CSE 2105
Digital Logic Design

MINIMIZATION TECHNIQUE

Simplification of Boolean function

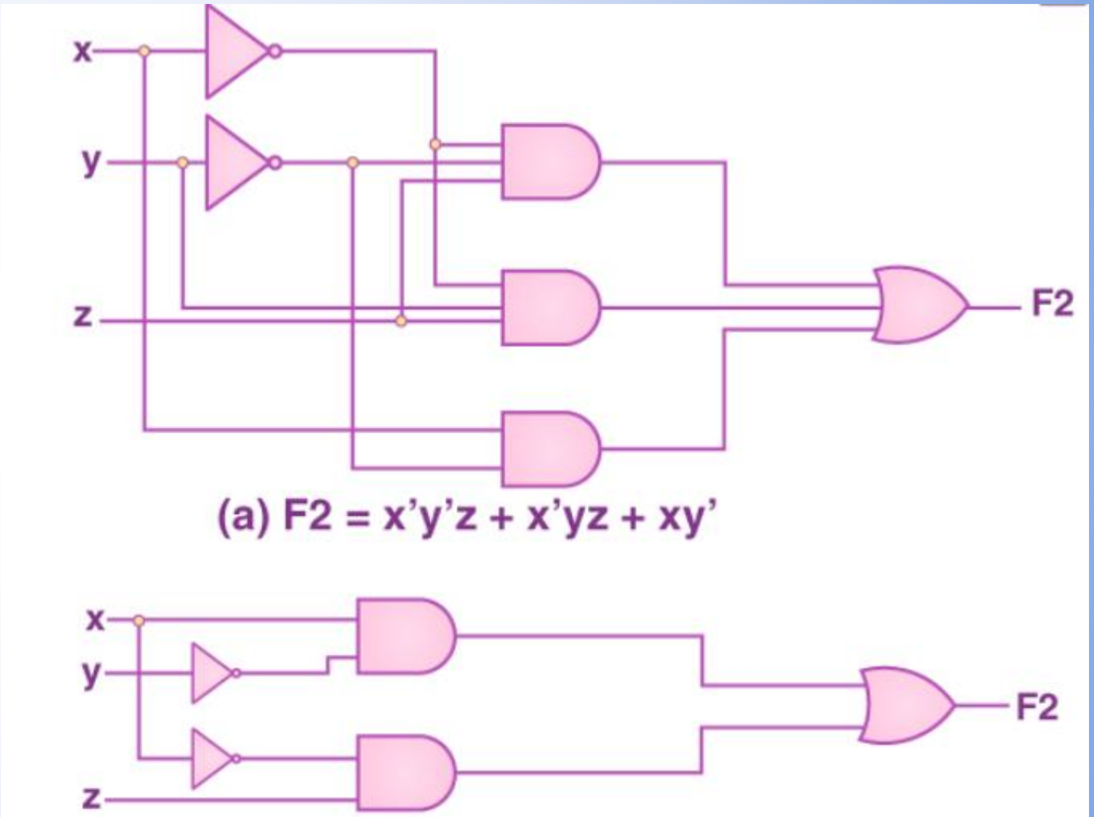
- Boolean algebra deals with binary variables and logic operation.
- A Boolean Function is described by an algebraic expression called Boolean expression which consists of binary variables, and the logic operation symbols.
- We can express every Boolean function in the form of a product of maxterms or a sum of minterms. But, the total number of literals present in such expressions is usually pretty high. So, it is preferable that we simplify the given algebraic expression to its most simplified form.
- Minimization refers to the process in which we simplify the algebraic expressions of any given Boolean function.
- This process is very important as it helps in the reduction of the overall cost, operational delay and complexity of an associated circuit.

Simplification of Boolean function

- $F = x'y'z + x'yz + xy'$

  $= x'z\,(y' + y) + xy'$

  $= x'z + xy'$ $[\ y'+y = 1]$
- Here, we can see that simplification reduces the number of gates used.



(a) F2 = x'y'z + x'yz + xy'
(b) F2 = xy' + x'z

Simplification of Boolean function

Some commonly used Boolean postulates used in simplification of Boolean function:

- $A + A = A$
- $A.A = A$
- $A + A' = 1$
- $A + A'B = (A + A')(A + B) = A + B$
- $A + AB = A(1 + B) = A$
- $A.A' = 0$
- $(A + B)(A + C) = A.A + A.C + A.B + B.C = A + AB + AC + BC = A(1 + \mathrm{B} + \mathrm{C}) + \mathrm{BC} = \mathrm{A}.1 + \mathrm{BC} = \mathrm{A} + \mathrm{BC}$

Simplification of Boolean function

- **Example:** $F = ABC'D' + ABC'D + AB'C'D + ABCD + AB'CD + ABCD' + AB'CD'$

$F = ABC' (D' + D) + AB'C'D + ACD (B + B') + ACD' (B + B')$

$= ABC' + AB'C'D + ACD + ACD'$

$= ABC' + AB'C'D + AC (D + D')$

$= ABC' + AB'C'D + AC$

$= A (BC' + C) + AB'C'D$

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

$= AB + AC + AB'C'D$

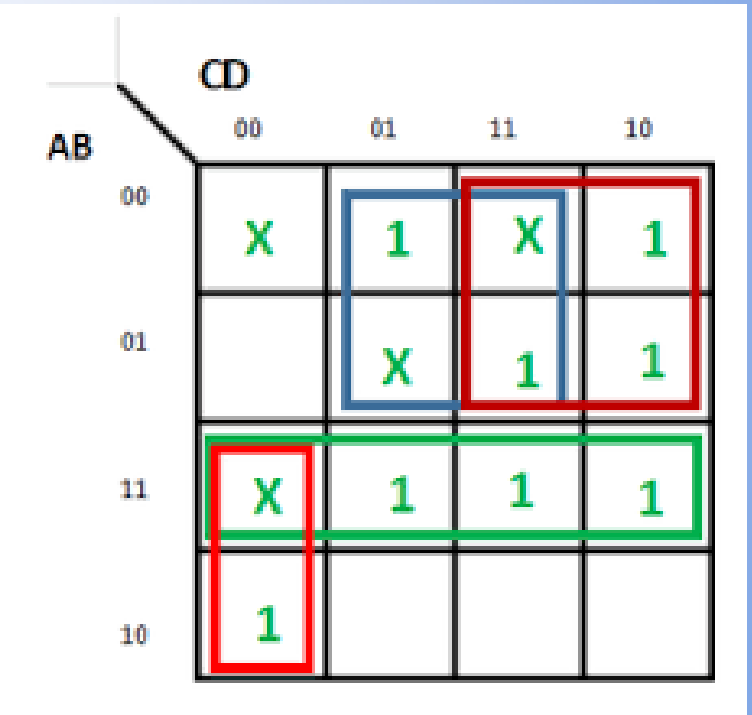
$= AB + AC + AC'D$

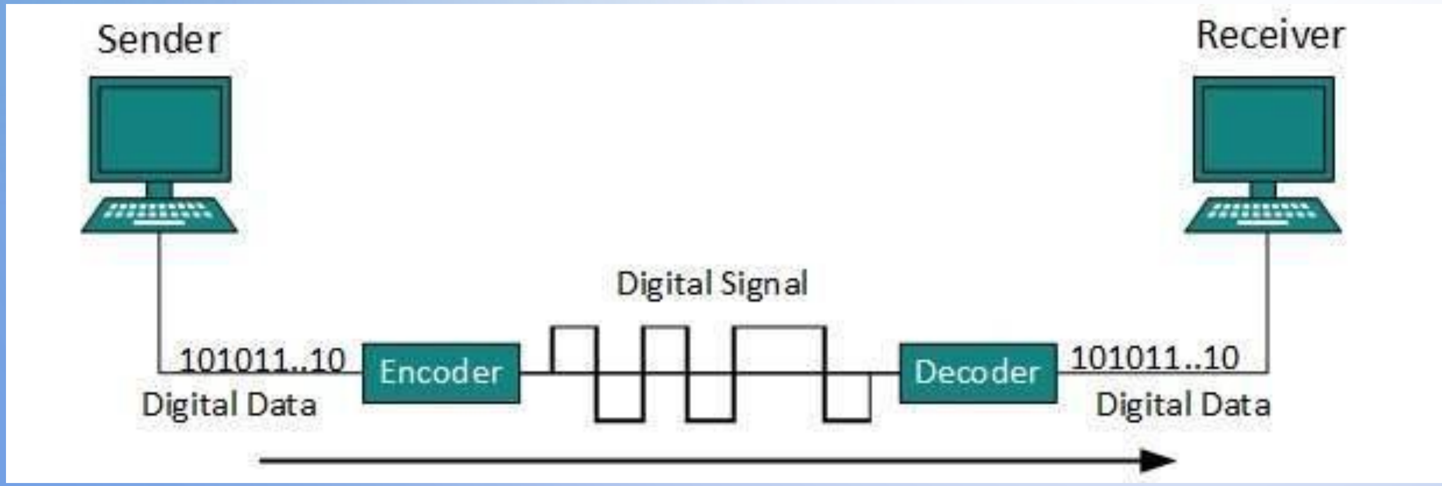
$= AB + AC + AD$

Gray Code

- A Gray code is an encoding of numbers so that adjacent numbers have a single digit differing by 1.
- The term Gray code is often used to refer to a ‘reflected’ code, or more specifically still, the binary reflected Gray code and is named after Frank Gray.
- For example, the representation of the decimal value ‘1’ in binary would normally be ‘001’ and ‘2’ would be ‘010’. In Gray code, these values are represented as ‘001’ and ‘011’. That way, incrementing a value from 1 to 2 requires only one bit to change, instead of two.

Application of Gray Code





Binary and Gray code

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101

K map

- Maurice Karnaugh, a telecommunications engineer, developed the Karnaugh map at Bell Labs in 1953 while designing digital logic based telephone switching circuits.
- **A Karnaugh map or a K-map refers to a pictorial method that is utilized to minimize various Boolean expressions.**
- Karnaugh maps reduce logic functions more quickly and easily compared to Boolean algebra. By reducing we mean simplifying, i.e, reducing the number of gates and inputs.

Applications of K map

- **Digital Circuits:** Karnaugh maps are widely used in the design of digital circuits. The simplified expressions obtained from K-Maps can be easily translated into logic gates, making it easier to design and implement the circuit.
- **Computer memory:** K-Maps are used in the design of computer memory. The simplified expressions obtained from K-Maps help in reducing the size and complexity of the memory circuit.
- **Communication systems:** K-Maps are used in the design of communication systems.
- **Consumer electronics:** K-Maps are used in the design of consumer electronics such as televisions, radios, and other electronic devices.

Steps of K-map

- Steps used to solve an expression using the K-map method:

1. Select a K-map according to the total number of input variables. A K-map for a function of n input variables will have 2^n squares and each adjacent squares will differ in only one variable.

2. In order for vertically and horizontally adjacent squares to differ in only one variable, the top-to-bottom labeling must be done in the order shown: $A'B', A'B, AB, AB', C'D', C'D, CD, CD'$.

3. Identify maxterms or minterms as given in the function.

4. For SOP, put the 1's in the blocks of the K-map with respect to the minterms (elsewhere 0's).

Steps of K-map

5. For POS, putting 0’s in the blocks of the K-map with respect to the maxterms (elsewhere 1’s).

6. Making rectangular groups of 1s in pairs/ quads/ octets and trying to minimize the number of groups but maximize the size of groups.

7. From the groups that have been created in step 5, find the product terms and then sum them up for the SOP form.

Example of K map

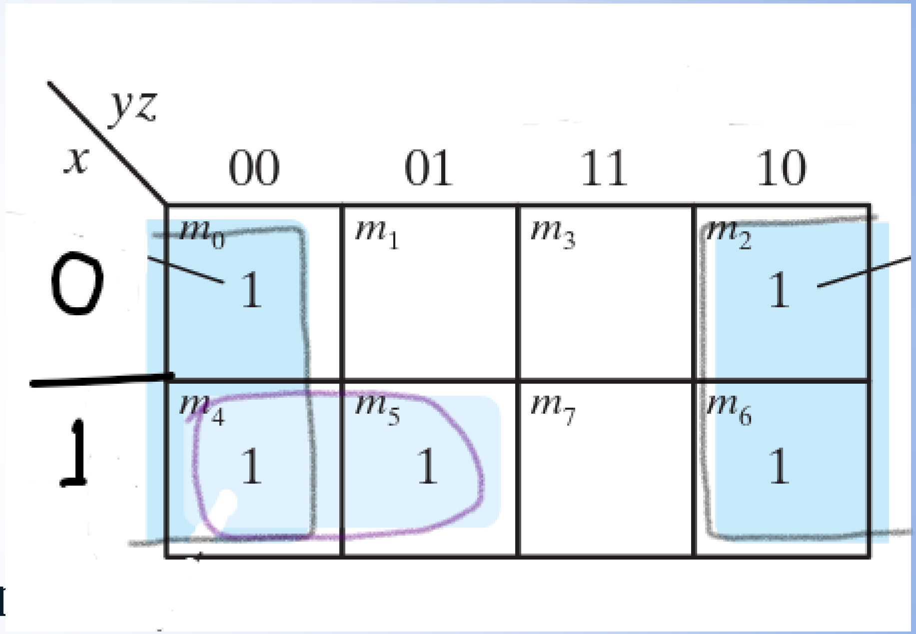
Simplify the Boolean function

F (x, y, z) = ∑(0, 2, 4, 5, 6)

F(x,y,z) = x’y’z’+x’yz’+xy’z’+xy’z’+xyz’

First, we combine the four adjacent squares in the first and last columns to give the single literal term z’. The remaining single square, representing minterm 5, is combined with an adjacent square that has already been used once. These two adjacent squares give the term xy’.

= z’ + xy’



Example of K map

Use a K map to simplify y = C'(A' B'D' + D) + AB'C + D'.

Solution:

1. Multiply out the first term to get the function in SOP form.
2. y = A'B'C'D' + C'D + AB'C + D'
3. For the C'D term, place a 1 in all squares with C'D in their

labels, that is, A B CD, ABCD, ABCD, A C'D.

4. For the AB'C term, place a 1 in all squares that have

an AB'C in their labels, that is, AB'CD', AB'CD.

5. For the D' term, place a 1 in all squares that have a D' in

their labels. That is, all squares in the leftmost and rightmost columns.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	1	1	0	1
AB	1	1	0	1
$A\bar{B}$	1	1	1	1

$y = A\bar{B} + \bar{C} + \bar{D}$

Example of K map

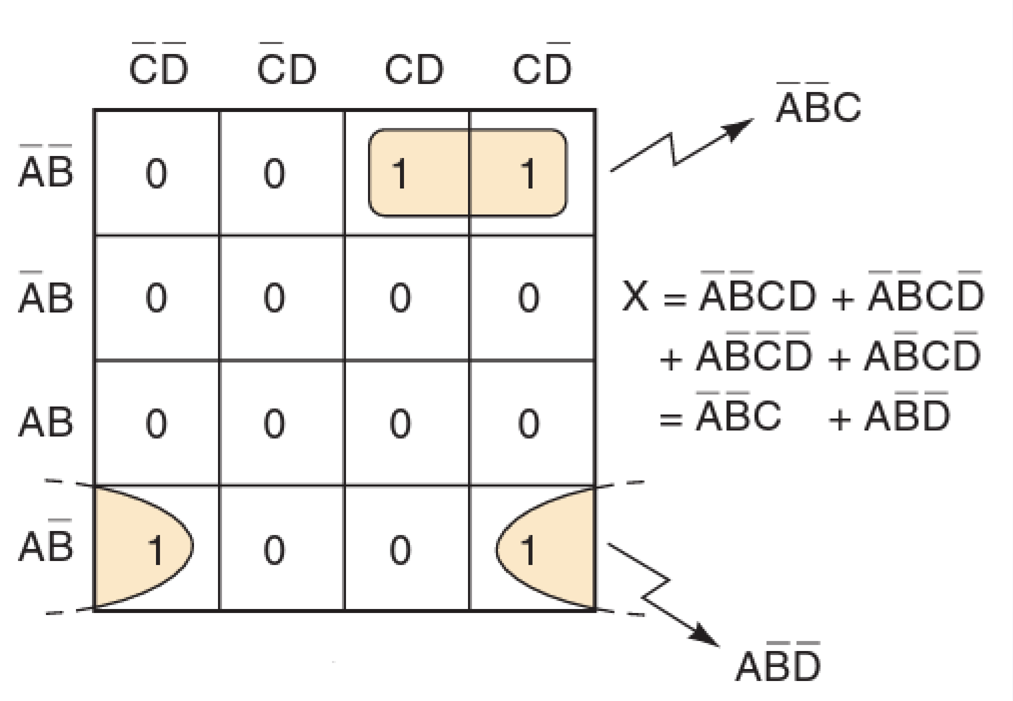
	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}B$	1	0
AB	1	0
$A\overline{B}$	0	0

$$X = \overline{A}B\overline{C} + AB\overline{C}$$
$$= B\overline{C}$$

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}B$	1	1
AB	0	0
$A\overline{B}$	0	0

$$X = \overline{A}B\overline{C} + \overline{A}BC$$
$$= \overline{A}B$$

Example of K map



Don’t care conditions

Till now, the Boolean expressions which have been discussed by us were completely specified, i.e., for each combination of input variable we have specified a minterm by representing them as 1 in the K-Map.

But, there may arise a case when for a given combination of input we may not have a specified output or the input combination may be invalid. The combinations for which we don't have any output expression specified are called don't care combination.

For Example, in 8421 code, input states 1001, 1010, 1011, 1100, 1101, 1110 and 1111 are invalid and the corresponding output is the don't care.

These don't care combinations in the K-Map are denoted by an X (cross) symbol.

Rules for Karnaugh Maps with Don't Care Conditions

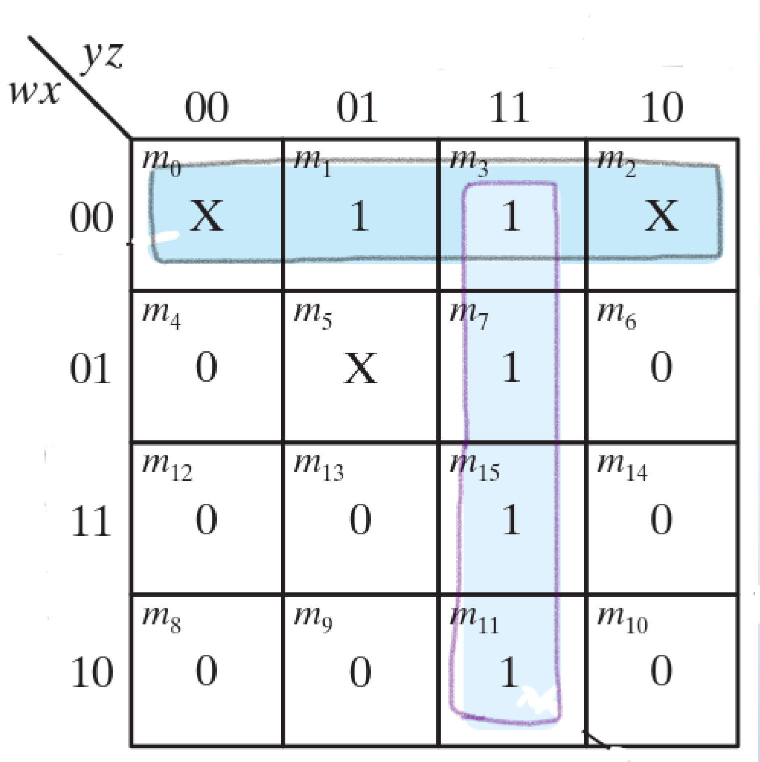
- After forming the K-Map, fill 1's at the specified positions corresponding to the given minterms. Fill X at the positions where don't care combinations are present.
- Now, Encircle the groups in the K-Map. One thing to be kept in mind is, now we can treat Don't Care conditions (X) as 1s if these help in forming the largest groups. No such group can be encircled whose all the elements are X.
- If still there are 1s left which doesn't get encircled in any of the groups, then these isolated 1s are encircled individually.
- Write the Boolean expression for each encircled group.
- The final minimal expression can be obtained by ORing each Boolean expressions that were obtained from each group.

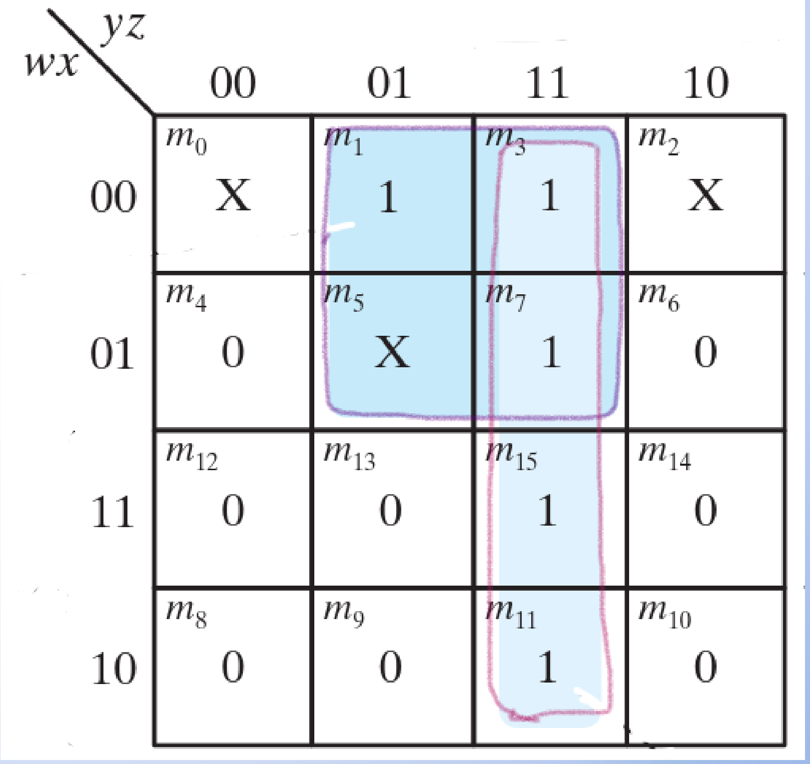
Examples of Don’t care conditions

Simplify the Boolean function $F\ (w, x, y, z)\ = \sum(1, 3, 7, 11, 15)$

which has the don’t-care conditions $d\ (w, x, y, z) = \sum(0, 2, 5)$

Solution: $F(w, x, y, z) =\ w’x’ +\ yz$ **OR** $w’z + yz$





Practice problems of K-Map

1. $F(x, y, z) = \sum(0,1,4,5,6)$
2. $d(x, y, z) = \sum(2,3,7)$
3. $F(A,B,C,D) = \sum(0,1,2,5,7,8,9,10,13,15)$
4. $F(w, x, y, z) = \sum(0,2,5,7,8,10,12,13,14,15)$
5. $F(A, B, C, D) = \sum(4, 7, 12, 10)$
6. $d(A, B,C,D) = \sum(0, 6, 8)$