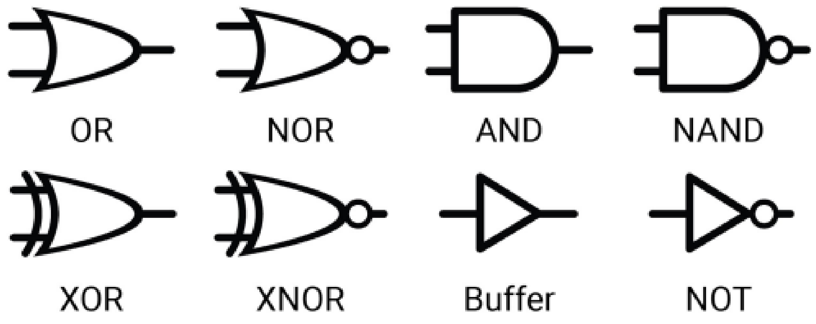
Computer System Architecture



Day 02 : Logic Gates

Supun Chandana Gajendrasinghe
Department of Network and Security
supun.g@nsbm.ac.lk

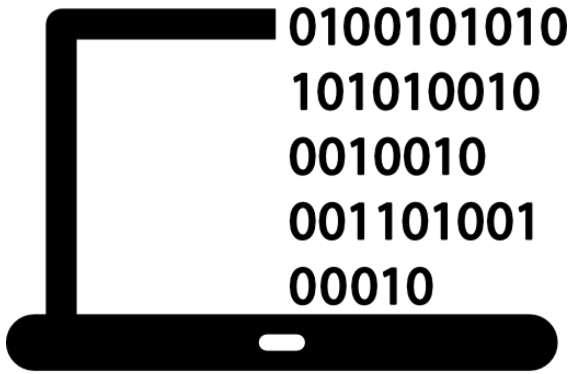
Outline

- Digital Electronics
- Logic Circuits
- Truth Table
- Basic Logic Gates
- Special Purpose Gates
- Universal Logic Gates
- Summary

Digital Electronics

- The electronics inside a modern computer are digital. **Digital electronics operate with only two voltage levels**.

- The fact that computers are digital is also a key reason they **use binary numbers**, since a binary system matches the underlying abstraction inherent in the electronics.

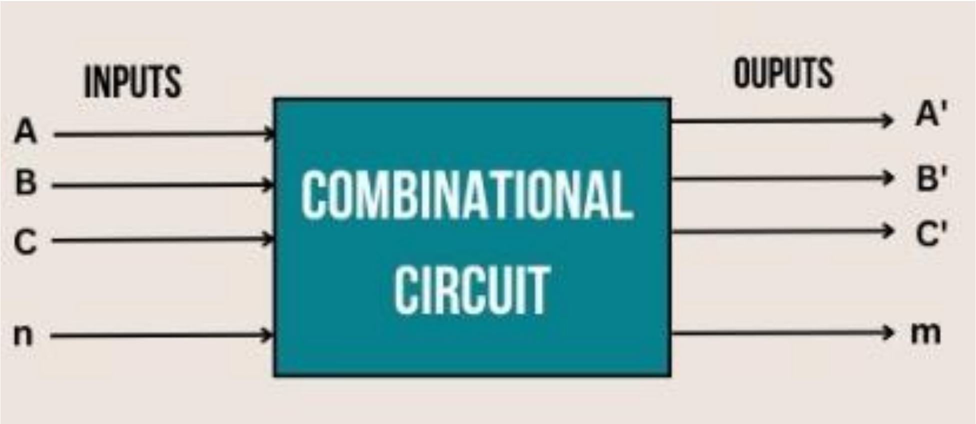


Logic Circuits

Combinational Logic Circuit

- Blocks **without memory** are called combinational

- The output of a combinational block depends only on the current input.

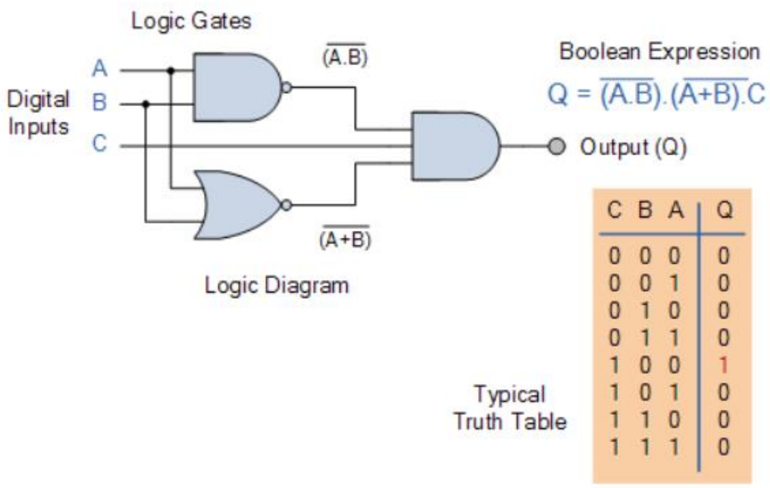


Eg: Adders, Multiplexers, Decoders

Logic Circuits Cont`d…

Combinational Logic Circuit

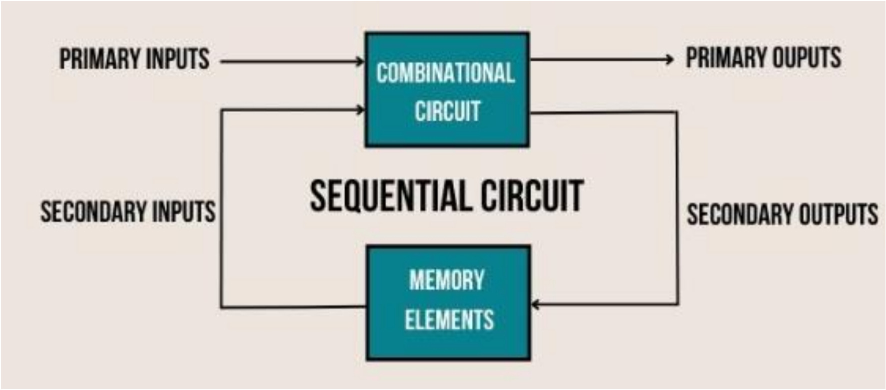
- There are 3 main ways of specifying the function of a combinational logic circuit
    I. Boolean Expression
    II. Truth Table
    III. Logic Diagram



Logic Circuits Cont`d...

Sequential Logic Circuit

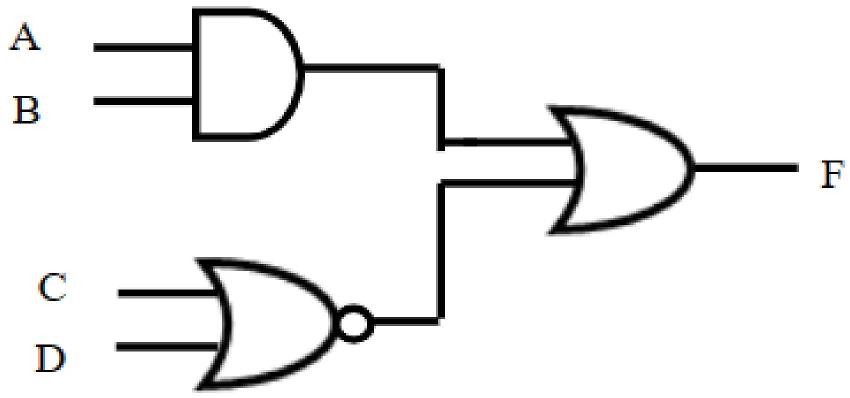
- Blocks **with memory**, the outputs can depend on both the inputs and the value stored in memory (Previous Input).



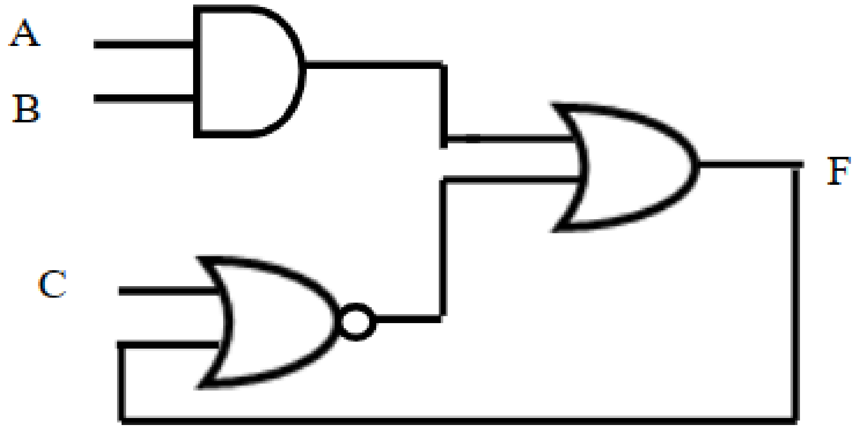
Eg: Flip-flops, Counters, Shift Registers.

Logic Circuits Cont`d…

Combinational Logic Circuit vs Sequential Logic Circuit



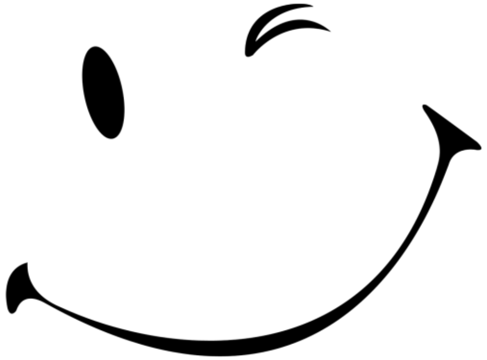
Combinational Logic Circuit



Sequential Logic Circuit

Logic Circuits Cont`d…

We will be discussing about Sequential Circuits and Combinational Logic Circuits in detail in a separate lecture.



Truth Tables

- Truth tables provide a way to describe the **relationship between the inputs and outputs of a logic device**.

- Because a combinational logic block contains no memory, it can be completely specified by defining the values of the outputs for each possible set of input values.

- Such a description is normally given as a truth table.

Truth Tables Cont`d…

- The truth table gives **all the possible values of logical variables** and the combination of the variables.

- The number of rows in the truth table should be equal to **2^n, where "n" is the number of variables in the equation**.

- 1 = True
- 0 = False

Truth Tables Cont`d…

Eg: Consider a logic function with one input (A) and one output (Q)

No of Entries for the truth table = $2^1 = 2$

A	Q
0	
1	

Truth Tables Cont`d…

Eg: Consider a logic function with two inputs (A) (B) and one output (Q)

No of Entries for the truth table = 2^2 = 2

A	B	Q
0	0	
0	1	
1	0	
1	1	

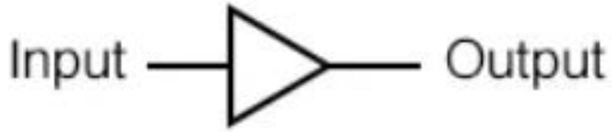
Logic Gates

- Logic gates operate using either 0 or 1 binary values.
- In digital systems, a “high” level usually signifies a binary ‘1 . Conversely, a “low” level is associated with a binary ‘0’
- Truth Table illustrates every potential combination of inputs and the corresponding output for a specific logic gate.

Logic Gates

Digital Buffer

The digital buffers are used in digital circuits for the amplification of digital signals to drive high current loads.

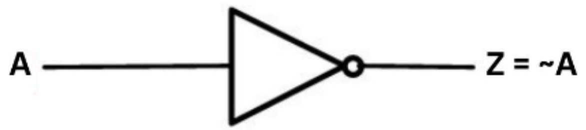


Input	Output
0	0
1	1

Logic Gates Cont`d…

NOT

NOT gates are often called inverters. A NOT gate reverses the logic state. **it has only a single input**



Symbol

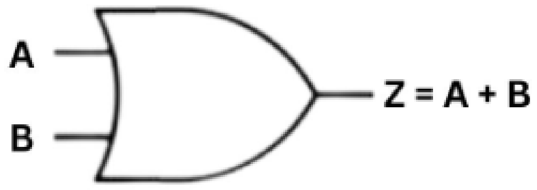
A	Output (~A)
0	1
1	0

Truth Table

Logic Gates Cont`d…

OR

The output is true if one or both inputs are true. If both inputs are false, then the output is false.



Symbol

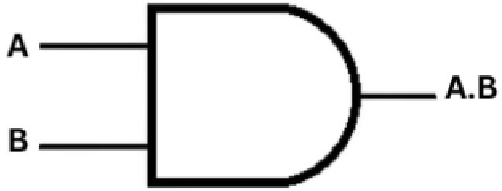
A	B	Output (A+B)
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table

Logic Gates Cont`d…

AND

The output is "true" when both inputs are "true." Otherwise, the output is "false."



Symbol

A	B	Output (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table

Special Purpose Gates

The derived or special gates are made for specific applications such as half adders, full adders, and subtractors. There are two derived logic gates made from OR and NOR gates.

- There are two universal gates:
  - XOR Gate
  - XNOR Gate

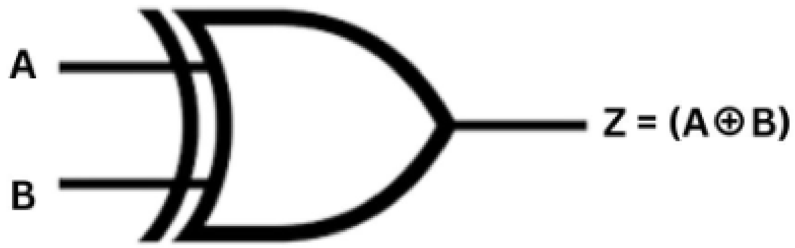
XOR Gate

XNOR Gate

Special Purpose Gates Cont`d…

XOR

The XOR (exclusive-OR) gate; The output is true if either, but not both, of the inputs are true. The output is false if both inputs are "false" or if both inputs are true. **It can accept only two inputs at a time.**



Symbol

A	B	Output (A⊕B)
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table

Special Purpose Gates Cont`d…

Activity

Construct the truth table of $Y = A\overline{B} + B\overline{A}$

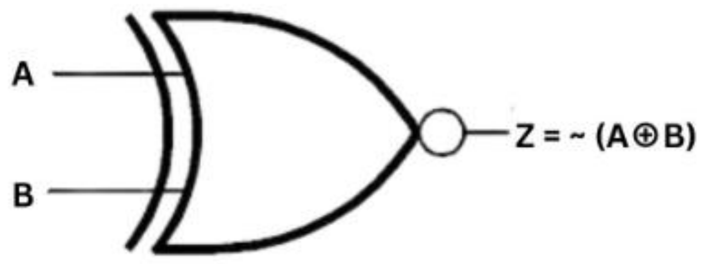
Compare you results with XOR gate output

Special Purpose Gates Cont`d…

XNOR

The XNOR (exclusive-NOR) gate is a combination of an XOR gate followed by an inverter. XNOR gate takes only two inputs and produces one output.

XNOR Gate = XOR Gate + NOT Gate

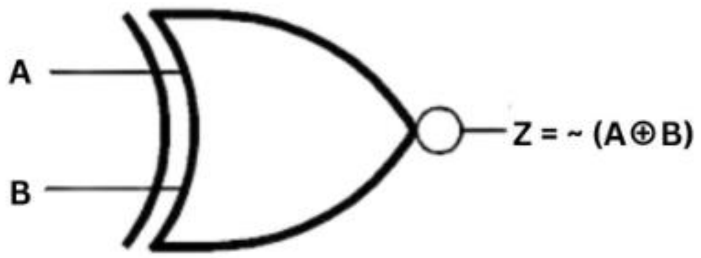


Symbol

Special Purpose Gates Cont`d…

Activity

Construct the truth table of XNOR Gate



Special Purpose Gates Cont`d…

Activity

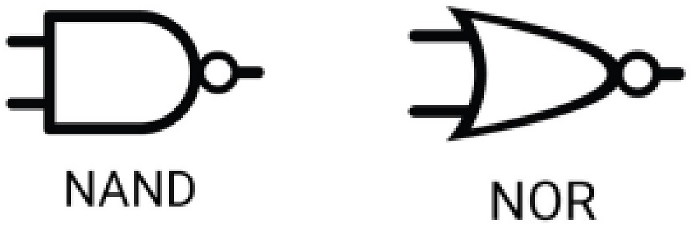
Construct the truth table of $Y = AB + \overline{A}\overline{B}$

Compare you results with XNOR gate output.

Universal Logic Gates

- Universal gates are those gates that **can perform the tasks of other gates** with minor adjustments.

- There are two universal gates:
  - NAND Gate
  - NOR Gate

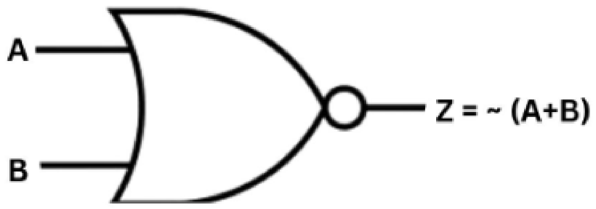


Universal Logic Gates Cont`d…

NOR

The NOR (NOT OR) gate is a combination OR gate followed by an inverter.

NOR Gate = OR Gate + NOT Gate

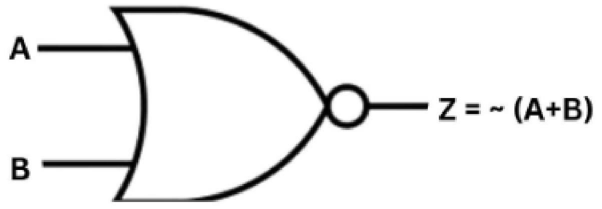


Symbol

Universal Logic Gates Cont`d…

Activity

Construct the truth table of NOR Gate

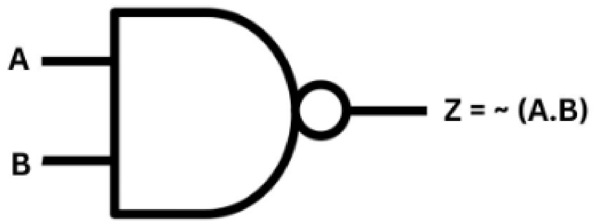


Universal Logic Gates Cont`d…

NAND

The NAND (Negated AND) gate operates as an AND gate followed by a NOT gate.

NAND Gate = AND Gate + NOT Gate

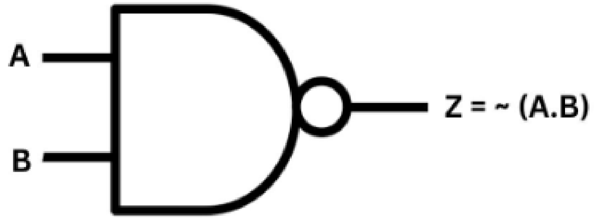


Symbol

Universal Logic Gates Cont`d…

Activity

Construct the truth table of NAND Gate



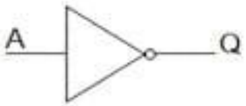
Summary

1. Basic Logic Gates
   I. NOT
   II. AND
   III. OR

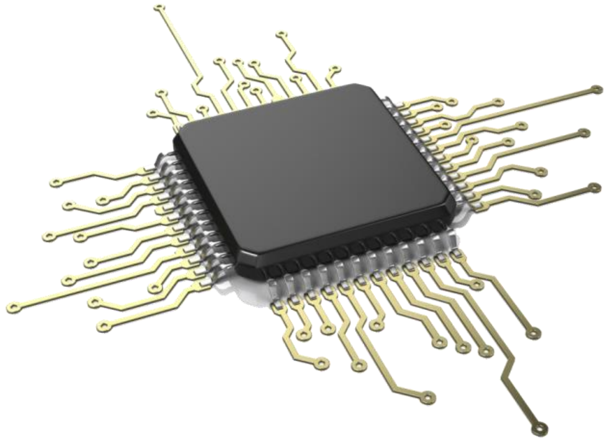
2. Derived Logic Gates / Special Purpose Logic Gates
   I. XOR
   II. XNOR

3. Universal Logic Gates
   I. NAND
   II. NOT

Summary Cont`d...

Gate	Symbol	Boolean Equation
NOT		$Q = \overline{A}$
AND		$Q = A.B$
OR		$Q = A + B$
NAND		$Q = \overline{A.B}$
NOR		$Q = \overline{A + B}$
XOR		$Q = A \oplus B$ or $Q = \overline{A}.B + A.\overline{B}$
XNOR		$Q = \overline{A \oplus B}$ or $Q = \overline{A}.\overline{B} + A.B$

Thank you!



Questions