Computer Organization and Architecture

COSC 2425

Lecture – 1

Aug 22th, 2022

Acknowledgement: Slides from Edgar Gabriel & Kevin Long

Organizational issues (I)

- Classes:
  - MoWe, 4:00 PM – 5:30 PM, SR 116
  - MoWe, 4:00 PM – 5:30 PM, Synchronous Online (Zoom)
- Labs:
  - Mo, 2:30 PM – 4:00 PM, SEC 100
- Free to attend class in either format.
- Exam has to be taken in person/in classroom.

- Email: pmantini@uh.edu
  - Office hours: Thursday, 1 PM – 2 PM or by appointment.
    - Only online office hours using Zoom
    - http://qil.uh.edu/coa/meeting/

Organizational Issues (III)

- TA's for the course:
  - Mirza, Samiha, Email: samiha.mirza1234@gmail.com
    - Office hours: TTH, 11:00 AM – 12:00 PM

-More TA's TBA

- http://qil.uh.edu/coa/meeting/

Course Content

Dates	Topics
Weeks 1 - 7	Performance evaluation, Instruction Set Architectures; Number representations; Computer Arithmetic;
October first week	Midterm
Weeks 8 - 15	Instruction Level Parallelism; Branch Prediction; Assembly Language Programming; Memory Hierarchies; Virtual Memory; Parallel Processors; Networking
December Finals week	Final Exam

Organizational Issues (IV)

Activity	Weight
3 Homework's (~15%)	**45%**
Midterm	**25%**
Final Exam	**30%**

Logistics

- Late policy for Assignments/Homework:
  - Late by 1 day - 25% off the grade
  - Late by 2 days - 50% off the grade
  - Late by more than 2 days – No Credit
- Late policy for midterm:
  - No late submission possible
- Collaboration policy:
  - No collaboration allowed for **homework's and exams**, **must be solved individually.**
  - Posting questions on online forums (Chegg.com, etc.) is **not allowed**

Important Notes

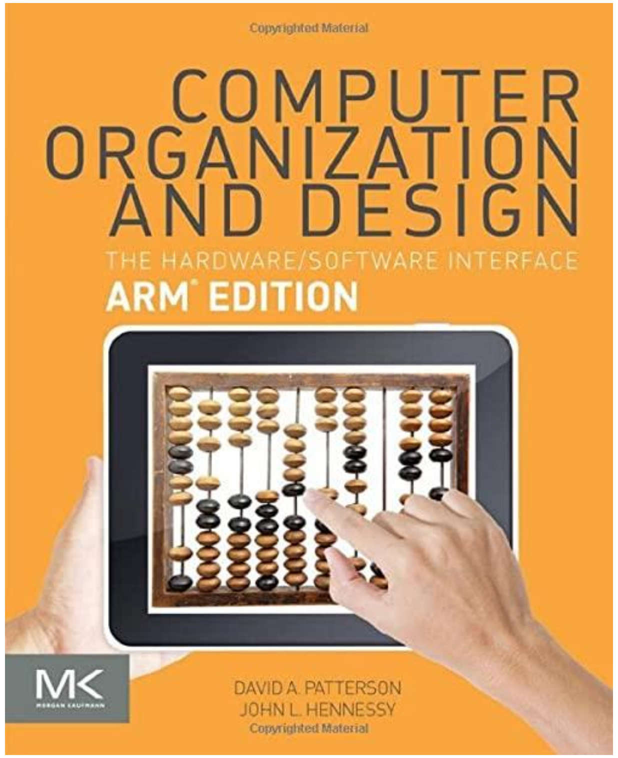
- 3-Day Policy: One has 3 days starting from the end of the class time in which the graded assignment/exam papers have been distributed and/or posted in order to object to the score of that assignment or exam. The objection shall be submitted electronically by emailing the TA and the instructor.

Academic Honesty

- The course has a ZERO-TOLERANCE policy for cheating, whether in exams or assignments. Plagiarism, copying and other anti-intellectual behavior are prohibited by the university regulations.
- Any violations to the university academic honesty code will be reported to the department and the university
- Solutions available on chegg.com or similar sites are automatically **disqualified.**

Textbook

Computer Organization and Design ARM Edition: The Hardware Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) 1st Edition by David A. Patterson (Author), John L. Hennessy



Course Expectations

1. Lectures and Lab (Required)
   1. Attend Lecture and Lab meeting (face-to-face/online)
   2. Participate and ask questions
2. Submit homework/assignments via **Blackboard** (Required)
3. Submit midterm and exam **in-person/in-classroom**(Required)
4. Attend Office hours (Optional)

Class Website

http://qil.uh.edu/coa

Accessing Videos (posted after each class)

Go to class website: qil.uh.edu/coa

Digital Image Processing   Home   Schedule   Meetings   Homework   Resources

Schedule

Week	Date	Topic	Video	Notes
1	Jan. 14, 2020	Introduction - 1: Class Logistics, motivation, applications		pdf
1	Jan. 16, 2020	Introduction - 2: Image Formation, Pin hole camera, optics of the eye, perspective projection		pdf
2	Jan. 21, 2020	Intro to Python: Introduction to python		pdf
2	Jan. 23, 2020	Image Acquisition: Image Acquisition, image formation model, photoconverters, a/d conversion, sampling and digitization, resampling		pdf
3	Jan. 28, 2020	Binary Image Processing - 1: Thresholding, Histograms, Bimodal histograms.		pdf
3	Jan. 30, 2020	Binary Image Processing - II: Blob coloring		pdf
4	Feb. 4, 2020	Binary Image Processing - III and Point Operations: Morphological operations		pdf

Deadlines

Accessing Videos

Digital Image Processing   Home   Schedule   Meetings   Homework   Resources

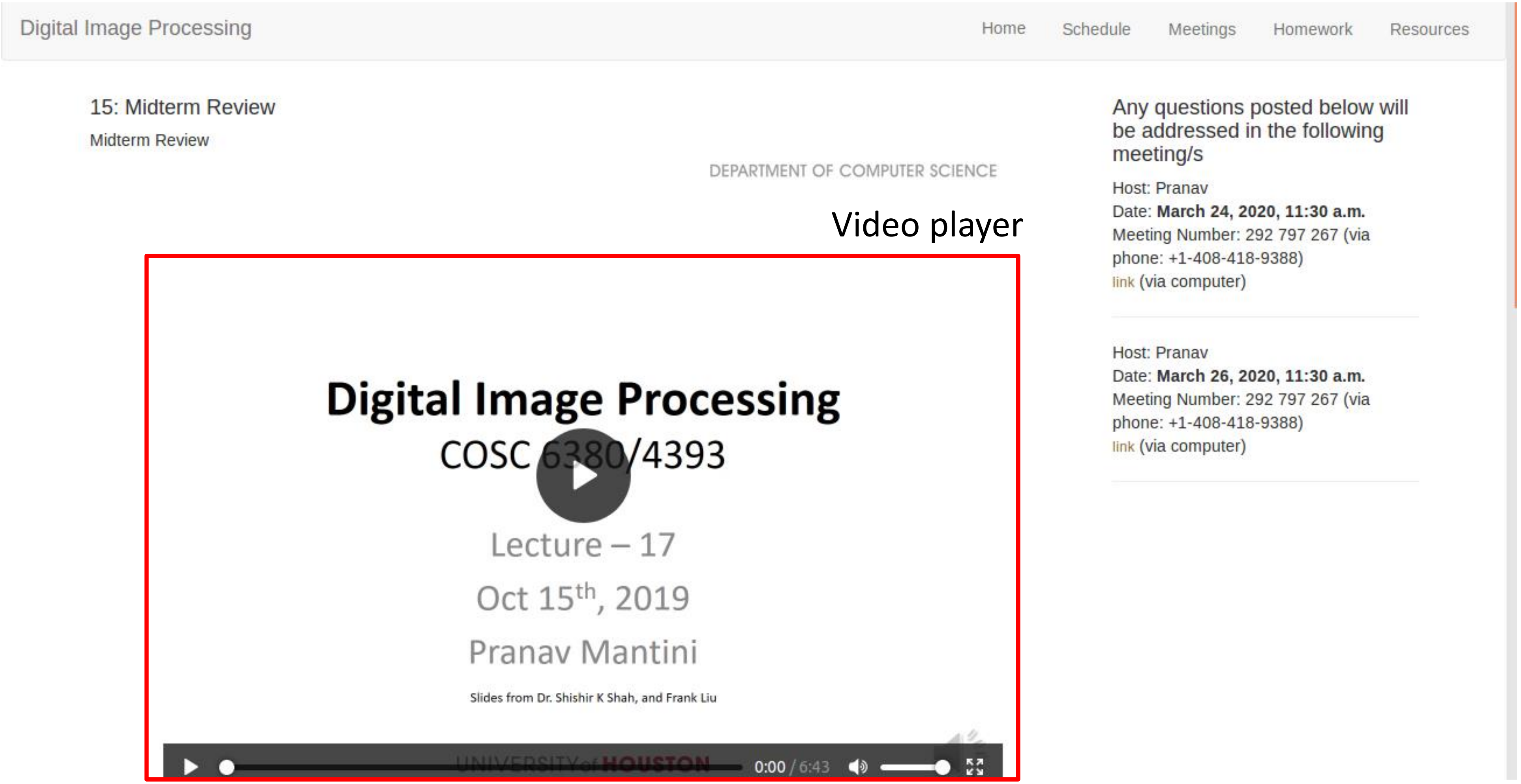
Schedule

Week	Date	Topic	Video	Notes
1	Jan. 14, 2020	Introduction - 1: Class Logistics, motivation, applications		pdf
1	Jan. 16, 2020	Introduction - 2: Image Formation, Pin hole camera, optics of the eye, perspective projection		pdf
2	Jan. 21, 2020	Intro to Python: Introduction to python		pdf
2	Jan. 23, 2020	Image Acquisition: Image Acquisition, image formation model, photoconverters, a/d conversion, sampling and digitization, resampling		pdf
3	Jan. 28, 2020	Binary Image Processing - 1: Thresholding, Histograms, Bimodal histograms.		pdf
3	Jan. 30, 2020	Binary Image Processing - II: Blob coloring		pdf
4	Feb. 4, 2020	Binary Image Processing - III and Point Operations: Morphological operations		pdf

Deadlines

Click for video

Watch Videos



Attending Meetings

1. Two types of meetings:
   1. Lecture meeting: Discuss course topics, address any questions from videos, and answer other questions if time permits.
   2. Office hours: Answer questions, help with assignments, etc. Please remember these meetings may not completely be one to one sessions. Please communicate with the TA and setup separate meeting with TA if you need to discuss something alone. (For example grades etc.)

Meeting

Digital Image Processing

Home Schedule Meetings Homework Resources

Upcomming Meetings

Date	Day	Type	Host	Meeting Number/Access Code	Meeting Link
March 24, 2020, 11:30 a.m.	Tuesday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
March 24, 2020, 2 p.m.	Tuesday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
March 25, 2020, 10 a.m.	Wednesday	Office Hours	**Sara**	299 920 079 (via phone: +1-408-418-9388)	link (via computer)
March 26, 2020, 11:30 a.m.	Thursday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
March 26, 2020, 2 p.m.	Thursday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
March 30, 2020, 10 a.m.	Monday	Office Hours	**Sara**	298 312 750 (via phone: +1-408-418-9388)	link (via computer)
March 31, 2020, 11:30 a.m.	Tuesday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
March 31, 2020, 2 p.m.	Tuesday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
April 2, 2020, 11:30 a.m.	Thursday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
April 2, 2020, 2 p.m.	Thursday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
April 3, 2020, 9 a.m.	Friday	Office Hours	**Khadija**	296 057 977 (via phone: +1-408-418-9388)	link (via computer)
April 6, 2020, 10 a.m.	Monday	Office Hours	**Sara**	298 312 750 (via phone: +1-408-418-9388)	link (via computer)
April 7, 2020, 11:30 a.m.	Tuesday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
April 7, 2020, 2 p.m.	Tuesday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)

Meeting

Digital Image Processing — Home | Schedule | Meetings | Homework | Resources

Upcomming Meetings

Date	Day	Type	Host	Meeting Number/Access Code	Meeting Link
March 24, 2020, 11:30 a.m.	Tuesday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
March 24, 2020, 2 p.m.	Tuesday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
March 25, 2020, 10 a.m.	Wednesday	Office Hours	**Sara**	299 920 079 (via phone: +1-408-418-9388)	link (via computer)
March 26, 2020, 11:30 a.m.	Thursday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
March 26, 2020, 2 p.m.	Thursday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
March 30, 2020, 10 a.m.	Monday	Office Hours	**Sara**	298 312 750 (via phone: +1-408-418-9388)	link (via computer)
March 31, 2020, 11:30 a.m.	Tuesday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
March 31, 2020, 2 p.m.	Tuesday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
April 2, 2020, 11:30 a.m.	Thursday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
April 2, 2020, 2 p.m.	Thursday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)
April 3, 2020, 9 a.m.	Friday	Office Hours	**Khadija**	296 057 977 (via phone: +1-408-418-9388)	link (via computer)
April 6, 2020, 10 a.m.	Monday	Office Hours	**Sara**	298 312 750 (via phone: +1-408-418-9388)	link (via computer)
April 7, 2020, 11:30 a.m.	Tuesday	Lecture Meeting	**Pranav**	292 797 267 (via phone: +1-408-418-9388)	link (via computer)
April 7, 2020, 2 p.m.	Tuesday	Office Hours	**Pranav**	298 783 225 (via phone: +1-408-418-9388)	link (via computer)

Attend Meeting

- Three ways
  - Via phone
    1. Call +1-346-248-7799
    2. Enter Access Code (See meeting page for specific access code)
  - Via computer
    1. Follow the link.
       1. No need to signup
       2. On windows it might install a plugin, accept and install
       3. On Linux/mac just follow the link and you should be able to enter the meeting
       4. Allow necessary permission in the browser to access the mic and audio
  - Via mobile App
    1. Download app
    2. Enter meeting number
    3. Enter Passcode

Attend Meeting

1. You can log-in ahead and wait for the meeting to start.
2. Make sure your mic is in mute when you enter the meeting.
    1. These meetings are like classrooms, to minimize background noise make sure to enter the meeting with your mic in mute, and continue to keep in mute.
    2. You can unmute when you have a question. We will figure out the details as we go along.

Introduction

The Computer Revolution

- Computers are pervasive

The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore’s Law
- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
- Computers are pervasive

Have you used a computer today?

Have you used a computer today?

- Check email/send messages

Have you used a computer today?

- Check email/send messages
  - On Phone/laptop (a personal device)
  - Retrieve data from a web server (Server computer)

Classes of Computers

- Personal computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff

- Server computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized

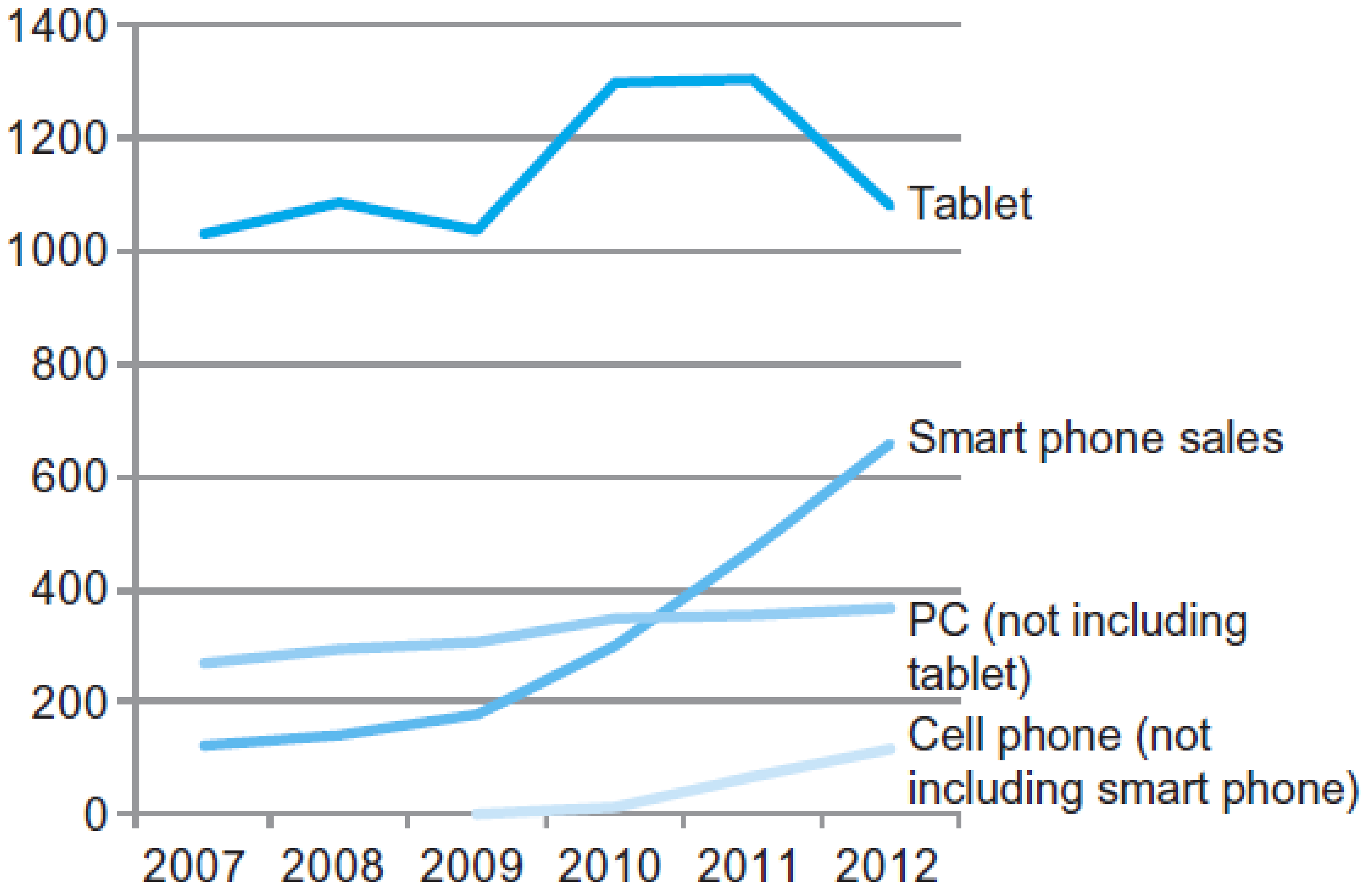
Have you used a computer today?

- Check email/send messages
  - On Phone/laptop (a personal device)
  - Retrieve data from a web server (Server computer)
- Perform Scientific computations
  - Supercomputers
- Use Navigation in car
  - Stereo/Navigation device (Embedded computer)

Classes of Computers

- Supercomputers
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

The PostPC Era



Breakdown

- Checking email? What is going on?

Breakdown

Authorize
Retrieve data
Respond

- Checking email

Request Data

emails

Desktop computer
At home

Gmail server
at data center (emails in db)

Breakdown

Authorize
Retrieve data
Respond

- Checking email

Request Data

Desktop computer
At home

Today: 20 emails
Yesterday: 10 emails
Total: 30 emails

Email1 => subject: Hi John

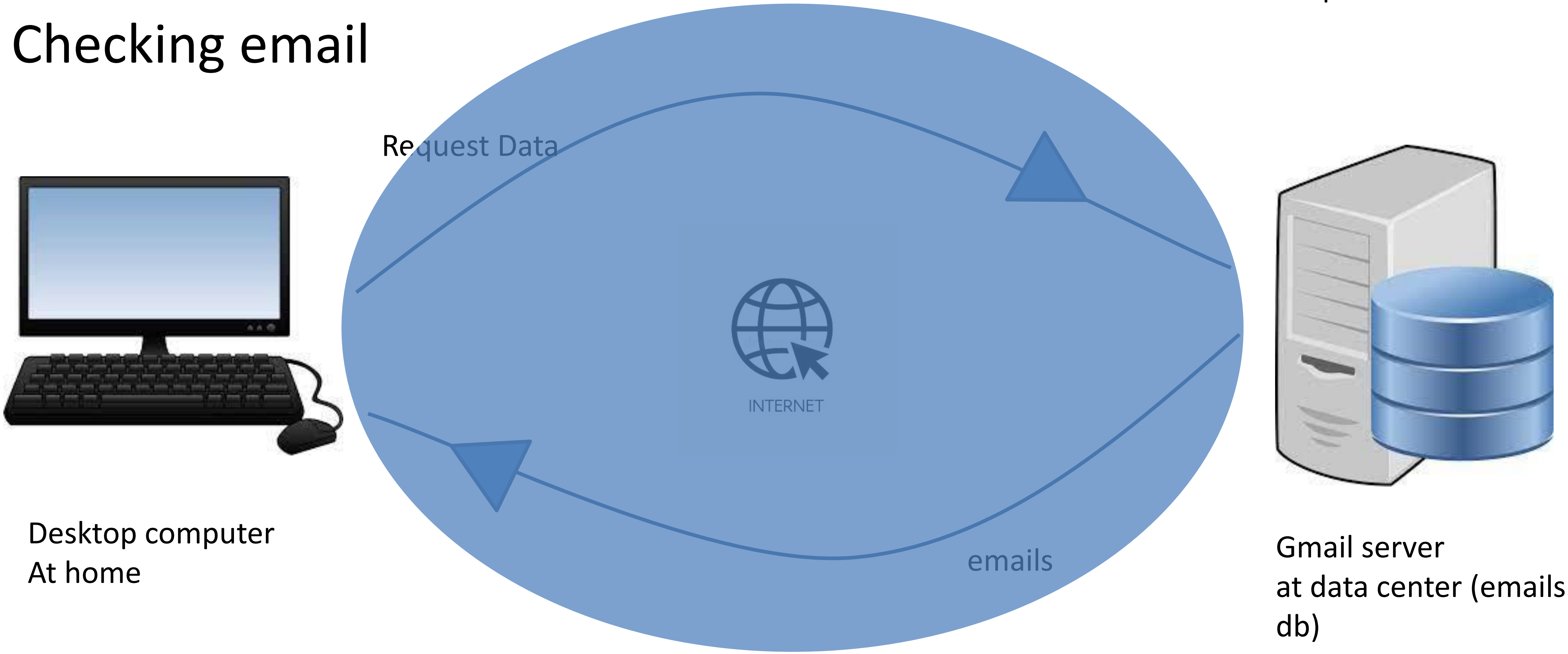
emails

Gmail server
at data center (emails in db)

Breakdown

Authorize
Retrieve data
Respond

- Checking email



Desktop computer
At home

Gmail server
at data center (emails in db)

How do computers work?

Today: 20 emails
Yesterday: 10 emails
Total: 30 emails

Email1 => subject: Hi John

How do computers work?

Today: 20 emails → Represent numbers
Yesterday: 10 emails
Total: 30 emails → Perform addition

Email1 => subject: Hi John → Represent text

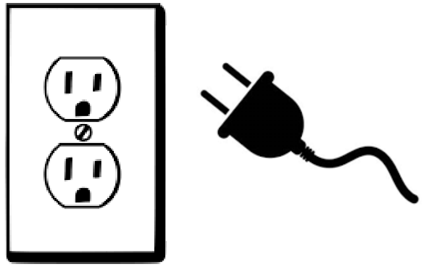
How do computers work?

Today: 20 emails → Represent numbers
Yesterday: 10 emails
Total: 30 emails → Perform addition

Email1 => subject: Hi John → Represent text

What is the basic idea behind a computer?
How does it do this?

Further Breakdown

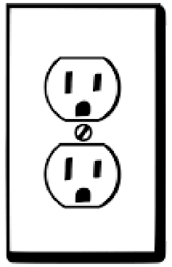


110v/240v
AC Current

Most electronic chips cannot handle AC current.
Too unstable, not unidirectional.

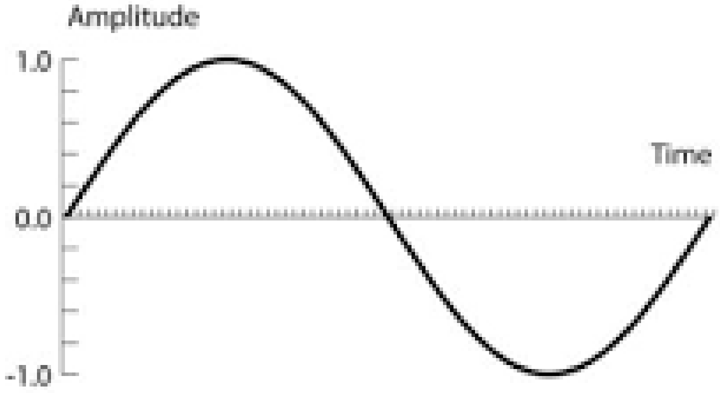
Requires stable constant low voltage. (~12 V)

Further Breakdown

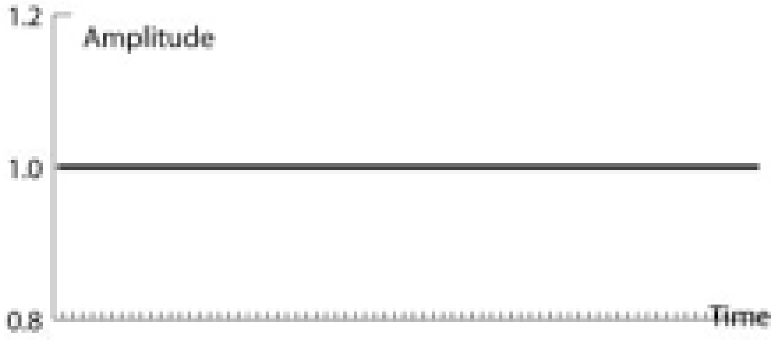


110v/240v
AC Current

PSU
12v DC current

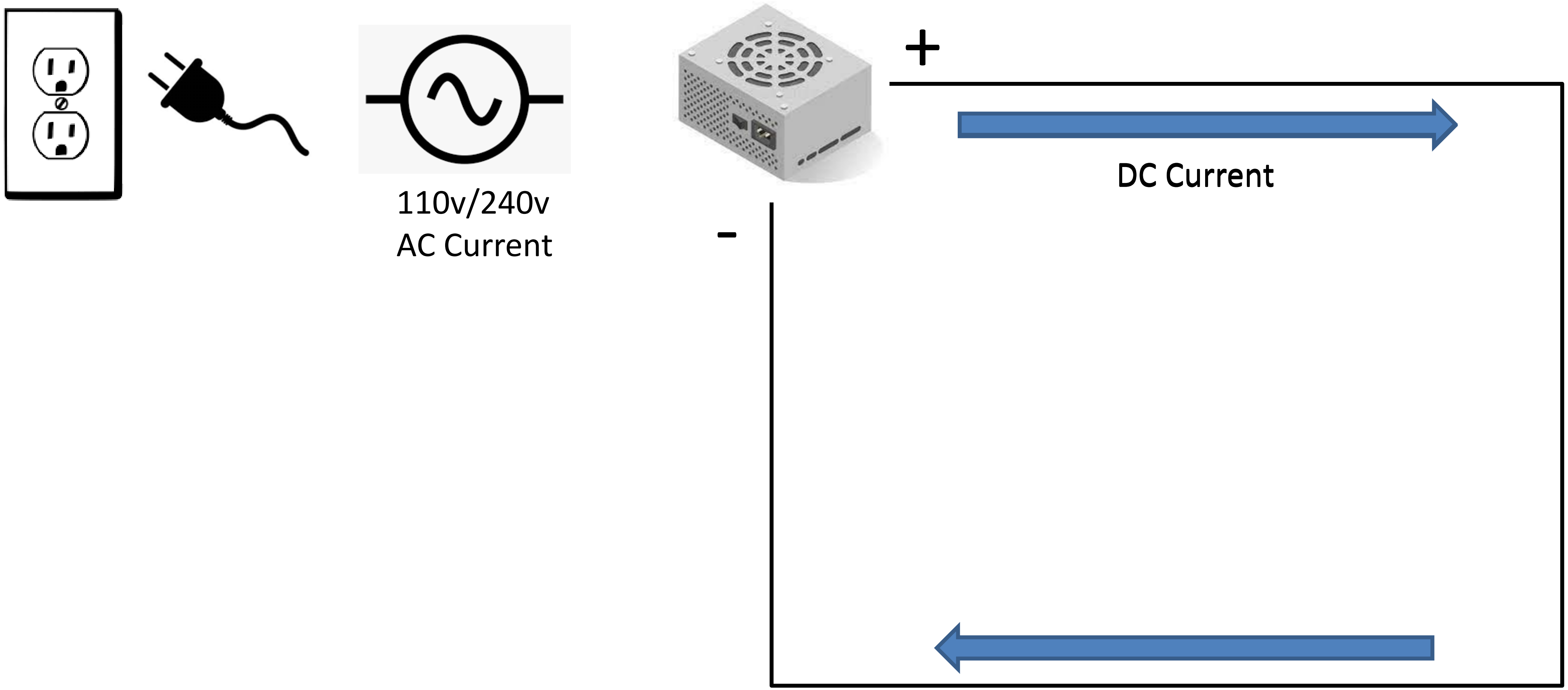


AC

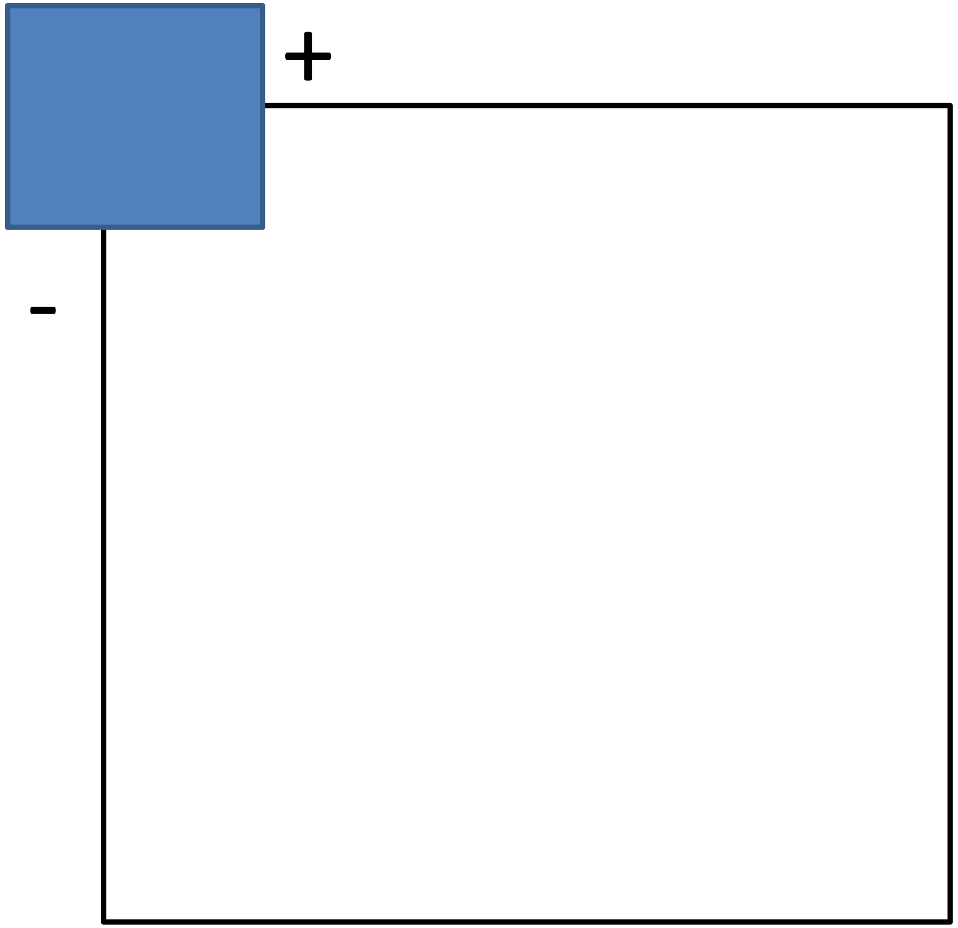


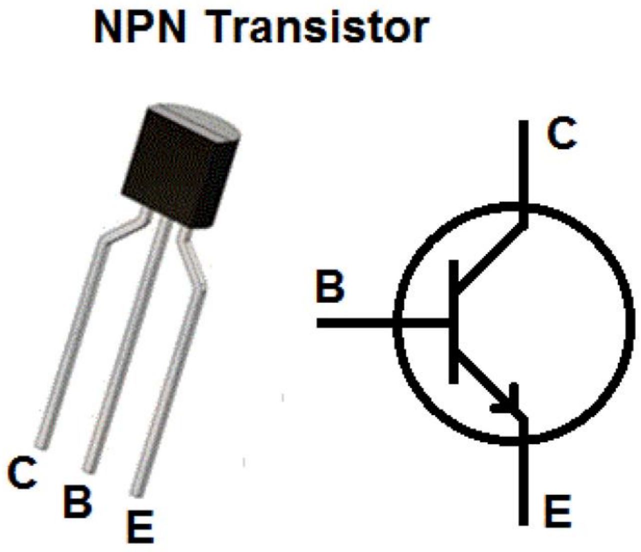
DC

Further Breakdown



Control the flow of current using transistors

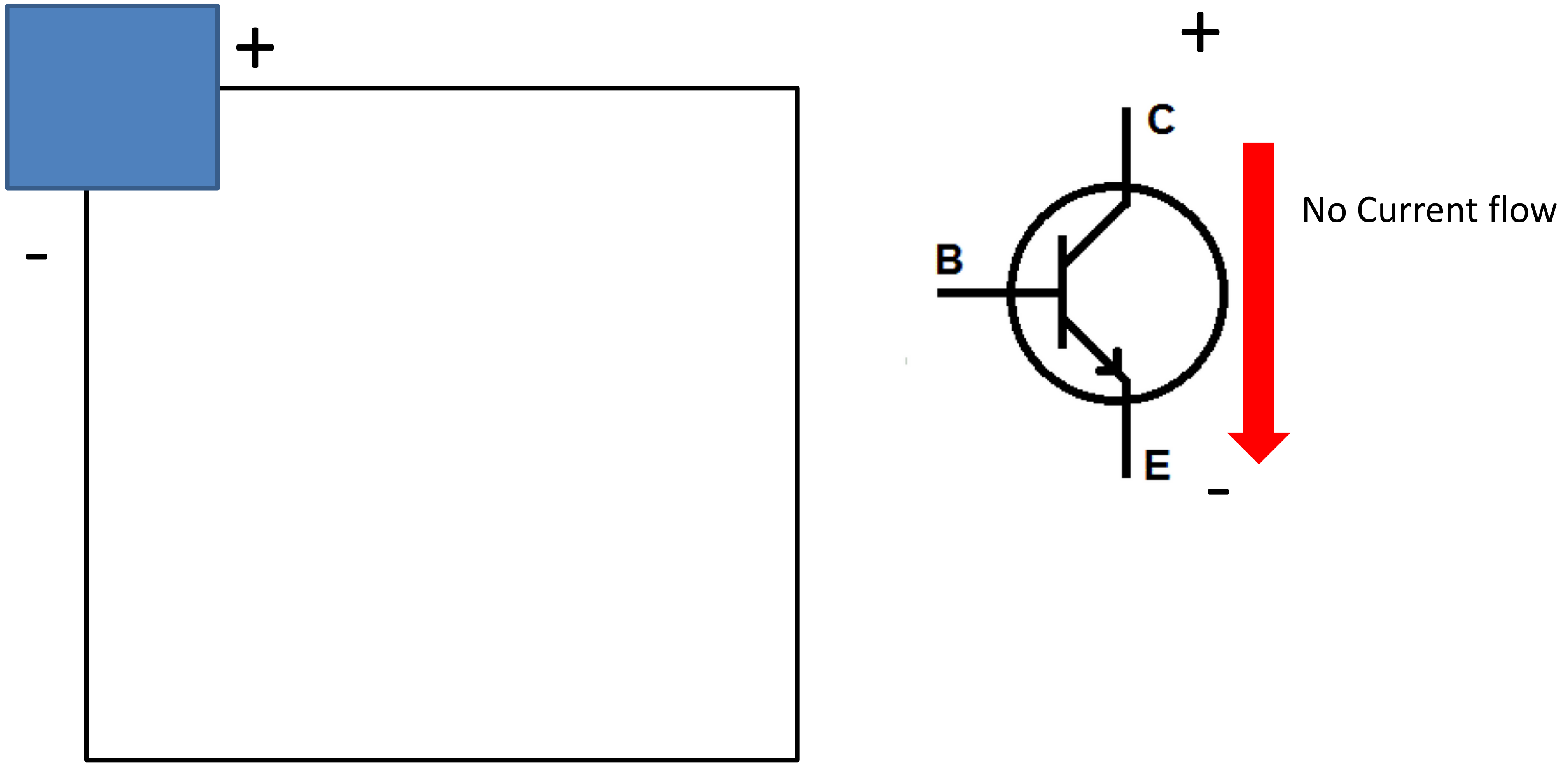




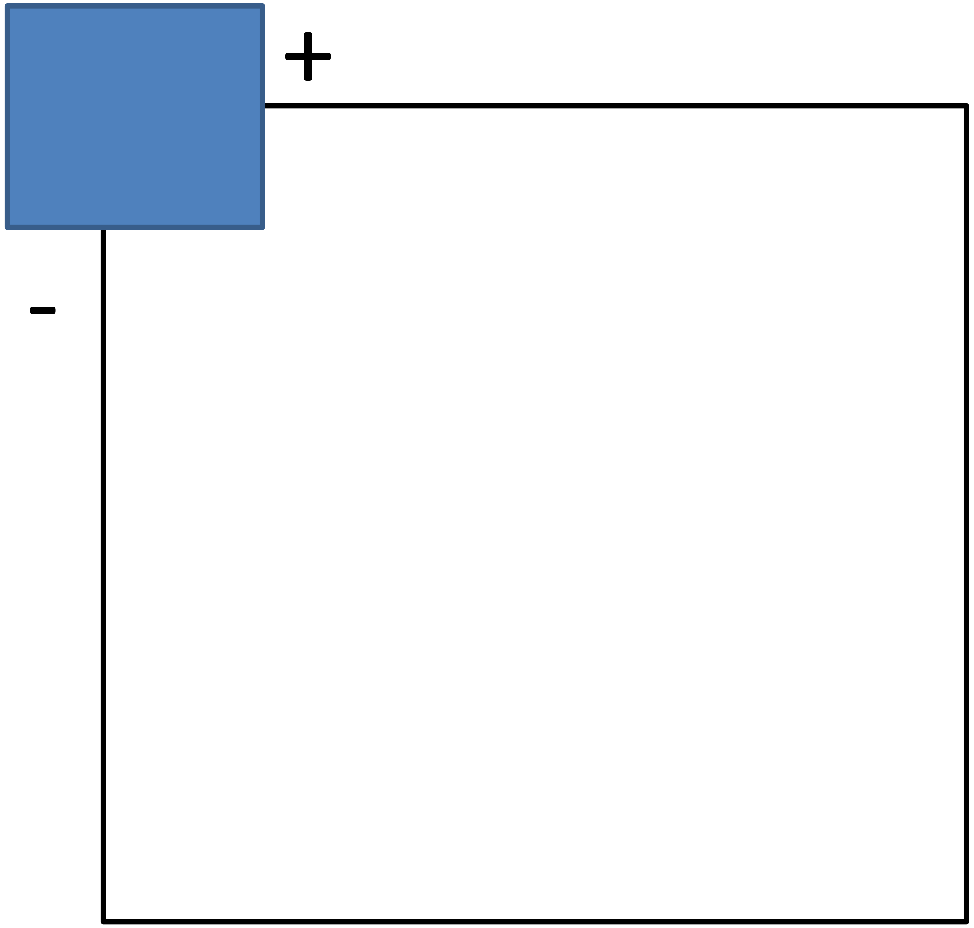
Transistor

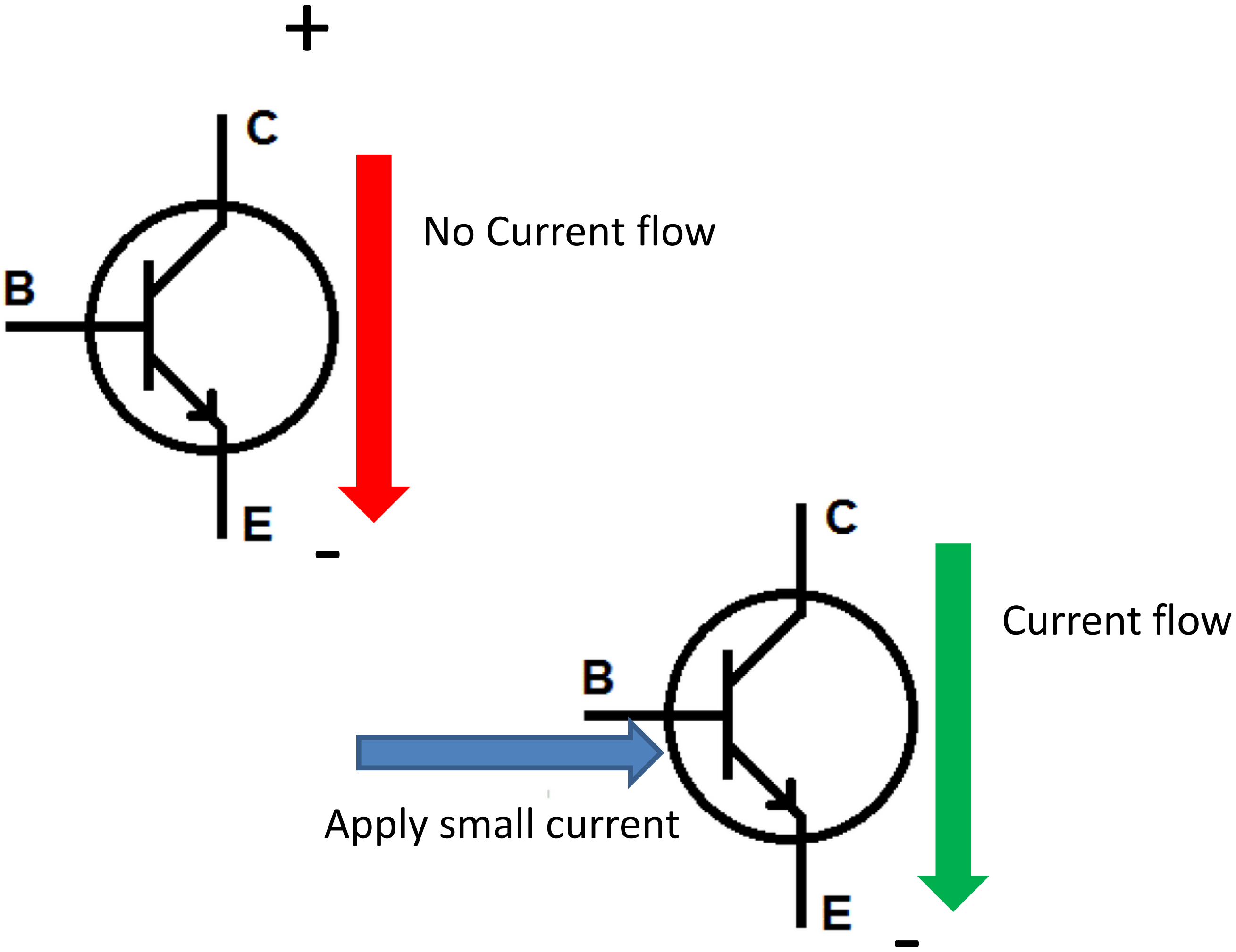
Symbol

Control the flow of current using transistors

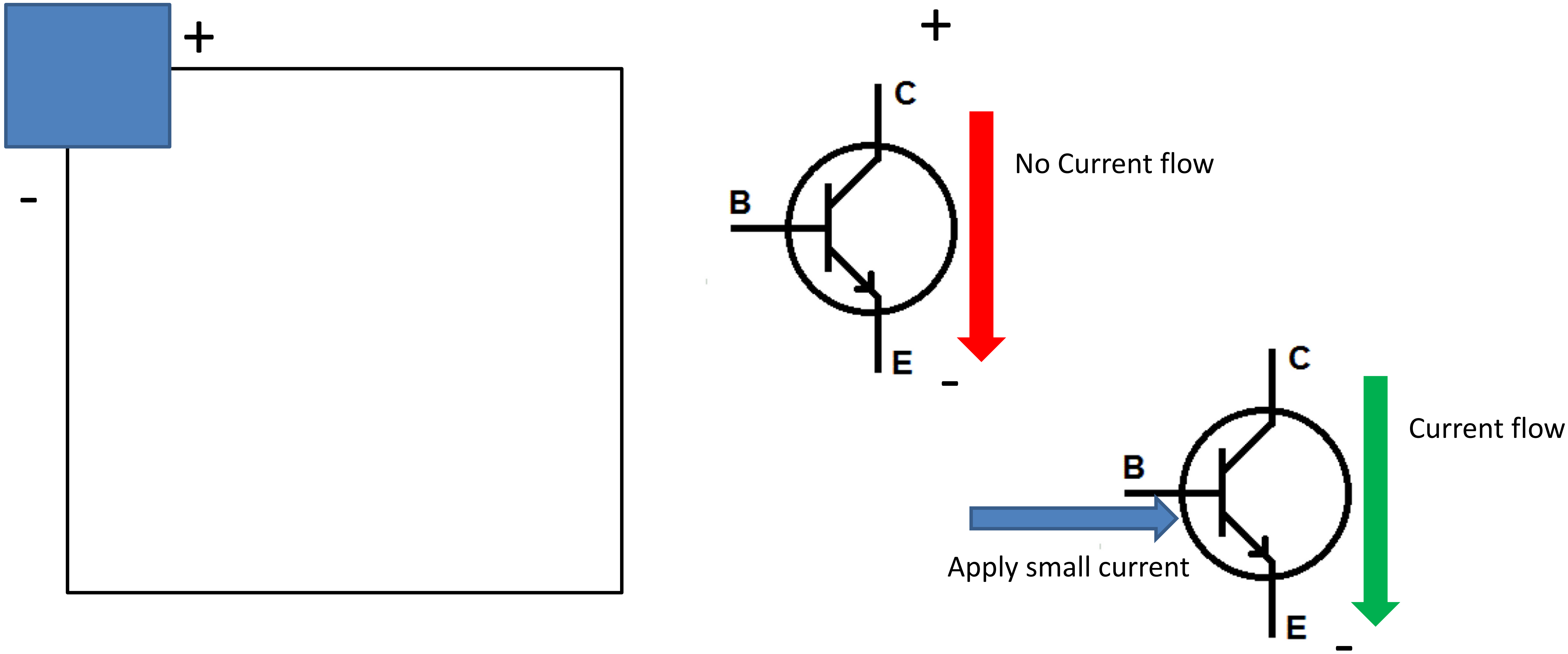


Control the flow of current using transistors

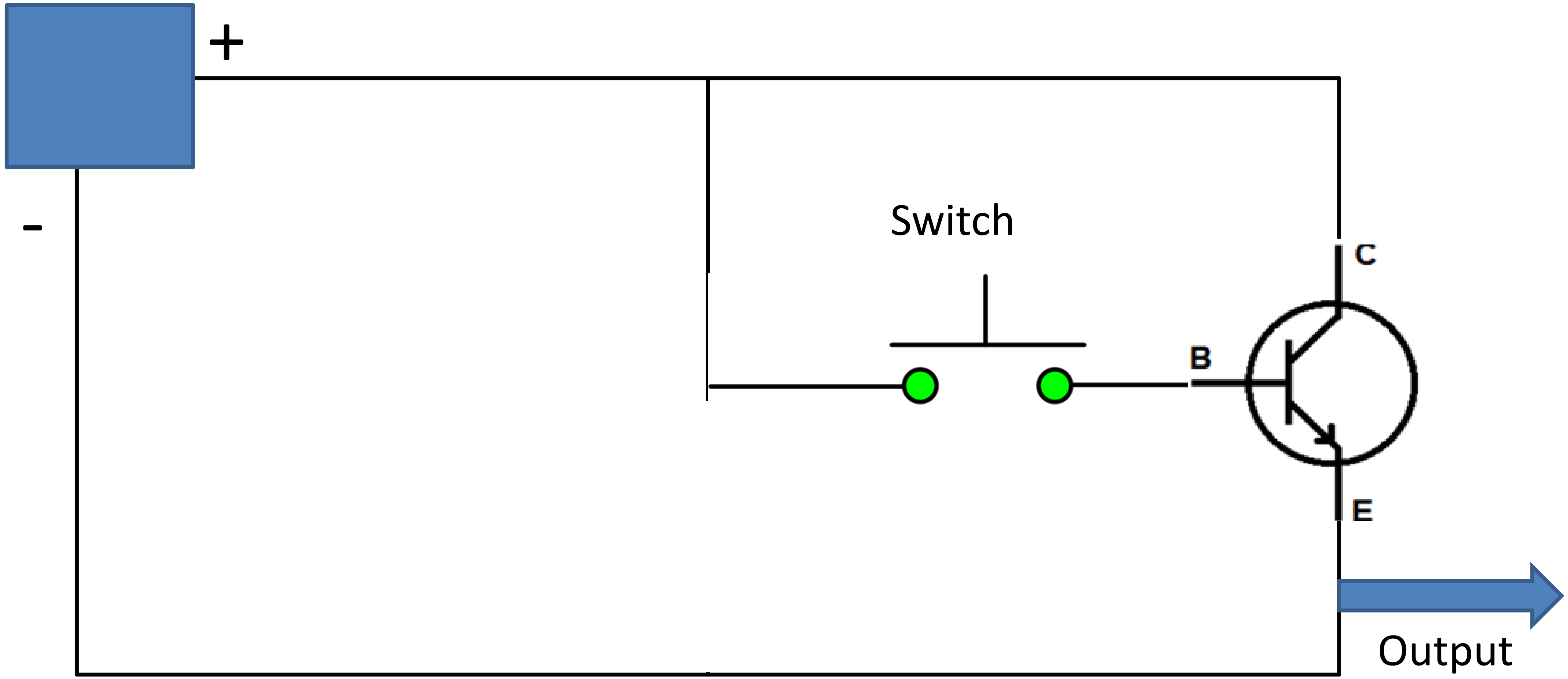




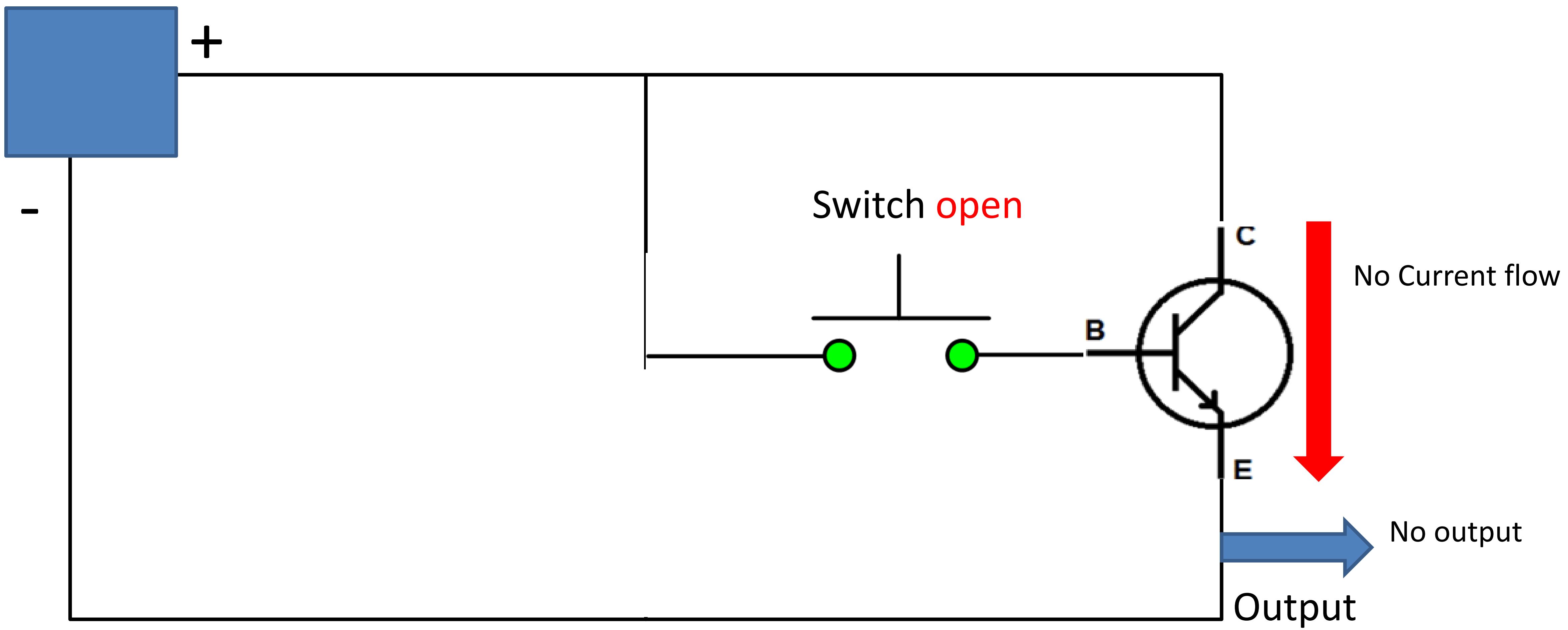
Control the flow of current using transistors



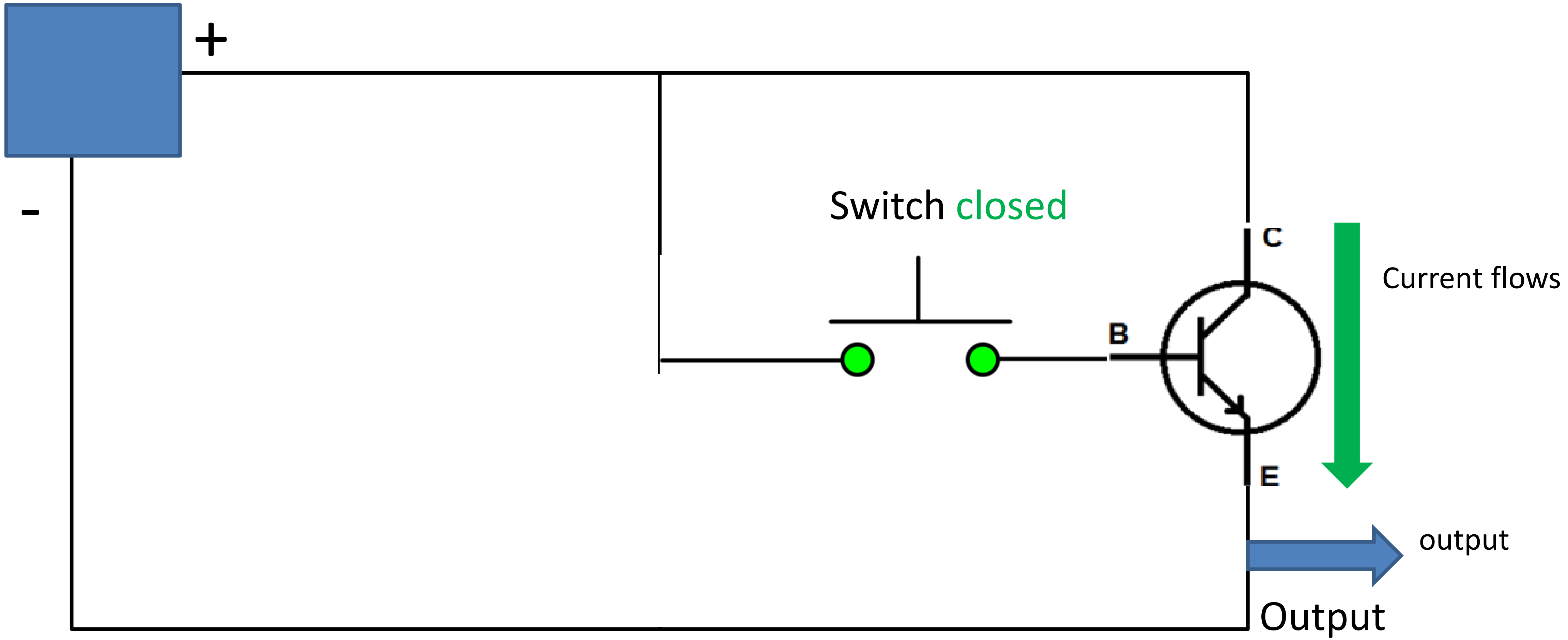
Control the flow of current using transistors



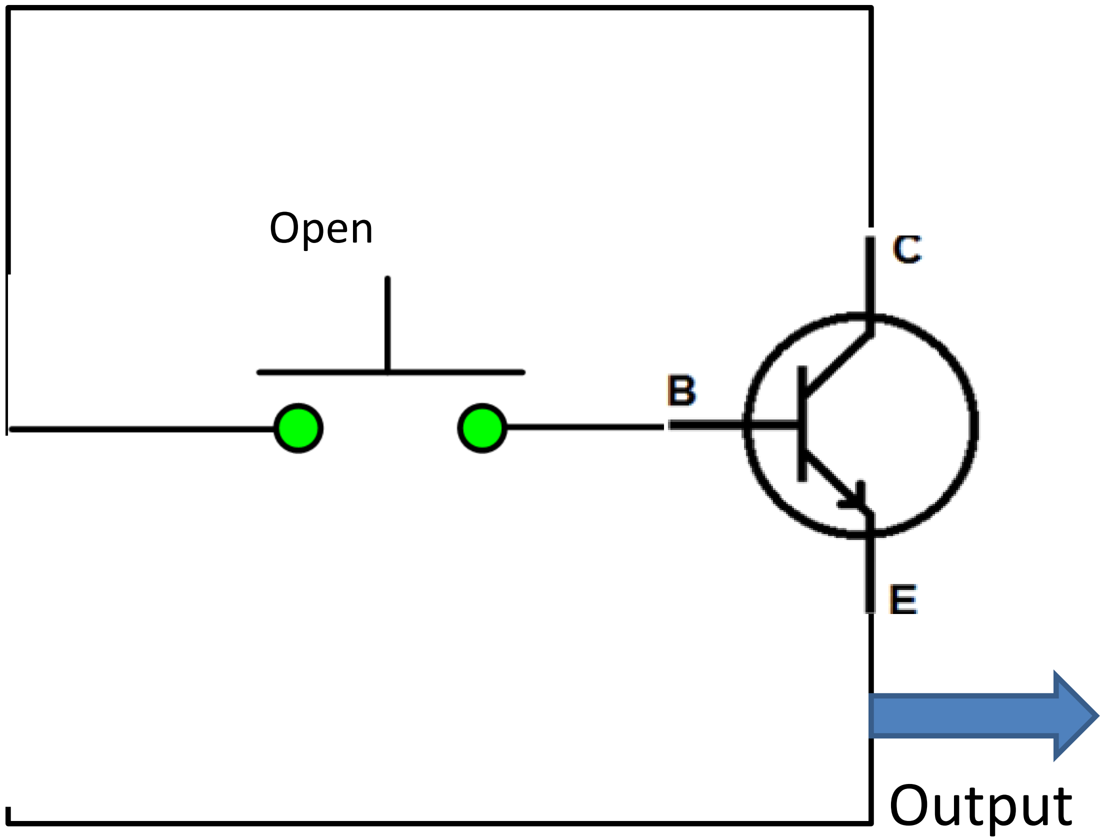
Control the flow of current using transistors



Control the flow of current using transistors

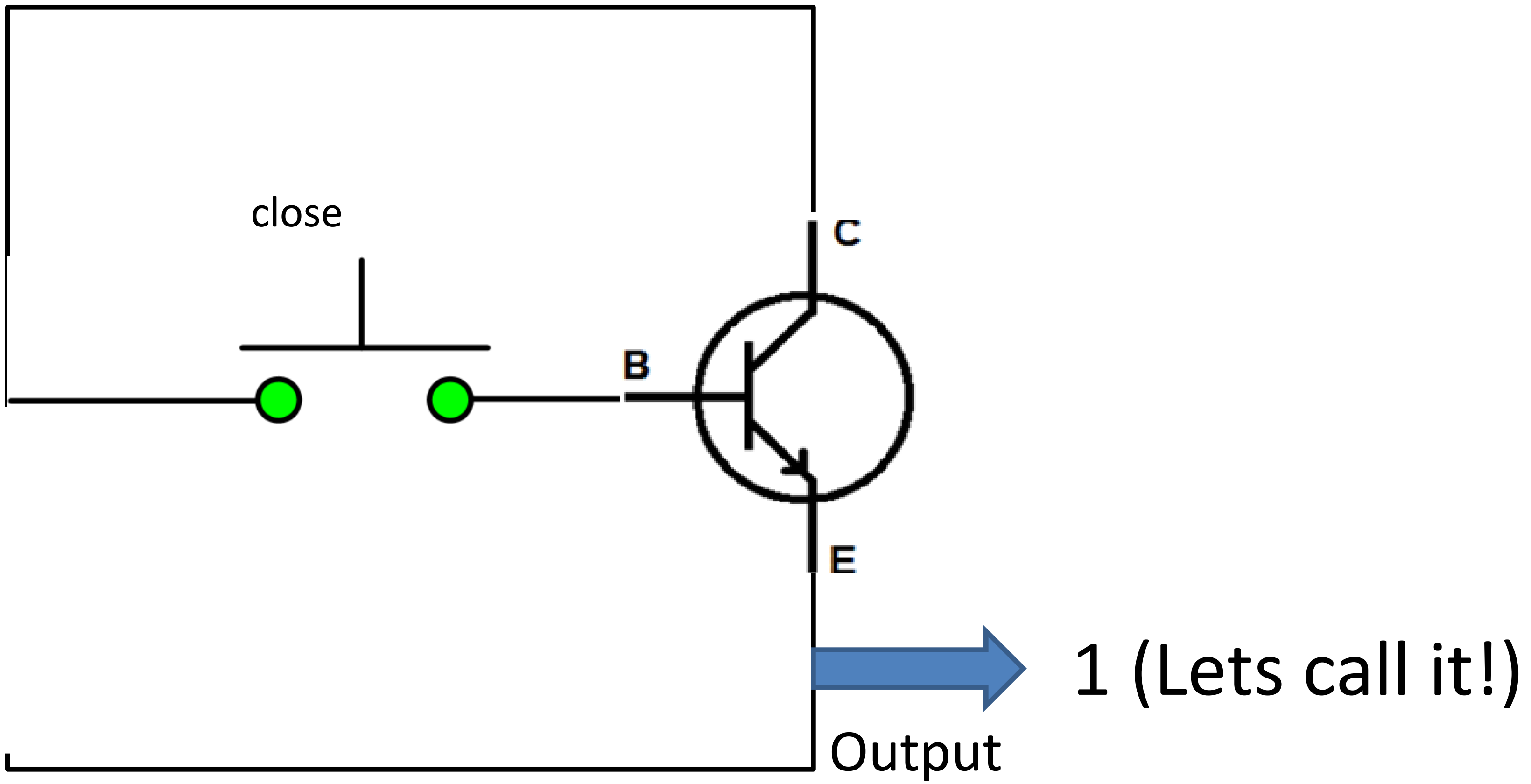


Control the flow of current using transistors

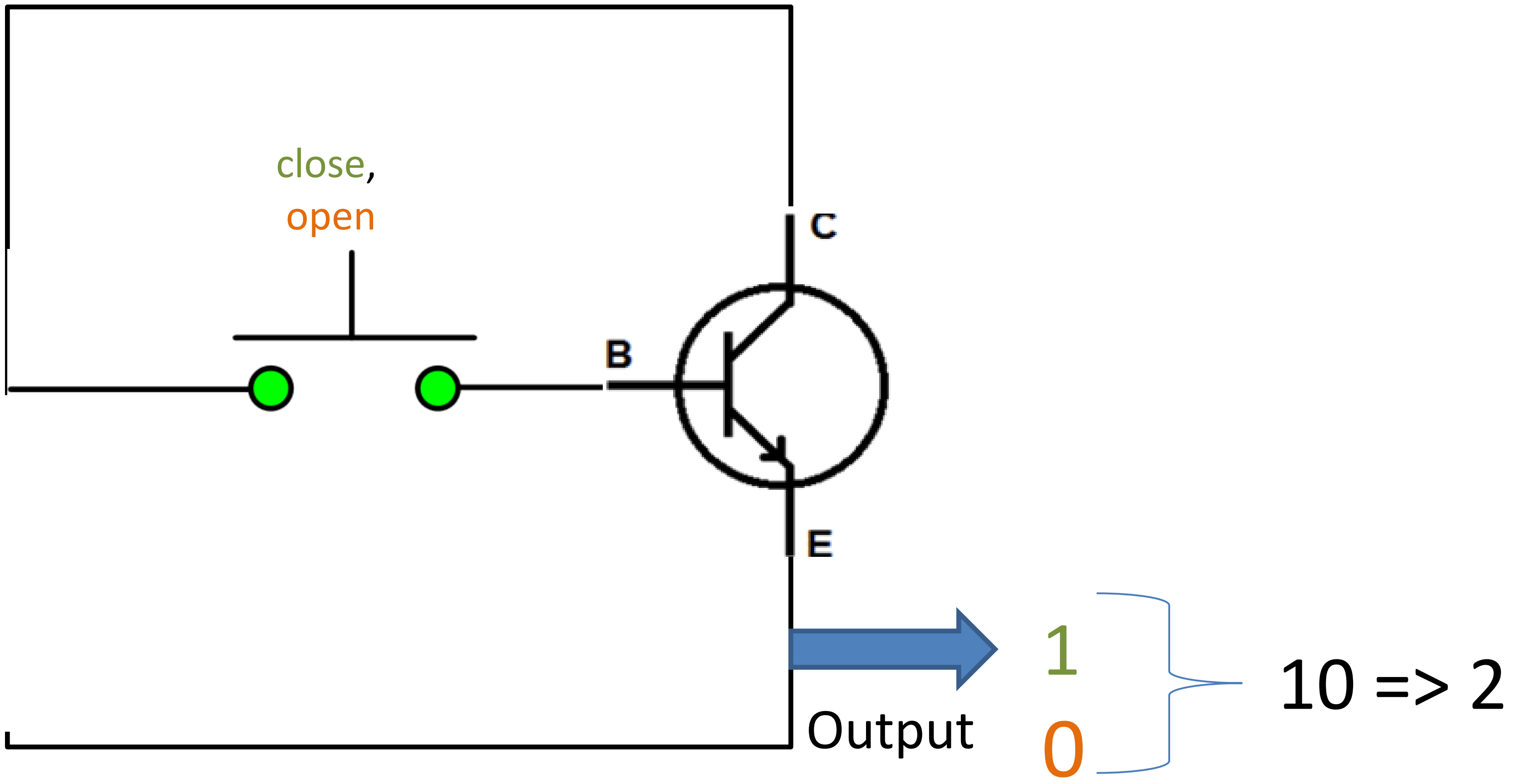


0 (Lets call it!)

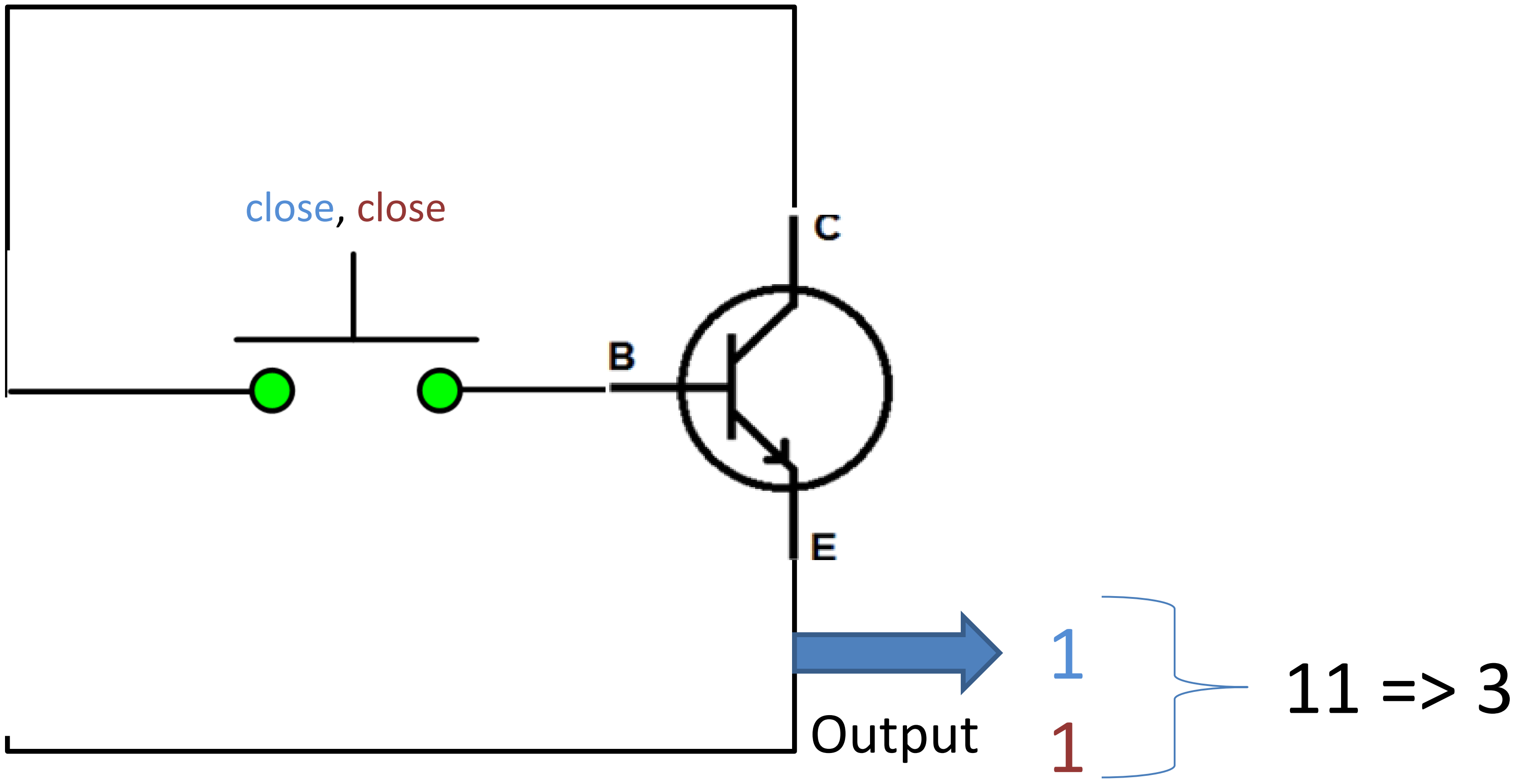
Control the flow of current using transistors



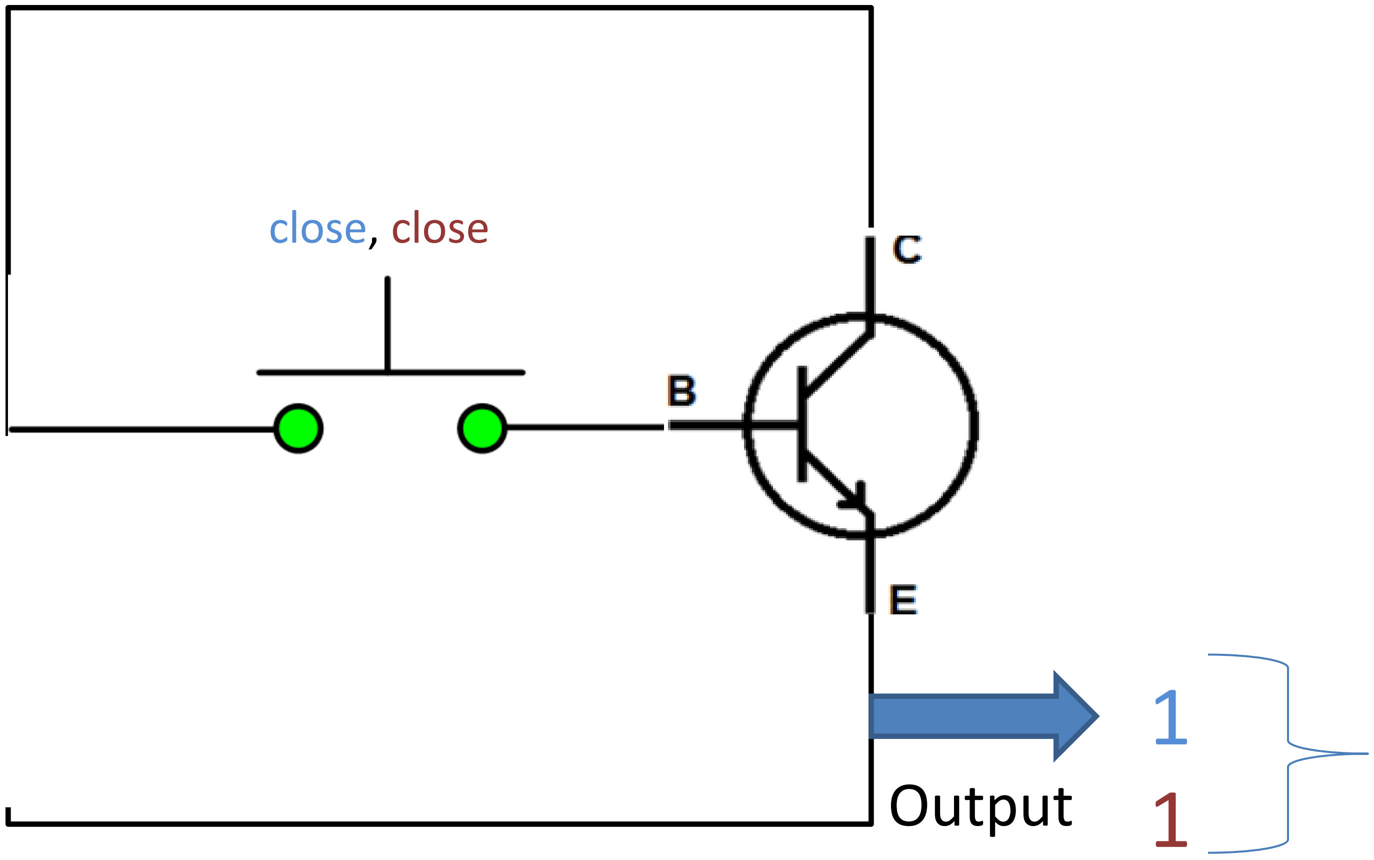
Control the flow of current using transistors



Control the flow of current using transistors



Control the flow of current using transistors



11 => 3

Binary numbers

Binary Representation of Numbers

Binary Number -> use two symbols for representation (0 & 1)

101101

Binary Representation of Numbers

Binary Number -> use two symbols for representation (0 & 1)

101101

index						
index	5	4	3	2	1	0
	1	0	1	1	0	1

Binary Representation of Numbers

Binary Number -> use two symbols for representation (0 & 1)

101101

index						
index	5	4	3	2	1	0
	1	0	1	1	0	1
	1×2^5 +	0×2^4 +	1×2^3 +	1×2^2 +	0×2^1 +	1×2^0
	1 X 32	0 X 16	1 X 8	1X 4	0 X 2	1 X 1

= 45

How do computers work?

Today: 20 emails → $\boxed{\text{Represent numbers}}$

Yesterday: 10 emails

Total: 30 emails → Perform addition

Email1 => subject: Hi John → Represent text

Represent Letters

- Come up with an arbitrary convention and associate with numbers, and then use binary representation.
- A => 65 => 01000001
- B => 66 => 01000010
- C => 67 => 01000011
- D => 68 => 01000100

Represent Letters

- Come up with an arbitrary convention to associate with numbers, and then use binary representation.
- A => 65 => 01000001
- B => 66 => 01000010
- C => 67 => 01000011
- D => 68 => 01000100
- Represent words CAB => 01000011 01000001 01000010

Represent Colors

- Three value for intensities of Red, Green, Blue
- (54, 72, 32)
  - (intensity for red, intensity for green, intensity for blue)
  - Represent in binary

How do computers work?

Today: 20 emails → Represent numbers
Yesterday: 10 emails
Total: 30 emails → Perform addition

Email1 => subject: Hi John → Represent text

Computations

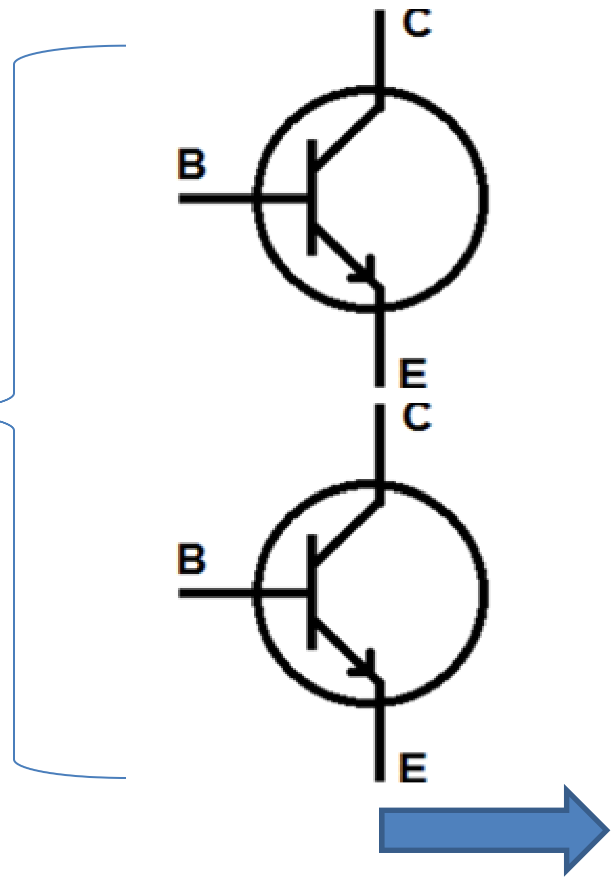
- Can we do Arithmetic?

Computations

- Can we do Arithmetic (Addition)?
  - Need to take at least two inputs, and operate on them.
  - We will build logic gates to accomplish this.
  - How do we build logic gates?
    - Cleverly place transistors to create circuits.

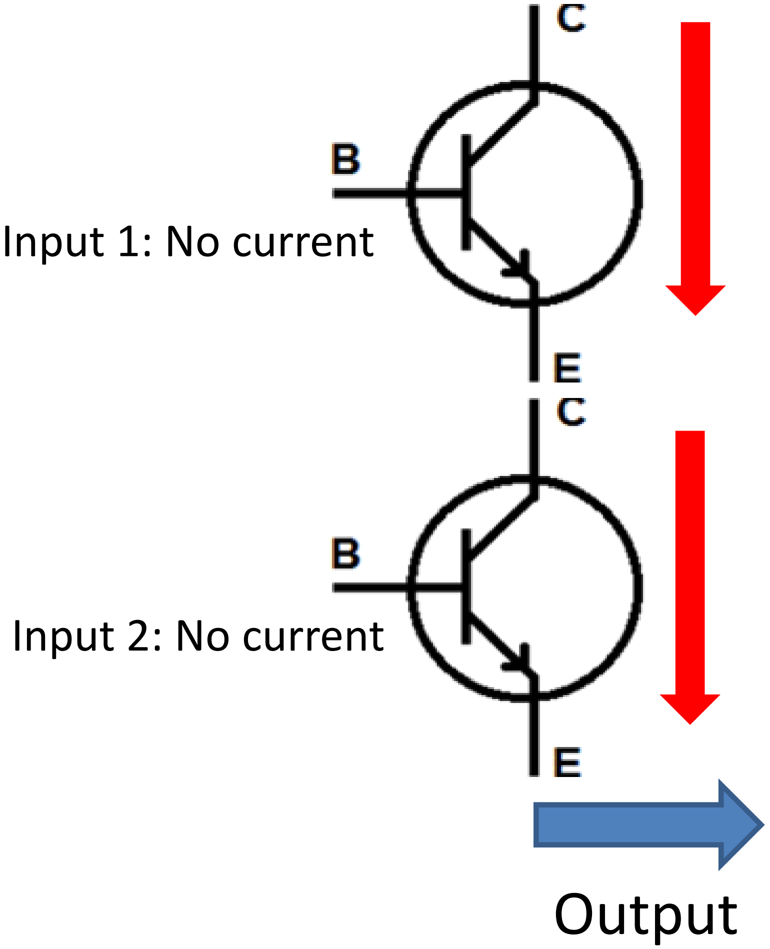
Two transistors in parallel

Two switches
to control
Or two
inputs



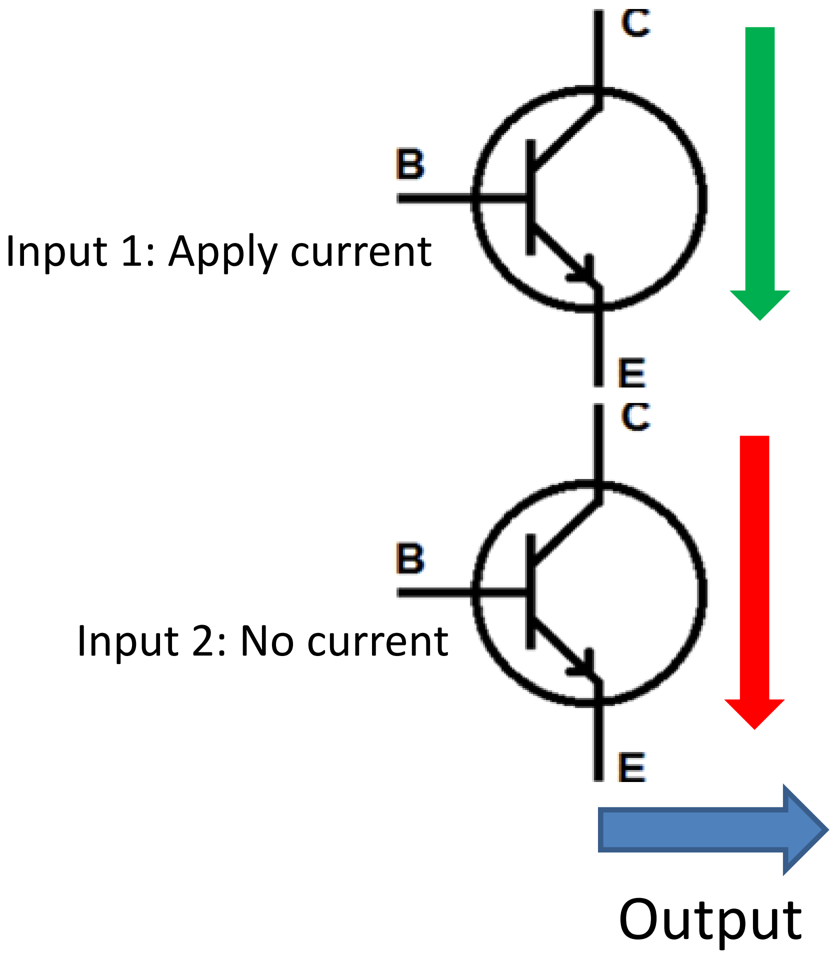
Output

Two transistors in parallel



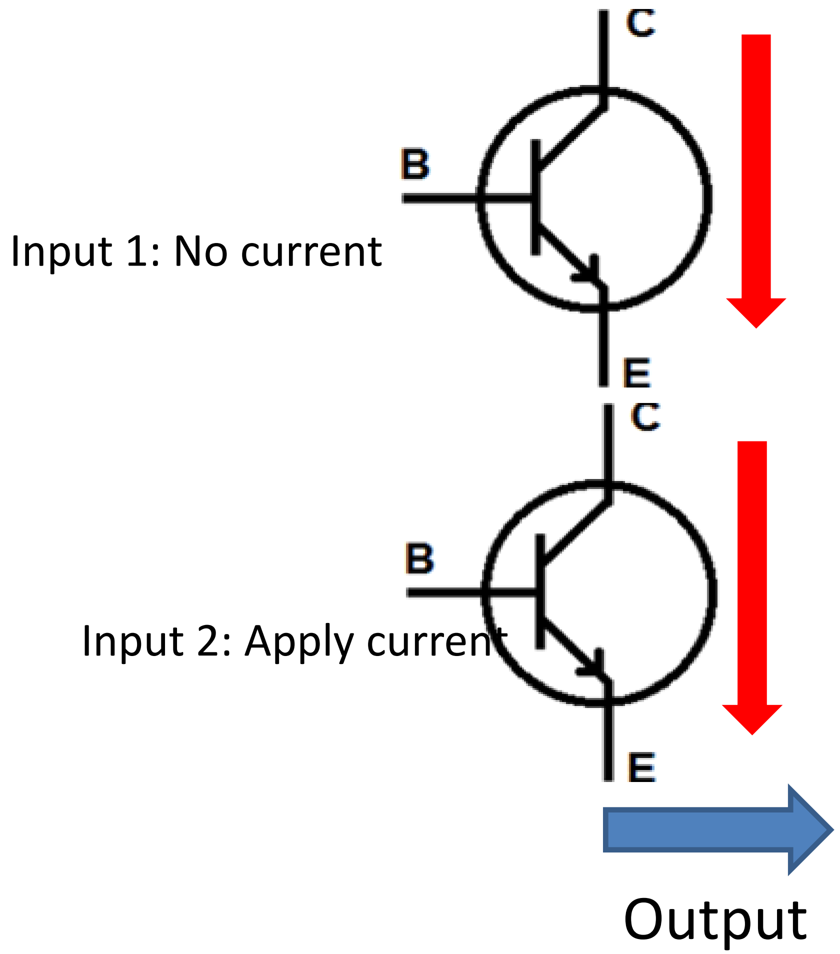
Input 1	Input 2	Output
0	0	?

Two transistors in parallel



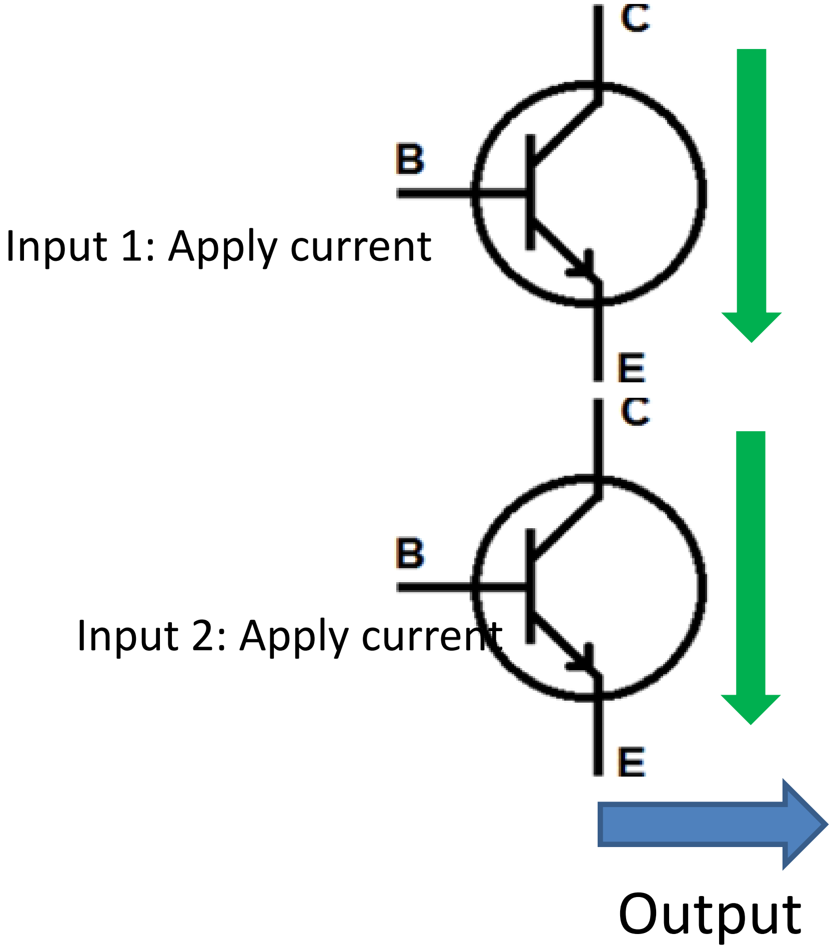
Input 1	Input 2	Output
0	0	0
0	1	?

Two transistors in parallel



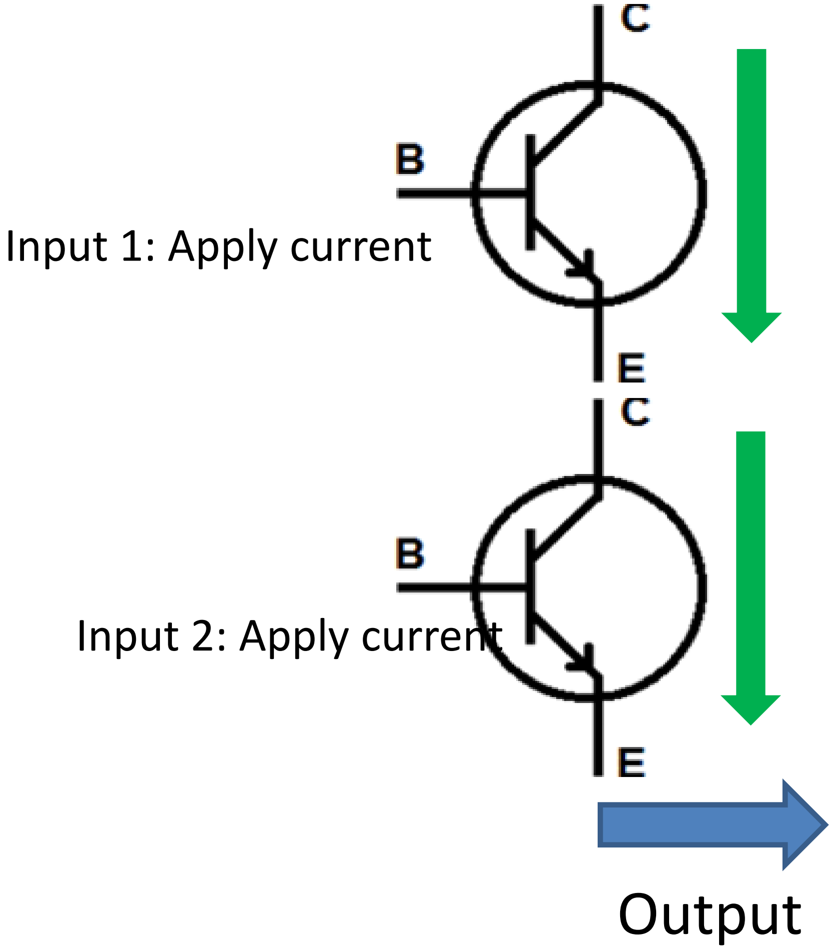
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	?

Two transistors in parallel



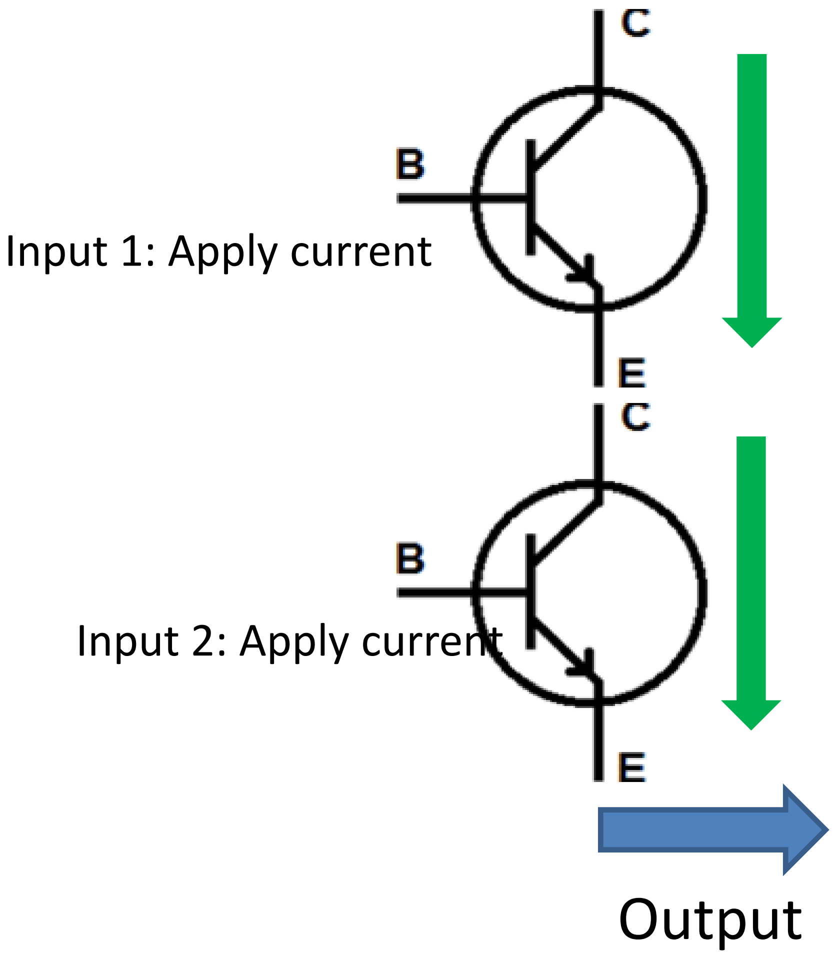
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	?

Two transistors in parallel



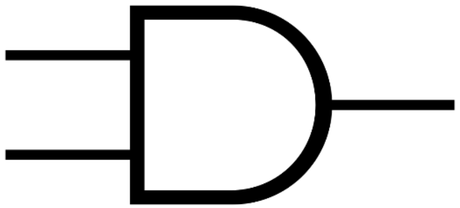
Input 1	Input 2	Output
0	0	0
1	0	0
0	1	0
1	1	1

Two transistors in parallel: Logic Gate



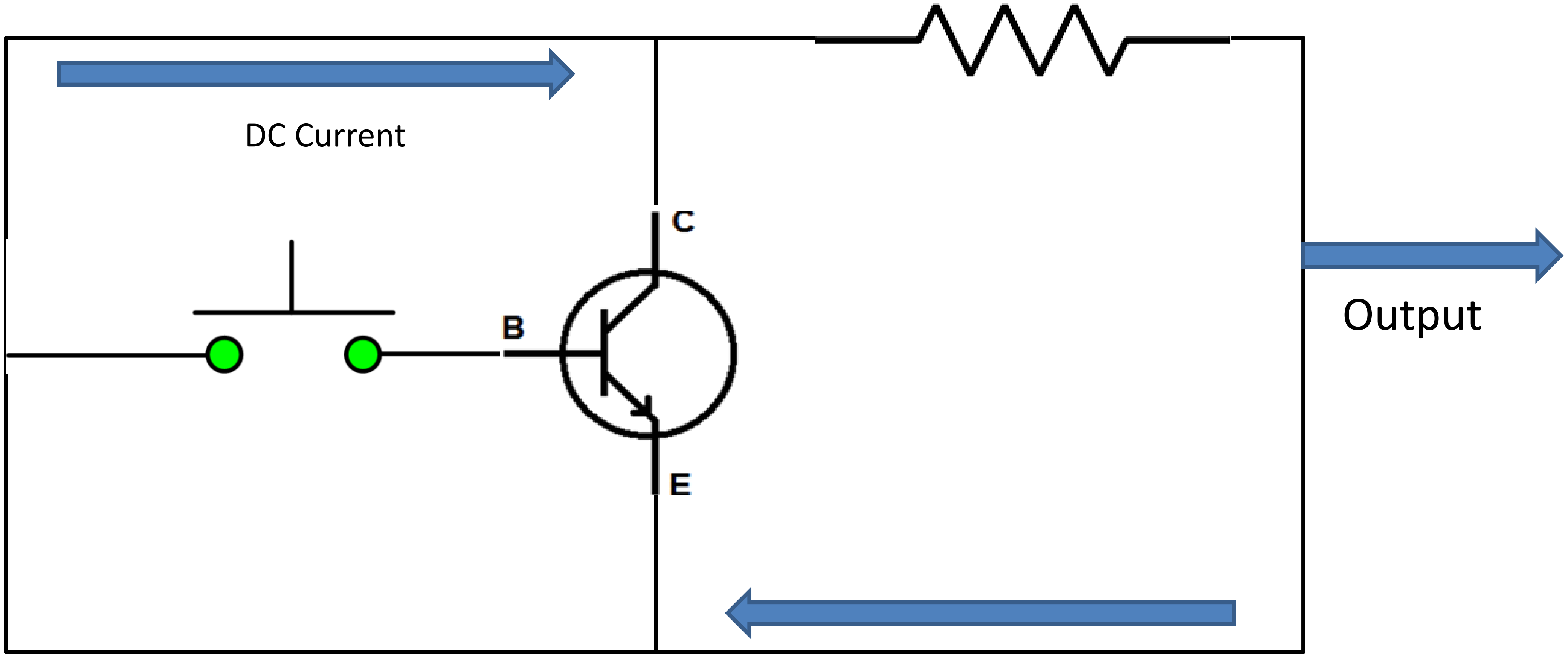
Input 1	Input 2	Output
0	0	0
1	0	0
0	1	0
1	1	1

AND Gate

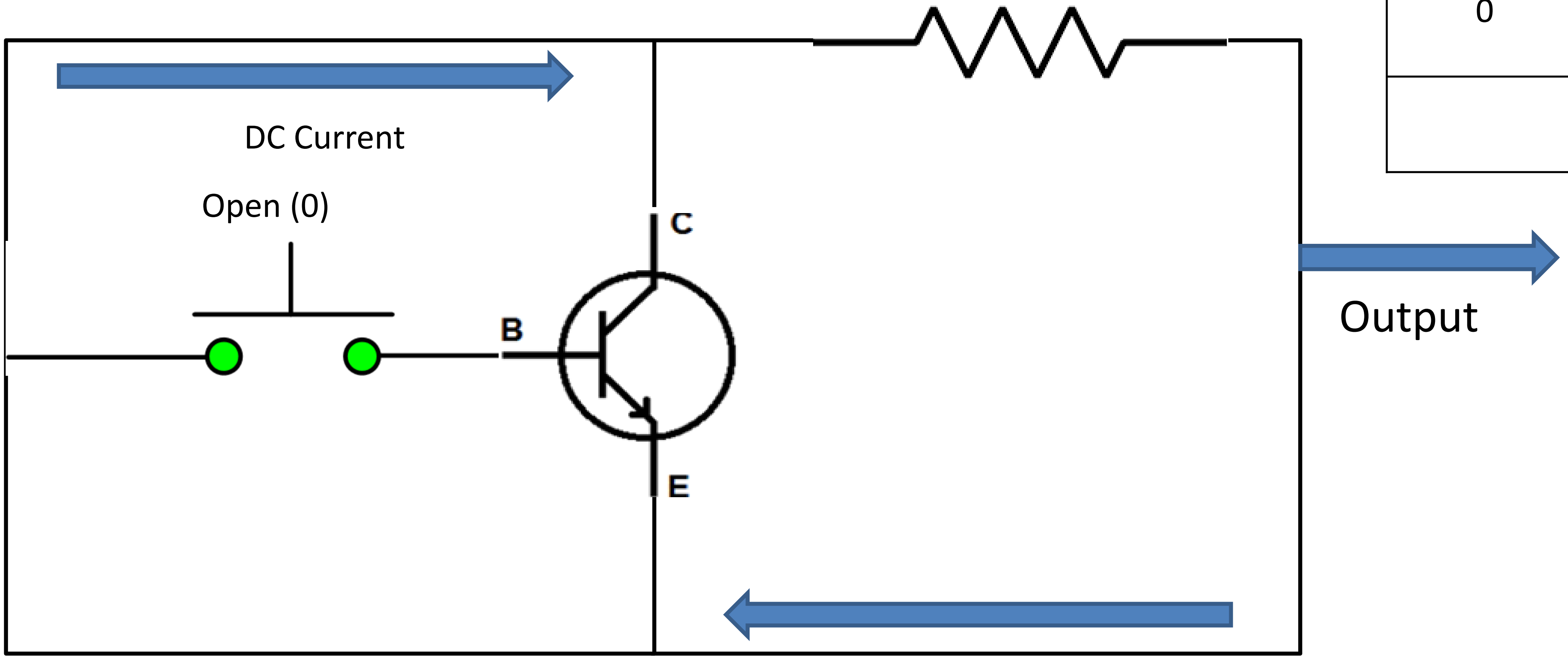


Symbol

Another logic gate

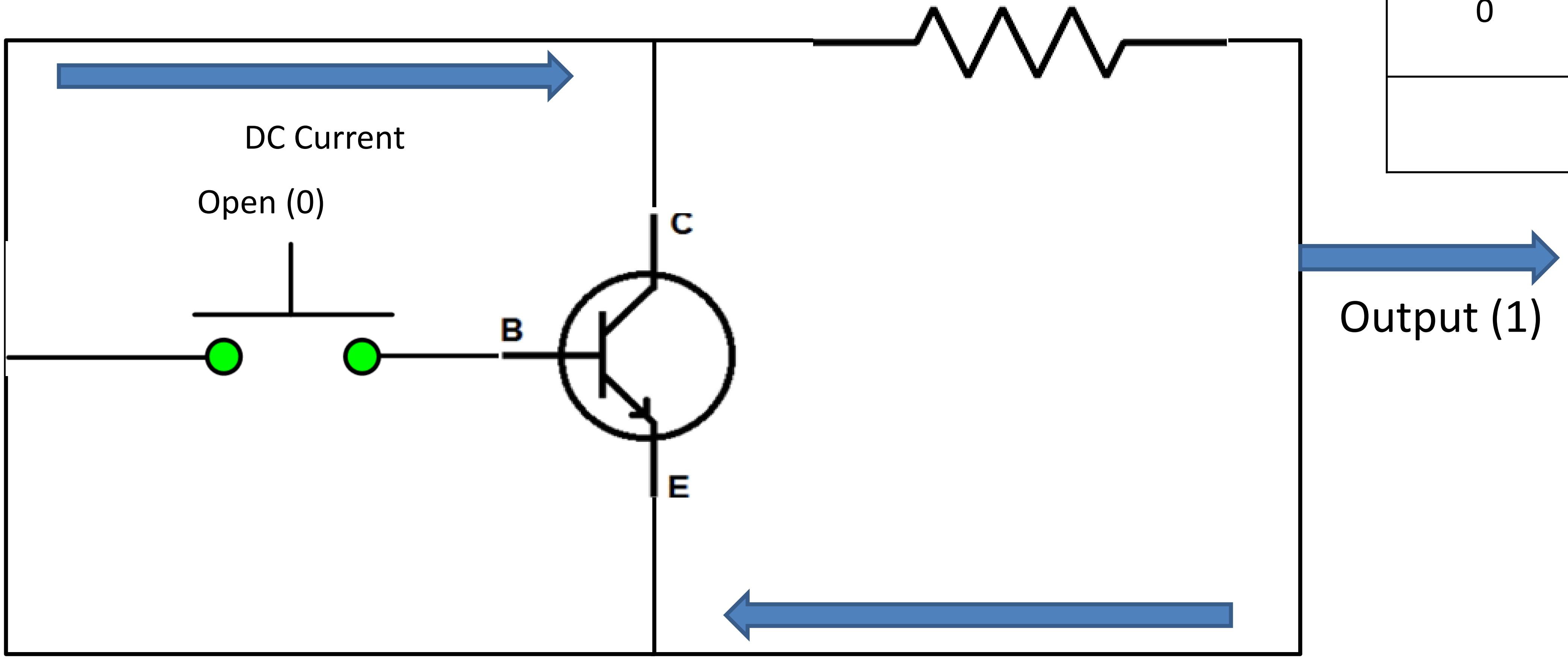


Another logic gate



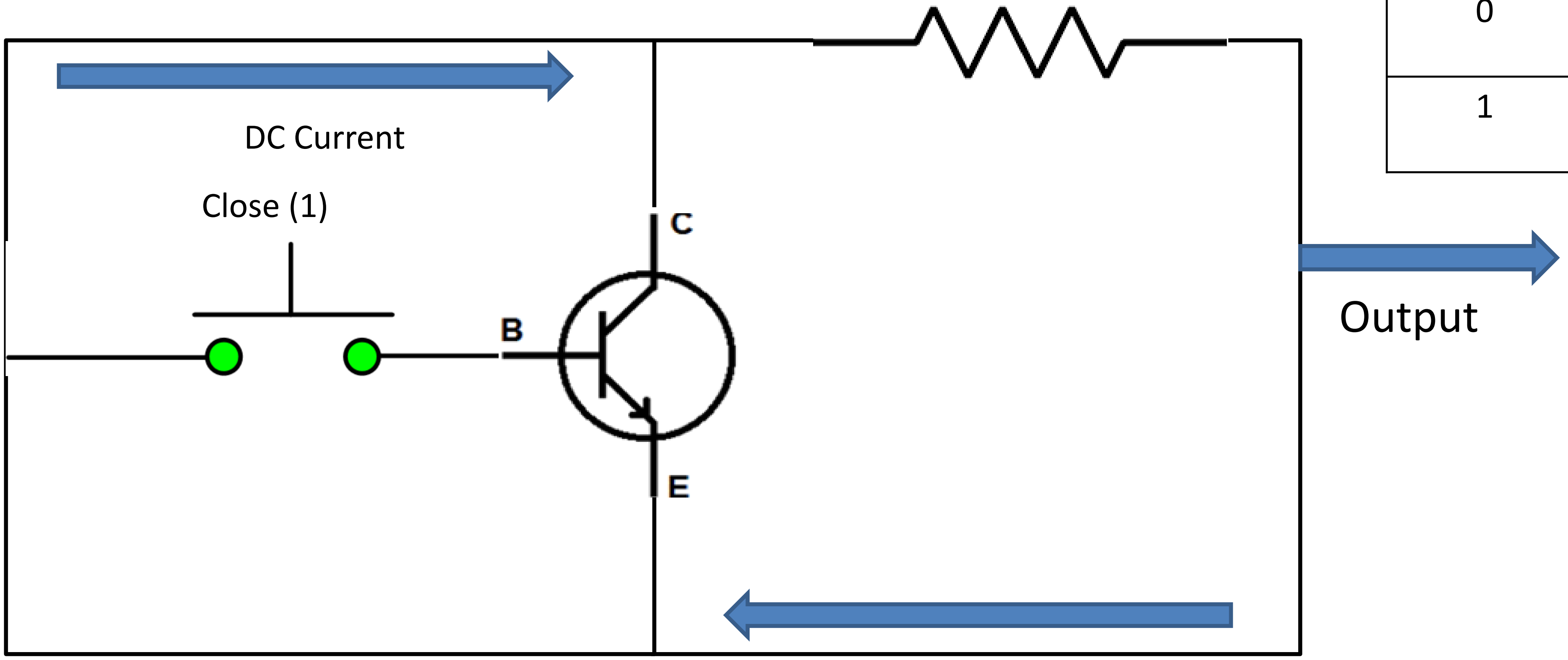
Input	Output
0	?

Another logic gate



Input	Output
0	1

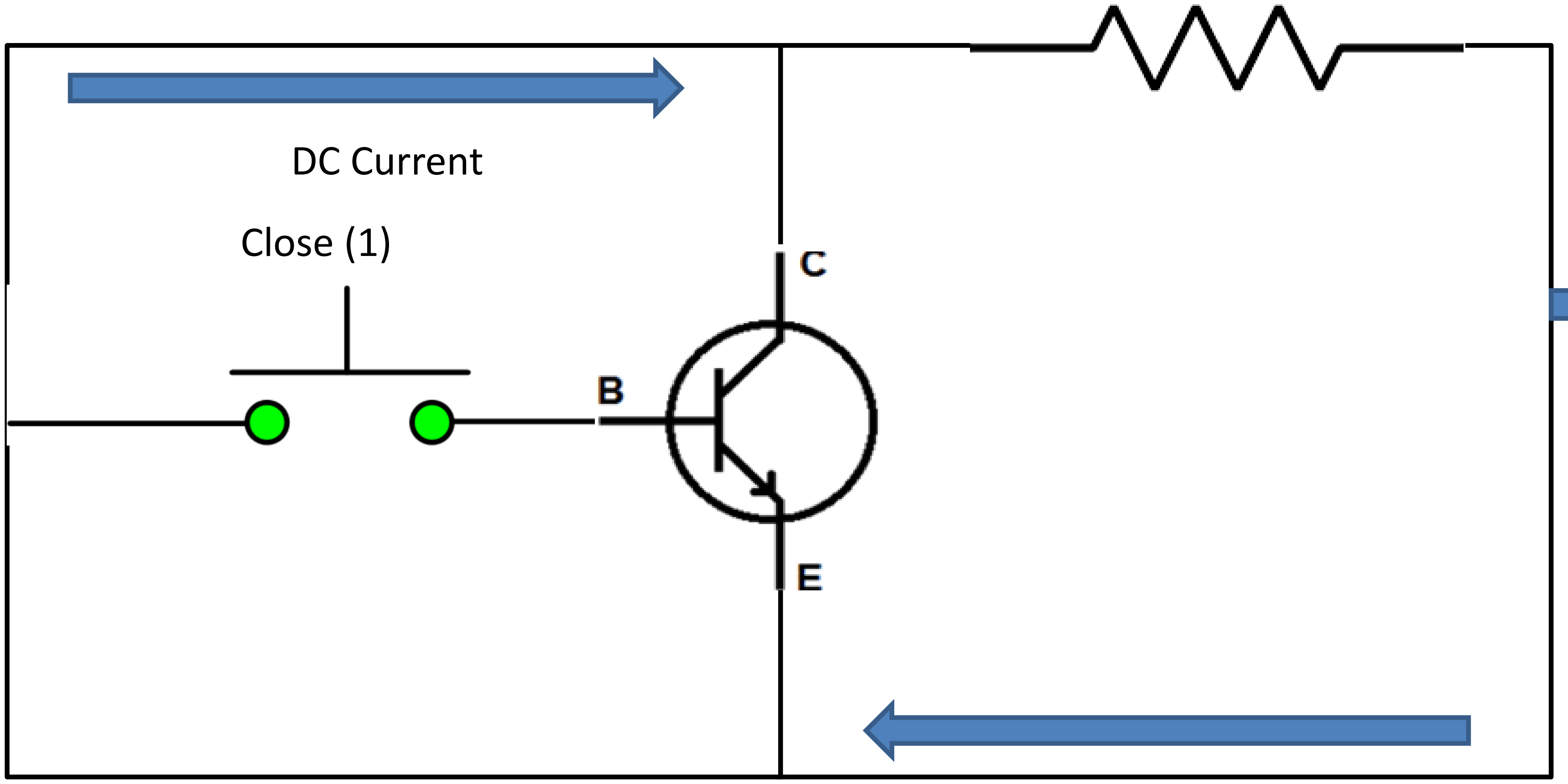
Another logic gate

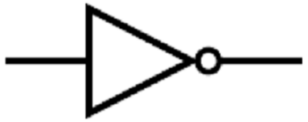


Input	Output
0	1
1	?

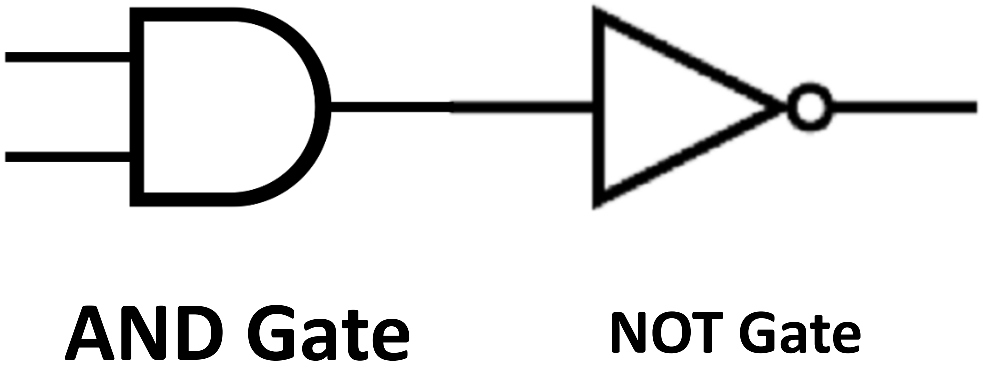
Another logic gate

Input	Output
0	1
1	0



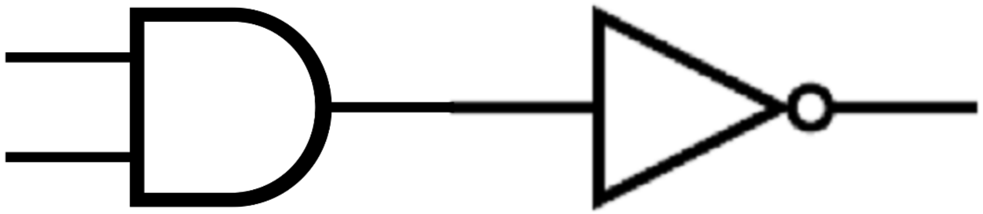


NAND Gate



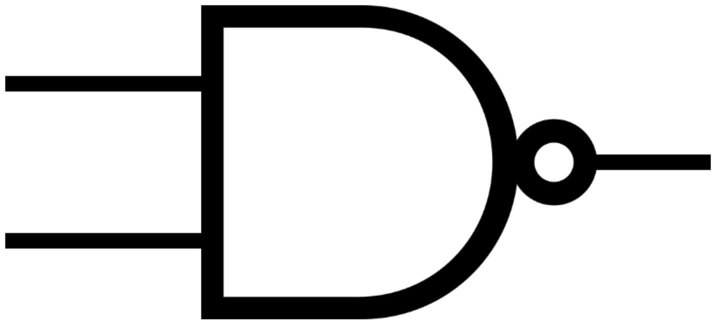
Input 1	Input 2	AND	NAND
0	0	0	
1	0	0	
0	1	0	
1	1	1	

NAND Gate



AND Gate

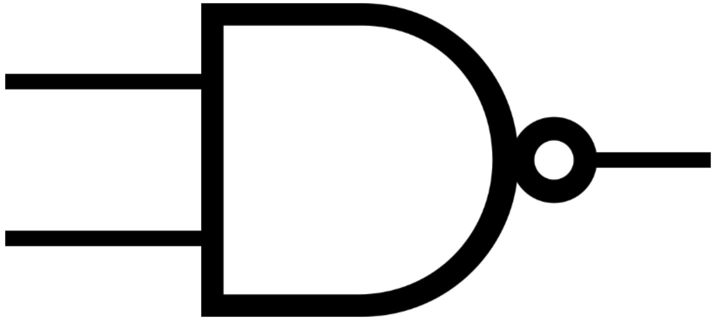
NOT Gate



NAND Gate

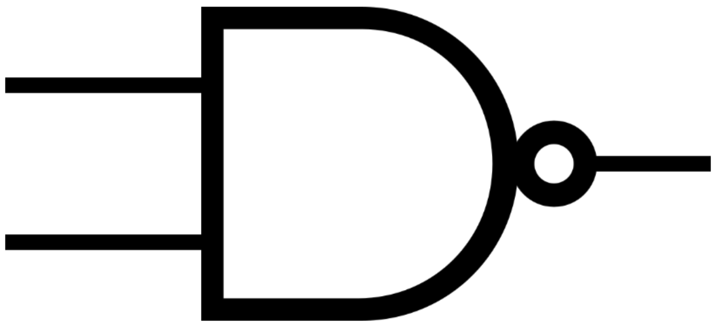
Input 1	Input 2	AND	NAND
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

One More Logic Gate



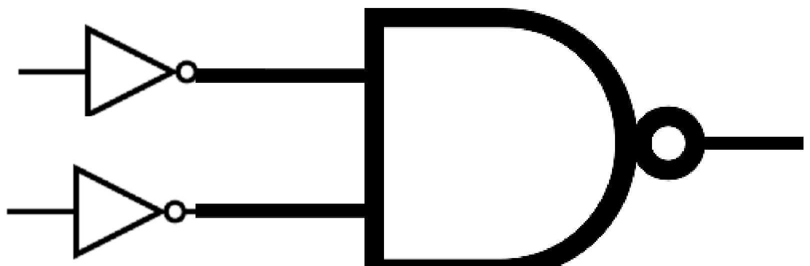
Input 1	Input 2	NAND
0	0	1
1	0	1
0	1	1
1	1	0

One More Logic Gate



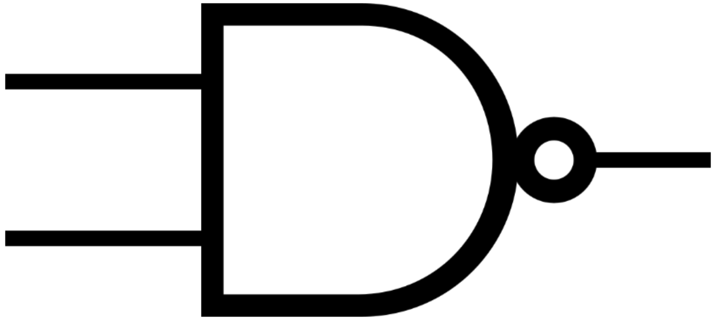
NAND Gate

Input 1	Input 2	NAND
0	0	1
1	0	1
0	1	1
1	1	0



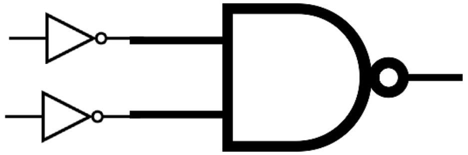
Input 1	Input 2	Output
0	0	?
1	0	
0	1	
1	1	

One More Logic Gate



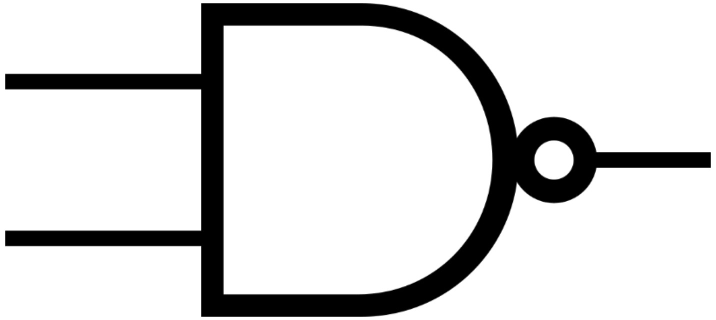
NAND Gate

Input 1	Input 2	NAND
0	0	1
1	0	1
0	1	1
1	1	0



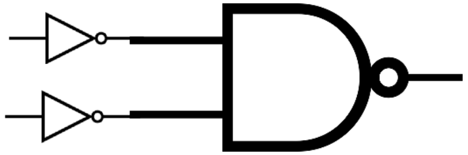
Input 1	Input 2	~Input 1	~Input 2	Output
0	0	1	1	0
1	0			
0	1			
1	1			

One More Logic Gate



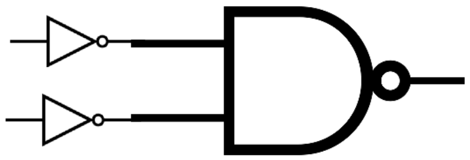
NAND Gate

Input 1	Input 2	NAND
0	0	1
1	0	1
0	1	1
1	1	0

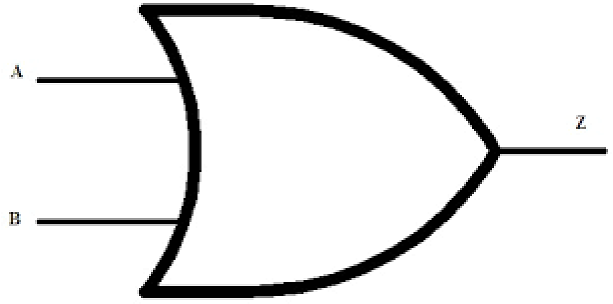


Input 1	Input 2	~Input 1	~Input 2	Output
0	0	1	1	0
1	0	0	1	1
0	1	1	0	1
1	1	0	0	1

OR Gate



Input 1	Input 2	Output
0	0	0
1	0	1
0	1	1
1	1	1



OR Gate

ADD Numbers (Simplified)

- Add two one-bit numbers, and produce two-bit results
- Input can be 0 or 1

ADD Numbers (Simplified)

- Add two one-bit numbers, and produce two-bit results
- Input can be 0 or 1

Input 1 (base 10)	Input 2 (base 10)	Output (base 10)
0	0	0
0	1	1
1	0	1
1	1	2

Decimal

ADD Numbers (Simplified)

- Add two one-bit numbers, and produce two-bit results
- Since input is one-bit numbers, Input can be 0 or 1

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	2

Decimal

Input 1	Input 2	Output 1	Output2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Binary

Design an Adder

Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

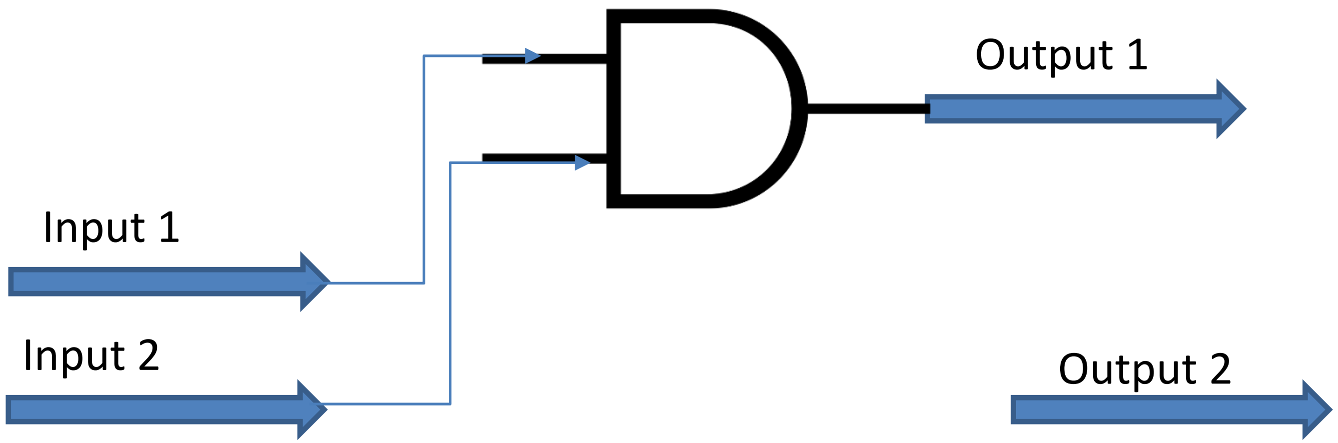
Binary

Design an Adder

Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Binary

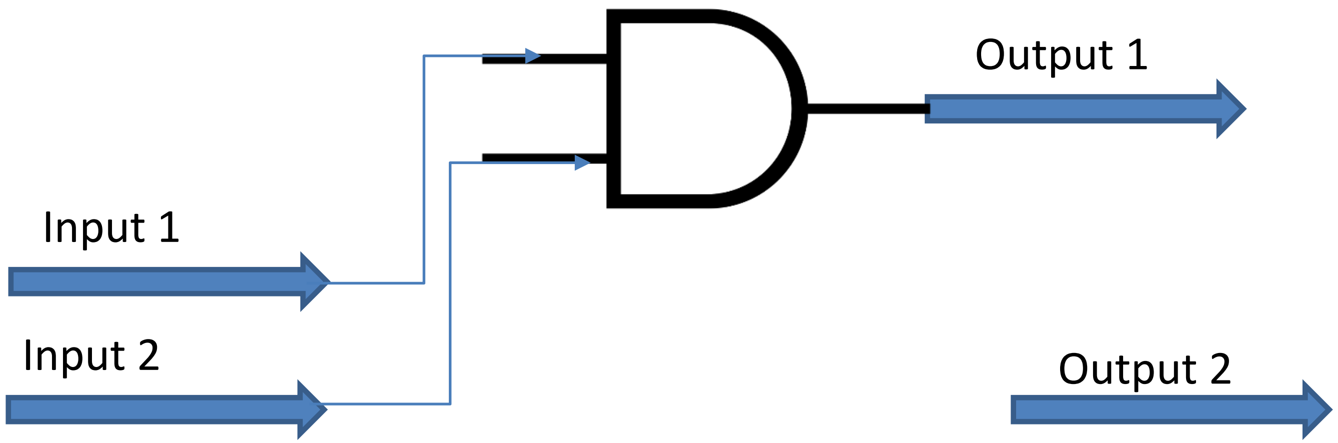
Design an Adder



Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Binary

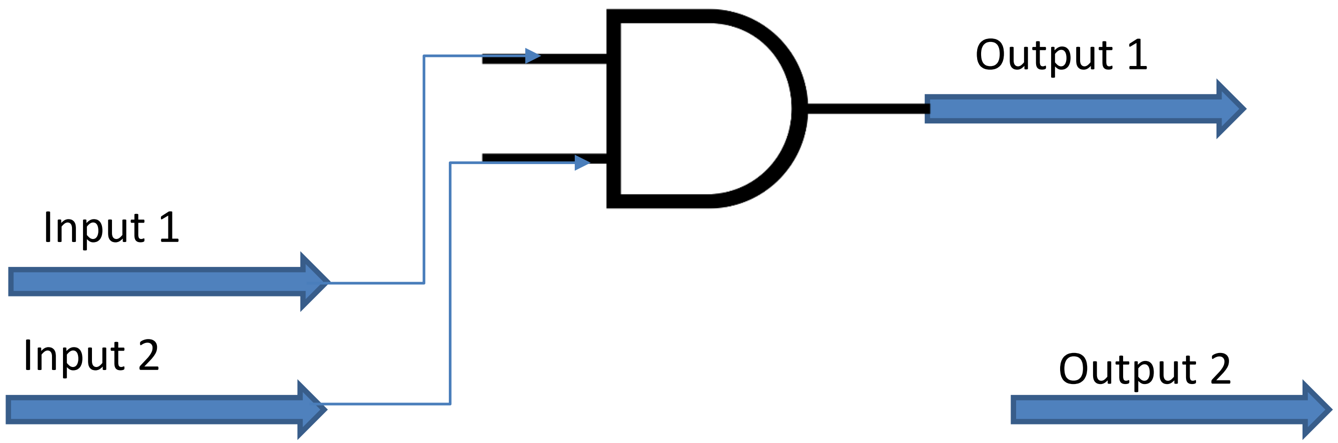
Design an Adder



Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Binary

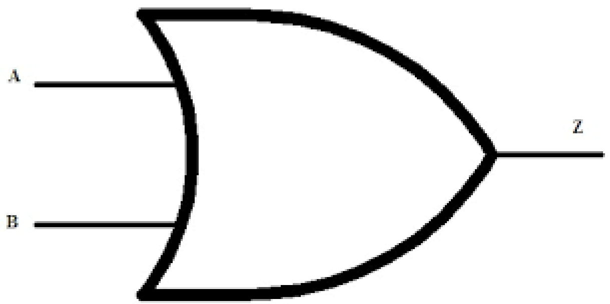
Design a Adder



It looks similar to OR Gate.
Except the last row is inverted where both inputs are 1, the result in inverted.

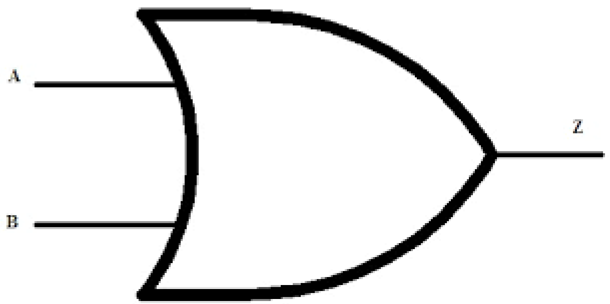
Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Binary



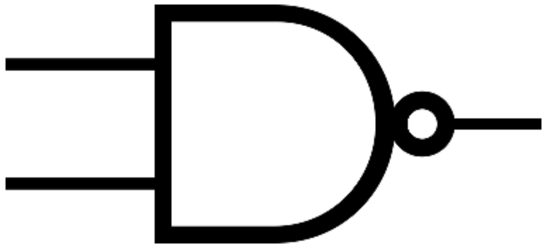
OR Gate

Input 1	Input 2	Output
0	0	0
1	0	1
0	1	1
1	1	1



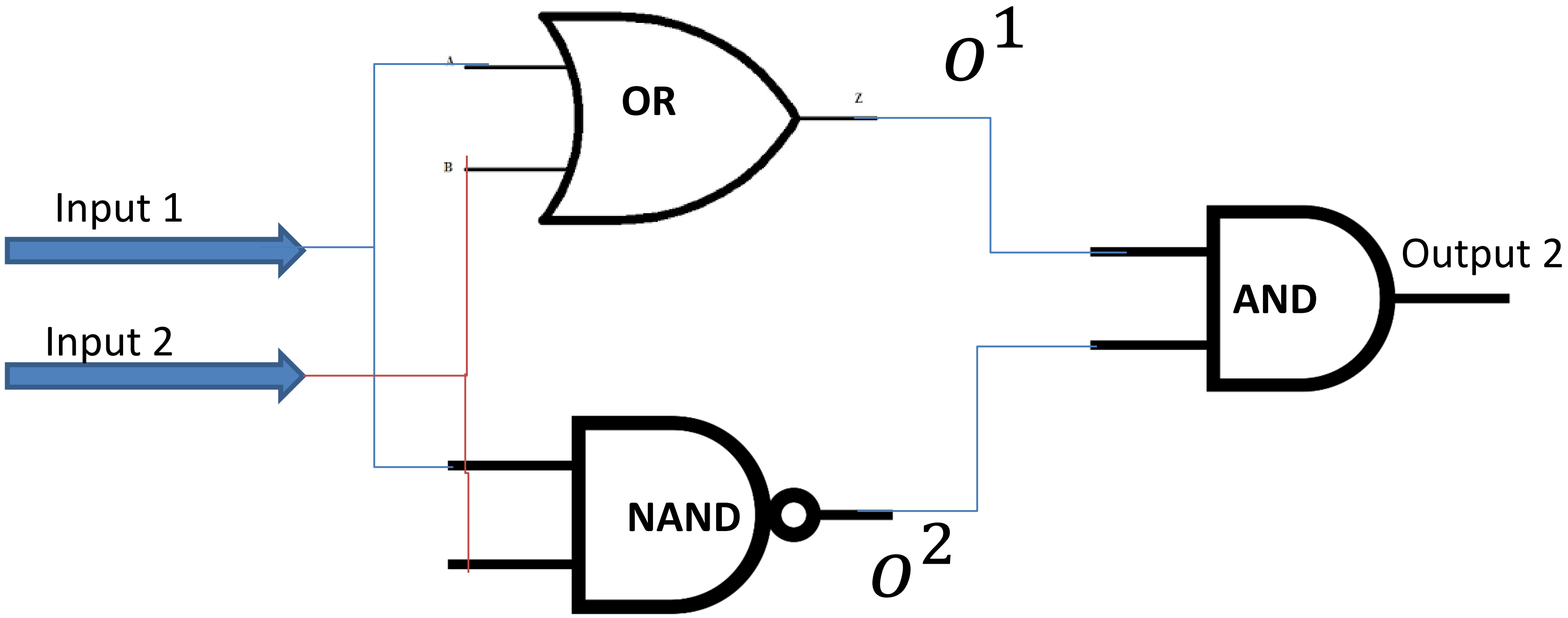
OR Gate

Input 1	Input 2	Output
0	0	0
1	0	1
0	1	1
1	1	1

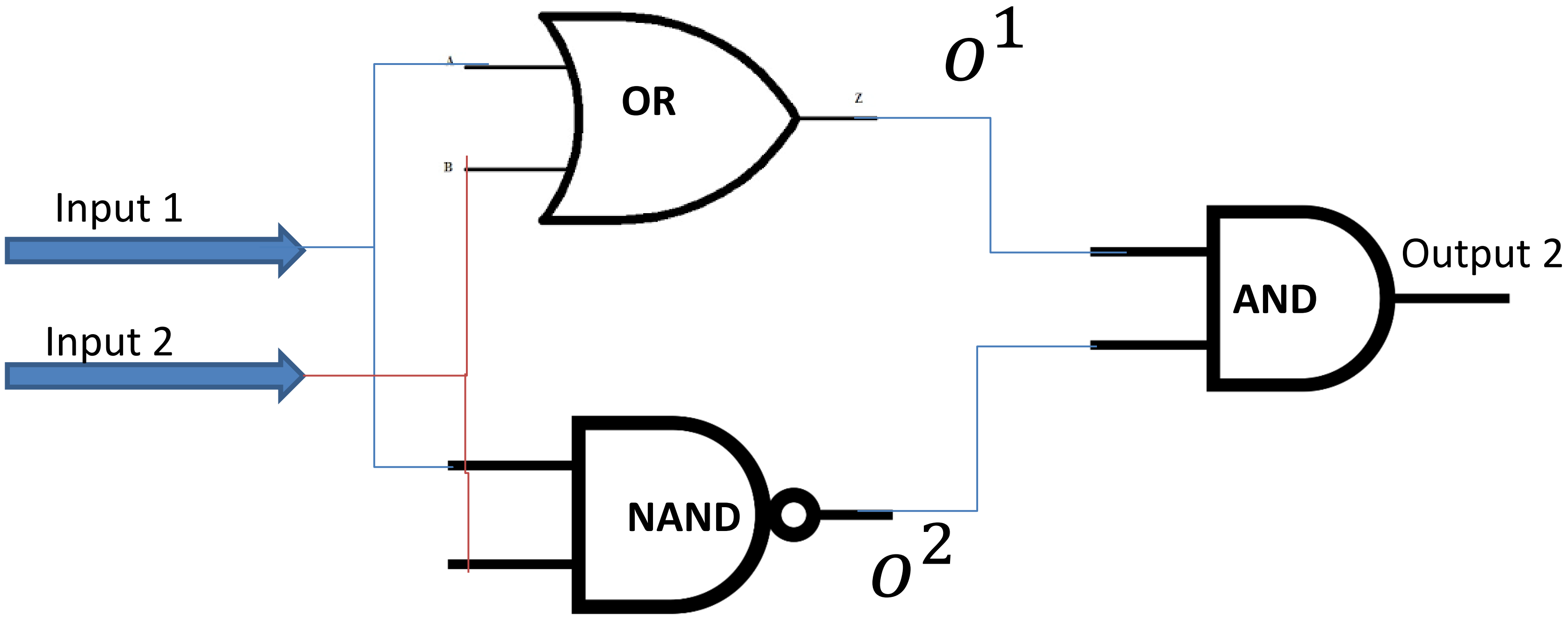


NAND Gate

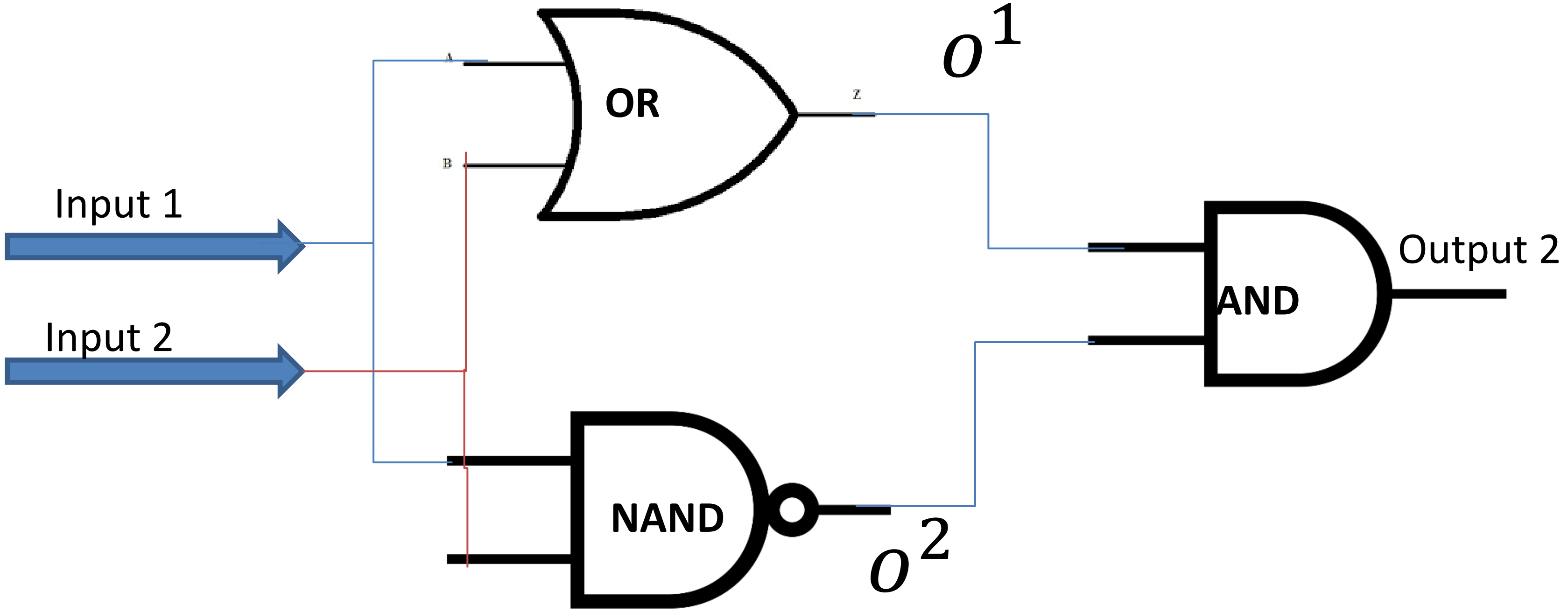
Input 1	Input 2	NAND
0	0	1
1	0	1
0	1	1
1	1	0

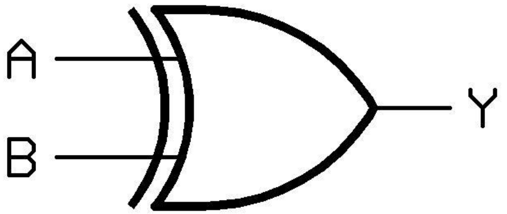


Input 1	Input 2	o^1	o^2	Output 2
0	0	0		
1	0	1		
0	1	1		
1	1	1		



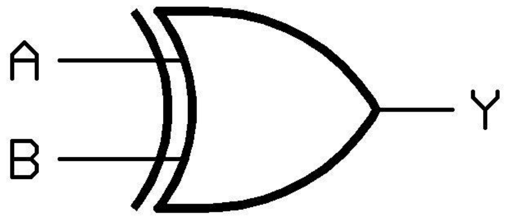
Input 1	Input 2	o^1	o^2	Output 2
0	0	0	1	
1	0	1	1	
0	1	1	1	
1	1	1	0	





XOR Gate

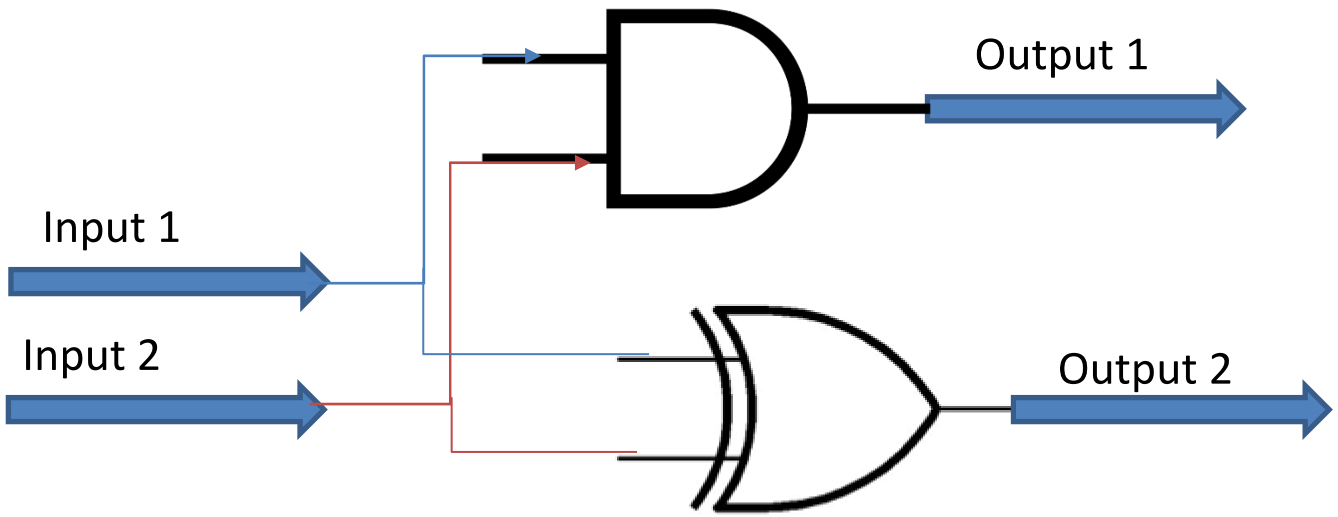
Input 1	Input 2	o^1	o^2	Output 2
0	0	0	1	0
1	0	1	1	1
0	1	1	1	1
1	1	1	0	0



XOR Gate

Input 1	Input 2	Output 2
0	0	0
1	0	1
0	1	1
1	1	0

Design an Adder



XOR Gate

It looks similar to OR Gate.
Except the last row is inverted where both inputs are 1, the result in inverted.

Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Binary

How do computers work?

Today: 20 emails → Represent numbers

Yesterday: 10 emails

Total: 30 emails → Perform addition

Email1 => subject: Hi John → Represent text

Many Such Circuits Exists

- There are many such circuits available to
  - Load data, store data, add, subtract and perform logical ops on data.

Questions

1. What operations can hardware perform? How to instruct computer to perform a certain operation? How are negative numbers/exponentials represented?
2. How do we perform addition, multiplication, division?
3. How do we improve the speed of the computer? Can we do things in parallel (compute while loading next data, etc.)
4. Where is data stored? How can we make it efficient?
5. Can we perform computations in parallel to improve performance?
6. How do we define performance?

Computer Architecture

Computer Architecture:

- refers to attributes of a system visible to a programmer
- includes the **instruction set architecture (ISA)** which defines instruction formats, instruction codes, number and names of registers, etc.

Computer Organization:

- refers to the operational units and their interconnections that realize the architectural specifications
- Examples: number of bits used to represent various data types (e.g., numbers, characters), I/O mechanisms,

- Computer Organization ≠ Computer Architecture

e.g. Intel and AMD CPUs offer nearly identical architecture (they support the same set of operations), but have a very different (internal) organization.

History of Computers

First Generation: Vacuum Tubes

- Vacuum tubes were used for digital logic and memory
- IAS computer
  - Fundamental design approach was the stored program concept
    - Attributed to the mathematician John von Neumann, but significant contributions and developments happening all around the world (Alan Turing, Konrad Zuse, etc.)
    - First publication of the idea was in 1945 for the EDVAC
  - Design began at the Princeton Institute for Advanced Studies
  - Completed in 1952
  - Prototype of all subsequent general-purpose computers

Second Generation: Transistors

- Invented at Bell Labs in 1947
- Smaller, Cheaper and Dissipates less heat than a vacuum tube
- Is a *solid state device* made from silicon
- fully transistorized computers were commercially available in the late 1950's

Image source: https://www.thoughtco.com/what-is-a-transistor-2698913

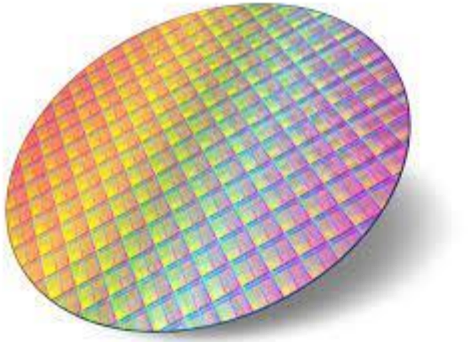
Second Generation Computers

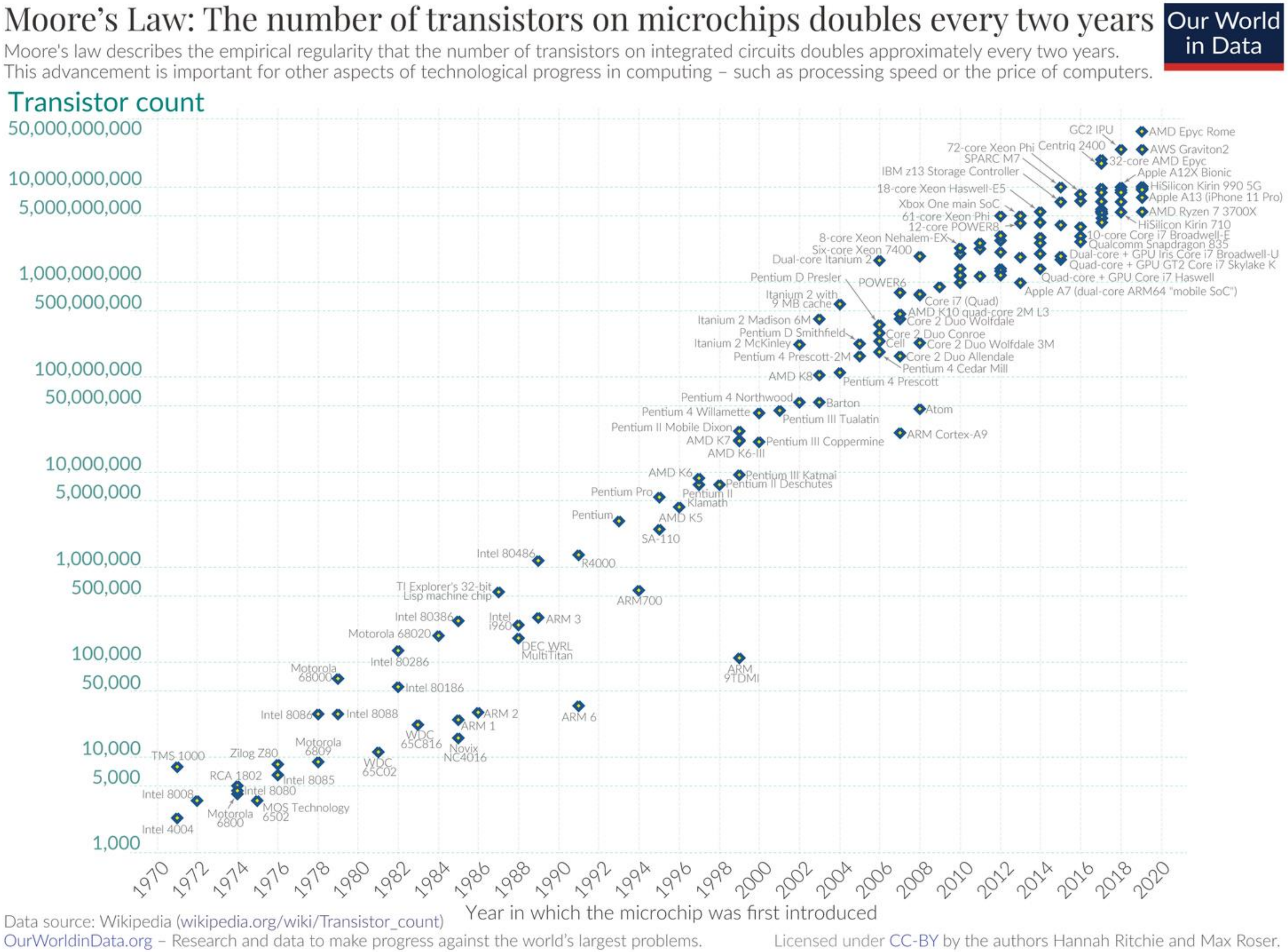
- Introduced:
  - More complex arithmetic and logic units and control units
  - The use of high-level programming languages
  - Provision of *system software* which provided the ability to:
    - Load programs
    - Move data to peripherals
    - Libraries perform common computations

Third Generation: Integrated Circuits

- 1958 – the invention of the integrated circuit
- Exploits the fact that transistors can be fabricated from a semiconductor such as silicon

- Many transistors can be produced at the same time on a single wafer of silicon
- Transistors can be connected with a processor metallization to form circuits





By Max Roser, Hannah Ritchie - https://ourworldindata.org/uploads/2020/11/Transistor-Count-over-time.png

Moore’s Law

- Moore observed that the number of transistors that could be integrated on a single chip doubled roughly every 18 month
- Cost of a chip manufacturing remained virtually unchanged => cost of computer logic and memory circuitry has decreased
- Logic and memory elements are placed closer together =>electrical path length is shortened, increasing operating speed
- The computer becomes smaller, making it more convenient to place in a variety of environments
- Reduction in power and cooling requirements