-

DEPARTMENT OF
COMPUTER SCIENCE ENGINEERING

III SEMESTER - R 2017

CS8382 DIGITAL SYSTEMS LABORATORY

LABORATORY MANUAL

Name : _____

Register No : _____

Section : _____

VISION

is committed to provide highly disciplined, conscientious and enterprising professionals conforming to global standards through value based quality education and training.

MISSION

- To provide competent technical manpower capable of meeting requirements of the industry
- To contribute to the promotion of Academic Excellence in pursuit of Technical Education at different levels
- To train the students to sell his brawn and brain to the highest bidder but to never put a price tag on heart and soul

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

VISION

To provide candidates with knowledge and skill in the field of Electrical and Electronics Engineering and thereby produce extremely well trained employable, socially responsible and innovative Electrical and Electronics Engineers.

MISSION

- To provide the students rigorous learning experience to produce creative solutions to society's needs.
- To produce electrical engineers of high calibre, conscious of the universal moral values adhering to professional ethical code.
- To provide highest quality learning environment for the students emphasizing fundamental concepts with strongly supported laboratory and prepare them to meet the global needs of the industry by continuous assessment and training.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

1. Fundamentals

To provide students with a solid foundation in mathematics, science and fundamentals of engineering enabling them to solve complex problems in order to develop real time applications.

2. Core Competence

To train the students to meet the needs of core industry with an attitude of learning new technologies.

3. Breadth

To provide relevant training and experience to bridge the gap between theory and practice which enable them to find solutions to problems in industry and research that contributes to the overall development of society.

4. Professionalism

To inculcate professional and effective communication skills to the students to make them lead a team and stand as a good decision maker to manage any constraint environment with good professional ethics at all strategies.

5. Lifelong Learning/Ethics

To practice ethical and professional responsibilities in the organization and society with commitment and lifelong learning needed for successful professional career.

PROGRAM OUTCOMES (POs)

a. Graduates will demonstrate knowledge of mathematics, science and electrical engineering.

b. Graduates will be able to identify, formulate and solve electrical engineering problems.

c. Graduates will be able to design and conduct experiments, analyze and interpret data.

d. Graduates will be able to design a system, component or process as per needs and specifications.

e. Graduates will demonstrate to visualize and work on laboratory and multidisciplinary tasks.

f. Graduates will demonstrate skills to use modern engineering tools, software and equipment to analyze problems.

g. Graduates will demonstrate knowledge of professional and ethical responsibilities.

h. Graduates will be able to communicate effectively by both verbal and written form.

i. Graduates will show the understanding of impact of engineering solutions on the society and also will be aware of contemporary issues.

j. Graduates will develop confidence for self-education and ability for lifelong learning.

k. Graduate who can participate and succeed in competitive examinations.

SYLLABUS

Objectives:

The student should be made to:

- Understand the various logic gates.
- Be familiar with various combinational circuits.
- Understand the various components used in the design of digital computers.
- Be exposed to sequential circuits
- Learn to use HDL

List of experiments:

1. Verification of Boolean Theorems using basic gates.
2. Design and implementation of combinational circuits using basic gates for arbitrary functions, code converters.
3. Design and implementation of combinational circuits using MSI devices:
   a. 4 – bit binary adder / subtractor
   b. Parity generator / checker
   c. Magnitude Comparator
   d. Application using multiplexers
4. Design and implementation of sequential circuits:
   a. Shift –registers
   b. Synchronous and asynchronous counters
5. Coding combinational / sequential circuits using HDL.
6. Design and implementation of a simple digital system (Mini Project).

Course Outcomes:

- Use Boolean simplification techniques to design a combinational hardware circuit.
- Design and Implement combinational and sequential circuits.
- Analyze a given digital circuit – combinational and sequential.
- Design the different functional units in a digital computer system.
- Design and Implement a simple digital system.

CS8381 DIGITAL SYSTEMS LABORATORY

Content

Expt.No.1: STUDY OF BASIC GATES

Aim:

To verify the truth table of basic digital IC's of AND, OR, NOT, NAND, NOR, EX-OR gates

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	NAND gate	IC 7400	1
6.	NOR gate	IC 7402	1
7.	EX-OR gate	IC 7486	1
8.	Connecting wires		As required

Theory:

Circuit that takes the logical decision and the process are called logic gates. Each gate has one or more input and only one output. OR, AND and NOT are basic gates. NAND, NOR and X-OR are known as universal gates. Basic gates form these gates.

AND gate

The AND gate performs a logical multiplication commonly known as AND function. The output is high when both the inputs are high. The output is low level when any one of the inputs is low.

OR gate

The OR gate performs a logical addition commonly known as OR function. The output is high when any one of the inputs is high. The output is low level when both the inputs are low.

NOT gate

A NOT gate is the physical realization of the complementation operation. The NOT gate is called an inverter. The output is high when the input is low. The output is low when the input is high.

NAND gate

The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and any one of the input is low .The output is low level when both inputs are high.

NOR gate

The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

EX-OR gate

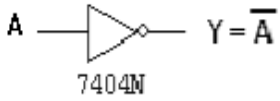
An Ex-OR gate performs the following Boolean function, $A \oplus B = (A . B') + (A' . B)$. It is similar to OR gate but excludes the combination of both A and B being equal to one. The exclusive OR is a function that give an output signal '0' when the two input signals are equal either '0' or '1'.

AND Gate Symbol: PIN Diagram:

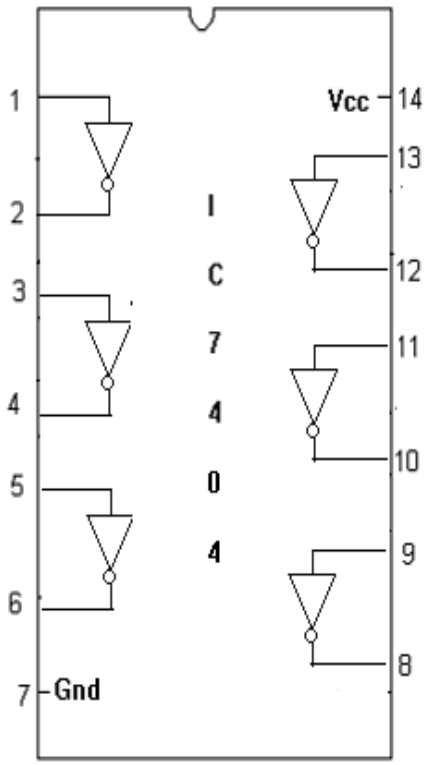
OR Gate:

OR GATE:

NOT Gate symbol:



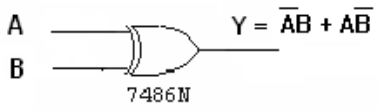
PIN Diagram:



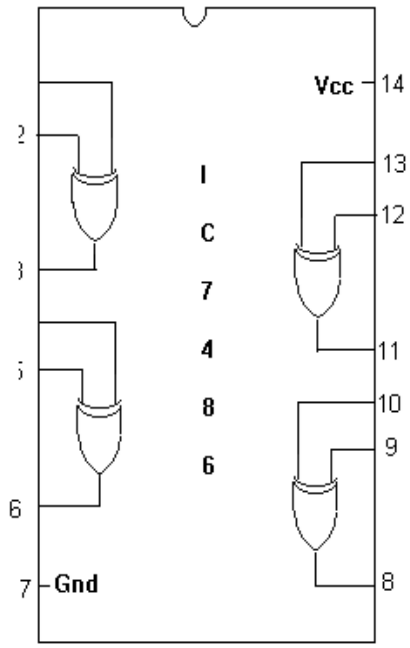
TRUTH TABLE :

A	$\overline{A}$
0	1
1	0

EXOR Gate symbol:



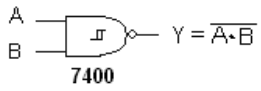
PIN Diagram:



TRUTH TABLE :

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

NAND Gate symbol:

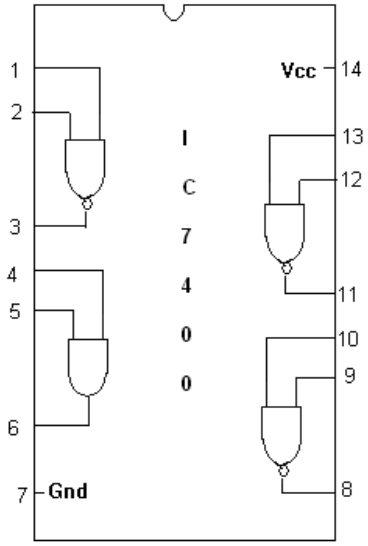


7400

TRUTH TABLE

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

PIN Diagram:



NOR Gate:

Procedure:

1. Connections are given as per the circuit diagram.
2. For all the IC's 7^{th} pin is grounded and 14^{th} pin is given +5 supply.
3. Apply the inputs and verify the truth table for all gates.

Result:

The truth tables of all the basic logic gates were verified.

Outcome:

At the completion of an experiment student will able to verify the truth table of all basic gates

Viva – Voce

1. List out the basic gate.
2. Mention the universal gate.
3. How many gates presented in IC 7408?
4. What is IC?
5. What are the applications of gates?
6. Write the truth table of AND gate.
7. Write the truth table of OR gate.
8. Write the truth table of NOT gate.
9. Write the truth table of NAND gate.
10. Write the truth table of NOR gate.
11. Write the truth table of EX- OR gate.
12. What are the classifications of IC?
13. What are types of linear integrated circuit?
14. What is meant by etching?
15. What are the advantages of IC?
16. Write the truth table of EX- NOR gate.

Expt.No.2:

VERIFICATION OF BOOLEAN THEOREMS USING LOGIC GATES

Aim: To verification of Boolean theorems using logic gates

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	NAND gate	IC 7400	1
6.	NOR gate	IC 7402	3
7.	EX-OR gate	IC 7486	1
8.	Connecting wires		As required

Theory:

BASIC Boolean Laws

1. Commutative Law

The binary operator OR, AND is said to be commutative if,

1. A+B = B+A
2. A.B=B.A

2. Associative Law

The binary operator OR, AND is said to be associative if,

1. A+(B+C) = (A+B)+C
2. A.(B.C) = (A.B).C

3. Distributive Law

The binary operator OR, AND is said to be distributive if,

1. A+(B.C) = (A+B).(A+C)
2. A.(B+C) = (A.B)+(A.C)

4. Absorption Law

1. A+AB = A
2. A+AB =$\overline{A}$+B

5. Idempotent Law

1. A+A = A
2. A.A = A

6. Complementary Law

1. A+A' = 1
2. A.A' = 0

7. De Morgan's Theorem

1. The complement of the sum is equal to the sum of the product of the individual complements.

$\overline{A+B} = \overline{A}.\overline{B}$

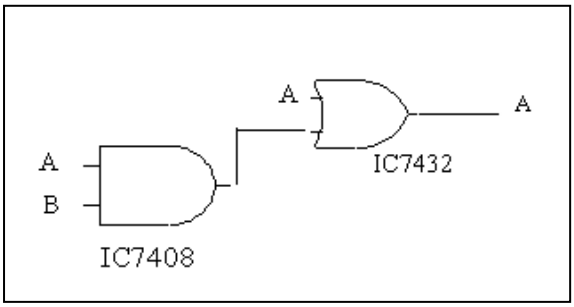
2. The complement of the product is equal to the sum of the individual complements.

$\overline{A.B} = \overline{A}+\overline{B}$

Design

1. Absorption Law

A+AB = A

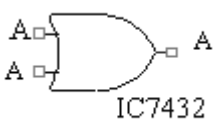


2. Involution (or) Double complement Law

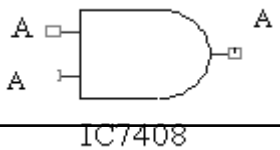
A = A

3. Idempotent Law

1. A+A = A

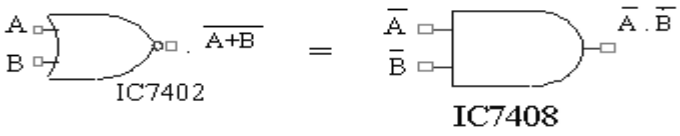


2. A.A = A



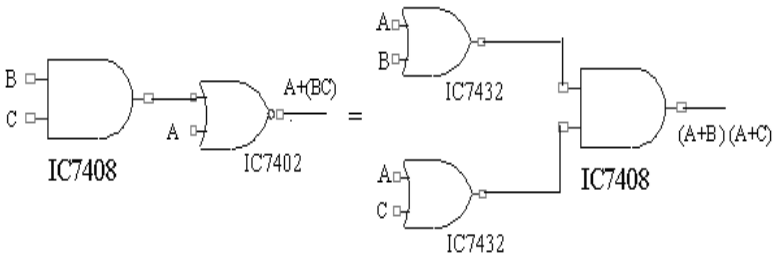
4. Demorgan's Law

A+B = A.B



5. Distributive Law

A+(B.C) = (A+B).(A+C)



Procedure:

1. Obtain the required IC along with the Digital trainer kit.
2. Connect zero volts to GND pin and +5 volts to V_{CC} .
3. Apply the inputs to the respective input pins.
4. Verify the output with the truth table.

Result:

Thus the above stated Boolean laws are verified.

Outcome:

At the completion of an experiment student will able to know the basic laws with their truth table.

Viva – Voce

1. What is Demorgan's law?
2. What is associative law?
3. What is mean by compliment gate?
4. Explain the basic laws in digital electronics
5. What is double complement?

Expt.No.3: HALF ADDER AND FULL ADDER

Aim:

To design and verify the truth table of the Half Adder & Full Adder circuits

Apparatus required:

S. No.	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	EX-OR gate	IC 7486	1
6.	Connecting wires	As required	

Theory:

The most basic arithmetic operation is the addition of two binary digits. There are four possible elementary operations, namely,

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (with 1 as carry)

The first three operations produce a sum of whose length is one digit, but when the last operation is performed the sum is two digits. The higher significant bit of this result is called a carry and lower significant bit is called the sum.

Half adder:

A combinational circuit which performs the addition of two bits is called half adder. The input variables designate the augend and the addend bit, whereas the output variables produce the sum and carry bits.

Full adder:

A combinational circuit which performs the arithmetic sum of three input bits is called full adder. The three input bits include two significant bits and a previous carry bit. A full adder circuit can be implemented with two half adders and one OR gate.

From the truth table, the expression for sum and carry bits of the output can be obtained as,

SUM = A'B'C + A'BC' + AB'C' + ABC

CARRY = A'BC + AB'C + ABC' +ABC

Half Adder

Truth table:

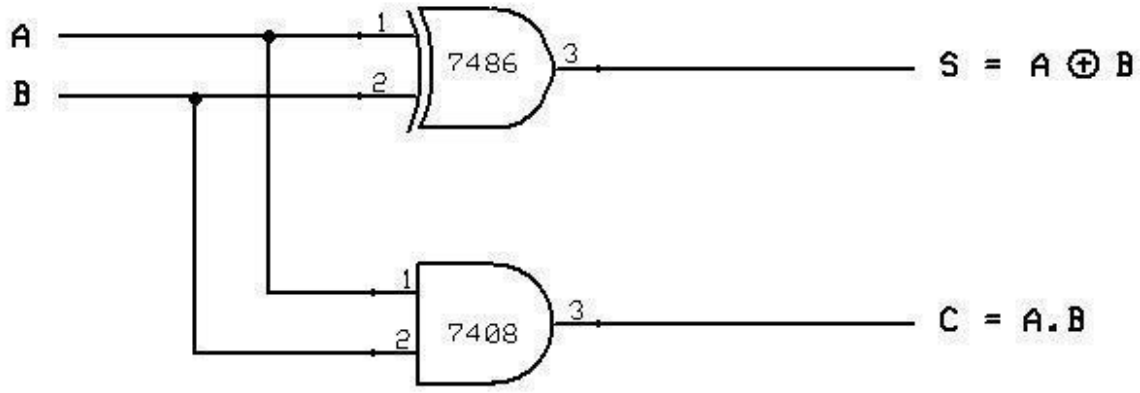
Sl.No.	Input		Output	
	A	B	S	C
1.	0	0	0	0
2.	0	1	1	0
3.	1	0	1	0
4.	1	1	1	1

From the truth table the expression for sum and carry bits of the output can be obtained as, Sum,

$S = A \oplus B$

Carry, $C = A \cdot B$

Circuit diagram:



Full adder

Truth table:

Sl.No.	Input			Output	
	A	B	C	S	C
1.	0	0	0	0	0
2.	0	0	1	1	0
3.	0	1	0	1	0
4.	0	1	1	0	1
5.	1	0	0	1	0
6.	1	0	1	0	1
7.	1	1	0	0	1
8.	1	1	1	1	1

Using Karnaugh maps the reduced expression for the output bits can be obtained as,

Sum:

A \ BC	B'C'	B'C	BC	BC'
A'	0	(1)	0	(1)
A	(1)	0	(1)	0

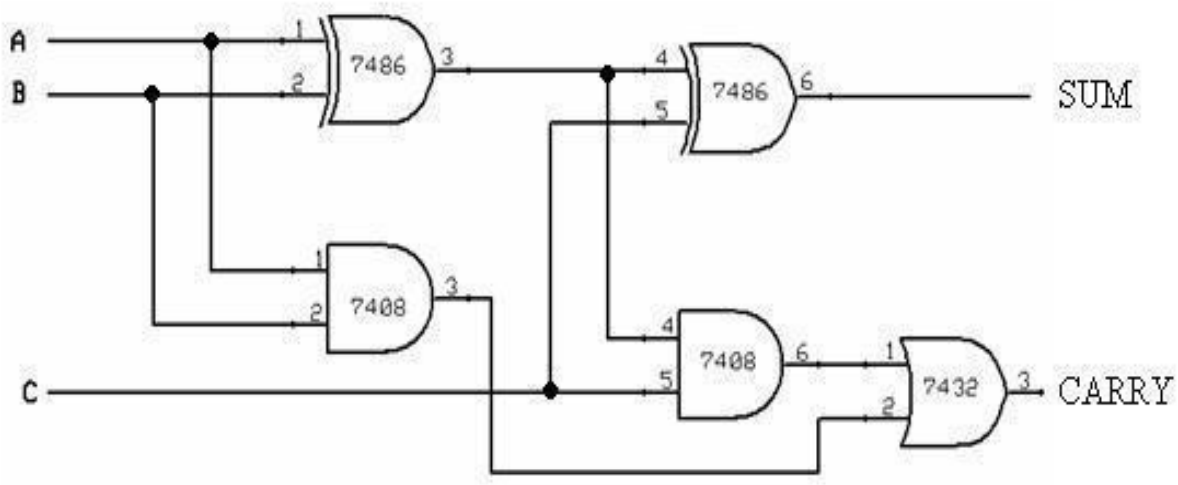
SUM = A'B'C + A'BC' + AB'C' + ABC = A ⊕ B ⊕ C

Carry:

A \ BC	B'C'	B'C	BC	BC'
A'	0	0	1	0
A	0	1	1	1

CARRY = AB + AC + BC

Logic Diagram:



Procedure:

1. Connections are given as per the circuit diagrams.
2. For all the IC's 7^{th} pin is grounded and 14^{th} pin is given +5 V supply.
3. Apply the inputs and verify the truth table for the half adder and full adder circuits.

Result:

The design of the half adder and full adder circuits was done and their truth tables were verified.

Outcome:

At the completion of an experiment student will able to design the half adder circuit and the full adder circuit.

Expt.No.4: HALF SUBTRACTOR AND FULL SUBTRACTOR

Aim:

To design and verify the truth table of the half subtractor & full subtractor circuits

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	EX-OR gate	IC 7486	1
6.	Connecting wires	As required	

Theory:

The subtraction of two binary digits has four possible operations. In all operations, each subtrahend bit is subtracted from the minuend bit. In case of the second operation the minuend bit is smaller than the subtrahend bit, hence 1 is borrowed.

Half subtractor:

A combinational circuit which performs the subtraction of two bits is called half subtractor. The input variables designate the minuend and the subtrahend bit, whereas the output variables produce the difference and borrow bits.

Full subtractor:

A combinational circuit which performs the subtraction of three input bits is called full subtractor. The three input bits include two significant bits and a previous borrow bit. A full subtractor circuit can be implemented with two half subtractors and one OR gate.

From the truth table the expression for difference and borrow bits of the output can be obtained as,

Difference, DIFF= A'B'C + A'BC' + AB'C' + ABC

Borrow, BORR = A'BC + AB'C + ABC' +ABC

Half subtractor

Truth table:

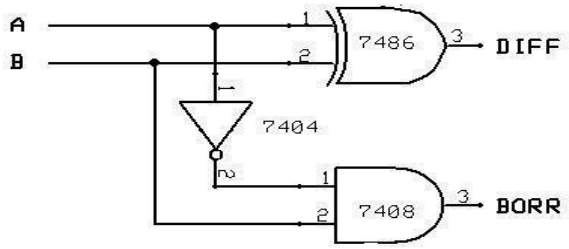
Sl.No.	Input		Output	
	A	B	Difference	Borrow
1.	0	0	0	0
2.	0	1	1	1
3.	1	0	1	0
4.	1	1	0	0

From the truth table the expression for difference and borrow bits of the output can be obtained as,

Difference, DIFF = A $\oplus$ B

Borrow, BORR = A'. B

Logic diagram:



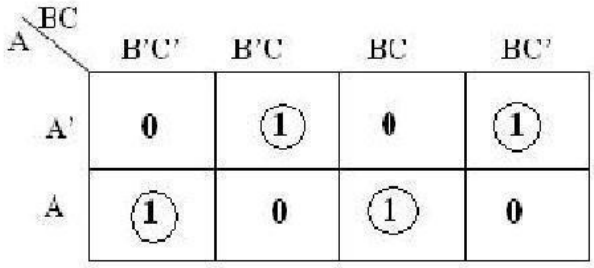
2. Full subtractor

Truth table:

Sl.No.	Input			Output	
	A	B	C	Difference	Borrow
1.	0	0	0	0	0
2.	0	0	1	1	1
3.	0	1	0	1	1
4.	0	1	1	0	1
5.	1	0	0	1	0
6.	1	0	1	0	0
7.	1	1	0	0	0
8.	1	1	1	1	1

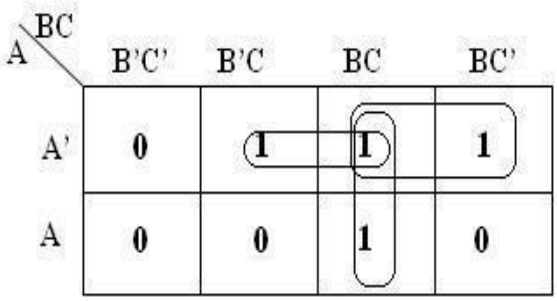
Using Karnaugh maps the reduced expression for the output bits can be obtained as,

Difference



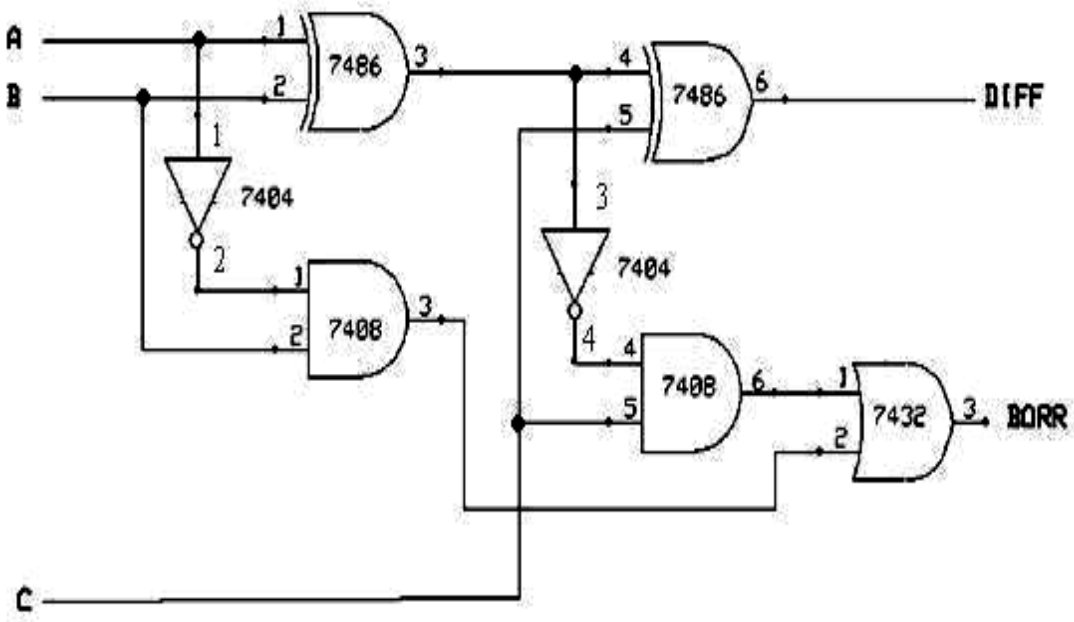
Difference = A'B'C + A'BC' + AB'C' + ABC

Borrow



Borrow = A'B + A'C + BC

Circuit diagram:



Expt.No.5: 4-BIT ADDER AND SUBTRACTOR

Aim:

To design and implement 4-bit adder and subtractor using IC 7483

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	IC	IC 7483	1
3.	NOT gate	IC 7404	1
4.	EX-OR gate	IC 7486	1
5.	Connecting wires		As required

Theory:

4 BIT Binary adder:

A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of next full adder in chain. The augends bits of 'A' and the addend bits of 'B' are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bits. The carries are connected in chain through full adder. The input carry to the adder is C_0 and it ripples through the full adder to the output carry C_4.

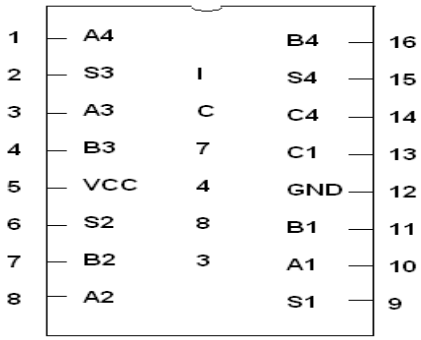
4 BIT Binary subtractor:

The circuit for subtracting A-B consists of an adder with inverters, placed between each data input 'B' and the corresponding input of full adder. The input carry C_0 must be equal to 1 when performing subtraction.

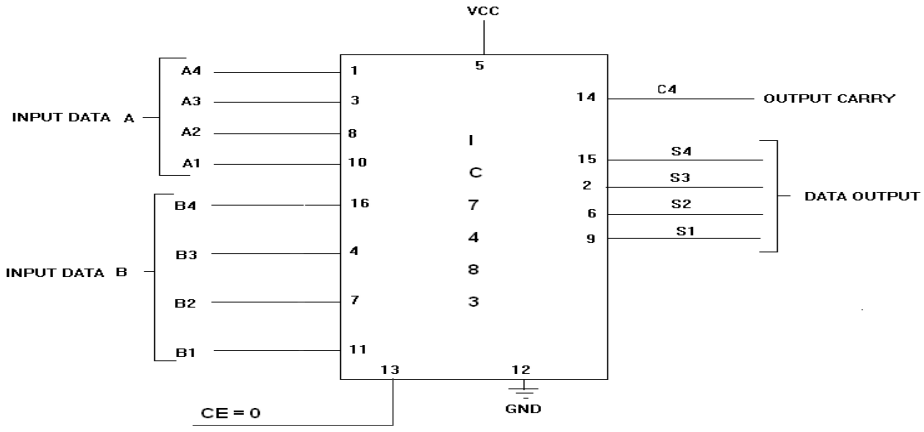
4 BIT Binary adder / subtractor:

The addition and subtraction operation can be combined into one circuit with one common binary adder. The mode input M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

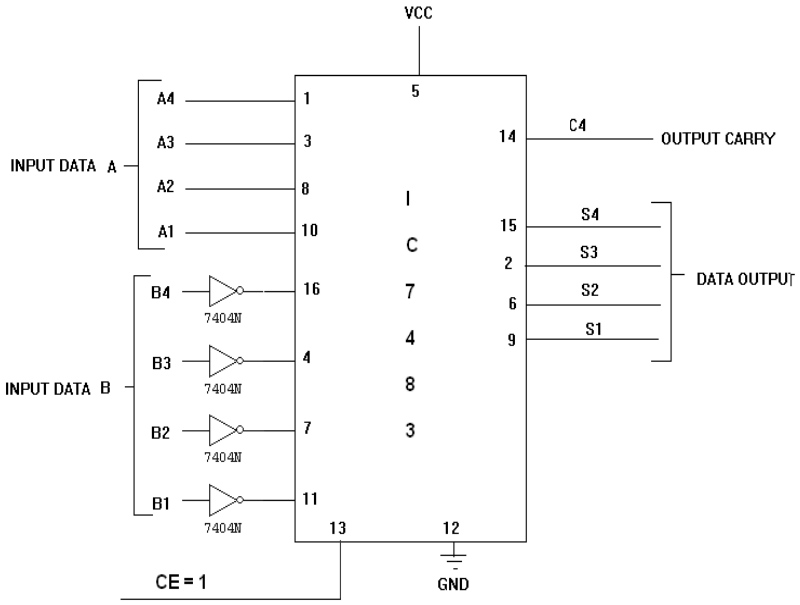
PIN Diagram for IC 7483:



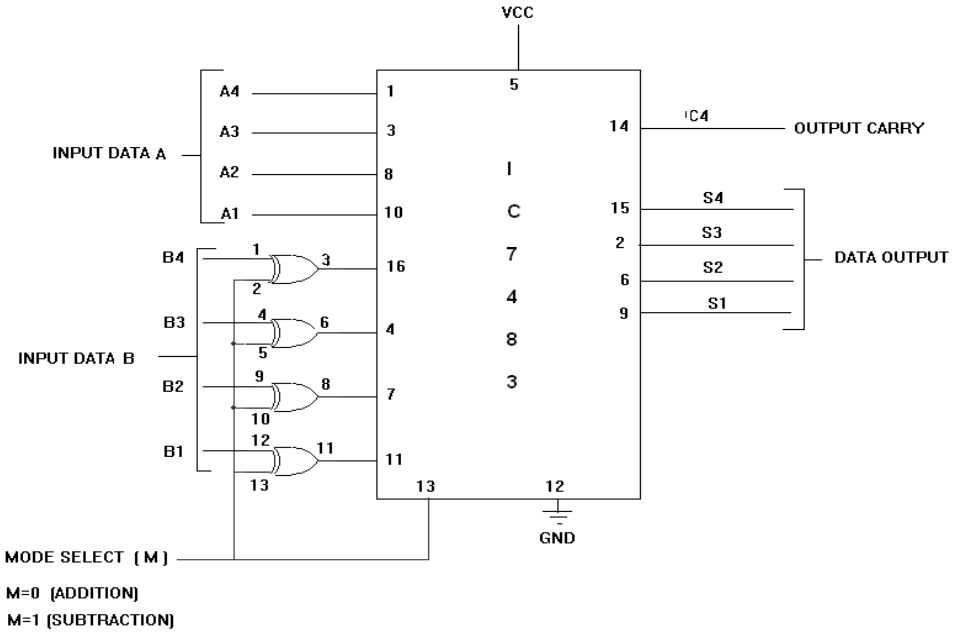
Logic Diagram: 4-Bit Binary Diagram:



Logic diagram: 4-Bit Binary Subtractor:



4-Bit Binary Adder /Subtractor:



Truth table:

Input Data A				Input Data B				Addition					Subtraction				
A4	A3	A2	A1	B4	B3	B2	B1	C	S4	S3	S2	S1	B	D4	D3	D2	D1
1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	0	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1
1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1

Procedure:

1. Connections are given as per the circuit diagrams.
2. Logical inputs were given as per circuit diagram.
3. Apply the inputs and verify the truth table for the 4-bit adder and subtractor.

.

Result:

The design of the 4-bit Binary adder and I subtractor circuit was done and its truth table was verified.

Outcome:

At the completion of an experiment student will able to design 4-bit binary adder and subtractor

Viva – Voce

1. What is expression for difference and borrow?
2. Write the truth table for half adder.
3. Write the truth table for full adder.
4. Write the truth table for half subtrator.
5. Write the truth table for full subtrator.
6. Draw the logic diagram of full subtrator.
7. What is adder?
8. List out the application of adders.
9. Draw the full adder using two half adder circuits.
10. What is combinational circuit?
11. What is different between combinational and sequential circuit?
12. What are the gates involved for binary adder?
13. List the properties of Ex-Nor gate?
14. What is expression for sum and carry?

Expt.No.6: MAGNITUDE COMPARATOR

Aim:

To design, construct and study the performance of 2 bit magnitude comparator

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	Magnitude comparator	IC 7485	2
6.	EX-OR gate	IC 7486	1
7.	Connecting wires		As required

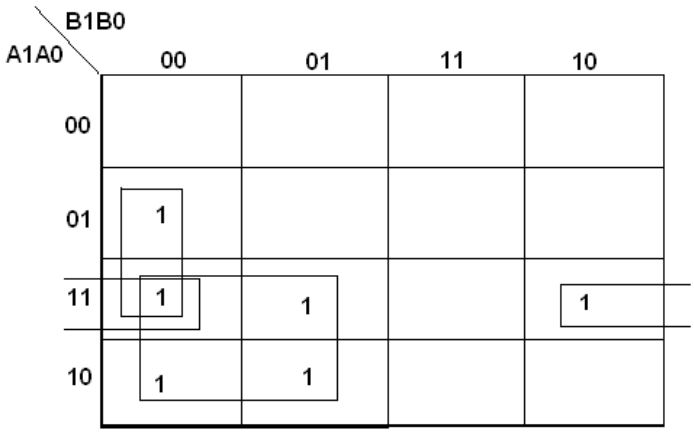
Theory:

The comparison of two numbers is an operator that determines one number is greater than, less than (or) equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitude. The outcome of the comparator is specified by three binary variables that indicate whether A>B, A=B (or) A<B.

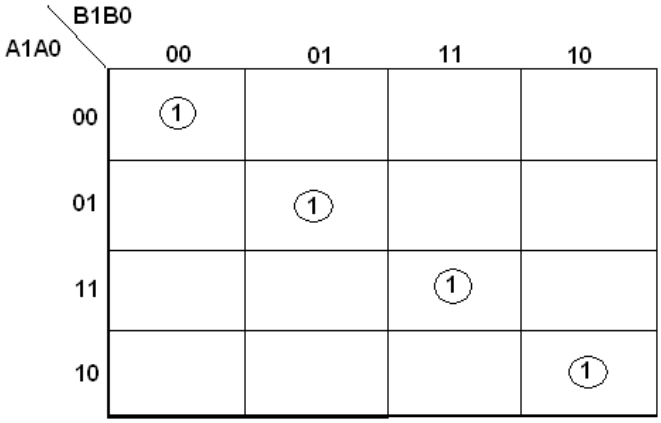
Truth table:

Inputs				Outputs		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

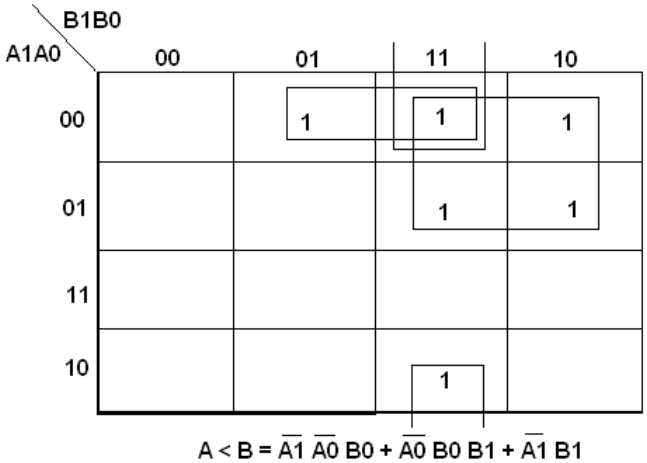
K-map



A > B = A0 $\overline{B0}$ B1 + A1 $\overline{B1}$ + A1 A0 $\overline{B0}$

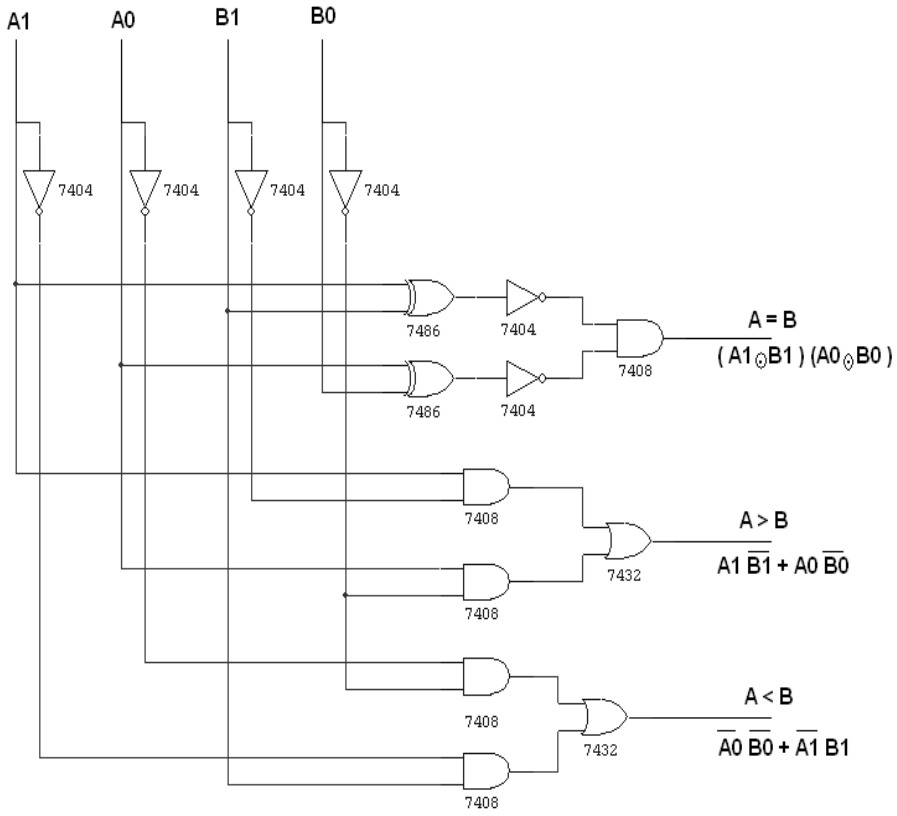


A = B =(A0$\odot$B0)(A1$\odot$B1)

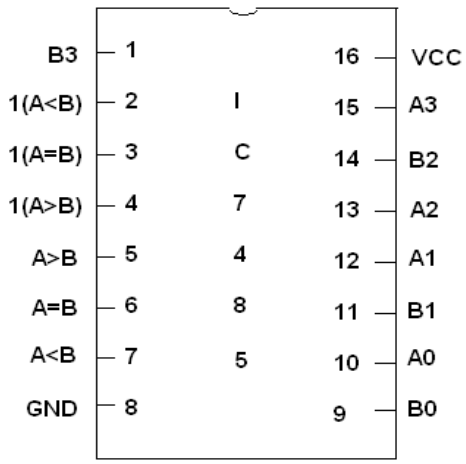


A < B = $\overline{A1}$ $\overline{A0}$ B0 + $\overline{A0}$ B0 B1 + $\overline{A1}$ B1

Logic Diagram:

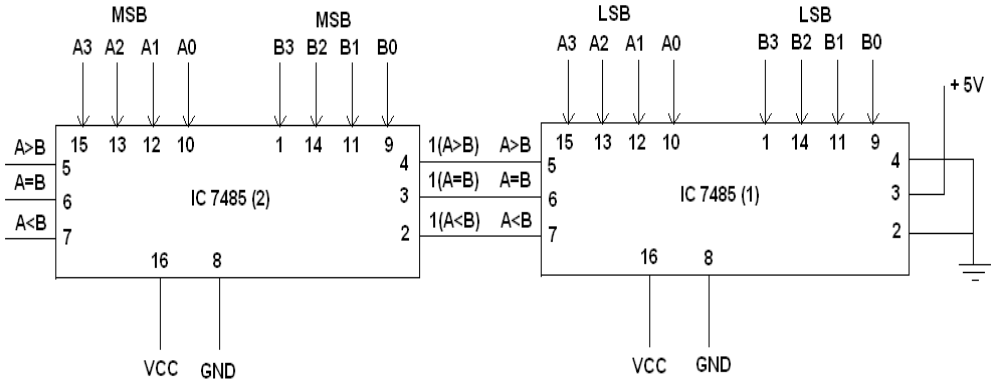


Pin Diagram for IC 7485:



Logic Diagram:

8 Bit Magnitude Comparator:



Truth table:

A	B	A>B	A=B	A<B
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	1	0
0 0 0 1 0 0 0 1	0 0 0 0 0 0 0 0	1	0	0
0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 1	0	0	1

Procedure:

1. Connections are given as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

Result:

Thus the 2-bit and 8-bit magnitude comparator was designed and verified using the logic gates.

Outcome:

At the completion of an experiment student will able to design the 2-bit and 8-bit magnitude comparator using logic gates.

Viva – Voce

1. What is magnitude comparator?
2. What is most significant bit?
3. Explain operation of AND gate.
4. Explain truth table of a comparator.
5. Explain magnitude comparator7485 IC.
6. What is 8-bit input Magnitude Comparator?
7. What is IC?
8. Explain the k-map simplification of A>B.
9. Explain the k-map simplification of A=B.
10. Explain the k-map simplification of A<B
11. Draw the logic diagram of 1-bit magnitude comparator.
12. What is the truth table of 1-bit magnitude comparator?
13. What is the use of magnitude comparator?

Expt.No.7: CODE CONVERSION

Aim:

To design, construct and study the performance of 4-bit different code converters

(i) Binary to gray code converter

(ii) Gray to binary code converter

(iii) BCD to excess-3 code converter

(iv) Excess-3 to BCD code converter

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	Magnitude comparator	IC 7485	2
6.	EX-OR gate	IC 7486	1
7.	Connecting wires		As required

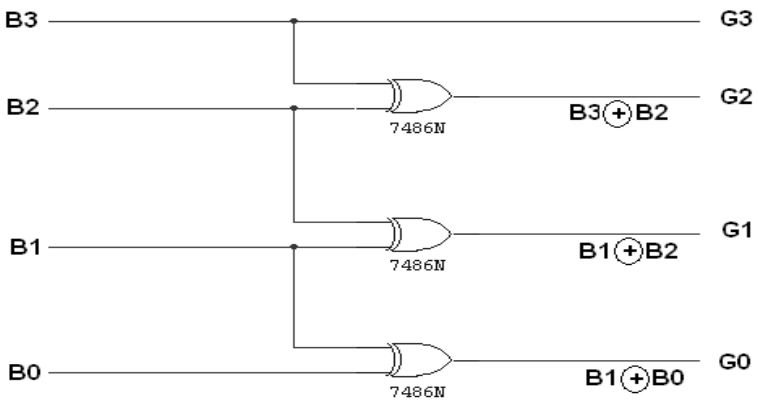
Theory:

The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code. The bit combination assigned to binary code to gray code. Since each code uses four bits to represent a decimal digit. There are four inputs and four outputs. Gray code is a non-weighted code. The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable. A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. To convert from binary code to Excess-3 code, the input lines must supply the bit combination of elements as specified by code and the output lines generate the corresponding bit combination of code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables. A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. These are various other possibilities for a logic diagram that implements this circuit. Now the OR gate whose output is C+D has been used to implement partially each of three outputs.

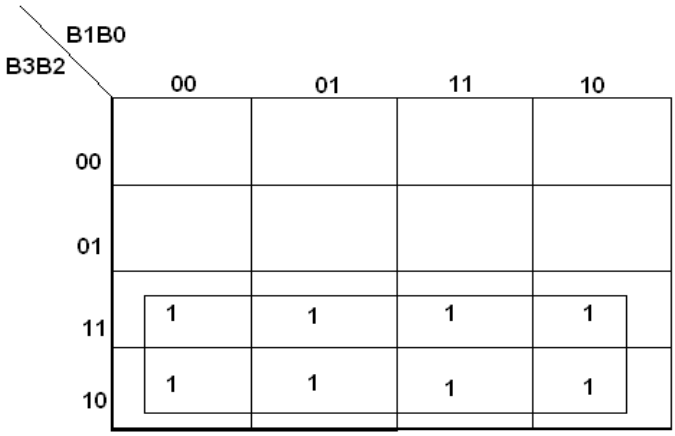
Logic diagram:

(i) Binary to gray code converter

Logic Diagram:

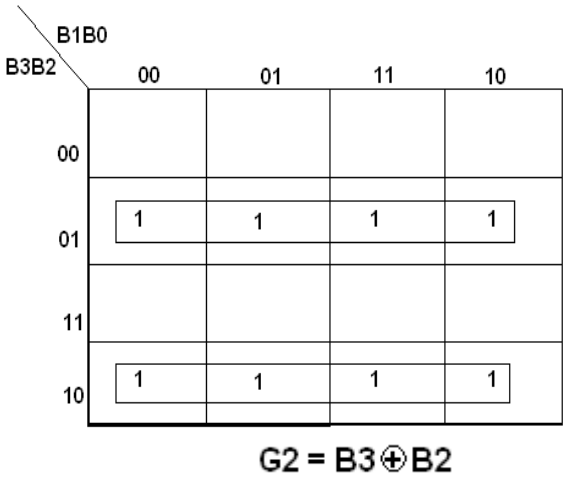


K map for G3:



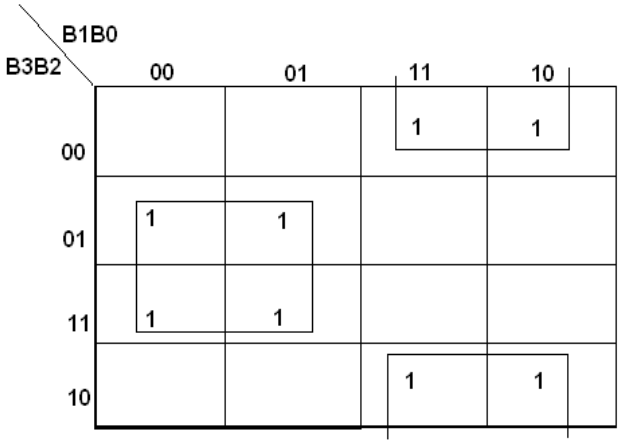
G3 = B3

K map for G2:



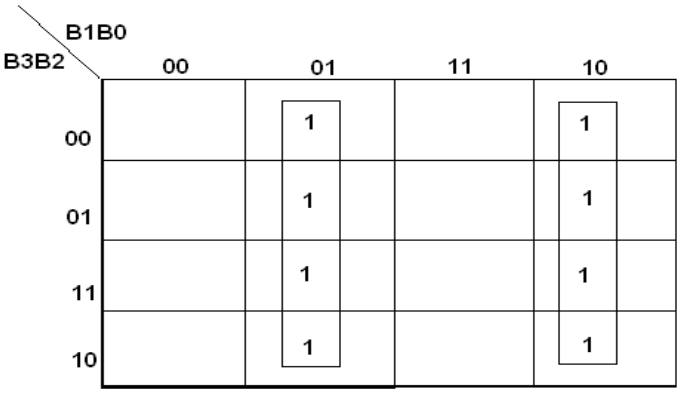
G2 = B3⊕B2

K map for G1:



G1 = B1⊕B2

K map for G0:



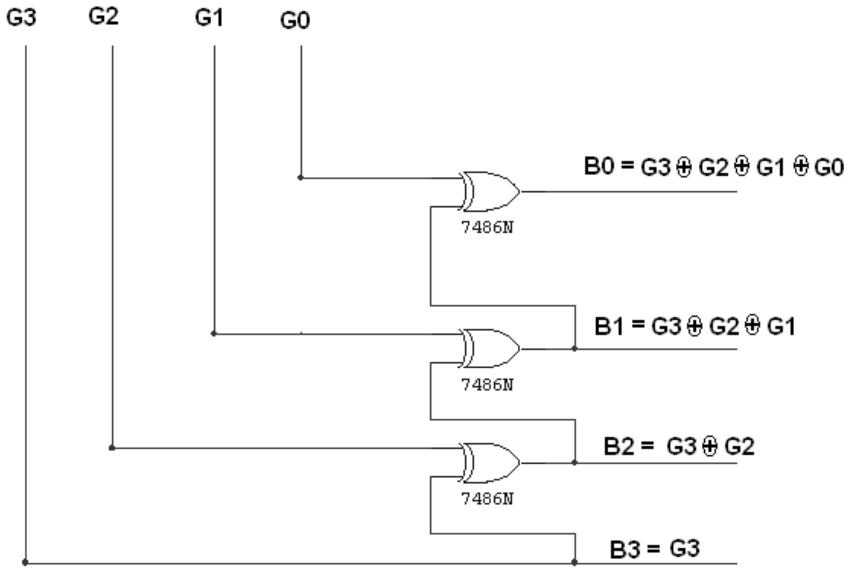
G0 = B1⊕B0

Truth table:

0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

(ii) Gray to binary code converter

Logic Diagram:



K map for B3:

G3G2 \ G1G0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

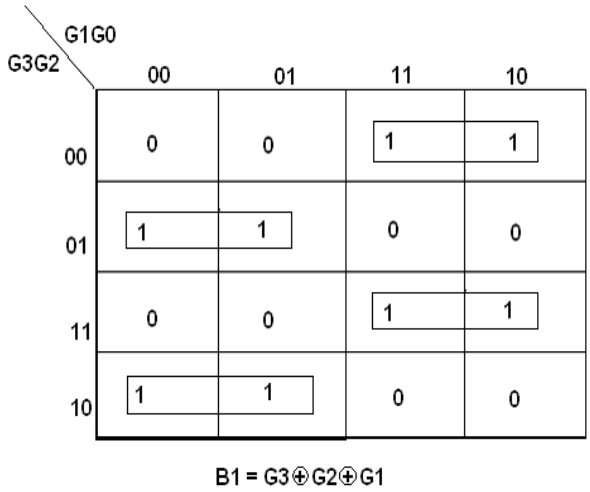
B3=G3

K map for B2:

G3G2 \ G1G0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

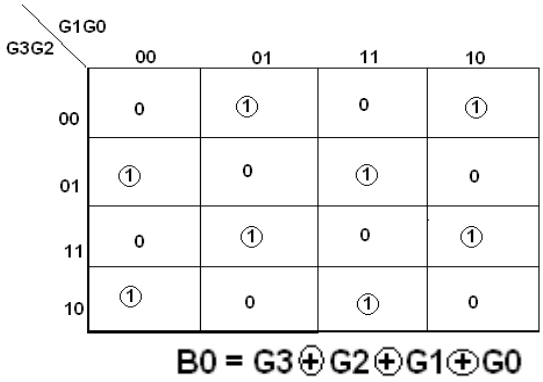
B2 = G3⊕G2

K map for B1:



B1 = G3⊕G2⊕G1

K map for B0:



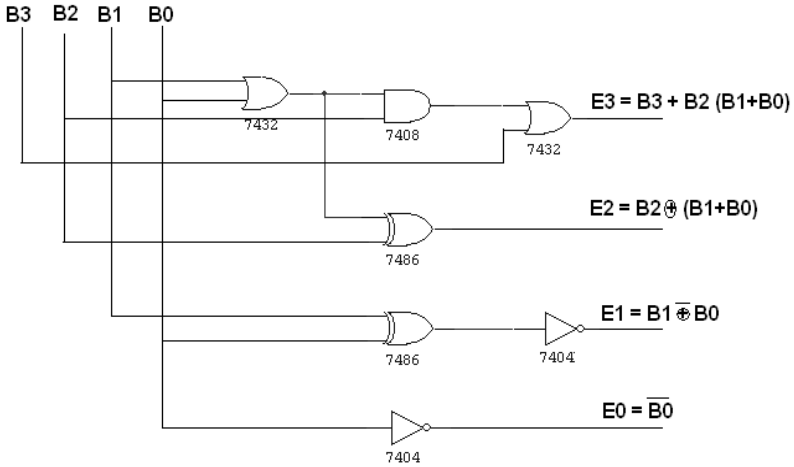
B0 = G3⊕G2⊕G1⊕G0

Truth table:

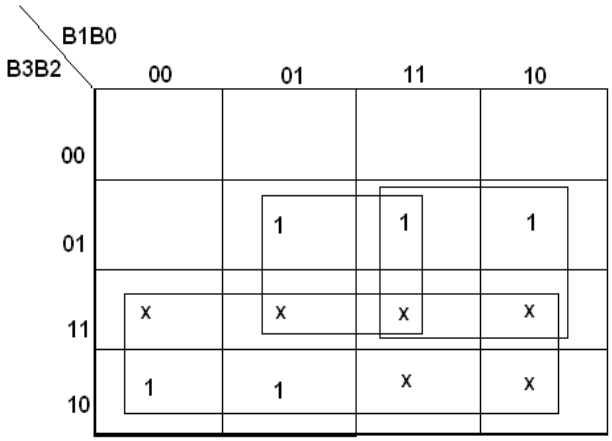
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

(iii) BCD to excess-3 code converter

Logic Diagram:

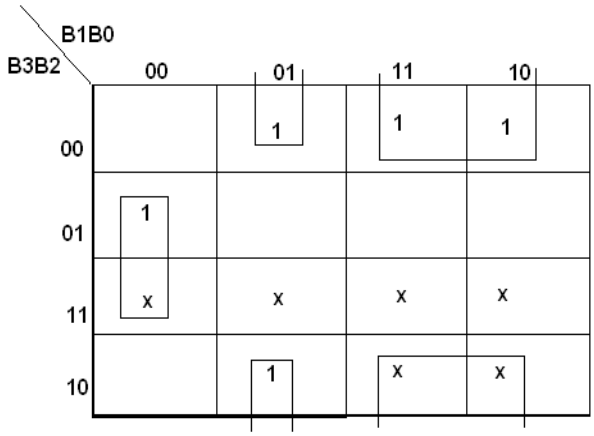


K map for E3:



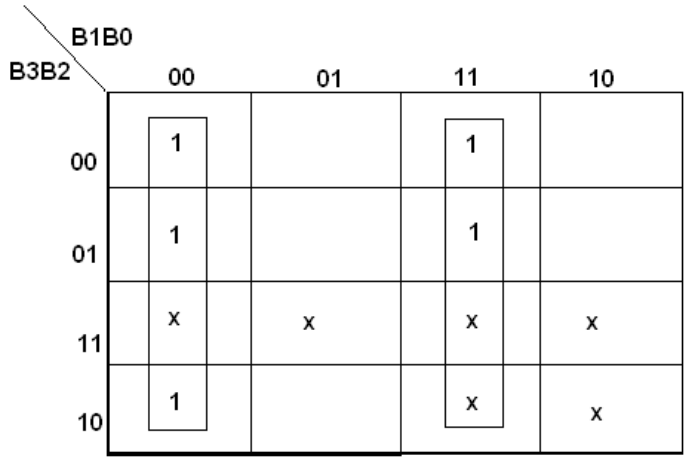
E3 = B3 + B2 (B0 + B1)

K map for E2:



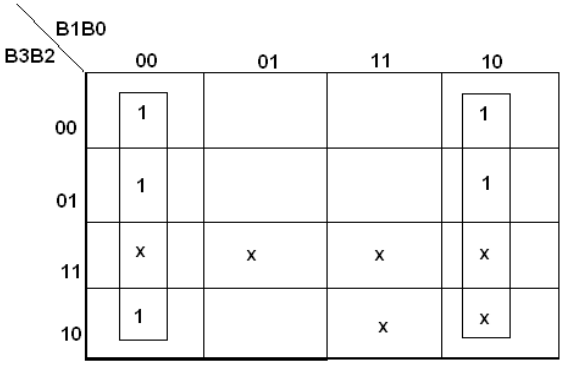
E2 = B2⊕(B1 + B0)

K map for E1:



$E1 = B1 \overline{\oplus} B0$

K map for E0:



$E0 = \overline{B0}$

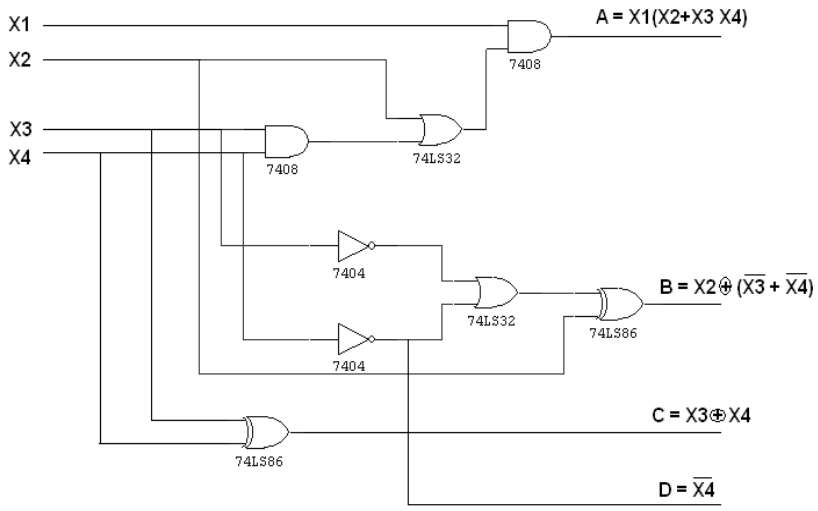
Truth table:

B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

(iv) Excess-3 to

BCD code converter

Logic Diagram:



K map for A:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	0	0
11	1	X	X	X
10	0	0	1	0

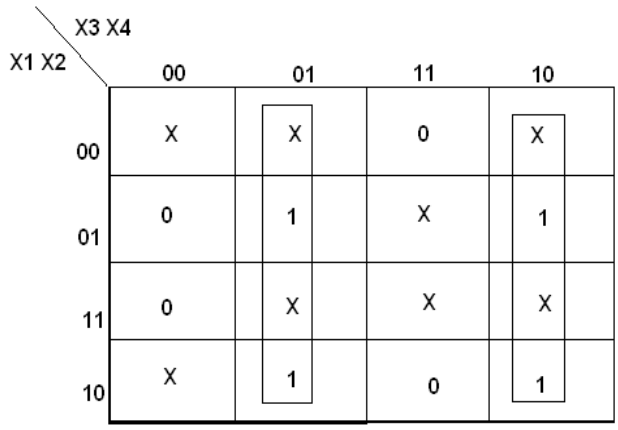
A = X1 X2 + X3 X4 X1

K map for B:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	1	0
11	0	X	X	X
10	1	1	0	1

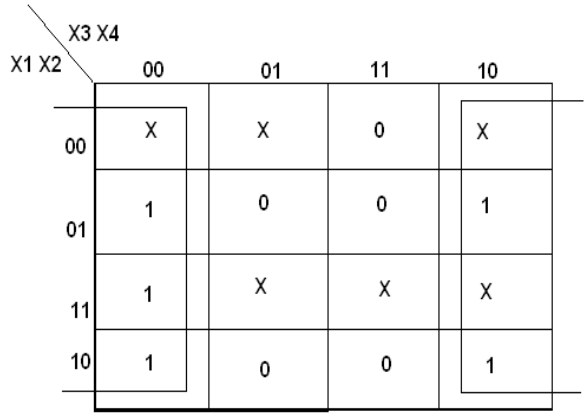
B = X2⊕ ($\overline{X3}$ + $\overline{X4}$)

K map for C:



$C = X3 \oplus X4$

K map for D:



$D = \overline{X4}$

Truth table:

B3	B2	B1	B0	G3	G2	G1	G0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

Procedure:

1. Connections were given as per circuit diagram.
2. Logical inputs were given as per truth table
3. Observe the logical output and verify with the truth tables.

Result:

Thus the code converters were designed and verified using the corresponding truth table.

Outcome:

At the completion of an experiment student will able to design the binary to gray converter.

Viva – Voce

1. What is binary code?
2. What is gray code?
3. What are the advantages of gray code?
4. What is unit distance code?
5. What is sequential code?
6. How to convert binary to gray code?
7. How to convert gray to binary code?
8. What is reflective code?
9. What are the advantages of EX – 3 code?
10. Which code is used to arithmetic operation in digital circuits?
11. Explain the operation of EX – OR.
12. What is K – Map?
13. Draw the truth table of EX- OR gate.
14. What is SOP?
15. What is POS?
16. What is minterm?

Expt.No.8: PARITY GENERATORS AND CHECKERS

Aim:

To implement the odd and even parity checkers using the logic gates and also to generate the odd parity and even parity numbers using the generators

Apparatus required:

Sl. No	Component	Type	Quantity
1	Trainer Kit	-	1
2	EX-OR	IC7486	1
3	NOT gate	IC 7404	1
4	Connecting wires	-	Required

Theory:

Parity checking is used for error detection in data transmission.

Odd parity checkers:

It counts the number of 1's in the given input and produces a 1 in the output when the number of 1's is odd.

Even parity checker:

It counts the number of 1's in the given input and produces a 1 in the output when the number of 1's is even.

Odd parity generators:

It generates an odd parity number. The odd parity checker circuit is used with the inverted output and also the input bits. So when the input is a 4-bit number then the output of the generator circuit will have 5 bits which is an odd parity number.

Even parity generator:

It generates an even parity number. The even parity checker circuit is used with the inverted output and also the input bits. So when the input is a 4-bit number then the output of the generator circuit will have 5 bits which is an even parity number.

Truth table:

Input				Checker output		Generator output	
A	B	C	D	odd	even	odd	even
0	0	0	1	1	0	00010	00011
0	0	1	0	1	0	00100	00101
0	0	1	1	0	1	00111	00110
0	1	0	0	1	0	01000	01001
0	1	0	1	0	1	01011	01010
0	1	1	0	0	1	01101	01100
0	1	1	1	1	0	01110	01111
1	0	0	0	1	0	10000	10001
1	0	0	1	0	1	10011	10010
1	0	1	0	0	1	10101	10100
1	0	1	1	1	0	10110	10111
1	1	0	0	0	1	11001	11000
1	1	0	1	1	0	11010	11011
1	1	1	0	1	0	11100	11101
1	1	1	1	0	1	11111	11110

Procedure:

1. The circuit is implemented using logic gates.
2. The inputs are given as per the truth table.
3. The corresponding outputs are noted.
4. The theoretical and practical values were verified.

Result:

The odd and even parity checkers are implemented using the logic gates and the odd parity and even parity numbers are generated using the corresponding generators.

Outcome:

At the completion of an experiment student will able to verify the odd and even parity checker using logic gates.

Viva – Voce

1. What is parity bit?
2. Why parity bit is added to message?
3. What is parity checker?
4. What is odd parity?
5. What is even parity?
6. What are the gates involved for parity generator?
7. List the procedures to convert gray code into binary.
8. Why weighted code is called as reflective codes?
9. What is a sequential code?
10. What is error deducting code?
11. What is ASCII code?
12. What is hamming code?
13. List the binary weighted code.
14. List the binary non weighted code.
15. Write the hamming code equation
16. List the procedures to convert binary code into gray
17. What are the applications of gray code?
18. What are the applications of Excess- 3 code?

Expt.No.9: MULTIPLEXER AND DEMULTIPLEXER

Aim:

To design and verify the truth table of a 4X1 multiplexer & 1X4 demultiplexer

Apparatus required:

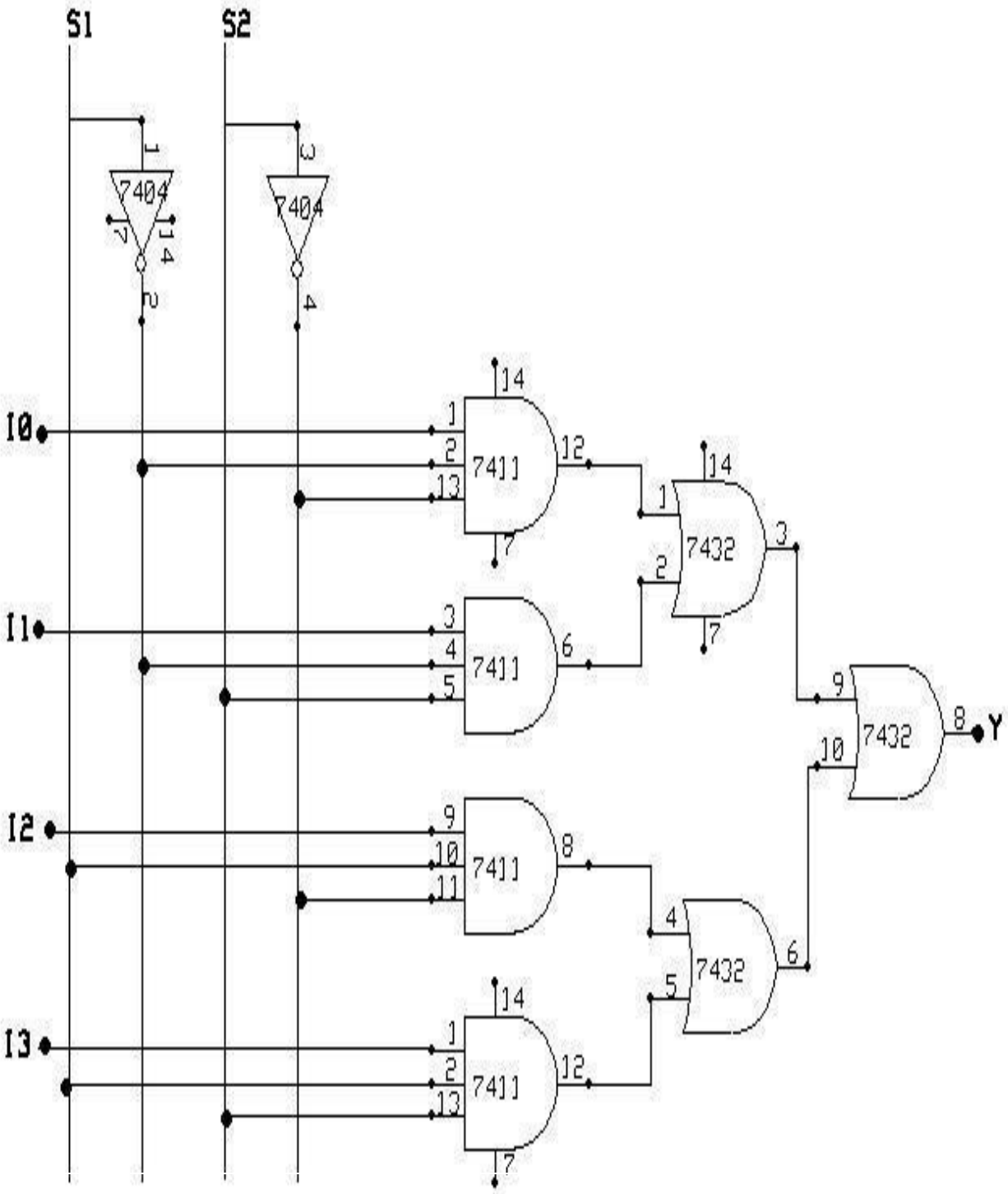
Sl. No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	OR gate	IC 7432	1
3.	NOT gate	IC 7404	1
4.	AND gate (three input)	IC 7411	1
5.	Connecting wires		As required

Theory:

Multiplexing means transmitting a large number of information units over a smaller number of channels or lines. A digital 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 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 determines which input is selected. A multiplexer is called a data selector, since it selects one of many inputs and steers the binary information to the output line. A Strobe is also provided to allow the designer to disable all output data until a specified time. Then, by allowing the STROBE to go low, the proper lead can be selected. This feature is very useful where data might be changing the same time DATA SELECT leads change. It is a very useful Medium Scale Integration (MSI) function and has a multitude of applications. It is used for connecting two or more sources to a single destination among the computer units and it is useful for constructing a common bus system. A decoder with an enable input can function as a demultiplexer. A Demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of specific output line is controlled by the bit values of n selection lines. The decoder and demultiplexer operations are obtained from the same circuit; a decoder with an enable input is referred to as a decoder / de-multiplexer. The Strobe lead can be used to active or de-active the entire IC, allowing time for the address lines to change the information is fed to the output. Demultiplexers are useful anytime information from one source must be fed several places.

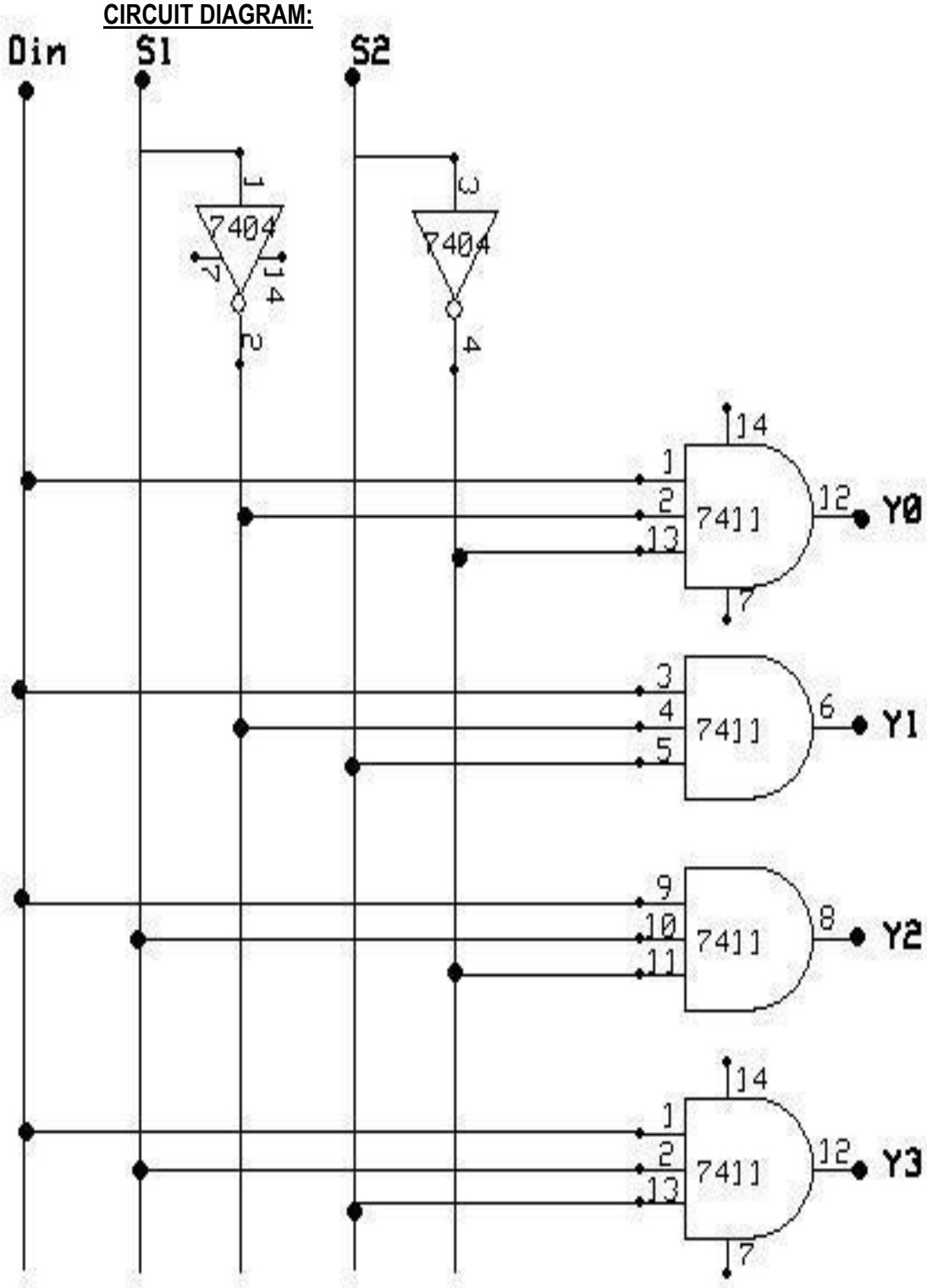
4 X 1 MULTIPLEXER

CIRCUIT DIAGRAM:



1X4 DEMULTIPLEXER

CIRCUIT DIAGRAM:



Result:

The design of the 4x1 Multiplexer and 1x4 Demultiplexer circuits was done and their truth tables were verified.

Outcome:

At the completion of an experiment student will able to design the multiplexer and the demultiplexer

Viva – Voce

1. What is multiplexer?
2. What is demultiplexer?
3. What are the advantages of multiplexer?
4. What are the advantages of demultiplexer?
5. What is select signal?
6. How to choose select signal in multiplexer?
7. How to choose select signal in demultiplexer?
8. Write the formula used in select signal.
9. What is the difference between the multiplexer and demultiplexer?
10. What is the application of multiplexer?
11. What is the application of demultiplexer?
12. Draw the truth table of multiplexer.
13. Draw the truth table of demultiplexer.
14. How many select signals are needed in 8*1 multiplexer?
15. How many select signals are needed in 8*1 demultiplexer?

EXP NO: 11 SHIFT REGISTER

Aim:

To design and implement the various shift register

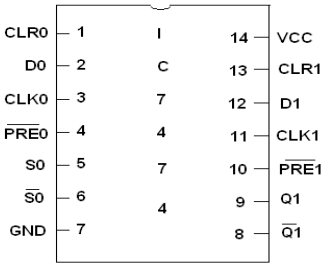
Apparatus required:

Sl. No	Name of the Apparatus	Range	Quantity
1.	D flip flop	IC 7474	2
2.	OR gate	IC 7432	1
3.	IC Trainer kit		1
5.	Connecting wires		As required

Theory:

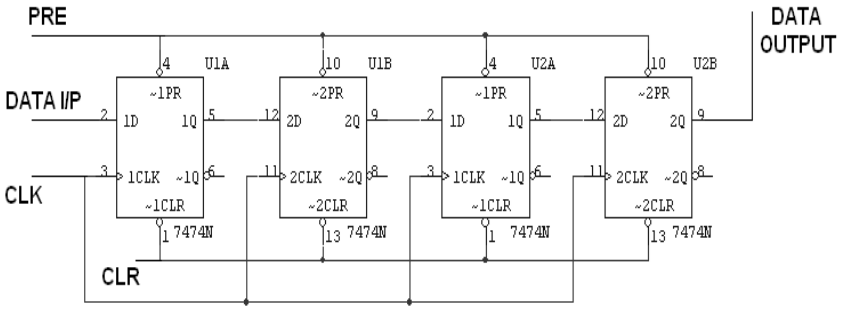
A register is capable of shifting its binary information in one or both directions is known as shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop.The simplest possible shift register is one that uses only flip flop. The output of a given flip flop is connected to the input of next flip flop of the register. Each clock pulse shifts the content of register one bit position to right.

PIN Diagram:



Logic Diagram:

SERIAL IN SERIAL OUT

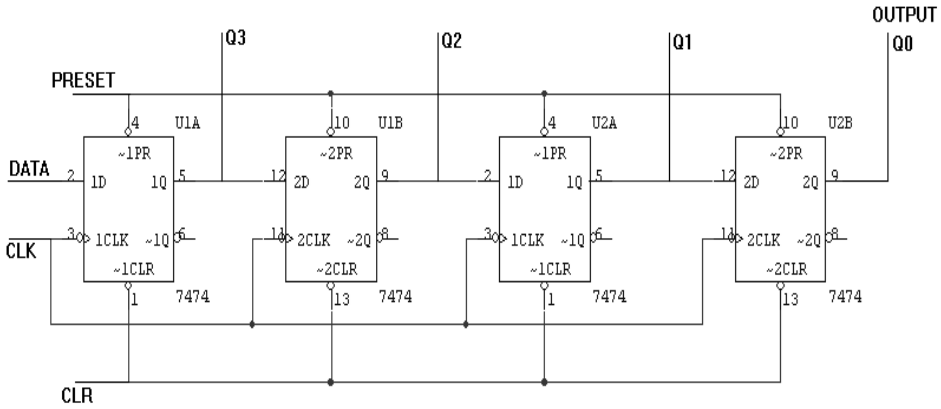


Truth Table:

CLK	Serial in	Serial out
1	1	0
2	0	0
3	0	0
4	1	1
5	X	0
6	X	0
7	X	1

Logic Diagram:

Serial in parallel out:



Truth Table:

CLK	DATA	OUTPUT			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

Parallel in Serial Out:

Truth Table:

CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1

Parallel in Parallel Out:

PARALLEL IN PARALLEL OUT:

CLR

Truth Table:

CLK	DATA INPUT				OUTPUT			
	D_A	D_B	D_C	D_D	Q_A	Q_B	Q_C	Q_D
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0

Procedure:

1. Connections are given as per circuit diagram
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

Result:

Thus the implementation of shift registers using flip flops was completed successfully.

Outcome:

At the completion of an experiment student will able to design the various types of shift register.

Expt.No.12: SYNCHRONOUS UP/DOWN COUNTER

Aim:

To design and implement 3 bit synchronous up/down counter

Apparatus required:

S.No	Name of the Apparatus	Range	Quantity
1.	JK Flip Flop	IC 7474	2
2.	OR gate	IC 7432	1
3.	NOT gate	IC 7404	1
4.	AND gate (three input)	IC 7411	1
5	XOR gate	IC 7486	1
6	IC Trainer Kit		1
7.	Connecting wires		As required

Theory:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. An up/down counter is one that is capable of progressing in increasing order or decreasing order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by up/down signal. When this signal is high counter goes through up sequence and when up/down signal is low counter follows reverse sequence.

K map:

JA = UD QB QC + UD QB QC

State Diagram:

Characteristic Table:

Q	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Logic Diagram:

74LS11

Truth Table:

Input Up/Down	Present State			Next State			A		B		C	
	Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	1	1	1	1	X	1	X	1	X
0	1	1	1	1	1	0	X	0	X	0	X	1
0	1	1	0	1	0	1	X	0	X	1	1	X
0	1	0	1	1	0	0	X	0	0	X	X	1
0	1	0	0	0	1	1	X	1	1	X	1	X
0	0	1	1	0	1	0	0	X	X	0	X	1
0	0	1	0	0	0	1	0	X	X	1	1	X
0	0	0	1	0	0	0	0	X	0	X	X	1
1	0	0	0	0	0	1	0	X	0	X	1	X
1	0	0	1	0	1	0	0	X	1	X	X	1
1	0	1	0	0	1	1	0	X	X	0	1	X
1	0	1	1	1	0	0	1	X	X	1	X	1
1	1	0	0	1	0	1	X	0	0	X	1	X
1	1	0	1	1	1	0	X	0	1	X	X	1
1	1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	0	0	X	1	X	1	X	1

Procedure:

1. Connections are given as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

Result:

Thus the 3-bit synchronous up/down counters was implemented successfully.

Outcome:

At the completion of an experiment student will able to design the synchronous up/down counter.

Expt.No.12:

SIMULATION OF COMBINATIONAL CIRCUITS USING HDL

Aim:

To write a verilog code for half adder, full adder and multiplexer

Tools Required:

Xilinx 9.2

Program:

Simulation wave for half adder

MULTIPLEXER:

Procedure:

1. Write and draw the Digital logic system.
2. Write the Verilog code for above system.
3. Enter the Verilog code in Xilinx software.
4. Check the syntax and simulate the above Verilog code (using ModelSim or Xilinx) and verify the output waveform as obtained.
5. Implement the above code in Spartan III using FPGA kit.

Result:

Thus the verilog code for half adder, full adder and multiplexer were simulated and verified successfully.

Outcome:

At the completion of an experiment student will able to be known the verilog code for half adder, full adder and multiplexer.

Expt.No.13:

SIMULATION OF SEQUENTIAL CIRCUITS USING HDL

Aim:

To write a verilog code for RS, D, JK flip flop and up counter

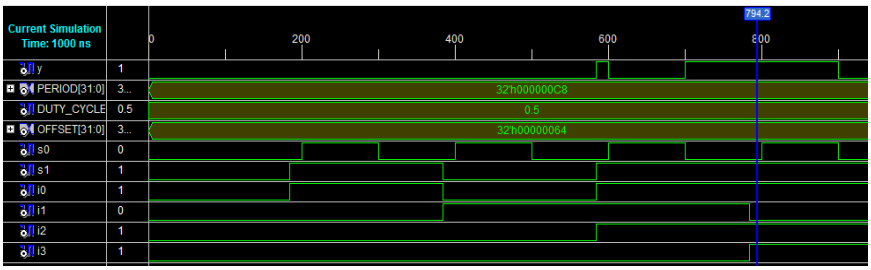
Tools required:

Xilinx 9.2

Program:

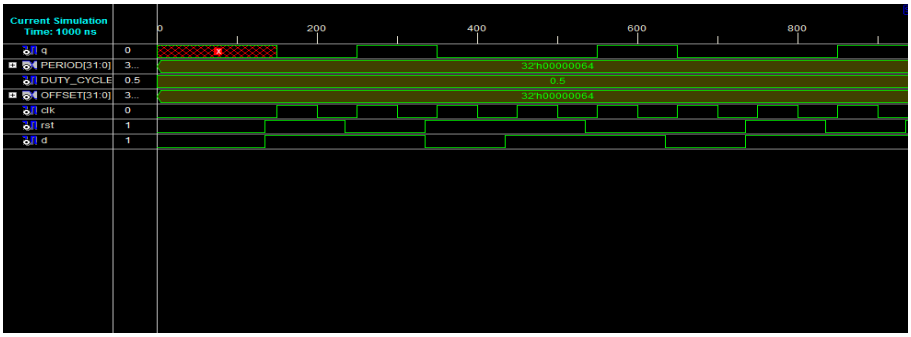
RS Flip Flop:

```
module sr1(clk, s, r, q);
    input clk;
    input s;
    input r;
    output reg q;
    always@(posedge clk or s or r)
    begin
    if(clk==1)
    if(s==1 && r==0)
    q<=1;
    else if(s==0 && r==0)
    q<=q;
    else
    q<=0;
    end
endmodule
```



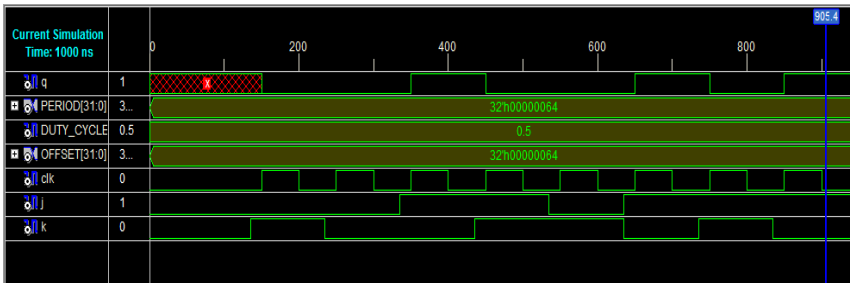
D Flip Flop:

```
module dff1(d, clk, rst, q);
    input d;
    input clk;
    input rst;
    output q;
    reg q;
    always@(posedge clk)
    if(rst)
    q<=0;
    else
    q<=d;
endmodule
```



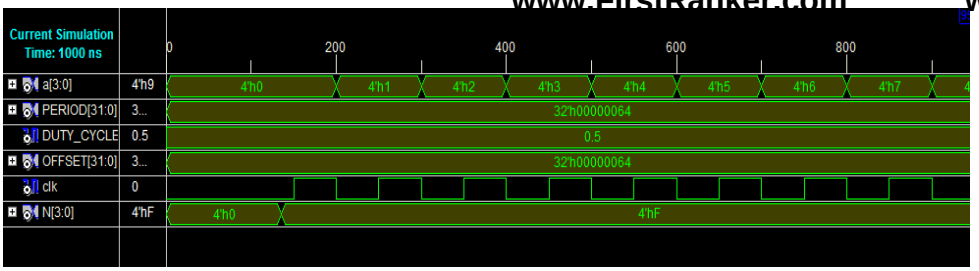
JK Flip Flop:

```
module jkflip(clk, j, k, q);
     input clk;
     input j;
     input k;
     output reg q;
     always@(posedge clk or j or k)
     begin
     if(clk==1)
     if(j==1 && k==0)
     q<=1;
     else if(j==0 && k==0)
     q<=q;
     else if(j==0 && k==1)
     q<=0;
     else
     q<=~q;
     end
endmodule
```



Up Counter:

```
module upcount(clk, N, a);
     input clk;
     input [3:0] N;
     output reg [3:0] a;
     initial a=4'b0000;
     always @(negedge clk)
     a=(a==N)?4'b0000: a+1'b1;
endmodule
```



Result:

Thus the verilog code for RS,D,JK Filp Flop and up counter were simulated and verified successfully.

Outcome:

At the completion of an experiment student will able to be known the verilog code for RS, D, JK Filp Flop and up counter .

ADDITIONAL EXPERIMENTS

Expt.No.8: ENCODER AND DECODER

Aim:

To study the operation of encoder and decoder circuits using logic gates

Apparatus required:

S. No	Name of the Apparatus		Range	Quantity
1.	Digital IC trainer			1
2.	NOT Gate	IC 7404		1
3.	OR Gate	IC 7432		1
4.	AND Gate	IC7408		1
5.	Bread Board			1
6.	NOT Gate	IC7404		1
8.	Connecting wires and probes			As required

Theory:

Decoder

In digital electronics, a decoder can take the form of a multiple-input, multiple-output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different e.g. n-to-2n , binary-coded decimal decoders. Decoding is necessary in applications such as data multiplexing, 7 segment display and memory address decoding.

The example decoder circuit would be an AND gate because the output of an AND gate is "High" (1) only when all its inputs are "High." Such output is called as "active High output". If instead of AND gate, the NAND gate is connected the output will be "Low" (0) only when all its inputs are "High". Such output is called as "active low output".

A slightly more complex decoder would be the n-to-2n type binary decoders. These types of decoders are combinational circuits that convert binary information from 'n' coded inputs to a maximum of 2n unique outputs. In case the 'n' bit coded information has unused bit combinations, the decoder may have less than 2n outputs. 2-to-4 decoder, 3-to-8 decoder or 4-to-16 decoder are other examples.

The input to a decoder is parallel binary number and it is used to detect the presence of a particular binary number at the input. The output indicates presence or absence of specific number at the decoder input. An encoder is a device, circuit, transducer, software program, algorithm or person that converts information from one format or code to another. The purpose of encoder is standardization, speed, secrecy, security, or saving space by shrinking size. Encoders are combinational logic circuits and they are exactly opposite of decoders. They accept one or more inputs and generate a multibit output code. Encoders perform exactly reverse operation than decoder. An encoder has M input and N output lines. Out

of M input lines only one is activated at a time and produces equivalent code on output N lines. If a device output code has fewer bits than the input code has, the device is usually called an encoder

Sl.No.	Inputs		Outputs			
	A	B	Y3	Y2	Y1	Y0
1.	0	0	0	0	0	1
2.	0	1	0	0	1	0
3.	1	0	0	1	0	0
4.	1	1	1	0	0	0

Sl.No.	Inputs								Outputs		
	D7	D6	D5	D4	D3	D2	D1	D0	A	B	C
1.	0	0	0	0	0	0	0	1	0	0	0
2.	0	0	0	0	0	0	1	0	0	0	1
3.	0	0	0	0	0	1	0	0	0	1	0
4.	0	0	0	0	1	0	0	0	0	1	1
5.	0	0	0	1	0	0	0	0	1	0	0
6.	0	0	1	0	0	0	0	0	1	0	1
7.	0	1	0	0	0	0	0	0	1	1	0
8.	1	0	0	0	0	0	0	0	1	1	1

Procedure:

1. Make the circuit connections as shown in the figure.
2. Check the corresponding truth table.

Result:

The design of the Encoder and Decoder circuit was done and the input and output were obtained

Outcome:

At the completion of an experiment student will able to design the encoder circuit and the decoder circuit

Viva – Voce

1. What is Encoder?
2. What is decoder?
3. List the application of encoder.
4. List the application of decoder.
5. Draw the truth table of encoder.
6. Draw the truth table of decoder.
7. What are logic gates used encoder?
8. What are logic gates used encoder?
9. What is the difference between decoder with demultiplexer?
10. What is the difference between encoder with multiplexer?
11. How to choose the select signal in encoder?
12. How to choose the select signal in decoder?
13. Draw the logic diagram of encoder.
14. Draw the logic diagram of encoder.
15. What is the difference between encoder with decoder?

Expt.No.2: IMPLEMENTATION OF BOOLEAN FUNCTIONS

Aim:

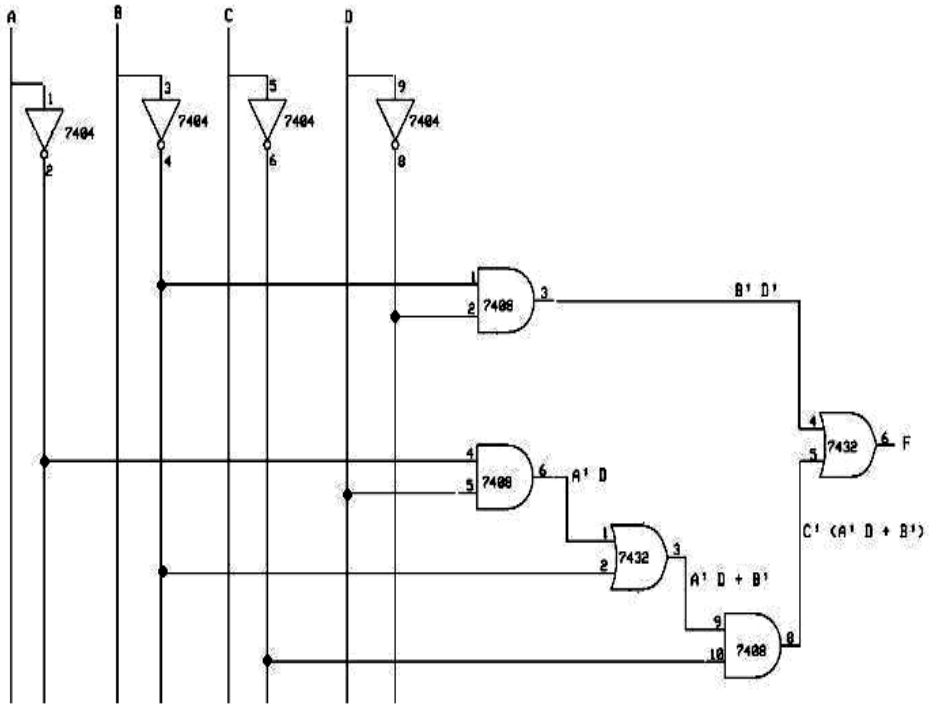
To design the logic circuit and verify the truth table of the given Boolean expression,

F (A, B, C, D) = Σ (0, 1, 2, 5, 8, 9, 10)

Apparatus required:

Sl.No	Name of the Apparatus	Range	Quantity
1.	Digital IC trainer kit		1
2.	AND gate	IC 7408	1
3.	OR gate	IC 7432	1
4.	NOT gate	IC 7404	1
5.	NAND gate	IC 7400	1
6.	NOR gate	IC 7402	1
7.	EX-OR gate	IC 7486	1
8.	Connecting wires		As required

Circuit diagram:



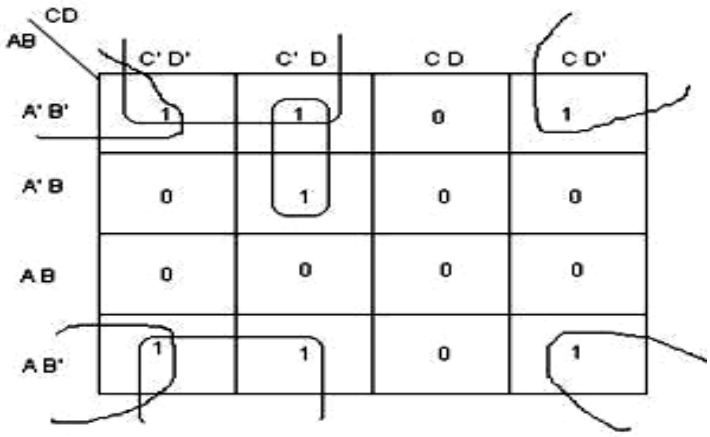
Design:

Given , F (A,B,C,D) = Σ (0,1,2,5,8,9,10)

Truth table:

Sl. No.	INPUT				OUTPUT
	A	B	C	D	F=D'B'+C'(B'+A'D)
1.	0	0	0	0	1
2.	0	0	0	1	1
3.	0	0	1	0	1
4.	0	0	1	1	0
5.	0	1	0	0	0
6.	0	1	0	1	1
7.	0	1	1	0	0
8.	0	1	1	1	0
9.	1	0	0	0	1
10.	1	0	0	1	1
11.	1	0	1	0	1
12.	1	0	1	1	0
13.	1	1	0	0	0
14.	1	1	0	1	0
15.	1	1	1	0	0
16.	1	1	1	1	0

The output function F has four input variables hence a four variable Karnaugh Map is used to obtain a simplified expression for the output as shown,



From the K-Map,

F = B' C' + D' B' + A' C' D

Since we are using only two input logic gates the above expression can be re-written as,

F = C' (B' + A' D) + D' B'

Now the logic circuit for the above equation can be drawn.

Procedure:

1. Connections are given as per the circuit diagram.
2. For all the IC's 7^{th} pin is grounded and 14^{th} pin is given +5 V supply.
3. Apply the inputs and verify the truth table for the given Boolean expression.

Result:

The truth table of the given Boolean expression was verified.

Outcome:

At the completion of an experiment student will able to design the Boolean expression.