EE1005 – Digital Logic Design

- **Lecture Slides**
- **Week 3**

Course Instructor:

Dr. Arslan Ahmed Amin

FAST National University of Computer and Emerging Sciences CFD Campus

Signed Binary to Decimal Conversion (1/2)

- If MSB is 0
  - Convert the binary number to decimal like unsigned number

Example

Convert signed $(0100\ 1100)_2$ to decimal equivalent

Solution

As MSB is 0 so number is positive

$$\text{Dec} = (1 \times 2^6) + (1 \times 2^3) + (1 \times 2^2)$$

$$\text{Dec} = 64 + 8 + 4$$

$$\text{Dec} = 76$$

$$\Rightarrow (0100\ 1100)_2 = (+76)_{10}$$

Signed Binary to Decimal Conversion (2/2)

- If MSB is 1, then follow these three steps
  - Compute the 2's complement of number
  - Convert to decimal
  - Place a negative sign with answer

Example

Convert signed $(1100\ 1100)_2$ to decimal equivalent

Solution

As MSB is 1 so number is negative

Step 1:

Compute 2's complement of $(1100\ 1100)_2$ which is $(00110100)_2$

Step 2:

Convert the complemented number to decimal

$(00110100)_2 = (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^2) = 32 + 16 + 4 = (52)_{10}$

Step 3:

Place a negative sign with answer

$\Rightarrow (1100\ 1100)_2 = (-52)_{10}$

Signed Decimal to Binary Conversion (1/2)

- If the number is positive
  - Convert to binary by using repeated division method like unsigned numbers

Example

Convert $(+25)_{10}$ to binary equivalent

Solution

As the number is positive, so we will use the repeated division method

$\Rightarrow (+25)_{10} = (11001)_2$

2	25		
2	12	-	1
2	6	-	0
2	3	-	0
2	1	-	1
	0	-	1

Signed Decimal to Binary Conversion (2/2)

- If the number is negative
  - Convert the magnitude of the number to binary by using correct number of bits
  - Compute the 2's complement to get the final answer

Example

Convert $(-25)_{10}$ to binary equivalent

Solution

As the number is negative so we need to follow two steps

Step 1:

Convert the magnitude of (-25) to binary

And we need at least 6 bits to represent (-25)

$(25)_{10} = (\mathbf{0}11001)_2$

Zero Appended to Complete 6 bits

Step 2:

Compute the 2's complement to get the final answer

2's complement of $(011001)_2$ is $(100111)_2$

$\Rightarrow (-25)_{10} = (100111)_2$

In case we have used 5 bits in step 1, the final answer would be $(00111)_2$, which is obviously incorrect

Arithmetic Subtraction in Signed Binary Numbers

- For subtraction we prefer 2's complement method
  - M-N case in which M>N
  - A carry out of the sign-bit position is discarded.

Example

Subtract 14 from 26 by using 2's complement method.

Solution

$26 = (11010)_2$

$14 = (01110)_2$

$\Rightarrow -14 = (10010)_2$

$$\begin{array}{rcr} 26 & = & 1\ 1\ 0\ 1\ 0 \\ \underline{-14} & = & +\ \underline{\ \ 1\ 0\ 0\ 1\ 0} \\ +12 & = & 1\ 0\ 1\ 1\ 0\ 0 \end{array}$$

End Carry Discarded

$\boxed{\text{Final Answer} = (01100)_2 = (+12)_{10}}$

Arithmetic Addition in Signed Binary Numbers

- M-N case in which M<N
- A carry out of the sign-bit position is discarded
- Take 2's complement and add a negative sign

Example

Compute (+6) + (-13) in binary.

Solution

$+6 = (00110)_2$

$+13 = (01101)_2$

$\Rightarrow -13 = (10011)_2$

$$\begin{array}{lcr} +6 & = & 0\ 0\ 1\ 1\ 0 \\ \underline{-13} & = & +\ \underline{1\ 0\ 0\ 1\ 1} \\ -7 & = & 1\ 1\ 0\ 0\ 1 \end{array}$$

- $\Rightarrow$ $\boxed{-(00111)_2 = (-7)_{10}}$

Outline

- Boolean Algebra
- Boolean Functions
- Canonical Forms
  - Minterms and Maxterms
- Standard forms
  - Sum of Products (SOP)
  - Product of Sums (POS)
- Digital Logic Gates
- Integrated Circuits

Basic Definitions (1/2)

- Boolean algebra, like any other deductive mathematical system, may be defined with a set of elements, a set of operators, and a number of unproved axioms
- Set
  - A set of elements is any collection of objects, usually having a common property
  - If S is a set, and x and y are certain objects, then the notation $x \in S$ means that x is a member of the set S and $y \notin S$ means that y is not an element of S
  - A set of elements is specified by braces: $A = \{1, 2, 3, 4\}$

Basic Definitions (2/2)

- Binary Operator
  - A binary operator defined on a set S of elements is a rule that assigns, to each pair of elements from S, a unique element from S
  - As an example, consider the relation $a * b = c$
  - We say that * is a binary operator if it specifies a rule for finding c from the pair (a, b) and also if $a, b, c \in S$
  - However, * is not a binary operator if $a, b \in S$, and if $c \notin S$

The Postulates of a Mathematical System

1) Closure
   - A set is said to be closed if the result obtained after applying a binary operator belongs to the same set
2) Associative Law
   - $(x * y) * z = x * (y * z)$ for all $x, y, z, \in S$
3) Commutative Law
   - $(x * y) = (y * x)$ for all $x, y, \in S$
4) Identity Element
   - $e * x = x * e = x$ for every $x \in S$
5) Inverse
   - $x * y = e$ (y is inverse and e is identity element)
6) Distributive Law
   - $x * (y . z) = (x * y) . (x * z)$

Two Valued Boolean Algebra

- A two-valued Boolean algebra is defined on a set of two elements, B = {0, 1}, with rules for the two binary operators + and . and complement as shown in the following operator tables

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

- These rules are exactly the same as the AND, OR, and NOT operations, respectively

Basic Theorems and Properties of Boolean Algebra (1/2)

- Duality
  - It states that every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators and identity elements are interchanged
  - If the dual of an algebraic expression is desired, we simply interchange OR and AND operators and replace 1's by 0's and 0's by 1's

Basic Theorems and Properties of Boolean Algebra (2/2)

Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

Operator Precedence

- The operator precedence for evaluating Boolean expressions
  1. Parentheses
  2. NOT
  3. AND
  4. OR

Boolean Functions (1/2)

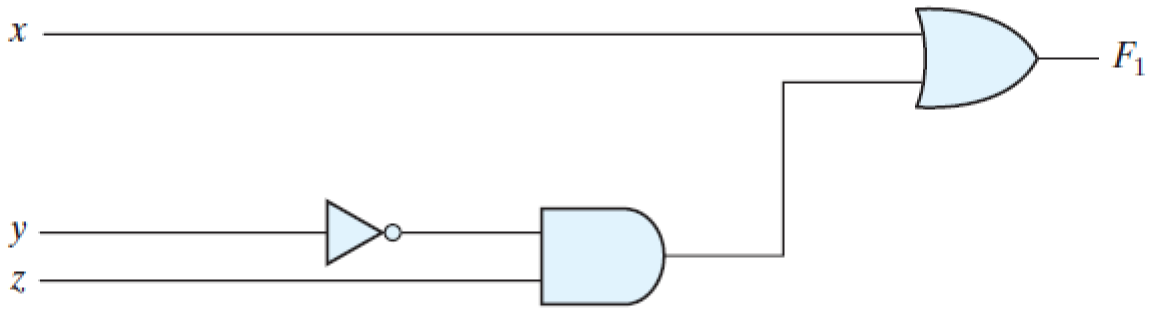
- A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols

$$F_1 = x + y'z$$

- A Boolean function can be represented in a truth table
- The number of rows in the truth table is 2^n, where n is the number of variables in the function
- A Boolean function can be transformed from an algebraic expression into a circuit diagram (also called schematic) composed of logic gates connected in a particular structure

Boolean Functions (2/2)

$$F_1 = x + y'z$$

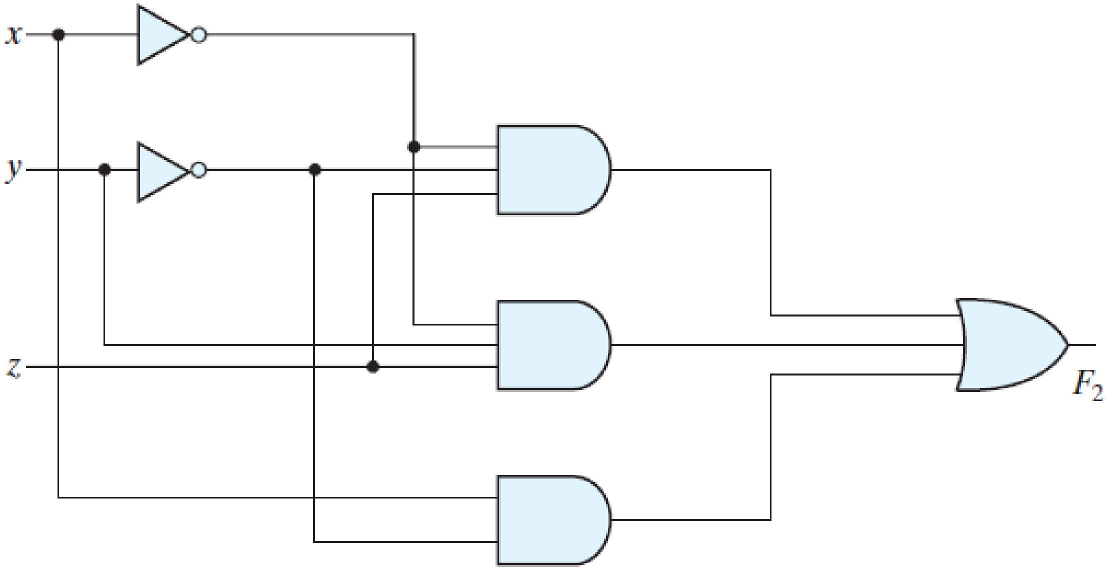


x	y	z	F_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Practice Problem

- Construct the truth table and circuit diagram for the following Boolean function

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



x	y	z	x'	y'	x'y'z	x'yz	xy'	F_2
0	0	0	1	1	0	0	0	0
0	0	1	1	1	1	0	0	1
0	1	0	1	0	0	0	0	0
0	1	1	1	0	0	1	0	1
1	0	0	0	1	0	0	1	1
1	0	1	0	1	0	0	1	1
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Algebraic Manipulation (1/2)

- When a Boolean expression is implemented with logic gates, each term requires a gate and each variable within the term designates an input to the gate
- We define a literal to be a single variable within a term
- The function $F_2 = x'y'z + x'yz + xy'$ has 3 terms and 8 literals
- By reducing the number of terms, the number of literals, or both in a Boolean expression, it is often possible to obtain a simpler circuit
- The manipulation of Boolean algebra consists mostly of reducing an expression for the purpose of obtaining a simpler circuit

Algebraic Manipulation (2/2)

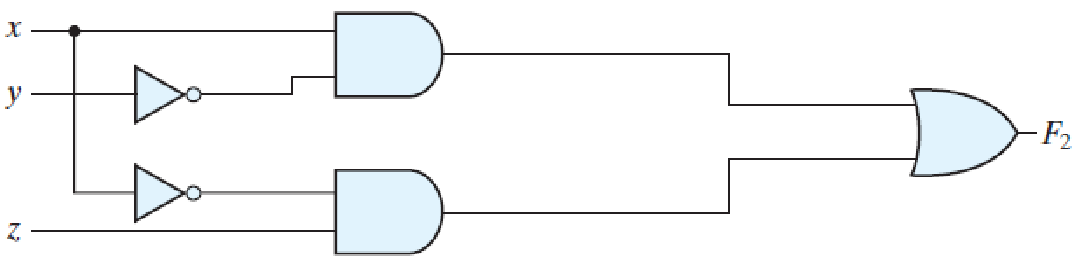
- The simple manipulation consists of applying basic theorems and relations
- The function $F_2 = x'y'z + x'yz + xy'$ can be simplified in the following way

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

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

$F_2 = x'z(1) + xy'$

$F_2 = x'z + xy'$



- The same function is reduced to two terms. The truth table will be the same and the schematic is now simplified

Complement of a Function

- The complement of a function F is F’ and can be obtained by an interchange of 1’s for 0’s and 0’s for 1’s in the value of the function
- The complement of a function may be derived algebraically through DeMorgan’s theorems
- DeMorgan’s theorems for any number of variables resemble the two-variable case

$$(A + B + C + \ldots + F)' = A'B'C' \ldots F'$$

$$(ABC \ldots F)' = A' + B' + C' + \ldots + F'$$

- The complement of a function is obtained by interchanging AND and OR operators and complementing each literal

Practice Problem

- Find the complements of the following functions by applying Demorgan's laws
  i. $F_1 = x'yz' + x'y'z$
  ii. $F_2 = x(y'z' + yz)$

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$

$$\begin{aligned} F_2' = [x(y'z' + yz)]' &= x' + (y'z' + yz)' = x' + (y'z')'(yz)' \\ &= x' + (y + z)(y' + z') \\ &= x' + yz' + y'z \end{aligned}$$

Another Way to Compute the Complement of a Boolean Function

- A simpler procedure for deriving the complement of a function is to take the dual of the function and complement each literal

Practice Problem

- Find the complements of the following functions by using duality
  i. $F_1 = x'yz' + x'y'z$
  ii. $F_2 = x(y'z' + yz)$

1. $F_1 = x'yz' + x'y'z$.
The dual of F_1 is $(x' + y + z')(x' + y' + z)$.
Complement each literal: $(x + y' + z)(x + y + z') = F_1'$

2. $F_2 = x(y'z' + yz)$.
The dual of F_2 is $x + (y' + z')(y + z)$.
Complement each literal: $x' + (y + z)(y' + z') = F_2'$

Property # 9 : $\overline{(x+y)} = \bar{x} \cdot \bar{y}$ De-Morgen Law

x	y	$\bar{x}$	$\bar{y}$	$x+y$	$\overline{x+y}$	$\bar{x} \cdot \bar{y}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0
					L.H.S	R.H.S

Again: Property 9 $\quad \overline{xy} = \bar{x} + \bar{y}$ De-Morgan's law

x	y	$\bar{x}$	$\bar{y}$	xy	$\overline{xy}$	$\bar{x}+\bar{y}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0
					L.H.S	R.H.S

It is verified that:

L.H.S = R.H.S

$$\overline{xy} = \bar{x} + \bar{y}$$

Example: $F = A + \overline{B}C$, Find $\overline{F} = ?$

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

$$= \overline{A} \cdot \overline{\overline{B}C}$$

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

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

Another Example: $F = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}$, Find $\bar{F}$

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

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

$$\bar{F} = \overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}BC} \cdot \overline{A\bar{B}}$$

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

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

Simplify F to minimum no. of literals.

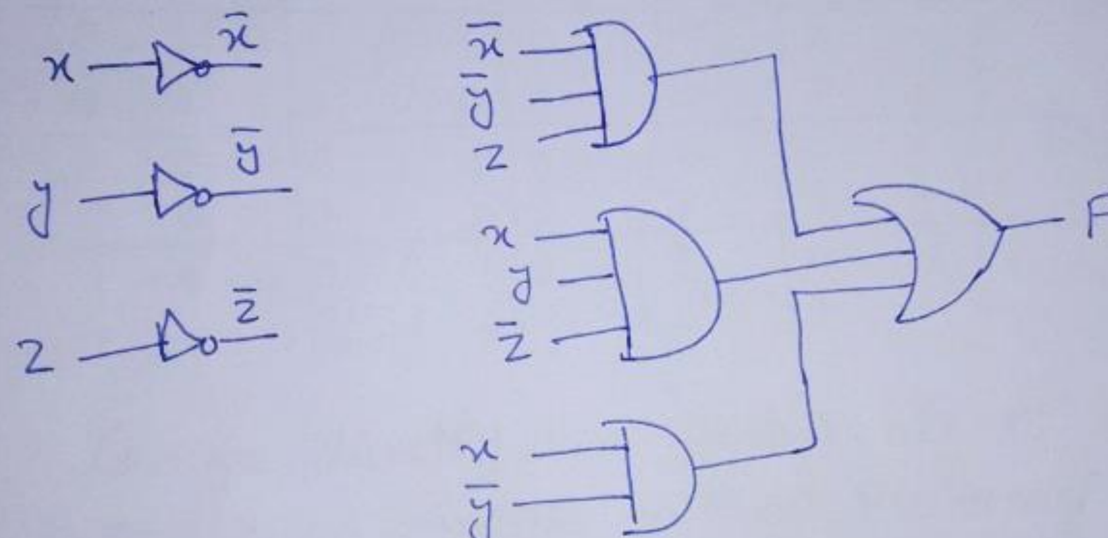
Given $F = A(\bar{A} + B)$
$= A\bar{A} + AB$
$= 0 + AB = AB$

Example: $F = (A+B)(A+\bar{B})$

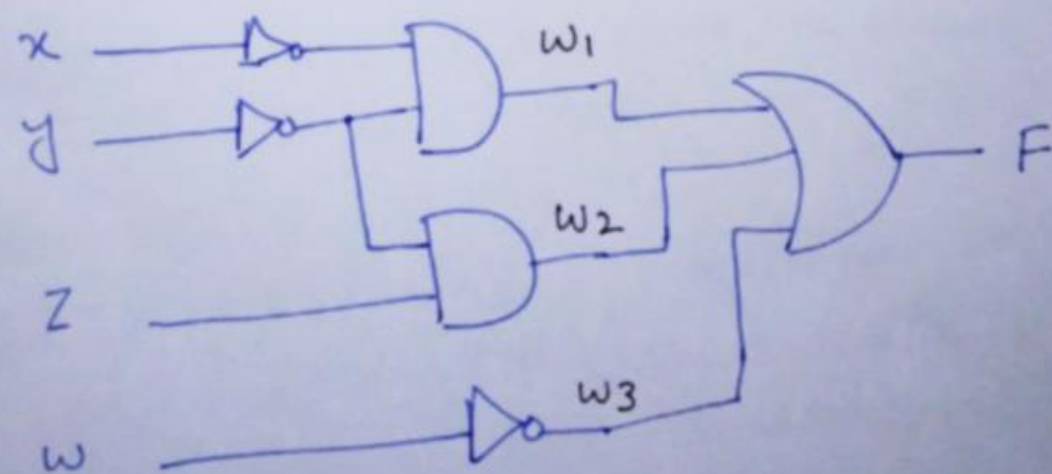
$= AA + A\bar{B} + BA + B\bar{B}$
$= A + A\bar{B} + BA + 0$
$= A(1 + \bar{B} + B) = A \cdot 1 = A$

① Drawing Circuit Diagram From Boolean Function:

$$F = \bar{x}\bar{y}z + xy\bar{z} + x\bar{y}$$



② Writing Boolean Function from Given Circuit



$w_1 = \bar{x}\,\bar{y}$; $w_2 = \bar{y}Z$; $w_3 = \bar{w}$

$F = w_1 + w_2 + w_3 = \bar{x}\,\bar{y} + \bar{y}Z + \bar{w}$

3 Drawing Truth Table of Boolean Function

10

$$F = \bar{x}\bar{y}z + xy\bar{z} + xyz$$

x	y	z	$\bar{x}$	$\bar{y}$	$\bar{z}$	$\bar{x}\bar{y}z$	$xy\bar{z}$	xyz	F
0	0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	1	0	0	1
0	1	0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	0	0	0	0	0
1	1	0	0	0	1	0	1	0	1
1	1	1	0	0	0	0	0	1	1

<u>standard Form</u>: When all variables are present in each literal.

Examples:

1. $F = \bar{x}\bar{y}z + xy\bar{z} + \bar{x}yz$ standard Form ✓
2. $G = \bar{x}\bar{y}z + x\bar{y}z + \bar{x}y$ Non standard Form ×
3. $H = \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}z$ Standard Form ✓
4. $I = xy\bar{z} + x\bar{y}z + \bar{x}yz + x\bar{y}\bar{z}$ stand Form ✓

Drawing Directly Truth Table from standard Function Form

11

① $F = \bar{x}\bar{y}z + xy\bar{z} + \bar{x}yz$

$\quad\;\; 0\,0\,1 \quad\; 1\,1\,0 \quad\; 0\,1\,1$

x	y	z	F	H	I
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	1	0	1
1	1	1	0	0	0

(4) How to write Boolean Function from Given Truth Table

A	B	C	F_1	F_2
0	0	0	0	1
0	0	1	1	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

Note:

if $A=0 \Rightarrow \overline{A}$

if $A=1 \Rightarrow A$

$$F_1 = \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}\,\overline{C} + ABC$$

$$F_2 = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C + \overline{A}\,\overline{B}C$$

Canonical Form

- Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form
- Minterms
  - The AND of N variables such that they equals to 1 is called minterm or standard product
  - There are 2^N possible minterms with N variables
  - Minterms are denoted by lower case m
- Maxterms
  - The OR of N variables such that the result is equal to 0 is called maxterm or standard sum
  - There are 2^N possible maxterms with N variables
  - Maxterms are denoted by upper case M

Calculating Minterms and Maxterms

- Each minterm is obtained from an AND term of the n variables, with each variable being primed if the corresponding bit of the binary number is a 0 and unprimed if a 1
- Each maxterm is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit is a 0 and primed if a 1
- The complement of a minterm is equal to its corresponding maxterm

Minterms and Maxterms for 3 Variables

x	y	z	Minterms		Maxterms	
			Term	Designation	Term	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	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

Representing a Boolean Function in Canonical Form

- Boolean function can be represented in canonical form in two ways
  - As a sum of minterms
  - As a product of maxterms
- To represent a Boolean function in canonical form all the terms of that function must contain all the variables

Assume following truth table is given

	A	B	C	F_1	F_2
0	0	0	0	1	0
1	0	0	1	0	1
2	0	1	0	1	1
3	0	1	1	1	0
4	1	0	0	0	0
5	1	0	1	0	1
6	1	1	0	1	1
7	1	1	1	1	0

Simplest way of writing expression for F_1, see location of 1's in truth table.

$$F_1 = \Sigma(0,2,3,6,7)$$

$$= m_0 + m_2 + m_3 + m_6 + m_7$$

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

These individual terms are called minterms.
i.e. terms corresponding to "1".

Note: if $A=1 \Rightarrow A$
$A=0 \Rightarrow \bar{A}$

Consider again F_1, now write expression for F_1 by looking at 0's location.

$$F_1 = \prod(1,4,5)$$

$$= M_1 \cdot M_4 \cdot M_5 = M_1 \cdot M_4 \cdot M_5$$

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

These individual terms are called maxterms i.e. terms corresponding to "0". Note: if $A=1 \Rightarrow \bar{A}$

$A=0 \Rightarrow A$

Now, let us write minterm / Max term express for F2.

Min term Expression for F2:

$F_2 = \sum(1, 2, 5, 6)$ ← Ist way

$= m_1 + m_2 + m_5 + m_6$ ← 2nd way

$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C}$ ← 3rd way

Maxterm Expression for F_2:

$$F_2 = \prod(0,3,4,7) \quad \leftarrow \text{1st way}$$

$$= M_0 \cdot M_3 \cdot M_4 \cdot M_7 \quad \leftarrow \text{2nd way}$$

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

SOP Expression: (Sum of Product)

Expression of function written using minterms

POS Expression: (Product of Sum)

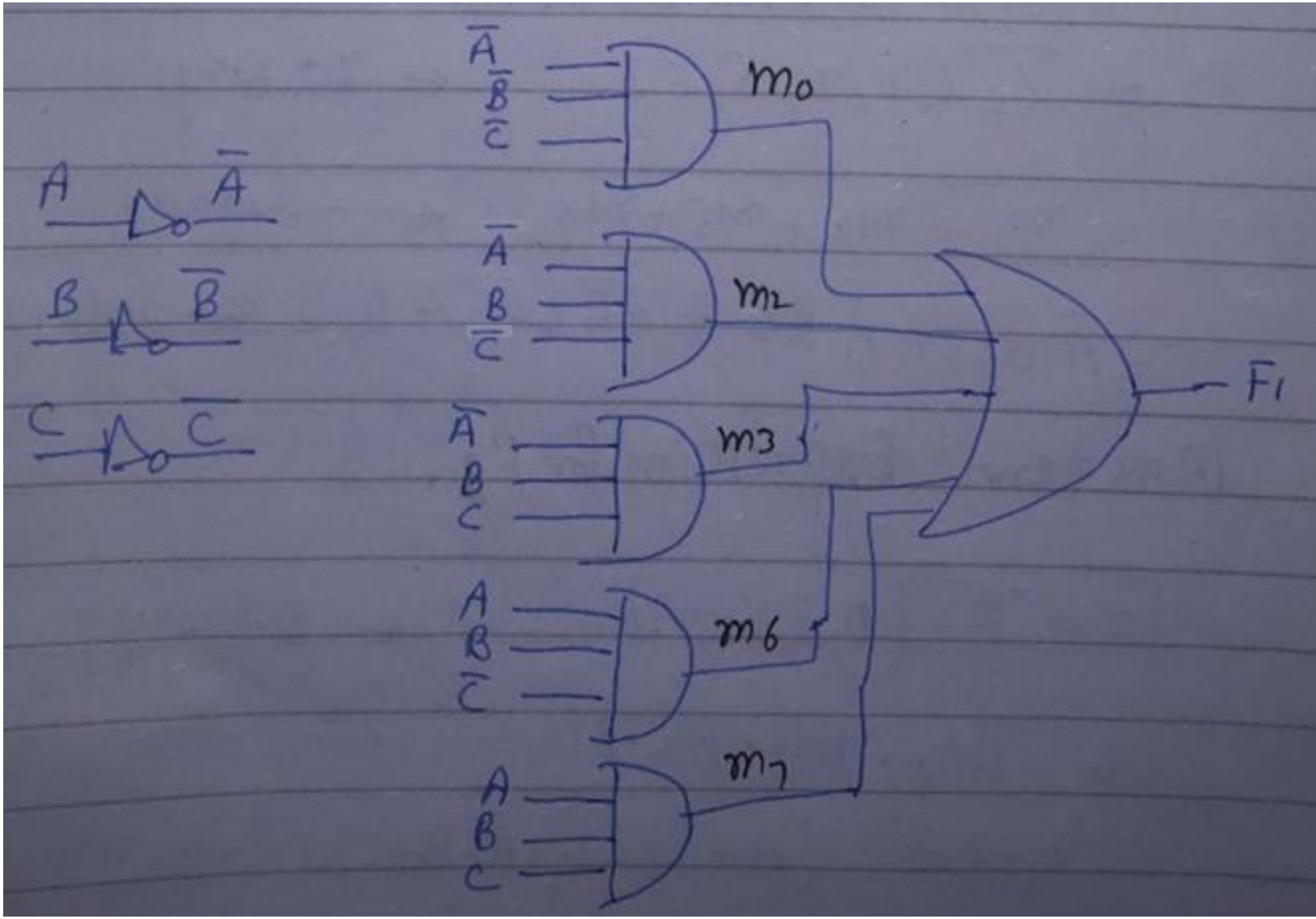
Expression of function using maxterms.

(6) Drawing Circuit Diagram using SOP/POS Expr.

SOP Expression Circuit:

$$F_1 = \sum(0,2,3,6,7) = m_0 + m_2 + m_3 + m_6 + m_7$$

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

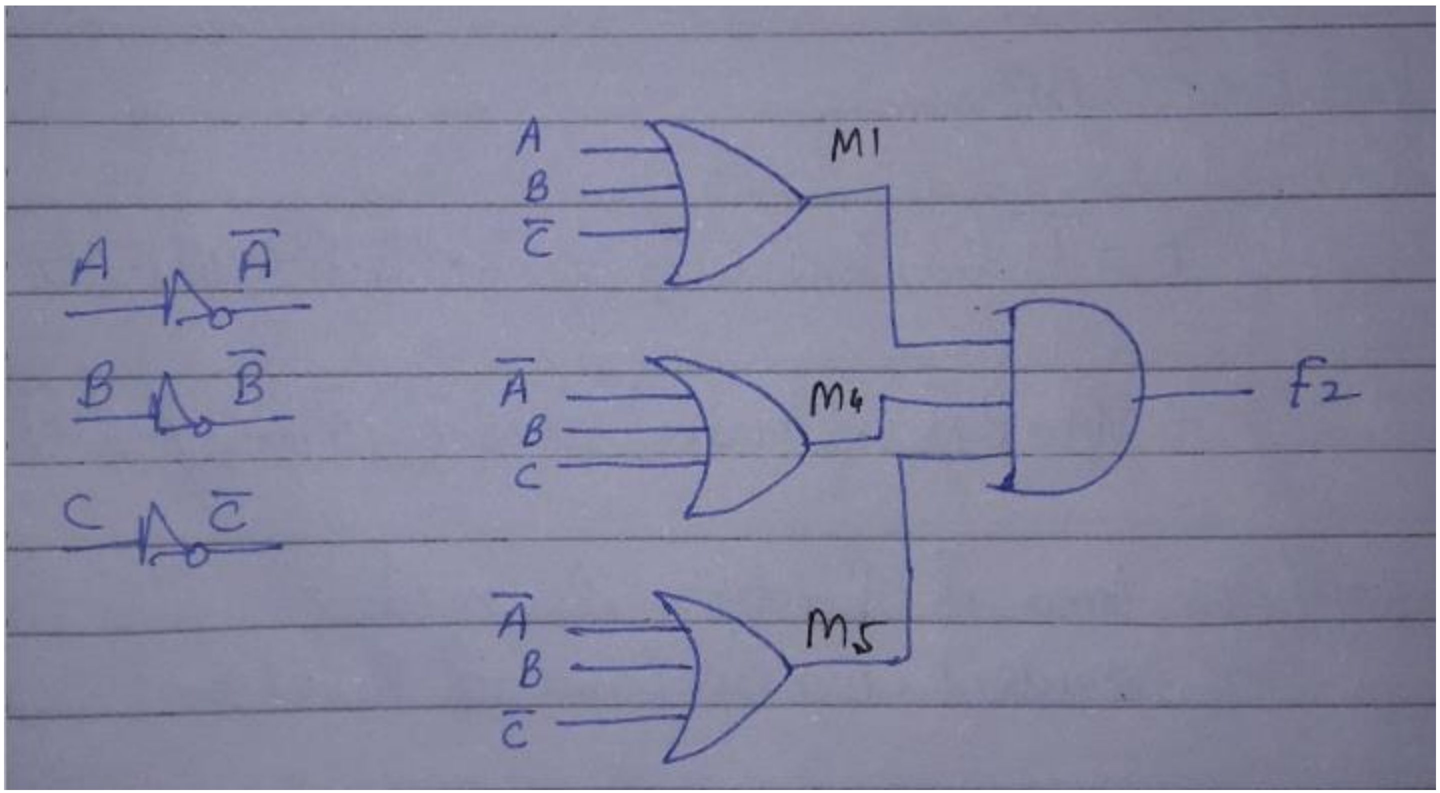

A
$\bar{A}$
B
$\bar{B}$
C
$\bar{C}$
m_0
m_2
m_3
m_6
m_7
F_1

POS Expression Circuit:

$$F_1 = \Pi(1, 4, 5)$$

$$= M_1 \cdot M_4 \cdot M_5$$

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


A
B
C̄
M1
A Ā
B B̄
C C̄
Ā
B
C
M4
f2
Ā
B
C̄
M5

(7) Conversion of given Function to standard SOP / POS form

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

A	B	C	$\overline{B}$	$\overline{B}C$	F
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

From above truth table, we can write;

SOP Expression:

$$F = \sum(1,4,5,6,7) = m_0 + m_4 + m_5 + m_6 + m_7$$

$$= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

POS Expression:

$$F = \prod(0,2,3) = M_0 \cdot M_2 \cdot M_3$$

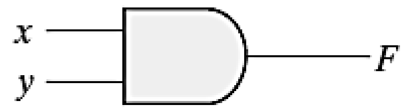
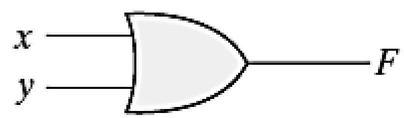
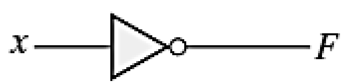
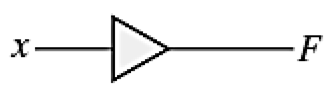
$$= (A+B+C)\cdot(A+\bar{B}+C)\cdot(A+\bar{B}+\bar{C})$$

Relationship b/w SOP/POS Forms:

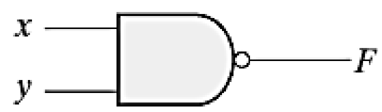
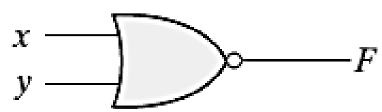
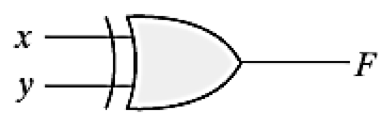
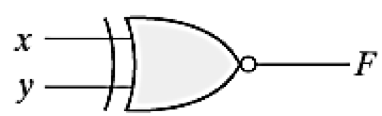
SOP: $F_{sop} = \Sigma(1, 4, 5, 6, 7) = m_0 + m_4 + m_5 + m_6 + m_7$

POS: $F_{pos} = \Pi(0, 2, 3) = M_0 \cdot M_2 \cdot M_3$

Digital Logic Gates (1/2)

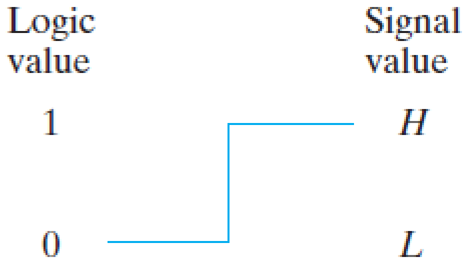
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1

Digital Logic Gates (2/2)

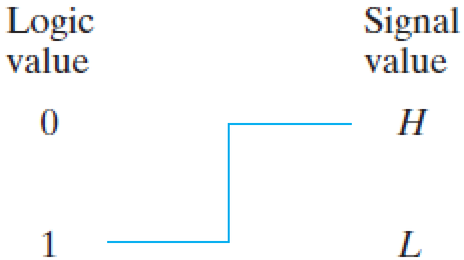
Name	Graphic symbol	Algebraic function	Truth table
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Positive and Negative Logic

- The binary signal at the inputs and outputs of any gate has one of two values
- One signal value represents logic 1 and the other logic 0
- Since two signal values are assigned to two logic values, there exist two different assignments of signal level to logic value
- Choosing the high-level H to represent logic 1 defines a positive logic system
- Choosing the low-level L to represent logic 1 defines a negative logic system

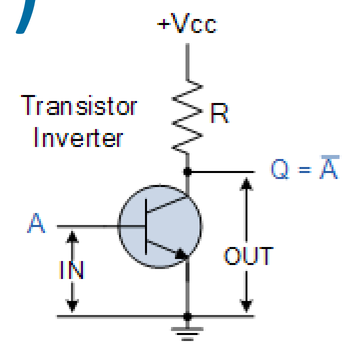


(a) Positive logic

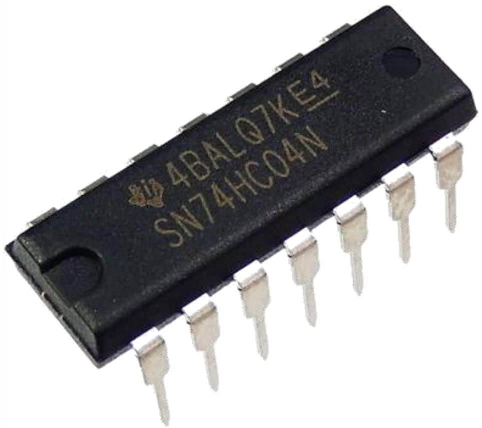


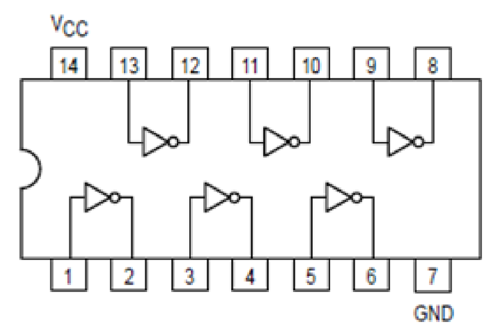
(b) Negative logic

Integrated Circuits (1/4)



- An integrated circuit (IC) is fabricated on a die of a silicon semiconductor crystal, called a chip, containing the electronic components for constructing digital gates
- The various gates are interconnected inside the chip to form the required circuit
- The number of pins may range from 3 on a small IC package to several thousand on a larger package
- Each IC has a numeric designation printed on the surface of the package for identification





Integrated Circuits (2/4)

- Level of Integration
  - Small Scale Integration (SSI)
    - The number of gates is usually fewer than 10
    - AND, OR, XOR etc ICs
  - Medium Scale Integration (MSI)
    - Have a complexity of approximately 10 to 1,000 gates in a single package
    - Decoders, adders, and multiplexers
  - Large Scale Integration (LSI)
    - These devices contain thousands of gates in a single package.
    - They include digital systems such as processors, memory chips, and programmable logic devices
  - Very Large Scale Integration (VLSI)
    - Contain millions of gates within a single package
    - Examples are large memory arrays and complex microcomputer chips

Integrated Circuits (3/4)

- Digital Logic Families
  - The circuit technology of an IC is referred to as a digital logic family
  - TTL (transistor-transistor logic)
    - 50 years old and standard method
  - ECL (emitter coupled logic)
    - For high speed operations
  - MOS (metal oxide semiconductor)
    - Suitable for circuits that need high component density
  - CMOS (complementary metal-oxide semiconductor)
    - CMOS is preferable in systems requiring low power consumption

Integrated Circuits (4/4)

- Fan-out
  - It specifies the number of standard loads that the output of a typical gate can drive without impairing its normal operation
- Fan-in
  - It is the number of inputs available in a gate
- Power dissipation
  - It is the power consumed by the gate that must be available from the power supply
- Propagation delay
  - It is the average transition delay time for a signal to propagate from input to output
- Noise margin
  - It is the maximum external noise voltage added to an input signal that does not cause an undesirable change in the circuit output