THE AMERICAN UNIVERSITY IN CAIRO

School of Sciences and Engineering

Fall 2021

CSCE230101 Digital Design I

Project 1 Report

Three-variable K-map logic minimization

Submitted to:

Dr. Mona Farouk

Submitted by:

Name: Abdelaziz Zakareya	ID: 900203361
Name: Shahd ElMahallawy	ID: 900194441

3/11/2021

Objective

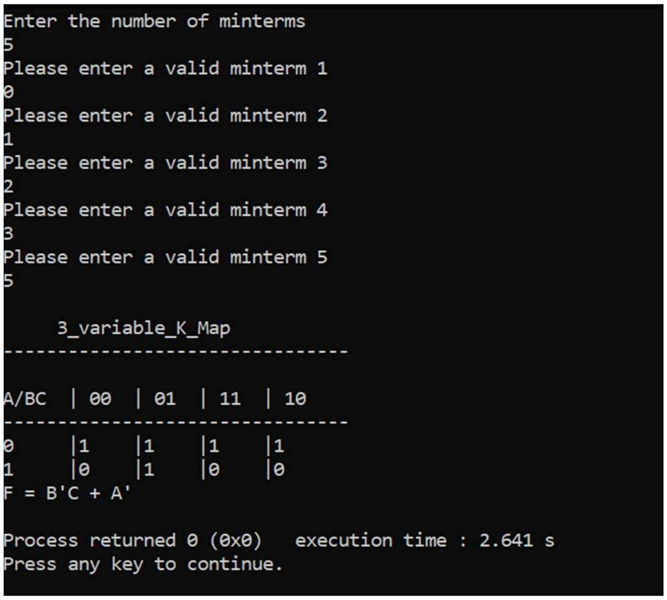
The objective of the project is to design a three-variable k-map that takes the inputs as decimal numbers and prints the k-map along with the simplified boolean expression.

Methods

1. First, we take the minterms from the user as decimal numbers and validate them.
2. We then have two cases, either the number of minterms = 8, so the K-map will have all the value to 1 and the boolean function F =1, or the number of minterms = 0, so the K-map will have all the values = 0, and the boolean function f = 0;
3. Otherwise, we generate the K-map by first initializing it to zero, and set the minterms equal to 1.
4. Then, we print the K-map visually.
5. We then start to find the implicants of the k-map. We first make a struct of Kmap to store the decimal representation, the binary representation, the mapping value of each kmap element.
6. We have only five possible cases in finding the implicants: a) First case, one cell is equal to 1, b) Second case, we check if two neighbouring cells in a row are equal to 1, c) Third case, we check if two neighbouring cells in a column are equal to 1, d) Fourth case, we check if four neighbouring cells in a square are equal to 1, e) Fifth case, we check if four neighbouring cell in a row are equal to 1. By using modulo (%), we guarantee that we will not go beyond the limit of our k-map, and hence we have only these five cases.
7. In every iteration, we get the maximum case of all the cases, because it will be the prime implicant in this case, and hence it will be the minimum implicant that can be realized. Then, we collect all the prime implicants.
8. However, each will have additional implicants that are not prime, so we need to get rid of them. We do this by first grouping the implicants as vectors of the struct kmap. Then we group all the implicants by making a vector of implicants. We first sort the vector of implicants according to the size of its implicants, and then sort each implicant vector ascendingly. In this case, we can get rid of duplicate implicants as well as non-prime implicants.
9. After that, we arrive at the simplest boolean expression in terms of binary representation, so the last step is to convert our binary representation into the letter A, B, C to arrive the boolean expression.

Testing and results

Test 1:



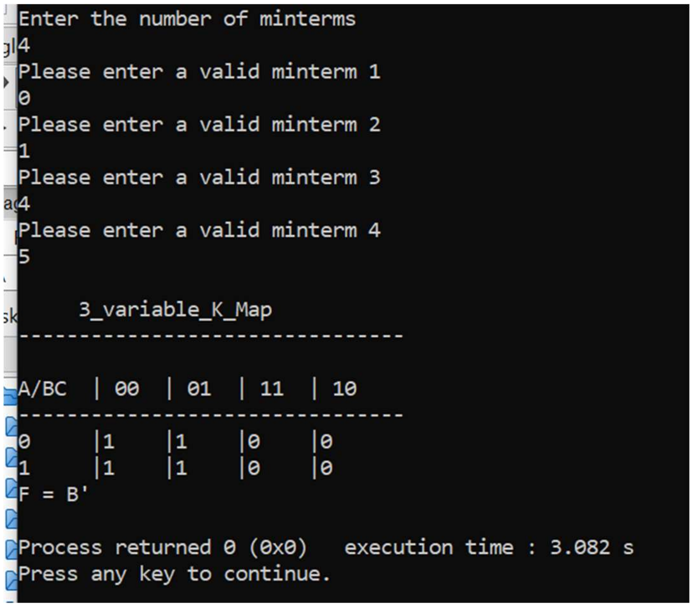
```
Enter the number of minterms
5
Please enter a valid minterm 1
0
Please enter a valid minterm 2
1
Please enter a valid minterm 3
2
Please enter a valid minterm 4
3
Please enter a valid minterm 5
5

     3_variable_K_Map
--------------------------------

A/BC  | 00  | 01  | 11  | 10
--------------------------------
0     |1    |1    |1    |1
1     |0    |1    |0    |0
F = B'C + A'

Process returned 0 (0x0)   execution time : 2.641 s
Press any key to continue.
```

Test 2:



```
Enter the number of minterms
4
Please enter a valid minterm 1
0
Please enter a valid minterm 2
1
Please enter a valid minterm 3
4
Please enter a valid minterm 4
5

     3_variable_K_Map
--------------------------------

A/BC  | 00  | 01  | 11  | 10
--------------------------------
0     |1    |1    |0    |0
1     |1    |1    |0    |0
F = B'

Process returned 0 (0x0)   execution time : 3.082 s
Press any key to continue.
```