

ECE-318

Digital Integrated Circuits

Laboratory Manual



Prepared By

Mr.D.Anil Prasad
Assistant Professor
Dept of ECE

Physical Lab-Incharge

Mr.N.Srinivasa Naidu
Assistant Professor
Dept of ECE

Head of the Department

Dr.K.Murali Krishna
Prof. & HOD
Dept of ECE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES

(Approved by AICTE, Permanently Affiliated to Andhra University,
Accredited by NBA & NAAC with “A” Grade)

Sangivalasa-531162, Bheemunipatnam Mandal, Vsp .Dt.

Phone: 08933- 225084,226395

VISION OF INSTITUTE

ANITS envisions to emerge as a world-class technical institution whose products represent a good blend of technological excellence and the best of human values.

MISSION OF THE INSTITUTE

To train young men and women into competent and confident engineers with excellent communication skills, to face the challenges of future technology changes, by imparting holistic technical education using the best of infrastructure, outstanding technical and teaching expertise and an exemplary work culture, besides molding them into good citizens

VISION OF THE DEPARTMENT

To become a centre of excellence in Education, research and produce high quality engineers in the field of Electronics and Communication Engineering to face the challenges of future technological changes.

MISSION OF THE DEPARTMENT

The Department aims to bring out competent young Electronics & Communication Engineers by achieving excellence in imparting technical skills, soft skills and the right attitude for continuous learning.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: To prepare graduates for successful career in Electronics industries, R&D organizations and/or IT industries by providing technical competency in the field of Electronics & Communication Engineering.

PEO2: To prepare graduates with good scientific and engineering proficiency to analyze and solve electronic engineering problems.

PEO3: To inculcate in students professionalism, leadership qualities, communication skills and ethics needed for a successful professional career.

PEO4: To provide strong fundamental knowledge in men and women students to pursue higher education and continue professional development in core engineering and other fields.

ECE 318 DIGITAL IC's LABORATORY

<i>Credits</i>	<i>Periods</i>			<i>Exam Hrs.</i>	<i>Sessional Marks</i>	<i>Exam Marks</i>	<i>Total Marks</i>
	<i>Theory</i>	<i>Tutorial</i>	<i>Lab</i>				
2	-	-	3	3	50	50	100

COURSE OBJECTIVES:

1. To understand the simplification methods (Boolean algebra & postulates, k-map method and tabular method) to simplify the given Boolean function.
2. To understand the fundamentals of digital logic and design various combinational and sequential circuits.
3. To understand formal procedure for the analysis and design of synchronous and asynchronous sequential logic

COURSE OUTCOMES:

After completion of the course the student will be able to

CO[1] Verify the logic behavior of IC gates and implement the given Boolean functions using basic logic gates and/ or Universal gates.

CO[2] Design, Analyze and Implement combinational circuits for given specifications

CO[3] Design, Analyze and Implement flip-flops and registers

CO[4] Design, Analyze and Implement counters to meet required specifications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	3	-	-	-	-	-	1	3	-
CO2	3	3	2	3	-	-	-	-	-	1	3	-
CO3	3	3	2	3	-	-	-	-	-	1	3	-
CO4	3	2	2	3	-	-	-	-	-	1	3	-

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	-	3	-	2
CO2	-	3	-	2
CO3	-	3	-	2
CO4	-	3	-	2

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

LIST OF EXPERIMENTS

1. Logic behavior of various IC gates.
2. Implementation of Boolean functions using Basic Gates
3. Implementation of Boolean functions using universal gates.
4. Adders and Subtractors
5. 4-bit binary parallel adder
6. 4-bit Magnitude comparator
7. Design with Multiplexer.
8. Applications of Multiplexers
9. FLIP-FLOPS.
10. Design and testing of Ripple Counters
11. Design and testing of Mod-K Synchronous Counters.
12. Design and testing of Shift Registers.

1. VERIFICATION OF LOGIC BEHAVIOR OF IC GATES

AIM:

To verify the logic behavior of various IC gates

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. 7400 quadruple two-input NAND gates
4. 7402 quadruple two-input NOR gates
5. 7404 hex inverters
6. 7408 quadruple two-input AND gates
7. 7432 quadruple two-input OR gates
8. 7486 quadruple two-input XOR gates

THEORY

In this experiment logic behavior of various IC gates is to be verified.

The pin assignments of various gates are shown in Fig. 1.1." Quadruple" means that there are four gates within the package. The digital circuits can be constructed by using standard integrated circuits (ICs) mounted on breadboards that are easily assembled in the laboratory.

The Digital IC trainer kit is required for performing the experiments which consists of the following equipment:

1. Light-emitting diode (LED) indicator lamps
2. Toggle switches to provide logic-1 and logic-0 signals
3. Pulsers with push buttons and debounce circuits to generate single pulses.
4. A clock-pulse generator with at least two frequencies: a low frequency of about 1 pulse per second to observe slow changes in digital signals and a higher frequency for observing waveforms in an oscilloscope.
5. A power supply of 5 V.
6. Socket strips for mounting the ICs.

The integrated circuits to be used in the experiments can be classified as small-scale integration (SSI) or medium-scale integration (MSI) circuits. SSI circuits contain individual gates or flip-flops, and MSI chips perform specific digital functions. The eight

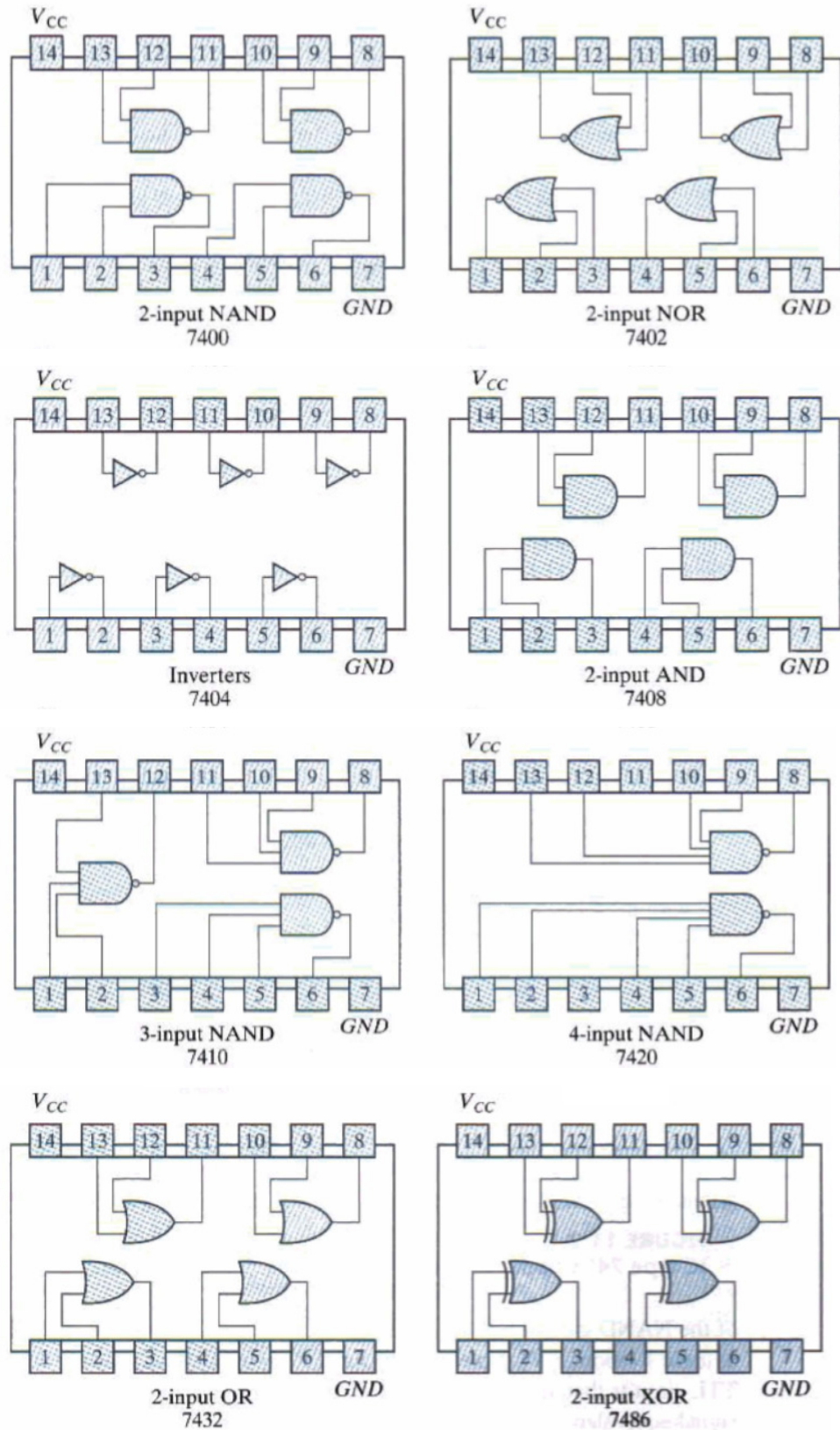
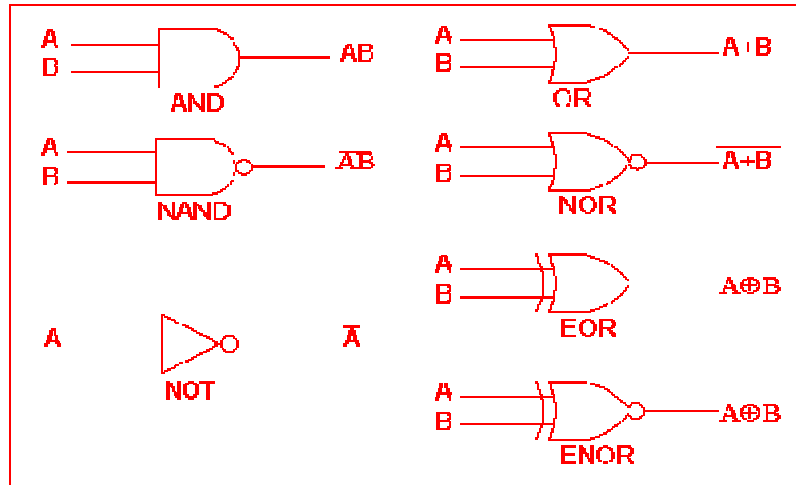


Fig. 1.1 Digital gates in IC package with Identification numbers and pin assignments

SSI gate ICs needed for the experiments-----two-input NAND, NOR, AND, OR, XOR gates, inverters and three- input and four-input NAND gates are shown in Fig. 1.1. The pin assignments for the gates are indicated in the diagram. The pins are numbered from 1 to 14, Pin number 14 is marked V_{CC} , and pin number 7 is marked GND (ground)



	x	y	$F=x.y$		x	y	$F=x+y$
AND GATE	0	0	0	OR GATE	0	0	0
	0	1	0		0	1	1
	1	0	0		1	0	1
	1	1	1		1	1	1
	x	y	$F=(x.y)'$		x	y	$F=(x+y)'$
NAND GATE	0	0	1	NOR GATE	0	0	1
	0	1	1		0	1	0
	1	0	1		1	0	0
	1	1	0		1	1	0
	x	y	$F=xy'+x'y$		x	y	$F=xy+\overline{xy}$
EX-OR GATE	0	0	0	EX-NOR GATE	0	0	1
	0	1	1		0	1	0
	1	0	1		1	0	0
	1	1	0		1	1	1

	x	F=x'		x	F=x
NOT	0	1	BUFFER	0	0
GATE	1	0		1	1

Fig. 1.2: Truth tables of Digital logic gates

These are the supply terminals, which must be connected to a power supply of 5 V for proper operation of the circuit. Each IC is recognized by its identification number for example, the two-input NAND gates are found inside the IC whose number is 7400. Since Boolean functions are expressed in terms of AND, OR, and NOT operations, it is easier to implement a Boolean function with these types of gates. The truth tables of the eight gates are shown in fig.1.2. Each gate has one or two binary input variables, designated by x and y, and one binary output variable designated by F.

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Place IC gate 7400 quadruple two-input NAND gate on the breadboard.
2. Connect pin number 14 to +5V and pin number 7 to GND.
3. Connect the input pins of the IC gate to switches and output to an indicator lamp and obtain the truth table.
4. Compare the results with the truth table shown in fig. 1.2.
5. Repeat step1 to step4 for other logic gates.

OBSERVATIONS:

	x	y	F=x.y		x	y	F=x+y
	0	0			0	0	
AND	0	1		OR	0	1	
GATE	1	0		GATE	1	0	
	1	1			1	1	

	X	y	$F=(x.y)'$		x	y	$F=(x+y)'$
NAND GATE	0	0		NOR GATE	0	0	
	0	1			0	1	
	1	0			1	0	
	1	1			1	1	

	x	y	$F=xy'+x'y$		x	y	$F=xy+\overline{xy}$
EX-OR GATE	0	0		EX-NOR GATE	0	0	
	0	1			0	1	
	1	0			1	0	
	1	1			1	1	

	X	$F=x'$
NOT GATE	0	
	1	

RESULT:

Logic behavior of various IC gates is verified.

2. IMPLEMENTATION OF BOOLEAN FUNCTION USING BASIC GATES

AIM:

1. To simplify the Boolean function $F = x'y'z + x'yz + xy'$ using Boolean algebra
2. To implement Boolean function without simplification and after simplification using basic IC gates

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. 7404 hex inverters
4. 7408 quadruple two-input AND gates
5. 7432 quadruple two-input OR gates

THEORY

This experiment illustrates the relationship between the Boolean function and the corresponding logic diagram. The Boolean functions are simplified using Boolean algebra discussed in the section Boolean algebra.

Boolean Algebra

Because binary logic is used in all of today's digital computers and devices, the cost of the circuits that implement it is an important factor addressed by designers. Finding simpler and cheaper, but equivalent, realizations of a circuit can reap huge payoffs in reducing the over-all cost of the design. Mathematical methods that simplify circuits rely primarily on Boolean algebra.

A Boolean algebra is defined as a closed algebraic system containing a set K of two or more elements and the two operators, \cdot and $+$

- Identity elements

$$a + 0 = a \quad 0 \text{ is the identity element for the } + \text{ operation}$$

$$a \cdot 1 = a \quad 1 \text{ is the identity element for the } \cdot \text{ operation}$$

- Commutative Property:

For every 'a' and 'b' in K ,

$$a + b = b + a$$

$$a \cdot b = b \cdot a$$

- Associative Property:

For every 'a', 'b', and 'c' in K,

$$a + (b + c) = (a + b) + c$$

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

- Distributive Property:

For every 'a', 'b', and 'c' in K,

$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

- The Existence of the Complement:

For every 'a' in K there exists a unique element called a' (or \bar{a}) (complement of a) such that,

$$a + a' = 1$$

$$a \cdot a' = 0$$

- Involution:

$$(a')' = a$$

Remember that:

$$aa' = 0 \qquad a+a'=1$$

- Absorption:

$$a + ab = a \qquad a(a+b) = a$$

- DeMorgan's Theorem:

A key theorem in simplifying Boolean algebra expressions is DeMorgan's Theorem.

It states: $(a + b)' = a'b'$ $(ab)' = a' + b'$

$$a + b = a \cdot b' \qquad a \cdot b = a + b'$$

Boolean functions

Boolean algebra is an algebra that deals with binary variables and logic operations. A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols. For a given value of the binary variables, the function can be equal to either 1 or 0.

As an example, consider the Boolean function $F_1 = x + y'z$

The function F_1 is equal to 1 if x is equal to 1 or if both y' and z are equal to 1. F_1 is equal to 0 otherwise. The complement operation dictates that when $y' = 1$, $y = 0$. Therefore, $F_1 = 1$ if $x = 1$ or if $y = 0$ and $z = 1$. A Boolean function expresses the logical relationship between binary variables and is evaluated by determining the binary value of the expression for all possible values of the variables

Truth Table

A Boolean function can be represented in a truth table. The number of rows in the truth table is 2^n , where n is the number of variables in the function. The binary combinations for the truth table are obtained from the binary numbers by counting from 0 through $2^n - 1$. Table 2.1 shows the truth table for the function F_1 considered in the earlier section.. There are eight possible binary combinations for assigning bits to the three variables x , y , and z . The column labeled F_1 contains either 0 or 1 for each of these combinations. The table shows that the function is equal to 1 when $x = 1$ or when $yz = 01$ and is equal to 0 otherwise.

Table 2.1: Truth table of the Boolean function F_1

x	Y	z	F1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Circuit Diagram

A Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

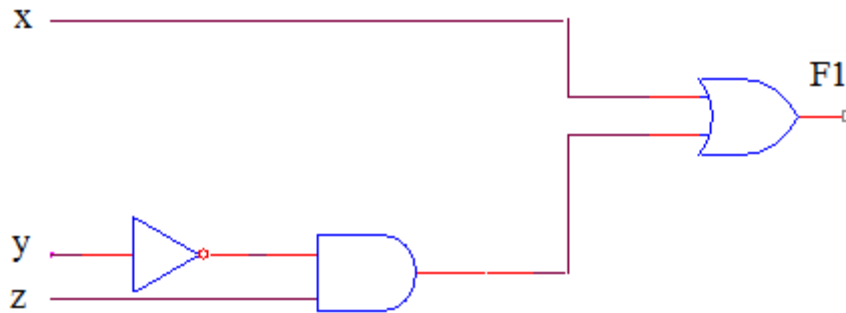


Fig. 2.1: Circuit diagram drawn from the Boolean function F_1 .

The logic circuit diagram (also called a schematic) for F_1 is shown in Fig. 2.1. There is an inverter for input y to generate its complement. There is an AND gate for the term $y'z$ and an OR gate that combines x with $y'z$. In logic-circuit diagrams, the variables of the function are taken as the inputs of the circuit and the binary variable F_1 is taken as the output of the circuit. The particular expression used to represent the function will dictate the interconnection of gates in the logic-circuit diagram. Here is a key fact that motivates the use of Boolean algebra: By manipulating a Boolean expression according to the rules of Boolean algebra, it is sometimes possible to obtain a simpler expression for the same function and thus reduce the number of gates in the circuit and the number of inputs to the gate which leads to reduction in the cost of the circuit.

Function Minimization

Using Boolean algebra the given Boolean function is to be minimized as follows

$$F = x'y'z + x'yz + xy' \text{ (equation 1)}$$

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

$$F = x'z + xy' \text{ (equation 2)}$$

Circuit Diagram

Circuit diagrams are drawn from the Boolean function without simplification (equation 1) and with simplification (equation 2) using Boolean algebra is shown in fig. 2.2 and fig. 2.3 respectively.

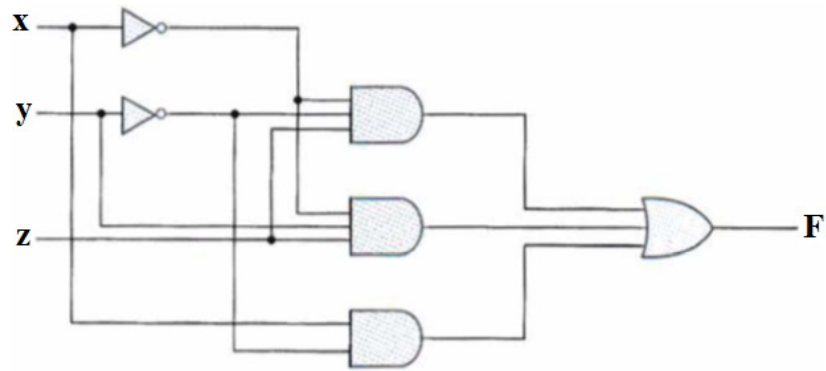


Fig. 2.2: circuit diagram of the Boolean function without simplification using basic gates

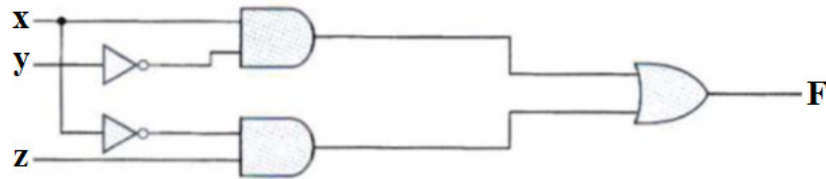


Fig. 2.3: circuit diagram of the Boolean function with simplification using basic gates

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the circuit diagram shown in fig. 2.2 using basic gates.
2. Obtain the truth table by connecting the inputs of the circuit shown in fig.2.2 to switches and the output to an indicator lamp.
3. Connect the circuit diagram shown in fig. 2.3 using basic gates without disturbing the earlier circuit.
4. Obtain the truth table by connecting the inputs of the circuit shown in fig.2.3 to switches and the output to an indicator lamp.
5. Compare the output of both the circuits for each of the possible input combinations.

OBSERVATIONS

Inputs of the circuits			Outputs of the circuits	
x	y	Z	F(without simplification)	F(with simplification)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

RESULT:

Output of each circuit implementing the Boolean function with and without simplification is verified and found to be same.

3. IMPLEMENTATION OF BOOLEAN FUNCTION USING UNIVERSAL GATES

AIM:

1. To simplify the Boolean function $F(A, B, C, D) = \Sigma(3, 7, 8, 12, 13, 15) + d(9, 14)$ using K-Map method.
2. To implement simplified Boolean function using Universal Gates.

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. 7400 quadruple two-input NAND gates

THEORY

In this experiment the given Boolean function is simplified using the K-map method and the simplified Boolean function is to be implemented using only 7400 ICs,

K-Map Method

The complexity of the digital logic circuits that implement a Boolean function is directly related to the complexity of the algebraic expression from which the function is implemented. Although the truth table representation of a function is unique, when it is expressed algebraically it can appear in many different, but equivalent, forms. Boolean expressions may be simplified using Boolean algebra. However, this procedure of minimization is awkward because it lacks specific rules to predict each succeeding step in the manipulative process.

		BC			
		00	01	11	10
A	0	0	1	0	1
	1	1	1	1	1

(a)

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(b)

Fig. 3.1 : Representation of Boolean function using (a) K-Map (b) Truth table

The map method presented here provides a simple, straightforward procedure for minimizing Boolean functions. This method may be regarded as a pictorial form of a truth table. The K-Map representation of the truth table is as shown in fig.3.1.

Four-Variable Map

A four variable map is shown in Fig. 3.2, There are sixteen minterms for four binary variables; therefore, the map consists of sixteen squares. The map method is also known as the Karnaugh map or K-map. A K-map is a diagram made up of squares, with each square representing one minterm of the function that is to be minimized. Note that the minterms are arranged, not in a binary sequence, but in a sequence similar to the Gray code. The characteristic of this sequence is that only one bit changes in value from one adjacent column to the next. To understand the usefulness of the map in simplifying Boolean functions, we must recognize the basic property possessed by adjacent squares: Any two adjacent squares in the map differ by only one variable, which is primed in one square and unprimed in the other.

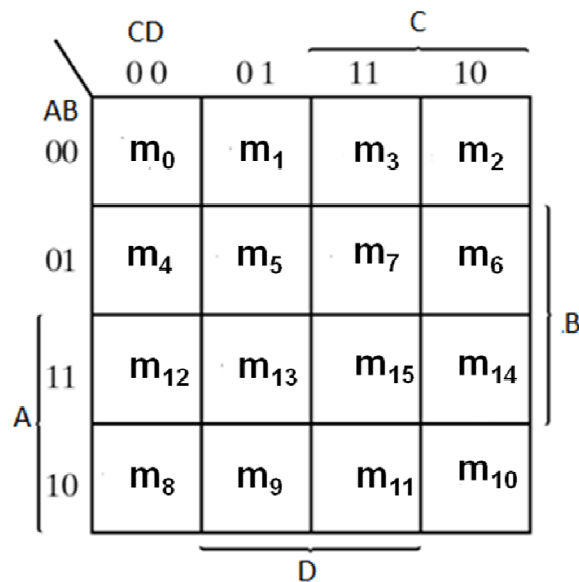


Fig.3.2 : Four variable K-map

Since any Boolean function can be expressed as a sum of minterms, it follows that a Boolean function is recognized graphically in the map from the area enclosed by those squares whose minterms are included in the function. The combination of adjacent squares that is useful during the simplification process is easily determined from inspection of the four variable map:

One square represents one minterm, giving a term with four literals

Two adjacent squares represent a term with three literals.

Four adjacent squares represent a term with two literals.

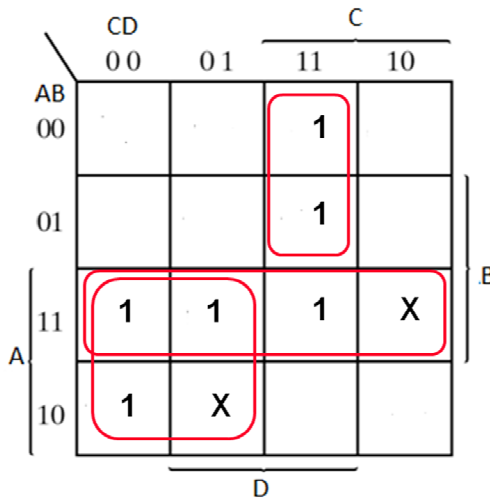
Eight adjacent squares represent a term with one literal.

Sixteen adjacent squares produce a function that is always equal to 1.

No other combination of squares can simplify the function

FUNCTION MINIMIZATION USING K-MAP

Given function is $F(A, B, C, D) = \Sigma(3, 7, 8, 12, 13, 15) + d(9, 14)$



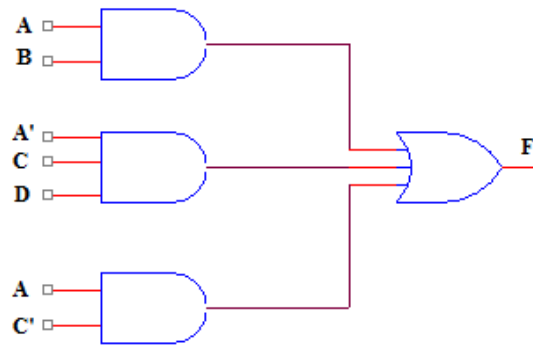
$$F = A'CD + AB + AC$$

TWO-LEVEL IMPLEMENTATION

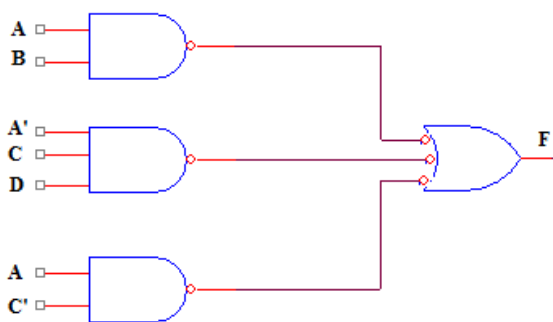
The implementation of Boolean functions with NAND gates requires that the functions be in sum-of-products form. The Boolean function simplified using K-Map is given as

$$F = A'CD + AB + AC$$

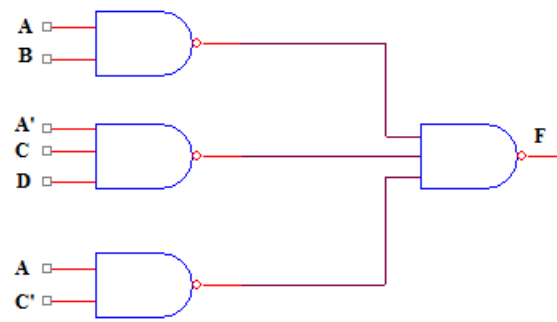
The function is implemented in (a) with AND and OR gates. In (b), the AND gates are replaced by NAND gates and the OR gate is replaced by a NAND gate with an OR-invert graphic symbol. A bubble denotes complementation and two bubbles along the same line represent double complementation, so both can be removed. Removing the bubbles on the gates of (b) produces the circuit of (a). Therefore, the two diagrams implement the same function and are equivalent.



(a)



(b)



(c)

Fig. 3.3: Three ways to implement the Boolean function F

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Implement the circuit diagram shown in fig. 3.3(c) using NAND gates.
2. Obtain the truth table by connecting the inputs of the circuit to switches and the output to an indicator lamp.

OBSERVATIONS:

Inputs of the circuits				Output
A	B	C	D	F
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	

RESULT:

The given Boolean function is simplified using K-Map method and implemented using only NAND gates.

4. ADDERS AND SUBTRACTORS

AIM:

1. To implement the Half adder and Full adder circuits and test for proper operation
2. To implement the Half subtractor and Full subtractor circuit and test for proper operation.

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. 7404 hex inverters
4. 7408 quadruple two-input AND gates
5. 7432 quadruple two-input OR gates
6. 7486 quadruple two-input XOR gates

THEORY

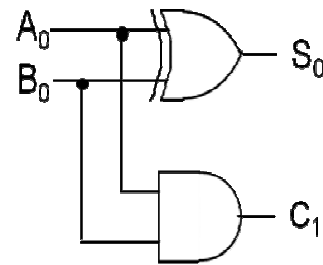
Half Adder

Half adder adds two binary numbers A_0, B_0

Where $A_0, B_0 \rightarrow$ single bit inputs
 $S_0 \rightarrow$ single bit sum
 $C_1 \rightarrow$ carry out

Truth Table

A_0	B_0	S_0	C_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

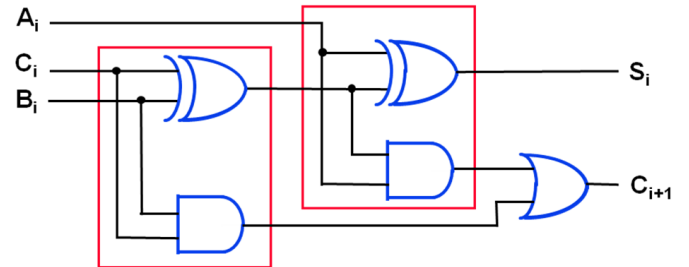


Full Adder using two half adders

Full adder includes a carry-in C_i

Truth Table

C_i	A_i	B_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$S_i = C_i' A_i' B_i + C_i' A_i B_i' + C_i A_i' B_i + C_i A_i B_i$$

$$= C_i' (A_i' B_i + A_i B_i') + C_i (A_i' B_i + A_i B_i)$$

$$= C_i \oplus A_i \oplus B_i$$

$$C_{i+1} = C_i B_i + C_i' A_i B_i + C_i A_i B_i'$$

$$= C_i B_i + A_i (C_i' B_i + C_i B_i')$$

$$= C_i B_i + A_i (C_i \oplus B_i)$$

Half Subtractor

Half subtractor subtracts two binary numbers A_0, B_0

Where $A_0, B_0 \rightarrow$ single bit inputs

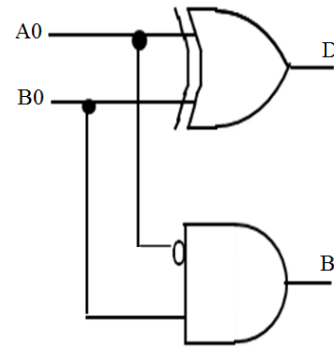
$D \rightarrow$ Difference

$B \rightarrow$ Borrow

Half Subtractor

Truth Table

A_0	B_0	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

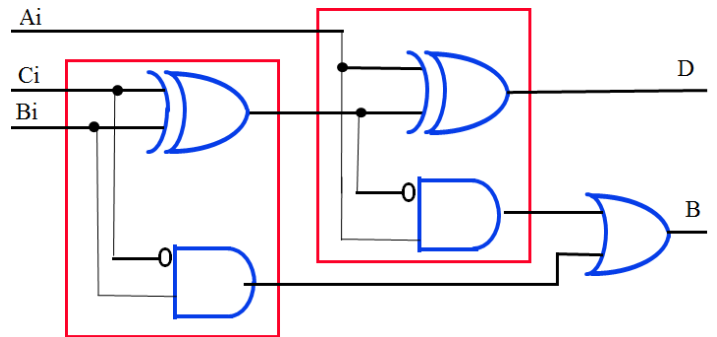


Full Subtractor using two half subtractors

Full subtractor includes a borrow C_i

Truth Table

C_i	A_i	B_i	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



$$\begin{aligned}
 D &= C_i' A_i' B_i + C_i' A_i B_i' + C_i A_i' B_i' + C_i A_i B_i \\
 &= C_i' (A_i' B_i + A_i B_i') + C_i (A_i' B_i' + A_i B_i) \\
 &= C_i \oplus A_i \oplus B_i \\
 B &= C_i' B_i + C_i' A_i B_i' + C_i B_i A_i \\
 &= C_i' B_i + A_i (C_i' B_i' + C_i B_i) \\
 &= C_i' B_i + A_i (C_i' B_i + C_i B_i')'
 \end{aligned}$$

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the circuit diagram of Half adder and verify its truth table.
2. Connect the circuit diagram of Full adder using two half adders and verify the truth table of Full adder circuit.
3. Connect the circuit diagram of Half subtractor and verify its truth table.
4. Connect the circuit diagram of Full subtractor using two half subtractors and verify the truth table of Full subtractor circuit.

OBSERVATIONS:

Tabulate the truth tables of Half adder, Full adder , Half subtractor and Full subtractor.

RESULTS:

The adder and subtractor circuits were implemented using basic gates and tested for their operation.

5. Four-bit Binary Parallel Adder

AIM:

1. To implement 4-bit binary parallel adder-subtractor circuit using IC type 74LS83 and verify its functionality.

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. 4-bit binary parallel adder IC type 74LS83
4. 7486 quadruple two-input XOR gates

THEORY

4-Bit Binary Parallel Adder

IC type 7483 is a four-bit binary parallel adder. The pin assignment is shown in Fig. 6.1

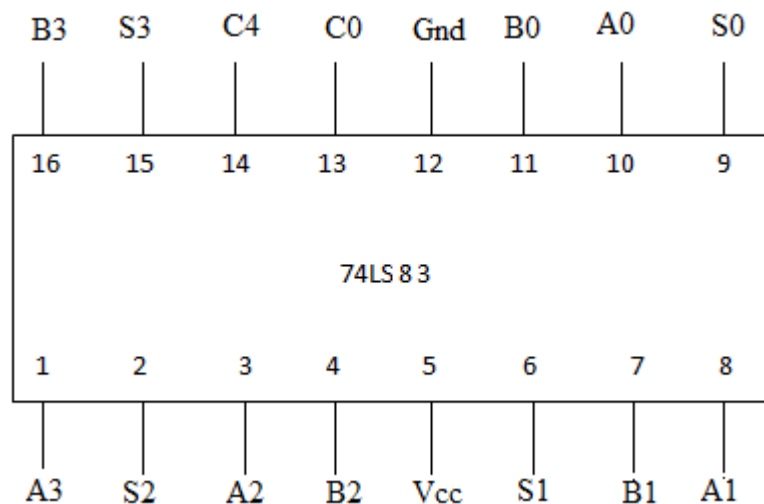


Fig. 5.1: 4-bit binary parallel adder

The 2 four-bit input binary numbers are A0 through A3 and B0 through B4. The four-bit sum is obtained from S0 through S4. C0 is the input carry and C4 the output carry, Test the four-bit binary adder 7483 by connecting the power supply and ground terminals. Then connect the four A inputs, B inputs and the input carry to toggle switches. The five outputs are applied to indicator lamps. Perform the addition of a few binary numbers and check that the output sum and output carry give the proper values. when the input carry is equal to 1, it adds 1 to the output sum.

4-Bit Binary Adder Subtractor Circuit

Two binary numbers can be subtracted by taking the 2's complement of the subtrahend adding it to the minuend. The 2's complement can be obtained by taking the 1's complement and adding 1. To perform $A - B$, we complement the four bits of B , add them to the four bits of A , and add 1 through the input carry. This is done as shown in Fig. 6.2, The four XOR gates complement the bits of B when the mode select $M = 1$ and leave the bits of B unchanged when $M = 0$. Thus, when the mode select M is equal to 1, the input carry C_0 is equal to 1 and the sum output is A plus the 2's complement of B . When M is equal to 0, the input carry is equal to 0 and the sum generates $A + B$.

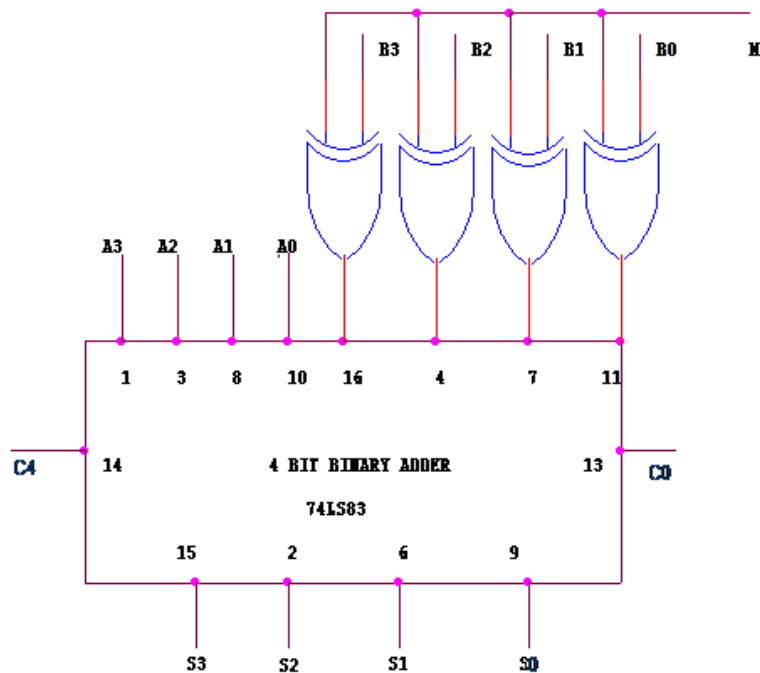


Fig.5.2 : 4-bit binary parallel adder-subtractor circuit

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the circuit diagram as shown in fig.5.2.
2. Connect the four A inputs, B inputs and the input carry as shown in the observation table.
3. The five outputs are applied to indicator lamps.
4. Perform the addition of a few binary numbers given in the observation table by making M=0 and check for proper values of the output sum and output carry
5. Perform the subtraction of a few binary numbers given in the observation table by making M=1 and check for proper values of the output sum and output carry

OBSERVATIONS:

INPUTS								OUT PUTS						
M	C0	A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0	C4
0	0	1	1	1	1	1	1	0	0					
0	1	1	1	1	1	1	1	0	0					
1	0	1	1	1	1	1	1	0	0					
1	1	1	1	1	1	1	1	0	0					
0	0	1	1	0	1	0	1	1	1					
0	1	1	1	0	1	0	1	1	1					
1	0	1	1	0	1	0	1	1	1					
1	1	1	1	0	1	0	1	1	1					

RESULTS:

The operation of the 4-bit binary parallel adder is verified and 4-bit binary parallel adder-subtractor circuit is designed and implemented for a given binary numbers.

6. Four-bit Magnitude Comparator circuit.

AIM:

1. To implement 4-bit Magnitude Comparator circuit using IC type 74LS83 and verify its functionality.

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. 4-bit binary parallel adder IC type 74LS83
4. 7402 quadruple two-input NOR gates
5. 7404 hex inverters
6. 7408 quadruple two-input AND gates
7. 7432 quadruple two-input OR gates
8. 7486 quadruple two-input XOR gates

THEORY

4-Bit Binary Parallel Adder

IC type 7483 is a four-bit binary parallel adder. The pin assignment is shown in Fig. 6.1

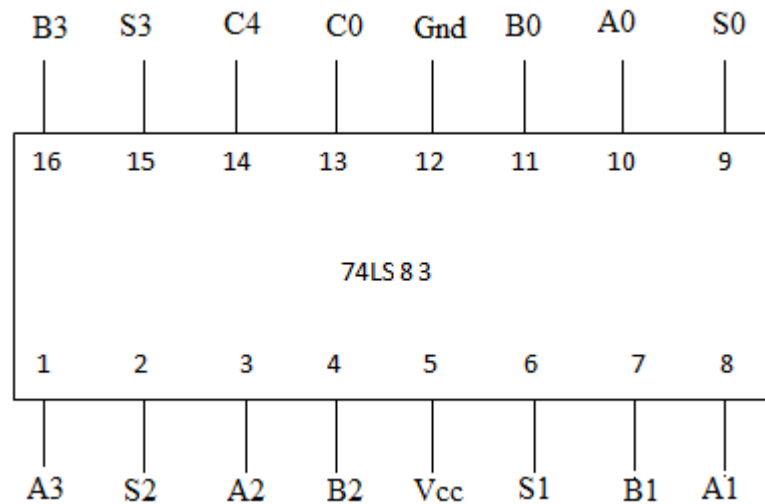


Fig. 6.1: 4-bit binary parallel adder

The 2 four-bit input binary numbers are A0 through A3 and B0 through B4. The four-bit sum is obtained from S0 through S4. C0 is the input carry and C4 the output carry, Test the four-bit binary adder 7483 by connecting the power supply and ground terminals. Then connect the four A inputs, B inputs and the input carry to toggle switches. The five outputs are applied to indicator lamps. Perform the addition of a few binary numbers and

check that the output sum and output carry give the proper values. when the input carry is equal to 1, it adds 1 to the output sum.

4-Bit Binary Adder Subtractor Circuit

Two binary numbers can be subtracted by taking the 2's complement of the subtrahend adding it to the minuend. The 2's complement can be obtained by taking the 1's complement and adding 1. To perform $A - B$, we complement the four bits of B, add them to the four bits of A, and add 1 through the input carry. This is done as shown in Fig. 6.2, The four XOR gates complement the bits of B when the mode select $M = 1$ and leave the bits of B unchanged when $M = 0$. Thus, when the mode select M is equal to 1, the input carry C_0 is equal to 1 and the sum output is A plus the 2's complement of B. When M is equal to 0, the input carry is equal to 0 and the sum generates $A + B$.

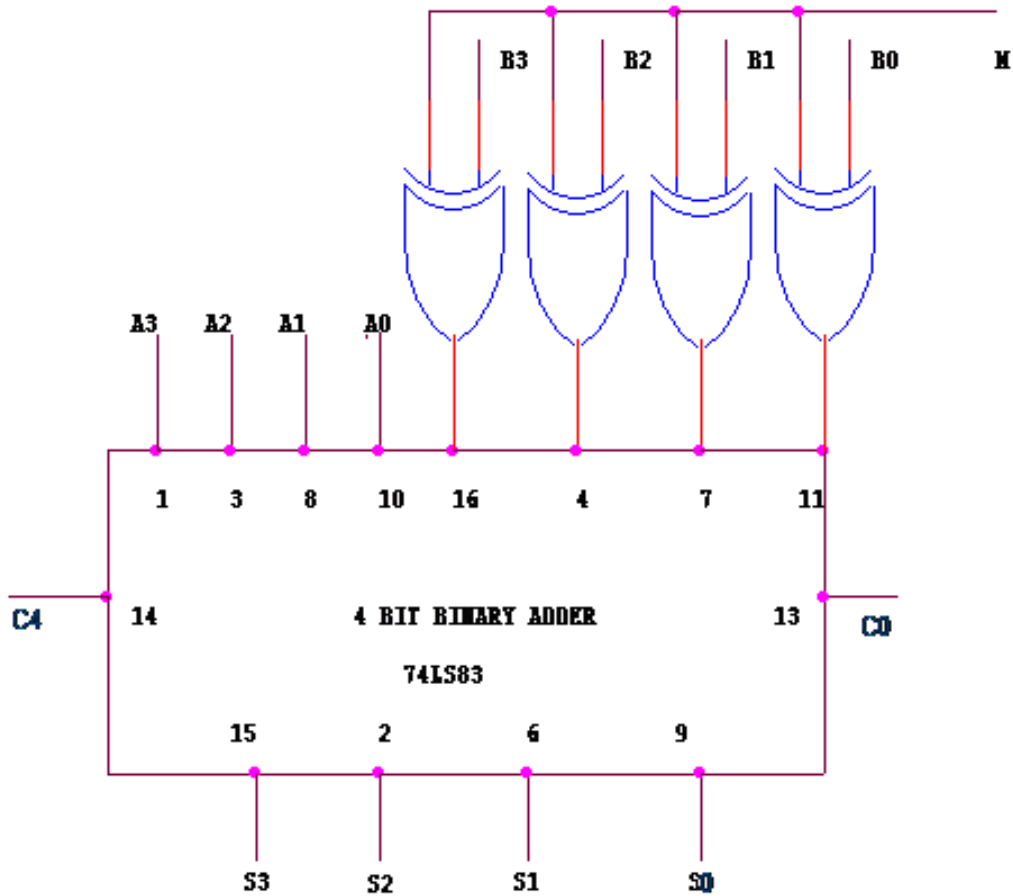


Fig.6.2 : 4-bit binary parallel adder-subtractor circuit

Magnitude Comparator

The comparison of two 4-bit binary numbers is an operation that determines whether one number is greater than, equal to, or less than the other number. Two numbers, A and B can be compared by first subtracting A-B as is done in Fig.6.2. If the output $S(S_3, S_2, S_1, S_0)$ is equal to zero, then $A=B$. The output carry from C_4 determines the relative magnitudes of the numbers: When $C_4=1$ $A \geq B$; when $C_4=0$, $A < B$; and when $C_4=1$ and $S \neq 0$, $A > B$.

It is necessary to supplement the subtractor circuit of Fig. 6.2 to construct the 4-bit magnitude comparator circuit. This is done with a combinational circuit as shown in fig.6.3 that has five inputs S_1 through S_4 and C_4 & and three outputs, designated by X, Y, and Z, so that

$$X=1 \quad \text{if}(A>B) \quad (C_4=1 \text{ and } S \neq 0000)$$

$$Y=1 \quad \text{if}(A<B) \quad (C_4=0)$$

$$Z=1 \quad \text{if}(A=B) \quad (S=0000)$$

4-bit magnitude comparator shown in fig.6.3 has five inputs S_1 through S_4 and C_4 taken from the output of the 4-bit binary adder-subtractor circuit shown in fig.6.2. The three outputs designated by X, Y and Z are used to indicate $A > B$, $A < B$ and $A = B$ of any of the two 4-bit binary numbers.

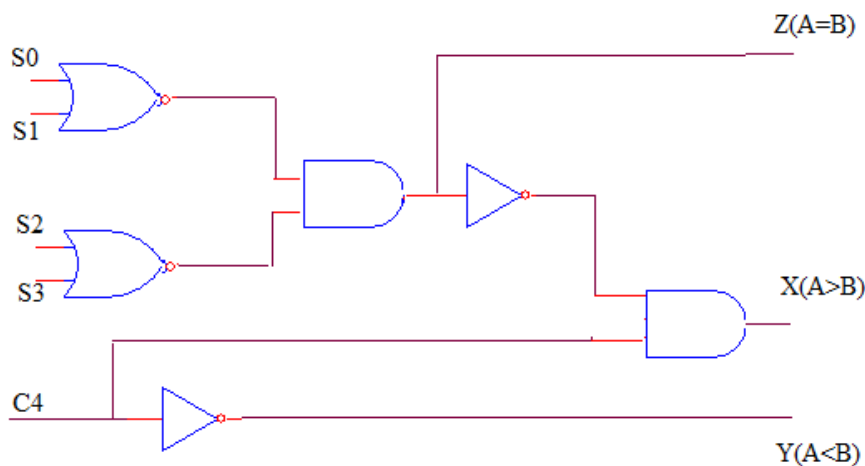


Fig. 6.3: 4-bit binary magnitude comparator

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the circuit diagram as shown in fig.6.2.
2. Connect the four A inputs, B inputs and the input carry to the logic switches and five outputs to indicator lamps.
3. Connect the circuit diagram as shown in fig.6.3 with five inputs taken from the outputs of the circuit shown in fig.6.2 and the three outputs designated by X, Y and Z are to be connected to the indicator lamps.
4. Verify the operation of the comparator circuit by using atleast two sets of numbers for A and B to check each of the outputs X, Y, and Z in the observation table.

OBSERVATIONS:

INPUTS								OUT PUTS									
								4-bit binary adder subtractor				4-bit magnitude comparator					
M	C0	A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0	C4	X	Y	Z
1	0	1	1	1	1	1	1	0	0								
1	1	1	1	1	1	1	1	0	0								
1	0	1	1	0	1	0	1	1	1								
1	1	1	1	0	1	0	1	1	1								
1	0	1	1	1	1	1	1	1	1								
1	1	1	1	1	1	1	1	1	1								

RESULTS

4-bit binary magnitude comparator is implemented using 4-bit binary parallel adder and tested for a given two binary numbers.

7. DESIGN WITH MULTIPLEXER.

AIM:

To construct 8:1 Mux using two 4:1 Muxs and verify its functionality.

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. Multiplexer IC type 74LS153
4. 7432 quadruple two input OR gate

THEORY

A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

IC type 74LS153

The pin assignment of IC type 74LS153 is as shown in Fig. 7.1.

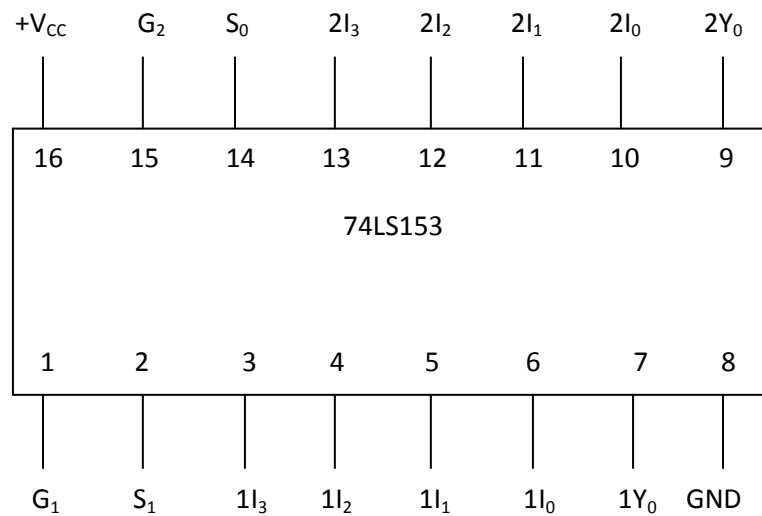


Fig. 7.1: Dual 4:1 Multiplexer

Construction Of 8:1 Mux Using Two 4:1 Muxs

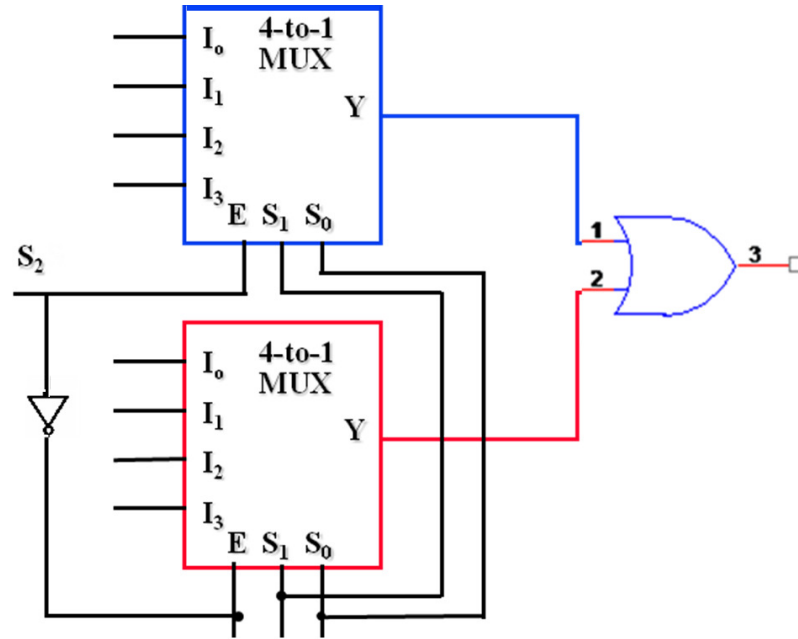
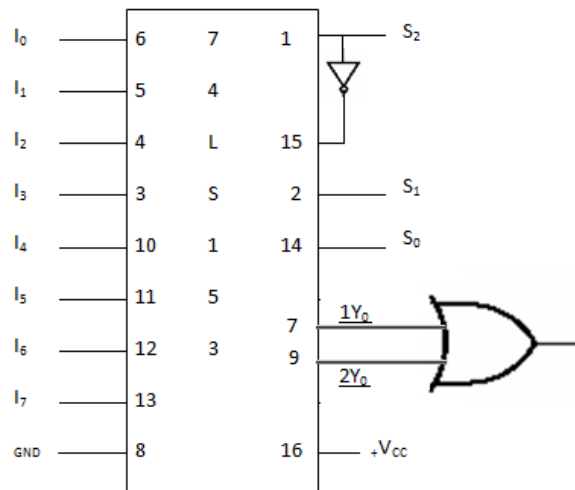


Fig. 7.2: Construction of 8:1 Mux using two 4:1 Mux and one OR gate

CIRCUIT DIAGRAM:

The circuit diagram for constructing 8:1 Multiplexer using two 4:1 multiplexers and an OR gate is as shown in fig.7.3



(a)

Fig.7.3: Implementation of (a) 8:1 Mux

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Make the connections as per the circuit diagram shown in fig.7.3(a) to construct 8:1 Mux using two 4:1 Mux and an OR gate.
2. Verify the truth table of the 8:1 Mux

OBSERVATIONS:

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	S_2	S_1	S_0	Y
0	X	X	X	X	X	X	X	0	0	0	
1	X	X	X	X	X	X	X	0	0	0	
X	0	X	X	X	X	X	X	0	0	1	
X	1	X	X	X	X	X	X	0	0	1	
X	X	0	X	X	X	X	X	0	1	0	
X	X	1	X	X	X	X	X	0	1	0	
X	X	X	0	X	X	X	X	0	1	1	
X	X	X	1	X	X	X	X	0	1	1	
X	X	X	X	0	X	X	X	1	0	0	
X	X	X	X	1	X	X	X	1	0	0	
X	X	X	X	X	0	X	X	1	0	1	
X	X	X	X	X	1	X	X	1	0	1	
X	X	X	X	X	X	0	X	1	1	0	
X	X	X	X	X	X	1	X	1	1	0	
X	X	X	X	X	X	X	0	1	1	1	
X	X	X	X	X	X	X	1	1	1	1	

RESULT:

8:1 Mux is constructed using two 4:1 Muxs and its truth table is verified.

8. APPLICATIONS OF MULTIPLEXER.

AIM:

To implement the Full adder using multiplexer of IC type 74LS153.

APPARATUS:

1. Digital IC trainer kit
2. Connecting wires
3. Multiplexer IC type 74LS153
4. 7432 quadruple two input OR gate

THEORY

A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

IC type 74LS153

The pin assignment of IC type 74LS153 is as shown in Fig. 8.1.

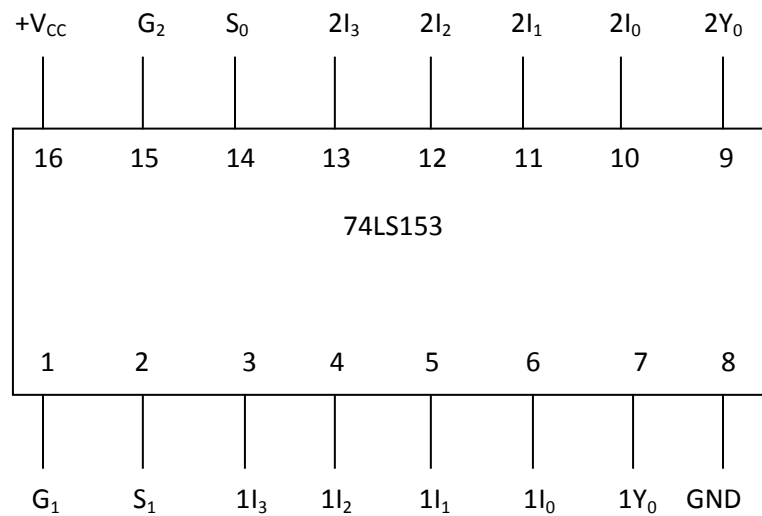


Fig. 8.1: Dual 4:1 Multiplexer

Implementation Of Full Adder Using 4:1 Multiplexer

Full Adder			Truth Table	
Inputs			Outputs	
A	B	C	Sum	carry
0	0	0	C	0
0	0	1		
0	1	0	C'	C
0	1	1		
1	0	0	C'	C
1	0	1		
1	1	0	C	1
1	1	1		

CIRCUIT DIAGRAM:

The circuit diagram for implementing Full adder circuit using IC type 74LS 153 is as shown in fig.8.2

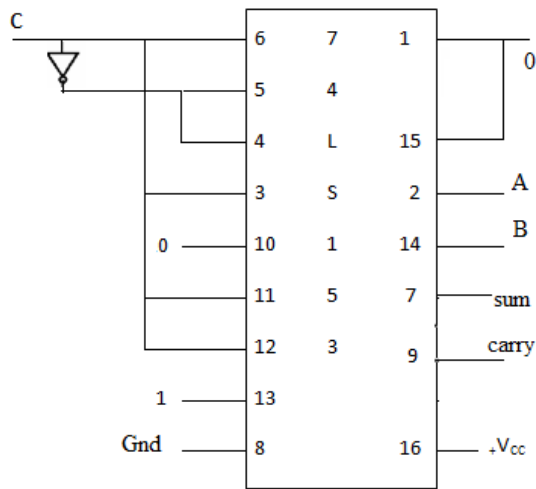


Fig.8.2: Implementation of Full adder using 4:1

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Make the connections as per the circuit diagram shown in fig.8.3 to implement Full adder using IC type 74LS 153
4. Verify the truth table of the Full adder.

OBSERVATIONS:

Full Adder

Inputs			Outputs	
A	B	C	Sum	carry
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

RESULT:

Full adder is implemented using two 4:1 Muxs and the verified.

9. FLIP-FLOPS

AIM:

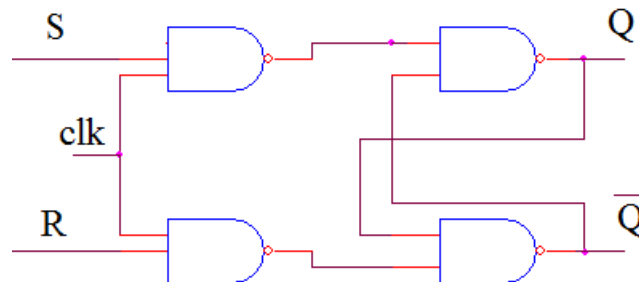
To construct and verify the characteristics of the following flip-flops using logic gates to understand their operation.

- i) S-R Flip-Flop
- ii) D- Flip-Flop
- iii) J-K Flip-Flop(Master-Slave of JK FF)
- iv) T-Flip-Flop

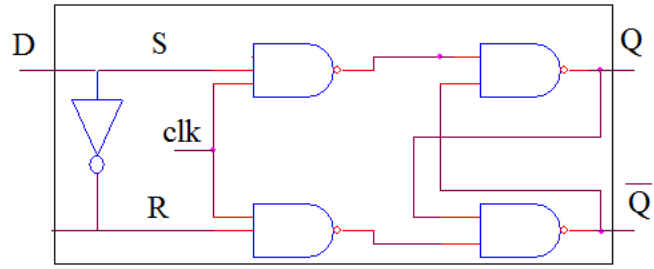
APPARATUS:

1. Digital IC trainer Kit
2. Connecting wires
3. 7404 hex inverters
4. 7400 quadruple two-input NAND gates
5. 7410 three-Input NAND Gates

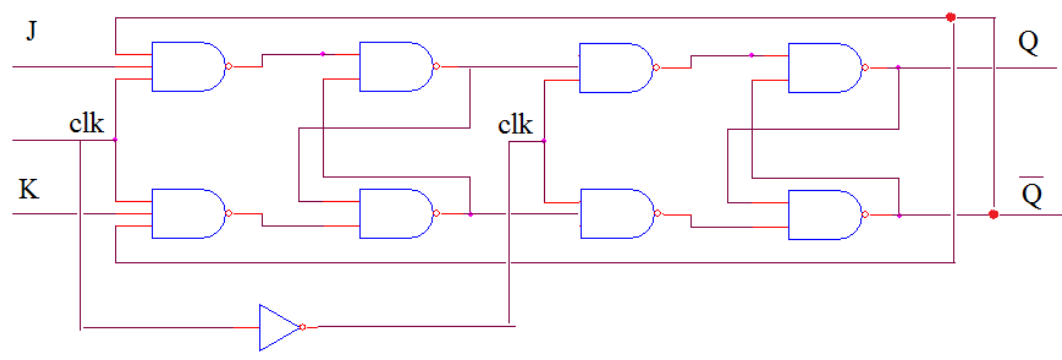
Circuit Diagram And Their Characteristic Tables Of Various Flip-Flops



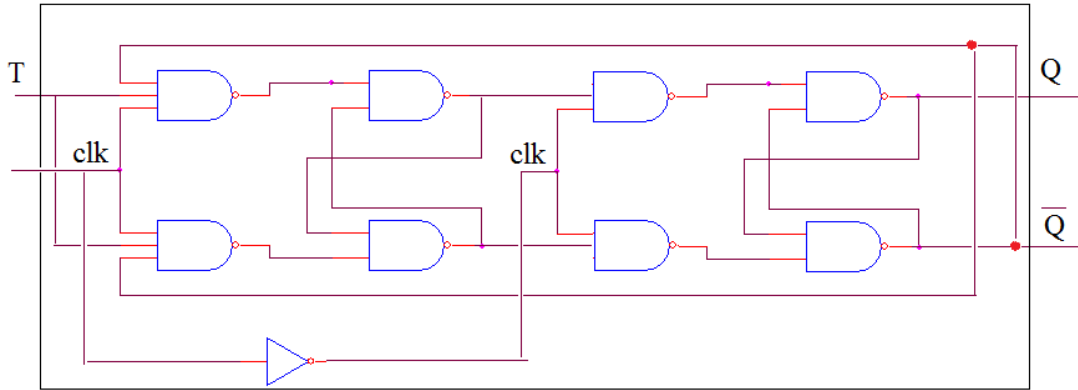
Clk	S	R	Q	Q'
0	X	X	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	NOT used	



Clk	D	Q	Q'
0	X	Memory	
1	0	0	1
1	1	1	0



Clk	J	K	Q	Q'
L	X	X	Memory	
H	X	X	Memory	
↓	0	0	Memory	
↓	0	1	0	1
↓	1	0	1	0
↓	1	1	Q'	Q



Clk	T	Q	Q'
X	0	Memory	
↓	1	Q'	Q

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the Circuit diagram of the each of the flip flop using NAND gates
2. Connect the inputs of the each circuit to logic switches and the outputs to indicator lamps.
3. Tabulate the observation tables and verify the characteristic of various Flip-flops.

OBSERVATION TABLES:

SR-FLIPFLOP

Clk	S	R	Q	Q _{n+1}
0	X	X	0	
0	X	X	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	
X	1	1	NOT used	

T-FLIPFLOP

Clk	T	Q	Q _{n+1}
X	0	0	
X	0	1	
↓	1	0	
↓	1	1	

JK-FLIPFLOP

Clk	J	K	Q	Q _{n+1}
X	0	0	0	
X	0	0	1	
↓	0	1	0	
↓	0	1	1	
↓	1	0	0	
↓	1	0	1	
↓	1	1	0	
↓	1	1	1	

D-FLIPFLOP

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

RESULT:

The characteristic table of various flip-flops is verified

10. DESIGN OF RIPPLE COUNTER

AIM:

1. To Design a 4-bit ripple counter for the given specification
2. To implement ripple counter and test their logic states

APPARATUS:

1. Digital IC trainer Kit
2. Connecting wires
3. Dual J– K FLIP-FLOPS (7473)–2Nos

IC type 7473

IC type 7473 is a dual JK Flip-flop. The pin assignment is shown in Fig. 10.1

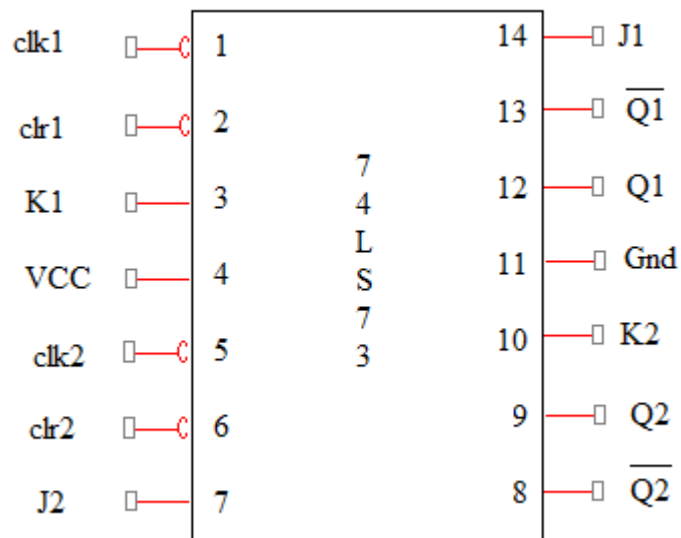


Fig.10.1:Pin diagram of dual JK-Flip-flop

CIRCUIT DIAGRAM

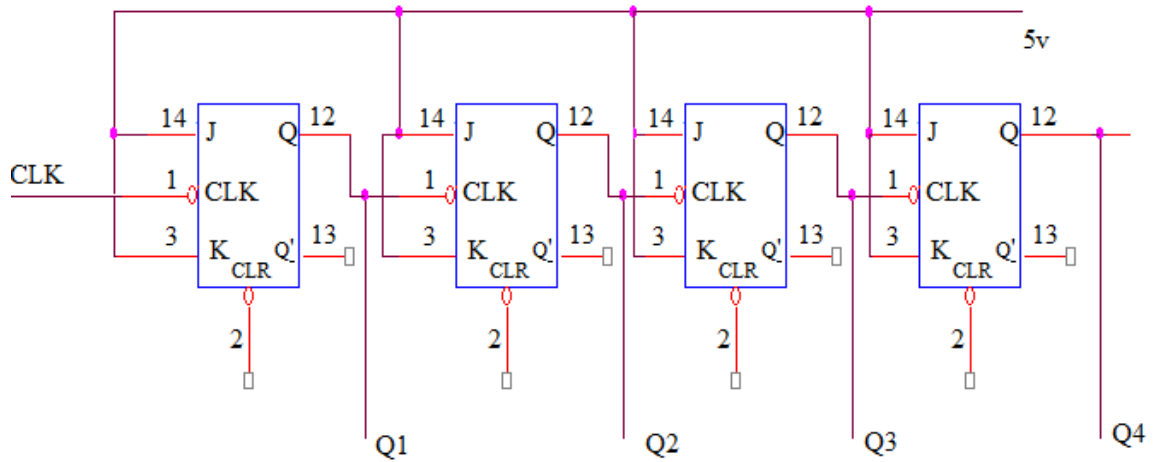


Fig.10.2: 4-bit binary ripple counter

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the Circuit diagram as shown in fig.10.2 using IC type 7473
2. Apply the clock and observe the logic states of Q1,Q2,Q3 and Q4

RESULT:

4-bit binary ripple counter is implemented using JK flip-flops and tested their logic states.

11. DESIGN OF MOD-K SYNCHRONOUS COUNTER

AIM:

1. To Design a MOD-6 synchronous counter using JK flip-flops.
2. To implement MOD-6 synchronous counter and test their logic states

APPARATUS:

1. Digital IC trainer Kit
2. Connecting wires
3. 7408 quadruple two-input AND gates
4. Dual J– K FLIP-FLOPS (7473)–2Nos

THEORY:

The design of a clocked sequential circuit starts from a set of specifications and culminates in a logic diagram or a list of Boolean functions from which the logic diagram can be obtained. In contrast to a combinational circuit, which is fully specified by a truth table, a sequential circuit requires a state table for its specification. The first step in the design of sequential circuits is to obtain a state table or an equivalent representation, such as a state diagram.

A synchronous sequential circuit is made up of flip-flops and combinational gates. The design of the circuit consists of choosing the flip-flops and then finding a combinational gate structure that, together with the flip-flops, produces a circuit which fulfills the stated specifications. The number of flip-flops is determined from the number of states needed in the circuit. The combinational circuit is derived from the state table by evaluating the flipflop input equations and output equations.

DESIGN OF MOD-6 SYNCHRONOUS COUNTER

State table :

Present state			Nest state			Excitation Values					
Q _C	Q _B	Q _A	Q _C	Q _B	Q _A	J _C	K _C	J _B	K _B	J _A	K _A
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	0	X	X	1	X	1

Excitation Table for J – K Flip-Flop

Q _n	Q _{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Simplification of flip-flop inputs using K-map

	Q _B Q _A					Q _B Q _A			
Q _C	00	01	11	10	Q _C	00	01	11	10
0	1	X	X	1	0	X	1	1	1
1	1	X	X	X	1	X	1	X	X
	J_A = 1					K_A = 1			

	$Q_B Q_A$					$Q_B Q_A$			
Q_C	00	01	11	10	Q_C	00	01	11	10
0	0	1	X	X	0	X	X	1	0
1	0	0	X	X	1	X	X	X	X

$J_B = \bar{Q}_C Q_A$
 $K_B = Q_A$

	$Q_B Q_A$					$Q_B Q_A$			
Q_C	00	01	11	10	Q_C	00	01	11	10
0	0	0	1	0	0	X	X	X	X
1	X	X	X	X	1	0	1	X	X

$J_C = Q_A Q_B$
 $K_C = Q_A$

CIRCUIT DIAGRAM

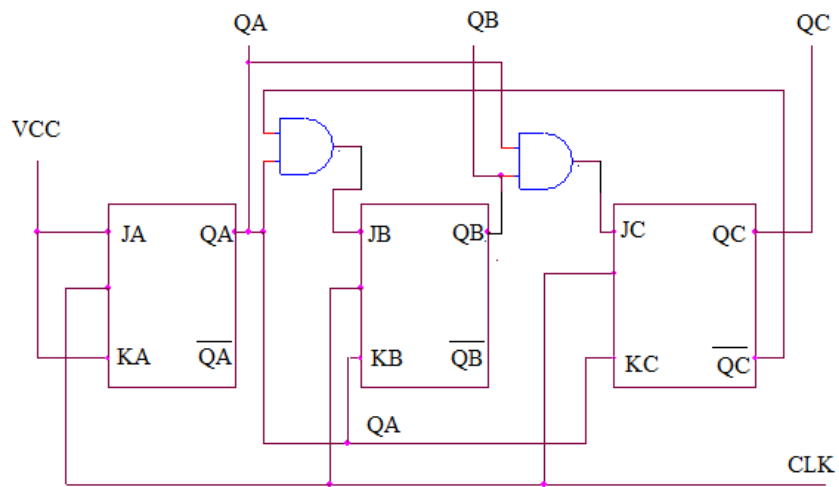


Fig.11.1:MOD-6 synchronous counter

IC type 7473

IC type 7473 is a dual JK Flip-flop. The pin assignment is shown in Fig. 11.2

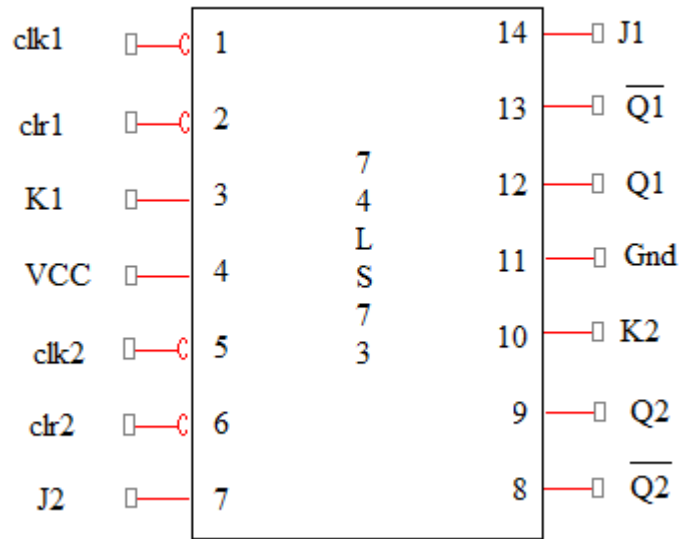


Fig.11.2:Pin diagram of dual JK Flip-flop

PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connect the Circuit diagram as shown in fig.11.1 using IC type 7473
2. Apply the clock and observe the logic states of QA, QB and QC

RESULT:

Mod-6 synchronous counter is designed using JK flip-flop, implemented and tested to verify their logic states.

12. UNIVERSAL SHIFT REGISTERS

AIM:

To verify the operation of universal shift register.

APPARATUS:

1. Digital IC trainer Kit
2. Connecting wires
3. Universal Shift Register(74LS194)

THEORY:

A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction, is called a shift register. The logical configuration of a shift register consists of a chain of flip-flops in cascade, with the output of one flip-flop connected to the input of the next flip-flop. All flip-flops receive common clock pulses, which activate the shift of data from one stage to the next.

If the flip-flop outputs of a shift register are accessible, then information entered serially by shifting can be taken out in parallel from the outputs of the flip-flops. If a parallel load capability is added to a shift register, then data entered in parallel can be taken out in serial fashion by shifting the data stored in the register.

A register capable of shifting in one direction only is a unidirectional shift register. One that can shift in both directions is a bidirectional shift register. If the register has both shifts and parallel-load capabilities, it is referred to as a universal shift register.

These bidirectional shift registers are designed to incorporate virtually all of the features a system designer may want in a shift register. The features are parallel input and parallel outputs, right shift and left shift serial input, operating-mode-control input's, and a direct overriding clear line. The register has four distinct modes of operation namely.

Parallel(broadside)load

Shift right(in the direction Q_A toward Q_D)

Shift left(in the direction Q_D toward Q_A)

Inhibit clock(do nothing)

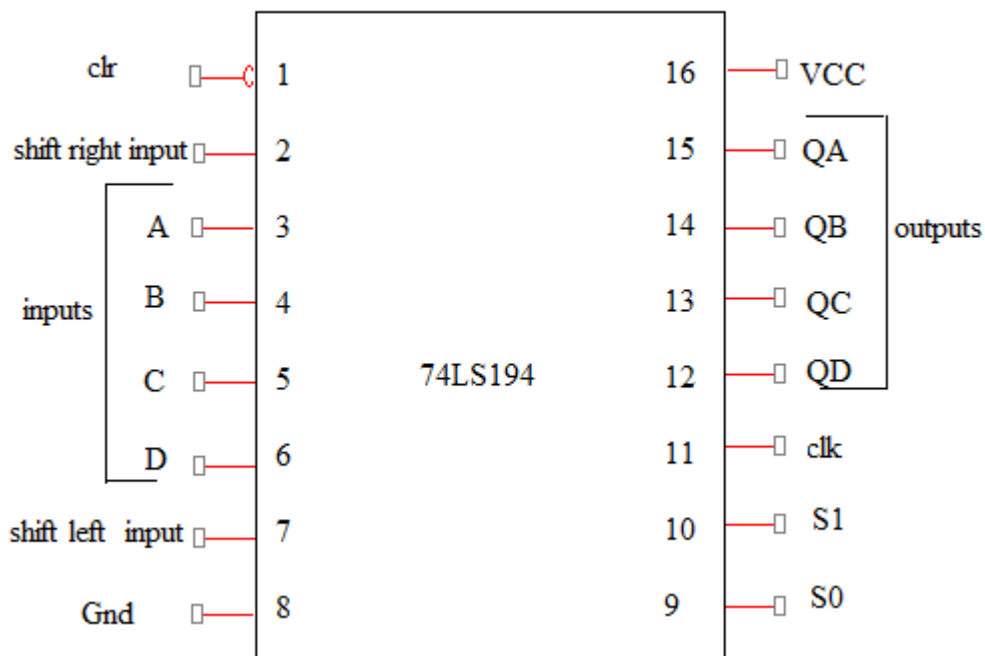
Synchronous parallel loading is accomplished by applying the four bit's of data and taking both mode control inputs, S_0 and S_1 , high. The data are loaded into the associated

flip-flops and appear at the outputs after the positive transition of the clock input. During loading serial data flow is inhibited.

Shift right is accomplished synchronously with the rising edge of the clock pulse when s_0 is high and s_1 is low. Serial data for this mode is entered at the shift right data inputs when s_0 is low and s_1 is high, data shifts left synchronously and new data is entered at the shift-left serial input. Clocking of the flip-flop is inhibited when both mode control inputs are low. The mode controls of the universal shift register should be changed only while the clock input is high.

IC Type 74LS194

The pin diagram of the universal shift register of IC type 74LS194 is as shown below.



PRECAUTIONS:

1. Avoid Loose and wrong Connections.
2. Identify V_{CC} and GND pins of the IC correctly before connections.

PROCEDURE:

1. Connections are to be made as per the circuit diagram of universal shift register as shown above.
2. For Different Combinations of inputs, the outputs QA, QB, QC ,QD are tabulated and truth table is to be verified.

FUNCTIONAL TABLE

Clr	Mode		Clk	Inputs						Outputs			
	S ₁	S ₀		Serial		Parallel				Q _A	Q _B	Q _C	Q _D
				Left	Right	A	B	C	D				
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	L	X	X	X	X	X	X	X	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}
H	L	H	↑	X	H	X	X	X	X	H	Q _{A0}	Q _{B0}	Q _{C0}
H	L	H	↑	X	L	X	X	X	X	L	Q _{A0}	Q _{B0}	Q _{C0}
H	H	L	↑	H	X	X	X	X	X	Q _{B0}	Q _{C0}	Q _{D0}	H
H	H	L	↑	L	X	X	X	X	X	Q _{B0}	Q _{C0}	Q _{D0}	L

↑ : Transition from low to high level

H : High level (steady state)

L : Low level(Steady state)

X : don't care

OBSERVATION

Clr	Mode		Clk	Inputs						Outputs			
	S ₁	S ₀		Serial		Parallel				Q _A	Q _B	Q _C	Q _D
				Left	Right	A	B	C	D				
L	X	X	X	X	X	X	X	X	X				
H	H	H	↑	X	X	a	b	c	d				
H	L	L	X	X	X	X	X	X	X				
H	L	H	↑	X	H	X	X	X	X				
H	L	H	↑	X	L	X	X	X	X				
H	H	L	↑	H	X	X	X	X	X				
H	H	L	↑	L	X	X	X	X	X				

Result: The operation of the Universal shift register is studied and verified.