MINIMIZATION OF LOGIC CIRCUITS

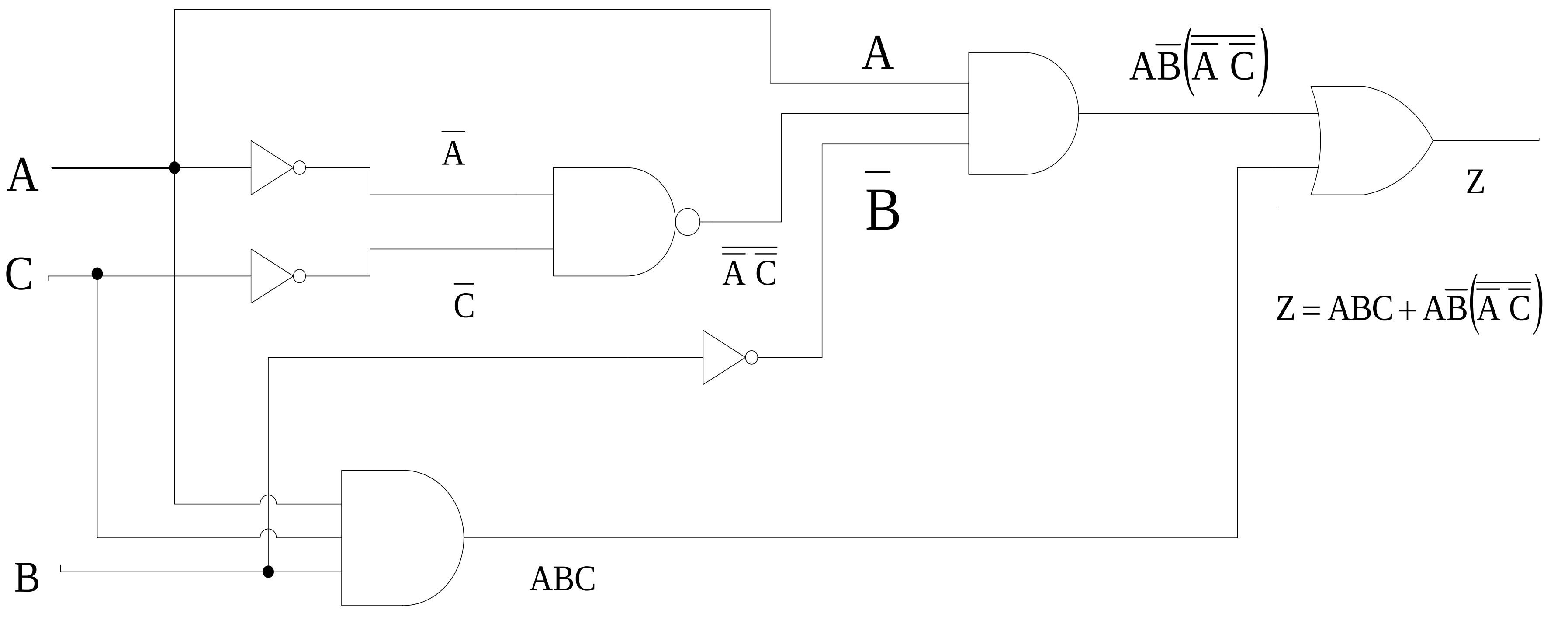
Algebraic Simplification

To simplify a logic expression –

1. The original expression is put into the SOP form by repeated application of De Morgan's theorem and multiplication of terms.

2. Once it is in this form, the product terms are checked for common factors, and terms combined whenever possible. This eliminates one or more terms.

Example : Simplify the following logic circuit



$$Z = ABC + A\overline{B}\,(\overline{\overline{A}\;\overline{C}})$$

$$= ABC + A\overline{B}\,(\overline{\overline{A}} + \overline{\overline{C}})$$

$$= ABC + A\overline{B}\,(A + C)$$

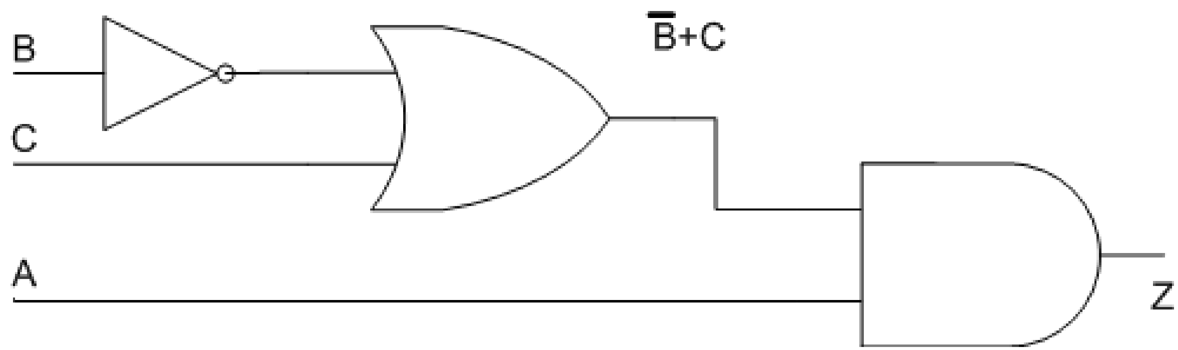
$$= ABC + A\overline{B}A + A\overline{B}C$$

$$= ABC + A\overline{B} + A\overline{B}C$$

$$= AC\,(B + \overline{B}) + A\overline{B}$$

$$= AC + A\overline{B}$$

$$= A\,(C + \overline{B})$$



Minimization of Logic Expression with Karnaugh Map (K- Map)

SSOP: Standard Sum of Products Form
SPOS: Standard Product of Sums Form

In each term, each logic variable appears once, either complemented or uncomplemented

SSOP Form - Min term Representation

$$\text{Example:} \quad A\,\overline{B}\,C$$

- binary digit 1 is assigned to each uncomplemented variable
- binary digit 0 is assigned to each complemented variable

$$\therefore \quad A\,\overline{B}\,C = 101_2 = 5_{10} = m_5$$

SPOS Form - Max term Representation

$$\text{Example} \qquad A + \overline{B} + C$$

- binary digit 0 is assigned to each uncomplemented variable

- binary digit 1 is assigned to each complemented variable

$$\therefore \quad A + \overline{B} + C \rightarrow 010_2 \rightarrow 2_{10} = M_2$$

Examples: Min Terms and Max Terms

$$f(A,B,C,D) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}BC\bar{D} + A\bar{B}C\bar{D} + ABCD$$

$$\begin{array}{cccc} 0000 & 0110 & 1010 & 1111 \\ 0 & 6 & 10 & 15 \\ m_0 + & m_6 + & m_{10} + & m_{15} \end{array}$$

Min Terms

$$= \sum m(0,6,10,15)$$

$$f(A,B,C,D) = (A+B+C+D) \quad (A+\bar{B}+C+\bar{D}) \quad (\bar{A}+B+\bar{C}+\bar{D}) \quad (\bar{A}+\bar{B}+\bar{C}+\bar{D})$$

$$\begin{array}{cccc} 0\;0\;0\;0 & 0\;1\;0\;1 & 1\;0\;1\;1 & 1\;1\;1\;1 \\ M_0 & M_5 & M_{11} & M_{15} \end{array}$$

$$= \prod M(0,5,11,15)$$

Max Terms

Karnaugh Map Representation of Logical Functions

The Karnaugh map (K-map) is a diagram used to simplify a logic equation or to convert a truth table to its corresponding logic circuit in a simple, orderly process.

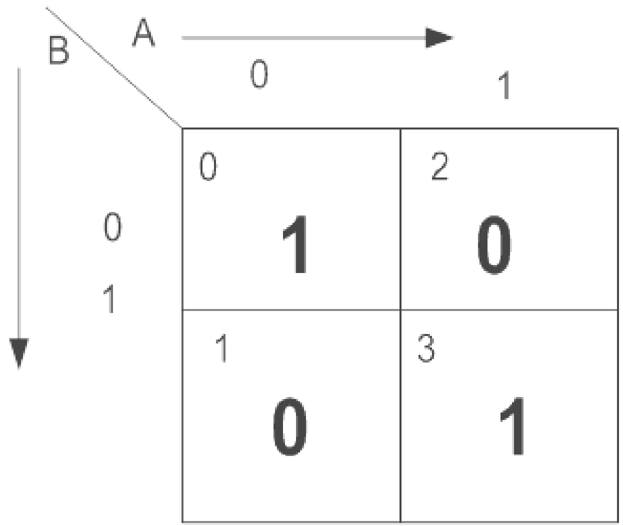
Easy to do for up to 4 variables. Can be done for 5 variables with some effort. Not feasible for more than 4 variables – for this ***Quine-McCluskey Algorithm*** should be used,

Two Variable K-Map

Example

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

K-Map Format



$$f(A, B) = \bar{A}\bar{B} + AB = m_0 + m_3$$

Minterm Representation

$$f(A, B) = (A+\bar{B})(\bar{A}+B) = M_1 M_2$$

Maxterm Representation

Three Variable K- Map

Ex.

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

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

$$= m_0 + m_1 + m_2 + m_6$$

AB → Grey Code

C \ AB	00	01	11	10
0	0: 1	2: 1	6: 1	4: 0
1	1: 1	3: 0	7: 0	5: 0

K- Map Representation

Four Variable K- Map

Ex. $$f = \overline{A}\,\overline{B}\,\overline{C}\,D + \overline{A}\,B\,\overline{C}\,D + A\,B\,\overline{C}\,D + A\,B\,C\,D$$

$$= m_1 + m_5 + m_{13} + m_{15}$$

Grey Code →

Grey Code ↓

CD \ AB	00	01	11	10
00	0	4	12	8
01	1 1	5 1	13 1	9
11	3	7	15 1	11
10	2	6	14	10

Simplification of Logical Functions using Karnaugh Map

The graphical arrangement of the variables in a Karnaugh Map allows an easy way to eliminate variables.

Can be easily extended to allow further simplification if there are Don’t Care States (discussed later)

The Essential Feature of the K- Map

Adjoining boxes of the K-Map, both horizontally and vertically (but not diagonally) correspond to minterms, or maxterms ,which differ in only a single variable - this variable appearing complemented in one term and uncomplemented in the other.

It is precisely to achieve this end that the Grey Code is used to number the rows and columns.

Example

CD \ AB	00	01	11	10
00	0	4	12 **1**	8 **1**
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Consider for example, minterms m_8 and m_{12} .They adjoin horizontally. We have, $m_8 = A\,\overline{B}\,\overline{C}\,\overline{D}$ $\quad m_{12} = A\,B\,\overline{C}\,\overline{D}$

These two minterms differ only in that the variable B appears complemented in one and uncomplemented in the other.

They can therefore be combined to yield

$$A\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D} = A\overline{C}\,\overline{D}(\overline{B} + B) = A\overline{C}\,\overline{D}$$

Thus two terms, each involving four variables, have been replaced by a single term involving three variables.

The variable which appeared complemented in one term and uncomplemented in the other has been eliminated.

In the K-map ,the presence of these minterms are noted by placing 1s in the appropriate boxes of the K-map. As these minterms correspond to adjoining boxes, so they can be combined.

CD \ AB	00	01	11	10
00	0	4	12 1	8 1
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$$A\overline{C}\,\overline{D}$$

Rule for Combining Two NeighbouringTerms:

Any pair of adjoining minterms (vertically or horizontally) can be combined into a single term involving one variable fewer than what was there in the original minterms themselves.

The variable which is complemented in one and uncomplemented in the other is removed.

Applying this rule to m_8 and m_{12} , we see that for both these minterms A = 1, C = 0, D = 0. But B = 0 in m_8 and B = 1 in m_{12} . So we eliminate B.

Combining, we get

$$A\,\overline{B}\,\overline{C}\,\overline{D} + A\,B\,\overline{C}\,\overline{D} = A\,\overline{C}\,\overline{D}(\overline{B} + B) = A\,\overline{C}\,\overline{D}$$

Example

$$AB\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + A\overline{B}C\overline{D}$$

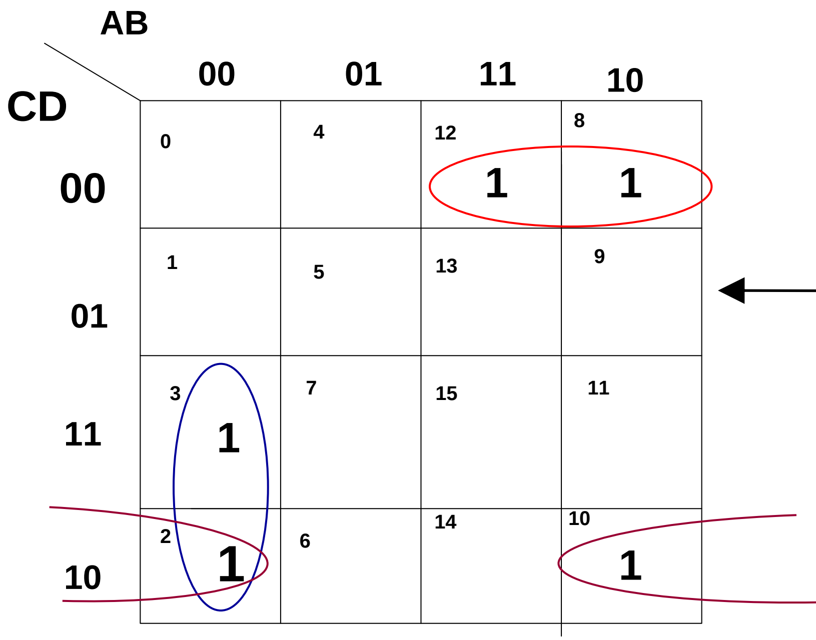
CD \ AB	00	01	11	10
00	0	4	12 **1**	8 **1**
01	1	5	13	9
11	3 **1**	7	15	11
10	2 **1**	6	14	10 **1**

Note the *wrap-around* feature of the K-Map in determining neighbours

$$f(A,B,C,D) = \sum m(2,3,8,10,12) = A\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{D}$$

Example

$$AB\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}CD + \overline{A}\overline{B}C\overline{D} + A\overline{B}C\overline{D}$$



Another way of combining for the same original expression gives different result/

RESULT NOT UNIQUE!

$$f = (A,B,C,D) = \sum m(2,3,8,10,12) = A\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C + \overline{B}\,C\overline{D}$$

On the same row, each box in the leftmost column adjoins the box in the right most column. As also, the topmost row adjoins the bottommost row

$$f = (A,B,C,D) = \sum m(2,3,8,10,12) = A\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{D} \quad (1a)$$

$$f = (A,B,C,D) = \sum m(2,3,8,10,12) = A\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C + \overline{B}\,C\overline{D} \rightarrow \quad (1b)$$

Either of these results is equally acceptable and equally economical --→ each requires a single OR gate and three AND gates.

In arriving at (1a), we used minterm m_8 twice, this repetitive use of a minterm is allowable since in using, say, m_8 twice, we have used the theorem

$$m_8 = A\overline{B}\,\overline{C}\,\overline{D} = A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}\,\overline{D}........$$

In arriving at (1b), we used minterm m_2 twice

$$m_2 = \overline{A}\,\overline{B}C\overline{D} = \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}C\overline{D}........$$

Larger groupings in a K-map

Two adjoining boxes can be combined to yield a term from which one variable has been eliminated. Similarly, when 2^n boxes adjoin, they can be combined to yield a single term from which n variables have been eliminated.

Ex.

CD \ AB	00	01	11	10
00	0	4	12	8
01	1 1	5 1	13	9
11	3 1	7 1	15	11
10	2	6	14	10

$$f(A,B,C,D) = (m_1 + m_5) + (m_3 + m_7)$$

$$= \overline{A}\,\overline{C}\,D + \overline{A}\,C\,D$$

$$= \overline{A}\,D(\overline{C} + C)$$

$$= \overline{A}\,D$$

Ex.

CD \ AB	00	01	11	10
00	0 1	4	12	8 1
01	1	5	13	9
11	3	7	15	11
10	2 1	6	14	10 1

$$=f(A,B,C,D)=\sum m(0,2,8,10)=\overline{B}\,\overline{D}$$

Ex.

CD \ AB	00	01	11	10
00				
01	1	1	1	1
11				
10				

$$f(A,B,C,D)=\overline{C}\,D$$

Ex.

CD \ AB	00	01	11	10
00		1	1	
01				
11				
10		1	1	

$$f(A,B,C,D)=B\overline{D}$$

Ex.

CD \ AB	00	01	11	10
00	1	1		
01	1	1		
11	1	1		
10	1	1		

$$f(A,B,C,D)=\overline{A}$$

Ex.

CD \ AB	00	01	11	10
00	1	1	1	1
01				
11				
10	1	1	1	1

$$f(A,B,C,D)=\overline{D}$$

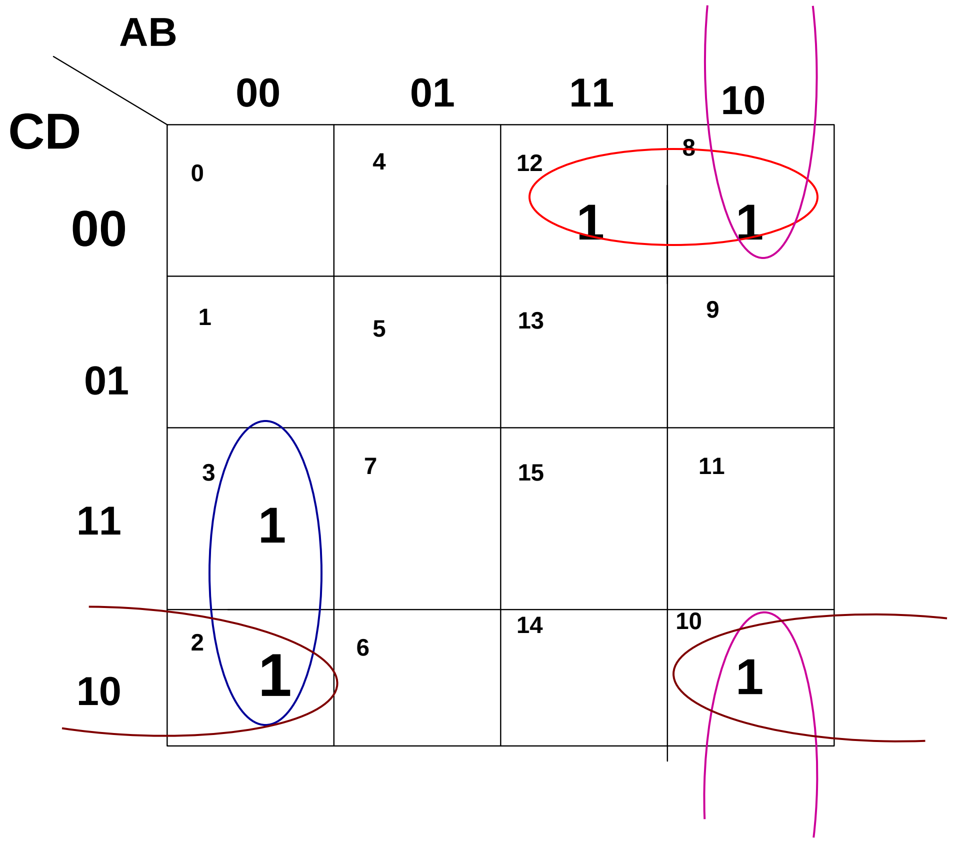
When a logical function has been expressed in Standard Form in terms of its minterms, the K-map may be used to simplify the function by applying the following principles –

1. The combination of boxes (minterms) which are selected must be such that each box is selected at least once.

   However, a particular box may be involved in a number of different combinations.

2. The individual combinations should be selected to contain as many boxes as possible so that all boxes will be included in as few different combinations as possible.

Ex.



$$f = (A,B,C,D) = (m_2 + m_3) + (m_8 + m_{12}) + (m_8 + m_{10})$$

$$\text{or } f = (A,B,C,D) = (m_2 + m_3) + (m_8 + m_{12}) + (m_2 + m_{10})$$

$$\text{Let, } p_1 = m_2 + m_3 \qquad p_2 = m_8 + m_{12}$$

$$p_3 = m_8 + m_{10} \qquad p_4 = m_2 + m_{10}$$

$$\text{So, } f(A,B,C,D) = p_1 + p_2 + p_3$$

$$\text{or, } f(A,B,C,D) = p_1 + p_2 + p_4$$

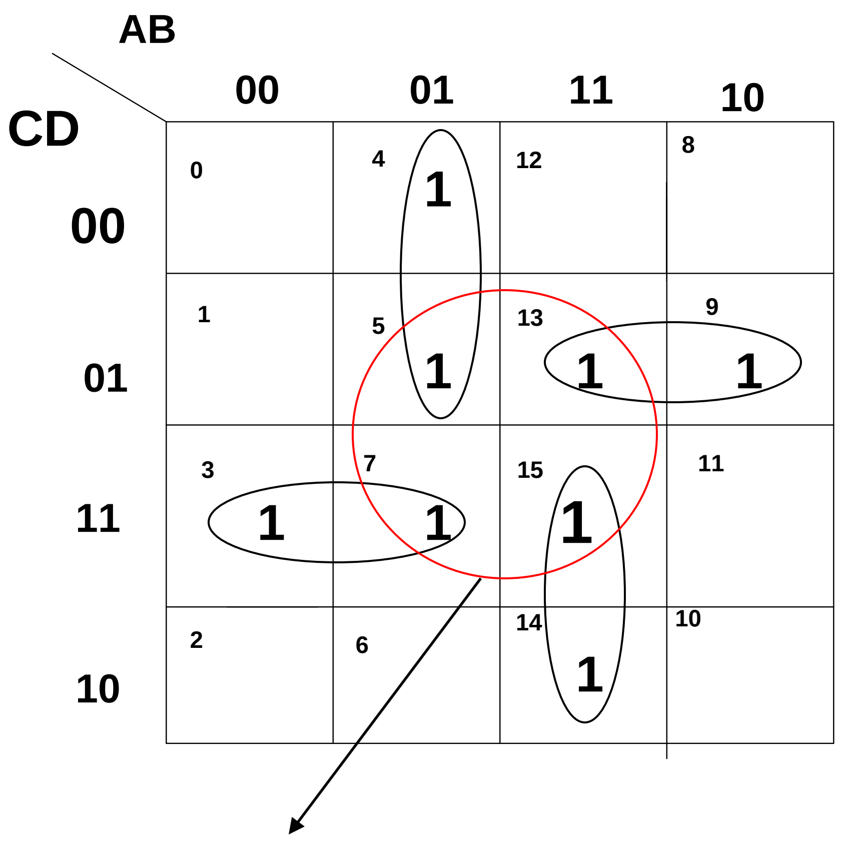
p_1, p_2, p_3, p_4 ----- Prime Implicants

There is no alternative but to use p_1 and p_2 as otherwise m_3 and m_{12} will not be accounted for.

$\Longrightarrow$ p_1, p_2 ----- Essential Prime Implicants

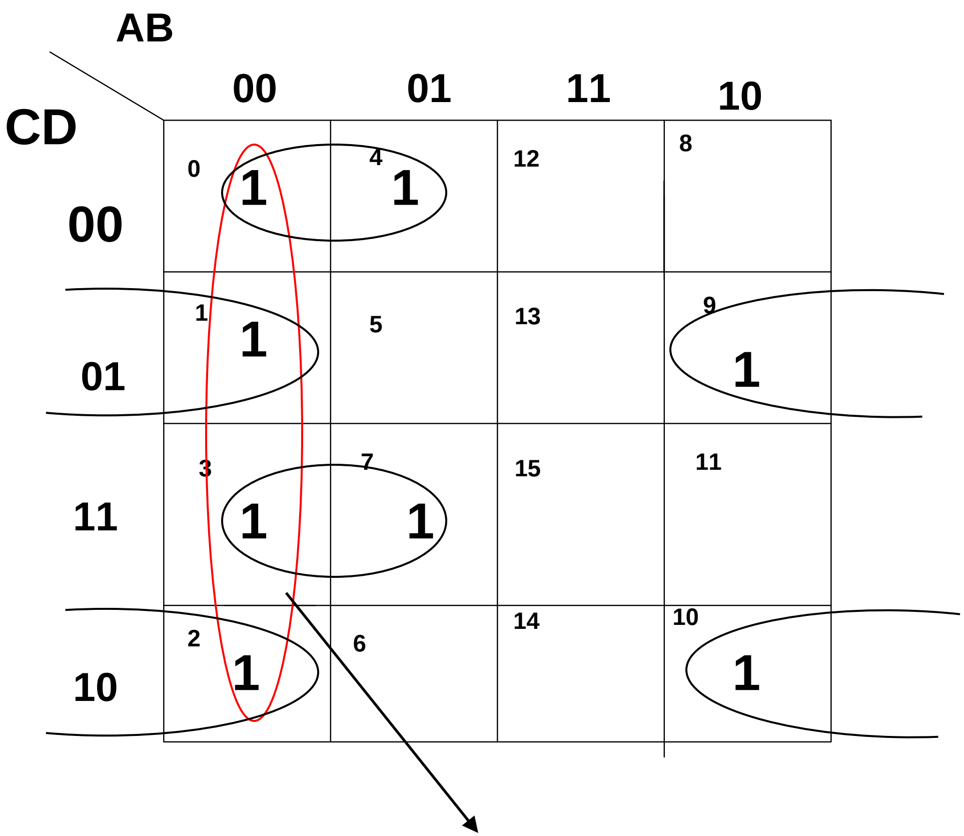
❖ *Hazard with preoccupation for finding prime implicants to make as large group as possible*

Ex.



Superfluous Grouping

Ex.



Superfluous Grouping

Rules of Making Combination
in
K- Map
to avoid such hazards

1. Encircle and accept as essential prime implicants any box or boxes that cannot be combined with any other.

2. Identify the boxes that can be combined with a single other box in only one way.

   Encircle such two-box combinations. A box which can be combined into a two-grouping but can be so done in more than one way is to be temporarily bypassed.

3. Identify the boxes that can be combined with three other boxes in only one way.

   If not all of the four boxes so involved are already covered in groupings of two, encircle these four boxes. Again, a box which can be grouped in groups of four in more than one way is to be temporarily bypassed.

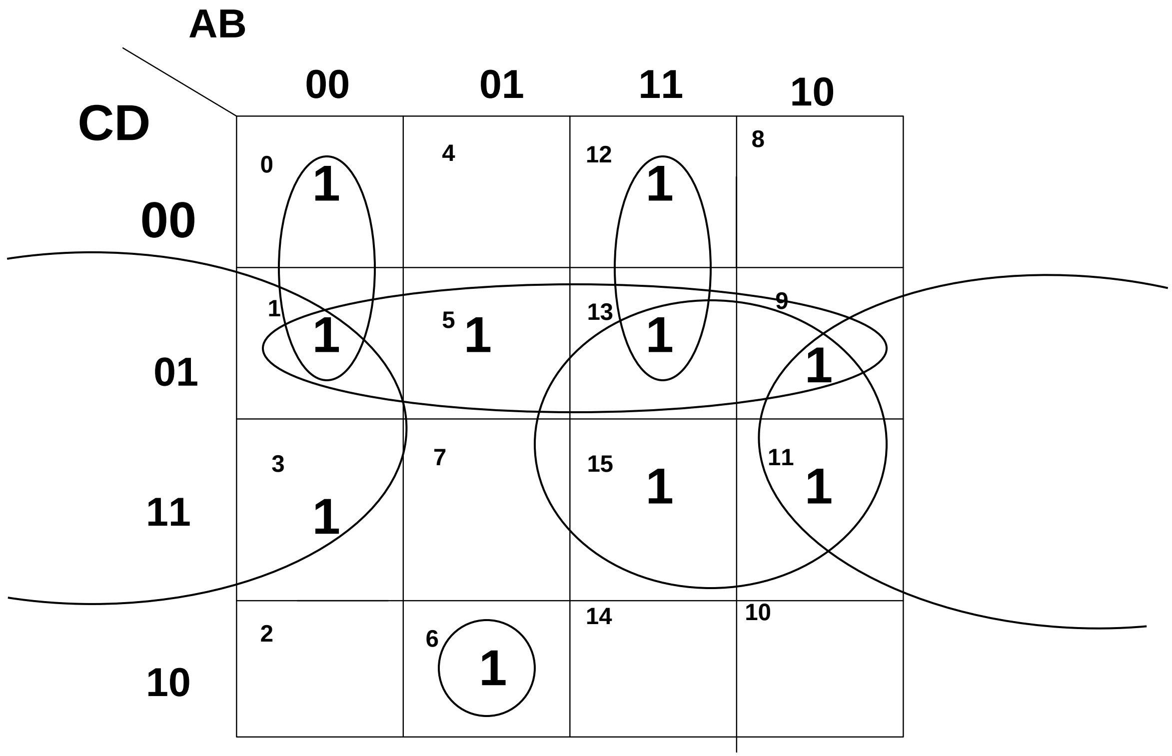
4. Repeat the procedure for groups of eight etc.

5. After the above procedure, if there still remains some uncovered boxes of 1's, they may be combined with each other or with already covered boxes in any manner.

Of course, we would want to include these left-over boxes in as few groupings as possible.

Ex. Minimize the logic expression

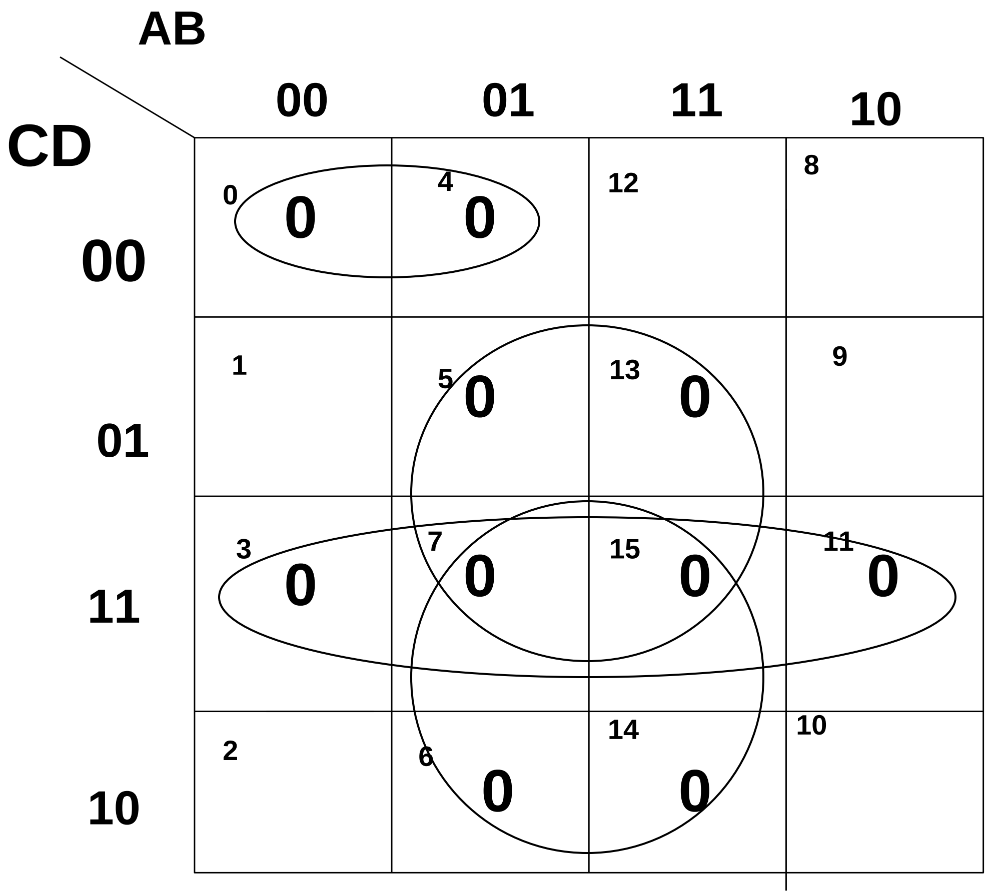
$$f = (A, B, C, D) = \sum m(0, 1, 3, 5, 6, 9, 11, 12, 13, 15)$$



$$f = (A, B, C, D) = \overline{A}BC\overline{D} + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + \overline{C}D + AD + \overline{B}D$$

Ex. Minimize the logic expression

$$f = (A,B,C,D) = \prod M(0,3,4,5,6,7,11,13,14,15)$$



$$f = (A,B,C,D) = (A + C + D)(\overline{B} + \overline{D})(\overline{C} + \overline{D})(\overline{B} + \overline{C})$$

Incompletely Specified Functions

(Functions with Don’t Care Conditions)

Sometimes, a function may be *specified incompletely*.

This may happen because of either of the following two reasons.

1) Sometimes, we simply do not care what value is assumed by the function for certain combinations of variables.

or

2) It may happen that we know that certain combinations of the variables will simply never occur. In this case, we do not care since the net effect is the same.

Ex. $f = (A, B, C, D) = \sum m(1,2,5,6,9) + d(10,11,12,13,14,15)$

d - don’t care – function f not specified for minterms m(10,11,12,13,14,15)

Representation in the K-Map

d (don’t care) – represented as x

CD \ AB	00	01	11	10
00	0	4	12 X	8
01	1 1	5 1	13 X	9 1
11	3	7	15 X	11 X
10	2 1	6 1	14 X	10 X

Minimization of logic function

i) Not using don’t cares

CD \ AB	00	01	11	10
00	0	4	12 X	8
01	1 1	5 1	13 X	9 1
11	3	7	15 X	11 X
10	2 1	6 1	14 X	10 X

$$f = (A,B,C,D) = \overline{A}\,\overline{C}\,D + \overline{B}\,\overline{C}\,D + \overline{A}\,C\,\overline{D}$$

i) Using don’t cares

Use only those don’t cares and assume as 1 which help in minimization. Ignore the other don’t cares

CD \ AB	00	01	11	10
00	0:	4:	12: X	8:
01	1: 1	5: 1	13: X	9: 1
11	3:	7:	15: X	11: X
10	2: 1	6: 1	14: X	10: X

$$f = (A,B,C,D) = \overline{C}\,D + C\,\overline{D}$$

Ex. Design a logic circuit which produces a 1 corresponding to a BCD input equal to or greater than 6.

Solution

Inputs				Output
A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

CD \ AB	00	01	11	10
00	0	4	12 X	8 1
01	1	5	13 X	9 1
11	3	7 1	15 X	11 X
10	2	6 1	14 X	10 X

$$f = (A, B, C, D) = A + BC$$

If we don’t use don’t cares,

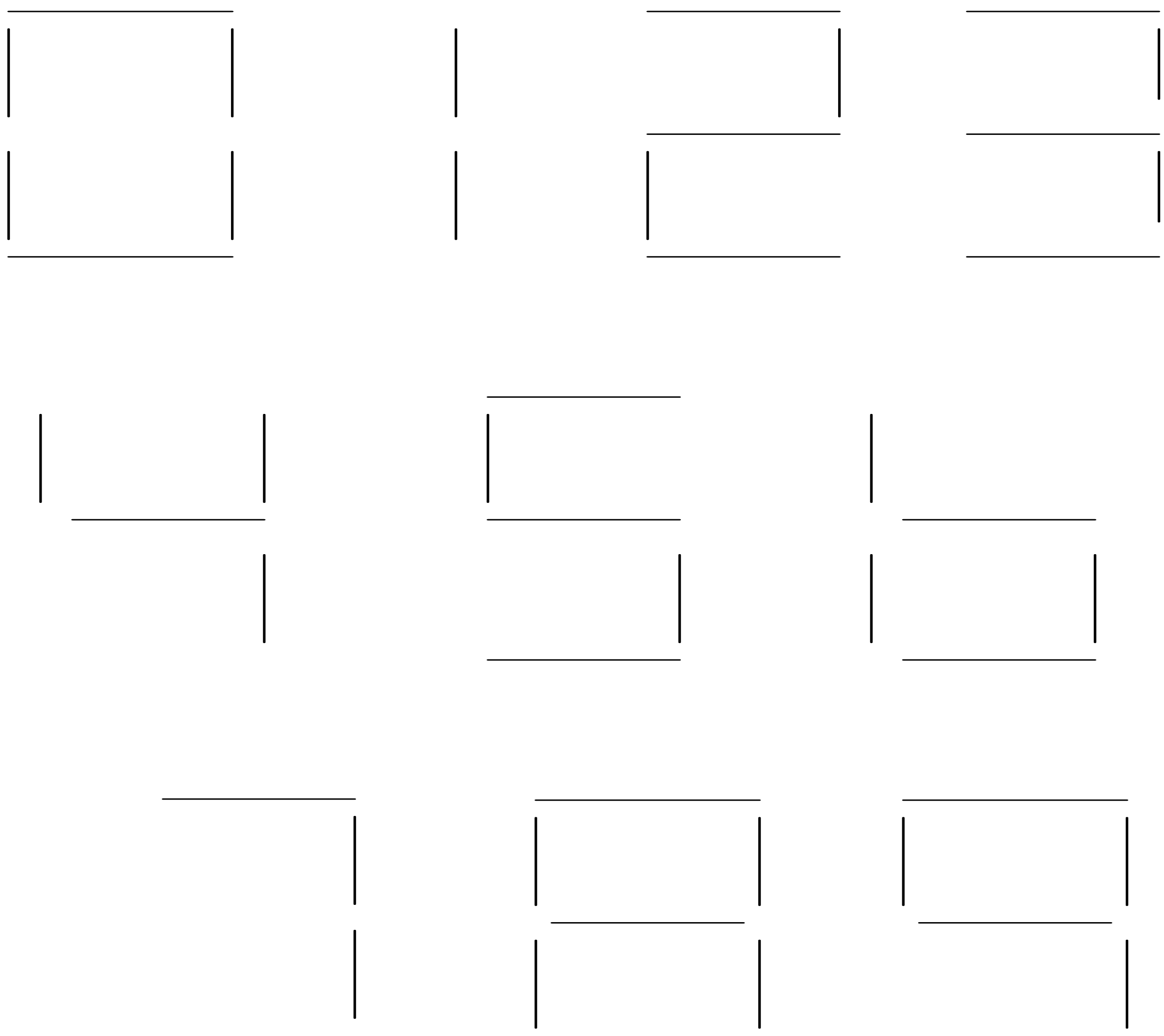
CD \ AB	00	01	11	10
00	0	4	12 X	8 1
01	1	5	13 X	9 1
11	3	7 1	15 X	11 X
10	2	6 1	14 X	10 X

$$f = (A, B, C, D) = A\,\overline{B}\,\overline{C} + \overline{A}\,B\,C$$

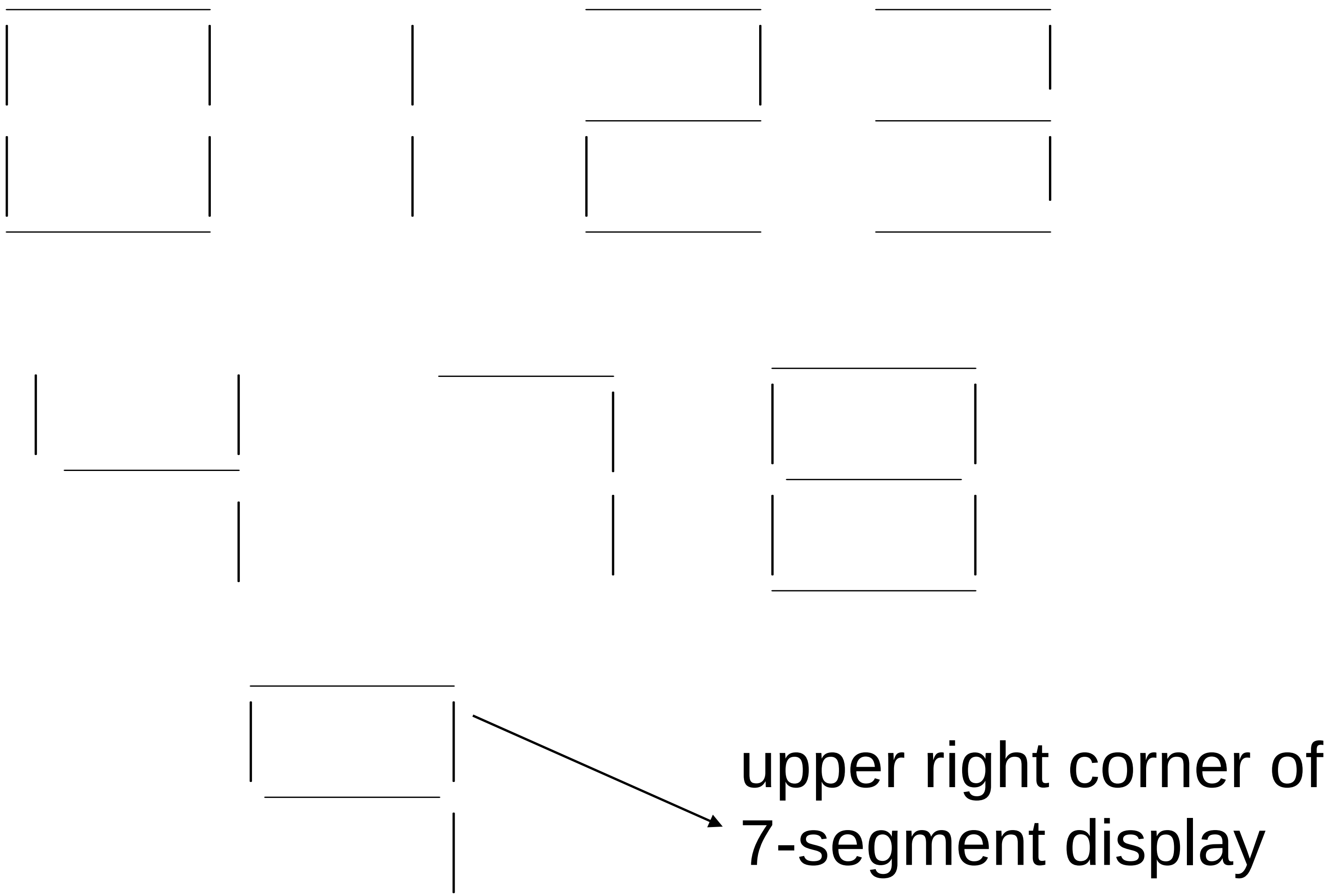
Ex. Design a logic circuit that produces a 1 when the 4-bit BCD input code translates to a decimal number that uses the upper right segment of a 7-segment display.

Solution

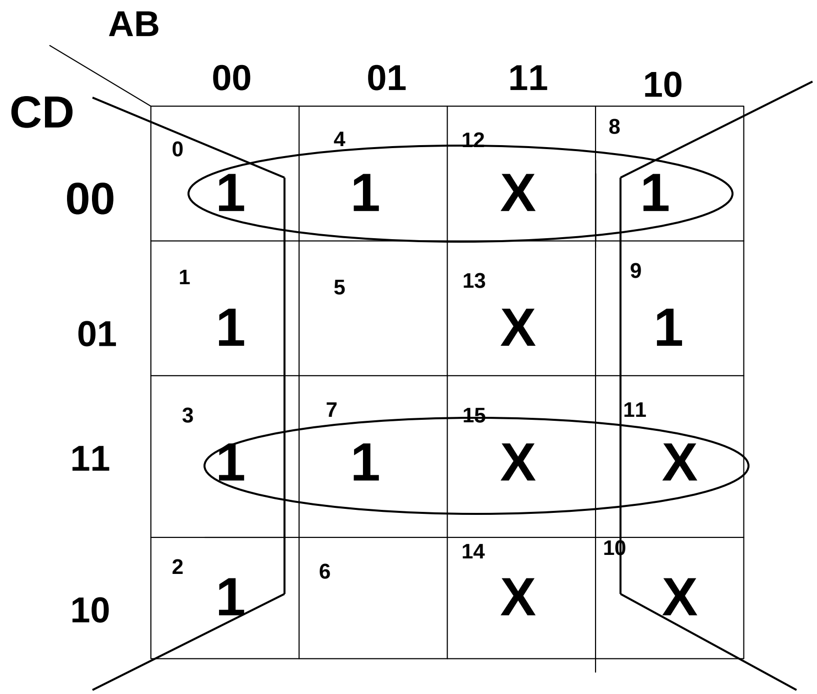
7- Segment display



Decimal numbers that use upper right corner of 7-segment display



$$\text{Ex.}\quad f = (A,B,C,D) = \sum m(0,1,2,3,4,7,8,9) + d(10,11,12,13,14,15)$$



$$\text{Ex.}\quad f = (A,B,C,D) = \overline{B} + \overline{C}\,\overline{D} + C\,D$$