FACULTY OF ENGINEERING
DAMIETTA UNIVERSITY
كلية الهندسة-جامعة دمياط

جامعة دمياط
Damietta University

Digital Circuits

Boolean Algebra and Logic Gate

Chapter 2

Boolean Algebra and Logic Gate

Lecture 5

Dr. Marwa Fayez Fahmy

Description of the Laws of Boolean Algebra

- **Annulment Law 'Theorem 2'**– A term AND´ed with a "0" equals 0 or OR´ed with a "1" will equal 1
  - A . 0 = 0 A variable AND'ed with 0 is always equal to 0
  - A + 1 = 1 A variable OR'ed with 1 is always equal to 1

- **Identity Law 'Postulate 2'**– A term OR´ed with a "0" or AND´ed with a "1" will always equal that term
  - A + 0 = A A variable OR'ed with 0 is always equal to the variable
  - A . 1 = A A variable AND'ed with 1 is always

- **Idempotent Law 'Theorem 1'** – An input that is AND´ed or OR´ed with itself is equal to that input
  - A + A = A A variable OR'ed with itself is always equal to the variable
  - A . A = A A variable AND'ed with itself is always equal to the variable

- **Complement Law 'Postulate 5'** – A term AND´ed with its complement equals "0" and a term OR´ed with its complement equals "1"

-
  - A . A = 0 A variable AND'ed with its complement is always equal to 0
  - A + A = 1 A variable OR'ed with its complement is always equal to 1

- **Commutative Law 'Postulate 3'**– The order of application of two separate terms is not important
  - A . B = B . A The order in which two variables are AND'ed makes no difference
  - A + B = B + A The order in which two variables are OR'ed makes no difference
- **Double Negation Law - Involution 'Theorem 3'**– A term that is inverted twice is equal to the original term
  - (A')' = A A double complement of a variable is always equal to the variable

- **Distributive Law** **'Postulate 4'** – This law permits the multiplying or factoring out of an expression.
-
  - A(B + C) = A.B + A.C (OR Distributive Law)
  - A + (B.C) = (A + B).(A + C) (AND Distributive Law)
-
- **Absorptive Law** **'Theorem 6'** – This law enables a reduction in a complicated expression to a simpler one by absorbing like terms.
-
  - A + (A.B) = A (OR Absorption Law)
  - A(A + B) = A (AND Absorption Law)

- **Associative Law** ’Theorem 4’– This law allows the removal of brackets from an expression and regrouping of the variables.
  - A + (B + C) = (A + B) + C = A + B + C (OR Associate Law)
  - A(B.C) = (A.B)C = A . B . C (AND Associate Law)

- **de Morgan´s Theorem** ‘Theorem 5 ‘ – There are two “de Morgan´s” rules or theorems,

(1) Two separate terms NOR´ed together is the same as the two terms inverted (Complement) and AND´ed for example: A+B = A . B

(2) Two separate terms NAND´ed together is the same as the two terms inverted (Complement) and OR´ed for example: A.B = A + B

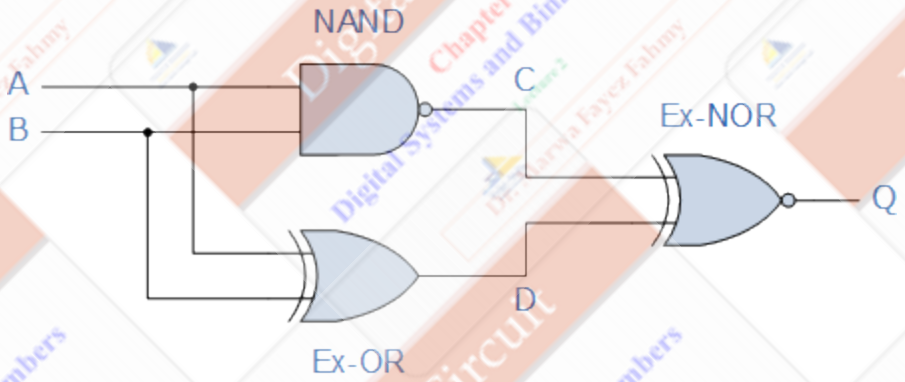
Operator Precedence

- The operator precedence for evaluating Boolean Expression is
  - Parentheses
  - NOT
  - AND
  - OR
- Examples
  - $x\,y' + z$
  - $(x\,y + z)'$

Boolean Algebra Examples

Boolean Algebra Example No1

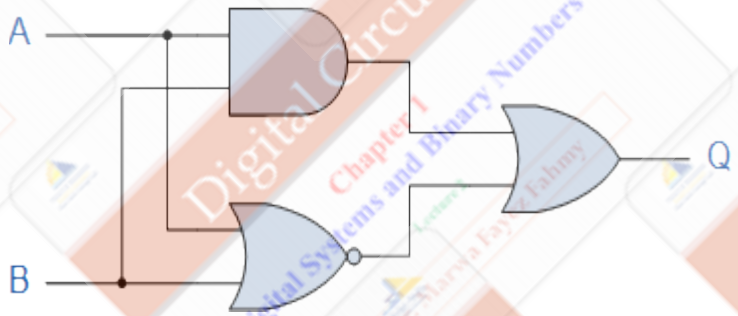
- Construct a Truth Table for the logical functions at points C, D and Q in the following circuit and identify a single logic gate that can be used to replace the whole circuit.

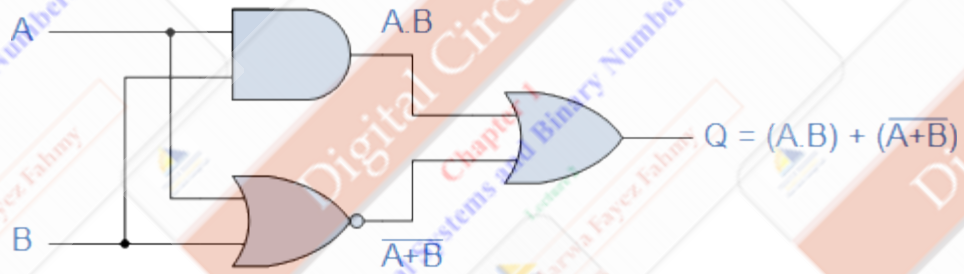


Inputs		Output at		
A	B	C	D	Q
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

Boolean Algebra Example No2

- Find the Boolean algebra expression for the following system.

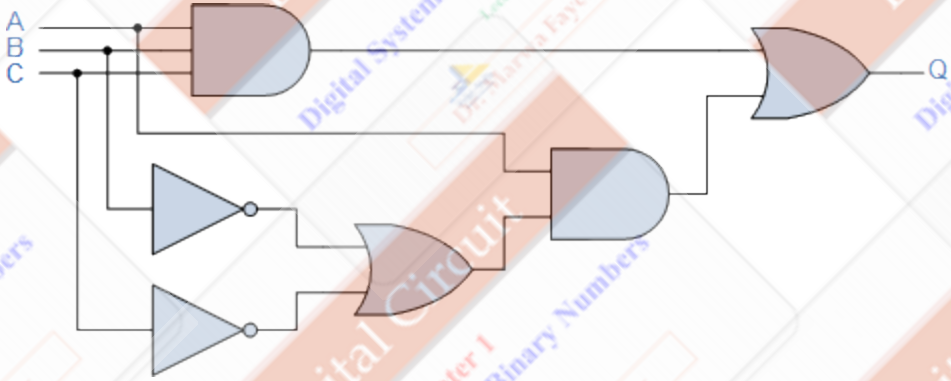


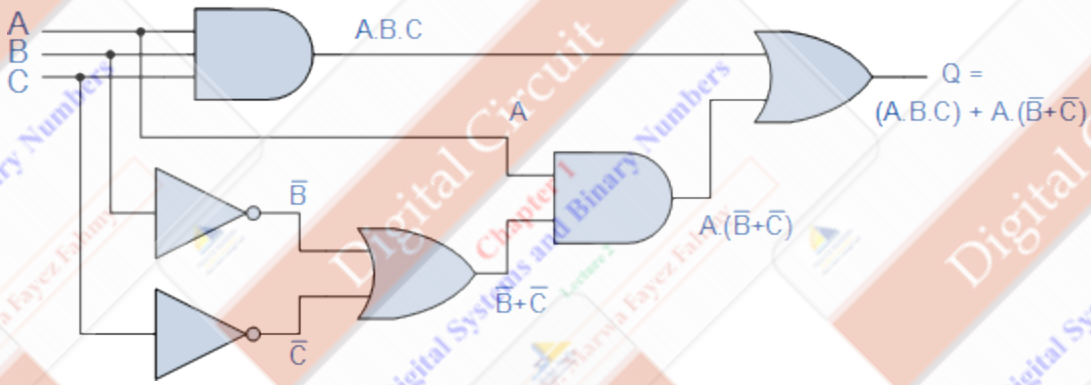


Inputs		Intermediates		Output
B	A	A.B	A + B	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Boolean Algebra Example No 3

- Find the Boolean algebra expression for the following system.





Inputs			Intermediates					Output
C	B	A	A.B.C	B	C	B+C	A.(B+C)	Q
0	0	0	0	1	1	1	0	0
0	0	1	0	1	1	1	1	1
0	1	0	0	0	1	1	0	0
0	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1

Report

Proof That

1. $xy + x'z + yz = xy + x'z$
2. $(x+y)\cdot(x'+z)\cdot(y+z) = (x+y)\cdot(x'+z)$ -- (dual)

- **Proof:**

$$xy + x'z + yz = xy + x'z + (x+x')yz$$
$$= xy + x'z + xyz + x'yz$$
$$= (xy + xyz) + (x'z + x'zy)$$
$$= xy + x'z$$

QED (2 true by duality).

Boolean Functions

- A Boolean function
  - Binary variables
  - Binary operators OR and AND
  - Unary operator NOT
  - Parentheses
- Examples
  - $F_1 = x\,y\,z'$
  - $F_2 = x + y'z$
  - $F_3 = x'\,y'\,z + x'\,y\,z + x\,y'$
  - $F_4 = x\,y' + x'\,z$

Boolean Functions

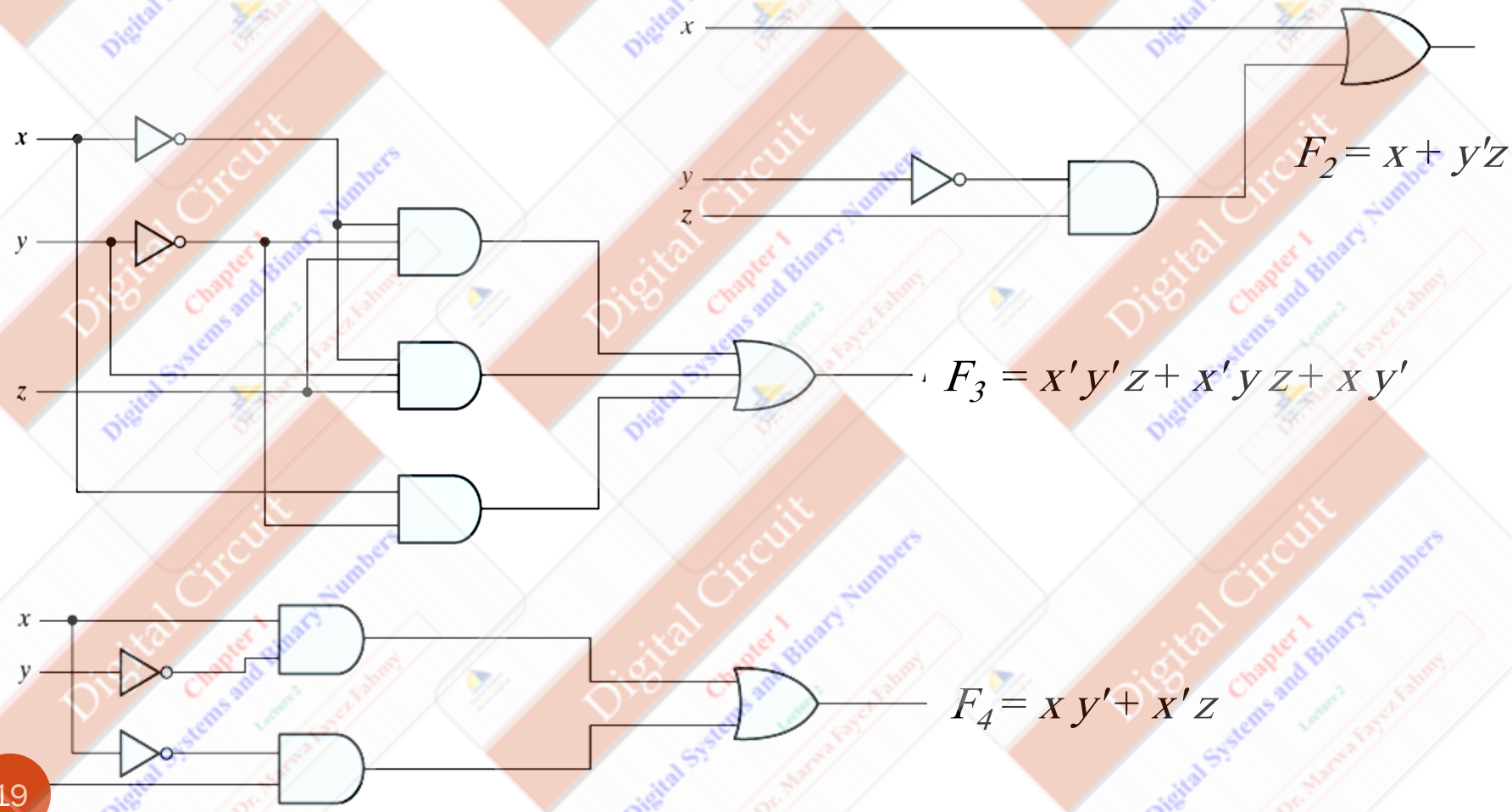
- The truth table of 2^n entries

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

- Two Boolean expressions may specify the same function
- $F_3 = F_4$

Boolean Functions

- Implementation with logic gates
  - F_4 is more economical



Algebraic Manipulation

- To minimize Boolean expressions
  - Literal: a primed or unprimed variable (an input to a gate)
  - Term: an implementation with a gate
  - The minimization of the number of literals and the number of terms → a circuit with less equipment
  - It is a hard problem (no specific rules to follow)
- Example 2.1

1. $x(x'+y) = xx' + xy = 0+xy = xy$
2. $x+x'y = (x+x')(x+y) = 1\,(x+y) = x+y$
3. $(x+y)(x+y') = x+xy+xy'+yy' = x(1+y+y') = x$
4. $xy + x'z + yz = xy + x'z + yz(x+x') = xy + x'z + yzx + yzx' = xy(1+z) + x'z(1+y) = xy + x'z$
5. $(x+y)(x'+z)(y+z) = (x+y)(x'+z)$, by duality from function 4. (*consensus theorem* with duality)

Complement of a Function

- An interchange of 0's for 1's and 1's for 0's in the value of F
  - By DeMorgan's theorem
  - $(A+B+C)' = (A+X)'$ let $B+C = X$
    $= A'X'$ by theorem 5(a) (DeMorgan's)
    $= A'(B+C)'$ substitute $B+C = X$
    $= A'(B'C')$ by theorem 5(a) (DeMorgan's)
    $= A'B'C'$ by theorem 4(b) (associative)
- *Generalizations*: a function is obtained by interchanging AND and OR operators and complementing each literal.
  - $(A+B+C+D+ \ldots +F)' = A'B'C'D' \ldots F'$
  - $(ABCD \ldots F)' = A'+ B'+C'+D' \ldots +F'$

Examples

- Example 2.2
  - $F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x+y'+z)(x+y+z')$
  - $F_2' = [x(y'z'+yz)]' = x' + (y'z'+yz)' = x' + (y'z')'(yz)'$
    $= x' + (y+z)(y'+z')$
    $= x' + yz' + y'z$
- Example 2.3: a simpler procedure
  - Take the dual of the function and complement each literal

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


إستفسار