Digital Logic Design
Combinational Logic
Part 1

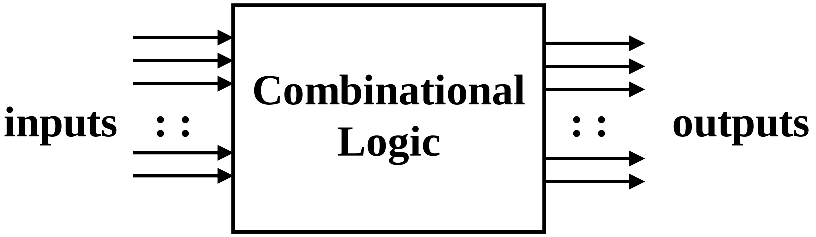
INTRODUCTION

Two classes of logic circuits

- Combinational Circuit

  Each output depends entirely on the immediate (present) inputs.

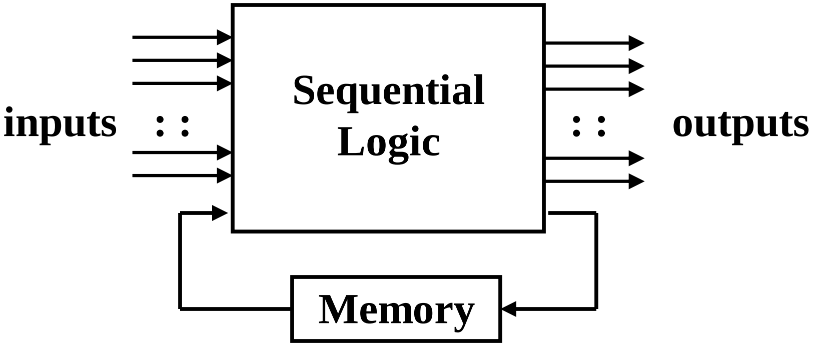
  - ✓ Gates
  - ✓ Decoders, multiplexers
  - ✓ Adders, multipliers



- Sequential Circuit

  - Each output depends on both present inputs and state.

  - ✓ Counters, registers
  - ✓ Memories



Boolean Algebra

- Digital circuits are hardware components for processing (manipulation) of binary input
- They are built of transistors and interconnections in semiconductor devices called integrated circuits
- A basic circuit is called a logic gate; its function can be represented mathematically.

BOOLEAN ALGEBRA

- Boolean values:
  - True (1)
  - False (0)

$\overline{A}$

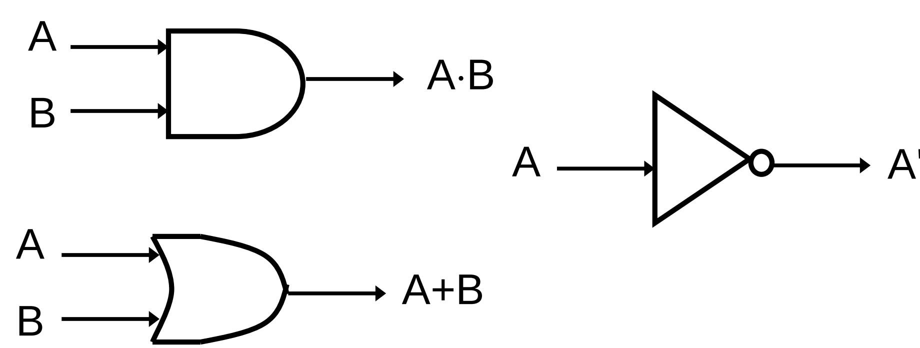
- Truth tables

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

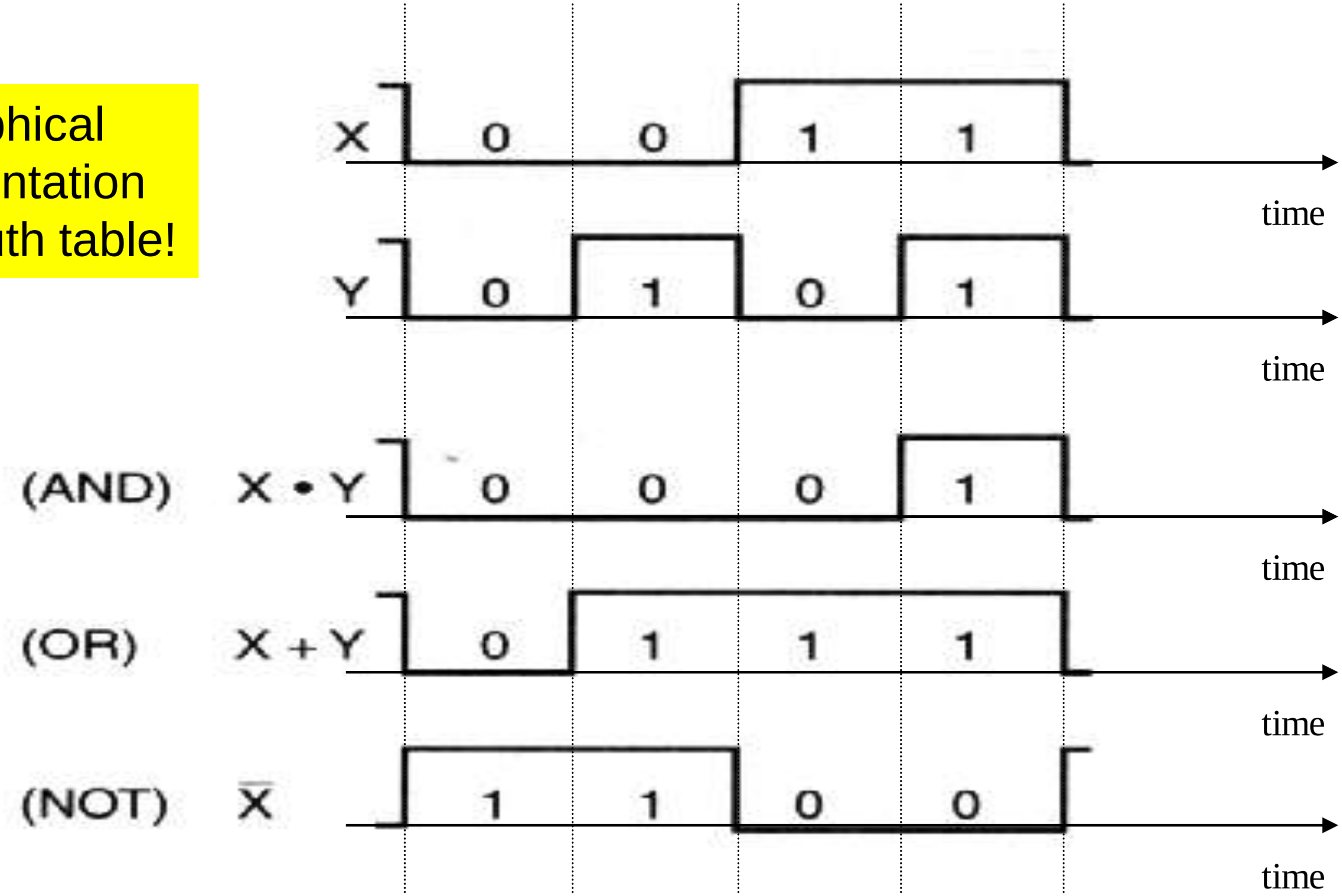
A	A'
0	1
1	0

- Logic gates



Timing Diagram

A graphical representation Of the truth table!



Boolean Expression

$$F(X, Y) = XY + Y'$$

variable   operator   literals

- **<u>A Boolean expression</u>** is made of Boolean variables and constants combined with logical operators: AND, OR and NOT
- A **<u>literal</u>** is each instance of a variable or constant.
- Boolean expressions are fully defined by their truth tables
- A Boolean expression can be represented using interconnected logic gates
  - Literals correspond to the input signals to the gates
  - Constants (1 or 0) can also be input signals
  - Operators of the expression are converted to logic gates
- Example: $a'bd + bcd + ac' + a'd'$ (4 variables, 10 literals, ?? gates)

Operator Precedence

- Given a Boolean expression, the order of operations depends on the precedence rules given by:

1. Parenthesis Highest Priority
2. NOT
3. AND
4. OR Lowest Priority

- Example: XY + WZ will be evaluated as:

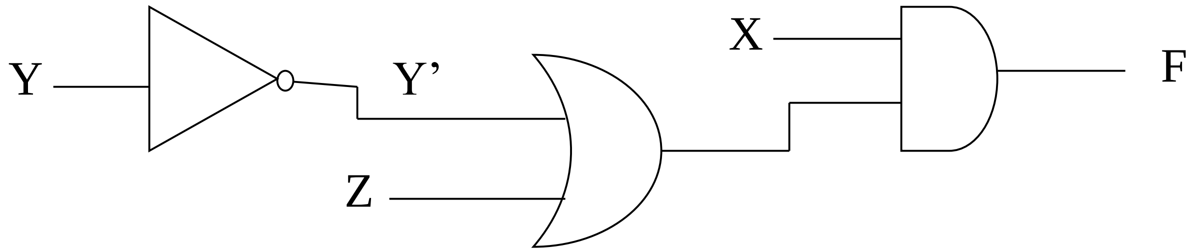
1. XY
2. WZ
3. XY + WZ

Example

F = X . (Y' + Z)

This function has three inputs X, Y, Z and the output is given by F

As can be seen, the gates needed to construct this circuit are: 2 input AND, 2 input OR and NOT



Example (Cont.)

A Boolean function can be represented with a truth table

F = X . (Y' + Z)					
X	Y	Z	Y'	Y' + Z	F=X.(Y'+Z)
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1

Identities of Boolean Algebra

Basic Identities of Boolean Algebra

1.	$X + 0 = X$	2.	$X \cdot 1 = X$	
3.	$X + 1 = 1$	4.	$X \cdot 0 = 0$	
5.	$X + X = X$	6.	$X \cdot X = X$	
7.	$X + \overline{X} = 1$	8.	$X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$			
10.	$X + Y = Y + X$	11.	$XY = YX$	Commutative
12.	$X + (Y + Z) = (X + Y) + Z$	13.	$X(YZ) = (XY)Z$	Associative
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	Distributive
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan's

src: Mano's Textbook

DeMorgan's Theorem

$$\overline{X+Y} = \overline{X} \bullet \overline{Y} \longleftrightarrow \overline{X \bullet Y} = \overline{X} + \overline{Y}$$

Truth Tables to Verify DeMorgan's Theorem

A)	X	Y	X + Y	$\overline{X+Y}$
	0	0	0	1
	0	1	1	0
	1	0	1	0
	1	1	1	0

B)	X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X} \cdot \overline{Y}$
	0	0	1	1	1
	0	1	1	0	0
	1	0	0	1	0
	1	1	0	0	0

use truth tables to prove that two Boolean expressions are equal !

Extended DeMorgan's Theorem: $\overline{X_1 + X_2 + \ldots. + X_n} = \overline{X_1}.\overline{X_2}.\ldots.\overline{X_n}$

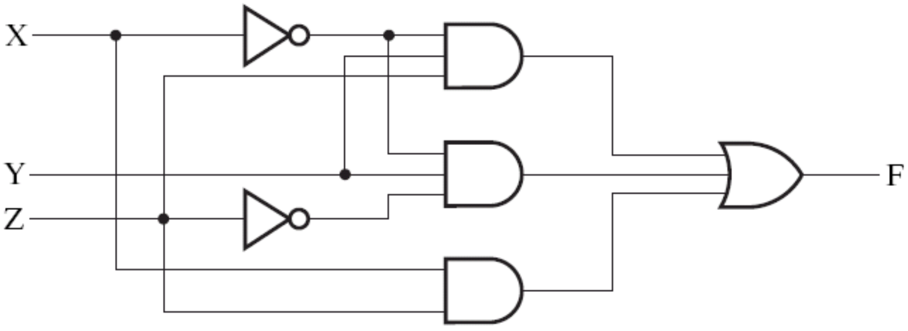
$\overline{X_1 X_2 \ldots. X_n} = \overline{X_1} + \overline{X_2} + \ldots. + \overline{X_n}$

Why Boolean Algebra?

- Boolean algebra identities and properties help reduce the size of expressions
- In effect, smaller sized expressions will require fewer logic gates for building the circuit
- As a result, less cost will be incurred for building simpler circuits
- The speed of simpler circuits is also high

Algebraic Manipulation (Example)

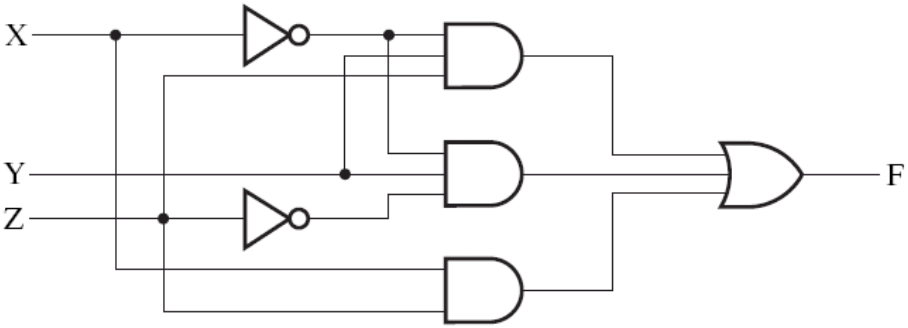
F = X'YZ + X'YZ' + XZ



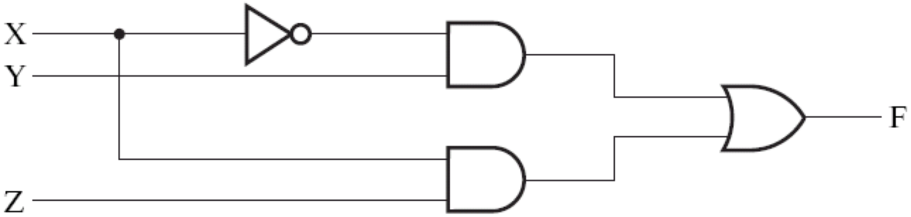
(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$

Algebraic Manipulation (Example)

F = X’YZ + X’YZ’ + XZ
= X’Y(Z+Z’) + XZ (id 14)
= X’Y.1 + XZ (id 7)
= X’Y + XZ (id 2)



(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$



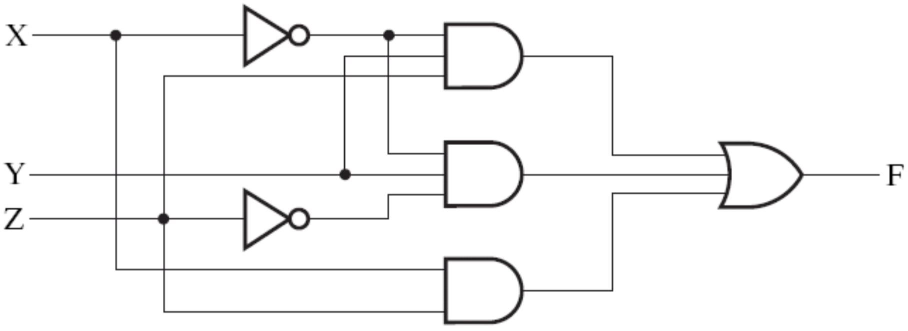
(b) $F = \overline{X}Y + XZ$

Algebraic Manipulation (Example)

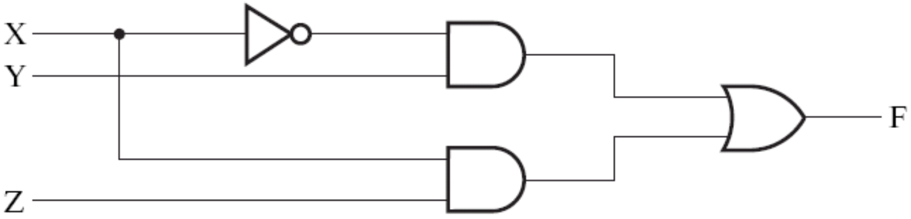
$F = X'YZ + X'YZ' + XZ$
$= X'Y(Z+Z') + XZ$ (id 14)
$= X'Y.1 + XZ$ (id 7)
$= X'Y + XZ$ (id 2)

X	Y	Z	(a) F	(b) F
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Verify !



(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$



(b) $F = \overline{X}Y + XZ$

Example

Reduce $F1=\overline{(A + \overline{B} + AB)}\ \overline{(AB + \overline{A}C + BC)}$

Using DeMorgan’s Theorem,

F1 = (A’.B.(A’+B’)).(A’+B’).(A+C’).(B’+C’)

= (A’.B.A’ + A’.B.B’).(A’+B’)(A+C’) .(B’+C’)

= (A’B + 0).(A’+B’)(A+C’) .(B’+C’)

= (A’BA’ + A’BB’) (A+C’) .(B’+C’)

= (A’B) (A+C’) .(B’+C’)

= (A’BA+A’BC’)(B’+C’)

= (0+A’BC’)(B’+C’)

= (A’BC’B’ + A’BC’C’)

= (0 + A’BC’) = <u>A’BC’</u>

Example

Simplify $G = \overline{(\overline{(A+\overline{B}+C)}.(\overline{AB}+\overline{C}\overline{D})+(\overline{ACD}))}$

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

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

$= (ACD+A\overline{B}CD) + (ABCD+ABCD)$

$= (ACD +ACD(\overline{B}+B) + ABCD)$

$= (ACD + ACD + ABCD)$

$= (ACD + ABCD)$

$= (ACD(1+B))$

$= \underline{ACD}$

Minterms

- A **product term** is a term where literals are ANDed.
  - Example: x'y', xz, xyz, …
- A **minterm** is a product term in which all variables appear exactly once, in normal or complemented form
  - Example: F(x,y,z) has 8 minterms: x'y'z', x'y'z, x'yz', ...
- In general, a function with n variables has 2^n minterms
- A minterm equals 1 at exactly one input combination and is equal to 0 otherwize
  - Example: x'y'z' = 1 only when x=0, y=0, z=0
- A minterm is denoted as m_i where i corresponds the input combination at which this minterm is equal to 1

Minterms

Minterms for Three Variables

Src: Mano's book

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

Variable complemented if 0
Variable uncomplemented if 1

m_i indicated the i^{th} minterm
i indicates the binary combination
m_i is equal to 1 for ONLY THAT combination

Maxterms

- A **sum term** is a term where literals are ORed.
  - Example: x’+y’, x+z, x+y+z, …
- A **maxterm** is a sum term in which all variables appear exactly once, in normal or complemented form
  - Example: F(x,y,z) has 8 maxterms: (x+y+z), (x+y+z’), (x+y’+z), ...
- In general, a function with n variables has 2^n maxterms
- A maxterm equals 0 at exactly one input combination and is equal to 1 otherwize
  - Example: (x+y+z) = 0 only when x=0, y=0, z=0
- A maxterm is denoted as M_i where i corresponds the input combination at which this maxterm is equal to 0

Maxterms

Src: Mano's book

Maxterms for Three Variables

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\overline{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\overline{Y}+\overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X}+Y+\overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X}+\overline{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	M_7	1	1	1	1	1	1	1	0

Variable complemented if 1
Variable not complemented if 0

M_i indicated the i^{th} maxterm
i indicates the binary combination
M_i is equal to 0 for ONLY THAT combination

Minterms and Maxterms

In general, a function of n variables has

- 2^n minterms: $m_0, m_1, \ldots, m_{2^n-1}$
- 2^n maxterms: $M_0, M_1, \ldots, M_{2^n-1}$

Minterms and maxterms are the complement of each other!

$$M_i = \overline{m_i} \qquad \forall i = 0, 1, 2, \ldots\ldots, (2^n - 1)$$

Example: F(X,Y):

$m_2 = XY'$ ➔ $m_2' = X' + Y = M_2$

Expressing Functions with Minterms

- A Boolean function can be expressed algebraically from a give truth table by forming the logical sum (OR) of ALL the minterms that produce 1 in the function

Example:

Consider the function defined by the truth table

X	Y	Z	m	F
0	0	0	m_0	1
0	0	1	m_1	0
0	1	0	m_2	1
0	1	1	m_3	0
1	0	0	m_4	0
1	0	1	m_5	1
1	1	0	m_6	0
1	1	1	m_7	1

$$F(X,Y,Z) = X'Y'Z' + X'YZ' + XY'Z + XYZ$$
$$= m_0 + m_2 + m_5 + m_7$$
$$= \Sigma m(0,2,5,7)$$

Expressing Functions with Maxterms

- A Boolean function can be expressed algebraically from a give truth table by forming the logical product (AND) of ALL the maxterms that produce 0 in the function

Example:

Consider the function defined by the truth table

$F(X,Y,Z) = \Pi\ M(1,3,4,6)$

X	Y	Z	M	F	F′
0	0	0	M_0	1	0
0	0	1	M_1	0	1
0	1	0	M_2	1	0
0	1	1	M_3	0	1
1	0	0	M4	0	1
1	0	1	M_5	1	0
1	1	0	M_6	0	1
1	1	1	M_7	1	0

Applying DeMorgan

$F' = m_1 + m_3 + m_4 + m_6$

$= \Sigma m(1,3,4,6)$

$F = F'' = [m_1 + m_3 + m_4 + m_6]'$

$= m_1'.m_3'.m_4'.m_6'$

$= M_1.M_3.M_4.M_6$

$= \Pi\ M(1,3,4,6)$

Note the indices in this list are those that are missing from the previous list in $\Sigma m(0,2,5,7)$

Sum of Minterms vs Product of Maxterms

- A Boolean function can be expressed algebraically as:
  - The sum of minterms
  - The product of maxterms
- Given the truth table, writing F as
  - $\sum m_i$ – for all minterms that produce 1 in the table, or
  - $\prod M_i$ – for all maxterms that produce 0 in the table
- Minterms and Maxterms are complement of each other.

Example

- Write E = Y′ + X′Z′ in the form of Σm_i and ΠM_i?

- Solution: **<u>Method1</u>**

First construct the Truth Table as shown

Second:

$E = \Sigma m(0,1,2,4,5)$, and

$E = \Pi M(3,6,7)$

X	Y	Z	m	M	E
0	0	0	m_0	M_0	1
0	0	1	m_1	M_1	1
0	1	0	m_2	M_2	1
0	1	1	m_3	M_3	0
1	0	0	m_4	M4	1
1	0	1	m_5	M_5	1
1	1	0	m_6	M_6	0
1	1	1	m_7	M_7	0

Example (Cont.)

Solution: **Method2_a**

$E = Y' + X'Z'$

$= Y'(X+X')(Z+Z') + X'Z'(Y+Y')$

$= (XY'+X'Y')(Z+Z') + X'YZ'+X'Z'Y'$

$= XY'Z+X'Y'Z+XY'Z'+X'Y'Z'+X'YZ'+X'Z'Y'$

$= m_5 + m_1 + m_4 + m_0 + m_2 + m_0$

$= m_0 + m_1 + m_2 + m_4 + m_5$

$= \Sigma m(0,1,2,4,5)$

To find the form ΠMi, consider the remaining indices

$E = \Pi M(3,6,7)$

Solution: **Method2_b**

$E = Y' + X'Z'$

$E' = Y(X+Z)$

$= YX + YZ$

$= YX(Z+Z') + YZ(X+X')$

$= XYZ+XYZ'+X'YZ$

$E = (X'+Y'+Z')(X'+Y'+Z)(X+Y'+Z')$

$= M_7 \,.\, M_6 \,.\, M_3$

$= \Pi M(3,6,7)$

To find the form Σm_i, consider the remaining indices

$E = \Sigma m(0,1,2,4,5)$

Example

Question: F (a,b,c,d) = $\sum m(0,1,2,4,5,7)$, What are the minterms and maxterms of F and and its complement $\overline{F}$?

Solution:

F has 4 variables; $2^4 = 16$ possible minterms/maxterms

$$F(a,b,c,d) = \sum m(0,1,2,4,5,7)$$
$$= \prod M(3,6,8,9,10,11,12,13,14,15)$$

$$\overline{F}(a,b,c,d) = \sum m(3,6,8,9,10,11,12,13,14,15)$$
$$= \prod M(0,1,2,4,5,7)$$

Canonical Forms

The sum of minterms and the product of maxterms forms are known as the **canonical forms** (الصيغ القانونية) of a function.

Standard Forms

- Sum of Products (SOP) and Product of Sums (POS) are also standard forms
  - AB+CD = (A+C)(B+C)(A+D)(B+D)
- The sum of minterms is a special case of the SOP form, where all product terms are minterms
- The product of maxterms is a special case of the POS form, where all sum terms are maxterms

SOP and POS Conversion

SOP → POS

F = AB + CD

= (AB+C)(AB+D)

= (A+C)(B+C)(AB+D)

= (A+C)(B+C)(A+D)(B+D)

Hint 1: Use X+YZ=(X+Y)(X+Z)

Hint 2: Factor

POS → SOP

F = (A'+B)(A'+C)(C+D)

= (A'+BC)(C+D)

= A'C+A'D+BCC+BCD

= A'C+A'D+BC+BCD

= A'C+A'D+BC

Hint 1: Use i (X+Y)(X+Z)=X+YZ

Hint 2: Multiply

Question1: How to convert SOP to sum of minterms?
Question2: How to convert POS to product of maxterms?

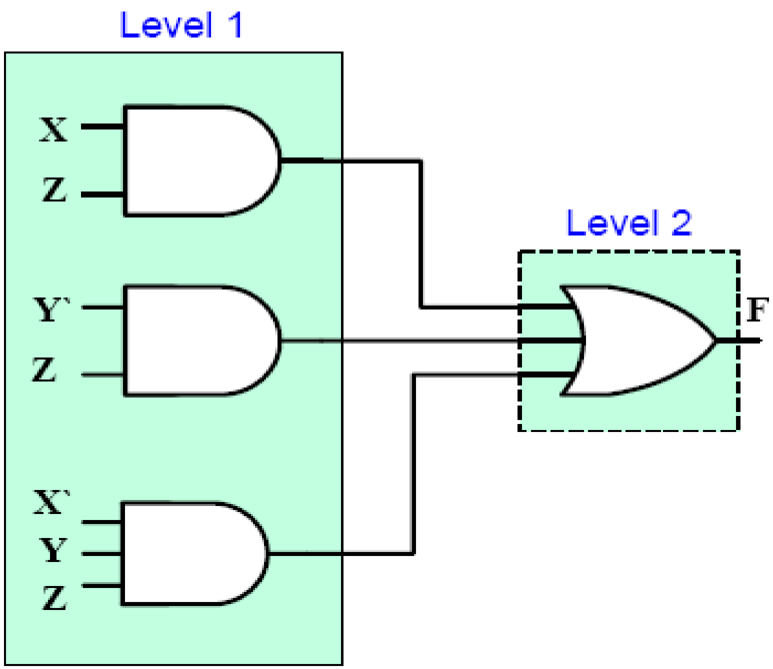
Implementation of SOP

Any SOP expression can be implemented using a

2-levels of gates

The 1st level consists of AND gates, and the 2nd level consists of a single OR gate

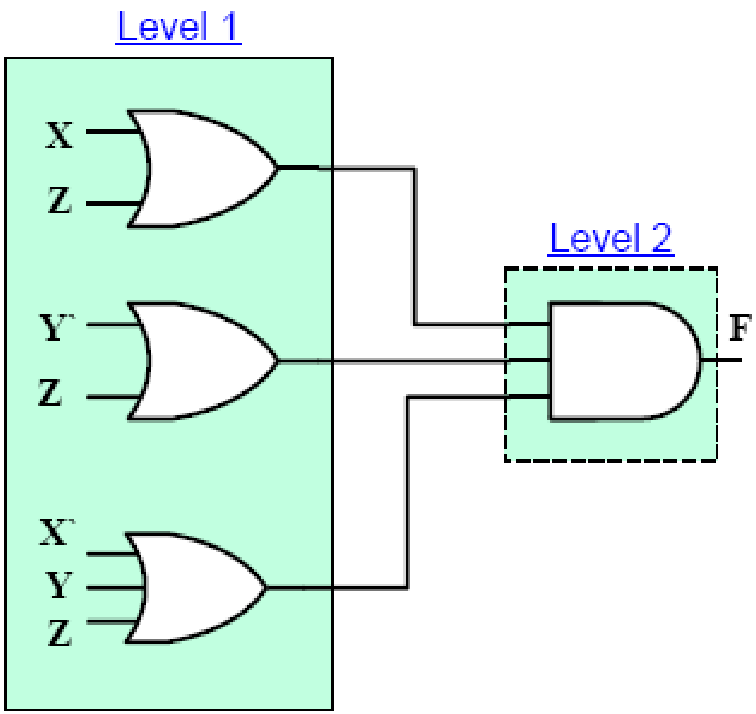
Also called 2-level Circuit



Two-Level Implementation (F = XZ + Y`Z + X`YZ)
Level-1: AND-Gates ; Level-2: One OR-Gate

Implementation of POS

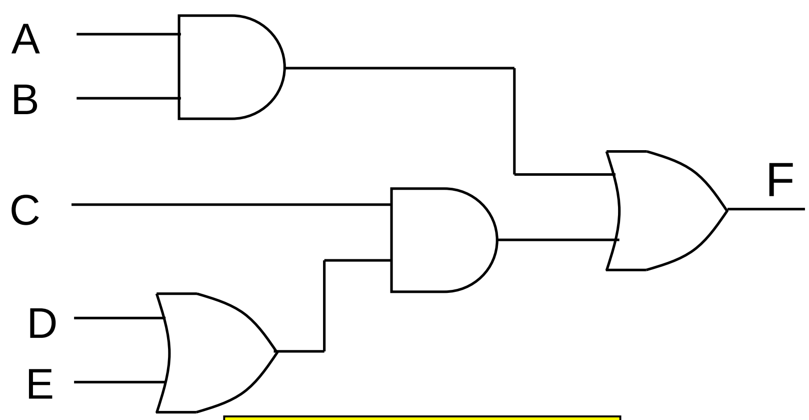
- Any POS expression can be implemented using a
  2-levels of gates
- The 1st level consists of OR gates, and the 2nd level consists of a single AND gate
- Also called 2-level Circuit



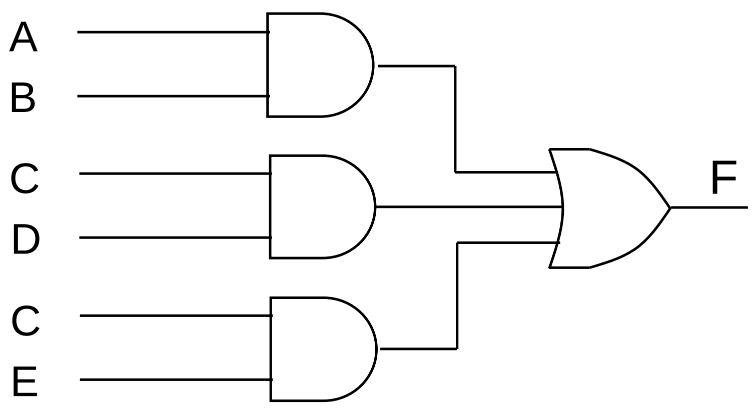
Two-Level Implementation {F = (X+Z)(Y`+Z)(X`+Y+Z)}
Level-1: OR-Gates ; Level-2: One AND-Gate

Implementation of SOP

- Consider F = AB + C(D+E)
  - This expression is NOT in the sum-of-products form
  - Use the identities/algebraic manipulation to convert to a standard form (sum of products), as in F = AB + CD + CE
- Logic Diagrams:



3-level circuit



2-level circuit