

# BIT – BInary digiT

- **Bit** (Binary Digit) = Basic unit of information, representing one of two discrete states. The smallest unit of information within the computer.
- Bit has one of two values:
  - 0 (off) or 1 (on)
  - 0 (False) or 1 (True)

*How do we describe the behavior of gates and circuits?*

## Boolean expressions

Uses Boolean algebra, a mathematical notation for expressing two-valued logic

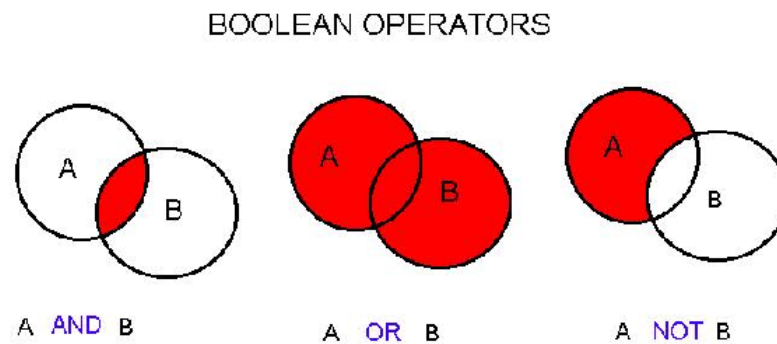
## Logic diagrams

A graphical representation of a circuit; each gate has its own symbol

## Truth tables

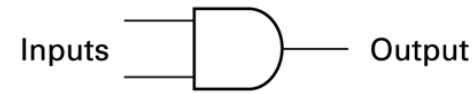
A table showing all possible input value and the associated output values

- **Boolean Operation**: An operation that manipulates one or more true/false values
- Specific operations
  - AND
  - OR
  - XOR (exclusive or)
  - NOT
- Using Truth Tables we can use different sets of logic operations to store, add, subtract, and more complicated operations with bit.



- **Logic gates** have:
  - **one or more inputs**
  - **only one output**
- The output is active only for certain input combinations.
- Logic gates are the building blocks of any digital circuit.

#### AND



Inputs	Output
0 0	0
0 1	0
1 0	0
1 1	1

#### OR



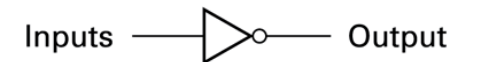
Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	1

#### XOR



Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	0

#### NOT



Inputs	Output
0	1
1	0

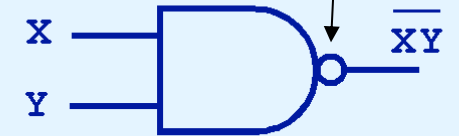
# Combining Logic Gates

- NAND and NOR are two very important gates.
- “ o ” is the **inversion bubble**.

## Inversion Bubble

X NAND Y

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

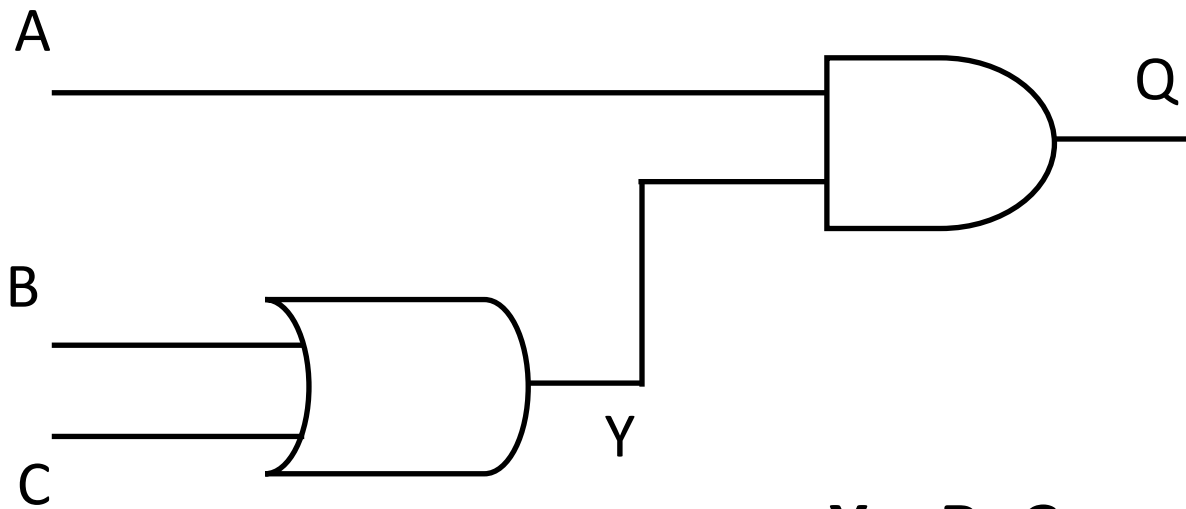


X NOR Y

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



# Combinational Circuits (example)



$$Y = B + C$$
$$Q = A(B + C)$$

Truth Table

A	B	C	Y	Q
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1