Karnaugh Maps

- **Minimizing** circuits helps reduce the number of components in the actual physical implementation.
- Reducing Boolean expressions can be done suing Boolean identities; however, using identities can be very difficult because **no rules** are given on how or when to use the identities.
- Karnaugh Map is a graphic representation of a Boolean function in the form of a map as way to simplify the Boolean functions.
- Karnaugh Map is due to M. Karnaugh, who introduced (1953) his version of the map in his paper.
- The map method provides a simple straight forward procedure for minimizing Boolean functions.
- Karnaugh Map, abbreviated as **K-map**, is actually pictorial form of the truth-table.
- The map is a diagram made up of squares each square represents one minterms. Thus, Karnaugh map of a Boolean function is graphical arrangement of minterms, for which the function is asserted.

Minterm	X	Y
$\bar{X}\bar{Y}$	0	0
$\bar{X}Y$	0	1
$X\bar{Y}$	1	0
XY	1	1

Minterms for Two Variables

Minterm	X	Y	Z
$\bar{X}\bar{Y}\bar{Z}$	0	0	0
$\bar{X}\bar{Y}Z$	0	0	1
$\bar{X}Y\bar{Z}$	0	1	0
$\bar{X}YZ$	0	1	1
$X\bar{Y}\bar{Z}$	1	0	0
$X\bar{Y}Z$	1	0	1
$XY\bar{Z}$	1	1	0
XYZ	1	1	1

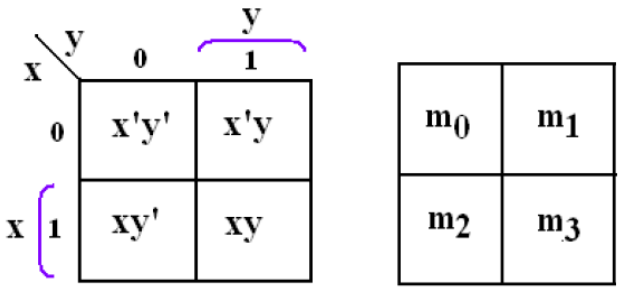
Minterms for Three Variables

- **The Rules of k-map simplifications for n-variables are:**

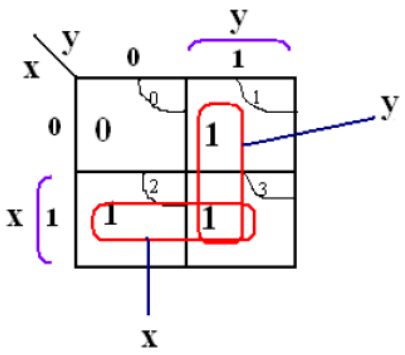
  1) **Groupings** can contain only **1**s; no 0s.
  2) Only 1s in **adjacent** cells can be grouped; diagonal grouping is not allowed.
  3) The number of 1s in a group must be a **power of 2**.
  4) The groups must be made as **large** as possible while still following all rules.
  5) All 1s must belong a group, even if it is a group of one.
  6) **Overlapping** groups are allowed.
  7) **Wrap** around is allowed.
  8) Use the **fewest** number of groups possible.

1- Two variables maps

- Any two-variable function has $2^2 = 4$ minterms.
- **A** three variable map is shown below,

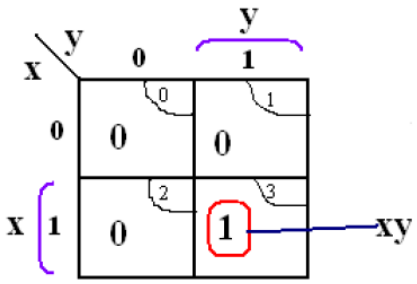


- **Example-1:** simplify F using k-map method, where F(x,y) = Σ(1,2,3)



F = x + y

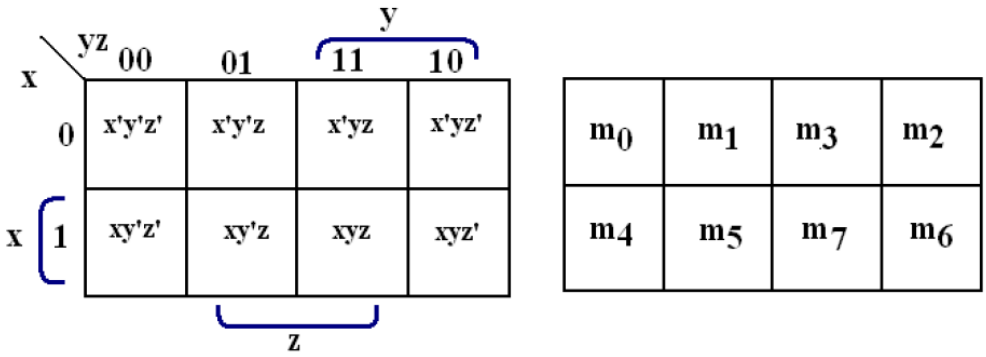
- **Example-2:** simplify F using k-map method, where F(x,y)=Σ(3)



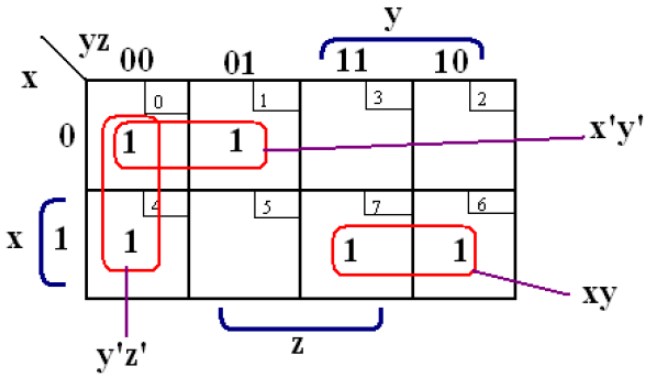
F = xy

2- Three variables maps

- A three variable map is shown below, note that the minterms are arranged, not in binary sequence, the characteristic of this sequence is that the only one bit changed from (1 to 0) or from (0 to 1) in the listing sequence.

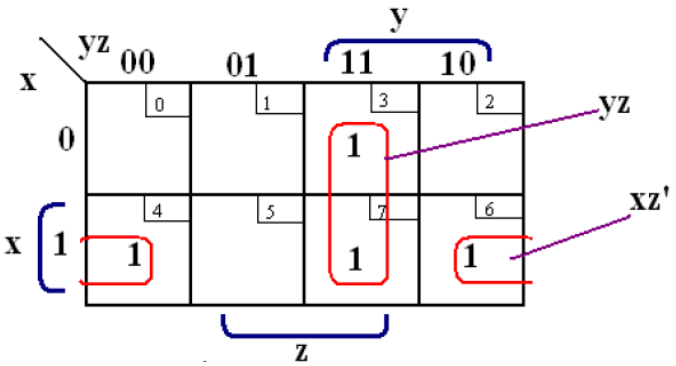


- **Example-3:** simplify F using map method, where F(x,y,z)=Σ(0,1,4,6,7)



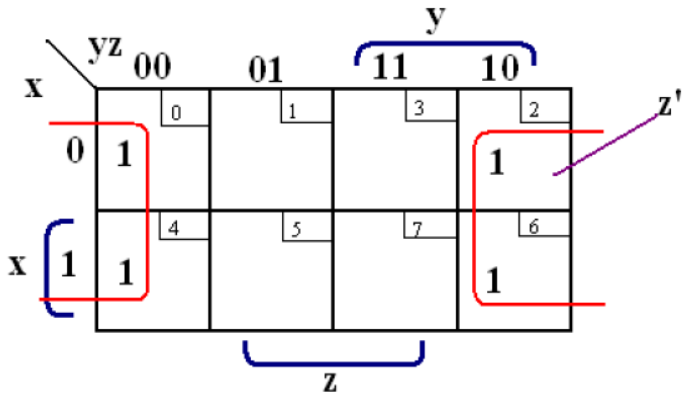
F = x'y' + xy + y'z'

- **Example-4:** simplify F using map method, where F(x,y,z)=Σ(3,4,6,7)



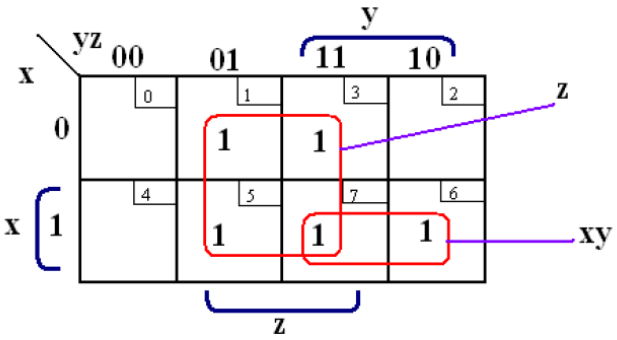
F = yz + xz'

- **Example-5:** simplify F using map method, where F(x,y,z)=Σ(0,2,4,6)



F = z'

- **Example-6:** simplify F using map method, where F(x,y,z)=Σ(1,3,5,6,7)



F = z + xy

- **Note:-** any combination of 4 adjacent squares in the 3-variables map, which represents the ORing of four adjacent minterms will result in an expression of only one literals.

- **Example-7:** simplify F using k-map method, where F= A'B'C' + A'C + A'B + AB'C + BC

  Sol.: the terms need to be expressed as a sum of minterms as explain before, so A'C missing B, A'B missing C and BC missing A

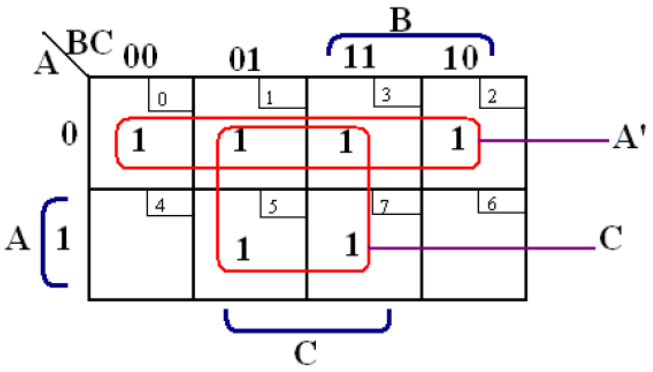
  F= A'B'C' +A'C (B+B') + A'B(C+C') + AB'C + BC(A+A')

  = A'B'C' +**A'BC**+ A'B'C + A'BC + **A'BC**' + AB'C + ABC + A'BC

  = A'B'C' +A'B'C +A'BC' + A'BC + AB'C + ABC

  ∴ F(A,B,C)=Σ(0,1,2,3,5,7)

  and the simplification of F using the map methods is as follows:

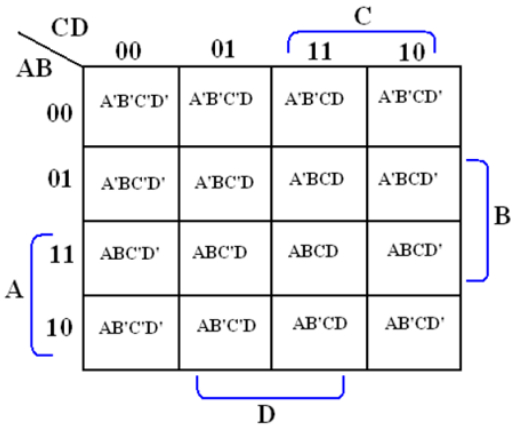


F = A’ + C

- H.W. simplify F using k-map method, where F= x’y’z + x’yz + xy’z + xyz
- H.W. simplify F using k-map method, where F= x’y’z’ + x’y’z + x’yz + x’yz’ + xy’z’ + xyz’

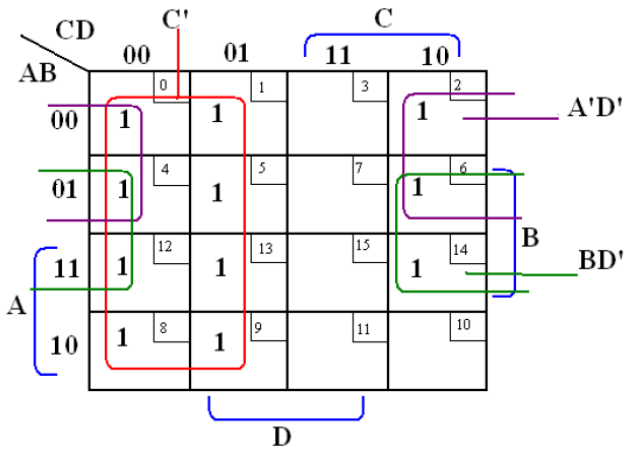
3- Four variables maps

- The combinations of adjacent squares that is useful during the simplification process easily determined for inspection of the 4-variable map



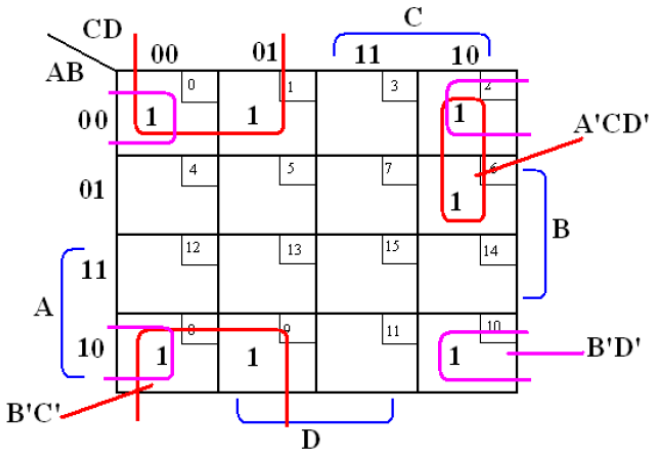
m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

- **Example-8:** simplify F using map method, where F(A,B,C,D)=Σ(0,1,2,4,5,6,8,9,12,13,14)



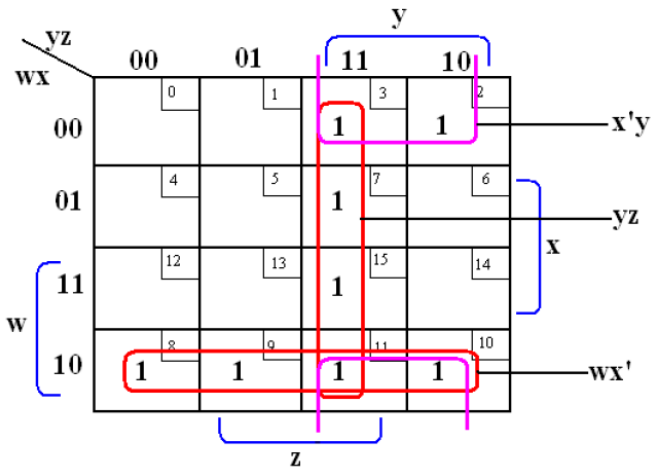
F = A'D' + BD' + C'

- **Example-9:** simplify F using map method, where F(A,B,C,D)=Σ(0,1,2,6,8,9,10)



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

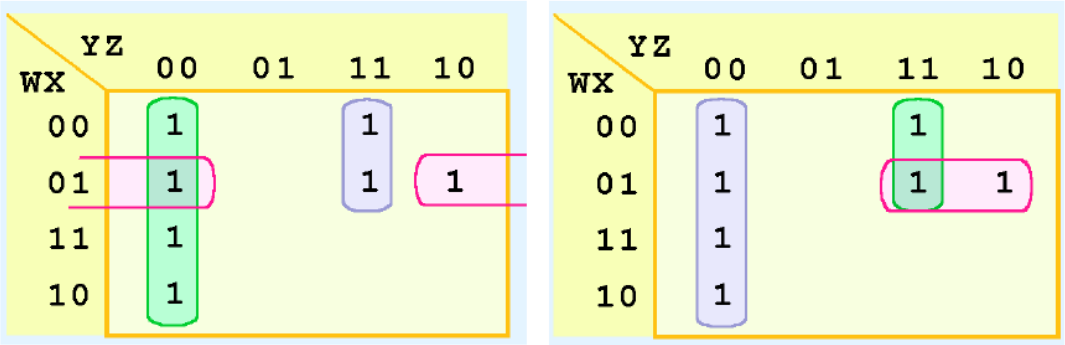
- **Example-10:** simplify F using map method, where F(w,x,y,z)=Σ(2,3,7,8,9,10,11,15)



F = x'y + yz + wx'

- **H.W.** simplify F using k-map method, where F= w'x'y'z' + w'x'y'z + w'x'yz' + w'xyz' + wx'y'z' + wx'y'z' + wx'y'z + wx'yz'

- Here, F1 and F2 are different, however, are **equivalent:**

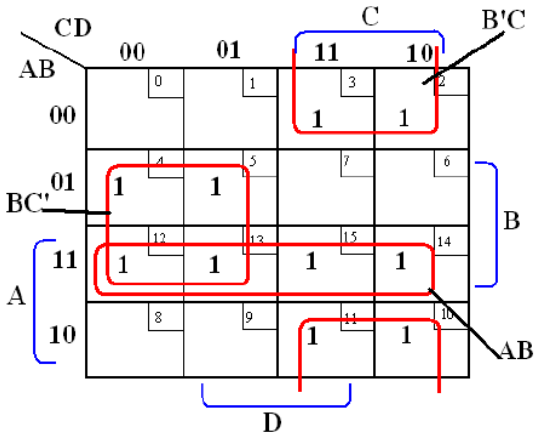


F(w, x, y, z) = **F1** = y'z' + w'yz + **w'xz'** F(w, x, y, z) = **F2** = y'z' + w'yz + **w'xy**

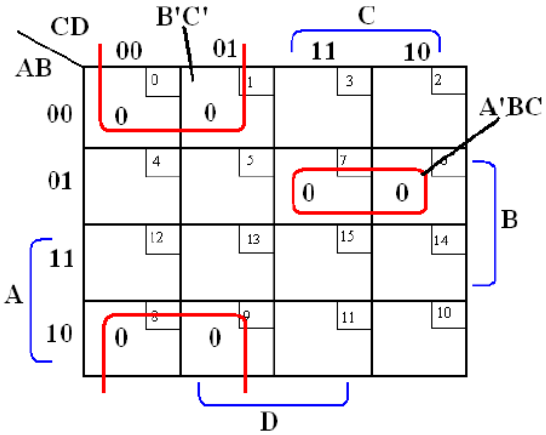
4- Product of Sum (PoS) Simplification

- The minimized Boolean function derived from the map in all previous example were expressed in the sum of product (SoP) form, with a minor or modification the product of sums forms can be obtained.
- The process for minimizing a POS expression is basically the same as for an SOP expression except that you group 0s to produce minimum sum terms instead of grouping 1s to produce minimum product terms.
- The rules for grouping the 0s are the same as those for grouping the 1s that you learned before.

- **Example=11:** simplify F using map method as
  a) Sum of products
  b) Product of sums,
  Where F(A,B,C,D)=Σ(2,3,4,5,10,11,12,13,14,15)



a) F = B'C + AB + BC'
(S.O.P)



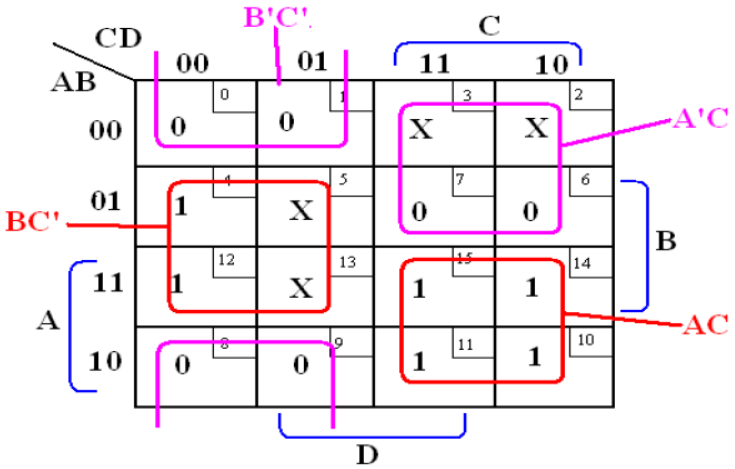
b) F' = (B'C' + A'BC)'
F = (B + C) (A + B' + C')
(P.O.S)

- **H.W:** simplify F using map method as
  a) Sum of products
  b) Product of sums,
  Where F(A,B,C)= (A+B+C) (A+B+C') (A+B'+C) (A+B'+C') (A'+B'+C)

5- Don't Care Condition

- There are certain situations where a function may not be completely specified, meaning there may be some inputs that are **undefined** for the function.
- **Real circuits don't** always need to have an **output** defined for every possible input.
- If a circuit is designed so that a particular set of inputs can **never happen**, we call this set of inputs a **don't care** condition.
- They are very **helpful** to us in K-map circuit simplification. Because they are input values that should not matter (and should never occur), we can let them have values of **either 0 or 1**, depending on which helps us the most.
- Don't care values are typically indicated with an "**X**" in the appropriate cell.

- **Example-12:** simplify F using map method as
  a) Sum of products
  b) Product of sums
  Where F(A,B,C,D)=Σ(4,10,11,12,14,15) and the don't care conditions d(A,B,C,D)=Σ(2,3,5,13)



a) F= BC' + AC
S.O.P

b) F' = (A'C + B'C')'
F= (A + C') (B + C)
P.O.S