Digital Design

Ms. Umarah Qaseem

1

3

Chapter 2

Boolean Algebra and Logic Gates,

Equivalent Circuits, Simplification, Canonical

4

Boolean Algebra

Boolean Algebra developed by George Boole (1815-1864)

11

11

Boolean Algebra

- An algebra that deals with binary variables and logic operations.
  - Developed by George Boole in 1854
  - Shannon introduced two-valued Boolean algebra in 1938.
  - Huntington formulated the formal definition of Boolean algebra through the postulates in 1904.

12

Postulates & Theorems

Postulate

- ❑ A basic assumption that is accepted without proof
- ❑ Accepted as true in order to provide a basis for logical reasoning

Theorem

- ❑ A statement which can be proved to be true using logical (deductive) reasoning
- ❑ A proposition deducible from basic postulates

13

13

Boolean Algebra

“plus” / ”OR”

“times” / ”AND”

- A (two-valued) **Boolean Algebra** is a 3-tuple {*B* , + , · }, where
  - *B* is a set of at least 2 elements
  - (+) and (·) are binary operations

There are 6 fundamental laws/postulates, or axioms that these operators satisfy

14

14

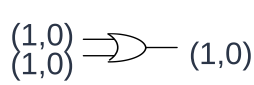
Postulates of Boolean Algebra

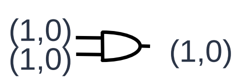
- **Closure** ➔ *x + y is in B and x . y is in B* (Obvious as result either 0 or 1)
- **Identity** ➔ *(1. x=x) and (0+x=x)*
- **Commutative** ➔ *x + y = y + x and x . y = y . x*
- **Distributive** ➔ *x + (y. z) = (x + y) (x + z) and x. (y + z) = (x . y) + (x . z)*
- **Complement** ➔ *x + x' = 1 and x.x'=0 (for every x there is an x')*
- **Postulate 6:** *There exist at least two elements x, y such that x!=y.*

15

Postulate 1

- A1. Closure: For every a, b belongs to B
- a + b is in B; Closure with respect to '+'
- a · b is in B; Closure with respect to '.'





16

16

Postulate 2

- A2. Existence of identity elements: The set B has two distinct identity elements, denoted as 0 and 1, such that for every element a B
- a + 0 = 0 + a = a
- a · 1 = 1 · a = a

additive identity element

multiplicative identity element

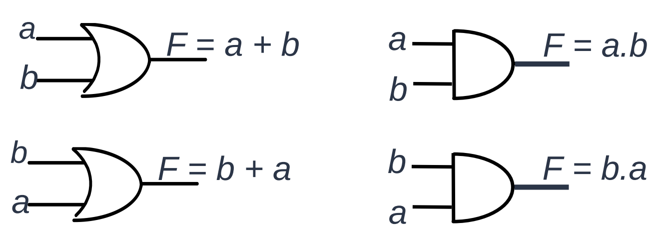
OR operation AND operation

17

17

Postulate 3

- A3. Commutative laws: For every a, b B
- a + b = b + a
- a · b = b · a



18

18

Postulate 4

- A4. Distributive laws: For every a, b, c $\in$ B
- a + (b · c) = (a+ b) ·(a + c) NOT Similar to Linear Algebra
- a · (b + c) = (a · b) + (a · c) Similar to Linear Algebra

19

19

Postulates 5

- A5. Existence of a complement: For every element a $\in$ B there exists an element a' such that
- a + a' = 1
- a · a' = 0
  - X+X'=1 ➔ 0 +0'=0+1=1 and 1+1'=1+0=1
  - X.X'=0 ➔ 0.0' = 0.1 =1 and 1.1' =1+0=1

20

20

Postulates 6

- **A6**. There exist at least two elements $a,b \in B$ such that
- $a \neq b$

The Duality Principle

- Each postulate of Boolean algebra contains a pair of expressions or equations such that one is transformed into the other and vice-versa by **interchanging the operators, + ↔ ·, and identity elements, 0 ↔ 1.**
- The two expressions are called the **duals** of each other.

Examples of Duals

Postulate	Duals	
	Expression 1	Expression 2
1	a, b, a + b ε B	a, b, a · b ε B
2	a + 0 = a	a · 1 = a
3	a + b = b + a	a · b = b · a
4	a + (b + c) = (a + b) + c	a · (b · c) = (a · b) · c
5	a + (b · c)=(a + b) · (a + c)	a · (b + c)=(a · b)+(a · c)
6	$a + \bar{a} = 1$	$a \bullet \bar{a} = 0$

23

Properties of Boolean Algebra

- Properties stated as theorems
- Provable from the postulates (axioms) of Boolean algebra

25

25

Basic Theorems for Boolean Algebra

Theorem # 1: (Idempotency Law)

- $(x + x) = x$
- $(x \,.\, x) = x$

34

Proof of theorem 1

THEOREM 1(a): $x + x = x$.

Statement	Justification
$x + x = (x + x) \cdot 1$	postulate 2(b)
$= (x + x)(x + x')$	5(a)
$= x + xx'$	4(b)
$= x + 0$	5(b)
$= x$	2(a)

THEOREM 1(b): $x \cdot x = x$.

Statement	Justification
$x \cdot x = xx + 0$	postulate 2(a)
$= xx + xx'$	5(b)
$= x(x + x')$	4(a)
$= x \cdot 1$	5(a)
$= x$	2(b)

35

Basic Theorems for Boolean Algebra

Theorem # 2:

- x + 1 = 1

  (any variable + 1 = 1)

- x . 0 = 0

  (any variable AND with 0 = 0)

36

Proof of theorem 2

THEOREM 2(a): $x + 1 = 1.$

Statement	Justification
$x + 1 = 1 \cdot (x + 1)$	postulate 2(b)
$= (x + x')(x + 1)$	5(a)
$= x + x' \cdot 1$	4(b)
$= x + x'$	2(b)
$= 1$	5(a)

THEOREM 2(b): $x \cdot 0 = 0$ by duality.

37

Basic Theorems for Boolean Algebra

Theorem # 3: (Involution or Double Negation)

- $(x')' = x$

38

Basic Theorems for Boolean Algebra

Theorem # 4: (Associative Law)

If x , y & z are three variables, then

- $x + (y + z) = (x + y) + z$
- $x . (y . z) = (x . y) . z$

39

Basic Theorems for Boolean Algebra

Theorem # 5: (Demorgan's Law)

If x & y are two variables, then

- $(x + y)' = x' \cdot y'$
- $(x \cdot y)' = x' + y'$

Demorgan's Laws are one of the most important set of Boolean Algebraic Laws.

40

Basic Theorems for Boolean Algebra

Theorem # 6: (Absorption Law)

- If x & y are two variables then,
  $x + xy = x$ or
  $x(x + y) = x$ (through duality)

41

Proof Absorption law

THEOREM 6(a): $x + xy = x$.

Statement	Justification
$x + xy = x \cdot 1 + xy$	postulate 2(b)
$= x(1 + y)$	4(a)
$= x(y + 1)$	3(a)
$= x \cdot 1$	2(a)
$= x$	2(b)

THEOREM 6(b): $x(x + y) = x$ by duality.

42

Proof of associative and de Morgans law

The algebraic proofs of the associative law and DeMorgan's theorem are long and will not be shown here. However, their validity is easily shown with truth tables. For example, the truth table for the first DeMorgan's theorem, $(x + y)' = x'y'$, is as follows:

x	y	$x + y$	$(x + y)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

x'	y'	$x'y'$
1	1	1
1	0	0
0	1	0
0	0	0

45

Rule Number Boolean Expression	
1	$A + 0 = A$
2	$A + 1 = 1$
3	$A \cdot 0 = 0$
4	$A \cdot 1 = A$
5	$A + A = A$
6	$A + \overline{A} = 1$
7	$A \cdot A = A$
8	$A \cdot \overline{A} = 0$
9	$\overline{\overline{A}} = A$
10	$A + AB = A$
11	$A + \overline{A}B = A + B$
12	$(A + B)(A + C) = A + BC$

46

46

Identity	**a)** $x + 0 = x$	**b)** $x .1 = x$
Commutative	**a)** $x + y = y + x$	**b)** $x .y = y.x$
Distributive	**a)** $x.(y + z) = (x .y) + (x.z)$	**b)** $x + (y.z) = (x+y).(x+z)$
Complement	**a)** $x + x' = 1$	**b)** $x.x' = 0$

47

47

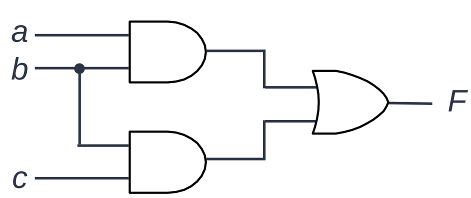
Implementation of Boolean Functions

48

Implementation of Boolean Functions

Draw the logic diagram for the following function:

$F = (a.b)+(b.c)$



49

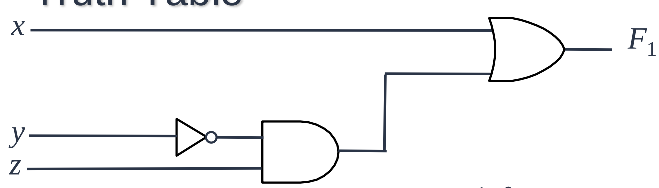
49

Implementation of Boolean Functions

- *Boolean Function expresses logical relationship between binary variables*
- Boolean Function can be represented by a Truth Table

X	Y	Z	F_1
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

$$F_1 = x + y'z$$



50

Operator Precedence

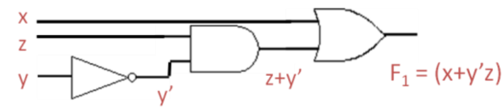
Operator Precedence for evaluating Boolean expressions is

- Parentheses
- NOT
- AND
- OR

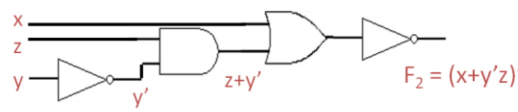
51

Implementation of Boolean Functions

x	y	z	F_1	F_2
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0



$F_1 = (x+y'z)$



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

52

Boolean Functions

- Boolean functions, truth tables or logic diagrams can be generated from each other
- Designers try to reduce complexity and number of gates to reduce circuit cost
- Two different boolean functions may represent the same logic and truth table
- A single variable in a boolean function is called literal
- Group of literals which are forming an input to a gate together, are called a term

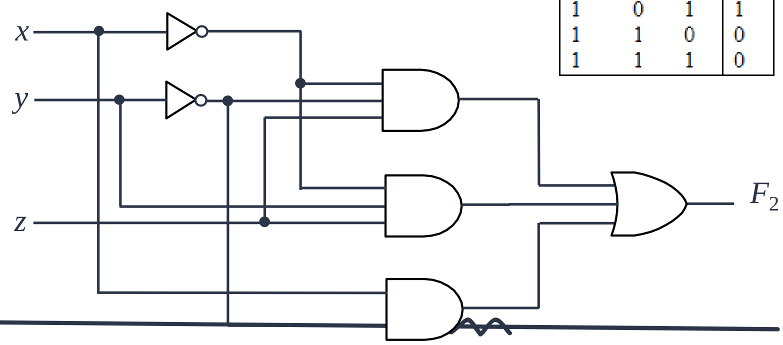
53

Implementation of Boolean Functions

- Example 2:

$$F_2 = x'y'z + x'yz + xy'$$

X	Y	Z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



54

Simplification

55

Simplification of Boolean Functions

X	Y	Z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

- Try another implementation using a simplified F_2:

$$F_2 = x'y'z + x'yz + xy'$$
$$= x'z(y'+y) + xy' \quad \text{Distributive Pos}$$
$$= x'z(1) + xy' \quad \text{Complement Pos}$$
$$= x'z + xy'$$

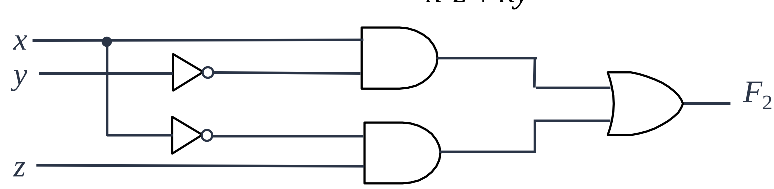
Identity	**a)** ***x + 0 = x*** **b)** ***x .1 = x***
Commutative	**a)** ***x + y = y + x*** **b)** ***x .y = y.x***
Distributive	**a)** ***x.(y + z) = (x .y) + (x.z)*** **b)** ***x + (y.z) = (x+y).(x+z)***
Complement	**a)** ***x + x' = 1*** **b)** ***x.x' = 0***

56

Implementation of Boolean Functions

X	Y	Z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

$$F_2 = x'y'z + x'yz + xy'$$
$$= x'z(y'+y) + xy' \quad \text{Distributive Pos}$$
$$= x'z(1) + xy' \quad \text{Complement Pos}$$
$$= x'z + xy'$$

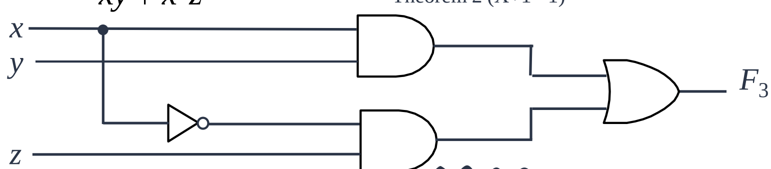


This implementation has fewer gates and fewer inputs to the gates (or wires) than the previous one.

57

Simplifying Boolean Functions

$$
\begin{aligned}
F_3 &= xy + x'z + yz \\
&= xy + x'z + yz(x + x') && \text{Identity .1 is same \& Complement Pos (x.x')=1} \\
&= xy + x'z + xyz + x'yz && \text{Distributed Pos} \\
&= xy + xyz + x'z + x'yz && \text{Rearrange order} \\
&= xy(1+z) + x'z(1+y) && \text{Distributed Pos} \\
&= xy + x'z && \text{Theorem 2 (X+1 =1)}
\end{aligned}
$$



58

Equivalent Circuits,

59

Boolean Function: Another Example

Boolean function

$F_2 = x' y' z + x' y z + x y'$

Implementation: two not gates, three 3-input AND gates, one 3-input OR gate

Simplifies to F_2
$$\begin{aligned} &= x' y' z + x' y z + x y' \\ &= x' z (y' + y) + x y' \\ &= x' z (1) + x y' \\ &= x' z + x y' \end{aligned}$$

Implementation: two not gates, two 2-input AND gates, one 2-input OR gate

60

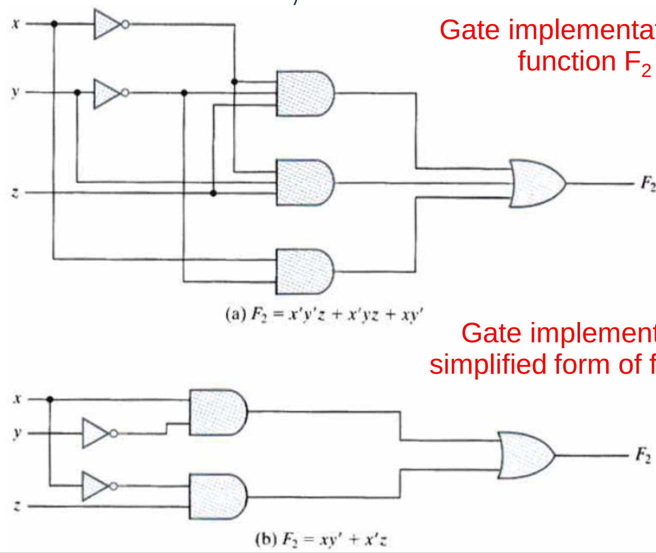
Truth table for Function F_2

$F_2 = x' y' z + x' y z + x y'$
$F_2 = x' z + x y'$

x	y	z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

61

Gate Implementation of F_2



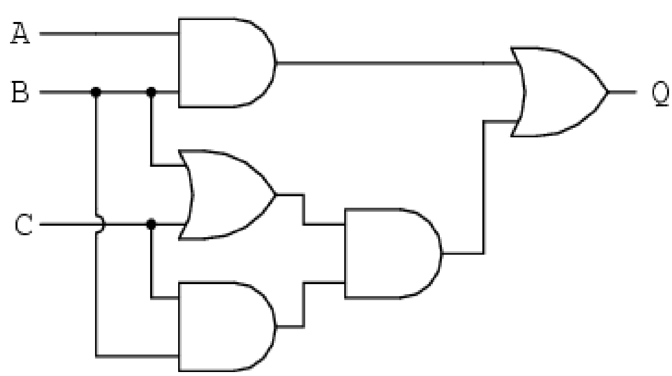
Gate implementation of function F_2

Gate implementation of simplified form of function F_2

62

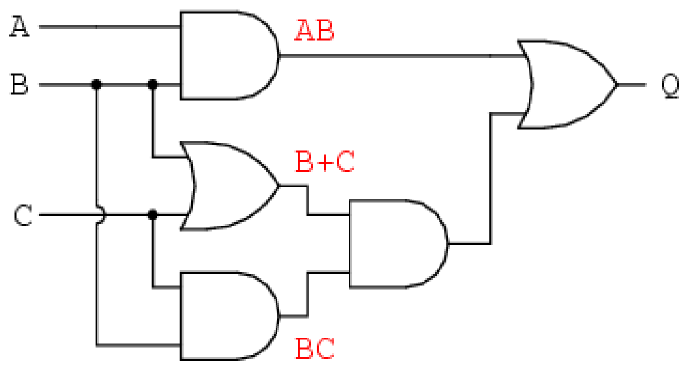
Equivalent Circuits & its Simplification

Example 2: Write down the equation for Q.



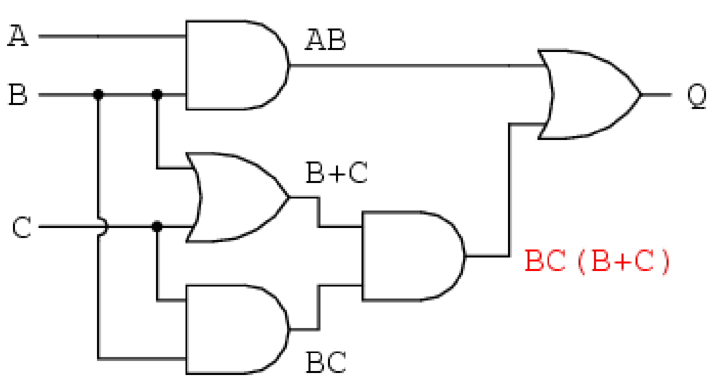
63

Equivalent Circuits & its Simplification



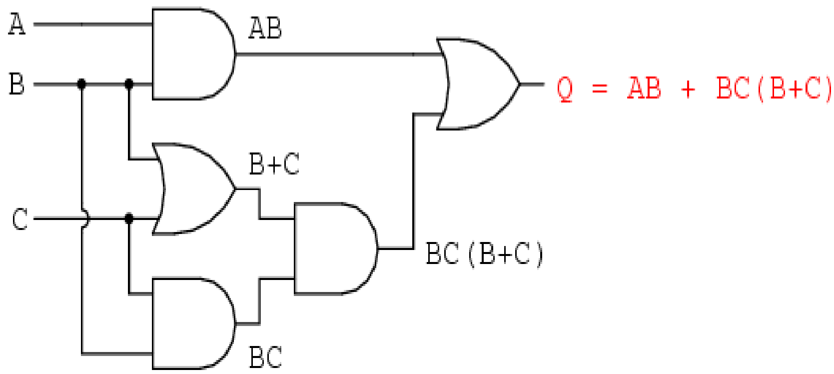
64

Equivalent Circuits & its Simplification



65

Equivalent Circuits & its Simplification Example 1



66

- Now simplifying the expression

AB + BC(B + C)

↓ Distributing terms

AB + BBC + BCC

↓ Applying identity **AA = A** to 2nd and 3rd terms

AB + BC + BC

↓ Applying identity **A + A = A** to 2nd and 3rd terms

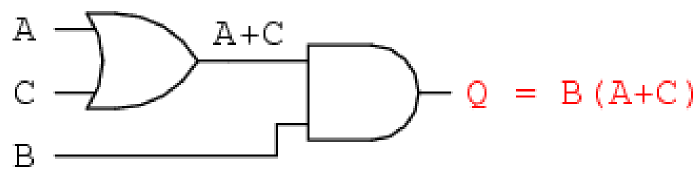
AB + BC

↓ Factoring **B** out of terms

B(A + C)

67

- Final Simplified Circuit....



68

68

LOGICAL EQUIVALENCE

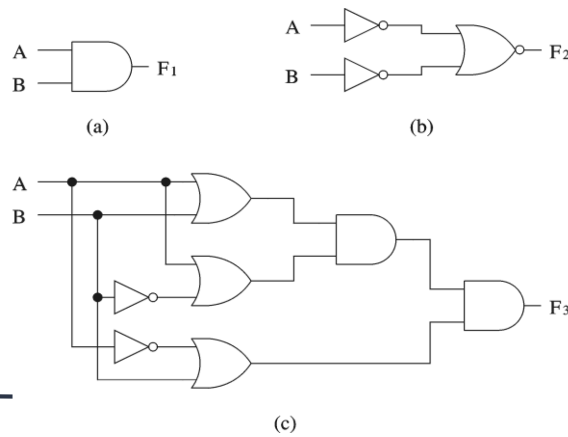
- One more example…

69

69

Logical Equivalence

- All three circuits implement F = A B function



70

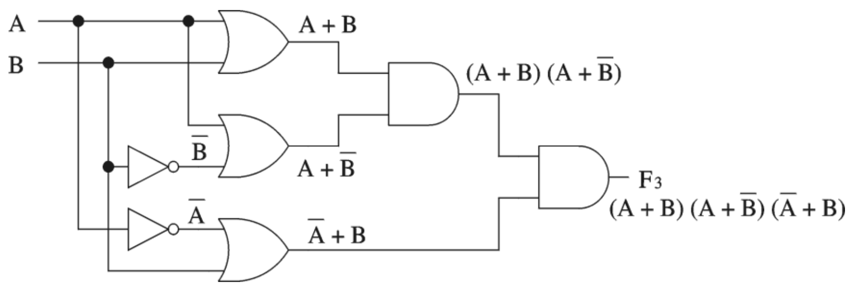
Logical Equivalence

- Proving logical equivalence of two circuits
  - Derive the logical expression for the output of each circuit
  - Show that these two expressions are equivalent
- Two ways:
  - You can use the truth table method
  - You can also use algebraic manipulation

71

Logical Equivalence

- Derivation of logical expression from a circuit
  - Trace from the input to output
  - Write down intermediate logical expressions along the path



72

Logical Equivalence

- Proving logical equivalence: Truth table method

A	B	F1 = A B	$F3 = (A + \overline{B})\ (\overline{A} + B)\ (A + B)$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

73

Rule Number Boolean Expression	
1	$A + 0 = A$
2	$A + 1 = 1$
3	$A \cdot 0 = 0$
4	$A \cdot 1 = A$
5	$A + A = A$
6	$A + \overline{A} = 1$
7	$A \cdot A = A$
8	$A \cdot \overline{A} = 0$
9	$\overline{\overline{A}} = A$
10	$A + AB = A$
11	$A + \overline{A}B = A + B$
12	$(A + B)(A + C) = A + BC$

74

Identity	**a)** $x + 0 = x$	**b)** $x . 1 = x$
Commutative	**a)** $x + y = y + x$	**b)** $x . y = y . x$
Distributive	**a)** $x.(y + z) = (x . y) + (x.z)$	**b)** $x + (y.z) = (x+y).(x+z)$
Complement	**a)** $x + x' = 1$	**b)** $x.x' = 0$

75

Practice the following three examples at home

76

76

Simplification by using Basic Rules

$x + x'y = x + y$

Proof:

$x + x'y = x(1) + x'y$ postulate 2b
$x + x'y = x(1 + y) + x'y$ theorem 2a
$= x.1 + xy + x'y$ (postulate 4b)
$= x + (x + x')y$ (postulate 2b, 4a)
$= x + (1)y$ (postulate 5a)
$= x + y$ (postulate 2b)

Duality : $x.(x'+y)=x.y$

77

77

Simplification by using Basic Rules

$$x' + xy = x' + y$$

Proof:

$x' + xy = x'(1) + xy$ postulate 2b
$= x'(1 + y) + xy$ theorem 2a
$= x'.1 + x'y + xy$ (postulate 4b)
$= x' + (x' + x)y$ (postulate 2b, 4a)
$= x' + (1)y$ (postulate 5a)
$= x' + y$ (postulate 2b)

Duality : $x'.(x+y) = x'.y$

78

78

- Example 3

A + B(A + C) + AC

↓ Distributing terms

A + AB + BC + AC

↓ Applying rule **A + AB = A** to 1st and 2nd terms

A + BC + AC

↓ Applying rule **A + AB = A** to 1st and 3rd terms

A + BC

79

79

Compliment of Functions

80

Complement of a Function

- DeMorgan's Theorem

$(x+y)' = x.'y'$
$(x.y)'=x'+y'$

$(A+B+C+D+...+F)'= A'B'C'D'.... F'$
$(ABCD...F)' = A'+B'+C'+D'+... + F'$

The complement of a function can be obtained by interchanging AND and OR operators and complementing each literal

81

Complement of a Function

- Example: Find the complement of:

$$
\begin{aligned}
F &= x'yz'+x'y'z \\
F' &= (x'yz'+x'y'z)' \\
&= (x'yz')'.(x'y'z)' \\
&=(x+y'+z).(x+y+z')
\end{aligned}
$$

82

Complement of a Function

- Example:

Find the Complement of F = (AB' + C) D' + E

$$
\begin{aligned}
F &= (AB'+C)D'+E \\
F' &= [(AB'+C)D'+E]' \\
&= [(AB'+C)D']'E' \\
&= [(AB'+C)'+D'']E' \\
&= [(AB')'C'+D]E' \\
&= (A'+B)C'E'+DE'
\end{aligned}
$$

83

Complement of a Function

- Difference between compliment and dual?
- 0→1 and + → . And vice versa
- BUT
- We don't compliment the literal in dual

84

Complement of a Function

- Example: Another easy method
- Find the complement of F by taking its dual and complementing each literal

$$F = x'yz' + x'y'z$$

Dual of F

$$= (x'+y+z').(x'+y'+z)$$

Complement each literal

$$=(x+y'+z).(x+y+z') = F'$$

85

Canonical and Standard forms

86

Definitions

Product terms – AND → $\bar{A}BZ$

Sum terms – OR → $X + \bar{A}$

87

Definitions

Sum of Products → $\bar{A}Z+BZ$

Product of Sums → $(X+\bar{A})(X+B)$

88

89

Canonical and Standard Forms

- Canonical form: consists of terms comprising of all variables
- Sum of Product (SOP)
  - Sum of Minterms e.g. xy + x' y'
- Product of Sum (POS)
  - Product of Maxterms e.g. (x+y).(x'+y')
- Standard form: a reduced (short) form of canonical expression.
  - Standard Product

  (XZ) + (Y' Z) + (X' YZ)
  - Standard Sums

  (X + Z) (Y' + Z) (X' + Y + Z)

90

Canonical and Standard Forms

- Canonical form : all variables
- Standard form : reduced form
- Non-Standard form : neither in SOP nor in POS e.g x(y+z)

91

Minterms and Maxterms

- *n* binary variables can be combined to form 2^n terms (AND terms), called *Minterms* or standard products.
- In a similar fashion, *n* binary variables can be combined to form 2^n terms (OR terms), called *Maxterms* or standard sums.
- **Boolean Functions expressed in Sum of MinTerms or Product of MaxTerms are said to be in Canonical Form**

92

Minterms and Maxterms

- Each Minterm is obtained from an AND term.
- Variable primed if corresponding bit 0, vice versa
- Each Maxterm is obtained from an OR term.
- Variable primed if corresponding bit 1, vice versa

x	y	z	minterm	maxterm
0	0	0	x'.y'.z'	x+y+z
0	0	1	x'.y'.z	x+y+z'
0	1	0	x'.y.z'	x+y'+z
0	1	1	x'.y.z	x+y'+z'
1	0	0	x.y'.z'	x'+y+z
1	0	1	x.y'.z	x'+y+z'
1	1	0	x.y.z'	x'+y'+z
1	1	1	x.y.z	x'+y'+z'

93

Minterms and Maxterms

x	y	z	minterm	designation	maxterm	designation
0	0	0	x'.y'.z'	m_0	x+y+z	M_0
0	0	1	x'.y'.z	m_1	x+y+z'	M_1
0	1	0	x'.y.z'	m_2	x+y'+z	M_2
0	1	1	x'.y.z	m_3	x+y'+z'	M_3
1	0	0	x.y'.z'	m_4	x'+y+z	M_4
1	0	1	x.y'.z	m_5	x'+y+z'	M_5
1	1	0	x.y.z'	m_6	x'+y'+z	M_6
1	1	1	x.y.z	m_7	x'+y'+z'	M_7

94

Minterm Is True when

Minterm	Condition	Shorthand
x'y'z'	x=0, y=0, z=0	m_0
x'y'z	x=0, y=0, z=1	m_1
x'yz'	x=0, y=1, z=0	m_2
x'yz	x=0, y=1, z=1	m_3
xy'z'	x=1, y=0, z=0	m_4
xy'z	x=1, y=0, z=1	m_5
xyz'	x=1, y=1, z=0	m_6
xyz	x=1, y=1, z=1	m_7

95

Sum of Minterms Form

- Every function can be written as a sum of minterms, which is a special kind of sum of products form
- The sum of minterms form for any function is **unique**
- **If you have a truth table for a function, you can write a sum of minterms expression just by picking out the rows of the table where the function output is 1.**

96

Example

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

97

97

Example

f = x'y'z' + x'y'z + x'yz' + x'yz + xyz'
= $m_0 + m_1 + m_2 + m_3 + m_6$
f(x,y,z) = ∑ (0,1,2,3,6)

f' = xy'z' + xy'z + xyz
= $m_4 + m_5 + m_7$
= ∑ (4,5,7)

f' contains all the minterms not in f

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

98

Maxterm Is false when

Maxterm	Condition	Shorthand
x + y + z	x=0, y=0, z=0	M_0
x + y + z'	x=0, y=0, z=1	M_1
x + y' + z	x=0, y=1, z=0	M_2
x + y' + z'	x=0, y=1, z=1	M_3
x' + y + z	x=1, y=0, z=0	M_4
x' + y + z'	x=1, y=0, z=1	M_5
x' + y' + z	x=1, y=1, z=0	M_6
x' + y' + z'	x=1, y=1, z=1	M_7

99

Product of Maxterms

- Every function can be written as a unique product of maxterms
- **If you have a truth table for a function, you can write a product of maxterms expression by picking out the rows of the table where the function output is 0.**

100

Example

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

101

Example

f = (x' + y + z)(x' + y + z')(x' + y' + z')
$= M_4\ M_5\ M_7$
$= \Pi\ (4,5,7)$

f' = (x + y + z)(x + y + z')(x + y' + z)
(x + y' + z')(x' + y' + z)
$= M_0\ M_1\ M_2\ M_3\ M_6$
$= \Pi\ (0,1,2,3,6)$

f' contains all the maxterms not in f

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

102

Representation of Functions: Activity

- Representation of Function in minterms

F1 = x'y'z+xy'z'+xyz = m1 + m4 + m7
F2 = x'yz+xy'z+xyz'+xyz = m3 + m5 + m6 + m7

- Representation of Function in maxterms

F1 = (x+y+z).(x+y'+z).(x+y'+z').(x'+y+z').(x'+y'+z)
= M0 . M2 . M3 . M5 . M6
F2 = (x+y+z).(x+y+z').(x+y'+z).(x'+y+z)
= M0 . M1 . M2 . M4

Boolean expression represented as sum of minterms or product of maxterms are said to be in Canonical Form

x	y	z	F1	F2
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

103

Minterms and Maxterms

- For n variables there are 2^n minterms and maxterms
- Each maxterm is the complement of corresponding minterm and vice versa

$$\overline{m}_j = M_j$$

$$\overline{m}_3 = \overline{\overline{X}YZ} = X + \overline{Y} + \overline{Z} = M_3$$

105

Sum of Minterms: Activity

- Given the truth table, express F1 in sum of minterms

x	y	z	F_1	F_1'
0	0	0	0	1
0	0	1	$\boxed{1}$	0
0	1	0	0	1
0	1	1	0	1
1	0	0	$\boxed{1}$	0
1	0	1	0	0
1	1	0	0	0
1	1	1	$\boxed{1}$	0

$$F_1(x, y, z) = \sum(1,4,7) = m_1 + m_4 + m_7$$

$$= (x'y'z) + (xy'z') + (xyz)$$

106

Product of Maxterms: Activity

- Given the truth table, express F1 in Product of Maxterms

x	y	z	F_1	F_1'
0	0	0	$\boxed{0}$	1
0	0	1	1	0
0	1	0	$\boxed{0}$	1
0	1	1	$\boxed{0}$	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

$$F_1(x,y,z) = \prod(0,2,3) = M_0 \cdot M_2 \cdot M_3 = (x+y+z)(x+y'+z)(x+y'+z')$$

107

Another Example

- Find Sum of Minterms and Product of MaxTerms of *F= xy+x'z*
- Find Sum of Minterms & Product of Maxterms from truth table

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

- $F(x,y,z)= \sum(1,3,6,7)$
- $F(x,y,z)= \prod(0,2,4,5)$

108

References

- Chapter 2 – Digital Design Morris Mano
- Digital Logic Circuits – V.D. Agrawal, Auburn University, Auburn
- Logic System Design – Seattle Pacific University
- Template is taken from slides carnival.

SlidesCarnival

109