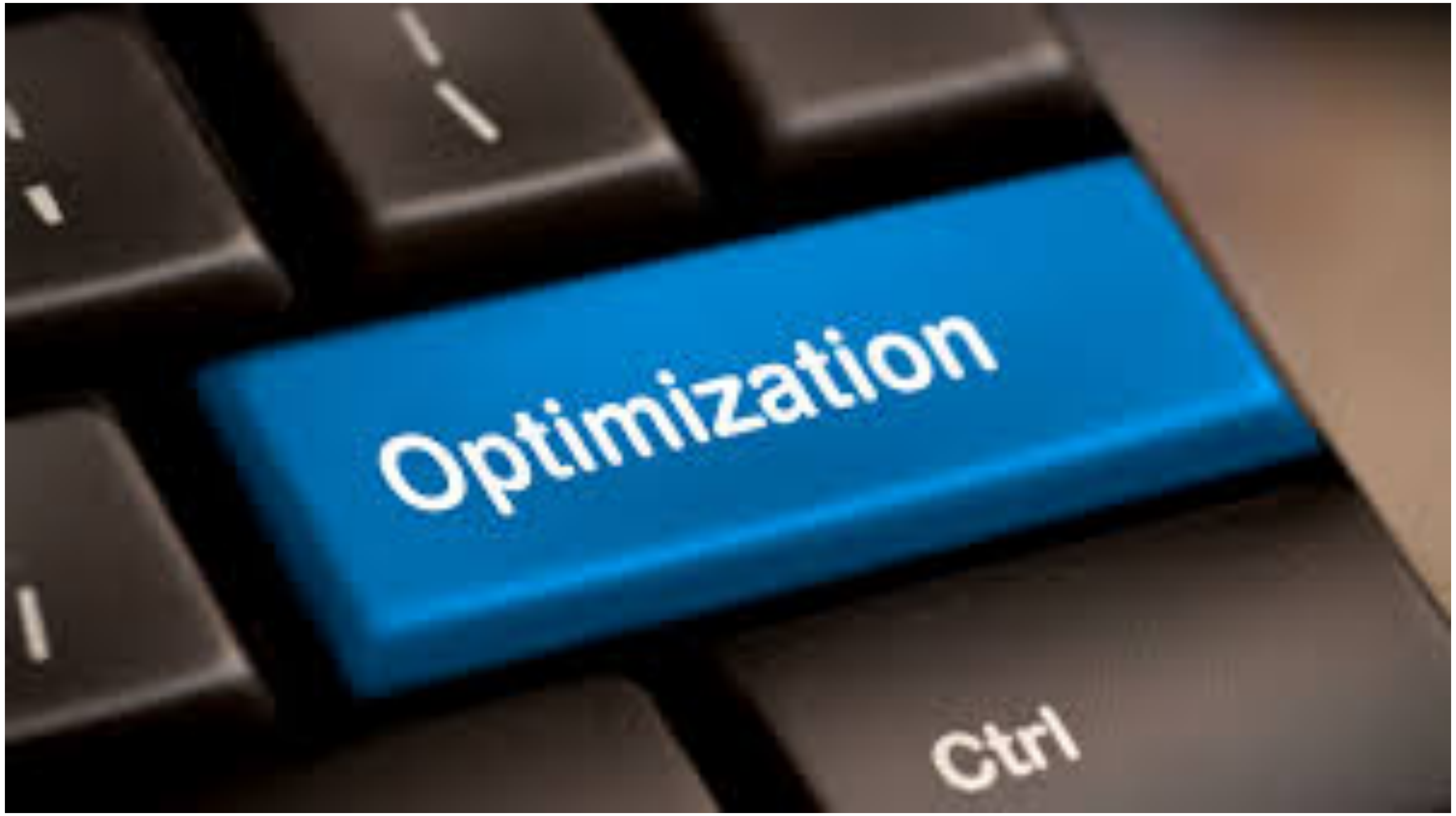
Digital Logic Design + Computer Architecture

Sayandeep Saha

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Logic Minimization

Life of an Engineer



Logic Minimization: Why?

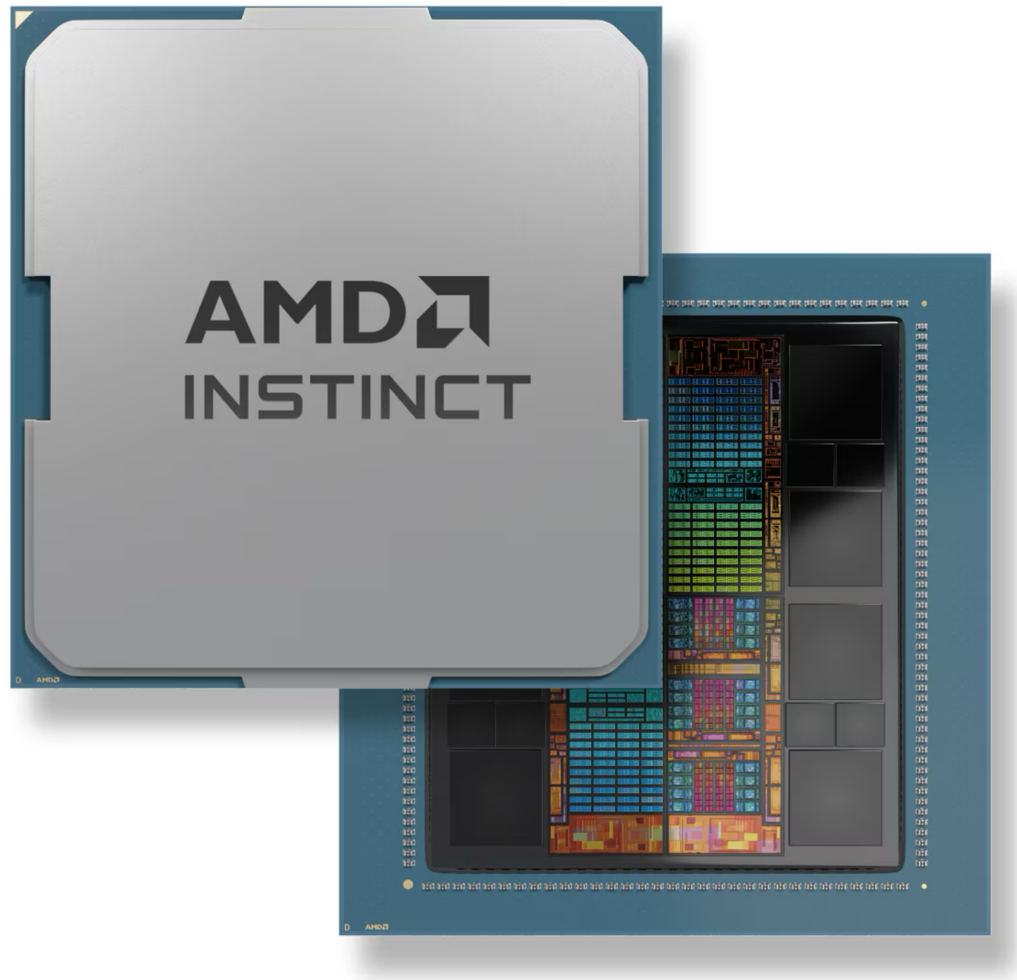
- Consider a switching expression. **How many gates do you need to implement this?** Consider each gate is 2-input, 1 output except the NOTs — 5 OR, 12 ANDs, 3 NOTs

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

- Now consider the following expression:' $x'z' + y'z' + yz + xz$
- Observe that both implements the same logic function!!! Now you need 4 ORs, 4 ANDs, and 3 NOTs.
- Can you do better?? — Yes $f(x, y, z) = x'z' + xy' + yz$
- Turns out that there can be more such expressions.
- Lower gate count => Lower transistor count => Lower area (and perhaps less power, and time)…
- So, now we have an engineering problem in hand — **how to minimize the switching expressions???**

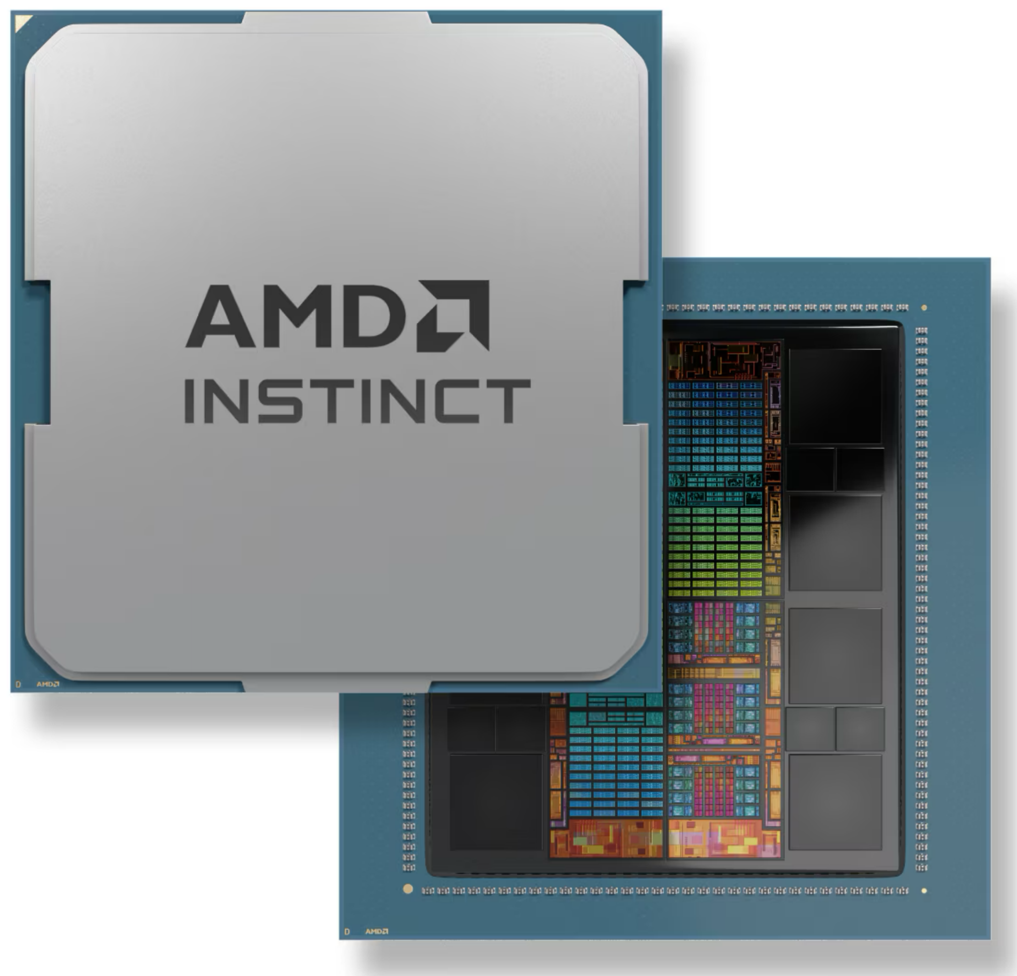
Bigger Picture

- Modern circuits contains billions of gates — e.g. AMD Instinct is a GPU processor containing 146,000,000,000 transistors; so a few billions of gates (if not trillions)…
- How do people minimized their gate network…Fortunately we have tools for that.
- Today we will be studying some of the fundamental techniques behind these tools.



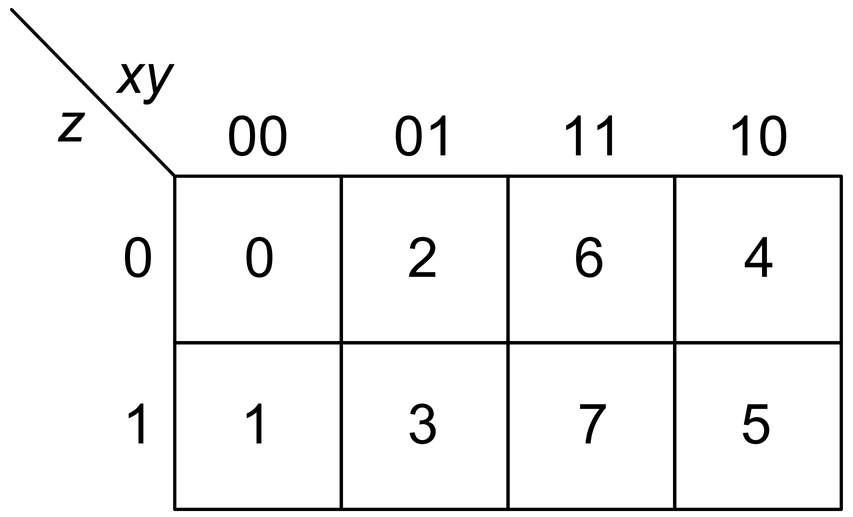
Bigger Picture

- Modern circuits contains billions of gates — e.g. AMD Instinct is a GPU processor containing 146,000,000,000 transistors; so a few billions of gates (if not trillions)…
- How do people minimized their gate network…Fortunately we have tools for that.
- Today we will be studying some of the fundamental techniques behind these tools.
  - Of course, a very very rudimentary intro

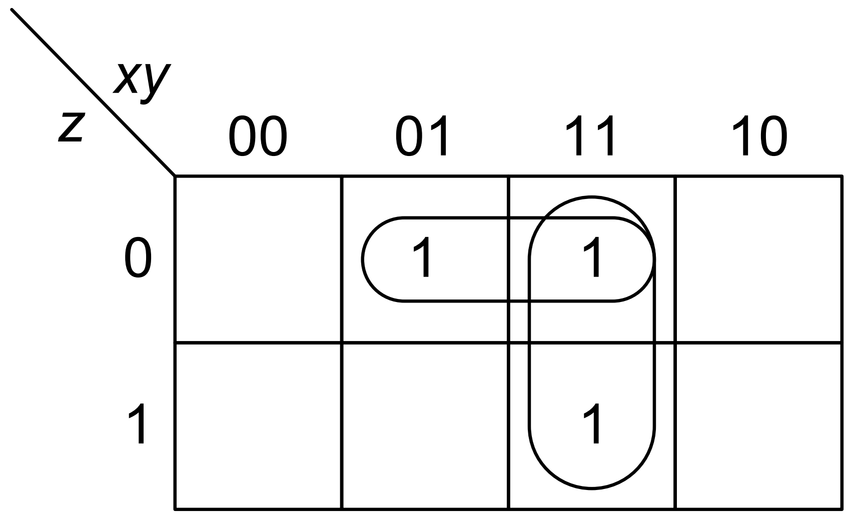


The Map Method

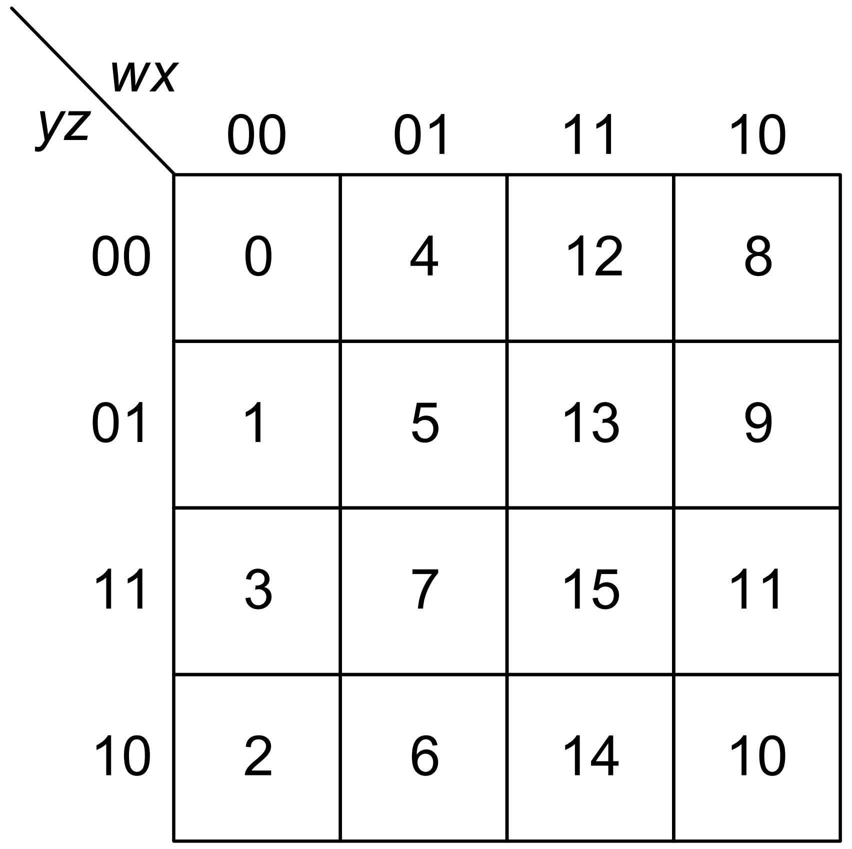
- **Karnaugh map**: modified form of truth table
- Combine terms using the **Aa + Aa’ = A (combining theorem)**



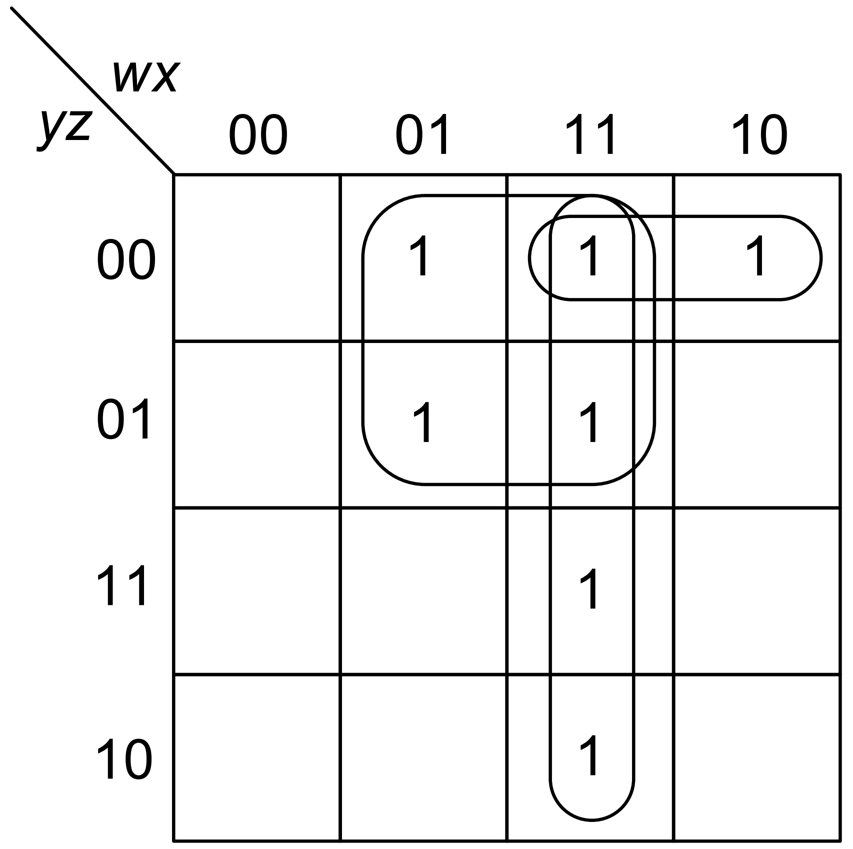
(a) Location of minterms in a three-variable map.



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



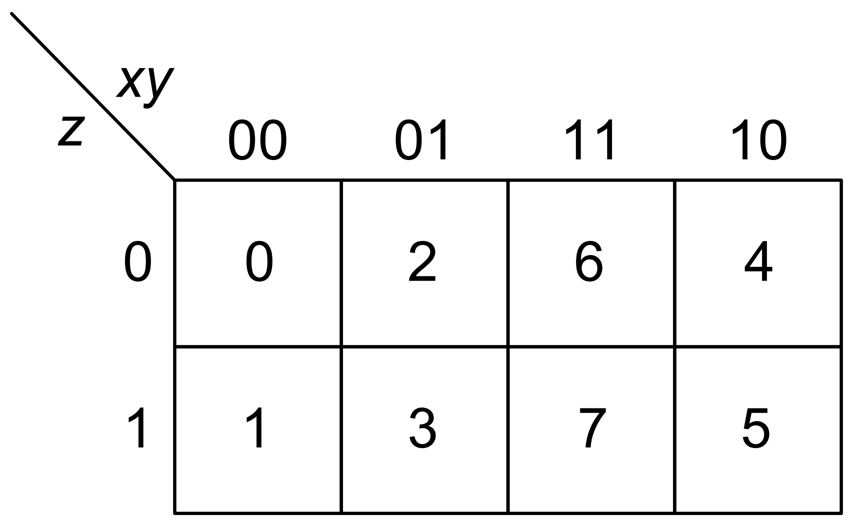
(c) Location of minterms in a four-variable map.



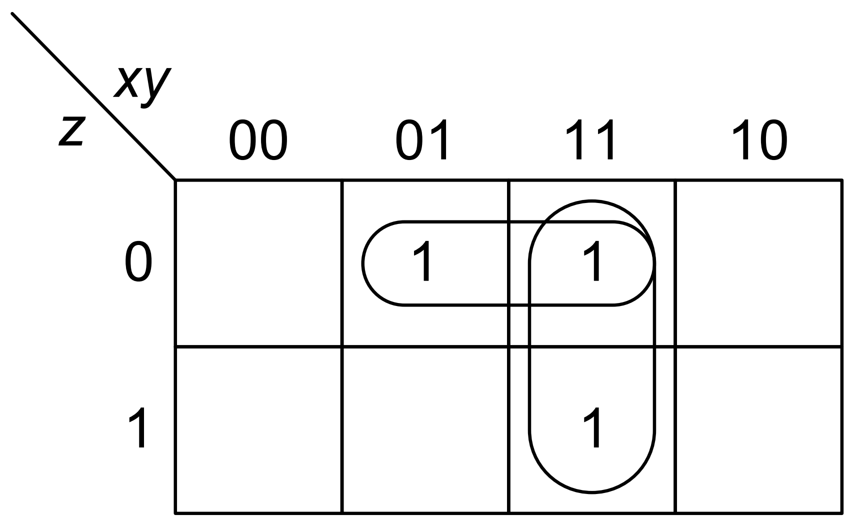
(d) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

The Map Method

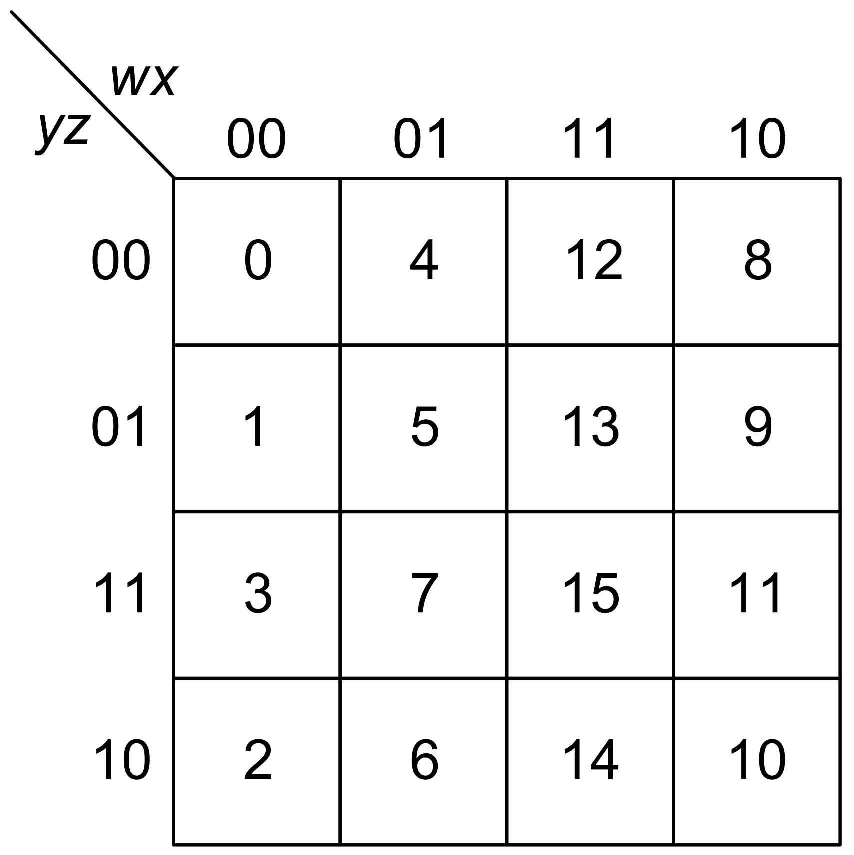
- **Karnaugh map**: modified form of truth table
- Combine terms using the **Aa + Aa' = A (combining theorem)**
- **Cube**:
  - Collection of $\mathbf{2}^{m}$ cells, each adjacent to ***m*** cells of the collection
  - The cube is said to **cover** the cells it is involved with
  - Expressed by a product of **n-m literals** for a function containing **n variables**
  - **m literals** not in the product said to be eliminated



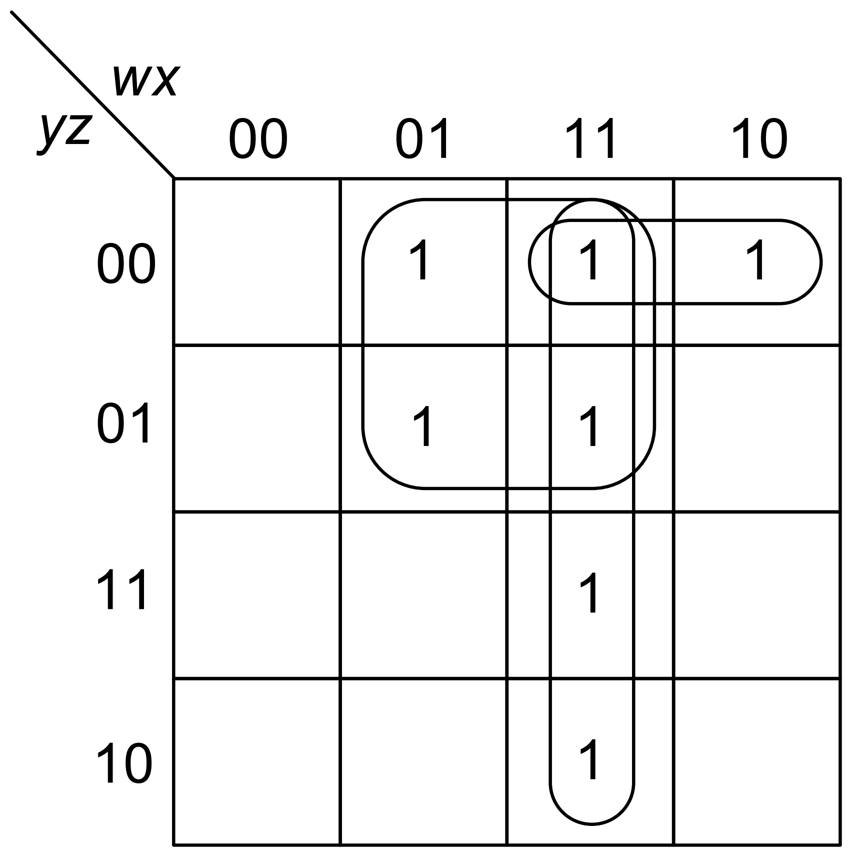
(*a*) Location of minterms in a three-variable map.



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



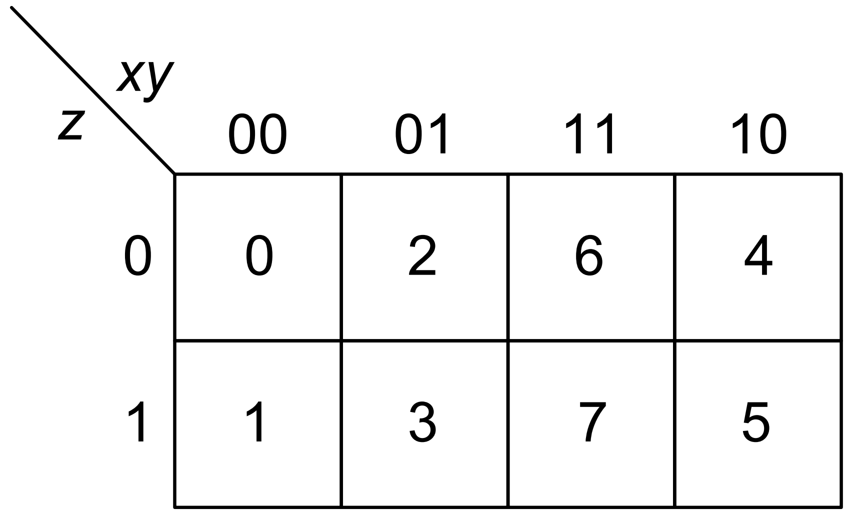
(*c*) Location of minterms in a four-variable map.



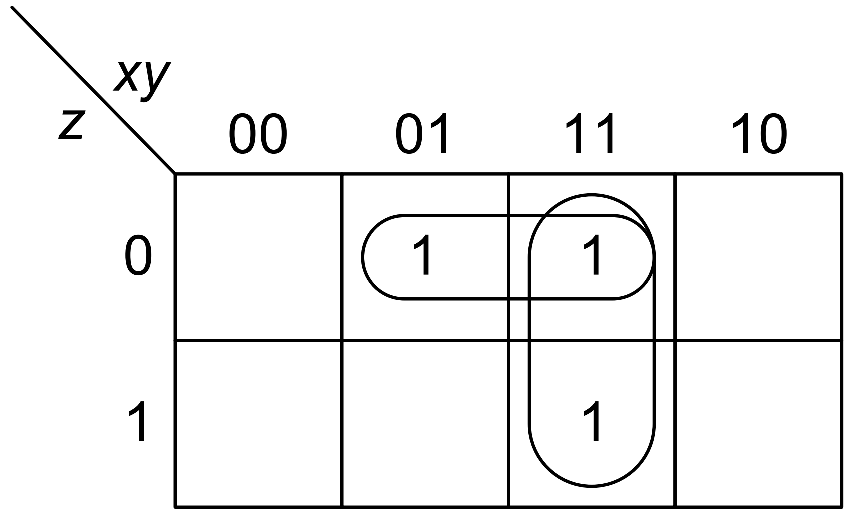
(*d*) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

The Map Method

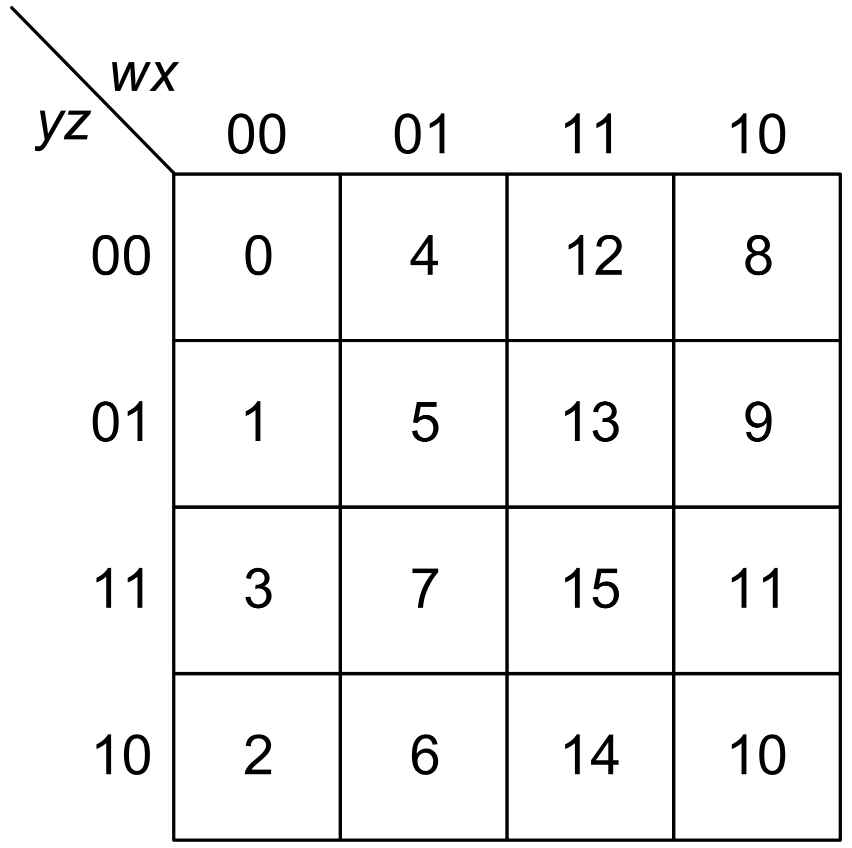
- **Karnaugh map**: modified form of truth table
- Combine terms using the **Aa + Aa' = A (combining theorem)**
- **<u>Cube</u>**:
  - Collection of $\mathbf{2}^{m}$ cells, each adjacent to ***m*** cells of the collection
  - The cube is said to **cover** the cells it is involved with
  - Expressed by a product of **n-m literals** for a function containing **n variables**
  - **m literals** not in the product said to be eliminated
  - **More Clarification**:
    - Consider the squares 2 and 6 in Fig (a)
    - The minterms are z'x'y and z'xy
    - Now apply the **combining theorem.**
    - Literal x and x' are eliminated.
    - The result is a 2-cube.



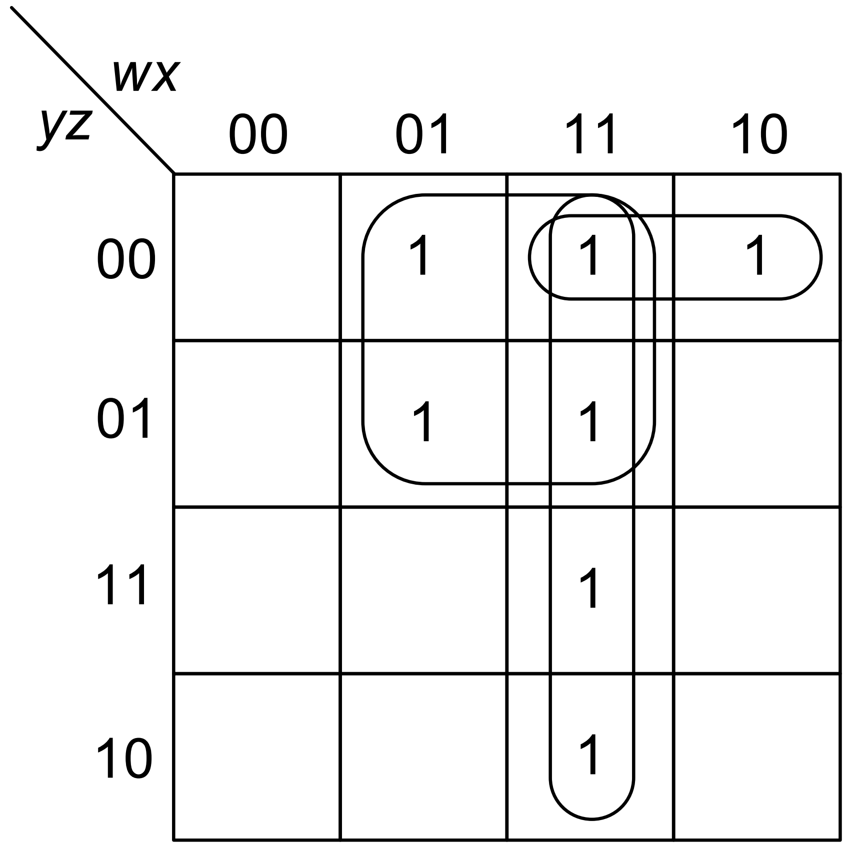
(a) Location of minterms in a three-variable map.



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

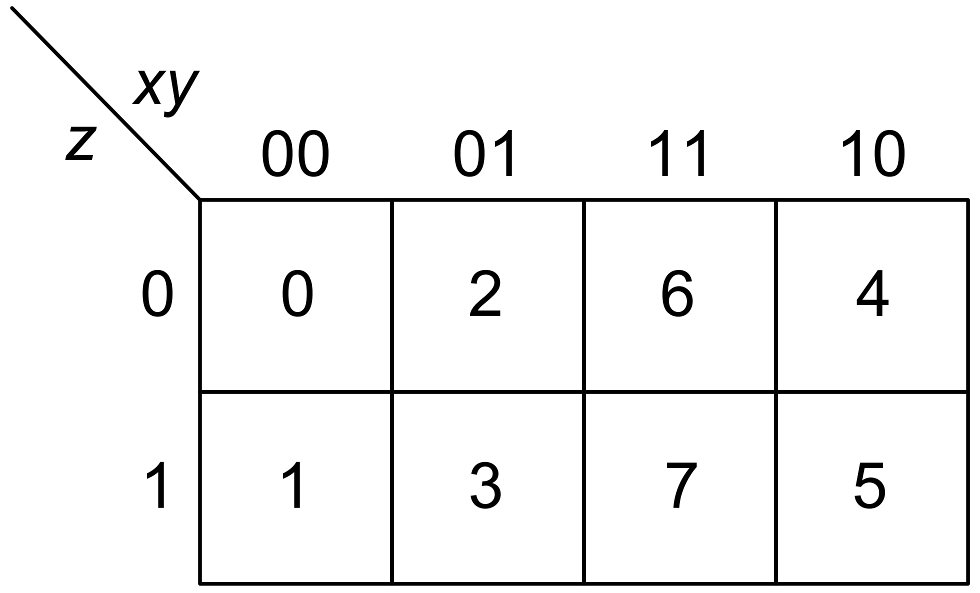


(c) Location of minterms in a four-variable map.

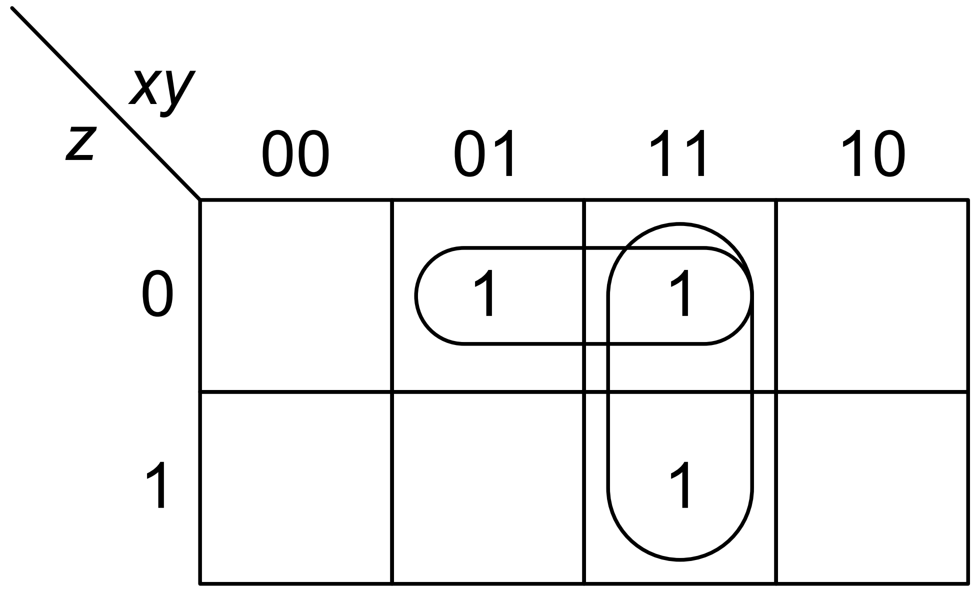


(d) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

The Map Method



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



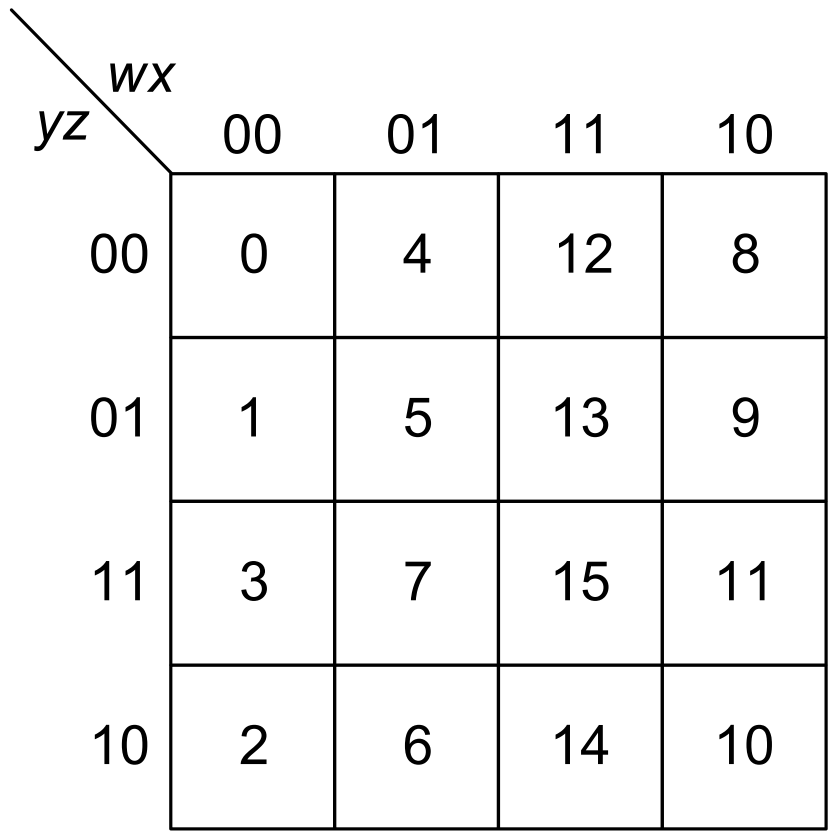
(*b*) Map for function $f(x,y,z) = \sum(2,6,7) = yz' + xy$.

- **Example**: $f = yz' + xy$
  - **Use of cell 6 in forming both cubes justified by idempotent law**
  - Corresponding algebraic manipulations:

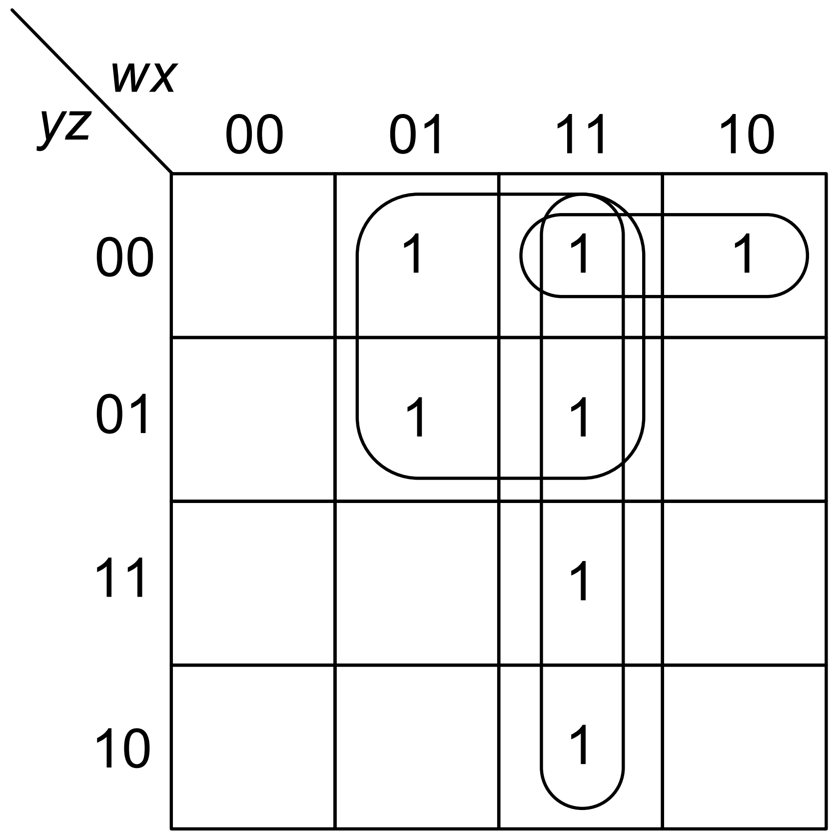
$$\begin{aligned} f &= x'yz' + xyz' + xyz \\ &= x'yz' + \underline{xyz' + xyz'} + xyz \text{ (idempotent law)} \\ &= yz'(x' + x) + xy(z' + z) \\ &= yz' + xy \end{aligned}$$

The Map Method

- **Example**: w'xy'z' + w'xy'z + wxy'z' + wxy'z = xy'(w'z' + w'z + wz' + wz) = xy'
- **Trick**:
  - In a cube, just keep the variables not changing their value.



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

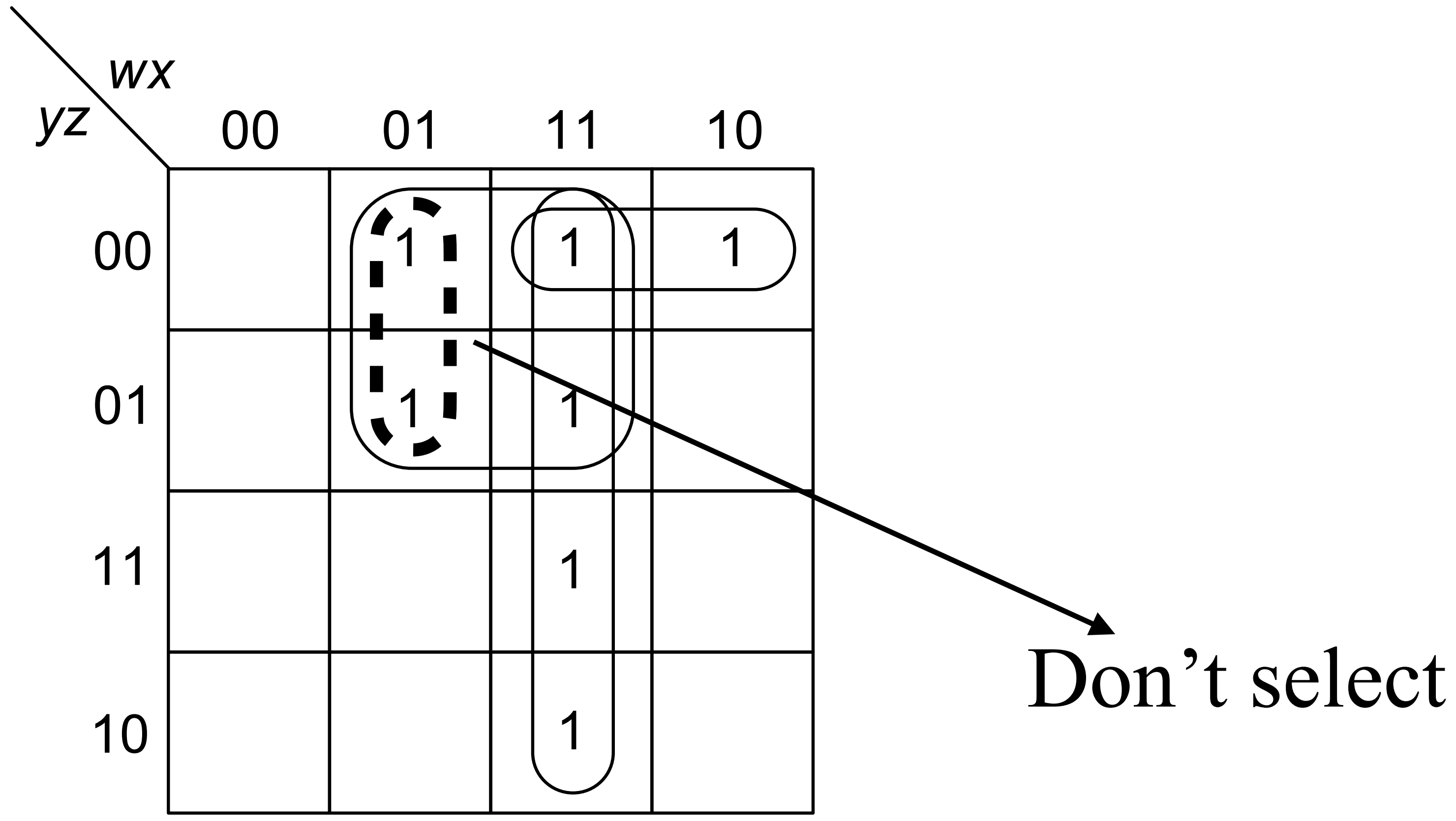


(d) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

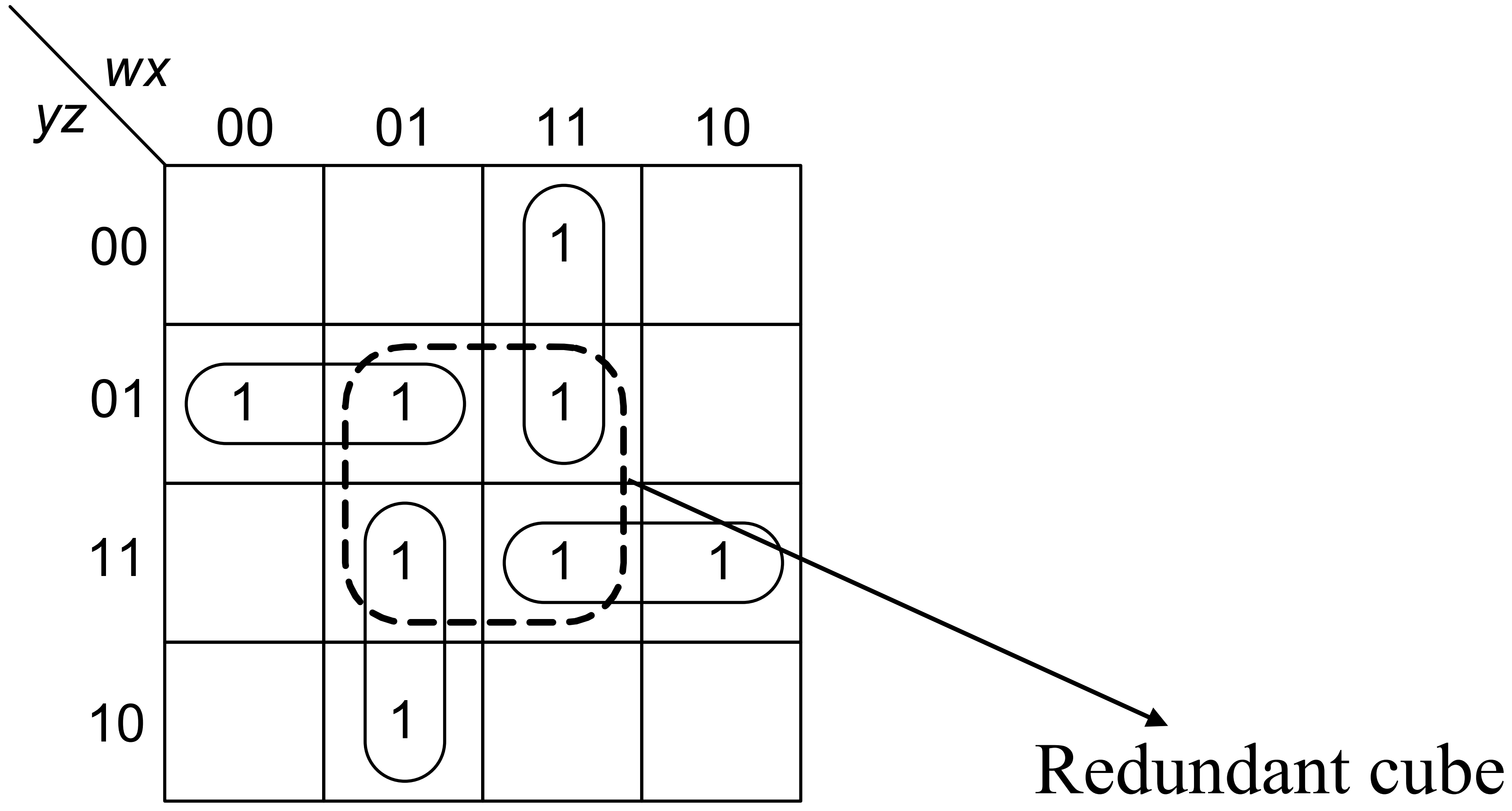
The Map Method

Rules for minimization:

- **Step 1**: cover those 1 cells by cubes that cannot be combined with other 1 cells; continue to 1 cells that have a single adjacent 1 cell (thus can form cubes of only two cells)
- **Step 2—:** Combine 1 cells that yield cubes of four cells, but are not part of any cube of eight cells, and so on..
  - A cube contained in a larger cube must never be selected
  - A cube contained in any combination of other cubes already selected in the cover is redundant (**consensus theorem)**
  - If there are more than one way of covering the map with cubes, select the cover with larger cubes
  - **Minimal expression**: collection of cubes that are as large and as few in number as possible, such that each 1 cell is covered by at least one cube
  - **Irredundant expressions**:
    - An SOP from where no term or literal can be deleted.
    - Not necessarily minimal
  - **Minimal and irredundant expressions may not be unique**
  - **But a minimal expression is always irredundant.**

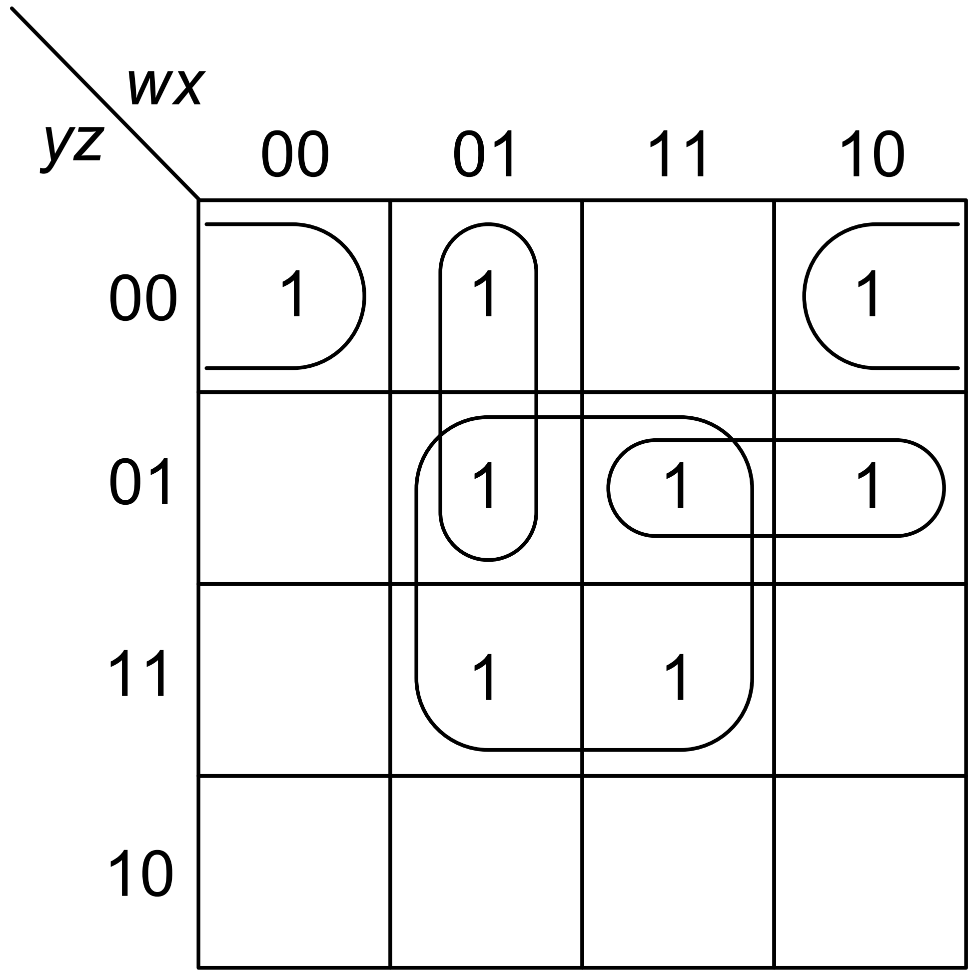


(d) Map for function $f(w,x,y,z) = \sum(4,5,8,12,13,14,15) = wx + xy' + wy'z'$.

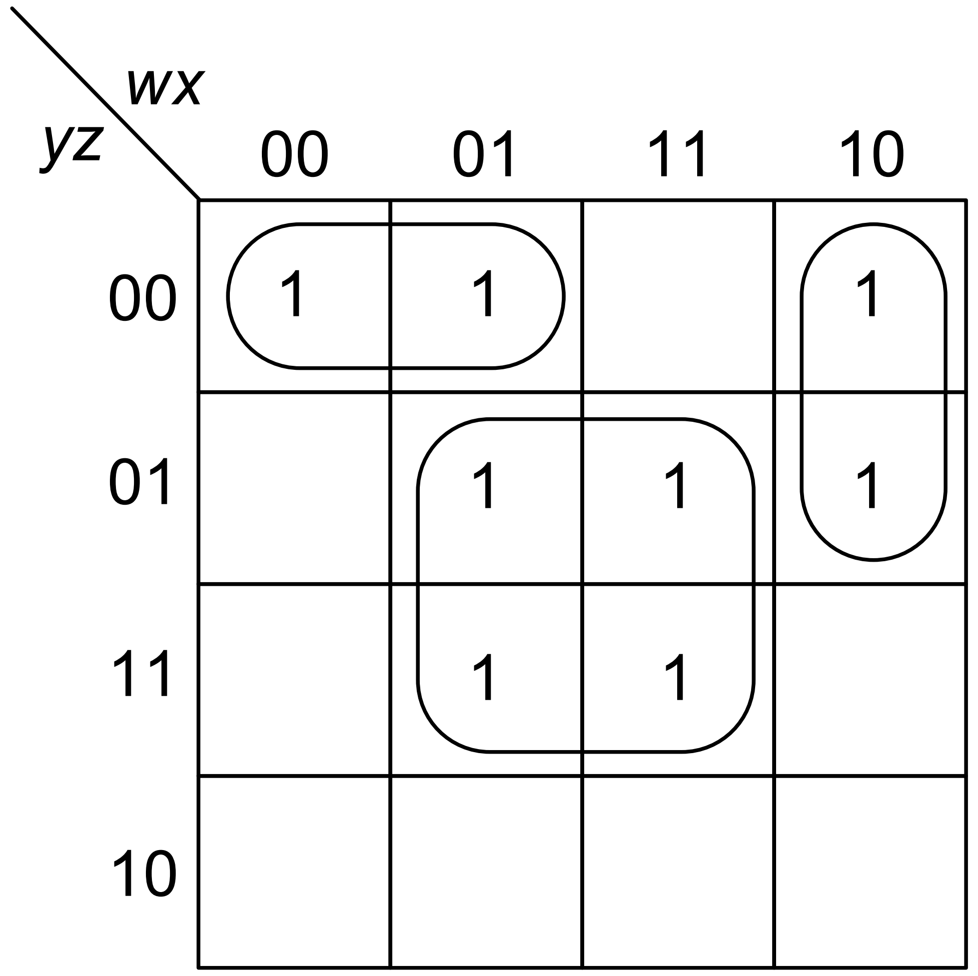


Let's try this..

The Map Method



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



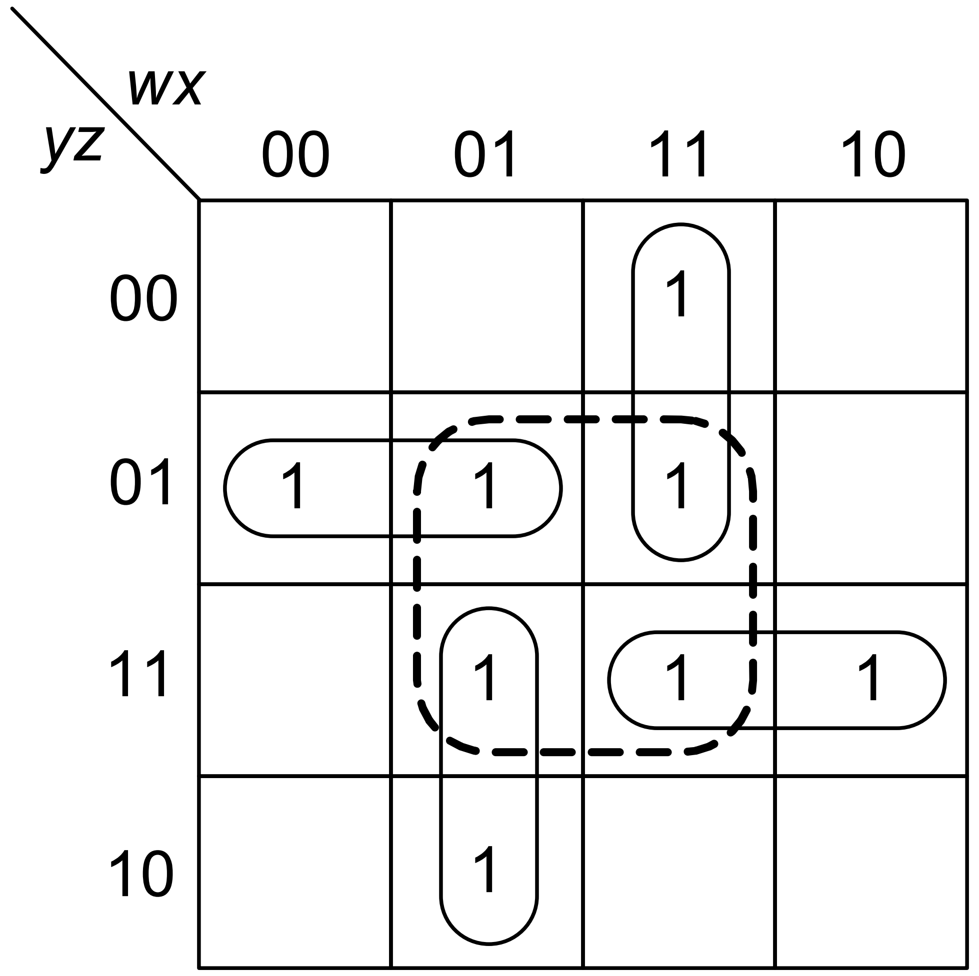
(b) $f = w'y'z' + wx'y' + xz$ is the
unique minimal expression.

Example: Two irredundant expressions for $f(w,x,y,z) = \sum(0,4,5,7,8,9,13,15)$

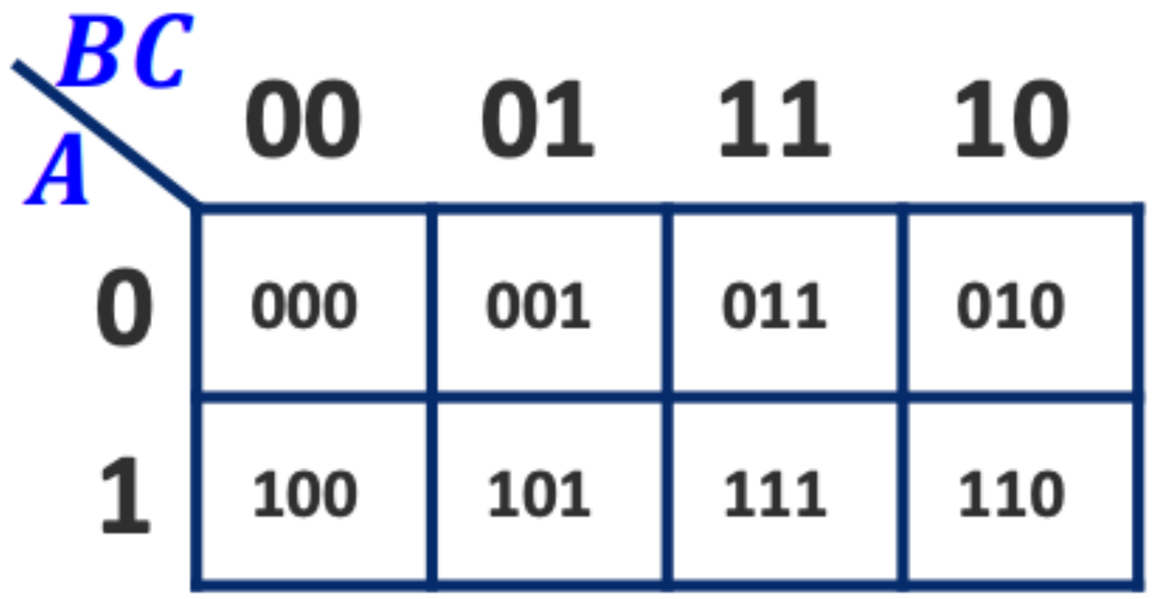
The Map Method

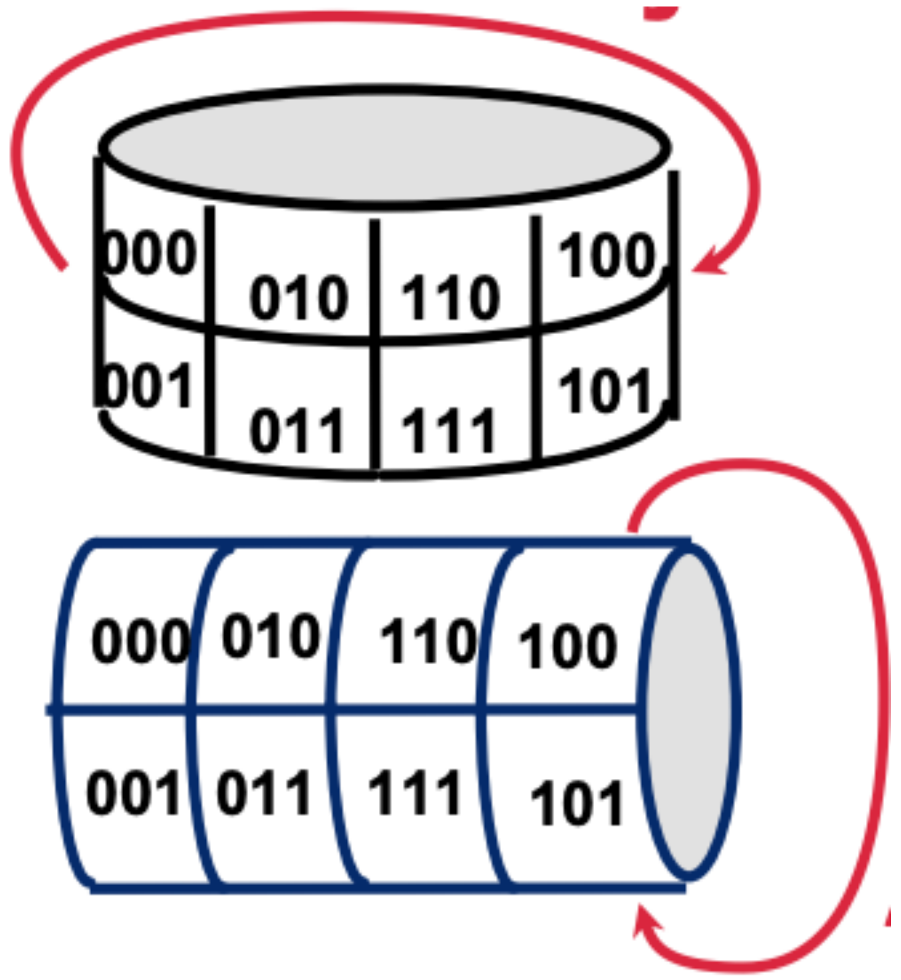
Example: $f(w,x,y,z) = \sum(1,5,6,7,11,12,13,15)$

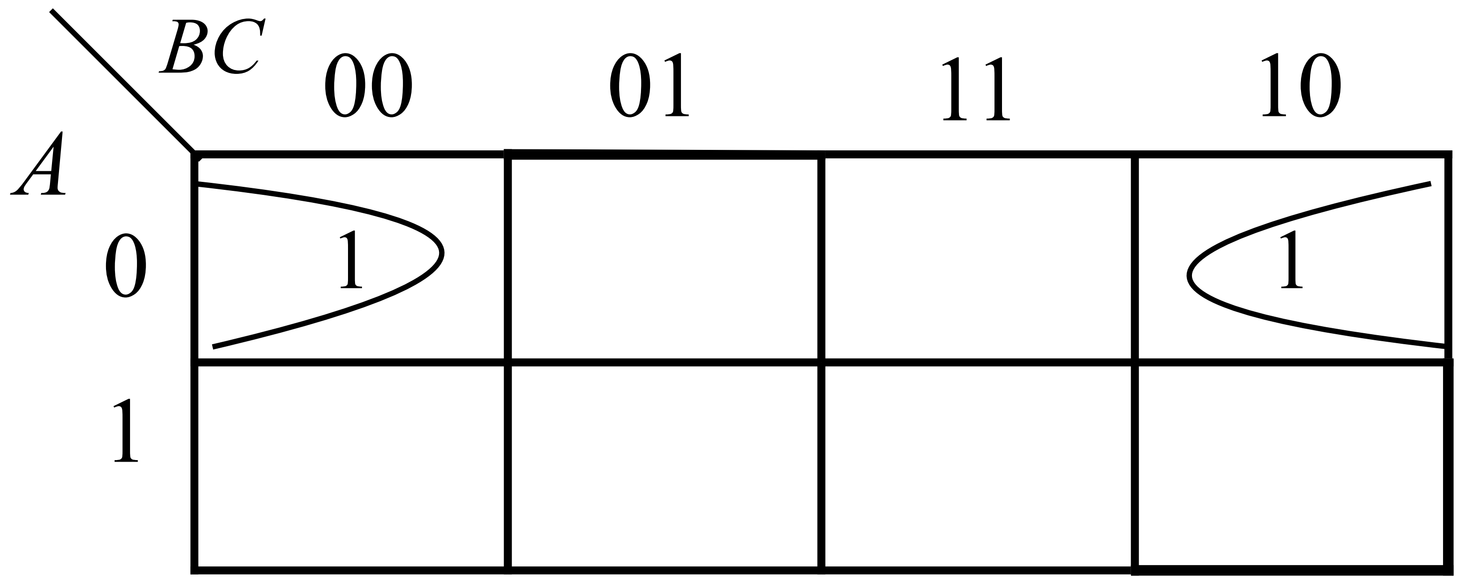
- Only one irredundant form: f = wxy’ + wyz + w’xy + w’y’z



The Map Method: Earth is not Flat

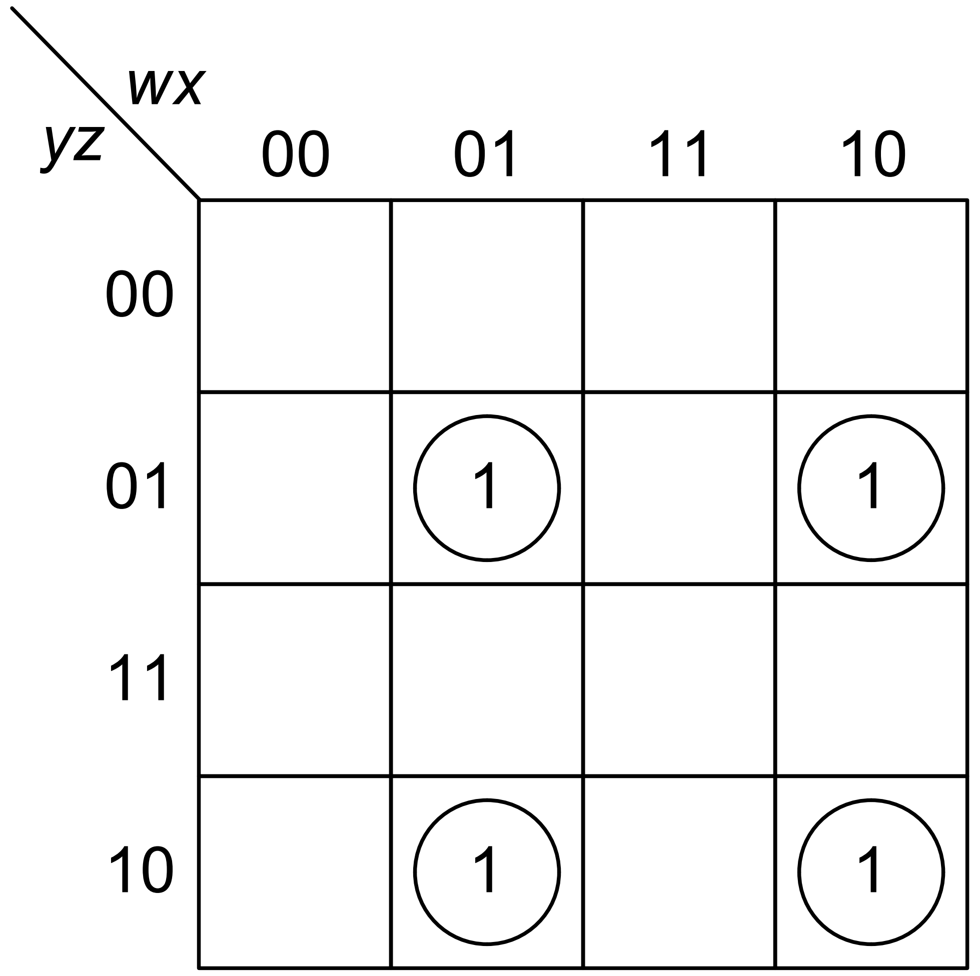




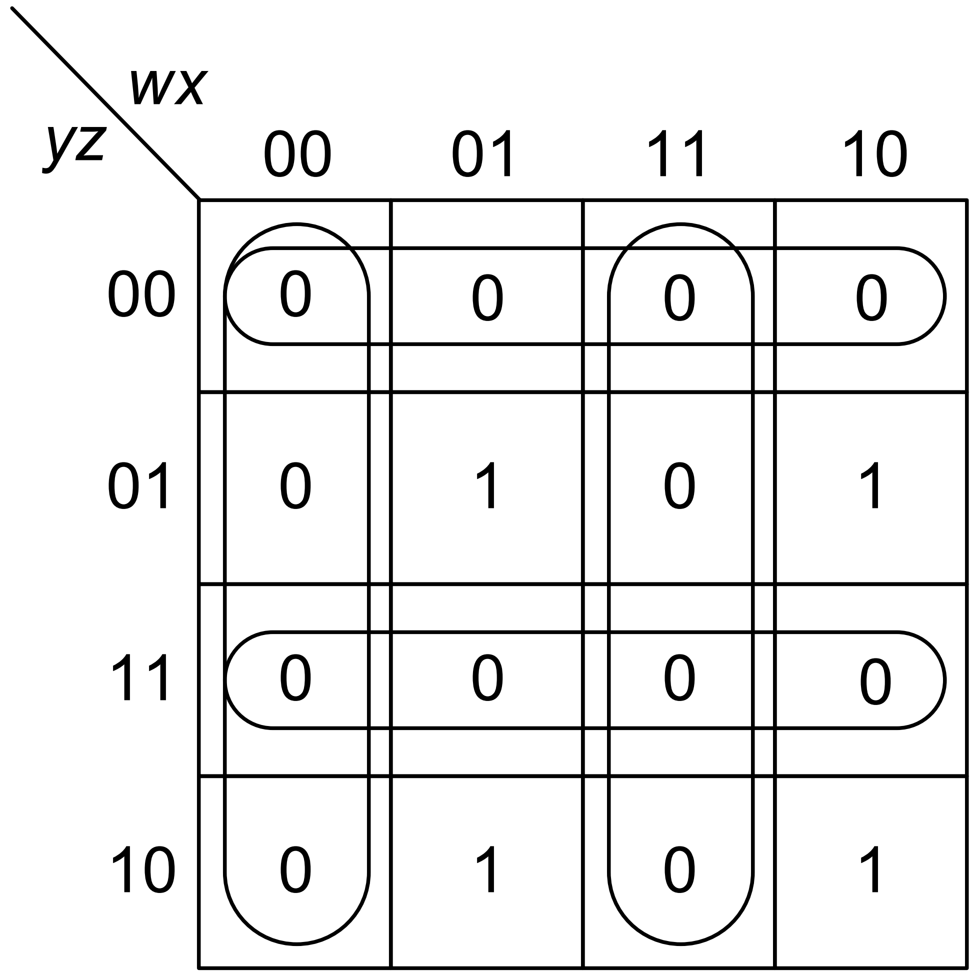


Minimal Product-of-Sums

- **Dual procedure**: product of a minimum number of sum factors, provided there is no other such product with the same number of factors and fewer literals
    - **Variable corresponding to a 1 (0) is complemented (uncomplemented)**
    - Cubes are formed of 0 cells
- **Example**: either one of minimal sum-of-products or minimal product-of-sums can be better than the other in literal count



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

Let's Try it..

- Implement $f(A, B, C, D) = \sum(0,2,8,12,13)$ with minimum number of gates.

Let's Try it..

- Implement **complement of** $f(A,B,C,D) = \prod(7,9,13)$.

Don’t-care Combinations

- **Don’t-care combination ϕ:** combination for which the value of the function is not specified.
  - Either input combinations may be invalid
  - Or precise output value is of no importance

- Since each don’t-care can be specified as either 0 or 1
  - a function with k don’t-cares corresponds to a class of 2^k distinct functions.
  - Our aim is to choose the function with the minimal representation
- Assign 1 to some don’t-cares and 0 to others in order to increase the size of the selected cubes whenever possible
- No cube containing only don’t-care cells may be formed

Code Converter

Example: code converter from BCD to excess-3 code
Combinations 10 through 15 are don't-cares

Truth table

Decimal number	BCD inputs				Excess-3 outputs			
	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

Code Converter

Example: code converter from BCD to excess-3 code

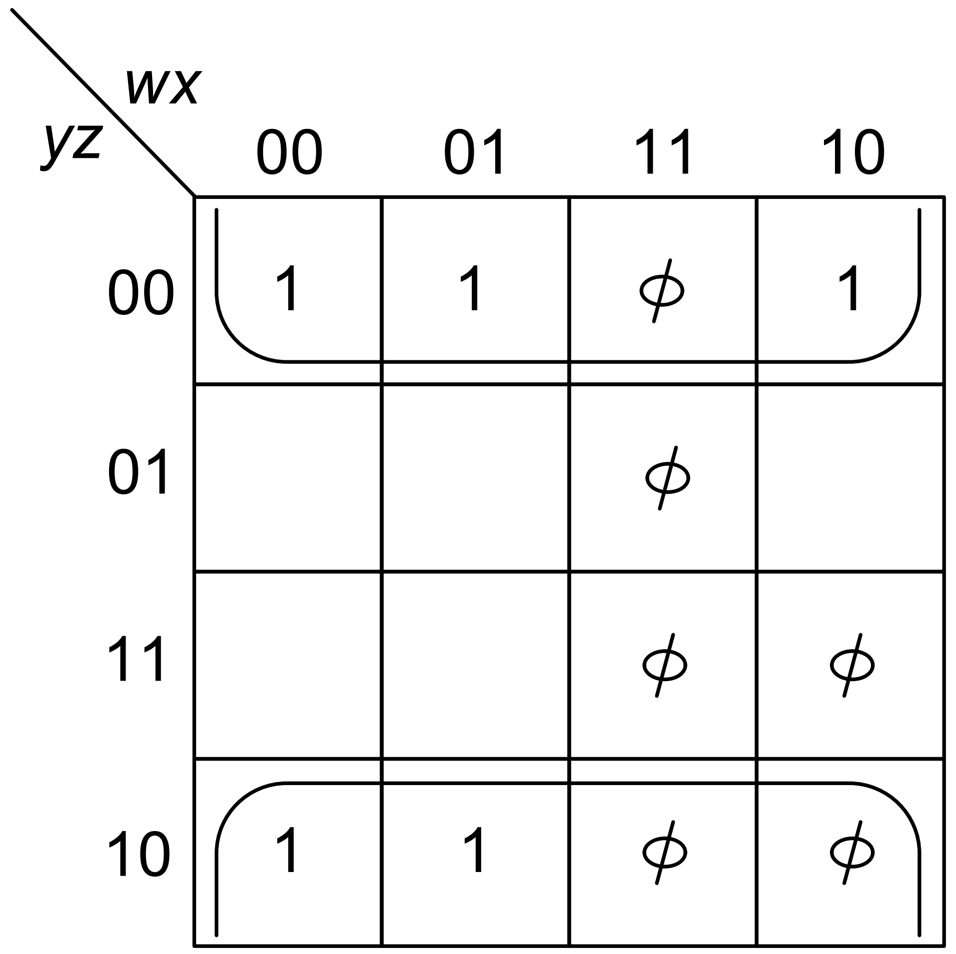
Combinations 10 through 15 are don't-cares

Truth table

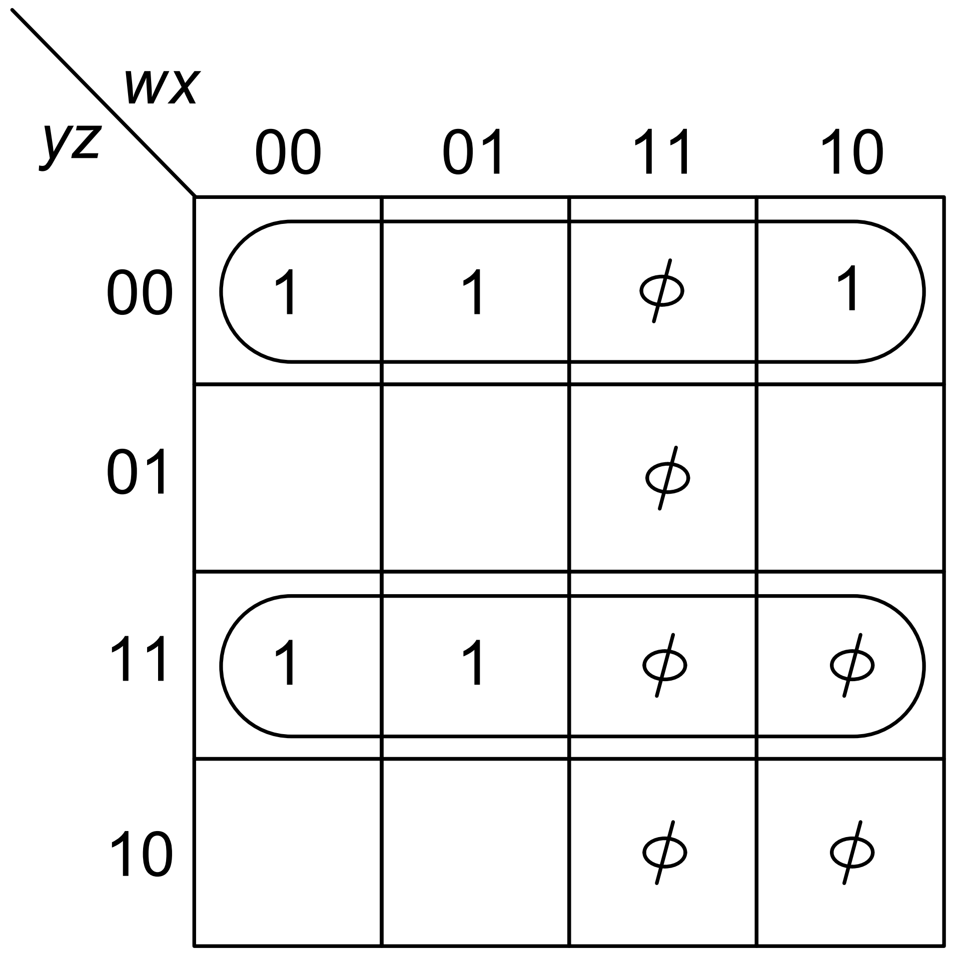
Decimal number	*BCD inputs*				*Excess-3 outputs*			
	w	x	y	z	f_4	f_3	f_2	f_1
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

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

Code Converter

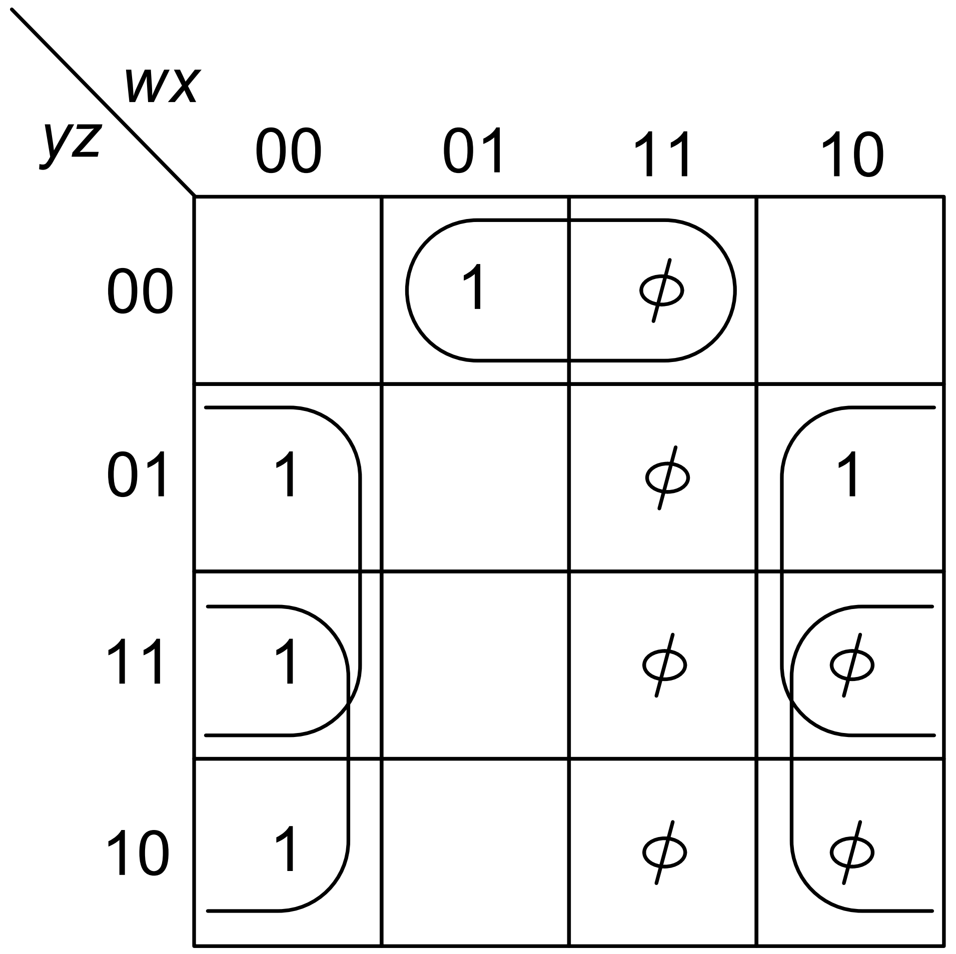


f_1 map

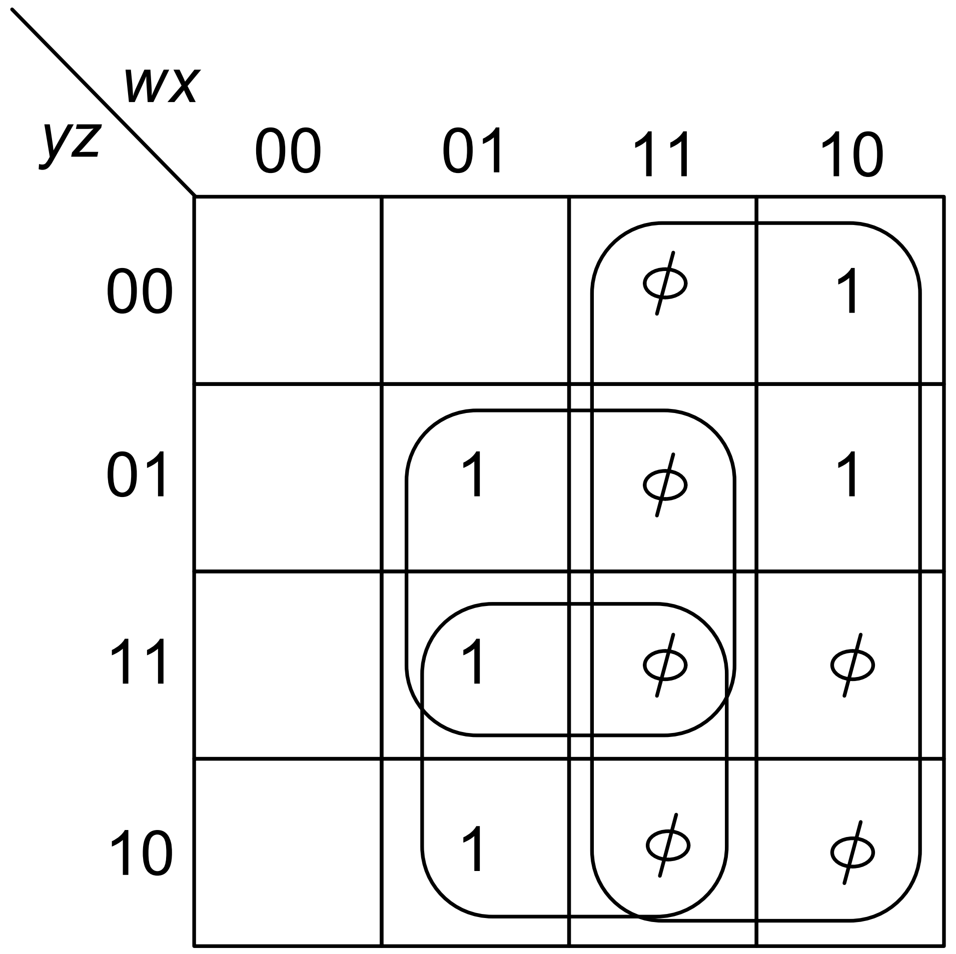


f_2 map

Increase the size of the cubes without making it necessary to increase the number of cubes, than would be required with fewer don’t cares assigned one.



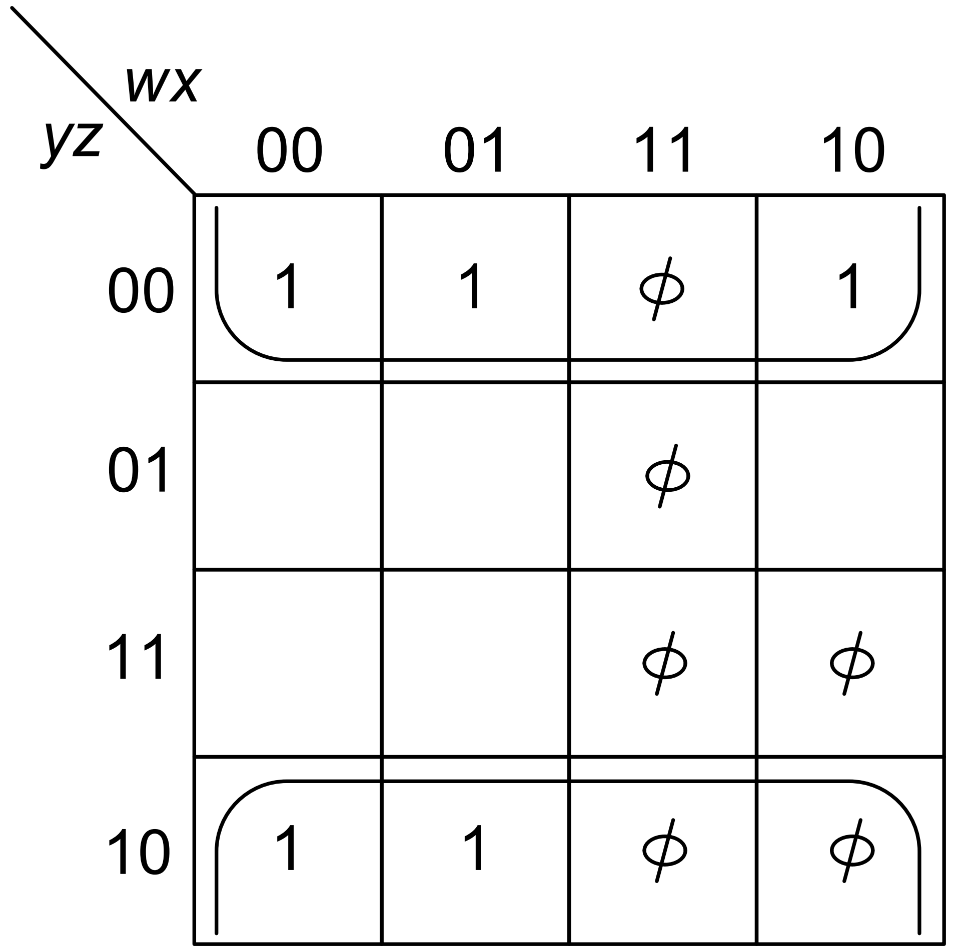
f_3 map



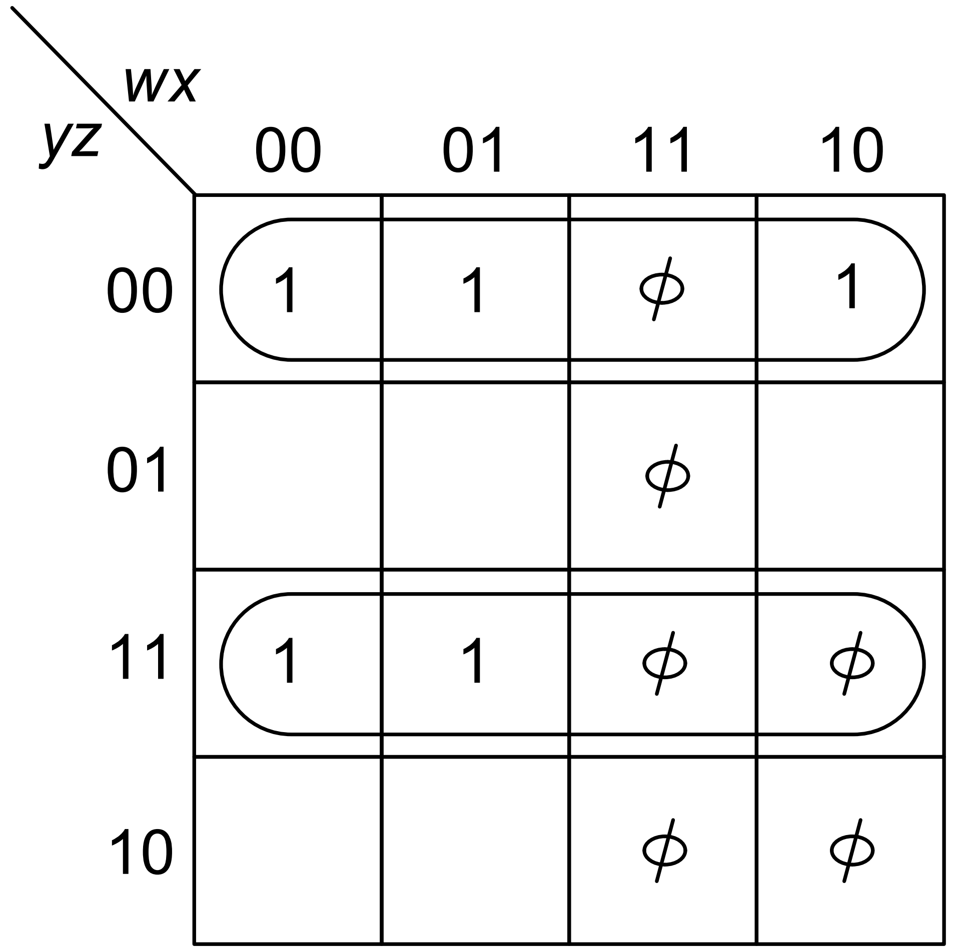
f_4 map

Lets do it!!!

Code Converter

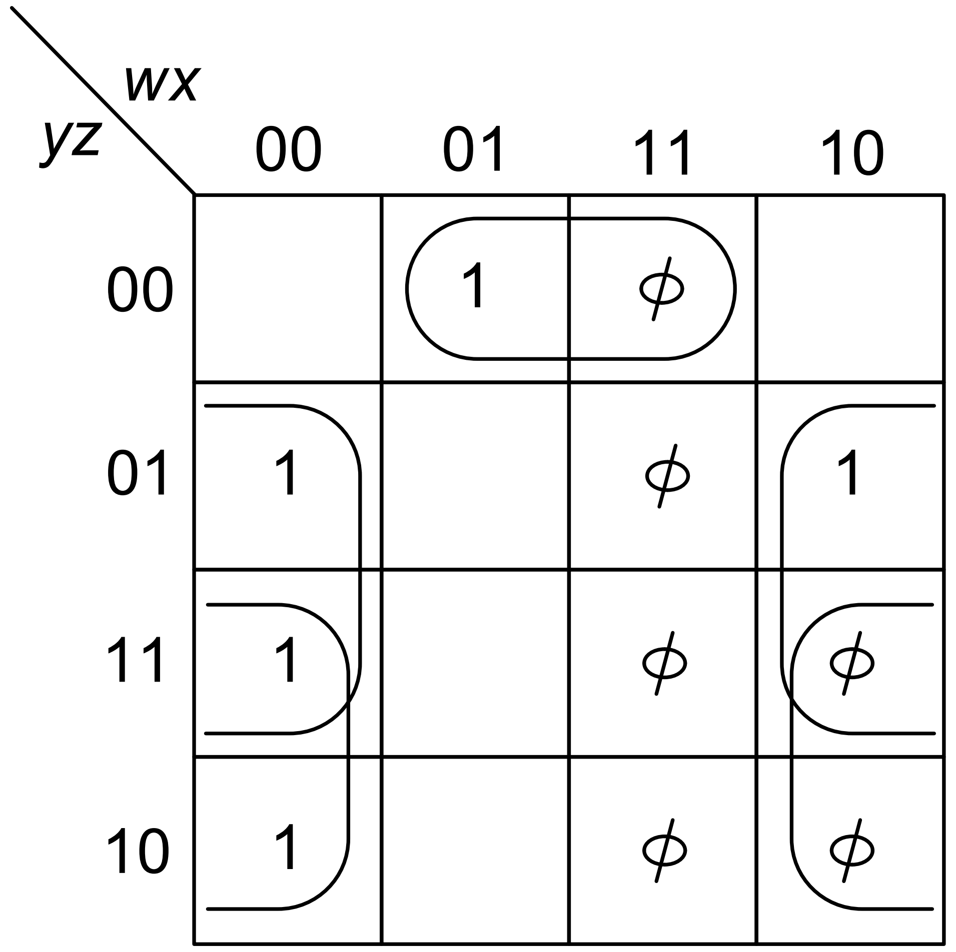


f_1 map

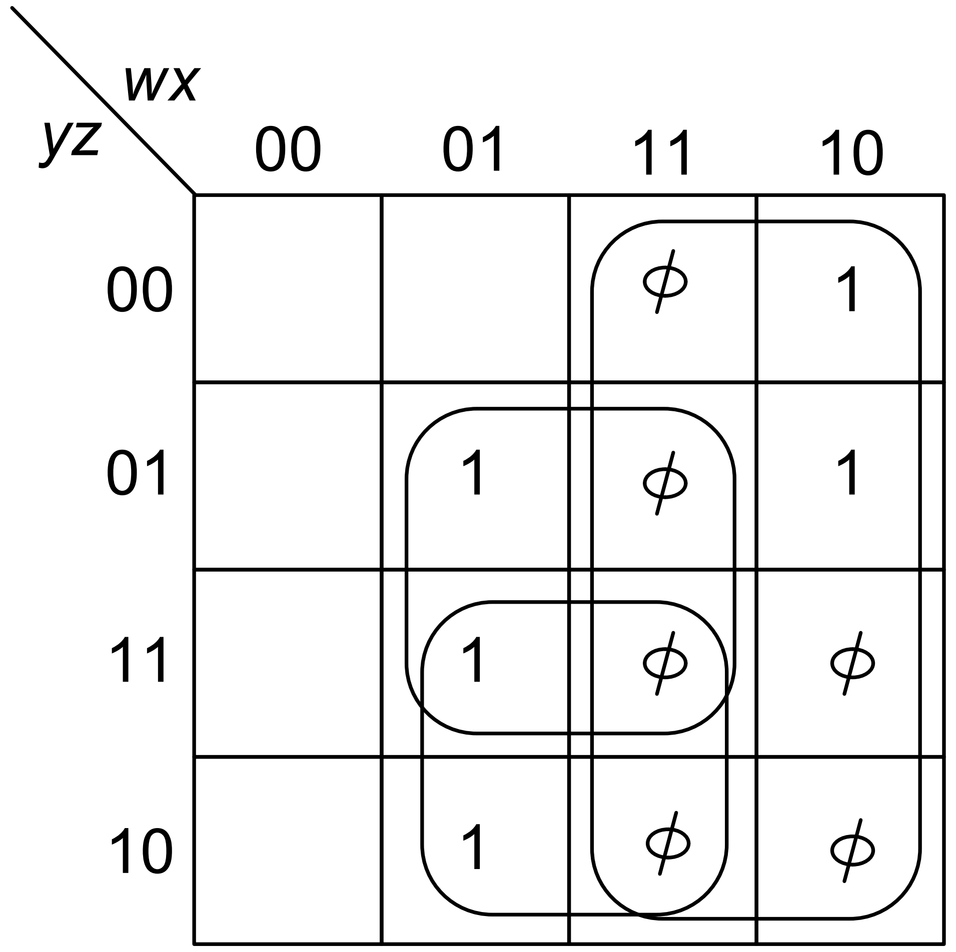


f_2 map

Increase the size of the cubes without making it necessary to increase the number of cubes, than would be required with fewer don't cares assigned one.



f_3 map

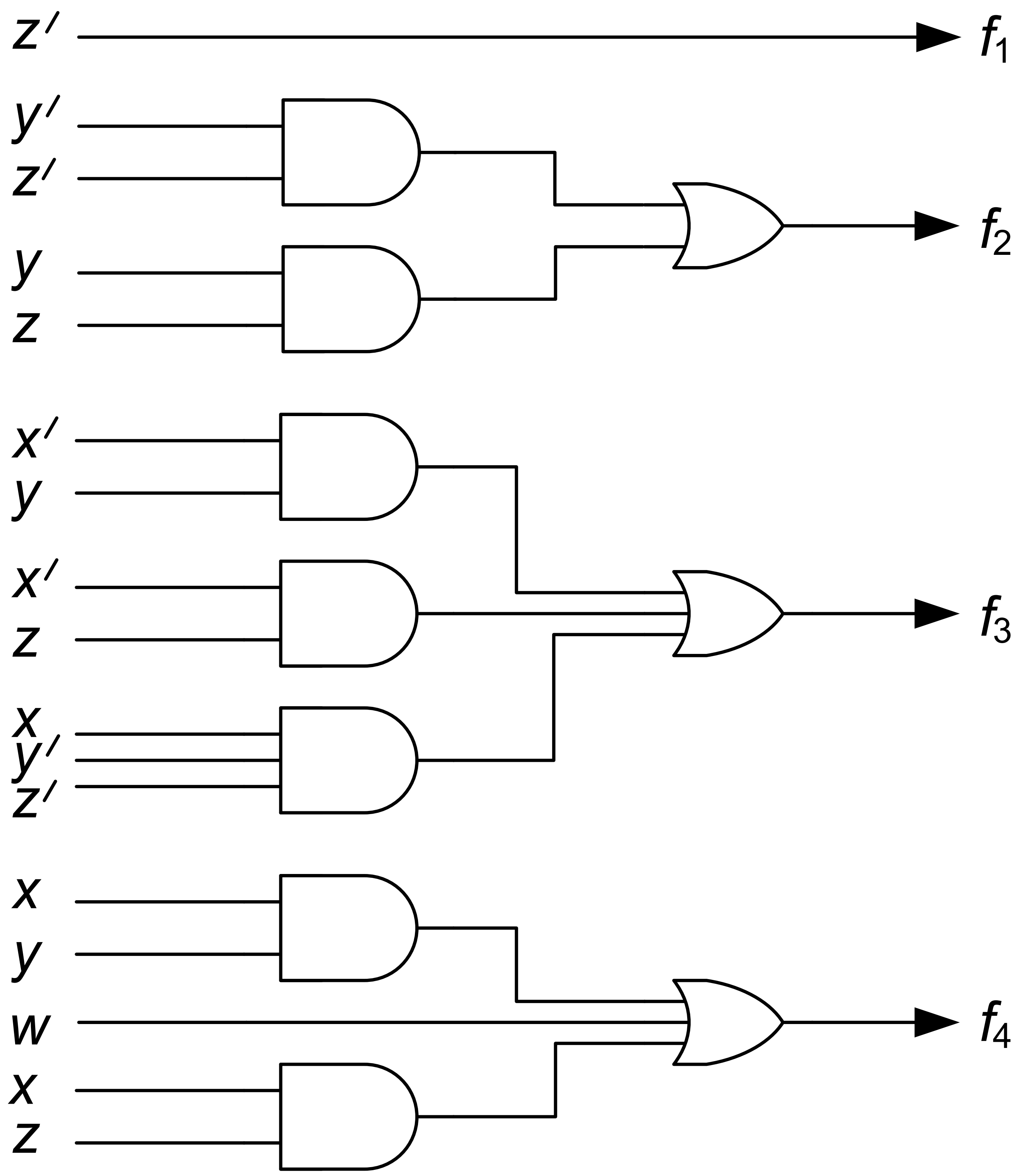


f_4 map

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

Logic Network for Code Converter

Two-level AND-OR realization:

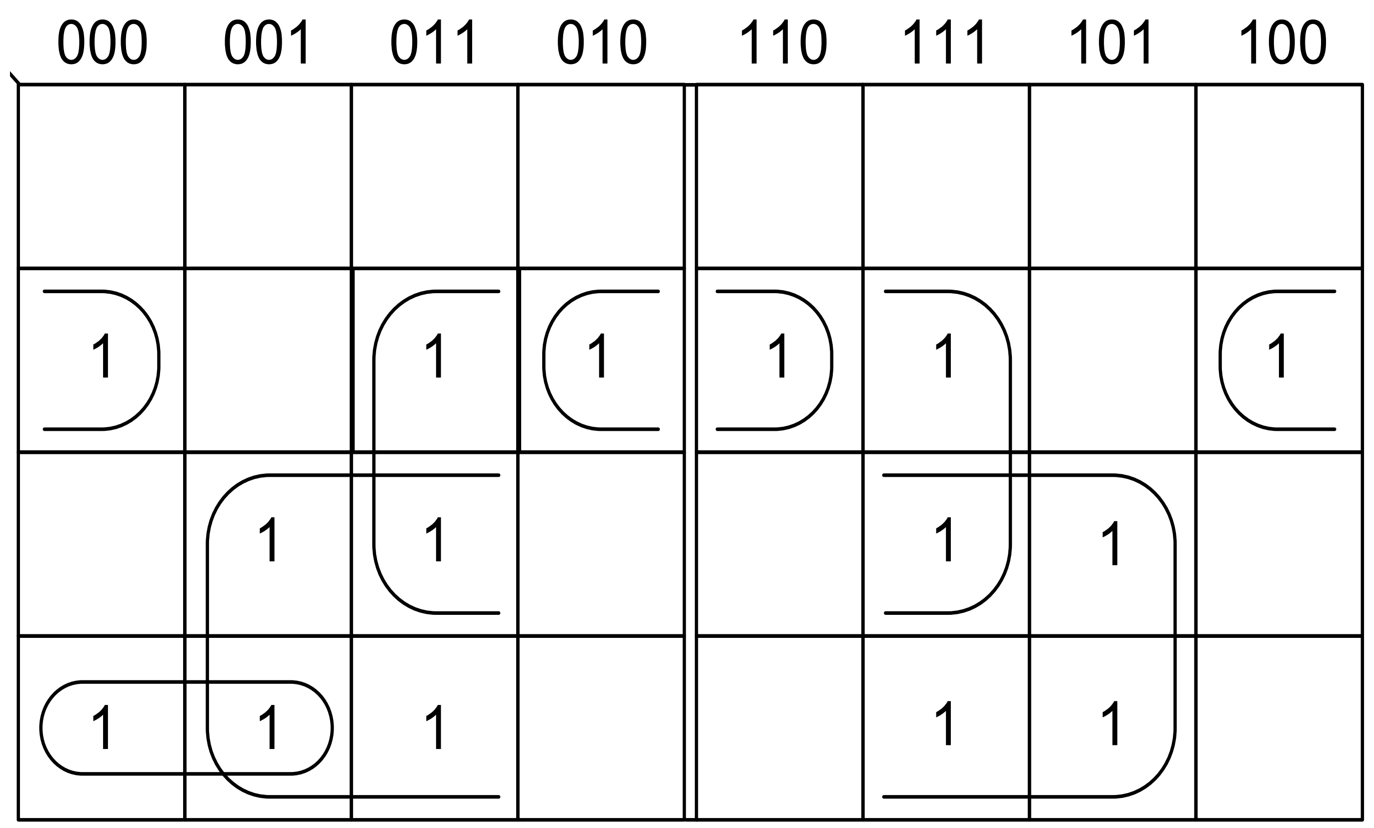


Five-variable Map

<u>General five-variable map</u>

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



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

Limitation of Simple Maps

- Maps are useful up to 5-6 variables, after that the calculation becomes formidable..
- How many cells are there in a 6 variable map??
- We also need something which is more amenable to a computer program.