Digital Logic Design Laboratory Manual

DEPARTMENT OF ELECTRICAL, ELECTRONICS AND COMMUNICATION ENGINEERING
GITAM SCHOOL OF TECHNOLOGY
GITAM
(Deemed to be University)
(Estd. u/s 3 of UGC act 1956 & Accredited by NAAC with "A+" Grade)

HYDERABAD-502329

List of experiments

1. **STUDY OF LOGIC GATES AND UNIVERSAL GATES**
2. **MINIMIZATION AND REALIZATION OF GIVEN FUNCTION USING LOGIC GATES**
3. **BASIC ADDER - SUBTRACTOR CIRCUITS USING LOGIC GATES**
4. **FUNCTION REALIZATION USING DECODER**
5. **FUNCTION REALIZATION USING MULTIPLEXER**
6. **BCD –TO- SEVEN SEGMENT DISPLAY**
7. **PRIORITY ENCODER**
8. **4-BIT BINARY PARALLEL ADDER- SUBTRACTOR USING MSI GATES**
9. **DESIGN OF FLIP-FLOPS USING SSI GATES**
10. **SHIFT REGISTERS**
11. **DESIGN OF ASYNCHRONOUS COUNTERS**
12. **DESIGN OF SYNCHRONOUS COUNTERS**
13. **FOUR BIT ARITHMETIC LOGIC UNIT**
14. **SERIAL ADDER- SUBTRACTOR**
15. **MEMORY UNIT**

Among the above experiments minimum of 10 experiments have to be completed

1. STUDY OF LOGIC GATES AND UNIVERSAL GATES

AIM:

A) To verify the truth tables of logic gates AND, OR, NOT, NAND, NOR, XOR
B) To verify NAND and NOR as universal gates

APPARATUS:

S.NO.	Component	Type	Quantity
1	Quad 2 input AND gates	IC 7408	1
2	Quad 2 input OR gates	IC 7432	1
3.	Hex inverter(NOT gate)	IC 7404	1
4.	Quad 2 input NAND gates	IC 7400	1
5	Quad 2 input NOR gates	IC 7402	1
6	Quad 2 input XOR gates	IC 7486	1
7	Digital IC trainer	--	1

PROCEDURE:

PART (A):

1. Supply connections are given at the corresponding pins of ICs.
2. Each IC is taken separately and the individual gates in each IC are tested by giving inputs and the truth tables are verified.
3. Same procedure is repeated for all ICs.

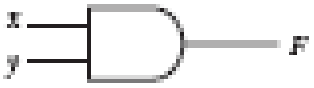
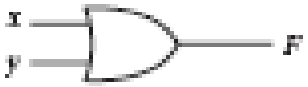
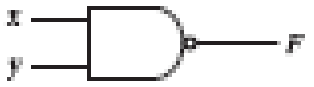
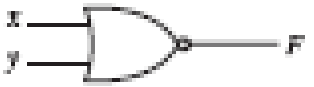
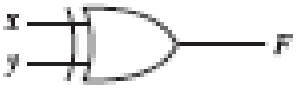
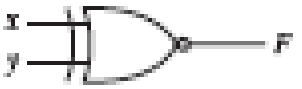
PART (B):

1. Supply connections are given at the corresponding pins of ICs.
2. For realization of individual gates using NAND gates alone, the Connections are made as per the logic diagrams.
3. Inputs are given and the truth tables of individual gates are verified.
4. The same procedure is repeated for realization of individual gates using NOR gates.

PRECAUTIONS:

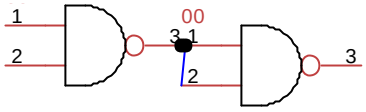
1. The power supply pins must be checked whether power is available at those pins using test probes.
2. No loose connections should be there and care must be taken to avoid shorting of pins.

DIGITAL LOGIC GATES:

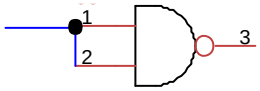
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

REALIZATION OF BASIC GATES OPERATIONS USING NAND GATES:

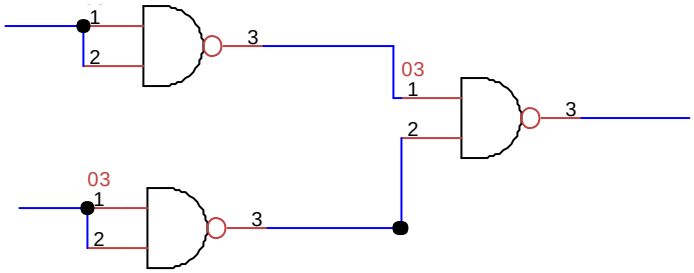
1. AND Operation:



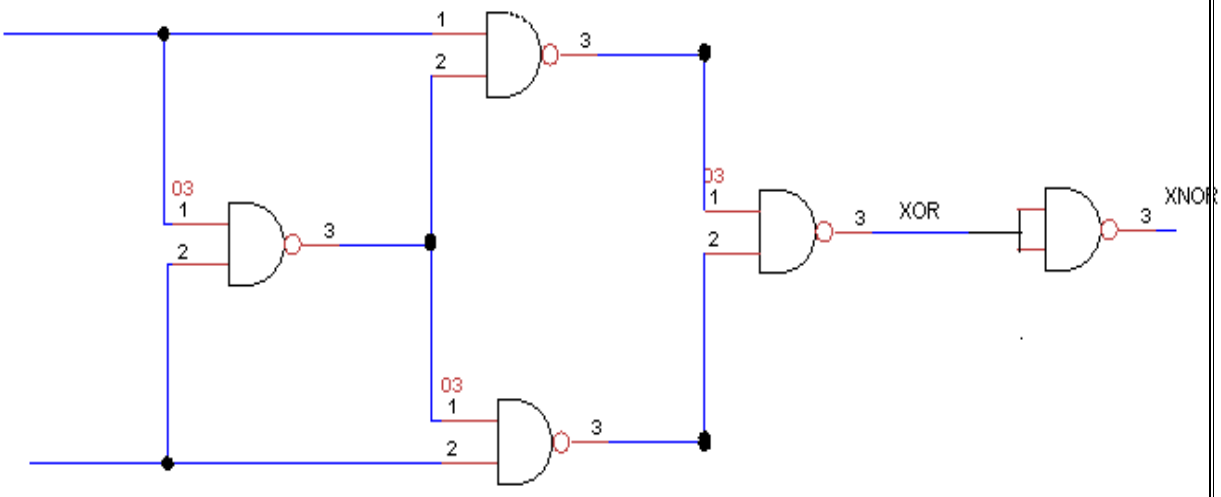
2. NOT Operation:



3. OR Operation:

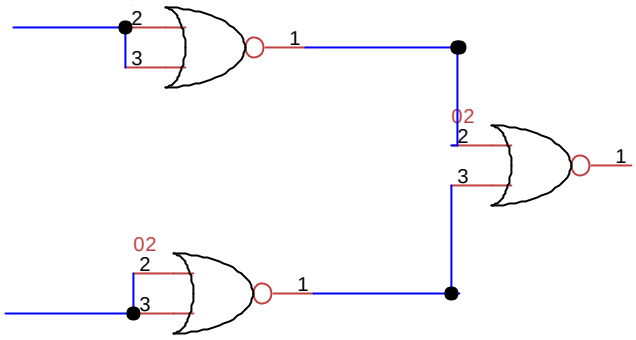


4. XOR & XNOR Operation:

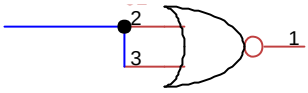


REALIZATION OF BASIC GATES OPERATIONS USING NOR GATE:

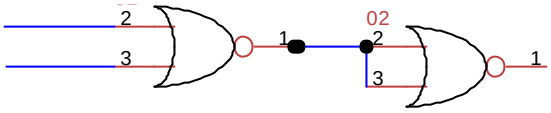
1. AND Operation:



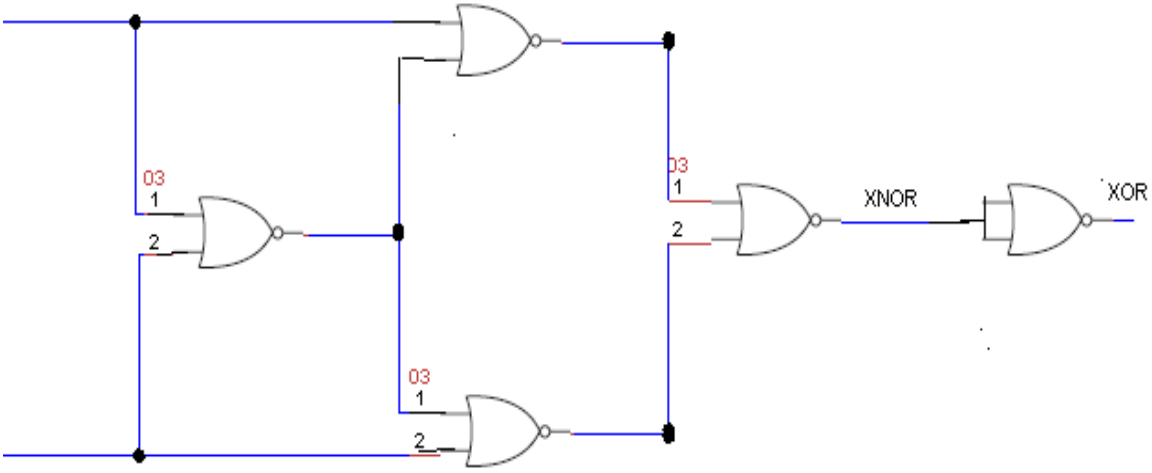
2. NOT Operation:



3. OR Operation:



4. XNOR & XOR Operation:



RESULT:

The truth tables of individual gates are verified and their realizations using NAND gates alone and NOR gates alone have been verified.

SAMPLE QUESTIONS:

1. If one of the inputs of an EX-OR gate is *high*, its output will be--------
   (same as other input/inverse of other input).
2. To INHIBIT(or DISABLE) an AND gate one of its input is connected to logic level-----(0/1).
3. To DISABLE a NOR gate one of its inputs needs to be connected
   to logic level----------(0/1)
4. The number of rows in a truth table of 4 variables are----------.
5. A 3 input NOR gate is required to detect the simultaneous occurrence of all the inputs in the LOW state. Its output is----------(active low /active-high).
6. The minimum number of bits required to distinguish 108 distinct objects is--------.
7. What is the difference between a positive logic system and negative logic system?
8. What are universal gates? Why that name?
9. Minimum number of NAND gates necessary to realize EX-OR gate using NAND gates only is --------.
10. Minimum number of NOR gates necessary to realize EX-OR gate using NOR gates only is --------.

2. MINIMIZATION AND REALIZATION OF GIVEN FUNCTION USING LOGIC GATES

AIM:

A) To implement a given Boolean function using basic gates after minimization

$F(W,X,Y,Z) = \Sigma(0,2,4,5,6)$

B) To implement a given Boolean function using two level NAND gates alone

C) To implement a given Boolean function using two level NOR gates alone

APPARATUS:

S.NO.	Component	Type	Quantity
1	Quad 2 input NAND gates	IC 7400	1
2.	Quad 2 input NOR gates	IC 7402	1
3	Quad 2 input AND gates	IC 7408	1
4	Quad 2 input OR gates	IC 7432	1
5.	Hex Inverter	IC 7404	1
3.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. The given Boolean function is simplified first using K- map method.
2. The simplified expression is realized using AND, OR, NOT gates.
3. The 4 input variables are connected to 4 toggle switches and the output of the circuit is connected to one of the output indicators.
4. The inputs are varied from 0000 to 1111 and the truth table of the given function is verified.

PART (B):

1. The given Boolean function is simplified first using K- map method and expressed in SOP form.
2. The simplified expression is realized using NAND gates alone and connected as shown in the logic diagram. Individual gates in the IC are checked for proper operation.
3. The 4 input variables are connected to 4 toggle switches and the output of the circuit is connected to one of the output indicators.
4. The input are varied from 0000 to 1111 and the truth table of the given function is verified.

PART (C):

1. The given Boolean function is simplified first using K- map method and expressed in POS form.
2. The simplified expression is realized using NOR gates alone and connected

as shown in the logic diagram.

3. The 4 input variables are connected to 4 toggle switches and the output of the circuit is connected to one of the output indicators.
4. The input are varied from 0000 to 1111 and the truth table of the given function is verified.
5. In case, the desired output is not found, tracing the logic signal values from input side to output or else output side to input should be checked and thus the problem is fixed.

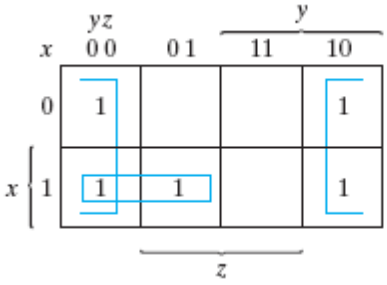
PRECAUTIONS:

1. Ensure that individual gates are properly working before connections are made.
2. care must be taken to avoid shorting of individual pins of ICs and loose connections.

TRUTH TABLE:

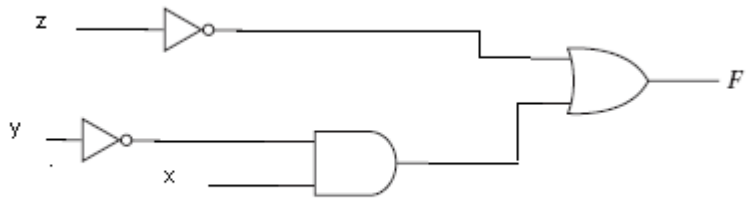
INPUTS			OUTPUT
X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

3- Variable K-map method of minimization:

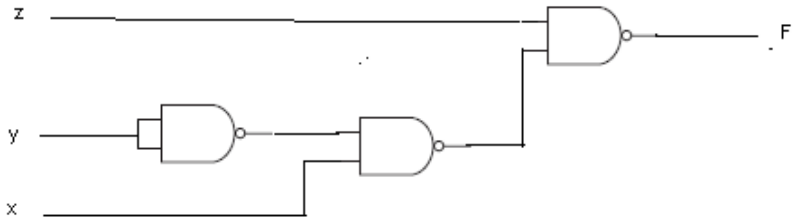


$F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$

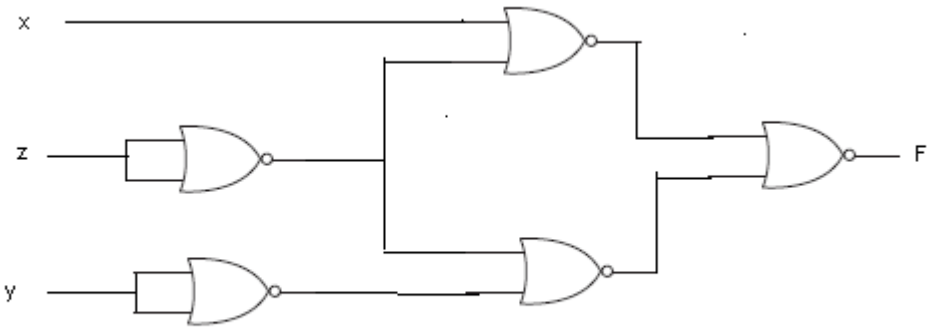
A) Using Basic Gates:



B) Using NAND Gates alone:



C) Using NOR Gates alone:



F= (x+z') (y'+ z')

RESULT:

The given Boolean function is simplified using K- map and then realized using Basic gates, NAND gates alone, NOR gates alone.

SAMPLE QUESTIONS:

1. A logic variable in complemented or uncomplemented form is known as a ----------.
2. A NOR-NOR realization is a ----------level realization.
3. A 1 in a cell of K-map can be combined with three other 1's in only one combination. The resulting term of these four 1's is----------.
4. What are the advantages of K-map method of minimization over algebraic manipulation?
5. What is algebraic manipulation?
6. What are minterms and maxterms ?
7. What are canonical forms and standard forms? State the difference between them if any?
8. What is the difference between two level implementation and multi-level implementation? Which is more advantageous than the other and why?
9. In how many gate levels standard form expressions can be implemented?
10. Which code is used for labeling the cells of K-map? Why?

3. BASIC ADDER - SUBTRACTOR CIRCUITS USING LOGIC GATES

AIM:

To construct and verify the truth tables of

A) Half- adder circuit

B) Half- subtractor circuit

C) Full- adder circuit constructed using half-adders and basic gates

D) Full- subtractor circuit constructed using half-subtractions and basic gates.

APPARATUS:

S.NO.	Component	Type	Quantity
1	Quad 2 input AND gates	IC 7408	1
2.	Quad 2 input OR gates	IC 7432	1
3.	Quad 2 input XOR gates	IC 7486	1
4.	Hex Inverter	IC 7404	1
5.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Supply connections are given to corresponding pins of ICs.
2. Connections are made as per the circuit diagram (i)
3. The two input variables x, y(addened bit & augend bit) are connected to two toggle switches and out variables S,C (Sum and Carry) are connected to two output indicators.
4. For various possible combination of inputs, operation is verified.

PART (B):

1. Connections are made as per the circuit diagram (ii)
2. The two input variables x, y (minuend bit & subtrahend bit) are connected to two toggle switches and out variables D, B(Difference and Borrow)are connected to two output indicators.
3. For various possible combination of inputs, operation of the circuit is verified.

PART (C):

1. Connections are made as per the circuit diagram (iii).
2. The three input variables x, y, z (addend bit, augend bit, input carry) are connected to three toggle switches and out variables S, C(Sum and Carry) are connected to two output indicators.
3. For various possible combination of inputs, truth table is verified.

PART (D):

1. Connections are made as per the circuit diagram (iv)

2. The three input variables x, y, z (minuend bit, subtrahend bit, previous borrow) are connected to three toggle switches and out variables D, B(Difference and Borrow)are connected to two output indicators.
3. For various possible combinations of inputs, operation of the circuit is verified.

PRECAUTIONS:

1. Ensure that individual gates are properly working before connections are made.
2. Care must be taken to avoid shorting of individual pins of ICs and loose connections.

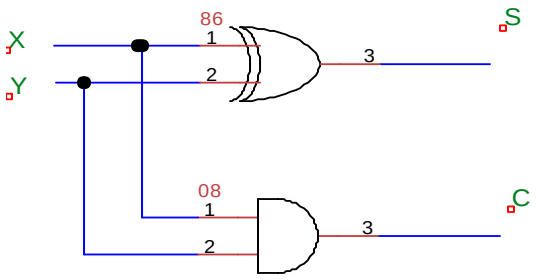
HALF ADDER:

TRUTH TABLE:

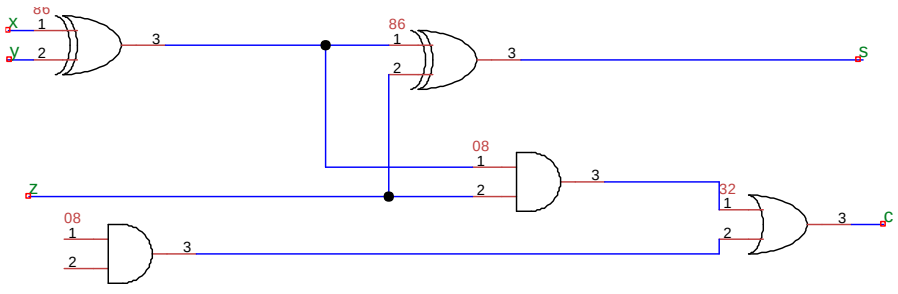
Inputs		Outputs	
X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S= x'y+xy' = x ⊕ y, C = xy

CIRCUIT DIAGRAM:



FULL ADDER:



TRUTH TABLE

Inputs			Outputs	
X	Y	Z	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

$S = x'y'z + x'yz' + xy'z' + xyz = x \oplus y \oplus z$

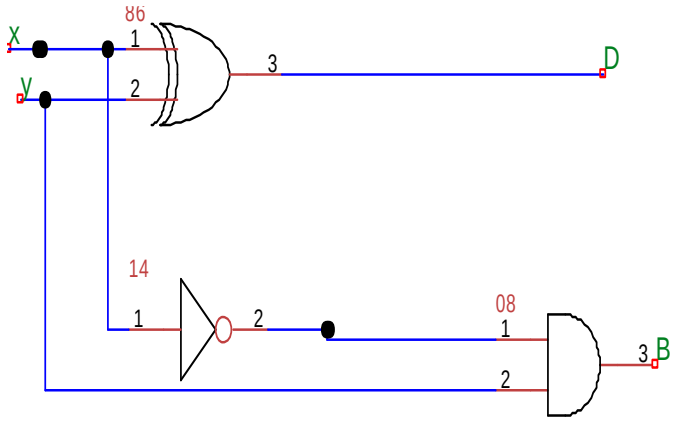
C = xy + yz + zx and C can be derived to be equal to $(x \oplus y)\, z + xy$

HALF SUBTRACTOR

TRUTH TABLE:

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

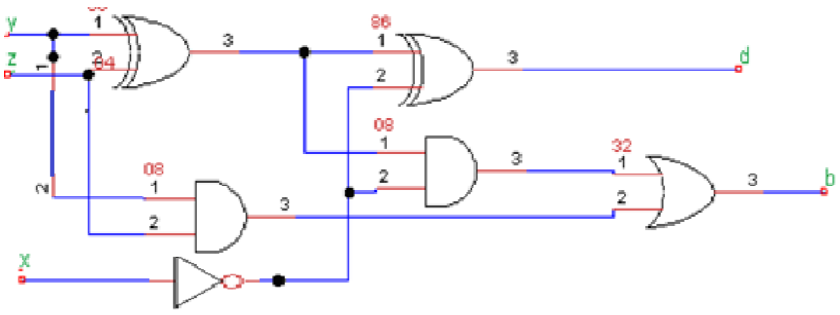
CIRCUIT DIAGRAM:



$D = x'y + xy' = x \oplus y$

$B = x'y$

FULL SUBTRACTOR:



Inputs			Outputs	
X	Y	Z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

D = x'y'z + x'yz '+ xy'z' + xyz = x $\oplus$ y $\oplus$ z

B = x'y + x'z + yz and C can be derived to be equal to $(y \oplus z)\, x' + yz$

RESULT:

The Half- adder, Half- subtractor, Full- adder and Full- subtractor circuits are constructed using basic gates and truth tables are verified.

SAMPLE QUESTIONS:

1. What is half adder?
2. What is full adder?
3. How will you implement a full adder using half adders?
4. What is a truth table?
5. How many outputs are there for a full adder? Why?
6. What is the difference between signed and unsigned binary numbers?
7. How many forms are available to represent a negative number in a digital system?
8. How will you get 2's complement of a binary number?
9. 2's complement of a 2's complement is----------. Show with an example.

10. The process of subtraction gets converted into that of addition by using----------

4. FUNCTION REALIZATION USING DECODER

AIM:

A) To verify the operation of binary to Octal decoder IC 74138 and realize 4 to 16 line decoder using 3 to 8 decoders.

B) To implement a Full adder circuit using decoder and basic gates

APPARATUS:

S.NO.	Component	Type	Quantity
1	3 to 8 line decoder	IC 74138	2
2	Quad 2 input OR gates	IC 7432	2
3.	Hex inverter	IC 7404	2
4.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Power supply connections are made at pins 8(GND) and 16(Vcc) of 74138 IC.
2. To enable the chip operation, the corresponding enable pins are given signals. E_1', E_2', E_3' are connected to LOW, LOW and HIGH respectively.
3. To verify the operation of binary to Octal decoder, three input pins A_2, A_1, A_0 are connected to input toggle switches and eight outputs O_0', O_1', O_2',.. O_7', are connected to output indicators and the truth table is verified.
4. To extend this 3 to 8 line decoder into a 4 to 16 line decoder, two separate 74138 ICs are taken and connections are made as per the circuit diagram (i) and the truth table is verified.

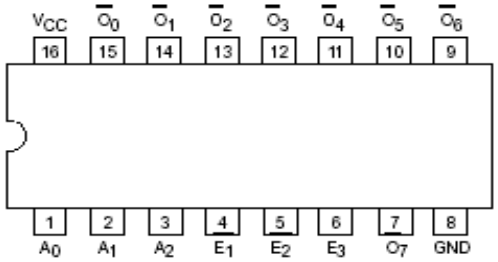
PART (B):

1. Power supply connections are made at pins 8(GND) and 16(V_{cc}) of 74138 IC.
2. To enable the chip operation, the corresponding enable pins are given signals. E_1', E_2', E_3 are connected to LOW, LOW and HIGH respectively.
3. Circuit connections are made as per circuit diagram (ii) and operation of full adder is verified.

PRECAUTIONS:

1. The corresponding chip enable signals must be enabled for normal operation.
2. Care must be taken to ensure that there are no shorting of pins
3. It is always good practice to ensure the correct operation of individual gates which are used as part of the circuit.

74138 3x8 Decoder



PIN NAMES

A_0–A_2	Address Inputs
E_1, E_2	Enable (Active LOW) Inputs
E_3	Enable (Active HIGH) Input
O_0–O_7	Active LOW Outputs (Note b)

TRUTH TABLE

INPUTS						OUTPUTS							
E_1	E_2	E_3	A_0	A_1	A_2	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

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

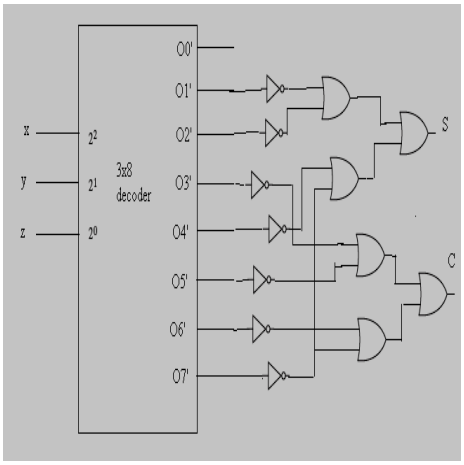
Full adder circuit using Decoder:

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

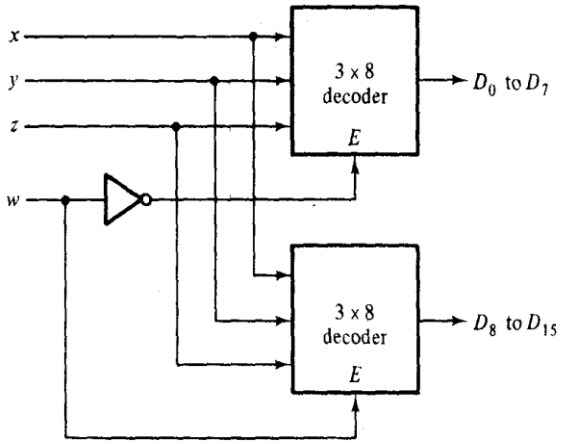
$C = \sum (3,5,6,7)$

CIRCUIT DIAGRAM:

Inputs			Outputs	
X	Y	Z	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



i) Full adder implementation using 3 x 8 decoder and gates



ii) 4x 16 Decoder implementation using Two 3x8 decoders

RESULT: The operation of 3 to 8 decoder is verified and it is extended as 4 to 16 Decoder and a full adder circuit is realized using 3 to 8 decoder.

SAMPLE QUESTIONS:

1. What is a decoder? How to specify its size?
2. What is a demultiplexer? How a decoder with enable input functions like demultiplexer?
3. Design a 2 to 4 line decoder with enable input using basic gates.
4. Is there any similarity between decoder and code converters?
5. Give some practical applications of decoder circuits?
6. A decoder with 64 output lines has ----------data inputs.
7. What is the function of enable pin in any standard IC?
8. How many minterms a 3x8 decoder generates?
9. Show experimentally implementation of
   (i). Half adder
   (ii). Full subtractor circuit using 3x8 decoder and the necessary gates.

5. FUNCTION REALIZATION USING MULTIPLEXER

AIM:

A) To verify the 8 to 1 multiplexer operation using MSI gates.

B) To implement a 4 bit odd parity checker circuit using multiplexer.

APPARATUS:

S.NO.	Component	Type	Quantity
1	8 to1 multiplexer	IC 74151	1
2.	Hex inverter	IC 7404	1
3.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Suitable power supply connections and chip enable signals are given at the corresponding pins of the MUX IC.
2. To verify the operation of 74151 IC as multiplexer, input pins (I_0, I_1, I_2…I_7) are connected to 8 input toggle switches and the out put pins Z and Z' are connected to output indicators.
3. The selection lines pins (S_2, S_1, S_0) are connected to three input toggle switches.
4. The selection lines are varied from binary 000 to 111 and we will observe the data at the corresponding inputs (0 or 1) will find its way to the output.

PART (B):

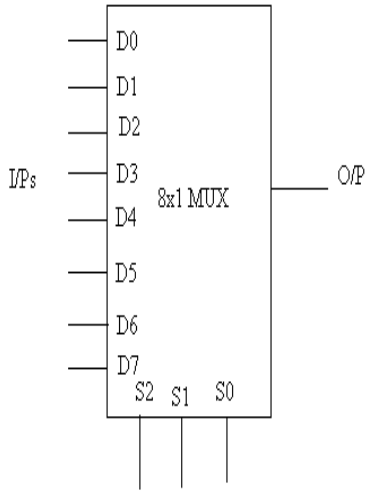
1. First, identify the size of MUX needed for implementing the given circuit using MUX.
2. Complete the procedure for implementing any Boolean function using MUX. Three input variables of the function are connected to selection lines pins And the fourth input variable is given as input to the MUX as per the table.
3. The corresponding inputs from the derived table are given to the MUX IC and the output pins are connected to output indicators.
4. The 4 input variables are varied from 0000 to 1111 and the truth table for the given circuit operation is verified.

PRECAUTIONS:

1. Suitable signals must be given to enable pins of IC in order to enable the chip.
2. The order in which we are connecting the input variables to selection

pins must be proper.

Logic Symbol & Function table for 8x1 MUX:



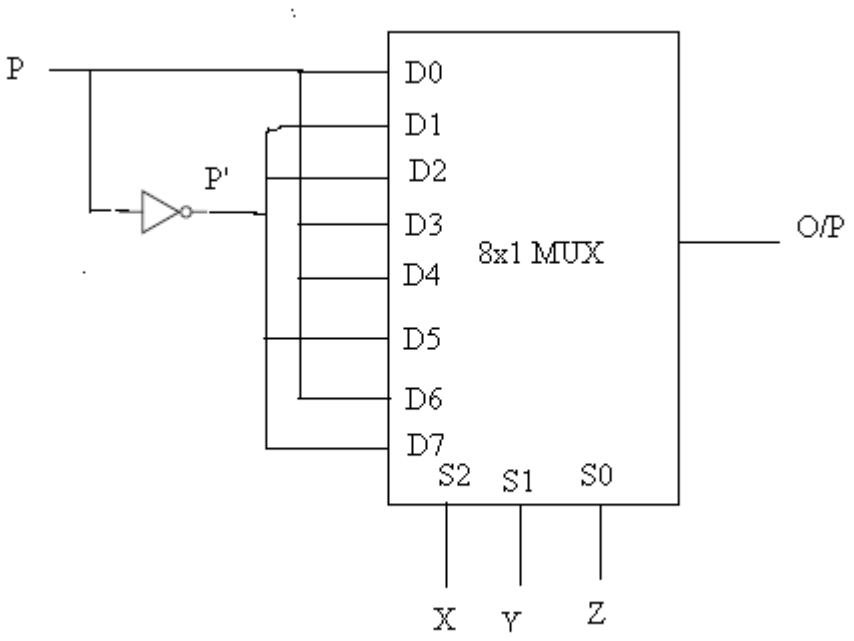
Chip Enable(E)	Selection lines S2	S1	S0	Output
1	x	x	x	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

4- bit PARITY CHECKER circuit implementation using MUX:

TRUTH TABLE:

Inputs				Output C	
X	Y	Z	P		
0	0	0	0	0	
0	0	0	1	1	C= P
0	0	1	0	1	
0	0	1	1	0	C= P′
0	1	0	0	1	
0	1	0	1	0	C= P′
0	1	1	0	0	
0	1	1	1	1	C= P
1	0	0	0	1	
1	0	0	1	0	C= P′
1	0	1	0	0	
1	0	1	1	1	C= P
1	1	0	0	0	
1	1	0	1	1	C= P
1	1	1	0	1	
1	1	1	1	0	C= P′

CIRCUIT DIAGRAM:



RESULT:

The Muliplexer operation is verified and 4- bit even parity checker circuit has been realized using multiplexer.

SAMPLE QUESTIONS:

1. What is a multiplexer? Why we call it as data selector?
2. How will you specify the size of a given MUX?
3. Give some practical applications of multiplexers.
4. Realize basic gates using a Multiplexer of suitable size.
5. Design a BCD – to- Gray code converter using Muliplexers.
6. How will you implement a multiplerxer with decoders?
7. Construct a 4x1 multiplexer from no. of 2x1 multiplexers only

6. BCD –TO- SEVEN SEGMENT DISPLAY

AIM:

A) To display any one of the decimal digits 0 through 9 using a Seven- segment indicator.
B) To display any one of the decimal digits 00 through 99 using Seven- segment indicators.

APPARATUS:

S.NO.	Component	Type	Quantity
1	BCD-to- Seven segment decoder	IC 7447	2
2.	Seven segment display devices(common anode)	FND 507/ A5611	2
3.	Resistors	470 Ohms	2
4.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Power supply connections are given to the corresponding pins of IC 7447.
2. The LT′,RBI′,RBO′ pins of the 7447 decoder are made HIGH .
3. Four pins A_3, A_2, A_1, A_0 are connected to four toggle switches as BCD inputs.
4. If the given seven segment display device is common anode type, the outputs a′, b′,c′,d′,e′,f′,g′ of the decoder(7447 IC) are connected directly to the corresponding seven segments a, b,c,d,e,f,g of the seven segment display device. If it is common cathode type, outputs of the decoder are connected through inveters to the display device. The common terminal of the device is connected through a limiting resistor to V_{cc} or GND, depending on the type of display device.
5. Inputs are varied from 0000 to 1001 and the equivalent decimal digits are displayed.

PART (B):

1. Two separate 7447 ICs and display devices are taken and above steps are repeated for separately.
2. RBI of LSB is connected to RBO of MSB and then to HIGH.
3. For any Random combination of inputs, decimal digits 00 to 99 are observed on display devices.

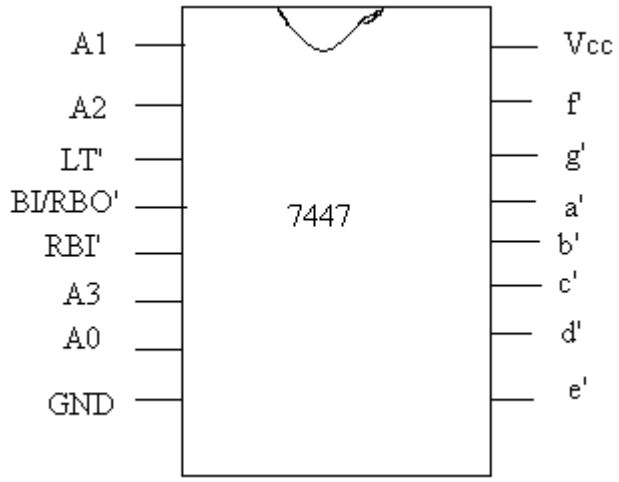
PRECAUTIONS:

1. Before making connections, verify whether all the segments in the given Seven

segment display devices are glowing or not by connecting V_{cc} to the common anode and GND to the segment to be tested.

2. For normal operation of 7447 ICs, the corresponding pins LT′ must be 1, RBI′ can be 0 or 1, RBO′ to be 1.

PIN diagram of 7447:



A0-A3 : BCD inputs

RBI'- Ripple blanking i/p

LT'- Lame test i/p

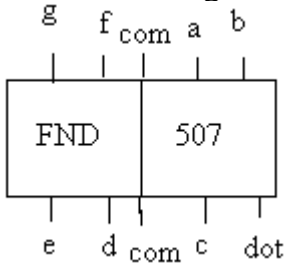
BI/RBO'-Blanking i/p / Ripple blanking o/p

a'-g' : 7 segment o/ps

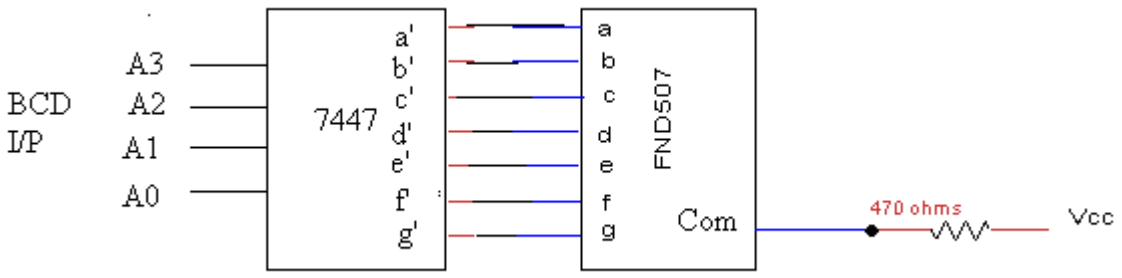
Summary of BCD- to- seven segment decoder functions:

LT	RBI	BI/RBO	BCD inputs	Display mode
0	X	1 output	X	Lamp test
X	X	0 input	X	Display blank
1	1	1 output	Any number	Normal decoding
1	0	Normally at logic 1 output. Goes to logic 0 during blanking interval	Any number	Normal decoding with zero blanking

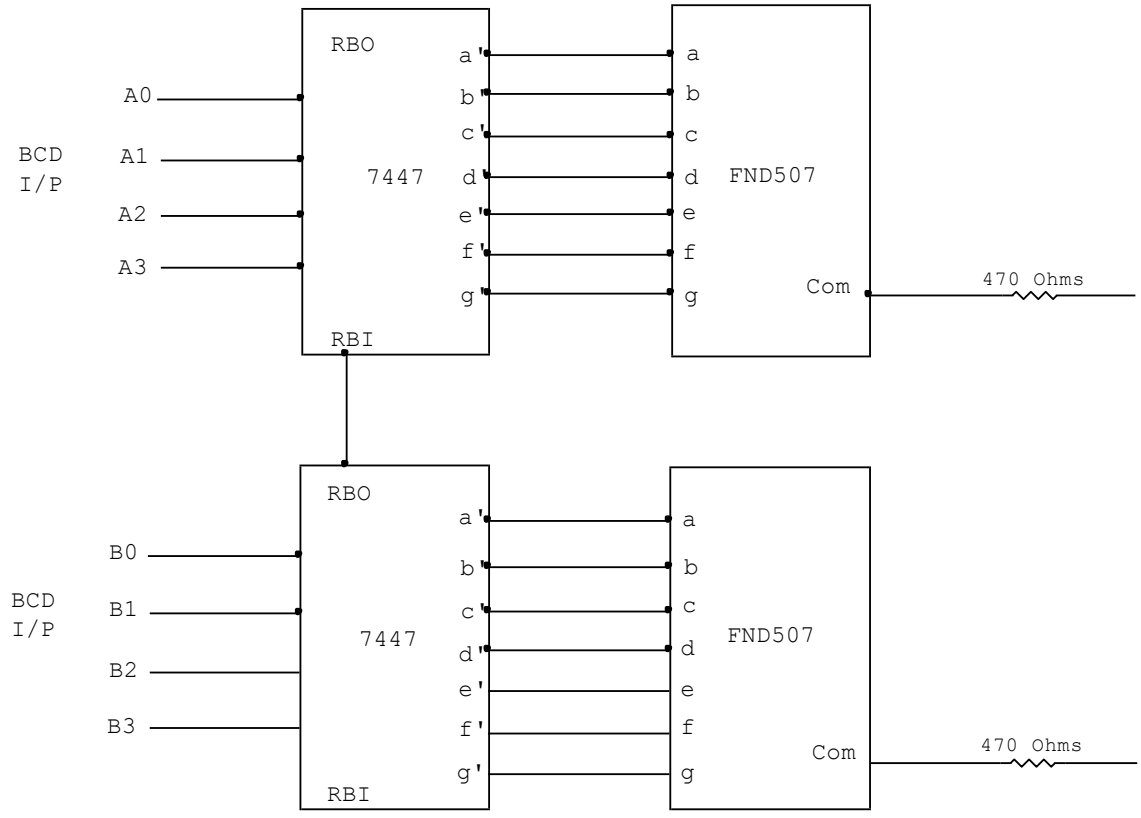
FND 507 PIN diagram:



BCD TO SEVEN SEGMENT DISPLAY: (Displaying 0 -9)



BCD TO SEVEN SEGMENT DISPLAY: (Displaying 00 -99)



RESULT:

The decimal digits from 00 to 99 are displayed using two seven segment display devices and two7447 BCD- to- seven segment decoders.

SAMPLE QUESTIONS:

1. Write some common applications of seven- segment display devices.
2. What is the function of $\overline{LT}$, $\overline{RBI}$, $\overline{RBO}$ pins?
3. What is the relation between the outputs of the decoder and the type of display device chosen? Which is better?
4. What is the function of the resistor in the above circuits?
5. What precautions we must take as the significant digits to be displayed increases?
6. Write some commonly available BCD- to- seven segment decoder/ driver ICs.
7. Draw the logic diagram to display  with minimum circuitry.

7. PRIORITY ENCODER

AIM:

A) To verify Octal - to- Binary encoder operation using 74148 Encoder

B) To design and verify hexadecimal - to – binary encoder using 74148 encoders and 74157 multiplexer

APPARATUS:

S.NO.	Component	Type	Quantity
1	74148	Octal- to- binary priority encoder	2
2.	74151	Quad 2:1 MUX	1
3.			
4.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Power supply connections are made at the corresponding pins of IC 74148.
2. Eight Octal input pins i.e. 0, 1, 2, 3, ….7 (Active- LOW) are connected to eight toggle switches and the chip enable pin EI (Active- LOW) is also connected to one more input toggle switch.
3. The pins A_2, A_1, A_o, E_o, G_S (Active- LOW) are connected as outputs to the five output indicators.
4. The inputs are varied and the truth table of Octal-to- binary encoder operation is verified.

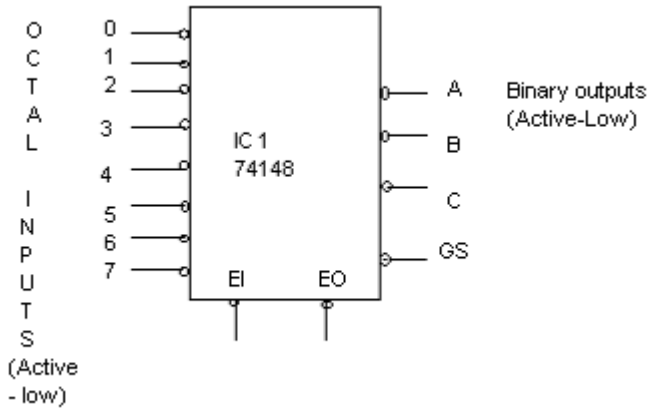
PART (B):

1. Since there are 16 symbols (0-F) in hexadecimal number system, two 74148 encoders are required. Two separate 74148 IC are taken and power supply connections are made.
2. Hexadecimal inputs 0 through 7 are applied to IC_1 input lines and inputs 8 through F to IC_2 input lines as above.
3. Whenever one of the inputs of IC_2 is active (LOW), IC_1 must be disabled, on other hand, if all the inputs of IC_2 are HIGH then IC_1 must be enabled. This is achieved by connecting the E_o line of IC_2 to the E_I line of IC_1.
4. A quad 2:1 MUX is required to get the proper 4- bit outputs. The complete circuit is shown in fig and connections are made as per the circuit diagram.
5. The G_s output of 74148 goes LOW whenever one of its inputs is active. Therefore, G_s of IC_2 is connected to select input of 74157, which selects A inputs if it is LOW, otherwise B inputs are selected. The outputs of the multiplexer are the required binary outputs and are active- LOW.

PRECAUTIONS:

1. While verifying the operations, it should be remembered that the inputs and outputs are all Active- LOW to avoid confusion.
2. For Active LOW inputs and outputs, B_3 is HIGH, A_3 is GND, Strobe input of MUX is GND.

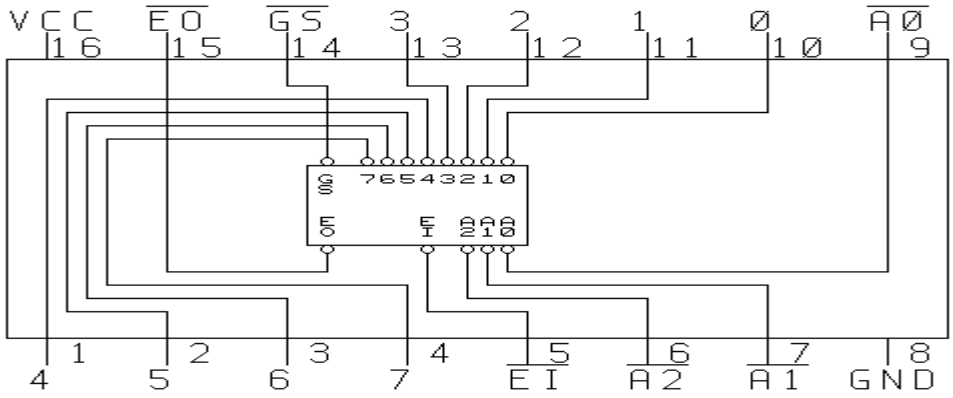
Octal- to- binary priority encoder Circuit diagram:



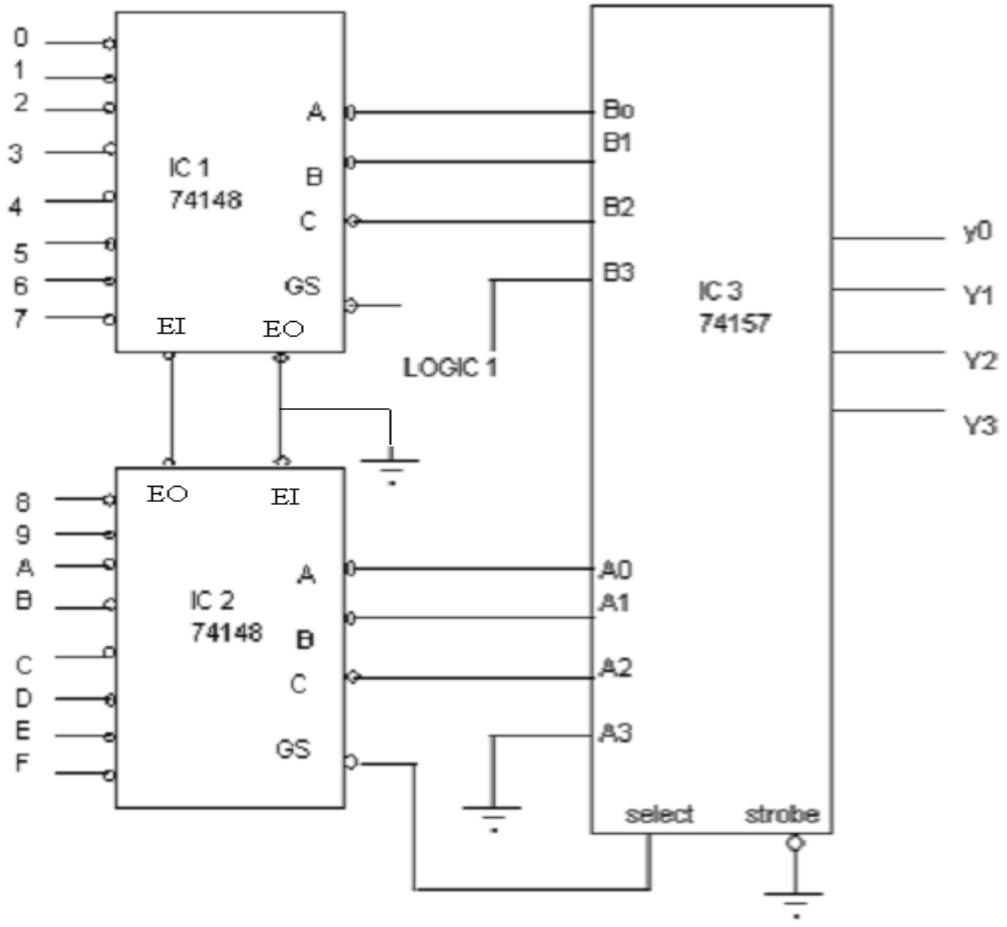
TRUTH TABLE OF Octal- to- binary priority encoder:

INPUTS									OUTPUTS				
E_I	0	1	2	3	4	5	6	7	A_2	A_1	A_o	G_S	E_o
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

74148 Pin diagram :



Hexadecimal- to- binary priority encoder:
Circuit diagram:



RESULT:

The operations of Octal- to- Binary and Hexadecimal- to- binary encoder operations are verified using 74148 ICs.

SAMPLE QUESTIONS:

1. What is the significance of E_I, E_O, G_S pins of 74148 IC?
2. Show the truth table for Decimal- to- BCD encoder (Active- LOW) operation?
3. State some practical applications of priority encoders?
4. What is valid bit indicator in priority indicators?
5. What are the two differences between encoder and priority encoder?
6. What is the difference between encoders and decoders?

8.4-BIT BINARY PARALLEL ADDER- SUBTRACTOR USING MSI GATES

AIM:

(A) To construct and test 4- bit binary parallel adder circuit and modify it into a 4- bit subtractor circuit

(B) To modify the 4- bit subtractor circuit into a magnitude comparator circuit that compares the magnitudes of two numbers

APPARATUS:

S.NO.	Component	Type	Quantity
1	4- bit binary parallel adder	IC 7483	1
2.	Quad 2 input XOR gates	IC 7400	1
3.	Hex inverter	IC 7404	1
4.	Quad 2 input AND gates	IC 7408	1
5.	Digital trainer	-	1

PROCEDURE:

PART (A):

1. Power supply connections are given to the corresponding pins of 7483 IC.
2. Four toggle switches are connected as Data inputs A to the A_3, A_2, A_1, A_0 pins of 7483.
3. B_3, B_2, B_1,B_o pins are connected as B Data inputs of 7483 to another four switches through four XOR gates .
4. External input Mode select (M) is connected as one input of all XOR gates and also connected to carry input of parallel adder. Another input of all XOR gates will be B data inputs. The sum bits S_3, S_2, S_1, S_0 and C_4 gives output and connect them to output indicators.
5. To perform Binary addition operation, Keep M=0 so that Cin=0, and give some inputs to A and B and verify the addition operation. Show that during addition, the output carry is equal to 1 when the sum exceeds 15.
6. When the mode select M is equal to 1, the input carry is equal to 1 and the sum output is A plus the 2's complement of B. show that when A>= B, the subtraction operation gives the correct answer, A-B, and the output carry c4 is equal to 1. But when A<B, the subtraction gives the 2's complement of B - A and the output carry is equal to 0.

PART (B):

1. Two numbers A and B can be compared by first performing subtraction operation as above.
2. If the output in S is equal to zero, we know that A=B. The output carry from C_4 determines the relative magnitude: when C_4 =1, we have A>=B; when C_4 =0, we have A<B; and when C_4 =1 and S not equal to 0, we have A>B.

3. To provide the comparison logic, it is necessary to supplement the subtractor circuit with a combinational circuit that has five inputs, S1 through S4 and C_4, and three outputs x, y, z so that

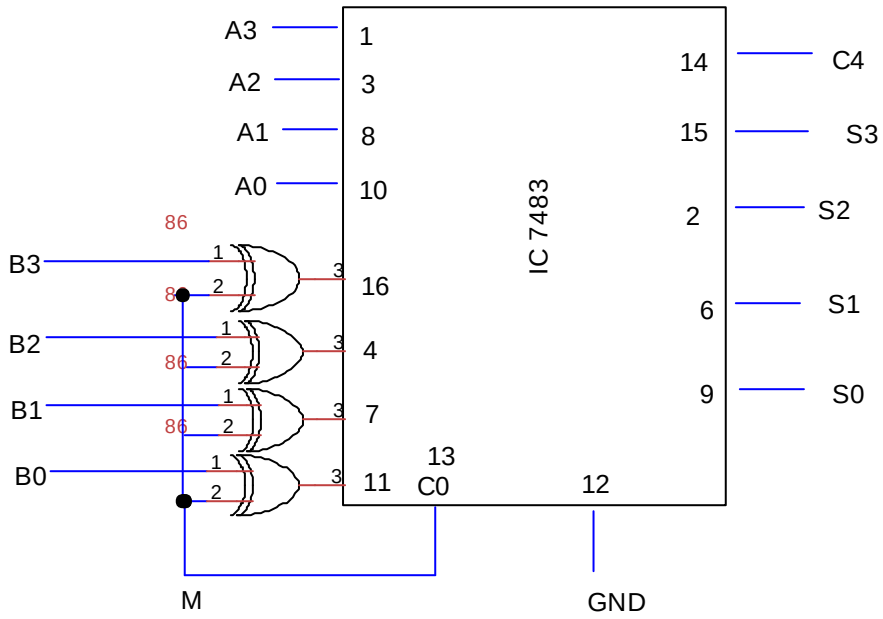
x = 1	if A=B	(S = 0000)
y = 1	if A<B	(C_4 = 0)
z = 1	if A>B	(C_4 =1 and S ≠ 0000)

Modify the Subtractor circuit as per the diagram and test its operation for different sets of A and B and check the outputs x, y, z.

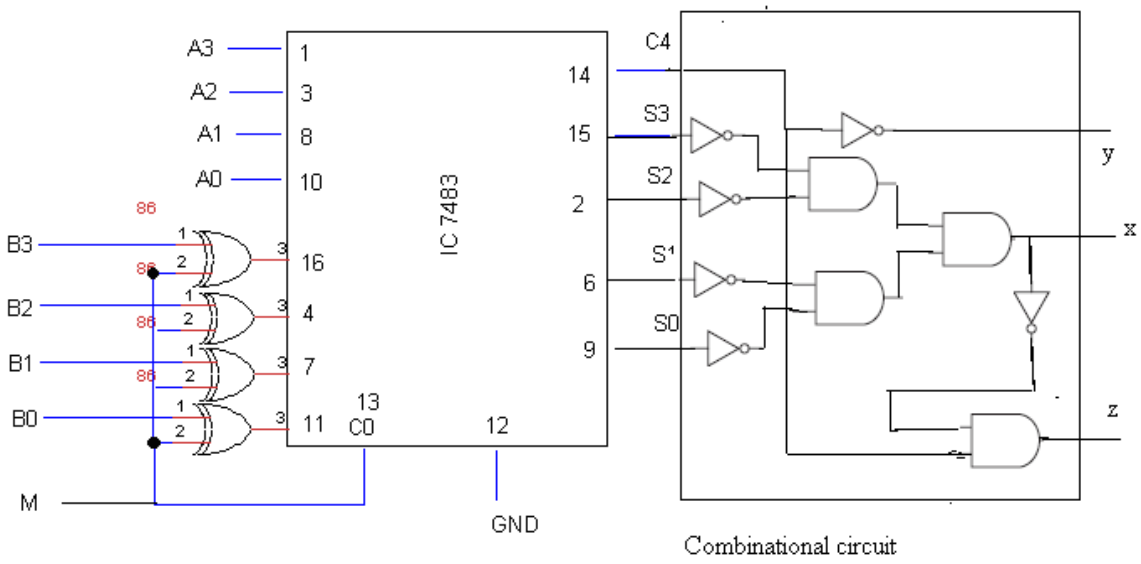
PRECAUTIONS:

1. Check the individual gates used in the circuit before connecting for ensuring proper operation.
2. Ensure that there are no shorting of pins.

4-BIT ADDER - SUBTRACTOR



Magnitude – Comparator:



RESULT:

The 4- bit binary parallel addition and subtraction operations are verified and modifying the same circuit magnitude comparison is also done.

SAMPLE QUESTIONS:

1. What are the various ways of representing negative numbers in Binary number system?
2. What is Ripple carry adder and what is its limitation? How will you overcome it?
3. What is the Carry propagation delay of an n-bit parallel adder?
4. Design a BCD adder circuit?
5. Describe the connections involved in obtaining i) A + B +1 i.e.add with carry
   ii) A – B - 1 i.e.subtract with borrow.

9. DESIGN OF FLIP-FLOPS USING SSI GATES

AIM: To construct and verify the operation of following latches and flip-flops.

a) SR latch using cross coupled NOR gates
b) S'R' latch using cross coupled NAND gates
c) Clocked SR flip-flop
d) D flip -flop
e) JK flip- flop
f) JK master – slave flip-flop
g) T- flip flop

APPARATUS:

S.NO.	Component	Type	Quantity
1	Quad 2 input NAND gate	IC 7400	2
2.	Quad 2 input NOR gate	IC 7402	
3.	Hex inverter	IC 7404	1
4.	Digital trainer	-	1

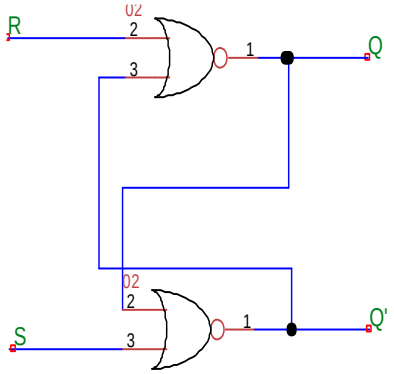
PROCEDURE:

1. Connect the circuits as shown in the figures for all latches and flip-flops.
2. For latch operation verification, give external inputs from toggle switches and verify the characteristic tables.
3. For verification of Flip-flops, give clock pulse from the clock pulse generator 1Hz on the digital trainer or give pulse inputs from the pulser on the same and verify the characteristic tables by varying the external inputs.

PRECAUTIONS:

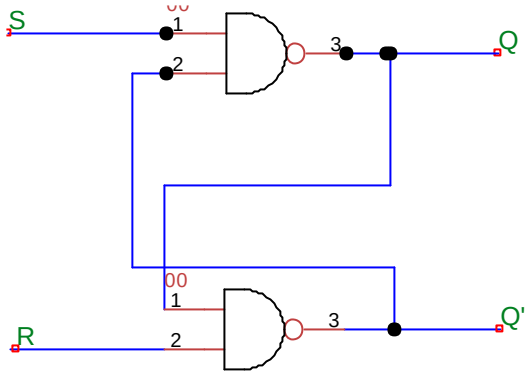
1. Ensure that the individual gates you are using are functioning properly before making connections.
2. Care must be taken to prevent loose connections and shorting of pins.

SR – LATCH:



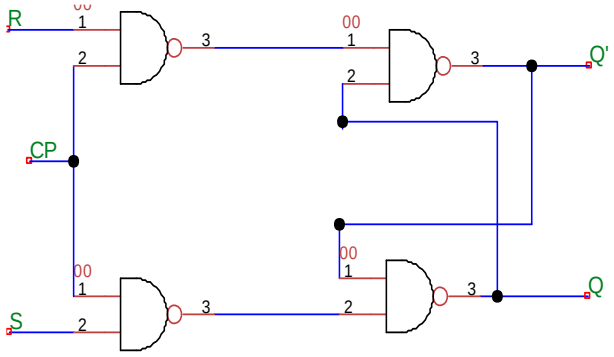
S	R	Q	Q’
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

S’R’ LATCH:



S	R	Q	Q’
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1

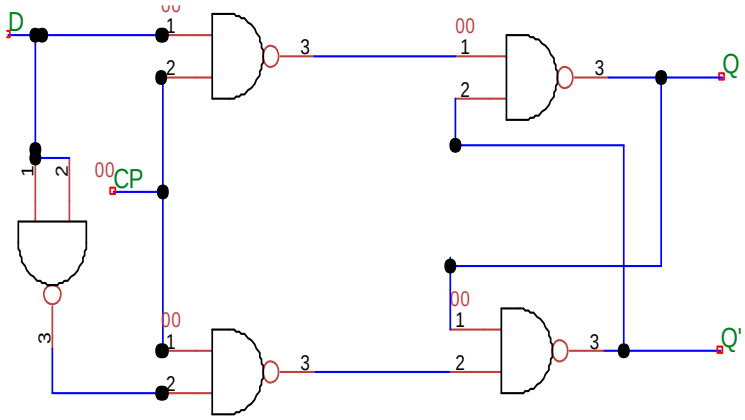
CLOCKED RS FLIP FLOP:



Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

Characteristic equation: Q(t+1) = S+ R’Q;

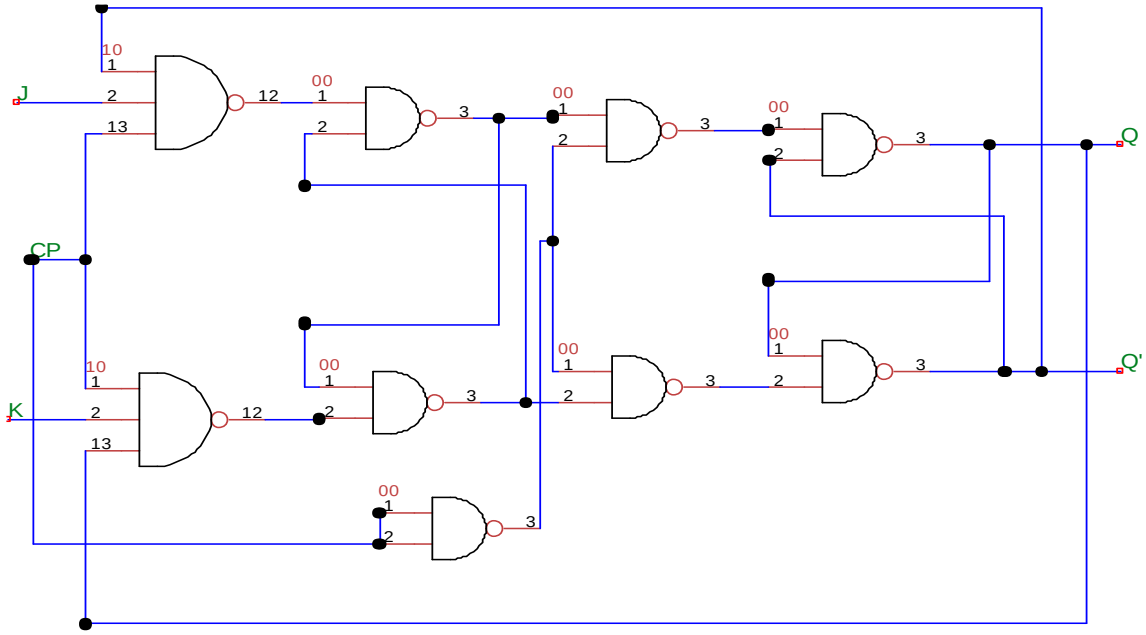
D FLIP FLOP:



Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Characteristic equation: Q(t+1) = D

JK MASTER SLAVE FLIP FLOP:

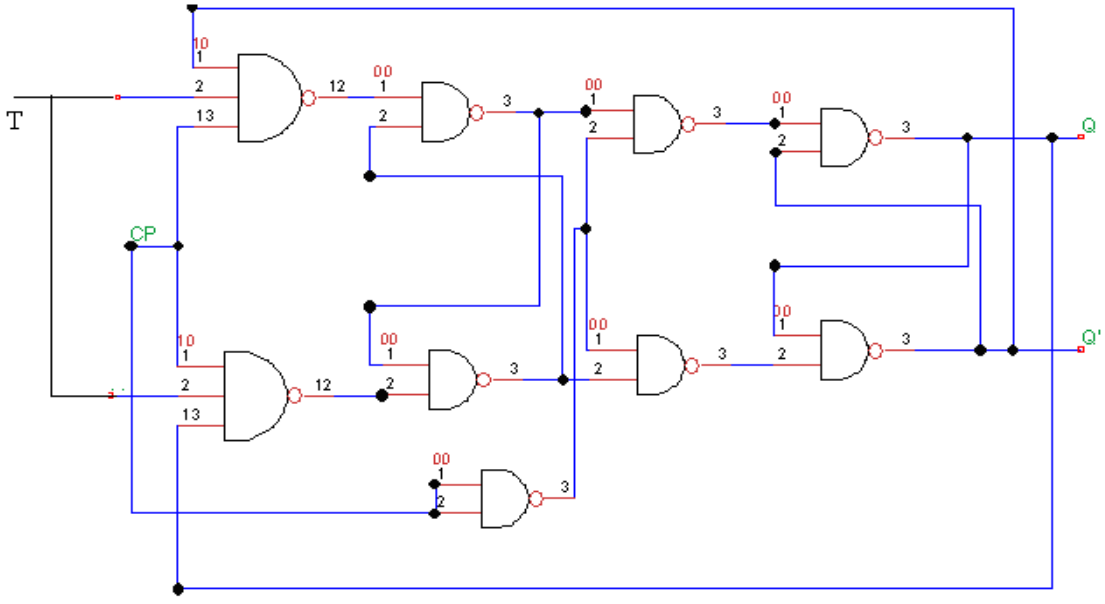


Characteristic table:

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Characteristic equation: Q(t+1) = JQ’ + K’ Q

T-FLIPFLOP:



Characteristic table:

Q	T	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic equation: Q(t+1) = TQ' + T'Q

RESULT:

The operation of various latches and flip-flops are verified using SSI gates.

SAMPLE QUESTIONS:

1. What is the difference between a combinational circuit and a sequential circuit?
2. What is a flip flop?
3. State some applications of Flip-flops?
4. Number of flip-flops required for storing n-bit of information is-------
5. 'Preset' and' clear' inputs are used in a Flip-flop for making Q=------- and ---------- respectively. They are called ---------- inputs to the flip-flop.
6. In a J-K flip flop if J=K=1, its Q output will be----------when a clock pulse is applied.
7. What is race around condition? How do you eliminate it?
8. In a T- Flip flop, the Q output----------when T=0 and clock pulse is applied.
9. A negative edge- triggered Flip flop changes state at the ----------of the clock pulse.
10. A Flip-flop with active- low preset input will have Q=---------- when preset' is connected to LOW.
11. Convert i) D flip-flop to JK flip-flop.
    ii) SR flip-flop to JK flip-flop.

.

10. SHIFT REGISTERS

AIM:

A) To verify the operation of shift register (74195 IC).

B) Verify the following operations of shift register using 74195 IC

- Ring counter
- Feedback shift register
- Bidirectional shift register
- Bidirectional shift register with parallel load

APPARATUS:

S.NO.	Component	Type	Quantity
1	4- bit shift register with parallel load	IC 74195	1
2.	Quadruple 2x1 Multiplexers	IC 74157	1
3.	Quad 2 input XOR gate	IC 7486	1
4.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. The function table for the 74195 shows the mode of operation of the register. When the clear input goes to 0, the four flip-flops clear to 0 asynchronously, That is, without the need of a clock. Synchronous operations are affected by a positive transition of the clock. To load the input data, the SH/LD must be equal to 0 and a positive clock-pulse transition must occur. To shift right, the SH/LD must be equal to 1. The J and K' inputs must be connected together to form the serial input.
2. Make connections according to the pin assignment to the inputs and outputs as shown in fig.
3. Vary the inputs and verify all the operations listed in the function table of the 74195 IC.

PART (B):

RING & JOHNSON COUNTERS:

1. A ring counter is a circular shift register with the signal from the serial output Q_D going into the serial input.
2. Connect the J and K' inputs together to form the serial input. Use the load condition to preset the ring counter to an initial value of 1000.
3. Rotate the single bit with the shift condition and check the state of the register after each clock pulse.
4. A switch tail ring counter (Johnson) uses the complement output of Q_D for the serial input. Preset the switch-tail ring counter to 0000 and predict the sequence of states that will result from shifting. Verify your prediction by

observing the state sequence after each shift.

FEEDBACK SHIFT REGISTER:

1. A feedback shift register is a shift register whose serial input is connected to some function of selected register outputs.
2. Connect a feedback shift register whose serial input is the XOR of outputs Q_C and Q_D.
3. Predict the sequence of states of the register starting from state 1000.Verify your prediction by observing the state sequence after each clock pulse.

BIDIRECTIONAL SHIFT REGISTER:

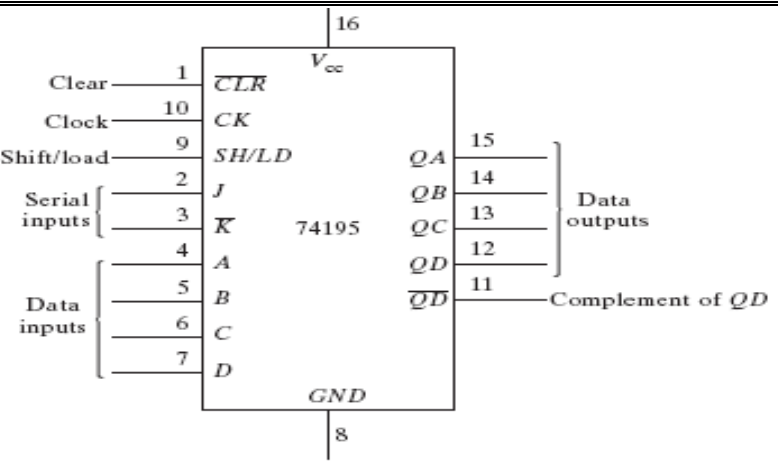
1. The 74195 can shift only right from Q_A towards Q_D. It is possible to convert the register to a bidirectional shift register by using the load mode to obtain a shift left operation (from Q_D toward Q_A). This is accomplished by connecting the output of each flip-flop to the input of the flip-flop on its left and using the load mode of the SH/LD input as a shift-left control. Input D becomes the serial input for the shift-left.
2. Connect the 74195 as a bidirectional shift register (without parallel load). Connect the serial input for shift right to a toggle switch. Construct the Shift left as a ring counter by connecting the serial output Q_A to the serial input D.
3. Clear the register and then check its operation by shifting a single 1 from the serial input switch. Shift right three more times and insert 0's from the serial input switch. Then rotate left with the shift- left (load) control. The single 1 should remain visible while shifting.

BIDIRECTIONAL SHIFT REGISTER WITH PARALLEL LOAD:

1. Construct a bidirectional shift register with parallel load using the 74195 register and the 74157 multiplexer.
2. Derive a table for the five operations (Asnch clear, shift right, shift left, parallel load, synchronous clear) as a function of the clear, clock, SH/LD inputs of 74195 and the strobe and select inputs of the 74157.
3. Connect the circuit and verify your function table. Use the parallel load condition to provide an initial value into the register.

PRECAUTIONS:

1. While verifying the individual operations, corresponding inputs must be applied.

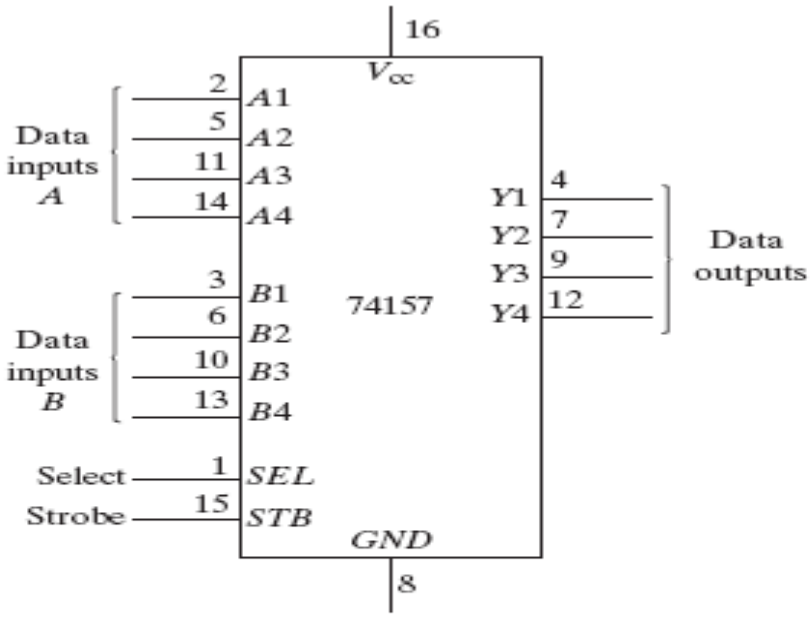


Function table

Clear	Shift/ load	Clock	J	$\overline{K}$	Serial input	Function
0	X	X	X	X	X	Asynchronous clear
1	X	0	X	X	X	No change in output
1	0	↑	X	X	X	Load input data
1	1	↑	0	0	0	Shift from QA toward QD, $QA = 0$
1	1	↑	1	1	1	Shift from QA toward QD, $QA = 1$

nc.

Fig. 11-16 IC Type 74195 Shift Register with Parallel Load

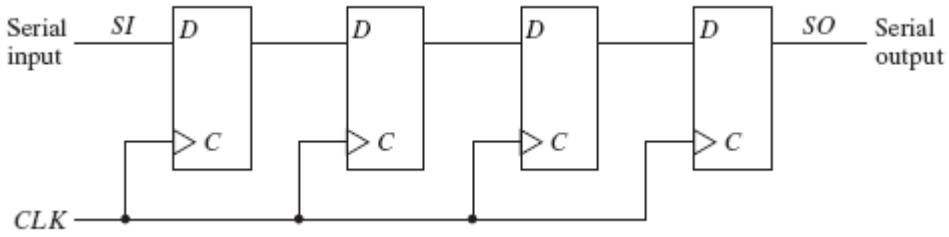


Function table

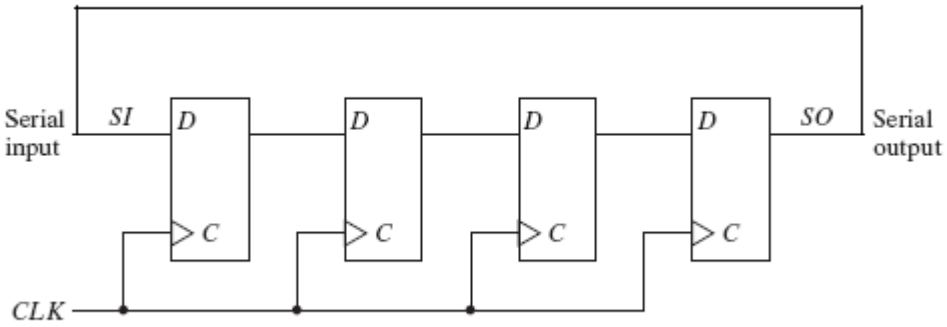
Strobe	Select	Data outputs Y
1	X	All 0's
0	0	Select data inputs A
0	1	Select data inputs B

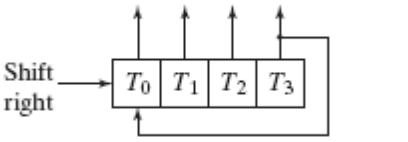
Fig. 11-17 IC Type 74157 Quadruple 2 × 1 Multiplexers

4- bit shift Register:

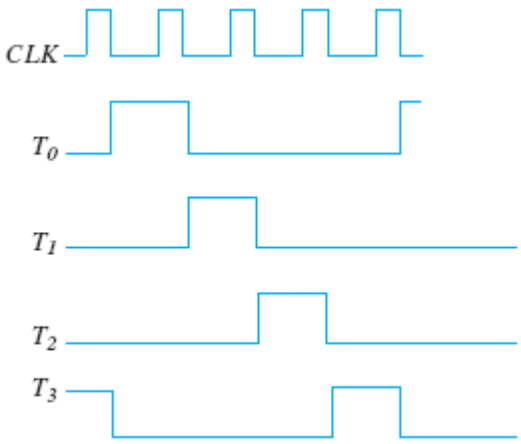


4- bit Ring counter:



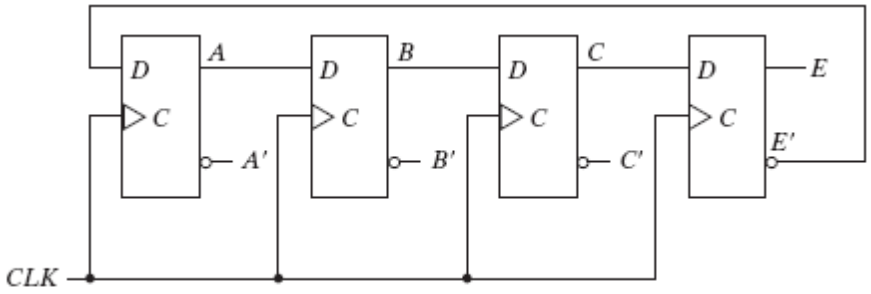


(a) Ring-counter (initial value = 1000)



(c) Sequence of four timing signals

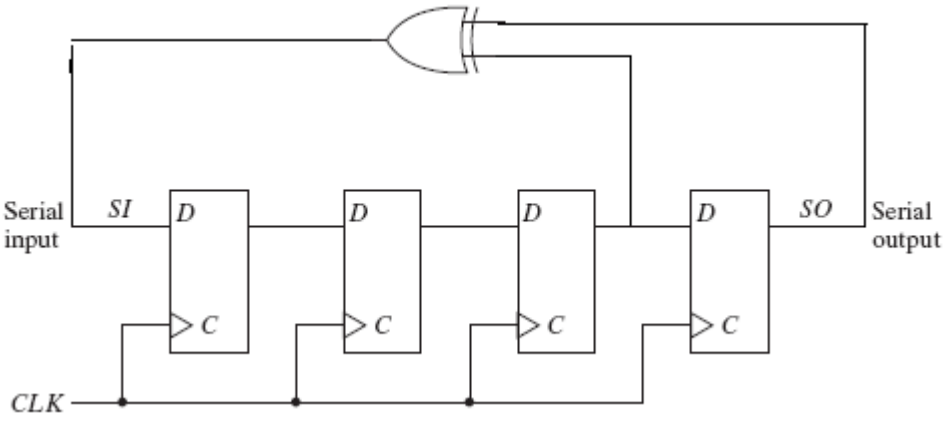
Johnson counter:



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs			
	A	*B*	*C*	*E*
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

Feedback shift Register:



RESULT:

The shift register operations are verified using 74195 IC.

SAMPLE QUESTIONS:

1. Write some applications of shift registers.
2. What is a universal shift register?
3. Draw the timing waveforms for the 4- bit ring counter?
4. What is serial and parallel loading of register? How to convert serial data to parallel and parallel data to serial? What type of register is used?
5. Write some commonly used shift register ICs?
6. How many no. of states a) a 5-bit ring counter b) a 5-bit Johnson counter will go through?

11. DESIGN OF ASYNCHRONOUS COUNTERS

AIM:

A) To construct and test 4- bit binary up/down Ripple counters using JK flip-flops.

B) To construct and test Mod-11 Ripple counter using JK flip-flops.

C) To construct and test decimal BCD counter using 74490 IC and extend it as decade decimal BCD counter.

APPARATUS:

S.NO.	Component	Type	Quantity
1	Dual JK flip-flop	IC 7476	2
2.	Dual BCD counter (asynchronous)	IC 74490	2
3.	Triple 3- input NAND gate	IC 7410	1
4.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Power supply connections are made for the two separate 7476 ICs.
2. The preset and clear pins of all flip-flops are disabled by connecting to HIGH for normal operation of flip- flops and all JK inputs to 1(Vcc or HIGH)
3. All connections are made as per the circuit diagram. For UP counter operation (0000 to 1111), the Q output of LSB flip-flop (to which clock of 1 Hz is applied) is connected as clock input for next higher significant bit flip- flop and so on. For DOWN counter operation(1111 to 0000), the Q' outputs are connected as clock input for next significant flip-flops.
4. The Q outputs of all four flip- flops are connected to 4 output indicators for showing the count.
5. Apply clock input from pulser (press push button in the pulser and release it to produce a single pulse) and verify the count and also counter operation. You can also give clock input of low frequency from clock generator(about 1 Hz).

PART (B):

1. Keep the above circuit as it is and take 3 –input NAND gate and give power supply connections.
2. Disconnect the clear input pins of all flip- flops from Vcc(HIGH). For MOD- 11 operation(count 0 to 10 and for $Q_4Q_3Q_2Q_1 = 1011$), connect the Q outputs Q_4 ,Q_2 and Q_1 as inputs to the 3 input NAND gate and its output is connected to clear inputs of all flip-flops.
3. Now, apply clock pulses and observe the count sequence from 0000 to 1010 and back to 0000.

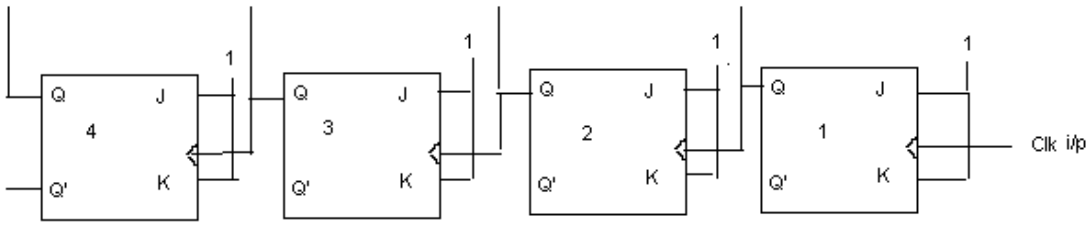
PART (C):

1. Give power supply connections to the corresponding pins of 74490 IC and disable the Set and Clear pins of the IC.
2. Connect the input to a pulser and the four outputs to indicator lamps. Verify the count goes from 0000 to 1001.
3. To extend it for decade decimal BCD counter, connect the circuit as per the diagram and give a clock input and observe the count sequence at the outputs of all individual counters.

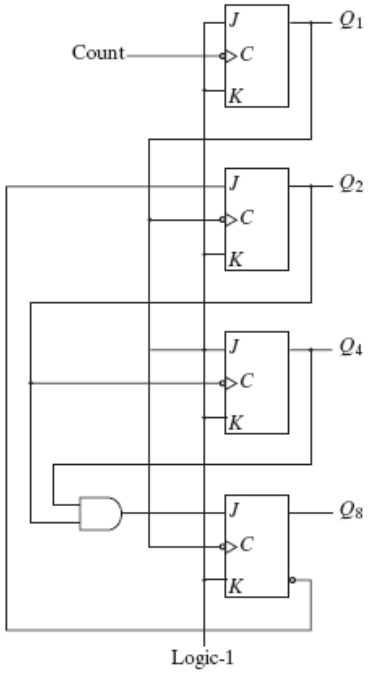
PRECAUTIONS:

1. No pins must be left open. If we are not using particular pins in that application, disable those pins.
2. Care has to be taken in identifying the LSB and MSBs.
2. Avoid loose connections and shorting of pins.

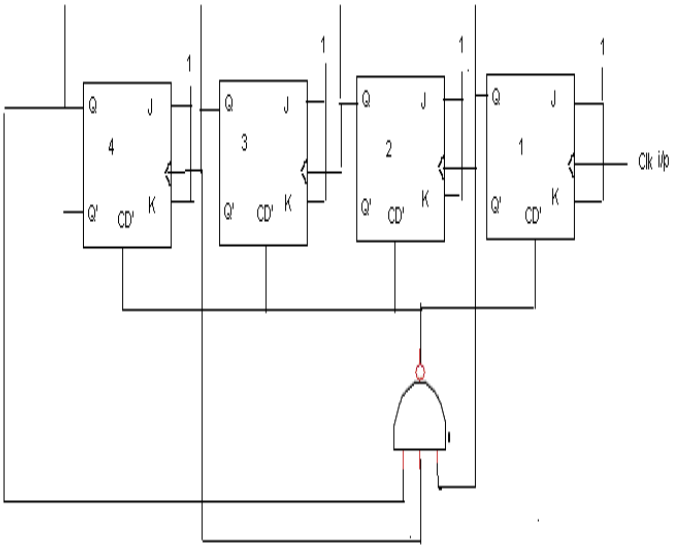
4- BIT RIPPLE (UP)COUNTER:



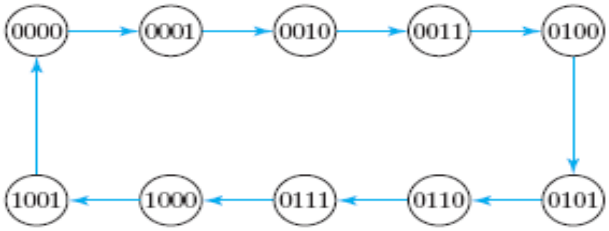
MOD-11 RIPPLE COUNTER:



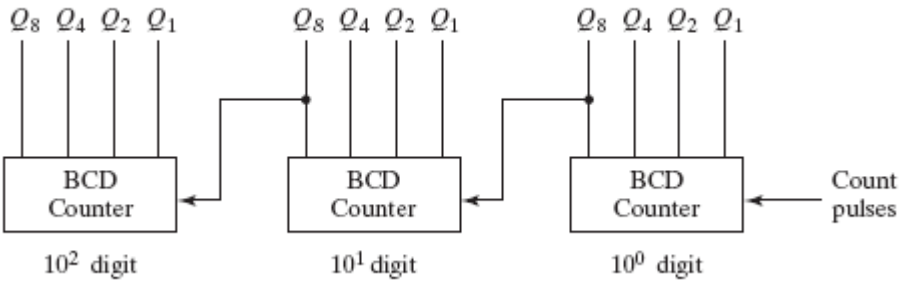
BCD RIPPLE COUNTER:



State diagram of Decimal BCD Counter



THREE DECADE DECIMAL BCD COUNTER:



COUNT SEQUENCES:

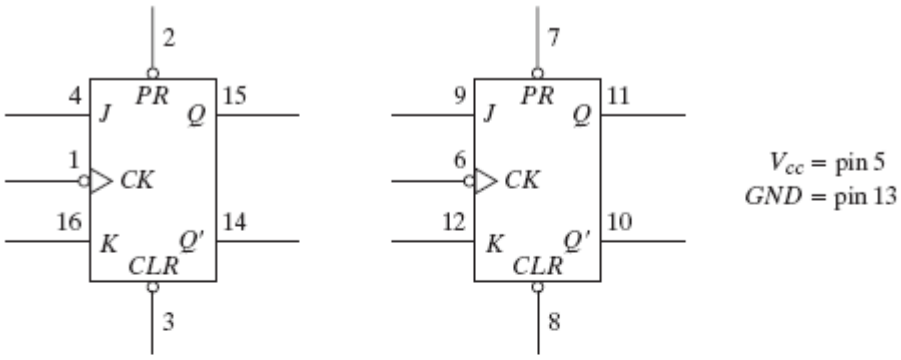
MOD-16 UP COUNTER:

Clock pulses	Count sequence $Q_4Q_3Q_2Q_1$
0	0 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 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 0 0 0

IC Type 7476 Dual *JK*

MOD-11 COUNTER:

Clock pulses	Count sequence $Q_4Q_3Q_2Q_1$
0	0 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 0 0 0
↑	1 0 0 1
↑	1 0 1 0
↑	0 0 0 0



Function table

Inputs					Outputs	
Preset	Clear	Clock	J	K	Q	Q'
0	1	X	X	X	1	0
1	0	X	X	X	0	1
0	0	X	X	X	1	1
1	1	⊓	0	0	No change	
1	1	⊓	0	1	0	1
1	1	⊓	1	0	1	0
1	1	⊓	1	1	Toggle	

RESULT:

The 4- bit Asynchronous counter UP/DOWN Mod-11 counter, and decade BCD counters constructed and their operation is verified.

SAMPLE QUESTIONS:

1. Draw the timing diagrams showing the relationship between clock and the four outputs of 4- bit UP counter(Ripple)
2. What is the main classification of counters and define?
3. What do you mean by modulus of a counter?
4. How do you determine the maximum frequency range of a given n- bit counter?
5. How many flip flops are needed for a Mod-100 counter?
6. What is the difference between counters and registers?
7. State some commonly available asynchronous counter ICs?
8. What is the difference between edge triggered and level triggered circuits?
9. A counter counts no. of ---------- to the circuit.
10. Draw the logic diagram of 4-bit binary ripple down counter.

12. DESIGN OF SYNCHRONOUS COUNTERS

AIM:

A) To construct and test 4- bit binary Up/Down Synchronous counters using JK flip-flops.

B) To construct and test Mod-11 counter using JK flip-flops.

APPARATUS:

S.NO.	Component	Type	Quantity
1	Dual JK flip-flop	IC 7476	2
2.	Quad 2 input AND gates	IC 7408	1
3.	Quad 2 input NAND gates	IC 7400	1
4.	Digital IC trainer	-	1

PROCEDURE:

PART (A):

1. Power supply connections are given to the corresponding pins of the ICs.
2. The Set(S_D')and Clear pins(C_D') of all flip-flops are disabled by connecting to HIGH for normal operation of flip- flops and a common clock pulse is given to all the clock inputs of all flip-flops.
3. The JK inputs of all flip- flops are connected as per the circuit diagram and the four outputs $Q_4Q_3Q_2Q_1$ are connected to four output indicators for observing the count sequence.
4. Now, apply clock pulse input from pulser and observe the count sequence for 4-bit counter operation. For DOWN counter, connect the Q' outputs(instead of Q's) as inputs of the AND gates in the circuit diagram and apply clock pulse and observe the count sequence.

PART (B):

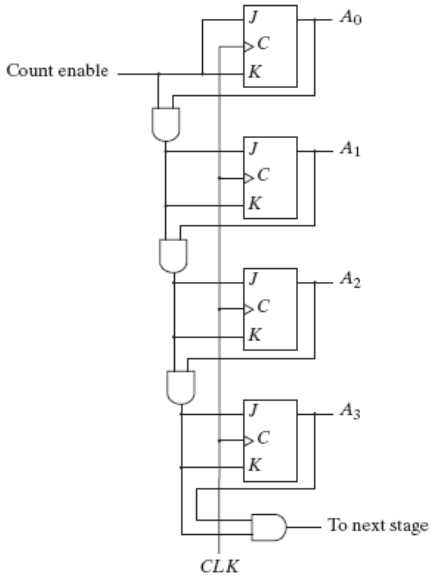
1. The excitation table for MOD-11 counter is derived and the JK inputs of all flip- flops are derived after simplification using K-maps and the circuit diagram is drawn.
2. The connections are made as per the circuit diagram.
3. Apply clock input and observe the count sequence from 0000 to 1010 and back to 0000.

PRECAUTIONS:

1. Care must be taken while deriving the JK inputs of each flip-flop from the excitation table.
2. Ensure before making connections, the proper functioning of each gate in the individual components.
3. There should not be any loose connections and avoid shorting of pins.

4- BIT SYNCHRONOUS BINARY UP COUNTER:

MOD-16 UP COUNTER:



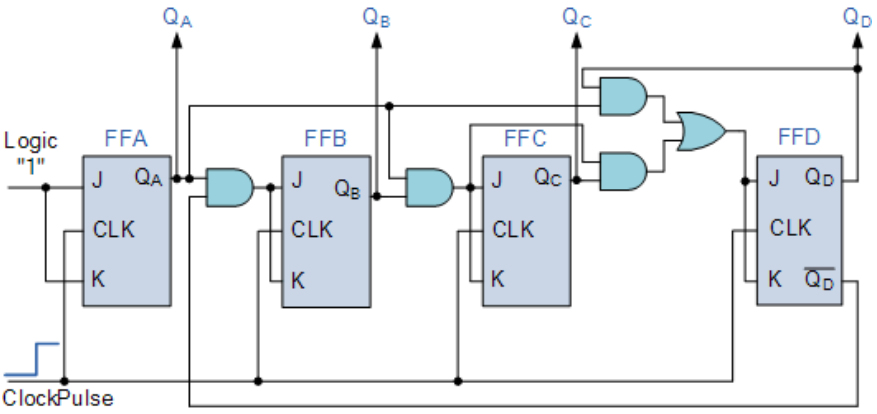
COUNT SEQUENCES:
MOD-16 UP COUNTER:

Clock pulses	Count sequence $Q_4Q_3Q_2Q_1$
0	0 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 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 0 0 0

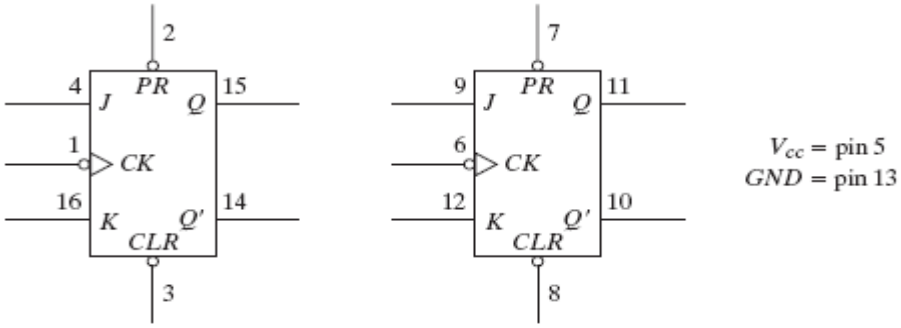
MOD-10 COUNTER:

Clock pulses	Count sequence $Q_4Q_3Q_2Q_1$
0	0 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 0 0 0
↑	1 0 0 1
↑	0 0 0 0

MOD 10 SYNCHRONOUS COUNTER:



IC Type 7476 Dual *JK* Master–Slave Flip-Flops



V_{cc} = pin 5
GND = pin 13

Function table

Inputs					Outputs	
Preset	Clear	Clock	*J*	*K*	*Q*	*Q'*
0	1	*X*	*X*	*X*	1	0
1	0	*X*	*X*	*X*	0	1
0	0	*X*	*X*	*X*	1	1
1	1	⊓	0	0	No change	
1	1	⊓	0	1	0	1
1	1	⊓	1	0	1	0
1	1	⊓	1	1	Toggle	

RESULT:

The 4-bit and MOD-11 synchronous counters are designed and their operation is verified.

SAMPLE QUESTIONS:

1. What is the difference in operations of synchronous and Ripple counters?
2. The modulus of a 4- bit counter is----------.

3. The count of a 4- bit binary DOWN counter is 0000. When a clock pulse is applied its count will be----------.
4. The cascade of divide- by 5 counter followed by divide-by-2 counter is in state 0000. When a clock pulse is applied, its state will be ----------.
   Assume negative edge triggered circuits.
5. State some commonly available synchronous counter ICs with their features?
6. State some applications of counters.
7. What is lock-out condition in counters? How to avoid it?
8. How will you ensure whether a counter is self- starting or not?
9. What do you mean by binary counters?
10. What do you mean by triggering?

13. FOUR BIT ARITHMETIC LOGIC UNIT

AIM:

To perform 4- bit addition and 4- bit subtraction and some logic operations using 74181 IC

APPARATUS:

S.NO.	Component	Type	Quantity
1	4-bit ALU	IC 74181	1
2.	Digital IC trainer	-	1

THEORY:

FUNCTIONAL DESCRIPTION

The 74LS181 is a 4-bit high speed parallel Arithmetic Logic Unit (ALU) controlled by the four Function Select Inputs(S0 . . . S3) and the Mode Control Input (M), it can perform all the 16 possible logic operations or 16 different arithmetic operations on active HIGH or active LOW operands. The Function Table lists these operations. When the Mode Control Input (M) is HIGH, all internal carries are inhibited and the device performs logic operations on the individual bits as listed. When the Mode Control Input is LOW, the carries are enabled and the device performs arithmetic operations on the two 4-bit words. The device incorporates full internal carry look ahead and provides for either ripple carry between devices using the C_{n+4} output, or for carry look ahead between packages using the signals P (Carry Propagate) and G (Carry Generate), P and G are not affected by carry in. When speed requirements are not stringent, the LS181 can be used in a simple ripple carry mode by connecting the Carry Output ($C_{n+4)}$ signal to the Carry Input (C_n) of the next unit. For high speed operation the LS181 is used in conjunction with the 9342 or 93S42 carry look ahead circuit. One carry look ahead package is required for each group of the four LS181 devices. Carry look ahead can be provided at various levels and offers high speed capability over extremely long word lengths. The A = B output from the LS181 goes HIGH when all four F outputs are HIGH and can be used to indicate logic equivalence over four bits when the unit is in the subtract mode. The A = B output is open collector and can be wired-AND with other A = B outputs to give a comparison for more than four bits. The A = B signal can also be used with the Cn+4 signal to indicate A>B and A<B. The Function Table lists the arithmetic operations that are performed without a carry in. An incoming carry adds a one to each operation. Thus, select code LHHL generates A minus B minus 1 (2s complement notation) without a carry in and generates A minus B when a carry is applied. Because subtraction is actually performed by complementary addition (1s complement), a carry out means borrow; thus a carry is generated when there is no underflow and no carry is generated when there is underflow. As indicated, the LS181 can be used with either active LOW inputs producing active LOW outputs or with active HIGH inputs producing active

HIGH outputs. For either case the table lists the operations that are performed to the operands labeled inside the logic symbol.

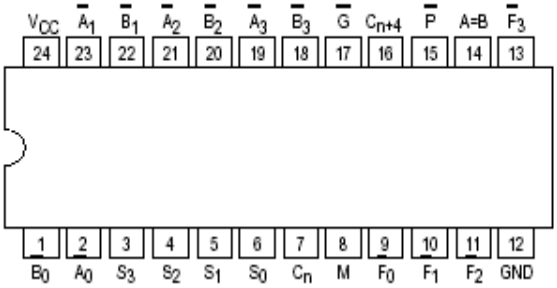
PROCEDURE:

1. Power supply connections are given to the corresponding pins of 74181 IC.
2. A Data inputs and B Data inputs are connected to eight toggle switches. Four select lines S0 –S3 and Mode control input (M) and carry input C_n are connected to input switches.
3. Fix data input A and B some values, For M=L and C_n=L, verify the arithmetic operations according to the function table. For M=H, verify logic operations by varying the selection lines and verify the function table.

PRECAUTIONS:

1. Care must be taken while verifying the operations while dealing with Active LOW inputs and outputs.

CONNECTION DIAGRAM DIP (TOP VIEW)



PIN NAMES

$\overline{A}_0-\overline{A}_3$, $\overline{B}_0-\overline{B}_3$	Operand (Active LOW) Inputs
S_0-S_3	Function — Select Inputs
M	Mode Control Input
C_n	Carry Input
$\overline{F}_0-\overline{F}_3$	Function (Active LOW) Outputs
$A = B$	Comparator Output
G	Carry Generator (Active LOW) Output
$\overline{P}$	Carry Propagate (Active LOW) Output
C_{n+4}	Carry Output

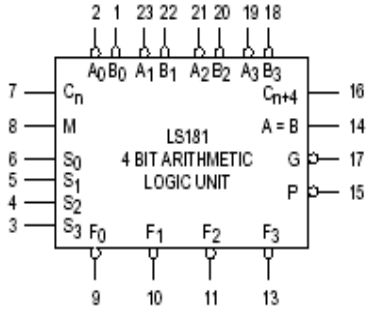
FUNCTION TABLE

MODE SELECT INPUTS				ACTIVE LOW INPUTS & OUTPUTS		ACTIVE HIGH INPUTS & OUTPUTS	
				LOGIC (M = H)	ARITHMETIC** (M = L) (C_n = L)	LOGIC (M = H)	ARITHMETIC** (M = L) (C_n = H)
S_3	S_2	S_1	S_0				
L	L	L	L	$\overline{A}$	A minus 1	$\overline{A}$	A
L	L	L	H	$\overline{AB}$	AB minus 1	$\overline{A}$ + B	A + $\overline{B}$
L	L	H	L	A + B	AB minus 1	AB	A + B
L	L	H	H	Logical 1 minus 1		Logical 0 minus 1	
L	H	L	L	$\overline{A}$ + B	A plus (A + B)	$\overline{AB}$	A plus AB
L	H	L	H	$\overline{B}$	AB plus (A + B)	B	(A + B) plus AB
L	H	H	L	$A \oplus \overline{B}$	A minus B minus 1	$A \oplus B$	A minus B minus 1
L	H	H	H	$\overline{A}$ + B	A + B	$\overline{AB}$	AB minus 1
H	L	L	L	AB	A plus (A + B)	$\overline{A + B}$	A plus AB
H	L	L	H	$A \oplus B$	A plus B	$A \oplus B$	A plus B
H	L	H	L	B	AB plus (A + B)	B	(A + B) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logical 0 A plus A*		Logical 1 A plus A*	
H	H	L	H	AB	AB plus A	A + B	(A + $\overline{B}$) plus A
H	H	H	L	AB	AB plus A	A + B	(A + B) Plus A
H	H	H	H	A	A	A	A minus 1

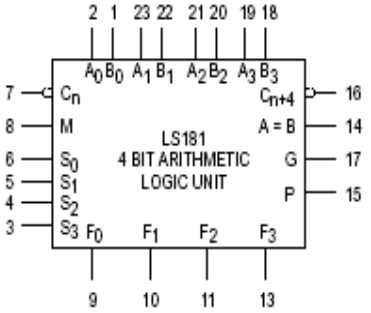
L = LOW Voltage Level
H = HIGH Voltage Level
*Each bit is shifted to the next more significant position
**Arithmetic operations expressed in 2s complement notation

LOGIC SYMBOLS

ACTIVE LOW OPERANDS



ACTIVE HIGH OPERANDS



14. SERIAL ADDER- SUBTRACTOR

AIM: To construct and test a 4-bit serial adder-subtractor circuit.

APPARATUS:

S.NO.	Component	Type	Quantity
1	4-bit shift register with parallel load and asynchronous clear	74195	2
2	Quad 2 input AND gates	7408	1
3.	Quad 2 input EX-OR gates	7486	1
4.	Quad 2 input NOR gates	7432	1
5.	Dual JK Master –Slave flip-flop	7476	1

THEORY:

A digital system is said to operate in a serial mode when information is transferred and manipulated one bit at a time. Information is transferred one bit at a time by shifting the bits out of the source register into the destination register. This in contrast to parallel transfer where all the bits of the register are transferred at the same time.

Operations in digital computers are usually done in parallel because this is a faster mode of operation. Serial operations are slower, but have the advantage of requiring less equipment.

The two binary numbers to be added serially are stored in two shift registers. Bits are added one pair at a time through a single full adder circuit. The carry out of the full adder is transferred to a flip-flop. The output of this flip-flop is then used as the carry input for the next pair of significant bits. The sum bit from the S output of the full adder could be transferred into a third shift register. By shifting the sum into A while the bits of A are shifted out, it is possible to use one register for storing both the augend and the sum bits. The serial input of register B can be used to transfer a new binary number while the addend bits are shifted out during the addition.

The operation of the serial adder is as follows. Initially register A holds the augend, register B holds the addend, and the carry flip-flop is cleared to 0. The outputs (SO) of A and B provide a pair of significant bits for the full adder at x and y. Output Q of the flip-flop provides the input carry at z. The shift control enables both registers and the carry flip-flop, so at the next clock pulse, both registers are shifted once to the right, the sum bit from S enters the leftmost flip-flop of A, and the output carry is transferred into flip-flop Q. The shift control enables the registers for a number of clock pulses equal to the number of bits in the register. For each succeeding clock pulse, a new sum bit is transferred to A, a new carry bit is transferred to Q, and both registers are shifted once to the right. This process continues until the shift control is disabled. Thus, the addition is accomplished by passing each pair of bits together with the previous carry

through a single full adder circuit and transferring the sum, one bit at a time, into register A.

Initially, register A and the carry flip-flop are cleared to 0 and then the first number is added from B. While B is shifted through the full adder, a second number is transferred to it through its serial input. The second number is then added to the content of register A while a third number is transferred serially into register B. This can be repeated to form the addition of two, three, or more numbers and accumulate their sum in register A.

PROCEDURE:

Construction of serial adder:

1. Provide a facility for register B to accept parallel data from four toggle switches and connect its serial input to ground so that 0's are shifted into register B during the addition.
2. Provide a toggle switch to clear the registers and the flip-flop. Another switch will be needed to specify whether register B is to accept parallel data or is to be shifted during the addition.
3. Make the connections as per the circuit diagram.

Testing the serial adder:

1. To test your serial adder, perform the binary addition 5+6+15=26. This is done by first clearing the registers and the carry flip-flop. Parallel load the binary value 0101 into register B.
2. Apply four pulses to add B to A serially and check that the result in A is 0101.
3. Parallel load 0110 into B and add it to A serially. Check that A has the proper sum.
4. Parallel load 1111 into B and add to A. Check that the value in A is 1010 and that the carry flip-flop is set.
5. Clear the registers and flip-flop and try a few other numbers to verify that your serial adder is functioning properly.

Serial adder- subtractor:

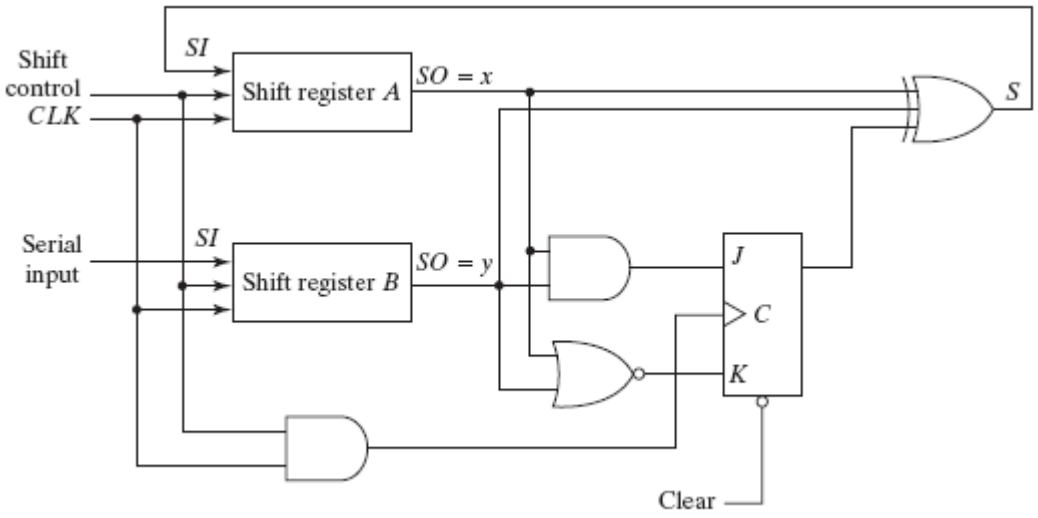
1. In a serial subtractor (that subtracts A-B), we will find that the output difference is the same as the output sum, but that the input to the J and K of the borrow flip- flop needs the complement of QD (available in the 74195).
2. Using the other two XOR gates from the 7486, convert the serial adder to a serial adder-subtractor with a mode control M. When M=0, the circuit adds A+B. When M=1, the circuit subtracts A-B and the flip-flop holds the borrow instead of the carry.
3. Test the adder part of the circuit by repeating the operations recommended above to ensure that the modification did not change the operation.
4. Test the serial subtractor part by performing the operations
   15-4-5-13= -7. Binary 15 can be register A by first clearing it to 0 and adding 15 from B. Check the intermediate results during the subtraction. Note

that -7 will appear as the 2's complement of 7 with a borrow of 1 in the flip-flop.

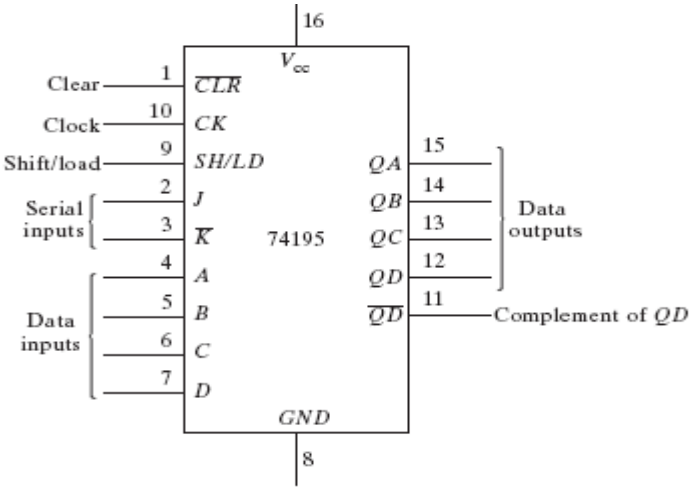
PRECAUTIONS:

1. While taking new data, it should be ensured that registers and flip-flops are cleared.
2. Clock pulses for the 7476 must be of the suitable type.

CIRCUIT DIAGRAM OF SERIAL ADDER:



74195 IC PIN CONFIGURATION:



Function table

Clear	Shift/ load	Clock	J	$\overline{K}$	Serial input	Function
0	X	X	X	X	X	Asynchronous clear
1	X	0	X	X	X	No change in output
1	0	↑	X	X	X	Load input data
1	1	↑	0	0	0	Shift from QA toward QD, $QA = 0$
1	1	↑	1	1	1	Shift from QA toward QD, $QA = 1$

nc.

Fig. 11-16 IC Type 74195 Shift Register with Parallel Load

RESULT: 4-bit serial adder- Substractor circuit is constructed and verified.

SAMPLE QUESTIONS:

1. What is a serial adder?
2. What is a parallel adder?
3. How many full adders are needed in implementing 4-bit serial adder?
4. How many full adders are needed in implementing 4-bit parallel adder?
5. How many shift registers are needed for implementing a serial adder?
6. Which type of flip-flop is used in this experiment? Can D-type flip-flop be used?

15. MEMORY UNIT

AIM: a) To investigate the behavior of a random-access memory (RAM) unit and its storage capacity
b) To simulate a ROM using RAM
c) To implement a combination circuit using ROM simulator that converts a 4-bit binary number to its equivalent gray code.

APPARATUS:

S.NO.	Component	Type	Quantity
1	16x4 RAM	IC 74189	1
2	Binary counter	IC 7493	1
3.	Hex inverter	IC 7404	1
4.	Digital IC trainer	-	1

THEORY:

A memory unit is a collection of cells capable of storing a large quantity of binary information. Binary information is transferred for storage and from which information is available when needed for processing.

There are two types of memories that are used in digital systems: random-access memory (RAM) and read-only memory (ROM). RAM can perform both read and write operations. ROM can only perform read operation.

IC type 74189 is a 16x4 random-access memory. The pin assignment to the inputs and outputs is shown in fig 1. The four address inputs select one of 16 words in the memory. The least significant bit of the address is A_0 and the most significant bit is A_3.

The chip select (CS) input must be equal to 0 to enable the memory. If CS is equal to 1, the memory is disabled and all four outputs are in a high impedance state. The write enable (WE) input determines the type of operation as indicated in the function table. The write operation is performed when WE=0. This is a transfer of the binary number from the data inputs into the selected word in memory. The read operation is performed when WE=1.This transfers the complement value stored in the selected word into the output data lines. The memory has three state outputs to facilitate memory expansion.

PROCEDURE:

I. Testing the RAM:

Since the outputs of the 74189 produce the complement values, we need to insert four inverters to change the outputs to their normal value. The RAM can be tested after making the following connections:

1. Connect the address inputs to a binary counter using the 7493 IC.
2. Connect the four data inputs to toggle switches and the data outputs to four 7404 inverters. Provide four indicator lamps for the address and four more for the outputs of the inverters.
3. Connect input CS to ground and WE to a toggle switch (or a pulser that provides a negative pulse).
4. Store a few words into the memory and then read them to verify that the write and read operations are functioning properly. The proper way to write is first to set the address in the counter and the inputs in the four toggle switches. To store the word in memory, flip the WE switch to the write position and then return it to the read position.

II.ROM simulator:

1. A ROM simulator is obtained from a RAM when operated in the read mode only.
2. The pattern of 1's and 0's is first entered into the simulating RAM by placing the unit momentarily in the write mode.
3. Simulation is achieved by placing the unit in the read mode and taking the address lines as inputs for the ROM. The ROM can then be used to implement any combinational circuit.

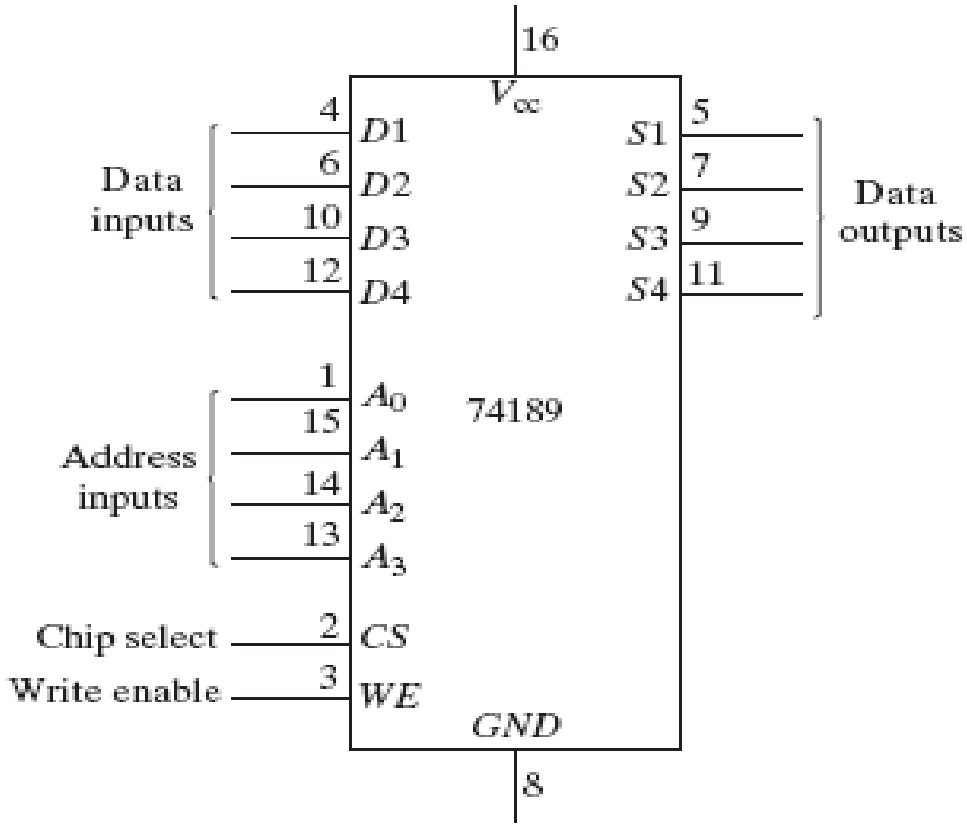
III. Combinational circuit implementation using ROM simulator:

1. Obtain the truth table of the code converter.
2. Store the truth table into the 74189 memory by setting the address inputs to the binary value and the data inputs to the corresponding gray code value.
3. After all 16 entries of the table are written in memory, the ROM simulator is set by connecting the WE line to logic-1 permanently.
4. Check the code converter by applying the inputs to the address lines and verifying the correct outputs in the data output lines.

PRECAUTIONS:

1. You must be careful when using the WE switch. Always leave the WE input in the read mode, unless you want to write into memory.
2. Be careful not to change the address or the inputs when WE is in the write mode.

IC Type 74189 16 × 4 RAM

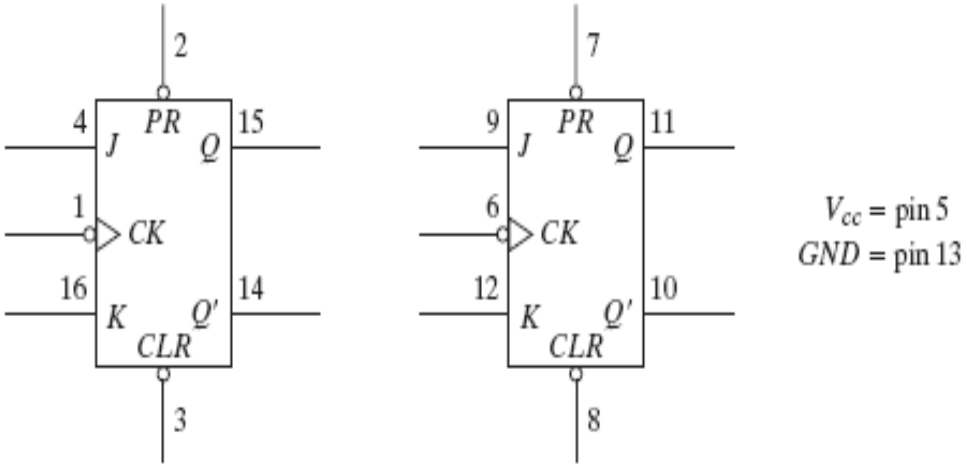


Function table

CS	WE	Operation	Data outputs
0	0	Write	High impedance
0	1	Read	Complement of selected word
1	X	Disable	High impedance

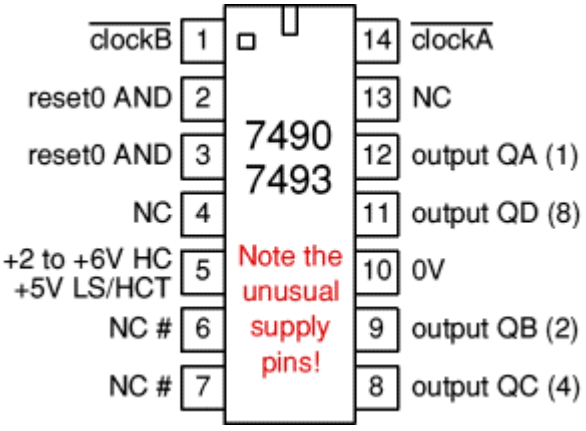
I Inc

IC Type 7476 Dual *JK* Master–Slave Flip-Flops



Function table

Inputs					Outputs	
Preset	Clear	Clock	*J*	*K*	*Q*	*Q'*
0	1	*X*	*X*	*X*	1	0
1	0	*X*	*X*	*X*	0	1
0	0	*X*	*X*	*X*	1	1
1	1	⊓	0	0	No change	
1	1	⊓	0	1	0	1
1	1	⊓	1	0	1	0
1	1	⊓	1	1	Toggle	



RESULT:

The behavior of RAM is investigated and using it, ROM is simulated and the given combinational circuit is implemented using ROM simulator.

SAMPLE QUESTIONS:

1. What are the different types of programmable devices?
2. What are the various various ways in which ROMs can be programmed?
3. How will you specify the size of a ROM?
4. What is the basic difference between RAM and ROM?
5. How many address lines and data input-output lines are needed to specify a memory unit of 256K X 64?