MINIMIZATION OF SWITCHING FUNCTIONS

Dr. Chandan Karfa

CSE IIT Guwahati

- ***Source:***

Z. Kohavi and N. Jha, Switching and Finite Automata Theory, 3rd Ed., Cambridge University Press, 2010.

Minimization Objectives

The switching function is given as **SoP** or PoS form.

1. the minimum number of appearances of literals (recall that a *literal* is available in complemented or uncomplemented form);

2. the minimum number of literals in a sum-of-products (or product-of-sums) expression;

3. **the minimum number of terms in a sum-of-products expression, provided that there is no other such expression with the same number of terms and fewer literals.**

SoP and 3rd objectives

Minimization Objectives

- A combination of the first and second product terms yields $x'z'(y + y') = x'z'$. Similarly, combinations of the second and third, fourth and fifth, and fifth and sixth terms yield a reduced expression for f :

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

$$= x'z' + y'z' + yz + xz$$

- In general, a sum-of-products expression, from which no term or literal can be deleted without altering its logic value, is called an ***irredundant, or irreducible, expression***.

- if we combine the first and second terms of F, the third and sixth, and the fourth and fifth, we obtain the expression:

$$F(x, y, z) = x'z' + xy' + yz$$

- By combining the first and fourth terms, the second and third, and the fifth and sixth, we obtain a third irredundant expression

$$f(x, y, z) = x'y + y'z' + xz.$$

- *An irredundant expression is not necessarily minimal, nor is the minimal expression always unique*.

- It is, therefore, desirable to develop procedures for generating the set of all minimal expressions, so that the appropriate one may be selected according to other criteria

The Karnaugh Map Method

- Combining rule: Aa + Aa' = A
- K-map can take two forms Sum of Product (SOP) and Product of Sum (POS) according to the need of problem.
- K-map is table like representation, but it gives more information than TRUTH TABLE.
- Each n-variable map consists of 2^n cells (squares), representing all possible combinations of these variables.
- The **cyclic code**: As a result of this coding, cells that have a common side correspond to combinations that differ by the value of just a single variable.

- In general, two cells that differ in just one variable value are said to be *adjacent* and play a major role in the simplification process, because they may be combined by means of the rule *Aa* + *Aa'* = *A*.

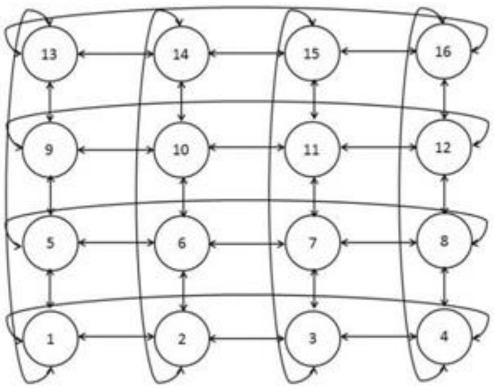
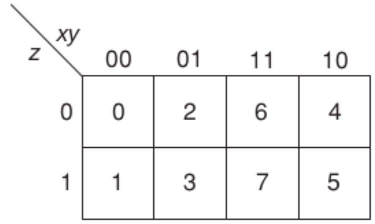


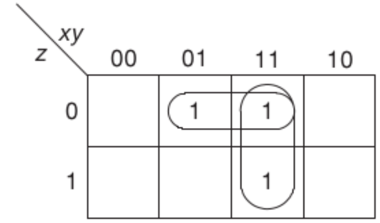
Figure 2. Torus Topology

STEPS TO SOLVE AN EXPRESSION USING K-MAP

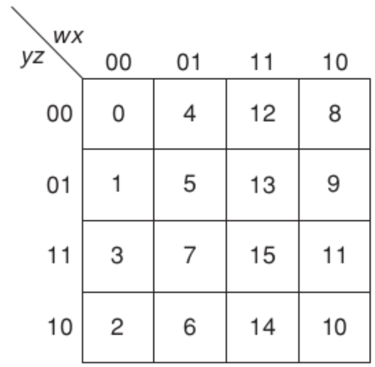
1. Select K-map according to the number of variables.
2. For SOP put 1's in blocks of K-map respective to the minterms (0's elsewhere)
3. For each minterm of *n* literals, there are *n* other minterms that have *n* – 1 literals in common with it, differing from it in just one literal.
4. Make rectangular groups containing total terms in power of two like 2,4,8 ..(except 1) and try to cover as many elements as you can in one group.



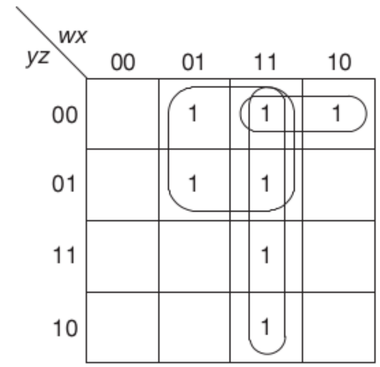
(*a*) Location of minterms in a three-variable map.



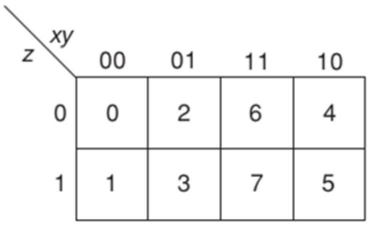
(*b*) Map for function $f(x, y, z) = \sum(2, 6, 7) = yz' + xy$.



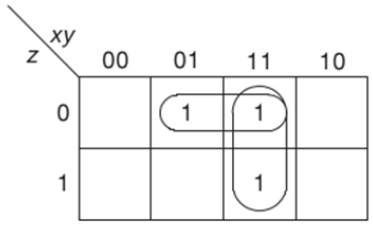
(*c*) Location of minterms in a four-variable map.



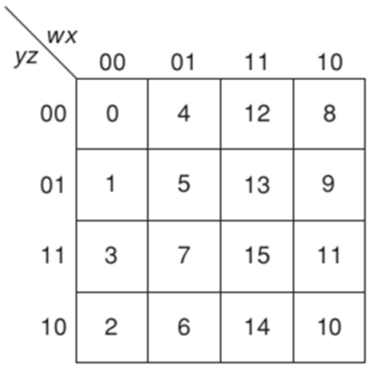
(*d*) Map for function $f(w, x, y, z) = \sum(4, 5, 8, 12, 13, 14, 15) = wx + xy' + wy'z'$.



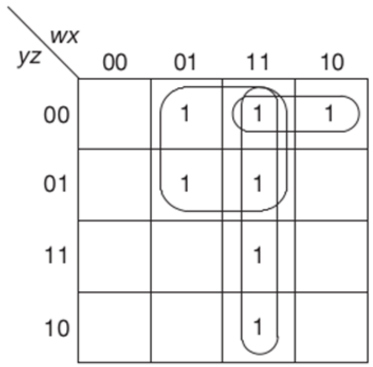
(a) Location of minterms in a three-variable map.



(b) Map for function $f(x, y, z) = \sum(2, 6, 7) = yz' + xy$.



(c) Location of minterms in a four-variable map.



(d) Map for function $f(w, x, y, z) = \sum(4, 5, 8, 12, 13, 14, 15) = wx + xy' + wy'z'$.

SIMPLIFICATION AND MINIMIZATION OF FUNCTIONS

- A collection of 2^m cells, each adjacent to m cells of the collection, is called a **cube** and the cube is said to cover these cells.
- Each cube can be expressed by a product containing n − m literals, where n is the number of variables on which the function depends.
- The *m* literals that are not contained in the product can be eliminated, because each of their 2^m combinations appear in the product with the same factor.
- The square array of four 1's in Fig.

$$\begin{aligned} w'xy'z' + w'xy'z + wxy'z' + wxy'z &= xy'(w'z' + w'z + wz' + wz) \\ &= xy'. \end{aligned}$$

- By the idempotent law any cell may be included in as many cubes as desired. (x + x = x)

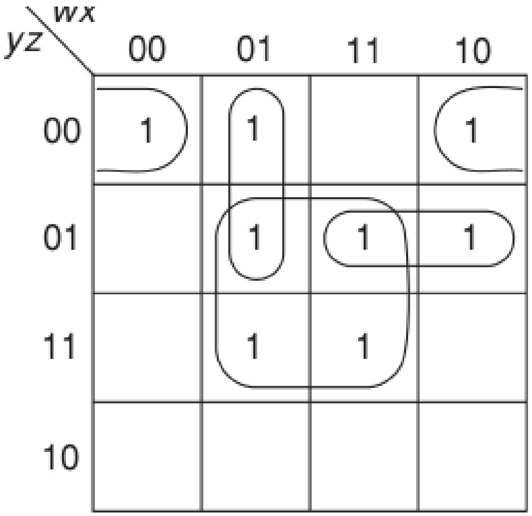
The Karnaugh Map Method

- we must cover all the 1-cells with the smallest number of cubes in such a way that each cube is as large as possible.
- a cube contained in a larger cube must never be selected.
- If there is more than one way of covering the map (i.e., its 1-cells) with the minimal number of cubes, we must select a covering that consists of larger cubes.
- Such a selection guarantees that the corresponding expression is indeed minimal and that no other expression containing the same number of terms, but fewer literals, exists

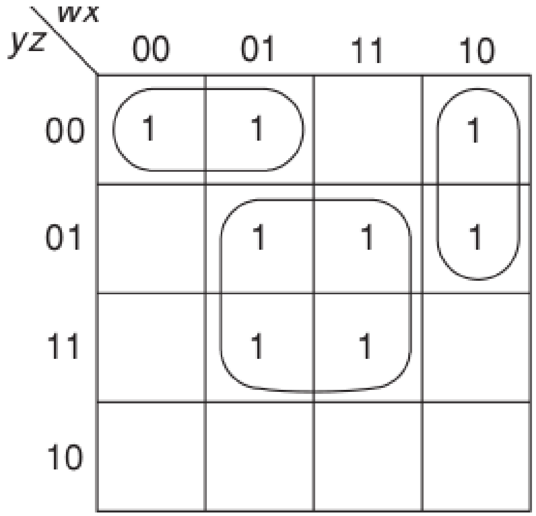
The Karnaugh Map Method

RULES FOR OBTAINING A SIMPLE EXPRESSION-

1. Start by covering with cubes those 1-cells that cannot be combined with any other 1-cell, and continue to those which have only a single adjacent 1-cell and thus can form cubes of only two 1-cells.
2. Next, combine those 1-cells that yield cubes of four but are not part of any cube of eight cells, and so on.
3. A minimal expression is one that corresponds to a collection of cubes that are as large and as few in number as possible, such that every 1-cell in the map of the function is covered by at least one cube.



(*a*) $f = x'y'z' + w'xy' + wy'z + xz$ is an irredundant expression.



(*b*) $f = w'y'z' + wx'y' + xz$ is the unique minimal expression.

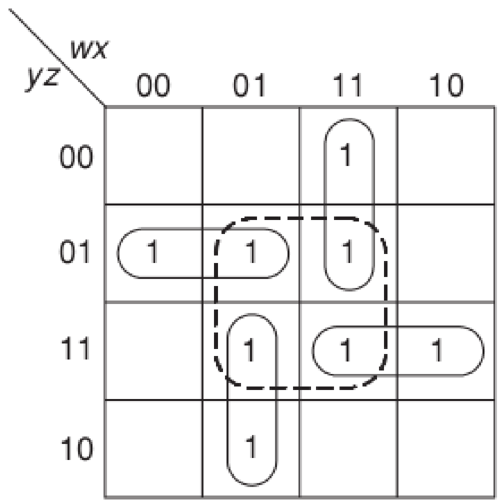
- Given a function f (w, x, y, z) = Σ(0, 4, 5, 7, 8, 9, 13, 15). We can obtain two irredundant expressions for this using k-map.

Question 1:

- There exist two more irredundant expressions for f in the last example, but neither of them is minimal. Identify them.

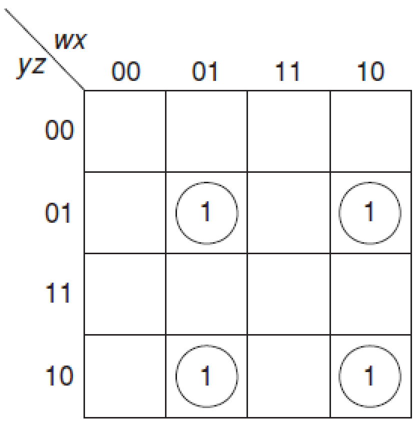
Example 2

- The function f(w, x, y, z) = **Σ**(1, 5, 6, 7, 11, 12, 13, 15) has only one irredundant form, as opposed to the preceding example. This unique minimal expression is derived from fig. and is found to be f=wxy'+ wyz + w'xy + w'y'z. Note that the dotted cube xz of four 1's becomes redundant if rule 1 is followed, since all its cells are covered by the other cubes.

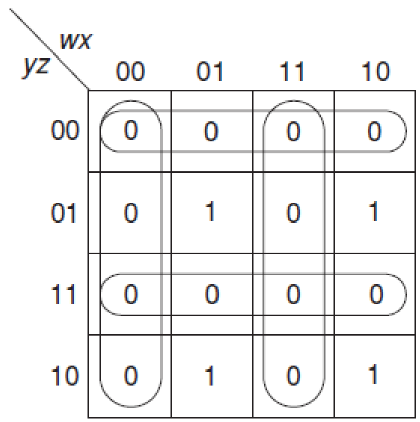


Determination of the minimal product of sums

- A variable corresponding to a 1 is complemented, and a variable corresponding to a 0 is uncomplemented.
- Cubes are formed of 0-cells instead of 1-cells and are selected in exactly the same manner as in the sum-of-products case.



(a) Map of $f(x, y, z) = \sum(5, 6, 9, 10)$
$= w'xy'z + wx'y'z + w'xyz' + wx'yz'$.



(b) Map of $f(x, y, z)$
$= \prod(0, 1, 2, 3, 4, 7, 8, 11, 12, 13, 14, 15)$
$= (y + z)(y' + z')(w + x)(w' + x')$.

Don’t-care combinations

- For certain input combinations the value of the output is unspecified, either because these input combinations are invalid or because the precise value of the output is of no importance
- Such situations may occur when the variables are not mutually independent. Combinations for which the value of **the function is not specified are called *don’t-care combinations.***
- The value of the function for such combinations is denoted by ϕ (or d).

Use don’t care to minimize functions:

- When employing the map of an incompletely specified function, we assign the value 1 to selected don’t-care combinations and the value 0 to others, in such a way as to increase the size of the selected cubes whenever possible.
- No cube containing only don’t-care cells can be formed, because it is not required that the function equal 1 for these combinations.

EXAMPLE-

<u>Design a code converter that converts BCD messages into Excess-3 code.</u>

- The converter has four input lines carrying signals labeled w, x, y, and z, and four output lines carrying signals f1 , f2 , f3 , and f4 .
- If the system operates properly then the input combinations will correspond to the decimal values 0 through 9, while the remaining six combinations, 10 through 15, will never occur and thus may be regarded as don't-care combinations.

Decimal	BCD inputs				Excess-3 outputs			
number	w	x	y	z	f_4	f_3	f_2	f_1
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

(*a*) Truth table for BCD and Excess-3 codes

$$f_1 = \sum(0, 2, 4, 6, 8) + \sum_\phi(10, 11, 12, 13, 14, 15)$$
$$f_2 = \sum(0, 3, 4, 7, 8) + \sum_\phi(10, 11, 12, 13, 14, 15)$$
$$f_3 = \sum(1, 2, 3, 4, 9) + \sum_\phi(10, 11, 12, 13, 14, 15)$$
$$f_4 = \sum(5, 6, 7, 8, 9) + \sum_\phi(10, 11, 12, 13, 14, 15)$$

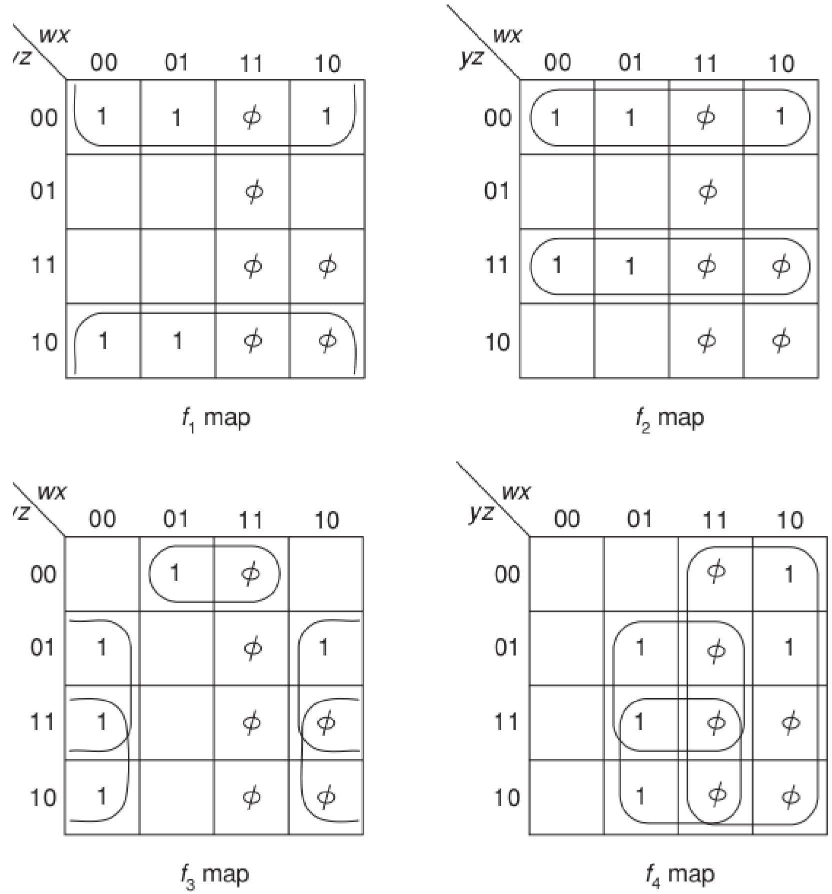
(*b*) Output functions

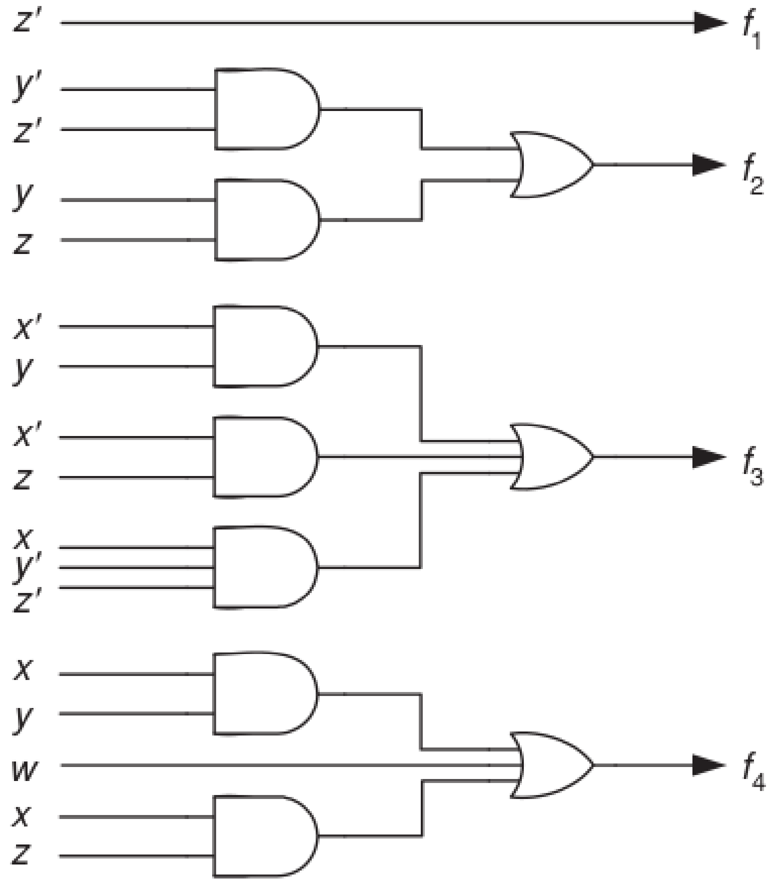
Decimal number	BCD inputs w	x	y	z	Excess-3 outputs f_4	f_3	f_2	f_1
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

(*a*) Truth table for BCD and Excess-3 codes

$$f_1 = \sum(0, 2, 4, 6, 8) + \sum_\phi(10, 11, 12, 13, 14, 15)$$
$$f_2 = \sum(0, 3, 4, 7, 8) + \sum_\phi(10, 11, 12, 13, 14, 15)$$
$$f_3 = \sum(1, 2, 3, 4, 9) + \sum_\phi(10, 11, 12, 13, 14, 15)$$
$$f_4 = \sum(5, 6, 7, 8, 9) + \sum_\phi(10, 11, 12, 13, 14, 15)$$

$$f_1 = z',$$
$$f_2 = y'z' + yz,$$
$$f_3 = x'y + x'z + xy'z',$$
$$f_4 = w + xy + xz.$$





if, owing to a malfunction in the message, an invalid input combination occurs then the output of the code converter will also be erroneous.

THE FIVE VARIABLE MAP

- A five-variable map contains $2^5 = 32$ cells.
- Each cell, in addition to being adjacent to four other cells, can be combined with a fifth cell on the other side of the center symmetry line.

yz \ vwx	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

Example With the aid of a map, minimize the function

$$f(v, w, x, y, z) = \sum(1, 2, 6, 7, 9, 13, 14, 15, 17, 22, 23, 25, 29, 30, 31).$$

From the cubes shown in Fig. 4.9, we obtain the minimal sum-of-products expression

$$f(v, w, x, y, z) = x'y'z + wxz + xy + v'w'yz'$$

yz \ vwx	000	001	011	010	110	111	101	100
00								
01	1		1	1	1	1		1
11		1	1			1	1	
10	1	1	1			1	1	

Question 2:

With the aid of a map, minimize the function
$f\,(v,w,x,y,z)$ =

SUM (0, 1, 2, 3, 7, 9, 13, 14, 15, 16, *18*, 21, 23, 25, 29, 31).

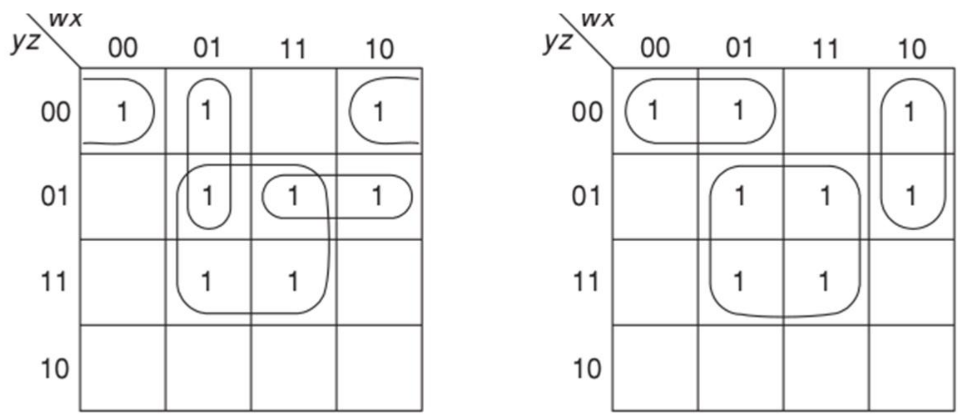
yz \ vwx	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

Prime implicants

- A switching function $f(x1, x2, ... , xn)$ is said to ***cover*** another function $g(x1, x2, . . . , xn)$, this action being denoted by $f \supseteq g$, if f assumes the value 1 whenever g does.
- If f covers h then h is said to ***imply*** f; h is said to be an implicant of f. The implication is often denoted by $h \rightarrow f$.
- If $f = wx + yz$ and $h = wxy$ then f covers h and h implies f.

Prime implicants

- A ***prime implicant*** p of a function f is a product term covered by f such that the delete on of any literal from p results in a new product that is not covered by f.
- p is a prime implicant if and only if p implies f but does not imply any product with fewer literals that in turn also implies f
- A prime implicant of $f = x'y + xz + y'z'$ is $x'y$, since it is covered by f and neither x' nor y alone implies f.



(a) $f = x'y'z' + w'xy' + wy'z + xz$ is an irredundant expression.

(b) $f = w'y'z' + wx'y' + xz$ is the unique minimal expression.

Minimization steps

- *Every irredundant sum-of-products equivalent to f is a union of prime implicants of f.*
- Generate the set of all prime implicants of *f* and from this set to select those prime implicants whose union yields a minimal expression for *f*.
  - *Use combining theorem* $Aa + Aa' = A$ to a pair of minterms repeatedly.
- Step1: Determination of the minimal expression is a systematic combination of terms.
- Step 2: Selecting the minimal set of prime implicants.

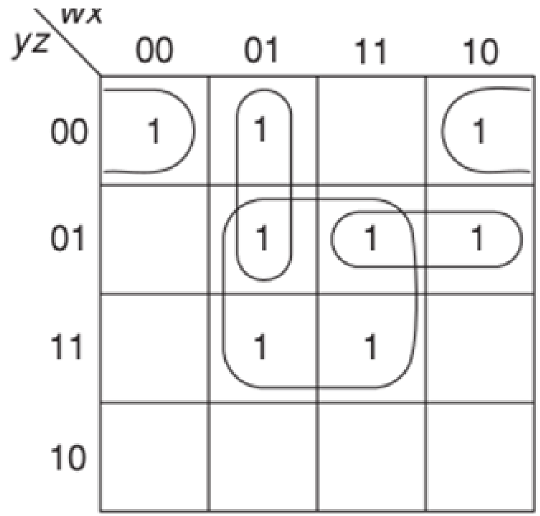
Prime implicants: example

$f(w, x, y, z) = (0, 4, 5, 7, 8, 9, 13, 15)$

The set of all prime implicants of f is

$P = \{xz, w'y'z, wx'y', x'y'z', w'xy', wy'z\}$

Note that xyz is not a prime implicant since it implies xz.

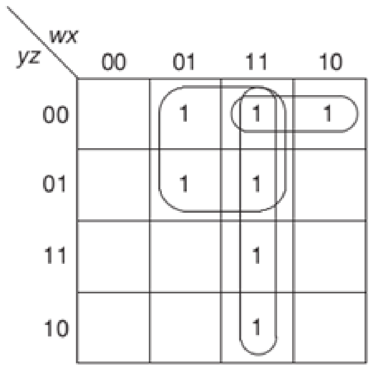


DERIVING MINIMAL EXPRESSIONS

- A prime implicant p of a function f is said to be an **essential prime implicant** if it covers at least one minterm of f that is not covered by any other prime implicant.
- Since every minterm of f must be covered by an expression for f, all essential prime implicants must be contained in any irredundant expression for this function.

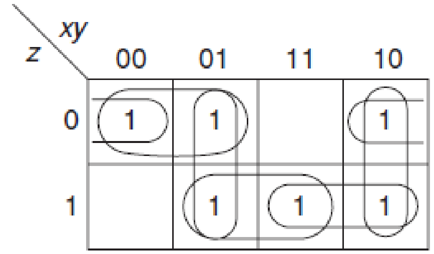
Examples

f (w, x, y, z) =**Σ**(4, 5, 8, 12, 13, 14, 15)



(*d*) Map for function $f(w, x, y, z)$
$= \sum(4, 5, 8, 12, 13, 14, 15) = wx + xy' + wy'z'$.

All are essential prime implicants



A map for the function $f(x, y, z) = \sum(0, 2, 3, 4, 5, 7)$.

The map for the function f (x, y, z) = Σ(0, 2, 3, 4, 5, 7), it is known as a cyclic prime implicant map since no prime implicant is essential, all prime implicants have the same size, and every cell is covered by exactly two prime implicants.

PROCEDURE FOR OBTAINING MINIMAL SOP

Determine all essential prime implicants and include them in the minimal expression.

Remove from the list of prime implicants all those that are covered by the essential prime implicants.

If the set derived in step 1 covers all the minterms of f then it is the unique minimal expression.

Otherwise, select additional prime implicants such that f is covered completely and such that the total number and size of the prime implicants thus added are minimal.

The tabulation procedure for the determination of prime implicants

- The Karnaugh map method described in the preceding sections is very useful for functions of up to six variables.
- In order to manipulate functions of a larger number of variables a more systematic procedure, preferably one that can be carried out by a computer, is necessary.
- The tabulation procedure, also known as the Quine–McCluskey method of reduction, satisfies the above requirements.
- It is suitable for hand computation and is also easily programmable.

Basic idea

- Two k-variable terms can be combined into a single $(k-1)$-variable term if and only if they have in common $k-1$ identical literals and differ in just a single literal.
- The combined term consists of the product of the $k-1$ identical literals while the variable, which is uncomplemented in one term and complemented in the other, is deleted.
- Two minterms are combined if they differ in one position only.
- **Need:** determine, in a simple and systematic way, which terms can (or cannot) be combined and to carry out all possible such combinations.
- If we consider the **binary representation of the minterms,** we observe that the necessary and sufficient condition for two minterms to be combinable is that their binary representations differ **in just one position.**
- To facilitate the combination process the minterms are arranged in groups according to the number of 1's in their binary representation.

Quine-McClusky Method

Arrange all minterms in groups, such that all terms in the same group have the same number of 1’s in their binary representation. Start with the least number of 1’s and continue with groups of increasing numbers of 1’s. The number of 1’s in a term is called the index of that term.

Compare every term of the lowest-index group with each term in the successive group; whenever possible, combine the two terms being compared by means of the combining theorem Aa + Aa' = A. Repeat this by comparing each term in a group of index i with every term in the group of index i+1 until all possible applications of the combining theorem have been exhausted.

Now compare the terms generated in step 2, in the same fashion: a new term is generated by combining two terms that differ by only a single 1 and whose dashes are in the same position. The process continues until no further combinations are possible. The remaining unchecked terms constitute the set of prime implicants of the function.

Step 1

	w	x	y	z	
0	0	0	0	0	✓
1	0	0	0	1	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

Step 2

	w	x	y	z	
0, 1	0	0	0	–	✓
0, 2	0	0	–	0	✓
0, 8	–	0	0	0	✓
1, 5	0	–	0	1	✓
1, 9	–	0	0	1	✓
2, 10	–	0	1	0	✓
8, 9	1	0	0	–	✓
8, 10	1	0	–	0	✓
5, 7	0	1	–	1	✓
5, 13	–	1	0	1	✓
9, 13	1	–	0	1	✓
7, 15	–	1	1	1	✓
13, 15	1	1	–	1	✓

Step 3

	w	x	y	z	
0, 1, 8, 9	–	0	0	–	*A*
0, 2, 8, 10	–	0	–	0	*B*
1, 5, 9, 13	–	–	0	1	*C*
5, 7, 13, 15	–	1	–	1	*D*

EXAMPLE

Determination of the set of prime implicants for the function f2 (w, x, y, z) = **Σ**(0, 1, 2, 5, 7, 8, 9, 10,13, 15).

The entire procedure is a mechanized process for combining and reducing all adjacent pairs of terms. The unchecked terms are the prime implicants of f, since each implies f and is not covered by any other term with fewer literals.

Step 1

	w	*x*	*y*	*z*	
0	0	0	0	0	✓
1	0	0	0	1	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

Step 2

	w	*x*	*y*	*z*	
0, 1	0	0	0	–	✓
0, 2	0	0	–	0	✓
0, 8	–	0	0	0	✓
1, 5	0	–	0	1	✓
1, 9	–	0	0	1	✓
2, 10	–	0	1	0	✓
8, 9	1	0	0	–	✓
8, 10	1	0	–	0	✓
5, 7	0	1	–	1	✓
5, 13	–	1	0	1	✓
9, 13	1	–	0	1	✓
7,15	–	1	1	1	✓
13, 15	1	1	–	1	✓

Step 3

	w	*x*	*y*	*z*	
0, 1, 8, 9	–	0	0	–	*A*
0, 2, 8, 10	–	0	–	0	*B*
1, 5, 9, 13	–	–	0	1	*C*
5, 7, 13, 15	–	1	–	1	*D*

Contd...

- The left-hand part of previous fig corresponding to the application of step 1, consists of all minterms, arranged in groups of increasing indices.
- The reduced terms, after the first application of step 2, are given in the center part. For example, the combination of the terms 0000 and 0001 is recorded by writing 000– in its first row, where the dash indicates that variable z is redundant.
- The terms 0000 and 0001 in the left-hand part of the figure are now checked off.
- The same rule is applied repeatedly until all combinable terms are recorded in the center part.

000-
100-

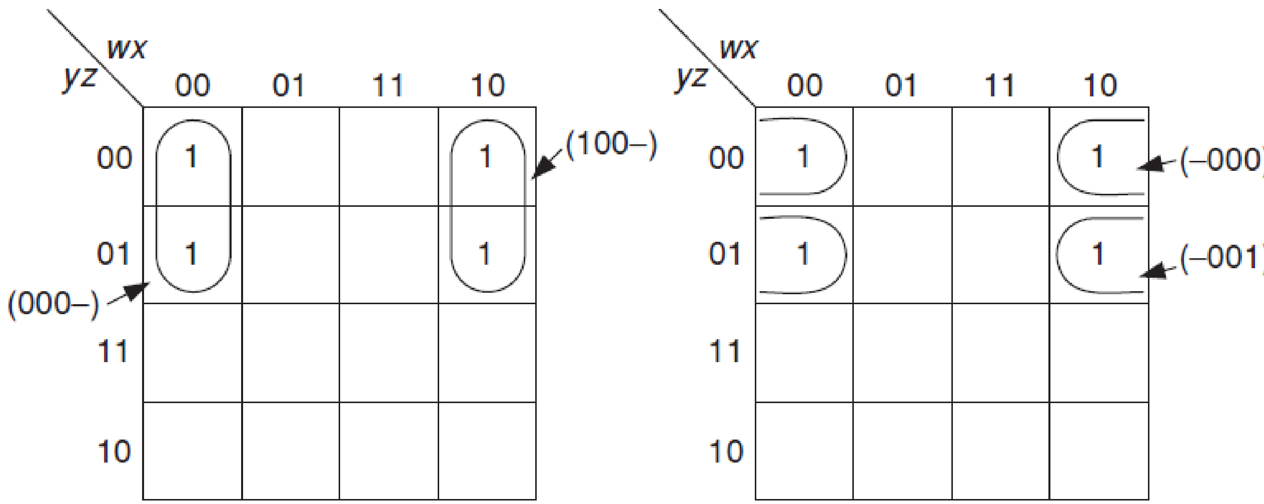
-00-

-000
-001

-00-

Each term is generating two ways: what does it indicates?

Every four cell cube can be formed by combining two adjacent two cells cubes in two ways



Decimal Representation

- Two minterms can be combined only if they differ by a power of 2, that is, only if the difference between their decimal codes is 2^i.
- The combined term consists of the same literals as the minterms with the exception of the variable whose weight is 2^i, which is deleted.
- Two terms whose codes differ by a power of 2 but which have the same index cannot be combined
- if a term with a smaller index has a higher decimal value than another term whose index is higher, then the two terms cannot be combined although they may differ by a power of 2

1	✓
2	✓
12	✓
17	✓
18	✓
20	✓
24	✓
13	✓
19	✓
21	✓
25	✓
15	✓
23	✓
27	✓
29	✓
31	✓

(*a*)

1, 17	(16)	*H*
2, 18	(16)	*G*
12, 13	(1)	*F*
17, 19	(2)	✓
17, 21	(4)	✓
17, 25	(8)	✓
18, 19	(1)	*E*
20, 21	(1)	*D*
24, 25	(1)	*C*
13, 15	(2)	✓
13, 29	(16)	✓
19, 23	(4)	✓
19, 27	(8)	✓
21, 23	(2)	✓
21, 29	(8)	✓
25, 27	(2)	✓
25, 29	(4)	✓
15, 31	(16)	✓
23, 31	(8)	✓
27, 31	(4)	✓
29, 31	(2)	✓

(*b*)

17, 19, 21, 23	(2, 4)	✓
17, 19, 25, 27	(2, 8)	✓
17, 21, 25, 29	(4, 8)	✓
13, 15, 29, 31	(2,16)	*B*
19, 23, 27, 31	(4, 8)	✓
21, 23, 29, 31	(2, 8)	✓
25, 27, 29, 31	(2, 4)	✓

(*c*)

17, 19, 21, 23, 25, 27, 29, 31 (2, 4, 8) *A*

(*d*)

The prime implicant chart

- The prime implicant chart displays pictorially the covering relationships between the prime implicants and minterms of a function.
- It consists of an array of u columns and v rows, where u and v designate the number of minterms for which the function takes on the value 1 and the number of prime implicants, respectively.
- The entries of the i_{th} row in the chart consist of x's placed at its intersections with columns corresponding to minterms covered by the i_{th} prime implicant.

	✓		✓	✓	✓	✓		✓	✓	✓
	0	1	2	5	7	8	9	10	13	15
$A = x'y'$	×	×				×	×			
✓ $B = x'z'$	×		×			×		×		
$C = y'z$		×		×			×		×	
✓ $D = xz$				×	×				×	×

	w	x	y	z	
0, 1, 8, 9	–	0	0	–	A
0, 2, 8, 10	–	0	–	0	B
1, 5, 9, 13	–	–	0	1	C
5, 7, 13, 15	–	1	–	1	D

Minimization Objectives

- The problem now is *to select a minimal subset of prime implicants such that each column contains at least one* $\times$ *in the rows corresponding to the selected subset and the total number of literals in the prime implicants selected is as small as possible.*

ESSENTIAL PRIME IMPLICANT

- If any column contains just a single × then the prime implicant corresponding to the row in which this × appears is essential and consequently must be included in any irredundant expression for f .
- The × is circled, and a check mark is placed next to the essential prime implicant.
- The row that corresponds to an essential prime implicant is referred to as an essential row.
- Once an essential prime implicant has been selected, all the minterms it covers are checked off.

▪ If, after all essential prime implicants and their corresponding columns have been checked, the entire function is covered, i.e., every column is checked off, then the union of all essential prime implicants yields the minimal expression.

▪ If this is not the case then additional prime implicants are necessary.

Example

- essential prime implicant *B* covers, in addition to columns 2 and Consequently columns 0, 2, 8, and 10 are checked off.
- The two essential prime implicants *B* and *D* of *f*2 cover all the minterms except 1 and 9. These minterms may be covered by either prime implicant *A* or *C*, and since both are expressed with the same number of literals, we obtain

	✓ 0	1	✓ 2	✓ 5	✓ 7	✓ 8	9	✓ 10	✓ 13	✓ 15
$A = x'y'$	×	×				×	×			
✓ $B = x'z'$	×		⊗			×		⊗		
$C = y'z$		×		×			×		×	
✓ $D = xz$				×	⊗				×	⊗

$$f_2(w, x, y, z) = x'z' + xz + x'y'$$

$$f_2(w, x, y, z) = x'z' + xz + y'z.$$

Don’t-care combinations

- During the process of generating the set of prime implicants, don’t-care combinations are regarded as true combinations, that is, combinations for which the function assumes value 1.
  - This, in effect, increases to the maximum the number of possible prime implicants.
- The don’t-care terms are, however, not considered in the next step, that of selecting a minimal set of prime implicants.

$$f_3(v, w, x, y, z) = \sum(13, 15, 17, 18, 19, 20, 21, 23, 25, 27, 29, 31) + \sum_{\phi}(1, 2, 12, 24)$$

Question 4:

- Find all the prime implicants of the following functions using the binary representations of the minterms using Quine-McClusky method.

$$f_3(v, w, x, y, z) = \sum(13, 15, 17, 18, 19, 20, 21, 23, 25, 27, 29, 31) + \sum_{\phi}(1, 2, 12, 24)$$

Don’t-care combinations

- Don’t-care minterms need not be listed as column headings in the prime implicant chart, since they do not have to be covered by the minimal expression.
- We actually leave the specification of the don’t-care terms open; that is, if a minimal expression contains a prime implicant derived from a don’t-care combination, this amounts to specifying that combination as 1; otherwise, the don’t-care combination is, in effect, assigned the value 0.

	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
	13	15	17	18	19	20	21	23	25	27	29	31
✓ $A = vz$			×		×		×	⊗	×	⊗	×	×
✓ $B = wxz$	×	⊗									×	×
$C = vwx'y'$									×			
✓ $D = vw'xy'$						⊗	×					
$E = vw'x'y$				×	×							
$F = v'wxy'$	×											
$G = w'x'yz'$				×								
$H = w'x'y'z$			×									

Selection of nonessential prime implicants

- The selection of nonessential prime implicants is facilitated by the initial listing of prime implicants in a descending order, according to the number of minterms they cover. Thus, prime implicants that are located in a higher group in the chart are expressed with fewer literals than those located in a lower group.
- A horizontal line across the chart separates one group from the other.
- Essential prime implicants: *A*, *B*, and *D*. TO select 18, select *E or G (*since both have the same number of literals)

	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
	13	15	17	18	19	20	21	23	25	27	29	31
✓ $A = vz$			×		×		×	⊗	×	⊗	×	×
✓ $B = wxz$	×	⊗									×	×
$C = vwx'y'$									×			
✓ $D = vw'xy'$						⊗	×					
$E = vw'x'y$				×	×							
$F = v'wxy'$	×											
$G = w'x'yz'$				×								
$H = w'x'y'z$			×									

$$f_3(v, w, x, y, z) = vz + wxz + vw'xy' + vw'x'y$$

$$f_3(v, w, x, y, z) = vz + wxz + vw'xy' + w'x'yz'.$$

Question 5:

- Find all the prime implicants of the following functions using the binary representations of the minterms using Quine-McClusky method.

$$f_4(v, w, x, y, z) = \sum(0, 1, 3, 4, 7, 13, 15, 19, 20, 22, 23, 29, 31).$$

Reduced chart

- While every irredundant expression must contain the essential prime implicants A and C, none may contain B, since B covers only terms already covered by A and C.
- The reduced chart, which results after the removal of rows A, B, and C and all columns covered by them is shown in fig
- f(v, w, x, y, z) = **Σ**(0, 1, 3, 4, 7, 13, 15, 19, 20, 22, 23, 29, 31)

	0	1	✓ 3	4	✓ 7	✓ 13	✓ 15	✓ 19	20	22	✓ 23	✓ 29	✓ 31
✓ $A = wxz$						×	×					×	×
$B = xyz$					×		×				×		×
✓ $C = w'yz$			×		×			×			×		
$D = vw'xy$										×	×		
$E = vw'xz'$									×	×			
$F = w'xy'z'$				×					×				
$G = v'w'x'z$		×	×										
$H = v'w'y'z'$	×			×									
$I = v'w'x'y'$	×	×											

(*a*) Prime implicant chart.

	0	1	4	20	22
D					×
E				×	×
F			×	×	
G		×			
H	×		×		
I	×	×			

Determination of the set of all irredundant expressions

- Associate a two-valued variable with each remaining prime implicants.
- The truth value of such a variable is 1 if the corresponding prime implicant is included in the irredundant expression, and is 0 if it is not.
- Define a *prime implicant function p* to be equal to 1 if each column is covered by at least one of the chosen prime implicants and 0 if it is not.
- For example, column 0 can be covered by either row *H* or row *I*. Consequently, either *H* or *I* must be included in any irredundant expression

Determination of the set of all irredundant expressions

we obtain the expression for p,

p = (H + I)(G + I)(F + H)(E + F)(D + E), which can also be written as a sum of products,

p = EHI + EFI + DFI + EGH + DFGH.

- For *p,* at least three rows are needed to cover the reduced chart,
  - for example rows *E*, *H*, and *I*, or rows *E*, *F*, and *I* .
- Four out of five irredundant expressions for *f*4 result in minimal expressions:

$$f_4(v, w, x, y, z) = wxz + w'yz + vw'xz' + v'w'y'z' + v'w'x'y',$$
$$f_4(v, w, x, y, z) = wxz + w'yz + vw'xz' + w'xy'z' + v'w'x'y',$$
$$f_4(v, w, x, y, z) = wxz + w'yz + vw'xy + w'xy'z' + v'w'x'y',$$
$$f_4(v, w, x, y, z) = wxz + w'yz + vw'xz' + v'w'x'z + v'w'y'z'.$$

	0	1	4	20	22
D					×
E				×	×
F			×	×	
G		×			
H	×		×		
I	×	×			

Reduction of the Chart: Deletion of Rows

- A row *U* of a prime implicant chart is said to *dominate* another row *V* of that chart if *U* covers every column covered by *V*.
- *If row U dominates row V and the prime implicant corresponding to row U does not have more literals than the prime implicant corresponding to row V, then row V can be deleted from the chart*

	✓1	✓3	✓4	✓5	✓6	✓7	10	11	✓12	✓13	✓14	✓15	18	19	✓20	✓21	✓22	✓23	✓25	26	✓27
✓$A = w'x$			×	×	×	×									⊗	⊗	×	×			
✓$B = v'x$			×	×	×	×			⊗	⊗	×	×									
$C = vx'y$													×	×						×	×
$D = vw'y$													×	×			×	×			
$E = wx'y$							×	×												×	×
$F = v'wy$							×	×			×	×									
$G = x'yz$		×						×						×							×
$H = w'yz$		×				×								×				×			
$I = v'yz$		×				×		×				×									
✓$J = v'w'z$	⊗	×		×		×															
✓$K = vwx'z$																			⊗		×

(*a*) Prime implicant chart.

	10	11	18	19	26
C			×	×	×
D			×	×	
E	×	×			×
F	×	×			
G		×		×	
H				×	
I		×			

Reduced prime implicant cha

	✓10	✓11	✓18	✓19	✓26
✓*C*			⊗	×	×
✓*E*	⊗	×			×
G		×		×	

(*c*) Final chart.

Minimal Expression: A + B + J + K + C + E

Reduction of the Chart: Deletion of Columns

- A column *i* in a prime implicant chart is said to *dominate* another column *j* of that chart if *i* has an × in every row in which *j* has an ×.
- Clearly, any minimal expression derived from a chart which contains both columns *i* and *j* can be derived from a chart which contains only the dominated column i.
- *if column i dominates column j, then column i can be deleted from the chart without affecting the search for a minimal expression.*
- *Column 11 dominates over 10. So, 11 can be deleted.*

	10	11	18	19	26
C			×	×	×
D			×	×	
E	×	×			×
F	×	×			
G		×		×	
H				×	
I		×			

Reduced prime implicant cha

- Reducing columns the *dominating* ones are removed, while of the rows the *dominated* ones are deleted.
- The removal of dominated rows and dominating columns may alternate a number of times

The Branching method

- It may happen that a prime implicant chart has no essential prime implicants, dominated rows, or dominating columns. Whenever this happens, a different approach must be taken, called the *branching method*
- where each prime implicant covers two minterms and each minterm is covered by two prime implicants. Such a chart is called a *cyclic* prime implicant chart

$$\breve{f}_6 = \sum(0, 1, 5, 7, 8, 10, 14, 15)$$

yz \ wx	00	01	11	10
00	1			1
01	1	1		
11		1	1	
10			1	1

(*a*) Cyclic map.

	0	1	5	7	8	10	14	15
$A = w'x'y'$	×	×						
$B = w'y'z$		×	×					
$C = w'xz$			×	×				
$D = xyz$				×				×
$E = wxy$							×	×
$F = wyz'$						×	×	
$G = wx'z'$					×	×		
$H = x'y'z'$	×				×			

(*b*) Cyclic prime implicant chart.

The Branching method

- It is necessary to make an arbitrary selection of one row and then apply the above reduction procedure. Let select row A.
- The row *B* is dominated by row *C* and row *H* is dominated by row *G*.
  - Remove B and H.
- Rows *C* and *G* must be selected they are now essential.
  - This selection, in turn, implies the inclusion of row *E* in

	0	1	5	7	8	10	14	15
$A = w'x'y'$	×	×						
$B = w'y'z$		×	×					
$C = w'xz$			×	×				
$D = xyz$				×				×
$E = wxy$							×	×
$F = wyz'$						×	×	
$G = wx'z'$					×	×		
$H = x'y'z'$	×				×			

(*b*) Cyclic prime implicant chart.

	5	7	8	10	14	15
B	×					
C	×	×				
D		×				×
E					×	×
F				×	×	
G			×	×		
H			×			

(*c*) Reduced chart after selection of row *A*.

F = A + C + G + E

The Branching method

- In general, there is no guarantee that the initial arbitrary selection will result in a minimal expression. It is, therefore, necessary to repeat the process for each row that could be substituted for the initially selected one.
- The column 0 can be covered by either row *A* or *H*. Consequently, one of these rows must be included in any minimal expression.
- The entire process must now be repeated for row *H* as the initial selection

	1	5	7	10	14	15
A	×					
B	×	×				
C		×	×			
D			×			×
E					×	×
F				×	×	
G				×		

(*d*) Reduced chart after selection of row *H*.

F = B + D + F + H

The Branching method

- The prime implicant chart of a function whose map is cyclic is itself always cyclic.
- It is possible to encounter cyclic charts in the process of reducing a prime implicant chart that corresponds to a noncyclic map.
- Moreover, a cyclic chart may result while applying the branching process and reducing another cyclic chart.
- Whenever such a situation occurs, another arbitrary row selection must be made and all alternative expressions must be obtained, such that a minimal one may be selected.

Can you automate the Quine-McClusky Method?

Heuristic two-level circuit minimization

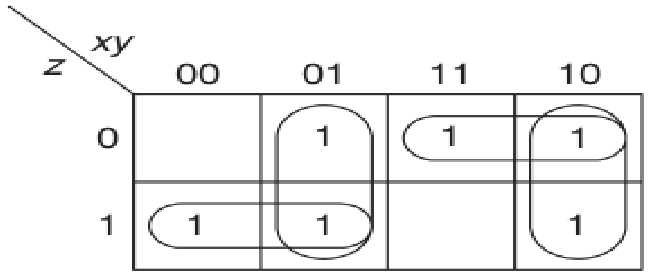
- The prime implicant chart method requires that first all prime implicants are found and then a minimal subset of these prime implicants that covers all the minterms of the function is chosen.
- If more than one subset is of minimal cardinality then the one with fewest literals is chosen.
- The problem with this approach is that it may become impractical for many functions of interest.
- One reason is that for an n-variable function, the number of prime implicants can be as large as $3^n/n$.
- Heuristic two-level circuit minimization tries to alleviate the above problem by reducing the number of prime implicants that need to be tackled.
- A very successful computer-aided design tool that encapsulates this approach is called ESPRESSO.

STEPS IN ESPRESSO

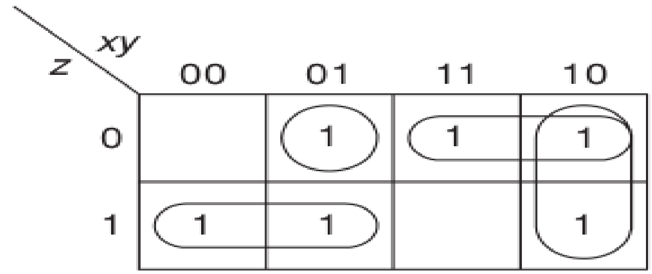
There are three main steps in ESPRESSO: expand, reduce and irredundant.

- The expand step targets implicants and expands them into prime implicants. Any implicants that are now covered by the expanded prime implicant are omitted from any further consideration.
- The reduce step transforms the prime implicants into implicants of the least possible size such that all the minterms of the function are still covered.This makes the implementation suboptimal but may lead to better solutions later.
- The irredundant step chooses a minimal subset of the prime implicants obtained so far such that the subset covers all the minterms of the function. This is like prime implicant chart covering. However, since the number of prime implicants targeted is usually much smaller, the process is not as time consuming.

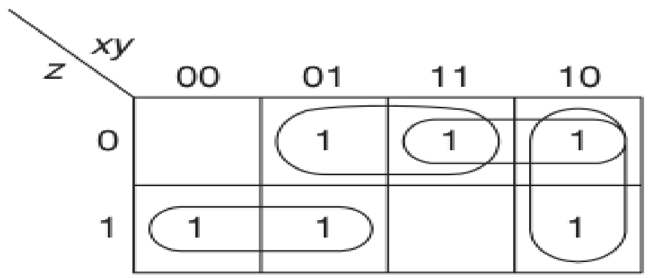
EXAMPLE



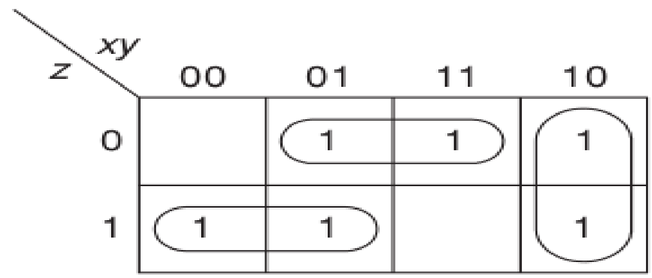
(*a*) Initial covering of *f*.



(*b*) After the reduce step.



(*c*) After the expand step.



(*d*) After the irredundant step.

EXAMPLE Contd..

- Consider the initial set of prime implicants, shown in Fig a, that covers all the minterms of function f .
- Such a set could be obtained by applying expand and irredundant steps to the initial set of minterms. Suppose that the prime implicant x'y is now reduced to the implicant x'yz',as shown in Fig b.
- When the implicant x'yz' is now expanded in another direction, the prime implicant yz' is obtained, as shown in Fig.c.
- The prime implicant xz' can now be removed in the irredundant step since its minterms are covered by the remaining prime implicants, thus obtaining the covering of minterms shown in Fig d.
- This corresponds to the minimal sum-of-products x'z+yz'+ xy'. This expression is obviously superior to the original expression, x'z+x'y+ xz'+ xy'.

Contd..

x	y	z	f
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1

expand and irredundant $\Rightarrow$

x	y	z	f
0	–	1	1
0	1	–	1
1	–	0	1
1	0	–	1

reduce $\Rightarrow$

x	y	z	f
0	–	1	1
0	1	0	1
1	–	0	1
1	0	–	1

expand $\Rightarrow$

x	y	z	f
0	–	1	1
–	1	0	1
1	–	0	1
1	0	–	1

irredundant $\Rightarrow$

x	y	z	f
0	–	1	1
–	1	0	1
1	0	–	1

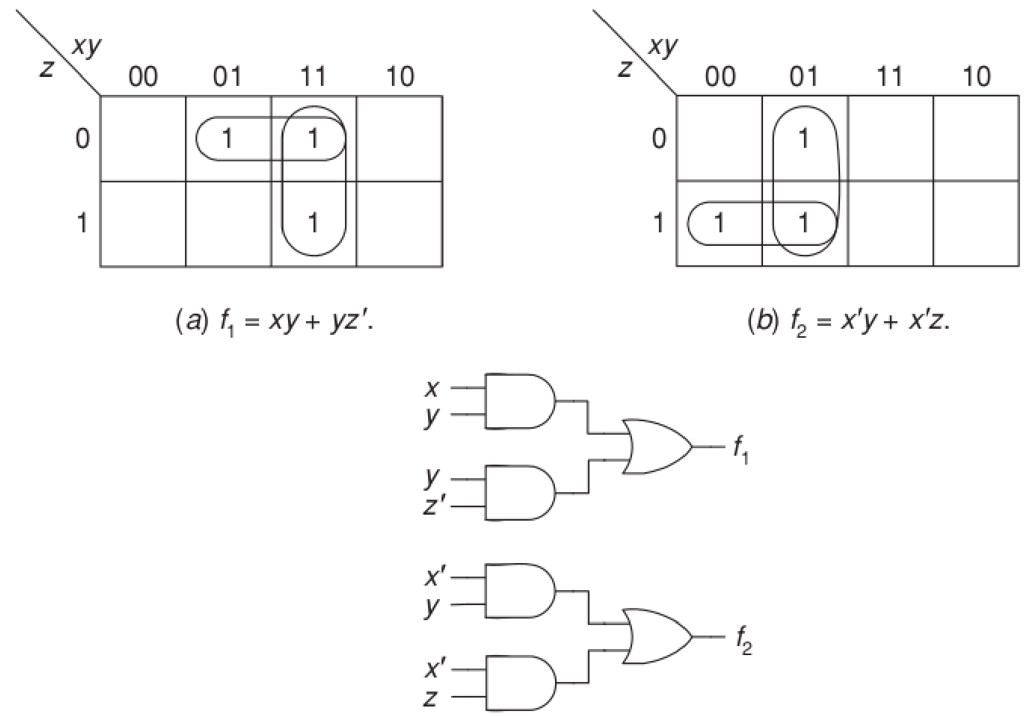
- The input to ESPRESSO is typically an encoded truth table Truth tables equivalent to the set of transformations performed in the example above are shown in Fig.
- The set of minterms of function f is subjected to expand and irredundant steps to obtain the initial covering containing the prime implicants x'z, x'y, xz' and xy'.
- Then, the reduction of prime implicant x'y to the implicant x'yz is depicted by the transformation of 01– to 010.
- The expansion step converts 010 to –10. Finally, the irredundant step eliminates 1–0.

Multi-output two-level circuit minimization

- In general, most circuits that we might want to design have multiple outputs.
- A trivial way to deal with an n-output circuit is to treat it as n single-output circuits and minimize them separately.

EXAMPLE

Consider the functions f1 and f2 shown in Fig. and the prime implicants shown in the maps. Since all four prime implicants are essential, the corresponding two-level circuit can be derived as shown in the figure.

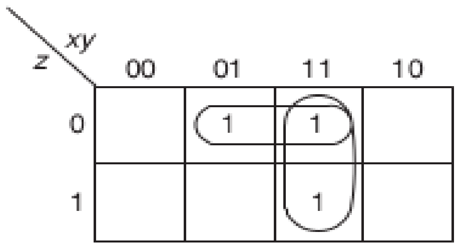


(a) $f_1 = xy + yz'$.

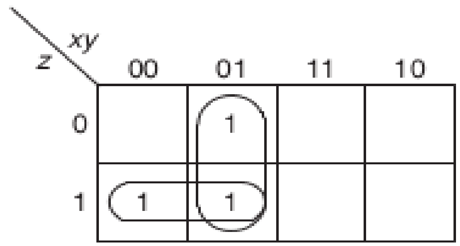
(b) $f_2 = x'y + x'z$.

(c) Two-level implementation.

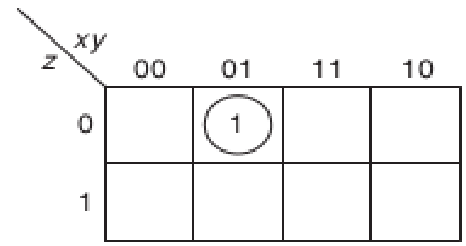
EXAMPLE 2



(*a*) f_1.



(*b*) f_2.



(*c*) $f_1 f_2$.

Function	Prime implicant	f_1 2	f_1 ✓6	f_1 ✓7	f_2 ✓1	f_2 2	f_2 ✓3
f_1	✓A = xy		×	×			
	B = yz'	×	×				
f_2	C = $x'y$					×	×
	✓D = $x'z$				×		×
$f_1 f_2$	E = $x'yz'$	×				×	

(*d*) Augmented prime implicant chart.

Function	Prime implicant	f_1 2	f_2 2
f_1	B = yz'	×	
f_2	C = $x'y$		×
$f_1 f_2$	E = $x'yz'$	×	×

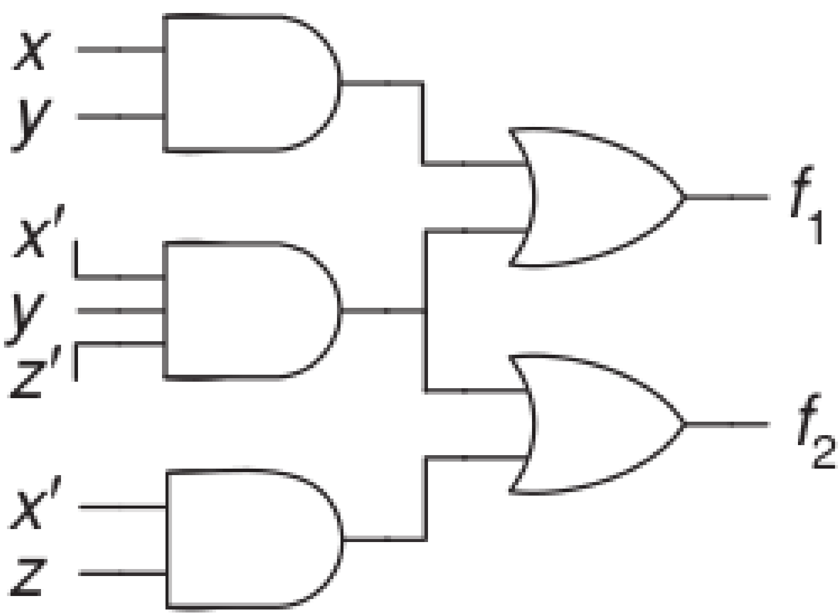
(*e*) Reduced chart.

EXAMPLE 2

- The previous approach, however, can be suboptimal. The reason is that it does not exploit the possibility of sharing logic among different outputs.
- Consider the functions f1 and f2 shown in previous Fig.
- Since none of the prime implicants of f1 and f2 is also a prime implicant of f1 f2 , all five multi-output prime implicants shown in these maps deserve further consideration.
- The augmented prime implicant chart is shown in Fig.
- The essential prime implicants and the minterms they cover are then checked.
- This leads to the reduced chart shown in Fig.
- Assuming that we are interested in minimizing the number of gates as a primary objective and the number of interconnections as a secondary objective, we cannot use the concept of dominated rows to reduce this chart further.
- Thus, we can use the prime implicant function p to resolve the situation as follows:

$$p = (B + E)(C + E) = BC + E.$$

Contd..



The minimum gate implementation contains AND gates realizing multi-output prime implicants A, D, and E in the first level.

The complete implementation is shown in Fig.