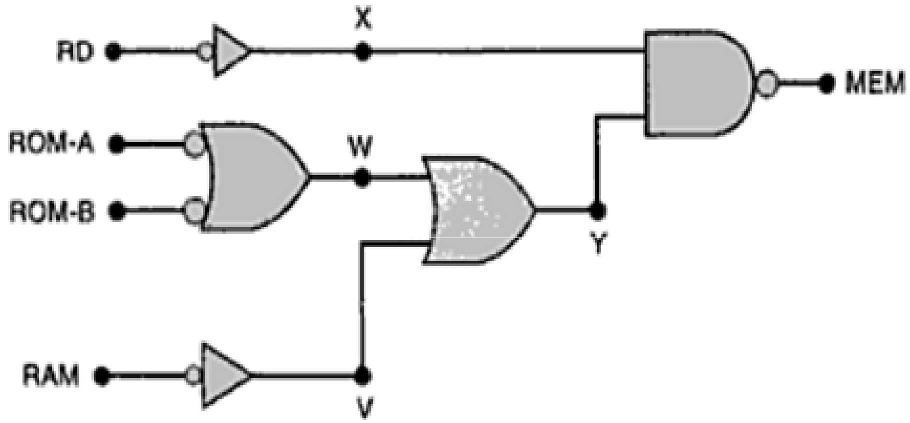
Digital Electronics and Microprocessors

Class 3

CHHAYADEVI BHAMARE

Microcomputer Application:-The logic circuit in fig generates an output MEM, that is used to activate a Memory IC in a particular Microcomputer. Determine the I/P conditions necessary to activate MEM

1.MEM is active LOW, and it will go LOW only when x and y are HIGH
2.X will be HIGH only when RD=0.
3.Y will be high when either W or V is HIGH
4. W will be high when either ROM-A or ROM-B=0
5. V will be HIGH when RAM=0



6. Putting this all together, MEM will go LOW only when RD=0 and at least one of the three inputs ROM-A,ROM-B, or RAM is low

Implementing (synthesizing) Circuits From Boolean Expressions

- It is important to be able to draw a logic circuit from a Boolean expression.
- The expression

$$x = A \cdot B \cdot C$$

could be drawn as a three input AND gate.

- A more complex example such as

$$y = AC + B\overline{C} + \overline{A}BC$$

could be drawn as two 2-input AND gates and one 3-input AND gate feeding into a 3-input OR gate. Two of the AND gates have inverted inputs.

Reference Example 3-7

NOR Gates and NAND Gates

- Combine basic AND, OR, and NOT operations.
- The NOR gate is an inverted OR gate. An inversion “bubble” is placed at the output of the OR gate.
- The Boolean expression is, $x = \overline{A + B}$

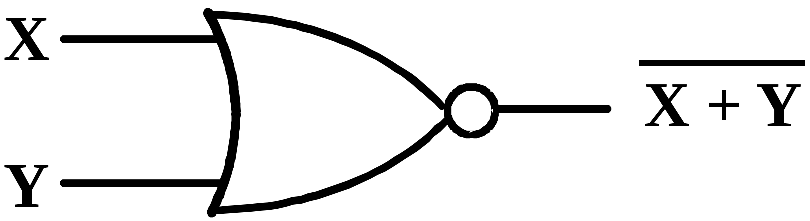
NOR Gates and NAND Gates

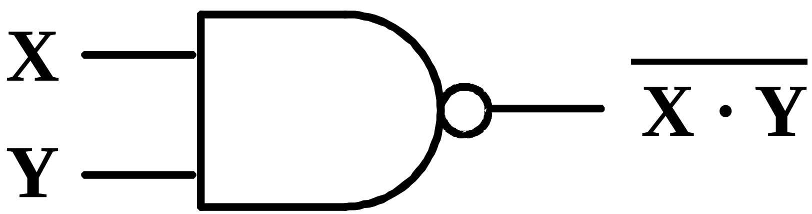
- The NAND gate is an inverted AND gate. An inversion “bubble” is placed at the output of the AND gate.
- The Boolean expression is

$$x = \overline{AB}$$

NOR Gates and NAND Gates

- The output of NAND and NOR gates may be found by simply determining the output of an AND or OR gate and inverting it.





X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Example:-

- Implement the logic circuit that has the expression x= $\overline{AB\bullet\left(\overline{C+D}\right)}$
- Refer Examples 3-8 to3-10 from T1

Boolean Theorems

Single Variable theorems **(Refer Fig3-25 on page 79 in T1)**

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

$$x \cdot x = x$$

$$x \cdot \overline{x} = 0$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + \overline{x} = 1$$

$$x + 0 = x$$

Boolean Theorems

- Multivariable theorems:
- Understanding all of the Boolean theorems will be useful in reducing expressions to their simplest form.

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

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

$$x(yz) = (xy)z = xyz$$

$$x(y + z) = xy + xz$$

$$(w + x)(y + z) = wy + xy + wz + xz$$

$$x + xy = x$$

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

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

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

Examples:- logic simplification using Boolean theorems

- Refer Examples 3-13 to 3-15 in T1

DeMorgan's Theorems

- When the OR sum of two variables is inverted, it is equivalent to inverting each variable individually and ANDing them.

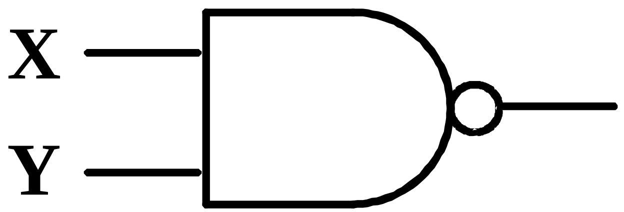
  $(x+y)' = x' . y'$

- When the AND product of two variables is inverted, it is equivalent to inverting each variable individually and ORing them.

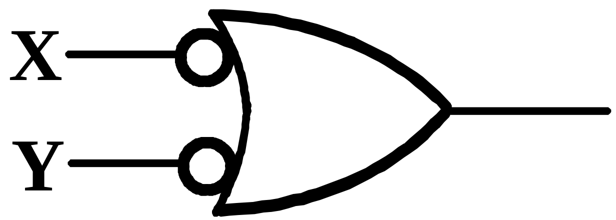
  $(x . y)' = x' + y'$

Implications of DeMorgan's Theorems

- A NAND gate is equivalent to an OR gate with inverted inputs.



$$\overline{X \cdot Y}$$

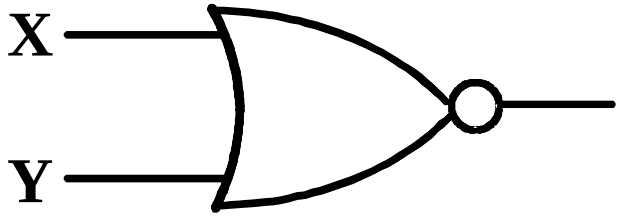


$$\overline{X} + \overline{Y} = \overline{X \cdot Y} = \text{NAND}$$

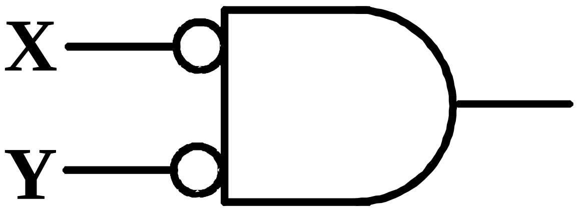
Invert-OR = NAND

Implications of DeMorgan's Theorems

- A NOR gate is equivalent to an AND gate with inverted inputs.



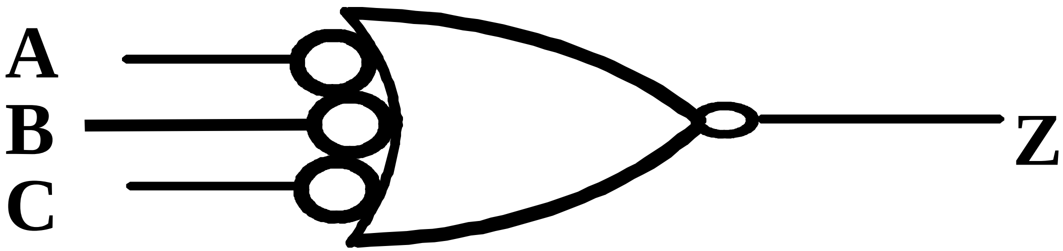
$\overline{X + Y}$



$\overline{X} \cdot \overline{Y} = \overline{X + Y} = \text{NOR}$

NOR=Invert-AND

Example



$$Z = \overline{\bar{A} + \bar{B} + \bar{C}} = ABC$$

Examples:- Simplify the following Boolean expressions using the rules of Boolean algebra

1. $W=(PQR+P'Q')(S+T)+(P'+Q')(S+T)+(S+T)$
2. $Y=(A+B)'CD+(A+B)'$
3. $H=KL'+K$

Solution:-

1. Let x=(S+T), let y=(P'+Q'):
   since x+xy=x, (P'+Q')(S+T)+(S+T)=(S+T)
   W=(PQR+P'Q')(S+T)+(S+T) = (S+T)

2. work for you
3. Work for you

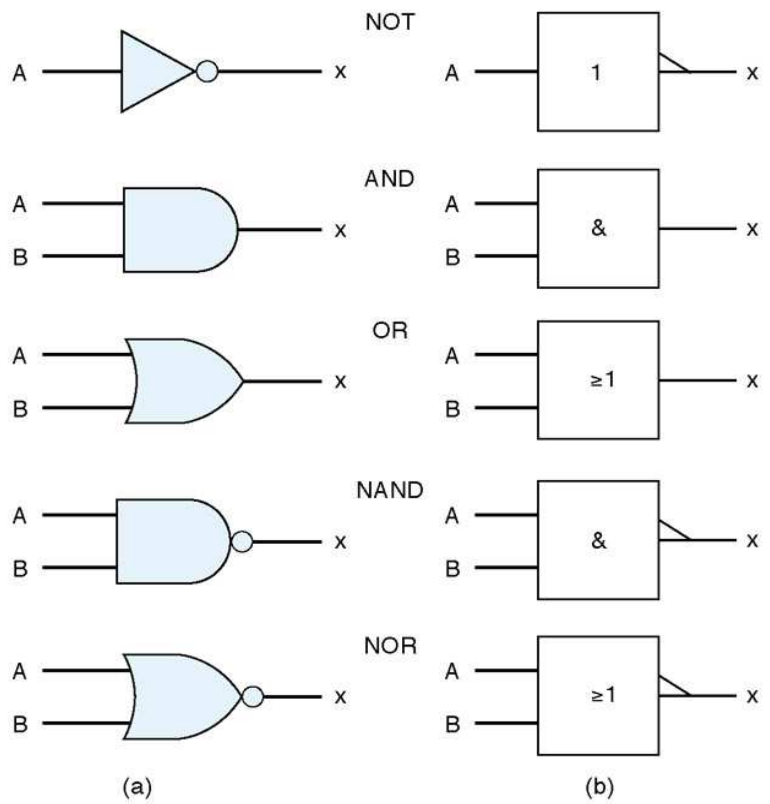
Universality of NAND and NOR Gates

- NAND or NOR gates can be used to create the three basic logic expressions (OR, AND, and INVERT)
- This characteristic provides flexibility and is very useful in logic circuit design.

- <u>Examples based on this will be covered in Tut2 (refer fig 3-29 and 2-30 before coming to tut2)</u>

IEEE/ANSI Standard Logic Symbols

- Compare the IEEE/ANSI symbols to traditional symbols.
- Rectangular symbols represent logic gates and circuits.
- A small triangle replaces the inversion bubble.
- These symbols are not widely accepted but may appear in some schematics.



Summary of Methods to Describe Logic Circuits

- The three basic logic functions are AND, OR, and NOT.
- Logic functions allow us to represent a decision process.

  Examples of decision process

  - If it is raining OR it looks like rain I will take an umbrella.
  - If I get paid AND I go to the bank I will have money to spend.
- Different methods to describe Logic circuits
  - Boolean Expression
  - Schematic diagram
  - Truth table
  - Timing diagram

Example:- Following English expression describing the way logic circuit needs to operate in order to drive a seatbelt warning indicator in a car

- If the driver is present AND the Driver in NOT buckled up AND the ignition switch is on, THEN turn on the warning light.
  - Describe the circuit using Boolean algebra, schematic diagram with logic symbols, truth table, and timing diagram

Number of Variables? **3**

Driver present
Buckled up
Ignition on

Example continued (refer page 99 of T1)

1. Boolean expression

$$\text{Warning light} = \text{driver present} \cdot \overline{\text{buckled up}} \cdot \text{ignition on}$$

2. Schematic diagram
3. Truth table

Combinational Logic (Chapter 4-T1)

- Basic logic gate functions will be combined in *combinational* logic circuits.
- Simplification of logic circuits will be done using Boolean algebra and a mapping technique.
- Troubleshooting of combinational circuits will be introduced.

Sum-of-Products Form

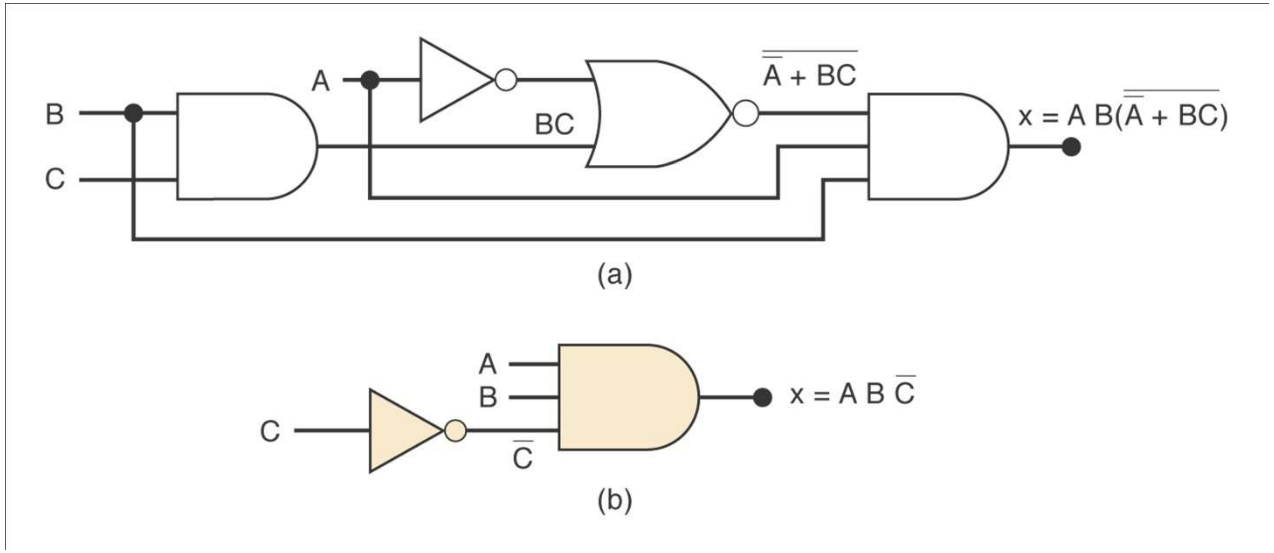
- A Sum-of-products (SOP) expression will appear as two or more AND terms ORed together.

$$X = ABC + \overline{A}B\overline{C}$$

$$X = AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$$

Simplifying Logic Circuits

- The circuits below both provide the same output, but the lower one is clearly less complex.



- We will study simplifying logic circuits using Boolean algebra and Karnaugh mapping

Algebraic Simplification

- Place the expression in SOP form by applying DeMorgan’s theorems and multiplying terms.
- Check the SOP form for common factors and perform factoring where possible.
- Note that this process may involve some trial and error to obtain the simplest result.

Reference Examples for algebraic simplification

- Examples 4.1 to 4.6 from T1

Designing Combinational Logic Circuits

- To solve any logic design problem:
  - Interpret the problem and set up its truth table.
  - Write the AND (product) term for each case where the output equals 1.
  - Combine the terms in SOP form.
  - Simplify the output expression if possible.
  - Implement the circuit for the final, simplified expression.

Example application:- Majority Circuit

A logic circuit having 3 inputs, A, B, C will have its output HIGH only when a majority of the inputs are HIGH.

Step 1 **Set up the truth table**

Step 2 **Write the AND term for each case where the output is a 1.**

A	B	C	x	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \bar{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\bar{B}C$
1	1	0	1	$\rightarrow AB\bar{C}$
1	1	1	1	$\rightarrow ABC$

Step 3 **Write the SOP form the output**

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

Step 4 **Simplify the output expression (ref. p. 119 of T1)**

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

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

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

$$= BC + AC + AB$$

<u>Step 5</u> Implement the circuit