Digital Logic Design

Lecture 6

Overview

- **Canonical and Standard Forms (Minterms, Maxterms, Conversions)**
- **How to write minterms/maxterms from truth table**
- **Writing a function in terms of its minterms/ maxterms**
- **Properties of minterms /maxterms.**
- **Literal cost**
- **Gate input cost**

Canonical Forms

- **Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form.**
- **It is useful to specify Boolean functions in a form that:**
  - **Allows comparison for equality.**
  - **Has a correspondence to the truth tables**
- **Canonical Forms in common usage:**
  - **Sum of Products (SOP)**
  - **Product of Sums (POS)**

Minterms

- **<u>Minterms</u> are AND terms with every variable present in either original or complemented form.**
- **Given that each binary variable may appear normal (e.g., x) or complemented (e.g.,), there are 2^n minterms for n variables.**
- **<u>Example:</u> Two variables (X and Y)produce** $\overline{\mathbf{X}}$
  2 x 2 = 4 combinations:

  $\overline{X}\,\overline{Y}$ **(both complemented)**

  $\overline{X}Y$ **(X complemented, Y normal)**

  $X\overline{Y}$ **(X normal, Y complemented)**

  XY **(both normal)**
- **Thus there are <u>four minterms</u> of two variables.**
- **A literal is a complemented variable if the corresponding bit of the related binary combination is 0 and is an uncomplemented variable if it is 1.**

Maxterms

- **<u>Maxterms</u> are OR terms with every variable in either original or complemented form.**
- **Given that each binary variable may appear normal (e.g., x) or complemented (e.g., $\overline{x}$), there are 2^n maxterms for *n* variables.**
- **<u>Example:</u> Two variables (X and Y) produce 2 x 2 = 4 combinations:**

$$\mathbf{X + Y} \text{ (both normal)}$$
$$\mathbf{X + \overline{Y}} \text{ (x normal, y complemented)}$$
$$\mathbf{\overline{X} + Y} \text{ (x complemented, y normal)}$$
$$\mathbf{\overline{X} + \overline{Y}} \text{ (both complemented)}$$

Maxterms and Minterms

- **Examples: Two variable minterms and maxterms.**

Index	Minterm	Maxterm
0	$\overline{x}\,\overline{y}$	$x + y$
1	$\overline{x}\,y$	$x + \overline{y}$
2	$x\,\overline{y}$	$\overline{x} + y$
3	$x\,y$	$\overline{x} + \overline{y}$

- **The index above is important for describing which variables in the terms are true and which are complemented.**

Standard Order

- **Minterms and maxterms are designated with a subscript**
- **The subscript is a number, corresponding to a binary pattern**
- **The bits in the pattern represent the complemented or normal state of each variable listed in a standard order.**
- **All variables will be present in a minterm or maxterm and will be listed in the <u>same order</u> (usually alphabetically)**
- **Example: For variables a, b, c:**
  - **Maxterms: $(a + b + \bar{c})$, $(a + b + c)$**
  - **Terms: $(b + a + c)$, $a\,\bar{c}\,b$, and $(c + b + a)$ are NOT in standard order.**
  - **Minterms: $a\,\bar{b}\,c$, $a\ b\ c$, $\bar{a}\,\bar{b}\ c$**
  - **Terms: $(a + c)$, $\bar{b}\ c$, and $(\bar{a} + b)$ do not contain all variables**

Purpose of the Index

- **The index for the minterm or maxterm, expressed as a binary number, is used to determine whether the variable is shown in the true form or complemented form.**
- **For Minterms:**
  - **"1" means the variable is "Not Complemented" and**
  - **"0" means the variable is "Complemented".**
- **For Maxterms:**
  - **"0" means the variable is "Not Complemented" and**
  - **"1" means the variable is "Complemented".**

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 Index 0 (base 10) = 000 (base 2) for three variables). All three variables are complemented for minterm 0 ($\overline{X}, \overline{Y}, \overline{Z}$) and no variables are complemented for Maxterm 0 (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 and Maxterm Relationship

- **Review: DeMorgan's Theorem**
  $\overline{x \cdot y} = \overline{x} + \overline{y}$ **and** $\overline{x + y} = \overline{x} \cdot \overline{y}$
- **Two-variable example:**
  $M_2 = \overline{x} + y$ **and** $m_2 = x \cdot \overline{y}$
  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**
- **giving:**
  $$M_i = \overline{m_i} \text{ and } m_i = \overline{M_i}$$
  Thus M_i is the complement of m_i.

Function Tables for Both

- **Minterms of 2 variables**

x y	m_0	m_1	m_2	m_3
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1

Maxterms of 2 variables

x y	M_0	M_1	M_2	M_3
0 0	0	1	1	1
0 1	1	0	1	1
1 0	1	1	0	1
1 1	1	1	1	0

- **Each column in the maxterm function table is the complement of the column in the minterm function table since M_i is the complement of m_i.**

Minterms for Three Variables

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

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

Observations

- **In the function tables:**
  - **Each <u>min</u>term has one and only one 1 present in the 2^n terms (in a row) (a <u>minimum</u> of 1s). All other entries are 0.**
  - **Each <u>max</u>term has one and only one 0 present in the 2^n terms (in a row) All other entries are 1 (a <u>max</u>imum of 1s).**
- **We can implement any function by "ORing" the minterms corresponding to "1" entries in the function table. These are called the minterms of the function.**
- **We can implement any function by "ANDing" the maxterms corresponding to "0" entries in the function table. These are called the maxterms of the function.**
- **This gives us two <u>canonical forms</u>:**
  - **<u>Sum of Products (SOP)</u>**
  - **<u>Product of Sums (POS)</u>**

for stating any Boolean function.

Minterm Function Example

- **Example: Find $F_1 = m_1 + m_4 + m_7$**
- **$F1 = \overline{x}\ \overline{y}\ z + x\ \overline{y}\ \overline{z} + x\ y\ z$**

x y z	index	m_1	+	m_4	+	m_7	$= F_1$
0 0 0	0	0	+	0	+	0	= 0
0 0 1	1	1	+	0	+	0	= 1
0 1 0	2	0	+	0	+	0	= 0
0 1 1	3	0	+	0	+	0	= 0
1 0 0	4	0	+	1	+	0	= 1
1 0 1	5	0	+	0	+	0	= 0
1 1 0	6	0	+	0	+	0	= 0
1 1 1	7	0	+	0	+	1	= 1

Minterm Function Example

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

Maxterm Function Example

- **Example: Implement F1 in maxterms:**

$$F_1 = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

$$F_1 = (x + y + z) \cdot (x + \overline{y} + z) \cdot (x + \overline{y} + \overline{z}) \cdot (\overline{x} + y + \overline{z}) \cdot (\overline{x} + \overline{y} + z)$$

x y z	i	$M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$ = F1
0 0 0	0	0 · 1 · 1 · 1 · 1 = 0
0 0 1	1	1 · 1 · 1 · 1 · 1 = 1
0 1 0	2	1 · 0 · 1 · 1 · 1 = 0
0 1 1	3	1 · 1 · 0 · 1 · 1 = 0
1 0 0	4	1 · 1 · 1 · 1 · 1 = 1
1 0 1	5	1 · 1 · 1 · 0 · 1 = 0
1 1 0	6	1 · 1 · 1 · 1 · 0 = 0
1 1 1	7	1 · 1 · 1 · 1 · 1 = 1

Maxterm Function Example

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

Canonical Sum of Minterms

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

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

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

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

Another SOP Example

- Example: $F = A + \overline{B}\,C$
- 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 SOP:

Shorthand SOP Form

- **From the previous example, we started with:**

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

- **We ended up with:**

$$F = m_1+m_4+m_5+m_6+m_7$$

- **This can be denoted in the formal shorthand:**

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

- **Note that we explicitly show the standard variables in order and drop the "m" designators.**

Canonical Product of Sums

- **Any Boolean Function can be expressed as a <u>Product of Sums (POS)</u>.**
  - **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 $v \cdot \overline{v}$ and then applying the distributive law again.**
- **Example: Convert to product of sums:**

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

Apply the distributive law:

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

Add missing variable z:

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

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

Another POS Example

- **Convert to Product of Sums:**

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

- **Use** $x + yz = (x+y)\cdot(x+z)$ **with** $x = (A\overline{C} + BC)$, $y = \overline{A}$, **and** $z = \overline{B}$ **to get:**

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

- **Then use** $x + \overline{x}y = x + y$ **to get:**

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

and a second time to get:

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

- **Rearrange to standard order,**

$$f = (\overline{A} + B + \overline{C})(A + \overline{B} + C)$$ **to give** $f = M_5 \cdot M_2$

Function Complements

- **The complement of a function expressed as a sum of products is constructed by selecting the minterms missing in the sum-of-products canonical forms.**
- **Alternatively, the complement of a function expressed by a Sum of Products form is simply the Product of Sums with the same indices.**
- **Example: Given** $F(x, y, z) = \Sigma_m(1,3,5,7)$

$$\overline{F}(x, y, z) = \Sigma_m(0,2,4,6)$$

$$\overline{F}(x, y, z) = \Pi_M(1,3,5,7)$$

Conversion Between Forms

- **To convert between sum-of-products and product-of-sums form (or vice-versa) we follow these steps:**
  - **Find the function complement by swapping terms in the list with terms not in the list.**
  - **Change from products to sums, or vice versa.**
- **Example: Given F as before:** $F(x,y,z) = \Sigma_m(1,3,5,7)$
- **Form the Complement:** $\overline{F}(x,y,z) = \Sigma_m(0,2,4,6)$
- **Then use the other form with the same indices – this forms the complement again, giving the other form of the original function:** $F(x,y,z) = \Pi_M(0,2,4,6)$

Standard Forms

- **<u>Standard Sum-of-Products (SOP) form:</u> equations are written as an OR of AND terms**
- **<u>Standard Product-of-Sums (POS) form:</u> equations are written as an AND of OR terms**
- **Examples:**
  - **SOP:** $A\,B\,C + \overline{A}\,\overline{B}\,C + B$
  - **POS:** $(A + B) \cdot (A + \overline{B} + \overline{C}) \cdot C$
- **These “mixed” forms are <u>neither SOP nor POS</u>**
  - $(A\,B + C)\,(A + C)$
  - $A\,B\,\overline{C} + A\,C\,(A + B)$

Standard Sum-of-Products (SOP)

- **A sum of minterms form for *n* variables can be written down directly from a truth table.**
  - **Implementation of this form is a two-level network of gates such that:**
  - **The first level consists of *n*-input AND gates, and**
  - **The second level is a single OR gate (with fewer than 2^n inputs).**
- **This form often can be simplified so that the corresponding circuit is simpler.**

Standard Sum-of-Products (SOP)

- **A Simplification Example:**
- $\mathbf{F(A, B, C) = \Sigma m(1,4,5,6,7)}$
- **Writing the minterm expression:**

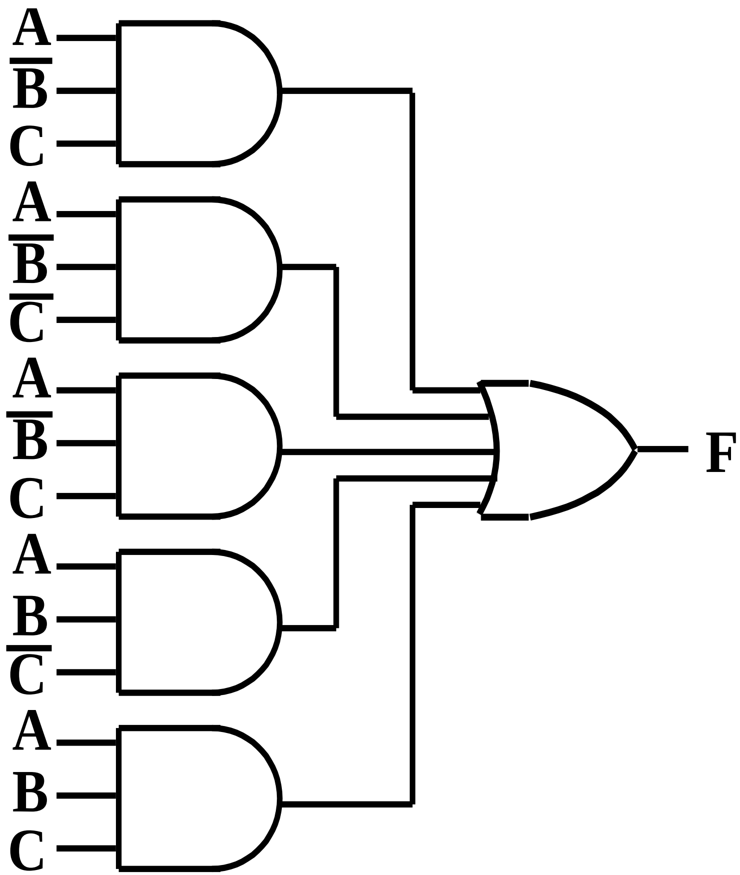
  $\mathbf{F = \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + AB\overline{C} + ABC}$
- **Simplifying:**

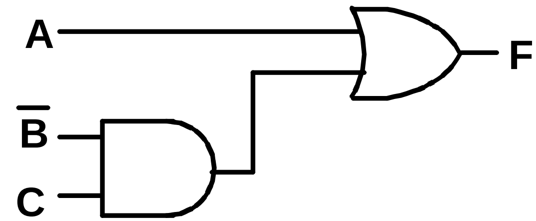
  F =

- **Simplified F contains 3 literals compared to 15 in minterm F**

AND/OR Two-level Implementation of SOP Expression

- The two implementations for F are shown below – it is quite apparent which is simpler!





SOP and POS Observations

- **The previous examples show that:**
  - **Canonical Forms (Sum-of-products, Product-of-Sums), or other standard forms (SSOP, SPOS) differ in complexity**
  - **Boolean algebra can be used to manipulate equations into simpler forms.**
  - **Simpler equations lead to simpler two-level implementations**
- **Questions:**
  - **How can we attain a “simplest” expression?**
  - **Is there only one minimum cost circuit?**
  - **The next part will deal with these issues.**

Cost Criteria

Two cost criteria:

i. Literal cost
   - the # of literal appearances in a Boolean expression

ii. Gate input cost (✓)
   - the # of inputs to the gates in the implementation

Cost Criteria

Literal cost:

- the # of literal appearances in a Boolean expression
- E.g.: $F = AB + C(D + E) \rightarrow 5$ literals

  $F = AB + CD + CE \rightarrow 6$ literals
- Adv.: is very simple to evaluate by counting literal appearances
- Disadv.: does not represent ckt complexity accurately in all cases
  - E.g.:

$$G = ABCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \rightarrow 8 \text{ literals}$$

$$G = (\overline{A} + B)(\overline{B} + C)(\overline{C} + D)(\overline{D} + A) \rightarrow 8 \text{ literals}$$

Cost Criteria

Gate input cost:

- the # of inputs to the gates in the implementation
- For SoP or PoS eqs, the gate input cost can be found by the sum of
  - all literal appearances
  - the # of terms excluding terms that consist only of a single literal
  - the # of distinct complemented single literals (optional)
- E.g.: p.2-51

$$G = ABCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \quad \rightarrow 8 + 2\,(+4) \text{ gate input counts}$$

$$G = (\overline{A} + B)(\overline{B} + C)(\overline{C} + D)(\overline{D} + A) \quad \rightarrow 8 + 4\,(+4)$$

Cost Criteria

- Literal cost has the advantage that it is very simple to evaluate by counting literal appearances.
- literal cost of eight

$$G = ABCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \quad \text{and} \quad G = (\overline{A} + B)(\overline{B} + C)(\overline{C} + D)(\overline{D} + A)$$

Examples

$$F_1 = AB + C(D+E)$$

$$Cost(F_1) = 5 + 3 + 0$$
$$= 8$$

$$F_2 = AB + CD + CE$$

$$Cost(F_2) = 6 + 3$$
$$= 9$$

$$F_3 = ABCD + \bar{A}\bar{B}\bar{C}\bar{D}$$

$$Cost(F_3) = 8 + 2 + 4$$
$$= 14$$

$$F_4 = (\bar{A} + B)(\bar{B} + C)(\bar{C} + D)(\bar{D} + A)$$

$$Cost(F_4)$$
$$= 8 + 4 + 4$$
$$= 16$$