11.3 Boolean algebra and equations

Boolean algebra

In 1847, mathematician ***George Boole*** developed an algebra to capture human logic as mathematical equations. A later section will show how Boolean algebra became, without Boole's knowledge, the foundation of digital circuit design.

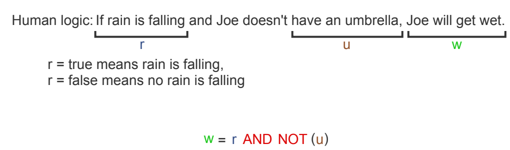
In algebra, a ***variable*** is a symbol that represents a value. ***Boolean algebra*** is an algebra whose only values are true or false, and whose operators are AND, OR, and NOT. AND, OR, and NOT are known as ***logic operators***.

Table 11.3.1: Logic operators.

Operator	Description
AND	AND outputs true only if both inputs are true.
OR	OR outputs true if either, or both, inputs are true.
NOT	NOT outputs true if the input is false. NOT outputs false if the input is true.

PARTICIPATION ACTIVITY 11.3.1: Boolean algebra can capture human logic as math equations.

Start 2x speed



PARTICIPATION ACTIVITY 11.3.2: Capturing human logic as a Boolean equation.

Inputs: h indicates a baby is hungry, s indicates sleepy, m indicates a mother is holding the baby.
Output: c indicates the baby will cry.
Match the equation to the human logic.

c = true **c = s AND NOT(m)** **c = h OR s** **c = h AND s** **c = h**

A baby will cry only if hungry

A baby will cry if hungry or sleepy

A baby will cry only if sleepy and not being held by a mother

A baby will cry only if both hungry and sleepy

A baby will always cry

Reset

PARTICIPATION ACTIVITY 11.3.3: Boolean algebra.

Inputs: b means the battery works, g means there's enough gas.
Output: c means a car will start.
Indicate whether the equation matches the human logic.

1) The car will start only if the battery works and there's enough gas.

c = b OR g

- Yes
- No

2) The car will never start.

c = false

- Yes
- No

3) The car will start if there's enough gas; the battery doesn't matter.

c = g

- Yes
- No

4) The car will not start if there's not enough gas.

c = NOT(g)

- Yes
- No

A Boolean expression is evaluated by evaluating parts and combining. NOT is evaluated first. AND is evaluated before OR.

PARTICIPATION ACTIVITY 11.3.4: Evaluating Boolean expressions.

Start 2x speed

Expression: a AND NOT(b)

Given: a = true, b = false

a AND NOT(b)
true AND NOT(false)
true AND true
true

Given: a = true, b = true

a AND NOT(b)
true AND NOT(true)
true AND false
false

PARTICIPATION ACTIVITY 11.3.5: Evaluating Boolean expressions.

Evaluate. Assume:
a = false
b = true
Type answers as: true or false

1) a AND b

Check Show answer

2) a OR b

Check Show answer

3) a AND NOT(b)

Check Show answer

4) NOT(a) AND NOT(b)

Check Show answer

5) (a AND b) OR NOT(a)
= (false) OR ?
Type only the ? part

Check Show answer

6) (a AND b) AND NOT(a)
= (?) AND ...
Type only the ? part

Check Show answer

7) (a AND b) OR NOT(a)

Check Show answer

Symbols

Note: Mathematicians use symbols like ∧, ∨, and ¬ for AND, OR, and NOT, respectively. This section uses the words for simple understanding; later sections use common digital-designer shorthand notation.

Boolean equations

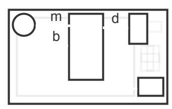
Boolean algebra was developed in the 1800s for purposes unrelated to digital circuits. In 1938, Claude Shannon applied Boolean algebra to the design of digital circuits. Previously, designing circuits directly as switches was hard and error-prone. Shannon showed that building and using logic gates (AND/OR/NOT) allowed use of Boolean algebra's properties to more-easily and correctly design complex circuits.

In digital circuits, 1 (the high voltage value) is Boolean algebra's true, and 0 is false.

A ***Boolean equation*** has a Boolean variable (left), an equal sign, and a Boolean expression (right), defining the left variable's value based on the right variables' values. A Boolean equation can describe a digital circuit, with the output on the left and the inputs on the right.

PARTICIPATION ACTIVITY 11.3.6: A Boolean equation can describe digital circuit behavior: Microwave door example.

Start 2x speed



Inputs: m: motor is operating
b: button is pressed

Output: d: open door

Goal: Open door if user presses door open button, but only if motor is not operating

d = b AND NOT (m)

Digital designers commonly use a shorthand notation for Boolean operators, shown below.

Table 11.3.2: Digital-designer shorthand notation for Boolean operators.

Operation	Shorthand	Notes
a AND b	ab	Intentionally looks like multiplication. Known as *abutment*.
a OR b	a + b	Intentionally looks like addition.
NOT(a)	a'	a' is also called the ***complement*** of a.

Example: a AND NOT(b) becomes ab'.

PARTICIPATION ACTIVITY 11.3.7: Digital-designer shorthand notation for Boolean operators.

Directly translate each to an expression using digital-designer shorthand notation.
Note: This material's activities require straightforward answers. Ex: For a AND b, type ab. All variations like ba or (ab) cannot be accounted for.

1) a OR b

Check Show answer

2) (a AND b) OR c

Check Show answer

3) NOT(a)

Check Show answer

4) NOT(a) AND b

Check Show answer

5) NOT(a) AND NOT(b)

Check Show answer

6) NOT(ab)

Check   Show answer

7) NOT(a OR b)

Check   Show answer

PARTICIPATION ACTIVITY   11.3.8: Boolean equations for digital circuits.

A digital system has the following inputs and outputs:

Inputs: d: door is open, w: window is open, e: alarm is enabled, n: time-of-day is night
Output: s: sounds alarm

Select the Boolean equation that describes the indicated goal.

1) Goal: Sound alarm only if door is open and alarm is enabled.
   - e = sd
   - s = e + d
   - s = ed

2) Goal: Sound alarm if alarm is enabled, and also the window is open or the door is open.
   - s = ewd
   - s = e(w + d)
   - s = e + w + d

3) Goal: Sound alarm if alarm is enabled and window is open at night.
   - s = ewn
   - s = e(w + n)
   - s = e + w + n

Logic circuits

*Digital circuits are sometimes called **logic circuits** due to the roots in Boolean algebra's logic operations of AND, OR, and NOT.*

Boolean functions

In Boolean algebra, a ***function*** is a relation of inputs' values to an output's values. A function can be described in various ways:

- As English: When inputs a, b are both 1's, the output y is 1. Else, y is 0.
- As an equation: y = ab
- As a table:

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

- As a circuit, as a drawing, a K-map (introduced later), etc.

Distinguishing the words "expression", "equation", and "function" will be useful.

Table 11.3.3: Expressions, equations, and functions.

Item	Notation	Notes
Expression	ab	An expression lacks an equal sign, and involves input variables.
Equation	y = ab	An equation has an =, with expressions of input variables on the right, and an output variable on the left. (In general math, both sides of an equation can be expressions, but in this material, the left side is usually just an output variable.)
Function	Various	A relation of input values to output values. Can be represented in various ways: equation, table, circuit, etc. A function may have more than one input, but has only one output.

PARTICIPATION ACTIVITY   11.3.9: Expressions, equations, and functions.

c, d are inputs, y is an output.

1) Is c + d a function?

- Yes
- No

2) Is y = c + d a function?

- Yes
- No

3) Is y = c + d an equation?

- Yes
- No

4) Does this table represent a function?

c	d	y
0	0	0
0	1	1
1	0	1
1	1	1

- Yes
- No

5) Is the following a function?

y is 1 if either or both of c, d is 1. Else, y is 0.

- Yes
- No

6) Is the following a function?

Output y is 1 if both inputs are 0's, otherwise y is 0. Also, if both inputs are 1's, y is 1.

- Yes
- No

7) Is the following a function?

Output y is 1 if both inputs are 0's.

- Yes
- No

Provide feedback on this section