general de-multiplexer circuit has 1 input signal, n control/select signals and 2^n output signals. De-multiplexer circuit can also be realized using a decoder circuit with enable.

PIN DIAGRAM:



DESIGN SOLUTION:

Let us consider 4 individuals as S, M, T, and H (inputs) and sprinkler as F (output). Also, ON as '1' and OFF as '0'.

Step 1: Construct the Truth Table for the given scenario.

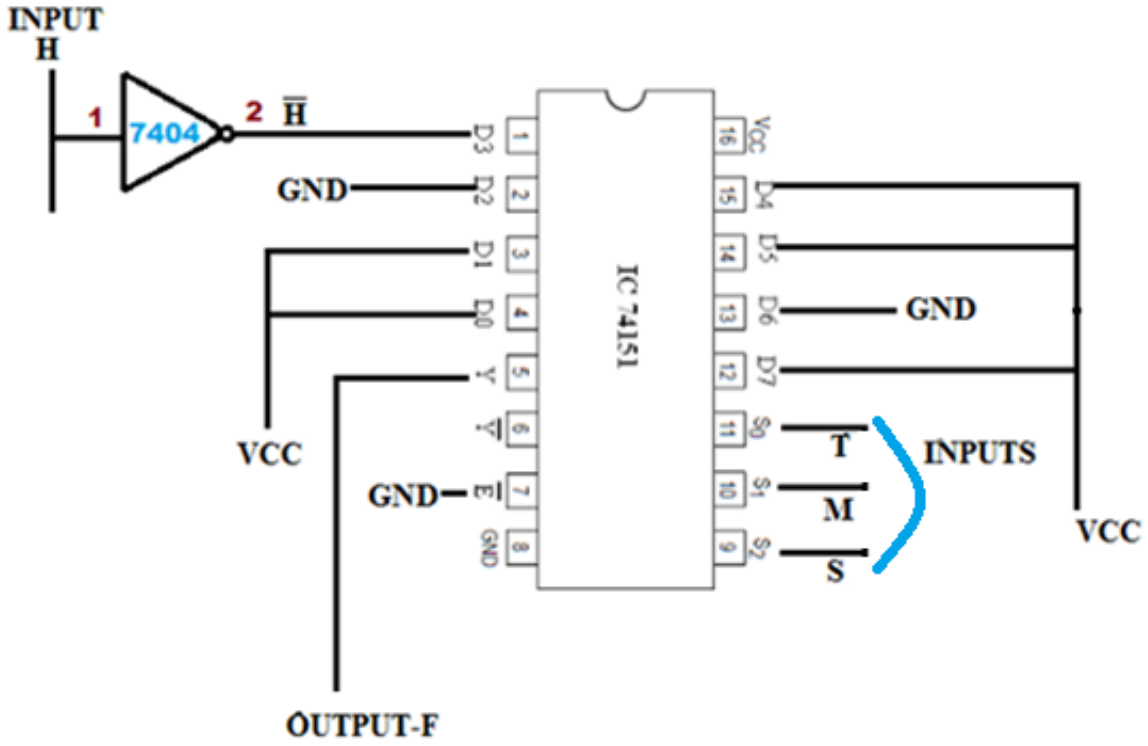
Decimal	Season-S	Moisture-M	Temperature-T	Humidity-H	Output-F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

From the above Truth Table, F = ∑m (0, 1, 2, 3, 6, 8, 9, 10, 11, 14, 15).

Step 2: Convert 16:1 to 8:1 MUX by taking H as the entered variable.

When	000	001	010	011	100	101	110	111
H = 0	1 0	1 2	0 4	1 6	1 8	1 10	0 12	1 14
H = 1	1 1	1 3	0 5	0 7	1 9	1 11	0 13	1 15
	D0 = 1	**D1 = 1**	**D2 = 0**	**D3 = $\overline{H}$**	**D4 = 1**	**D5 = 1**	**D6 = 0**	**D7 = 1**

Step 3: Construct the Logic circuit to realize the given problem using 74151 MUX IC



PROCEDURE:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT:

Lawn sprinkling system has been realized using multiplexer IC 74151.

VIVA QUESTIONS:

1. What is a multiplexer?
2. What is a de-multiplexer?
3. What are the applications of multiplexer and de-multiplexer?
4. What is the difference between multiplexer & de-multiplexer?

EXPERIMENT: 5 STUDY of FLIP FLOPS

AIM: To verify state tables of JK and D Flip Flops using logic gates.

LEARNING OBJECTIVE:
To learn about JK and D Flip Flops.

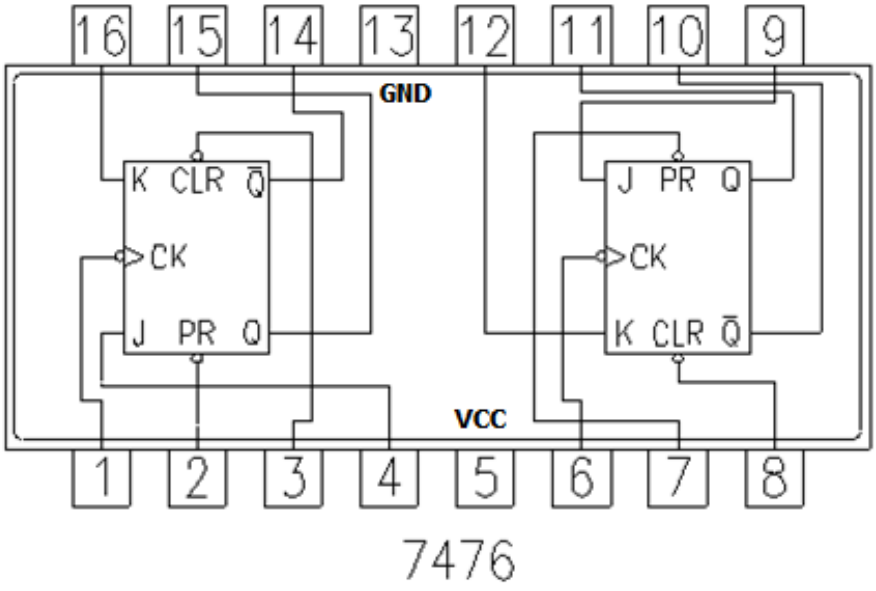
COMPONENTS REQUIRED:
IC 7476, IC 7474, IC 7410, IC 7400, Patch Cords & IC Trainer Kit.

THEORY:
A Flip flop is a bistable electronic circuit that has two stable states. That is, its output is either +5V (logic 1) or 0V (logic 0). A Flip Flop can be referred as memory device since its output will remain unchanged until its input is not changed. It is used to store one binary digit. There four different types of flip flops. They are SR, JK, D and T flip flop.

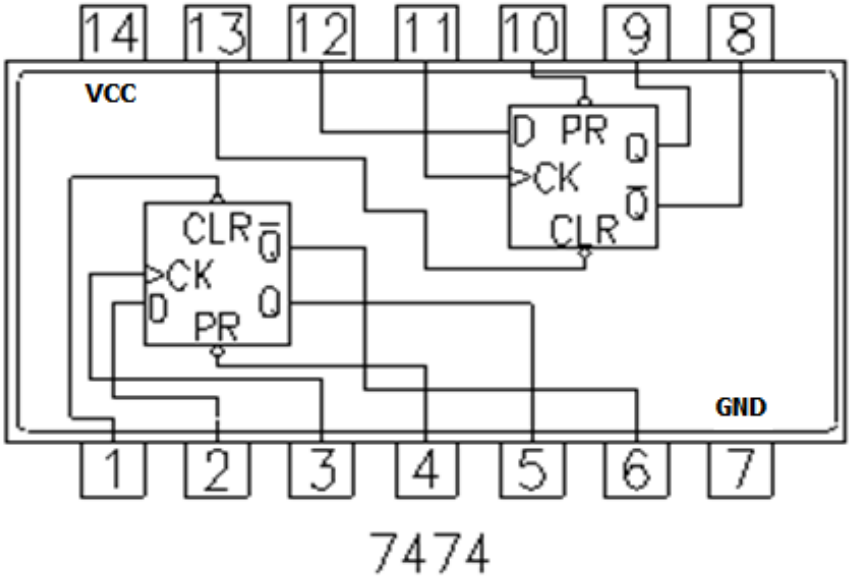
JK Flip Flop: A JK flip-flop is a sequential bi-state single-bit memory device named after its inventor by Jack Kilby. In general, it has one clock input pin (CLK), two data input pins (J and K), and two output pins (Q and $\bar{Q}$). JK flip flop can either be triggered upon the leading-edge of the clock or on its trailing edge and hence can either be positive- or negative- edge-triggered, respectively.

D Flip Flop: D flip flop or Data flip flop is a type of flip flop that has only one data input that is 'D' and one clock pulse input with two outputs Q and $\bar{Q}$. This Flip Flop is also called a delay flip flop because when the input data is provided into the D flip flop, the output follows the input data delay by one clock pulse.

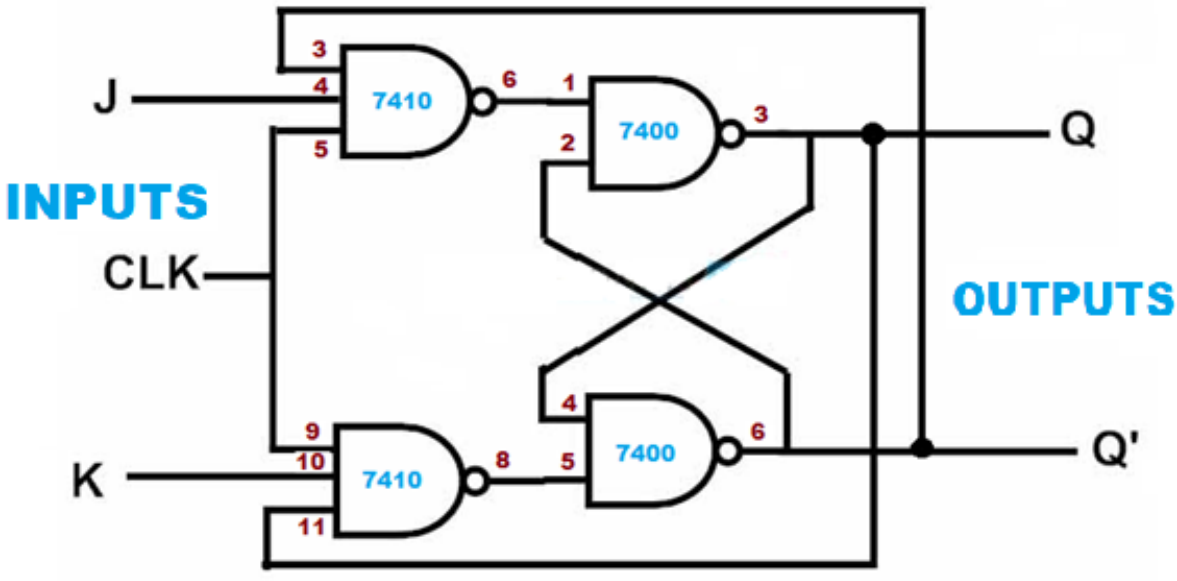
PIN DIAGRAMS:



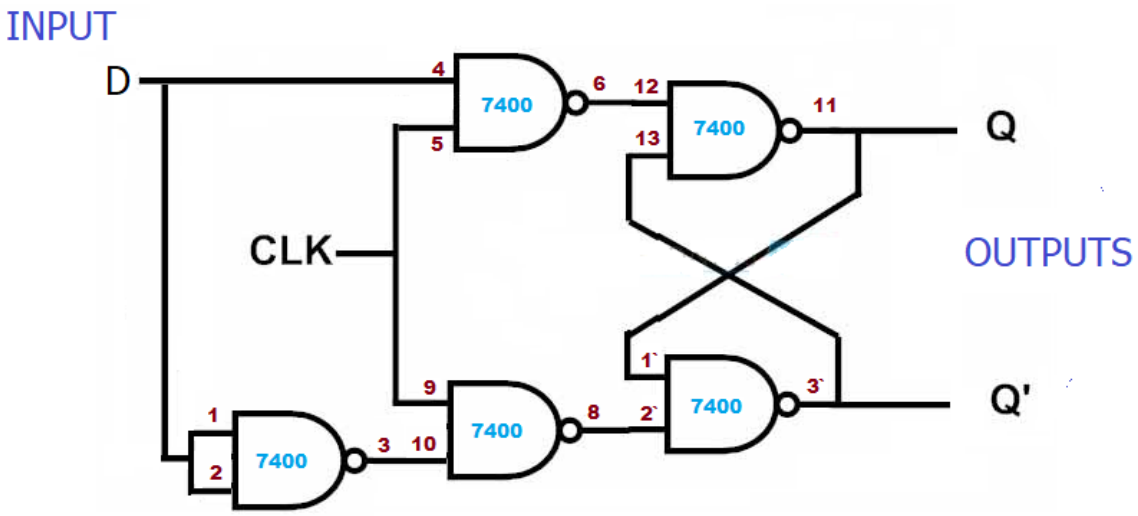
7476



CIRCUIT DIAGRAMS using only NAND gates:



JK Flip-Flop



D Flip-Flop

STATE TABLES:

JK Flip-Flop

CLK	J	K	Q	$\bar{Q}$
0	X	X	No Change	
1	0	0	No Change	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle	

D Flip-Flop

CLK	D	Q	$\bar{Q}$
0	X	No Change	
1	1	1	0
1	0	0	1

PROCEDURE:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT: State tables of JK and D Flip-Flops has been verified.

VIVA QUESTIONS:

1. What are flip flops?
2. What is an excitation table?
3. List out the applications of flip flop.
4. What is race around condition?

EXPERIMENT: 6 MASTER-SLAVE D Flip-Flop

AIM:

To implement Master-Slave D Flip-Flop using only NAND gates.

LEARNING OBJECTIVE:

To learn about Master-Slave D Flip-Flop.

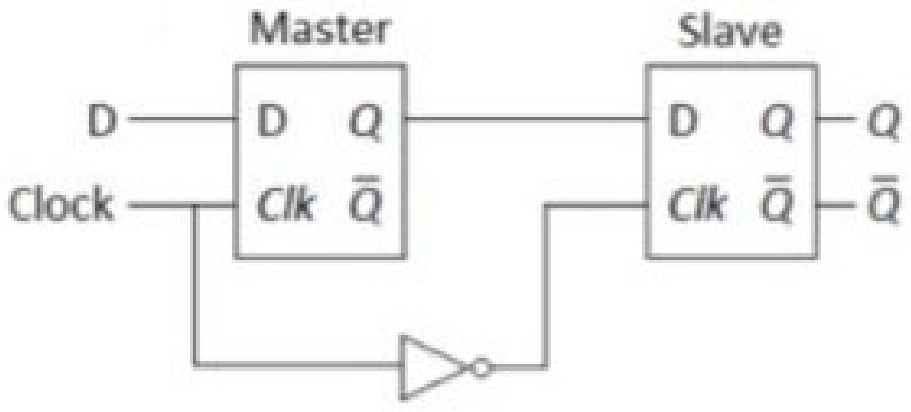
COMPONENTS REQUIRED:

IC 7400, Patch Cords & IC Trainer Kit.

THEORY:

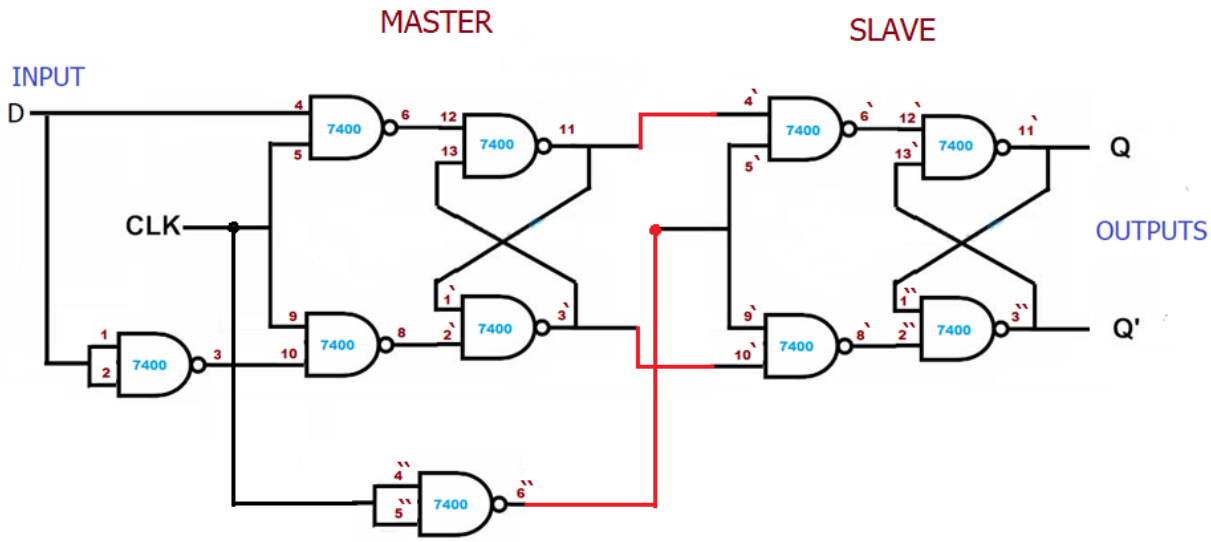
Master-slave is a combination of two flip-flops connected in series, where one acts as a master and another act as a slave. Each flip-flop is connected to a clock pulse complementary to each other, i.e., if the clock pulse is in high state, the master flip-flop is in enable state, and the slave flip-flop is in disable state, and if clock pulse is low state, the master flip-flop is in disable state, and the slave flip flop is in enabled state.

In this master-slave also, two D flip-flop connected to each other in series with clock pulse invited to each other. The basic mechanism of this master-slave is also similar to other master-slave flip-flops. D master-slave flip-flop can be level triggered, or edge triggered.

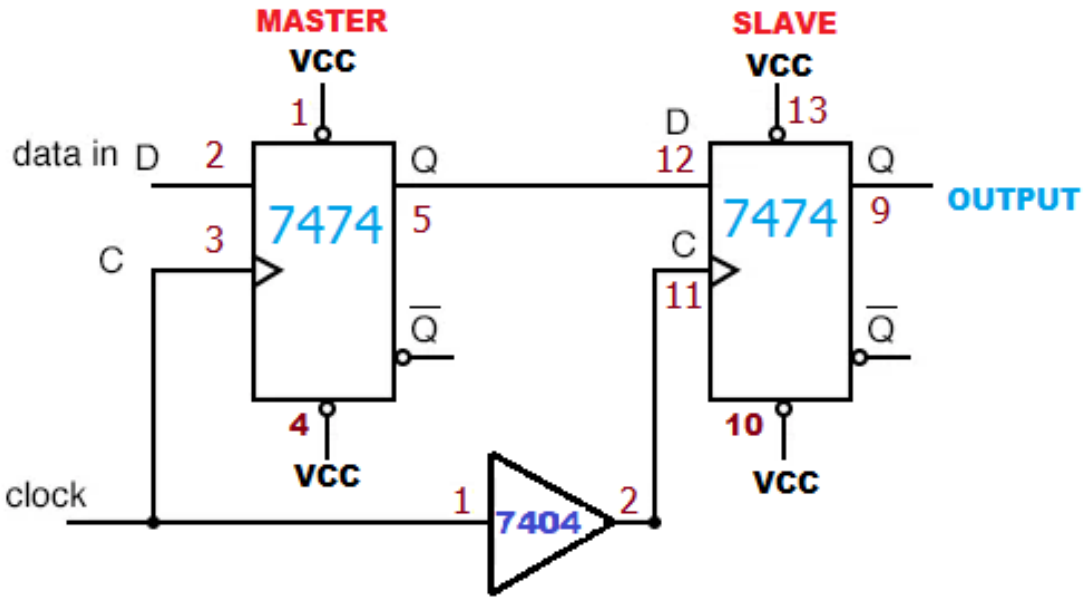


Master-slave D flip-flop Diagram

CIRCUIT DIAGRAM:



OR



TRUTH TABLE:

D	Q (Previous)	Clock	Q
0	0	1	0
0	1	1	0
1	0	1	1
1	1	1	1
0	0	0	0
0	1	0	1
1	0	0	0
1	1	0	1

PROCEDURE:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT: Truth Table of Master-Slave D Flip Flops has been verified.

VIVA QUESTIONS:

1. Where Master-Slave flip-flop can be used?
2. Master-Slave flip flop is also referred as?
3. D stands for what in D Flip-Flop?

EXPERIMENT: 7 CONVERSION of JK Flip-Flop to T Flip-Flop

AIM:

To design and demonstrate the conversion of JK Flip-Flop to T Flip-Flop.

LEARNING OBJECTIVE:

To learn about conversion of JK Flip-Flop to T Flip-Flop.

COMPONENTS REQUIRED:

IC 7476, Patch Cords & IC Trainer Kit.

THEORY:

J-K Flip-Flop: JK flip-flop shares the initials of Jack Kilby, who won a Nobel prize for his fabrication of the world's first integrated circuit, some people speculate that this type of flip-flop was named after him because a flip-flop was the first device that Kilby build when he was developing integrated circuits. J-K flip-flop is the gated version of SR flip-flop with an addition of extra input i.e., clock input. It prevents invalid output conditions when both the inputs are at the same value.

T Flip-Flop: T flip-flop means Toggle flip-flop. It changes the output on each clock edge and gives an output that is half the frequency of the signal to the input.

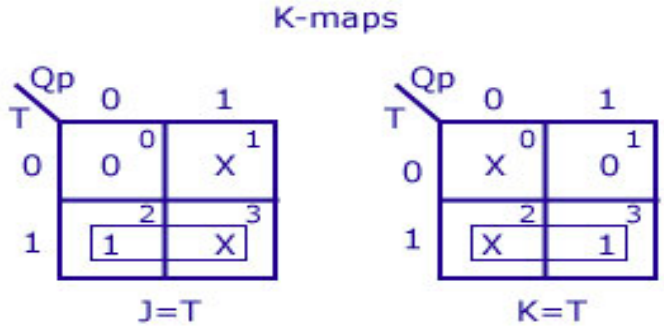
DESIGN SOLUTION:

Step 1: Construct the conversion table.

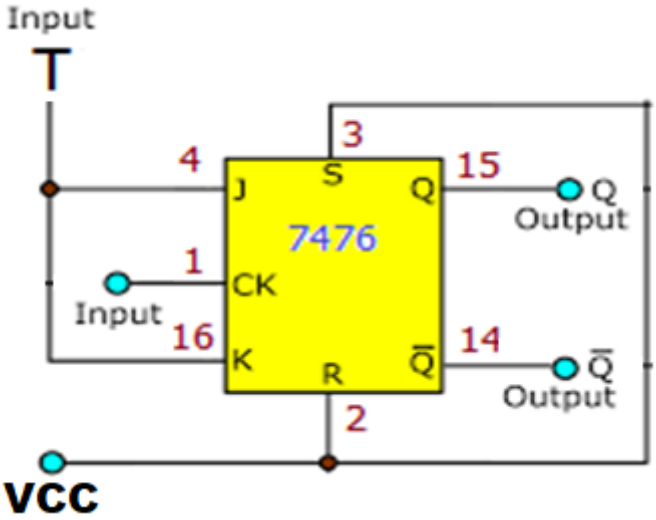
Conversion Table

T Input	Outputs		J-K Inputs	
	Qp	Qp+1	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

Step 2: Using the K-map, find the Boolean expression for J and K in terms of T.



Step-3: Construct the circuit diagram using 7476 IC.



TRUTH TABLE:

T Flip-Flop

CLK	T	Q (Previous)	Q	State
1	0	0	0	No Change
1	0	1	1	
1	1	0	1	Toggles
1	1	1	0	

PROCEDURE:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT: Truth Table of T Flip-Flop has been verified.

VIVA QUESTIONS:

1. Where T flip-flop can be used?
2. Difference between JK and T Flip-Flops.

EXPERIMENT: 8 3-bit SERIAL in SERIAL out SHIFT REGISTER

AIM:

To design 3-bit serial in serial out shift register using D Flip Flop's.

LEARNING OBJECTIVE:

To learn about serial in serial out shift register using D Flip Flop's.

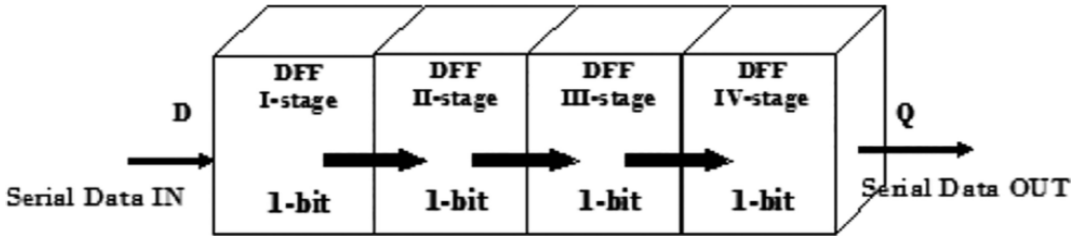
COMPONENTS REQUIRED:

IC 7474, Patch Cords & IC Trainer Kit.

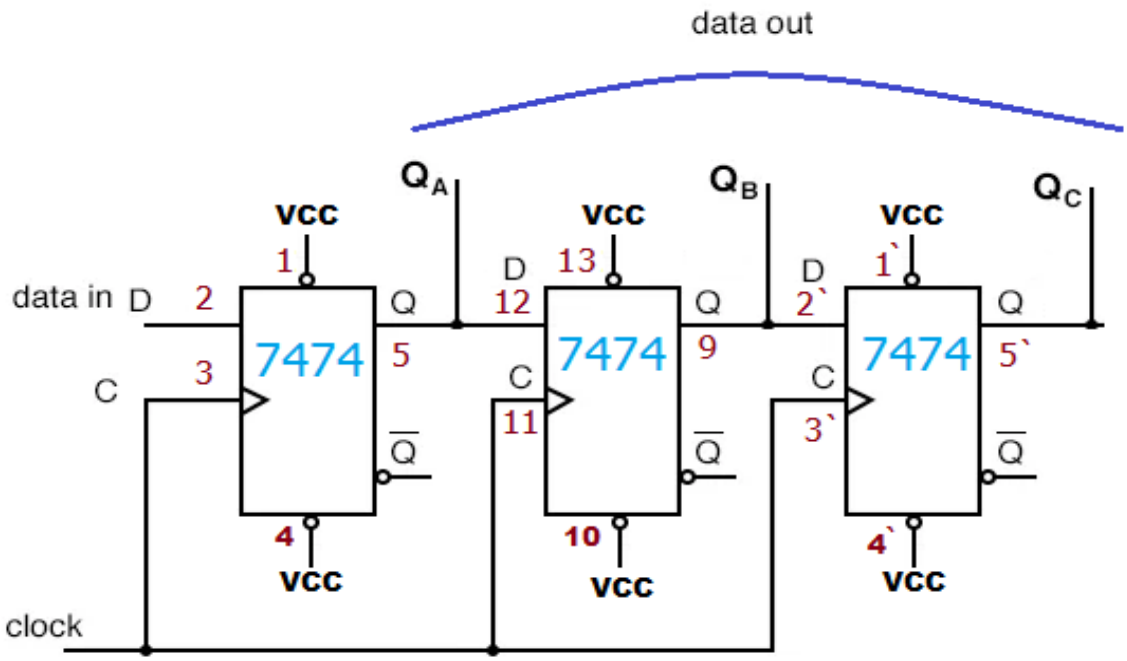
THEORY:

The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as Serial-In Serial-Out shift register. Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register.

The logic circuit given below shows a serial-in serial-out shift register. The circuit consists of four D flip-flops which are connected in a serial manner. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.



CIRCUIT DIAGRAM:



TRUTH TABLE:

Clock pulses	Data In	Q_A	Q_B	Q_C
0	0	0	0	0
1	1	1	0	0
2	0	0	1	0
3	1	1	0	1

PROCEDURE:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

RESULT:

Truth Table of 3-bit serial in serial out shift register using D Flip Flop's has been verified.

VIVA QUESTIONS:

1. Where shift registers can be used?
2. List different types of shift registers based on applying inputs and accessing outputs.
3. What is the output after two clock pulses for a bit sequence 1101 serially entered into a 4-bit parallel out shift register which is initially clear?

EXPERIMENT: 9 2-bit SYNCHRONOUS COUNTER using JK FLIP FLOP

AIM:

To design 2-bit synchronous counter for the given sequence using JK Flip Flop.

LEARNING OBJECTIVE:

To learn about synchronous counter using JK Flip Flops.

COMPONENTS REQUIRED:

IC 7476, Patch Cords & IC Trainer Kit.

THEORY:

The synchronous counter can be defined as, a counter which uses a clock signal for transforming their transition. So, these counters mainly depend on the input of the clock to modify state values. In this counter, all flip flops (FFs) are associated with the same clock signal to activate simultaneously. An alternate name of this counter is a simultaneous counter where there is no ripple effect & propagation delay in these counters.

As compared to synchronous, asynchronous type designing is very simple but the asynchronous counter has a limitation of maximum operating frequency. To overcome this limitation, these counters are mainly designed by providing simultaneous clocking so, the output changes in synchronization through the input of the clock.

DESIGN SOLUTION:

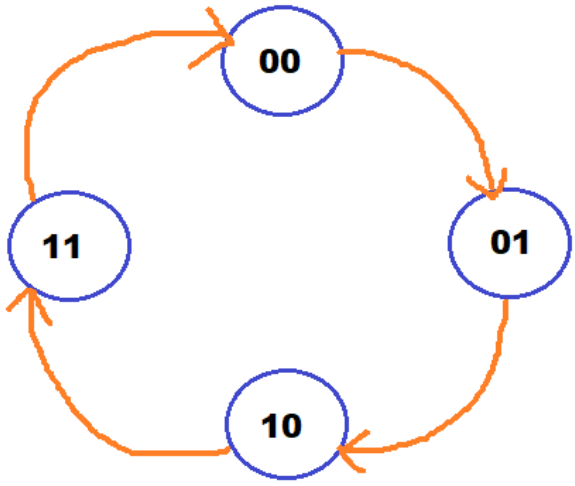
Step 1: Find the number of Flip Flops required

Here, number of Flip Flops required for designing 2-bit synchronous counter is 2.

Step 2: Excitation Table of JK Flip Flop

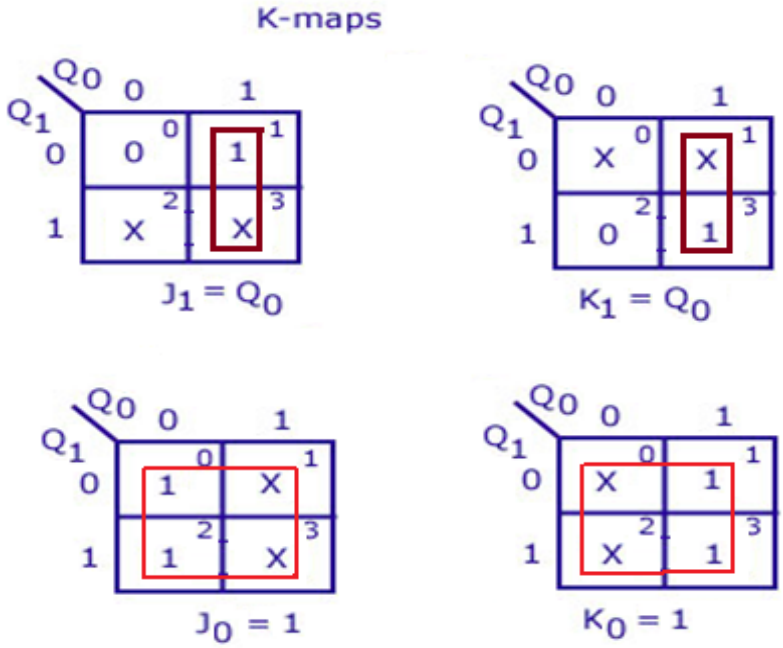
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3: State diagram and State Table

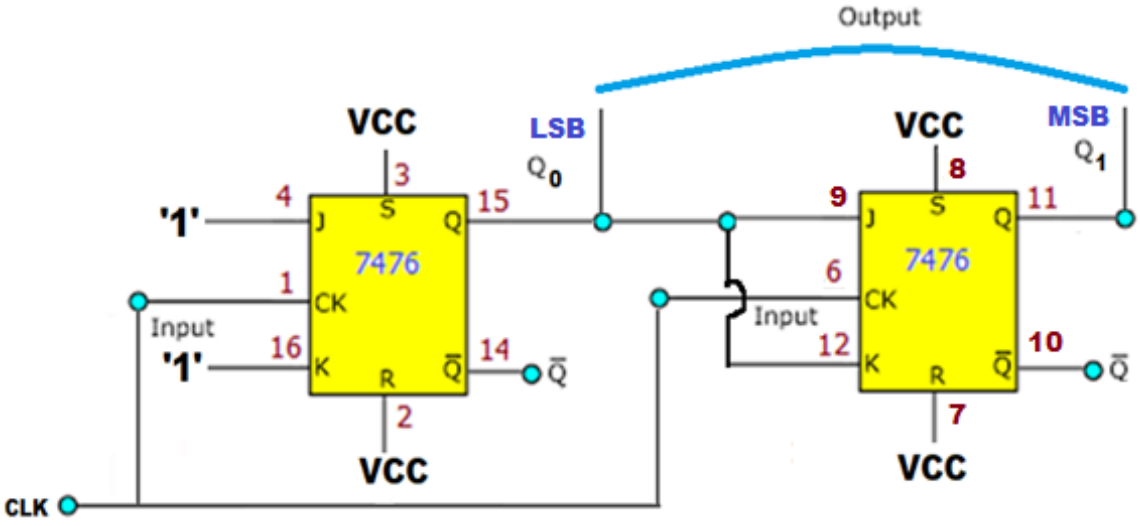


Present State		Next State		Inputs			
Q_1	Q_0	Q_1^+	Q_0^+	J_1	K_1	J_0	K_0
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

Step 4: Using the K-map, find the Boolean expression for J_1, K_1 and J_0, K_0.



Step 5: Circuit Diagram



PROCEDURE:

1. Check all the components and patch chords
2. Prepare the excitation table of the FF's and determine the FF inputs which must be present for the desired next state from the present state.
3. In terms of FF outputs prepare Karnaugh map for each FF input and simplify the Karnaugh map's expression into a minimized form.
4. Make connection according to the expression.
5. Verify the truth table.

RESULT: Implementation of 2-bit synchronous counter is verified.

VIVA QUESTIONS:

1. What are synchronous counters?
2. What are the advantages of synchronous counters?
3. What is an excitation table?

VHDL PART

EXPERIMENT: 0 Write the Verilog/VHDL code for basic gates. Simulate and verify its working.

VHDL Code

```
Port:   a: in STD_LOGIC;
        b: in STD_LOGIC;
        c: out STD_LOGIC;
        d: out STD_LOGIC;

begin
        c <= a and b;
        d <= a or b;
```

Link for Tool Introduction with Basic gate Simulation: https://youtu.be/wE1lqB8kz2s

EXPERIMENT: 1 Write the Verilog/VHDL code for 8:1 MUX. Simulate and verify its working.

VHDL Code

```
Port:   d: in STD_LOGIC_ VECTOR (7 downto 0);
        s: in STD_LOGIC_ VECTOR (2 downto 0);
        y: out STD_LOGIC;


begin
        y <=  d(0) when s =”000” else
              d(1) when s =”001”  else
              d(2) when s =”010”  else
              d(3) when s =”011”  else
              d(4) when s =”100”  else
              d(5) when s =”101”  else
              d(6) when s =”110”  else
              d(7);
```

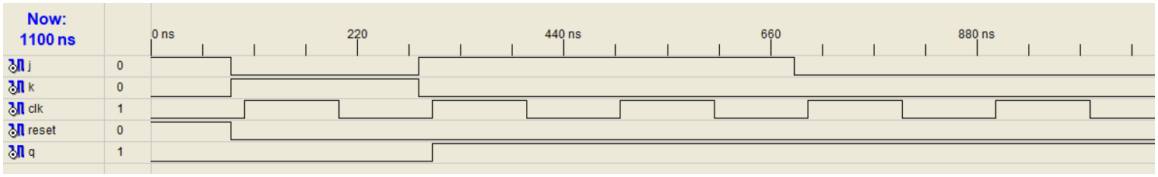
Link for 8:1 MUX Simulation: https://youtu.be/LDPOjAvsCX8

EXPERIMENT: 2 Write the Verilog/VHDL code for a JK Flip-Flop and D Flip-Flop with positive-edge triggering. Simulate and verify its working.

VHDL Code for a JK Flip-Flop

```
Port:   j: in STD_LOGIC;
        k: in STD_LOGIC;
        clk: in STD_LOGIC;
        reset: in STD_LOGIC;
        q: out STD_LOGIC;
begin
        process (j, k, clk, reset)
        begin
                if(reset = '1') then
                        q <= '0';
                elsif (clk = '1') then
                        if(j /= k) then
                                q <= j;
                        elsif(j = '1' and k = '1') then
                                q <= not j;
                        end if;
                end if;
        end process;
```

Simulation Waveform:



VHDL Code for a D Flip-Flop

```
Port:   data: in STD_LOGIC;
        clk: in STD_LOGIC;
        reset: in STD_LOGIC;
        q: out STD_LOGIC;
begin
        process (clk, reset)
        begin
                if reset='1'
                        then q <='0';
                elsif (clk ='1')
                       then q <= d;
                end if;
        end process;
```

Link for D Flip-Flop Simulation: https://youtu.be/5BiN4nQrzrg

EXPERIMENT: 3 Write the Verilog/VHDL code for a mod-8 down counter. Simulate and verify its working.

VHDL Code

```
Port:   c: in STD_LOGIC;
        r: in STD_LOGIC;
        q: inout STD_LOGIC_VECTOR (2 downto 0);

begin
        process (c, r)
        begin
                if (r ='1')
                    then q <= "111";
                elsif (c = '1')
                    then q <= q - 1;
                end if;
        end process;
```

Simulation Waveform: