**Digital Logic Design Lecture 6**

1

**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**

Chapter 2 - Part 1 2

**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)**

Chapter 2 - Part 1 3

**Minterms**

▪ **Minterms 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.**

**~~x~~**

▪ **Example: Two variables (X and Y)produce 2 x 2 = 4 combinations:**

**XY (both complemented)**

**XY (X complemented, Y normal)**

**XY (X normal, Y complemented)**

**XY (both normal)**

▪ **Thus there are four minterms 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.**

Chapter 2 - Part 1 4

**Maxterms**

▪ **Maxterms 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., ~~x)~~, there are 2*n* maxterms for *n* variables.**

▪ **Example: Two variables (X and Y) produce 2 x 2 = 4 combinations:**

**X** +**Y X** +**Y X** +**Y X** +**Y**

**(both normal)**

**(x normal, y complemented) (x complemented, y normal) (both complemented)**

Chapter 2 - Part 1 5

**Maxterms and Minterms**

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

**Index Minterm Maxterm**

| **~~x y~~** |
| --- |
| **~~x~~ y** |
| **x ~~y~~** |

**0 x + y**

**1 x + ~~y~~**

**2 ~~x~~ + y**

**3 x y ~~x~~ + ~~y~~**

▪ **The ~~index above is important for describi~~ng which variables in the terms are true and which are complemented.**

Chapter 2 - Part 1 6

**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 same order (usually alphabetically)** ▪ **Example: For variables a, b, c:**

• **Maxterms: (a + b + ~~c)~~, (a + b + c)**

• **Terms: (b + a + c), a ~~c~~ b, and (c + b + a) are NOT in standard order.**

• **Minterms: a ~~b~~ c, a b c, a b c**

• **Terms: (a + c), bc, and (a+ b) do not contain all variables**

Chapter 2 - Part 1 7

**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”.**

Chapter 2 - Part 1 8

**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 ( ) and no variables are**

**X,Y,Z**

**complemented for Maxterm 0 (X,Y,Z).**

• **Minterm 0, called m0is .**

**XYZ**

• **Maxterm 0, called M0is (X + Y + Z).**

• **Minterm 6 ?**

• **Maxterm 6 ?**

Chapter 2 - Part 1 9

**Index Examples – Four Variables**

**Index Binary Minterm Maxterm**

**i Pattern mi Mi**

**0 0000**

**abcda** + **b** + **c** + **d**

**1 0001 3 0011**

**abcd ?**

**?**

**a** + **b** + **c** + **d**

**5 0101**

**abcd a** + **b** + **c** + **d**

**7 0111**

**?**

**a** + **b** + **c** + **d**

**10 1010 13 1101 15 1111**

**abcd a** + **b** + **c** + **d ~~c~~ ?**

**ab d**

**abcda** + **b** + **c** + **d**

Chapter 2 - Part 1 10

**Minterm and Maxterm Relationship**

▪ **Review: DeMorgan's Theorem**

**~~x ·y~~** = **~~x~~** + **~~y x~~** ~~+~~ **~~y~~** = **~~x~~** ⋅ **~~y~~**

**and**

▪ **Two-variable example:**

**M2**= **~~x~~** + **y m2**= **x·~~y~~**

**and**

**Thus M2 is the complement of m2 and vice-versa.** ▪ **Since DeMorgan's Theorem holds for *n* variables, the above holds for terms of *n* variables**

▪ **giving:**

**Mi**= **mi mi**= **Miand**

**Thus Miis the complement of mi.**

Chapter 2 - Part 1 11

**Function Tables for Both**

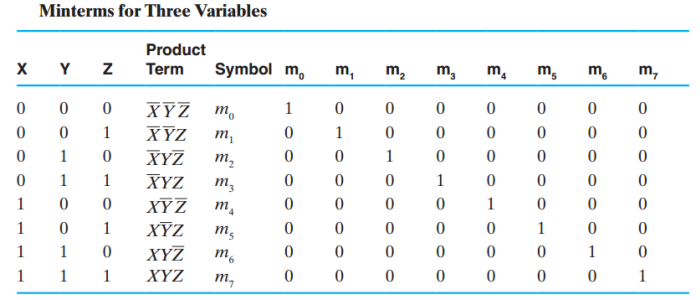
▪ **Minterms of Maxterms of 2 variables 2 variables**

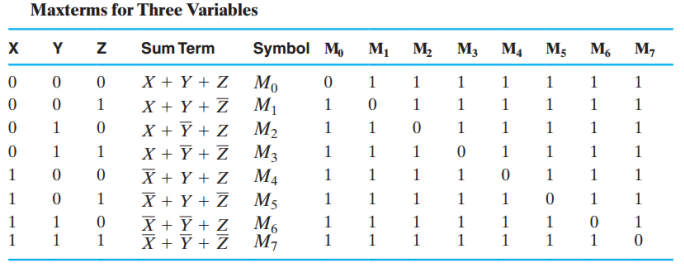
| **x y** | **m0** | **m1** | **m2** | **m3** |
| --- | --- | --- | --- | --- |
| **0 0** | **1** | **0** | **0** | **0** |
| **0 1** | **0** | **1** | **0** | **0** |
| **1 0** | **0** | **0** | **1** | **0** |
| **1 1** | **0** | **0** | **0** | **1** |

| **x y** | **M0** | **M1** | **M2** | **M3** |
| --- | --- | --- | --- | --- |
| **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 Miis the complement of mi.**

Chapter 2 - Part 1 12

Chapter 2 - Part 1 13

Chapter 2 - Part 1 14

**Observations**

▪ **In the function tables:**

• **Each minterm has one and only one 1 present in the 2*n* terms (in a row) (a minimum of 1s). All other entries are 0.** • **Each maxterm has one and only one 0 present in the 2*n* terms (in a row) All other entries are 1 (a maximum 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 canonical forms:**

• **Sum of Products (SOP)**

• **Product of Sums (POS)**

**for stating any Boolean function.**

Chapter 2 - Part 1 15

**Minterm Function Example**

▪ **Example: Find F1 = m1 + m4 + m7** ▪ **F1 = ~~x y~~ z + x ~~y z~~ + x y z**

**x y z index m1 + m4 + m7 = F1**

**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**

Chapter 2 - Part 1 16

**Minterm Function Example**

▪ **F(A, B, C, D, E) = m2 + m9 + m17 + m23** ▪ **F(A, B, C, D, E) =**

Chapter 2 - Part 1 17

**Maxterm Function Example**

▪ **Example: Implement F1 in maxterms: F1 = M0 · M2 · M3 · M5 · M6 F1** = **(x** + **y** + **z) ·(x** + **~~y~~** + **z)·(x** + **~~y~~** + **~~z~~)**

**·(~~x~~** + **y** + **~~z~~)·(~~x~~** + **~~y~~** + **z)**

**x y z i M0**⋅ **M2**⋅ **M3**⋅ **M5**⋅ **M6 = F1**

⋅

⋅ **1**

⋅

**0 0 0 0 0 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** ⋅

Chapter 2 - Part 1 18

**Maxterm Function Example**

**M3 M8 M11 M14 F(A,B,C,D)** = ⋅ ⋅ ⋅▪

▪ **F(A, B,C,D) =**

Chapter 2 - Part 1 19

**Canonical Sum of Minterms**

▪ **Any Boolean function can be expressed as a Sum of Products.**

• **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 v** + **~~v~~**

**missing a variable v with a term ( ).**

**f** = **x** + **~~x y~~**

▪ **Example: Implement as a sum of minterms.**

**First expand terms:**

**f** = **x(y** + **~~y~~)** + **~~x y~~**

**Then distribute terms:**

**f** = **xy** + **x~~y~~** + **~~x y~~**

**Express as sum of minterms: f = m3 + m2 + m0**

Chapter 2 - Part 1 20

**Another SOP Example**

▪ **Example:**

**F** = **A** + **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:**

Chapter 2 - Part 1 21

**Shorthand SOP Form**

▪ **From the previous example, we started with: F** = **A** + **B C**

▪ **We ended up with:**

**F = m1+m4+m5+m6+m7**

▪ **This can be denoted in the formal shorthand: F(A,B,C)** = Σ**m(1,4,5,6,7)**

▪ **Note that we explicitly show the standard variables in order and drop the “m” designators.**

Chapter 2 - Part 1 22

**Canonical Product of Sums**

▪ **Any Boolean Function can be expressed as a Product of Sums (POS).**

• **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**

**v** ⋅ **~~v~~**

**law , “ORing” terms missing variable v with a term equal to and then applying the distributive law again.**

▪ **Example: Convert to product of sums:**

**f(x, y, z)** = **x** +**~~x y~~**

**Apply the distributive law:**

**x** +**~~x y~~** =**(x** +**~~x~~)(x** +**~~y~~)** =**1** ⋅**(x** +**~~y~~)** =**x** +**~~y~~ Add missing variable z:**

**x** +**~~y~~** +**z**⋅ **~~z~~** = **(x** +**~~y~~** +**z)**(**x**+**~~y~~** +**~~z~~** )

**Express as POS: f = M2 · M3**

Chapter 2 - Part 1 23

**Another POS Example**

▪ **Convert to Product of Sums:**

**f(A,B,C)** =**A C** +**BC** + **A B**

**x** =**(AC** +**BC), y** =**A**

▪ **Use x + y z = (x+y)·(x+z) with , and to get:**

**z** =**B**

**f** =**(AC** +**BC** + **A)(AC** +**BC** +**B)**

**x** +**~~x~~ y** =**x** + **y**

▪ **Then use to get:**

**f** = **(C** +**BC** + **A)(AC** +**C** +**B)**

**and a second time to get:**

**f** =**(C** +**B** + **A)(A** +**C** +**B)**

▪ **Rearrange to standard order,**

**f** =**(A** +**B** +**C)(A** +**B** +**C)**

**to give f = M5· M2**

Chapter 2 - Part 1 24

**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)** = Σ**m (1,3,5,7) F~~(~~x, y, z)** = Σ**m(0,2,4,6)**

**F(x, y, z)** = Π**M(1,3,5,7)**

Chapter 2 - Part 1 25

**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)** = Σ**m(1,3,5,7)**

**F(x, y, z)** = Σ**m(0,2,4,6)**

▪ **Form the Complement:**

▪ **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)** = Π**M(0,2,4,6)**

Chapter 2 - Part 1 26

**Standard Forms**

▪ **Standard Sum-of-Products (SOP) form: equations are written as an OR of AND terms** ▪ **Standard Product-of-Sums (POS) form: equations are written as an AND of OR terms** ▪ **Examples:**

• **SOP:** • **POS:**

**A B C** + **~~A B~~ C** + **B (A**+**B)·(A**+**B** +**C)·C**

▪ **These “mixed” forms are neither SOP nor POS**

**(A B** + **C) (A** + **C)**

•

**ABC**+ **AC(A**+ **B)**•

Chapter 2 - Part 1 27

**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.**

Chapter 2 - Part 1 28

**Standard Sum-of-Products (SOP)** ▪ **A Simplification Example:**

**F(A,B,C)** = Σ**m(1,4,5,6,7)**▪

▪ **Writing the minterm expression:**

**F = A B C + A B C + A B C + ABC + ABC** ▪ **Simplifying:**

**F =**

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

Chapter 2 - Part 1 29

**AND/OR Two-level Implementation of SOP Expression**

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

**B**

**A**

**C F**

**A B C A B C A B C A B C**

**F**

**B C**

Chapter 2 - Part 1 30

**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**

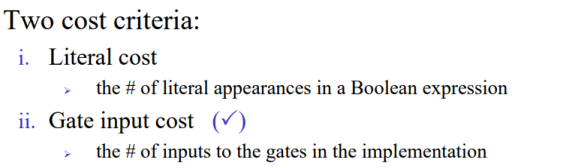
▪ **Questions:**

• **How can we attain a “simplest” expression?** • **Is there only one minimum cost circuit?**

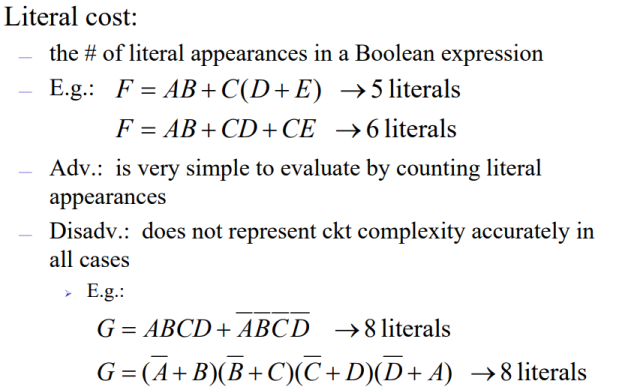
• **The next part will deal with these issues.**

Chapter 2 - Part 1 31

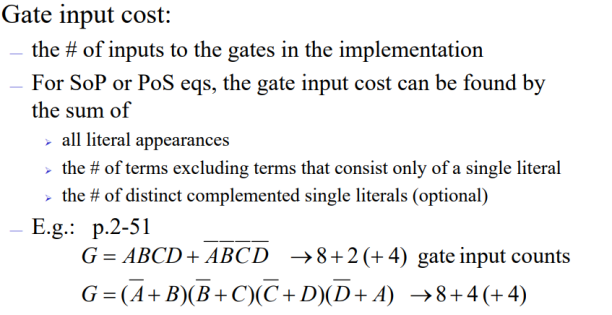
Cost Criteria

Chapter 2 - Part 1 32

Cost Criteria

Chapter 2 - Part 1 33

Cost Criteria

Chapter 2 - Part 1 34

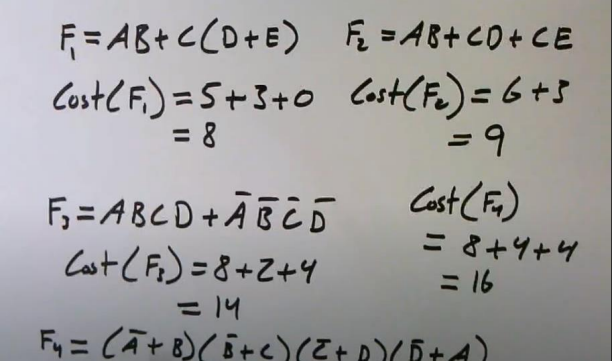
Cost Criteria

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

▪ literal cost of eight

Chapter 2 - Part 1 35

Examples

Chapter 2 - Part 1 36