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, 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 for each value. Instead of being organized into columns and rows like a truth table, the Karnaugh map is an array of cells in which each cell represents a binary value of the input variables.

Developing Karnaugh Maps

3-Variable K-Map (Karnaugh Map)

A B \ C	0	1
0 0		
0 1		
1 1		
1 0		

K-Map without Cell Address

A B \ C	0	1
0 0	0	1
0 1	2	3
1 1	6	7
1 0	4	5

K-Map with Cell Address

4-Variable K-Map (Karnaugh Map)

A B \ C D	0 0	0 1	1 1	1 0
0 0				
0 1				
1 1				
1 0				

K-Map without Cell Address

A B \ C D	0 0	0 1	1 1	1 0
0 0	0	1	3	2
0 1	4	5	7	6
1 1	12	13	15	14
1 0	8	9	11	10

K-Map with Cell Address

The numbers in the blue boxes are known as CELL ADDRESS numbers. They are the decimal of the binary numbers of the same cell. They help us identify the position of each cell very easily

Filling up Karnaugh Maps

Karnaugh maps can be filled up from:

1. A given Boolean expression (SOP/POS), by creating a truthtable from it. (You must standardize the expression if it is not standard).
2. A given truth-table.
3. Boolean expressions, written in numbers

Filling up Karnaugh Maps

Filling up a 3 Variable K-map from a given truth-table

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Step-1: Draw K-map with Cell address

A B \ C	0	1
0 0	0	1
0 1	2	3
1 1	6	7
1 0	4	5

Step-2: Fill K-map with Y Values from Truth-table

A B \ C	0	1
0 0	0 0	1 1
0 1	2 0	3 0
1 1	6 1	7 1
1 0	4 1	5 1

Filling up Karnaugh Maps

Representing Boolean expressions in numbers

	A	B	C	Y
	0	0	0	0
1	0	0	1	1
	0	1	0	0
	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$F(A,B,C)=\Sigma(1,4,5,6,7)$

$Y=A'.B'.C+A.B'.C'+A.B'.C+A.B.C'+A.B.C$

OR

$F(A,B,C)=\Pi(0,2,3)$

$Y=(A+B+C).(A+B'+C).(A+B'+C')$

Filling up Karnaugh Maps

Filling up a 3 Variable K-map from Boolean expressions written in numbers

F(A,B,C)=Σ(0,4,5,6)

Step-1: Draw K-map with Cell address

A B \ C	0	1
0 0	0	1
0 1	2	3
1 1	6	7
1 0	4	5

Step-2: Fill K-map with Y Values from Numbered expression

A B \ C	0	1
0 0	0 1	1 0
0 1	2 0	3 0
1 1	6 1	7 0
1 0	4 1	5 1

Filling up Karnaugh Maps

Filling up a 4 Variable K-map from Truth-table:

DEC	A	B	C	D	Y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

A B \ C D	0 0	0 1	1 1	1 0
0 0	0 1	1 1	3 1	2 1
0 1	4 0	5 0	7 1	6 1
1 1	12 0	13 0	15 1	14 0
1 0	8 0	9 1	11 1	10 0

The Boolean expression representation of the truthtable in numbers can be written as follows

$F(A,B,C)=\Sigma(0,1,2,3,6,7,9,11,15)$

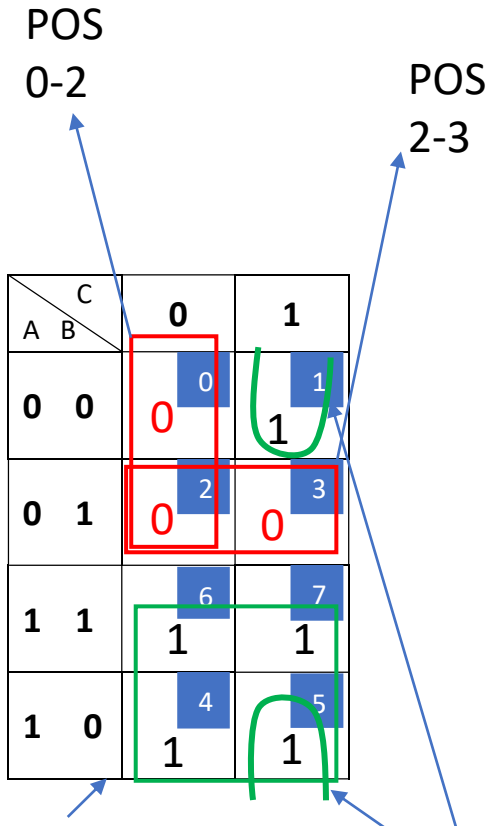
OR

$F(A,B,C)=\Pi(4,5,8,10,12,13,14)$

Rules for Making groups in K-MAP

1. Groups are made either for ‘0’s only or for ‘1’s only.
    a. Groups of ‘1’s give SOP expressions.
    b. Groups of ‘0’s give POS expressions.
2. The size of the groups should be as big as possible. For 4 variable K-MAP group of 16 should be made if possible, else group of 8 if possible, else group of 4 if possible, else group of 2 if possible, else only the ungrouped expression is written.
3. Group should be rectangular or square look alike shaped.
4. Groups can be formed only in vertical or horizontal direction. Diagonal groups are not allowed.
5. A value which is already grouped can be included in making another group, only if the size of that new group becomes large after including this value.
6. Make the total number of groups as less as possible.

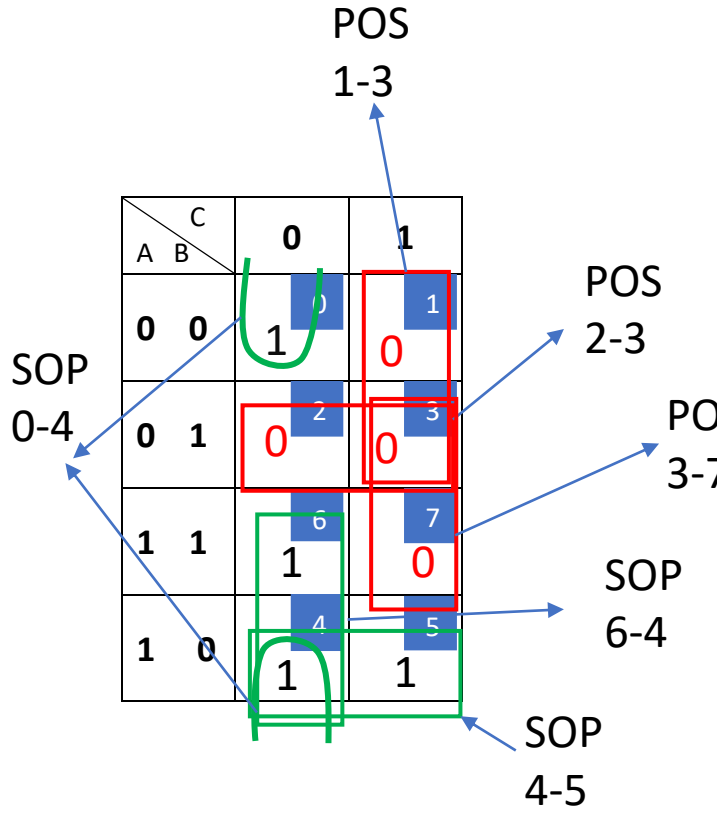
Solving K-MAP using detailed expressions:



ABC'+ABC+AB'C'+AB'C
AB(C'+C)+AB'(C'+C)
AB+AB'=A(B+B')=A

A'B'C+AB'C
B'C(A'+A)
B'C

SOP EXPRESSION
Y=A+B'C



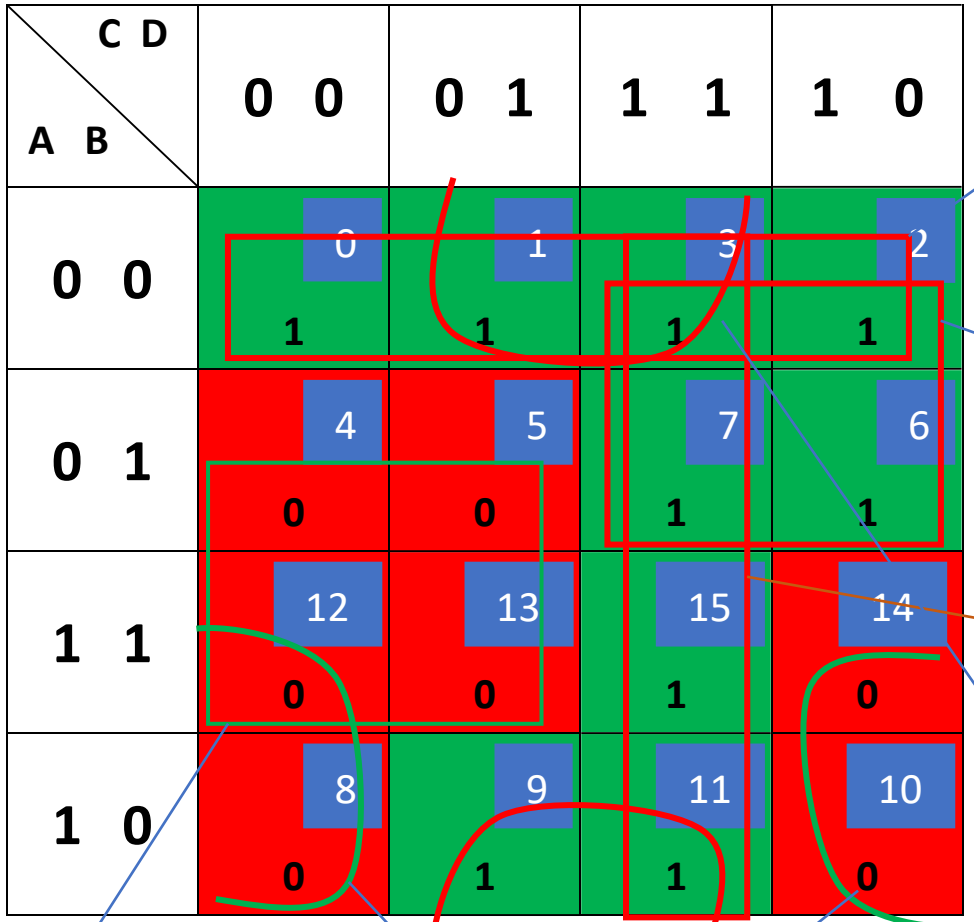
A'B'C'+AB'C'
B'C'(A'+A)
B'C'

ABC'+AB'C'
AC'(B+B')
AC'

AB'C'+AB'C
AB'(C'+C)
AB'

SOP EXPRESSION
Y=AC'+B'C'+AB'

Solving K-MAP using detailed expressions:



0-1-3-2 (SOP)

A'.B'.C'.D'+A'B'C'D+A'B'CD'+A'B'CD

A'B'C'(D'+D)+A'B'C(D'+D)

A'B'C'+A'B'C=A'B'(C+C')=A'B'

3-2-7-6 (SOP)

A'B'CD+A'B'CD'+A'BCD+A'BCD'

A'B'C(D'+D)+A'BC(D'+D)

A'B'C+A'BC= A'C(B'+B)=A'C

SOP EXPRESSION

Y=A'B'+A'C+CD+B'D

3-7-15-11 (SOP)

=A'B'CD+A'BCD+ ABCD+ AB'CD

=A'CD(B'+B)+ACD(B+B')

=A'CD+ACD=CD(A'+A)

=CD

1-3-9-11 (SOP)

=A'B'C'D+A'B'CD+ AB'C'D+ AB'CD

=A'B'D(C'+C)+AB'D(C'+C)

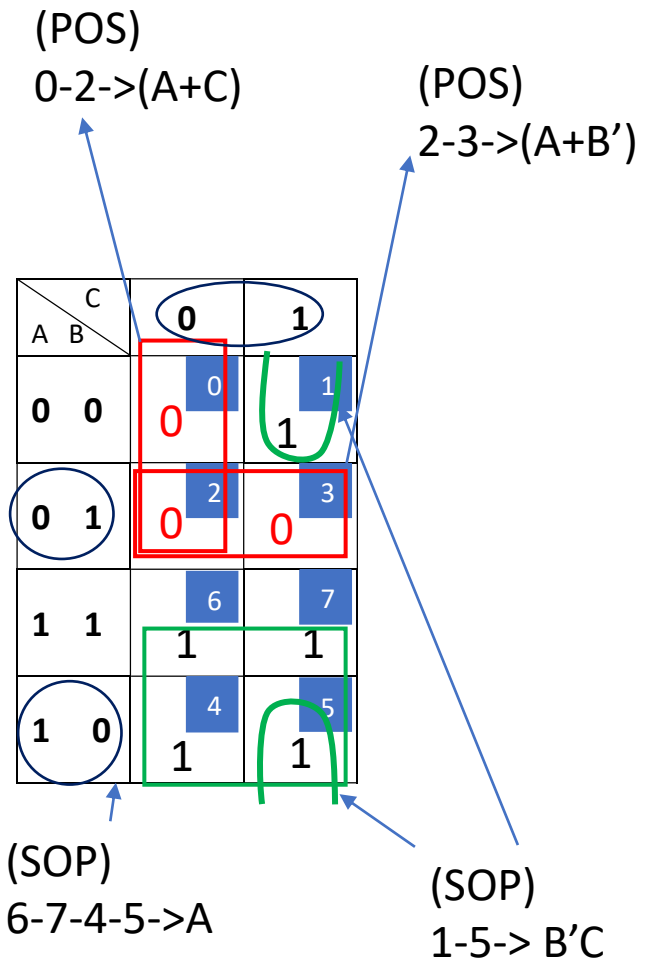
= A'B'D + AB'D =B'D(A'+A)

=B'D

4-5-12-13 (POS)

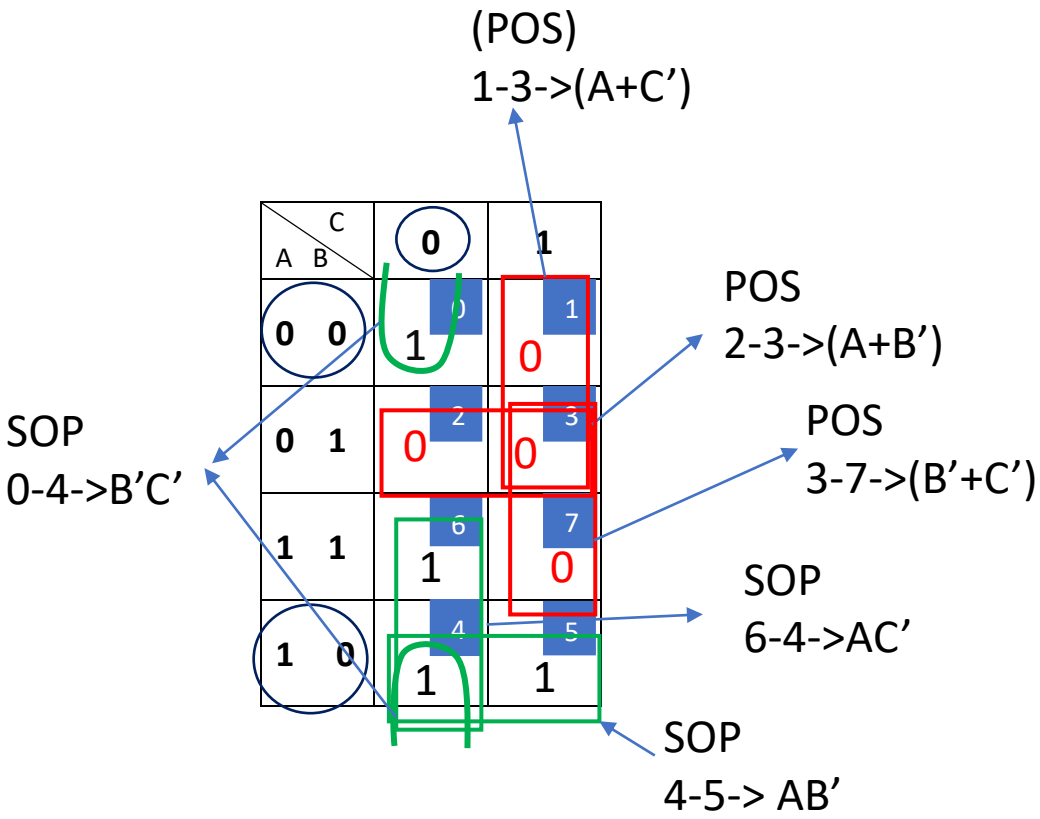
12-8-14-10 (POS)

Solving K-MAP using shorthand method:



SOP EXPRESSION
Y=A+B'C

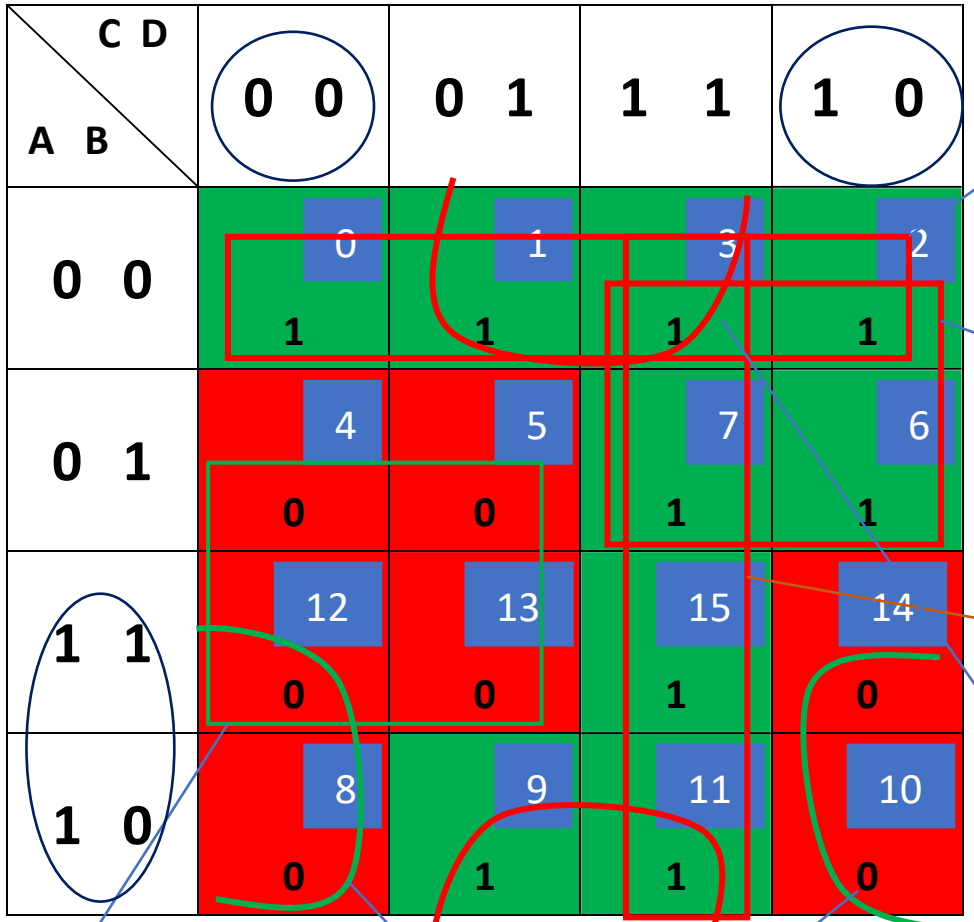
POS EXPRESSION
(A+C).(A+B')



SOP EXPRESSION
Y=AB'+B'C'+AC'

POS EXPRESSION
(A+C')(A+B')(B'+C')

Solving K-MAP using shorthand method:



0-1-3-2 (SOP)->A'B'

3-2-7-6 (SOP)->A'C

3-7-15-11 (SOP)->CD

1-3-9-11 (SOP)->B'D

SOP EXPRESSION

Y=A'B'+A'C+CD+B'D

4-5-12-13 (POS)
->(B'+C)

12-8-14-10 (POS)
->(A'+D)

POS EXPRESSION

(A'+D) (B'+C)

K-Map with don’t care condition:

For the following truth table, develop a k-Map and find the simplified SOP and simplified POS expression

I) WITHOUT USING DON’T CARE II) USING DON’T CARE CONDITION

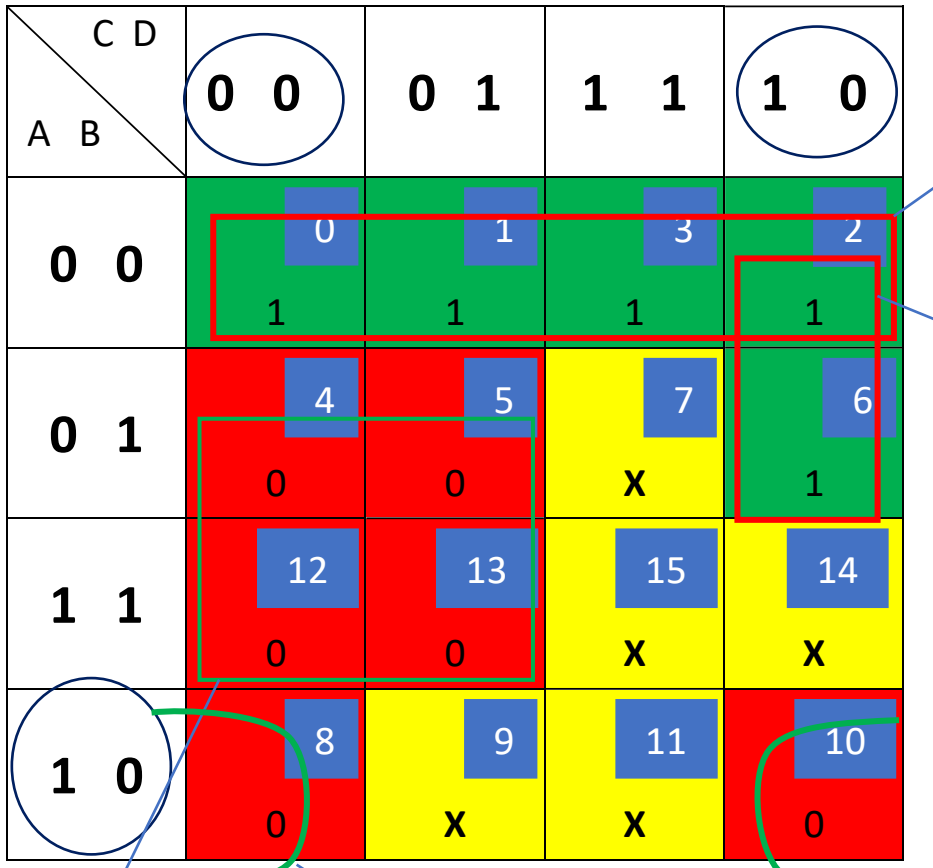
DEC	A	B	C	D	Y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	X
8	1	0	0	0	0
9	1	0	0	1	X
10	1	0	1	0	0
11	1	0	1	1	X
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	X
15	1	1	1	1	X

A B \ C D	0 0	0 1	1 1	1 0
0 0	0 1	1 1	3 1	2 1
0 1	4 0	5 0	7 X	6 1
1 1	12 0	13 0	15 X	14 X
1 0	8 0	9 X	11 X	10 0

Don’t cares are values, that are not useful in regular use of circuit.
Only if a don’t care value can increase the size of a group, whether it is SOP/POS, that don’t care is included in any SOP or POS group and behave like a ‘1’ or ‘0’, respectively.

X = 1/0 (but we don’t care about it)

I) WITHOUT USING DON'T CARE



0-1-3-2 (SOP)->A'B'

2-6 (SOP)->A'CD

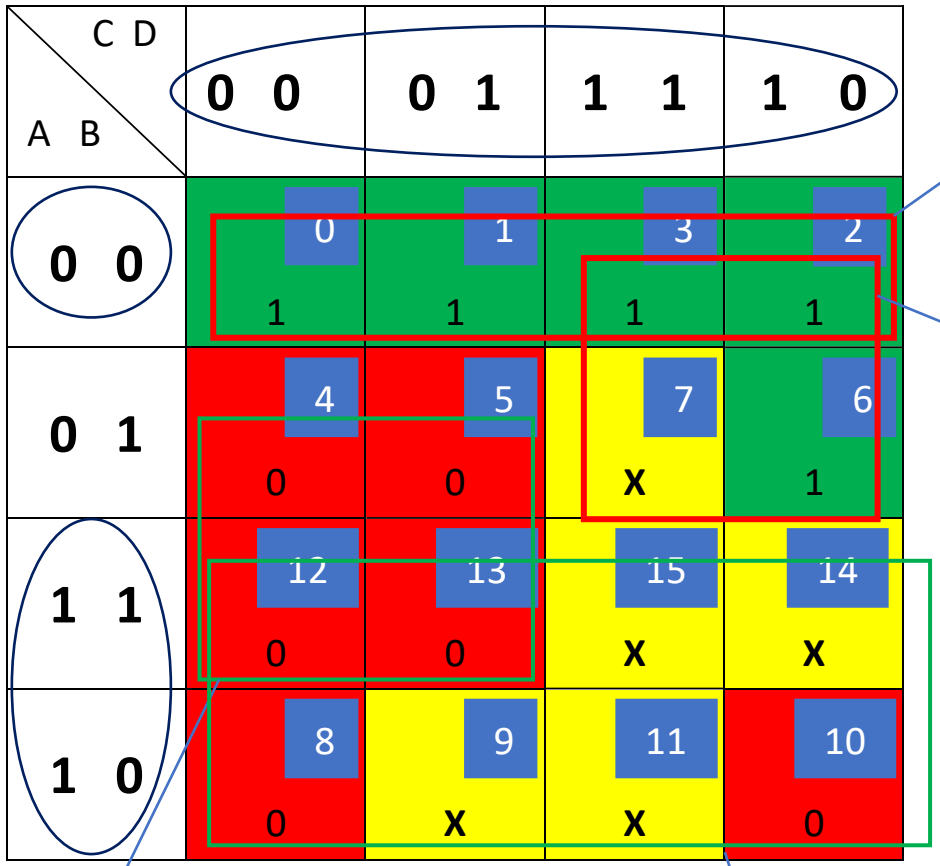
SOP EXPRESSION

Y=A'B'+A'CD

4-5-12-13 (POS)
->(B'+C)

8-10 (POS)
->(A'+B+D)

II) USING DON'T CARE



0-1-3-2 (SOP)->A'B'

3-2-7-6 (SOP)->A'C

SOP EXPRESSION

Y=A'B'+A'C

POS EXPRESSION

Y=A'(B'+C)

4-5-12-13 (POS)
->(B'+C)

14-15-12-13-8-9-11-10 (POS)
->A'

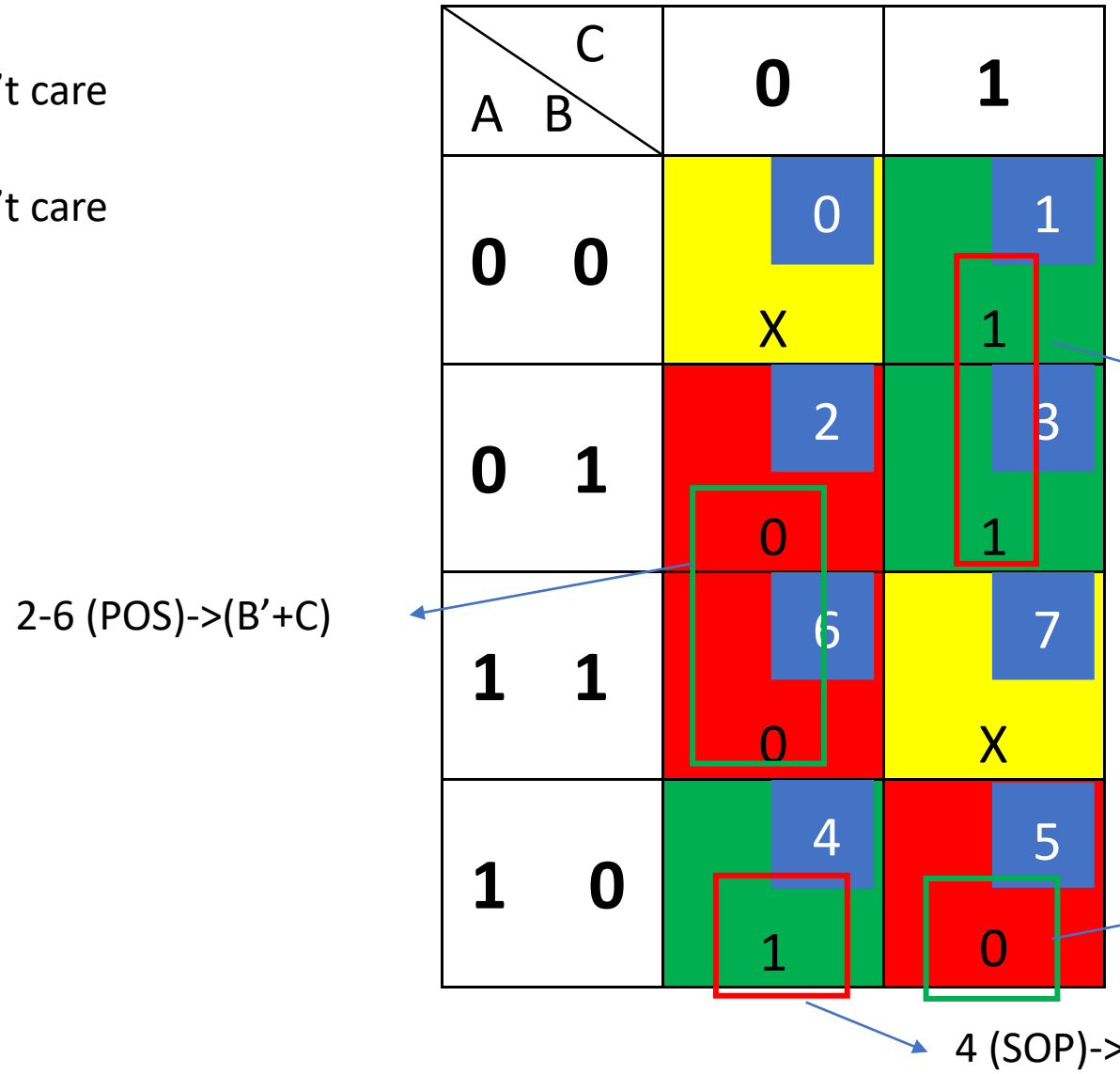
Classroom Task

f(A,B,C)=Σ(1,3,4) & d(A,B,C)=(0,7)

Find the most simplified

i) SOP without using don't care
ii) SOP using don't care
iii) POS without using don't care
iv) POS using don't care

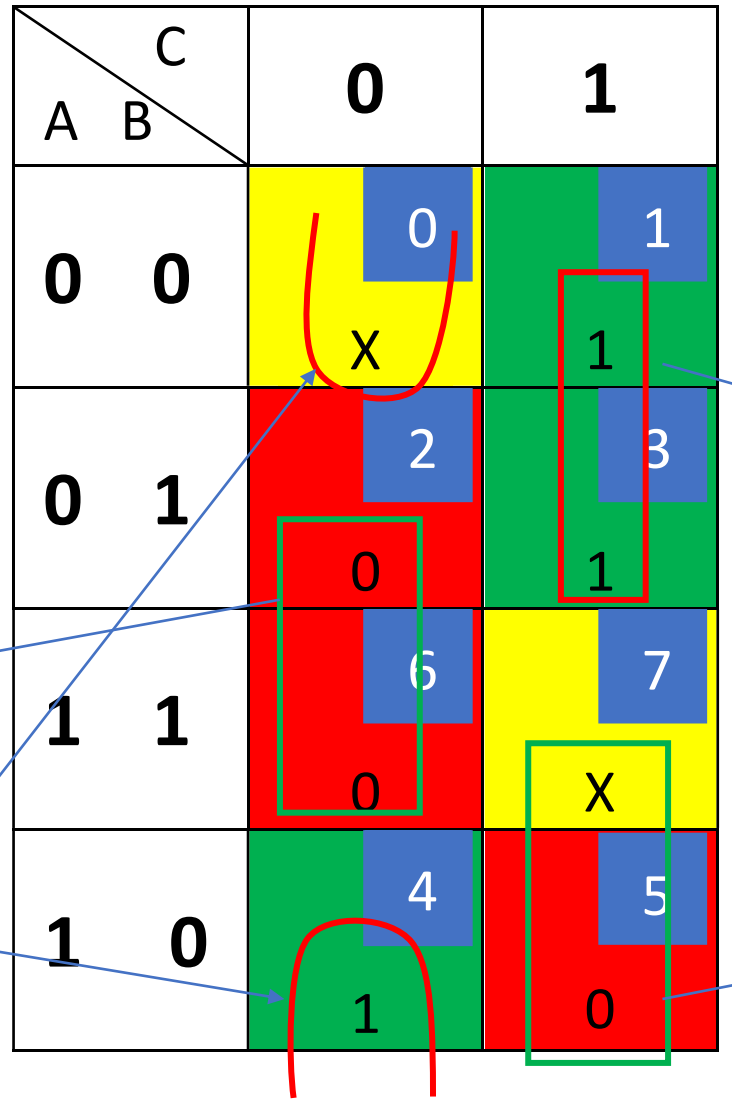
WITHOUT USING DON'T CARE CONDITION



SOP EXPRESSION
Y=A'C+AB'C'

POS EXPRESSION
Y=(A'+B+C')(B'+C)

USING DON'T CARE CONDITION



SOP EXPRESSION

Y=A'C+B'C'

1-3 (SOP)->A'C

2-6 (POS)->(B'+C)

POS EXPRESSION

Y=(A'+C)(B'+C)

7-5 (POS)->(A'+C')

SOP

0-4-> B'C'