Transistors, Boolean Logic and Logical Gates

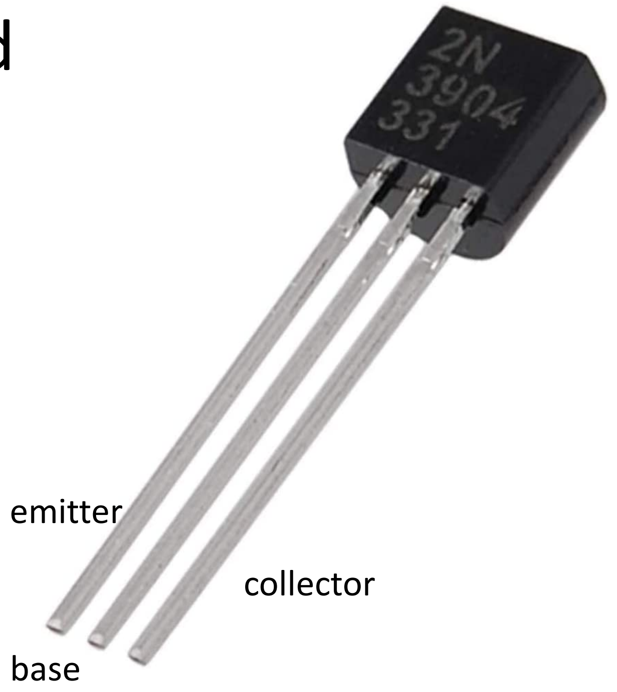
Garrett Dancik, PhD

Fall 2021

Course Notes: https://gdancik.github.io

Why Binary?

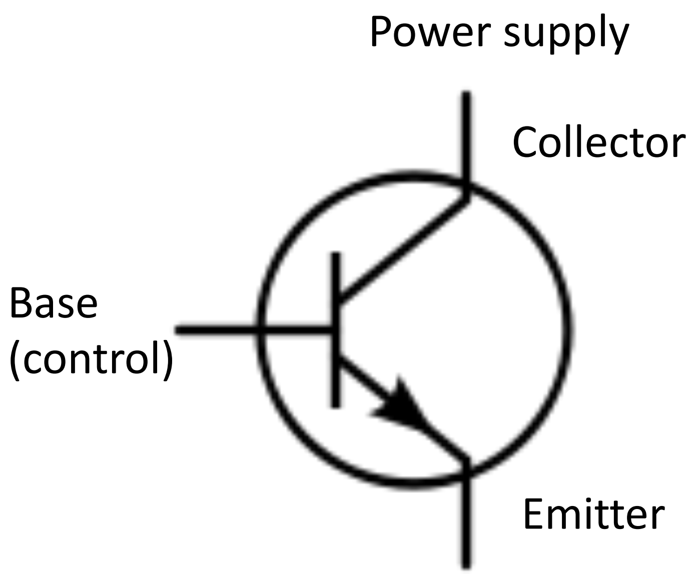
- The foundation of computer hardware (processing and memory) are devices that can operate in two stable states.
  - A transistor can either turn current on (1) or off (0)
  - Magnetic core memory (common until 1975): a core can be magnetized in either the clockwise or counter-clockwise direction
  - A hard drive consists of regions of magnetic material which can be magnetized (1) or not (0)



https://www.amazon.com/MCIGICM-200pcs-Transistor-Bipolar-Transistors/dp/B06XRBLKDR

Transistor basics

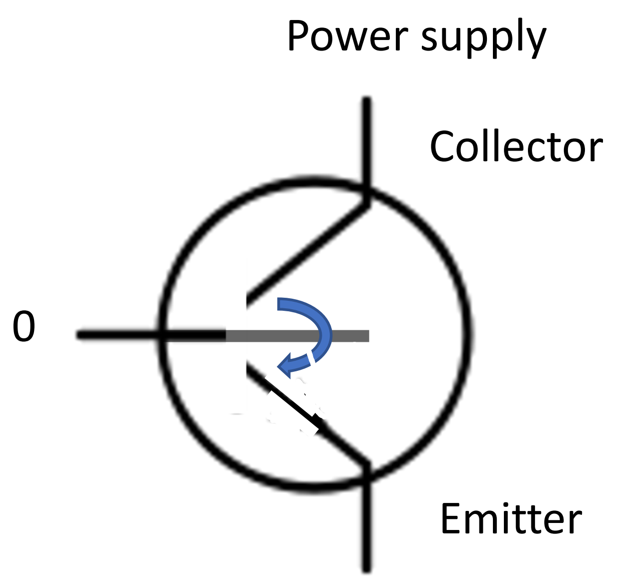
- A transistor is a semi-conductor that can amplify or switch electronic signals.



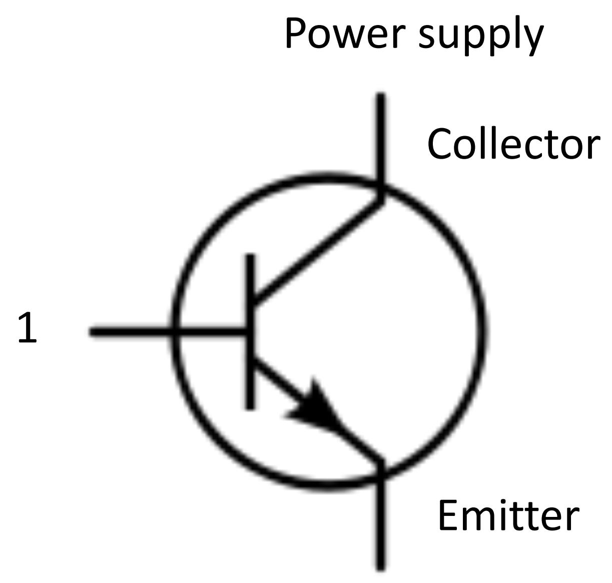
- The base is used to open/close the switch
  - If the control line is set to 1, then the switch is closed, and current will flow from the collector to the emitter. The transistor is ON (1). (This is the state of the current figure)
  - If the control line is set to 0, then switch is open, and current will not flow from the collector to the emitter. The transistor is OFF (0)

https://commons.wikimedia.org/wiki/File:Transistor.symbol.npn.svg

A transistor can be OFF (0) or ON (1)



No Voltage = OFF
Binary value = 0



High voltage = ON
Binary value = 1

Additional details: https://www.youtube.com/watch?v=stM8dgcY1CA

Boolean logic

- Computer circuits are constructed based on Boolean logic
  - Boolean logic is a branch of mathematics (algebra) that deals with *true* and *false* values
- In computer logic,
  - *true* represents a binary 1 or a transistor that is ON
  - *false* represents a binary 0 or a transistor that is OFF
- Boolean operations include AND, OR, NOT, NOR, NAND, and XOR

Truth table for AND and OR

Truth table for: *a* AND *b*

Inputs		Output
a	b	*a AND b* (also written as $a \cdot b$)
True (1)	True (1)	True (1)
True (1)	False (0)	False (0)
False (0)	True (1)	False (0)
False (0)	False (0)	False (0)

The expression *a AND b* is True only if both *a* and *b* are True

Truth table for: *a* OR *b*

Inputs		Output
a	b	*a OR b* (also written as $a + b$)
True (1)	True (1)	True (1)
True (1)	False (0)	True (1)
False (0)	True (1)	True (1)
False (0)	False (0)	False (0)

The expression *a OR b* is True if either *a* or *b* are True (including if both are True)

Truth table for NOT

Truth table for: *NOT a*

Inputs	Output
a	*NOT a* (also written as ā)
True (1)	False (0)
False (0)	True (1)

The expression *NOT a* is True if *a* is False and is False if *a* is True

- Boolean logic examples:
  - grade is between 90 and 100
    - grade >= 90 AND grade <= 100
  - User has typed 'Y' or 'y'
    - userInput == 'Y' OR userInput == 'y'
  - User has not typed 'Y' or 'y'
    - userInput != 'Y' AND userInput != 'y'

Truth table for NAND and NOR

Truth table for: *a* NAND *b*

Inputs		Output
a	b	*a NAND b* (also written as $\overline{\mathrm{a \cdot b}}$)
True (1)	True (1)	False (0)
True (1)	False (0)	True (1)
False (0)	True (1)	True (1)
False (0)	False (0)	True (1)

The expression *a NAND b* is equivalent to *NOT* (a *AND* b) and is True if either *a* or *b* are False

Truth table for: *a* NOR *b*

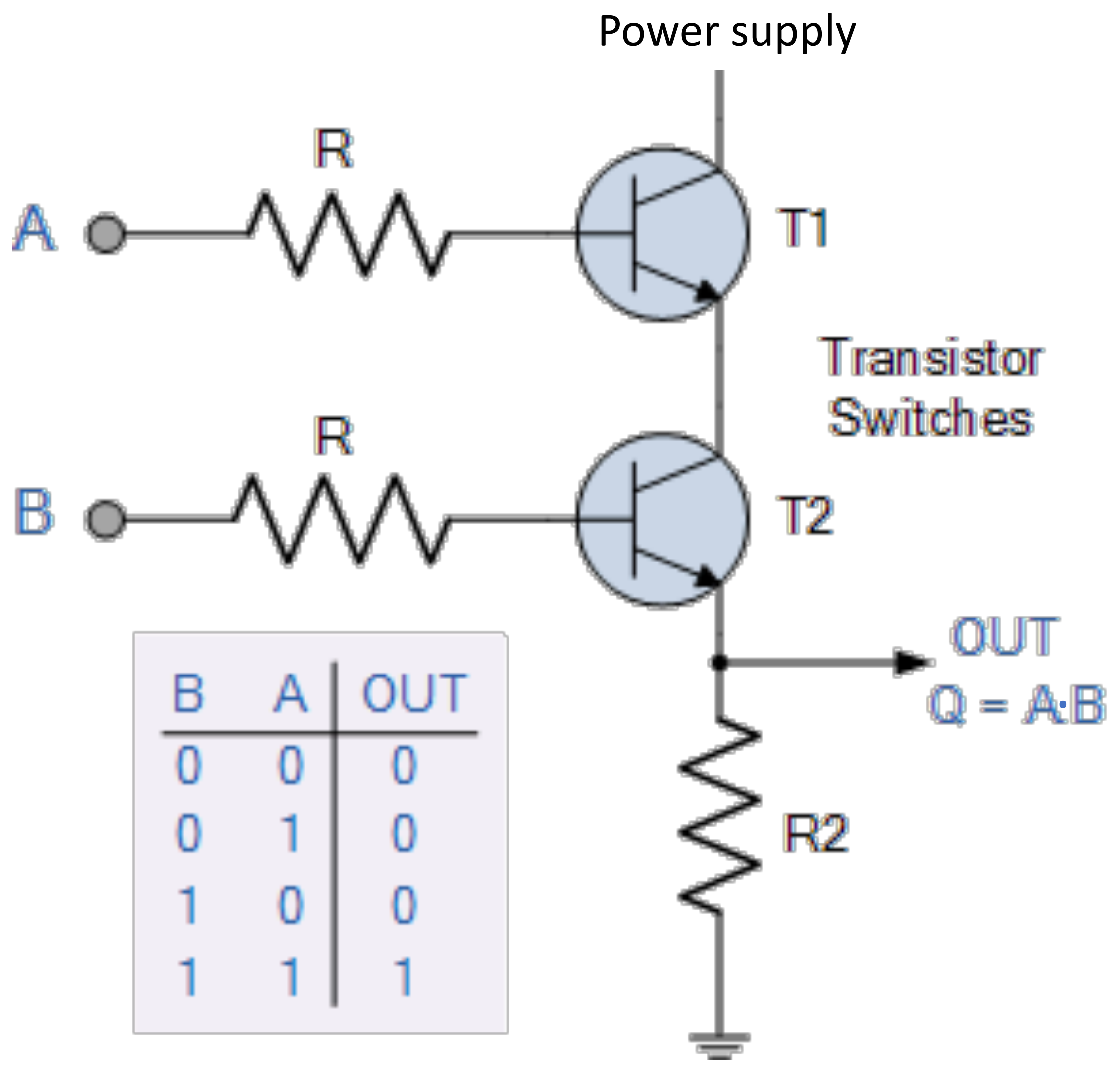
Inputs		Output
a	b	*a NOR b* (also written as $\overline{\mathrm{a} + b}$)
True (1)	True (1)	False (0)
True (1)	False (0)	False (0)
False (0)	True (1)	False (0)
False (0)	False (0)	True (1)

The expression *a NOR b* is equivalent to NOT (*a* OR *b)* and is True only if both *a* and *b* are False

Logic Gates

- A *logic gate* is an electronic device that takes one or more binary inputs and produces a single binary output.
- Gates are created from transistors
- Types of gates: AND, OR, NAND, NOR (and XOR and XAND)
- An AND gate is shown on the right
  - Recall: If the base (input) is 1, the transistor switch will be closed
  - Only if A and B are both 1, will current flow from the power supply through the transistors, resulting in voltage at the output (OUT)
  - This gives a digital implementation of the Boolean AND operator

Transistor AND Gate

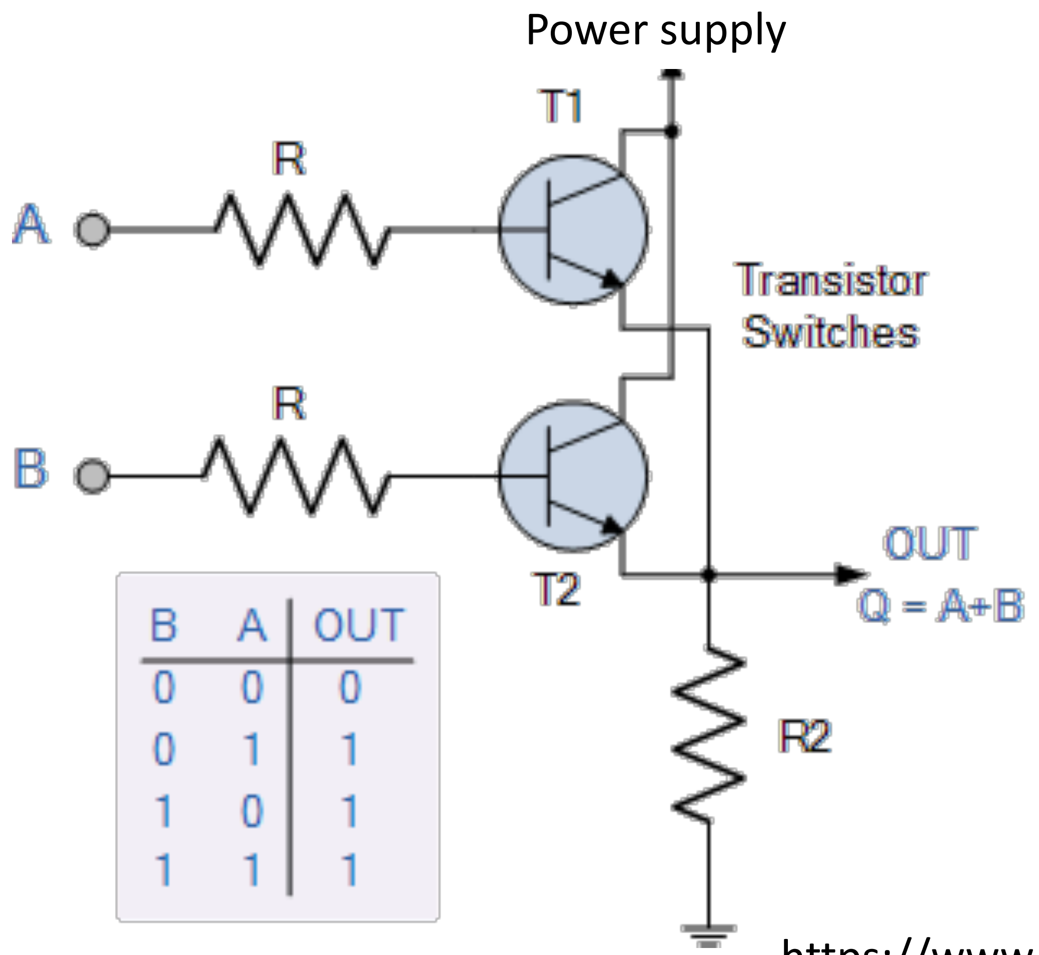


B	A	OUT
0	0	0
0	1	0
1	0	0
1	1	1

https://www.electronics-tutorials.ws/logic/logic_2.html

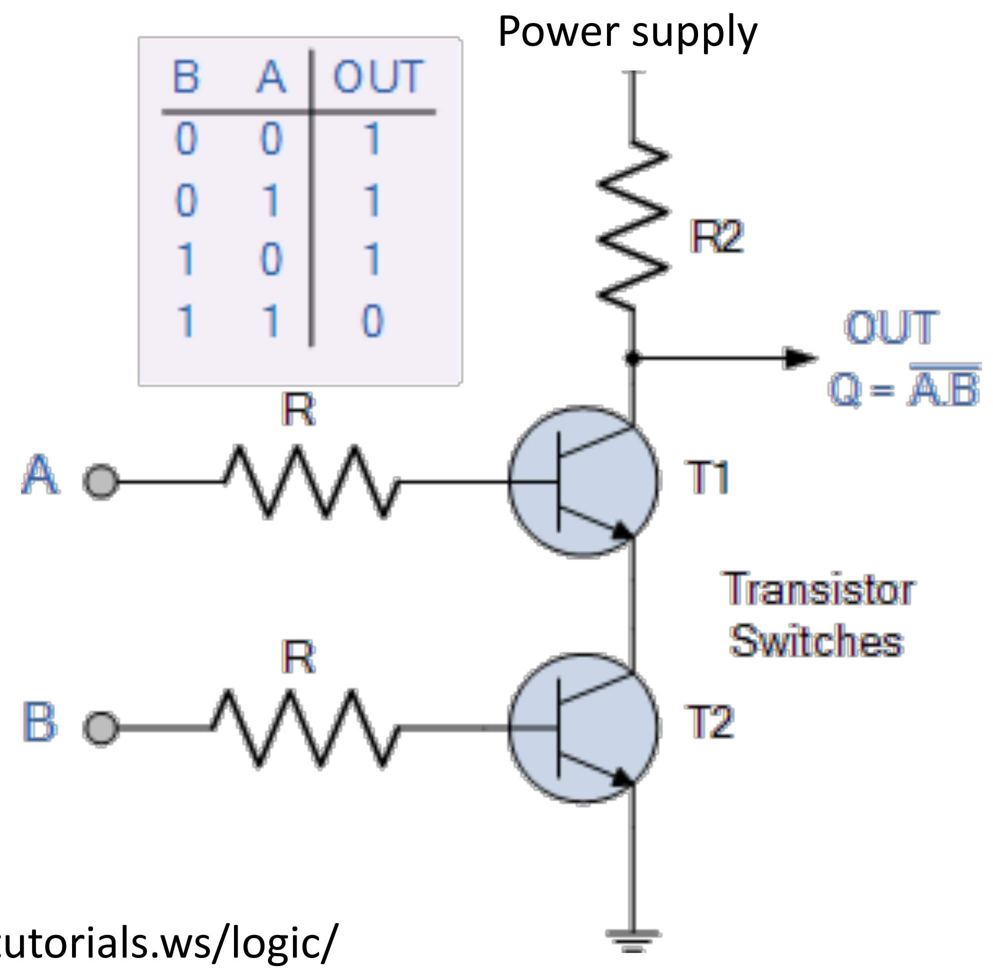
Additional Gates

Transistor OR Gate



B	A	OUT
0	0	0
0	1	1
1	0	1
1	1	1

Transistor NAND Gate

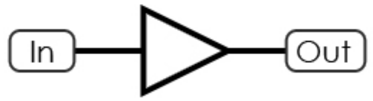


B	A	OUT
0	0	1
0	1	1
1	0	1
1	1	0

https://www.electronics-tutorials.ws/logic/

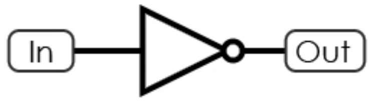
Logic Gates - Symbols and Truth Tables

BUF
(Buffer)



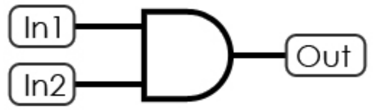
In	Out
0	0
1	1

NOT
(Inverter)



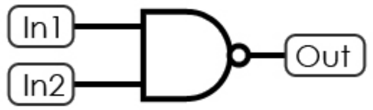
In	Out
0	1
1	0

AND



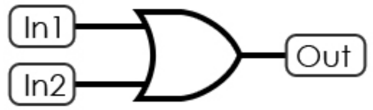
In1	In2	Out
0	0	0
0	1	0
1	0	0
1	1	1

NAND
(NOT AND)



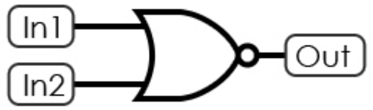
In1	In2	Out
0	0	1
0	1	1
1	0	1
1	1	0

OR



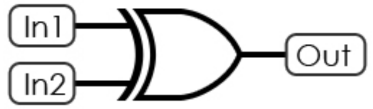
In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	1

NOR
(NOT OR)



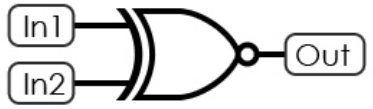
In1	In2	Out
0	0	1
0	1	0
1	0	0
1	1	0

XOR
(Exclusive Or)



In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	0

XNOR
(NOT XOR)



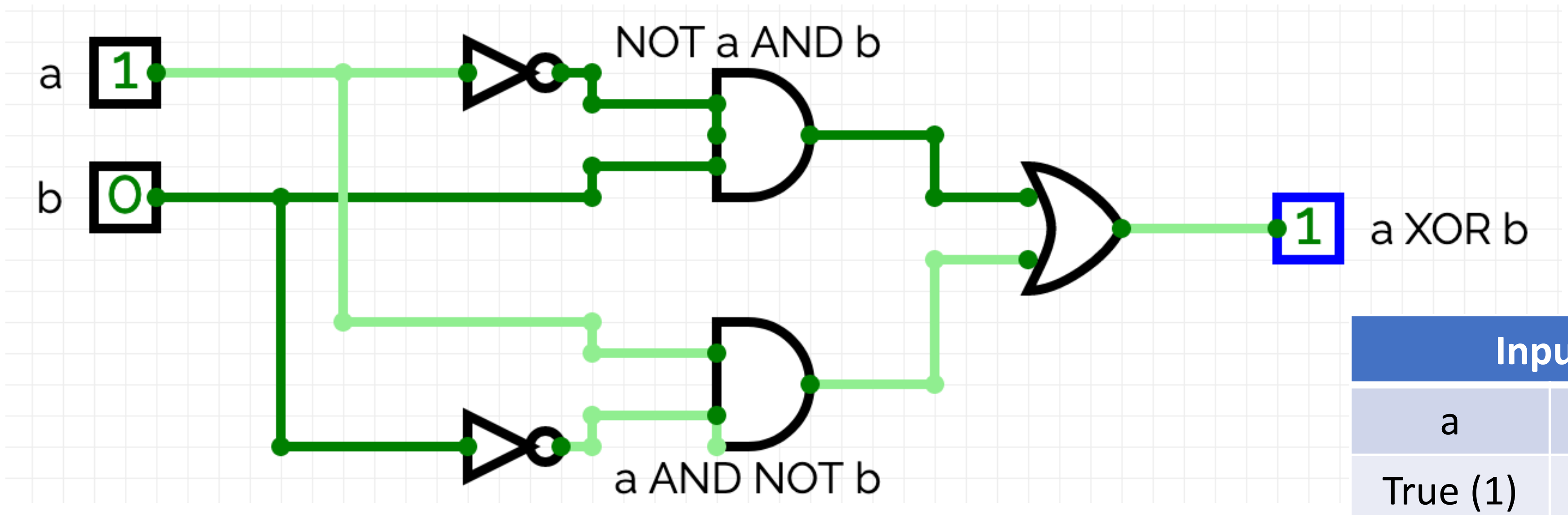
In1	In2	Out
0	0	1
0	1	0
1	0	0
1	1	1

A circle behind a symbol indicates that the output signal is inverted.

http://www.exclusivearchitecture.com/?page_id=2425

Logic gates are the building blocks of computer systems

Exclusive OR (XOR)



Inputs		Output
a	b	*a XOR b*
True (1)	True (1)	False (0)
True (1)	False (0)	True (1)
False (0)	True (1)	True (1)
False (0)	False (0)	False (0)

XOR is an *exclusive or* which returns True only if *a* and *b* are different

https://circuitverse.org/users/89029/projects/xor-75562e33-e621-4231-aad9-dc5f6eb649af

Digital Circuits

- A circuit is a collection of logic gates that transform binary inputs into binary outputs.
  - If the outputs depend only on the current inputs, the circuit is a *combinational circuit*
  - If the outputs also depend on previous inputs, the circuit is a *sequential circuit*
- Every output in a circuit can be represented as a Boolean expression
- We will use https://circuitverse.org/ to simulate circuits