Course code	Course Name	L-T-P - Credits	Year of Introduction
CS234	DIGITAL SYSTEMS LAB	0-0-3-1	2016

Pre-requisite: CS203 Switching theory and logic design

Course Objectives:

1. To familiarize students with digital ICs, the building blocks of digital circuits
2. To provide students the opportunity to set up different types of digital circuits and study their behaviour

List of Exercises/Experiments : (minimum 12 exercises/experiments are mandatory)

1. Familiarizations and verification of the truth tables of basic gates and universal gates.
2. Verification of Demorgan's laws for two variables.
3. Implementation of half adder and full adder circuits using logic gates.
4. Implementation of half subtractor and full subtractor circuits using logic gates.
5. Implementation of parallel adder circuit.
6. Realization of 4 bit adder/subtractor and BCD adder circuits using IC 7483.
7. Implementation of a 2 bit magnitude comparator circuit using logic gates.
8. Design and implementation of code convertor circuits
9. a) BCD to excess 3 code b) binary to gray code
10. Implementation of multiplexer and demultiplexer circuits using logic gates. Familiarization with various multiplexer and demultiplexer ICs.
11. Realization of combinational circuits using multiplexer/demultiplexer ICs.
12. Implementation of SR, D, JK, JK master slave and T flip flops using logic gates. Familiarization with IC 7474 and IC 7476.
13. Implementation of shift registers using flip flop Integrated Circuits.
14. Implementation of ring counter and Johnson counter using flip flop Integrated Circuits.
15. Realization of asynchronous counters using flip flop ICs.
16. Realization of synchronous counters using flip flop ICs. Familiarization with various counter Integrated Circuits.
17. Implementation of a BCD to 7 segment decoder and display.
18. Simulation of Half adder, Full adder using VHDL.

(Note: The experiments may be done using hardware components and/or VHDL)

Course outcome:

Students will be able to:

1. identify and explain the digital ICs and their use in implementing digital circuits.
2. design and implement different kinds of digital circuits.

INTRODUCTION

STUDY OF LOGIC GATES

Aim: Truth Table verification of logic gates

General Characteristics:

a. Pin 'V_{cc}' on an I.C is to be connected to a regulated power supply of +5 volts.

b. Pin 'GND' on an I.C is to be connected to the power supply ground.

c. = High or '1' state implies a voltage between 2.4V and 5.0V in the input state.

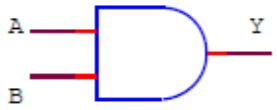
Procedure:

1. Fix the I.C on the I.C trainer kit.

2. Connections are made as shown, using the pin details of the gates. Toggle switches an LED's in the trainer are used as inputs and outputs respectively.

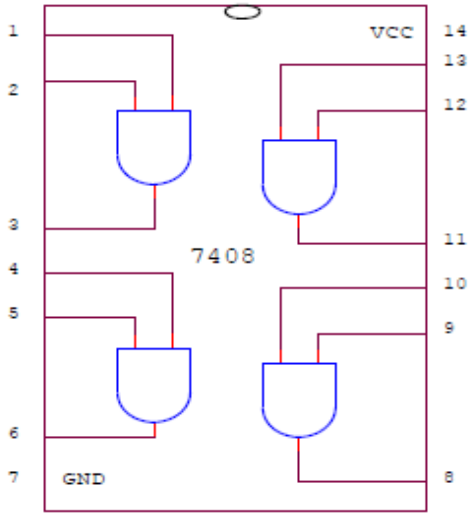
3. Switch on the supply on the trainer and verify the truth table of the gates

AND GATE (7408)

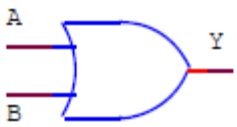


Truth Table

A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

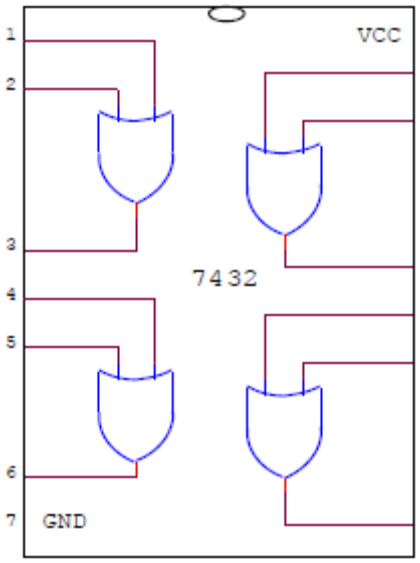


OR GATE (7432)

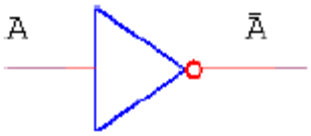


Truth Table

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

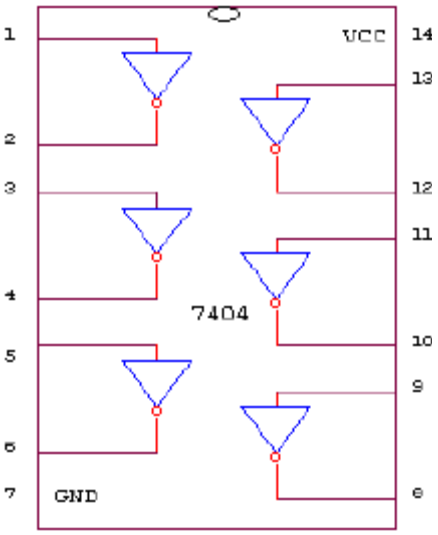


NOT GATE (7404)

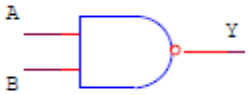


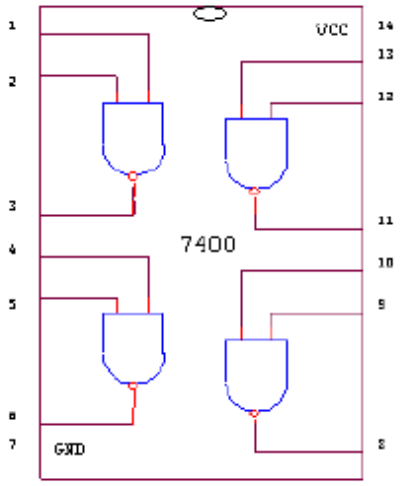
Truth Table

A	Y=Ā
0	1
1	0



NAND GATE (7400)

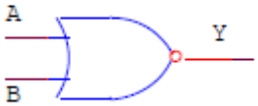


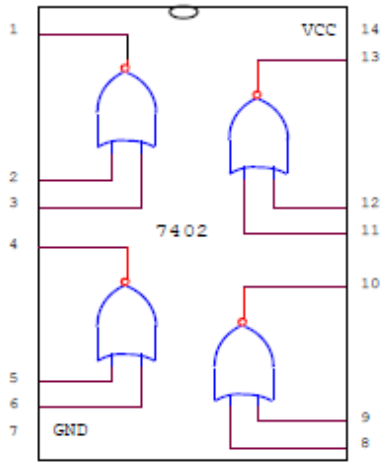


Truth Table

A	B	$Y = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR GATE (7402)





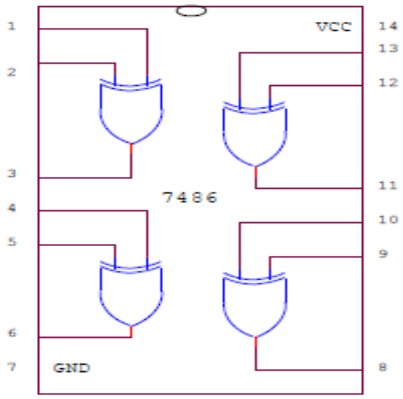
Truth Table

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

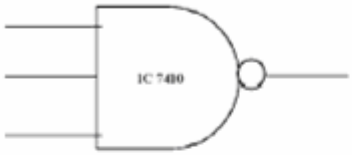
EX-OR GATE (7486)

Truth Table

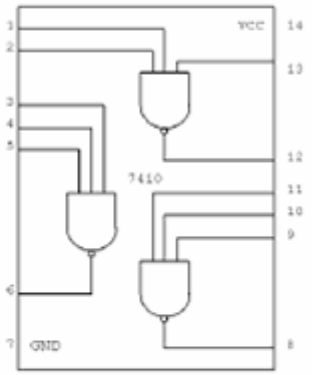
A	B	Y=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0



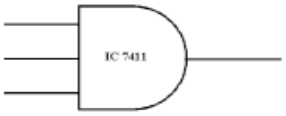
3 Input NAND Gate(7410)



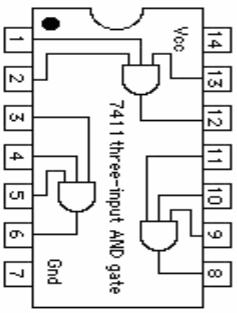
Inputs			outputs
W	X	Y	Z = W. X. Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



3 Input AND Gate(7411)



Inputs			outputs
W	X	Y	Z = W. X. Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



Result: Truth Tables of Logic Gates were verified

REALIZATION OF GATES USING UNIVERSAL BUILDING BLOCKS (NAND ONLY)

Aim: To construct logic gates NOT, AND, OR, EX-OR,EX-NOR of basic gates using NAND gate and verify their truth tables .

Apparatus:

1. IC's - 7400
2. Electronic Circuit Designer
3. Connecting patch chords.

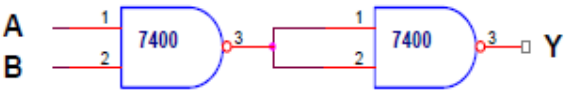
Circuit Diagrams:

TRUTH TABLE

NOT Gate

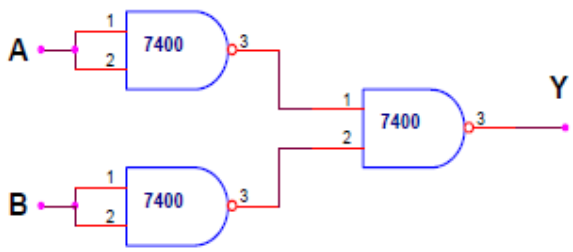
A	Y
0v	5v
5v	0v

AND Gate



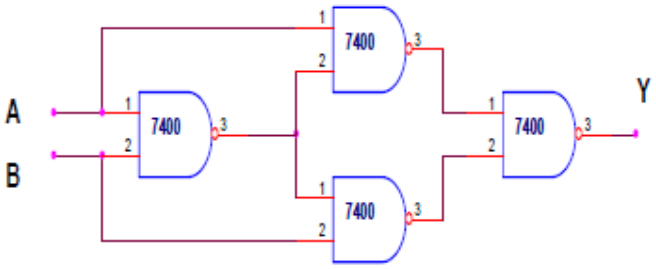
A	B	Y
0v	0v	0v
0v	5v	0v
5v	0v	0v
5v	5v	5v

OR Gate



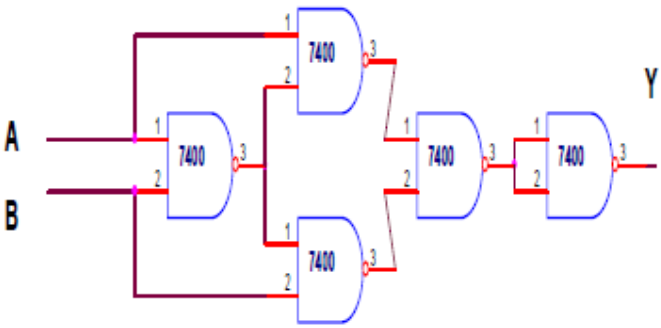
A	B	Y
0v	0v	0v
0v	5v	5v
5v	0v	5v
5v	5v	5v

EX-OR Gate



A	B	Y
0v	0v	0v
0v	5v	5v
5v	0v	5v
5v	5v	0v

EX-NOR Gate



A	B	Y
0v	0v	5v
0v	5v	0v
5v	0v	0v
5v	5v	5v

Result : Other Logic Gates were constructed using NAND Gate and their truth tables were verified.

Exercise : Realisation of other gates using NOR Gates.

VERIFICATION OF DEMORGAN'S THEOREM

AIM:

1. To verify De-Morgan's theorem for two variables
2. To realize sum of product (SOP) and product of sum (POS) expressions using universal gates.

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

THEORY:

1. De Morgan theorem states that
   a) AB'=A'+B'
   b) (A+B)'=A'.B'

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

2. 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

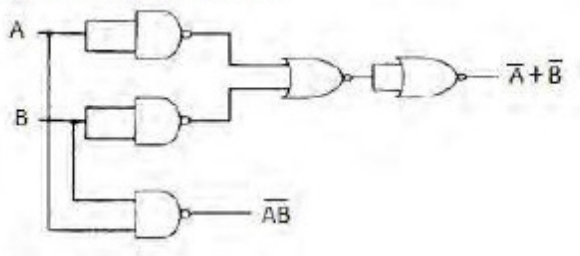
DEMORGAN'S THEOREM:

a) AB'=A'+B'

TRUTH TABLE:

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

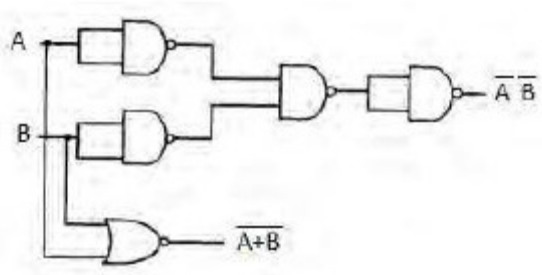
CIRCUIT DIAGRAM:



b) (A+B)'=A'.B'

TRUTH TABLE:

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



PROCEDURE:

1) Test all the IC packages using 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:

De-Morgan's theorem and postulate of Boolean algebra were verified.

DESIGN OF COMBINATIONAL LOGIC CIRCUITS

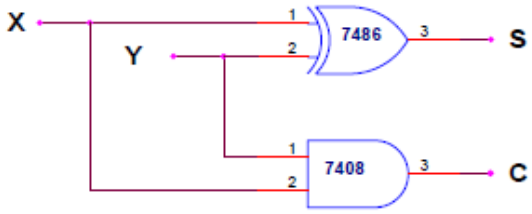
Aim: - To design and construct Half-adder, Full-adder, Half-subtractor, Full- subtractor

Apparatus: -

1. IC's - 7486, 7432, 7408, 7400
2. Electronic Circuit Designer
3. Connecting patch chords.

Circuit Diagram:-

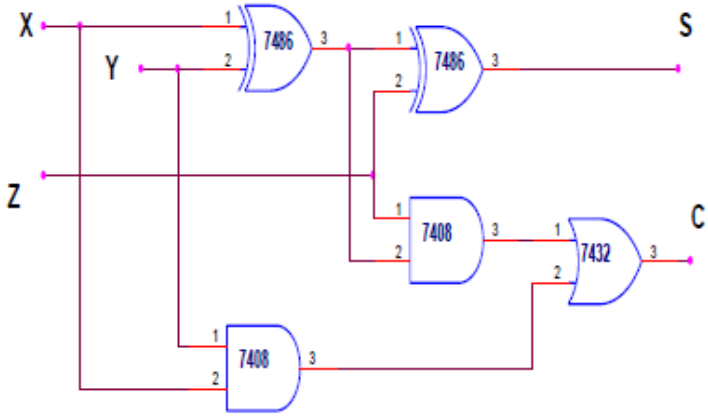
Half Adder:



TRUTH TABLE

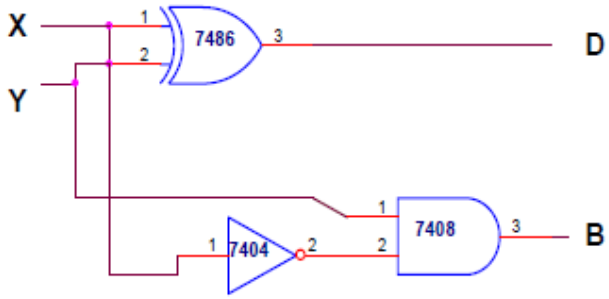
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder:



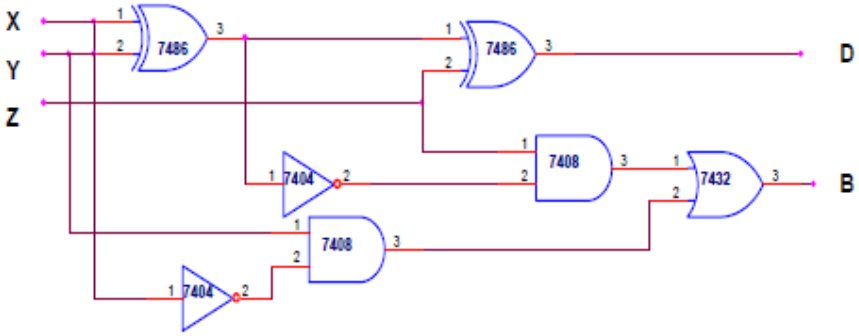
A	B	C_{N-1}	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Half Subtractor



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

Full Subtractor



A	B	C N-1	D	B
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

Procedure: -

1. Verify the gates.

2. Make the connections as per the circuit diagram.

3. Switch on V_{CC} and apply various combinations of input according to truth table.

4. Note down the output readings for half/full adder and half/full subtractor, Sum/difference and the carry/borrow bit for different combinations of inputs verify their truth tables.

Precautions:

1. All the connections should be made properly.

2. IC should not be reversed.

Result: Combinational logic circuits like Half-adder, Full-adder, Half-subtractor, Full-subtractor are constructed and truth tables are verified.

4-BIT PARALLEL ADDER/SUBTRACTOR USING IC 7483

AIM:

To perform the addition of two 4-bit numbers.

COMPONENTS REQUIRED:

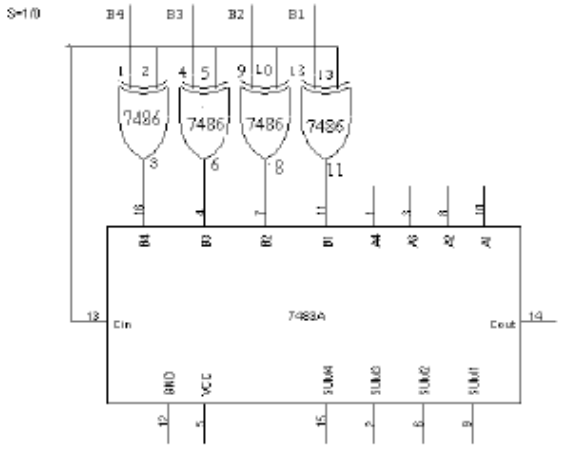
Digital Trainer Kit		01
XOR Gate	IC 7486	01
Parallel Adder/ Subtractor	IC 7483	01
Patch chords / Connecting wires		25

THEORY:

Parallel adders are ripple carry type in which the carry output of each full adder stage is connected to the carry input of the next higher order stage, the 7483 IC is a 4-bit parallel adder used to perform subtraction as well, the two 4-bit binary numbers $A_4A_3A_2A_1$ and $B_4B_3B_2B_1$ are added to give a sum $S_4S_3S_2S_1$ and carry C_{out} when carry in (C_{IN}) and 'S'is '0'.The 1's compliment subtraction is carried out when carry out (Cout) connected to Cin and 'S'= '1', The 2's compliment subtraction is carried out when carry in (C_{IN}) and 'S'is '1'.

PROCEDURE:

1. Connections are made as shown in the logic diagram using the pin details of the gates and IC 7483.
2. Connect V_{cc} & GND to respective pins of ICs
3. Switch on the Trainer kit.
4. Apply the inputs and observe the outputs.
5. Compare the practical value with the theoretical value.



Observation and calculations:

- **4 Bit adder operations.**

1. Let A = 1100 ($A_4 A_3 A_2 A_1$) , B = 0011 ($B_4B_3 B_2 B_1$) Then
   Sum ($S_4 S_3 S_2 S_1$) = 1111 and $C_{out} = 0$ (With S = 0 and $C_{in} = 0$).
2. Let A = 1001 ($A_4 A_3 A_2 A_1$) , B = 1101($B_4B_3 B_2 B_1$) Then
   Sum ($S_4 S_3 S_2 S_1$) = 0110 and $C_{out} = 1$ (With S = 0 and $C_{in} = 0$).

TRUTH TABLE:

PARALLEL ADDER				
Parameters	Theoretical		Practical	
Augend	1100	1001	1100	1001
Addend	0011	1101	0011	1101
Sum	1111	0110	1111	0110
Carry	0	1	0	1

- **4 Bit subtraction operation.**

One's Complement Method

Perform Subtraction of: $A_4 A_3 A_2 A_1$= 1 0 0 1
$B_4B_3 B_2 B_1$= 0 0 1 1

One's complement of the Subtrahend is

$B_4B_3 B_2 B_1$	=		1	0	0	0
$A_4 A_3 A_2 A_1$	=	(+)	1	0	0	1
		①	0	1	0	1
		→				1 (+)
($S_4 S_3 S_2 S_1$)	=		0	1	1	0

Two's Complement Method

Perform Subtraction of: $A_4 A_3 A_2 A_1$= 1 0 0 1
$B_4B_3 B_2 B_1$= 0 0 1 1

One's complement of the Subtrahend is

$B_4B_3 B_2 B_1$	=		1	1	0	0	
						1	
			1	1	0	1	
$A_4 A_3 A_2 A_1$	=		1	0	0	1	(+)
($S_4 S_3 S_2 S_1$)		①	0	1	1	0	
		→	End around Carry is disregarded				

TRUTH TABLE:

PARALLEL SUBTRACTOR				
	One's Complement		Two's Complement	
Parameters	Theoretical	Practical	Theoretical	Practical
Minuend	1001	1001	1001	1001
Subtrahend	0011	0011	0011	0011
Difference	0110	0110	0110	0110

RESULT:

Thus the operation of Parallel adder and Parallel Subtractor were studied and checked using IC 7483.

MODE CONTROLLED 4-BIT BINARY ADDER/SUBTRACTOR CIRCUIT

Aim:

To design and set up the following adder/ subtractor circuit using a 4-bit binary adder IC 7483

Components Required:

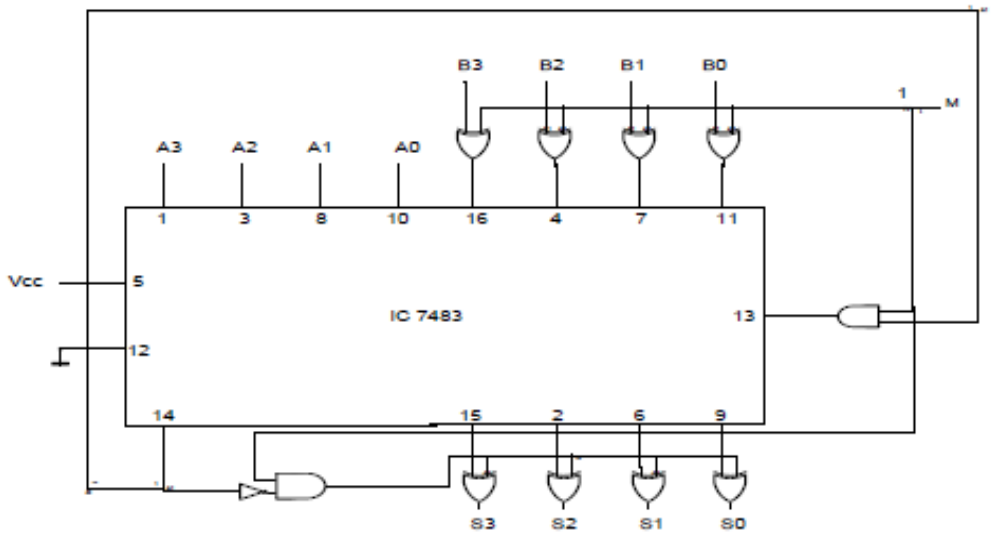
IC 7483, IC 7486, breadboard, logic probe etc..

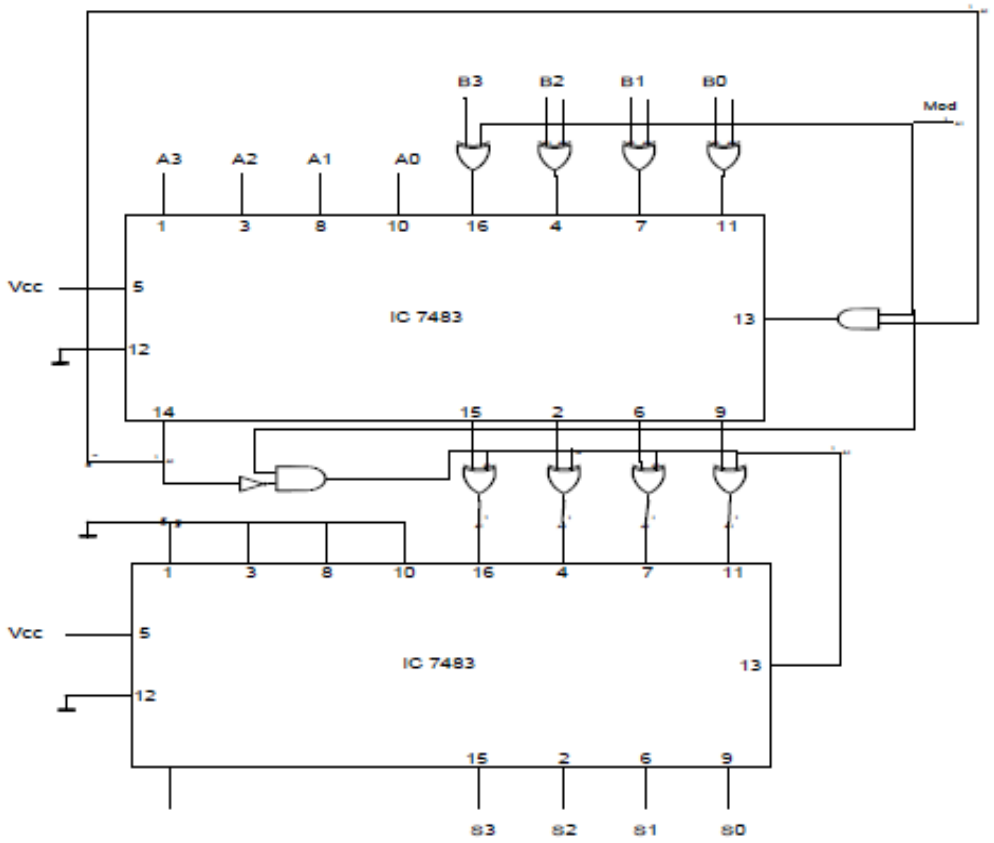
Principle :

IC 7483 performs the addition of two 4-bit binary numbers $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ and carry input to give the output $S_3S_2S_1S_0$ and carry out. So for adding the two numbers $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$, the two numbers are given to input terminals 1,3,8,10 and 16,4,7,11 of the IC 7483 and carry in the terminal 13 is set to zero. To subtract two numbers by two's compliment method, we are adding the 2's compliment of the second number to each of the four bits of the first numbers. The final carry is neglected and the difference is taken from $S_3S_2S_1S_0$.

In the circuit we set mode control such that when the mode control is zero, addition is performed and subtraction is performed when the mode control is one. We use XOR gates to feed the input so that when mode control is one, the complement of each of the four bits are fed and when mode control is zero, the input as such is fed.

In 1's complement subtraction, the complement of the subtraction is taken and added with the other number. The final carry is then added to the LSB of the result. In case there is no carry, the complement of the result is taken and this will be a negative number. This indicates that, subtraction is performed from a smaller number.





Result : A mode controlled 4 bit binary adder-subtractor circuit was set up using IC 7483 and its output was verified.

VERIFICATION OF TRUTH TABLES OF FLIPFLOPS USING GATES

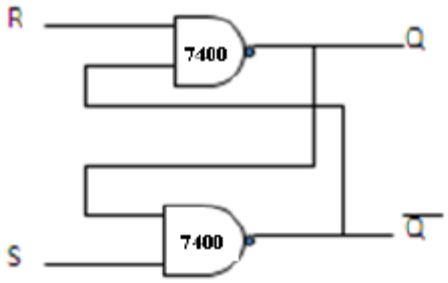
Aim: - To design and construct basic flip-flops R-S ,J-K,J-K Master slave flip-flops using gat and verify their truth tables

Apparatus: -

1. IC's - 7404, 7402, 7400
2. Electronic circuit designer
3. Connecting patch chords

Circuit Diagrams:-

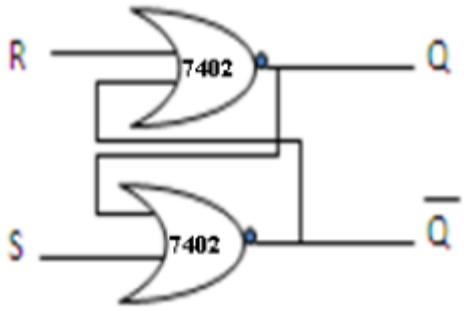
Basic flipflop using NAND gates



Truth Table

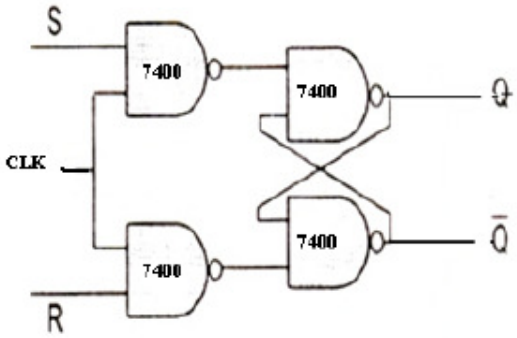
S	R	Q
0	0	Forbidden
0	1	1
1	0	0
1	1	No Change

Basic flipflop using NOR gates



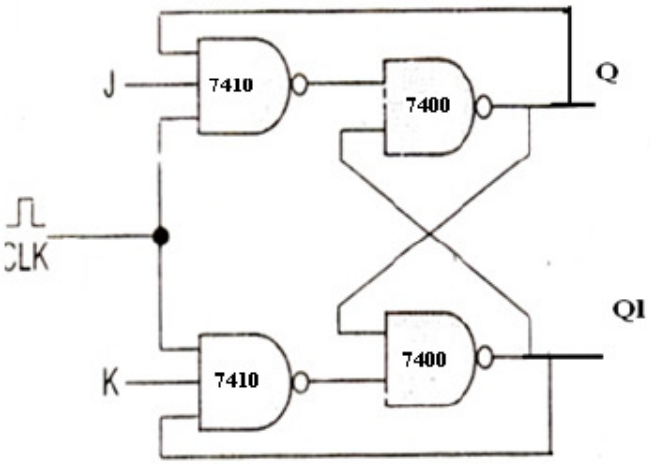
S	R	Q
0	0	No Change
0	1	0
1	0	1
1	1	Forbidden

R-S flip-flop using NAND gates



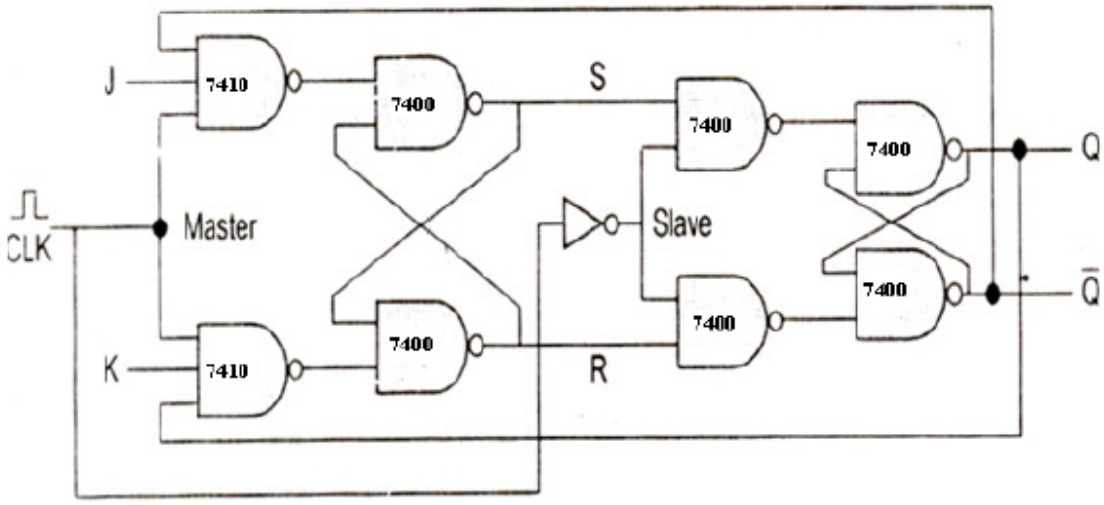
S	R	Q
0	0	No Change
0	1	0
1	0	1
1	1	Forbidden

J-k flip-flop using NAND gates



J	K	Q
0	0	No Change
0	1	0
1	0	1
1	1	Race around

J-K Master Slave using NAND gates



J	K	Q
0	0	
0	1	0
1	0	1
1	1	

Procedure:

1. Connect the Flip-flop circuits as shown above.

2. Apply different combinations of inputs and observe the outputs

Precautions: All the connections should be made properly.

Result: Different Flip-flops using gates are constructed and their truth tables are verified

DESIGN OF CODE CONVERTORS (BINARY TO GRAY AND GRAY TO BINARYCONVERSION)

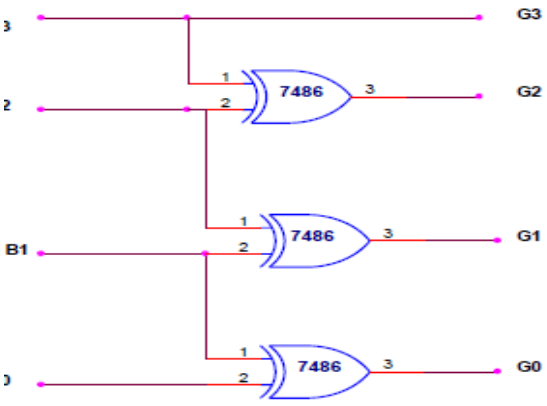
Aim: To design code converters and verify their truth tables

Apparatus:

1. IC - 7486
2. Electronic circuit designer
3. Connecting patch chords

Circuit Diagrams:

Binary to Gray Code Converter

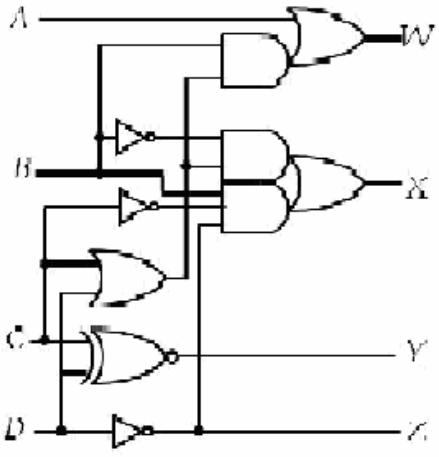


Truth table

B3	B2	B1	BO	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

BCD to Excess 3 Code Converter

- Excess-3 code for a decimal digit is the binary value for the decimal number plus 3.



input ABCD	output WXYZ
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

CD / AB	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$W = A + BC + BD$

CD / AB	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	x	x	x	x
10	0	1	x	x

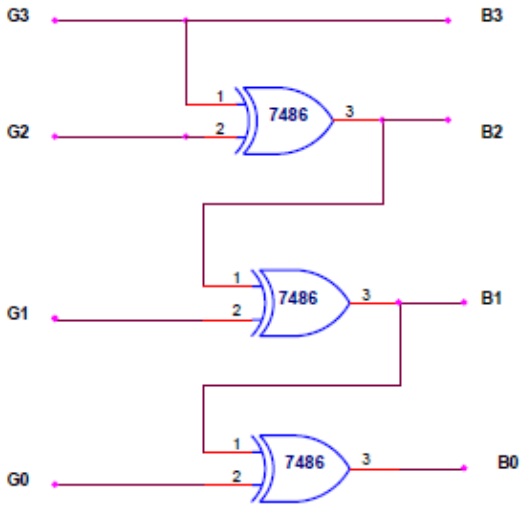
$X = B'C + B'D + BC'D'$

CD / AB	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	x	x	x	x
10	1	0	x	x

$Y = CD + C'D'$

D = Z'

Gray to Binary Code Converter



G3	G2	G1	GO	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

Procedure: -

1. The circuit connections are made as shown in fig.
2. Pin (14) is connected to +Vcc and Pin (7) to ground.
3 In the case of binary to gray conversion, the inputs B0, B1, B2 and B3 are given at respective pins and outputs G0, G1, G2, G3 are taken for all the 16 combinations of the input.
4. In the case of gray to binary conversion, the inputs G0, G1, G2 and G3 are given at respective Pins and outputs B0, B1, B2, and B3 are taken for all the 16 combinations of inputs.
5. The values of the outputs are tabulated.

Result: code converters are designed and their truth tables are verified.

Precautions: All the connections should be made properly.

DESIGN OF MULTIPLEXERS/DEMULTIPLEXERS

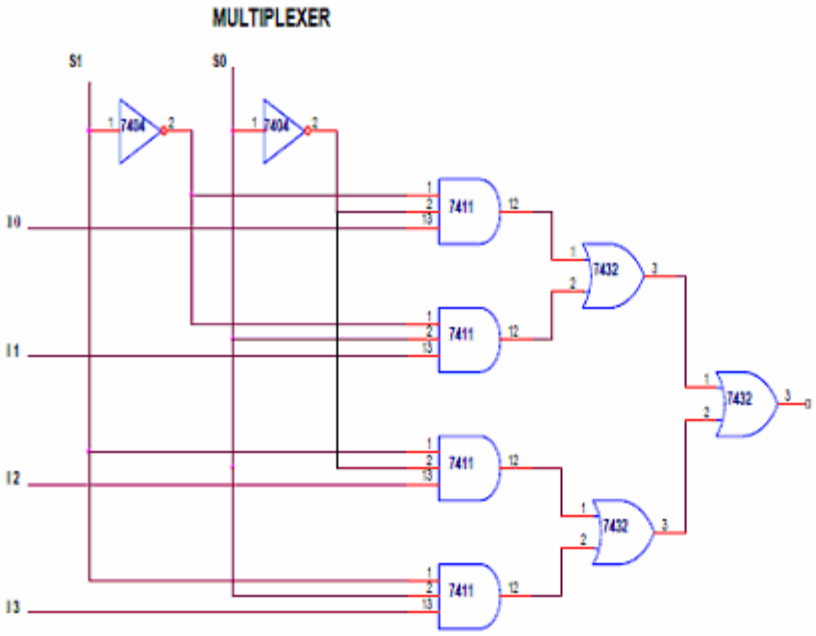
Aim: To design Multiplexer and Demultiplexer and verify their truth tables

Apparatus:

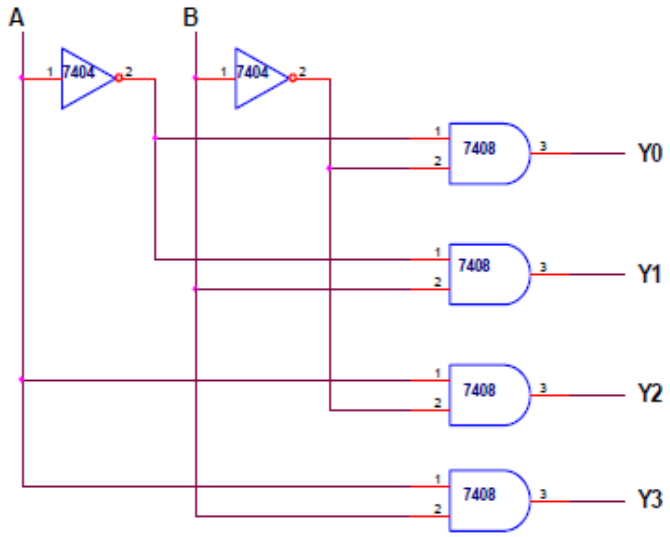
1. IC - 7404,7411,7432,7408
2. Electronic circuit designer
3. Connecting patch chords

Circuit Diagrams:

S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3



Demultiplexer



A	B	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
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.

Precautions: All the connections should be made properly.

Result: Multiplexer and Demultiplexer are constructed and the truth tables are verified

3 VARIABLE FUNCTION USING IC 74151

AIM:

To Realize the 3 variable function using IC 74151.

COMPONENTS REQUIRED:

Digital Trainer Kit		01
NOT Gate	IC 7404	01
8×1 MUX	IC 74151	01
Patch chords / Connecting wires		20

TRUTH TABLE:

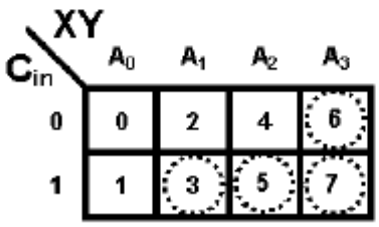
Full Adder using MUX

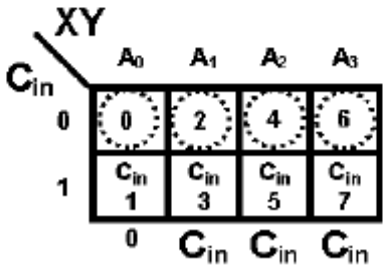
INPUTS			OUTPUTS	
X	Y	C_{IN}	S (Sum)	C_{OUT} (Carry)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Implementation table for Sum and Carry:

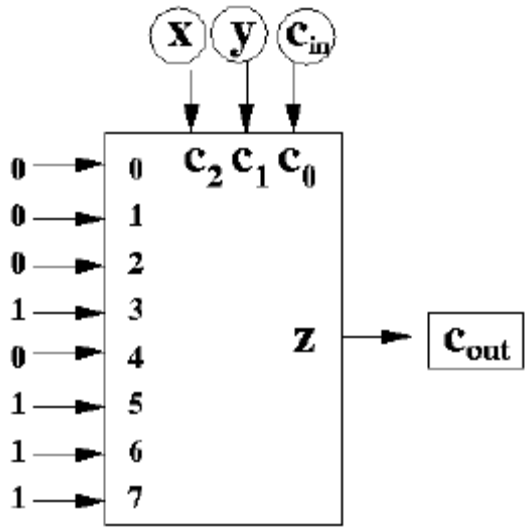
Sum = Σm (1,2,4,7)

Carry = Σm (3,5,6,7)

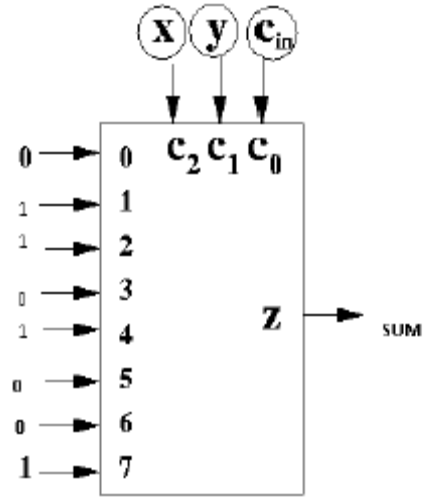




LOGIC DIAGRAM FOR CARRY:



LOGIC DIAGRAM FOR SUM:



RESULT:

The functions of MUX using 74151 IC & Realization of Arithmetic circuits using MUX IC 74151 were studied.

Exercise : Familiarisation of MUX/DeMUX ICs IC 74150 and IC 74154

REALIZATION OF 1:8 DEMUX USING IC 74138

AIM:

To study the function of DEMUX using IC 74138.

COMPONENTS REQUIRED:

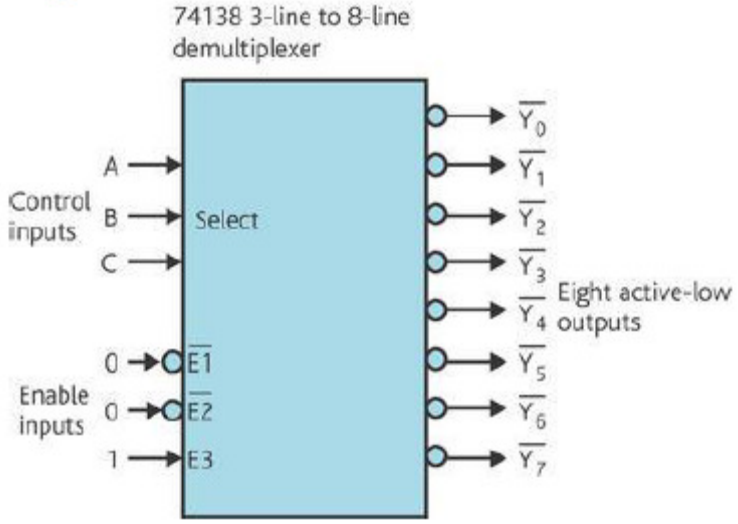
Digital Trainer Kit		01
NOT Gate	IC 7404	01
OR Gate	IC 7432	01
NAND Gate (2 Input)	IC 7410	01
NAND Gate (3 Input)	IC 7420	01
1×4 DEMUX	IC 74139	02
Patch chords / Connecting wires		20

THEORY:

The Demultiplexer is a combinational logic circuit which takes one input data source and selectively distributes it to N output channels just like a multiposition switch. The data distributor, known more commonly as a Demultiplexer or "Demux" for short, is the exact opposite of the Multiplexer.The demultiplexer takes one single input data line and then switches it to any one of a number of individual output lines one at a time. The demultiplexer converts a serial data signal at the input to a parallel data at its output lines.

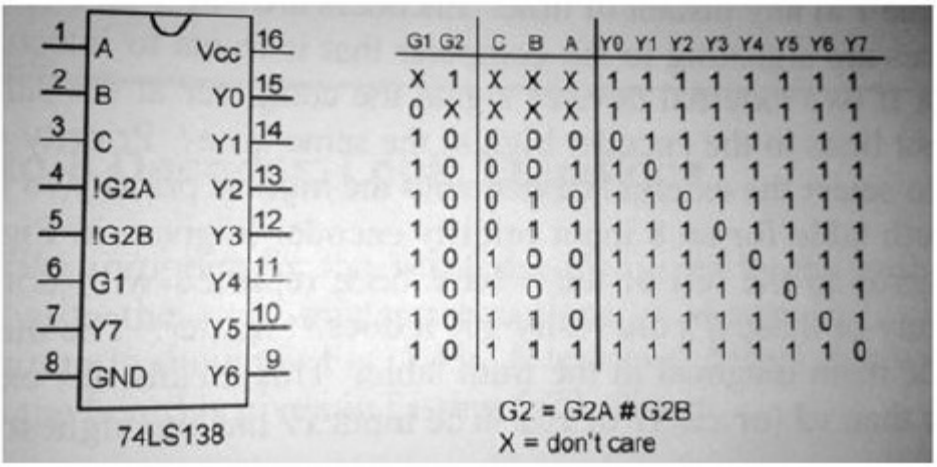
LOGIC SYMBOL:

1:8 DE MUX Using 74138 I.C



TRUTH TABLE:

74LS138 (3- to- 8 ACTIVE -LOW DECODER)



G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

G2 = G2A # G2B
X = don't care

PROCEDURE:

1. Connections are made as shown in the logic diagram using the pin details of the gates.
2. Connect V_{cc} and GND to respective pins of each IC.
3. Connect the data, select and enable inputs to the toggle switches and outputs to the LED's
4. Switch on the supply on the Trainer
5. Verify the truth table of the De-Multiplexer.

RESULT:

The function of DEMUX/DECODER using IC 74138 alization of Code Converters using DEMUX 74139 were studied.

RING COUNTER AND JOHNSON (TWISTED RING) COUNTER

AIM:

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

COMPONENTS REQUIRED:

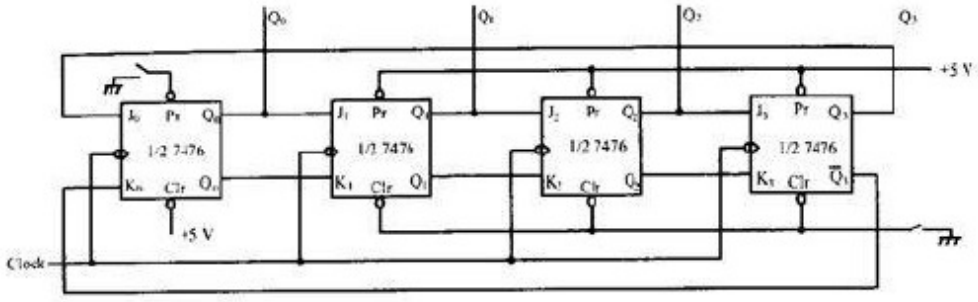
Digital IC trainer kit, IC 7476

THEORY:

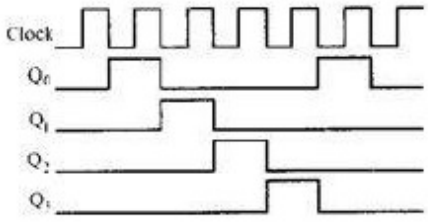
Ring counter and Johnson counters are basically shift registers Ring

Ring counter:

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



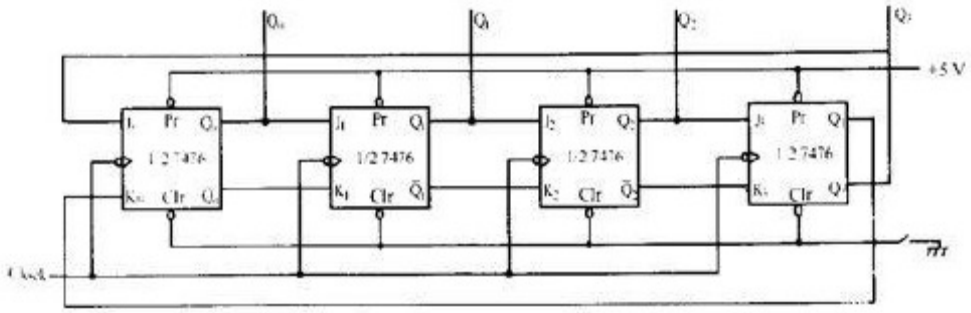
Clk	Q_0 Q_1 Q_2 Q_3
1	1 0 0 0
2	0 1 0 0
3	0 0 1 0
4	0 0 0 1

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

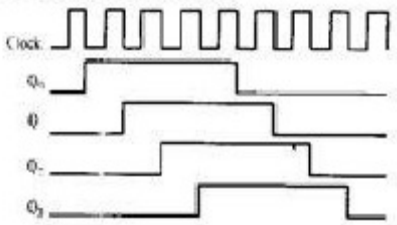
Johnson counter (Twisted ring counter)

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



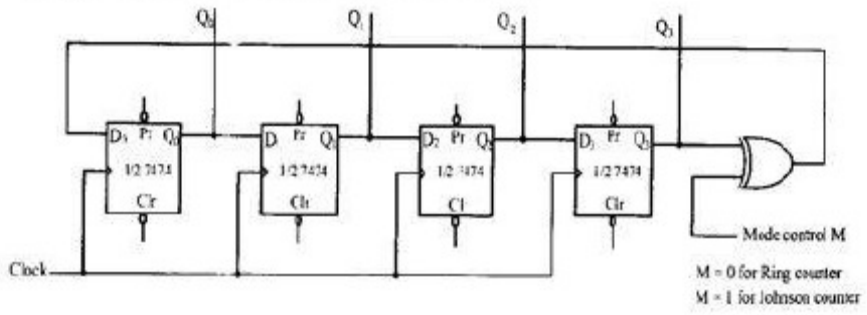
Waveforms for Johnson counter



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 :

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

REALIZATION OF SHIFT REGISTERS USING IC7474

AIM:

To implement different types of shift registers like Serial In Serial Out [SISO], Serial In Parallel Out [SIPO], Parallel In Parallel Out [PIPO] and Parallel In Serial Out [PISO] using D-flip flops and to verify their observation table.

COMPONENTS REQUIRED:

Trainer Kit	01
IC 7474	02
Ic 7408	01
Patch chord	20

THEORY:

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. A register capable of shifting its binary information either to the left or to the right is called a shift register. The logical configuration of a shift register consists of a chain of flip flops connected in cascade with the output of one flip flop connected to the input of the next flip flop. All the flip flops receive a common clock pulse which causes the shift from one stage to the next.

The Q output of a D flip flop is connected to the D input of the flip flop to the left. Each clock pulse shifts the contents of the register one bit position to the right. The serial input determines, what goes into the right most flip flop during the shift. The serial output is taken from the output of the left most flip flop prior to the application of a pulse. Although this register shifts its contents to its left, if we turn the page upside down we find that the register shifts its contents to the right. Thus a unidirectional shift register can function either as a shift right or a shift left register.

The binary information (data) in a register can be moved within or into or out of the register upon application of clock pulses. This type of bit movement or shifting is essential for certain arithmetic and logic operations used in microprocessors. This gives rise to group of registers

called shift registers. They are very important in applications involving the storage and transfer of data in a digital system.

TYPES OF SHIFT REGISTERS :

Serial In Serial Out [SISO]:

In this type of register, the output of one flip-flop is connected to the input of the next flip-flop. Output of the register is obtained from the last flip-flop. Depending on the direction of the input given shifting takes place in this. Bit by bit loading is done with every clock pulse and shifting takes place with every clock pulse.

Serial In Parallel Out [SIPO]:

This is similar to SISO except that the output is taken from each flip-flop. Thereby the shifted value is shown at once.

Parallel In Parallel Out [PIPO]:

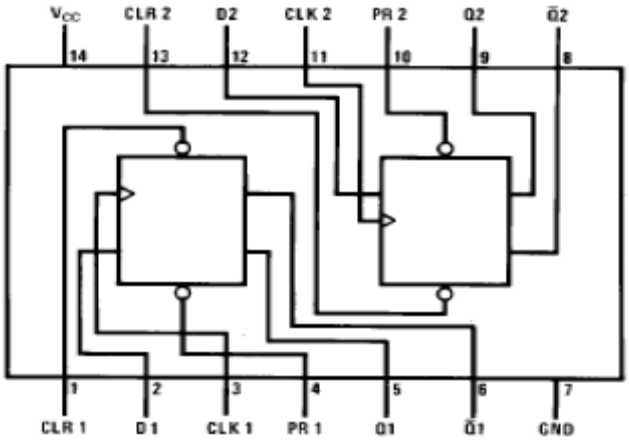
Upon giving clock pulse, data is loaded in parallel in all flip-flops. Output is taken from each of the flip-flop.

Parallel In Serial Out [PISO]:

Here we use a control input Load/ (Shift)' such that if Load/ (Shift)' = 1, data is loaded in all flip-flops in parallel and when the Load/ (Shift)' = 0, data is shifted with every clock pulse. Output is obtained from the last flip-flop.

IC 7474 consists of two D flip-flops with PRESET & CLEAR. The pin diagram is as shown in figure.

PIN DIAGRAM OF IC 7474:



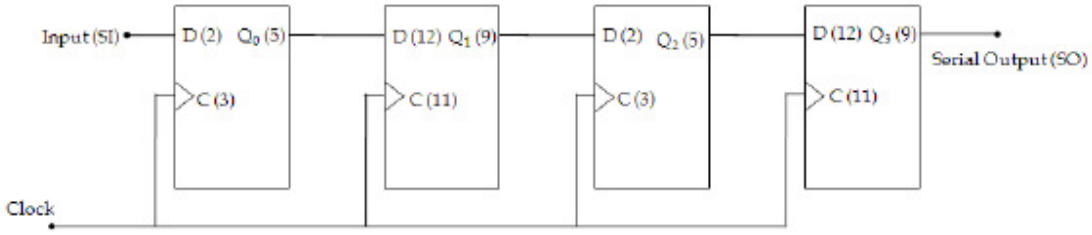
PROCEDURE:

1. Test all the ICs manually/ using IC tester.
2. Connect VCC and the ground.
3. Connect the appropriate pins to the input and output LEDs and switches.
4. Verify the truth table with respect to the clock.

FUNCTION TABLE FOR IC 7474:

Inputs				Outputs	
Preset	Clear	Clock	D	Q	Q`
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	1	1
1	1	1	0	0	1
1	1	1	1	1	0
1	1	0	X	No change	No change

LOGIC DIAGRAM:
Serial In Serial Out:

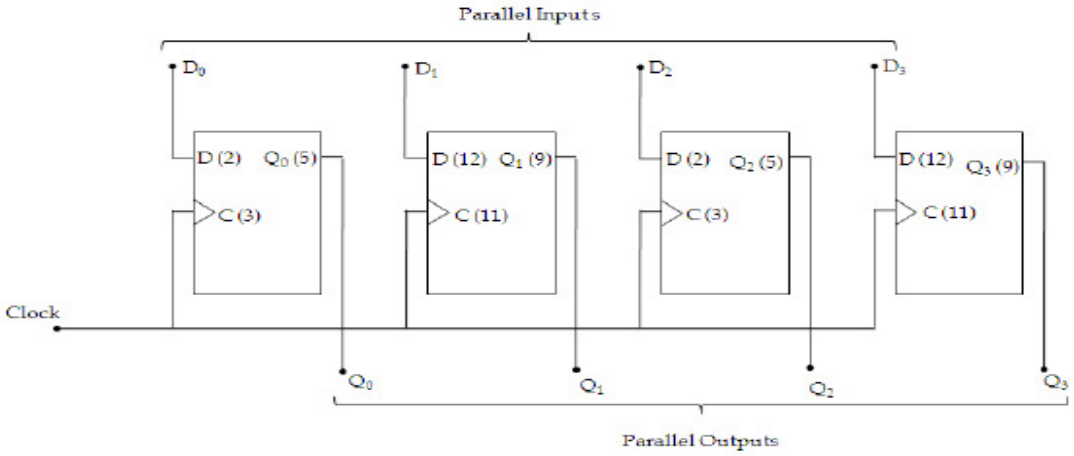


Figure(1). Connection diagram for Serial in Serial out Shift Register (Right Shift)
Observation Table:

Observation Table:

Input	Clock	Shift Register				Serial Output
(SI)	Pulse	Q_0	Q_1	Q_2	Q_3	(SO)
0	↑	0	-	-	-	-
1	↑	1	0	-	-	-
1	↑	1	1	0	-	-
0	↑	0	1	1	0	0
1	↑	1	0	1	1	1
0	↑	0	1	0	1	1

Serial In Parallel Out:



Observation Table:

Input (SI)	Clock Pulse	Outputs			
		Q_0	Q_1	Q_2	Q_3
0	↑	0	-	-	-
1	↑	1	0	-	-
1	↑	1	1	0	-
0	↑	0	1	1	0
1	↑	1	0	1	1
0	↑	0	1	0	1

Parallel In Parallel Out:

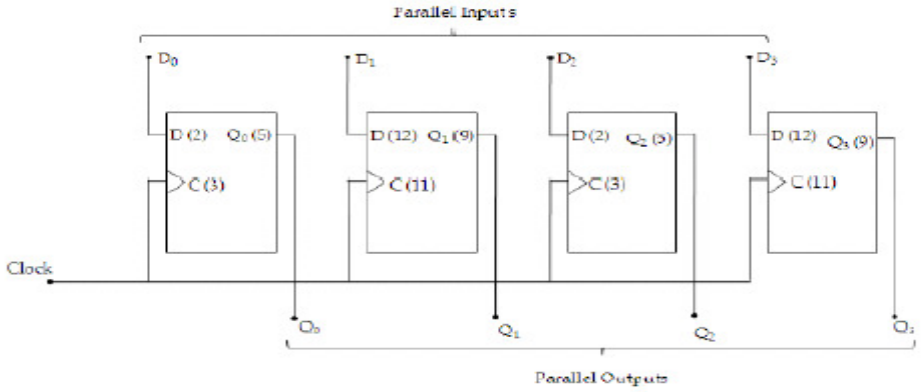


Figure. Connection diagram for Parallel in Parallel Out Shift Register

Observation Table:

Clock Pulse	Inputs				Outputs			
	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3
↑	0	1	0	1	0	1	0	1
↑	1	1	1	0	1	1	1	0
↑	1	0	1	0	1	0	1	0
↑	0	1	1	0	0	1	1	0

Parallel In Serial Out:

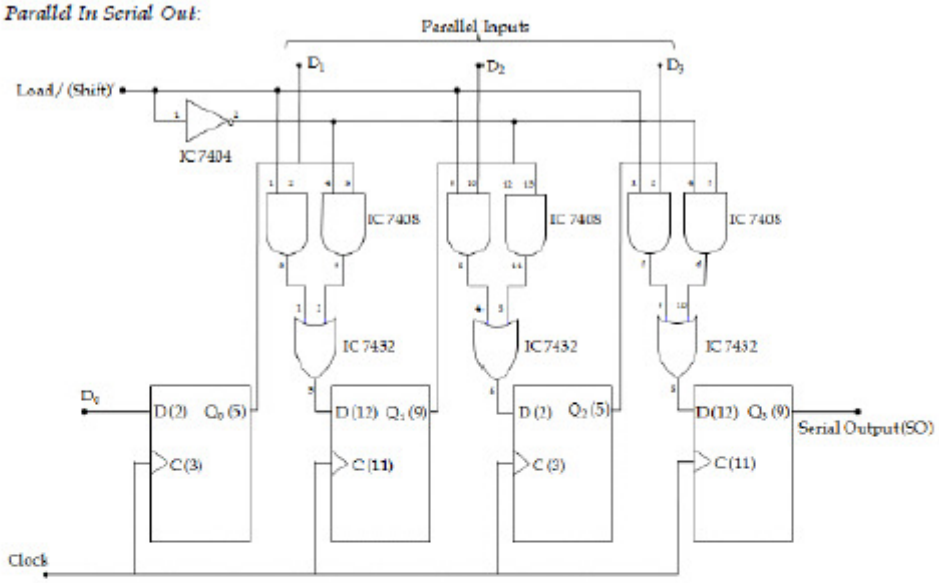


Figure. Connection diagram for Parallel in Serial Out Shift Register

Observation Table:

Load/ (Shift)'	Clock Pulse	Inputs				Serial Output (SO)
		D_0	D_1	D_2	D_3	
1	↑	0	1	0	1	1
0	↑	0	0	1	0	0
0	↑	0	0	0	1	1
0	↑	0	0	0	0	0
1	↑	1	1	0	1	1

RESULT:

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

REALIZATION OF MOD – N COUNTER USING 7490

AIM:

To realize a Modulo N-counter using 7490 and verify the expected truth table and display the output waveform for a square wave input of given frequency. (N–to be specified, N≤ 9).

COMPONENTS REQUIRED:

Trainer Kit	01
IC 7490	01
IC 7400	01
IC 7410	01
Patch chord	25

THEORY:

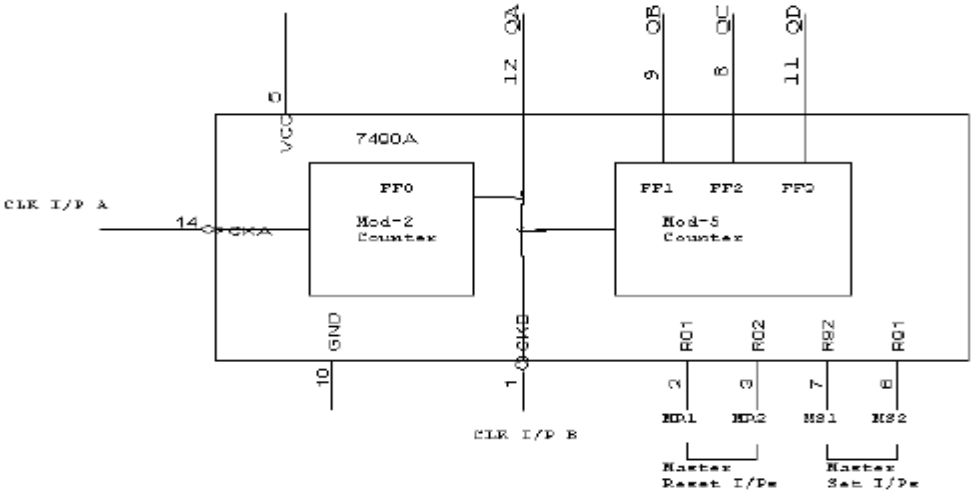
7490 is a ÷10 counter using 4 master slave JK flip-flops.

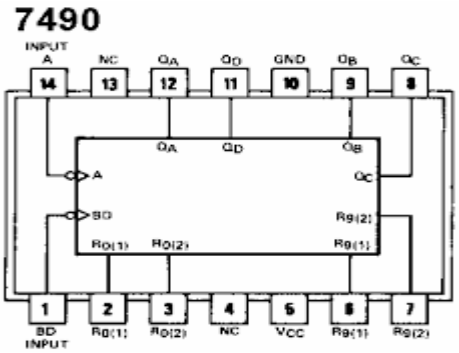
It contains ÷2 and ÷5 counters, which can be cascaded to give a ÷10 counter.

PROCEDURE:

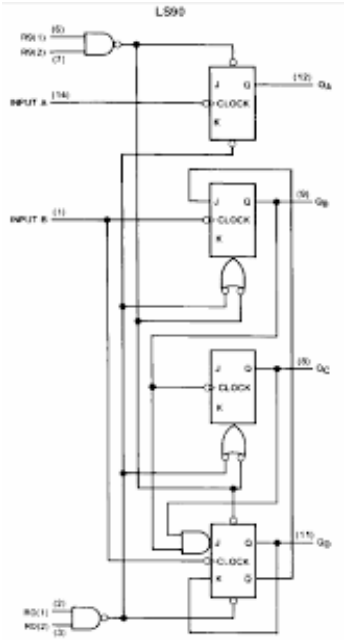
1. Connections are made as shown in the circuit diagram using 7490 ,7410 and 7400 IC.
2. Apply the clock pulse and verify the truth table

LOGIC DIAGRAM:

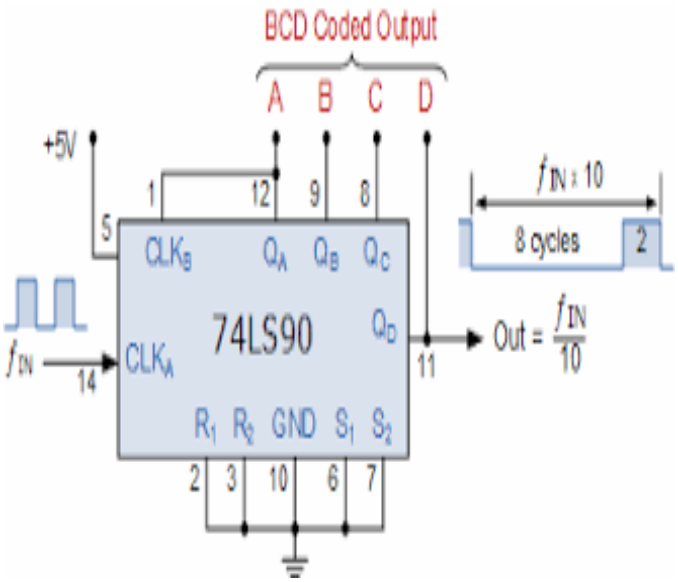




Pin Out of IC 7490



Internal Diagram of IC 7490



Circuit diagram of a decade counter

Working :

IC7490A monolithic counter contains four master- slave flip-flops and additional gating to provide a divide-by-two counter and a three-stage binary counter for which the count cycle length is divide-by-five. The counter has a gated zero reset and also has gated set-to-nine inputs for use in BCD nine's complement applications. To use the maximum count length (decade or four-bit binary), the B input is connected to the QA output. The input count pulses are applied to input A and the outputs are as described in the appropriate Function Table. A symmetrical divide-by-ten count can be obtained from the counters by connecting the QD output to the A input and applying the input count to the B input which gives a divide-by-ten square wave at output QA.

Note 1: Output QA is connected to input B for BCD count.

Note 2: Output QD is connected to input A for bi-quinary count

Function Tables

BCD Count Sequence (Note 1)

Count	Outputs			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

BCD Bi-Quinary (5-2) (Note 2)

Count	Outputs			
	Q_A	Q_D	Q_C	Q_B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

Reset/Count Function Table

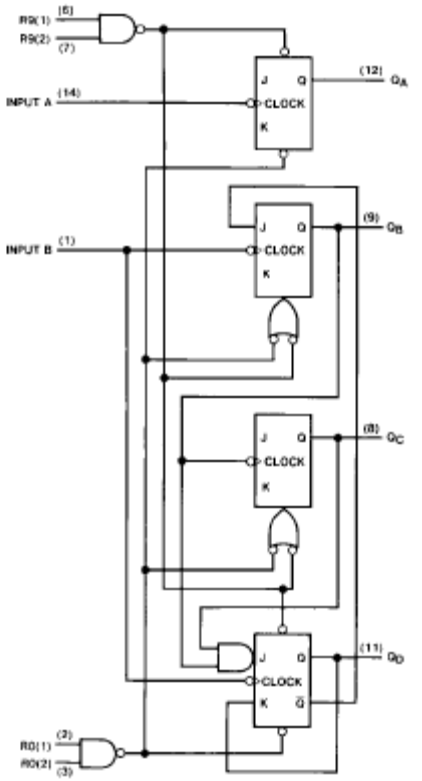
Reset Inputs				Outputs			
R0(1)	R0(2)	R9(1)	R9(2)	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

H = HIGH Level
L = LOW Level
X = Don't Care

Note 1: Output QA is connected to input B for BCD count.

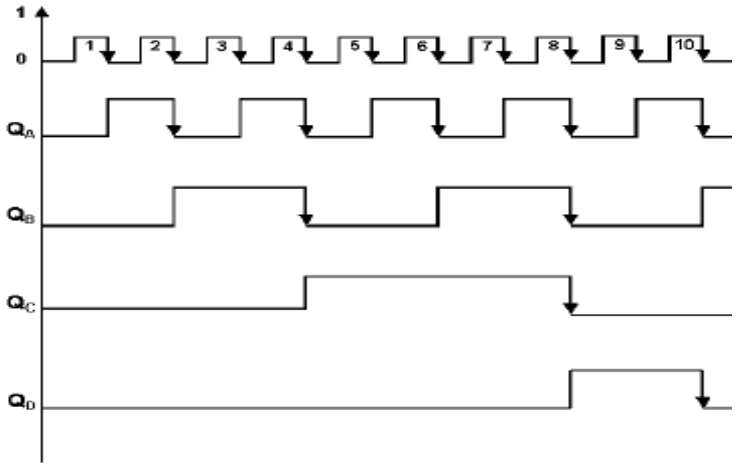
Note 2: Output QD is connected to input A for bi-quinary count

Logic Diagram



The J and K inputs shown without connection are for reference only and are functionally at a HIGH level.

EXPECTED WAVEFORM:



TRUTH TABLE:

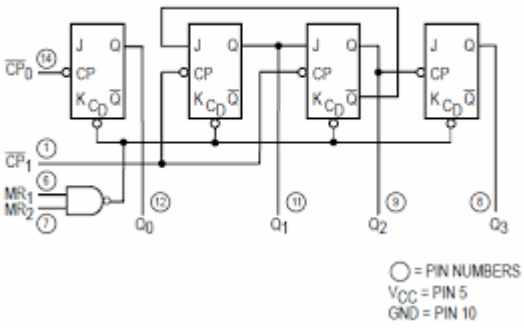
Clock pulse	Outputs			
	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

RESULT:

Thus the Mod-N Counter is constructed and verified.

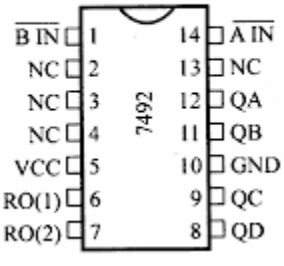
Other Counter ICs :

74LS92 (Divide by 12 Counter)



7492A、74LS92、74HC92　十二分频计数器

二分频和六分频；有复位输入。



'92A, 'LS92
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	H	L	L	L
7	H	L	L	H
8	H	L	H	L
9	H	L	H	H
10	H	H	L	L
11	H	H	L	H

'92A, 'LS92, '93A, 'LS93
RESET/COUNT FUNCTION TABLE

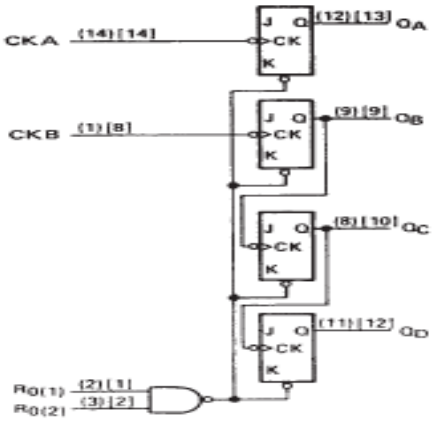
RESET INPUTS		OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

NOTES: A. Output Q_A is connected to input CKB for BCD count.
B. Output Q_D is connected to input CKA for bi-quinary count.
C. Output Q_A is connected to input CKB.
D. H = high level, L = low level, X = irrelevant

IC 74LS 93 4 Bit Binary Counter :

'93A, 'LS93

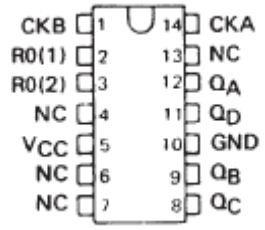
('93A) ['L93]



'92A, 'LS92, '93A, 'LS93
RESET/COUNT FUNCTION TABLE

RESET INPUTS		OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

NOTES: A. Output Q_A is connected to input CKB for BCD count.
B. Output Q_D is connected to input CKA for bi-quinary count.
C. Output Q_A is connected to input CKB.
D. H = high level, L = low level, X = irrelevant



'93A, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

STATIC DISPLAY

Aim :

1. To verify the truth table of BCD to Seven Segment decoder IC 7447.
2. To set up a static display to count from 0 to 9 using counter IC 7490.

Components Required :

IC 7490 Decade Counter.

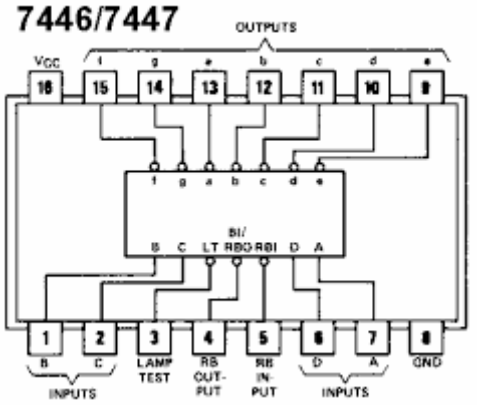
IC 7447 BCD to Seven Segment Decoder

FND 507 Seven Segment Display.

Resistors 220 Ω, ¼ W (7 Nos.).

Working:

IC 7447A features active-low outputs designed for driving common-anode LEDs or incandescent indicators directly. All of the circuits have full ripple-blanking input/output controls and a lamp test input. Display patterns for BCD input counts above nine are unique symbols to authenticate input conditions. All of the circuits incorporate automatic leading and/or trailing-edge, zero-blanking control (RBI and RBO). Lamp test (LT) of these devices may be performed at any time when the BI/RBO node is at a high logic level. All types contain an overriding blanking input (BI) which can be used to control the lamp intensity (by pulsing) or to inhibit the outputs. IC 7490 is wired as a decade counter, which counts in a straight BCD sequence from 0 to 9. The BCD output of IC 7450 is converted to seven segment display code using IC 7447. The seven LEDs of the seven segment display are so arranged to display the count in decimals from 0 to 9.



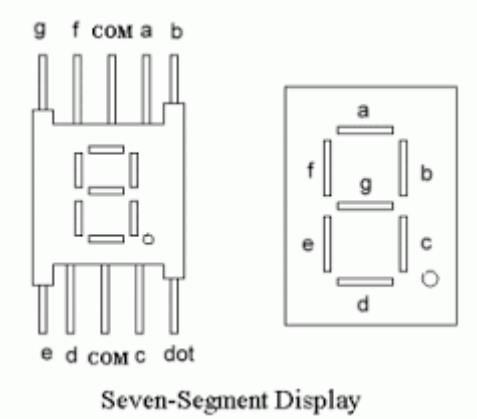
IC 7447 Pin Diagram

Function Table

46A, 47A

Decimal or Function	Inputs						BI/RBO (Note 1)	Outputs						
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L

IC 7447 Truth Table



Seven-Segment Display

Design of current limiting resistors:

Supply Voltage V_{cc} = 5 V

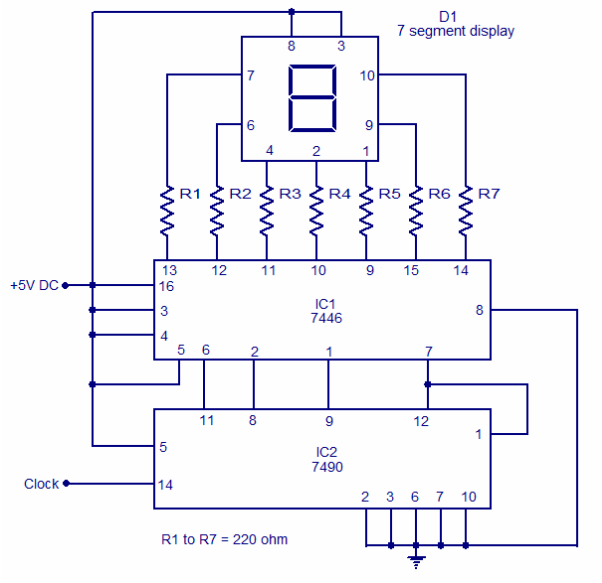
Drop across each LED, V_d = 1.5 V

Current through each LED, I_d = 15 mA.

Value of Resistance R = (V_{cc} – V_d) /I_d = (5 V– 1.5 V)/15 mA = 220Ω

Procedure :

The circuit for static display is wired properly on a breadboard and the circuit is powered using a 5 V dc power supply. A 1 Hz clock is applied to the clock input of IC 7490. The seven segment display will display counts in a straight sequence from 0 to 9.



Circuit Diagram of Static Display

Result :

The truth table of IC 7447 was verified and a circuit of static display was set up to count in a straight sequence from 0 to 9.