

Canonical Forms

Minterms and Maxterms

- ❖ A minterm (aka standard product) is an AND term containing all variables.
- ❖ A maxterm (aka standard sum) is an OR term containing all variables.
- ❖ Minterms are labelled m_0 to m_{2^n-1} and maxterms are labelled M_0 to M_{2^n-1} (n is the number of variables), as in this three-variable example:

$\bar{x}\bar{y}\bar{z}$	m_0	0	$x + y + z$	M_0
$\bar{x}\bar{y}z$	m_1	1	$x + y + \bar{z}$	M_1
$\bar{x}y\bar{z}$	m_2	2	$x + \bar{y} + z$	M_2
$\bar{x}yz$	m_3	3	$x + \bar{y} + \bar{z}$	M_3
$x\bar{y}\bar{z}$	m_4	4	$\bar{x} + y + z$	M_4
$x\bar{y}z$	m_5	5	$\bar{x} + y + \bar{z}$	M_5
$xy\bar{z}$	m_6	6	$\bar{x} + \bar{y} + z$	M_6
xyz	m_7	7	$\bar{x} + \bar{y} + \bar{z}$	M_7

- ❖ Note that for any number p , $M_p = \bar{m}_p$.
- ❖ Any Boolean function can be expressed as a sum of minterms or a product of maxterms; when expressed either way, it is said to be in canonical form.
- ❖ To express a function already given in non-canonical algebraic form in canonical form, put it into a standard form and then add the missing variables.
- ❖ A function $F = m_a + m_b + m_c + \dots$ can be expressed as $F = \Sigma(a, b, c, \dots)$.
- ❖ Similarly, $F = M_a M_b M_c \dots$ can be expressed as $F = \Pi(a, b, c, \dots)$.

Gray Code and BCD

Gray code is used as an alternate form of encoding decimal numbers in 1s and 0s (the usual form is straight binary). We use Gray code because any two adjacent numbers differ in only one bit. In Karnaugh maps we are looking for terms that differ in only one bit, which will appear adjacent to each other because we use Gray code.

BCD (binary-coded decimal) simply represents decimal numbers one digit at a time.

decimal	binary	octal	hex	BCD	Gray
0	0000	00	0	00000000	0000
1	0001	01	1	00000001	0001
2	0010	02	2	00000010	0011
3	0011	03	3	00000011	0010
4	0100	04	4	00000100	0110
5	0101	05	5	00000101	0111
6	0110	06	6	00000110	0101
7	0111	07	7	00000111	0100
8	1000	10	8	00001000	1100
9	1001	11	9	00001001	1101
10	1010	12	A	00010000	1111
11	1011	13	B	00010001	1110
12	1100	14	C	00010010	1010
13	1101	15	D	00010011	1011
14	1110	16	E	00010100	1001
15	1111	17	F	00010101	1000

Karnaugh Maps

Constructing Karnaugh Maps

- ❖ A Karnaugh map contains exactly as much information as a truth table. It is simply rearranged in such a way that it makes simplification easy.
- ❖ Theoretically we could draw n-dimensional maps for n variables. In practice, we are limited to two dimensions, but we can fit two variables on each axis. Karnaugh maps of five variables require an extra trick.
- ❖ To construct a Karnaugh map, start with a matrix. It should be as square as possible, but also must have a number of rows and columns that is a power of 2, and the total number of rows plus columns must add up to twice the number of variables.
- ❖ Then, label the rows and columns using Gray code.
- ❖ Then, assign a variable to every bit in the numbering.
- ❖ Finally, fill in the matrix with 1s, 0s (optional), and d's (for don't care).

Using Karnaugh Maps

- ❖ Every square in the matrix represents a minterm.
- ❖ To use a Karnaugh map, draw boxes on the matrix such that: all 1s are enclosed, no 0s are enclosed, and every box is a square or rectangle containing a number of rows and columns that is a power of two.
- ❖ To maximize the simplification, use the largest and fewest boxes possible.
- ❖ Don't care states can be included or excluded in the boxes.
- ❖ Remember that Karnaugh maps wrap around on all edges.
- ❖ After drawing the boxes, convert each into an algebraic term. Sum these terms together to find the expression for the function.
- ❖ To get the product of sums form instead of the sum of products, draw boxes around the 0s instead of the 1s. This gives the complement of the function. Invert it (using De Morgan's Theorem) to obtain a product of sums.