

Quine McCluskey Method

Boolean function simplification is one of the basics of Digital Electronics. The quine-McCluskey method also called the **tabulation method** is a very useful and convenient method for simplification of the Boolean functions for a large number of variables (greater than 4). This method is useful over [K-map](#) when the number of variables is larger for which K-map formation is difficult. This method uses prime implicants for simplification.

In this method, we construct multiple tables according to the question and at the last, we make a prime implicant table which is used to obtain essential prime implicants which are present in the simplified boolean expression. This method requires prior knowledge of decimal to binary representation and the basics of boolean algebra. It is a suitable method for a large number of input variables which can be easily solved by this method but the computation complexity is high. Majorly, this method includes the use of minterms, and prime implicants and obtains essential prime implicants which are further used in the simplified boolean functions.

Quine McCluskey Method (QMC):

- Quine McCluskey method also known as the tabulation method is used to minimize the [Boolean functions](#).
- It simplifies boolean expression into the simplified form using prime implicants.
- This method is convenient to simplify boolean expressions with more than 4 input variables.
- It uses an automatic simplification routine.

Terminologies:

[Implicant](#): Implicant is defined as a group of 1's(for minterm).

Prime implicant: It is the largest possible group of 1's(for minterm).

Essential Prime implicant: Essential prime implicants are groups that cover at least one minterm which cannot be covered by other applicants.

Note: This method uses decimal to binary representation.

Steps for Quine McCluskey Method:

1. Arrange the given minterms according to the number of ones present in their binary representation in ascending order.
2. Take the minterms from the continuous group if there is only a one-bit change to make their pair.
3. Place the '-' symbol where there is a bit change accordingly and keep the remaining bits the same.

4. Repeat steps 2 to 3 until we get all prime implicants (when all the bits present in the table are different).
5. Make a prime implicant table that consists of the prime implicants (obtained minterms) as rows and the given minterms (given in problem) as columns.
6. Place '1' in the minterms (cell) which are covered by each prime implicant.
7. Observe the table, if the minterm is covered by only one prime implicant then it is an essential to prime implicant.
8. Add the essential prime implicants to the simplified boolean function.

Example: Simplify using tabulation method : $F(A,B,C,D) = \sum m(0,1,2,4,6,8,9,11,13,15)$

Solution: Convert the given minterms into their binary representation and arrange them according to the number of ones present in the binary representation.

TABLE 1					
Group	Minterm	A	B	C	D
0	0	0	0	0	0
1	1	0	0	0	1
	2	0	0	1	0
	4	0	1	0	0
	8	1	0	0	0
2	6	0	1	1	0
	9	1	0	0	1
3	11	1	0	1	1
	13	1	1	0	1
4	15	1	1	1	1

As 0 has no 1 in its representation it is kept in one group(0). Similarly, 1, 2, 4, and 8 contain one 1 in their representation so it is kept in the next

group(1). 6 and 9 in the next group(2), 11, and 13 in the next group(3), 15 in the last group(4).

Now, for table-2 take minterms from successive groups(simultaneous group only) which have an only a 1-bit difference in their representation and form their pair by merging them and making a group of the pairs which are from the same groups that are merged (for example 0 is from group 0 and 1 is from group 1 so it is added to the group 0. 0 belongs to group 0 in table 1 and 2 belongs to group 1 in table 1 so its kept in the same group in table 2. Similarly, make all the possible pairs with the help of the above table and mark – where there is a bit difference.

TABLE-2					
Group	Pair	A	B	C	D
0	(0,1)	0	0	0	–
	(0,2)	0	0	–	0
	(0,4)	0	–	0	0
	(0,8)	–	0	0	0
1	(1,9)	–	0	0	1
	(2,6)	0	–	1	0
	(4,6)	0	1	–	0
	(8,9)	1	0	0	–
2	(9,11)	1	0	–	1
	(9,13)	1	–	0	1
3	(11,15)	1	–	1	1
	(13,15)	1	1	–	1

For table 3 repeat the same step by taking pairs of successive groups merging them where there is only a 1-bit difference and keeping them in groups according to the groups from where they are merged and placed – in bit difference.

TABLE-3					
Group	Quad	A	B	C	D
0	(0,1,8,9)	–	0	0	–
	(0,2,4,6)	0	–	–	0
1	(9,11,13,15)	1	–	–	1

After table 3 the process is stopped as there is no 1-bit difference in the remaining group minterms in the simultaneous groups of table 3.

Now, the remaining quads present in table 3 represent the prime implicants for the given boolean function. So, we construct prime implicants table which contains the obtained prime implicants as rows and the given minterms as columns. Place 1 in the corresponding place which the minterm can represent. Add the minterm to the simplified boolean expression if the given minterm is only covered by this prime implicant.

PRIME IMPLICANT TABLE											
Minterms → Prime Implicants ↓	0	1	2	4	6	8	9	11	13	15	
$B'C'$ (0,1,8,9)		1	1				1	1			
$A'D'$ (0,2,4,6)			1		1	1	1				
AD (9,11,13,15)							1	1	1	1	
Simplified Boolean function = $B'C' + A'D' + AD$											

$B'C'$ is in simplified function as minterm 1 is only covered by $B'C'$. Similarly, minterms 2,4,6 are only covered by $A'D'$ and minterms 11,13,15 are only covered by AD .

Example: Simplify using tabulation method : $F(A,B,C,D,E,F,G) = \sum m(20,28,52,60)$

Solution: Convert the given minterms in their binary representation and arrange them according to number of one's present in the binary representation.

TABLE-1								
Group	Minterms	A	B	C	D	E	F	G
0	20	0	0	1	0	1	0	0
1	28	0	0	1	1	1	0	0
	52	0	1	1	0	1	0	0
2	60	0	1	1	1	1	0	0

As 20 has 2 1s in its representation it is kept in one group(0). Similarly, 28 and 52 contain 3 1s in their representation so it is kept in the next group(1). 60 in the next group(2).

Now, for table-2 take minterms from successive groups(simultaneous group only) which have an only a 1-bit difference in their representation and form their pair by merging them and making a group of the pairs which are from the same groups that are merged (for example 20 is from group 0 and 28 is from group 1 so it is added to the group 0. 20 belongs to group 0 in table 1 and 52 belongs to group 1 in table 1 so its kept in the same group in table 2. Similarly, make all the possible pairs with the help of the above table and mark – where it is a bit different.

TABLE-2								
Group	Pair	A	B	C	D	E	F	G
0	(20,28)	0	0	1	–	1	0	0
	(20,52)	0	–	1	0	1	0	0
1	(28,60)	0	–	1	1	1	0	0

TABLE-2								
Group	Pair	A	B	C	D	E	F	G
	(52,60)	0	1	1	–	1	0	0

For table 3 repeat the same step by taking pairs of successive groups merging them where there is only a 1-bit difference and keeping them in groups according to the groups from where they are merged and placed – in bit difference.

TABLE-3								
Group	Quad	A	B	C	D	E	F	G
0	(20,28,52,60)	0	–	1	–	1	0	0

After table 3 the process is stopped as there is no 1-bit difference in the remaining group minterms in the simultaneous groups of table 3.

Now, the remaining quads present in table 3 represent the prime implicants for the given boolean function. So, we construct prime implicants table which contains the obtained prime implicants as rows and the given minterms as columns. Place 1 in the corresponding place which the minterm can represent. Add the minterm to the simplified boolean expression if the given minterm is only covered by this prime implicant.

$A'CEF'G'$ is obtained from table 3 as A, F, G contains 0 so $A'F'G'$, C, and E contain 1 so CE.

Prime Implicants Table				
Minterms →				
Prime Implicants ↓	20	28	52	60
$A'CEF'G'(20,28,52,60)$	1	1	1	1

Prime Implicants Table				
Minterms →				
Prime Implicants ↓	20	28	52	60
Simplified Boolean Function = $A'CEF'G'$				

$A'CEF'G'$ is in simplified function as it is the only prime implicant that covers all minterms.

Advantages of Quine McCluskey Method:

- This method is suitable for a large number of inputs ($n > 4$) for which K-map construction is a tedious task.
- It does not require pattern recognition.

Disadvantages of Quine McCluskey Method:

- The computational complexity of this method is high.