Laboratory Manual

CS 09 408 (P) Digital Systems Lab

INDEX

CYCLE I

A. Familiarization of digital ICs and digital IC trainer kit	
1	Verification of truth tables
B. Study of combinational circuits	
2.	Verification of De Morgan's theorem, the postulates of Boolean algebra and Realization of Sum of Product and Product of Sum expression using universal gates
C. Adders and Subtractor	
3	Half adder and Half Subtractor
4	Full adder and Full Subtractor
D. Digital comparator, Parity generator and checker, and Code converters	
5	Single bit digital comparator
6	Odd parity generator and checker
7	BCD to Excess-3 code converter
E. Flip flops using gates and familiarization of ICs	
8	RS, gated RS, D, T, and JK master slave flip-flops
F. Multiplexer, Demultiplexer, Shift registers and Counters	
9	Multiplexer and Demultiplexer using gates
10	Shift registers
11	Ring counter and twisted ring (Johnson) counter
12	Asynchronous UP and DOWN counter
13	Variable modulo asynchronous counter (Decade counter)

EXPT NO: 1

FAMILIARISATION OF DIGITAL ICs AND DIGITAL IC TRAINER KIT

AIM:

To familiarize with logic gate IC packages and to verify the truth tables of logic gates.Also familiarize with digital IC trainer kit

COMPONENTS REQUIRED

DIGITAL IC TRAINER KIT: The equipment mainly used to test and set up digital circuits. Integrated circuits can be fitted in sockets or bread board. There are built in voltage sources and clock signals. In order to feed monopulses manually, a debouncer switch is also provided. A number of select switches are provided to obtain '0' or '1' state voltages as digital inputs. Green and Red LEDs are provided to represent low and high states respectively to visualize the digital outputs.

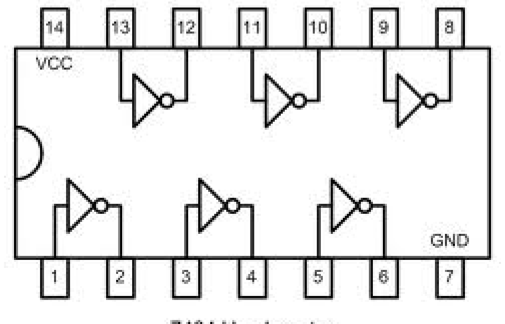
IC PACKAGES:	7404-Hex inverter gates
	7400-Quad two input NAND gates
	7402-Quad two input NOR gates
	7408-Quad two input AND gates
	7432-Quad two input OR gates
	7486-Quad two input XOR gates

PROCEDURE

1) Test the IC using digital IC tester before conducting the experiment
2) Verify the dual in line package pin out the IC before feeding the input
3) Set up the circuit and observe the output. Enter the input and output stages in truth table corresponding to the input combinations

PIN CONFIGURATION AND TRUTH TABLE

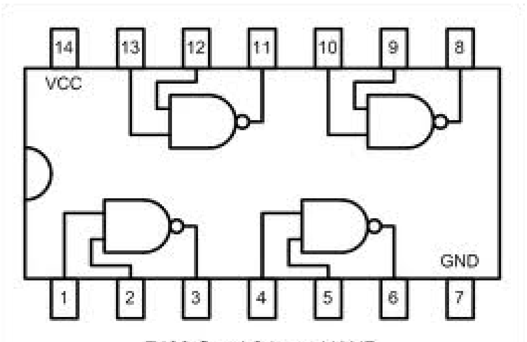
1. 7404-Hex inverter gates



7404 Hex Inverter

NOT	Input	Outputs					
	A	**Y_1**	**Y_2**	**Y_3**	**Y_4**	**Y_5**	**Y_6**
Pins	1,3,5,9,11,13	2	4	6	8	10	12
	0						
	1						

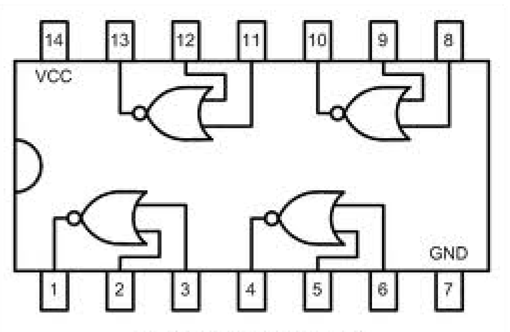
2. 7400-Quad two input NAND gates



7400 Quad 2 Input NAND

NAND	Inputs		Outputs			
	A	**B**	Y_1	Y_2	Y_3	Y_4
Pins	1,4,9,12	2,5,10,13	3	6	8	11
	0	0				
	0	1				
	1	0				
	1	1				

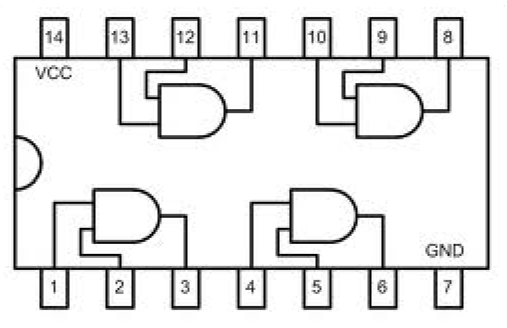
3. 7402-Quad two input NOR gates



7402 Quad 2 Input NOR

NOR	Inputs		Outputs			
	A	**B**	Y_1	Y_2	Y_3	Y_4
Pins	2,5,8,11	3,6,9,12	1	4	10	13
	0	0				
	0	1				
	1	0				
	1	1				

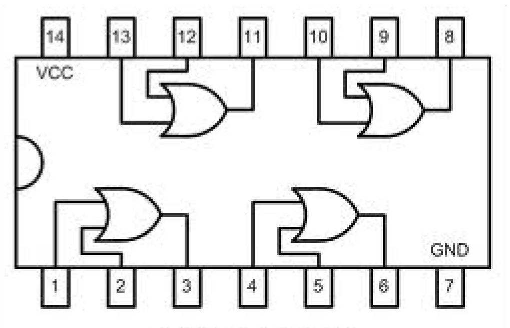
4. 7408-Quad two input AND gates



7408 Quad 2 Input AND

AND	Inputs		Outputs			
	A	**B**	Y_1	Y_2	Y_3	Y_4
Pins	1,4,9,12	2,5,10,13	3	6	8	11
	0	0				
	0	1				
	1	0				
	1	1				

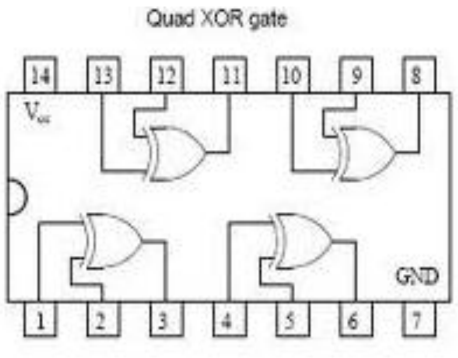
5. 7432-Quad two input OR gates



7432 Quad 2 Input OR

OR	Inputs		Outputs			
	A	**B**	Y_1	Y_2	Y_3	Y_4
Pins	1,4,9,12	2,5,10,13	3	6	8	11
	0	0				
	0	1				
	1	0				
	1	1				

6. 7486-Quad two input XOR gates



XOR	Inputs		Outputs			
	A	**B**	**Y_1**	**Y_2**	**Y_3**	**Y_4**
Pins	1,4,9,12	2,5,10,13	3	6	8	11
	0	0				
	0	1				
	1	0				
	1	1				

RESULT

Familiarized the digital IC trainer kit &logic gate IC packages and verified the truth tables of logic gates.

EXPT NO: 2

STUDY OF COMBINATIONAL CIRCUITS

AIM

1. To verify De-Morgan's theorem for two variables
2. Minimize the function $F = \bar{A}\bar{B} + A\bar{B} + \bar{A}B$ and verify
3. To realize sum of product (SOP) and product of sum (POS) expressions using universal gates

COMPONENTS REQUIRED: IC Trainer kit, IC 7400, IC 7402

PRINCIPLE

1. De Morgan theorem states that
   a) $\overline{(A + B + C + D + \cdots.)} = \bar{A}\bar{B}\bar{C}\bar{D}$ ....
   b) $\bar{A}\bar{B}\bar{C}\bar{D}$ .... $= \overline{(A + B + C + D + \cdots.)}$

De-Morgan's theorem is highly useful to simplify the Boolean expression

2. The given function can be written as $F = \bar{A}\bar{B} + A\bar{B} + \bar{A}B + \bar{A}\bar{B}$

$F = \bar{B}(\bar{A} + A) + \bar{A}(B + \bar{B}) \quad = \bar{B} + \bar{A}$

3. Gates NAND and NOR are known as universal gates, because any logic gates or Boolean expression can be realized by either NAND or NOR gate alone. Each product term in the SOP expression is called minterm and each sum term in the POS expression is called maxterm. SOP expression can be economically realized using NAND gates and POS expression can be economically realized using NOR gates

Realization of SOP expression

Using NAND gates

i) Use NAND gates for each minterm
ii) Use one NAND gate for whole summation

Using NOR gates

i) Invert all the variables in each minterm
ii) Use NOR gates for each minterm having inverted variables
iii) Use NOR gate for whole summation
iv) Use another NOR gate at the output for inverting.

Realization of POS expression

Using NOR gates

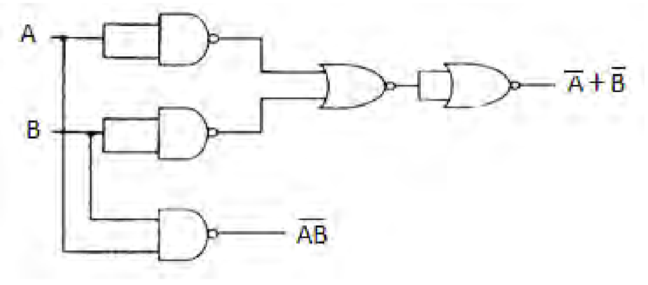
i) Use NOR gates for each maxterm
ii) Use one NOR gate for whole multiplication

Using NAND gates

i) Invert all the variables in each maxterm
ii) Use NAND gates for each maxterm having inverted variables
iii) Use NAND gate for whole multiplication
iv) Use another NAND gate at the output for inverting

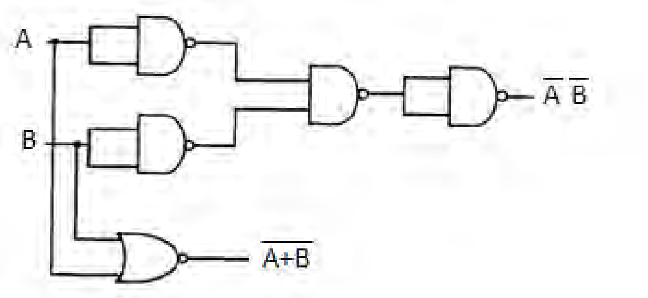
$1.\,a)\ \overline{AB} = \bar{A} + \bar{B}$ *by de morgans theorem*

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

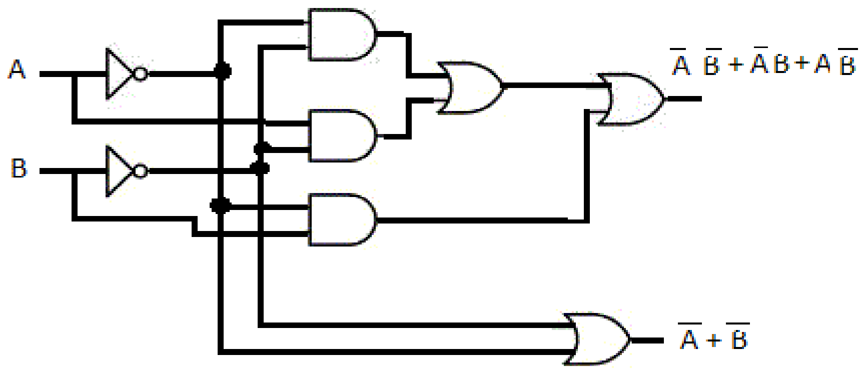


$b)\ \overline{A+B} = \bar{A}\bar{B}$ *by De Morgans theorem*

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



2. Minimization

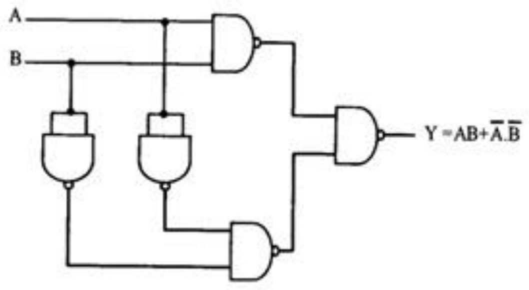


3. a)

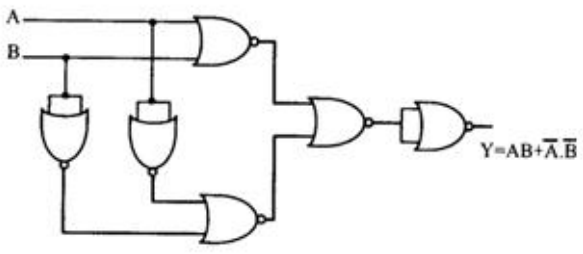
Realisation of SOP expression $Y = AB + \overline{A} \cdot \overline{B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Truth table



Using NAND gates only



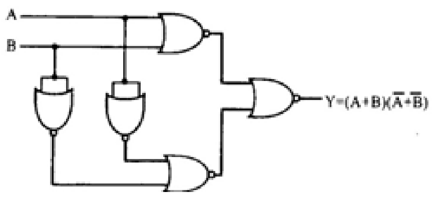
Using NOR gates only

b)

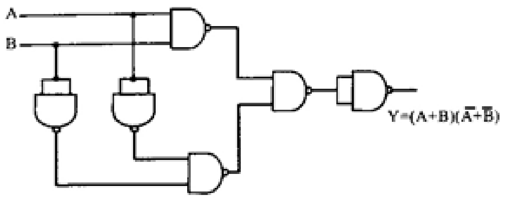
Realisation of POS expression $Y = (A + B)(\bar{A} + \overline{B})$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Truth table



Using NOR gates only



NAND gates only

PROCEDURE

1) Test all the components and IC packages using multimeter and digital IC tester
2) Set up the circuit one by one and verify their truth table
3) Observe the output corresponding to input combinations and enter it in truth table

RESULT AND DISCUSSION

De-Morgan's theorem and postulate of Boolean algebra were verified. Sum of products and product of sum expressions were realized using universal gates

<u>Model Questions</u>:

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. Realize the given logic function using not more than five 2-input NAND gates
   $F = \bar{A}\bar{B}\bar{C} + A\bar{B}C + \bar{A}\bar{B}D + \bar{B}C\bar{D}$
5. Convert the following expression into SOP&POS:-$(AB + C)(B + \bar{C}D)$ and realize using universal gates
6. Implement the function F (A, B, C) =∑m (0, 1, 4).
7. Design a circuit with four inputs and one output, such that the output goes to '1'whenever two or more of inputs are '1'. For other cases the output remains at '0'.

EXPT NO: 3

HALF ADDER AND HALFSUBTRACTOR

AIM

1. To design and set up half adder and half subtractor using
   a. EXOR gates and AND gates
   b. NAND gates

COMPONENTS REQUIRED

IC Trainer kit, IC 7400, IC 7486

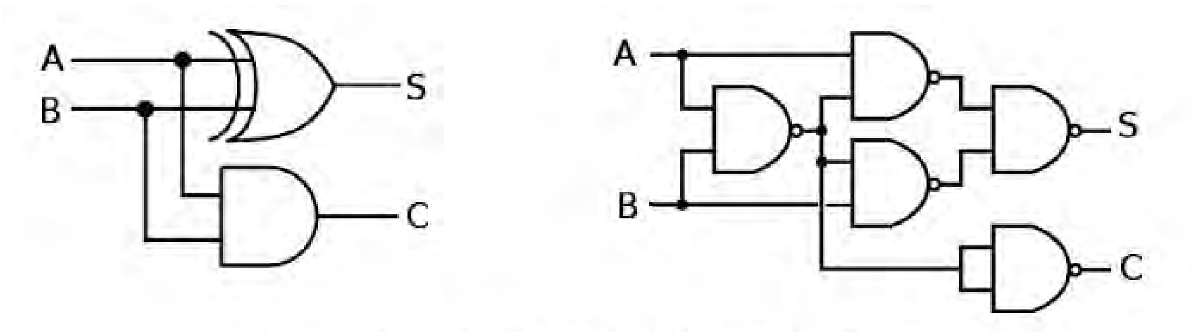
PRINCIPLE

The simplest binary adder is called half adder. Half adder has two input bits and two output bits. One output bit is the sum and the other is the carry. They are represented by‘S’ and ‘C’ respectively in logic symbol.

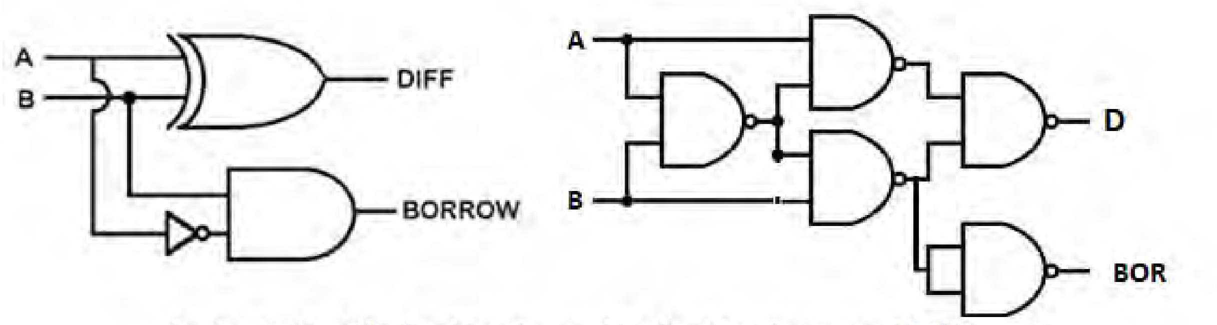
The simplest binary subtractor is called half Subtractor. It has two input bits and two output bits. One output bit is the Difference and the other is borrowed. They are represented by ‘D’ and ‘B’ respectively in logic symbol

Truth table of Half Adder			
Inputs		Output	
A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table of Half Subtractor			
Inputs		Output	
A	**B**	**D**	**BOR**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Logic circuit of Half adder and Haff adder using NAND gate only



Logic circuit of half subtractor and half subtractor using NAND gates

PROCEDURE

1. Verify whether all the wires and components are in good condition
2. Set up a half adder circuit and feed all the input combinations
3. Observe the output corresponding to input combinations and enter it in the Truth table
4. Repeat the above steps for half subtractor circuits

RESULT AND DISCUSSION

Half adder and the half subtractor circuits are set up using logic gates and verified the result.

<u>Model Questions</u>:

1. Design half adder cum Subtractor using mode control.

EXPT NO: 4

FULL ADDER AND FULL SUBTRACTOR

AIM

1. To design and set up full adder and full subtractor using
    a. EXOR gates and AND gates
    b. NAND gates

COMPONENTS REQUIRED

IC Trainer kit,ICS 7400, IC 7486

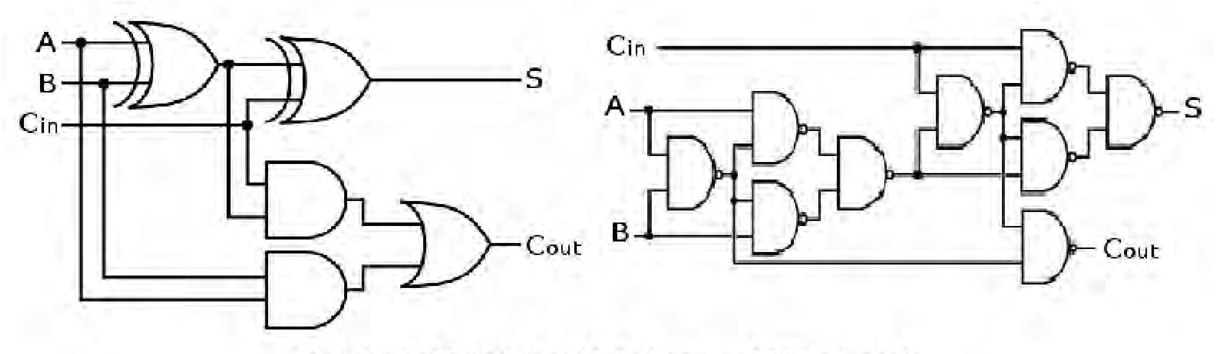
PRINCIPLE

A half adder has no provision to add a carry from the lower order bits when binary numbers are added. When two input bits and a carry are to be added the number of input bits becomes three and the input combination increases to eight. For this full adder is used. Like half adder it also has a sum bit and a carry bit. The new carry generated is represented by ‘C_{out}’ and the carry generated from the previous addition is represented by ‘C_{in}’

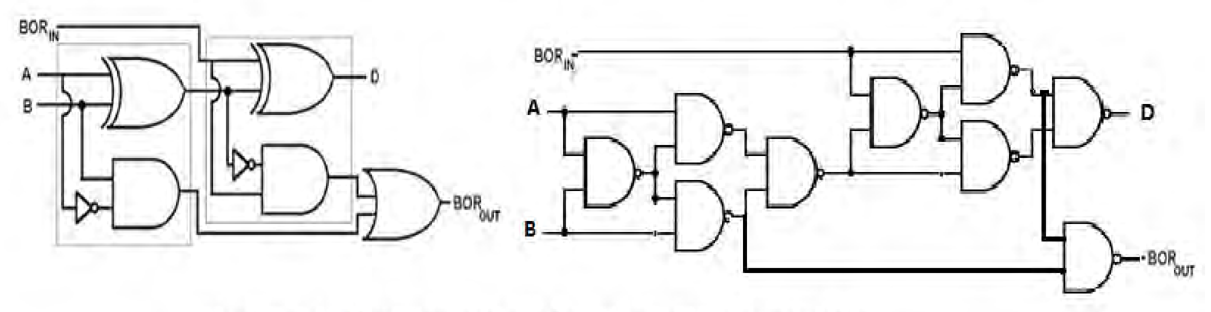
When two input bits and a borrow have to be subtracted the number of input bits equal to three and the input combinations increases to eight, for this a full subtractor is used.

Truth table of Full Adder				
Inputs			Output	
A	**B**	**C_{in}**	**S**	**C_{out}**
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

Truth table of Full Subtractor				
Inputs			Output	
A	**B**	**BOR_{in}**	**D**	**BOR_{out}**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Logic circuit of Full adder and Full adder using NAND gates only



Logic circuit of full subtractor and full subtractor using NAND gates

PROCEDURE

1. Verify whether all the wires and components are in good condition
2. Set up full adder circuit and feed all the input combinations
3. Observe the output corresponding to input combinations and enter it in the Truth table
4. Repeat the above steps for Full subtractor circuits

RESULT AND DISCUSSION

Fulladder and the Full subtractor circuits are set up using logic gates and verified the result.

<u>Model Questions</u>:

1. Difference between half adder and a full adder
2. Design full adder cum Subtractor using mode control
3. Design a full adder using two half adder and suitable gate

EXPT NO: 5

DIGITAL COMPARATOR

AIM

To design and setup single bit comparator using logic gates and verify the truth table.

PRINCIPLE

Digital or Binary Comparators compares the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

The digital comparator accomplishes this using several logic gates that operate on the principles of Boolean algebra. The purpose of a **Digital Comparator** is to compare a set of variables or unknown numbers, for example A (A1, A2, A3, ... An, etc) against that of a constant or unknown value such as B (B1, B2, B3, ... Bn, etc) and produce an output condition or flag depending upon the result of the comparison. There are two main types of Digital Comparator available and these are.

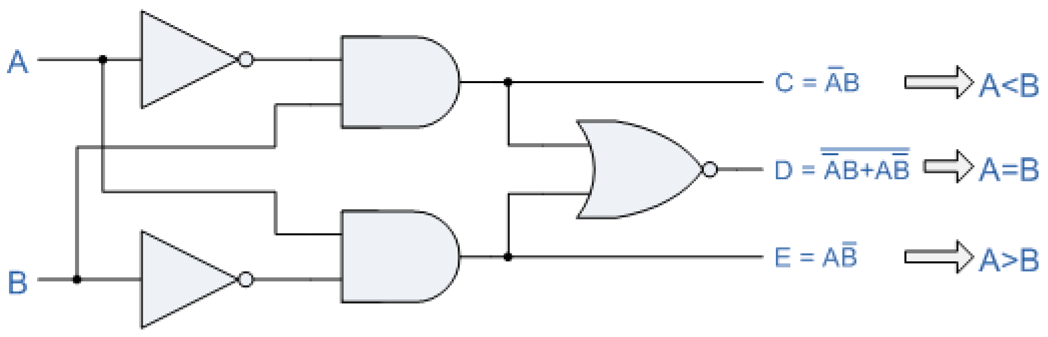
Identity Comparator - an *Identity Comparator* is a digital comparator that has only one output terminal for when the inputs A = B

Magnitude Comparator - a *Magnitude Comparator* is a type of digital comparator that has three output terminals, one each for equality (A = B), greater than (A > B) and less than (A < B)

This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the truth table of simple 1-bit comparator.

Input		Output		
B	**A**	**A>B**	**A=B**	**A<B**
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Digital comparators actually use Exclusive-NOR gates within their design for comparing their respective pairs of bits. When we are comparing two binary or BCD values or variables against each other, we are comparing the "magnitude" of these values, logic "0" against a logic "1" which is where the term Magnitude Comparator comes from.



RESULT

Designed and setup single bit comparator using logic gates and verified the truth table.

Model Questions:

1. Design a two bit digital comparator
2. Which logic is used in single bit digital comparator

EXPT NO: 6

PARITY GENERATOR AND CHECKER

AIM

To design and verify the truth table of a three bit odd parity generator and checker

COMPONENTS REQUIRED

IC Trainer kit, ICS 7400, IC 7486

PRINCIPLE

A parity bit is used for the purpose of detecting errors during transmission of binary information. A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors. An error is detected if the checked parity does not correspond with the one transmitted. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker.

In even parity the added parity bit will make the total number of 1's an even amount and in odd parity the added parity bit will make the total number of 1's an odd amount.

In a three bit odd parity generator the three bits in the message together with the parity bit are transmitted to their destination, where they are applied to the parity checker circuit. The parity checker circuit checks for possible errors in the transmission.

Since the information was transmitted with odd parity the four bits received must have an odd number of 1's. An error occurs during the transmission if the four bits received have an even number of 1's, indicating that one bit has changed during transmission. The output of the parity checker is denoted by PEC (parity error check) and it will be equal to 1 if an error occurs, i.e., if the four bits received has an even number of 1's.

<u>ODD PARITY GENERATOR</u>

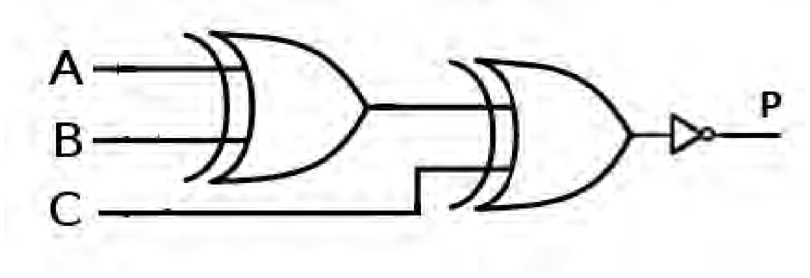
INPUT (Three bit message)			**OUTPUT** (Odd Parity bit)
A	**B**	**C**	**P**
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

From the truth table the expression for the output parity bit is,
P (A, B, C) = Σ m (0, 3, 5, 6)   Also written as,

$$P = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$

$$P = (\overline{A \oplus B \oplus C})$$

CIRCUIT DIAGRAM: ODD PARITY GENERATOR



ODD PARITY CHECKER

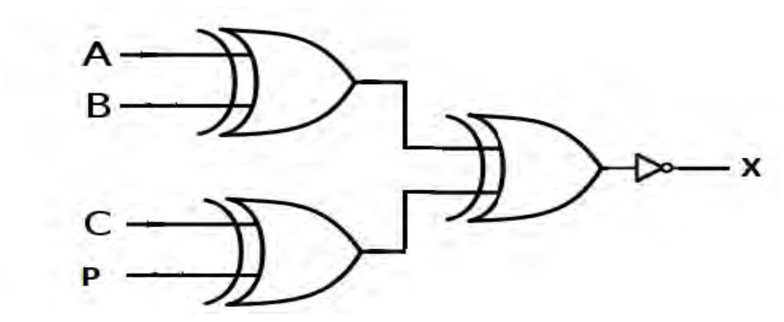
INPUT (four bit messageReceived)				OUTPUT (Parity error check)
A	**B**	**C**	**P**	**X**
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

From the truth table the expression for the output parity checker bit is,
X (A, B, C, P) = Σ (0, 3, 5, 6, 9, 10, 12, 15)

The above expression is reduced as,

$$X = (\overline{A \oplus B \oplus C \oplus P})$$

CIRCUIT DIAGRAM: ODD PARITY CHECKER



PROCEDURE:

1. Connections are given as per the circuit diagrams.
2. for all the ICs 7th pin is grounded and 14th pin is given +5 V supply.
3. Apply the inputs and verify the truth table for the Parity generator and checker.

RESULT:

The design of the three bit odd Parity generator and checker circuits was done and their truth tables were verified.

<u>Model Questions</u>:

1. Design 2-bit odd parity generator and checker
2. Need of parity generator and checker
3. Difference between odd parity and even parity

EXPT NO: 7

BCD to EXCESS -3 CODE CONVERTER

AIM

To design and set up the circuit of BCD to Excess-3 converter

COMPONENTS REQUIRED

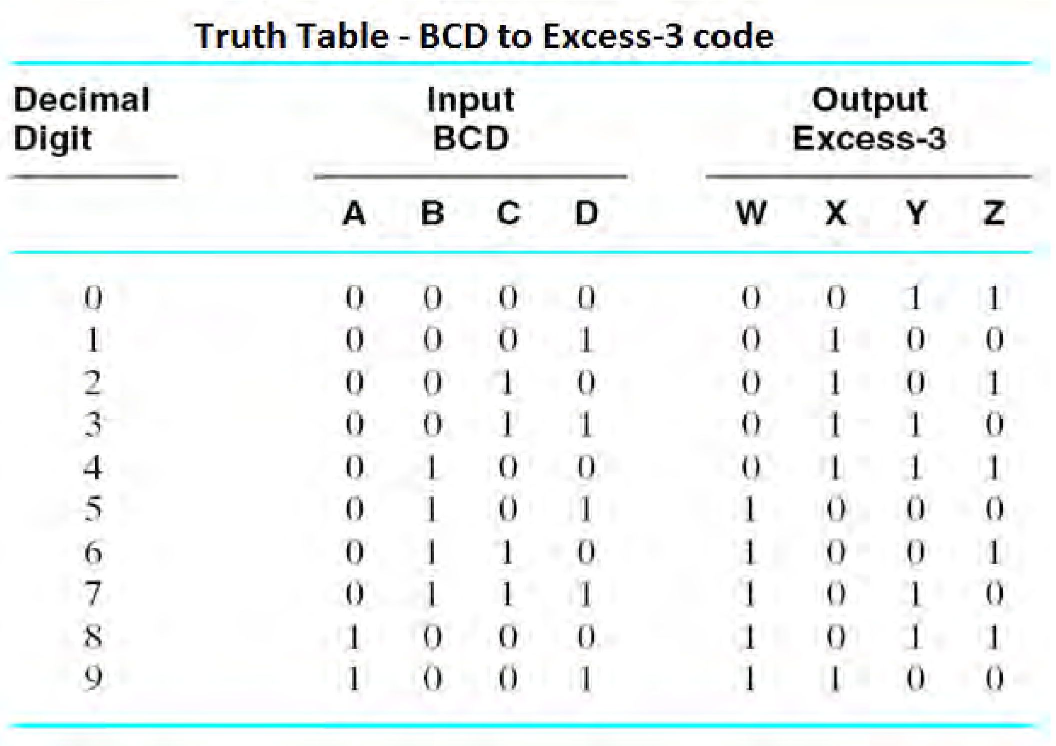
IC Trainer kit,IC 7486, IC 7408, IC 7404, IC 7432

PRINCIPLE

The Excess-3code for a decimal digit is the binary combination corresponding to the decimal digit plus 3. For example, the excess-3 code for decimal digit 5 is the binary combination for 5+3=8, which is 1000.Each BCD digit four bits with the bits with the bits, from most significant to least significant, labeled A,B,C,D. As the same for excess-3 labeled W,X,Y,Z.

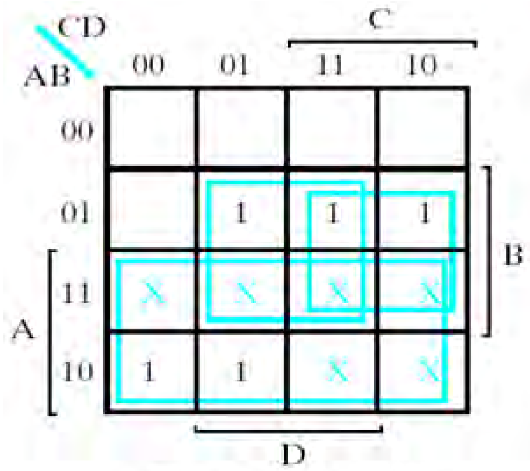
PROCEDURE

1. Test all the components using multi meter and digital IC tester
2. Verify the truth table of the circuit by feeding input bit combinations

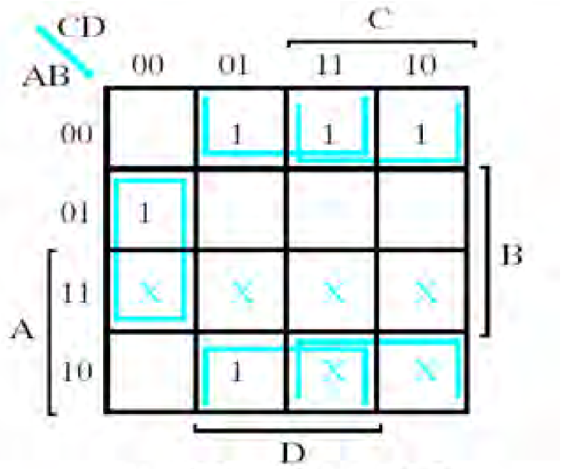


Truth Table - BCD to Excess-3 code

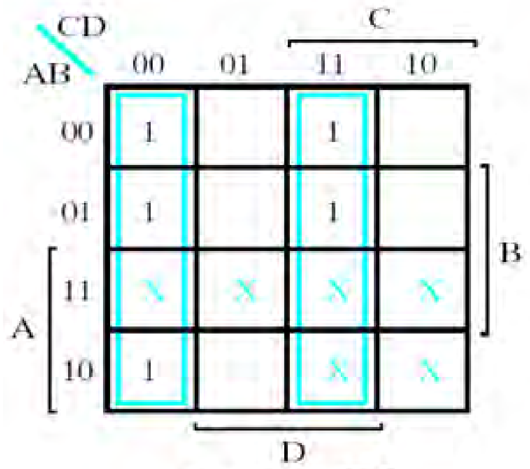
Decimal Digit	Input BCD				Output Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0



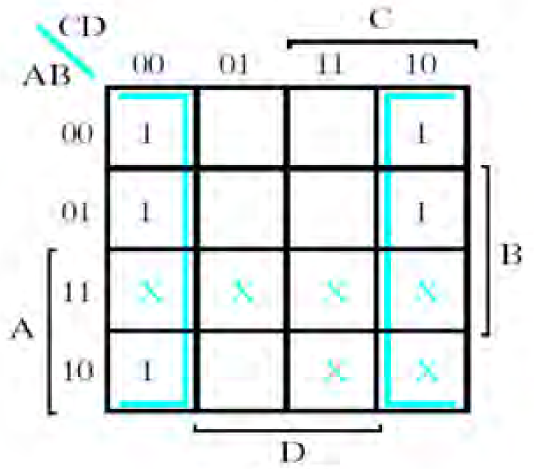
$W = A + BC + BD$



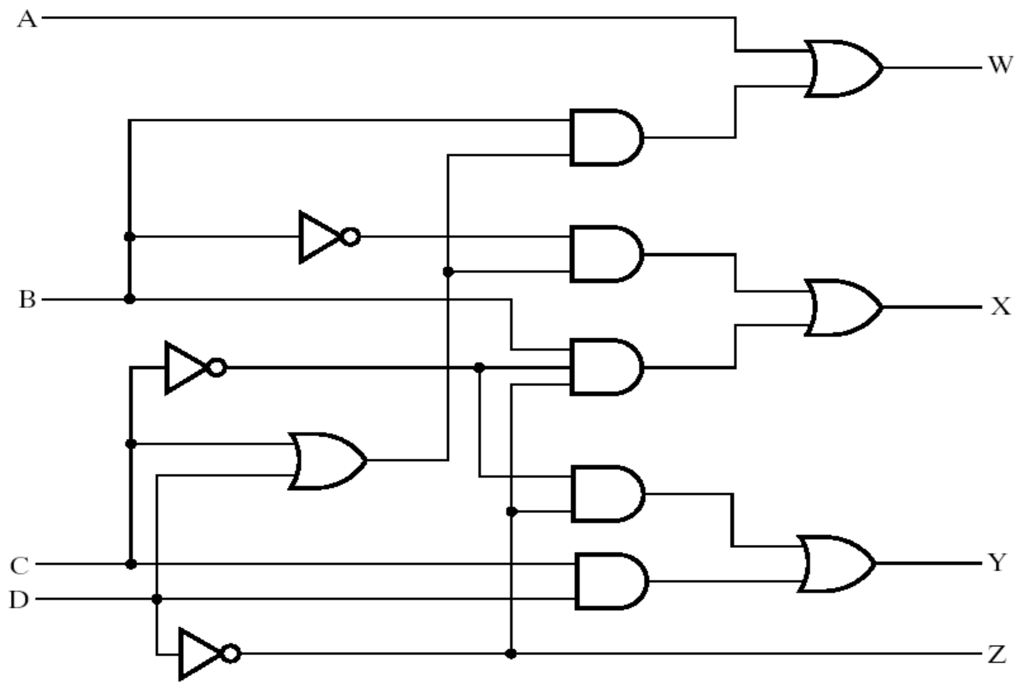
$X = \overline{B}C + \overline{B}D + B\overline{C}\overline{D}$



$Y = CD + \overline{C}\overline{D}$



$Z = \overline{D}$



RESULT

The circuit of BCD to Excess-3 converter has set up and verified the result

Model Questions:

1. Significance of excess-3 code
2. Significance of Gray code
3. Design 3-bit binary to gray code converter
4. Design 5-bit gray code to binary

EXPT NO: 8

FLIP FLOPS USING GATES AND FAMILIARIZATION OF ICs

AIM

To Setup the following flip flops using gates and verify the truth table also familiarize the flip flop ICs

1. $\bar{R}\bar{S}$ flip flop
2. GatedRS flip flop
3. D flip flop
4. T flip flop
5. JK flip flop
6. MS JK flip flop

COMPONENTS REQUIRED

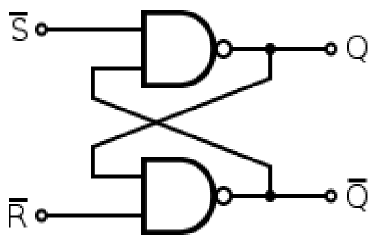
Digital IC trainer kit, IC 7400, IC 7410, IC 7473, IC 7474, IC 7476

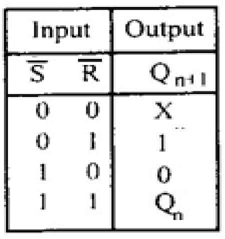
PRINCIPLE

Flip flops are the basic building blocks in any memory systems since its output will remain in its state until it is forced to change it by some means

$\bar{S}\bar{R}$ FLIP FLOP

$\bar{S}$ and $\bar{R}$stands for set and reset. There are four input combination possible at the inputs. But $\bar{S}$=$\bar{R}$=1 is forbidden since the output will be indeterminate



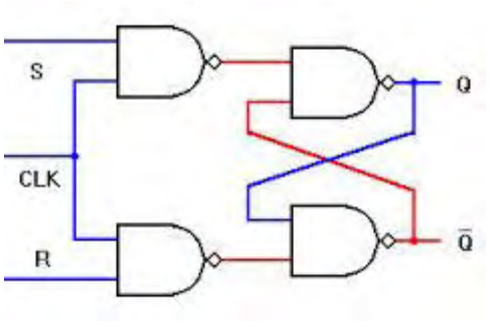


Input		Output
$\bar{S}$	$\bar{R}$	Q_{n+1}
0	0	X
0	1	1
1	0	0
1	1	Q_n

X: Forbidden state

GATED SR FLIP FLOP

S and R stands for Set and Reset. There are four input combination possible at the inputs. But S=R=1 is forbidden since the output will be indeterminate.

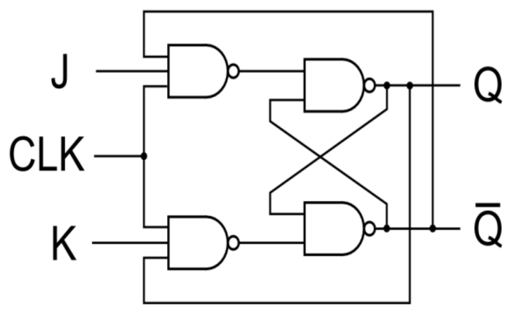


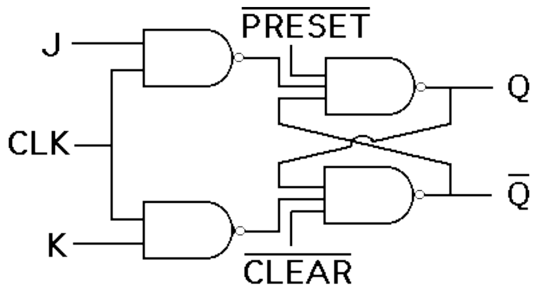
Input		Output
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	X

X: Forbidden state

J K FLIP FLOP

The indeterminate output state of SR FF when S=R=1 is avoided by converting it to a JK FF.When flip flop is switched on its output state is uncertain. When an initial state is to be assigned two separate inputs called preset and clear are used. They are active low inputs

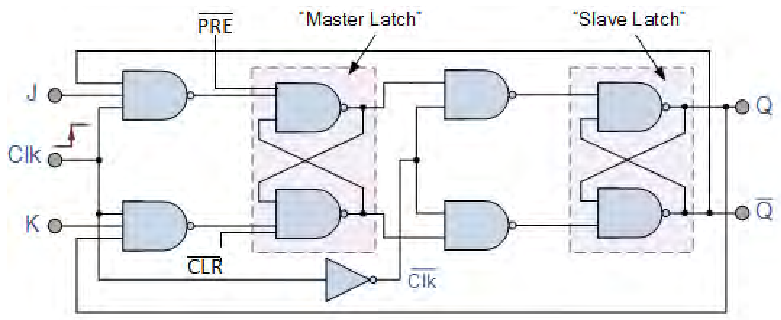


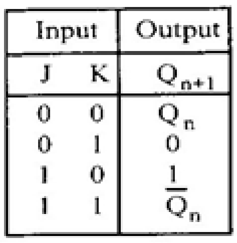


Input		Output
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q}_n$

MASTER SLAVE J K FLIP FLOP

The race around condition of JK FF is rectified in master slave JK FF. Racing is toggling of output more than ones during the positive clock edge. MSJK FF is created by cascading two JK FF. The clock is fed to the first stage (Master) and is inverted and fed to the second stage (slave). This ensure that the master is always followed by the slave and eliminate the chance of racing

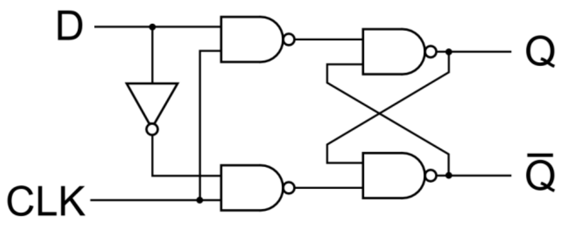


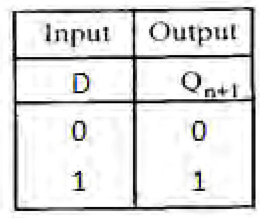


Input		Output
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q}_n$

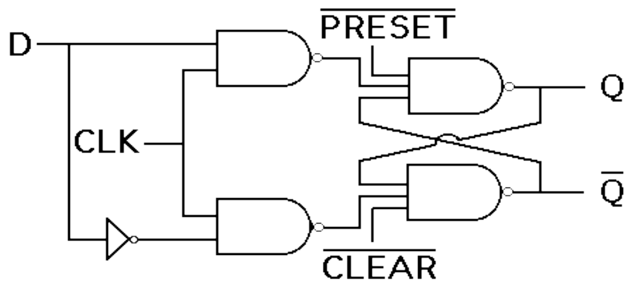
D FF

It has only one input called as D input or Data input. The input data is transferred to the output after a clock is applied. D FF can be derived from JK FF by using J input as D input and J is inverted and fed to K input



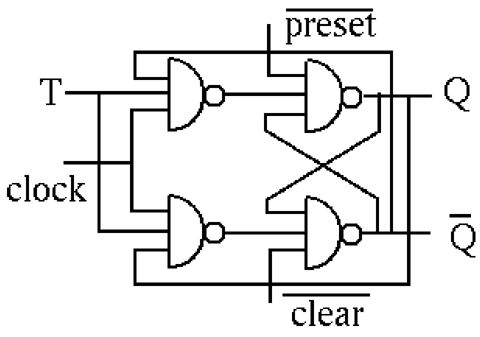


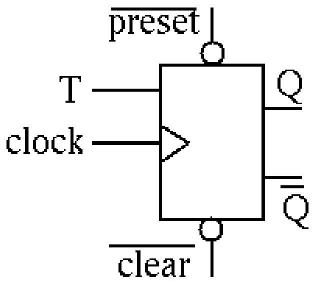
Input	Output
D	Q_{n+1}
0	0
1	1

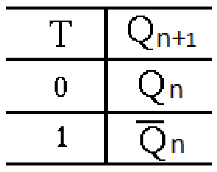


T FF

T stands for Toggle. The output toggles when a clock pulse is applied. T FF can be derived from JK FF by shorting J and K input

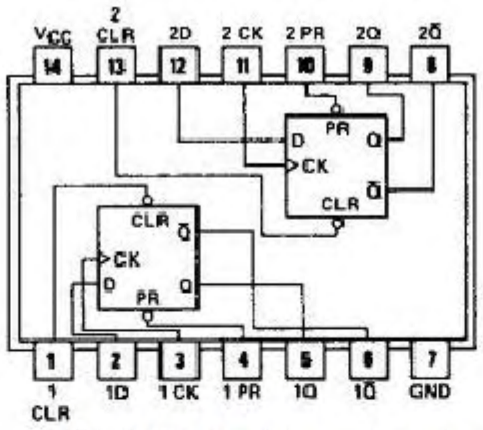




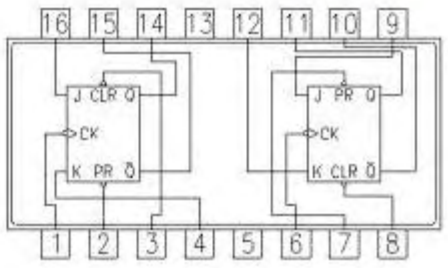


T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

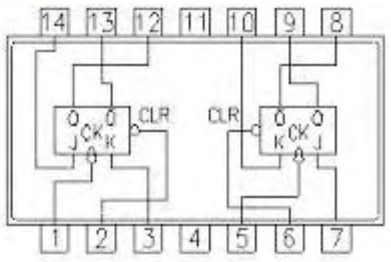
FLIP FLOP ICs



7474 **D Flip-flop** with preset and clear



7476
Dual J/K M/S Flip-Flop
with Preset and Clear



7473
Dual J-K M/S Flip-Flop with Clear

PROCEDURE

1. Test all components and IC packages using digital IC tester and multimeter
2. Set up FF using Gates and verify their truth tables
3. Verify the Truth tables of 7473, 7474, and 7476 ICs

RESULT AND DISCUSSION

The flip flops $\bar{R}\bar{S}$ flip flop, Gated RS flip flop, D flip flop, T flip flop, JK flip flop and MS JK flip flops were set up using NAND gates and verified. ICs 7474, 7473 and 7476 are familiarized

Model Questions

1. List four Basic Flip-flop applications**?**
2. What advantage does a J-K Flip-flop have over an S-R**?**
3. What is meant by Race around condition**?**

EXPT NO: 9

MULTIPLEXER AND DEMULTIPLEXER USING GATES

AIM

To design Multiplexer and Demultiplexer and verify their truth tables

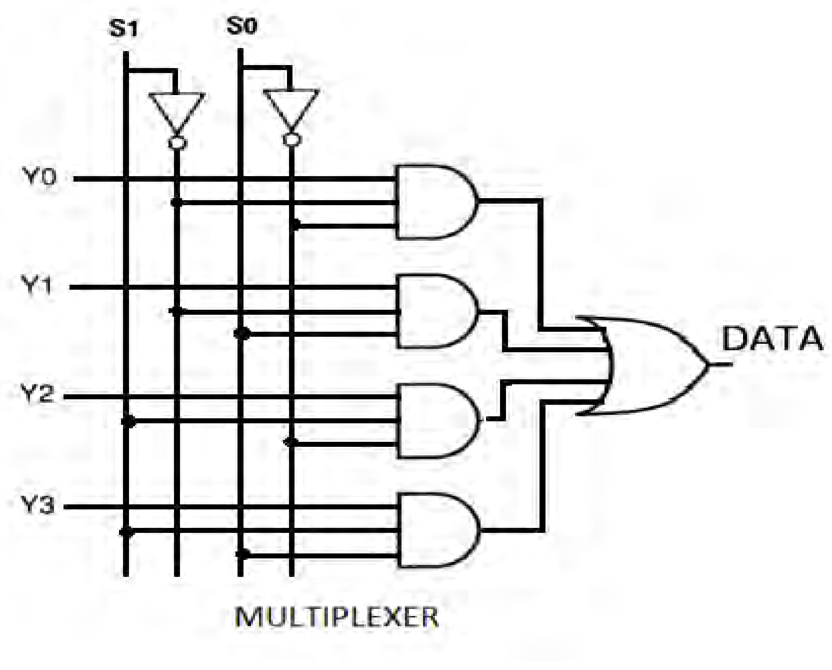
COMPONENTS REQUIRED: -

Digital IC trainer kit, IC

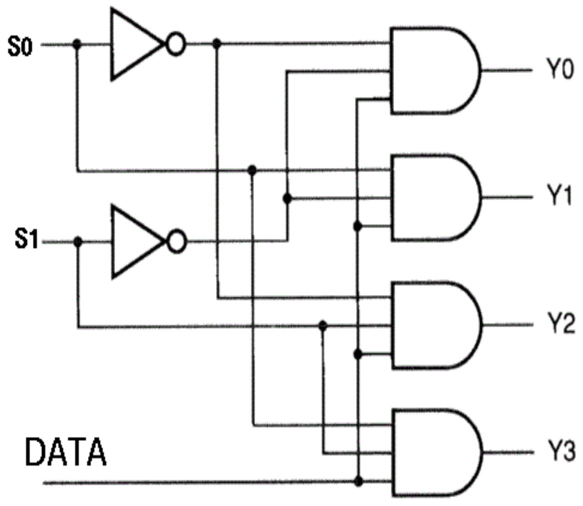
PRINCIPLE

In electronics, a **multiplexer** (or **mux**) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has *n* select lines, which are used to select which input line to send to the output. An electronic multiplexer can be considered as a multiple-input, single-output switch. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a **data selector**.

Conversely, a **demultiplexer** (or **demux**) is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. An electronic demultiplexer can be considered as a single-input, multiple-output switch. A multiplexer is often used with a complementary demultiplexer on the receiving end.



TRUTH TABLE		
S1	**S0**	**DATA**
0	0	Y0
0	1	Y1
1	0	Y2
1	1	Y3



DEMULTIPLEXER

TRUTH TABLE						
INPUTS			OUTPUTS			
DATA	S1	S0	Y3	Y2	Y1	Y0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

PROCEDURE:

1. Connections are made as per the circuit diagram
2. Switch on the power supply
3. Apply different combinations of inputs and observe the outputs; compare the outputs with the truth tables.

RESULT:

Multiplexer and Demultiplexer are constructed and verified the truth tables.

<u>Model Questions</u>:

1. Difference between Decoder and Demultiplexer
2. Difference between encoder and multiplexer
3. Implement the function F (A, B, C) =∑m (0, 1, 3, 4) using multiplexer.
4. Design an 8:1 multiplexer
5. Design an 1 to 16 demultiplexer
6. Design a full adder using multiplexer

EXPT NO: 10

SHIFT REGISTERS

AIM

To set up and verify the performance of shift registers using D FF

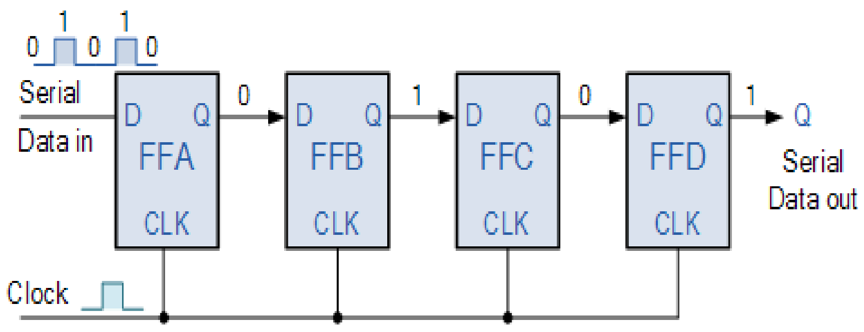
COMPONENTS REQUIRED: -

Digital IC trainer kit, IC 7476, IC 7408

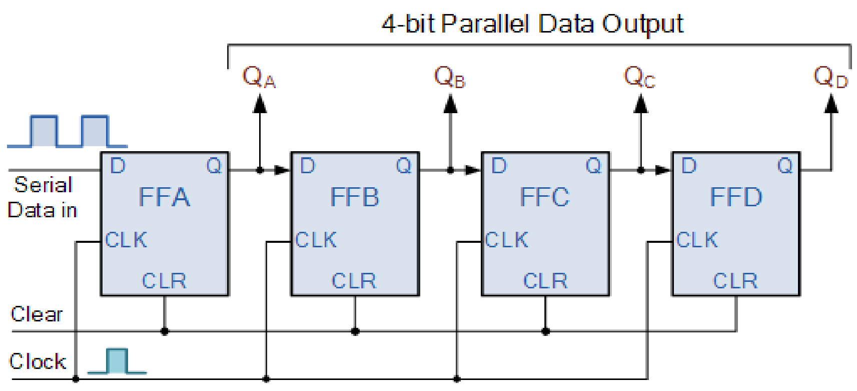
PRINCIPLE

Registers are simply a group of flip flops that can be used to store a binary number. A shift register is nothing but a register which can accept binary number and shift it. The data can be entered in the shift register either in serial or in parallel. The output can be taken either in serial or in parallel. Since there are two ways to shift data in to a register and two ways to shift data out of the register four types of registers can be constructed.

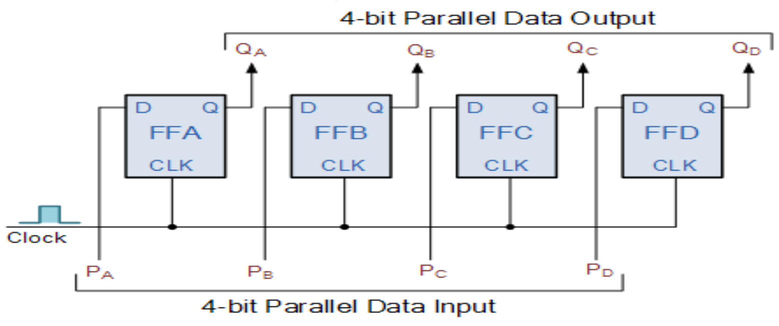
1. Serial In Serial Out (SISO)



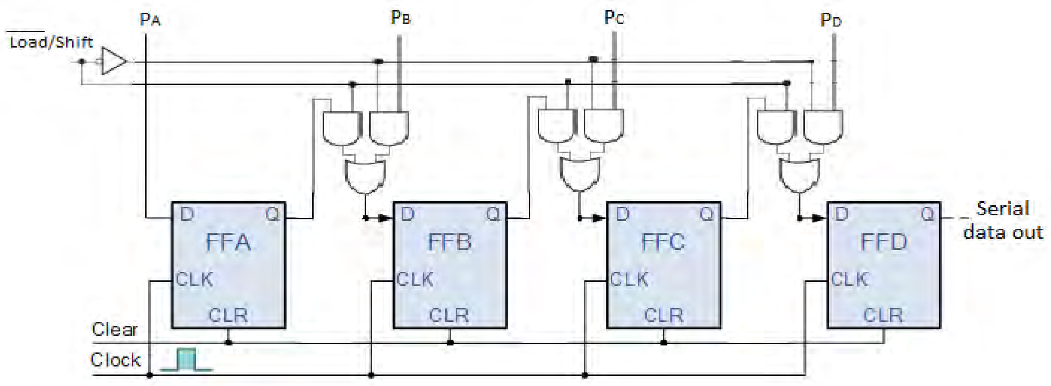
2. Serial In Parallel Out (SIPO)



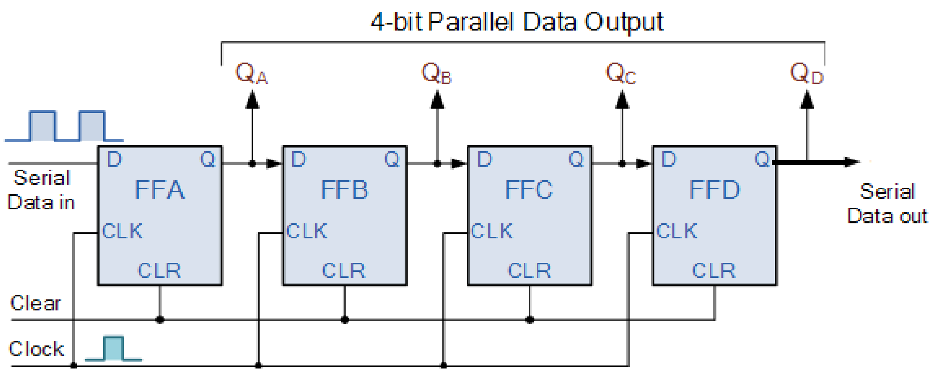
3. Parallel In Parallel Out (PIPO)



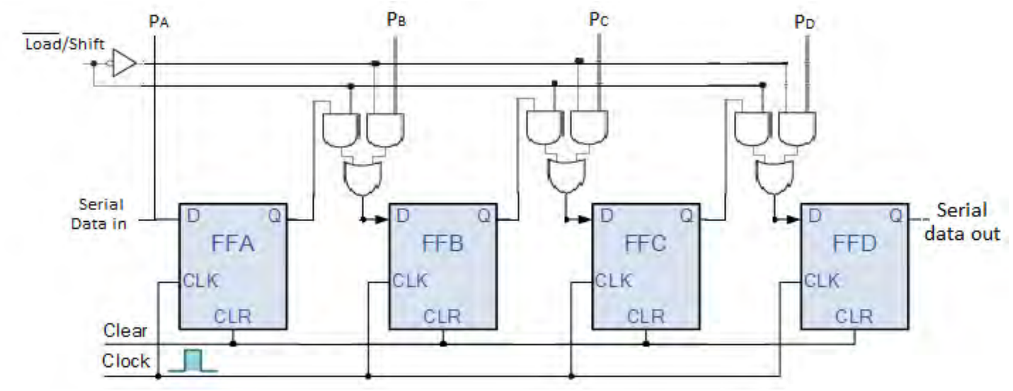
4. Parallel In Serial Out (PISO)



Serial input shift register (SISO+SIPO): As its name suggests serial input shift register allows data to enter serially. The output data can be available in parallel or serial



Serial output shift register (SISO+PISO using mode control): As its name suggests serial output shift register allows data to out serially. The input data can be available in parallel or serial



PROCEDURE

1. Test all the components using multimeter and digital IC tester
2. Set up serial input shift register using D FF. Clear all FF using clear pin. Feed 1011 to the serial input starting from LSB using the PRESET and CLEAR pins.

RESULT AND DISCUSSION

The performance of shift registers using D FF are set up and studied

<u>Model Questions</u>:

1. Design a bidirectional shift register using mode control
2. Draw the waveforms for each shift register

EXPT NO: 11

RING COUNTER AND JOHNSON (TWISTED RING) COUNTER

AIM

To design and set up four bit Johnson and ring counter using JK FF

COMPONENTS REQUIRED

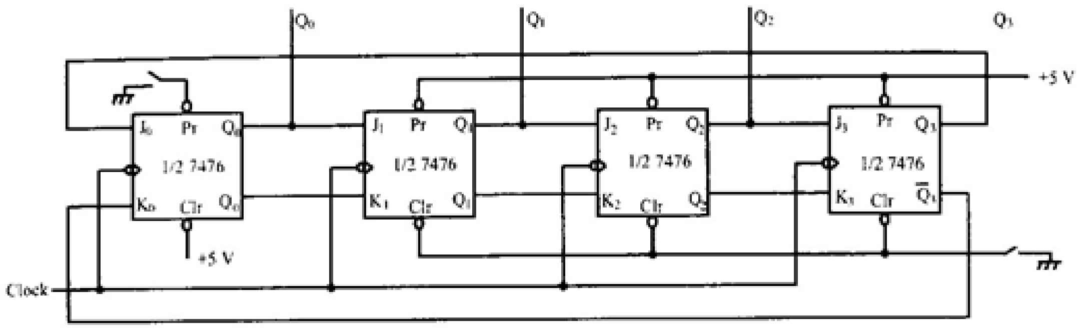
Digital IC trainer kit, IC 7476

PRINCIPLE

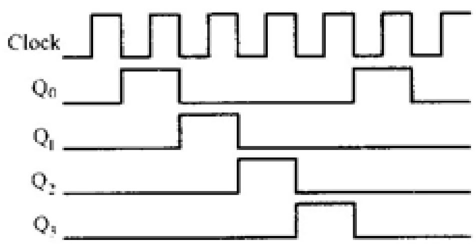
Ring counter and Johnson counters are basically shift registers

<u>Ring counter</u>

It is made by connecting Q&Q' output of one JK FF to J&K input of next FF respectively. The output of final FF is connected to the input of first FF. To start the counter the first FF is set by using preset facility and the remaining FF are reset input. When the clock arrives the set condition continues to shift around the ring



Waveforms for ring counter



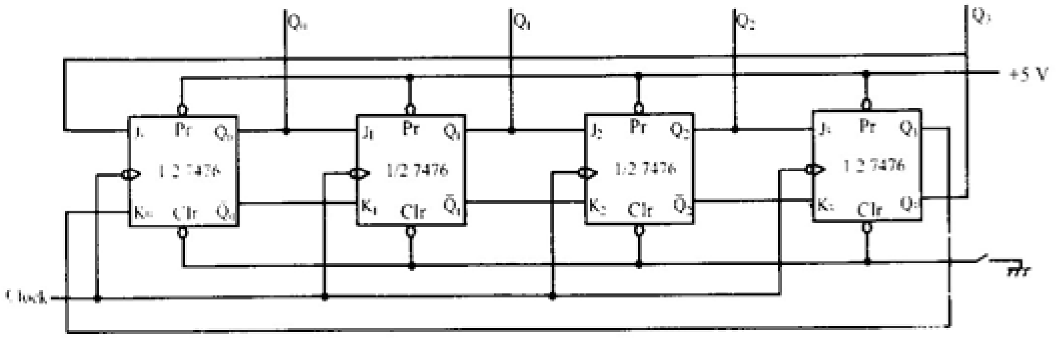
Clk	Q_0	Q_1	Q_2	Q_3
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

As it can be seen from the truth table there are four unique output stages for this counter. The modulus value of a ring counter is *n*, where *n* is the number of flip flops. Ring counter is called divided by N counter where N is the number of FF

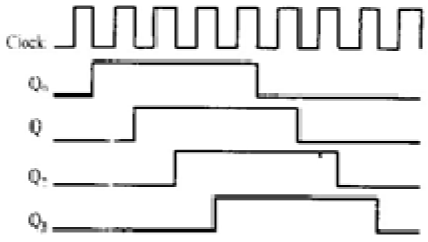
Johnson counter (Twisted ring counter)

The modulus value of a ring counter can be doubled by making a small change in the ring counter circuit. The Q' and Q of the last FFS are connected to the J and K input of the first FF respectively. This is the Johnson counter

Johnson counter



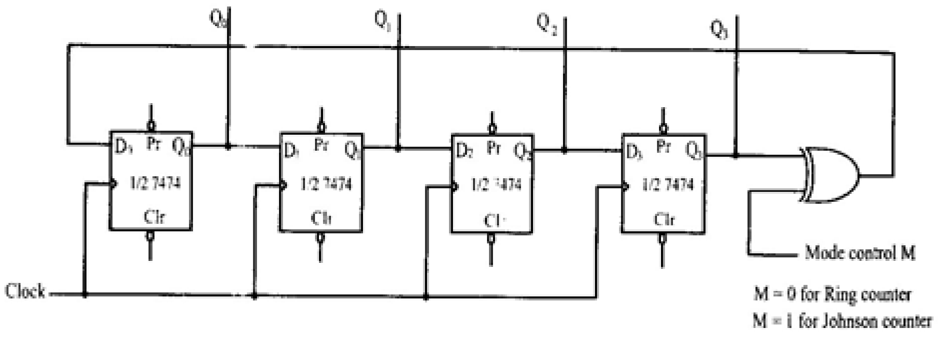
Waveforms for Johnson counter



Clk	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Initially the FFs are reset. After first clock pulse FF0 is set and the remaining FFs are reset. After the eight clock pulse all the FFS are reset. There are eight different conditions creating a mode 8 Johnson counter. Johnson counter is called a twisted ring counter or divide by 2N counter

Ring/Johnson counter using mode control



PROCEDURE

1. Set up the ring counter and set clear Q outputs using PRESET and apply mono pulse.
2. Note down the state of the ring counter on the truth table for successive clock 0.
3. Repeat the steps for Johnsons counter

RESULT AND DISCUSSION

Four bit ring counter and the Johnson counter were set up using the JK FF and verified

<u>Model Questions</u>:

1. Design a ring counter using D flip flop
2. Design a Johnson counter using D flip flop
3. Design a ring counter/Johnson counter with mode control using JK flip flop

EXPT NO: 12

ASYNCHRONOUS COUNTERS

AIM

To set up and study the working of asynchronous up counter and down counter.

COMPONENTS REQUIRED: Digital IC trainer kit, IC 7476

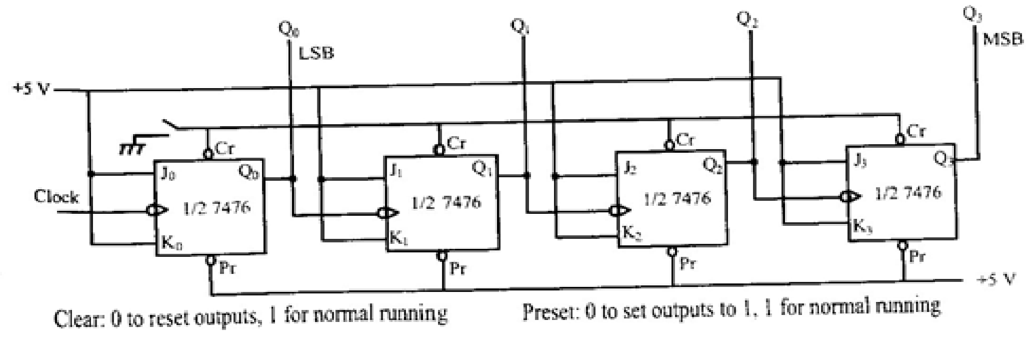
PRINCIPLE

A Counter is a circuit that produces a set of unique output combinations in relation to the number of applied input pulses. The number of unique outputs of a counter is known as modulus and the counter having N number of unique output is called mod N counter. Normally up or down counter have N=2^n where n is the number of flip flops.
In Asynchronous counters the FFs are not given the clock pulse simultaneously. There for the propagation delay increases simultaneously. There for the propagation delay increases with the number of FFs used. FF must be used in toggle mode to count states.

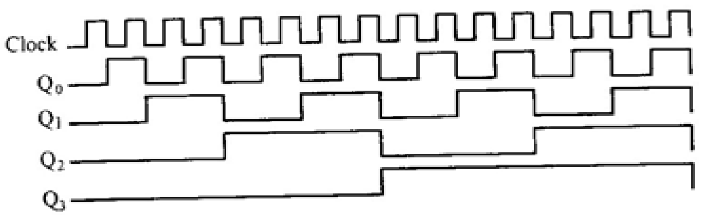
4 bit binary up counter (Ripple counter)

In this circuit all FFs are clocked by Q output of the preceding FF. JK inputs of all FF are connected to high states. 7476 in dual JK master slave FF with preset and clear. A ripple counter comprising of n FF is used to count up to 2^n pulses. A circuit with four FF gives a maximum count of 2^4=16.The counter gives a natural count from 0 to 15 and resets on 16 th pulse. With application of the first clock pulse Q_0 changes from 0 to 1 $Q_1Q_2Q_3$ remain unaffected. With second clock pulses Q_0 become 0 and Q_1 become 1. At the 16thclockpulse all Q output resets and repeat the cycle.

4-bit binary up counter


Q0
LSB
Q1
Q2
Q3
MSB
+5 V
Clock
Cr
J0
Q0
1/2 7476
K0
Pr
J1
Q1
K1
J2
Q2
K2
J3
Q3
K3
+5 V
Clear: 0 to reset outputs, 1 for normal running
Preset: 0 to set outputs to 1, 1 for normal running

Waveforms for 4-bit binary up counter

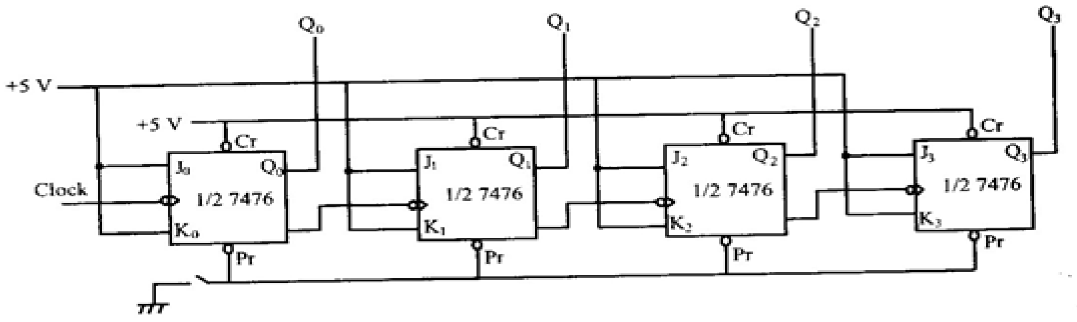


Clk	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

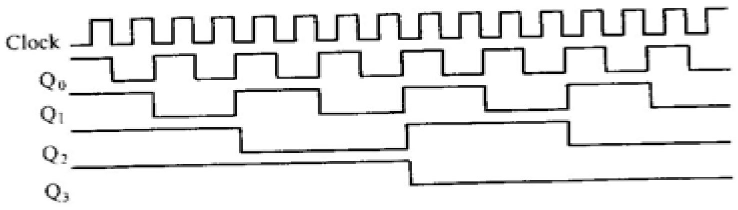
4 bit binary down counter

Here all FFs are clocked by Q' output of the preceding FF. The counter gives a natural count from 15 to 0 and sets on 16 th pulse. With application of the first clock pulse Q_0 changes from 1 to 0 $Q_1Q_2Q_3$ remain unaffected. With second clock pulses Q_0 become 1 and Q_1 become 0. At the 16thclockpulse all Q output sets and repeat the cycle.

4-bit binary down counter



Waveforms for 4-bit binary down counter



Clk	Q_3	Q_2	Q_1	Q_0
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
13	0	0	1	0
14	0	0	0	1
15	0	0	0	0

PROCEDURE

Test all ICs using Multimeters and IC tester. Then set up the circuit for up counter. Connect the preset point to +5V to disable it. Clear all flip flop outputs initially connecting common CLEAR terminal to Logic 0. After the usage of clear point connect them into Logic 1, and apply mono pulse. Counter start counting up.

Change clock inputs of all flip flops, except FF0 from Q outputs to the Q' outputs. Preset all FFs by connecting common PRESET terminal to logic 0.Apply mono pulse to down counter.

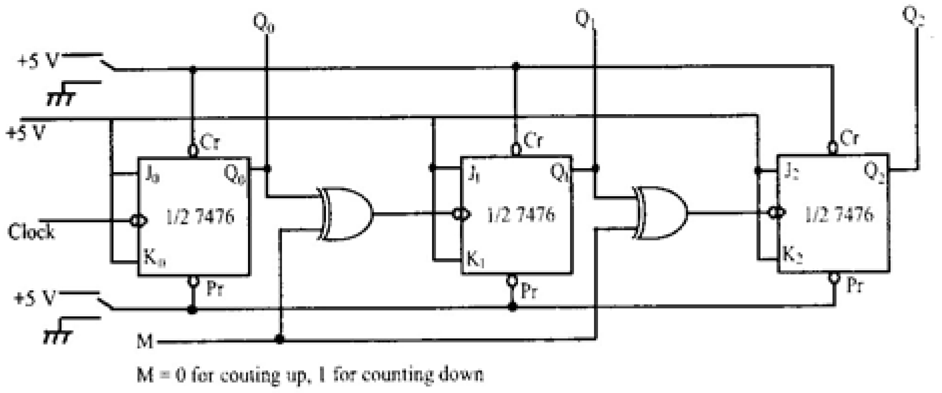
RESULT

Asynchronous up counter and down counter is designed and truth table is verified.

Model Questions:

1. Design an up counter using T flip flop
2. Design an up counter using D flip flop
3. Design an asynchronous UP/DOWN counter using mode control

3-bit up/down counter



EXPT NO: 13

DECADE COUNTER

AIM

To study about variable modulo asynchronous counters and study the working of Decade counter (a modulo 10 counter)

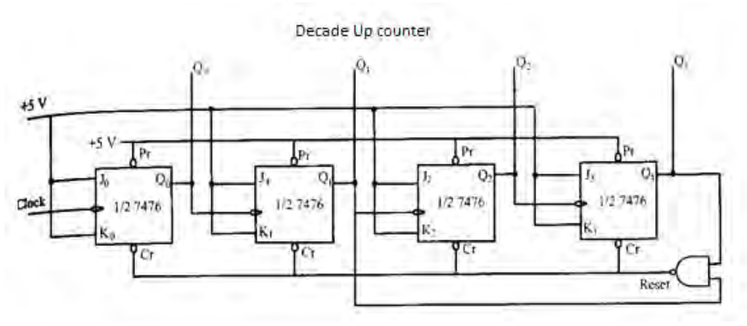
COMPONENTS REQUIRED- ICS 7476 and IC Trainer kit

PRINCIPLE

A counter is a circuit that provides a set of unique output combinations in relation to the number of applied input pulses. The number of unique outputs of a counter is known as it mod or modulus value.

Decade up counter

The circuit of the decade up counter is similar to 4 bit ripple up counter but with the aid of logic circuit the count is limited to 0 to 9. As soon as the count reaches 10_D i.e 1010, the NAND gates clear the FF and the counting restarts from zero



PROCEDURE

Set up the circuit for Decade up counter. Connect the preset point to +5V to disable it. Clear all flip flop outputs initially connecting common CLEAR terminal to Logic 0. After the usage of clear point connect them into Logic 1, and apply mono pulse. Counter start counting up.

RESULT

Asynchronous decade up counter has designed and verified truth table.

Model Questions:

1. Design a Modulo 12 (MOD 12) counter
2. Design an asynchronous decade down counter
3. Design BCD up counter

CS09 408(P): Digital Systems Lab

Teaching scheme **Credits:** 2
3 hours practical per week

Objectives
· To give a hands on experience on digital electronics components and systems; which are fundamental building blocks of the Computer systems.
· To deal extensively with the characteristic and features of indispensable digital electronic circuits and systems through structured experiments.

1. Verification of truth tables of AND, OR, NOT, NAND, NOR and XOR gates, used for gating digital signals.
2. TTL characteristics
3. Verification of the postulates of Boolean algebra and DE Morgan’s theorem using logic gates.
4. Half and full adders, half and full Subtractor.
5. Digital comparator, parity generator and checker, and code converter
6. Characteristics and operations of RS, gated RS, D, T, and JK master slave flip-flops
7. Multiplexer and Demultiplexer using gates
8. Shift register, ring counter, and twisted ring counter.
9. Decade counter and variable modulo asynchronous counter
10. Astable multivibrator and Schmitt trigger using gates, astable and monostable
11. Multivibrator and frequency divider using 555.

Reference Books
1. C Nagarath J., *Electronics Analog & Digital*, Prentice Hall India
2. Millman & Halkias, *Integrated Electronics*, Tata McGraw Hill.

Internal Continuous Assessment *(Maximum Marks-50)*
60%-Laboratory practical and record
30%- Test/s
10%- Regularity in the class

Semester End Examination *(Maximum Marks-50)*
70% - Procedure, conducting experiment, results, tabulation, and inference
20% - Viva voce
10% - Fair record