Computational Simulation of Quine-McCluskey Algorithm using C/C++

Digital Logic Circuits (EEE2040.01-00), Yonsei University

Taewan Ham¹

wani7@yonsei.ac.kr

I. Introduction

Karnaugh Map method is often used to simplify Boolean functions. It is an intuitive and effective method. However, when the number of variables become larger, additional method is required due to its complexity. Quine-McCluskey algorithm provides more systematic approach than Karnaugh Map method. Therefore, it enables the usage of computer. The goal of this project is to create a computational simulation of Quine-McCluskey algorithm using C/C++.

II. Theoretical explanation of Quine-McCluskey Algorithm

Quine-McCluskey algorithm is consisted in two big steps. The first step is figuring out the prime implicants. Prime implicants can be found by representing min-terms in binary notation and repetitively combining two min-terms if they have only one different variable. To reduce the calculation, each min-terms are grouped based on the number of 1's according to their binary notation. The combination continues until there is no available implicants to combine. Implicants that can't be combined anymore is called prime implicants.

The second step is selecting the minimum set of prime implicants. After the first step, set of prime implicants is selected. Based on this prime implicants, we can construct a prime implicant chart. The row of the chart means prime implicants. The column of the chart means min-terms. If a prime implicant is the only implicant covering the min-term, the prime implicant covering it is called 'essential prime implicant'. After finding the essential prime implicant, we must find the minimum set of prime implicants that can cover all the remaining min-terms. The set of prime implicants can be found by using combination (trial and error) or Petrick's method. The selected set of prime implicants can be expressed in sum-of-products form.

Following is an example of using Quine-McCluskey algorithm on four variable Boolean function. Let's say $f(A, B, C, D) = \Sigma m(4, 8, 9, 10, 11, 12, 14, 15)$. Each min-terms are represented in binary notation and grouped according to their number of 1's.

| min-term | Binary notation | Number of 1's (Group) |
|----------|-----------------|-----------------------|
| m(4) | 0100 | 1 |
| m(8) | 1000 | |

¹ Undergraduate, School of Electrical and Electronic Engineering, Yonsei University

| m(9) | 1001 | 2 |
|-------|------|---|
| m(10) | 1010 | |
| m(12) | 1100 | |
| m(11) | 1011 | 3 |
| m(14) | 1110 | |
| m(15) | 1111 | 4 |

Table 1. Initial procedure

After the initial procedure each min-terms are combined if only one variable differs. The different digit is notated as '_'.

| size 2 | | size 4 | |
|-----------|------|----------------|-----|
| m(4, 12) | _100 | m(8,9,10,11) | 10 |
| m(8, .9) | 100_ | m(8,10,12,14) | 1_0 |
| m(8, 10) | 10 0 | m(10,11,14,15) | 1 1 |
| m(8, 12) | 1_00 | | |
| m(9, 11) | 10 1 | | |
| m(10, 11) | 10_1 | | |
| m(10, 14) | 1 10 | | |
| m(12, 14) | 11_0 | | |
| m(11, 15) | 1_11 | | |
| m(14, 15) | 111 | | |

Table 2. Combining min-terms and repeating it

After finding the prime implicants, we need to construct a prime implicant chart.

| | 4 | 8 | 9 | 10 | 11 | 12 | 14 | 15 |
|------------------|---|---|---|----|----|----|----|----|
| m(4, 12) | X | | | | | X | | |
| m(8, 9, 10, 11) | | X | X | X | X | | | |
| m(8, 10, 12, 14) | | X | | X | | X | X | |
| m(10, 11, 14, | | | | X | X | | X | X |
| 15) | | | | | | | | |

Table 3. Prime implicant chart

Based on this prime implicant chart, we conduct either trial/error method or Petrick's method to solve the minimum sum-of-products of the Boolean function. In trial/error method, we pick the minimum set of min-terms that can eliminate the 'x marks' in the prime implicant chart. In Petrick's method, we make a logical function which is consisted of sum-of-products of Boolean expression that makes each column true. Then, we simplify the logical function by using X + X = X and X + XY = X.

Ⅲ. Implementation of Quine-McCluskey Algorithm in C/C++ code

The overall of the code is consisted in three steps. The first step is getting input from the user. I made error exception using while loop. The second step is finding the prime implicants. The third step is finding the essential prime implicants. For further explanation,

please check out the code comment.

IV. Results of the Simulation

Function (1) - (6) are given test functions. The result of each test functions is shown in Table 4. Screenshots of the result is attached in the end of the paper.

| | (2) $F(a, b, c, d) = \sum m(0, 3, 4, 5, 7, 9, 11, 13)$ |
|--|--|
| QM operation | QM operation |
| ABCDguf | ABCDguf |
| | |
| 0 0 1 0 1 2 1 0 m(2) | 0 0 0 0 0 1 1 0 m(0) |
| 01001310 m(4) | 0 0 1 1 2 2 1 0 m(3) |
| 0 1 0 1 2 2 1 0 m(5) | 0 1 0 0 1 2 1 0 m(4) |
| 0 1 1 0 2 2 1 0 m(6) | 0 1 0 1 2 3 1 0 m(5) |
| 1 0 0 1 2 2 1 0 m(9) | 0 1 1 1 3 2 1 0 m(7) |
| 1 0 1 0 2 2 1 0 m(10) | 1 0 0 1 2 2 1 0 m(9) |
| 1 0 1 1 3 3 1 0 m(11) | 1 0 1 1 3 2 1 0 m(11) |
| 1 1 0 0 2 2 1 0 m(12) | 1 1 0 1 3 2 1 0 m(11) 1 1 0 1 3 2 1 0 m(13) |
| 1 1 0 0 2 2 1 0 m(12) 1 1 0 1 3 4 1 0 m(13) | 1 1 0 1 3 2 1 0 m(13) |
| 1 1 1 1 4 2 1 0 m(15) | QM operation |
| 1 1 1 1 4 2 1 0 m(13) | |
| QM operation | A B C D g u f |
| l · · · · · | |
| ABCDguf | $0.1000010 \parallel m(04)$ |
| 0.11010101011 (2.6) | 010-11010 m(45) |
| 0-1101010 m(26) | 0-1112010 m(37) |
| -1 0 1 0 1 0 1 0 m(2 10) | -1 0 1 1 2 0 1 0 m(3 11) |
| 0 1 0 -1 1 1 1 0 m(4 5) | 01-112010 m(57) |
| 01-101010 m(46) | -1 1 0 1 2 0 1 0 m(5 13) |
| -1 1 0 0 1 1 1 0 m(4 12) | 1 0 -1 1 2 0 1 0 m(9 11) |
| -1 1 0 1 2 1 1 0 m(5 13) | 1 -1 0 1 2 0 1 0 m(9 13) |
| 10-112110 m(911) | |
| 1-1012110 m(913) | List of Prime Implicants |
| 1 0 1 -1 2 0 1 0 m(10 11) | m(04) |
| 1 1 0 -1 2 1 1 0 m(12 13) | m(45) |
| 1 -1 1 1 3 1 1 0 m(11 15) | m(37) |
| 1 1 -1 1 3 1 1 0 m(13 15) | m(311) |
| | m(57) |
| QM operation | m(513) |
| ABCDguf | m(911) |
| | m(913) |
| -1 1 0 -1 1 0 1 0 m(4 5 12 13) | |
| 1 -1 -1 1 2 0 1 0 m(9 11 13 15) | Prime Implicants Chart |
| | 0 3 4 5 7 9 11 13 |
| List of Prime Implicants | |
| m(26) | 1 0 2 0 0 0 0 0 |
| m(210) | 0 0 2 3 0 0 0 0 |
| m(46) | 0 2 0 0 2 0 0 0 |
| m(10 11) | $0\ 2\ 0\ 0\ 0\ 2\ 0$ |
| m(451213) | 0 0 0 3 2 0 0 0 |
| m(9 11 13 15) | 0 0 0 3 0 0 0 2 |
| | 0 0 0 0 0 2 2 0 |
| Prime Implicants Chart | 0 0 0 0 0 2 0 2 |
| 2 4 5 6 9 10 11 12 13 15 | |
| | Essential Prime Implicants |
| 200200000 | A'C'D' |

```
2\; 0\; 0\; 0\; 0\; 2\; 0\; 0\; 0\; 0
0\ 2\ 0\ 2\ 0\ 0\ 0\ 0\ 0
0000022000
0\; 2\; 1\; 0\; 0\; 0\; 0\; 1\; 2\; 0
0000102021
----Essential Prime Implicants----
BC'
AD
F(a, b, c, d) = \sum m(0, 1, 2, 3, 5, 7, 8, 10, 12, 13, 15)
                                                              18, 20, 21, 26, 27, 30, 31)
----QM operation----
                                                              ----QM operation----
ABCDguf
                                                              ABCDEguf
00000310 \parallel m(0)
                                                              0\ 0\ 0\ 0\ 0\ 0\ 4\ 1\ |||\ m(\ 0\ )
0\ 0\ 0\ 1\ 1\ 3\ 1\ 0\ |||\ m(\ 1\ )
                                                              0\ 0\ 0\ 0\ 1\ 1\ 2\ 1\ |||\ m(\ 1\ )
00101310 \parallel m(2)
                                                              0\ 0\ 0\ 1\ 0\ 1\ 2\ 1\ |||\ m(\ 2\ )
00112310 \parallel m(3)
                                                              0\ 0\ 1\ 0\ 0\ 1\ 2\ 1\ |||\ m(\ 4\ )
0 1 0 1 2 3 1 0 ||| m(5)
                                                              0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ |||\ m(\ 8\ )
0 1 1 1 3 3 1 0 ||| m(7)
                                                              0\ 1\ 0\ 1\ 1\ 3\ 2\ 1\ |||\ m(\ 11\ )
10001310 \parallel m(8)
                                                              0 1 1 0 1 3 1 1 ||| m( 13 )
                                                              0 1 1 1 0 3 2 1 ||| m(14)
10102210 ||| m(10)
1 1 0 0 2 2 1 0 ||| m(12)
                                                              0 1 1 1 1 4 4 1 ||| m( 15 )
1 1 0 1 3 3 1 0 ||| m( 13 )
                                                              10001221 ||| m(17)
                                                              1 0 0 1 0 2 2 1 ||| m( 18 )
1 1 1 1 4 2 1 0 ||| m( 15 )
                                                              10100221 \parallel m(20)
----OM operation----
                                                              10101321 \parallel m(21)
ABCDguf
                                                              1\ 1\ 0\ 1\ 0\ 3\ 3\ 1\ |||\ m(\ 26\ )
                                                              1 1 0 1 1 4 3 1 ||| m( 27 )
0 0 0 -1 0 1 1 0 ||| m( 0 1 )
                                                              1\ 1\ 1\ 1\ 0\ 4\ 3\ 1\ |||\ m(\ 30\ )
0\ 0\ -1\ 0\ 0\ 2\ 1\ 0\ |||\ m(\ 0\ 2\ )
                                                              11111531 \parallel m(31)
-1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ |||\ m(\ 0\ 8\ )
0 0 -1 1 1 2 1 0 ||| m( 1 3 )
                                                              ----QM operation----
                                                              ABCDEguf
0 - 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \parallel m(15)
0\ 0\ 1\ -1\ 1\ 1\ 1\ 0\ |||\ m(\ 2\ 3\ )
-101011110 \parallel m(210)
                                                              0\ 0\ 0\ 0\ -1\ 0\ 0\ 1\ |||\ m(\ 0\ 1\ )|
10-101110 \parallel m(810)
                                                              0\ 0\ 0\ -1\ 0\ 0\ 0\ 1\ |||\ m(\ 0\ 2\ )
1 - 100101011 m(812)
                                                              0.0 - 1.0 0.0 0.1 \parallel m(.0.4)
0 - 1 1 1 2 1 1 0 \parallel m(37)
                                                              0 - 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ ||| \ m(\ 0 \ 8)
01-112210 ||| m(57)
                                                              -10001101 \parallel m(117)
-1 1 0 1 2 1 1 0 ||| m( 5 13 )
                                                              -10010101 | || m(218)
1 1 0 -1 2 0 1 0 ||| m( 12 13 )
                                                              -101001011111m(420)
-1 1 1 1 3 1 1 0 ||| m( 7 15 )
                                                              1 0 -1 0 1 2 0 1 ||| m( 17 21 )
1 1 -1 1 3 1 1 0 ||| m( 13 15 )
                                                              1 -1 0 1 0 2 0 1 ||| m( 18 26 )
                                                              1 0 1 0 -1 2 0 1 ||| m( 20 21 )
----QM operation----
                                                              0 1 -1 1 1 3 1 1 ||| m( 11 15 )
ABCDguf
                                                              -1 1 0 1 1 3 1 1 ||| m( 11 27 )
                                                              0 1 1 -1 1 3 0 1 ||| m( 13 15 )
0\ 0\ -1\ -1\ 0\ 0\ 1\ 0\ |||\ m(\ 0\ 1\ 2\ 3\ )
                                                              0 1 1 1 -1 3 1 1 ||| m( 14 15 )
-10-100010 \parallel m(02810)
                                                              -1 1 1 1 0 3 1 1 ||| m( 14 30 )
0 - 1 - 1 1 1 0 1 0 \parallel m(1357)
                                                              1 1 0 1 -1 3 1 1 ||| m( 26 27 )
-1 1 -1 1 2 0 1 0 ||| m( 5 7 13 15 )
                                                              1 1 -1 1 0 3 1 1 ||| m( 26 30 )
                                                              -1 1 1 1 1 4 2 1 ||| m( 15 31 )
                                                              1 1 -1 1 1 4 2 1 ||| m( 27 31 )
----List of Prime Implicants----
m(812)
                                                              1 1 1 1 -1 4 2 1 ||| m( 30 31 )
m(1213)
m(0123)
                                                              ----QM operation----
m(02810)
                                                              ABCDEguf
m(1357)
```

```
m(571315)
                                                          -1 1 -1 1 1 3 0 1 ||| m( 11 15 27 31 )
                                                          -1 1 1 1 -1 3 0 1 ||| m( 14 15 30 31 )
                                                          1 1 -1 1 -1 3 0 1 ||| m( 26 27 30 31 )
----Prime Implicants Chart----
0 1 2 3 5 7 8 10 12 13 15
                                                          ----List of Prime Implicants----
0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 2\ 0\ 0
                                                          m(01)
00000000220
                                                          m(02)
                                                          m(04)
22220000000
20200021000
                                                          m(08)
0\; 2\; 0\; 2\; 2\; 2\; 0\; 0\; 0\; 0\; 0
                                                          m(117)
00002200021
                                                          m(218)
                                                          m(420)
----Essential Prime Implicants----
                                                          m(1721)
                                                          m(1826)
BD
                                                          m(2021)
                                                          m(1315)
                                                          m(11 15 27 31)
                                                          m(14 15 30 31)
                                                          m(26273031)
                                                          ----Prime Implicants Chart----
                                                          0 1 2 4 8 11 13 14 15 17 18 20 21 26 27 30 31
                                                          4020000000000000000
                                                          4002000000000000000
                                                          4000100000000000000
                                                          0200000000000000000
                                                          002000000000000000
                                                          0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 0
                                                          0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 2\ 0\ 0\ 0\ 0
                                                          0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 2\ 0\ 0\ 0
                                                          0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 0\ 0\ 0\ 0
                                                          0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 3\ 0\ 0\ 0\ 0\ 0\ 0\ 0
                                                          0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 3\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 3
                                                          0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 3\ 0\ 0\ 0\ 0\ 0\ 2\ 3
                                                          0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 3
                                                          ----Essential Prime Implicants----
                                                          A'C'D'E'
                                                          A'BCE
                                                          BDE
                                                          BCD
F(a, b, c, d, e) = \sum m(1, 3, 4, 5, 9, 11, 12, 13, 18, 22, e)
                                                                      23, 26, 30, 31)
           22, 23, 27, 28, 29, 30)
----QM operation----
                                                          ----QM operation----
ABCDEguf
                                                          ABCDEguf
                                                          0 0 0 0 1 1 3 1 ||| m(1)
0\ 0\ 0\ 0\ 0\ 0\ 4\ 1\ |||\ m(\ 0\ )
0\ 0\ 0\ 0\ 1\ 1\ 3\ 1\ |||\ m(\ 1\ )
                                                          0\ 0\ 0\ 1\ 1\ 2\ 2\ 1\ |||\ m(\ 3\ )
0\ 0\ 0\ 1\ 0\ 1\ 3\ 1\ |||\ m(\ 2\ )
                                                          0\ 0\ 1\ 0\ 0\ 1\ 2\ 1\ |||\ m(\ 4\ )
                                                          0 0 1 0 1 2 3 1 ||| m(5)
0\ 0\ 0\ 1\ 1\ 2\ 4\ 1\ |||\ m(\ 3\ )
                                                          01001231 \parallel m(9)
0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ |||\ m(\ 4\ )
01000131 \parallel m(8)
                                                          0 1 0 1 1 3 2 1 ||| m( 11 )
0\ 1\ 0\ 0\ 1\ 2\ 3\ 1\ |||\ m(9)
                                                          0 1 1 0 0 2 2 1 ||| m( 12 )
0 1 0 1 0 2 3 1 ||| m( 10 )
                                                          0\ 1\ 1\ 0\ 1\ 3\ 3\ 1\ |||\ m(\ 13\ )
0\ 1\ 0\ 1\ 1\ 3\ 4\ 1\ |||\ m(\ 11\ )
                                                          10010221 \parallel m(18)
10011331 ||| m(19)
                                                          10110331 ||| m(22)
```

```
10101321 ||| m(21)
                                                              10111421 ||| m(23)
1 0 1 1 0 3 2 1 ||| m( 22 )
                                                              1 1 0 1 0 3 2 1 ||| m( 26 )
10111431 \parallel m(23)
                                                              1\ 1\ 1\ 1\ 0\ 4\ 3\ 1\ |||\ m(\ 30\ )
1 1 0 1 1 4 2 1 ||| m(27)
                                                              1 1 1 1 1 5 2 1 ||| m(31)
1 1 1 0 0 3 2 1 ||| m(28)
1 1 1 0 1 4 2 1 ||| m( 29 )
                                                              ----OM operation----
1 1 1 1 0 4 2 1 ||| m( 30 )
                                                              ABCDEguf
----QM operation----
                                                              0 0 0 -1 1 1 1 1 ||| m( 1 3 )
ABCDEguf
                                                              0.0 - 1.0 1 1 1 1 \parallel m(1.5)
                                                              0 - 1001121 \parallel m(19)
0\ 0\ 0\ 0\ -1\ 0\ 2\ 1\ |||\ m(\ 0\ 1\ )
                                                              0\ 0\ 1\ 0\ -1\ 1\ 1\ 1\ |||\ m(\ 4\ 5\ )
0\ 0\ 0\ -1\ 0\ 0\ 2\ 1\ |||\ m(\ 0\ 2\ )
                                                              0 - 1 1 0 0 1 1 1 \parallel m(4 12)
0.0 - 1.0 0.0 0.1 \parallel m(.0.4)
                                                              0 - 1011211 \parallel m(311)
0 - 1000021 \parallel m(08)
                                                              0 - 1 1 0 1 2 2 1 \parallel m(5 13)
0\ 0\ 0\ -1\ 1\ 1\ 2\ 1 \parallel m(1\ 3)
                                                              0 1 0 -1 1 2 1 1 ||| m( 9 11 )
0 - 1001121 \parallel m(19)
                                                              0 1 -1 0 1 2 1 1 ||| m( 9 13 )
0 0 0 1 -1 1 2 1 ||| m( 2 3 )
                                                              0 1 1 0 -1 2 1 1 ||| m( 12 13 )
                                                              10-110211||| m(1822)
0 - 1 \ 0 \ 1 \ 0 \ 1 \ 2 \ 1 \parallel m(2 \ 10)
0\ 1\ 0\ 0\ -1\ 1\ 2\ 1 \parallel m(\ 8\ 9\ )
                                                              1 -1 0 1 0 2 1 1 ||| m( 18 26 )
0 1 0 -1 0 1 2 1 ||| m( 8 10 )
                                                              1 0 1 1 -1 3 1 1 ||| m( 22 23 )
0 - 1 \ 0 \ 1 \ 1 \ 2 \ 3 \ 1 \parallel m(3 \ 11)
                                                              1 -1 1 1 0 3 2 1 ||| m( 22 30 )
-1 0 0 1 1 2 1 1 ||| m( 3 19 )
                                                              1 1 -1 1 0 3 1 1 ||| m( 26 30 )
0\ 1\ 0\ -1\ 1\ 2\ 2\ 1\ |||\ m(\ 9\ 11\ )|
                                                              1 -1 1 1 1 4 1 1 ||| m( 23 31 )
0 1 0 1 -1 2 2 1 ||| m( 10 11 )
                                                              1 1 1 1 -1 4 1 1 ||| m( 30 31 )
-1 1 0 1 1 3 1 1 ||| m( 11 27 )
10-111301 ||| m(1923)
                                                              ----OM operation----
1-1011311 ||| m(1927)
                                                              ABCDEguf
1 0 1 -1 1 3 0 1 ||| m( 21 23 )
1-1101301 ||| m(2129)
                                                              0 - 1 \ 0 - 1 \ 1 \ 1 \ 0 \ 1 \parallel m(1 \ 3 \ 9 \ 11)
1 0 1 1 -1 3 0 1 ||| m( 22 23 )
                                                              0 - 1 - 101101 \parallel m(15913)
1 -1 1 1 0 3 0 1 ||| m( 22 30 )
                                                              0 -1 1 0 -1 1 0 1 ||| m( 4 5 12 13 )
1 1 1 0 -1 3 0 1 ||| m( 28 29 )
                                                              1 -1 -1 1 0 2 0 1 ||| m( 18 22 26 30 )
1 1 1 -1 0 3 0 1 ||| m( 28 30 )
                                                              1 -1 1 1 -1 3 0 1 ||| m( 22 23 30 31 )
----QM operation----
                                                              ----List of Prime Implicants----
ABCDEguf
                                                              m(13911)
                                                              m(15913)
0 0 0 -1 -1 0 1 1 ||| m( 0 1 2 3 )
                                                              m(451213)
0 - 1 0 0 - 1 0 1 1 \parallel m(0 1 8 9)
                                                              m(18 22 26 30)
0 -1 0 -1 0 0 1 1 ||| m( 0 2 8 10 )
                                                              m(22 23 30 31)
0-10-11111 ||| m(13911)
0 - 1 \ 0 \ 1 - 1 \ 1 \ 1 \ 1 \ || \ m(2 \ 3 \ 10 \ 11)
                                                              ----Prime Implicants Chart----
                                                              1 3 4 5 9 11 12 13 18 22 23 26 30 31
0 1 0 -1 -1 1 1 1 ||| m(8 9 10 11)
-1 -1 0 1 1 2 0 1 ||| m( 3 11 19 27 )
                                                              210021000000000
                                                              2002200200000
----OM operation----
                                                              0\ 0\ 1\ 2\ 0\ 0\ 1\ 2\ 0\ 0\ 0\ 0\ 0
ABCDEguf
                                                              0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 2\ 0\ 1\ 2\ 0
                                                              0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 1\ 0\ 2\ 1
0 - 10 - 1 - 1001 \parallel m(0123891011)
----List of Prime Implicants----
                                                              ----Essential Prime Implicants----
m(04)
                                                              A'C'E
m(1923)
                                                              A'CD'
m(2123)
                                                              ADE'
m(2129)
                                                              ACD
m(22 23)
m(22 30)
m(2829)
```

```
m(2830)
m(3 11 19 27)
m(0123891011)
----Prime Implicants Chart----
0 1 2 3 4 8 9 10 11 19 21 22 23 27 28 29 30
200010000000000000
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 3\ 0\ 0\ 0\ 0
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 3\ 0\ 0\ 0\ 0
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 2\ 0
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 3\ 0\ 0\ 0\ 0
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 2
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 0
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 2
0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 2\ 2\ 0\ 0\ 0\ 1\ 0\ 0\ 0
211201112000000000
----Essential Prime Implicants----
A'B'D'E'
C'DE
A'C'
```

V. Code manual

<Input>

- Step 1: Enter number of variables. If input does not satisfy the condition, the system prints a warning and repeat Step 1.
- Step 2 : Enter number of min-terms. Don't-care terms are ignored. If input does not satisfy the condition, the system prints a warning and repeat Step 1.
 - Step 3: Enter list of min-terms

<Output>

- 1. Each stage of Quine-McCluskey algorithm.
- 2. List of prime implicants
- 3. Prime Implicant Chart
- 4. Essential Prime Implicants

VI. Full text of the code

Full text of the code is attached to the report for anti-plagiarism test.

/* ------"Computational Simulation of Ouine-McCluskey Algorithm using C/C++"

```
- Digital Logic Circuits, Yonsei university-
______
Date | Author
11/30/2022 | Taewan Ham
______
#include <iostream>
#include <math.h>
#include <vector>
#include <set>
#include <string>
#include <algorithm>
using namespace std;
//variables
int num_of_var = 0; // number of variables (4 or 5)
int num_of_min = 0; // number of min-terms
bool bool_tmp = true; // temporary variable for while loop in step 1
bool flag = true; // temporary variable for while loop in step 2
int min_index = 0; // index for vector minterm_list
int size_imp = 1; // size of implicants in tmp_table
int 1 = 0; // index number for qm_table and tmp_table
int u = 0; // used
int g = 0; // group
int f = 0; // f in init_var4,5_table
// arrays
// size of the column of each table is fixed to 8 to use as parameter in functions
int init_table[32][8];
int init_4var_table[16][8] = { // initial table of four variable minterms
 // A, B, C, D, g, u, f - g: group number, f: 0 or 1, u: used(1) not used(0)
         \{0, 0, 0, 0, 0, 0, 0, 0\}, //m0
         {0, 0, 0, 1, 1, 0, 0, 0}, //m1
         \{0, 0, 1, 0, 1, 0, 0, 0\}, //m2
         \{0, 0, 1, 1, 2, 0, 0, 0\}, //m3
         \{0, 1, 0, 0, 1, 0, 0, 0\}, //m4
         \{0, 1, 0, 1, 2, 0, 0, 0\}, //m5
         \{0, 1, 1, 0, 2, 0, 0, 0\}, //m6
         {0, 1, 1, 1, 3, 0, 0, 0}, //m7
         {1, 0, 0, 0, 1, 0, 0, 0}, //m8
         {1, 0, 0, 1, 2, 0, 0, 0}, //m9
{1, 0, 1, 0, 2, 0, 0, 0}, //m10
         \{1, 0, 1, 1, 3, 0, 0, 0\}, //m11
         \{1, 1, 0, 0, 2, 0, 0, 0\}, //m12
         \{1, 1, 0, 1, 3, 0, 0, 0\}, //m13
         \{1, 1, 1, 0, 3, 0, 0, 0\}, //m14
         \{1, 1, 1, 1, 4, 0, 0, 0\}, //m15
};
int init_5var_table[32][8] = { // initial table of five variable minterms
 // A, B, C, D, E, g, u, f - g: group number, f: 0 or 1, u: used(1) not used(0)
         {0, 0, 0, 0, 0, 0, 0, 0}, //m0
         \{0, 0, 0, 0, 1, 1, 0, 0\}, //m1
         \{0, 0, 0, 1, 0, 1, 0, 0\}, //m2
         \{0, 0, 0, 1, 1, 2, 0, 0\}, //m3
         \{0, 0, 1, 0, 0, 1, 0, 0\}, //m4
         \{0, 0, 1, 0, 1, 2, 0, 0\}, //m5
         \{0, 0, 1, 1, 0, 2, 0, 0\}, //m6
         \{0, 0, 1, 1, 1, 3, 0, 0\}, //m7
         \{0, 1, 0, 0, 0, 1, 0, 0\}, //m8
         \{0, 1, 0, 0, 1, 2, 0, 0\}, //m9
         {0, 1, 0, 1, 0, 2, 0, 0}, //m10
         \{0, 1, 0, 1, 1, 3, 0, 0\}, //m11
```

```
{0, 1, 1, 0, 0, 2, 0, 0}, //m12
         {0, 1, 1, 0, 1, 3, 0, 0}, //m13
         \{0, 1, 1, 1, 0, 3, 0, 0\},\
                                    //m14
                                    //m15
         \{0, 1, 1, 1, 1, 4, 0, 0\},\
         \{1, 0, 0, 0, 0, 1, 0, 0\}, //m16
         \{1, 0, 0, 0, 1, 2, 0, 0\}, //m17
         \{1, 0, 0, 1, 0, 2, 0, 0\}, //m18
         {1, 0, 0, 1, 1, 3, 0, 0}, //m19
         \{1, 0, 1, 0, 0, 2, 0, 0\}, //m20
         \{1, 0, 1, 0, 1, 3, 0, 0\}, //m21
         {1, 0, 1, 1, 0, 3, 0, 0},
         {1, 0, 1, 1, 1, 4, 0, 0},
                                    //m23
         \{1, 1, 0, 0, 0, 2, 0, 0\}, //m24
         \{1, 1, 0, 0, 1, 3, 0, 0\}, //m25
         \{1, 1, 0, 1, 0, 3, 0, 0\}, //m26
         \{1, 1, 0, 1, 1, 4, 0, 0\}, //m27
         {1, 1, 1, 0, 0, 3, 0, 0}, //m28
         \{1, 1, 1, 0, 1, 4, 0, 0\}, //m29
         {1, 1, 1, 1, 0, 4, 0, 0},
                                    //m30
         \{1, 1, 1, 1, 1, 5, 0, 0\},\
                                    //m31
};
int tmp_table[32][8]; // table used in step 2
int qm_table[32][8]; // table used in step 2 (quine-mccluskey table)
int num_list[32] = \{ \}; // number list (index : 0 \sim 31, initial value : 0)
// vectors
vector <int> vector_tmp; // temporary vector
vector <int> epi; // list of essential prime implicant
vector <vector <int>> minterm_list; // list of minterms used in step 2
vector <vector <int>>> tmp_minterm_list; // list of minterms used in step 2
vector <vector <int>> final_minterm_list; // final list of minterms (prime implicants)
vector <vector <int>> final_implicants; // final list of implicants (prime implicants)
// functions
bool compare_digits(int arr[][8], int n, int m, int var_num); // compare each two digits
number (n & m)
void initialize_table(int arr[][8]); // initialize elements of arr in to 0
void print_QM_op(int var_num); // prints column info of qm operation in step 2
string binary to literal(vector <int> binary, int var num); // changes binary notation in
to literal notation
int main(void) {
         // ---- Step 1 : getting input from user ----
         while (bool_tmp) { // getting number of variables
                  printf("Please enter number of variables : ");
                  scanf s("%d", &num of var); // input of number of variables (ex: 4, 5)
                  if (num_of_var == 4 || num_of_var == 5) bool_tmp = false;
                  else printf("Only 4 or 5 variables operation is available...\n"); //
repeat until it gets the correct input
         if (num_of_var == 4) { // if variable is 4
                  g = 4; u = 5; f = 6;
                  std::copy(&init_4var_table[0][0], &init_4var_table[0][0] + 16 * 8,
&init_table[0][0]);
         else { // if variable is 5
                  g = 5; u = 6; f = 7;
                  std::copy(&init_5var_table[0][0], &init_5var_table[0][0] + 32 * 8,
&init_table[0][0]);
         }
         bool_tmp = true;
         printf("\n");
```

```
while (bool_tmp) {
                   printf("Please enter number of min-terms : ");
                   scanf_s("%d", &num_of_min); // input of number of minterms (maximum:
2^(num_of_var)-1)
                   if (num of var == 4 && num of min < 16) bool tmp = 0;</pre>
                   else if (num_of_var == 5 && num_of_min < 32) bool_tmp = 0;</pre>
                   else printf("Please check your input again...\n"); // repeat until it
gets the correct input
         }
         printf("\n");
         printf("Please enter minterms : ");
         for (int i = 0; i < num_of_min; i++) { // getting minterms as input</pre>
                   int tmp = 0;
                   scanf_s("%d", &tmp);
                   init_table[tmp][f] = 1; // update initial table
         }
         // ---- Step 2 : Finding Prime implicants ----
         int cnt = 0;
         for (int i = 0; i < int(pow(2, num_of_var)); i++) { //initializing qm_table with
init_table
                   if (init_table[i][f] == 1) {
                            for (int j = 0; j < 8; j++) qm_table[1][j] = init_table[i][j];</pre>
                            minterm_list.push_back(std::vector <int>());
                            minterm_list[l].push_back(i);
                            1++;
                   }
         }
         while (flag) {
                   int group = 0;
                   int s = 0; // size of tmp_table
                   flag = false; // initialize flag
                   size_imp *= 2;
                   while (group < num_of_var) { // operation (qm_table -> tmp_table)
                            for (int i = 0; i < 1; i++) {</pre>
                                      if (qm_table[i][f] == 1 && qm_table[i][g] == group) {
                                               for (int j = 0; j < 1; j++) {
                                                         if (qm_table[j][f] == 1 &&
qm table[j][g] == group + 1) {
(compare_digits(qm_table, i, j, num_of_var)) { // if two number satisfies the condition
                                                                             qm_table[i][u]++;
// check 'used'
                                                                             qm table[i][u]++;
                                                                             bool fflag =
false:
//tmp_minterm_list.push_back(std::vector <int>());
                                                                            for (int n = 0; n
< minterm_list[i].size(); n++) vector_tmp.push_back(minterm_list[i][n]); // record minterms</pre>
                                                                            for (int n = 0; n
< minterm_list[j].size(); n++) vector_tmp.push_back(minterm_list[j][n]);</pre>
sort(vector_tmp.begin(), vector_tmp.end());
                                                                            for (int n = 0; n
< tmp minterm list.size(); n++) {</pre>
std::set<vector<int>> s;
s.insert(tmp_minterm_list[n]);
```

```
auto
rep = s.insert(vector_tmp);
                                                                                       if
(!rep.second) { // if it is repeated
         fflag = true;
         break;
                                                                                       }
                                                                              if (!fflag) { //
if it is not a repeated operation
tmp_minterm_list.push_back(vector_tmp);
sort(tmp_minterm_list[s].begin(), tmp_minterm_list[s].end());
                                                                                       for
(int k = 0; k < num_of_var; k++) {
         if (qm_table[i][k] == qm_table[j][k]) tmp_table[s][k] = qm_table[i][k];
         else tmp_table[s][k] = -1;
                                                                                       }
tmp_table[s][f] = 1;
tmp_table[s][g] = group;
                                                                                       s++;
                                                                                       flag =
true; // flag is false if there is no available combining
                                                                             }
vector_tmp.clear();
                                                                   }
                                                          }
                                                }
                                      }
                             }
                             group++;
                   for (int i = 0; i < 1; i++) { // find prime implicants
                             if (qm_table[i][f] == 1 && qm_table[i][u] == 0) {
                                      final_minterm_list.push_back(minterm_list[i]);
                                      for (int j = 0; j < num_of_var; j++) {</pre>
                                                vector_tmp.push_back(qm_table[i][j]);
                                      final_implicants.push_back(vector_tmp);
                             vector_tmp.clear();
                   }
                   // print each stage of operation
                   printf("\n----QM operation----\n");
                   print_QM_op(num_of_var);
                   for (int q = 0; q < 1; q++) {
                             for (int w = 0; w < 8; w++) printf("%d ", qm_table[q][w]);
printf("||| m( ");</pre>
                             for (int e = 0; e < minterm_list[q].size(); e++) printf("%d ",</pre>
minterm_list[q][e]);
                             printf(")\n");
                   }
```

```
// qm reset, qm = tmp, tmp reset
                 initialize_table(qm_table);
                 std::copy(\&tmp_table[0][0], \&tmp_table[0][0] + s * 8, \&qm_table[0][0]);
                 initialize_table(tmp_table);
                 // minterm list reset, min = tmp, tmp reset
                 minterm_list.assign(tmp_minterm_list.size(),
std::vector<int>(tmp_minterm_list.size()));
                 std::copy(tmp_minterm_list.begin(), tmp_minterm_list.end(),
minterm_list.begin());
                 tmp_minterm_list.assign(0, std::vector<int>(0));
                 // initialize index values
                 1 = s;
                 s = 0; // initializing size of qm table
        }
        // ---- Step 3: Prime Implicant Chart ----
        int pi num = final minterm list.size(); // number of rows of the prime implicant
chart
        std::vector <int> col_chart; // column of the prime implicant chart
        // printing prime implicants
        printf("\n----List of Prime Implicants----\n");
        for (int i = 0; i < pi_num; i++) {</pre>
                 printf("m( ");
                 num_list[final_minterm_list[i][j]]++; // count how many time
the min-term is included in prime implicants
                 printf(")\n");
        }
        // make a vector that contains min-term list
        for (int i = 0; i < 32; i++) {
                 if (num_list[i] != 0) {
                          col_chart.push_back(i);
        }
        int col size = col chart.size();
        std::vector< vector <int>>> pi_chart; // prime implicant chart
        // creating prime implicant chart (pi chart)
        for (int i = 0; i < pi num; i++) {</pre>
                 bool fflag = true; // flag for epi
                 for (int j = 0; j < col_size; j++) {</pre>
                          bool fflag_ = true; // flag for pushing back 0 in pi_chart
                          for (int k = 0; k < final_minterm_list[i].size(); k++) {</pre>
                                  if (final_minterm_list[i][k] == col_chart[j]) {
frequency of min-term
                                           if (num_list[col_chart[j]] == 1 && fflag ==
true) {
                                                    epi.push_back(i); // essential
prime implicant
                                                    fflag = false; // prevent repeat
of same epi value
                                           fflag = false;
                                  }
```

```
if(fflag_) vector_tmp.push_back(0); // push 0 if there is a
number in col_chart[] but not in final_minterm_list[][]
                   pi_chart.push_back(vector_tmp);
                   vector_tmp.clear();
         }
         printf("\n----Prime Implicants Chart----\n");
         for (int i = 0; i < col_size; i++) printf("%d ", col_chart[i]); // print column</pre>
info (which is list of min-terms
         printf("\n----\n");
         for (int i = 0; i < pi_num; i++) { // print prime implicant chart</pre>
                   for (int j = 0; j < col_size; j++) {</pre>
                            printf("%d ", pi_chart[i][j]);
                   printf("\n");
         }
         printf("\n----Essential Prime Implicants----\n");
         for (int i = 0; i < epi.size(); i++) { // print essential prime implicants</pre>
                   string str_tmp = binary_to_literal(final_implicants[epi[i]],
num_of_var);
                   for(int j = 0; j < str_tmp.size(); j++) printf("%c", str_tmp[j]); //</pre>
print literals
                   printf("\n");
         }
         return 0;
}
bool compare_digits(int arr[][8], int n, int m, int var_num) {
         int result = 0;
         for (int i = 0; i < var_num; i++) {</pre>
                   if (arr[n][i] != arr[m][i]) result++;
         if (result == 1) return true; // if n, m have only one digit in common return
true
         else return false;
}
void initialize table(int arr[][8]) {
         for (int i = 0; i < 32; i++) {</pre>
                   for (int j = 0; j < 8; j++) arr[i][j] = 0;</pre>
}
void print_QM_op(int var_num) {
         for (int i = 0; i < var_num; i++) printf("%c ", 65 + i);</pre>
         printf("g u f"); // group, used, f
         printf("\n----\n");
}
string binary_to_literal(vector <int> binary, int var_num) {
         char l_tmp; // literal
         char c_tmp; // asterisk
         string str_tmp; // output variable
for (int i = 0; i < var_num; i++) {</pre>
                   if (binary[i] == 1) {
                            1_tmp = char(i + 65); // change it to literal
                            str_tmp += 1_tmp;
                   else if (binary[i] == 0) {
```

VII. Screenshots of the result

```
-QM operation----
 ABCDguf
                            0 0 1 0 1 2 1 0 0 1 0 1 0 1 0 0 1 0 0 1 3 1 0
       0 0 1 3 1 0
0 1 2 2 1 0
              2 2 2 2
                      1 0
       0 1 2 2 1 0
1 0 2 2 1 0
    0 0
1
1
1
       1 1 3 3 1 0 |||
0 0 2 2 1 0 |||
0 1 3 4 1 0 |||
1 1 4 2 1 0 |||
    θ
    1 0
 11114210
     --QM operation-
 ABCDguf
0 -1 1 0 1 0 1 0 ||| m( 2 6 )

-1 0 1 0 1 0 1 0 ||| m( 2 10 )

0 1 0 -1 1 1 1 1 0 ||| m( 4 5 )

0 1 -1 0 1 0 1 0 ||| m( 4 5 )

-1 1 0 0 1 1 1 0 ||| m( 4 6 )

-1 1 0 0 1 1 1 0 ||| m( 5 13 )

1 0 -1 1 2 1 1 0 ||| m( 9 11 )

1 -1 0 1 2 1 1 0 ||| m( 9 13 )

1 0 1 -1 2 0 1 0 ||| m( 10 11

1 1 0 -1 2 1 1 0 ||| m( 12 13

1 -1 1 1 3 1 1 0 ||| m( 11 15

1 1 -1 1 3 1 1 0 ||| m( 13 15
                                           2 10 )
4 5 )
                                           4 12 )
5 13 )
9 11 )
                                            10 11 )
                                      m( 12 13 )
m( 11 15 )
m( 13 15 )
     --QM operation----
 ABCDguf
-1 1 0 -1 1 0 1 0 ||| m( 4 5 12 13 )
1 -1 -1 1 2 0 1 0 ||| m( 9 11 13 15 )
  ----List of Prime Implicants----
 m(26)
m(210)
 m(46)
 m( 10 11 )
 m( 4 5 12 13 )
 m( 9 11 13 15 )
  ----Prime Implicants Chart--
 2 4 5 6 9 10 11 12 13 15
        02000000
        0 0
                Θ
                    2 0 0 0 0
        020
                    θ
                        0 0
                                θ
        0 0
                    220
                Θ
                                θ
 0
        10000120
        0010202
        -Essential Prime Implicants-
 BC'
AD
```

Figure 1. (1)

```
---QM operation----
ABCDguf
                  ||| m( 0 )
||| m( 3 )
||| m( 4 )
||| m( 5 )
||| m( 7 )
||| m( 9 )
||| m( 11 )
||| m( 13 )
       Θ Θ
           11θ
     11
         2
            2
                θ
         1
            2
     θ
       Θ
                θ
            3
     θ
       1
         2
                θ
         3
    1
            2
              1 θ
       1 2
            2
              1θ
  0
    θ
    1 1 3 2 1
  Θ
                θ
    013210
----QM operation----
ABCDguf
  -1 0 0 0 0
               1 0 |||
                         m(
                             04)
                                  )
                    1 0 -1 1
             θ
                  θ
                             45
  -1 1
           2
             θ
                  θ
                         m(
                             3
   Θ
          2
             θ
                         m(
-1
      1
                  θ
                             3 11 )
               1 -1 1 2
                             5
             θ
                  θ
                         m(
                               7)
               1 2
-1 1 0
             θ
                  θ
                         m(
                             5
                               13)
                               11 )
13 )
                             9
1 0 -1 1 2
             θ
                         m(
                  θ
1 -1 0 1 2 0
               1
                  θ
                         m(
----List of Prime Implicants----
m( 0 4 )
m( 4 5 )
m( 3 7 )
m( 3 11 )
m( 5 7 )
  3 7 )
3 11 )
5 7 )
5 13 )
H ( )
m( 9 11 )
m( 9 13 )
    -Prime Implicants Chart--
0 3 4 5 7 9 11 13
   200000
  0230000
  2002000
Θ
  2000020
  0032000
θ
  θ
    Θ
      3 0 0 0 2
      00220
  0 0
           2 0 2
  0000
  --Essential Prime Implicants---
A'C'D'
```

Figure 2. (2)

| QM operation A B C D g u f | |
|---|--|
| 0 0 0 0 3 1 0 | QM operation |
| 0 0 0 1 1 3 1 0 m(1) 0 0 1 0 1 3 1 0 m(2) 0 0 1 1 2 3 1 0 m(3) 0 1 0 1 2 3 1 0 m(5) 0 1 1 1 3 3 1 0 m(7) 1 0 0 0 1 3 1 0 m(8) 1 0 1 0 2 2 1 0 m(10) 1 1 0 0 2 2 1 0 m(12) 1 1 0 1 3 3 1 0 m(13) 1 1 1 1 4 2 1 0 m(13) 1 1 1 1 4 2 1 0 m(15) QM operation | A B C D G u + |
| 0 0 1 0 1 3 1 0 | 0 0 0 0 0 3 1 0 m(0) |
| 0 0 1 1 2 3 1 0 | |
| 0 1 0 1 2 3 1 0 (5) 0 1 1 1 3 3 1 0 | |
| 0 1 1 1 3 3 1 0 m(7) 1 0 0 0 1 3 1 0 m(8) 1 0 1 0 2 2 1 0 m(10) 1 1 0 0 2 2 1 0 m(12) 1 1 0 1 3 3 1 0 m(13) 1 1 1 1 4 2 1 0 m(13) 1 1 1 1 4 2 1 0 m(15) QM operation A B C D g u f | |
| 1 0 0 0 1 3 1 0 m(8) 1 0 1 0 2 2 1 0 m(10) 1 1 0 0 2 2 1 0 m(12) 1 1 0 1 3 3 1 0 m(13) 1 1 1 1 4 2 1 0 m(15) QM operation A B C D g u f | 0 1 1 1 3 3 1 0 m(7) |
| 1 1 0 0 2 2 1 0 m(12) 1 1 0 1 3 3 1 0 m(13) 1 1 1 1 4 2 1 0 m(15) QM operation A B C D g u f | 10001310 m(8) |
| 1 1 0 1 3 3 1 0 m(13) 1 1 1 1 4 2 1 0 m(15) QM operation A B C D g u f | 1 0 1 0 2 2 1 0 m(10) |
| 1 1 1 1 4 2 1 0 m(15) QM operation A B C D g u f | 1 1 0 0 2 2 1 0 m(12) |
| QM operation A B C D g u f | 1 1 1 1 4 2 1 0 m(15) |
| A B C D g u f | |
| 0 0 0 -1 0 1 1 0 m(0 1) 0 0 -1 0 0 2 1 0 m(0 2) -1 0 0 0 0 1 1 0 m(0 8) 0 0 -1 1 1 2 1 0 m(1 3) 0 -1 0 1 1 1 1 0 m(1 5) 0 0 1 -1 1 1 1 0 m(2 3) -1 0 1 0 1 1 1 0 m(2 10) 1 0 -1 0 1 1 1 0 m(2 10) 1 0 -1 0 1 1 1 0 m(8 10) 1 -1 0 0 1 0 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(13 15) QM operation A B C D g u f | |
| 0 0 -1 0 0 2 1 0 | A B C D g u + |
| 0 0 -1 0 0 2 1 0 | |
| 0 0 -1 1 1 2 1 0 m(1 3) 0 -1 0 1 1 1 1 1 0 m(2 3) -1 0 1 0 1 1 1 1 0 m(2 10) 1 0 -1 0 1 1 1 1 0 m(8 10) 1 -1 0 0 1 0 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(8 12) 0 -1 1 1 2 2 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(13 15) QM operation A B C D g u f | 0 0 -1 0 0 2 1 0 m(0 2) |
| 0 -1 0 1 1 1 1 0 m(1 5) 0 0 1 -1 1 1 1 0 m(2 3) -1 0 1 0 1 1 1 0 m(2 10) 1 0 -1 0 1 1 1 0 m(8 10) 1 -1 0 0 1 0 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(3 7) 0 1 -1 1 2 2 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(13 15) | -1 0 0 0 0 1 1 0 m(0 8) |
| 0 0 1 -1 1 1 1 0 m(2 3) -1 0 1 0 1 1 1 0 m(2 10) 1 0 -1 0 1 1 1 0 m(8 10) 1 -1 0 0 1 0 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(13 15) QM operation A B C D g u f | 0 -1 0 1 1 1 1 0 m(1 3) |
| -1 0 1 0 1 1 1 0 m(2 10) 1 0 -1 0 1 1 1 0 m(8 10) 1 -1 0 0 1 0 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(3 7) 0 1 -1 1 2 2 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(13 15) QM operation A B C D g u f | 0 0 1 -1 1 1 1 0 m(2 3) |
| 1 -1 0 0 1 0 1 0 m(8 12) 0 -1 1 1 2 1 1 0 m(3 7) 0 1 -1 1 2 2 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(5 13) -1 1 1 1 3 1 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) QM operation A B C D g u f | -1 0 1 0 1 1 1 0 m(2 10) |
| 0 -1 1 1 2 1 1 0 m(3 7) 0 1 -1 1 2 2 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(13 15) QM operation | 10-101110 m(810) |
| 0 1 -1 1 2 2 1 0 m(5 7) -1 1 0 1 2 1 1 0 m(5 13) 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) QM operation A B C D g u f | 1-1001010 m(812) A-111211A m(87) |
| 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) QM operation A B C D g u f | 01-112210 =(57) |
| 1 1 0 -1 2 0 1 0 m(12 13) -1 1 1 1 3 1 1 0 m(7 15) 1 1 -1 1 3 1 1 0 m(7 15) QM operation A B C D g u f | -1 1 0 1 2 1 1 0 m(5 13) |
| 1 1 -1 1 3 1 1 0 m(13 15) QM operation A B C D g u f | 1 1 0 -1 2 0 1 0 m(12 13) |
| QM operation A B C D g u f | |
| A B C D g u f | |
| 0 0 -1 -1 0 0 1 0 m(0 1 2 3) -1 0 -1 0 0 0 1 0 m(0 2 8 10) 0 -1 -1 1 1 1 0 1 0 m(1 3 5 7) -1 1 -1 1 2 0 1 0 m(5 7 13 15) List of Prime Implicants m(8 12) m(12 13) m(0 1 2 3) m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 0 2 2 2 0 2 2 2 2 0 0 0 0 | |
| -1 0 -1 0 0 0 1 0 m(0 2 8 10) 0 -1 -1 1 1 0 1 0 m(1 3 5 7) -1 1 -1 1 2 0 1 0 m(5 7 13 15) List of Prime Implicants m(8 12) m(12 13) m(0 1 2 3) m(0 2 8 10) m(0 2 8 10) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | A B C D g u f |
| -1 0 -1 0 0 0 1 0 m(0 2 8 10) 0 -1 -1 1 1 0 1 0 m(1 3 5 7) -1 1 -1 1 2 0 1 0 m(5 7 13 15) List of Prime Implicants m(8 12) m(12 13) m(0 1 2 3) m(0 2 8 10) m(0 2 8 10) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | 0 0 -1 -1 0 0 1 0 m(0 1 2 3) |
| 0 -1 -1 1 1 0 1 0 m(1 3 5 7) -1 1 -1 1 2 0 1 0 m(5 7 13 15) List of Prime Implicants m(8 12) m(12 13) m(0 1 2 3) m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | |
| List of Prime Implicants m(8 12) m(12 13) m(0 1 2 3) m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 | 0-1-111010 m(1357) |
| m(8 12) m(12 13) m(0 1 2 3) m(0 1 2 3) m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | -1 1 -1 1 2 0 1 0 m(5 7 13 15) |
| m(8 12) m(12 13) m(0 1 2 3) m(0 1 2 3) m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | List of Prime Implicants |
| m(0 1 2 3) m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | n(8 12) |
| m(0 2 8 10) m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | |
| m(1 3 5 7) m(5 7 13 15) Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 | |
| Prime Implicants Chart 0 1 2 3 5 7 8 10 12 13 15 | n(1357) |
| 0 1 2 3 5 7 8 10 12 13 15 | H(5 7 13 15) |
| 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 2 2 0 2 2 2 2 | Prime Implicants Chart |
| 0 0 0 0 0 0 0 0 2 2 0 2 2 2 2 0 0 0 0 0 | 0 1 2 3 5 7 8 10 12 13 15 |
| 0 0 0 0 0 0 0 0 2 2 0 2 2 2 2 0 0 0 0 0 | 0 0 0 0 0 0 2 0 2 0 0 |
| 2 0 2 0 0 0 2 1 0 0 0 0 2 0 2 2 2 0 0 0 0 0 0 0 0 2 2 0 0 0 2 1 Essential Prime Implicants B'D' | 00000000220 |
| 0 2 0 2 2 2 0 0 0 0 0 0 0 0 0 2 2 0 0 0 2 1 Essential Prime Implicants B'D' | |
| 0 0 0 0 2 2 0 0 0 2 1 Essential Prime Implicants B'D' | |
| B'D' | |
| B'D' | Essential Drine Implicants |
| BD | |
| | BD |

Figure 3. (3)

```
-QM operation
CDEguf
                                  0 1 0 0 0 1 0 0 1 0 1 0 1
                  00010011100110011
                          00100101101001111
                                            01111333422233445
                                                     42221212422223333
                                                                                                    0
1
2
4
8
11
13
14
15
17
18
20
21
27
30
31
          00001111100001111
                                                             11111111111111111
                                                                                        000
                                                                                                                  -QM operation
CDEguf
                0
0
-1
0
0
                          0
-1
0
0
0
      0
0
0
-1
0
                                  -1
0
0
1
                                                00001
                                                         000000
                                                                                                                1
2
4
8
17
18
                                                                                           000012
                                                                                                                           )
     11-11361
111-1311
111-1311
11110311
101-1311
1-110311
1-111421
1-111421
                                                                                   13 15
14 15
14 39
26 27
26 39
15 31
27 31
30 31
                                                                        11111111
             -QM operatio
CDEguf
        1 -1 1 1 3 0 1 ||| m( 11 15 27 31
1 1 1 -1 3 0 1 ||| m( 14 15 30 31
1 -1 1 -1 3 0 1 ||| m( 26 27 30 31
        --List of Prime Implicants-
0 1 )
0 2 )
0 4 )
0 8 )
1 17 )
2 18 )
4 20 )
17 21 )
18 26 )
20 21 )
13 15 )
14 15 27 31 )
m( 11 15 27 31 )
m( 14 15 30 31 )
m( 26 27 30 31 )
```

Figure 4. (4)

```
operation
Eguf
                                      В
                                                                                                       D
                                                                                                                                                                        g
      000000001111111
                                        00000111100001111
                                                                       00001000001110111
                                                                                                       0011000111011001
                                                                                                                                         0101001011011010
                                                                                                                                                                        01121122333344344
                                                                                                                                                                                                        4334133343223222
                                                                                                                                                                                                                                            11111111111111111
                                                                                                                                                                                                                                                                             0
1
2
3
4
8
9
10
11
19
21
22
23
27
28
                                                                                                                                                                                                                                                                                                                                                                                                                             ,,,,,,
                                                                                                                                                                                                                                                                                                                                          00000000
                                                                       QM operation
CDEguf
                                      В
                                                                     ĊВ
                                   0 0 0 -1
0 0 -1 0
0 -1 0 0
-1 0 0
0 0 -1 1
-1 0 0 1
0 0 1 -1
-1 0 1 0
1 0 0 -1
1 0 -1
                                                                                                                                                                                                                          220222222
                                                                                                                                                                                                                                                                                                                                                                                                                                      1
2
4
8
3
9
3
10
9
      00000000000
                                                                                                                                                                                        000011111
                                                                                                                                                                                                                                                         1111111111
                                                                                                                                                                                                                                                                                                                                                          0000112288
                                                                                                                                                                                                                                                                                                                                                                                                                                                                               000000
                                                                                                                                                                                                                                                                                                                                                                                                                                                               *
                                                 A CONTRACTOR OF THE PARTY OF TH
                                                                            1 0
0 0
1 0
1 0
                                                                                                                                                                                                                                                                                                      10
11
19
11
27
23
27
23
29
23
29
30
                                                                                                                                                                                                                                                8
3
9
10
11
19
21
21
22
22
28
28
                          8 1 2 3 )
8 1 8 9 )
8 2 8 18 )
1 3 9 11 )
2 3 18 11 )
8 9 18 11 )
3 11 19 27
9 9 9 9 -1
                                                                                                                                                                                                                                                              0 0 1 ||| mC 0 1 2 3 8 9 10 11
                                        19 23 )
21 23 )
21 29 )
22 23 )
22 30 )
28 29 )
28 30 )
3 11 19
6 1 2 3
      THE THE TERM
```

Figure 5. (5)

```
D
                                          E
                                                    g
  0000000111111
             00001111000111
                     00110011011011
                                                              32233223232232
                                                                                                     1
3
4
5
9
11
12
13
18
22
23
30
31
                                                                                                                              000
                                          11011101001001
                                                    12122323234345
                                                                        11111111111111
                                                                                   100010011111
                                                                                                                                    ---QM operation
BCDEguf
                  θ
-1
θ
            θ
Θ
-1
                                -1
0
0
                                               1
1
1
                                                                                                           =(
  9 9
                                                         1
1
1
                                                                   1
1
2
                                                                             111
                                                                                                                                    3
5
9
                                                                                                                          111
        0 1 0 0

0 1 0 -1

-1 1 0 0

-1 0 1 1

-1 1 0 1

1 0 -1 1

1 1 0 -1

1 1 0 -1

0 -1 1 0

-1 0 1 0

0 1 1 -1

-1 1 1 0

-1 1 1 1

1 -1 1 1

1 1 1 1 1
                                                                                                        5 )
12
11
13
11
13
2 22
26
23
30
30
31
                                                                                                4
4
3
5
9
12
18
12
22
26
23
30
                                                                                     n(
n(
n(
n(
n(
n(
n(
n(
 9 9 9 9 9 1 1 1 1 1 1 1 1
                                                     1112111112111
                                                             111111111111111
                                              11222222233344
                                                                                                                      3
                                                                                                                          )
                 -QM operation
C D E g u f
                   0 -1 1
-1 0 1
1 0 -1
-1 1 0
1 1 -1
                                                                                                    1
4
18
22
                                                                                                             3 9 11 )
5 9 13 )
5 12 13 )
8 22 26 30
2 23 30 31
                                                  11123
  0 0 0 1 1
                                                        0
0
0
0
                                                                 11111
                                                                                         --List of Pri
1 3 9 11 )
1 5 9 13 )
4 5 12 13 )
18 22 26 30
22 23 30 31
4 5 12 13 )
18 22 26 30
22 23 30 31
                                                  Prime
                                                                         Implicants
Implicants Chart-
11 12 13 18 22 23
                                  2
9
9
                                          1
0
0
0
                                                           8
2
2
8
                                                                                                               8 8 8 1
                 0
1
0
0
                         0
2
2
0
0
                                                                   0
0
1
0
                                                                            8
8
2
2
                                                                                     0
0
0
1
                                                                                             0
0
1
0
                                                  0
0
1
0
0
                                                                                                      8822
                 Essenstial Prime Implicants
ACD
A'CD'
A'C'E
```

Figure 6. (6)