Digital Logic Design
(EL-227)
Spring-2021

LAB 05
Simplification of Digital Circuits and Karnaugh map

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), KARACHI

Date: 8th March 2021

Lab Session 05: Simplification of Digital Circuits and Karnaugh Map

OBJECTIVES:

The objectives of this lab is:

- To learn K-map and its usage in order to obtain cost effective circuit for implementation
- Simplification of circuits using De-Morgan's Theorem

APPARATUS:

- Logic trainer
- Logic probe

COMPONENTS:

ICs 74LS02, 74LS00, 74LS08, 74LS32, 74LS04, Jumper Wire

Introduction:

De-Morgan‘s laws provide mathematical verification of the equivalency of the NAND and negative-OR gates and the equivalency of the NOR and negative-AND gates. The complement of a product of variables is equal to the sum of the complements of the variables. The complement of two or more AND variables is equivalent to the OR of the complements of the individual variables. The De Morgan’s statements are,

Statement 1:

“The negation of conjunction is the disjunction of the negations”. Or we can define that as “The compliment of the product of 2 variables is equal to the sum of the compliments of individual variables”.

$$(A.B)' = A' + B'$$

Statement 2:

“The negation of disjunction is the conjunction of the negations”. Or we can define that as “The compliment of the sum of two variables is equal to the product of the compliment of each variable”.

$$(A + B)' = A'.B'$$

Truth Tables

The De Morgan’s laws are simply explained by using the truth tables. The truth table for De Morgan’s first statement($(A.B)' = A' + B'$) is given below.

A	B	A'	B'	A.B	(A.B)'	A'+B'
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

Table 1: Statement 1

The truth table for De Morgan's second statement ((A + B)' = A'.B') is given below.

A	B	A'	B'	A+B	(A+B)'	A'.B'
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Table 1: Statement 1

Figures of the about two statement shown below:

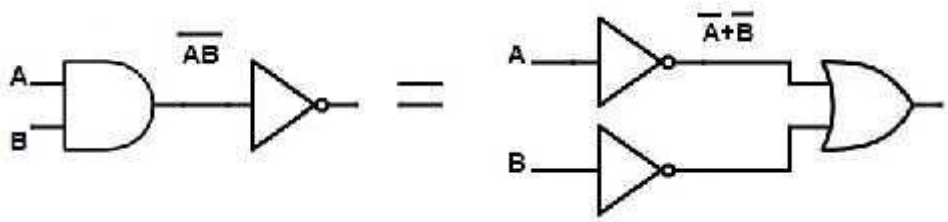


Figure 1 : NAND gate= Bubbled OR gate

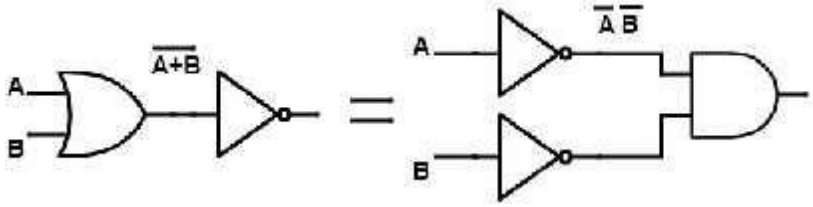


Figure 2 : NOR gate= Bubbled AND gate

Simpler expressions yield simpler hardware:

The proof is shown in table, which shows the truth table and the resulting logic circuit simplification.

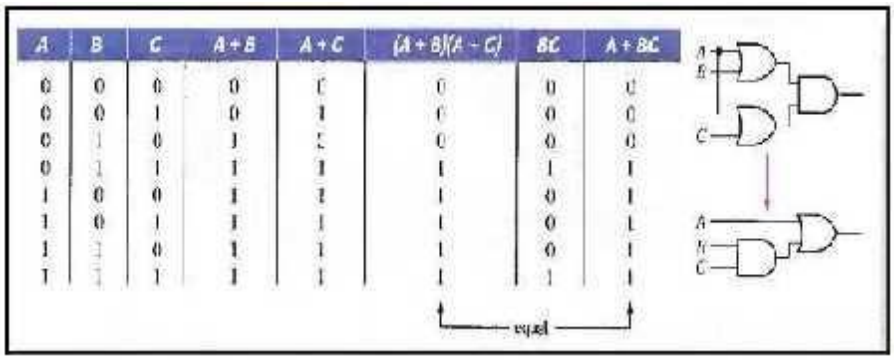


Figure 2 : Simplification of circuit

K-MAP:

The Karnaugh map (K-map) is a method used to simplify Boolean expressions. K-Map is a grid- like representation of a truth table that gives more insight. The required Boolean results are transferred from a truth table onto a two-dimensional grid where the cells are ordered in gray code and each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. Optimal groups of 1s or 0s are identified, which represent the terms of a canonical form of the logic in the original truth table. These terms can be used to write a minimal Boolean expression representing the required logic.

Karnaugh map is used to obtain optimized logic representation so that it can be implemented using a minimum number of logic gates. The sum-of-product form can always be implemented using AND gates feeding into an OR gate, and a product-of-sum form leads to OR gates feeding an AND gate.

Universality of logic Gates:

1. The NAND Gate as a Universal Logic Element

Any logic expression can be implemented using only NAND gates or only NOR gates and no other type of gate. NAND gates alone in the proper combination, can be used to perform each of the basic Boolean operations OR, AND, and INVERT.

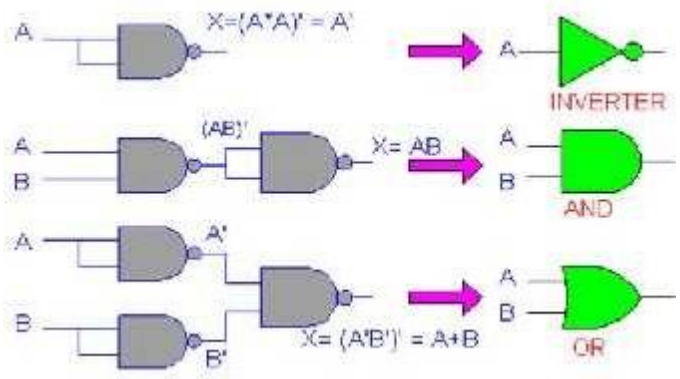


Figure 3 : NAND gate based Basic Gates

2. The NOR Gate as a Universal Logic Element

It can be shown that NOR gate can be arranged to implement any of the Boolean operations.

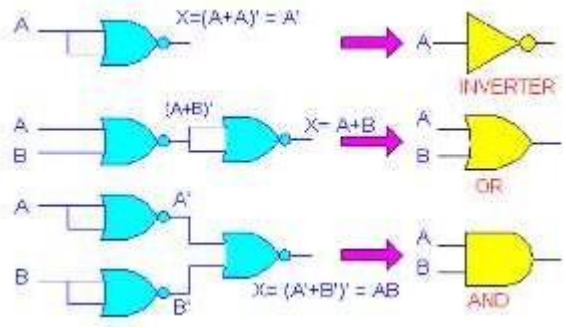


Figure 4 : NOR gate based Basic Gates

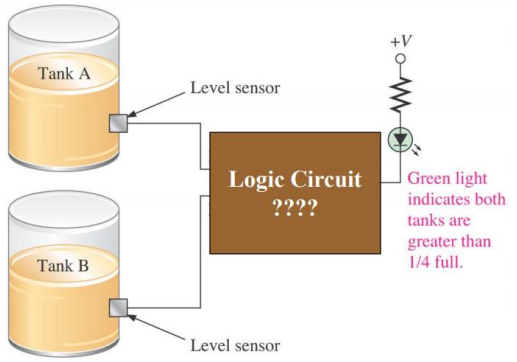
Report Experiment 05

Implement the following scenario/ Logic on Trainer

Two tanks store certain liquid chemicals that are required in a manufacturing process. Each tank has a sensor that detects when the chemical level drops to 25% of full. The sensors produce a HIGH level of 5 V when the tanks are more than one-quarter full. When the volume of chemical in a tank drops to one-quarter full, the sensor puts out a LOW level of 0 V.

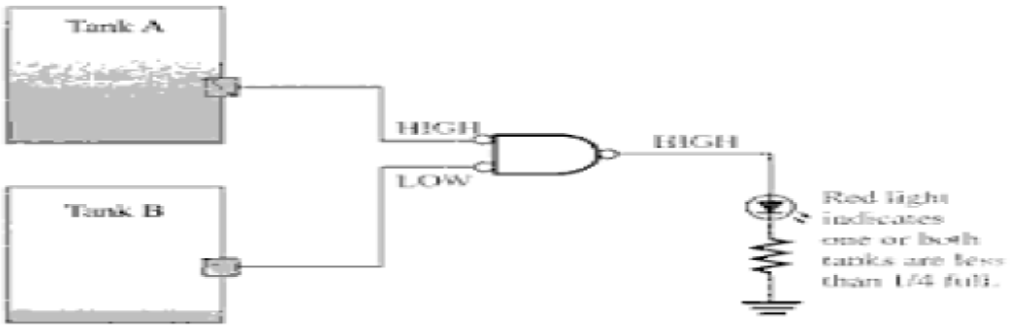
It is required that a single red light-emitting diode (LED) on an indicator panel show when both tanks are more than one-quarter full. Show how a NAND gate can be used to implement this function.

APPLICATION



Solution

- It is required that a single red light-emitting diode on an indicator panel show when both tanks are more than one-quarter full. Show how a NAND gate can be used to implement this function.
- If tank A and tank B are above one-quarter full, LED is off.
- As long as both sensor outputs are LOW(0V), indicating that both tanks are less than one-quarter full, NAND gate output is LOW(0V). red LED circuit is arranged so that a LOW voltage turns it on.



Exercise # 02 Analysis and Design Logic Circuit on Logic Works for the following scenario/ Logic.

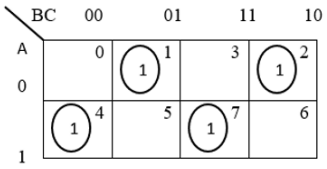
For the process described in Exercise 01 it has been decided to have a red LED display come on when at least one of the tanks falls to the quarter-full level rather than have the green LED display indicate when both are above one quarter. Design circuit on logic works that shows how this requirement can be implemented.

Exercise # 04 Design the even and odd parity generator circuits for a four data on Logic Works.

3-bit even parity generator truth table

3 Bit Message			Even Parity Generator
A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The karnaugh map (k-map) simplification for three-bit input even parity is



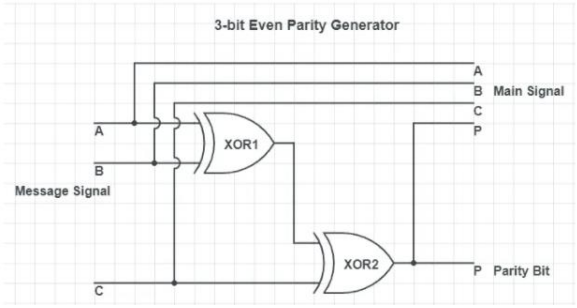
Solving the truth table for all the cases where P is 1 using Sum-of-Products method:

$$P = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,B\,C$$

$$= \overline{A}\,(\overline{B}\,C + B\,\overline{C}) + A\,(\overline{B}\,\overline{C} + B\,C)$$

$$= \overline{A}\,(B \oplus C) + A\,\overline{(B \oplus C)}$$

$$P = A \oplus B \oplus C$$



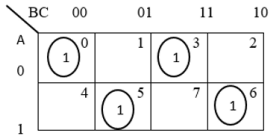
3-bit Odd Parity Generator truth table

3 Bit Message			Odd Parity Bit Generator
A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	x

Odd Parity Generator Truth Table

A B C	Odd Parity
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	0

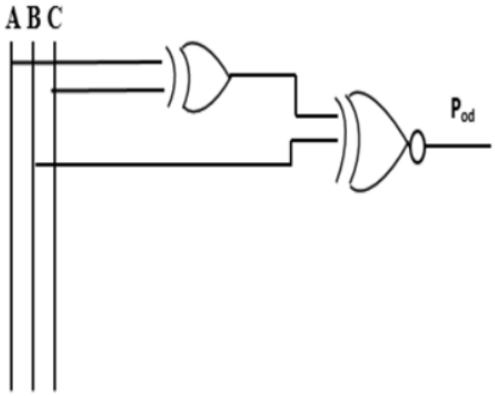
The Kavanaugh map (k-map) simplification for three-bit input odd parity is



From the above odd parity truth table, the parity bit simplified expression is written as

$$P_{od} = (\bar{A}\,\bar{B}\,\bar{C} + A\,\bar{B}\,C) + (\bar{A}\,B\,C + A\,B\,\bar{C})$$
$$= \bar{C}\,(A\,B + \bar{A}\,\bar{B}) + B\,(A\,\bar{B} + \bar{A}\,B)$$
$$= \bar{C}\,(A \oplus B) + C(A \oplus B)$$
$$= \bar{C}\,\bar{X} + C\,X = C \odot X = C \odot \overline{A \oplus B}$$

The logic diagram of this odd parity generator is shown below.



Exercise # 03 Analysis and Design Logic Circuit on Logic Works for the following scenario/ Logic.

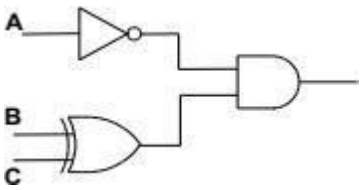
As part of an aircraft's functional monitoring system, a circuit is required to indicate the status of the landing gears prior to landing. A green LED display turns on if all three gears are properly extended when the "gear down" switch has been activated in preparation for landing.

A red LED display turns on if any of the gears fail to extend properly prior to landing. When a landing gear is extended, its sensor produces a LOW voltage. When a landing gear is retracted, its sensor produces a HIGH voltage. Implement a circuit to meet this requirement.

Exercise # 05 Implement the following scenario/ Logic on Trainer

A certain system contains two identical circuits operating in parallel. As long as both are operating properly, the outputs of both circuits are always the same. If one of the circuits fails, the outputs will be at opposite levels at some time. Devise a way to monitor and detect that a failure has occurred in one of the circuits **implement the following scenario/ Logic on Trainer**.

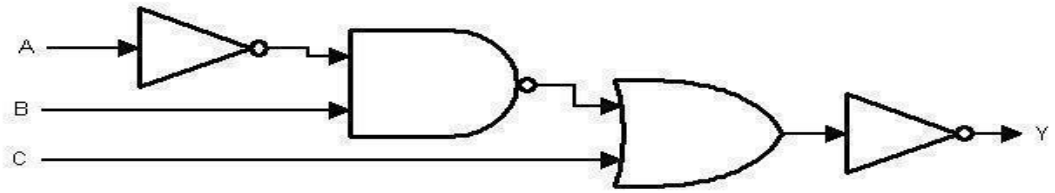
Exercise # 06 implement the given circuit on Trainer and draw Truth tables:



Exercise # 07 Implement the following scenario/ Logic on Trainer

A device is needed to indicate when two LOW levels occur simultaneously on its inputs and to produce a HIGH output as an indication. Specify the device and implement on Trainer.

Exercise # 08 Write the Boolean expression for given circuit and also implement the given circuits on Trainer and draw Truth tables:



TRUE/FALSE

1. An inverter performs a NOT operation.
2. A NOT gate cannot have more than one input.
3. If any input to an OR gate is zero, the output is zero.
4. If all inputs to an AND gate are 1, the output is 0.
5. A NAND gate can be considered as an AND gate followed by a NOT gate.
6. A NOR gate can be considered as an OR gate followed by an inverter.
7. The output of an exclusive-OR is 0 if the inputs are opposite.

LAB TASK#1:

Design circuit diagram in Logic Works given expressions by using either NAND or NOR gate.

1. **ABC + D' + E'**
2. **ABC + DE**

LAB TASK#2:

Simplify the following Boolean expression: and design circuit in Logic Works.

$$\bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$$

$$\overline{A\bar{B} + A\bar{C}} + \bar{A}\bar{B}C$$

Basic rules of Boolean algebra.

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \bar{A} = 0$
3. $A \cdot 0 = 0$	9. $\bar{\bar{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \bar{A}B = A + B$
6. $A + \bar{A} = 1$	12. $(A + B)(A + C) = A + BC$

A, B, or C can represent a single variable or a combination of variables.

LAB TASK#3:

Apply DeMorgin's theorems to each of the following expression and design circuit diagram on Logic Works.

(a) $\overline{(A + B + C)D}$

(b) $\overline{ABC + DEF}$

(c) $\overline{A\bar{B} + \bar{C}D + EF}$

LAB TASK#4

The Boolean expression for an exclusive-OR gate is AB' + A'B. With this as a starting point, use DeMorgin's theorems and any other rules or laws that are applicable to design an expression for the exclusive-NOR gate and design circuit in Logic Works.

LAB TASK#5:

.Convert the following Boolean expression into standard SOP form and design circuit diagram in Logic Works:

$$A\bar{B}C + \bar{A}\bar{B} + AB\bar{C}D$$

LAB TASK#6:

.Convert the following Boolean expression into standard POS form and design circuit diagram in Logic Works:

$$(A + \bar{B} + C)(\bar{B} + C + \bar{D})(A + \bar{B} + \bar{C} + D)$$

LAB TASK#7:

Develop a truth table for the standard SOP expression $\bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$.

LAB TASK#8:

Use a Karnaugh map to minimize the following standard SOP expression and design circuit diagram on Logic Works

$$A\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

LAB TASK#9:

Use a Karnaugh map to minimize the following standard POS expression and design circuit diagram on Logic Works

$$(A + B + C)(A + \bar{B} + C)(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)$$