MAHAKAL INSTITUTE OF TECHNOLOGY, UJJAIN

Approved By: All India Council of Technical Education (New Delhi)

DEPARTMENT OF
ELECTRONICS & COMMUNICATION ENGINEERING

LAB MANUAL

Name of Student : ……………………..

Name of Lab : Digital Systems

Subject Code : CS-304

Branch : Computer Science & Engineering

Year/Semester : II / III

Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (MP)

INDEX

S. No.	Name of Experiment	Date	Sign	Remark
1.	To study and verify the Truth Tables of AND, OR, NOT, NAND, NOR and EX-OR logic gates.			
2.	Implementation of AND, NOT and OR functions with NAND and NOR Gate.			
3.	To verify the operation of Half Adder & Full Adder circuits.			
4.	To verify the operation of Half Subtractor & Full Subtractor circuits.			
5.	To study and verify the Binary to Gray and Gray to Binary code conversion circuits.			

List of Experiments

Digital Systems (CS-304)

1) To study and verify the Truth Tables of AND, OR, NOT, NAND, NOR and EX-OR logic gates.
2) Implementation of AND, NOT and OR functions with NAND and NOR Gate.
3) To verify the operation of Half Adder & Full Adder circuits.
4) To verify the operation of Half Subtractor & Full Subtractor circuits.
5) To study and verify the Binary to Gray and Gray to Binary code conversion circuits.

Lab Manual

Digital Systems (CS-304)

Total number of Experiments: 05
Total number of Turns: 05

S. No.	Name of Experiment	Turns needed to complete
1.	To study and verify the Truth Tables of AND, OR, NOT, NAND, NOR and EX-OR logic gates.	1
2.	Implementation of AND, NOT and OR functions with NAND and NOR Gate.	1
3.	To verify the operation of Half Adder & Full Adder circuits.	1
4.	To verify the operation of Half Subtractor & Full Subtractor circuits.	1
5.	To study and verify the Binary to Gray and Gray to Binary code conversion circuits.	1

Distribution of Lab Hours:	**1 Hour – 40 Minutes**
Explanation of Experiment:	**20 Min.**
Performance of Experiment:	**50 Min.**
File Checking:	**10 Min.**
Attendance:	**05 Min.**
Viva/Quiz:	**05 Min.**
Solving of Queries:	**10 Min.**

DEPLOYMENT OF THE EXPERIMENT:

Turn	Deployment
1	Introduction of Lab & Experiment No.1
2	Experiment No.2
3	Experiment No.3
4	Experiment No.4
5	Experiment No.5

EQUIPMENT REQUIREMENT:

1. Particular KIT
2. Patch cords
3. Power cable

BATCH DISTRIBUTION: Per Batch of 30 Students

REFERENCE BOOKS:

S. No.	Title of the Book	Authors	Publication
1.	Fundamentals of Digital Circuits	A. Anand Kumar	PHI
2.	Digital Logic & Computer Design	Moris Mano	PHI
3.	Digital Principles and applications	Malvino & Leech	TMH

COURSE OUTCOMES:

After the completion of the course student will be able to:

1) Demonstrate the truth table of various expressions and combinational circuits using logic gates.
2) Design, test and evaluate various combinational circuits such as adders, subtractors, comparators, multiplexers and demultiplexers.
3) Construct flips-flops, counters and shift registers.

EXPERIMENT NO. 1

Unit/Topic: 1/Logic Gates

OBJECTIVE:

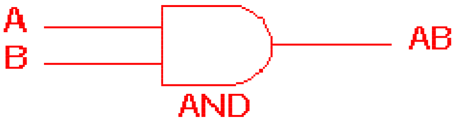
To study and verify the Truth Tables of AND, OR, NOT, NAND, NOR and EX-OR Logic Gates for positive logic.

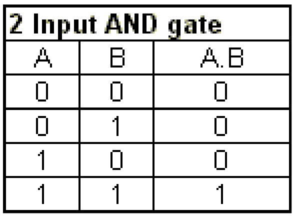
APPARATUS REQUIRED:

Digital logic trainer and Patch cords

THEORY:

AND Gate: A multi-input circuit in which the output is 1 only if all inputs are 1.The symbolic representation of the AND gate is:

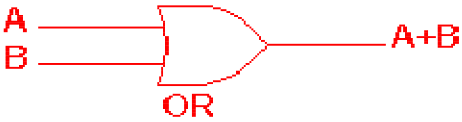


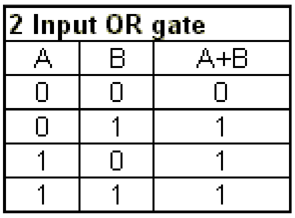


2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot (.) is used to show the AND operation i.e. A.B .

OR Gate: A multi-input circuit in which the output is 1 when any input is 1. The symbolic representation of the OR gate is shown:

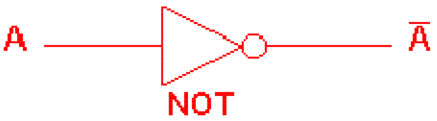


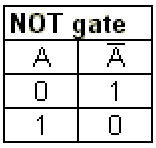


2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.

NOT Gate: The output is 0 when the input is 1, and the output is 1 when the input is 0. The symbolic representation of an inverter is:

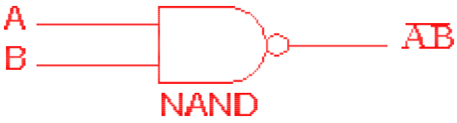


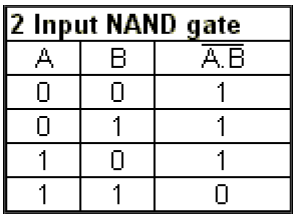


NOT gate	
A	$\overline{A}$
0	1
1	0

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, as shown at the outputs.

NAND Gate: AND followed by INVERT. It is also known as universal gate. The symbolic representation of the NAND gate is:

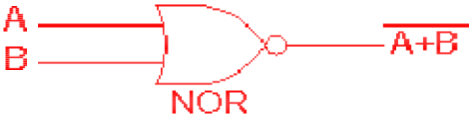


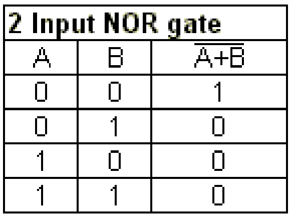


2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

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.

NOR Gate: OR followed by inverter. It is also known as universal gate. The symbolic representation is:

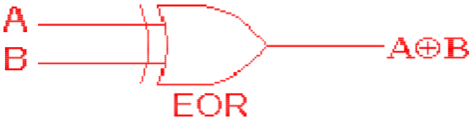


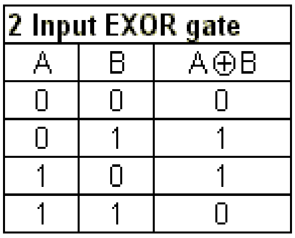


2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

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.

EX-OR Gate: The output of the Exclusive OR gate, is 0 when it’s two inputs are the same and it’s output is 1 when its two inputs are different. It is also known as Anti-coincidence gate.





2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

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 (⊕) is used to show the EX-OR operation.

Observation Table: LED ON (RED light): Logic 1
LED OFF (Green Light): Logic 0
Input variables: A, B
Output variable: Y

S. No.	Input(A)	Input(B)	Output (NOT) ___ Y = A’	Output (AND) Y=AB	Output (OR) Y=A+B	Output (NAND) ____ Y = (AB)’	Output (NOR) ___ _ Y= (A+B)’	Output (EX-OR) ⊕ Y= A⊞ B
1								
2								
3								
4								

Results and Analysis:

NOT Gate: When logic 1 is applied to one of NOT gate of 7404 IC, then output becomes zero. When input LED is ON (RED), the output LED become OFF (Green) vice versa.

OR Gate: The output of an OR gate is a 1 if one or the other or both of the inputs are 1, but a 0 if both inputs are 0. When One or the other or Both of the input LEDS are ON (RED Light), then output LED is ON (RED) otherwise Output LED is OFF(Green Light)

AND Gate: The output of an AND gate is only 1 if both its inputs are 1. For all other possible inputs the output is 0.When both the LEDS are On, then output LED is ON (RED Light) otherwise Output LED is OFF.

NOR Gate: The output of the NOR gate is a 1 if both inputs are 0 but a 0 if one or the other and both the inputs are 1.

NAND Gate: The output of the NAND gate is a 0 if both inputs are 1 but a 1 if one or the other or both the inputs are 0.

EX-OR gate: The output of the XOR gate is a 1 if either but not both inputs are 1 and a 0 if the inputs are both 0 or both 1.

RESULT:

EXPECTED VIVA QUESTIONS:

1. List out the Basic Gates.
2. What are the applications of Gates?
3. Write the truth table of EX- OR Gate.
4. Write the truth table of NOR Gate.
5. What are Universal Gates?

NAME OF FACULTY: PROF. MANISH BARVE

SIGNATURE:

DATE:

EXPERIMENT NO. 2

Unit/Topic: 2 /NAND-NOR Implementation

OBJECTIVE:

Implementation of AND, NOT and OR functions with NAND and NOR Gate.

EQUIPMENTS NEEDED:

1. Digital board **DB02**.
2. DC Power Supply +5V from external source or **Digital Lab ST2611**.
3. Digital multi-meter or **Digital Lab ST2611**.

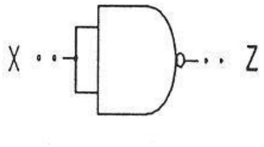
THEORY:

NAND & NOR Gates are said to be Universal Gate because any digital system can be implemented with them. To show that any Boolean function can be implemented with NAND Gates. It is only need to show that logic operations AND, OR, NOT can be implemented with NAND & NOR Gates. The implementation of AND, OR, NOT operation with NAND Gate is shown in Logic diagram in Experiment section. The NOT operation is obtained from a two input NAND Gate. The inputs of NAND Gate are shorted to get NOT operation. The AND operation requires two NAND Gates. The first produces the inverted AND and second acts as an inverter to produce normal output. The OR operation is achieved through a NAND Gate with additional inverters in each input.

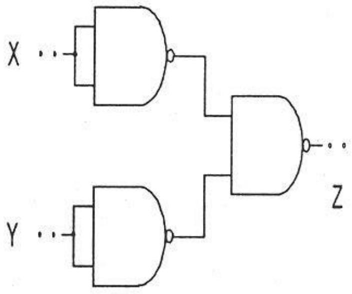
The NOR function is dual of the NAND function. All procedures and rules for NOR logic form a dual of the corresponding procedures and rules developed for NAND logic. The conversion of NOR to AND, OR, NOT is shown Logic diagram in Experiment section. NOT operation is obtained from a 2 input NOR Gate with both the inputs shorted. The OR operation requires two NOR Gates. The first produces the inverted OR and the second acts as an inverter to obtain normal output. The AND operation is achieved through a NOR Gate with additional inverters at each input. The logic diagram and truth table of Logic Gates is shown in experiment section.

Logic Diagram & Truth Table:

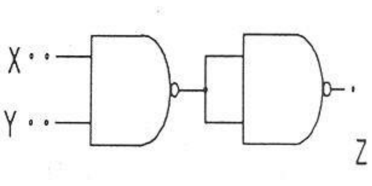
(Logic 1 = +5 V & logic 0 = Gnd)



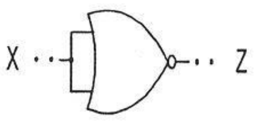
NOT Gate using NAND Gate



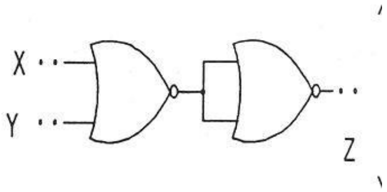
OR Gate using NAND Gate



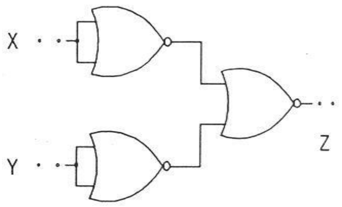
AND Gate using NAND Gate



NOT Gate using NOR Gate



OR Gate using NOR Gate



AND Gate using NOR Gate

Truth Table for NOT gate:-

X	Y
0	1
1	0

Truth Table for OR gate:-

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table for AND gate:-

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

PROCEDURE:

Connect +5 V and ground to their indicated position on **DB02** from external DC Power Supply or from DC power block of **Digital Lab ST2611.**

1. Connect inputs 00, 01, 10, 11 as per truth table to pins X and Y of OR Gate.
2. Switch on the Power Supply.
3. Observe output Z of Gate on multi meter or on logic probe or on LED display of **Digital Lab ST2611** and prove truth table.
4. Repeat above steps for remaining Logic Gates.

RESULT:

EXPECTED VIVA QUESTIONS:

1. What are Universal Gates?
2. What are the applications of Universal Gates?
3. Design XOR gate using NAND gates only?
4. Design XOR gate using NOR gates only?

NAME OF FACULTY: PROF. MANISH BARVE

SIGNATURE:

DATE:

EXPERIMENT NO. 3

Unit/Topic: 3/Combinational circuit (Adder)

OBJECTIVE:

To verify the operation of Half Adder & Full Adder circuits.

APPARATUS REQUIRED:

Digital logic trainer
Patch cords.

THEORY:

Half Adder:

With the help of half adder, we can design circuits that are capable of performing simple addition with the help of logic gates.

Let us first take a look at the addition of single bits.

0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 10

These are the least possible single-bit combinations. But the result for 1+1 is 10. Though this problem can be solved with the help of an EXOR Gate, if you do care about the output, the sum result must be re-written as a 2-bit output.

Thus the above equations can be written as

0+0 = 00
0+1 = 01
1+0 = 01
1+1 = 10

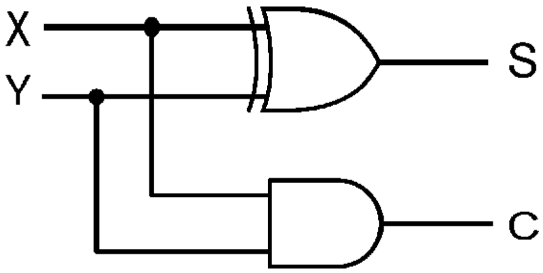
Here the output ‘1’of ‘10’ becomes the carry-out. The result is shown in a truth-table below. ‘SUM’ is the normal output and ‘CARRY’ is the carry-out.

INPUTS		**OUTPUTS**	
X	**Y**	**SUM**	**CARRY**
0	**0**	**0**	**0**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**1**

S = X’.Y +X.Y’

C = X.Y

From the equation it is clear that this 1-bit adder can be easily implemented with the help of EXOR Gate for the output 'SUM' and an AND Gate for the carry. Take a look at the implementation below.



Half Adder

For complex addition, there may be cases when you have to add two 8-bit bytes together. This can be done only with the help of full-adder logic.

PROCEDURE:

1. Make connections as shown in the figure.
2. Connect +5V to pin no. 14 and ground to pin no.7 of IC7408 and IC74136. (See IC Pin diagram as shown in the figure 87)
3. Connect the input terminal or input pin of IC7408 (pin no. 1 & 2) with input pin of IC74136 (pin no 1 & 2) and now connect them with any two of the 8 bit data switches
4. Connect output of EX-OR Gate i.e. pin no. 3 to input of logic probe or 8 bit LED display.
5. This will give the output for Summation.
6. Connect output of AND Gate i.e. pin no. 3 to input of logic probe or 8 bit LED display.
7. This will give the output for carry
8. Apply 0 (0V) to pin no. 1 and 2 of IC74136 (same will apply at the IC7408) shown in figure as per Truth Table.
9. Observe the output coming at 8 bit LED display or Logicprobe.
10. If the output is 0 (0V) the LED will not glow or if you have connected the Logic probe it will display "L" for low output
11. Apply the 01, 10, 11 combination to pin no. 1 and pin no. 2 respectively and observe the output.
12. LED should glow or Logic probe will show "H" if the output is 1 or (5V high output)
13. Note all the output coming for these combinations of input and verify it by the truth table.

OBSERVATION TABLE:

Input 1	Input 2	Carry	Sum
X	Y	C	S

Full Adder:

This type of adder is a little more difficult to implement than a half-adder. The main difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. The first two inputs are X and Y and the third input is an input carry designated as C_{IN}. When a full adder logic is designed we will be able to string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

The output carry is designated as C_{OUT} and the normal output is designated as S. Take a look at the truth-table.

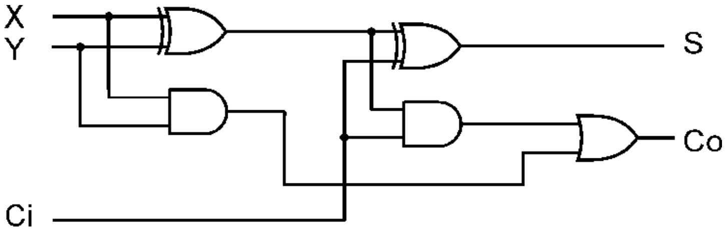
INPUTS			OUTPUTS	
X	**Y**	**C_{in}**	**C_{out}**	**SUM (S)**
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**0**	**1**
0	**1**	**0**	**0**	**1**
0	**1**	**1**	**1**	**0**
1	**0**	**0**	**0**	**1**
1	**0**	**1**	**1**	**0**
1	**1**	**0**	**1**	**0**
1	**1**	**1**	**1**	**1**

$$S = X'Y'C_{in} + X'Y\,C_{in}' + XY'\,C_{in}' + XY\,C_{in}$$

$$C_{OUT} = X.Y + X.\,C_{in} + Y.\,C_{in}$$

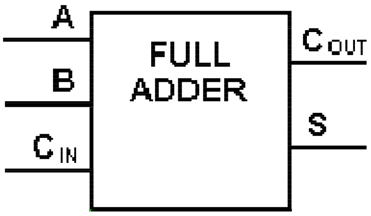
From the above truth-table, the full adder logic can be implemented. We can see that the output S is an EXOR between the input A and the half-adder SUM output with B and CIN inputs. We must also note that the COUT will only be true if any of the two inputs out of the three are HIGH.

Thus, we can implement a full adder circuit with the help of two half adder circuits. The first will half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add CIN to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus, COUT will be an OR function of the half-adder Carry outputs.



.

FULL ADDER



Single Bit full Adder

With this type of symbol, we can add two bits together taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude. In a computer, for a multi-bit operation, each bit must be represented by a full adder and must be added simultaneously. Thus, to add two 8-bit numbers, you will need 8 full adders which can be formed by cascading two of the 4-bit blocks.

PROCEDURE:

1. Make connections as shown in the figure 89.
2. Connect +5V to pin no. 14 and ground to pin no.7 of IC7408, IC7432 and IC 74136. (See IC Pin diagram as shown in the figure 89)
3. For Input1 and Input2 i.e. X&Y; connect the input terminal or input pin of IC 74136 (pin no. 1 & 2) with any two of the 8 bit data switches.
4. Also connect the same pins with input pin of IC7408 (pin no 1 & 2) for first carry.
5. The output of first sum will come at IC74136 pin no.3 feed this output back to IC74136 input (pin no. 4) for second summation.
6. Also connect it with AND Gate’s pin no.4 for carry input of second carry generator
7. For Input 3 i.e. Ci; connect an 8 bit data switch with IC74136 input (pin no. 5).
8. Also connect it with AND gates pin no. 5 for carry input of second carry generator
9. Connect output of EX-OR Gate (IC74136) i.e. pin no. 6 to input of logic probe or 8 bit LED display.

10. This will give the output for Full adder.
11. Second carry will appear at out put (pin no. 6) of AND gate
12. To Get a full carry connect the IC7408 pin no. 3 & 6 with the input terminal of IC7432 (pin no. 1 & pin no.2 respectively)
13. Connect the output terminal of IC7432 (pin no. 3) at input of logic probe or 8 bit LED display.
14. This will give the output for Full carry
15. Apply 0 (0V) to pin no. 1, 2 and 5 of IC74136 shown in the figure 89 as per TruthTable.
16. Observe the output coming at 8 bit LED display or Logic probe.
17. If the output is 0 (0V) the LED will not glow or if you have connected the Logic probe it will display "L" for low output
18. LED should glow or Logic probe will show "H" if the output is 1 or (5V high output)
19. Apply the all other combination given in truth to pin no.1, pin no.2 and pin no.5 respectively and observe the output. Note all the output coming for these combinations of input and verify it by the truth table

OBSERVATRION TABLE:-

Input 1	**Input 2**	**Input 3**	**Output Carry**	**Sum**
X	**Y**	**C_{in}**	**Co**	**S**
0	**0**	**0**		
0	**0**	**1**		
0	**1**	**0**		
0	**1**	**1**		
1	**0**	**0**		
1	**0**	**1**		
1	**1**	**0**		
1	**1**	**1**		

RESULT:

EXPECTED VIVA QUESTIONS:

1. What is meant by combinational circuit?
2. What is a Half adder?
3. What is a Full adder?
4. What is a Parallel adder?
5. State the limitations of Karnaugh Map.

NAME OF FACULTY: PROF. MANISH BARVE

SIGNATURE:

DATE:

EXPERIMENT NO. 4

Unit/Topic: 4/Combinational circuit (Subtractor)

OBJECTIVE:

To verify the operation of Half Subtractor & Full Subtractor circuits.

APPARATUS REQUIRED:

Digital logic trainer
Patch cords.

THEORY:

The arithmetic operation, subtraction of two binary digits has four possible elementary operations, namely,

0 - 0 = 0
0 - 1 = 1 with 1 borrow
1 - 0 = 1
1 - 1 = 0

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.

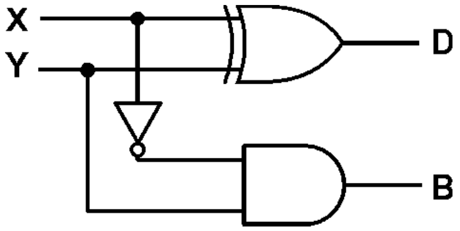
Half Subtractor:

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. **Half Subtractor** The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow).

INPUTS		**OUTPUTS**	
X	**Y**	**D (Difference)**	**B (Borrow)**
0	**0**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**1**	**0**
1	**1**	**0**	**0**

The Logic equation is:

D = X’Y+XY’,
B = X’.Y



Half Subtractor

PROCEDURE:

1. Connect +5V to pin no. 14 and ground to pin no.7 of IC7408, IC7404 and IC 74136.
2. Connect the input terminal or input pin of IC 74136 (pin no. 1 & 2) with any two of the 8 bit data switches for Input 1 and Input 2 respectively
3. Connect output of EX-OR Gate i.e. pin no. 3 of IC 74136 to input of logic probe or 8 bit LED display.
4. This will give the output for Difference.
5. Also connect the Input1 at the input pin no.1 of AND Gate IC7408 for 'Borrow' input1.
6. Connect the Input2 with NOT Gate IC7404 pin no.1. Connect the output of inverted signal i.e. IC7404's pin no. 2 with AND Gate pin no. 2.
7. Connect output of AND Gate i.e. pin no. 3 to input of logic probe or 8 bit LED display.
8. This will give the output for 'Borrow.'
9. Apply 0 (0V) to pin no. 1 and 2 of IC74136 (same will apply at the IC7408 and IC7404 input pins respectively) shown in the figure 91 as per Truth Table.
10. Observe the output coming at 8 bit LED display or Logic probe.
11. If the output is 0 (0V) the LED will not glow or if you have connected the Logic probe it will display "L" for low output
12. LED should glow or Logic probe will show "H" if the output is 1 or (5V high output)
13. Apply the 01, 10, 11 combination to pin no. 1and pin no. 2 respectively and observe the output.

OBSERVATION TABLE:

INPUTS		**OUTPUTS**	
X	Y	**D (Difference)**	**B (Borrow)**
0	**0**		
0	**1**		
1	**0**		
1	**1**		

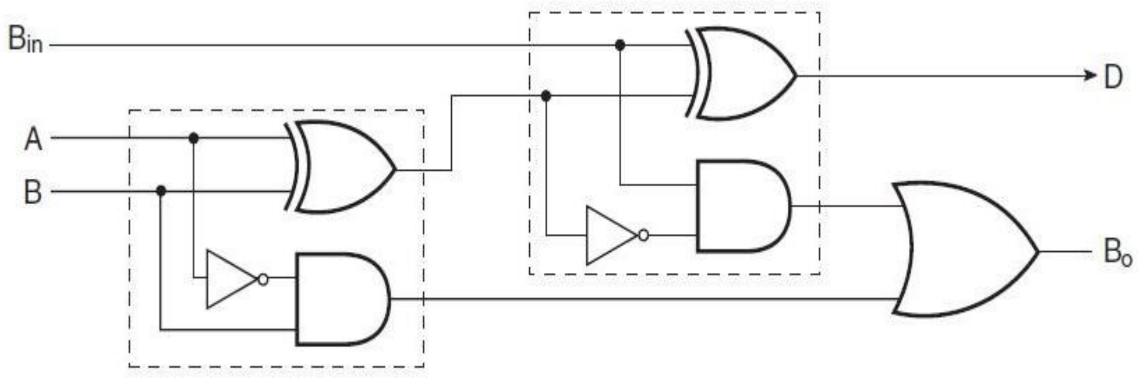
Full Subtractor:

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. As in the case of the addition using logic gates, a full subtractor is made by combining two half-subtractors and an additional OR-gate. A full subtractor has the borrow in capability (denoted as BOR_{IN} in the diagram below) and so allows cascading which results in the possibility of multi-bit subtraction. The circuit diagram for a full subtractor is given below.

INPUTS			**OUTPUTS**	
A	**B**	**B_{in}**	**D (DIFF.)**	**B_O (Borrow)**
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**

$D = A'B'B_{in} + A'BB_{in}' + AB'B_{in}' + ABB_{in} = A \oplus B \oplus B_{in}$

$B_O = A'B + A'B_{in} + BB_{in}$



PROCEDURE:

1. Collect the components necessary to accomplish this experiment.
2. Plug the IC chip into the breadboard.
3. Connect the supply voltage and ground lines to the chips. PIN7 = Ground an PIN14 = +5V.
4. According to the pin diagram of each IC mentioned above, make the connection according to circuit diagram.
5. Connect the inputs of the gate to the input switches of the LED.
6. Connect the output of the gate to the output LEDs.
7. Once all connections have been done, turn on the power switch of the breadboard
8. Operate the switches and fill in the truth table (Write "1" if LED is ON and "0" if LED is OFF
9. Apply the various combination of inputs according to the truth table and observe the condition of Output LEDs.

OBSERVATION TABLE:

Input Variable: A, B, B_{in}
Output Variable: D, B_O
RED Light LED: Logic 0
Green Light LED: Logic 1

INPUTS			**OUTPUTS**	
A	**B**	**B_{in}**	**D (DIFF.)**	**B_O (Borrow)**
0	**0**	**0**		
0	**0**	**1**		
0	**1**	**0**		
0	**1**	**1**		
1	**0**	**0**		
1	**0**	**1**		
1	**1**	**0**		
1	**1**	**1**		

RESULT:

EXPECTED VIVA QUESTIONS:

1. Draw the logic diagram of Full Subtrator.
2. What is different between Combinational and Sequential Circuit?
3. What is expression for difference and borrow for Full Subtractor.
4. Draw the logic diagram of Half Subtrator.

NAME OF FACULTY: PROF. MANISH BARVE

SIGNATURE:

DATE:

EXPERIMENT NO. 5

Unit/Topic: 5/Code Converters

OBJECTIVE:

To study and verify the Binary to Gray and Gray to Binary code conversion circuits.

1. Binary to Gray Code
2. Gray to Binary Code

APPARATUS REQUIRED:

1. Digital board DB06.
2. DC Power Supply +5 V from external source or ST2611 Digital lab.
3. Digital Multimeter or Digital Lab ST2611.

THEORY:

The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different digital system. It is sometimes necessary to use the output of one system as the input to another. A conversion circuit must be inserted between the two systems if each uses different codes for the same information. Thus, a code converter is a circuit that makes the two systems compatible even though each uses a different binary code.

The binary number system is a system that uses only the digits 0 & 1 as codes. To represent a group of 2n distinct element in a binary code requires a minimum of n bits. This is because it is possible to arrange n bits in 2n distinct ways. Although the minimum number of bits required to code 2n distinct quantities is n, there is no maximum number of bits that may be used for binary code. For example, a group of four distinct quantities can be represented by a two bit code, with each quantity assigned one of the following bit combinations: 00, 01, 10, and 11. A group of eight elements requires a three bit code, with each element assigned to one and only one of the following 000, 001, 010, 011, 100, 101, 110, and 111. (Refer table 1).

Gray code (reflected code) is shown in Table 1. Number in the gray code changes by only one bit as it proceeds from one number to the next. For example in going from decimal 7 to 8, the gray code number changes from 0100 to 1100; these number differ only in MSB. So it is with the entire gray code; every number differs by only one bit from the preceding number. The logic diagram for binary code to gray code converter and gray code to binary code converter is shown in fig. 1 and fig. 2.

Logic diagram & Truth Table:

(Logic 1 = +5V & Logic 0= GND)

Decimal	Gray Code	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001

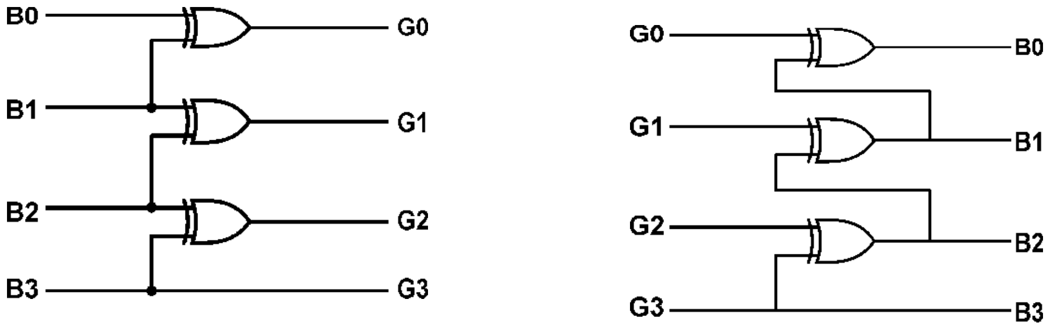


Fig. 7.1 a. Binary to Gray Code b. Gray to Binary Code

The logic equations for Binary to Gray Code conversion:

$G0 = B0 \oplus B1$
$G1 = B1 \oplus B2$
$G2 = B2 \oplus B3$
$G3 = B3$

Gray to Binary Code conversion:

$B0 = G3 \oplus G2 \oplus G1 \oplus G0$
$B1 = G3 \oplus G2 \oplus G1$
$B2 = G3 \oplus G2$
$B3 = G3$

PROCEDURE:

1. Connect +5 V and ground to their indicated position on **DB06** experiment board from external DC power supply or from DC power block of **Digital Lab ST2611.**
2. Connect inputs B0, B1, B2, B3 as per truth table 2 to binary to gray code converter as shown in figure.
3. Switch ON the power supply.
4. Observe output G0, G1, G2, G3 on multimeter or on LED Display of **Digital Lab ST2611.**
5. Repeat above step for remaining inputs and prove truth table.
6. Repeat above steps for gray to binary code converter and prove truth table.

OBSERVATION TABLE:

1. Binary to Gray Code Conversion:

Decimal	**B3**	**B2**	**B1**	**B0**	**Decimal**	**G3**	**G2**	**G1**	**G0**
0	0	0	0	0	0				
1	0	0	0	1	1				
2	0	0	1	0	2				
3	0	0	1	1	3				
4	0	1	0	0	4				
5	0	1	0	1	5				
6	0	1	1	0	6				
7	0	1	1	1	7				
8	1	0	0	0	8				
9	1	0	0	1	9				
10	1	0	1	0	10				
11	1	0	1	1	11				
12	1	1	0	0	12				
13	1	1	0	1	13				
14	1	1	1	0	14				
15	1	1	1	1	15				

2. Gray to Binary Code Conversion:

Decimal	G3	G2	G1	G0	Decimal	B3	B2	B1	B0
0	0	0	0	0	0				
1	0	0	0	1	1				
2	0	0	1	0	2				
3	0	0	1	1	3				
4	0	1	0	0	4				
5	0	1	0	1	5				
6	0	1	1	0	6				
7	0	1	1	1	7				
8	1	0	0	0	8				
9	1	0	0	1	9				
10	1	0	1	0	10				
11	1	0	1	1	11				
12	1	1	0	0	12				
13	1	1	0	1	13				
14	1	1	1	0	14				
15	1	1	1	1	15				

RESULT:

EXPECTED VIVA QUESTIONS:
1. What is gray code?
2. What are the advantages of gray code?
3. How to convert binary to gray code?
4. How to convert gray to binary code?

NAME OF FACULTY: PROF. MANISH BARVE

SIGNATURE:

DATE: