# COMPUTER SYSTEMS ARCHITECUTURE

# LAB MANUAL



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION SYSTEMS**

**KARPAGAM UNIVERSITY**

**COIMBATORE-641021**

## LIST OF PRACTICALS

1. Verification of logic gates
2. code converters
3. Realization of multiplexers using logic gates
4. Encoder and Decoder
5. Adders using logic gates.
6. Subtractors using logic gates.
7. Parity generator
8. Flip-Flops.



9. EX. NO:1 Verification of logic gates

**AIM:**

To study about logic gates and verify their truth tables.

**APPARATUS REQUIRED:**

| SL No. | COMPONENT | SPECIFICATION | QTY |
|---|---|---|---|
| 1. | AND GATE | IC 7408 | 1 |
| 2. | OR GATE | IC 7432 | 1 |
| 3. | NOT GATE | IC 7404 | 1 |
| 4. | NAND GATE 2 I/P | IC 7400 | 1 |
| 5. | NOR GATE | IC 7402 | 1 |
| 6. | X-OR GATE | IC 7486 | 1 |
| 7. | NAND GATE 3 I/P | IC 7410 | 1 |
| 8. | IC TRAINER KIT | - | 1 |
| 9. | PATCH CORD | - | 14 |

**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:**

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.

**X-OR GATE:**

The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

**PROCEDURE:**

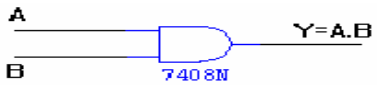
(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

**AND GATE:**

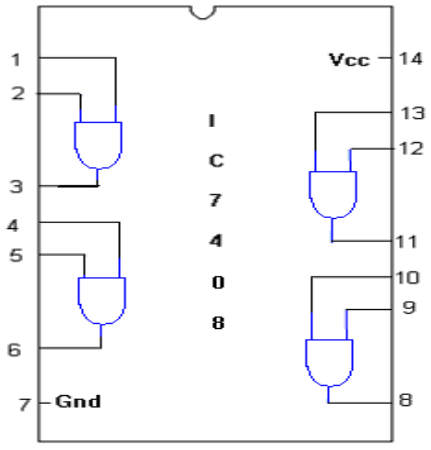
**SYMBOL:**




TRUTH TABLE

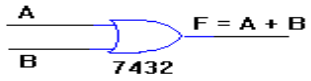
| A | B | A.B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**PIN DIAGRAM:**

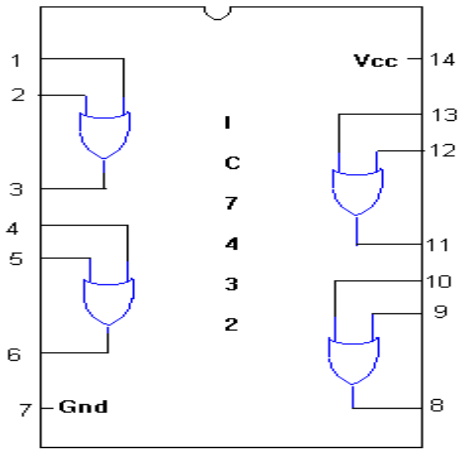


## OR GATE:

SYMBOL :




PIN DIAGRAM :

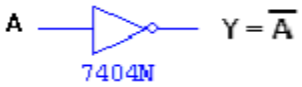



TRUTH TABLE

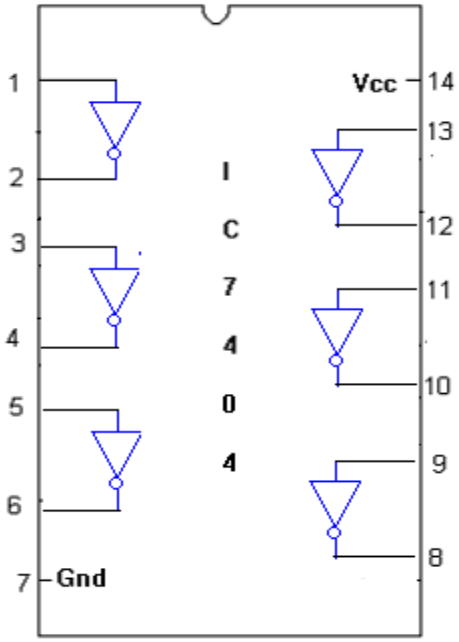
| A | B | A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## NOT GATE:

## SYMBOL:




## PIN DIAGRAM:

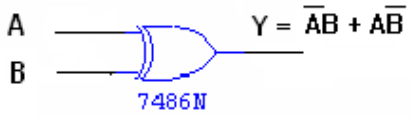



TRUTH TABLE :

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

## X-OR GATE :

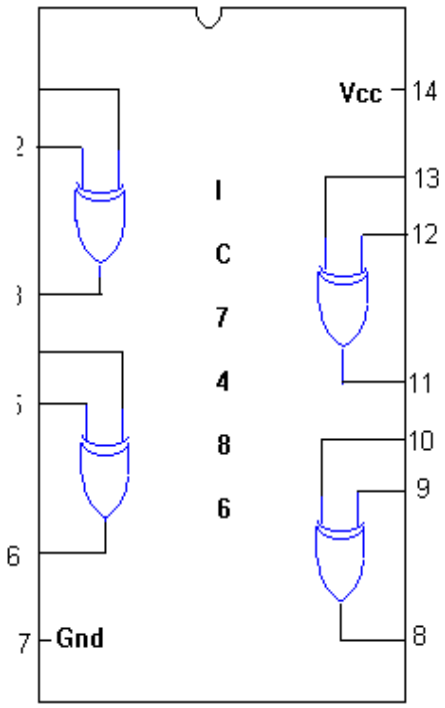
**SYMBOL :**




TRUTH TABLE :

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

**PIN DIAGRAM :**



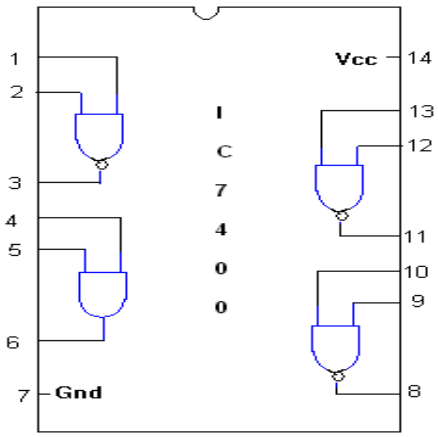

## 2-INPUT NAND GATE:

**SYMBOL:**


TRUTH TABLE

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

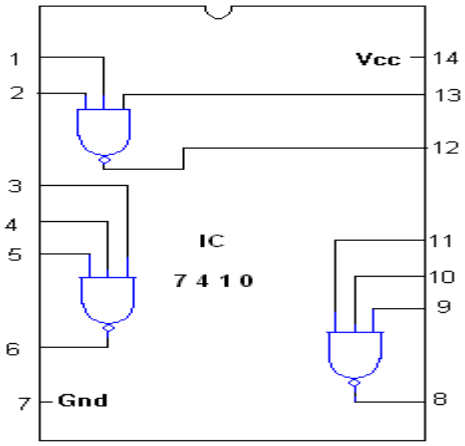
**PIN DIAGRAM:**




## 3-INPUT NAND GATE :

SYMBOL :


PIN DIAGRAM :

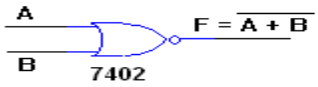



TRUTH TABLE

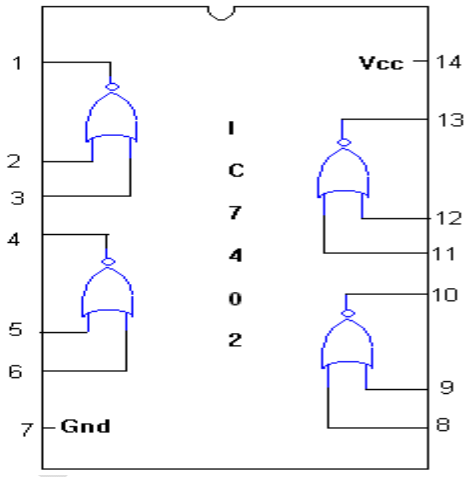
| A | B | C | $\overline{A.B.C}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

## NOR GATE:

SYMBOL :




PIN DIAGRAM :




TRUTH TABLE

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

**RESULT:**

**EX. NO: 2**

## CODE CONVERTOR

**AIM:**

To design and implement 4-bit

(i) Binary to gray code converter

(ii) Gray to binary code converter

**APPARATUS REQUIRED:**

| Sl.No. | COMPONENT | SPECIFICATION | QTY. |
|---|---|---|---|
| 1. | X-OR GATE | IC 7486 | 1 |
| 2. | AND GATE | IC 7408 | 1 |
| 3. | OR GATE | IC 7432 | 1 |
| 4. | NOT GATE | IC 7404 | 1 |
| 5. | IC TRAINER KIT | - | 1 |
| 6. | PATCH CORDS | - | 35 |

**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.

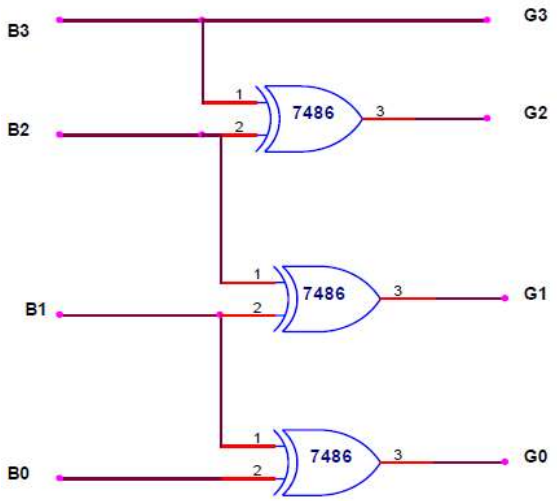
**Procedure: -**

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

## LOGIC DIAGRAM:

**Circuit Diagrams:**

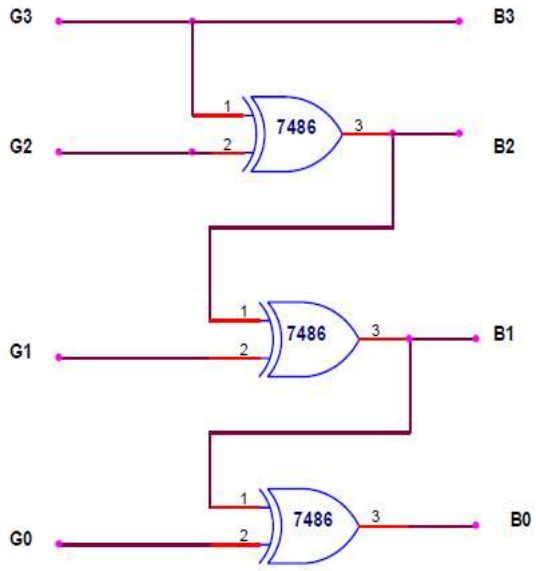
### Binary to Gray Code Converter




**Truth table**

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

### Gray to Binary Code Converter




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

**RESULT:**

**EX. NO. 3**

## REALIZATION OF MULTIPLEXER USING LOGIC GATES

**AIM:**

To design and implement multiplexer using logic gates

**APPARATUS REQUIRED:**

| Sl.No. | COMPONENT | SPECIFICATION | QTY. |
|---|---|---|---|
| 1. | 2/3 I/P AND GATE | IC 7408/IC7411 | 4/2 |
| 2. | OR GATE | IC 7432 | 1 |
| 3. | NOT GATE | IC 7404 | 1 |
| 2. | IC TRAINER KIT | - | 1 |
| 3. | PATCH CORDS | - | 32 |

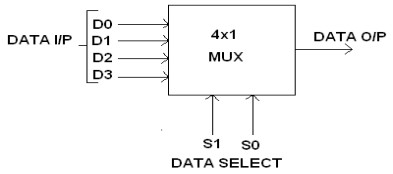
**THEORY:**

**MULTIPLEXER:**

Multiplexer 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 a particular input line is controlled by a set of selection lines. Normally there are $2^n$ input line and n selection lines whose bit combination determine which input is selected.

**BLOCK DIAGRAM FOR 4:1 MULTIPLEXER:**

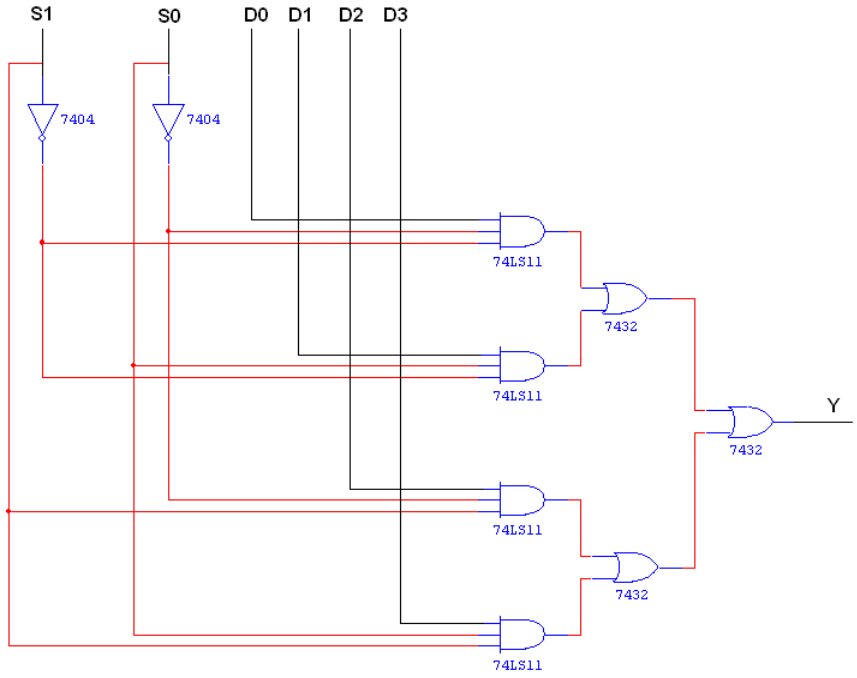



**FUNCTION TABLE:**

| S1 | S0 | INPUTS Y |
|---|---|---|
| 0 | 0 | D0 → D0 S1' S0' |
| 0 | 1 | D1 → D1 S1' S0 |
| 1 | 0 | D2 → D2 S1 S0' |
| 1 | 1 | D3 → D3 S1 S0 |

**Y = D0 S1' S0' + D1 S1' S0 + D2 S1 S0' + D3 S1 S0**

**CIRCUIT DIAGRAM FOR MULTIPLEXER:**



**TRUTH TABLE:**

| S1 | S0 | Y = OUTPUT |
|---|---|---|
| 0 | 0 | D0 |
| 0 | 1 | D1 |
| 1 | 0 | D2 |
| 1 | 1 | D3 |

**RESULT:**

**EX. NO. 4**

## ENCODER AND DECODER

**AIM:**

To design and implement encoder and decoder using logic gates.

**APPARATUS REQUIRED:**

| Sl.No. | COMPONENT | SPECIFICATION | QTY. |
|---|---|---|---|
| 1. | 2 I/P NAND GATE | IC 7408 | 2 |
| 2. | OR GATE | IC 7432 | 3 |
| 3. | NOT GATE | IC 7404 | 1 |
| 2. | IC TRAINER KIT | - | 1 |
| 3. | PATCH CORDS | - | 27 |

**THEORY:**

**ENCODER:**

An encoder is a digital circuit that perform inverse operation of a decoder. An encoder has $2^n$ input lines and n output lines. In encoder the output lines generates the binary code corresponding to the input value. In octal to binary encoder it has eight inputs, one for each octal digit and three output that generate the corresponding binary code. In encoder it is assumed that only one input has a value of one at any given time otherwise the circuit is meaningless. It has an ambiguila that when all inputs are zero the outputs are zero. The zero outputs can also be generated when D0 = 1.

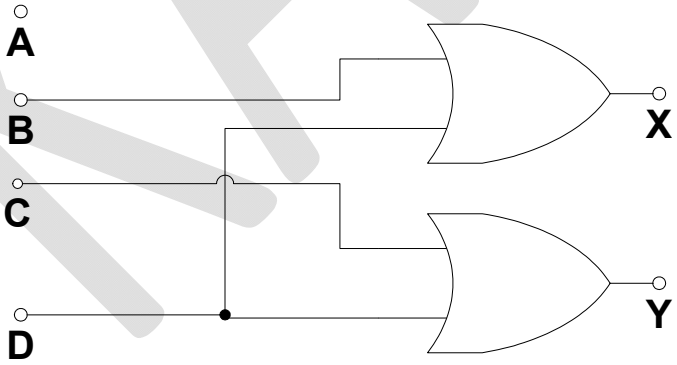
**DECODER:**

A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each input code word produces a different output code word i.e there is one to one mapping can be expressed in truth table. In the block diagram of decoder circuit the encoded information is present as n input producing $2^n$ possible outputs. $2^n$ output values are from 0 through out $2^n - 1$.

**PIN DIAGRAM FOR IC 7445:**

**BCD TO DECIMAL DECODER:**

**LOGIC DIAGRAM FOR ENCODER:**



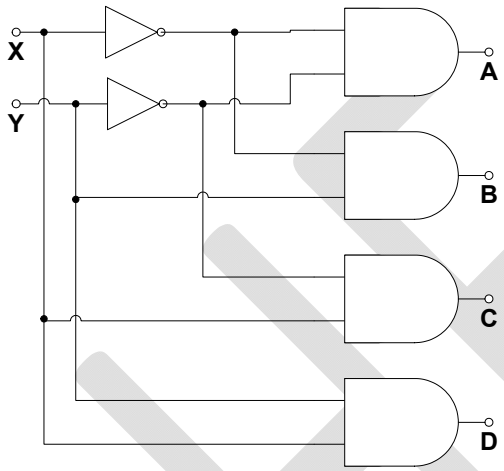

**TRUTH TABLE:**

| Inputs | | | | Outputs | |
|---|---|---|---|---|---|
| A | B | C | D | Y | X |
| 1 | 0 | 0 | 0 | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

**LOGIC DIAGRAM FOR DECODER:**

Circuit diagram:




**TRUTH TABLE:**

| Inputs | | Outputs | | | |
|---|---|---|---|---|---|
| X | Y | A | B | C | D |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

**PROCEDURE:**

(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

**RESULT:**

**EX. NO. 5 ADDER USING GATES**

**AIM:**

To design and construct half adder, full adder, half subtractor and full subtractor circuits and verify the truth table using logic gates.

**APPARATUS REQUIRED:**

| Sl.No. | COMPONENT | SPECIFICATION | QTY. |
|---|---|---|---|
| 1. | AND GATE | IC 7408 | 1 |
| 2. | X-OR GATE | IC 7486 | 1 |
| 3. | NOT GATE | IC 7404 | 1 |
| 4. | OR GATE | IC 7432 | 1 |
| 3. | IC TRAINER KIT | - | 1 |
| 4. | PATCH CORDS | - | 23 |

**THEORY:**

**HALF ADDER:**

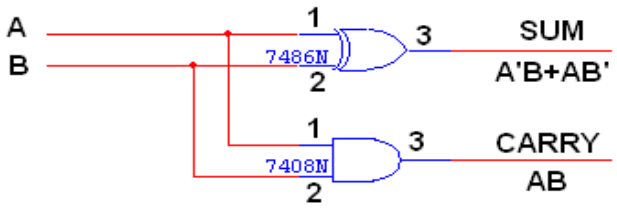
A half adder has two inputs for the two bits to be added and two outputs one from the sum ' S' and other from the carry ' c' into the higher adder position. Above circuit is called as a carry signal from the addition of the less significant bits sum from the X-OR Gate the carry out from the AND gate.

**FULL ADDER:**

A full adder is a combinational circuit that forms the arithmetic sum of input; it consists of three inputs and two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR Gate, carry output will be taken from OR Gate.

**LOGIC DIAGRAM:**

**HALF ADDER**



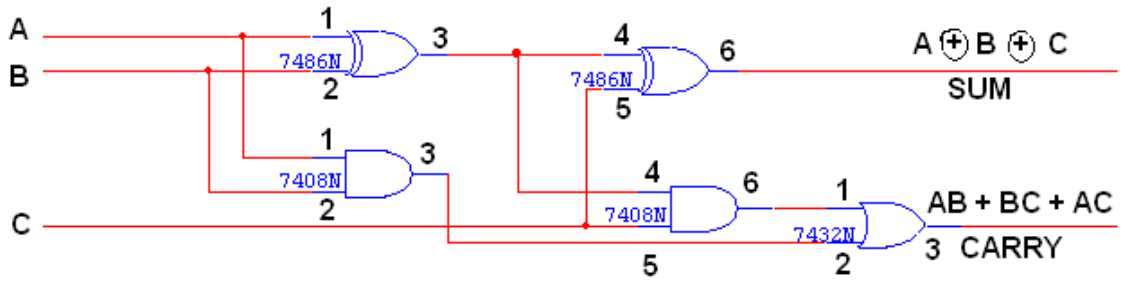

**TRUTH TABLE:**

| A | B | CARRY | SUM |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**LOGIC DIAGRAM:**

**FULL ADDER**

**FULL ADDER USING TWO HALF ADDER**




**TRUTH TABLE:**

| A | B | C | CARRY | SUM |
|---|---|---|---|---|
| 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 |

**RESULT:**

**EX. NO. 6**

## SUBTRACTOR USING GATES

**AIM:**

To design and construct half subtractor and full subtractor circuits and verify the truth table using logic gates.

**APPARATUS REQUIRED:**

| Sl.No. | COMPONENT | SPECIFICATION | QTY. |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1. | AND GATE | IC 7408 | 1 |
| 2. | X-OR GATE | IC 7486 | 1 |
| 3. | NOT GATE | IC 7404 | 1 |
| 4. | OR GATE | IC 7432 | 1 |
| 3. | IC TRAINER KIT | - | 1 |
| 4. | PATCH CORDS | - | 23 |

**THEORY:**

**HALF SUBTRACTOR:**

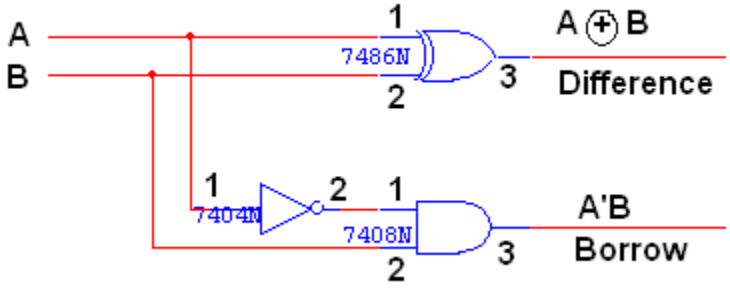
The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

**FULL SUBTRACTOR:**

The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs. The two half subtractor put together gives a full subtractor .The first half subtractor will be C and A B. The output will be difference output of full subtractor. The expression AB assembles the borrow output of the half subtractor and the second term is the inverted difference output of first X-OR.

**LOGIC DIAGRAM:**

**HALF SUBTRACTOR**




**TRUTH TABLE:**

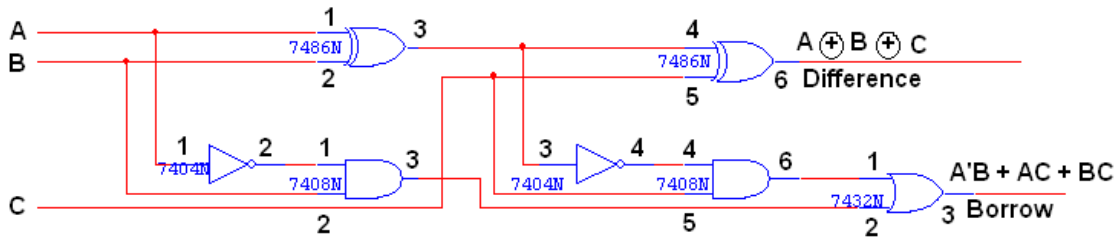
| A | B | BORROW | DIFFERENCE |
|---|---|---|---|
| | | | |

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

**LOGIC DIAGRAM:**

**FULL SUBTRACTOR**

**FULL SUBTRACTOR USING TWO HALF SUBTRACTOR:**




**TRUTH TABLE:**

| A | B | C | BORROW | DIFFERENCE |
|---|---|---|---|---|
| 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 |

**RESULT:**

**EXPT NO. :7**

## PARITY CHECKER /GENERATOR

**AIM:**

To design and implement parity checker generator using logic gate.

**APPARATUS REQUIRED:**

| Sl.No. | COMPONENT | SPECIFICATION | QTY. |
|---|---|---|---|
| 1. | NOT GATE | IC 7404 | 1 |
| 1. | XOR GATE | IC 7486 | 2 |
| 2. | IC TRAINER KIT | - | 1 |
| 3. | PATCH CORDS | - | 30 |

**THEORY:**

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

In even parity, the added parity bit will make the total number is even amount. In odd parity, the added parity bit will make the total number is odd amount. The parity checker circuit checks for possible errors in the transmission. If the information is passed in even parity, then the bits required must have an even number of 1's. An error occur during transmission, if the received bits have an odd number of 1's indicating that one bit has changed in value during transmission.

**FUNCTION TABLE:**

| **INPUTS** | | | **OUTPUTS** | |
|---|---|---|---|---|
| **Number of High Data Inputs (I0 – I7)** | **PE** | **PO** | **∑E** | **∑O** |
| **EVEN** | **1** | **0** | **1** | **0** |
| **ODD** | **1** | **0** | **0** | **1** |
| **EVEN** | **0** | **1** | **0** | **1** |
| **ODD** | **0** | **1** | **1** | **0** |
| **X** | **1** | **1** | **0** | **0** |
| **X** | **0** | **0** | **1** | **1** |

**PROCEDURE:**

(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

**RESULT:**