



A A B University

Faculty of Computer Sciences

Introduction to Digital Technologies and Circuits

Week 5:

Digital Logic Circuits

Combinational Circuits

Asst. Prof. Dr. **Mentor Hamiti**
mentor.hamiti@universitetiaab.com



- Boolean Algebra
- Logic (Boolean) Functions
- Representations of Boolean Functions
 - Switching Circuits
 - Truth Tables
 - Timing Diagrams
 - Venn Diagrams
 - K-Diagrams



- Minimization of Logic Functions
- Combinational Logic Circuits
- Logic Gates
 - Basic Logic Gates
 - Universal Logic Gates
 - Special (exclusive) Logic Gates



- Minimization of Logic Functions:
 - Algebraic Minimization
 - Graphic Minimization
 - **Table Minimization**



- Example 1:

- *Simplify the following Boolean (Logic) Functions:*

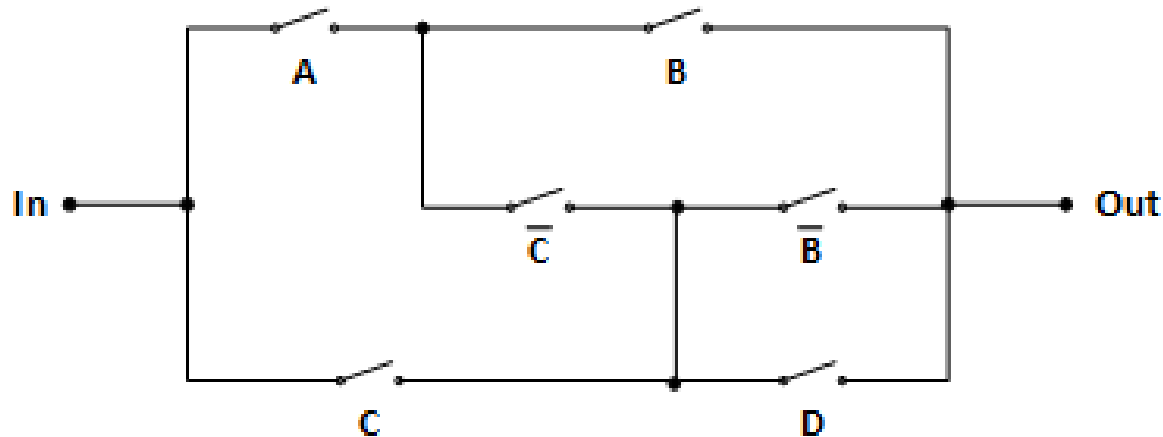
- a) $F_1 = A + ABC + AB + A'BC$

- b) $F_2 = (A + C')(B + C')(A + B')$



■ Example 2:

- For the given switching circuit:
 - a) Find the corresponding logic function and
 - b) Simplify the algebraic expression
 - c) Draw the equivalent switching circuit





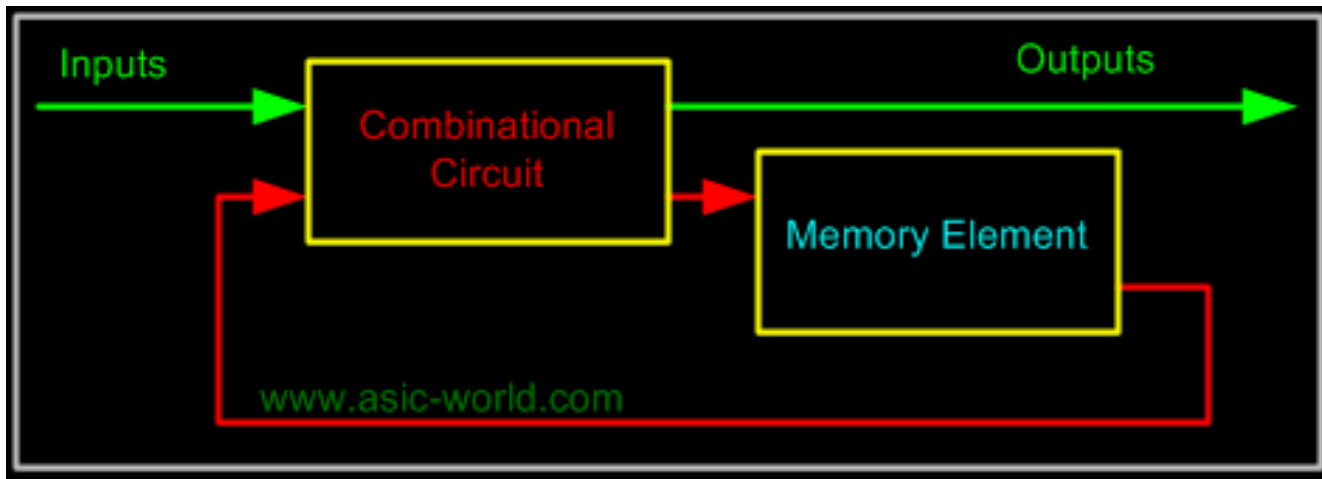
■ Example 3:

- *For the given timing diagram, use the appropriate K-Diagram to find the simplified algebraic expression!*





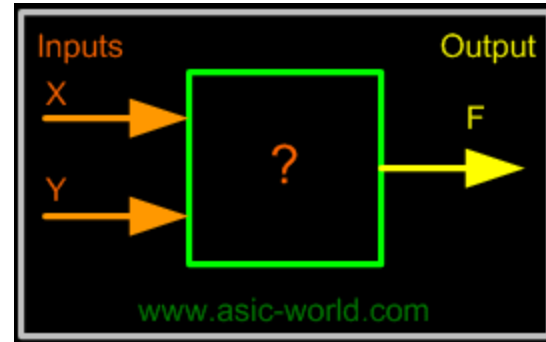
- There are two types of Digital Logic Circuits:
 - Combinational Logic Circuits
 - Sequential Logic Circuits



- Combinational logic output depends on the inputs levels, whereas sequential logic output depends on stored levels and also the input levels.



- A Combinational Logic Circuit can be expressed as a logic design and implemented as a collection of individual connected **Logic Gates**.



- A fixed logic system has two possible choices for representing true and false:
 - **Positive Logic**
In a positive logic system, a high voltage is used to represent logical true (1), and a low voltage for a logical false (0).
 - **Negative Logic**
In a negative logic system, a low voltage is used to represent logical true (1), and a high voltage for a logical false (0).
- In positive logic circuits it is normal to use +5V for true and 0V for false.



- Logic gates are the building blocks of any digital circuit.
- **Logic gates** are electronic circuits/devices which makes the logical decisions. They have one or more inputs and only one output. The output is active only for certain input combinations. Logic gates are also called switches.
- Logic gates can be categorized into there groups:
 - Basic Logic Gates
 - Universal Logic Gates
 - Special (exclusive) Logic Gates



- Basic Logic Gates:

- AND Gate $F = A \cdot B$

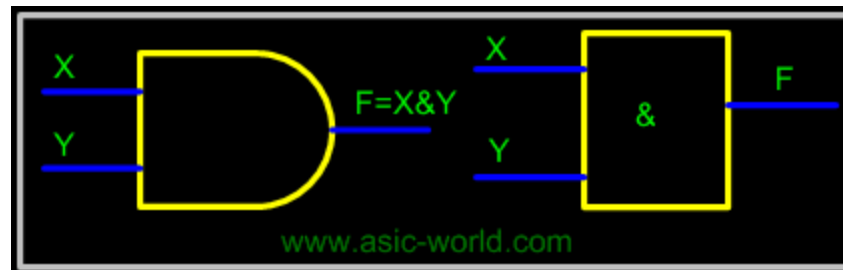
- OR Gate $F = A + B$

- NOT Gate $F = \bar{A}$



- The AND gate performs logical multiplication, commonly known as AND function.
- The AND gate has two or more inputs and single output.
- The output of AND gate is HIGH only when all its inputs are HIGH (i.e. even if one input is LOW, Output will be LOW).
- If X and Y are two inputs, then output F can be represented mathematically as $F = X.Y$, Here dot (.) denotes the AND operation.

Symbol:



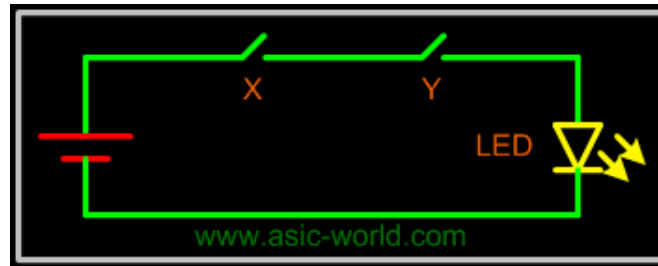
AND Gate



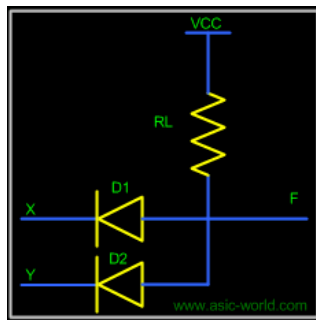
- Truth Table:

X	Y	F=(X.Y)
0	0	0
0	1	0
1	0	0
1	1	1

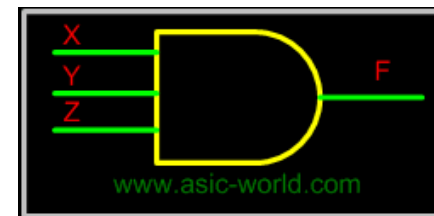
- Switch Representation:



- Circuit:



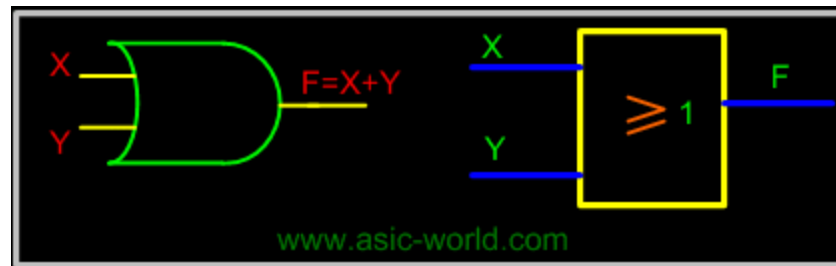
AND Gate with 3 inputs:





- The OR gate performs logical addition, commonly known as OR function.
- The OR gate has two or more inputs and single output.
- The output of OR gate is HIGH only when any one of its inputs are HIGH (i.e. even if one input is HIGH, Output will be HIGH).
- If X and Y are two inputs, then output F can be represented mathematically as $F = X+Y$. Here plus sign (+) denotes the OR operation.

Symbol:



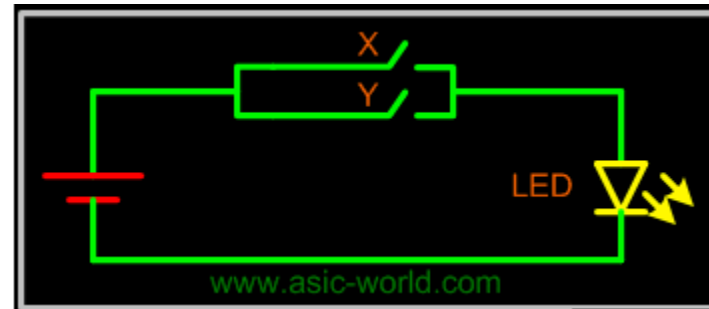
OR Gate



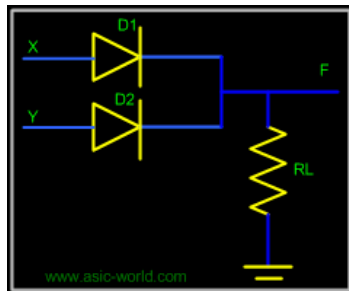
- Truth Table:

X	Y	F=(X+Y)
0	0	0
0	1	1
1	0	1
1	1	1

- Switch Representation:



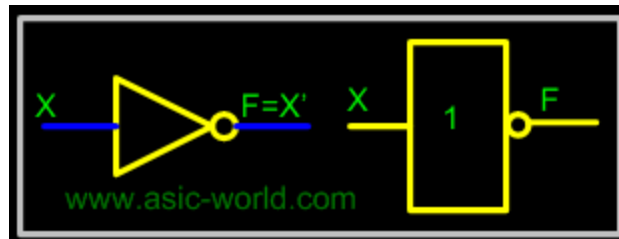
- Circuit:





- The NOT gate performs the basic logical function called inversion or complementation. NOT gate is also called inverter.
- The purpose of this gate is to convert one logic level into the opposite logic level. It has one input and one output. When a HIGH level is applied to an inverter, a LOW level appears on its output and vice versa.
- If X is the input, then output F can be represented mathematically as $F = X'$ or $F = \bar{X}$. There are a couple of other ways to represent inversion!

Symbol:



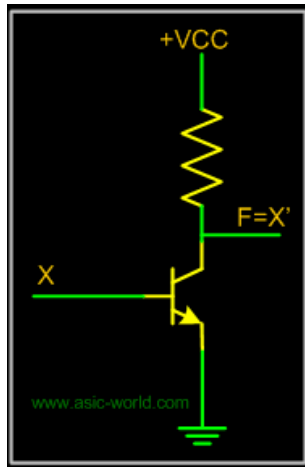
OR Gate



- Truth Table:

X	Y=X'
0	1
1	0

- Circuit:



Universal Logic Gates



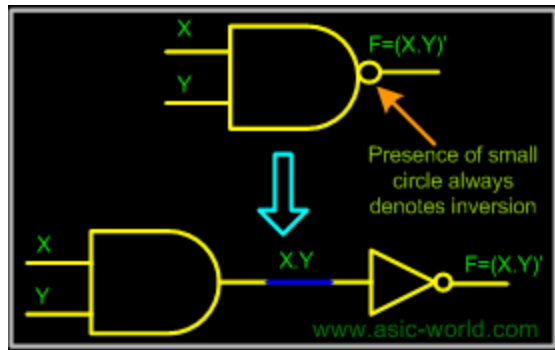
- Universal Logic Gates:

- NAND Gate $F = \overline{A \cdot B}$

- NOR Gate $F = \overline{A + B}$

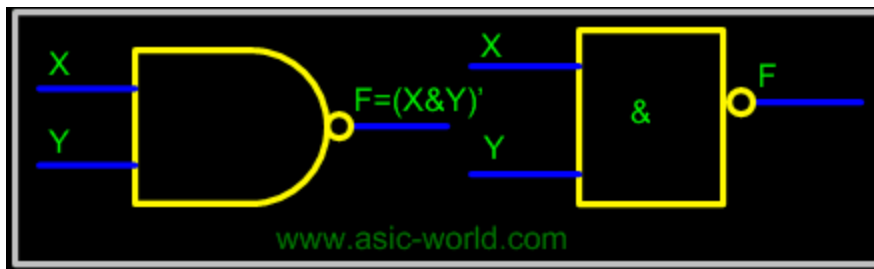


- NAND gate is a cascade of AND gate and NOT gate:



- It has two or more inputs and only one output. The output of NAND gate is HIGH when any one of its input is LOW (i.e. even if one input is LOW, Output will be HIGH).

Symbol:



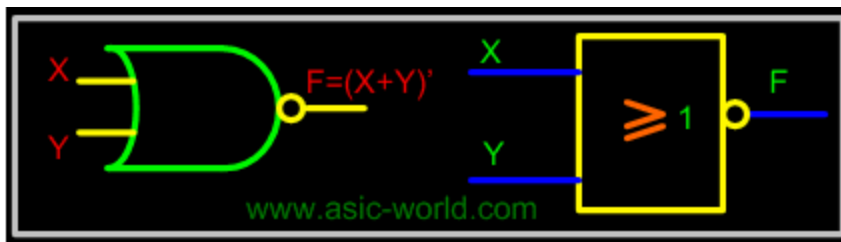
Truth table:

X	Y	F=(X.Y)'
0	0	1
0	1	1
1	0	1
1	1	0



- NOR gate is a cascade of OR gate and NOT gate.
- It has two or more inputs and only one output.
- The output of NOR gate is HIGH when any all its inputs are LOW (i.e. even if one input is HIGH, output will be LOW).

Symbol:



Truth table:

X	Y	$F=(X+Y)'$
0	0	1
0	1	0
1	0	0
1	1	0

Special (exclusive) Logic Gates



- Exclusive Logic Gates:

- XOR Gate $F = A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$

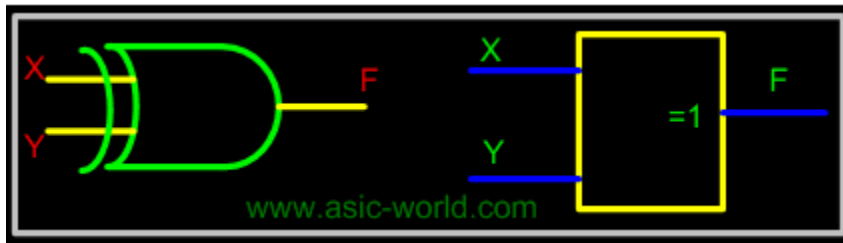
- XNOR Gate $F = A \otimes B = A \cdot B + \bar{A} \cdot \bar{B}$

XOR Gate



- An Exclusive-OR (XOR) gate is a gate with two or more inputs and one output.
- The output of a two-input XOR gate assumes a HIGH state if one and only one input assumes a HIGH state. This is equivalent to saying that the output is HIGH if either input X or input Y is HIGH exclusively, and LOW when both are 1 or 0 simultaneously.

Symbol:



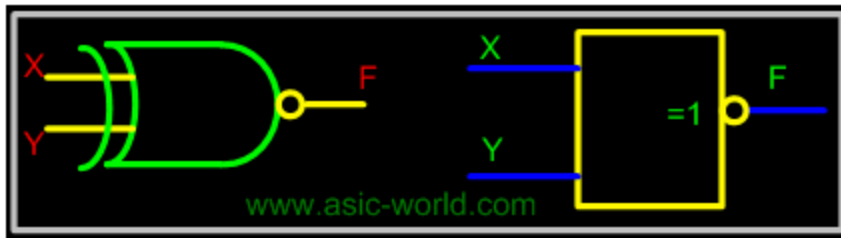
Truth table:

X	Y	F=(X Y)
0	0	0
0	1	1
1	0	1
1	1	0



- An Exclusive-NOR (XNOR) gate is a gate with two or more inputs and one output.
- The output of a two-input XNOR gate assumes a HIGH state if all the inputs assume the same state. This is equivalent to saying that the output is HIGH if both input X and input Y are HIGH exclusively or the same as input X and input Y are LOW exclusively, and LOW when both are not the same.

Symbol:



Truth table:

X	Y	$F=(X \oplus Y)'$
0	0	1
0	1	0
1	0	0
1	1	1



- Example 4:
 - Synthesis of Combinational Logic Circuit
 - Using Basic Gates
 - Using Universal Gates

$$F = AB + A'$$



■ Example 5:

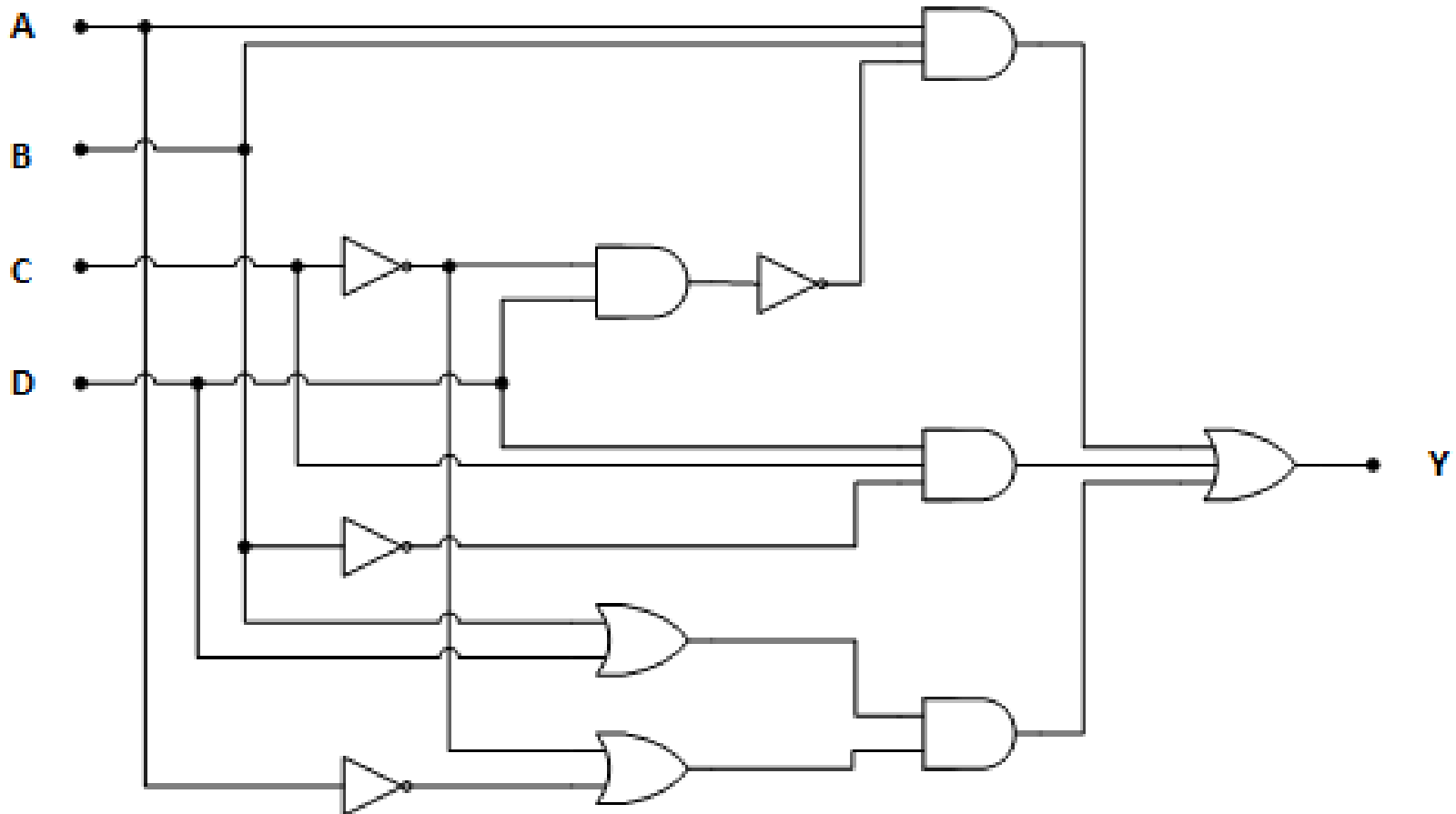
- *Logic Function is given by the following Truth Table. Design the Combinational Logic Circuit, with the use of:*
 - a) Basic Logic Gates*
 - b) Universal Logic Gates*

<i>i</i>	<i>F</i>
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	0



■ Example 6:

- *Review the possibility to minimize the following Combinational Logic Circuit?!*





- Questions?!

