

Mariam Elsaqa – 900193220

Alaa Alkady - 900191122

Digital Design 1 Project 1

Three-Variable K-map Logic Minimization

Spring 2021

Project Report

Dr. Mona Farouk

Department of Computer Science and Engineering, AUC

How to use this program:

To run this c++ program, you need to enter the number of minterms you have, then the program will ask you to enter the minterms you have, you should enter them in this form: 0 1 2 3 5, not comma separated. Then, the program will display the kmap and the simplified Boolean expression.

The program outputs the kmap in this form:

0	1	3	2
4	5	7	6

Program Design:

- Data structure used:

Vectors, 2D arrays and dynamic arrays. Vectors are used to store the implicants and 2D arrays to handle the kmap and dynamic array to store the string of simplified Boolean expression.

- In this program there are two structs:

First one (MapElement) to handle each element in the kmap and the other one (output) to handle the implicants.

- There are 10 functions:

- 1. getInput: -**

This is a void function that is used to get the needed information (Number of minterms and the minterms in decimal format) from the user.

2. kmap: -

This is a void function that is used to initialize and build the kmap by filling it with its decimal form and binary form, then check if the user entered this decimal number, if yes fill this spot with '1'.

3. HighestNumberOfP: -

This function takes 5 vectors (which stores all possible combinations in a cell that is initialized by '1') and returns the vector of maximum size. It compares the sizes of the five vectors passed to it and store the one with the highest size as a return value.

4. findPI: -

This is a void function that combine adjacent cells and compare them together. It calls HighestNumberOfP function as the vector of maximum size is the vector that should be stored as an implicant. It also counts the cell as included if it is used in the process.

5. includeCount: -

This is a void function that set each implicant with its insertion count.

6. includeCountAfterRemovingDuplicates: -

This void function is called in removeDuplicates to update the insertion count after removing the repeated implicants. The insertion count decreases by one.

7. removeDuplicates: -

This is a void function that is used to compare the resulted implicants and remove the repeated ones to leave it only once.

8. findEPI: -

This is a void function that is used to highlight Essential Prime implicants by checking if their insertion count is one or more. EPI is important to be highlighted as they have to be included in the finalized simplified Boolean expression.

9. simplifiedBoolExp: -

This is a void function where the finalized simplified Boolean expression is formed.

It compares the minterms and check if they are the same or not, if yes we check if it is 0 or 1, if 0 we add “ ’ ” after the letter. Letters are identified using ASCII as A is equal 65, so if the counter is 0 the letter is A, if the counter is 1 the letter is B, if the counter is 2 the letter is C.

At the end we erase the last character using the built in function erase as there will be an extra “ + “ sign.

10. PrintResults: -

This void function only calls the functions that contain the final results (kmap and simplifiedBoolExp) to display the results to the user.

In each function we call the preceding and needed function, so in the **main function** we only call the functions that communicate with the user and print the results.