



UNIVERSITI
TEKNOLOGI
MALAYSIA
www.utm.my

SCSR1013 DIGITAL LOGIC

MODULE 4a: BOOLEAN ALGEBRA

FACULTY OF COMPUTING



Introduction to Boolean Algebra

- It is the **set of rules** used to simplify a given **logic expression** without changing its functionality.

- Example: $F = \bar{A}B + BC + ABC$



be minimized

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

- It is used when number of variables are less, i.e., 1, 2 or 3 vars.



- **Variable**

- A symbol that represents a logical quantity
- Usually italic uppercase (A , B , C , D)
- A single variable can have a 1 or 0 value

- **Complement**

- The inverse of a variable
- Indicated by an overbar (\overline{A}) or prime (A')
- If $A = 1$, then $\overline{A} = 0$

- **Literal**

= both variable and its complement in a term

$$\overline{A} + B + C \rightarrow 3 \text{ literals}$$

Example:

$$f = x'y' + x'y = \overline{\overline{x}}\overline{\overline{y}} + \overline{x}y$$

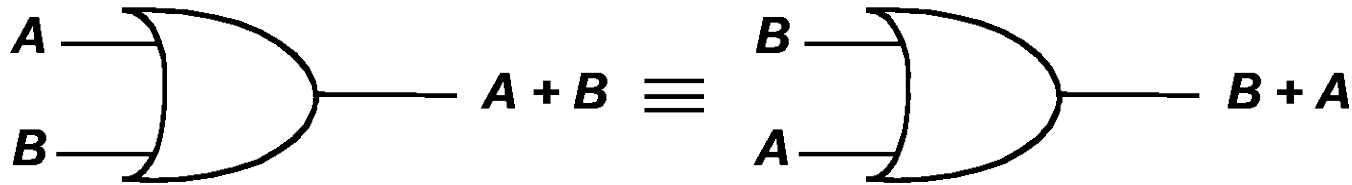
Total Number of Literals = 4 which are x' , y' , x and y

Total Number of variables = 2 which are x and y

Total Number of terms = 2 which are $x'y'$ and $x'y$

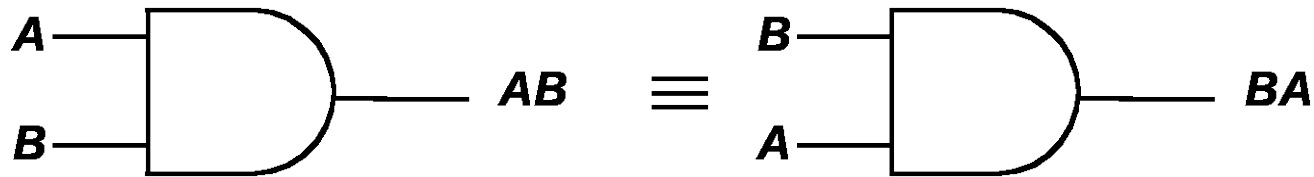


- Basic laws of BA
 - **Commutative Laws**
 - For addition and multiplication
 - **Associative Laws**
 - For addition and multiplication
 - **Distributive Laws**



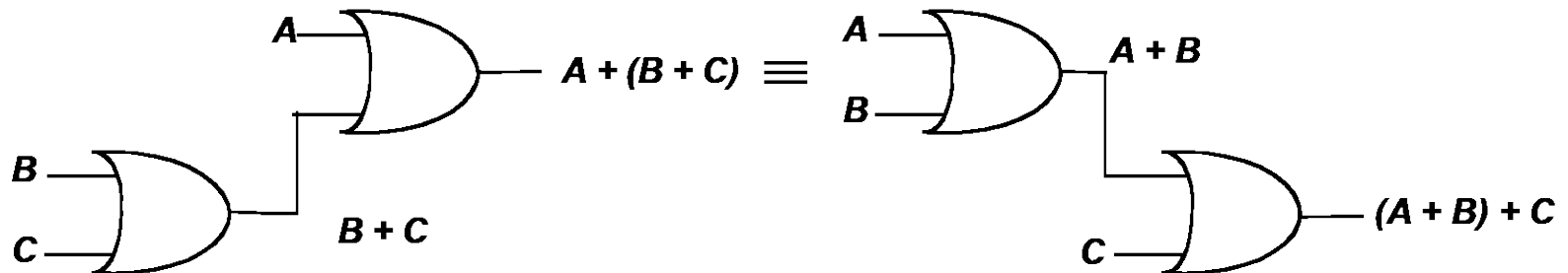
Commutative law of addition

- $A + B = B + A$
 - the **order** in which the variables are OR-ed makes no difference
 - in logic circuits, addition and the OR operation are the same.



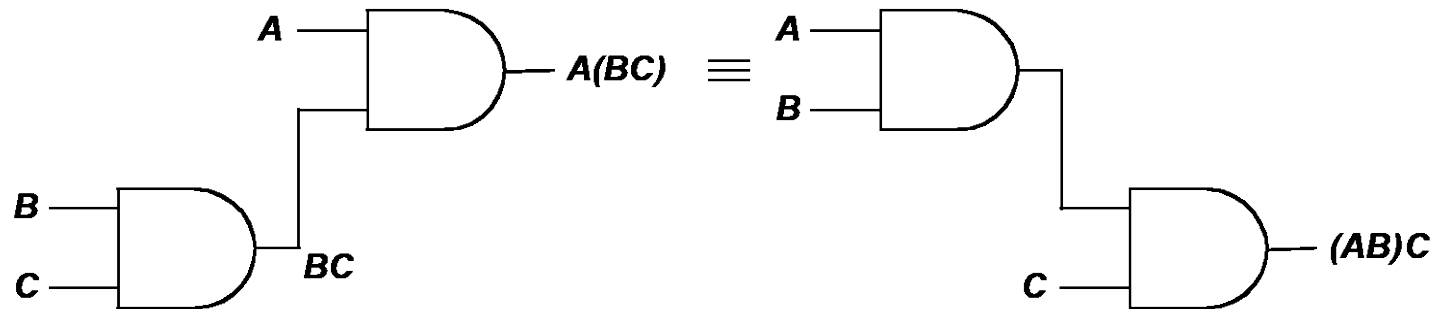
Commutative law of multiplication

- $AB = BA$
 - the **order** in which the variables are AND-ed makes no difference
 - in logic circuits, multiplication and the AND operation are the same.



Associative law of addition

- $A + (B + C) = (A + B) + C$
- the **grouping** in which the variables are OR-ed makes no difference



Associative law of multiplication

- $A(BC) = (AB)C$
 - the **grouping** in which the variables are AND-ed makes no difference

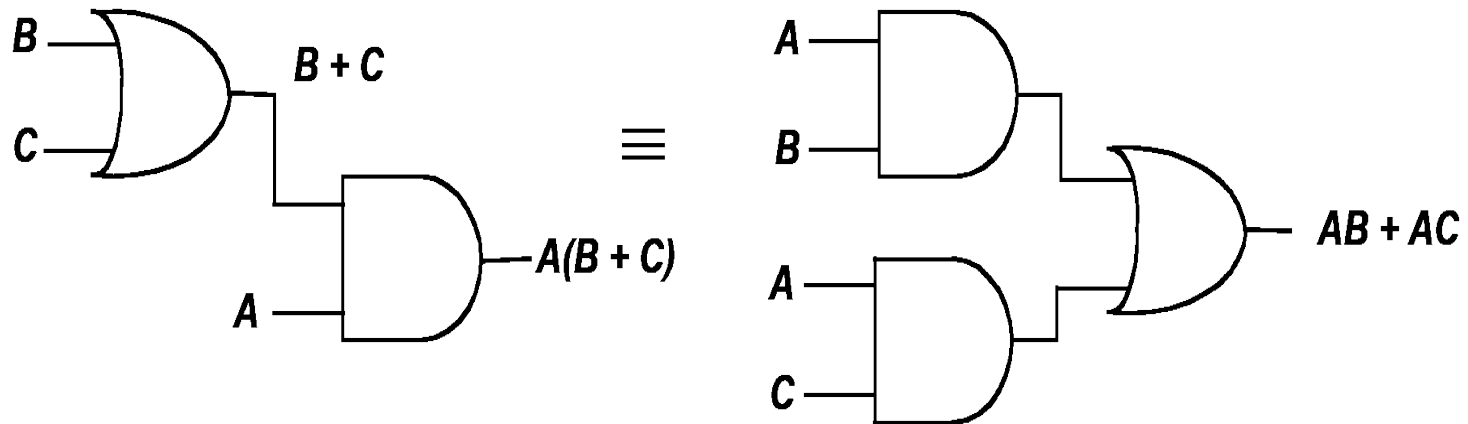


Exercise 4a.1:

Prove the Associate Law for $A(BC) = (AB)C$ using truth table.

A	B	C	AB	BC	A(BC)	(AB)C
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

- $A(B + C) = AB + AC$
 - expresses the process of factoring, variable A is **factored out** of the product terms, like $AB + AC = A(B+C)$
 - ... **A** is factored to the **(B + C)**



Why we have this ???

- want to simplify the Boolean equation
- thus we have simple circuit
- then reduce cost of production !

1

The rules can be proven by using truth table.

Rule 10

A	B	AB	A + AB
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

↑ equal ↑

Rules of Boolean Algebra

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

Rules of Boolean Algebra

2 The rules can be proven by using Boolean algebra laws and rules.

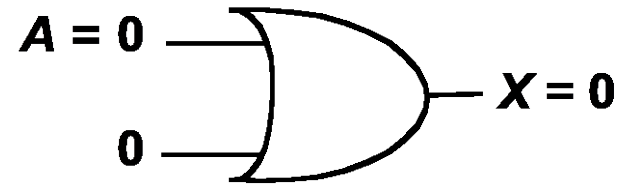
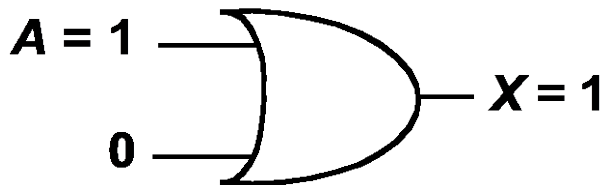
Rule 10:

$$\begin{aligned} A + AB &= A(1 + B) && \text{Factoring (distributive law)} \\ &= A \cdot 1 && \text{Rule 2: } (1 + B) = 1 \\ &= A && \text{Rule 4: } A \cdot 1 = A \end{aligned}$$

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

Rules of Boolean Algebra: PROOF

1	$A + 0 = A$
---	-------------

Rule 1:

$$X = A + 0 = A$$

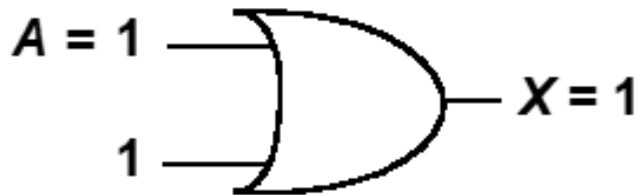
A *variable* OR-ed with **0** is always equal to the *variable*.

continue...

Rules of Boolean Algebra: PROOF

2	$A + 1 = 1$
---	-------------

Rule 2:

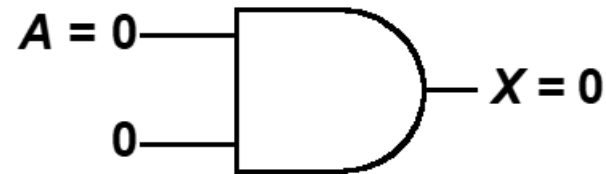
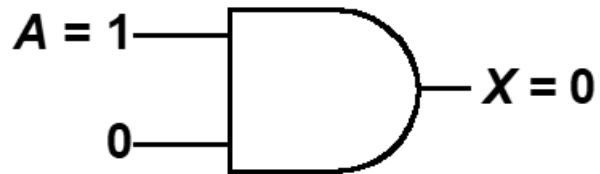


$$X = A + 1 = 1$$

A *variable* OR-ed with **1** is always equal to the **1**.

continue...

Rule 3:



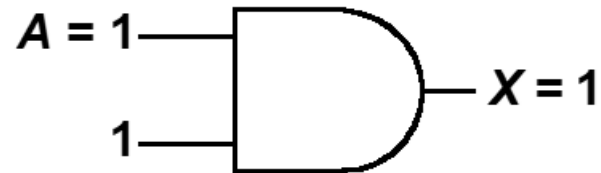
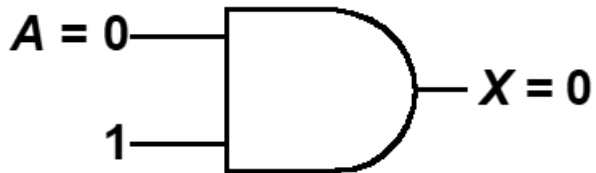
$$X = A \bullet 0 = 0$$

A variable AND-ed with 0 is always equal to 0.

continue...

4	$A \cdot 1 = A$
---	-----------------

Rule 4:

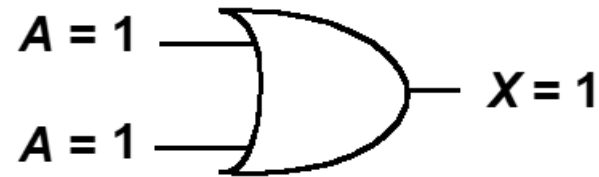
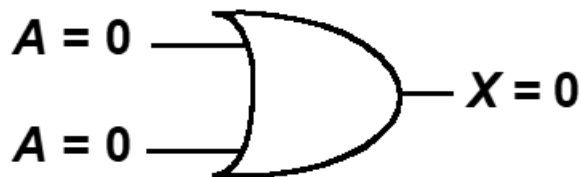


$$X = A \cdot 1 = A$$

A variable AND-ed with 1 is always equal to variable.

continue...

Rule 5:

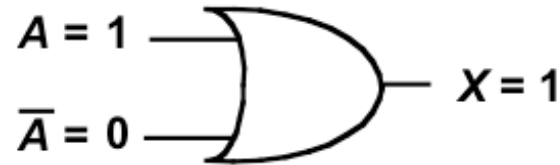


$$X = A + A = A$$

A variable OR-ed with itself is always equal to the variable.

continue...

Rule 6:

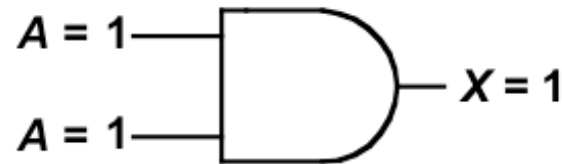
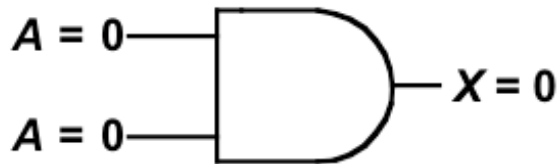


$$X = A + \bar{A} = 1$$

A *variable* OR-ed with its *complement* is always equal to 1.

continue...

Rule 7:

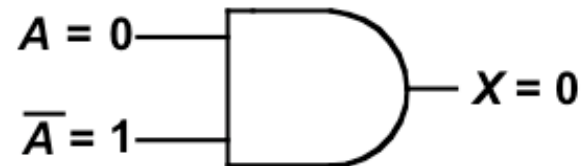
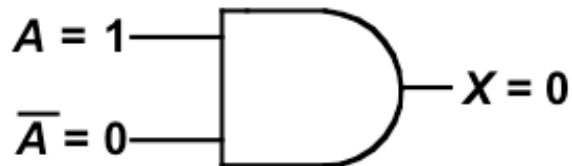


$$X = A \bullet A = A$$

A variable AND-ed with itself is always equal to the variable.

continue...

Rule 8:

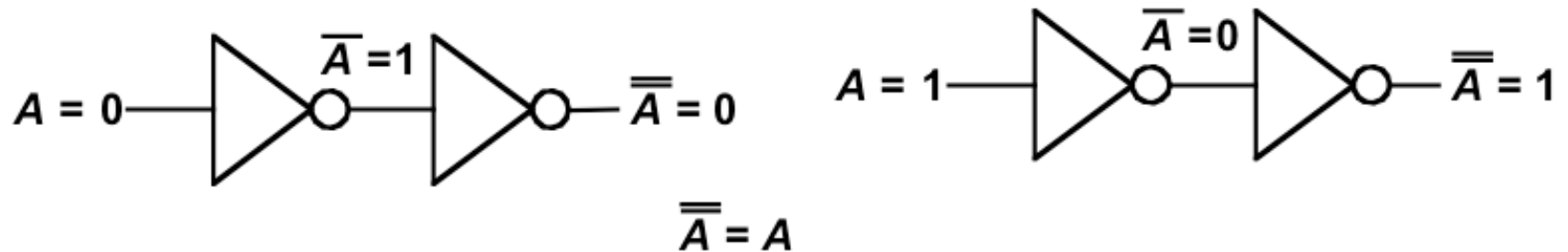


$$X = A \bullet \bar{A} = 0$$

A variable AND-ed with its complement is always equal to 0

continue...

Rule 9:



The *double complement* of a variable is always equal to the *variable*

continue...



Rules 10, 11, and 12 can be proven by using Boolean algebra laws.

10	$A + AB = A$
----	--------------

Rule 10:

$$\begin{aligned} A + AB &= A(1 + B) && \text{Factoring (distributive law)} \\ &= A \cdot 1 && \text{Rule 2: } (1 + B) = 1 \\ &= A && \text{Rule 4: } A \cdot 1 = A \end{aligned}$$

continue...



Rules 10, 11, and 12 can be proven by using Boolean algebra laws.

11	$A + \bar{A}B = A + B$
----	------------------------

Rule 11:

$$\begin{aligned} A + \bar{A}B &= (A + AB) + \bar{A}B \\ &= (AA + AB) + \bar{A}B \\ &= AA + AB + A\bar{A} + \bar{A}B \\ &= (A + \bar{A})(A + B) \\ &= 1 \cdot (A + B) \\ &= A + B \end{aligned}$$

Rule 10: $A = A + AB$

Rule 7: $A = AA$

Rule 8: adding $A\bar{A} = 0$

Factoring

Rule 6: $A + \bar{A} = 1$

Rule 4: drop the 1

continue...



Rules 10, 11, and 12 can be proven by using Boolean algebra laws.

12	$(A + B)(A + C) = A + BC$
----	---------------------------

Rule 12:

$$\begin{aligned}(A + B)(A + C) &= AA + AC + AB + BC \\&= A + AC + AB + BC \\&= A(1 + C) + AB + BC \\&= A \cdot 1 + AB + BC \\&= A(1 + B) + BC \\&= A \cdot 1 + BC \\&= A + BC\end{aligned}$$

Distributive law

Rule 7: $AA = A$

Factoring (distributive law)

Rule 2: $1 + C = 1$

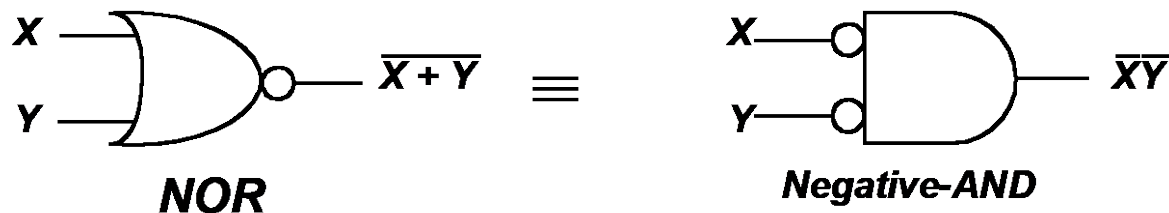
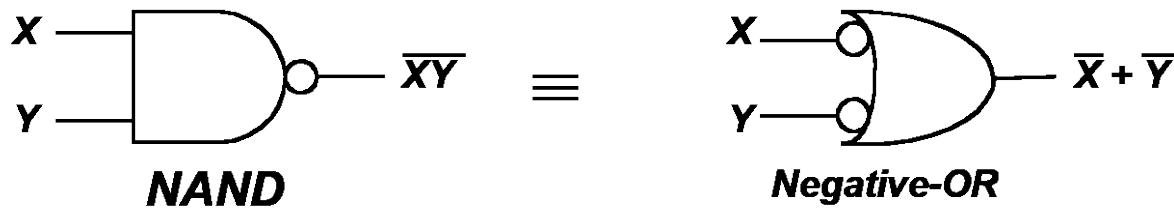
Factoring (distributive law)

Rule 2: $1 + B = 1$

Rule 4: $A \cdot 1 = A$

DeMorgan's Theorems

- To minimize the **variety** and **number of logic gates** IC.
- Provides mathematical verification for:
 - NAND \equiv negative-OR
 - NOR \equiv negative-AND



- **DM theorem 1:**

- The complement of a product of variables is equal to the sum of the complements of the variables

$$\overline{XY} = \overline{X} + \overline{Y}$$

- **DM theorem 2:**

- The complement of a sum of variables is equal to the product of the complements of the variables

$$\overline{X + Y} = \overline{X} \overline{Y}$$



DeMorgan's Theorems Application

Example of applying DeMorgan's Theorems.

DM theorem 1:

The complement of a product of variables is equal to the sum of the complements of the variables

$$\overline{XYZ} = \bar{X} + \bar{Y} + \bar{Z}$$

$$\overline{WXYZ} = \bar{W} + \bar{X} + \bar{Y} + \bar{Z}$$

DM theorem 2:

The complement of a sum of variables is equal to the product of the complements of the variables

DeMorgan's Theorem II

$$\overline{X + Y + Z} = \bar{X}\bar{Y}\bar{Z}$$

$$\overline{W + X + Y + Z} = \bar{W}\bar{X}\bar{Y}\bar{Z}$$



Truth Table Verification

A	B	$A \cdot B$	$(A \cdot B)'$	A'	B'	$(A' + B')$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

- Both $(A \cdot B)'$ and $(A' + B')$ produce identical outputs.

Example of applying DeMorgan's Theorems.

$$\overline{(AB + C)(A + BC)} = \overline{(AB + C)} + \overline{(A + BC)} \quad (\text{Theorem I})$$

$$\overline{(AB + C)} + \overline{(A + BC)} = (\overline{AB})\overline{C} + \overline{A}(\overline{BC}) \quad (\text{Theorem II})$$

$$= (\overline{A} + \overline{B})\overline{C} + \overline{A}(\overline{B} + \overline{C}) \quad (\text{Theorem I})$$

(Theorem I)

$$\overline{XY} = \overline{X} + \overline{Y}$$

(Theorem II)

$$\overline{X + Y} = \overline{X}\overline{Y}$$

Excercise

Self-Test:

Prove that \overline{AB} is **equal** or **not equal** with $\overline{A}\overline{B}$ by using the truth table.

Answer:

A	B	\overline{A}	\overline{B}	AB	$\overline{A}\overline{B}$	\overline{AB}



Image Upload



Exercise 4a.2:

Apply DeMorgan's theorems to each of the following expressions:

(a) $\overline{(A + B + C)D}$

(b) $\overline{ABC + DEF}$

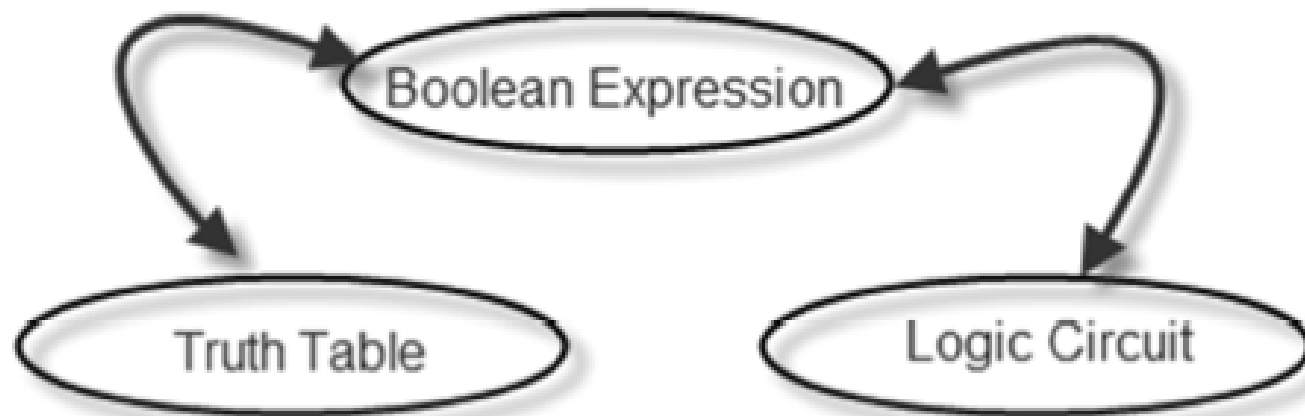
(c) $\overline{A\bar{B} + \bar{C}D + EF}$

Combinational logic is a type of digital circuit where the **output depends only on the present input values**, not on any past input. There's no memory element inside.

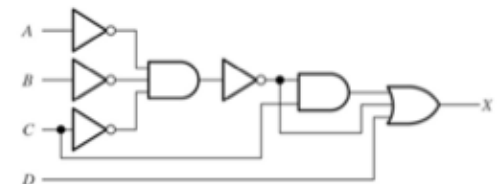
Module 4

Combinational Logic Representation

$$\overline{(AB + C)(A + BC)}$$



A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

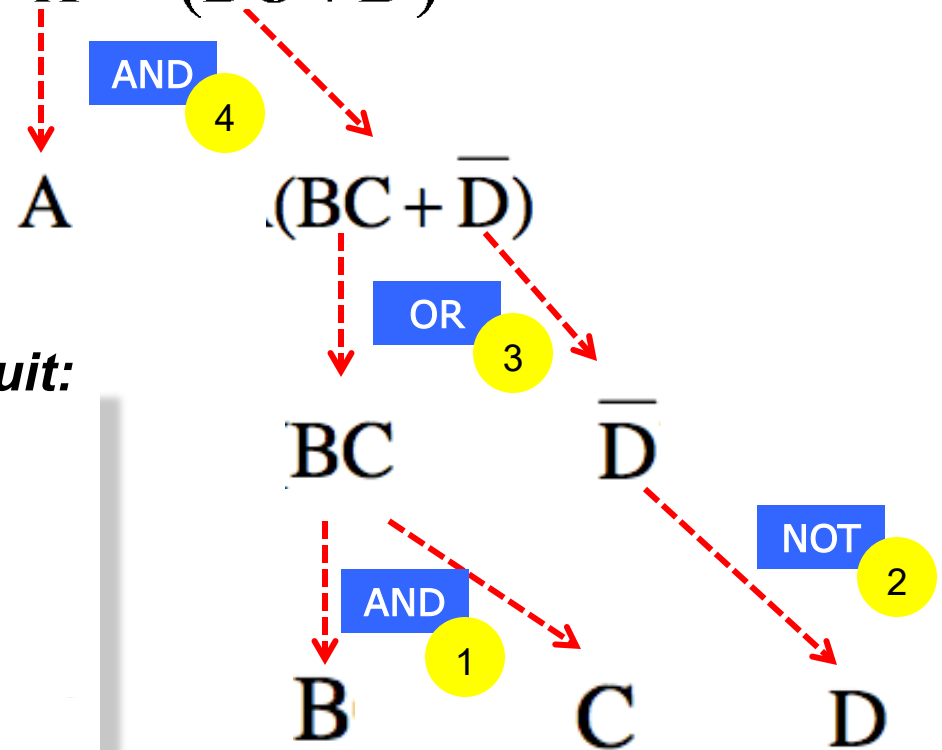




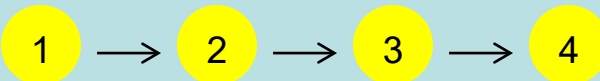
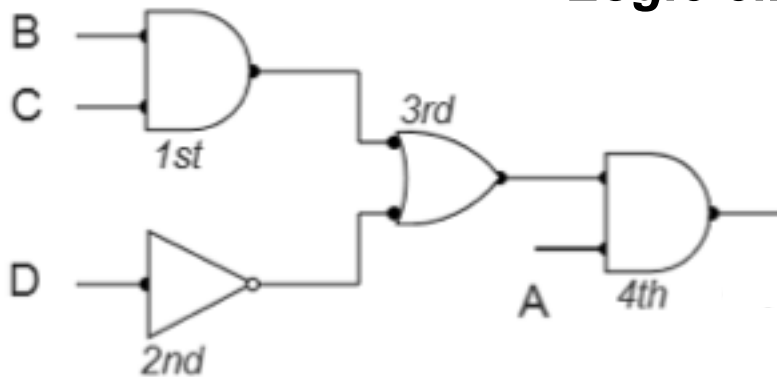
From Boolean Expression to Logic Circuit

- Treat a Boolean expression as if you try to solve a normal mathematical equation
- Just like a normal mathematic we have the precedence of operation
 - start with () then multiplication (AND) then addition (OR)
 - Therefore to draw the circuit following the same sequence

Example: $X = A(BC + \bar{D}) = A \quad (BC + \bar{D})$



Logic circuit:





Exercise 4a.3:

Draw the logic circuit represented by each expression:

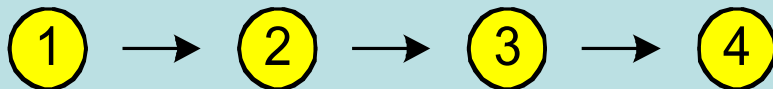
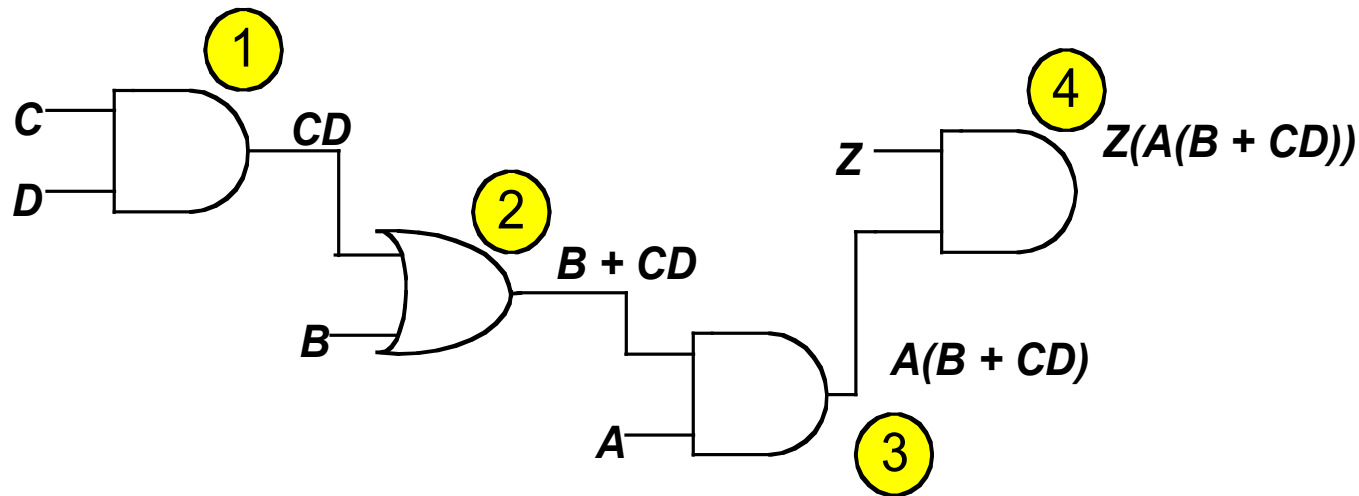
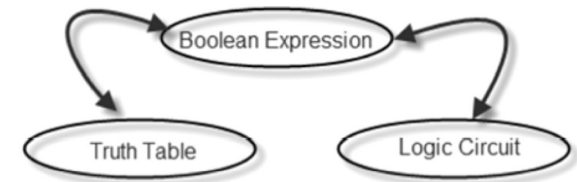
(i) $A\bar{B} + \bar{A}B$

(ii) $AB + \bar{A}\bar{B} + \bar{A}BC$

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

Logic circuit to Boolean expression

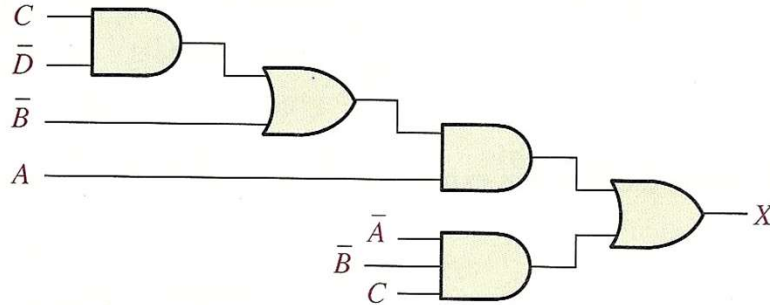
- To derive the Boolean expression for a given circuit, follow **left-to-right** rule.
 - Begin from the left-most inputs and work towards the last.



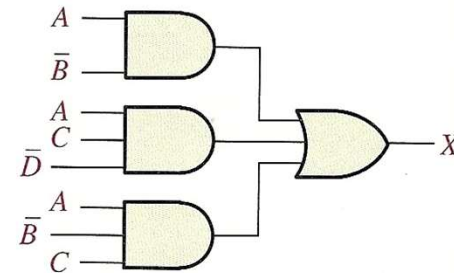


Exercise 4a.4:

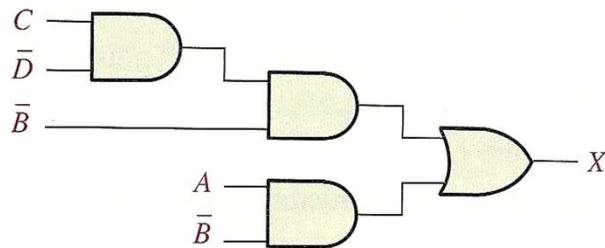
Determine which of the logic circuits are equivalent.



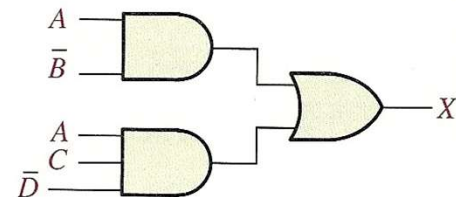
(a)



(b)

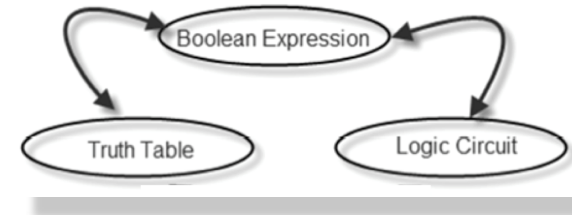


(c)



(d)

Boolean expression to truth table



- A truth table shows the output for all possible input values.
- From a Boolean expression, a truth table can be developed.

x = number of input variables

Possible combinations of values, $n = 2^x$

- Example:

$$F = A + B \rightarrow x = 2; n = 2^2 = 4$$



Steps in construction a truth table

- **Step 1:** Identify **x** and **n** from the Boolean exp.
- **Step 2:** Find the values of the variables that make the expression equal to **1**.
(Hint: use the rules for Boolean addition and multiplication)
- **Step 3:** List in a table
 - all the **n** combinations of 1s and 0s (**inputs**)
 - the values of variables from step 2 (**outputs=1**)
 - all the **other output** values will be **0**



Example: $F = A + B$

1. $x = ?$
 $n = 2^x ?$

2. Combination of inputs $\rightarrow F = 1$

$$A + B = 1$$

3. Fill in the table:

$x=2, n=4$

INPUT		OUTPUT
A	B	F
0	0	
0	1	
1	0	
1	1	

Example: $A(B + CD)$

Step 1: $x = 4$; $n = 2^x = 2^4 = 16$

Step 2:

$$A(B + CD) = 1 \cdot 1 = 1 \rightarrow A = 1$$

What makes $B + CD = 1$?

$$B + CD = 1 + 0 = 1$$

$$B + CD = 0 + 1 = 1$$

$$B + CD = 1 + 1 = 1 \rightarrow B = 1 \text{ or } 0$$

What makes $CD = 1$?

$$CD = 1 \cdot 1 = 1 \rightarrow C = 1; D = 1$$

Therefore, the output of $A(B + CD)$ will be 1 if

$$A = 1, B = 1, C = 0/1, D = 1/0$$

$$A = 1, B = 0, C = 1, D = 1$$

Step 3 : Fill in the table with results from Step 2

INPUTS				OUTPUT
A	B	C	D	$A(B + CD)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

- Tips on ‘**table-making**’ :
 - For n possible combinations, the input part of the table will register the binary value of 0 to $n-1$.
 - (e.g. $n=16$; 0 to 15)

- Remember the sequence

	2^4	2^2	2^1	2^0	
	8	4	2	1	
0	0	0	0	0	(0)
0	0	0	0	1	(1)
0	0	0	1	0	(2)
0	0	0	1	1	(3)

From Truth Table to Logic Circuit

Step 1: Get the product term from HIGH outputs

1

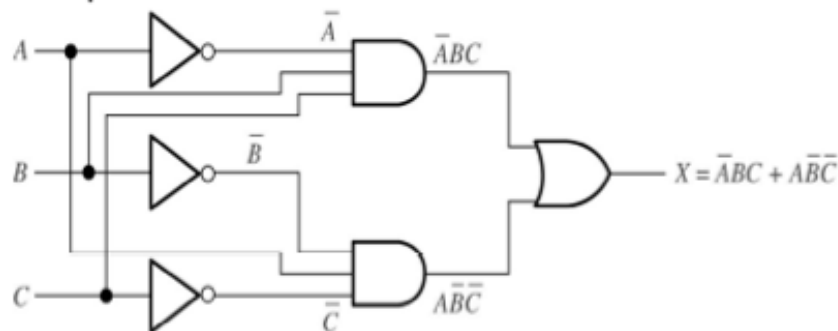
Step 2: From the product term get the expression

- This is done by OR-ing the product terms

$$X = \bar{A}BC + A\bar{B}\bar{C}$$

INPUTS			OUTPUT	PRODUCT TERM
A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A}BC$
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	0	
1	1	1	0	

Step 3: Implement the circuit



Example:

Given the truth table below, produce the logic circuit (Module v5: page 116).

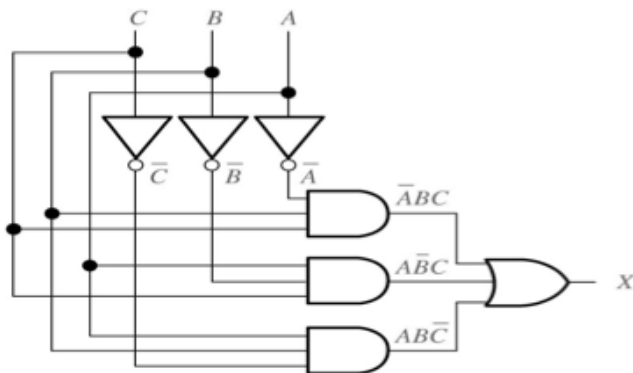
INPUTS			OUTPUT
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



INPUTS			OUTPUT	PRODUCT TERM
A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A}BC$
0	0	0	0	
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	0	



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

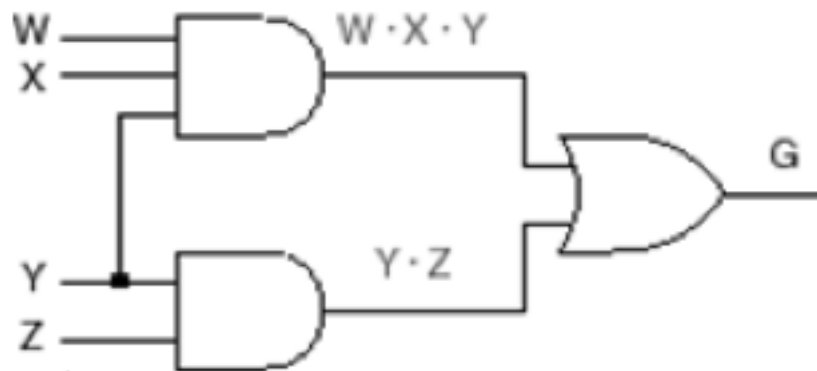




Logic Circuit to Truth Table

- Requires 2 steps
 1. From circuit derive the Boolean expression of the output
 2. From Boolean expression produce a Truth Table

Both step has been mentioned



Example:

Produce a truth table from the above circuit.

Step1: derive the Boolean expression of the output , $G = WXY + YZ$

Step 2: from the expression produce the truth table

G will be 1 if either term = 1

term 1 which is WXY will only be 1 if $W = 1$, $X = 1$ and $Y = 1$

term2 which is YZ will only be 1 if $Y = 1$ and $Z = 1$

W	X	Y	Z	G
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1