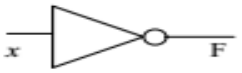
LECTURE 4: DIGITAL LOGIC CIRCUIT DESIGN

Digital Logic Gates

Commonly used digital logic gates are given in the following table.

Name	Graphic Symbol	Algebraic Function	Truth Table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
XOR		$F = xy' + x'y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
XNOR		$F = xy + x'y'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Extension to Multiple Inputs

The commonly used gates, except the buffer and the inverter, can be extended to have more than two inputs if the binary operation is commutative and associative (e.g. AND, OR).

NAND and NOR are commutative but not associative

$$\left[(x+y)'+z\right]' \neq \left[x+(y+z)'\right]'.$$

We define the multiple NOR and NAND gate as a complemented OR or AND gate.

e.g. three variable NOR gate → $(x+y+z)'$.

LECTURE 4: DIGITAL LOGIC CIRCUIT DESIGN

Integrated Circuits

Integrated circuits are classified based on the levels of integration.

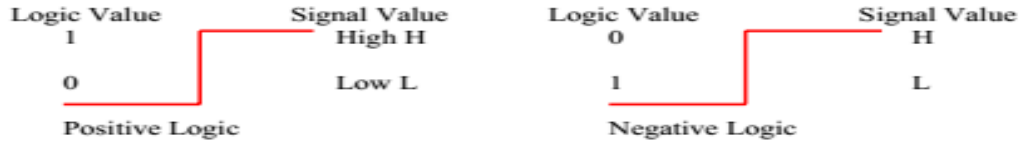
SSI **< 10 gates**

MSI **10 – 100 gates (e.g., decoders, adders, MUXs)**

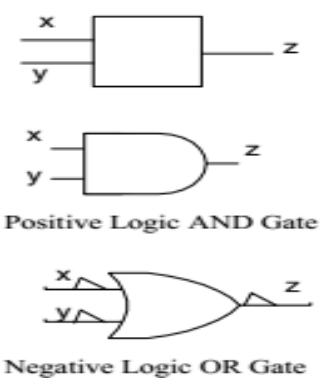
LSI **100 – few 1000s (e.g., processors, memory chips)**

VLSI **> 1000s**

Positive and Negative Logic



Demonstration of positive & Negative Logic



x y	z
L L	L
L H	L
H L	L
H H	H

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

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

Canonical and Standard Forms

Minterms and Maxterms for three binary values

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

A function is said to be in Canonical Form if it is expressed either in Sum of Minterms form or Product of Maxterms form.

LECTURE 4: DIGITAL LOGIC CIRCUIT DESIGN

Note: A term that is a Minterm or a Maxterm must contain all the input variables.

Functions of Three Variables

x	y	z	F_1	F_2	F_1'
0	0	0	0	0	1
0	0	1	1	0	0
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	0

$F_1 = x'y'z + xy'z' + xyz$

$F_2 = x'yz + xy'z + xyz' + xy$

$F_1 = m_1 + m_4 + m_7$

$= \sum m(1,4,7) = \sum(1,4,7)$

$F_2 = m_3 + m_5 + m_6 + m_7$

$= \sum(3,5,6,7)$

$F_1' = m_0 + m_2 + m_3 + m_5 + m_6$

$= \sum(0,2,3,5,6)$

$= x'y'z' + x'yz' + x'yz + xy'z + xyz'$

$F_1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$

$= M_0.M_2.M_3.M_5.M_6$

$= \prod(0,2,3,5,6)$

Similarly, $F_2 = \prod(0,1,2,4)$

Example 1: Express $F = A + B'C$ in Sum Of Minterms or SOP form.

A	B	C	$A+B'C$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

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

$= AB(C+C') + AB'(C+C') + (A+A')B'C$

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

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

$= m7 + m6 + m5 + m4 + m1$

$= \sum(1,4,5,6,7)$

Example 2: Express F = $xy + x'z$ in Product of Maxterms or POS form.

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$F = (xy + x')(xy + z)$

$= (x + x')(y + x')(x + z)(y + z)$

$= (x'+y)(x + z)(y + z)$

$= (x'+y + zz')(x + yy'+z)(xx'+y + z)$

$= (x'+y + z)(x'+y + z')(x + y + z)(x + y'+z)(x + y + z)(x'+y + z)$

$= (x + y + z)(x + y'+z)(x'+y + z)(x'+y + z')$

$= M_0.M_2.M_4.M_5$

$= \prod(0,2,4,5)$

LECTURE 4: DIGITAL LOGIC CIRCUIT DESIGN

Conversion between Canonical Forms

Example:

$F(A,B,C) = \sum(1,4,5,6,7)$

$F'(A,B,C) = \sum(0,2,3)$

$\therefore F(A,B,C) = (m_0 + m_2 + m_3)'$

$= M_0.M_2.M_3 = \prod(0,2,3)$

Standard Forms

A Standard form can be obtained upon simplifying a Sum of Minterms or Product of Maxterms expression. A term in a SOP or POS form need not contain all the variables of a function.

S O P → Sum Of Products — $F_1 = y'+xy+x'yz'$

P O S → Product Of Sums — $F_2 = x(x'+z)(x'+y+z'+w)$

A non-standard form can be changed to SOP form.

$F3 = (AB + CD)(A'B'+C'D')$

$= A'B'CD + ABC'D'$

Truth Tables of 16 Functions of Two Binary Variables

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Operator Symbol			.		/		/		⊕	+	↓	⊙	'	⊂	'	⊃	↑	

Boolean Expressions for the 16 Functions of Two Variables

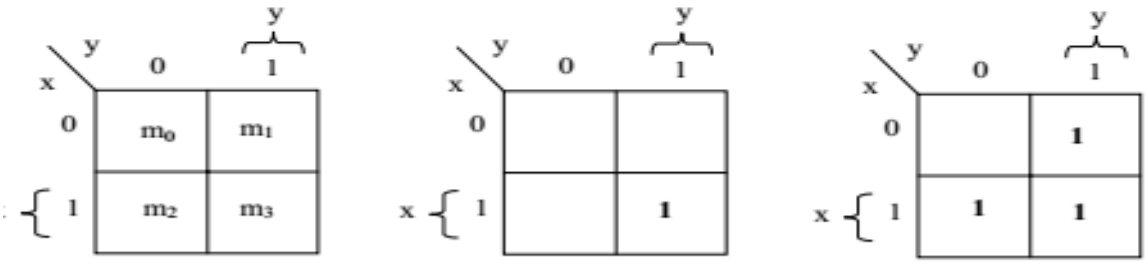
Boolean Functions	Operator Symbols	Name	Comments
$F_0 = 0$		Null	Binary Constant 0
$F_1 = xy$	$x.y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + xy'$	$x \oplus y$	Exclusive-OR	x or y but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$x \odot y$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

LECTURE 4: DIGITAL LOGIC CIRCUIT DESIGN

Karnaugh Map

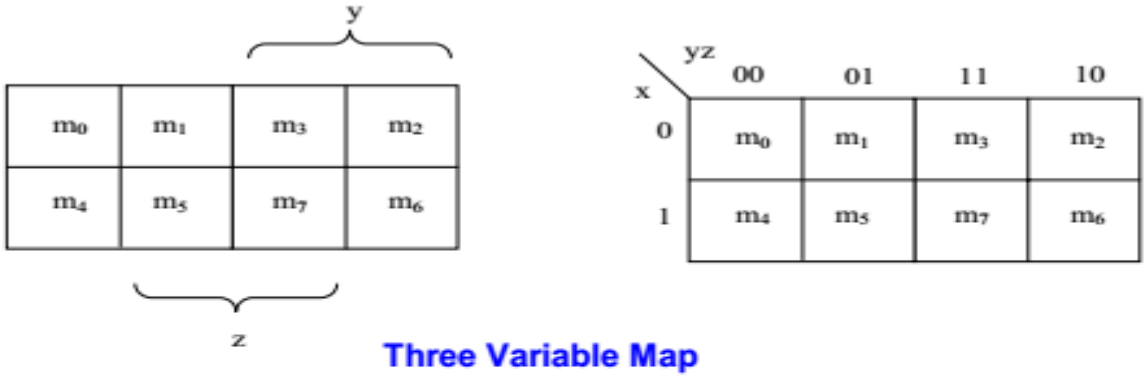
The map is a diagram mad up of squares, each small square represents a minterm. Any function is made up of minterms and can be represented on the map. Adjacents minterms can be combined to form simpler terms.

Two Variable Map



Two Variable Map $F = x.y$ $F = x + y$

Three Variable Map



Three Variable Map

Simplification

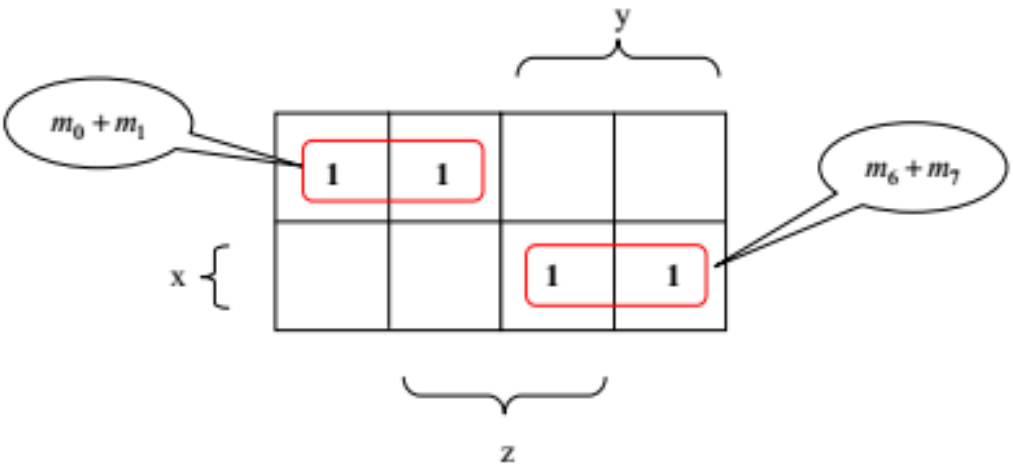
Suppose a function of three variables x, y, z is given by: $F = \sum(0,1,6,7)$. It is clear from the map that m_0 and m_1 are neibours. Also m_6 and m_7.

$$m_0 + m_1 = x'y'z' + x'y'z = x'y'$$

Similarly

$$m_6 + m_7 = xyz' + xyz = xy$$

This is shown in the map by the two groups of two minterms each.



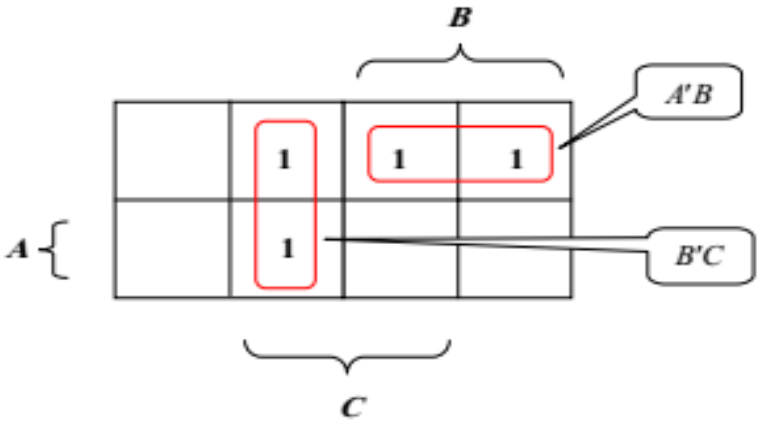
Example

Simplify the Boolean function $F = A'C + A'B + AB'C$

Solution: To simplify the expression we have to put it in a sum of minterms form.

$$F = A'C + A'B + AB'C$$
$$= A'B'C + A'BC + A'BC' + A'BC + AB'C$$
$$= \sum(1,3,2,3,5)$$
$$= \sum(1,2,3,5)$$

From the map:

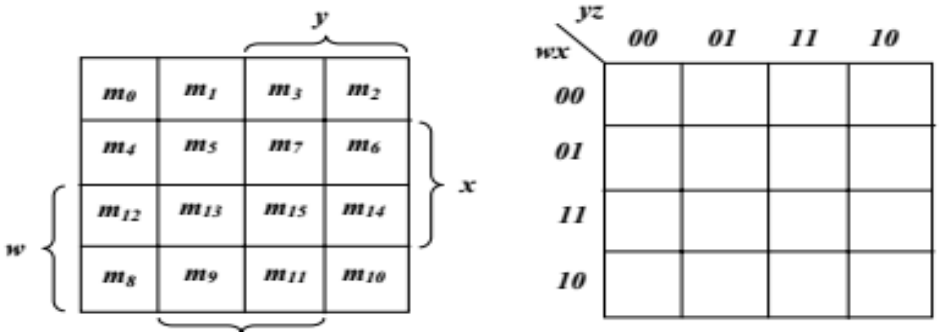


$$F = A'B + B'C$$

LECTURE 4: DIGITAL LOGIC CIRCUIT DESIGN

Four Variable Map

The map of four variables is shown below. It consists of sixteen small squares, arrange as a 4 x 4 small squares. Each small square represents one minterm. Each minterm has four neibouring (or adjacent) minterms.



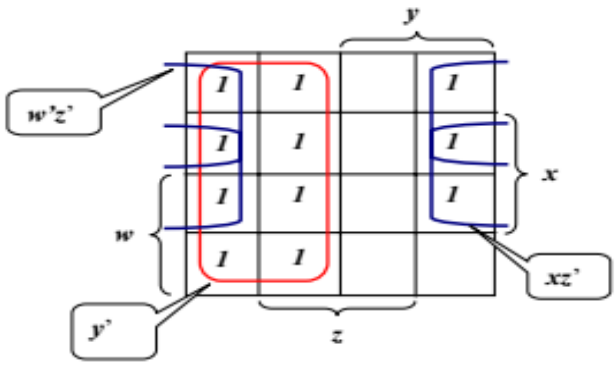
The map minimization of four variable functions depends on combining adjacent squares. The combination of adjacent squares help in the simplification of the functions.

One square represents minterm (4 literals).
Two adjacent squares represent a term with three literals.
Four adjacent squares represent a term of two literals.
Eight adjacent squares represent a term of one literal.
Sixteen adjacent squares represent the function (F = 1).

Example

Simplify the Boolean function

$F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$ using Karnaugh map.



$$F = y' + w'z' + xz'$$