CANONICAL FORM, STANDARD FORM AND NON-STANDARD FORM

Digital logic design

Iqra Chaudhary (Lecturer CS dept. NUML)

Minterms and maxterms for 3-variable function

x	y	z	F	minterm		maxterm	
0	0	0		$x'y'z'$	$m0$	$x+y+z$	$M0$
0	0	1		$x'y'z$	$m1$	$x+y+z'$	$M1$
0	1	0		$x'yz'$	$m2$	$x+y'+z$	$M2$
0	1	1		$x'yz$	$m3$	$x+y'+z'$	$M3$
1	0	0		$xy'z'$	$m4$	$x'+y+z$	$M4$
1	0	1		$xy'z$	$m5$	$x'+y+z'$	$M5$
1	1	0		xyz'	$m6$	$x'+y'+z$	$M6$
1	1	1		xyz	$m7$	$x'+y'+z'$	$M7$

Canonical form: Sum of Minterms and product of maxterms form

Sum of Minterms:

F1=$x'y'z'+xy'z'+xy'z+xyz'+x'yz'$

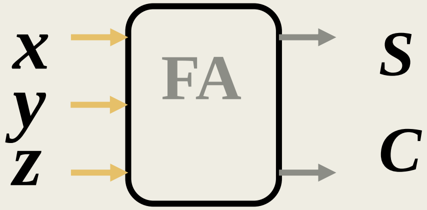
Product of maxterms:

F1=$(x+y+z')\cdot(x+y'+z')\cdot(x'+y'+z')$

x	y	z	F	minterm		maxterm	
0	0	0	1	$x'y'z'$	$m0$	$x+y+z$	$M0$
0	0	1	0	$x'y'z$	$m1$	$x+y+z'$	$M1$
0	1	0	1	$x'yz'$	$m2$	$x+y'+z$	$M2$
0	1	1	0	$x'yz$	$m3$	$x+y'+z'$	$M3$
1	0	0	1	$xy'z'$	$m4$	$x'+y+z$	$M4$
1	0	1	1	$xy'z$	$m5$	$x'+y+z'$	$M5$
1	1	0	1	xyz'	$m6$	$x'+y'+z$	$M6$
1	1	1	0	xyz	m7	$x'+y'+z'$	$M7$

Specification Given: Implement Adder that can add three bits (knows as full adder)

- Full Adder
  - *Adds 1-bit plus 1-bit plus 1-bit*
  - *Produces Sum and Carry*



$$\begin{array}{cc} & x \\ + & y \\ + & z \\ \hline C & S \end{array}$$

Implement Full adder

- Step 1: Number of input=3 and Number of output=2
- Step 2: Drive the truth table
- Step 3: Obtain the equation from the truth table and **simplify it**

$$S = xy'z' + x'yz' + x'y'z + xyz = x \oplus y \oplus z$$

	0	1	0	1
x	1	0	1	0

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

	0	0	1	0
x	0	1	1	1

$$C = xy + xz + yz$$

C= $x'yz + xy'z + xyz' + xyz = z(x'y + xy') + xy(z + z') = z(x \oplus y) + xy$

Step 4: Draw the circuit diagram from simplified expression

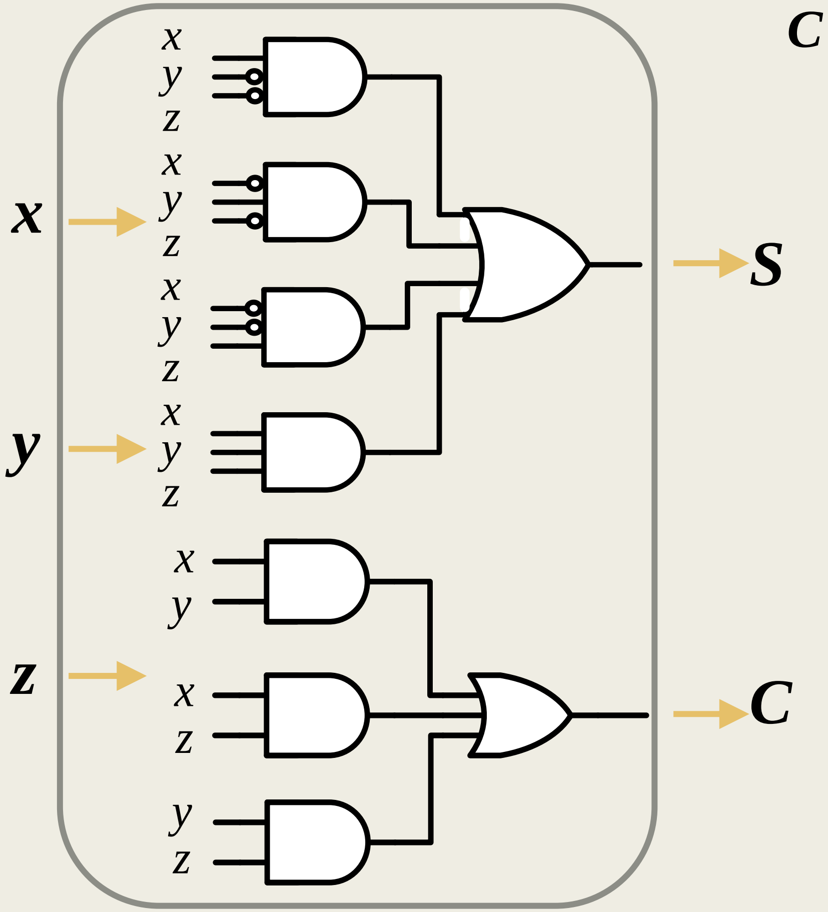
x	y	z	S	C	minterm
0	0	0	0	0	x'y'z'
0	0	1	1	0	x'y'z
0	1	0	1	0	x'yz'
0	1	1	0	1	x'yz
1	0	0	1	0	xy'z'
1	0	1	0	1	xy'z
1	1	0	0	1	xyz'
1	1	1	1	1	xyz

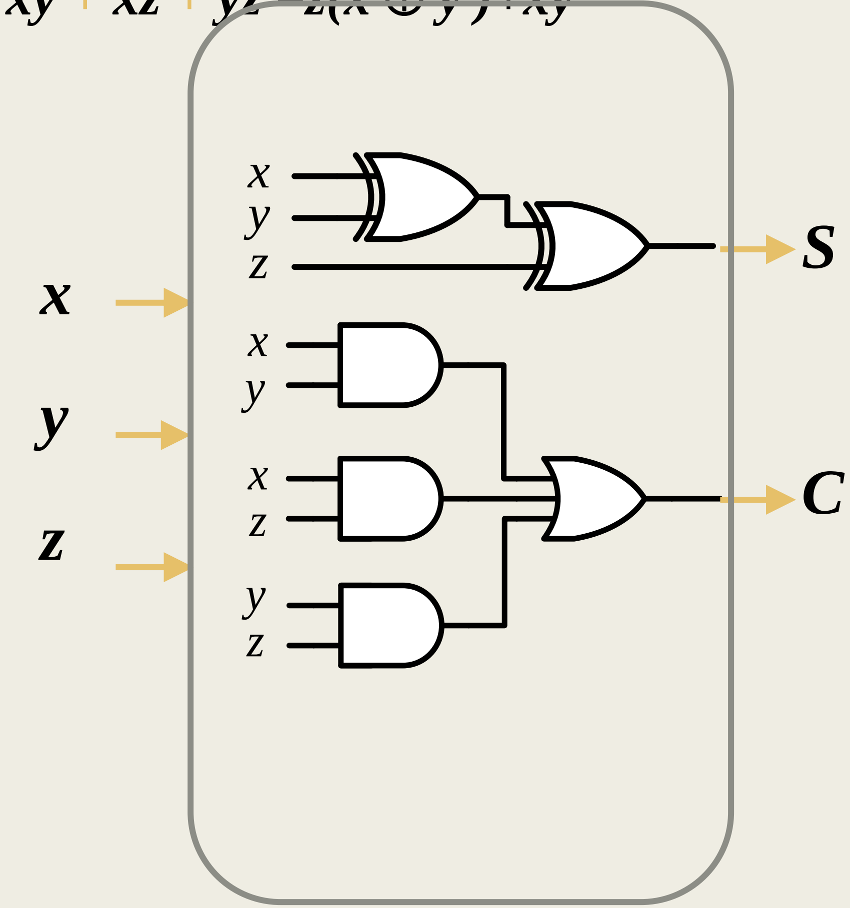
Implement Full adder

- Circuit diagram:

$$S = xy'z' + x'yz' + x'y'z + xyz = x \oplus y \oplus z$$

$$C = xy + xz + yz = z(x \oplus y) + xy$$





Multiple-variable exclusive-OR operation is defined as an *odd function.*

Specification Given: 3bits binary to gray code convertor

x y z	F1	F2	F3	minterm	maxterm
0 0 0				$x'y'z'$	$x+y+z$
0 0 1				$x'y'z$	$x+y+z'$
0 1 0				$x'yz'$	$x+y'+z$
0 1 1				$x'yz$	$x+y'+z'$
1 0 0				$xy'z'$	$x'+y+z$
1 0 1				$xy'z$	$x'+y+z'$
1 1 0				xyz'	$x'+y'+z$
1 1 1				xyz	$x'+y'+z'$

Canonical Form:Sum of minterms

- Example: Function is expressed as sum of minterms

  $F=x'y'z+x'yz+xy'z'+xy'z=m1+m3+m4+m5 =\sum m(1,3,4,5)$

x	y	z	F	minterm	
0	0	0	0	$x'y'z'$	$m0$
0	0	1	1	$x'y'z$	$m1$
0	1	0	0	$x'yz'$	$m2$
0	1	1	1	$x'yz$	$m3$
1	0	0	1	$xy'z'$	$m4$
1	0	1	1	$xy'z$	$m5$
1	1	0	0	xyz'	$m6$
1	1	1	0	xyz	$m7$

Function (F) = sum of all minterms where truth table has value 1

Canonical Form:Sum of minterms

Canonical form: Sum of minterms

- Example: F=m1+m4+m5+m6+m7
  F=∑m(1,4,5,6,7)

- Is this the only canonical form?
  - No, other forms exist (dual form, etc.)

A	*B*	*C*	*F*	minterm	design-ation
0	0	0	0	*A'B'C'*	*m0*
0	0	1	1	*A'B'C*	*m1*
0	1	0	0	*A'BC'*	*m2*
0	1	1	0	*A'BC*	*m3*
1	0	0	1	*AB'C'*	*m4*
1	0	1	1	*AB'C*	*m5*
1	1	0	1	*ABC'*	*m6*
1	1	1	1	*ABC*	*m7*

Dual Canonical Form: Product of maxterm

Inputs			Output
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Product of maxterm

$$X = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + \overline{C})$$

Sum of minterm

$$X = \overline{A}BC + A\overline{B}\overline{C} + AB\overline{C} + ABC$$

Dual Canonical Form:Product of maxterm

Example:

- F = m1+m4+m5+m6+m7
- F = M0·M2·M3
  = (A+B+C)(A+B’+C)(A+B’+C’)
  – From DeMorgan:
    F’ = A’B’C’+A’BC’+A’BC
    = m0 + m2 + m3

- ***Both canonical forms express same function***

A	*B*	*C*	*F*	Maxterm	Designation
0	0	0	0	*A+B+C*	*M0*
0	0	1	1	*A+B+C’*	*M1*
0	1	0	0	*A+B’+C*	*M2*
0	1	1	0	*A+B’+C’*	*M3*
1	0	0	1	*A’+B+C*	*M4*
1	0	1	1	*A’+B+C’*	*M5*
1	1	0	1	*A’+B’+C*	*M6*
1	1	1	1	*A’+B’+C’*	*M7*

Minterm and Maxterm Relationship

Review: DeMorgan's Theorem:

$$\overline{x \cdot y} = \overline{x} + \overline{y} \qquad \overline{x + y} = \overline{x} \cdot \overline{y}$$

- Two-variable example:

$$M_2 = \overline{x} + y \qquad m_2 = x \cdot \overline{y}$$

Thus M_2 is the complement of m_2 and vice-versa.

- Since DeMorgan's Theorem holds for *n* variables, the above holds for terms of *n* variables

$$M_i = \overline{m_i} \qquad m_i = \overline{M_i}$$

Thus M_i is the complement of m_i.

Index Example in Three Variables

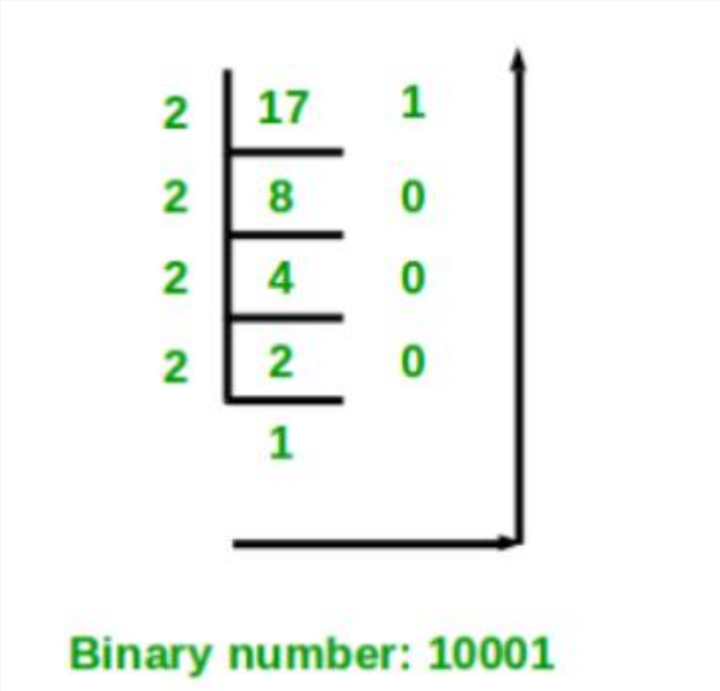
- Example: (for three variables)
- Assume the variables are called X, Y, and Z.
- The standard order is X, then Y, then Z.
- The <u>Index 0</u> (base 10) = 000 (base 2) for three variables). All three variables are complemented for <u>minterm 0</u> ($\overline{X}, \overline{Y}, \overline{Z}$) and no variables are complemented for <u>Maxterm 0</u> (X,Y,Z).
  - *Minterm 0, called* m_0 *is* $\overline{X}\,\overline{Y}\,\overline{Z}$.
  - *Maxterm 0, called* M_0 *is* $(X + Y + Z)$.
  - *Minterm 6 ?*
  - *Maxterm 6 ?*

Index Examples – Four Variables

Index i	Binary Pattern	Minterm m_i	Maxterm M_i
0	0000	$\bar{a}\bar{b}\bar{c}\bar{d}$	$a+b+c+d$
1	0001	$\bar{a}\bar{b}\bar{c}d$	?
3	0011	?	$a+b+\bar{c}+\bar{d}$
5	0101	$\bar{a}b\bar{c}d$	$a+\bar{b}+c+\bar{d}$
7	0111	?	$a+\bar{b}+\bar{c}+\bar{d}$
10	1010	$a\bar{b}c\bar{d}$	$\bar{a}+b+\bar{c}+d$
13	1101	$ab\bar{c}d$	?
15	1111	$abcd$	$\bar{a}+\bar{b}+\bar{c}+\bar{d}$

Minterm Function Example

- $F(A, B, C, D, E) = m_2 + m_9 + m_{17} + m_{23}$
- $F(A, B, C, D, E) =$
  - *m17=10001=AB'C'D'E*



Maxterm Function Example

- $F(A,B,C,D) = M_3 \cdot M_8 \cdot M_{11} \cdot M_{14}$
- F(A, B,C,D) =

Example

EXAMPLE 4–19

Convert the following SOP expression to an equivalent POS expression:

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC$$

Solution

The evaluation is as follows:

$$000 + 010 + 011 + 101 + 111$$

Since there are three variables in the domain of this expression, there are a total of eight (2^3) possible combinations. The SOP expression contains five of these combinations, so the POS must contain the other three which are 001, 100, and 110. Remember, these are the binary values that make the sum term 0. The equivalent POS expression is

$$(A + B + \overline{C})(\overline{A} + B + C)(\overline{A} + \overline{B} + C)$$

Binary Boolean Functions

- Possible combinations:

x	y	F0	F1	F2	F3	F4	F5	F6	F7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
Boolean function		F0=0	F1=xy	F2=xy'	F3=x	F4=x'y	F5=y	F6=xy'+x'y	F7=x+y
Name		Null	AND	Inhibition	Transfer	Inhibition	Transfer	Exclusive-OR	OR

x	y	F8	F9	F10	F11	F12	F13	F14	F15
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
Boolean function		F8=(x+y)'	F9=xy+x'y'	F10=y'	F11=x+y'	F12=x'	F13=x'+y	F14=(xy)'	F15=1
Name		NOR	Equivalence	Complement	Implication	Complement	Implication	NAND	Identify

Boolean Function Optimization

- Canonical Forms (Sum-of-minterms, Product-of-Maxterms), or other standard forms (SOP, POS) differ in complexity
- Minimizing the gate input (or literal) cost of a (a set of) Boolean equation(s) reduces circuit cost.
- Boolean Algebra rules and graphical techniques (K-maps) are tools to minimize cost criteria values.

STANDARD FORMS

Standard forms of Boolean Expressions

- Sum-of-Products form
- Product-of-Sums form

Standard forms of Boolean Expressions

- <u>Standard Sum-of-Products (SOP) form:</u> equations are written as an OR of AND terms

AB + ABC

B+ABC + CDE + $\overline{B}C\overline{D}$

$$\overline{A}B + \overline{A}B\overline{C} + AC$$

- <u>Standard Product-of-Sums (POS) form:</u> equations are written as an AND of OR terms

$$(\overline{A} + B)(A + \overline{B} + C)$$

$$(\overline{A} + \overline{B} + \overline{C})(C + \overline{D} + E)(\overline{B} + C + D)$$

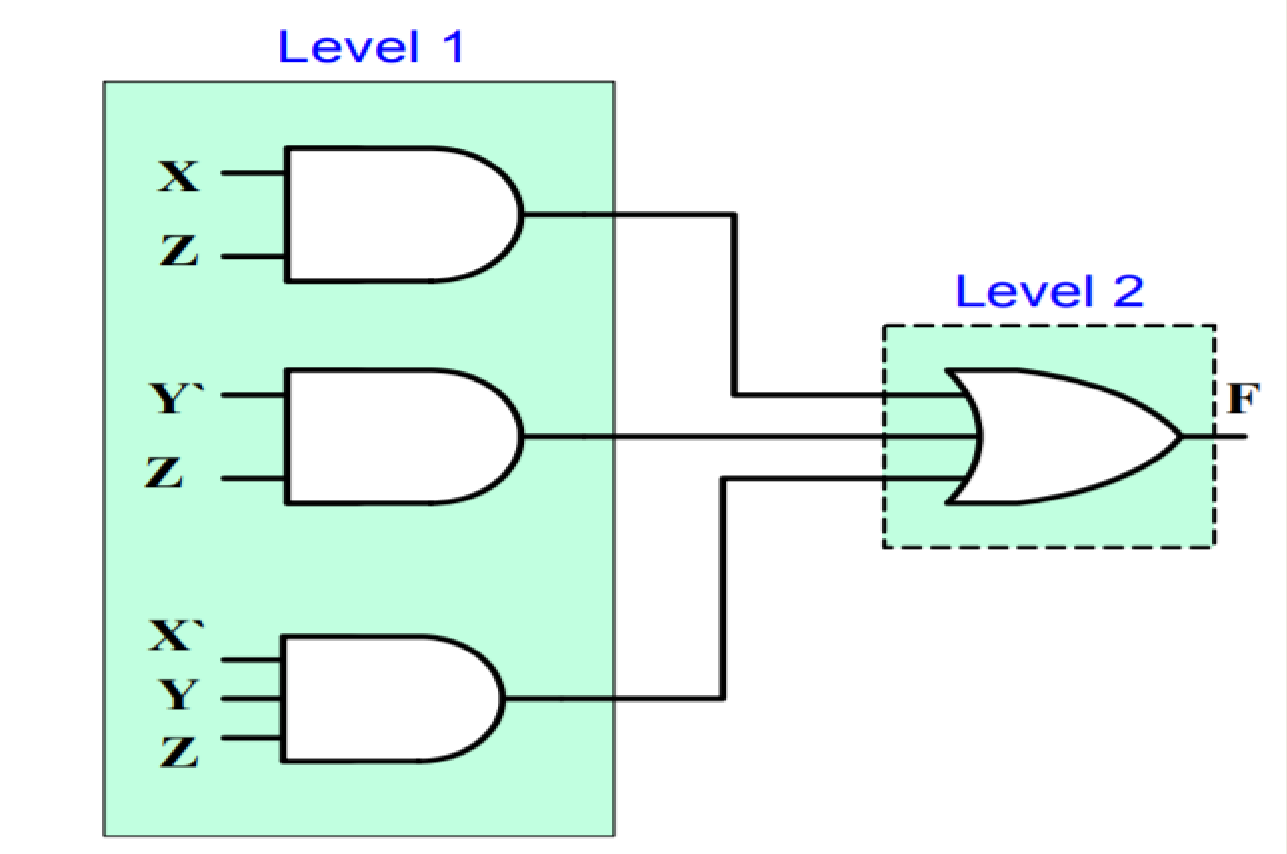
$$(A + B)(A + \overline{B} + C)(\overline{A} + C)$$

Sum of Products Expression (SOP):

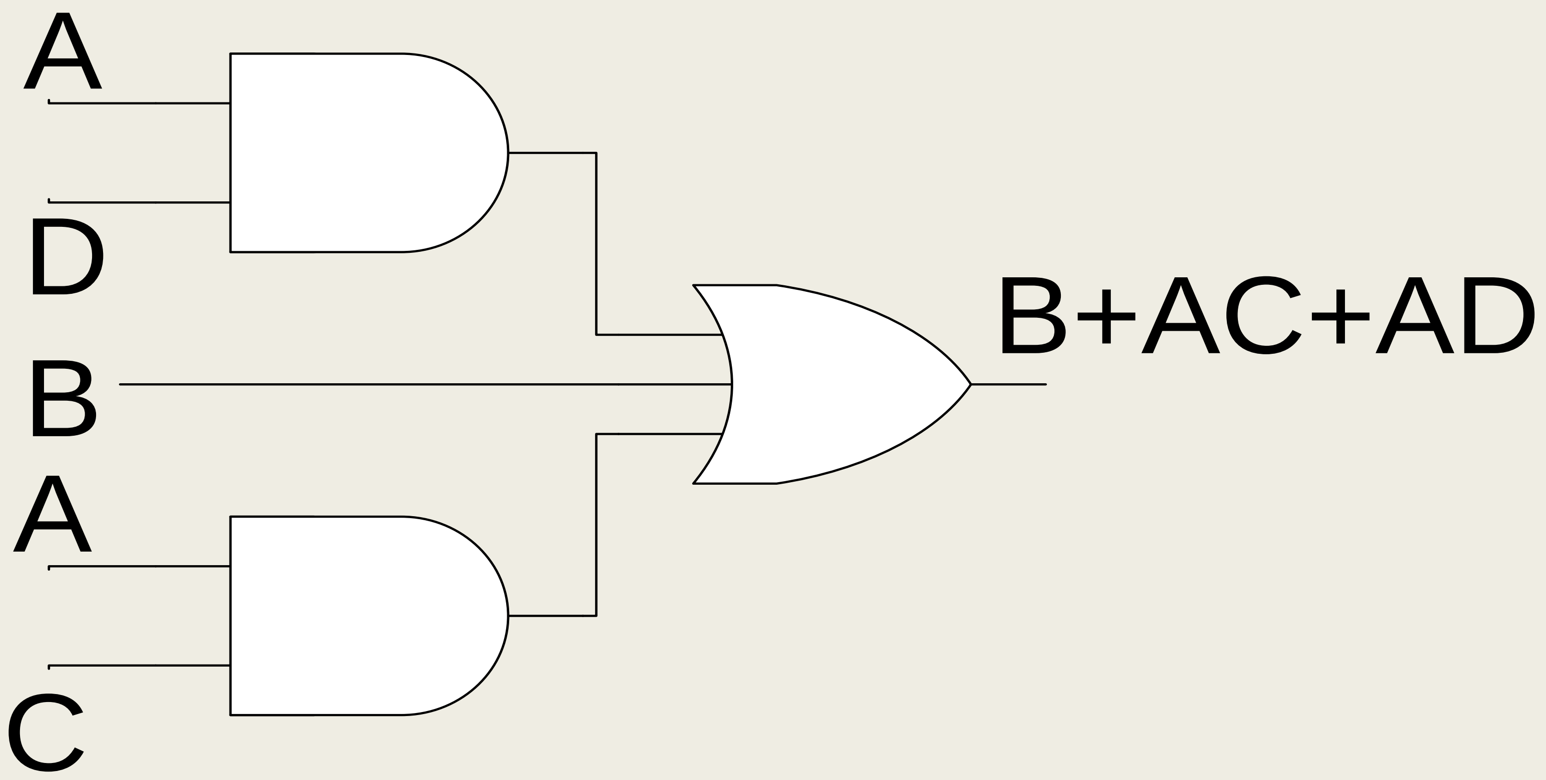
- Any SOP expression can be implemented in 2-levels of gates.
- The first level consists of a number of AND gates which equals the number of product terms in the expression. Each AND gate implements one of the product terms in the expression.
- The second level consists of a SINGLE OR gate whose number of inputs equals the number of product terms in the expression.

Sum of Products Expression (SOP):

- Example Implement the following SOP function F = XZ + Y`Z + X`YZ



Implementation of SOP expression

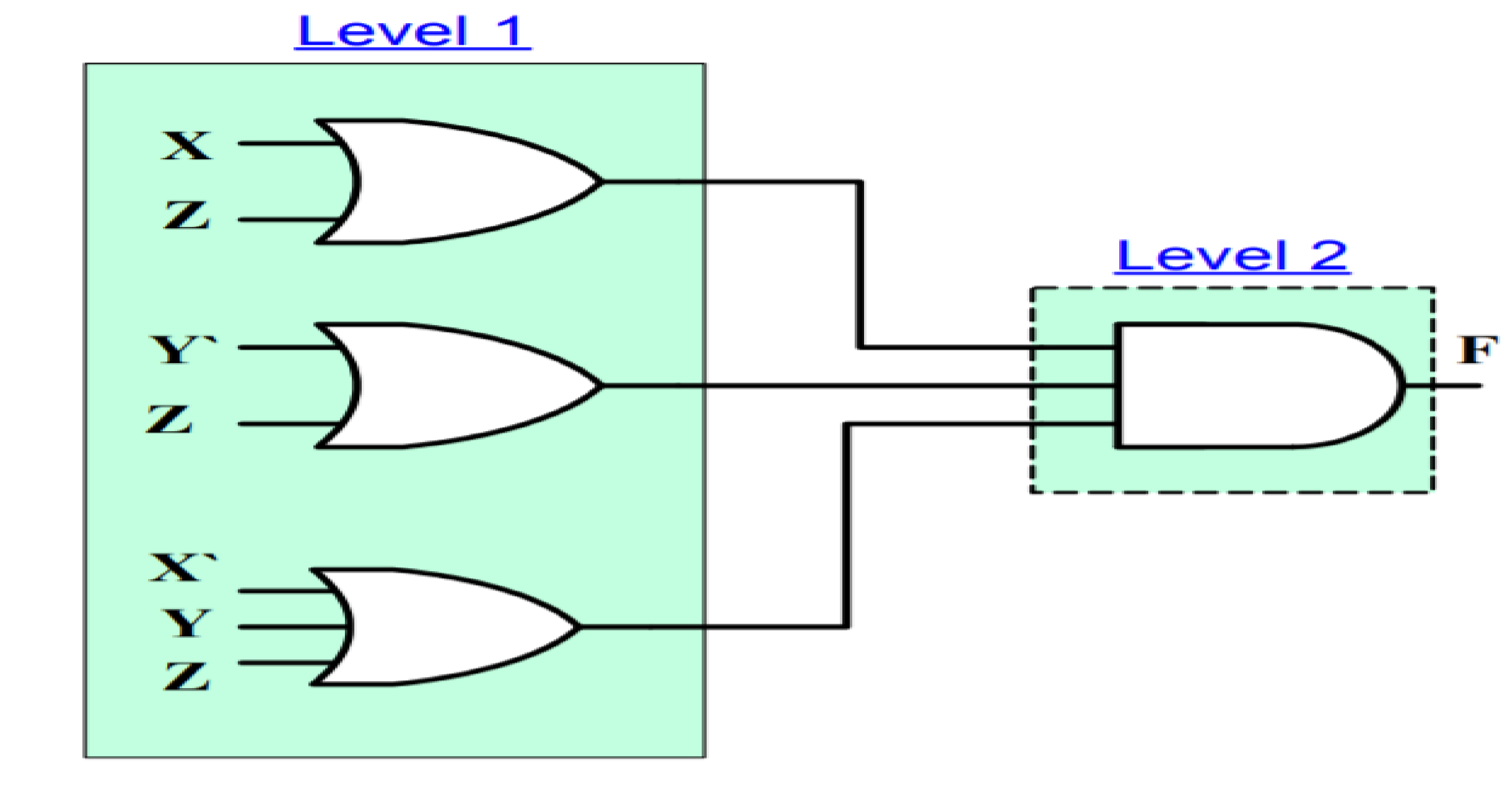


Product of Sums Expression (POS):

- Any POS expression can be implemented in 2-levels of gates
- The first level consists of a number of OR gates which equals the number of sum terms in the expression, each gate implements one of the sum terms in the expression.
- The second level consists of a SINGLE AND gate whose number of inputs equals the number of sum terms.

Product of Sums Expression (POS):

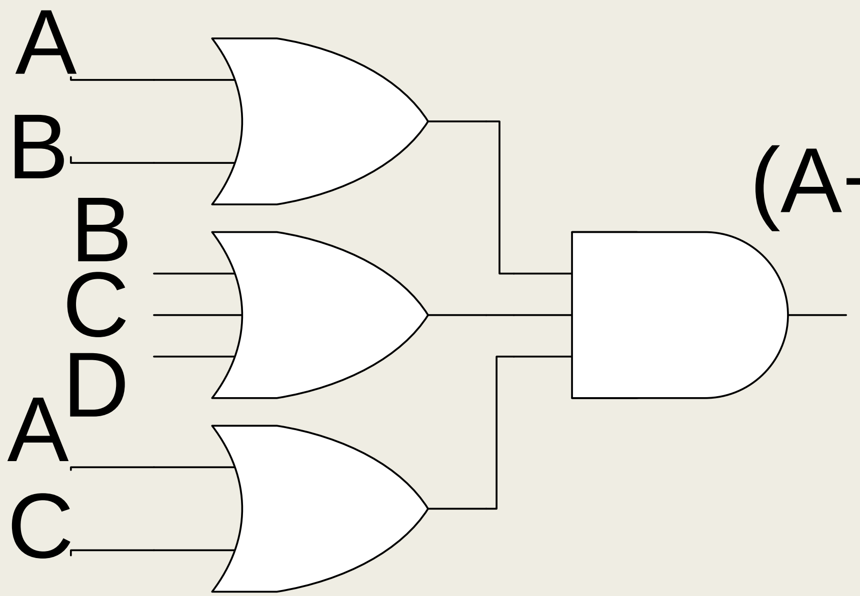
- Example Implement the following SOP function F = (X+Z)(Y` +Z)(X` +Y+Z)



Two-Level Implementation {F = (X+Z)(Y`+Z)(X`+Y+Z)}

Level-1: OR-Gates ; **Level-2:** One AND-Gate

Implementation of POS expression



$(A+B)(B+C+D)(A+C)$

NON-STANDARD FORMS

Non-Standard Forms

- **Examples:**

- These “mixed” forms are <u>neither SOP nor POS</u>

$$A\,B\,C + \overline{A}\,\overline{B}\,(C + B)$$

$$(A + B) \cdot (A + \overline{B} \,.\, \overline{C}) \cdot C$$

$$(A\;B + C)\;(A + C)$$

$$A\,B\,\overline{C} + A\,C\,(A + B)$$

Conversion of general expression to SOP form

$$AB + B(CD + EF) = AB + BCD + BEF$$

$$(A + B)(B + C + D) = AB + AC + AD + B + BC + BD$$

$$= AC + AD + B$$

$$\overline{\overline{(A + B)} + C} = \overline{\overline{(A + B)}}\,\overline{C} = (A + B)\overline{C} = A\overline{C} + B\overline{C}$$

CONVERSION INTO CANONICAL FORM

Conversion into Canonical

- Any Boole function can be expressed as a Sum of Minterms.
  - *For the function table, the minterms used are the terms corresponding to the 1's*
  - *For expressions, expand all terms first to explicitly list all minterms. Do this by "ANDing" any term missing a variable v with a term (*$v + \bar{v}$ *).*
- Example: Implement $f = x + \bar{x}\ \bar{y}$ as a sum of minterms.

  First expand terms: $f = x(y + \bar{y}) + \bar{x}\ \bar{y}$

  Then distribute terms: $f = xy + x\bar{y} + \bar{x}\,\bar{y}$

  Express as sum of minterms: f = $m_3 + m_2 + m_0$

Conversion into Canonical

$$F = A + \overline{B}C$$

- Example:
- There are three variables, A, B, and C which we take to be the standard order.
- Expanding the terms with missing variables:
- Collect terms (removing all but one of duplicate terms):
- Express as SOM:

Conversion into Canonical

Canonical form: Sum of minterms

- Example: F=A+B'C
  - Expansion of minterms:
    A=ABC+ABC'+AB'C+AB'C'
    B'C=AB'C+A'B'C
  - F = ABC+ABC'+AB'C+AB'C'+A'B'C
- Alternate forms:
  - F=m1+m4+m5+m6+m7
  - $$F(A,B,C) = \Sigma m(1,4,5,6,7)$$

A	B	C	F	minterm	design-ation
0	0	0	0	A'B'C'	m0
0	0	1	1	A'B'C	m1
0	1	0	0	A'BC'	m2
0	1	1	0	A'BC	m3
1	0	0	1	AB'C'	m4
1	0	1	1	AB'C	m5
1	1	0	1	ABC'	m6
1	1	1	1	ABC	m7

Converting SOP to Truth Table

- Examine each of the products to determine where the product is equal to a 1.
- Set the remaining row outputs to 0

$$X = \overline{A}B\overline{C} + A\overline{B} + ABC$$

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Conversion into Canonical

- Any Boolean Function can be expressed as a Product of Maxterms (POM).
  - *For the function table, the maxterms used are the terms corresponding to the 0's.*
  - *For an expression, expand all terms first to explicitly list all maxterms. Do this by first applying the second distributive law , "ORing" terms missing variable v with a term equal to* $v \cdot \bar{v}$ *and then applying the distributive law again.*
- Example: Convert to product of maxterms:

$$f(x,y,z) = x + \bar{x}\,\bar{y}$$

Apply the distributive law:

$$x + \bar{x}\,\bar{y} = (x + \bar{x})(x + \bar{y}) = 1 \cdot (x + \bar{y}) = x + \bar{y}$$

Add missing variable z:

$$x + \bar{y} + z \cdot \bar{z} = (x + \bar{y} + z)(x + \bar{y} + \bar{z})$$

Express as POM: $f = M_2 \cdot M_3$

Converting POS to Truth Table

- Opposite process from the SOP
- Each sum term results in a 0
- Set the remaining row outputs to 1

$$X = (\overline{A}+B+\overline{C})(A+\overline{B})(A+B+C)$$

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Thank You