American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE 2206: Digital Logic Design Laboratory

Title: Deriving logic equations and truth table from a given statement or expression and construction of combinational circuits

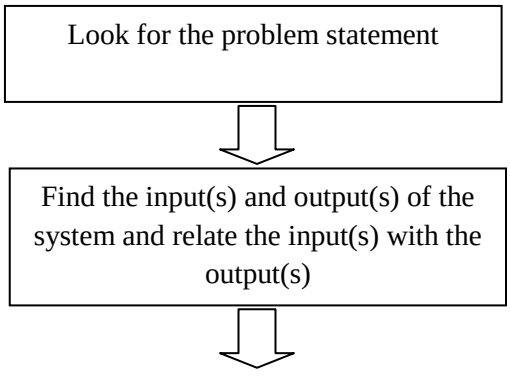
Abstract:
This experiment is designed to-
1. Help students implement the logic circuits derived from a given statement in the breadboard using gate ICs and observe whether the output verifies the truth table of the given logic statement or not.
2. Perform relevant theoretical work by deriving the logic circuit and truth table from the given logic equation/statement and get familiarized with Boolean algebra and De Morgan's law.
3. Simplify the logic expressions with K-Map and verify accuracy by breadboard implementation.

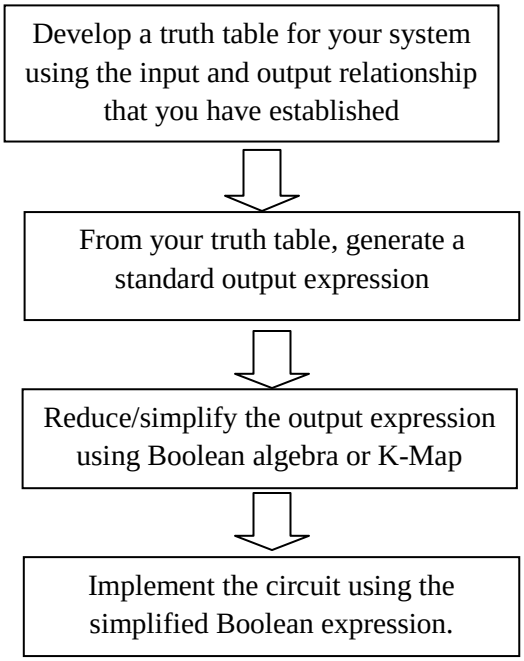
Introduction:
From any given logic statement, it is possible to construct a digital logic circuit. The first step in this process is to construct a truth table and then determine a standard SOP (sum of products) or POS (product of sums). At the same time, it is also possible to derive a logic expression from a given combinational circuit diagram by observing the individual logic operations performed in the circuit and matching them with their corresponding logic gates. Expressions are simplified using Boolean algebra and De Morgan's law or K-Map to reduce the number of gates used. Then the circuit is implemented in the breadboard using gate ICs and observed whether the output verifies the truth table of the given statement.

This experiment shows the students a practical verification of deriving logic equations and truth table from combinational circuits. Knowing how to derive logic equations and truth table from combinational circuits helps a person with detecting the output logic expressions from any unknown logic circuit.

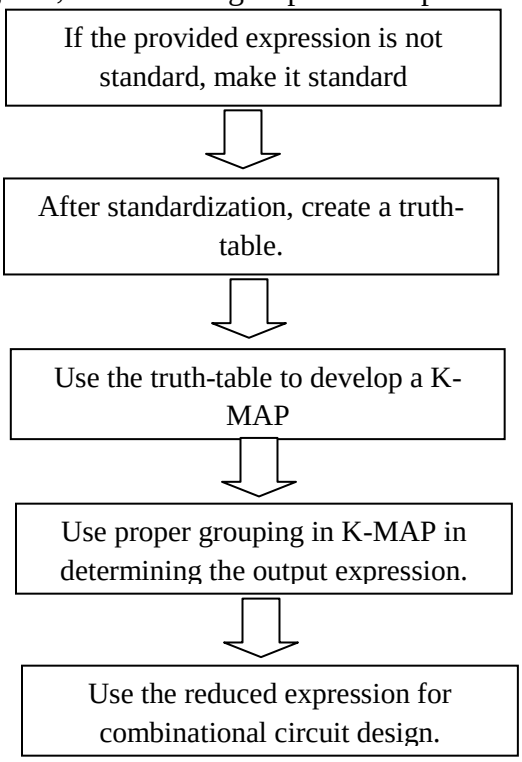
Theory and Methodology:
Combinational circuits are built with logic gates and other components. It does not include any values to be taken from a previous state of the circuit. Designing such a combinational digital system requires use of one of the following methods:

1. If a problem statement is given, the following steps will help designing the system





2. Or if an expression is given, the following steps will help in designing the system



Some useful definitions related to these procedures are given below:

Boolean algebra: In Boolean algebra, a variable is a symbol used to represent an action, a condition, or data. A single variable can only have a value of 1 or 0.

1. Variable: A symbol used to represent a logical quantity that can have a value of 1 or 0, usually designated by an italic letter.

2. Complement: The inverse or opposite of a number. In Boolean algebra, the inverse function, expressed with a bar over the variable.

3. Sum term: The Boolean sum of two or more literals equivalent to an OR operation

4. Product term: The Boolean product of two or more literals equivalent to an AND operation.

5. Sum of Products (SOP):

When two or more product terms are summed by boolean addition, the resulting expression is a sum of product. Ex. $Y = \bar{A}B + AB\bar{C} + \overline{AC} + A\bar{B}$

Implementing an SOP expression simply requires ORing the outputs of two or more AND gates. A product term is produced by an AND operation, and the sum (addition) of two or more product terms is produced by an OR operation. Therefore, an SOP expression can be implemented by AND-OR logic in which the outputs of a number (equal to the number of product terms in the expression) of AND gates connect to the inputs of an OR gate.

A standard SOP expression is one in which all the variables in the domain appear in each product term. Ex. $Y = \bar{A}B\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$

Standard SOP expressions are important in constructing truth-tables and in Karnaugh map simplification method.

The SOP expression is equal to 1 only if one or more of the product terms in the expression is equal to 1.

6. **Product of Sums (POS):**

When two or more sum terms are multiplied, the resulting expression is a product of sums (POS). Ex. $Y = (A + B)(A + C)(A + B + \bar{C})$

Implementing a POS expression simply requires ANDing the outputs of two or more OR gates. A sum term is produced by an OR operation, and the product of two or more sum terms is produced by an AND operation. Therefore, a POS expression can be implemented by logic in which the outputs of a number (equal to the number of sum terms in the expression) of OR gates connect to the inputs of an AND gate.

A standard POS expression is one in which all the variables in the domain appear in each sum term in the expression. Ex. $(A + B + C)(A + \bar{B} + C)(A + B + \bar{C})$

A POS expression is equal to 0 only if one or more of the sum terms in the expression is equal to 0.

7. Karnaugh Map:

A Karnaugh map provides a systematic method for simplifying Boolean expressions and, if properly used, will produce the simplest SOP or POS expression possible le, known as the minimum expression.

A Karnaugh map is similar to a truth table because it presents all of the possible values of input variables and the resulting output of each valued. Instead of being organized into columns and rows like truth table, the Karnaugh map is an array of cells in which each cell presents binary value of the input variables. The cells are arranged in a way so that the simplification of a given expression is simply a matter of properly grouping the cells. Karnaugh maps can be used for expressions with two, three, four and five variables. The number of cells in a Karnaugh map is

equal to the total number of possible input variable combinations as is the number of rows in a truth table.

Problem1. A Building has 4 floors which share the same water tank for water supply. In order to start the motor, each floor has a designated switch- Ground Floor with switch A, 1st Floor with switch B, 2nd Floor with switch C and 3rd Floor with switch D. The motor starts if someone presses the switch from the 3rd floor or from both ground and 2nd floor or from 1st and 2nd floor. Your job is to design the system.

Problem2. For the expression $\overline{AB + AC} + \bar{A}\bar{B}C$, find the truth-table, reduced expression using K-MAP and the logic gate diagram.

Pre-Lab Homework: Students must study the Boolean algebra rules and De Morgan's Law. They should also study how to derive truth table from logic expressions. They need to perform simulation of the circuits shown in the circuit diagram sections using PSIM and MUST present the simulation results to the instructor before the start of the experiment.

Apparatus:

1. Digital trainer board
2. IC 7432:1 pcs
3. IC 7408:1 pcs
4. IC 7404:2 pcs
5. IC 7402:1 pcs
6. IC 7400:1 pcs
7. IC 7486:1 pcs
8. Connecting wires

Precaution: The IC contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages. For proper operation, V_{in} and V_{out} should be constrained to the range GND to V_{CC}. Connect the ICs according to their pin configuration carefully and use connecting the wires with the ICs to make sure that they are firmly connected. Check whether all the data switches and output showing LEDs are working.

Experimental Procedure:

Problem1:

a) Draw a truth table to represent the output Y.
b) Use the truth table outputs to form standard SOP and POS expressions.
c) Minimize the SOP expression using Boolean algebra and K-Map. Perform hardware implementation of the circuit and compare with your truth table output.

Problem2:

a) Draw a step-by-step truth table to represent the outputs at each gate (1-6) and then the final output at Y.

b) Use output Y to form standard SOP expression.
c) Minimize the SOP expression using Boolean algebra and K-Map. Perform hardware implementation of the circuit and compare with your truth table output.

Simulation and Measurement:

Compare the simulation results with your experimental data and comment on the differences if any.

Results and Discussion:

Students will implement the circuit in the Trainer Board and match the theoretically obtained truth table by matching outputs for individual input configurations. If the practically obtained truth table does not match they will also investigate the errors.

Students will summarize the experiment and discuss it as a whole. They will observe that the method of deriving logic equations and truth tables is valid and effective. They will also see whether they can make the circuit efficient by reducing the number of gates used. They will also include any limitations of the process. If the connection is not loose, gates are properly biased, all the output LEDs are working and the ICs that are being used are not broken; and then the practically obtained truth table should exactly match the theoretical one. Which will, in turn validate the process.

Report Questions:

1. Construct the derived equations (i) and (ii), using Universal gates (both NAND and NOR).
2. Develop the truth table for a certain three-input logic circuit with the output expression Y=ABC+(AB)'C+A'BC+AB'C+A(B'+C).
3. Implement the following logic expressions with logic gates Y=ABC+AB+AC

References:

1. Thomas L. Floyd, "Digital Fundamentals", available Edition, Prentice Hall International Inc.