

FOREWORD

It is my great pleasure to present this laboratory manual for Second year engineering students for the subject of Digital Electronics keeping in view the vast coverage required for visualization of concepts of Digital Electronics with simple language.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9000 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9000 Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr S.D.Deshmukh
Principal

LABORATORY MANUAL CONTENTS

This manual is intended for the Second year students of CSE branches in the subject of Digital Electronics . This manual typically contains practical/Lab Sessions related Digital Electronics covering various aspects related the subject to enhanced understanding.

Students are advised to thoroughly go though this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Prof.D.S.Deshpande
HOD, CSE

Ms. Neha R. Khatri
Assist Prof, CSE Dept.

DOs and DON'Ts in Laboratory:

1. Make entry in the Log Book as soon as you enter the Laboratory.
2. All the students should sit according to their roll numbers starting from their left to right.
3. All the students are supposed to enter the terminal number in the log book.
4. Do not change the terminal on which you are working.
5. All the students are expected to get at least the algorithm of the program/concept to be implemented.
6. Strictly observe the instructions given by the teacher/Lab Instructor.

Instruction for Laboratory Teachers::

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.
2. Students should be taught for taking the printouts under the observation of lab teacher.
3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

SUBJECT INDEX

1. Study of digital ICs & verification of Logic Gates
2. Study & verification of operation of Half Adder & Full Adder.
3. Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
4. Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.
5. Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
6. Design and develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.
7. Design and implement a mod-n ($n < 8$) synchronous up counter using JK Flip-Flop ICs and demonstrate its working.
8. Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.
9. Design and implement a ring counter using 4-bit shift register and demonstrate its working.
10. Design and develop the Verilog / VHDL code for switched tail counter. Simulate and verify its working.
11. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ($n \leq 9$) and demonstrate its working.
12. Design a 4 bit R-2R ladder D/A converter using op-amp. Determine its accuracy and resolution.

Experiment No.1

Aim: To Study logic gates such as AND, OR, NOT, NAND, NOR, XOR

Apparatus: Bread board, wires, IC-7402(AND),7432(OR),7404 (NOT)

Theory:

1. AND:

Logical AND operation is defined as “**the output is 1 iff all the inputs are 1**”

Circuit of logical AND is shown below. It has N inputs ($N \geq 2$) and one output. Digital signals are applied at the input terminal marked A,B,C...,N, the other terminal being grounded(not shown in diagram)The output is obtained at the terminal marked Y, and it is also a digital signal.

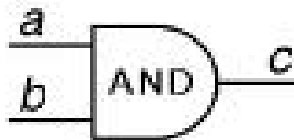


Fig1. LOGIC DIAGRAM OF AND GATE

Mathematically, **AND** operation is written as

$$Y=A \text{ AND } B \text{ AND } C$$

$$Y=A .B. C$$

$$Y=ABC \dots N$$

Truth Table for AND operation

INPUTS		OUTPUTS
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

2. OR:

Logical **OR** operation is defined as “**the output is 1 if at least one of the inputs is 1**”.

Circuit of logical OR is shown below. It has N inputs ($N \geq 2$) and one output. Digital signals are applied at the input terminal marked A, B, C...,N, the other terminal being grounded(not shown in diagram).The output is obtained at the terminal marked Y, and it is also a digital signal.



Fig 2.LOGIC DIAGRAM OF OR GATE

Mathematically,

OR operation is written as

$$Y=A \text{ OR } B \text{ OR } C$$

$$Y=A +B+C+\dots+N$$

Truth Table for **OR** operation

INPUTS		OUTPUTS
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3.NOT:

Logical NOT operation is also called as Inverter. It has one input (A) and one output (Y).Its logic Equation is written as

$$Y = \text{NOT } A$$

And is read as “Y equals not A” or “Y equals complement of A”.

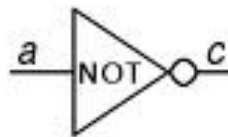


Fig 3.LOGIC DIAGRAM OF NOT GATE

INPUT	OUTPUTS
A	Y
0	1
1	0

4. NAND Gate:

The output is 1 when either of inputs A or B is 1, or if neither is 1. In other words, it is normally 1, going 0 only if both A and B are 1.

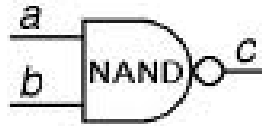


Fig 4.LOGIC DIAGRAM OF NAND GATE

INPUTS		OUTPUTS
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

4. NOR Gate:

A HIGH output (1) results if both the inputs to the gate are LOW (0). If one or both input is HIGH (1) or LOW output (0) results. NOR is the result of the negation of the OR operator. NOR is a functionally complete operation -- combinations of NOR gates can be combined to generate any other logical function. By contrast, the OR operator is *monotonic* as it can only change LOW to HIGH but not vice versa.

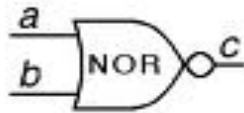


Fig 5.LOGIC DIAGRAM OF NOR GATE

INPUTS		OUTPUTS
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

5. XOR Gate:

A HIGH output (1) results if one, and only one, of the inputs to the gate is HIGH (1). If both inputs are LOW (0) or both are HIGH (1), a LOW output (0) results.

XOR gate is short for exclusive OR. This means that precisely one input must be 1 (true) for the output to be 1 (true). A way to remember XOR is "one or the other but not both."

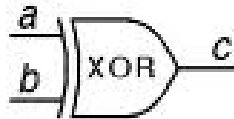


Fig 6.LOGIC DIAGRAM OF XOR GATE

INPUTS		OUTPUTS
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Procedure: -

- 1) To pin number 14 Of all IC's Vcc is applied & pin number 7 is grounded.
- 2) Assemble the circuit on breadboard according to the pin configuration.
- 3) Give the logical inputs and check for the proper output.

Conclusion: Hence verified the logical AND, OR, NOT, NAND, NOR, XOR Operation.

Experiment No.2

Aim: To study & verification of operation of half and full adder.

Apparatus: Bread board, wires, IC-7402(AND), 7432(OR), 7486 (XOR)

Theory:

Half Adder:

Half Adder is a combinational circuit that performs addition of two bits. It has two inputs and two outputs. The two I/Ps are the two 1-bit numbers A and B designated as augend and addend bits. The two O/Ps are the sum 'S' of A and B and the carry bit, denoted by 'C'.

Truth Table of a half adder can be derived by performing binary addition of augend and addend bits as follows:

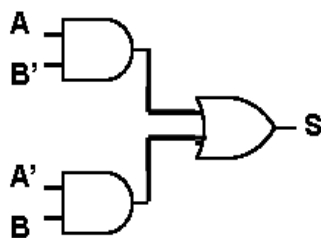
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

From the truth table, Boolean Expression can be derived as:

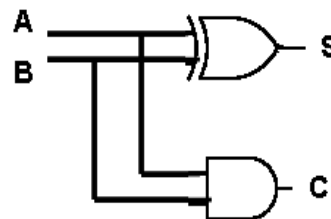
$$S = A'B + AB' = A \oplus B$$

$$C = AB$$

A Half Adder circuit can be implemented using AND & OR logic gates or by using XOR & AND logic gates. Both these implementations are shown in the image below:

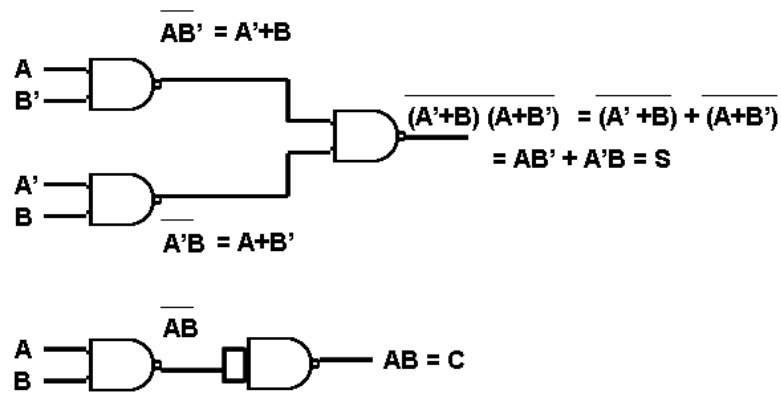


Half Adder implementation using AND-OR



Half Adder implementation using AND-XOR

Since NAND is considered as Universal Logic Gate which means that all other logic gates can be derived from it, below is the implementation of a **Half Adder** circuit using NAND logic gate:



Full Adder:-

Full Adder is a combinational circuit that performs addition of three bits. It consists of three inputs and two outputs. Two of the inputs denoted by A and B are augend and addend bits that are to be added, & third input denoted by C_i represents the carry bit from the previous lower significant position. The two O/Ps are the sum 'S' of A and B and the carry bit, denoted by C_o .

Truth Table of a full adder can be derived as follows:

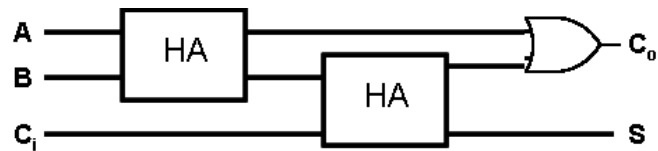
C_i	A	B	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

From the truth table, Boolean Expression can be derived as:

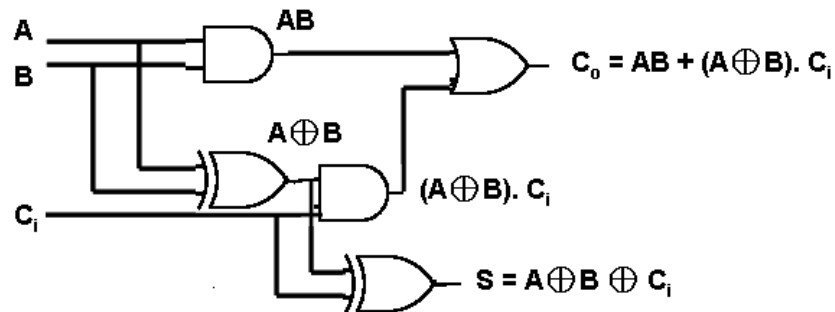
$$S = C_i \oplus A \oplus B$$

$$C_o = AB + (A \oplus B) C_i$$

A full adder circuit can be realized using half adder circuits as shown in the image below:



Using OR, XOR and AND logic gates, a **full adder** circuit can be implemented as below:



Procedure: -

- 1) To pin number 14 of all IC's V_{cc} is applied & pin number 7 is grounded.
- 2) Assemble the circuit on breadboard according to the pin configuration.
- 3) Give the logical inputs and check for the proper output.

Conclusion: Hence verified the truth table for half adder and full adder.

Experiment No.3

Aim: Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.

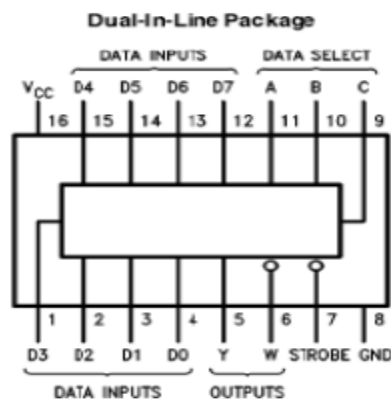
Apparatus: Bread board, wires, IC 74LS151.

Theory:

Simplify the function using MEV technique $f(a,b,c,d)=\sum m(2,3,4,5,13,15)+dc(8,9,10,11)$

Decimal	LSB	f	MEV map entry
0 } ₀	0000	0	0-----D ₀
1 } ₁	0001	0	
2 } ₂	0010	1	1-----D ₁
3 } ₃	0011	1	
4 } ₄	0100	1	1-----D ₂
5 } ₅	0101	1	
6 } ₆	0110	0	0-----D ₃
7 } ₇	0111	0	
8 } ₈	1000	X	X-----D ₄
9 } ₉	1001	X	
10 } ₁₀	1010	X	X-----D ₅
11 } ₁₁	1011	X	
12 } ₁₂	1100	0	d----D ₆
13 } ₁₃	1101	1	
14 } ₁₄	1110	0	d----D ₇
15 } ₁₅	1111	1	

Pin Diagram of Ics Used:



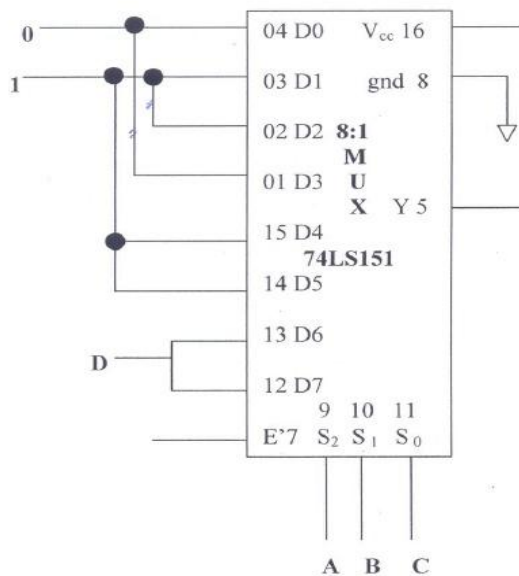
Theory:

Map Entered Variable Method:

Rules for entering values in a MEV K Map:

Rule No.	MEV f		Entry in MEV Map	Comments
1.	0	0	0	If function equals 0 for both values of MEV, enter 0 in appropriate cell of MEV Map
	1	0		
2.	0	1	1	If function equals 1 for both values of MEV, enter 1.
	1	1		
3.	0	0	MEV	If function equals MEV enter MEV
	1	1		
4.	0	1	-----	If the function is compliment of MEV enter MEV.
	1	0	MEV	
5.	0	-	-	If function equals don't care for both values of MEV, enter -
	1	-		
6.	0	-	0	If f=0 for MEV=0 and f=0 for MEV=1, enter 0.
	1	0		
7.	0	0	0	If f=0 for MEV=0 and f=- for MEV=1, enter 0.
	1	-		
8.	0	-	1	If f=-for MEV=0 and f=1 for MEV=1, enter 1.
	1	1		
9.	0	1	1	If f=1 for MEV=0 and f=- for MEV=-, enter -.
	1	-		

Circuit Diagram:



Procedure:

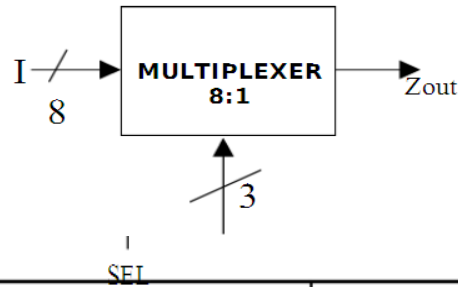
- (1) Verify all components and patch chords whether they are in good condition not.
- (2) Make connection as shown in the circuit diagram.
- (3) Give supply to the trainer kit.
- (4) Provide input data to circuit via switches.
- (5) Verify truth table sequence and observe outputs.

Conclusion: Thus we have studied the pin diagram of IC 74151 Multiplexer by solving simple example

Experiment No.4

Aim: Design and develop VHDL code for 8:1 MUX. Simulate and verify its working

Theory:



TruthTable

INPUTS			OUTPUTS
SEL (2)	SEL (1)	SEL (0)	Zout
0	0	0	I(0)
0	0	1	I(1)
0	1	0	I(2)
0	1	1	I(3)
1	0	0	I(4)
1	0	1	I(5)
1	1	0	I(6)
1	1	1	I(7)

-- VHDL code for 8 to 1 mux (Behavioral modeling).

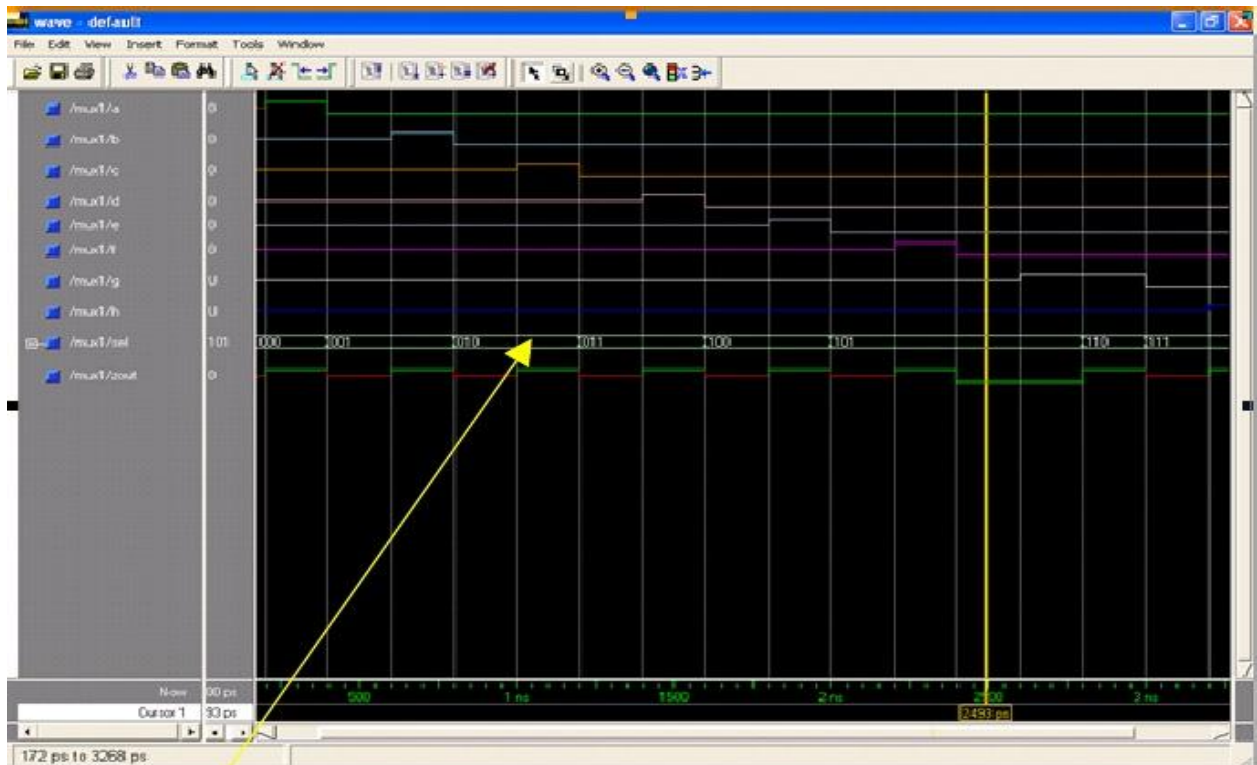
```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mux1 is
    Port ( I : in std_logic_vector(7 downto 0);
          sel : in std_logic_vector(2 downto 0);
          zout : out std_logic);
end mux1;

architecture behavioral of mux1 is
begin
    zout <= I(0) when sel="000" else
            I(1) when sel="001" else
            I(2) when sel="010" else
            I(3) when sel="011" else
            I(4) when sel="100" else
            I(5) when sel="101" else
            I(6) when sel="110" else
            I(7);
end behavioral;
    
```

Mux Simulation Results



Conclusion: - Hence we have studied 8:1 multiplexer.

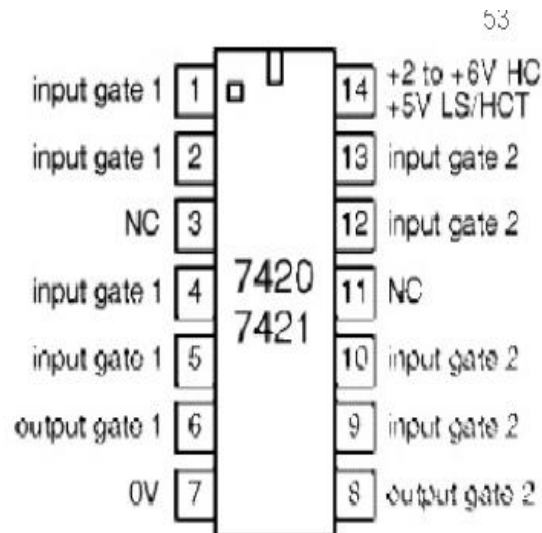
Experiment No.5

Aim: Realize a J-K Master/Slave FF using NAND gates and verify its truth table.

Apparatus: Bread board, wires, IC 74LS00, IC 74LS10, IC 74LS20.

Theory:

Pin Details of the ICs: 7400

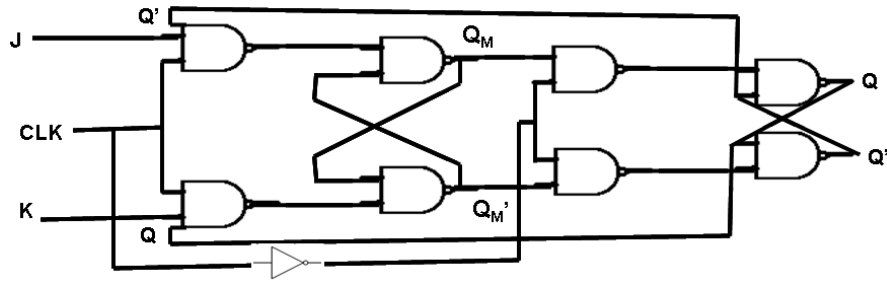


Master-Slave Flip Flop:-

The circuit of Master-Slave Flip Flop is basically two latches connected serially. The first latch is called the **Master** and the second is termed **Slave**. In a Master-Slave Flip Flop inputs are fed at the +ve edge and output is available at the -ve edge.

Let **Q_n** and **Q_{n+1}** represent the present state and next state of the flip flop, here is the truth table and circuit diagram of a Master-Slave Flip Flop:

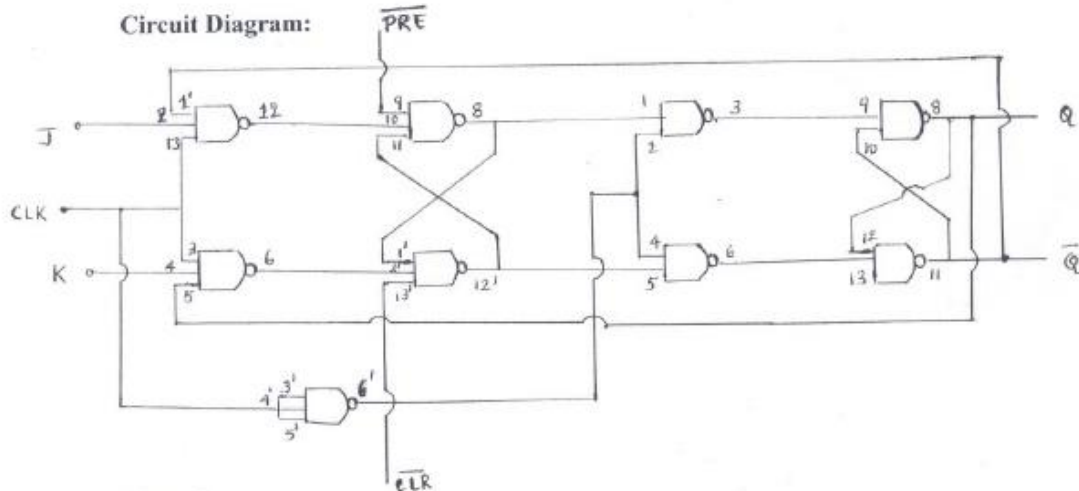
CLK	J	K	Q_{n+1}	Q_{n+1}'
1	0	0	Q_n	Q_n'
1	0	1	0	1
1	1	0	1	0
1	1	1	Q_n'	Q_n



Clock is connected directly to Master and inverted to Slave. When clock is high – Master is functional and when clock is low – Slave is functional. Q and Q' will be same as Q_M and Q_M'. Thus, what a master latch does when the clock goes high, the slave latch copies when the clock goes low.

For example, when J=1, K=1, let Q_n=0 and Q_n'=1. When clock is high then Q_{n+1}=Q_M=1, Q_M'=0. When clock is low then Q=Q_M=1, Q'=Q_M'=0. Thus, even when J=1, K=1, the outputs are complement to each other.

Clk	J	K	Q	--- Q	comment
	0	0	Q ₀	--- Q ₀	No change
	0	1	0	1	Reset
	1	0	1	0	Set
	1	1	Q ₀	Q ₀	toggle



Procedure:

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

Conclusion: - Hence we have studied the working of master slave J K flip flop

Experiment No.6

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

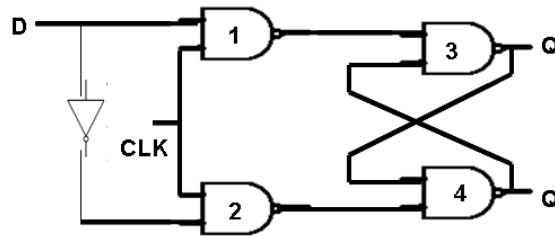
Theory:

D Flip Flop - Delay Flip Flop:-

D Flip Flop or a Delay Flip Flop has only data input D and two outputs which are complementary to each other and are denoted by Q and Q'. The Q output is identical to the D input except with one pulse time delay, hence the name D Flip Flop. D flip flop is used to avoid the **forbidden state** of S-R Latch. Since the inputs to the cross coupled NAND gates (gate 3 & 4) are always in opposite states, the invalid state (forbidden state) never occurs.

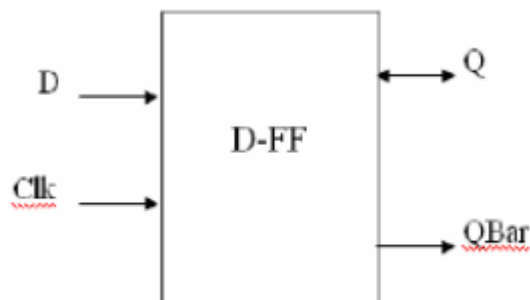
Let **Q_n** and **Q_{n+1}** denote the **present state** and **next state** of the flip flop, here is the truth table and circuit diagram of a D Flip Flop

D	Q _{n+1}
0	0
1	1



Application:

It is used in applications where delay is required.



--VHDL code for D Flip Flop Counter.

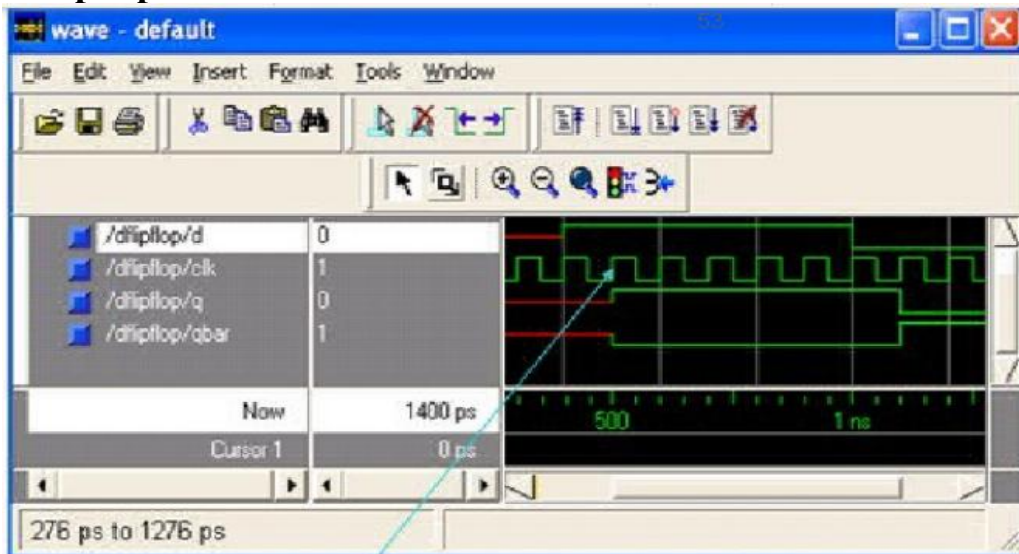
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity d_ff is
  Port ( D,Clk : in std_logic;
        Q : inout std_logic;
        Qbar : out std_logic);
end d_ff;

architecture behavioral of d_ff is
begin
  process(clk)
  begin
    if rising_edge(clk) then
      Q<= D;
    end if;
  end process;
  Qbar<= not Q;

end behavioral;
```

D-flipflop simulation result-



Positive edge triggered
(Output changes when clock signal goes low to high)

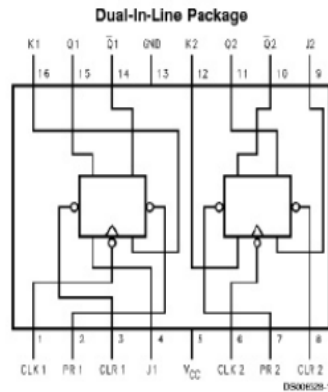
Conclusion:- Thus we have studied working of D Dlip Flip with its VHDL code.

Experiment No.7

Aim: Design and implement a mod-n ($n < 8$) synchronous up counter using JK Flip-Flop ICs and demonstrate its working.

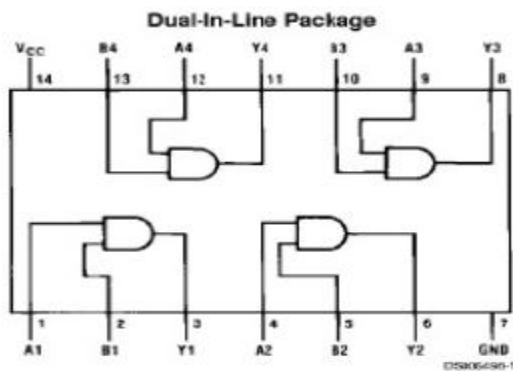
Apparatus: - Bread board, wires, IC 7486, IC 7408.

Pin diagram of 7476



Function Table

Inputs					Outputs	
PR	CLR	CLK	J	K	Q	\bar{Q}
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
H	H	\neg	L	L	Q_0	\bar{Q}_0
H	H	\neg	H	L	H	L
H	H	\neg	L	H	L	H
H	H	\neg	H	H	Toggle	



IC 7408

Theory:

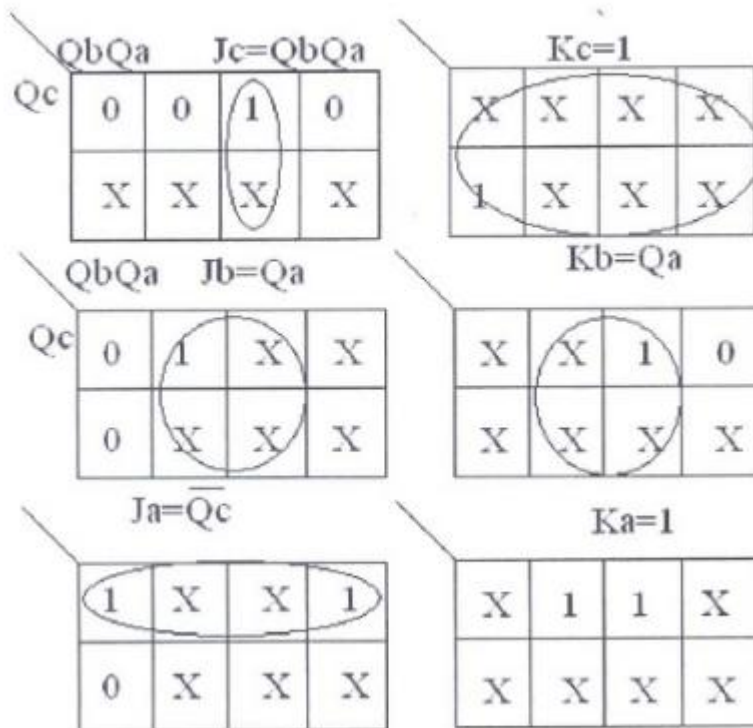
The ripple counter requires a finite amount of time for each flip flop to change state. This problem can be solved by using a synchronous parallel counter where every flip flop is triggered in synchronism with the clock, and all the output which are scheduled to change do so simultaneously.

The counter progresses counting upwards in a natural binary sequence from count 000 to count 100 advancing count with every negative clock transition and get back to 000 after this cycle.

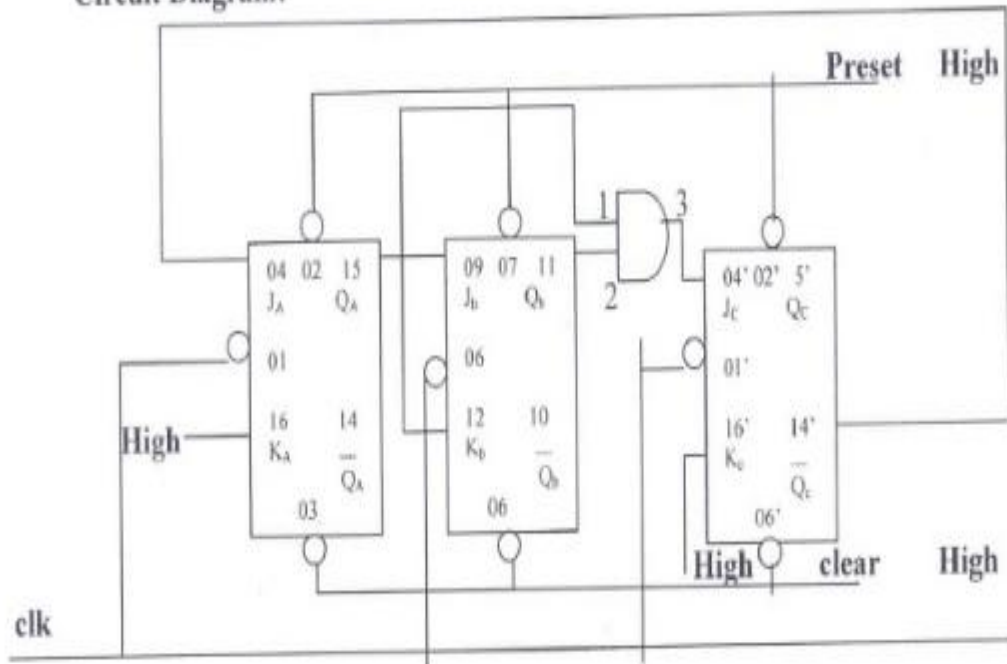
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Circuit Diagram:

K-Maps:



Circuit Diagram:



Procedure:

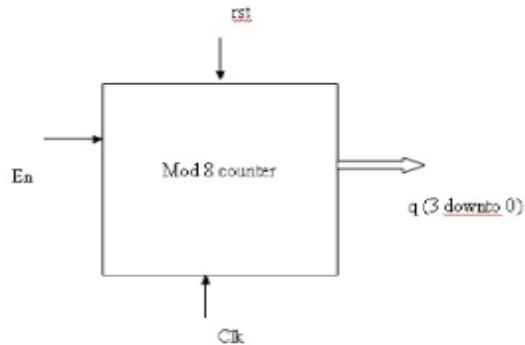
- (1) Verify all components and patch chords whether they are in good condition or not.
- (2) Make connection as shown in the circuit diagram.
- (3) Give supply to the trainer kit.
- (4) Provide input data to circuit via switches.
- (5) Verify truth table sequence and observe outputs.

Conclusion:- Thus we have studied the implementation of mod-n ($n < 8$) synchronous up counter using JK Flip-Flop.

Experiment No.8

Aim: Design and develop the VHDL code for mod-8 up counter. Simulate and verify its working.

Theory:-



Truth Table

<u>rst</u>	<u>Clk</u>	<u>En</u>	<u>Q</u>
1	X	0	0000
0	1	1	0001
0	1	1	0010
0	1	1	0011
0	1	1	0100
0	1	1	0101
0	1	1	0110
0	1	1	0111

05906490-1

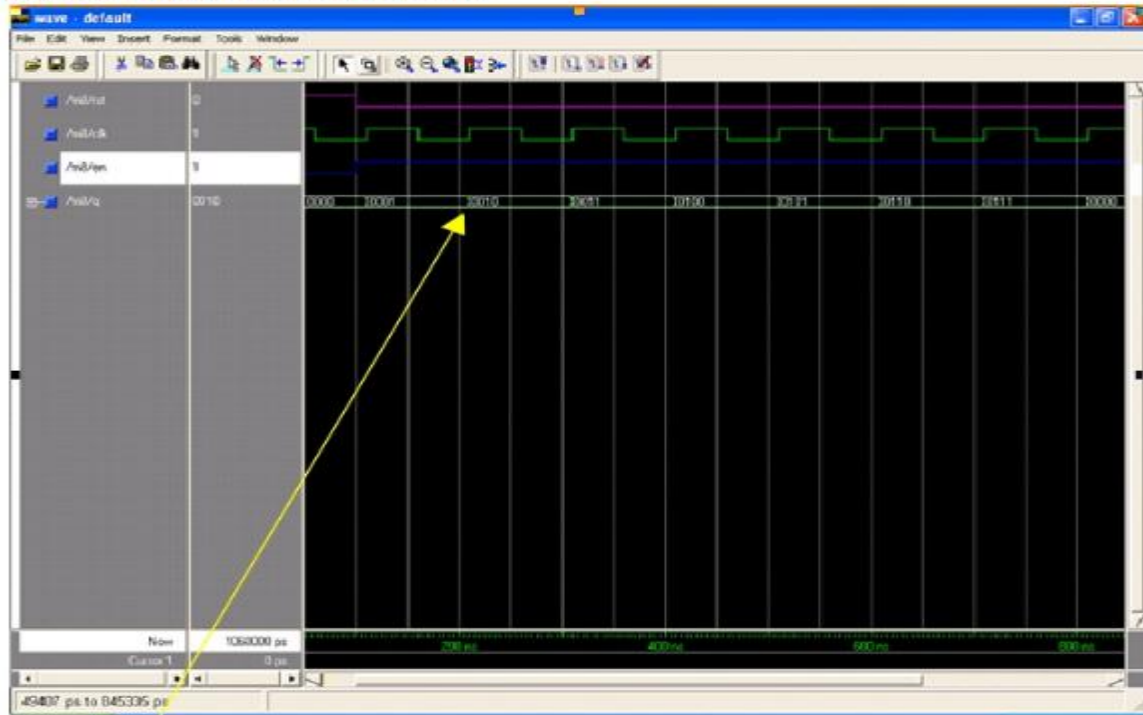
--VHDL code for Mod-8 Counter.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mod_8 is
    Port ( rst, clk, en: in std_logic;
          q : inout std_logic_vector(3 downto 0));
end mod_8;

architecture behavioral of mod_8 is
begin
    process(clk,rst) is
    begin
        if rst='1' then q<="0000";
        elsif rising_edge(clk) then
            if en='1' then
                Q<=Q+1;
            end if;
            if Q="0111" then
                Q<= "0000";
            end if;
        end if;
    end process;
end behavioral;
```

Mod-8 Counter Simulation Results



Conclusion:- Thus we have studied VHDL code for mod-8 up counter.

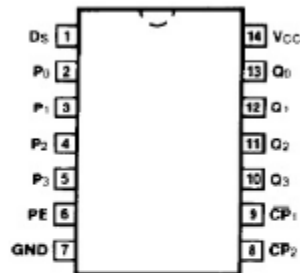
Experiment No.9

Aim: Design and implement a ring counter using 4-bit shift register and demonstrate its working.

Apparatus: - Bread board, wires, IC 7495.

Pin Diagram of ICs:

IC-7495



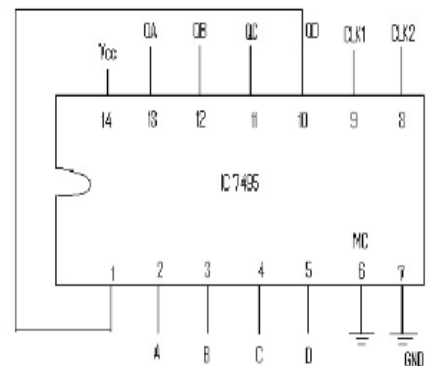
Theory:

Ring Counter is a basic register with direct feedback such that contents of the register simply circulate around the register when the clock is running. Here last output Q_d in a shift register is connected back to the serial input.

Function Table:

Clk	Qa	Qb	Qc	Qd
0	1	0	0	0
1	0	1	0	0
2	0	1	0	0
3	0	0	0	1

Circuit Diagram:



There are couples of ways to define a Register used in Digital Electronics.

“Registers are data storage devices that are more sophisticated than latches.”

“A register is a group of binary cells suitable for holding binary information.”

“A group of cascaded flip flops used to store related bits of information is known as a register.”

Application of Registers

These are used in computers for

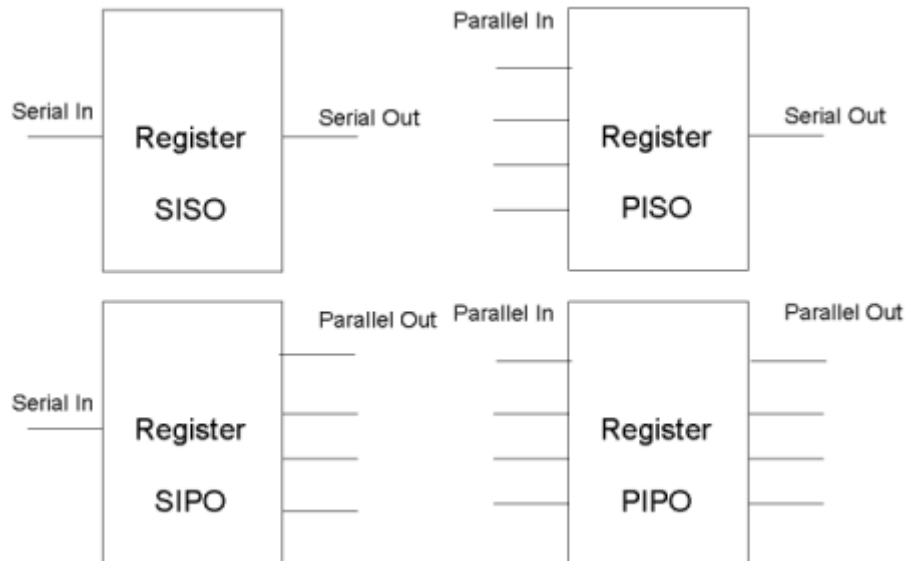
- Temporary storage
- Data transferring
- Data manipulation
- As counters

Shift Register

A register that is used to assemble and store information arriving from a serial source is called a shift register. Each flip flop output of a shift register is connected to the input of the following flip flop and a common clock pulse is applied to all flip flops, clocking them synchronously. Hence the shift register is a synchronous sequential circuit. An n-bit shift register consists of n Flip Flops and the gates control the shift operation.

There are four types of Shift Registers:

1. Serial-In, Serial-Out (SISO)
2. Parallel-In, Serial-Out (PISO)
3. Serial-In, Parallel-Out (SIPO)
4. Parallel-In, Parallel-Out (PIPO)



Procedure:

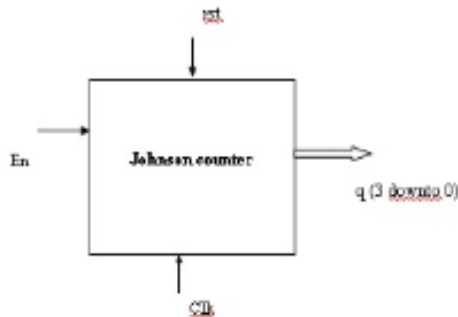
- 1) Verify all components and patch chords whether they are in good condition or not.
- 2) Make connection as shown in the circuit diagram.
- 3) Give supply to the trainer kit.
- 4) Provide input data to circuit via switched.
- 5) Verify truth table sequence and observe outputs.

Conclusion: - Hence we have studied implementation of ring counter using 4-bit shift register.

Experiment No.10

Aim: Design and develop the VHDL code for switched tail counter. Simulate and verify its working.

Theory:-



Truth Table

rst	clk	en	Q
1	0	0	0000
0	1	1	0001
0	1	1	0000
0	1	1	1000
0	1	1	1100
0	1	1	1110
0	1	1	1111
0	1	1	0111
0	1	1	0011

--VHDL code for Johnson counter.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

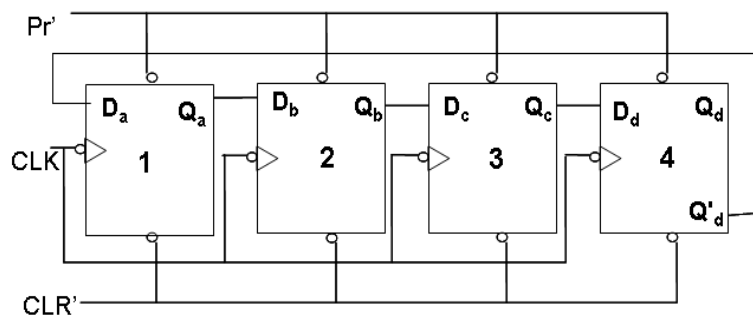
entity jc is
  Port ( clk, en, rst : in std_logic;
        q : inout std_logic_vector(3 downto 0));
end jc;

architecture behavioral of jc is
begin
  Process(clk,rst)
  begin
    if rst='1' then q<="0001";
    elsif rising_edge (clk) then
      if en='1' then
        q<=(not q(0)) & q(3 downto 1);
      end if;
    end if;
  end process;
end behavioral;
```

Johnson Counter:-

Johnson counter or Switch Tail or Twisted Ring Counter is a synchronous counter. An n-bit ring counter circulates a single bit among the flip flops to provide n distinct states. The number of states can be doubled if the shift register is connected as a switch tail ring counter.

A switch tail ring counter is a circular shift register with the complement output of the last flip flop connected to the input of the first flip flop. The circular connection is made from the complement output of the rightmost flip flop to the input of the leftmost flip flop. The register shifts its contents once to the right with every clock pulse and at the same time, the complement value of flip flop 4 are transferred to flip flop 1.



Starting from cleared states, the 4-bit switch tail ring counter goes through a sequence of 8 states. In general, a k-bit switch tail ring counter will go through a sequence of **2k states**.

Starting from all 0's, each shift operation inserts 1's from the left until the register is filled with all 1's. In the following sequence 0's are inserted from the left until the register is again filled with all 0's.

CLK	Qa	Qb	Qc	Qd	Qd'
0	0	0	0	0	1
1	1	0	0	0	
2	1	1	0	0	
3	1	1	1	0	
4	1	1	1	1	0
5	0	1	1	1	
6	0	0	1	1	
7	0	0	0	1	
8	0	0	0	0	1

Mod of Johnson Counter is $2n$, therefore it is known as Divide by $2N$ Counter.

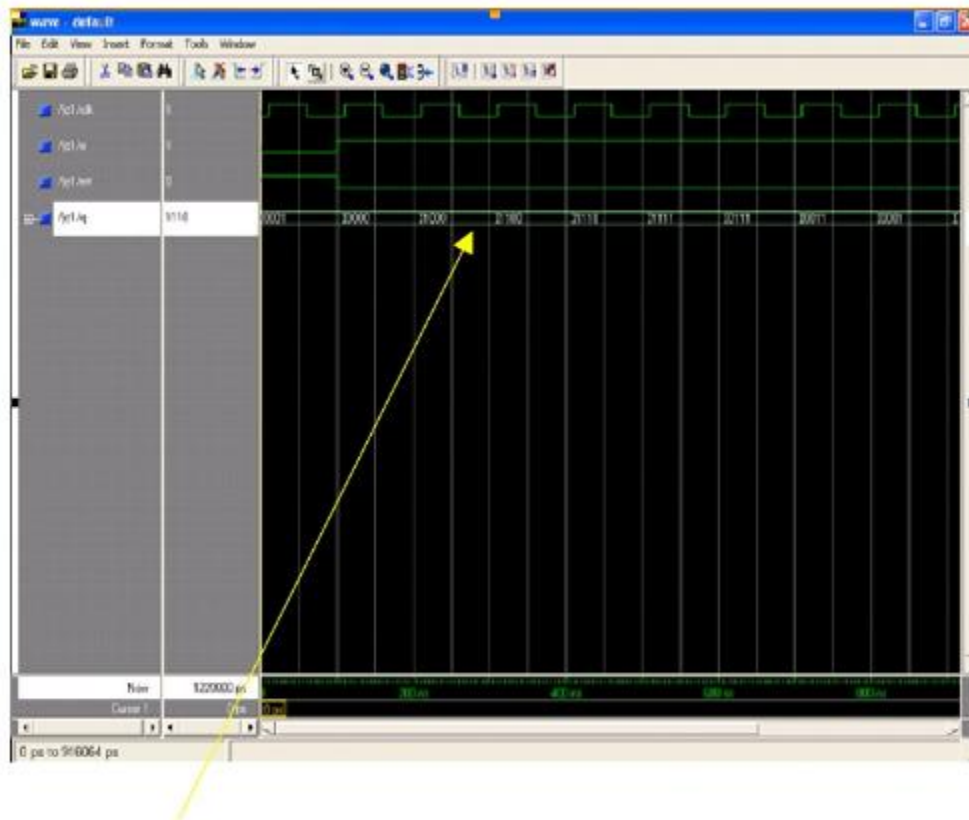
Frequency of Output = frequency of Clock Pulse / mod $f_{\text{Johnson}} = f / 2n$

Disadvantage of Johnson Counter

Disadvantage of Johnson Counter is that it doesn't count in a binary sequence.

Johnson counters can be constructed for any number of timing sequences. The number of flip flops needed is one half the numbers of timing signals.

Switched Tail Counter simulation result:-



Output

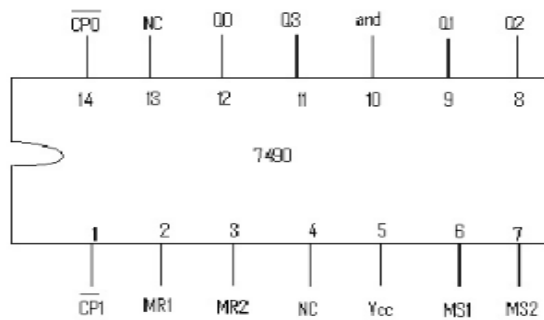
Conclusion: - Thus we studied VHDL code for switched tail counter.

Experiment No.11

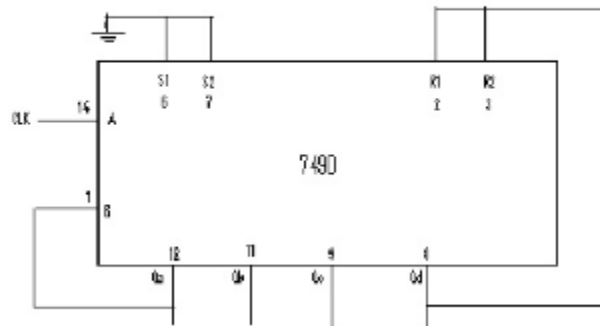
Aim: Design and implement a ring counter using 4-bit shift register and demonstrate its working. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ($n \leq 9$) and demonstrate its working.

Apparatus: - Bread board, wires, IC 7490.

Pin Diagram of 7490



Circuit Diagram:



Theory:

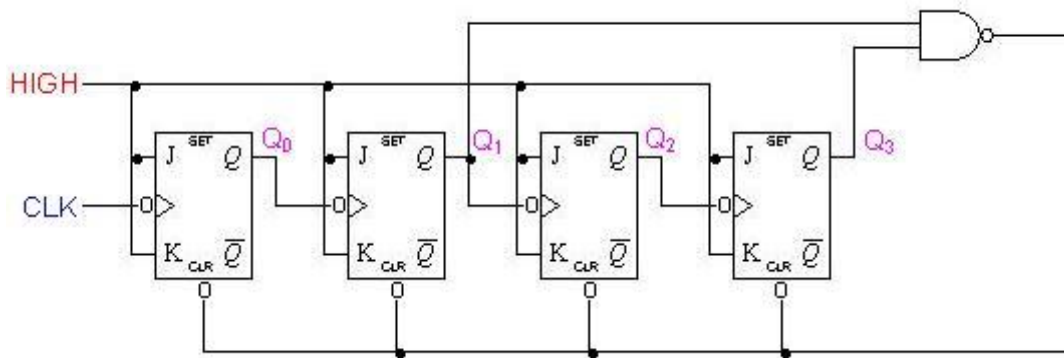
Asynchronous counter is a counter in which the clock signal is connected to the clock input of only first stage flip flop. The clock input of the second stage flip flop is triggered by the output of the first stage flip flop and so on. This introduces an inherent propagation delay time through a flip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

Function Table:

Clock	Qa	Qb	Qc
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
8	1	1	1

The binary counters previously introduced have two to the power n states. But counters with states less than this number are also possible. They are designed to have the number of states in their sequences, which are called truncated sequences. These sequences are achieved by forcing the counter to recycle before going through all of its normal states.

A common modulus for counters with truncated sequences is ten. A counter with ten states in its sequence is called a *decade counter*. The circuit below is an implementation of a decade counter.



Once the counter counts to ten (1010), all the flip-flops are being cleared. Notice that only Q1 and Q3 are used to decode the count of ten. This is called partial decoding, as none of the other states (zero to nine) have both Q1 and Q3 HIGH at the same time. The sequence of the decade counter is shown in the table below:

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

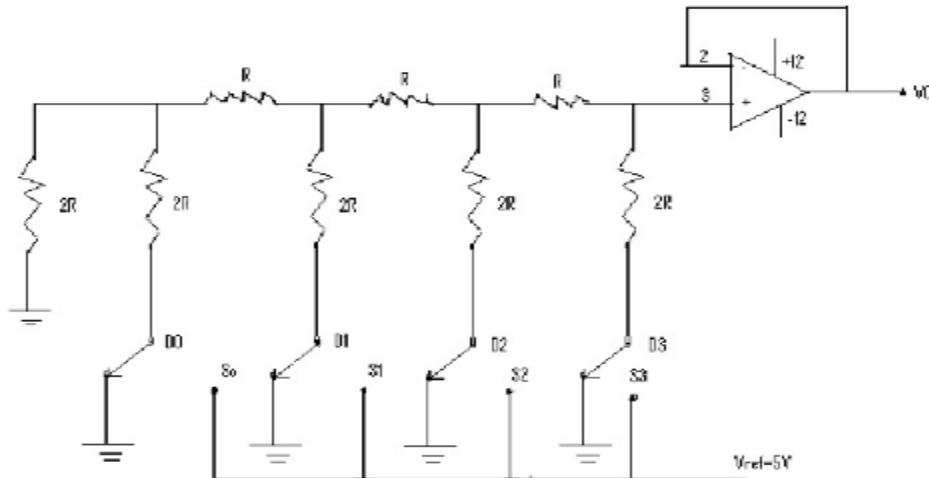
Conclusion: - Thus we studied implementation of ring counter using 4-bit shift register.

Experiment No.12

Aim: Design a 4 bit R-2R ladder D/A converter using op-amp. Determine its accuracy and resolution.

Apparatus: - Bread board, resistors, DMM, OP-741, connecting wires.

Circuit diagram:



$V_{ref} = 5V$

$$V_o = (2^3 D_3 + 2^2 D_2 + 2 D_1 + 2 D_0) V$$

$$V_o = (8D_3 + 4D_2 + 2D_1 + 2D_0) V_{ref} / 2^4$$

Smallest increment or change in the output voltage = $V_R \cdot 2^{-4} = (2R/3R)$

Tabular column:

Decimal equivalent	Digital inputs D3 D2 D1 D0	Analog output voltage	
		Theoretical value	Practical value
0	0 0 0 0		
1	0 0 0 1		
2			
3			
⋮			
15	1 1 1 1		

Procedure:

- 1) Make connections as shown in the circuit diagram
- 2) In different digital inputs measure the analog output voltage using multimeter.
- 3) Tabulate the results compare the theoretical output values with the practical ones.

Conclusion: - Thus we have studied a 4 bit R-2R ladder D/A converter using op-amp.