Digital Logic Design Lab# 4

OBJECTIVES:

- Writing a function in terms of Min &Max Terms from a statement.
- Learning how to do Minimization of Boolean function using K-maps. (SOP and POS form)
- To obtain the simplest implementation for given function by circuit optimization that require cost criteria.
- To study the realization of basic gates using universal gates (NAND gate & NOR gate)
- To learn technology mapping (NAND-NAND & NOR-NOR implementation) and its significance in order to obtain cost effective circuit for implementation.

EQUIPMENT: Logic trainer, Logic probe

COMPONENTS: ICs 74LS08, 74LS32, 74LS04, 74LS02, 74LS00

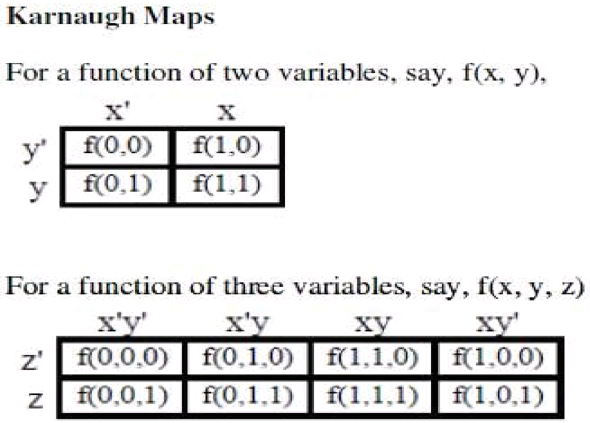
THEORY:

When a Boolean expression is implemented with logic gates, each term requires a gate, and each variable within the term designates an input to the gate. We define a **literal** as a single variable within the term that may or may not be complemented. By reducing the number of terms, the number of literals, or both in a Boolean expression, it is often possible to reduce the cost of circuit and obtain a simpler circuit. **Boolean algebra is applied to reduce an expression for obtaining a simpler circuit.** A Boolean function can be written in a variety of ways when expressed algebraically. There are, however, a few ways of writing algebraic expressions that are considered to be standard forms. The standard forms facilitate the simplification procedures for Boolean expressions and frequently result in more desirable logic circuits. The standard forms contain product terms and sum terms. An example of a product term is XYZ. This is a logical product consisting of an AND operation among three literals. An example of a sum term is X+Y+Z. This is a logical sum consisting of OR operation among the literals. Any Boolean expression can be written in form of **SOP and POS forms.**

A Boolean function can be represented by a **Karnaugh map** in which each cell corresponds to a minterm. The cells are arranged in such a way that any two immediately adjacent cells correspond to two minterms of distance 1. K-map is used to simplify Boolean expressions. K-Map is a grid- like representation of a truth table that gives more insight. Optimal groups of 1s or 0s are identified, which represent the terms of a canonical form of the logic in the original truth table. The minimization will result in reduction of the number of gates (resulting from less number of terms) and the number of inputs per gate (resulting from less number of variables per term). The minimization will reduce cost and power consumption of the logic circuit.

Cost: Literal means variable or its complement. Literal cost is the number of Literal appearances in Boolean expression corresponding to logic circuit diagram expressed as L.

Gate Input Cost is the no of inputs to gate in implementation corresponding exactly to given equation.



Karnaugh Maps

For a function of two variables, say, f(x, y),

	x'	x
y'	f(0,0)	f(1,0)
y	f(0,1)	f(1,1)

For a function of three variables, say, f(x, y, z)

	x'y'	x'y	xy	xy'
z'	f(0,0,0)	f(0,1,0)	f(1,1,0)	f(1,0,0)
z	f(0,0,1)	f(0,1,1)	f(1,1,1)	f(1,0,1)

The design of a combinational circuit starts from the specification of the problem and culminates in a logic diagram or netlist that describes a logic diagram. The procedure involves the specification, formulation, optimization, & technology mapping.

Technology mapping is actually transformation of logic diagram or netlist to a new diagram using the available implementation technology. Typically NAND and NOR gates are more desirable to use in technology mapping due to the following reasons:

1) NAND and NOR gates are said to be universal gates where universal gate is a gate which can implement any Boolean function without needing any other type of gate.
2) Using universal gate in technology mapping may further reduce cost of optimized logic diagram.
3) Universal gates are easier to fabricate with electronic components.

A convenient way to implement a Boolean function with NAND gates only (NAND-NAND implementation) is to begin with the optimized logic diagram of the circuit consisting of AND, OR and NOT gates. The function is converted to pure NAND logic by replacing each gate in logic diagram with its representation using NAND gates only as shown in figure 5-1. The function table of the NAND gate is given in table 5-1. After that, all inverter pairs are cancelled. The same conversion procedure is applied to implement a Boolean function with NOR gates only (NOR-NOR implementation).

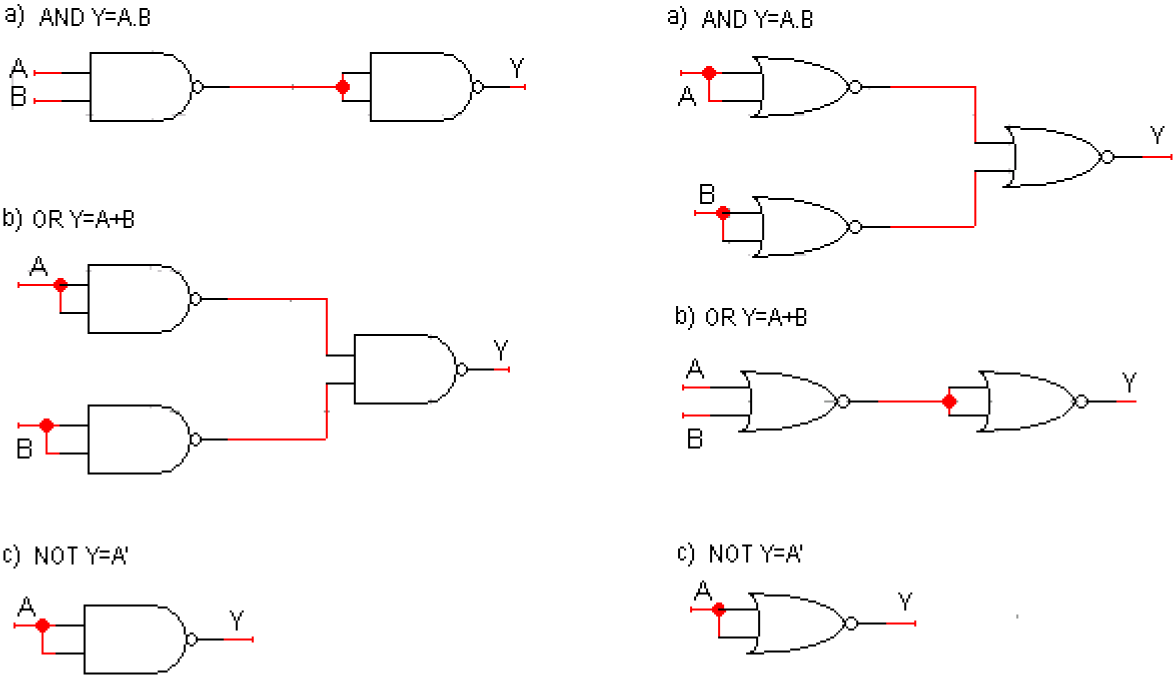


Figure 1: NAND-NAND and NOR-NOR representation of basic logic gates

We will use 74LS00 and 74LS02 ICs for NAND-NAND & NOR-NOR implementation. 74LS00 IC contains four 2-input NAND gates. The function table and connection diagram for this IC are shown below in table 5-2 and figure 5-2 respectively:

Function Table:

Inputs		Output
A	**B**	**Y**
L	L	H
L	H	H
H	L	H
H	H	L

Table 5-1: Nand gate truth table

Connection Diagram:

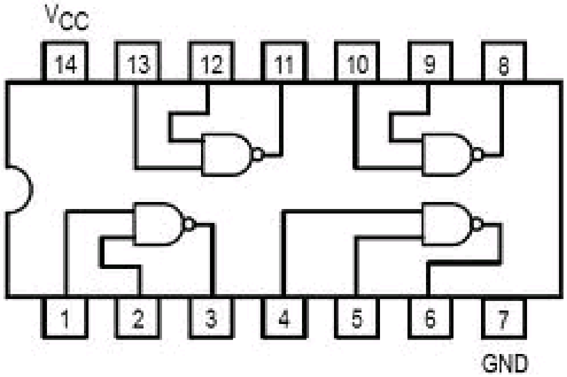


Figure 5-2: 74LS00 Connection diagram

74LS02 IC contains four 2-input NOR gates. The function table and connection diagram for this IC are shown below in table 5-2 and figure 5-3 respectively:

Function Table:

Inputs		Output
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

Table 2: Nor gate truth table

Connection Diagram:

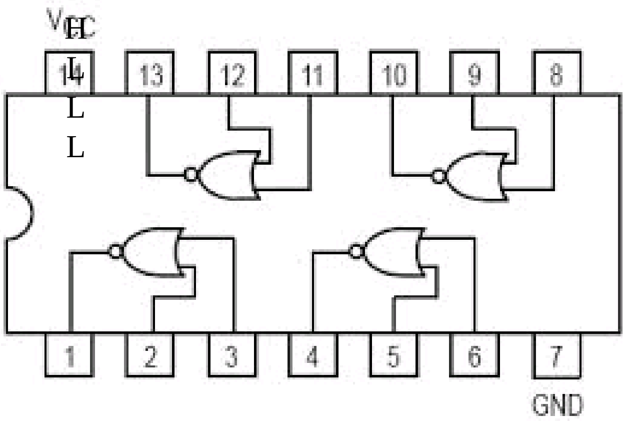


Figure 3: 74LS02 Connection diagram

LAB TASKS

Lab Task # 1:

Q. Do Minimizations of the following 4 Variable Boolean Function using K-MAP.

$$ABC'D+A'BCD+A'B'C'+A'B'D+AC+AB'C+B'$$

Lab Task # 2:

Q. Find Max terms from the following Min terms and write its expression. Also, implement the resultant expression on Logic Works.

$$F(X,Y,Z) = \Sigma m\ (1,3,6,7)$$

Lab Task #3:

Q. For the Boolean function F1(A, B, C,D) = ∑ m(0,2,4,6,7,8,10,12,14,15) do the following:

a) Find truth table and minimal SOP expression for Boolean function F1 using K-Maps and implement it on Logic Works 4

Lab Task #4:

For the logic circuit given below in figure do the following:

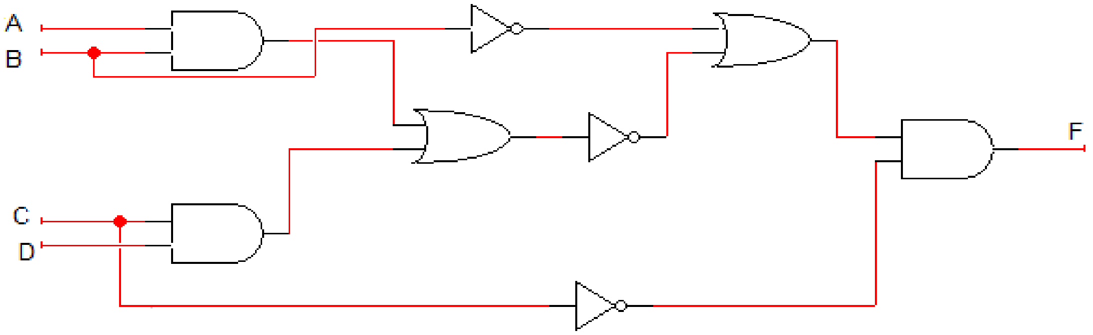


Figure: logic Circuit 1

a) Write the Boolean expression for **F**

b) Draw the truth table for Boolean expression **F.**

c) Write the Boolean function **F** in canonical form using minterms.

d) Write the Boolean function **F** in canonical form using maxterms.

e) Fill the following table in order to determine the gate cost for the implementation of Boolean function **F** found in part (a)

IC type	Required No. of Gates	Gates per IC	Required No. of ICs
Total no. of ICs			

Table: Gate Cost task 1 part (a)

f) Transform the given diagram of logic circuit to new logic diagram using NAND gates only, in the space given below. Show complete working. Implement the transformed logic circuit on logic trainer.

g) Fill the following table in order to determine the gate cost for the implementation of Boolean function **F** using NAND gates only.

IC type	Required No. of Gates	Gates per IC	Required No. of ICs
74LS00 (NAND gate IC)			
Total no. of ICs			

Table: Gate Cost task 1 part (f)

h) Transform the given diagram of logic circuit to new logic diagram using NOR gates only, in the space given below. Show complete working. Implement the transformed logic circuit on logic trainer.