Boolean Logic

Will Leeson

What comes to mind when you think of "Logic"?

Boolean Logic

- Developed by George Boole in 1854
- A systematic approach to logic
- Two Values
  - True (1)
  - False (0)
- Variables
- Three “Basic” operators
- Several “Secondary” Operators

X	Y	X Φ Y
T	T	T
T	F	F
F	T	F
F	F	T

Some Terminology

- Constant
  - A value that does not change
  - In algebra: 1, -30, 2.541, π
  - In boolean logic: True, False
- Variable
  - A value that can span many values
  - Usually represented by a single letter (x, y, z, etc.)
  - Same for both algebra and boolean logic

Some Terminology

- Operator
  - A symbol representing a set function
  - Unary and Binary operators
  - In algebra: +, -, /, ^, etc.
  - In boolean logic: $\land$, $\lor$, $\leftrightarrow$, $\neg$
- Operand
  - The values an operator acts on
  - Algebra: **1** + **3**, **27** / **x**, -**3**, etc.
  - Boolean logic: **True** $\land$ **False**, **X** $\leftrightarrow$ **Y**, $\neg$**X**
- Expression
  - A combination of operators and operands
  - Follows rules according to the mathematical language

Conjunction

- Binary Operator
- In words - and
- In symbols - ∧
- Only true is both expressions are true
  - “Did you go to dinner **and** a movie.”
  - “If you are happy **and** you know it, clap your hands”

X	Y	X ∧ Y
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction

- Binary Operator
- In words - or
- In symbols - $\vee$
- True when either expression is true
  - “My friends must enjoy listening to Folk **or** R&B music”
  - “Are there shellfish **or** cheese in this dish? I’m deathly allergic.”

X	Y	X $\vee$ Y
T	T	T
T	F	T
F	T	T
F	F	F

Negation

- Unary Operator
- In words - not
- In symbols - ¬
- True when the expression is False
    - “I am **not** 30 years old.”
    - “They are **not** a fan of the New York Jets.”

X	¬X
T	F
F	T

Conditional

- Binary Operator
- In words - If X then Y
- In symbols - ➔
- True unless X is true and Y is false
  - “**If** I’ve been to Pluto, **then** I’ve been to Mars.”
  - “**If** I’ve seen a cute dog, **then** I’ve said out loud ‘Ooo, cute dog’”

X	Y	$X \rightarrow Y$
T	T	T
T	F	F
F	T	T
F	F	T

Biconditional

- Binary Operator
- In words - X if and only if Y
- In symbols - $\leftrightarrow$
- True if X equals Y
    - “Johnny can have dessert **if and only if** I did all of my homework”
    - “I will go to the concert **if and only if** I know the band that is playing.”

X	Y	$X \leftrightarrow Y$
T	T	T
T	F	F
F	T	F
F	F	T

Exclusive Disjunction

- Binary Operator
- In words - (exclusive) or
- In symbols - ⊕
- True if either X or Y is true, not both
  - “Would you like the chicken **or** the fish?”
  - “I need to take my pill **or** the lactose in the pizza will be a problem.”

X	Y	X ⊕ Y
T	T	F
T	F	T
F	T	T
F	F	F

Order of operation

- In algebra, PEMDAS
    - Parentheses
    - Exponent
    - Multiplication/Division
    - Addition/Subtraction
- In boolean logic, IPAOEBC
    - Inverse (Not)/Parentheses
    - And
    - Or/EXOR
    - Biconditional/Conditional

Truth Tables

- A way to structure Boolean Formula
  - Break down the formula into “atoms”
  - Define the atoms using True and False
  - Combine atoms using order of operations
  - Repeat until none are left

X	Y	$X \rightarrow Y$
T	T	T
T	F	F
F	T	T
F	F	T

Truth Tables

$$(X \lor Y) \land \neg X$$

X	Y	(X ∨ Y)	¬X	(X∨Y) ∧ ¬X
T	T	T	F	F
T	F	T	F	F
F	T	T	T	T
F	F	F	T	F

Truth Tables

$$(A \rightarrow B) \lor (B \rightarrow A)$$

A	B	$(A \rightarrow B)$	$(B \rightarrow A)$	$(A \rightarrow B) \lor (B \rightarrow A)$
T	T	T	T	T
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

Tautology!

Truth Tables

$$(X \lor Y) \land \neg(X \lor Y)$$

X	Y	$(X \lor Y)$	$\neg(X \lor Y)$	$(X \lor Y) \land \neg(X \lor Y)$
T	T	T	F	F
T	F	T	F	F
F	T	T	F	F
F	F	F	T	F

Contradiction!

Truth Tables

$$X \wedge Y \leftrightarrow Z \vee Y$$

X	Y	Z	$X \wedge Y$	$Z \vee Y$	$X \wedge Y \leftrightarrow Z \vee Y$
T	T	T	T	T	T
T	T	F	T	T	T
T	F	T	F	T	F
T	F	F	F	F	T
F	T	T	F	T	F
F	T	F	F	T	F
F	F	T	F	T	F
F	F	F	F	F	T

Logical Equivalence

X	Y	$X \to Y$
T	T	T
T	F	F
F	T	T
F	F	T

$$\equiv$$

X	Y	$\neg X$	$\neg X \lor Y$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

Logical Equivalence

X	Y	$X \leftrightarrow Y$
T	T	T
T	F	F
F	T	F
F	F	T

$\equiv$

X	Y	$(X \land Y) \lor (\neg X \land \neg Y)$
T	T	T
T	F	F
F	T	F
F	F	T

Logical Equivalence

X	Y	$X \oplus Y$
T	T	F
T	F	T
F	T	T
F	F	F

$\equiv$

X	Y	$(X \lor Y) \land \neg(X \land Y)$
T	T	T
T	F	F
F	T	F
F	F	T

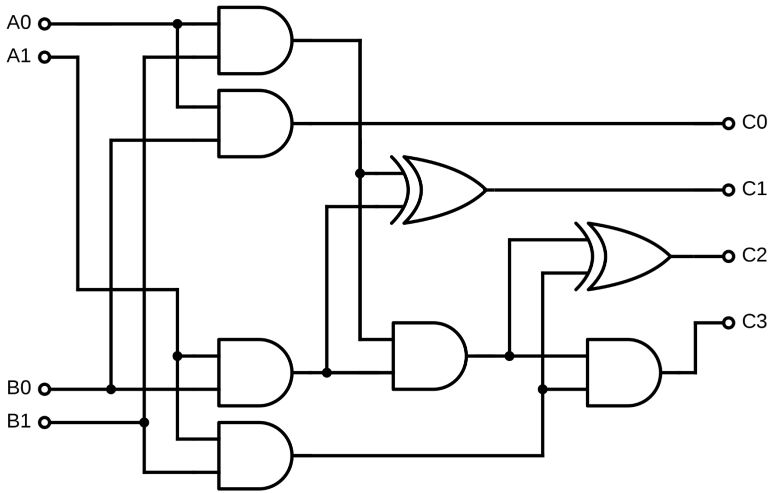
So what does this have to do with Computers?

Computers are machines

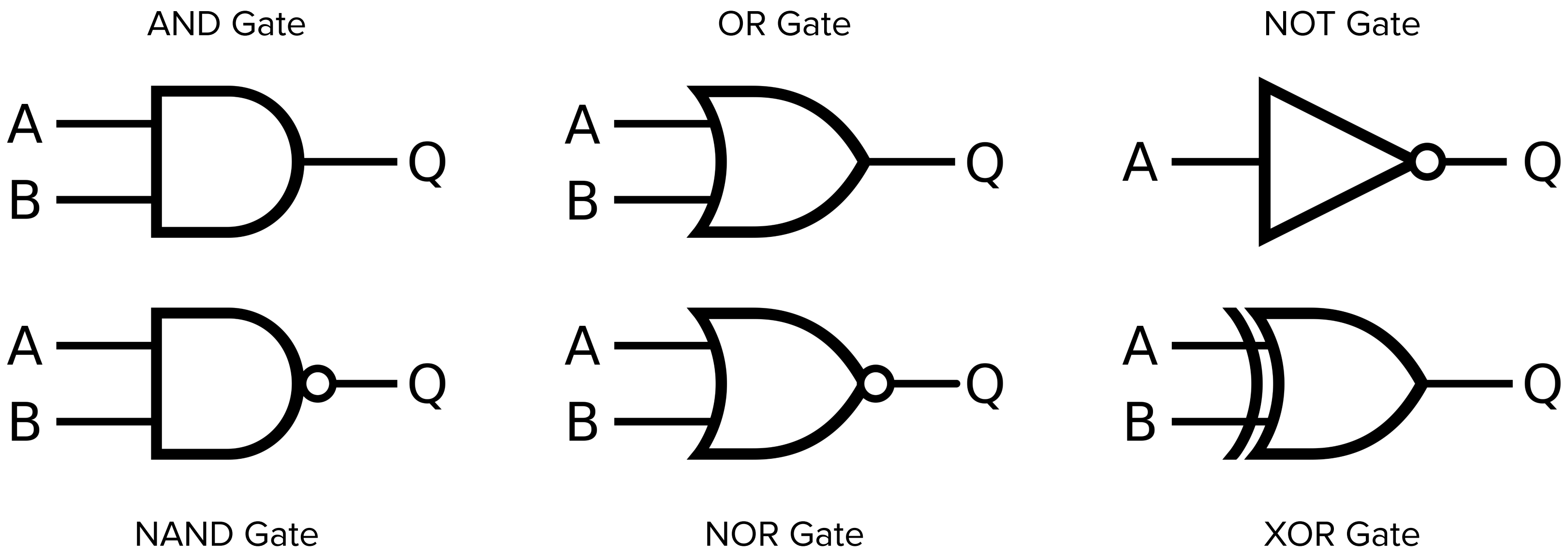
- They do not think for themselves
- They follow a set of instructions
    - Can be informed by external stimulus
    - Can be informed by “randomness”
- Programs rarely don’t make “decisions”
    - If they clicked button X, do Y
    - If X and Y or Z, do A
- When writing programs, you will use boolean logic

Computers are machines

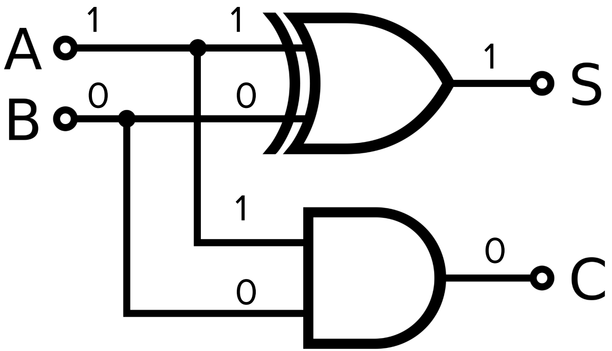
- Computers are wires with electricity running through them
- They don’t know what X+Y means
  - We must translate X+Y to electricity
  - This is where Boolean Algebra comes in
- Different “gates” enact boolean operations
- Circuits are combinations of gates serving different purposes

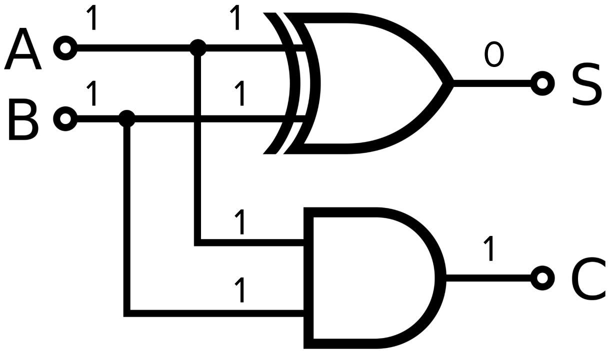


Gate diagrams



Addition Circuit





In decimal: 1 + 0 = 1

In binary: 1 + 0 = 1

In logic: $S = 1 \oplus 0 = 1$

$C = 1 \wedge 0 = 0$

In decimal: 1 + 1 = 2

In binary: 1 + 1 = 10

In logic: $S = 1 \oplus 1 = 0$

$C = 1 \wedge 1 = 1$

And we can go on from there…