EXPERIMENT NO. 2

APPLICATIONS OF LOGIC GATES AND K-MAP

NAME: ; ID: ;

GROUP NO.: ;DATE .

Learning Outcomes

1. Use a Karnaugh map to simplify the expression.
2. Build and test logic that implements the simplified expression for following applications
   a. Addition of two bits
   b. 4-bit Gray to Binary Code conversion
   c. Invalid BCD code detector
   d. Select one of the two inputs and pass it to output.
   e. Compare two bits

Equipments and components Required

Analog/Digital IC tester, Power Supply, Bread Board, LED Panel, Panel of toggle switches,Digital Multi Meter (DMM).

7408 – Quad 2-input AND	7432 – Quad 2-input OR
7486 – Quad 2-input XOR	7404 _ Quad 1-input NOT

Experiment:

Verify following parameters and complete the checklist before circuit implementation.

S.No.	Description	Remark (√ OR ×)
1	Check DC voltage from power supply using Multimeter.	
2	Is proper voltage range of DC power supply selected?	
3	Are all switches and LEDs working?	
4	Are all ICs tested using IC tester?	
5	Are required ICs placed correctly on bread board?	
6	Identified pin no.1 of IC?	
6	Do you know Logical function and pin configuration of all ICs? (Refer Data sheets)	
7	DC power supply is OFF till you complete the connections.	
8	All the truth tables with theoretically calculated values and required circuit diagrams are complete?	

<u>Run 01#:</u> Following truth table as Table 2.1 shows output of a logical circuit that adds two binary bits. The circuit is known as Half Adder. The circuit has two inputs and two outputs S (SUM) and C (CARRY).Derive logical expression for S And C from the given truth table in SOP form. Build and test logic that implements the simplified expression.

Include Pin numbers of each logic gate on the drawing.

<table>
<tr><td colspan="2">i/p</td><td>o/p</td><td rowspan="2">Logic
For ‘S’</td><td rowspan="2">Observed output (S)</td><td>o/p</td><td rowspan="2">Logic
for ‘C’</td><td rowspan="2">Observed Output (C)</td></tr>
<tr><td>A</td><td>B</td><td>S</td><td>C</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td></td><td></td><td>0</td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td></td><td></td><td>0</td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td></td><td></td><td>0</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td><td></td><td>1</td><td></td><td></td></tr>
<tr><td colspan="8">Logical Expression (SOP form) for ‘S’ =</td></tr>
<tr><td colspan="8">Logical Expression (SOP form) for ‘C’ =</td></tr>
</table>

Table 2.1: Truth Table for Half Adder

Fig 2.1: Circuit Diagram for Half Adder

<u>Run #02</u>: As an application of XOR gates, we would be implementing a circuit for converting a 4-bit gray code to its equivalent binary number. Given a 4 bit gray code $G_3G_2G_1G_0$ it equivalent binary number given by:

B_3 =…. **B_2 = ……..** **B_1 = ……….** **B_0 = ……….**

Draw circuit diagram as Fig 2.2. ***Include Pin numbers of each logic gate on the drawing.***

Fig 2.2: Gray to Binary code converter

Note down your observations in following truth table as Table 2.2.

$G_3G_2G_1G_0$	$B_3B_2B_1B_0$ (output)	$B_3B_2B_1B_0$ (Observed Output)	$G_3G_2G_1G_0$	$B_3B_2B_1B_0$ (output)	$B_3B_2B_1B_0$ (Observed Output)
00000			1100		
0001			1101		
0011			1111		
0010			1110		
0110			1010		
0111			1011		
0101			1001		
0100			1000		

Table 2.2: Truth Table for Gray to Binary code conversion

Run 03 #: Implement a logic circuit to act as a invalid BCD code detector using Logic Gates. Complete the truth table shown as Table 2.3 with your observations.

Decimal equivalent	B3	B2	B1	B0	Valid BCD	Invalid BCD	Observed output
0	0	0	0	0	1	0	
1	0	0	0	1	1	0	
2	0	0	1	0	1	0	
3	0	0	1	1	1	0	
4	0	1	0	0	1	0	
5	0	1	0	1	1	0	
6	0	1	1	0	1	0	
7	0	1	1	1	1	0	
8	1	0	0	0	1	0	
9	1	0	0	1	1	0	
10	1	0	1	0	0	1	
11	1	0	1	1	0	1	
12	1	1	0	0	0	1	
13	1	1	0	1	0	1	
14	1	1	1	0	0	1	
15	1	1	1	1	0	1	

Table 2.3: Truth Table for Invalid BCD code detector

(a) Complete and use K-map shown as Figure 2.3 to simplify boolean expression for the invalid codes. Draw the circuit diagram as Fig 2.4,space given below.

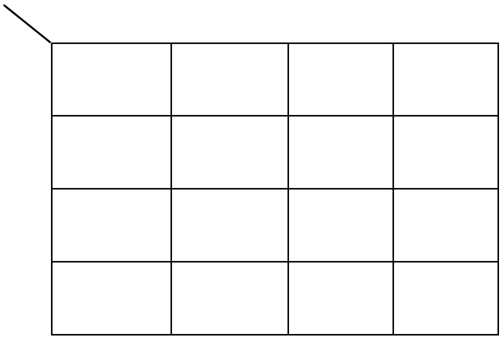

Fig2.3: k-map for invalid BCD code detector

Simplified logical Expression for Invalid BCD code detector circuit:=

Fig 2.4: Circuit Diagram for invalid BCD code detector

Run 04 #: Following truth table as in Table 2.4 shows output of a logical circuit which selects one of the two inputs and pass it to output. The circuit is called as a 2:1 Multiplexer (MUX). Complete K-map shown as Figure 2.5 and use it to simplify logical expression for 2:1 MUX. Draw the circuit diagram as Fig 2.6,space given below.

Input			o/p	Z (Observed output)
A1	A0	S	Z	
0	0	0	0	
0	0	**1**	0	
0	1	0	1	
0	1	**1**	0	
1	0	0	0	
1	0	**1**	1	
1	1	0	1	
1	1	**1**	1	

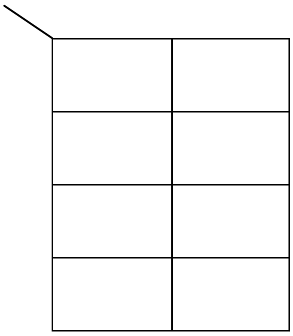

Fig 2.5:K-map for 2:1 MUX

Table 2.4: Truth Table for2:1 MUX;

Fig 2.6: Circuit diagram for 2:1 MUX

Implement the circuit and record your observations as observed output in Table 2.4

Run 05 #: Following truth table as in Table 2.5 shows output of a logical circuit which compare two bits …. The circuit is called as a 1 bit comparator.Complete K-map shown as Figure 2.7 and simplify logical expression. Draw the circuit diagram as Fig 2.8.Record your observations in Table 2.6

Input		A>B	A>B (Observed output)	A<B	A<B	A=B	A=B
A	B				(Observed output)		(Observed output)
0	0	0		0		1	
0	1	0		1		0	
1	0	1		0		0	
1	1	0		0		1	

Table 2.5: Truth Table for 1 bit comparator

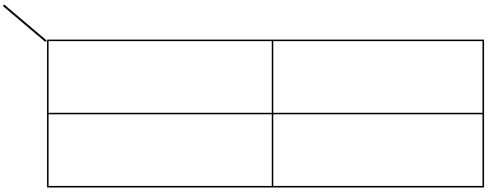

Fig 2.7: K-map for 1 bit comparator

Fig 2.8: Circuit Diagram for 1 bit comparator