Experiment - 1

Exp. Date:

Aim: Study of all the Logic Gates.

Requirement: Digital IC Trainer Kit, Bread Board, Multi-meter, connecting wires, power supply, IC 7400, IC 7402, IC 7404, IC 7408, IC 7432 IC 7486

Procedure/ Design Steps:

1. Place the ICs on the Bread board or ICs base of digital trainer kit.
2. Provide +5 volts supply to pin no. 14 and connect pin no. 7 to the ground for all the ICs.
3. Do the connections as per the ICs pin no. shown in circuit diagram.
4. Verify the truth table of all the basic gates.

Theory:

Logic Gates:
Logic gates are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output. The relationship between the input and the output is based on certain logic. Based on this, logic gates are named as AND gate, OR gate, NOT gate etc.

Basic Gates:

1. The AND Gate (IC-7408)
   The AND gate is an electronic circuit that gives a true output (1) only if all its inputs are true. A dot (.) is used to show the AND operation i.e. A.B. Note that the dot is sometimes omitted i.e. AB

   Logic Diagram & Pin Diagram:

   Truth Table & Observation:

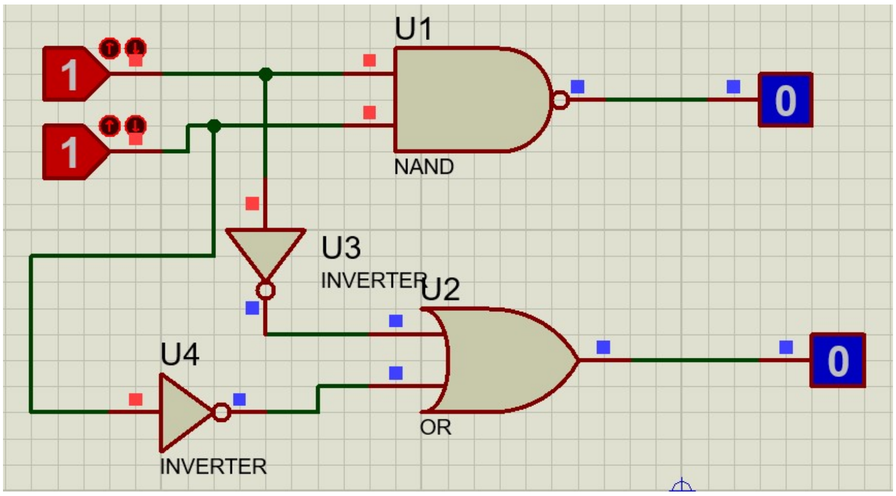


Figure 1: Proteus

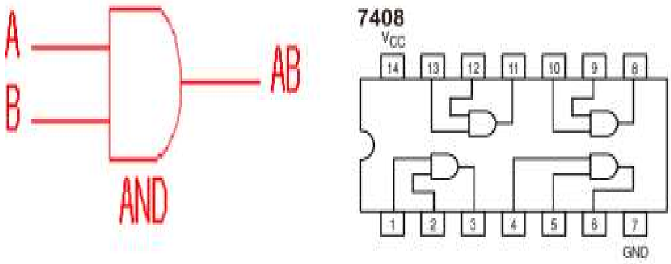


Figure 2: Logic Diagram of AND gate and Pin Diagram

Inputs		Output(T)	Output(P)
A	B	Y=A.B	Y=A.B
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Table 2: Truth Table for AND Gate

2. The OR Gate (IC-7432): The OR gate is an electronic circuit that gives a true output (1) if one or more of its inputs are true. A plus (+) is used to

show the OR operation.

Logic Diagram & Pin Diagram:

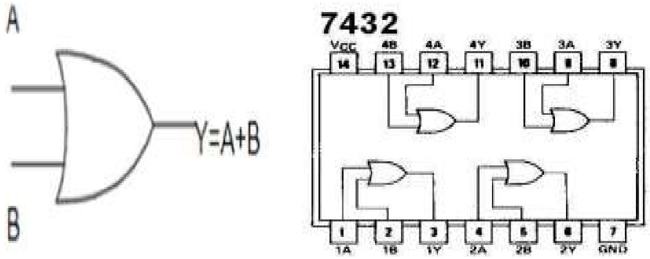


Figure 3: Logic Diagram of OR gate and Pin Diagram

Truth Table & Observation:

Inputs		Output(T)	Output(P)
A	B	Y=A+B	Y=A+B
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Table 3: Truth Table for OR Gate

3. The NOT Gate (IC-7404): The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A. This is also shown as A’, or *A* with a bar over the top.

Logic Diagram & Pin Diagram:

Truth Table & Observation:

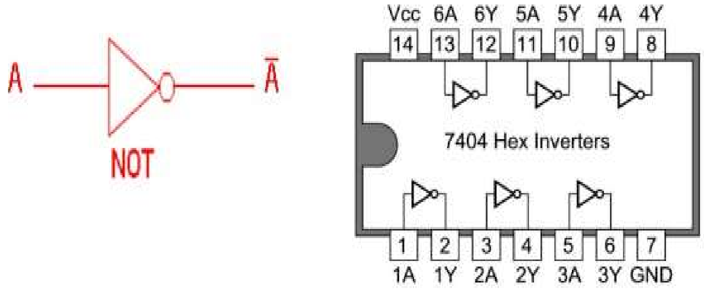


Figure 4: Logic Diagram of NOT gate and Pin Diagram

Inputs	Output(T)	Output(P)
A	Y=A	Y=A
0	1	1
1	0	0

Table 4: Truth Table for NOT Gate

4. The NAND Gate (IC-7400): This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

Logic Diagram & Pin Diagram:
Truth Table & Observation:

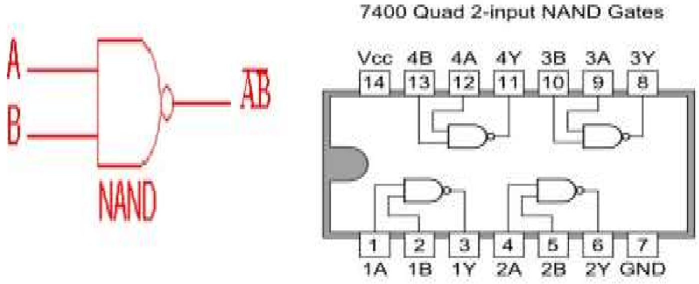


Figure 5: Logic Diagram of NAND gate and Pin Diagram

Inputs		Output(T)	Output(P)
A	B	Y=$\overline{A.B}$	Y=$\overline{A.B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Table 5: Truth Table for NAND Gate

5. The NOR gate (IC-7402):- This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

Logic Diagram & Pin Diagram:

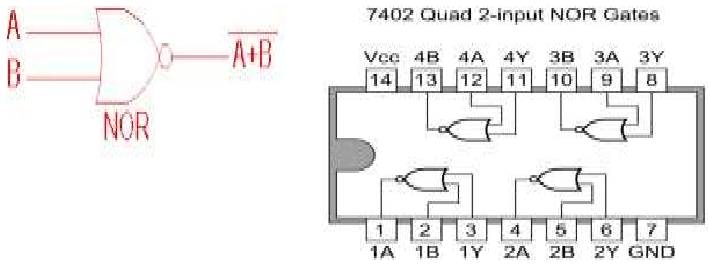


Figure 6: Logic Diagram of NOR gate and Pin Diagram

Truth Table & Observation:

6. EX-OR Gate(IC-7486):- The ’Exclusive-OR’ gate is a circuit which will give a high output if either, but not both, of its two inputs are high. An encircled plus sign (L) is used to show the EOR operation. XOR or Ex-OR gate is a special type of gate. It can be used in the half adder, full adder and subtractor. The exclusive-OR gate is abbreviated as EX-OR gate or sometime as X-OR gate. It has n input n >= 2 and one output.

Inputs		Output(T)	Output(P)
A	B	Y=$\overline{A+B}$	Y=$\overline{A+B}$
0	0	1	1
0	1	0	1
1	0	0	0
1	1	0	0

Table 6: Truth Table for NOR Gate

Boolean expression

$$Y = A^{\mathrm{M}} B$$

alternatively

$$Y = \bar{A}B + A\bar{B}$$

Logic Diagram & Pin Diagram:

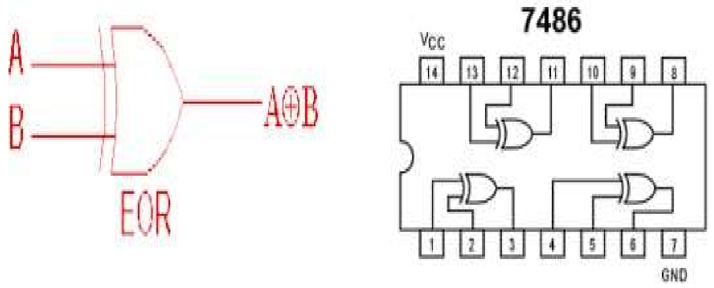


Figure 7: Logic Diagram of EXOR gate and Pin Diagram

Truth Table & Observation:

7. EX-NOR Gate(IC-74266):- The ’Exclusive-NOR’ gate circuit does the opposite to the EOR gate. It will give a low output if either, but not both, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

Inputs		Output(T)	Output(P)
A	B	$Y = A \,\text{L}\, B$	$Y = A \,\text{L}\, B$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Table 7: Truth Table for EXOR Gate

Boolean expression

$$Y = \overline{A \;\text{M}\; B}$$

alternatively

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

Logic Diagram & Pin Diagram:

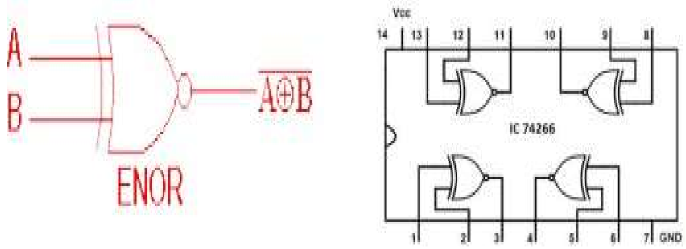


Figure 8: Logic Diagram of EX-NOR gate and Pin Diagram

Truth Table & Observation:

Result and Conclusion:

Inputs		Output(T)	Output(P)
A	B	$Y = \overline{A \oplus B}$	$Y = \overline{A \oplus B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

Table 8: Truth Table for EX-NOR Gate

....

....

....

Submission Date:

Professor In-charge

Experiment - 2

Exp. Date:

Aim: : Study the Design of Universal Gates.

Requirements: Digital IC Trainer Kit, Bread Board, Multi-meter, connecting wires, power supply, NAND IC 7400 and NOR IC 7402.

Procedure/ Design Steps:

1. Place the ICs on the Bread board or ICs base of digital trainer kit.
2. Provide +5 volts supply to pin no. 14 and connect pin no. 7 to the ground for all the ICs.
3. Do the connections as per the ICs pin no. shown in circuit diagram.
4. Verify the truth table of all the basic gates.

Theory:

Universal Gates:

A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families. In fact, an AND gate is typically implemented as a NAND gate followed by an inverter not the other way around. Likewise, an OR gate is typically implemented as a NOR gate followed by an inverter not the other way around.

1. The NAND Gate (IC-7400): This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

   Logic Diagram:

   Truth Table & Observation:

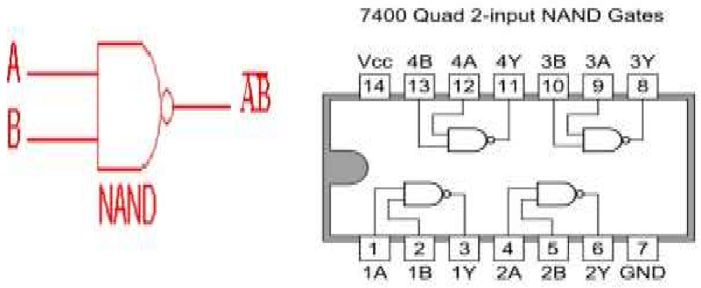


Figure 9: Logic Diagram of NAND gate and Pin Diagram

Inputs		Output(T)	Output(P)
A	B	Y=$\overline{A.B}$	Y=$\overline{A.B}$
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

Table 11: Truth Table for NAND Gate

2. The NOR gate (IC-7402):- This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

   Logic Diagram:

   Truth Table & Observation:

1. NAND Gate as a Universal gate
   To prove that any Boolean function can be implemented using only NAND gates, we will show that the AND, OR, and NOT operations can be performed using only NAND gates.

   (a) Implementing NOT Gate using NAND gate
   Logic Diagram:

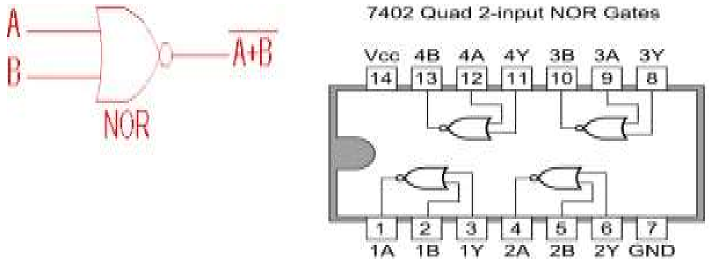


Figure 10: Logic Diagram of NOR gate and Pin Diagram

Inputs		Output(T)	Output(P)
A	B	$Y=\overline{A+B}$	$Y=\overline{A+B}$
0	0	1	
0	1	0	
1	0	0	
1	1	0	

Table 12: Truth Table for NOR Gate

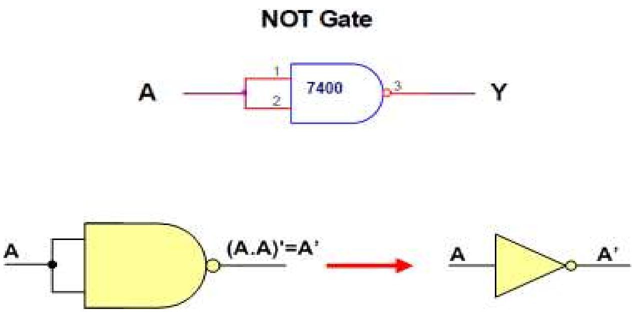


Figure 11: Logic Diagram of NOT gate using NAND gate

Truth Table & Observation:

(b) Implementing AND Using only NAND Gates

An AND gate can be replaced by NAND gates as shown in the figure (The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter).

Inputs	Output(T)	Output(P)
A	$Y = A$	$Y = A$
0	1	
1	0	

Table 13: Truth Table for NOT gate

Logic Diagram:

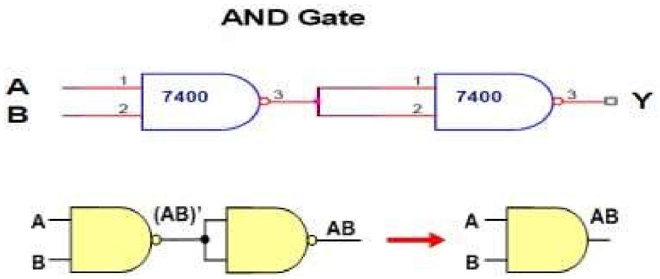


Figure 12: Logic Diagram of AND gate using NAND gate

Truth Table & Observation:

Inputs		Output(T)	Output(P)
A	B	Y =A.B	Y=A.B
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Table 14: Truth Table for AND Gate

(c) Implementing OR Using only NAND Gates
An OR gate can be replaced by NAND gates as shown in the figure (The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters).

Logic Diagram:

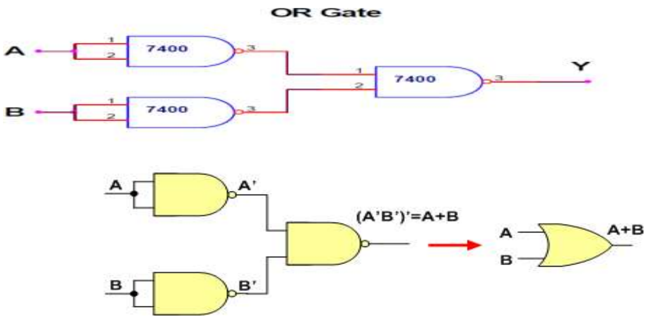


Figure 13: Logic Diagram of OR gate using NAND gate

Truth Table & Observation:

Inputs		Output(T)	Output(P)
A	B	Y =A+B	Y=A+B
0	0	0	
0	1	1	
1	0	1	
1	1	1	

Table 15: Truth Table for OR Gate

2. NOR Gate is a Universal Gate:
   To prove that any Boolean function can be implemented using only NOR gates, we will show that the AND, OR, and NOT operations can be performed using only NOR gates

   (a) Implementing an Inverter Using only NOR Gate

      i. Implementing NOT Gate using NOR gate
      Logic Diagram:

      Truth Table & Observation:

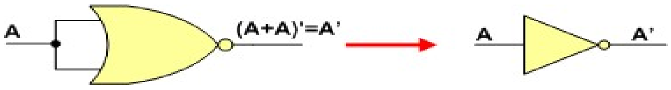


Figure 14: Logic Diagram of NOT gate using NOR gate

Inputs	Output(T)	Output(P)
A	$Y = A$	$Y = A$
0	1	
1	0	

Table 16: Truth Table for NOT gate

ii. Implementing OR Using only NOR Gates

Logic Diagram:

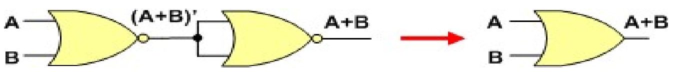


Figure 15: Logic Diagram of OR gate using NOR gate

Truth Table & Observation:

Inputs		Output(T)	Output(P)
A	B	Y =A+B	Y=A+B
0	0	0	
0	1	1	
1	0	1	
1	1	1	

Table 17: Truth Table for OR Gate

iii. Implementing AND Using only NOR Gates
An AND gate can be replaced by NOR gates as shown in the figure (The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters)

Logic Diagram:

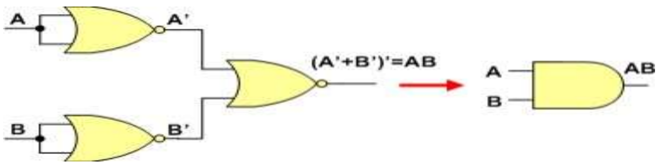


Figure 16: Logic Diagram of AND gate using NOR gate

Truth Table & Observation:

Inputs		Output(T)	Output(P)
A	B	Y =A.B	Y=A.B
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Table 18: Truth Table for AND Gate

Result and Conclusion:

. . . .

. . . .

. . . .

Submission Date:

Professor In-charge

Experiment - 3

Exp. Date:

Aim: Study the Design of Half & Full Adder.

Requirement: Digital IC Trainer Kit, Power Supply and Connecting Wires.

Procedure/ Design Steps:

1. Implement the truth table for half and full adder
2. Simply the output using K-map
3. Implement the circuit using trainer kit
4. Verify the truth table of all the outputs (sum and carry)

Theory:

An adder is a digital logic circuit in electronics that implements addition of numbers. In many computers and other types of processors, adders are used to calculate addresses, similar operations and table indices in the ALU and also in other parts of the processors. These can be built for many numerical representations like excess-3 or binary coded decimal. Adders are classified into two types: half adder and full adder. The half adder circuit has two inputs: A and B, which add two input digits and generate a carry and sum. The full adder circuit has three inputs: A and C, which add the three input numbers and generate a carry and sum. This article gives brief information about half adder and full adder in tabular forms and circuit diagrams.

Half Adder and Full Adder Circuit

An adder is a digital circuit that performs addition of numbers. The half adder adds two binary digits called as augend and addend and produces two outputs as sum and carry; XOR is applied to both inputs to produce sum and AND gate is applied to both inputs to produce carry. The full adder adds 3 one bit numbers, where two can be referred to as operands and one can be referred to as bit carried in. And produces 2-bit output, and these can be referred to as output

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 21: Truth Table for Half Adder

Inputs			Outputs	
A	B	CIN	COUT	S
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

Table 22: Truth Table for Full Adder

carry and sum.
Truth table for half and full adder is shown in Table 21 & 22.
The circuit implemntation for half adder and full adder is shown in Figures 19 and 20.

Result:

....

....

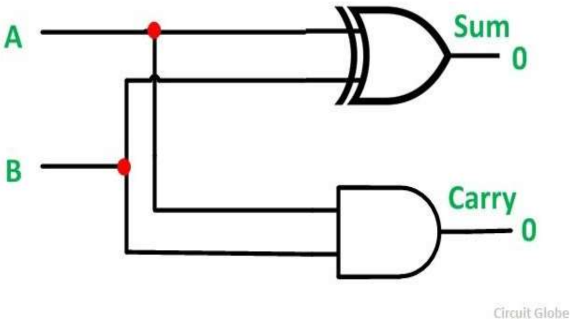


Figure 19: Half adder

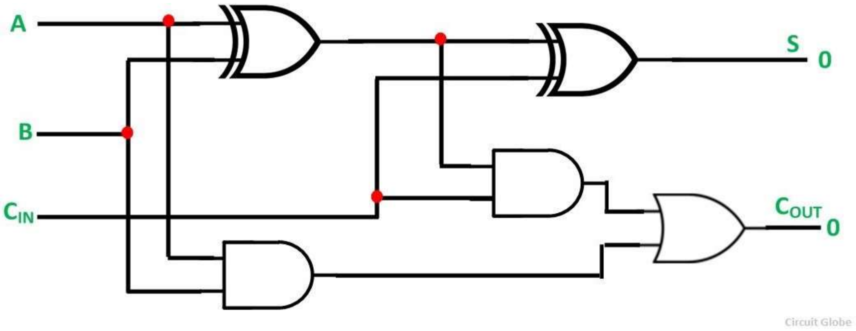


Figure 20: Full adder

. . . .

Submission Date:

Professor In-charge

Experiment - 4

Exp. Date:

Aim: Study the Design of Half & Full Subtractor.

Requirement: Digital IC Trainer Kit, Power Supply and Connecting wires.

Procedure/ Design Steps:

1. Implement the truth table for half and full subtractor
2. Simply the output using K-map
3. Implement the circuit using trainer kit
4. Verify the truth table of all the outputs (difference and borrow)

Theory:
In digital electronics, half subtractor and full subtractor are one of the most important combinational circuit used. Half subtractor and full subtractor are basically electronic devices or we can say logical circuits which performs subtraction of two binary digits. In this article, we are going to discuss half subtractor and full subtractor theory and also discuss the terms like half subtractor and full subtractor boolean expression, half subtractor and full subtractor circuit diagram etc.

Half subtractor:
As like addition operation of 2 binary digits, which produces SUM and CARRY, the subtraction of 2 binary digits also produces two outputs which are termed as difference and borrow. The simplest possible subtraction of 2-bit binary dig- its consists of four possible operations, they are 0-0, 0-1, 1-0 and 1-1. The operations 0-0, 1-0 and 1-1 produces a subtraction of 1-bit output whereas, the remaining operation 0-1 produces a 2-bit output. They are referred as differ- ence and borrow bit respectively. This borrow bit is used for subtraction of the next higher pair bit.
So, we can define half subtractor as a combinational circuit which is capable

of performing subtraction of 2-bit binary digits is known as a half subtractor. Here, the binary digit from which the other digit is subtracted is called minuend and the binary digit which is to be subtracted is known as the subtrahend.

Half subtractor truth table and circuit diagram: Half subtractor truth table:

Inputs		Outputs	
A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Table 23: Truth Table for Half subtractor

In the above truth table of half subtractor, the two input variables A and B represents minuend and subtrahend respectively. The two output functions difference and borrow are termed as D and B respectively. Using the truth table of half subtractor, we can design the half subtractor circuit diagram as below.

Half subtractor circuit diagram:

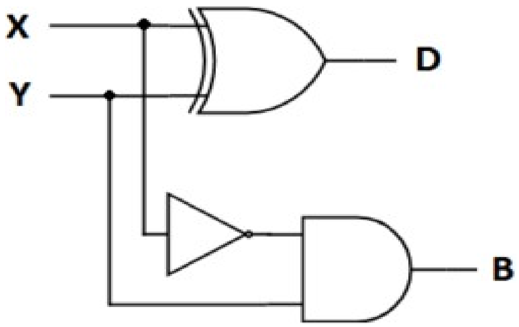


Figure 21: Half subtractor

Full Subtractor:

When there is a situation where the minuend and subtrahend number contains more significant bit, then the borrow bit which is obtained from the subtraction of 2-bit binary digits is subtracted from the next higher order pair of bits. In such situation, the subtraction involves the operation of 3 bits. Such situation of subtraction can't handle by a simple half subtractor. So, combining two half subtractor we can form another combinational circuit which can perform this type of operation. This circuit is known as the full subtractor.

So we can define full subtractor as a combinational circuit which takes three inputs and produces two outputs difference and borrow. Below is the truth table of the full subtractor, we have used three input variables X, Y and Z which refers to the term minuend, subtrahend and borrow bit respectively. The two outputs difference and borrow are named as D and B respectively.

Full subtractor truth table:

Inputs			Outputs	
A	B	CIN	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Table 24: Truth Table for Full Subtractor

Full subtractor circuit diagram:

Result:

....

....

....

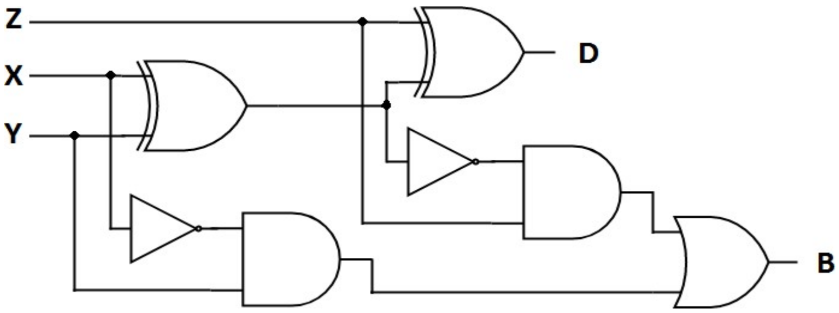


Figure 22: Full subtractor

Submission Date:

Professor In-charge

Experiment – 5

Exp. Date:

AIM: To design and implement 4-bit BCD adder using IC 7483.

COMPONENTS AND EQUIPMENTS REQUIRED:

Requirement: Digital Trainer kit, IC 7483, EX-OR GATE IC 7486

THEORY:

4 BIT BINARY ADDER:

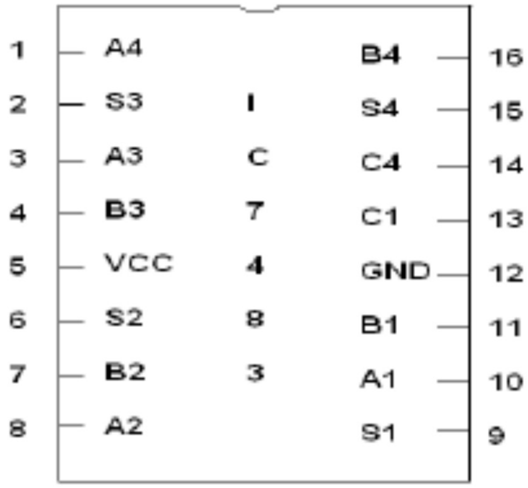
A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of the next full adder in the chain. The augend bits of 'A' and the addend bits of 'B' are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bits. The carry's are connected in chains through the full adder. The input carry to the adder is C0 and it ripples through the full adder to the output carry C4.

BIT BCD ADDER:

Consider the arithmetic addition of two decimal digits in BCD, together with an input carry from a previous stage. Since each input digit does not exceed 9, the output sum cannot be greater than 19, the 1 in the sum being an input carry. The output of two decimal digits must be represented in BCD and should appear in the form listed in the columns.

ABCD adder that adds 2 BCD digits and produces a sum digit in BCD. The 2 decimal digits, together with the input carry, are first added in the top 4-bit adder to produce the binary sum.

PIN DIAGRAM FOR IC 7483:



PROCEDURE

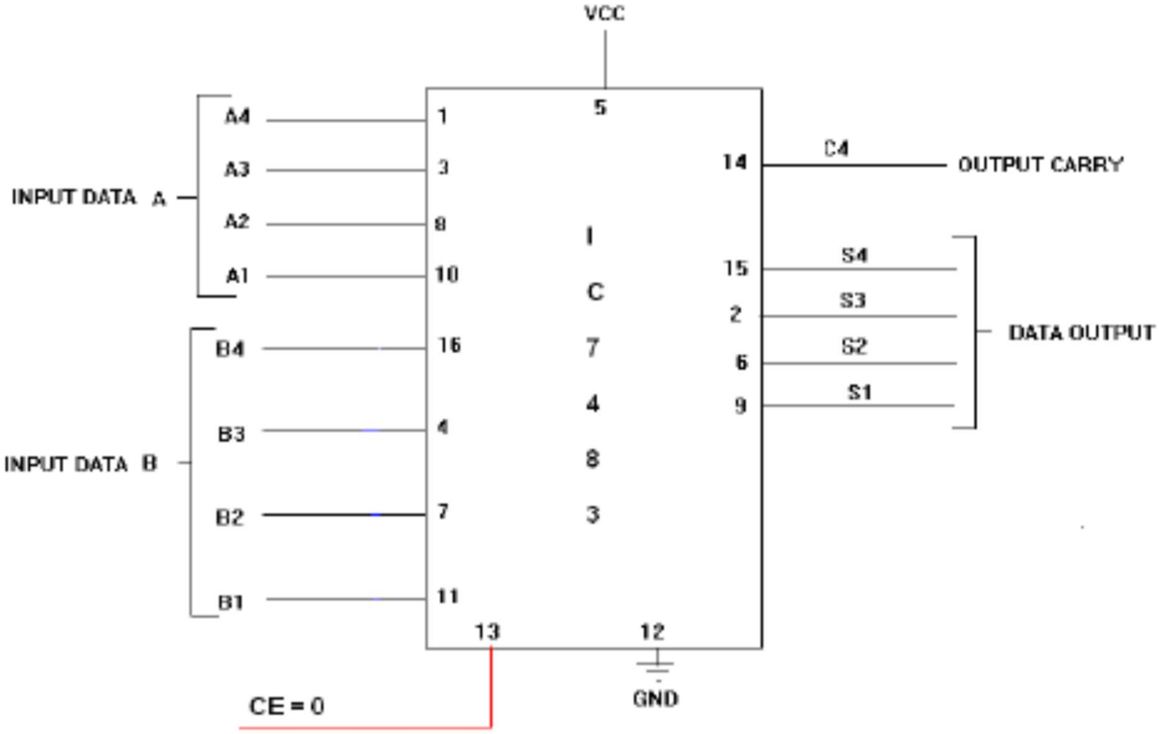
1. Realize adder using suitable gates and IC 7483.

2. Realize the BCD adder using IC7483,and K map for converting binary to BCD.

3. Verify the output using different input combinations.

4. Set up the circuits and observe the outputs.

LOGIC DIAGRAM:
4-BIT BINARY ADDER



Truth Table for 4 bit adder

Sr. No	Input data A				Input Data B				Addition				
	A3	A2	A1	A0	B3	B2	B1	B0	C	S3	S2	S1	S0
1	1	0	0	0	0	0	1	0	0	1	0	1	0
2	1	0	0	0	1	0	0	0	1	0	0	0	0
3	0	0	1	0	1	0	0	0	0	1	0	1	0
4	0	0	0	1	0	1	1	1	0	0	1	0	0
5	1	0	1	0	1	0	1	1	1	0	0	1	0

RESULT: Realized the circuit for 4-bit BCD adder using IC 7483 and observed the outputs for different inputs.

Experiment - 6

Exp. Date:

Aim: Design and Implement 3 - bit binary to gray code converter.

Requirement: Digital IC trainer Kit, Power Supply and Connecting Wires

Procedure/ Design Steps:

1. Implement the truth table for 3 input binary to gray code conveter
2. Simply the output using K-map
3. Implement the circuit using trainer kit
4. Verify the truth table of all the outputs (gray code).

Theory:
In computers, we need to convert binary to gray and gray to binary. The conversion of this can be done by using two rules namely binary to gray conversion and gray to binary conversion. In the first conversion, the MSB of the gray code is constantly equivalent to the MSB of the binary code. Additional bits of the gray code's output can get using EX-OR logic gate concept to the binary codes at that present index as well as the earlier index. Here MSB is nothing but the most significant bit. In the first conversion, the MSB of the binary code is constantly equivalent to the MSB of the particular binary code. Additional bits of the binary code's output can get using EX-OR logic gate concept by verifying gray codes at that present index. If the present gray code bit is zero then after that copy earlier binary code, as well copy reverse of earlier binary code bit. This article discusses an overview of code converters which includes binary to gray code converter as well as gray to binary code converter.

Binary to Gray Code Converter
Binary code is a very simple representation of data using two values such as 0's and 1's, and it is mainly used in the world of the computer. The binary code could be a high (1) or low (0) value or else even a modify in value. Gray code

Binary Inputs			Gray outputs		
B2	B1	B0	G2	G1	G0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Table 19: Binary to Gray code converter

or reflected binary code estimates the binary code nature that is arranged with on & off indicators, usually denoted with ones & zeros. These codes are used to look at clarity as well as error modification in binary communications.

The conversion of binary to gray code can be done by using a logic circuit. The gray code is a non-weighted code because there is no particular weight is assigned for the position of the bit. A n-bit code can be attained by reproducing a n-1 bit code on an axis subsequent to the rows of $2^n - 1$, as well as placing the most significant bit of 0 over the axis with the most significant bit of 1 beneath the axis. The step by step gray code generation is shown below.

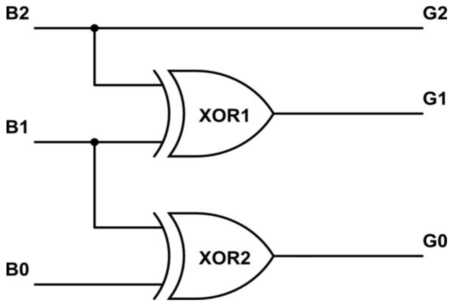


Figure 17: 3 -bit Binary to Gray code converter

Truth table for 3-input binary to gray code converter is shown below

Result and Conclusion:

....

....

....

....

....

Submission Date:

Professor In-charge

Experiment - 7

Exp. Date:

Aim: Design and Implement_Magnitude Comparator.

Requirement: Digital IC trainer Kit, Power Supply and Connecting Wires

Procedure/ Design Steps:

1) Switch ON the power supply button to supply 5V to the circuit.

2) Press Switch 1 for input A and Switch 2 for input B.

The switch in ON state is 🟩 and the switch in OFF state is 🟥

S3) i)When the input A is greater than the input B,LED 1 lits up.

ii)When the input A is lesser than the input B,LED 2 lits up.

iii)When the input A is equal to the input B,LED 3 lits up.

The LED in OFF state is ○ and the LED in ON state is 🔴 .

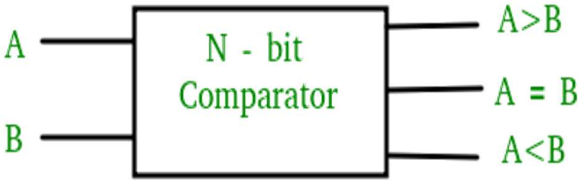
4) Click on "Add" Button to add data to the Truth Table.

Step-5) Repeat Steps 2 to 4 for another set of data.

6) Click "Print" to get the print out of the Truth Table.

Theory:

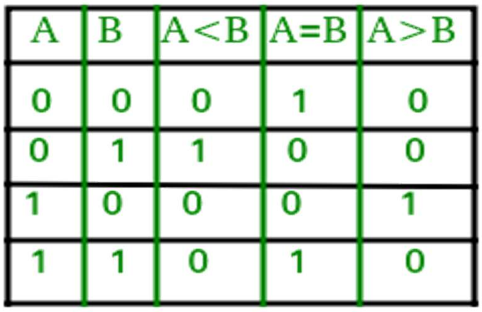
A magnitude digital comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for A > B condition, one for A = B condition and one for A < B condition.



Block Diagram of Comparator

1-Bit Magnitude Comparator:

A comparator used to compare two bits is called a single bit comparator. It consists of two inputs each for two single bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers. The truth table for a 1-bit comparator is given below :



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

Figure-2: Truth Table of 1-Bit Comparator

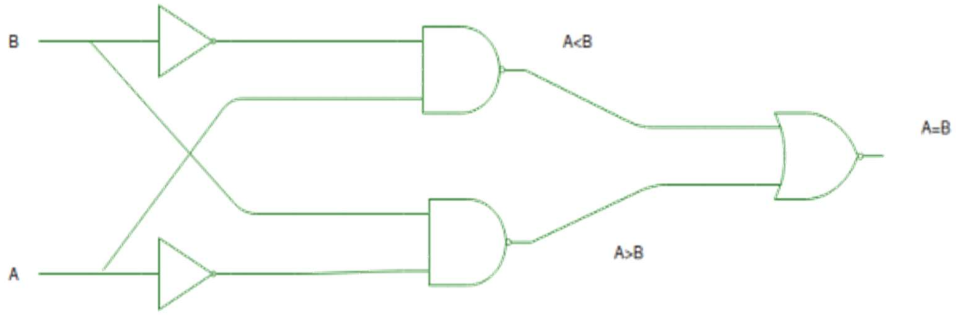
From the above truth table logical expressions for each output can be expressed as follows:

A > B : AB'

A < B : A'B

A = B : A'B' + AB

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below :



Logic Circuit of 1-Bit Comparator

Result and Conclusion:

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

Submission Date:

Professor In-charge

Experiment - 8

Exp. Date:

Aim: Design and Implement 4 - bit gray to binary code converter.

Requirement: Digital IC Trainer Kit, Power supply and connecting wires.

Procedure/ Design Steps:

1. Implement the truth table for 4 input gray to binary code converter
2. Simply the output using K-map
3. Implement the circuit using trainer kit
4. Verify the truth table of all the outputs (binary code).

Theory

This conversion method also follows the EX-OR gate operation between grey & binary bits. The below steps may useful to know how to perform gray code to binary conversion.

1. To convert grey code to binary, bring down the most siginificant digit of the given grey code number, because, the first digit or the most siginificant digit of the grey code number is same as the binary number.
2. To obtain the successive second binary bit, perform the EX-OR operation between the first bit or most siginificant digit of binary to the second bit of the given grey code.
3. To obtain the successive third binary bit, perform the EX-OR operation between the second bit or most siginificant digit of binary to the third MSD (most significant digit) of grey code and so on for the next successive binary bits conversion to find the equivalent

The truth table for the 4-input gray to binary code converter is shown below

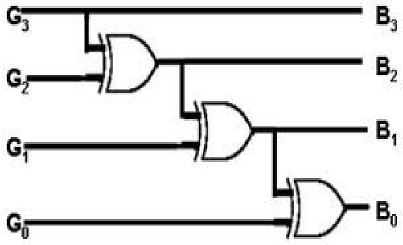


Figure 18: 4 -bit Gray to Binary code converter

Decimal No.	Gray Inputs				Binary outputs			
Decimal No.	G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	0	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1
10	1	1	1	1	1	0	1	0
11	1	1	1	0	1	0	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

Table 20: Gray to Binary code converter

Result:

....
....
....

Submission Date:

Professor In-charge

Experiment - 9

Exp. Date:

Aim: Design 4x1 multiplexer using digital logic gates

Requirement: Digital IC Trainer Kit, Power Supply and Connecting Wires.

Procedure/ Design Steps:

1. Implement the truth table for 4x1 multiplexer with two selection lines
2. Write Boolean expression for the output
3. Implement the circuit using trainer kit
4. Verify the truth table of all the outputs

Theory:

Multiplexer is a combinational circuit that has maximum of 2^n data inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called as Mux.

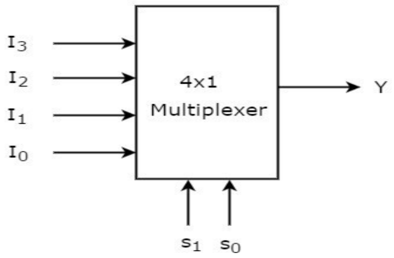
4x1 Multiplexer

4x1 Multiplexer has four data inputs I3, I2, I1 & I0, two selection lines s1 & s0 and one output Y. The block diagram of 4x1 Multiplexer is shown in figure 29.

One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. Truth table of 4x1 Multiplexer is shown in Table 28.

From Truth table, we can directly write the Boolean function for output, Y as

$$Y = S1'S0'I0 + S1'S0I1 + S1S0'I2 + S1S0I3$$

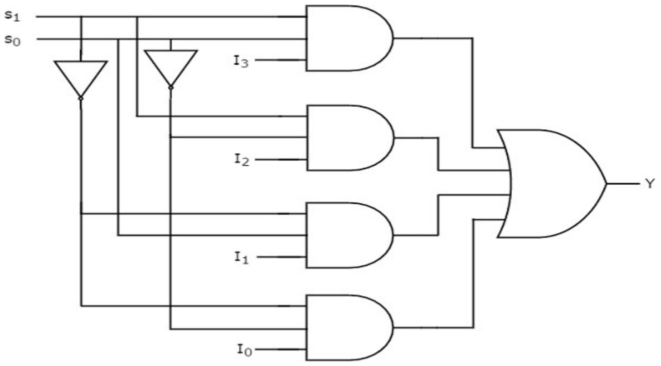


4x1 Multiplexer

Selection Lines		Output
S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Truth table for 4x1 multiplexer

We can implement this Boolean function using Inverters, AND gates & OR gate. The circuit diagram of 4x1 multiplexer is shown in figure 30.



Circuit diagram for 4x1 multiplexer

Result and Conclusion:

. . . .

. . . .

. . . .

Submission Date:

Professor In-charge