Binary Logic

Lecture 4 – Monday March 13, 2017

Outline

- Boolean Logic
- Boolean Algebra
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- Logic Gates
- Logic Circuits
- Summary

Outline

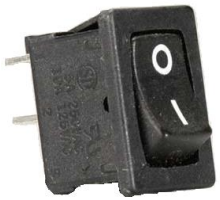
- **Boolean Logic**
- Boolean Algebra
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- Logic Gates
- Logic Circuits
- Summary

Boolean Logic

- Boolean logic is a branch of mathematics that deals with rules for manipulating two logical values **true** and **false**.
- It was named after George Boole.

George Boole (1815-1864)
British mathematician and philosopher

- Boolean logic is simply a way of comparing individual bits.
- Why is Boolean logic so relevant to computers?



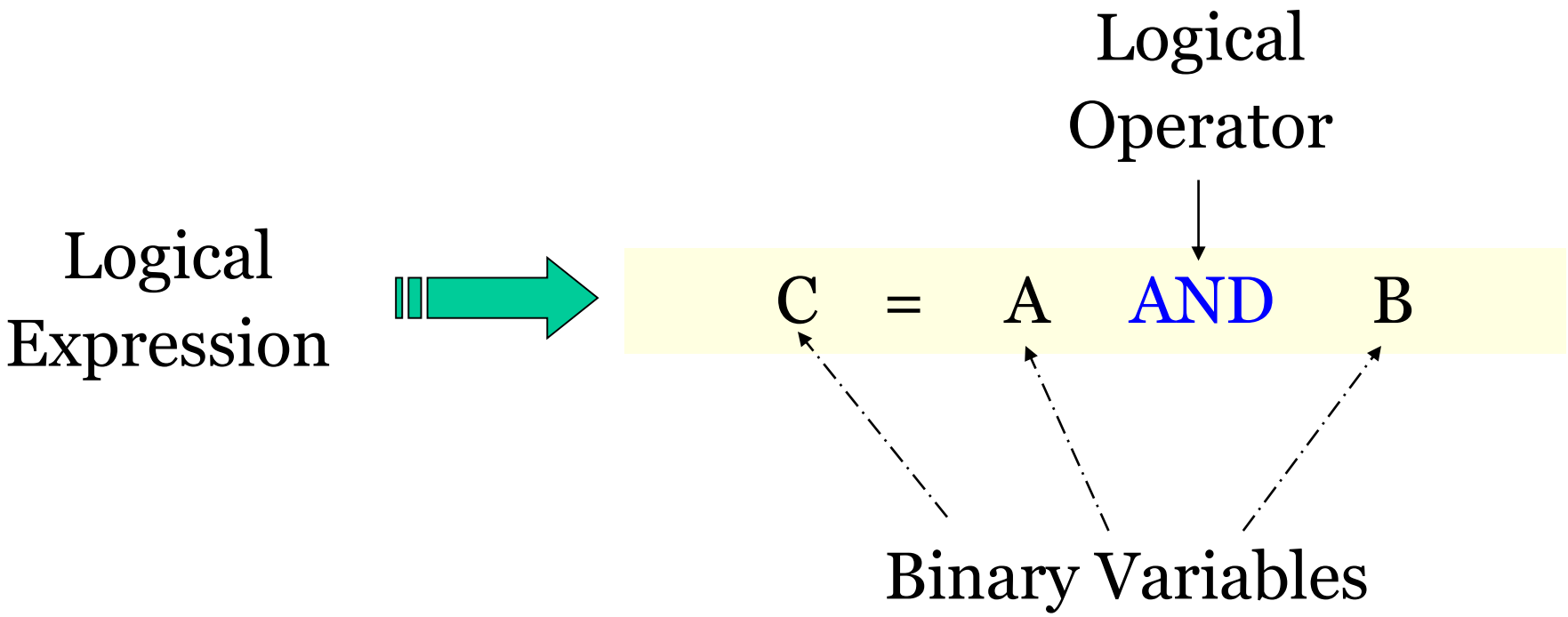
- Straightforward mapping to binary digits!
- Binary digit values can be thought of as: ON/OFF, High/Low, Yes/No, 1/0

Outline

- Boolean Logic
- **Boolean Algebra**
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- Logic Gates
- Logic Circuits
- Summary

Boolean Algebra

- An algebra in which elements have one of two values and the algebraic operations defined on the set are logical OR, a type of addition, and logical AND, a type of multiplication.

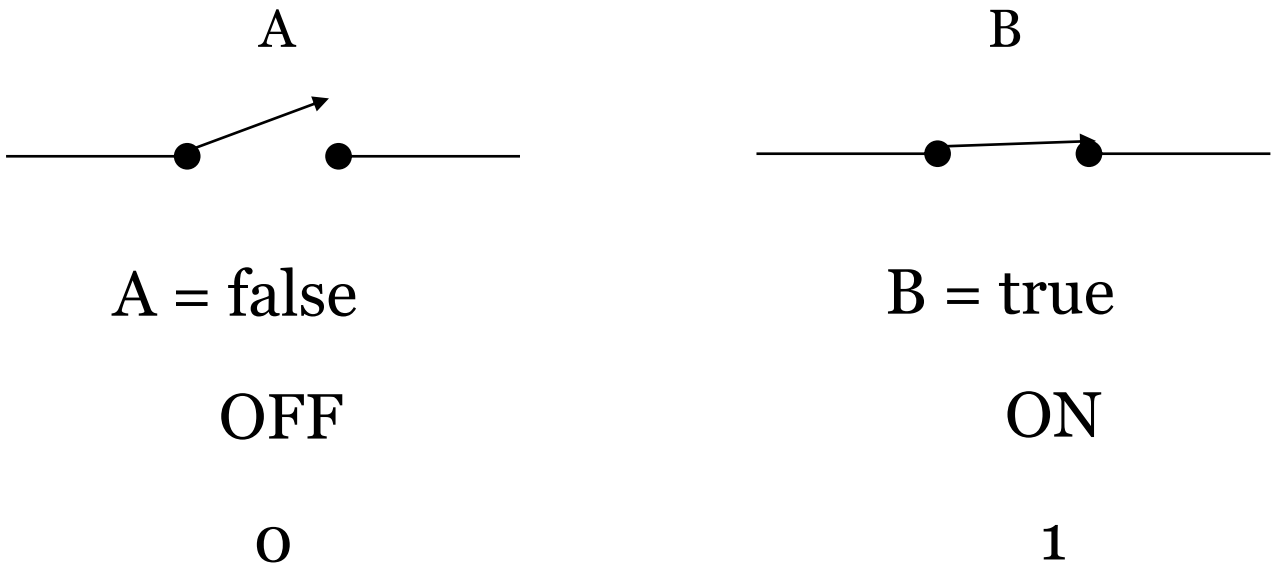


Outline

- Boolean Logic
- Boolean Algebra
- **Binary Variables**
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- Logic Gates
- Logic Circuits
- Summary

Binary Variables

- Binary variables take on one of two values, **true** or **false**.
- Variable identifiers: A, B, C, X, Y, ….



Outline

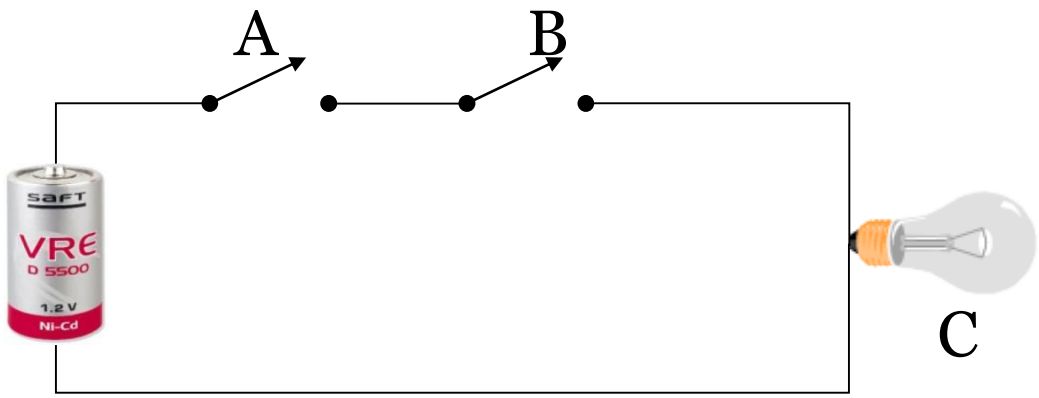
- Boolean Logic
- Boolean Algebra
- Binary Variables
- **Binary Operators**
  - **Logical AND**
  - **Logical OR**
  - **Logical Negation**
- Boolean Expressions
- Logic Gates
- Logic Circuits
- Summary

Binary Operators

- Logical AND
- Logical OR
- Logical Negation (NOT).

Binary Operators: Logical AND

- AND is denoted by a dot (·)
- Consider two logic variables A and B and the result is C.
- C is true if and only if A is true AND B is true



In order for current to flow, both switches must be closed

C = A.B

Truth Table

Inputs		Output
A	B	C=A.B
0	0	0
0	1	0
1	0	0
1	1	1

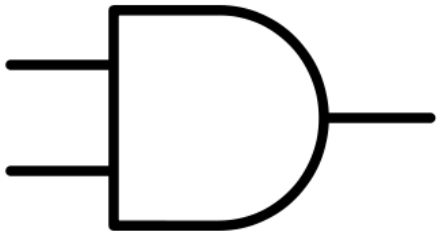
Binary Operators: Logical AND

ON/OFF

Person Sensor

Alarm Switch

ON/OFF



ON/OFF

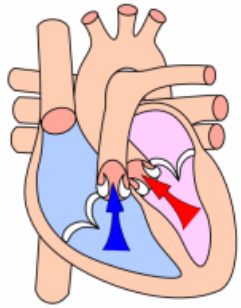
Burglar Alarm

If both the Person Sensor **AND** the Alarm Switch are on then the Burglar Alarm is activated

Binary Operators: Logical AND

- (A AND B) yields true only if both A and B are true.

Example: Adult blood pressure is considered normal at 120/80 where the first number is the systolic pressure and the second is the diastolic pressure.



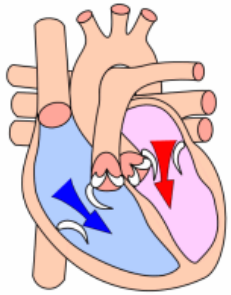
Systolic Pressure

A = True if Systolic Pressure = 120

B = True if Diastolic Pressure = 80

C = True $\Rightarrow$ Blood pressure is normal

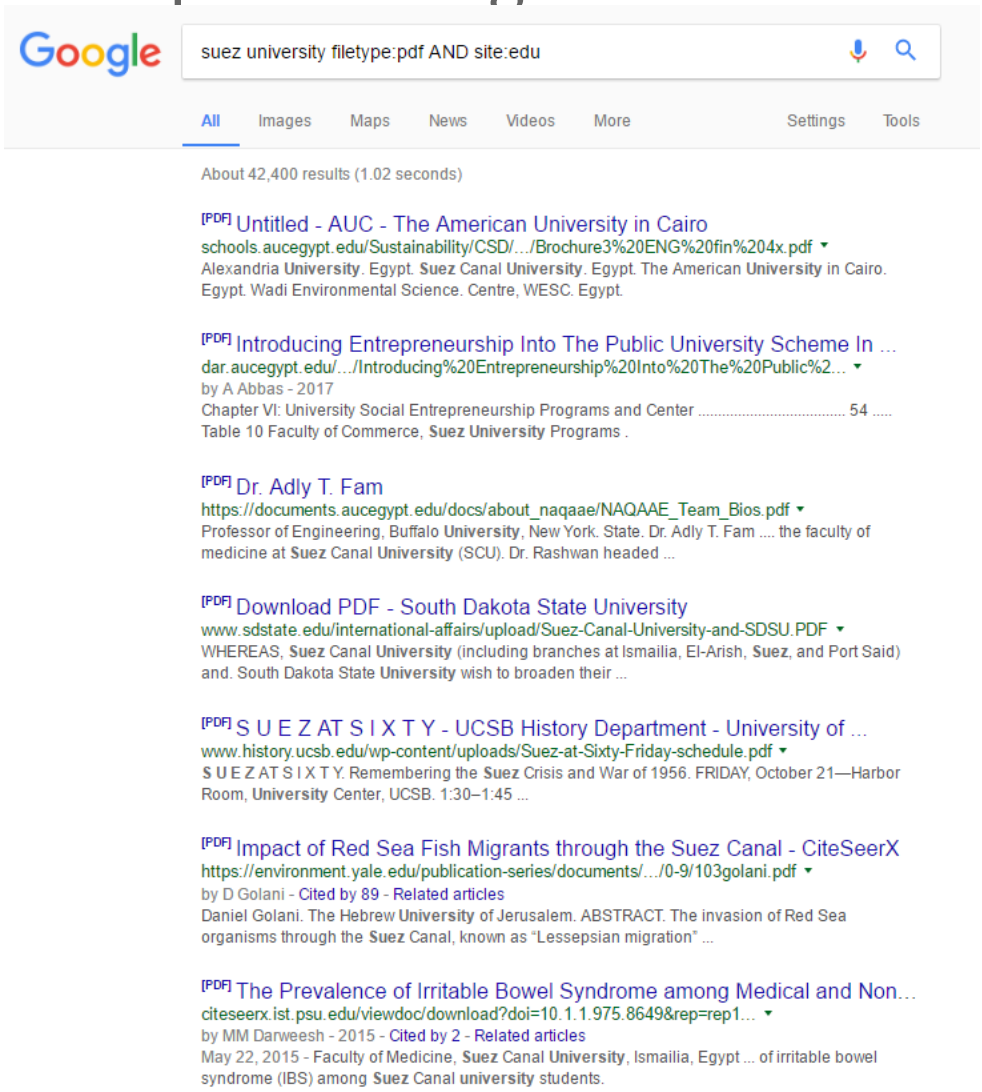
C = A.B



Diastolic Pressure

Binary Operators: Logical AND

- Example-2: Google Search



operator:parameter

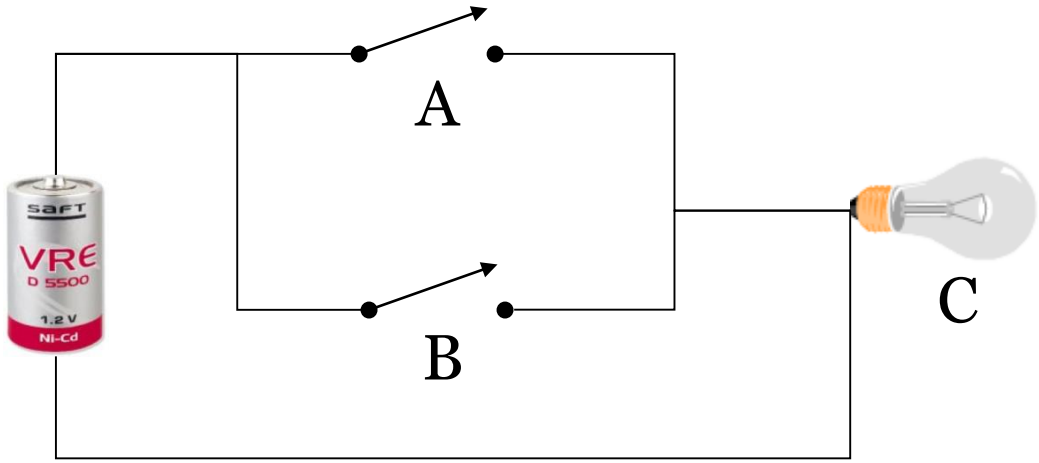
filetype:pdf will search for pdf only.

site:edu will search all site in edu top domain.

suez university filetype:pdf AND site:edu will search for pdf and all site in edu top domain.

Binary Operators: Logical OR

- OR is denoted by a plus (+)
- Consider two logic variables A and B and the result is C.
- C is true if A is true OR B is true



Current flows if either switch is closed

C = A+B

Truth Table

Inputs		Output
A	B	C = A+B
0	0	0
0	1	1
1	0	1
1	1	1

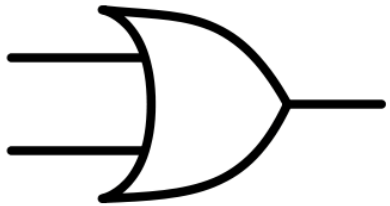
Binary Operators: Logical OR

ON/OFF

Front Doorbell Switch

Back Doorbell Switch

ON/OFF



ON/OFF

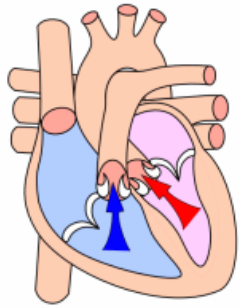
Doorbell

If either the Front Doorbell Switch **OR** the Back Doorbell Switch is pressed then the Doorbell rings

Binary Operators: Logical OR

- (A OR B) yields true only if either A or B, or both are true.

Example: Adult blood pressure is considered normal at 120/80 where the first number is the systolic pressure and the second is the diastolic pressure.



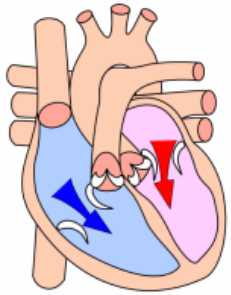
Systolic Pressure

A = True if Systolic Pressure ≠ 120

B = True if Diastolic Pressure ≠ 80

C = True ⇒ Blood pressure is abnormal

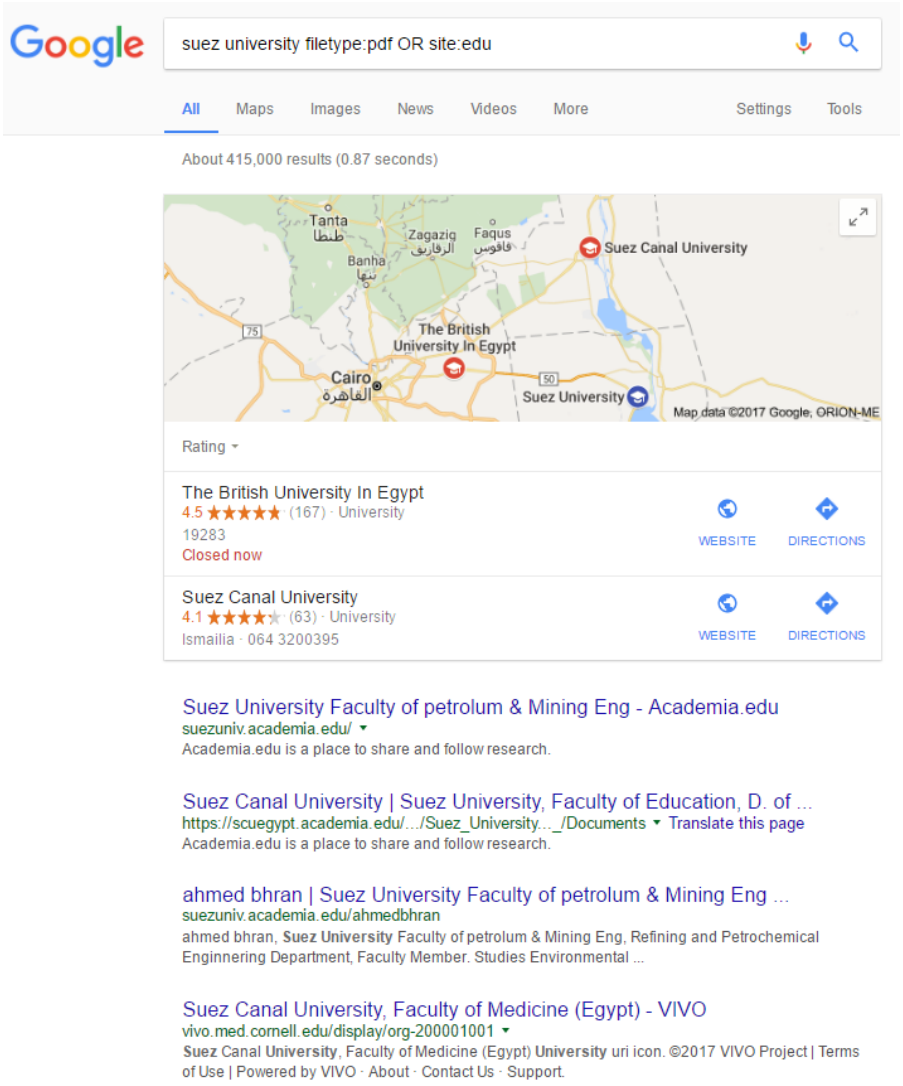
C = A+B



Diastolic Pressure

Binary Operators: Logical OR

- Example-2: Google Search



operator:parameter

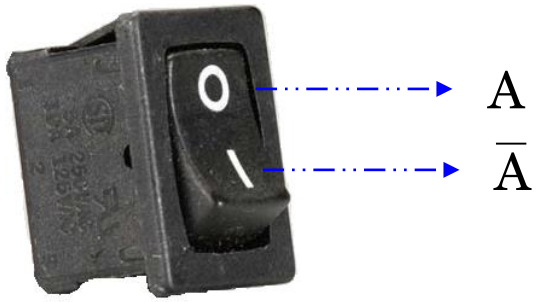
filetype:pdf will search for pdf only.

site:edu will search all site in edu top domain.

suez university filetype:pdf OR site:edu will search for pdf or all site in edu top domain.

Binary Operators: Logical Negation

- NOT is denoted by a bar ($\overline{\ }$) over, a single quote mark (') after, or ~ before the variable.
- Consider a logic variable A and the result is C.
- C is true if A is false and vice versa.
- $C = \overline{A}$ or $C = A'$ or $C = \sim A$

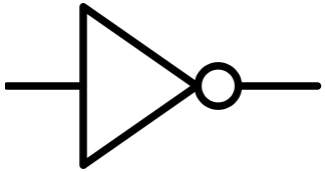


Truth Table

Input	Output
A	$C = \overline{A}$
0	1
1	0

Binary Operators: Logical Negation

Temperature detector (Above 20°C)
ON/OFF



Central Heating
ON/OFF

If the temperature is above 20°C then the Central Heating is switched off.

If the temperature is below 20°C then the Central Heating is switched on

Binary Operators: Logical Negation

- Inverts its operand.

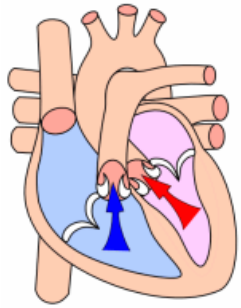
Example: Adult blood pressure is considered normal at 120/80 where the first number is the systolic pressure and the second is the diastolic pressure.

A = True if Systolic Pressure = 120

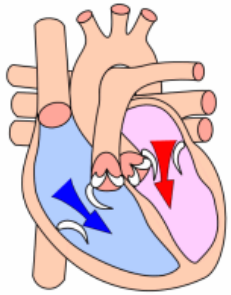
$\overline{A}$ = False $\Rightarrow$ Systolic Pressure $\neq$ 120

B = True if Diastolic Pressure = 80

$\overline{B}$ = False $\Rightarrow$ Diastolic Pressure $\neq$ 80



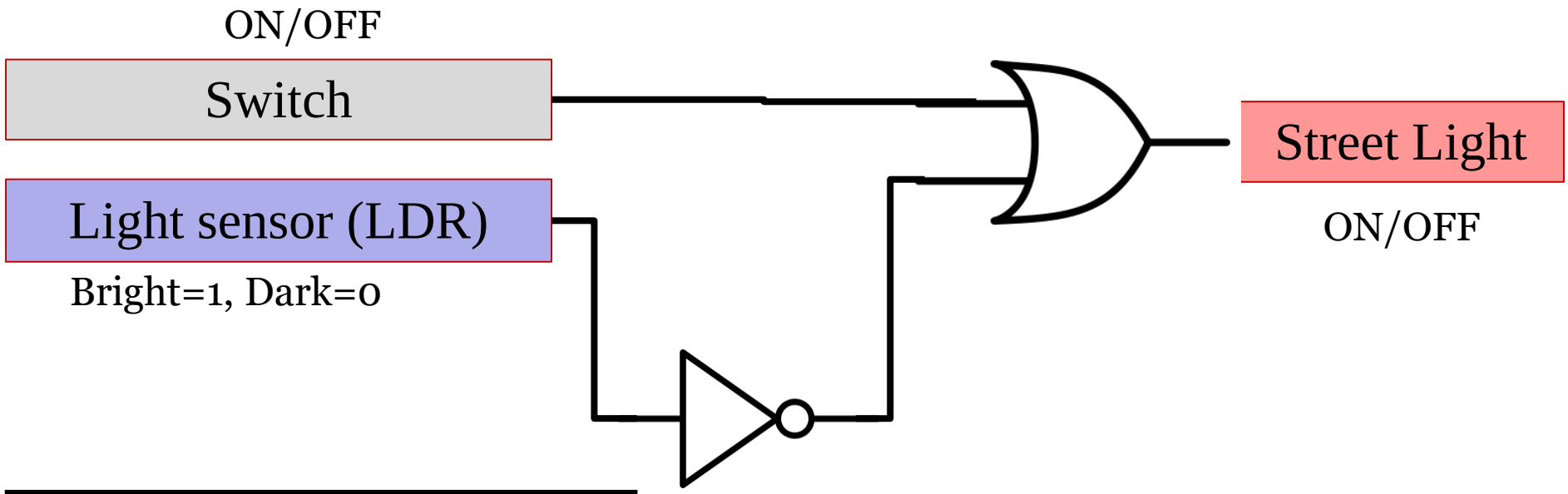
Systolic Pressure



Diastolic Pressure

Logic Gates: NOT Gate

- Street Light



Switch	LDR	Street light
On	Dark	On
On	Bright	On
Off	Dark	On
Off	Bright	Off

Logic Gates

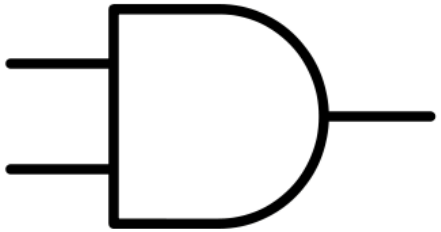
- **Fire alarm security system**

Hot=1, Cold=0

Heat Detector

Smoke Detector

Smoke=1, No smoke=0



ON/OFF

Siren

Exercise: The system shown is not functioning well. Suggest a modification in order to make the system works.

Outline

- Boolean Logic
- Boolean Algebra
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- **Boolean Expressions**
- Logic Gates
- Logic Circuits
- Summary

Boolean Expressions

- Basic operations:

  $C = A.B$ is read "C is equal to A and B."

  $z = x + y$ is read "z is equal to x OR y."

  $D = \overline{A}$ is read "D is equal to NOT A."

- Using the basic operations, we can form more complex expressions:

$$Z = (A.B + C) + \overline{X}.Y$$

If A=True, B=False, C=True, X=True, Y=False.

Z=(True.False+True)+False.False

=(False+True)+False=True+False=True

Boolean Expressions

- Example

Assuming that x=-10, y=50, and z=60 determine the value of the following Boolean expression:

(0<x<50)AND(50<y<100)OR([y-x]=z)

(0<x<50) ⇒ (0<[-10]<50) ⇒ FALSE

(50<y<100) ⇒ (50<50<100) ⇒ FALSE

([50-(-10)]=60) ⇒ (60=60) ⇒ TRUE

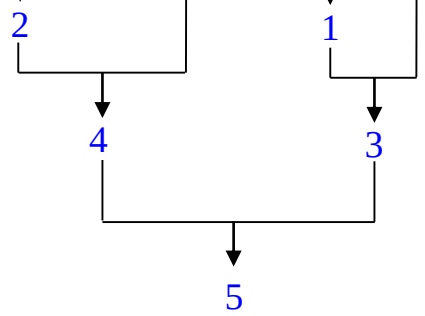
FALSE AND FALSE ⇒ FALSE

FALSE OR TRUE ⇒ TRUE

Boolean Expressions: Operator Precedence

- NOT has the highest precedence, followed by AND, and then OR.
- All higher-precedence operators are evaluated before any lower-precedence operators.
- Operators at the same precedence are evaluated left-to-right.

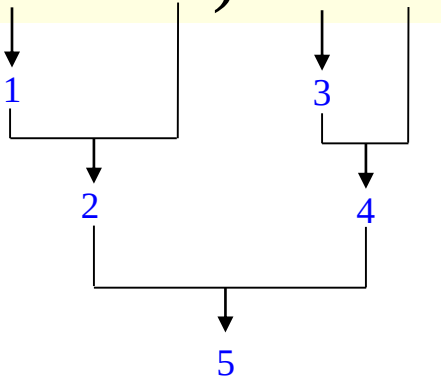
$$Z = A.B + C + \overline{X}.Y$$



Boolean Expressions: Operator Precedence

- Parentheses can be used to override operator precedence.

$$Z = (A.B + C) + \overline{X}.Y$$



Boolean Expressions: Truth Table

- A truth table represents all possible values of an expression given the possible values of its inputs.
- **How do we build a truth table?**
  - Step 1: Create columns for all variables
  - Step 2: Determine the number of rows needed (how many rows should appear?) $\Rightarrow$ For n inputs, # of rows is 2^n .
  - Step 3: Define all possible values for the inputs starting from all 0's to all 1's, e.g. for 3 input variables from 000 to 111
  - Step 4: Find the value of the expression for each input value and fill in the table.

Boolean Expressions: Truth Table

Example: Boolean expression **F = X + Y.Z**

Inputs: X, Y, Z

Outputs: F

# of inputs=n=3

# of columns=#of inputs + #of outputs=4

# of rows=$2^n=2^3=8$

Truth Table

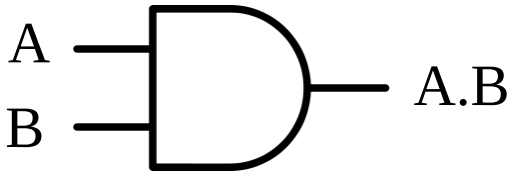
Inputs			Output
X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Outline

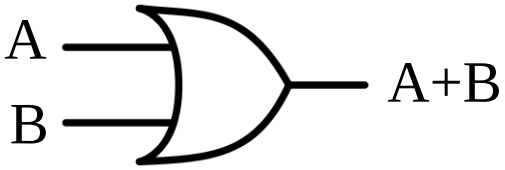
- Boolean Logic
- Boolean Algebra
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- **Logic Gates**
- Logic Circuits
- Summary

Logic Gates

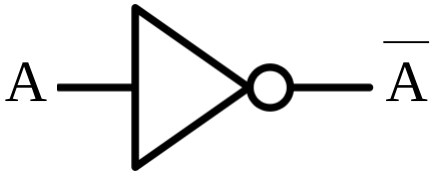
- A logic gates is an electronic device that operates on a collection of binary inputs to produce a binary output.
- The three basic logic gates are: AND gate, OR gate, NOT gate.



AND gate



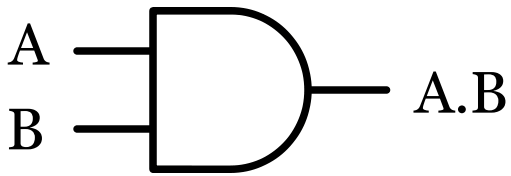
OR gate



NOT gate

Logic Gates: AND Gate

- AND (product) of two inputs.



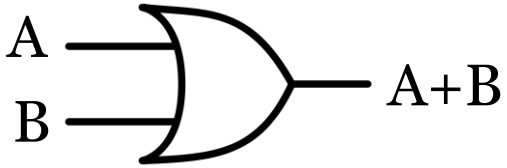
AND gate

Truth Table

Inputs		Output
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gates: OR Gate

- OR (sum) of two inputs.



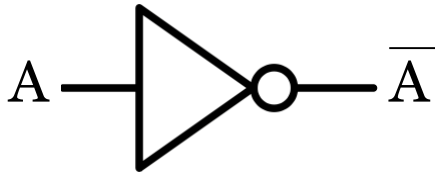
OR gate

Truth Table

Inputs		Output
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gates: NOT Gate

- NOT (complement) of one input.



NOT gate

Truth Table

Input	Output
A	$\overline{A}$
0	1
1	0

Outline

- Boolean Logic
- Boolean Algebra
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- Logic Gates
- **Logic Circuits**
- Summary

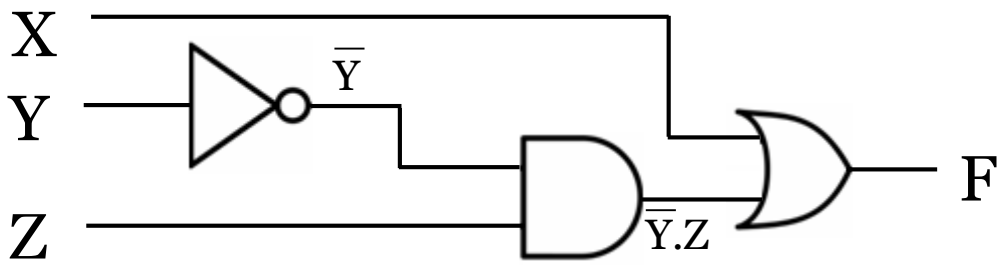
Logic Circuits

- A circuit is a collection of logic gates that implements one or more Boolean expressions.

Example-1: Boolean expression $F = X + \overline{Y}.Z$

Inputs: X, Y, Z

Outputs: F



Logic Circuit

Truth Table

Inputs			Output
X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logic Circuits

Example-2: Boolean expressions

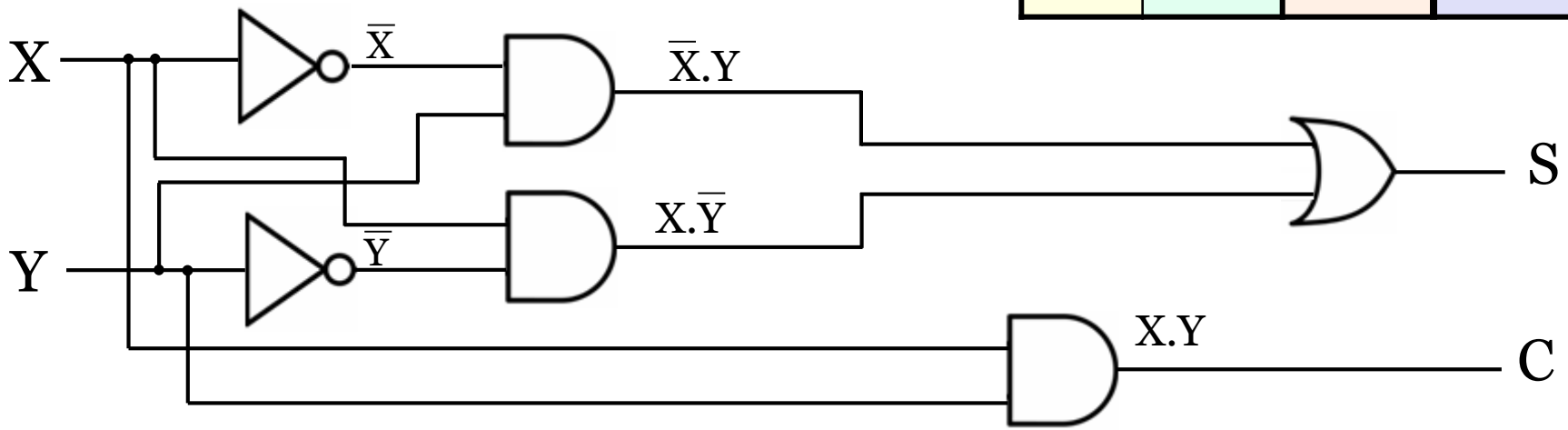
$S = \overline{X}.Y + X.\overline{Y}$

$C = X.Y$

Inputs: X, Y Outputs: S, C

Truth Table

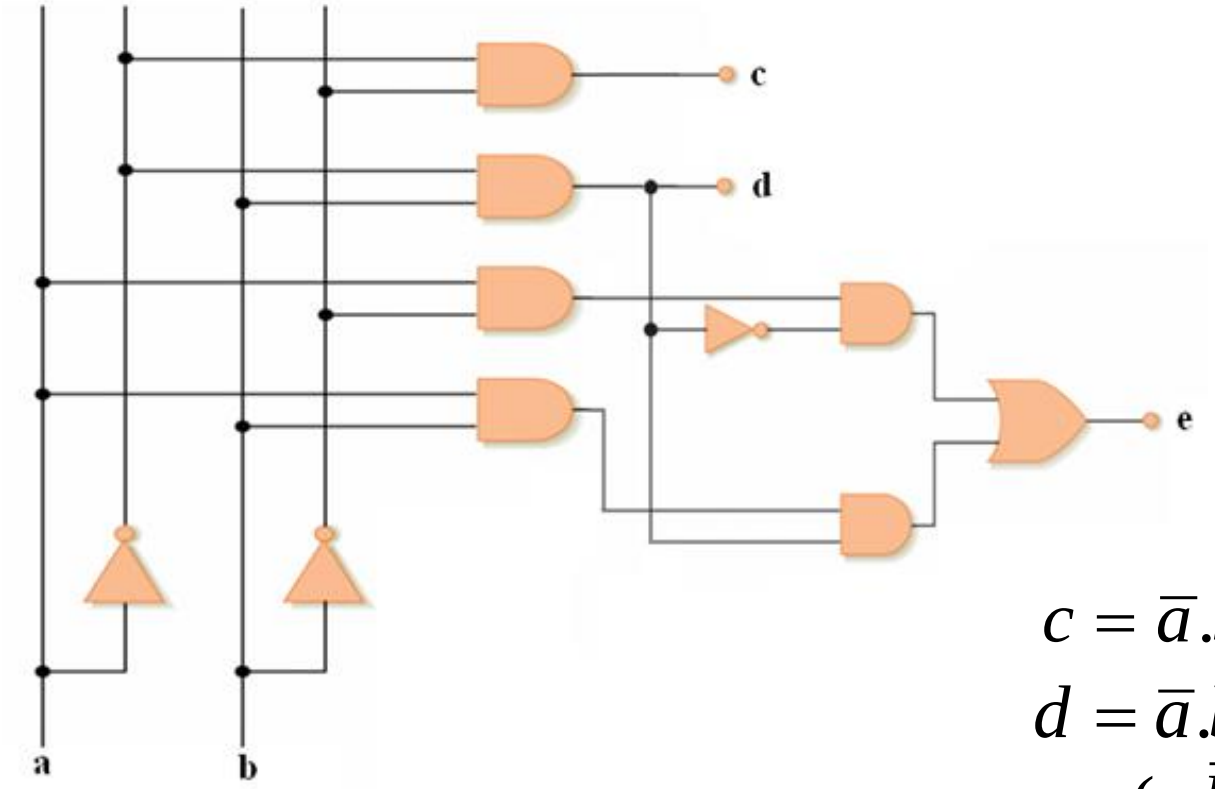
Inputs		Outputs	
X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Logic Circuit - Half Adder

Logic Circuits

Example-3: Determine the Boolean expression of the following circuit and construct the corresponding truth table considering that a and b are the input signals.



$$c = \bar{a}.\bar{b}$$

$$d = \bar{a}.b$$

$$e = (a.\bar{b}).\bar{d} + (a.b).d$$

Outline

- Boolean Logic
- Boolean Algebra
- Binary Variables
- Binary Operators
  - Logical AND
  - Logical OR
  - Logical Negation
- Boolean Expressions
- Logic Gates
- Logic Circuits
- **Summary**

Summary

- Boolean Algebra is a mathematical tool used in the analysis and design of digital circuits.
- OR, AND, NOT: basic Boolean operations.
- OR: HIGH output when any input is HIGH.
- AND: HIGH output only when all inputs are HIGH.
- NOT: output is the opposite logic level as the input.
- The order of evaluation in Boolean Expressions is: Parentheses, NOT, AND, OR.
- A logic gates is an electronic device that operates on a collection of binary inputs to produce a binary output.