Canonical and Standard Forms

CSE 4205: Digital Logic Design

Aashnan Rahman
Junior Lecturer
Department of Computer Science and Engineering (CSE)
Islamic University of Technology

01

Minterms and Maxterms

ANDed or ORed Terms

Minterms

A product (**AND**ed term) that contains **all the variables** of a particular function in either in **original form** or **complemented form**.

Also known as **Standard Product**.

- **n** variables can be combined using **AND** to form $\mathbf{2^n}$ **minterms**. $(0 \text{ to } 2^n - 1)$
- **Symbol →** m_j
- **1 priority**

1) 0 → complemented form
2) 1 → original form

Minterms

			Minterms	
x	**y**	**z**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0
0	0	1	$x'y'z$	m_1
0	1	0	$x'yz'$	m_2
0	1	1	$x'yz$	m_3
1	0	0	$xy'z'$	m_4
1	0	1	$xy'z$	m_5
1	1	0	xyz'	m_6
1	1	1	xyz	m_7

Maxterms

A sum (**OR**ed term) that contains **all the variables** of a particular function in either in **original form** or **complemented form**.

Also known as **Standard Sum**.

- **n** variables can be combined using **OR** to form **2^n Maxterms**. $(0 \text{ to } 2^n - 1)$
- **Symbol** → M_j
- **0 priority**
    1) 1 → complemented form
    2) 0 → original form

Maxterms

- **Maxterm** and its **corresponding Minterm** are **complements** to each other.

$$m_j = M_j'$$

Maxterms

			Maxterms	
x	**y**	**z**	**Term**	**Designation**
0	0	0	$x + y + z$	M_0
0	0	1	$x + y + z'$	M_1
0	1	0	$x + y' + z$	M_2
0	1	1	$x + y' + z'$	M_3
1	0	0	$x' + y + z$	M_4
1	0	1	$x' + y + z'$	M_5
1	1	0	$x' + y' + z$	M_6
1	1	1	$x' + y' + z'$	M_7

Boolean Functions using Minterms (SoP)

Any boolean function can be expressed as a sum of minterms (or product of maxterms).

From a given truth table, **minterms** are produced from those combinations of variables which produces **1 (True)** .

The required function is the **AND** or **sum** of the **minterms** having **output 1**.

Boolean Functions using Minterms (SoP)

x	y	z	Function f_1		Function f_2		minterms
0	0	0	0		0		m_0
0	0	1	1	$x'y'z$	0		m_1
0	1	0	0		0		m_2
0	1	1	0		1	$x'yz$	m_3
1	0	0	1	$xy'z'$	0		m_4
1	0	1	0		1	$xy'z$	m_5
1	1	0	0		1	xyz'	m_6
1	1	1	1	xyz	1	xyz	m_7

Boolean Functions using Minterms (SoP)

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

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

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

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

Boolean Functions using Maxterms (PoS)

Any boolean function can be expressed as a product of maxterms (or sum of minterms).

From a given truth table, **maxterms** are produced from those combinations of variables which produces **0 (False)** .

The required function is the **OR** or **product** of the **maxterms** having **output 0**.

Boolean Functions using Minterms (SoP)

x	y	z	Function f_1		Function f_2		maxterm
0	0	0	0	$x+y+z$	0	$x+y+z$	M_0
0	0	1	1		0	$x+y+z'$	M_1
0	1	0	0	$x+y'+z$	0	$x+y'+z$	M_2
0	1	1	0	$x+y'+z'$	1		M_3
1	0	0	1		0	$x'+y+z$	M_4
1	0	1	0	$x'+y+z'$	1		M_5
1	1	0	0	$x'+y'+z$	1		M_6
1	1	1	1		1		M_7

Boolean Functions using Minterms (SoP)

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$f_1 = (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z)(x + y' + z')$$
$$= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$
$$= \prod(0, 2, 3, 5, 6)$$

$$f_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z)$$
$$= M_0 M_1 M_2 M_4$$
$$= \prod(0, 1, 2, 4)$$

Complement of Boolean Functions

Taking each combination of variables (**minterms**) that produces 0 (**False**) in the function and **OR** them.

Or, taking each combination of variables (**maxterms**) that produces 1 (**True**) in the function and **AND** them.

02

Standard and Canonical Form

SoP and Pos Forms

Canonical Form

A Boolean expression is in **canonical form** when **each term** in the expression contains **all the variables** in the domain (either complemented or not).

There exists two such forms

- Sum of minterms / Products (SoP)
- Product of maxterms / Sums (PoS)

Sum of Minterms

If the function is not in this form,

- The expression is expanded into a **sum of ANDed terms**.
- Each term is inspected whether it **contains all the variables**, if missing it is introduced by **ANDing** the term with (x + x′).

$$F = A + B'C$$

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

$$= m_1 + m_4 + m_5 + m_6 + m_7$$ **Notation 1**

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$ **Notation 2**

Sum of Minterms

A	B	C	F
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

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

Product of Maxterms

If the function is not in this form,

- The expression is expressed into a **product of ORed terms**.
- Each term is inspected whether it **contains all the variables**, if missing it is introduced by **ORing** the term with (x.x′).

$$F = xy + x'z$$

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0 M_2 M_4 M_5 \end{aligned}$$

Notation 1

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Notation 2

Conversion between Canonical Forms

The complement of a function expressed as SOP equals SOP of missing minterms from the original function

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

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

If the complement of $\boldsymbol{F'}$ is taken following **De Morgan's Law**, we will get the original function $\boldsymbol{F}$ in different form

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

Form this conversion, it is proved that:

$$m_j = M_j'$$

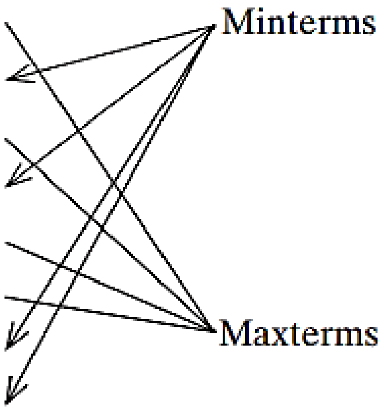
Conversion between Canonical Forms

$F = xy + x'z$

$F(x, y, z) = \Sigma(1, 3, 6, 7)$

$F(x, y, z) = \Pi(0, 2, 4, 5)$

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



Standard Form

- **Canonical forms:**
  - Both are easily formed from truth table
  - **Each minterm or maxterm must contain all variables; either primed or unprimed**
- **Standard form:**
  - Terms may have **one**, **two**, **three** or **any** number of literals.
  - **Two types:** SOP and POS

Standard Form (SoP)

- Has **ANDed** terms (**products**) which are finally **ORed** (**sum**)
- Logic diagrams contain a group of **AND gates followed** by a **single OR gate**. It's assumed that the complements of variables are directly available in their input. **Known as two-level-implementation**

SOP: $F_1 = y' + xy + x'yz'$

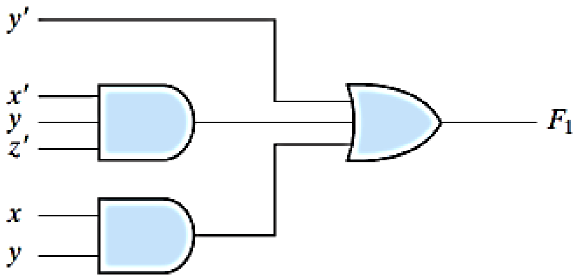
Standard Form (PoS)

- Has **ORed** terms (**sums**) which are finally **ANDed** (**product**)
- Logic diagrams contain a group of **OR gates followed** by a **single AND gate**. It's assumed that the complements of variables are directly available in their input. **Another two-level-implementation**

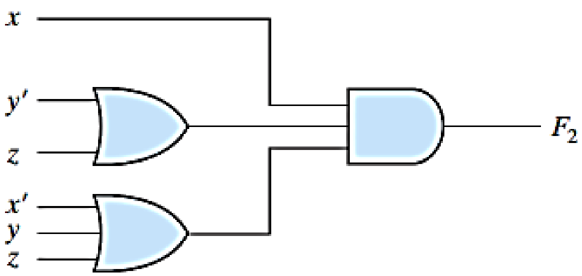
POS: $F_2 = x(y' + z)(x' + y + z')$

SoP and PoS

Two level implementation



(a) Sum of Products



(b) Product of Sums

Non -Standard Form

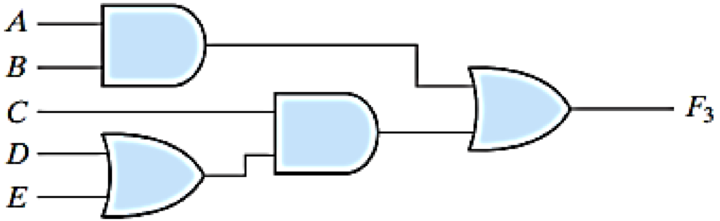
- The boolean expression is neither in SoP or PoS form.

Nonstandard Form: $F_3 = AB + C(D + E)$

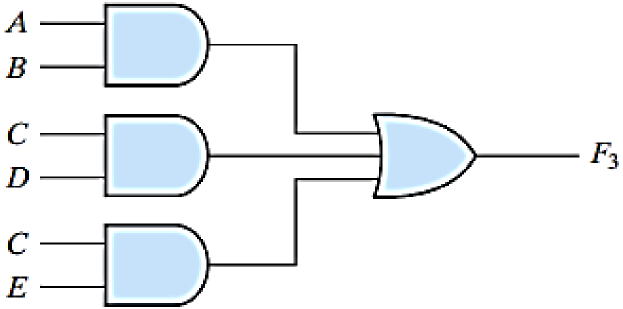
Standard Form: $F_3 = AB + C(D + E) = AB + CD + CE$

Can be converted to standard form following the distributive law.

Non -Standard Form



(a) $AB + C(D + E)$



(b) $AB + CD + CE$

Worksheets

- Worksheet 1

Thank You !!

Feel free to ask any questions