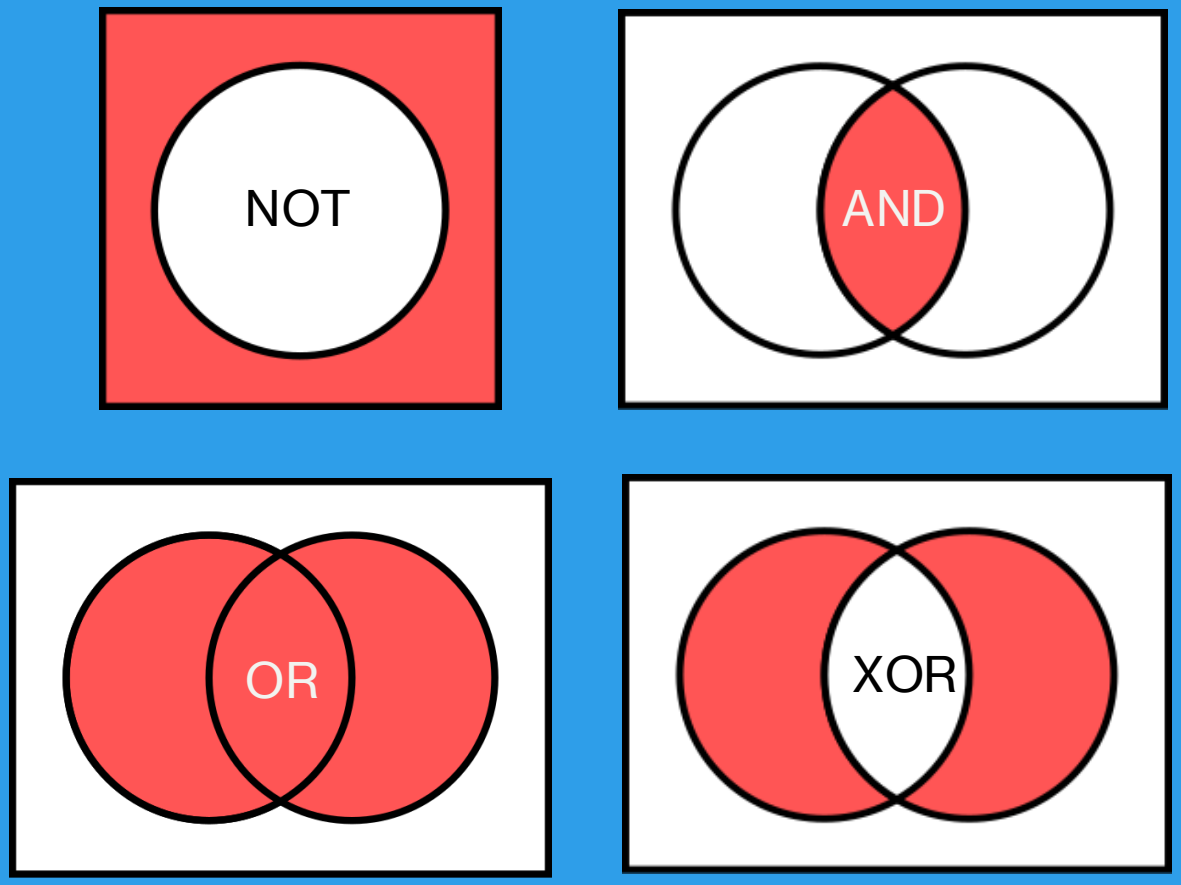
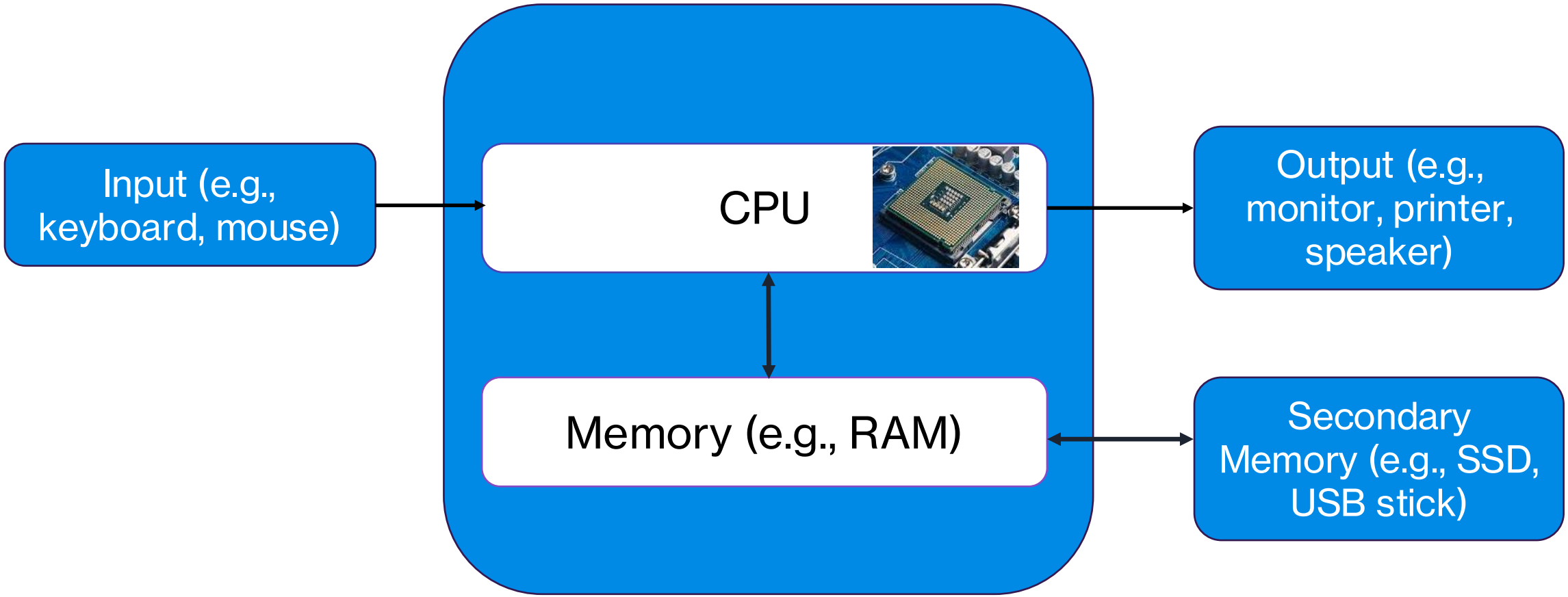
Computer Systems



Boolean Algebra

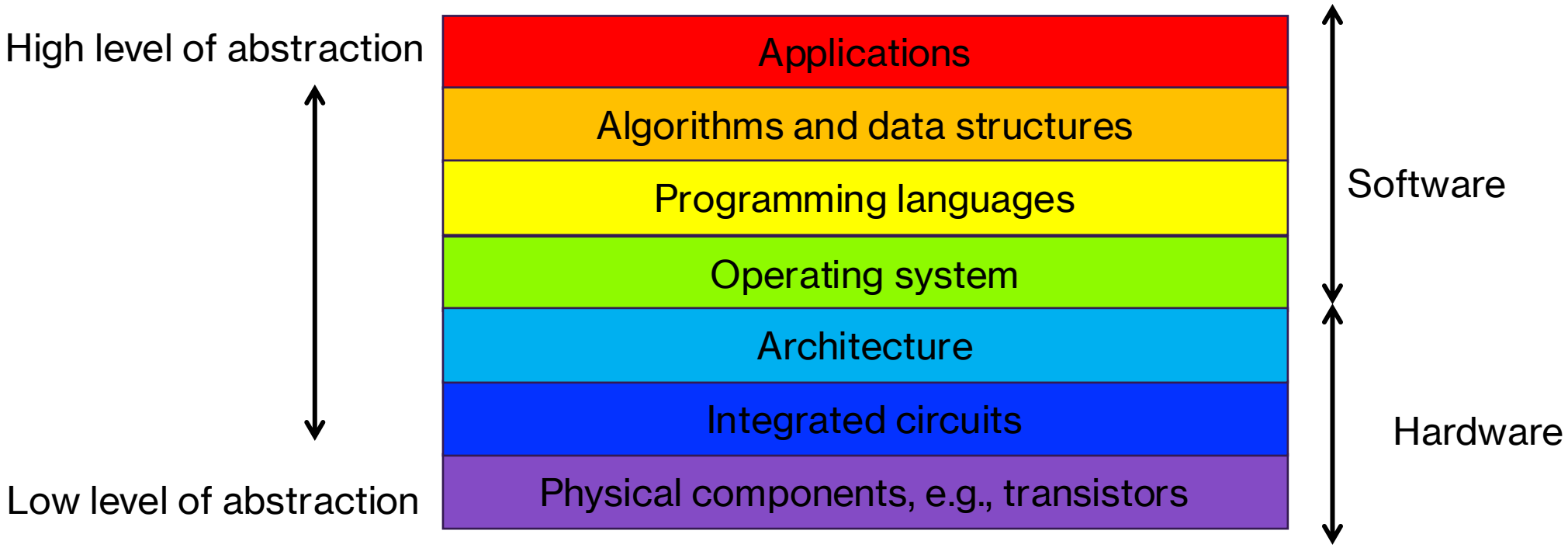
CS51 – Fall 2026

A simplified view of a computer



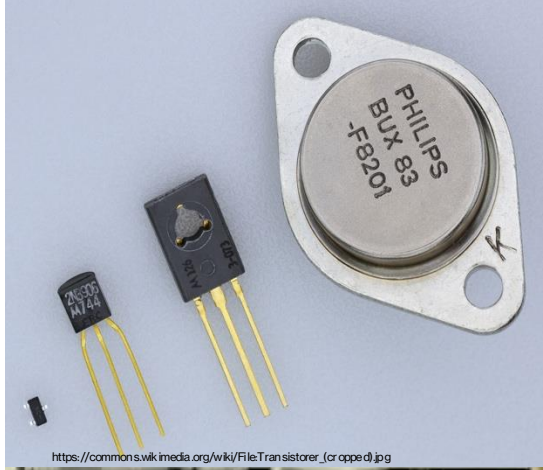
Abstraction

- Critical technique for managing complexity. We will simplify and generalize information by hiding details when they are not important. This allows us to cultivate a higher level of understanding without being overwhelmed with insignificant details.



Transistors

- Transistors are semiconductor devices made out of silicon.
- They act as switches that can be open or closed by applying electricity.
- Integrated circuits in modern computers are built off billions of transistors.
  - They are **small** – you can fit a few thousand of them across the width of a human hair!
  - They are **fast** – they can switch states millions of times per second.
  - They are **reliable** – they can run without any issue for decades.
- Invented in 1947 by Bardeen, Brattain, and Shockley at Bell Lab.
  - They shared the 1956 Nobel Prize in Physics for their achievement.
- A lot of development of transistors and circuits took place in Santa Clara Valley, which by the 1980s started being referred to as the **Silicon Valley**.



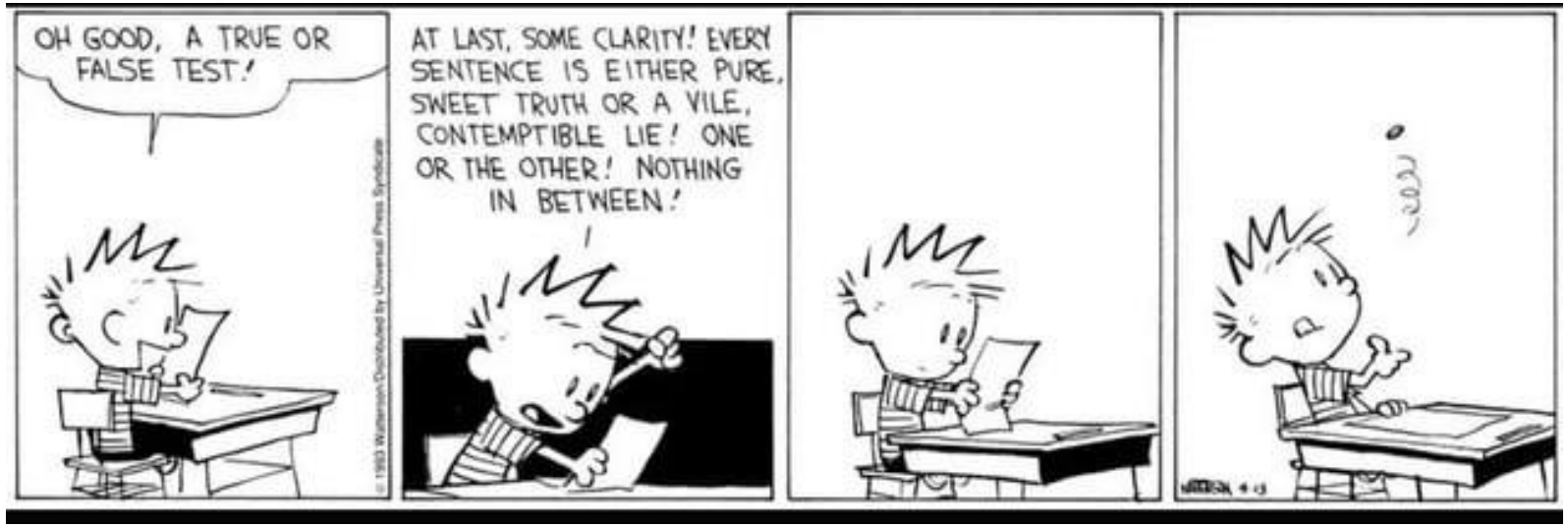
https://commons.wikimedia.org/wiki/File:Transistorer_(cropped).jpg

Think

- *What is a Boolean (e.g., a Boolean variable)*

Think

- *What is a Boolean (e.g., a Boolean variable)*
  - A variable/value that only has two possible values True/1 or False/0



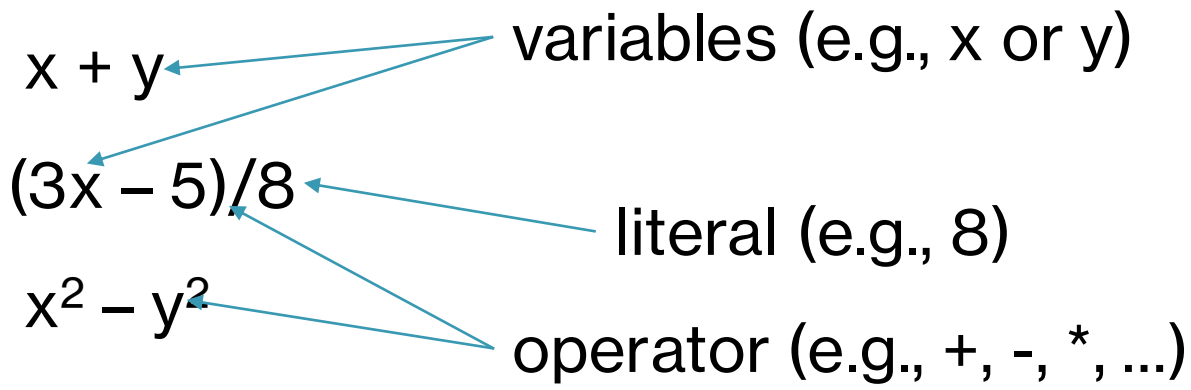
Boolean Algebra

Binary representation and Boolean algebra

- Transistors are powered with electricity and can be in one of two states, allowing us to represent information in **binary**.
- Binary means two states. Common interchangeable representations are:
  - 0 and 1,
  - False and True,
  - Low and High.
- Working only with two states has the advantage of giving us a distinct signal, thus minimizing errors.
- It also works with a well-established branch of Mathematics called **Boolean algebra** that combines Boolean variables.

George Boole

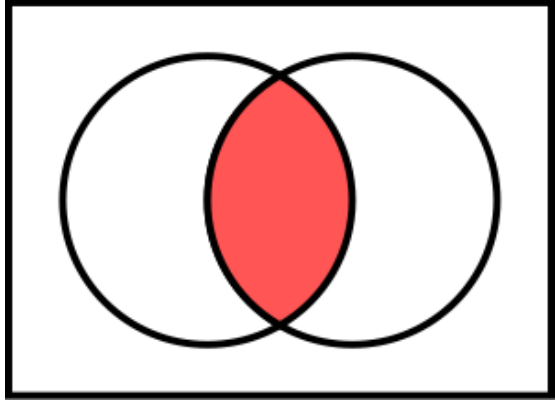
Algebra as you know it



Boolean Algebra

- Boolean algebra is a branch of algebra where variables can only have two values
- Literals: True, False (alternatively: 1, 0)
- Operators?

AND operation – Conjunction



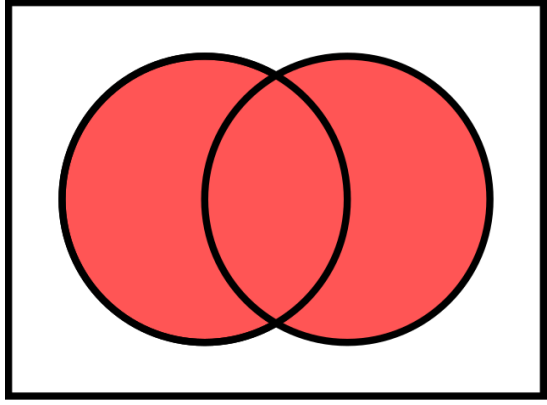
- Binary operator, i.e., it is applied two operands (Boolean variables or literals).
- Also denoted as ∧.
- Produces True (1) only if *both* operands are True (1).
- We can use a **truth table** to enumerate all possible values of operands and the effect of the logical operations when applied to them.

X	Y	X AND Y
False	False	False
False	True	False
True	False	False
True	True	True

Equivalently

X	Y	X ∧ Y
0	0	0
0	1	0
1	0	0
1	1	1

OR operation – Disjunction



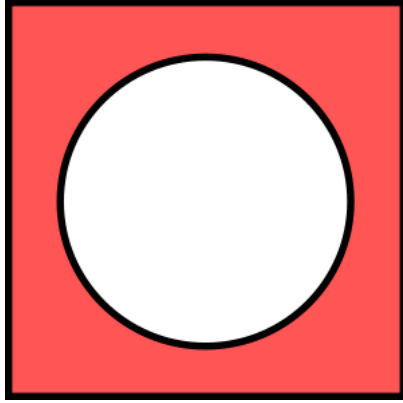
- Binary operator, i.e., it is applied two operands. Also denoted as ∨.
- Produces True (1) if *at least* one operand is True (1).
- Truth table for OR:

X	Y	X OR Y
False	False	False
False	True	True
True	False	True
True	True	True

Equivalently

X	Y	X ∨ Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT operation – Negation



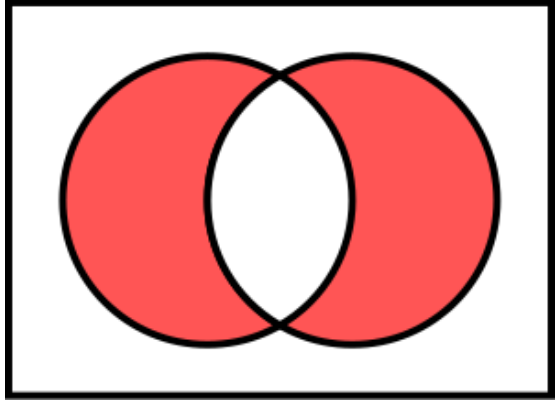
- Unary operator, i.e., it is applied to only one operand. Also denoted as ¬.
- It flips False (0) to True (1) and, equivalently, True (1) to False (0).
- Truth table for NOT:

X	NOT X
False	True
True	False

Equivalently

X	¬ X
0	1
1	0

XOR operation – Exclusive OR



- Binary operator, i.e., it is applied two operands. Also denoted as $\oplus$.
- Produces True (1) if *exactly* one operand is True (1).
- Truth table for XOR:

X	Y	X XOR Y
False	False	False
False	True	True
True	False	True
True	True	False

Equivalently

X	Y	X $\oplus$ Y
0	0	0
0	1	1
1	0	1
1	1	0

PRACTICE TIME – Evaluate these expressions

- 0 ∨ (¬ 0)
- 1 ∧ (¬ 1)
- ¬ (1 ∧ 0)
- 0 ∧ (¬ 1)
- ¬ (1 ∨ (¬ 1))
- 1 ⊕ (¬ 1)
- (1 ∧ (1 ⊕ ¬(0 ∨ ¬ 1)))

AND

X	Y	X ∧ Y
0	0	0
0	1	0
1	0	0
1	1	1

NOT

X	¬ X
0	1
1	0

OR

X	Y	X ∨ Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

ANSWER – Evaluate these expressions

- $0 \vee (\neg 0) = 0 \vee 1 = 1$
- $1 \wedge (\neg 1) = 1 \wedge 0 = 0$
- $\neg (1 \wedge 0) = \neg 0 = 1$
- $0 \wedge (\neg 1) = 0 \wedge 0 = 0$
- $\neg (1 \vee (\neg 1)) = \neg (1 \vee 0) = \neg 1 = 0$
- $1 \oplus (\neg 1) = 1 \oplus 0 = 1$
- $(1 \wedge (1 \oplus \neg(0 \vee \neg 1))) = (1 \wedge (1 \oplus \neg(0 \vee 0))) = (1 \wedge (1 \oplus \neg 0)) = (1 \wedge (1 \oplus 1)) = 1 \wedge 0 = 0$

AND

X	Y	X ∧ Y
0	0	0
0	1	0
1	0	0
1	1	1

NOT

X	¬ X
0	1
1	0

OR

X	Y	X ∨ Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

Boolean functions of three+ variables

- AND($X_1, X_2, \ldots, X_n$) = 0 if any argument is 0, 1 if all arguments are 1.
  - This is equivalent to writing $X_1 \wedge X_2 \wedge \ldots \wedge X_n$
- OR($X_1, X_2, \ldots, X_n$) = 1 if any argument is 1, 0 if all arguments are 0.
  - This is equivalent to writing $X_1 \vee X_2 \vee \cdots \vee X_n$

X	Y	Z	AND(X,Y,Z)	OR(X,Y,Z)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Think

- 2 + 3 * 5 =

- (2 + 3) * 5 =

Operator precedence in Algebra

- 2 + 3 * 5 = 17

- (2 + 3) * 5 = 25

Think

- $1 \vee 1 \wedge 0 =$

- $\neg 0 \wedge 0 =$

Operator precedence in Boolean Algebra

- $1 \vee 1 \wedge 0 = 1$
- $\neg 0 \wedge 0 = 0$

highest precedence (happens first)

$\neg$

$\wedge$

$\vee$

lowest precedence (happens later)

Operator precedence in Boolean Algebra

- $(1 \vee 1) \wedge 0 = 0$
- $\neg(0 \wedge 0) = 1$

highest precedence (happens first)

$\neg$

$\wedge$

$\vee$

lowest precedence (happens later)

Think

- $x(y + z) =$

Distributive Law in Algebra

- $x(y + z) = xy + xz$

Think

- $x \wedge (y \vee z) =$

Distributive Law in Boolean Algebra

- $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
- *How can you check this?*

Distributive Law in Boolean Algebra

- $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

x	y	z	y ∨ z	x ∧ (y∨ z)	(x ∧ y)	(x ∧ z)	(x ∧ y) ∨ (x ∧ z)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

PRACTICE TIME – De Morgan's laws

- ¬ (X ∧ Y) = ¬ X ∨ ¬ Y
- ¬ (X ∨ Y) = ¬ X ∧ ¬ Y

X	Y	X ∧ Y	¬ (X ∧ Y)	X ∨ Y	¬ (X ∨ Y)	¬ X	¬ Y	¬ X ∧ ¬ Y	¬ X ∨ ¬ Y
0	0								
0	1								
1	0								
1	1								

ANSWER – De Morgan's laws

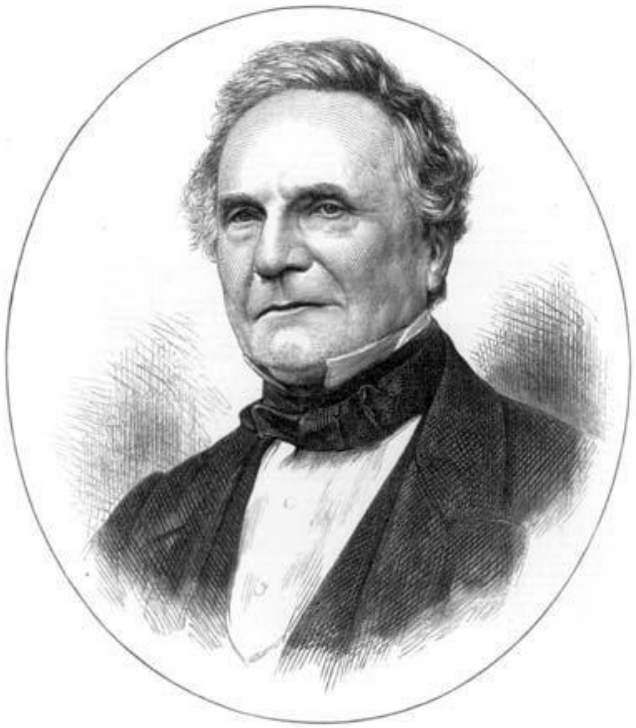
- $\neg (X \wedge Y) = \neg X \vee \neg Y$
- $\neg (X \vee Y) = \neg X \wedge \neg Y$

X	Y	$X \wedge Y$	$\neg (X \wedge Y)$	$X \vee Y$	$\neg (X \vee Y)$	$\neg X$	$\neg Y$	$\neg X \wedge \neg Y$	$\neg X \vee \neg Y$
0	0	0	1	0	1	1	1	1	1
0	1	0	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	0	1
1	1	1	0	1	0	0	0	0	0

A mathematical lineage to the first computer

Augustus De Morgan

Ada Lovelace

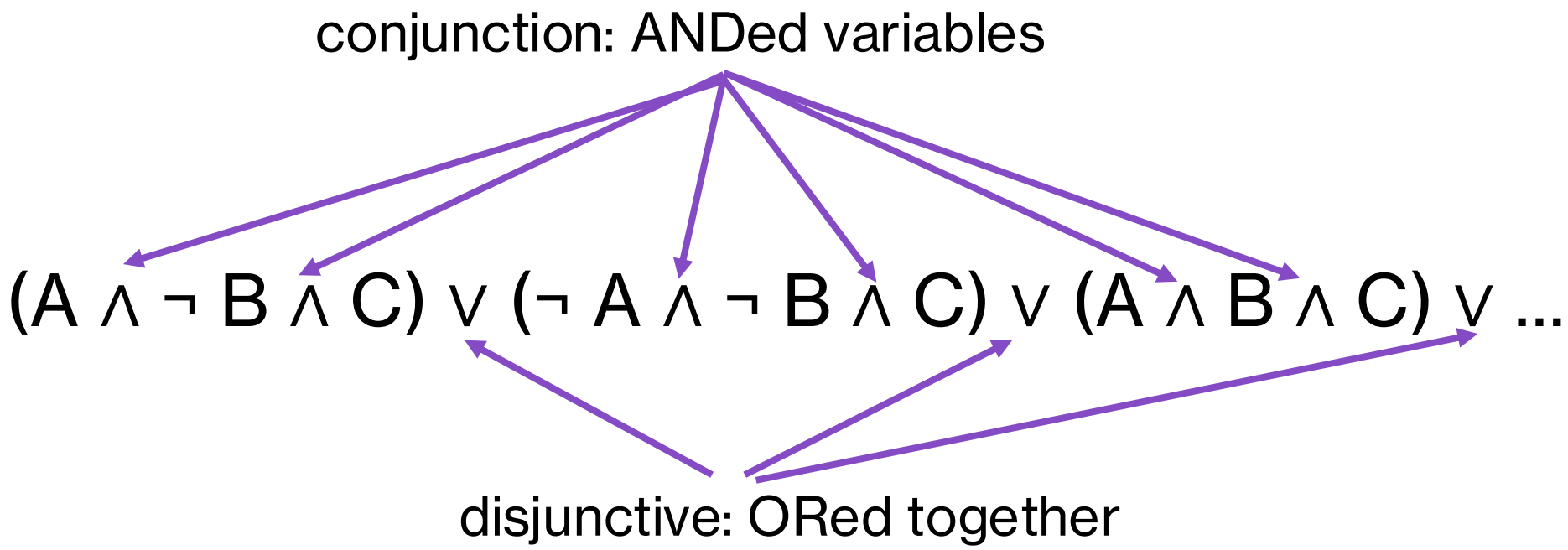


Charles Babbage

DNFs and CNFs

Disjunctive normal form (DNF)

- **Theorem (Boole, 1847):** Any Boolean function can be represented as a disjunction (ORs) of conjunctions (ANDs) of its arguments and their negation (NOT). This is also known as **disjunctive normal form** (DNF).



Disjunctive normal form (DNF)

One way this can happen is by using “laws”, specifically:

$$
\begin{aligned}
(\neg\neg x) &\rightsquigarrow x \\
(\neg(x \vee y)) &\rightsquigarrow ((\neg x) \wedge (\neg y)) \\
(\neg(x \wedge y)) &\rightsquigarrow ((\neg x) \vee (\neg y)) \\
(x \wedge (y \vee z)) &\rightsquigarrow ((x \wedge y) \vee (x \wedge z)) \\
((x \vee y) \wedge z) &\rightsquigarrow ((x \wedge z) \vee (y \wedge z))
\end{aligned}
$$

double negation

De Morgan’s laws

distributive law

Minterms

Boolean expressions consisting of the conjunction (AND) of all variables of a Boolean function. Each variable will appear exactly once in either its true (if the variable was 1) or complemented form (if the variable was 0). The minterm is formed using the values of the variables which make the value of the minterm equal to 1.

Row number	X	Y	Z	Minterm
0	0	0	0	$m_0 = \neg X \wedge \neg Y \wedge \neg Z$
1	0	0	1	$m_1 = \neg X \wedge \neg Y \wedge Z$
2	0	1	0	$m_2 = \neg X \wedge Y \wedge \neg Z$
3	0	1	1	$m_3 = \neg X \wedge Y \wedge Z$
4	1	0	0	$m_4 = X \wedge \neg Y \wedge \neg Z$
5	1	0	1	$m_5 = X \wedge \neg Y \wedge Z$
6	1	1	0	$m_6 = X \wedge Y \wedge \neg Z$
7	1	1	1	$m_7 = X \wedge Y \wedge Z$

Minterms

Boolean expressions consisting of the conjunction (AND) of all variables of a Boolean function. Each variable will appear exactly once in either its true (if the variable was 1) or complemented form (if the variable was 0). The minterm is formed using the values of the variables which make the value of the minterm equal to 1.

Row number	X	Y	Z	$m_3 = \neg X \wedge Y \wedge Z$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	**0**	**1**	**1**	**1**
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

Example – MAJ function

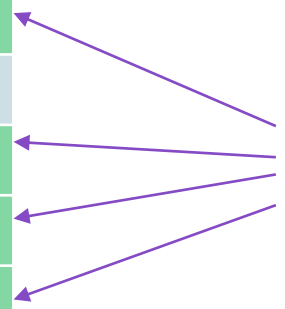
- MAJ($X_1, X_2, \ldots, X_n$) = 1 if (strictly) more arguments are 1 than 0, 0 otherwise.

X	Y	Z	MAJ(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Example – DNF of MAJ function

- MAJ($X_1, X_2, \ldots, X_n$) = 1 if (strictly) more arguments are 1 than 0, 0 otherwise.

X	Y	Z	MAJ(X,Y,Z)	Minterm
0	0	0	0	$m_0 = \neg X \wedge \neg Y \wedge \neg Z$
0	0	1	0	$m_1 = \neg X \wedge \neg Y \wedge Z$
0	1	0	0	$m_2 = \neg X \wedge Y \wedge \neg Z$
0	**1**	**1**	**1**	$\mathbf{m_3 = \neg X \wedge Y \wedge Z}$
1	0	0	0	$m_4 = X \wedge \neg Y \wedge \neg Z$
1	**0**	**1**	**1**	$\mathbf{m_5 = X \wedge \neg Y \wedge Z}$
1	**1**	**0**	**1**	$\mathbf{m_6 = X \wedge Y \wedge \neg Z}$
1	**1**	**1**	**1**	$\mathbf{m_7 = X \wedge Y \wedge Z}$



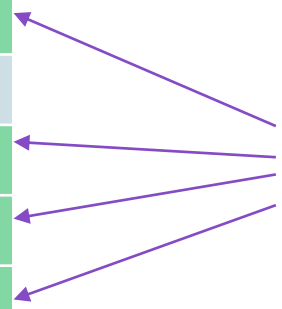
OR these together

- Isolate the rows where the output is equal to 1 and then connect with a disjunction (OR) the corresponding minterms.

Example – DNF of MAJ function

- MAJ($X_1, X_2, \ldots, X_n$) = 1 if (strictly) more arguments are 1 than 0, 0 otherwise.

X	Y	Z	MAJ(X,Y,Z)	Minterm
0	0	0	0	$m_0 = \neg X \wedge \neg Y \wedge \neg Z$
0	0	1	0	$m_1 = \neg X \wedge \neg Y \wedge Z$
0	1	0	0	$m_2 = \neg X \wedge Y \wedge \neg Z$
0	**1**	**1**	**1**	$\mathbf{m_3 = \neg X \wedge Y \wedge Z}$
1	0	0	0	$m_4 = X \wedge \neg Y \wedge \neg Z$
1	**0**	**1**	**1**	$\mathbf{m_5 = X \wedge \neg Y \wedge Z}$
1	**1**	**0**	**1**	$\mathbf{m_6 = X \wedge Y \wedge \neg Z}$
1	**1**	**1**	**1**	$\mathbf{m_7 = X \wedge Y \wedge Z}$



OR these together

- Isolate the rows where the output is equal to 1 and then connect with a disjunction (OR) the corresponding minterms.
- MAJ(X, Y, Z) = $m_3 \vee m_5 \vee m_6 \vee m_7$ = (¬X ∧ Y ∧ Z) ∨ (X ∧ ¬Y ∧ Z) ∨ (X ∧ Y ∧ ¬Z) ∨ (X ∧ Y ∧ Z)

PRACTICE TIME – DNF of ODD function

ODD($X_1, X_2, \ldots, X_n$) = 1 if an odd number of arguments is 1, 0 otherwise

X	Y	Z	ODD(X,Y,Z)	Minterm
0	0	0	0	$m_0 =$
0	0	1	1	$m_1 =$
0	1	0	1	$m_2 =$
0	1	1	0	$m_3 =$
1	0	0	1	$m_4 =$
1	0	1	0	$m_5 =$
1	1	0	0	$m_6 =$
1	1	1	1	$m_7 =$

ODD(X, Y, Z) =

PRACTICE TIME – DNF of ODD function

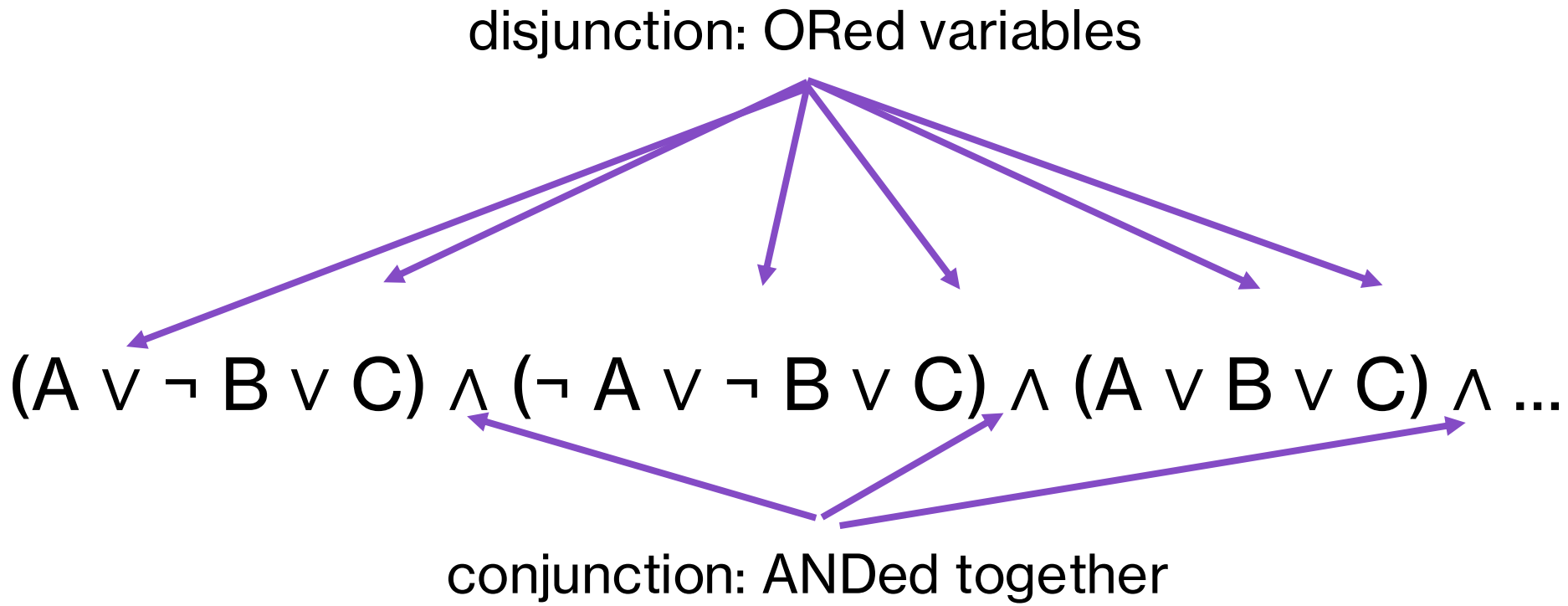
ODD($X_1, X_2, \ldots, X_n$) = 1 if an odd number of arguments is 1, 0 otherwise

X	Y	Z	ODD(X,Y,Z)	Minterm
0	0	0	0	$m_0 = \neg X \wedge \neg Y \wedge \neg Z$
0	**0**	**1**	**1**	$\boldsymbol{m_1 = \neg X \wedge \neg Y \wedge Z}$
0	**1**	**0**	**1**	$\boldsymbol{m_2 = \neg X \wedge Y \wedge \neg Z}$
0	1	1	0	$m_3 = \neg X \wedge Y \wedge Z$
1	**0**	**0**	**1**	$\boldsymbol{m_4 = X \wedge \neg Y \wedge \neg Z}$
1	0	1	0	$m_5 = X \wedge \neg Y \wedge Z$
1	1	0	0	$m_6 = X \wedge Y \wedge \neg Z$
1	**1**	**1**	**1**	$\boldsymbol{m_7 = X \wedge Y \wedge Z}$

ODD(X, Y, Z) = $m_1 \vee m_2 \vee m_4 \vee m_7$ = (¬X ∧ ¬Y ∧ Z) ∨ (¬X ∧ Y ∧ ¬Z) ∨ (X ∧ ¬Y ∧ ¬Z) ∨ (X ∧ Y ∧ Z)

Conjunctive normal form (CNF)

- **Theorem (Boole, 1847):** Any Boolean function can be represented as a conjunction (AND)s of disjunction (ORs) of its arguments and their negation (NOT). This is also known as **conjunctive normal form** (CNF).



Maxterms

Boolean expressions consisting of the disjunction (OR) of all variables of a Boolean function. Each variable will appear exactly once in either its true (if the variable was 0) or complemented form (if the variable was 1). The maxterm is formed using the values of the variables which make the value of the maxterm equal to 0.

Row number	X	Y	Z	Minterm	Maxterm
0	0	0	0	$m_0 = \neg X \wedge \neg Y \wedge \neg Z$	$M_0 = X \vee Y \vee Z$
1	0	0	1	$m_1 = \neg X \wedge \neg Y \wedge Z$	$M_1 = X \vee Y \vee \neg Z$
2	0	1	0	$m_2 = \neg X \wedge Y \wedge \neg Z$	$M_2 = X \vee \neg Y \vee Z$
3	0	1	1	$m_3 = \neg X \wedge Y \wedge Z$	$M_3 = X \vee \neg Y \vee \neg Z$
4	1	0	0	$m_4 = X \wedge \neg Y \wedge \neg Z$	$M_4 = \neg X \vee Y \vee Z$
5	1	0	1	$m_5 = X \wedge \neg Y \wedge Z$	$M_5 = \neg X \vee Y \vee \neg Z$
6	1	1	0	$m_6 = X \wedge Y \wedge \neg Z$	$M_6 = \neg X \vee \neg Y \vee Z$
7	1	1	1	$m_7 = X \wedge Y \wedge Z$	$M_7 = \neg X \vee \neg Y \vee \neg Z$

Maxterms

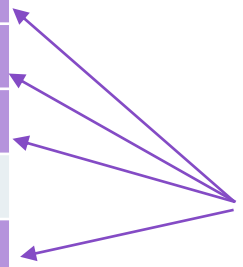
Boolean expressions consisting of the disjunction (OR) of all variables of a Boolean function. Each variable will appear exactly once in either its true (if the variable was 0) or complemented form (if the variable was 1). The maxterm is formed using the values of the variables which make the value of the maxterm equal to 0.

Row number	X	Y	Z	$M_5 = \neg X \vee Y \vee \neg Z$
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	**1**	**0**	**1**	**0**
6	1	1	0	1
7	1	1	1	1

Example – CNF of MAJ function

- MAJ($X_1, X_2, \ldots, X_n$) = 1 if (strictly) more arguments are 1 than 0, 0 otherwise.

X	Y	Z	MAJ(X,Y,Z)	Maxterm
0	**0**	**0**	**0**	$\boldsymbol{M_0 = X \vee Y \vee Z}$
0	**0**	**1**	**0**	$\boldsymbol{M_1 = X \vee Y \vee \neg Z}$
0	**1**	**0**	**0**	$\boldsymbol{M_2 = X \vee \neg Y \vee Z}$
0	1	1	1	$M_3 = X \vee \neg Y \vee \neg Z$
1	**0**	**0**	**0**	$\boldsymbol{M_4 = \neg X \vee Y \vee Z}$
1	0	1	1	$M_5 = \neg X \vee Y \vee \neg Z$
1	1	0	1	$M_6 = \neg X \vee \neg Y \vee Z$
1	1	1	1	$M_7 = \neg X \vee \neg Y \vee \neg Z$



AND these together

- Isolate the rows where the output is equal to 0 and then connect with a conjunction (AND) the corresponding maxterms.
- MAJ(X, Y, Z) = $M_0 \wedge M_1 \wedge M_2 \wedge M_4$ = (X ∨ Y ∨ Z) ∧ (X ∨ Y ∨ ¬Z) ∧ (X ∨ ¬Y ∨ Z) ∧ (¬X ∨ Y ∨ Z)

PRACTICE TIME – CNF of ODD function

ODD($X_1, X_2, \ldots, X_n$) = 1 if an odd number of arguments is 1, 0 otherwise

X	Y	Z	ODD(X,Y,Z)	Maxterm
0	0	0	0	$M_0 =$
0	0	1	1	$M_1 =$
0	1	0	1	$M_2 =$
0	1	1	0	$M_3 =$
1	0	0	1	M_4
1	0	1	0	$M_5 =$
1	1	0	0	$M_6 =$
1	1	1	1	$M_7 =$

ODD(X, Y, Z) =

ANSWER – CNF of ODD function

ODD($X_1, X_2, \ldots, X_n$) = 1 if an odd number of arguments is 1, 0 otherwise

X	Y	Z	ODD(X,Y,Z)	Maxterm
0	0	0	0	$M_0 = X \vee Y \vee Z$
0	0	1	1	$M_1 = X \vee Y \vee \neg Z$
0	1	0	1	$M_2 = X \vee \neg Y \vee Z$
0	1	1	0	$M_3 = X \vee \neg Y \vee \neg Z$
1	0	0	1	$M_4 = \neg X \vee Y \vee Z$
1	0	1	0	$M_5 = \neg X \vee Y \vee \neg Z$
1	1	0	0	$M_6 = \neg X \vee \neg Y \vee Z$
1	1	1	1	$M_7 = \neg X \vee \neg Y \vee \neg Z$

ODD(X, Y, Z) $= M_0 \wedge M_3 \wedge M_5 \wedge M_6 =$ (X ∨ Y ∨ Z) ∧ (X ∨ ¬Y ∨ ¬Z) ∧ (¬X ∨ Y ∨ ¬Z) ∧ (¬X ∨ ¬Y ∨ Z)

Gates

CPU: central processing unit

CPU

Inside a CPU

Gates

Inside a CPU

Gates

NOT

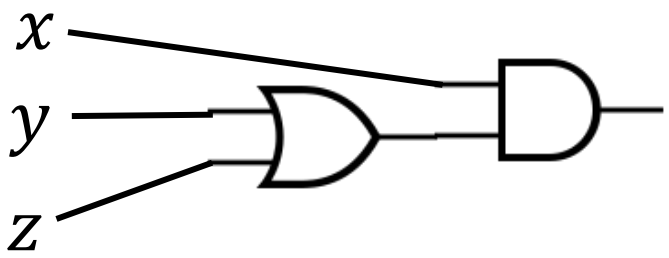
AND

OR

XOR

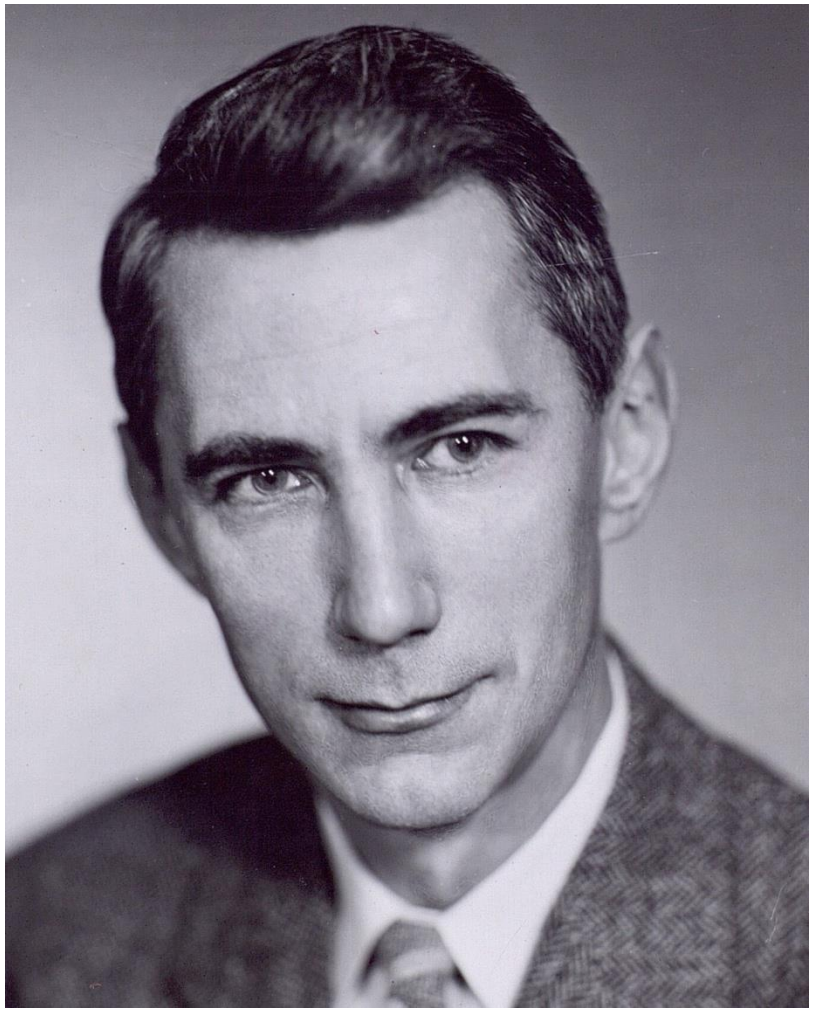
Gates

$$x \wedge (y \vee z) =$$



From Boolean Algebra to Circuits

- In 1937, **Claude Shannon** translated Boolean logic into a physical format with electronics.
- This work became the foundation of **circuit design** and made it possible to design the computers we know today.
- Shannon also introduced many of the core ideas of abstraction, encoding, and compression we use today.



Claude Shannon

Practice Problems

Practice Problems – Problem 1

- You are given the following truth table for a Boolean function with three variables, A, B, and C, and an output column F.
- What are the disjunctive and conjunctive normal forms based on this truth table?

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Practice Problems – Answer 1

- Disjunctive normal form:

  $F = (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge \neg C)$

- Conjunctive normal form:

  $F = (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)$

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0