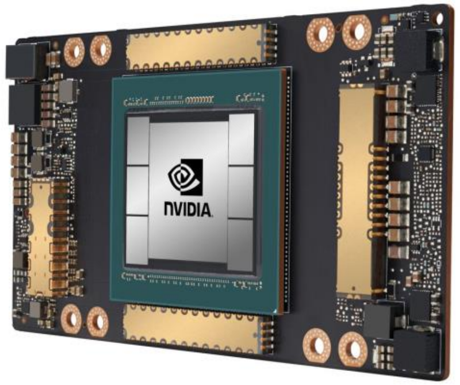
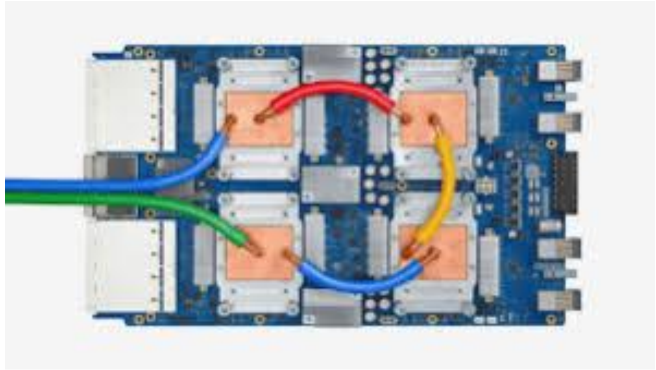
# CSCB58: Computer Organization




Prof. Gennady Pekhimenko

University of Toronto

Fall 2020



*The content of this lecture is adapted from the lectures of Larry Zheng and Steve Engels*

# CSCB58 Week 1

# Who are We?

Instructor:

Assistant Professor **Gennady Pekhimenko**

Office: BA5232 and IC454 (normally ;))

pekhimenko@cs.toronto.edu

http://www.cs.toronto.edu/~pekhimenko/

# Who are We?

TAs:

- Bojian Zheng, PhD student (bojian.zheng@mail.utoronto.ca)
- Qiongsi Wu, MSc. Student (qiongsi.wu@mail.utoronto.ca)
- Anand Jayarajan, PhD student (anandj@cs.toronto.edu)
- Mustafa Quraish, PhD student (mustafa@cs.toronto.edu)

# How to Connect?

- Lectures:

https://vectorinstitute.zoom.us/j/93537472871 (password: 597255)

- PRA (Mon):

https://utoronto.zoom.us/j/94009704386 (password: 708254)

- PRA (Tue):

https://utoronto.zoom.us/j/95205933572 (password: 045630)

- PRA (Thu):

https://utoronto.zoom.us/j/94993998484 (password: 713436)

# Today's outline

- **Why** CSCB58
- **What** is in CSCB58
- **How** to do well in CSCB58

- Start learning

# **Why** take CSCB58?

# “Why are you making me take this?”

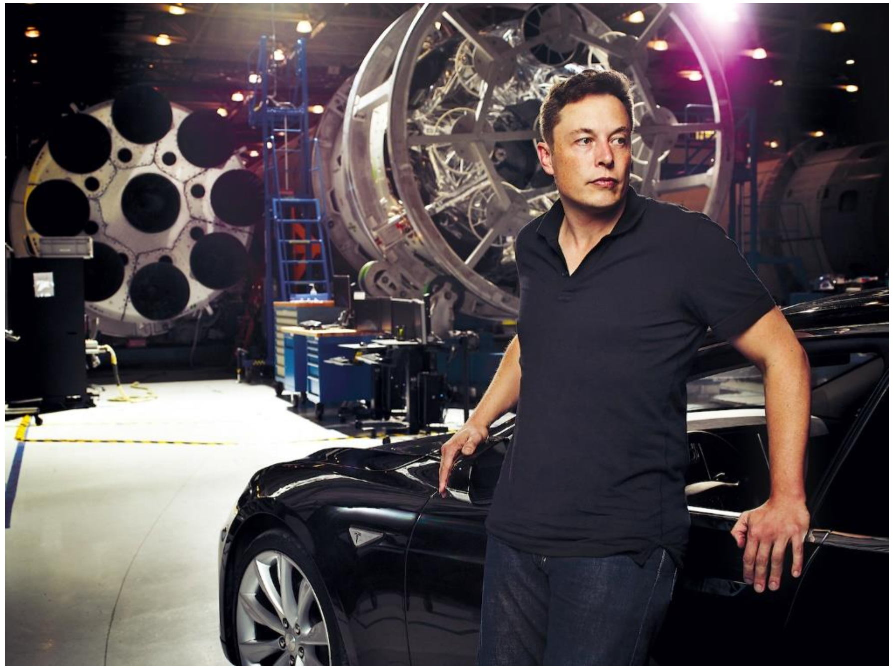
- CSCB58 isn’t needed if you’re just a causal technology user

  You can still drive a car, even if you don’t understand how the engine works

# “Why are you making me take this?”

- Computer science majors aren’t casual technology users

  At the very least, you’ll need to know how the programs you write are affected by hardware



# Learning the Magic

Si

magic

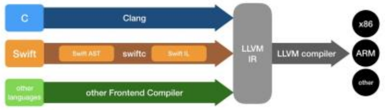
Hardware

magic

OS

magic

Compiler




CSCB58

CSCC69

CSD70

# More specifically…

- How do we express 1's and 0's using a piece of silicon?
- How does the computer do everything with just 1's and 0's?
- What is stored in that "fortnite.exe" file, what exactly happens when I double-click on it?
- How does the CPU run an if-statement, or for loop, or recursion?

**CSCB58 has all the answers!**

# After learning CSCB58…

- You’ll know everything about how a computer is physically built, and you can build one if you want.

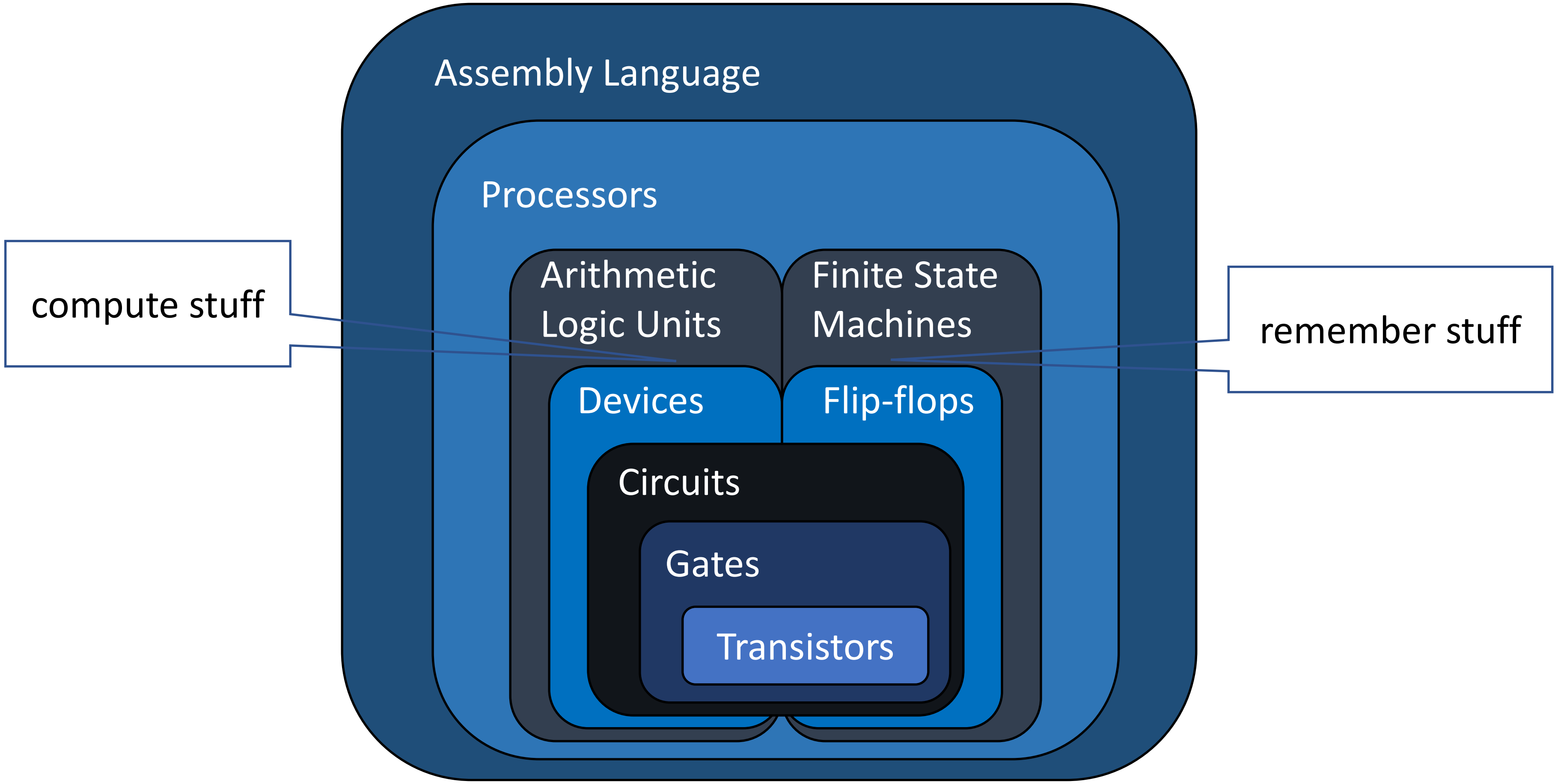
- With your hardware knowledge, you will be able to engineer the performance of your software like never before.

***People who are really serious about software should make their own hardware.***

***-- Alan Kay***

# **What's** in CSCB58?

# The architecture of a computer **hardware**, level by level, bottom-up



# We learn the whole real deal

- Computing from the ground up:
  - From atom level to assembly level
- Above the assembly level is the Operating System, which **virtualizes** the hardware

- Almost everything you learn from CS courses are **virtualizations/illusions**, except for **CSCB58**

# We learn how to handle **abstractions**

- At each level, we see how the previous layer is ***abstracted***
- In the end, we want to know how the underlying hardware affects us as programmers … so we can ignore the detail.

# **How** to do well in CSCB58

# First of all …

Be interested

# Course website

https://cscb58f20.ml/

**All course materials are here**

# Marking scheme

Grades will be based on lab exercises, classroom quizzes, assignment, and the final exam.

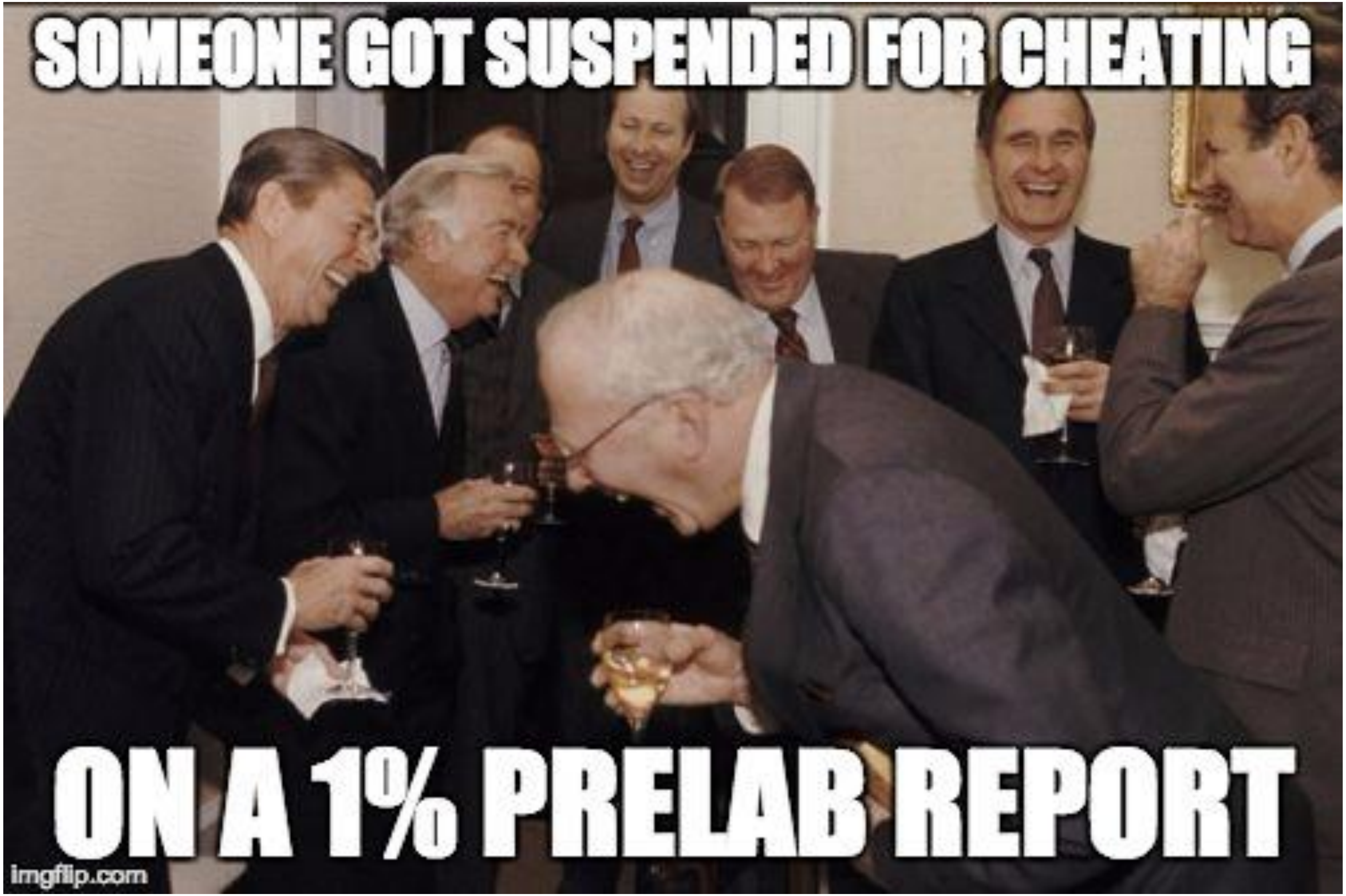
| Type | Description | Due Date | Weight |
|---|---|---|---|
| Lab | 10 weeks | weekly | 50% (5% each) |
| Quiz | Appx. 10 weeks | weekly | 10% (appx. 1% each) |
| Assignment | Assembly project | TBA | 15% |
| Final exam | Take-home; Must get at least 40% | TBA | 25% |

# Labs (starting from Week 2)

- Hands-on exercises in which you will build real pieces of hardware.
- Pre-labs and in-labs are all done **individually**.
- ONLY go to the lab section that you are registered to on ACORN. If you want to switch lab section, find someone who is willing to switch and get permission from Gennady.

# Prelab Reports

- For most of the labs, you will be required to submit a prelab report (a PDF file) to MarkUs before the labs start
- Must be completed **individually**
- Submission deadline is typically before the lab you are taken
- To get the mark for the prelab report
  - do your work and submit something meaningful
  - **don’t plagiarize**
  - not submitting anything would be much better than plagiarizing


SOMEONE GOT SUSPENDED FOR CHEATING
ON A 1% PRELAB REPORT
imgflip.com

# Lab software

- We will use Logisim (latest version). NOTE: please, use the version that will be provided/referenced by TAs

- The reference of the software will be (has been) posted on the course website

- **Task for this week: download the software, read the reference, and familiarize yourself with it**

# Weekly Quizzes

- We will use Quercus for online quizzes
- Starting in Week 2 (including **Monday PRA**)
- Useful practices for tests and exams
- Quizzes will be **every Friday in-class** (first **15 minutes**)

# New: Assembly Programming Project

- This year, we'll ask you to write a larger project in assembly than in prior years
- The project will be due on the last day of class

- You will be asked to write assembly code to build a game
  - Memory (storing game state)
  - Syscalls (to access input/output)
  - …

# Tests

- No midterm

- Final exam
  - Some time end of semester
  - Will be take-home
  - Must get >= 40%
  - Details will come later

# Discussion board on Piazza

https://piazza.com/class/kemxsy9by7o707

- All course announcements will be posted here.
- **Daily** reading is required.

# Office hours


Fridays, 12pm (please, give heads up/email if you are coming)

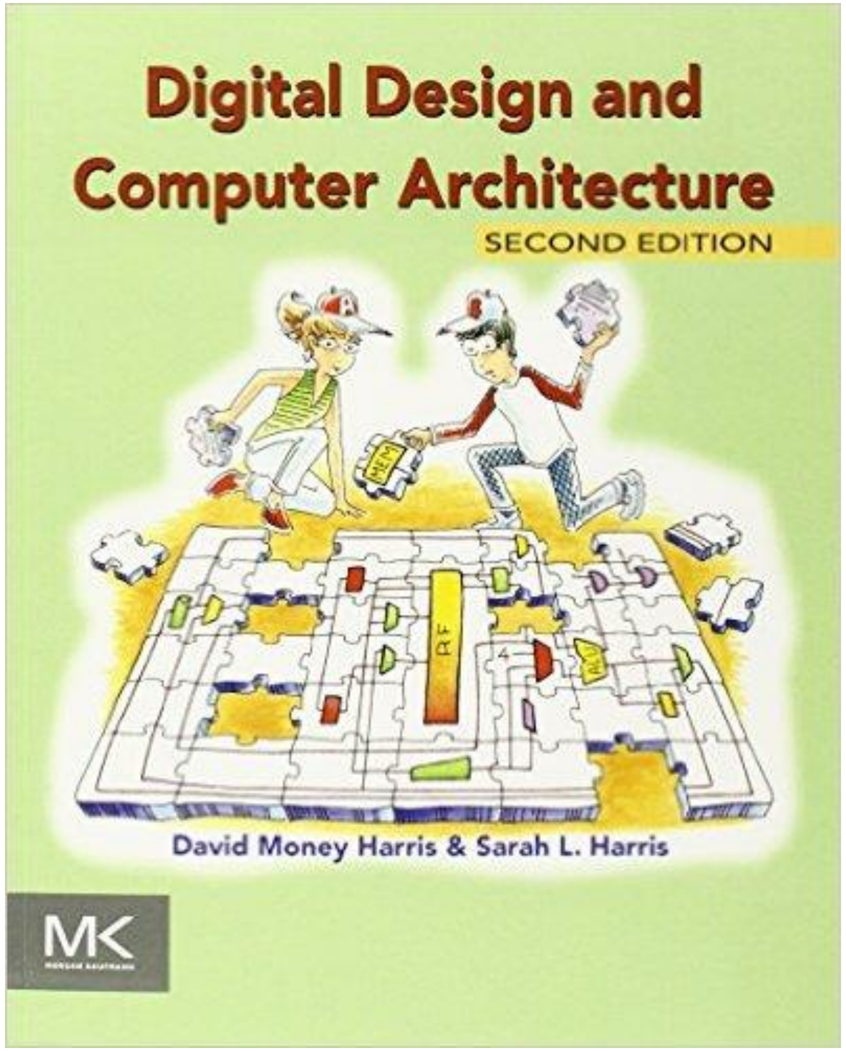
# Student Feedback

- Give us frequent (e.g., every week) feedbacks on how things are going. It is very useful for improving your learning experience in a timely manner, especially in a current situation
- Anonymous feedback form:
  - link on the course website
- Or even better, just send me an email

# Textbook: DDCA

Digital Design and Computer Architecture, 2nd edition, 2012by David Harris, Sarah Harris (1st edition also fine)

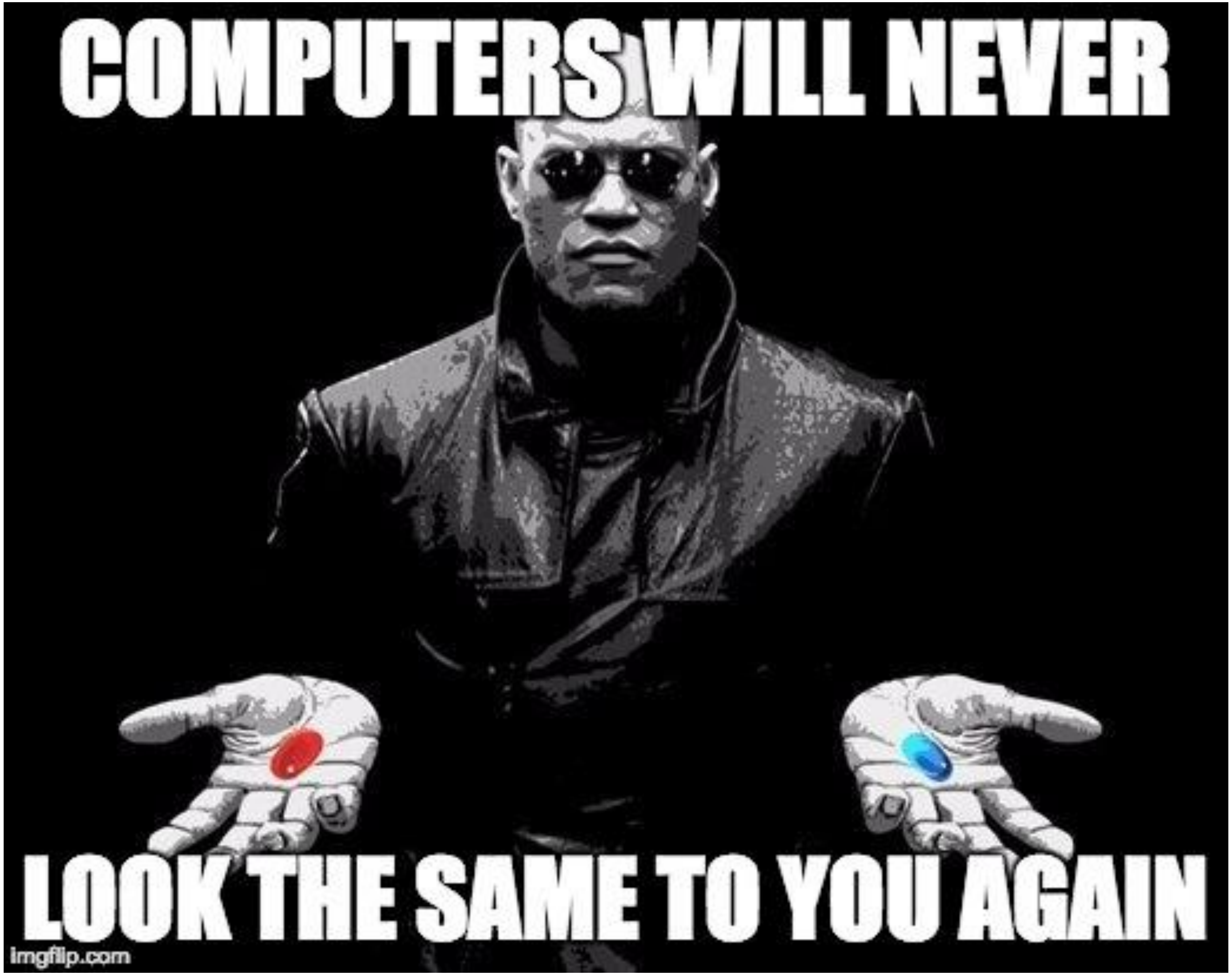
Available online at UofT library (Link)



# A typical week of CSCB58

- Wednesday/Friday: go the lectures
- Before your lab: submit prelab report
- Monday/Tuesday/Thursday: labs
- Thursday/Friday morning: next week's lab handout released, start working on the prelab.
- Saturday/Sunday: keep studying ;)

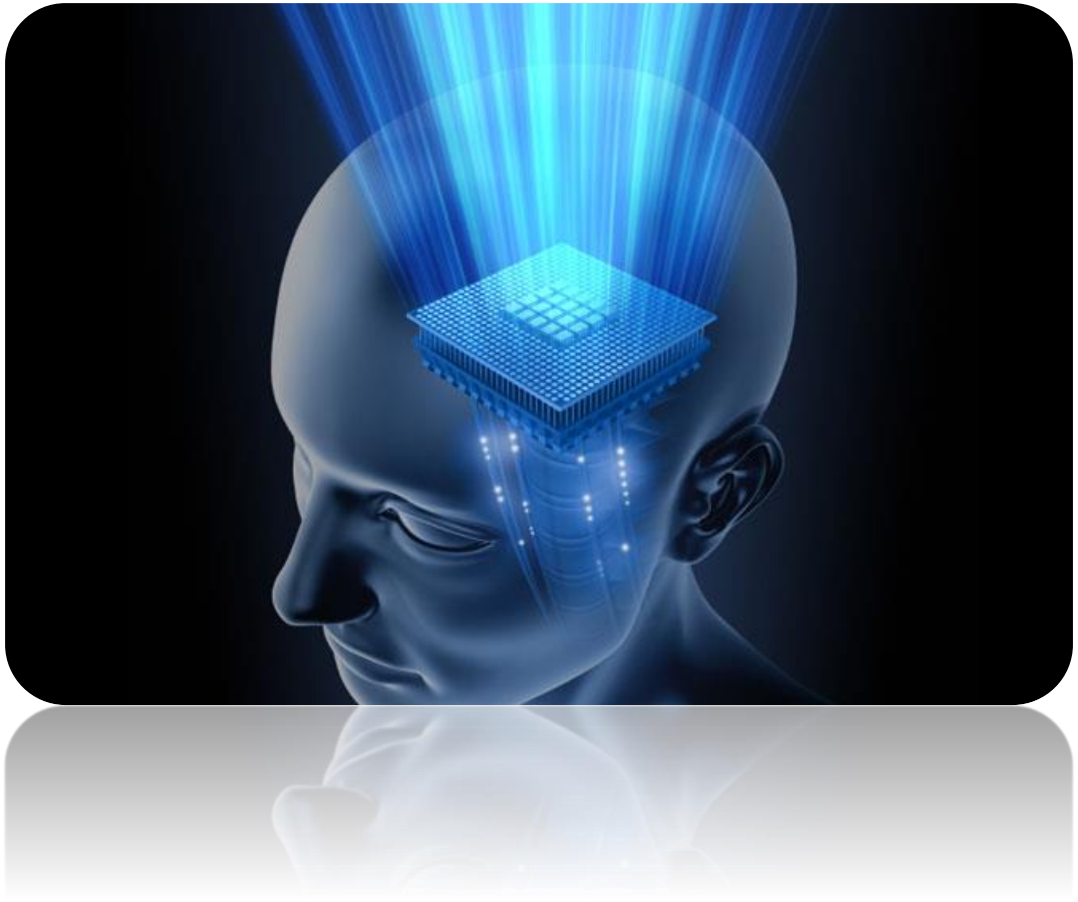
**It will be a lot of work,**
**and a lot of fun!**


COMPUTERS WILL NEVER
LOOK THE SAME TO YOU AGAIN
imgflip.com

# Let the learning begin

# Basic Logic Gates

# You already know something…

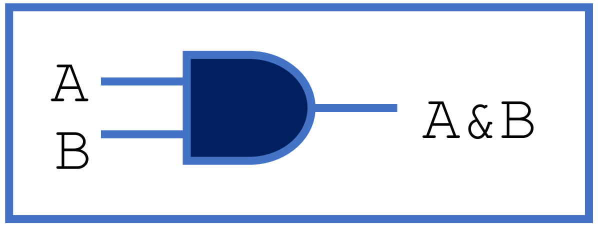


# Logic from math course

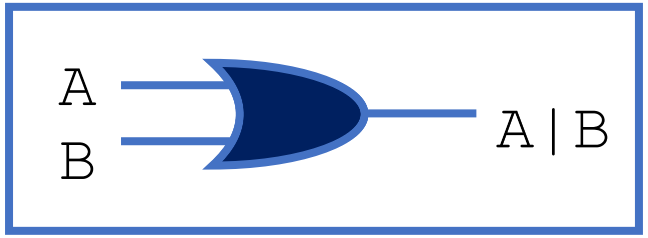
- Create an expression that is true iff the variables `A` and `B` are true, or `C` and `D` are true.

```
G = (A & B) | (C & D)
```

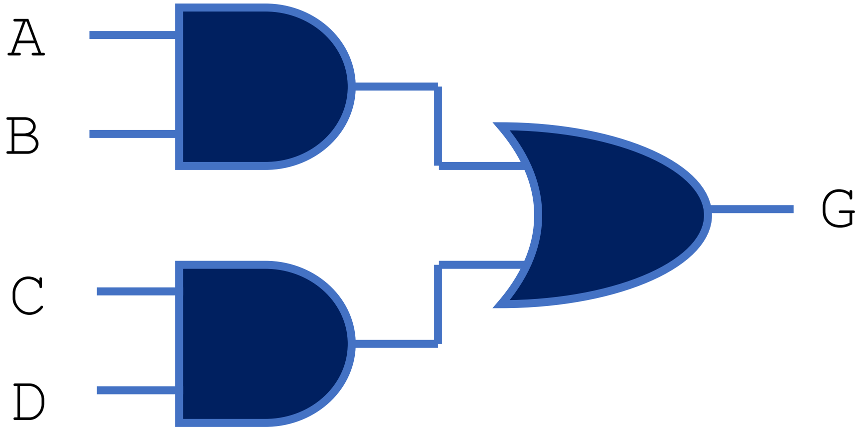
G = (A & B) | (C & D)




AND Gate




OR Gate




You just designed your first circuit in CSCB58!

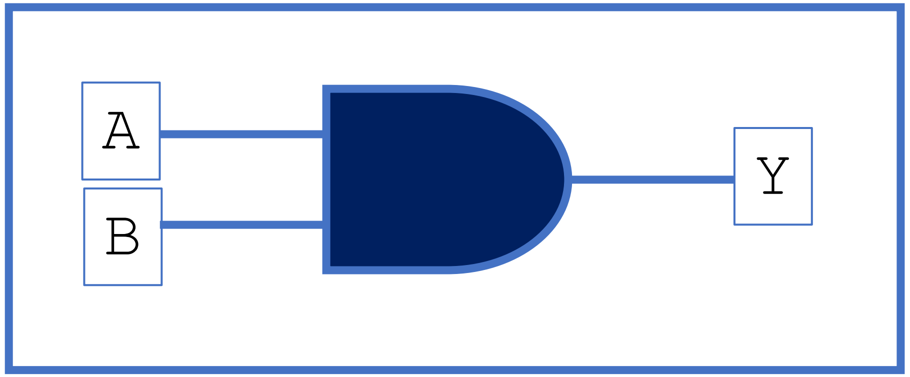
# Gates = Boolean logic

- If we know the logical expression, we already know how to put logic gates together to form a circuit

- Just need to know which logic operations are represented by which gate!

**Let’s meet all the gates**

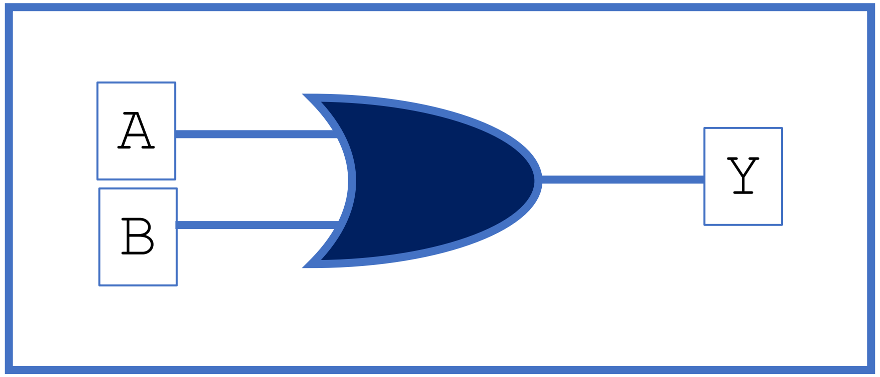
# AND Gates




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

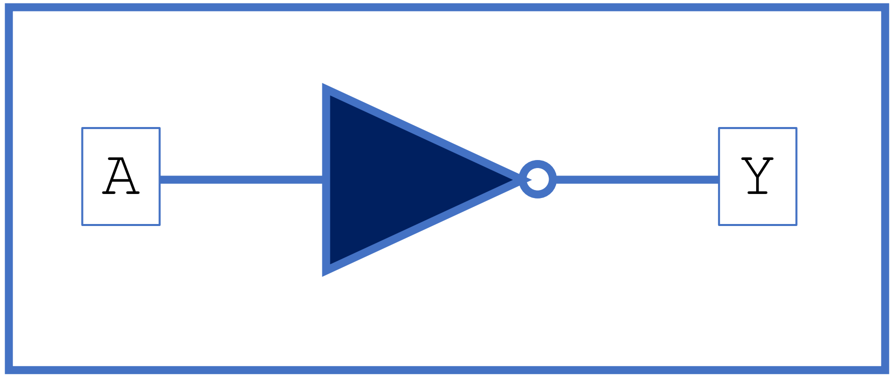
Truth table

# OR Gates



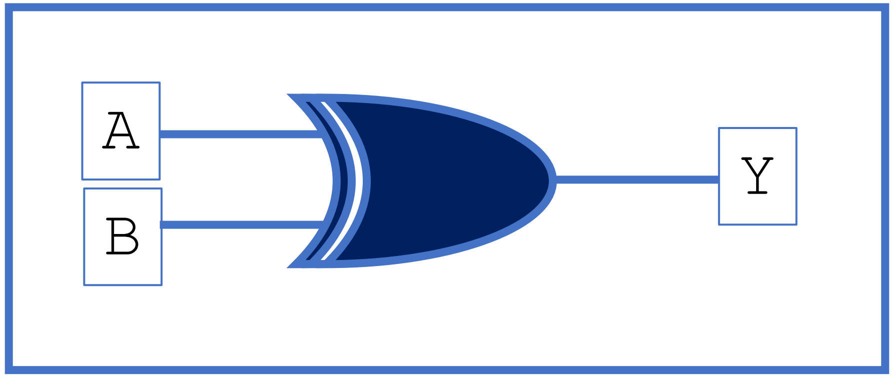

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# NOT Gates




| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

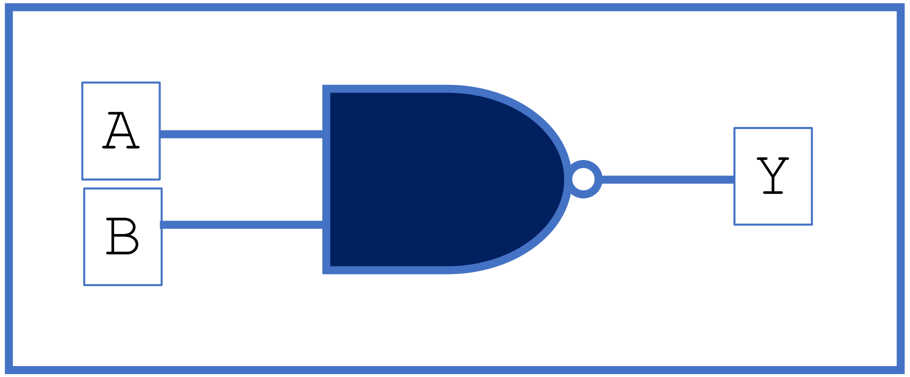
# XOR Gates




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

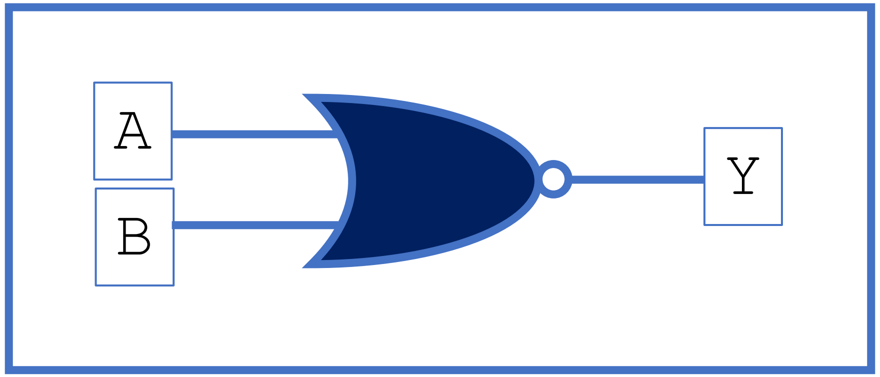
# Bill Gates

# NAND Gates



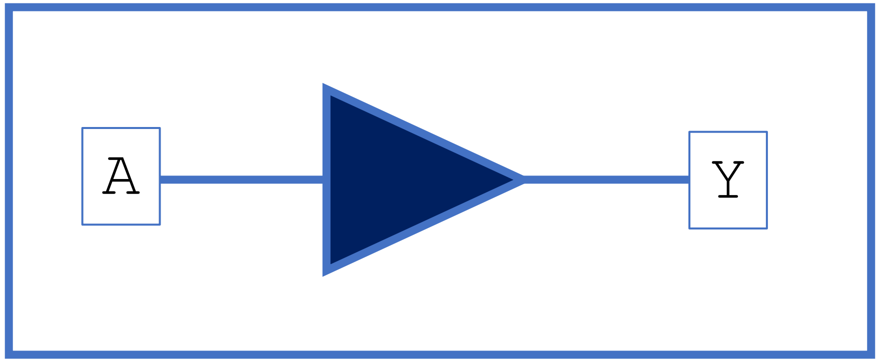

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# NOR Gates



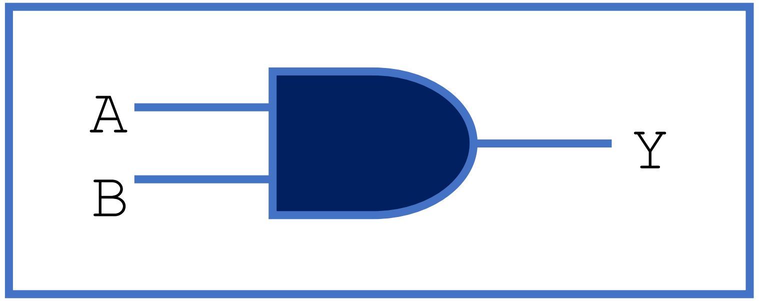

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

# Buffer



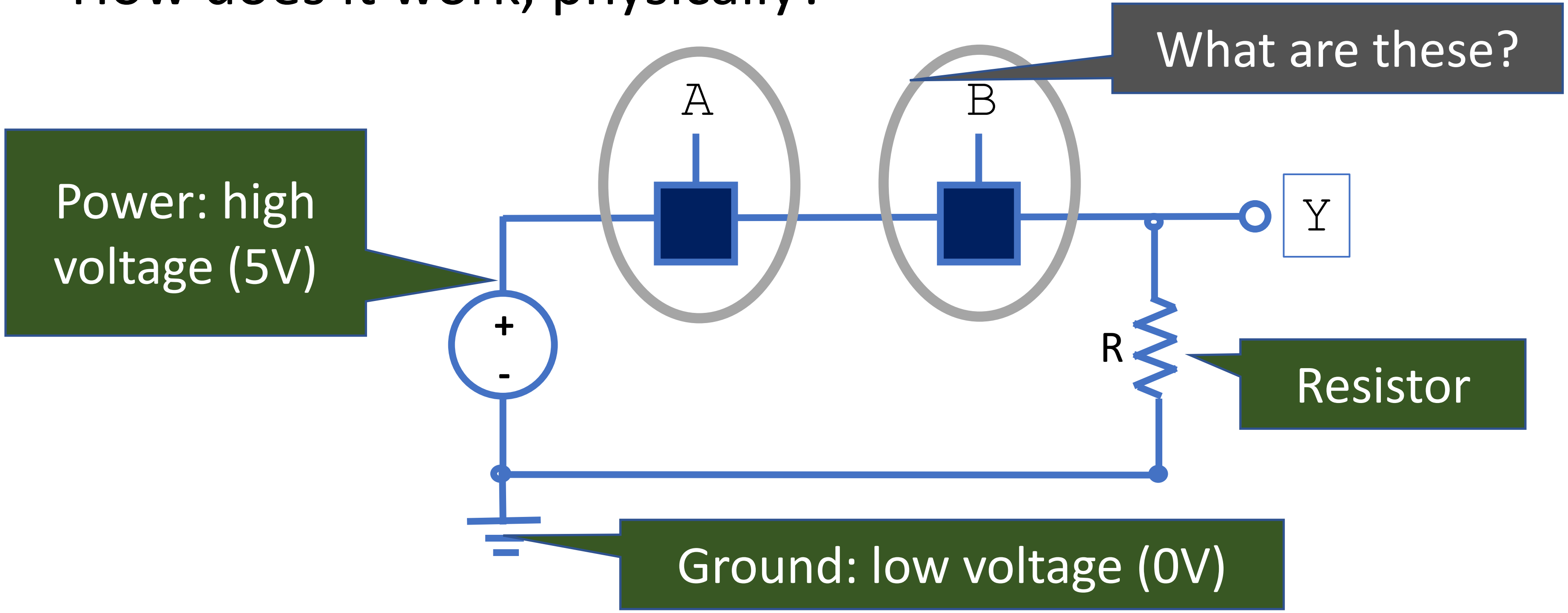

**This is not as silly as you might think now, as we'll see later…**

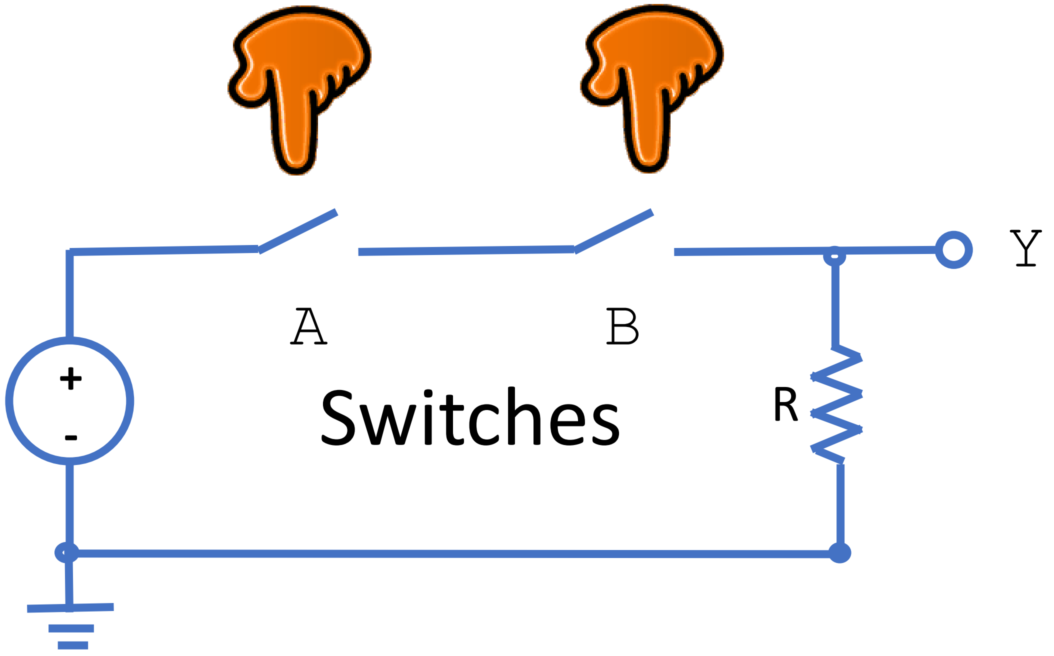
| A | Y |
|---|---|
| 0 | 0 |
| 1 | 1 |




# AND Gate

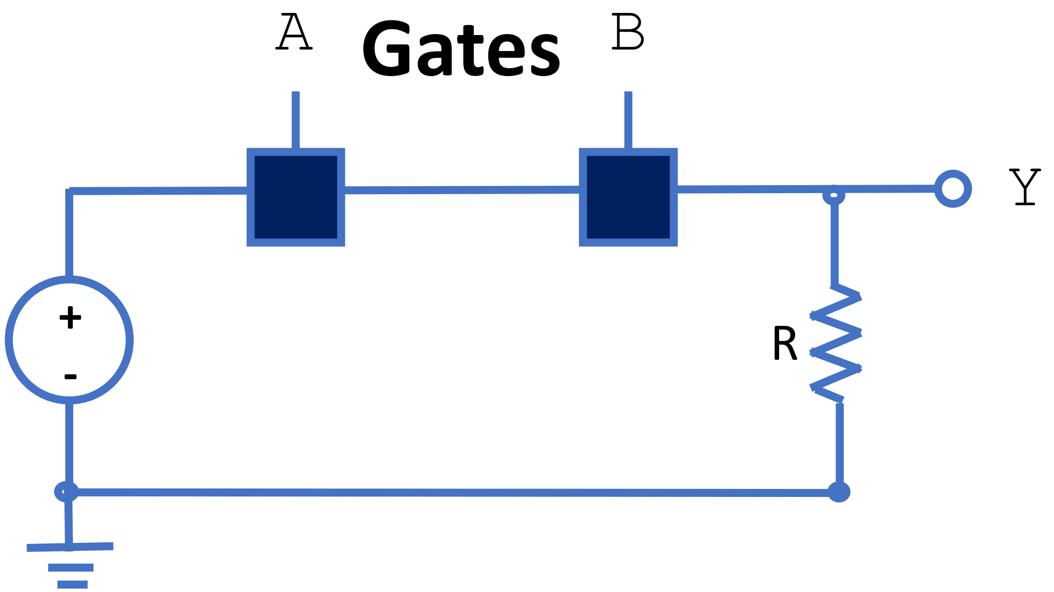
This is just a symbol...
What does it really look like, inside?
How does it work, physically?






When and only when **both** A are B are switched **ON**, Y has **high** voltage.

# Gates



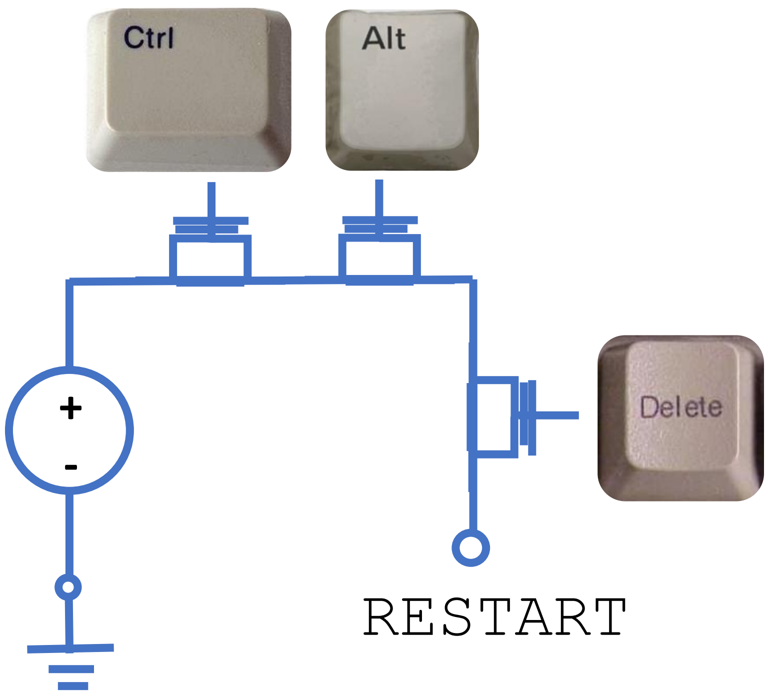


- Gate is like a switch, but controlled by the voltage of the input signal, instead of by a finger.
- Gate A is switched **ON** when signal A is of **high** voltage.
- When and only when **both** A and B have **high** voltage, Y has **high** voltage.
- High voltage is **1 (True),** low voltage is **0 (False).**
- **Y is True iff both A and B are True (Y = A & B).**

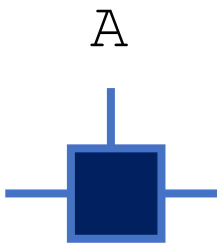
# Thinking in hardware

- Although CSCB58 has elements that are similar to other courses, it is very different in significant ways
- Unlike other software courses, CSCB58 is not about creating programs and algorithms, but rather devices and machines
  - Very important concept to grasp early in this course!
  - For instance: We need to understand what certain terms mean in the context of hardware.

# Example: Ctrl-Alt-Delete




| Ctrl | Alt | Del | RESTART |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |




**Gate is switched ON when signal A is of high voltage …**

**Why?**

**How?**

What does the inside of a gate look like?

Answer: There are **transistors**

# Transistors

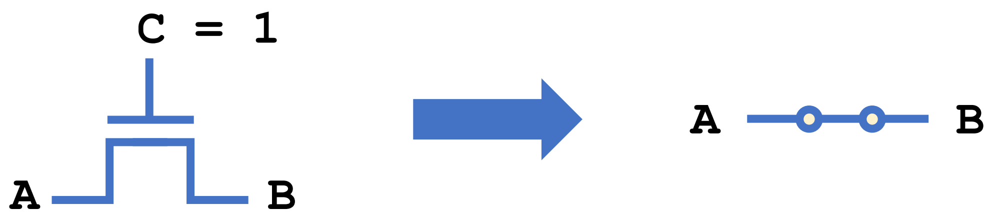
# One of the greatest inventions of the $20^{th}$ century

- Invented by William Shockley, John Bardeen and Walter Brattain in 1947, replacing previous vacuum-tube technology.
  - Nobel Prize for Physics in 1956.

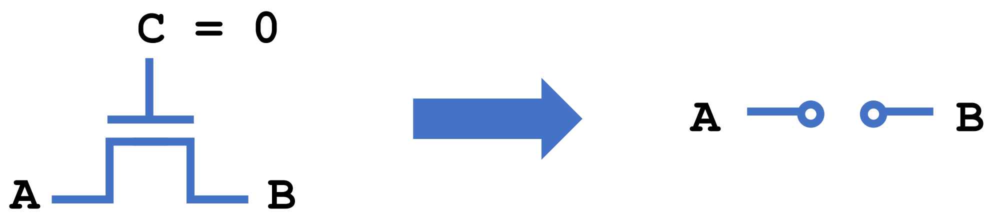
**Building block for the hardware of all your computers and electronic devices**

# What do transistors do?

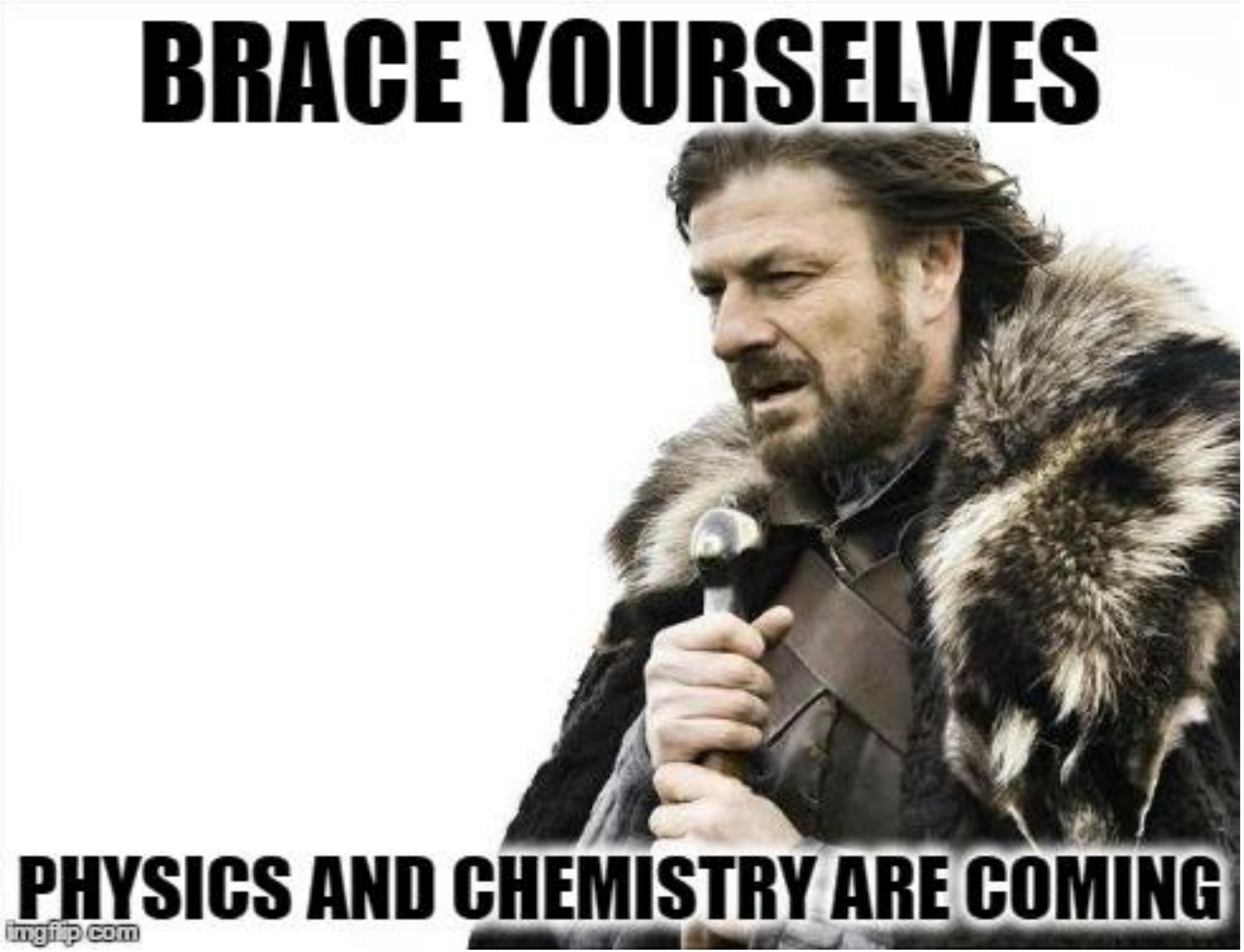
- Transistors connect Point A to Point B, based on the value at Point C.
  - If the value at Point C is high, A and B are connected.




  - And if the value at Point C is low, A and B are not.



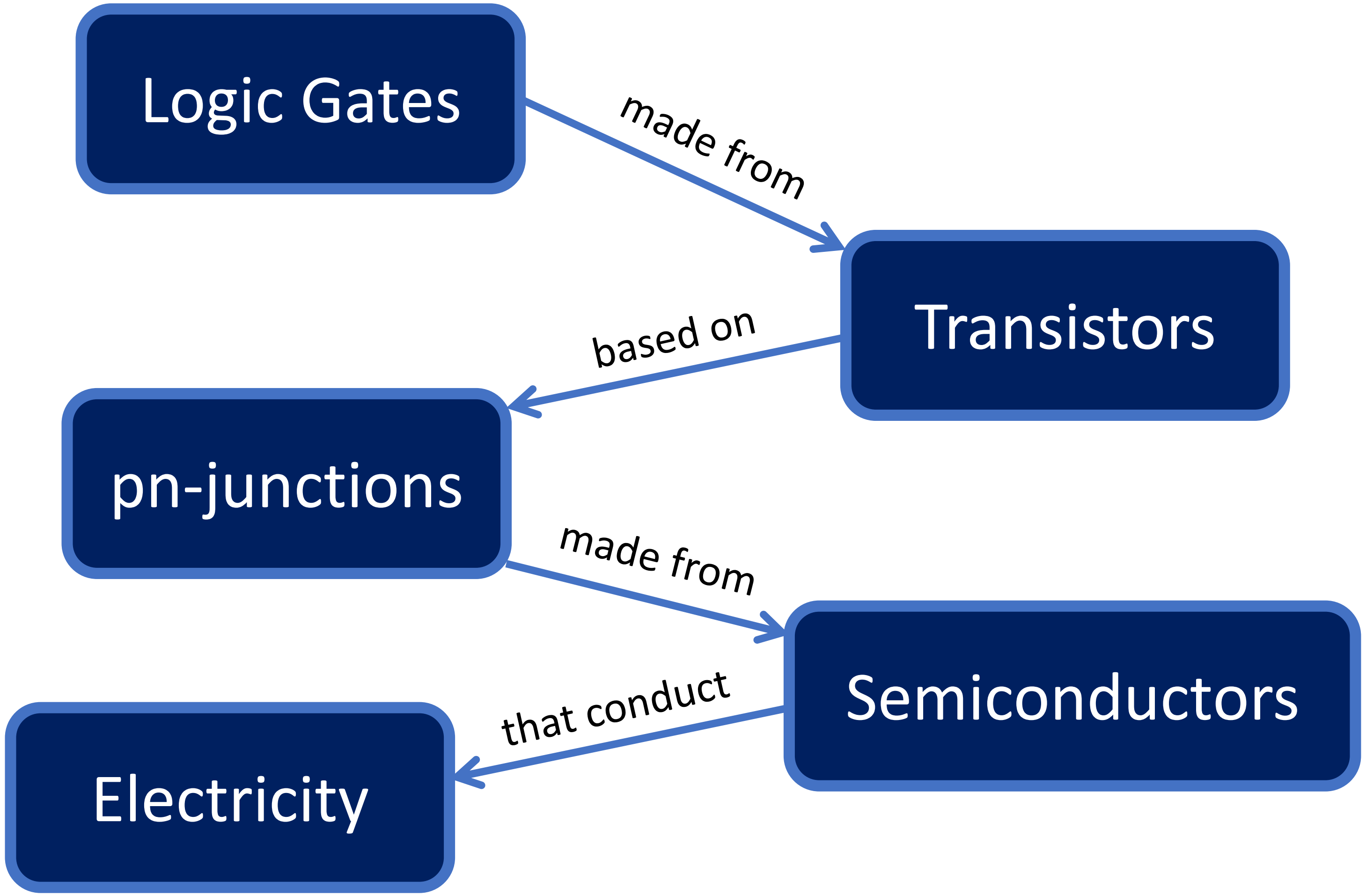

- **Need to know a little about electricity now….**


BRACE YOURSELVES
PHYSICS AND CHEMISTRY ARE COMING
imgflip.com

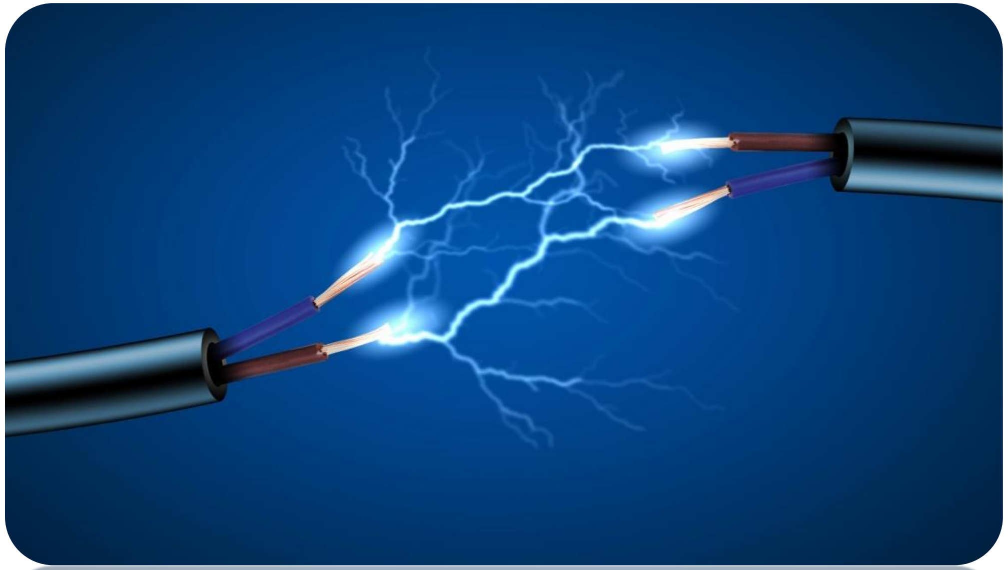
# Outline of the story

- Electricity, basic concepts
- Insulators, conductors, in between …, **Semiconductors**
- Impure semiconductors, **p-type / n-type**
- Put p-type and n-type together -- **pn-junction**
- Apply voltage to a pn-junction – **principle of transistors**
- A real-world manufacturing of transistor -- **MOSFET**

# Where do transistors fit?

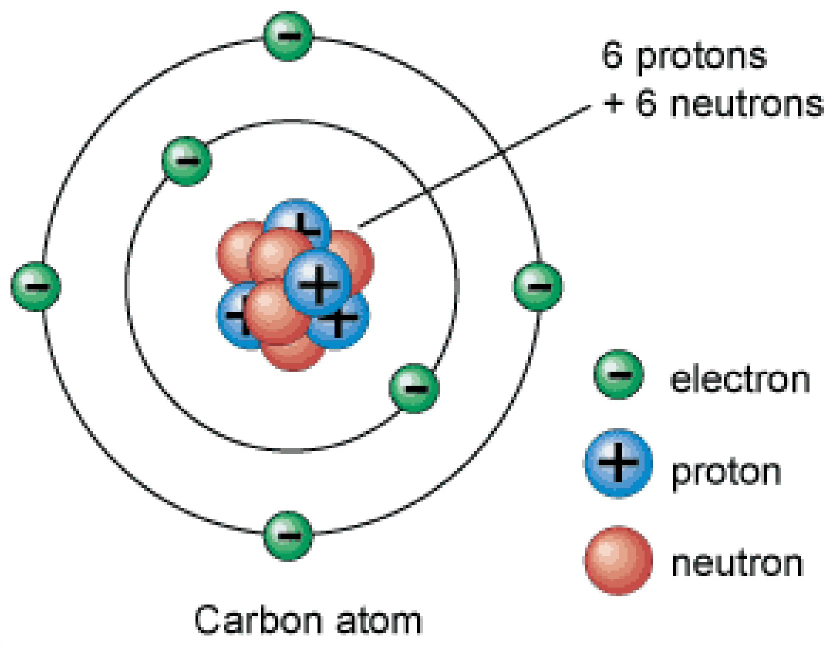


# Electricity Basics



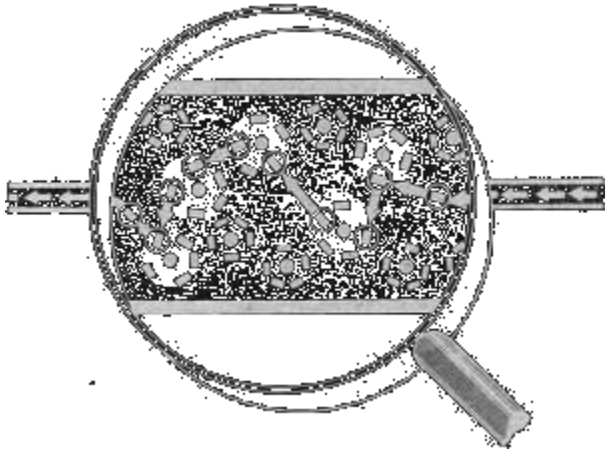
# Everything is made out of atoms …

- **Protons** are big (hardly move) and positively charged.
- **Electrons** are small (easily move) and negatively charged.
- **Neutrons** are big and, of course, neutral.
- Overall, an atom is **neutral**.



# What is Electricity?

- Electricity is the **flow** of charged particles (usually electrons) through a material.



# How do electrons flow?

They flow …

# How do electrons flow?

- Electrons want to flow from regions of **high electrical potential** (many electrons) to regions of **low electrical potential** (fewer electrons).
  - Like water flows from high to low.

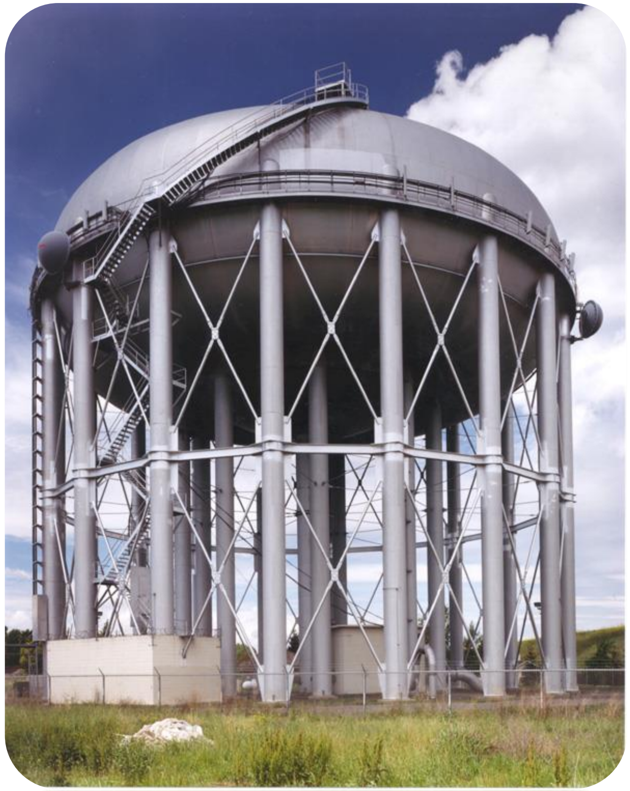
- This potential is referred to as **voltage** (V).
- The rate of this flow is called the **current** (I).
- Resistance (**I = V / R**) is like how narrow the water pipe is.

# Direction

The direction of the current is **opposite** to the direction of the electron movement, because electrons are **negatively** charged.

# Water Analogy

- To help picture this concept of voltage and current, imagine a reservoir:
  - Electrons flow from high to low potential like water would flow from the reservoir to the ground.
  - Voltage is like the elevation of the water above the ground.
  - Current is the rate at which the water flows.
- The relationship between voltage (V) and current (I) is called resistance: R = V/I



# A Note about Current

- Even though current is caused by electrons flowing through a material, the convention is to measure current as the **movement of positive charges**.
  - *Protons don't actually move*. When electrons move from point A to point B, the result is that B becomes more negative and A becomes more positive.
  - Scientists historically viewed current in terms of this creation of positive charge in a material.
  - It's not completely clear why scientists decided this.  Just go with it ☺

# More on Resistance

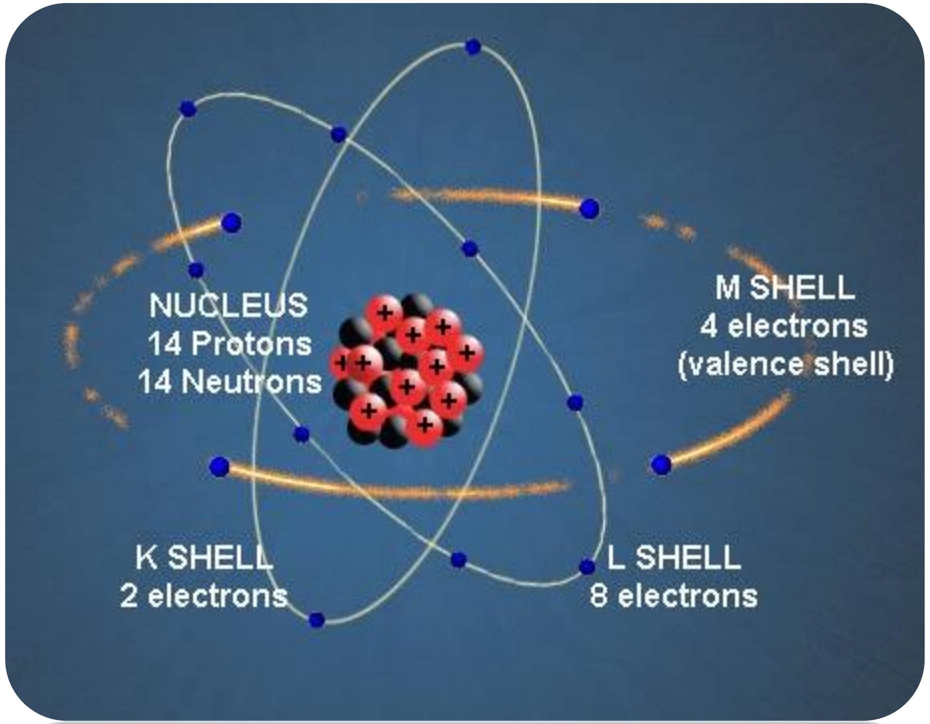
- Electrical resistance indicates how well a material allows electricity to flow through it:
  - High resistance (aka **insulators**) don’t conduct electricity at all.
  - Low resistance (aka **conductors**) conduct electricity well and are generally used for wires.

- **Semiconductors** are somewhere in between conductors and insulators, which makes it interesting…

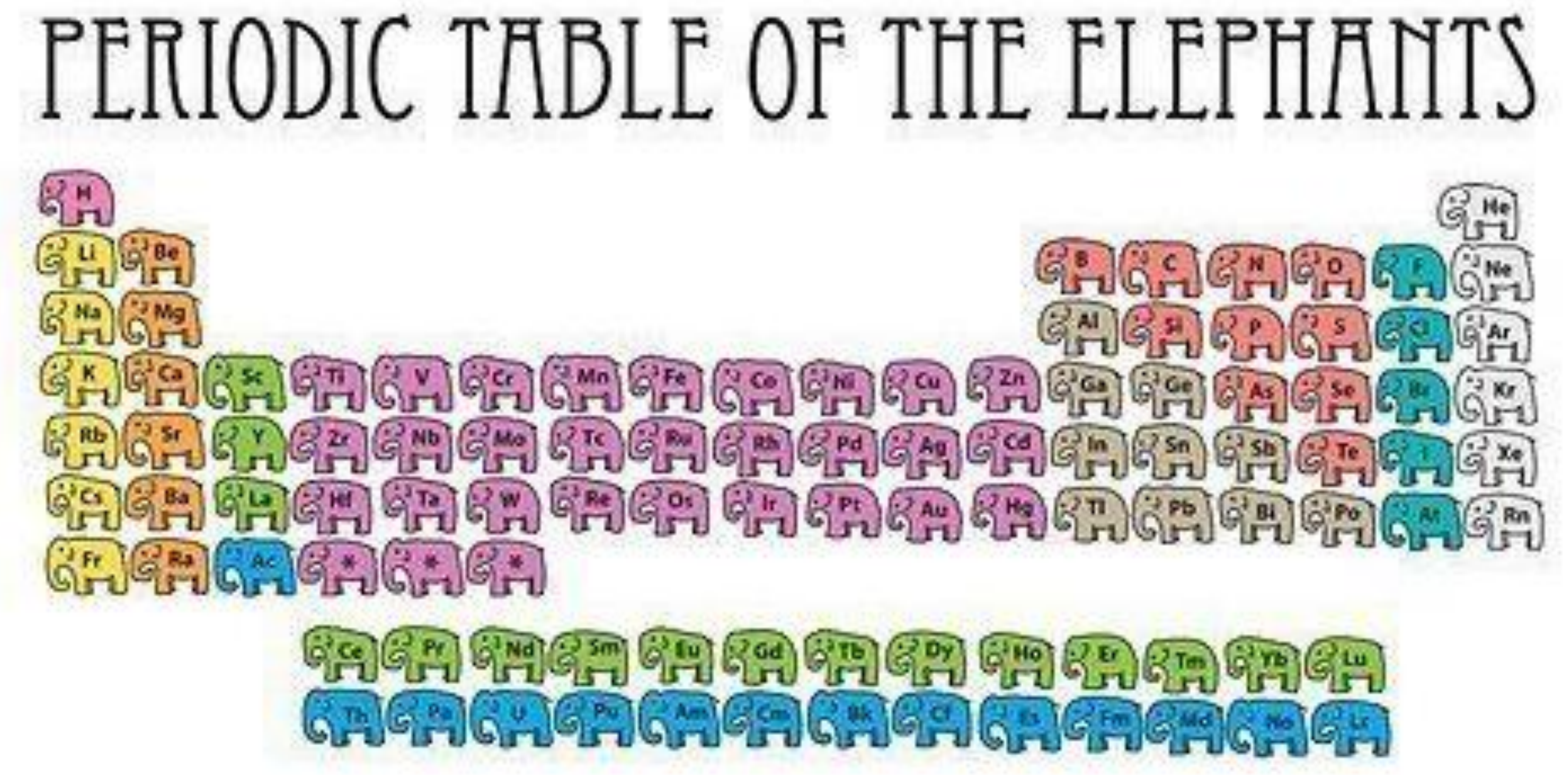
# Outline of the story

- Electricity, basic concepts
- Insulators, conductors, in between …, **Semiconductors**
- Impure semiconductors, **p-type / n-type**
- Put p-type and n-type together -- **pn-junction**
- Apply voltage to a pn-junction – **principle of transistors**
- A real-world manufacturing of transistor -- **MOSFET**

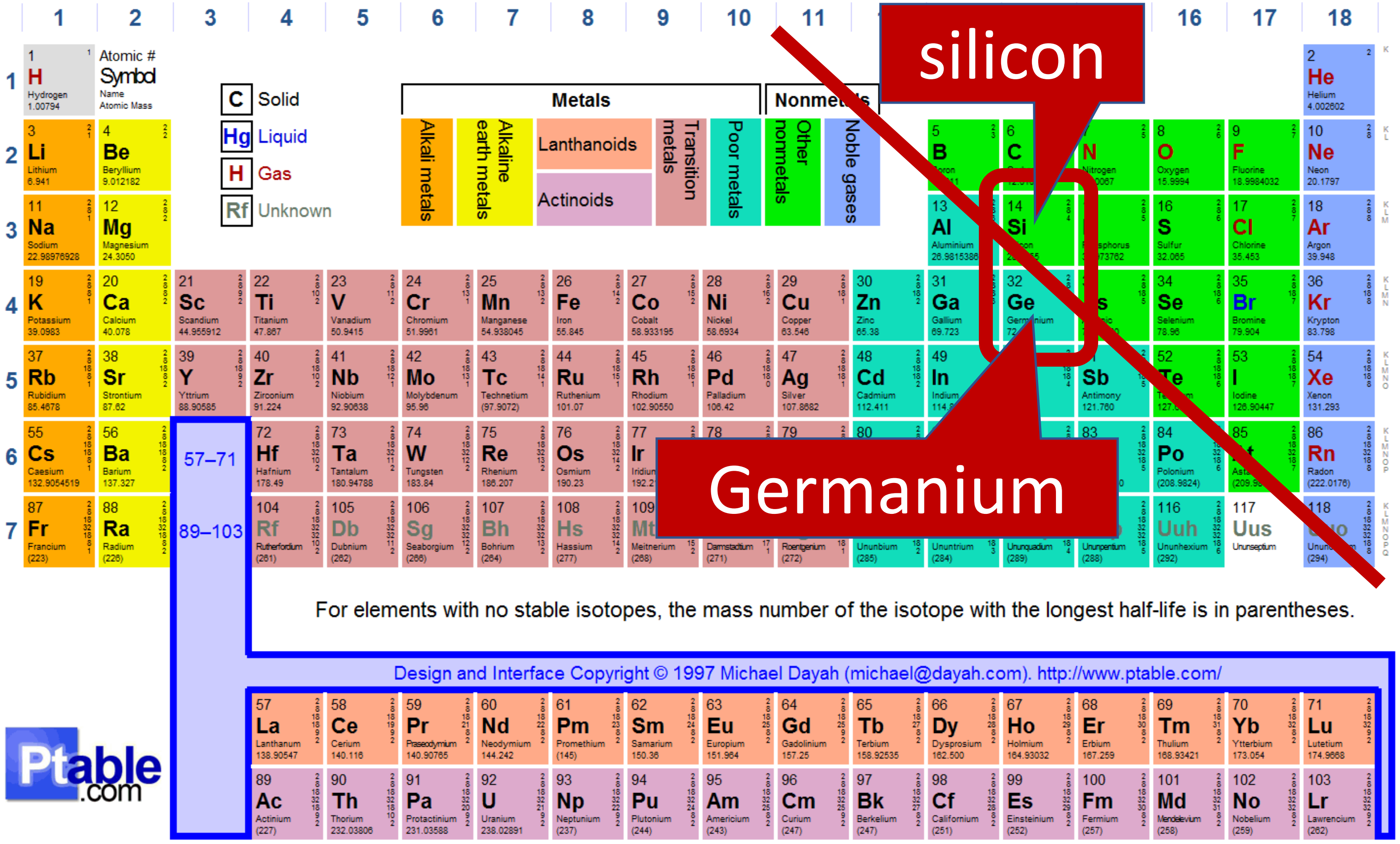
# Semiconductors



# Here comes the chemistry



# Periodic Table of Elements



# Conductivity of Semiconductors

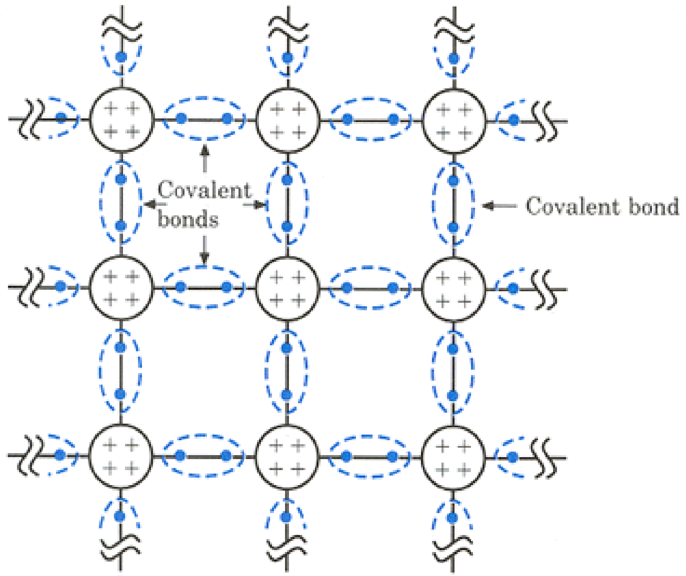
Semiconductor materials (e.g., silicon and germanium) straddle the boundary between **conductors** and insulators, behaving like one or the other, depending on factors like temperature and impurities in the material.

# Impurity

# Pure semiconductor is pretty stable

- Each atom has **4** valence electrons, forming bonds with other atoms, and the structure is **pretty stable**.

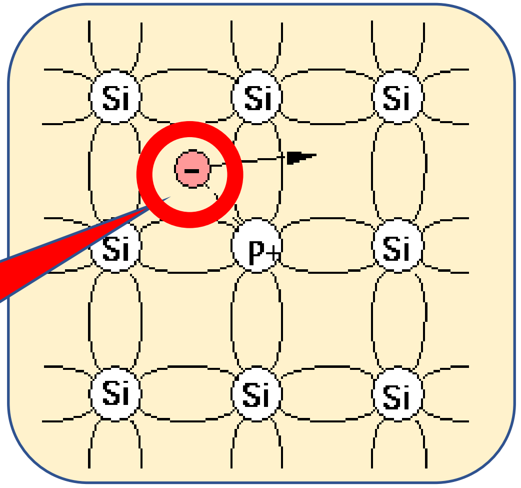
- At room temperature, very close to insulator.



# Encourage semiconductor's conductivity

**N-type:**

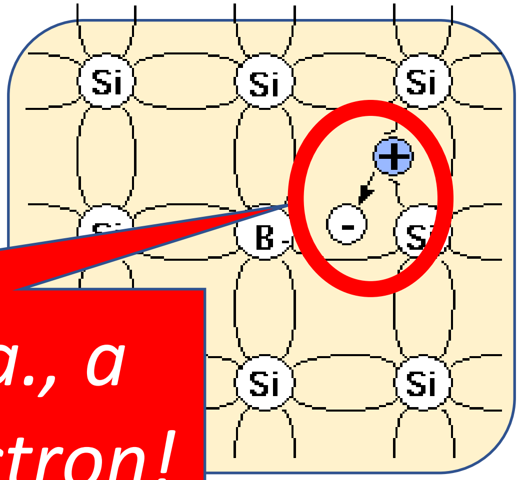
Add some atoms with **5** valence electrons, such as Phosphorus.




*An extra electron!*

**P-type:**

Add some atoms with **3** valence electrons, such as Boron.



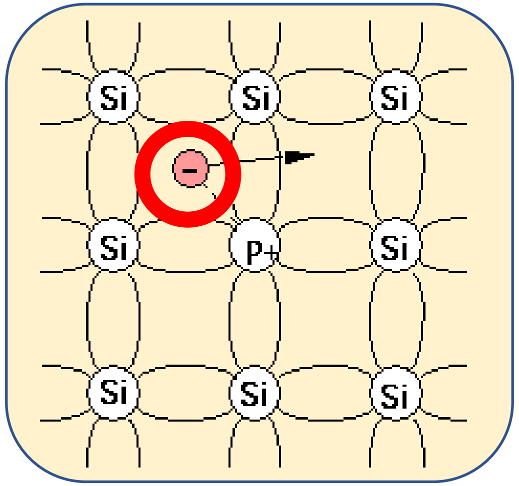

*A missing electron, a.k.a., a "hole", like a positive electron!*

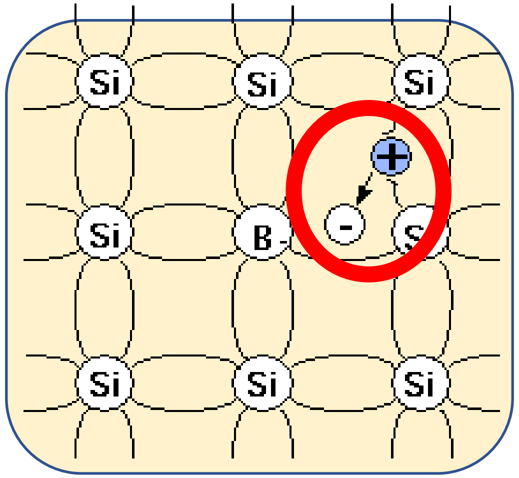
# Encourage semiconductor's conductivity

The extra electrons and the holes are **charge carriers**, which can move **freely** through the materials.

Thus the conductivity is encouraged.

This process of adding stuff is called **doping**, (n or p type).



Free electrons move like

Free holes move like

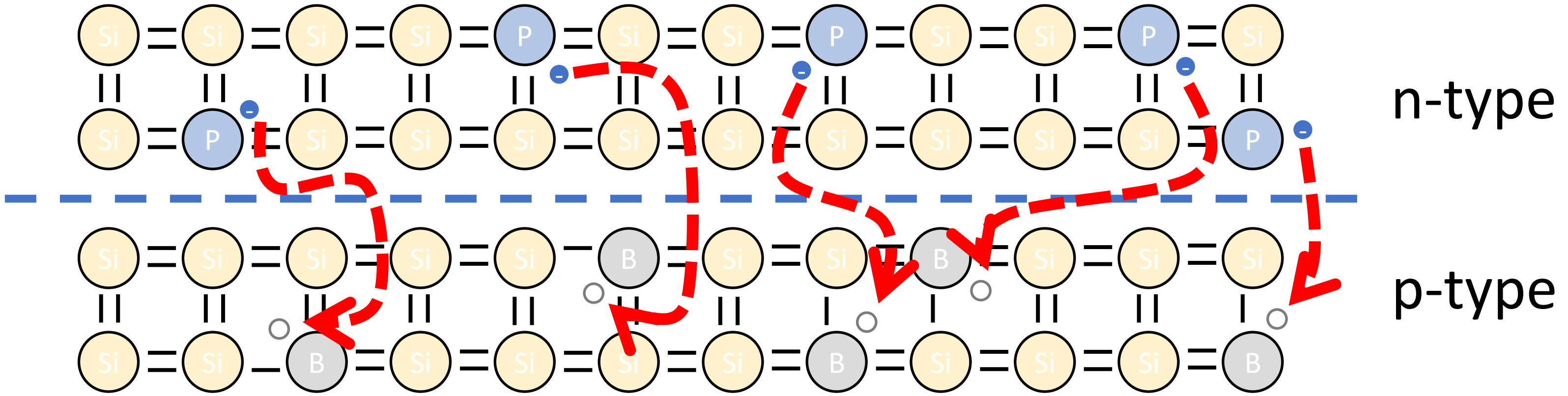
# Outline of the story

- Electricity, basic concepts
- Insulators, conductors, in between …, **Semiconductors**
- Impure semiconductors, **p-type / n-type**
- Put p-type and n-type together -- **pn-junction**
- Apply voltage to a pn-junction – **principle of transistors**
- A real-world manufacturing of transistor -- **MOSFET**

# PN-junctions

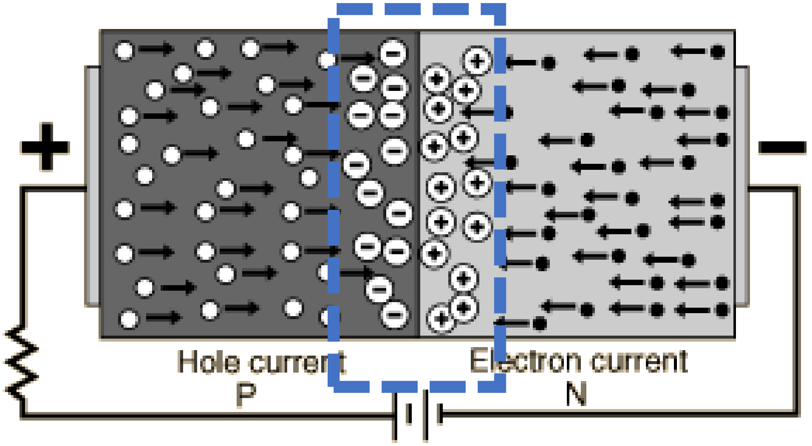
# Bringing p and n together

- What happens if you brought some p-type material into contact with some n-type material?



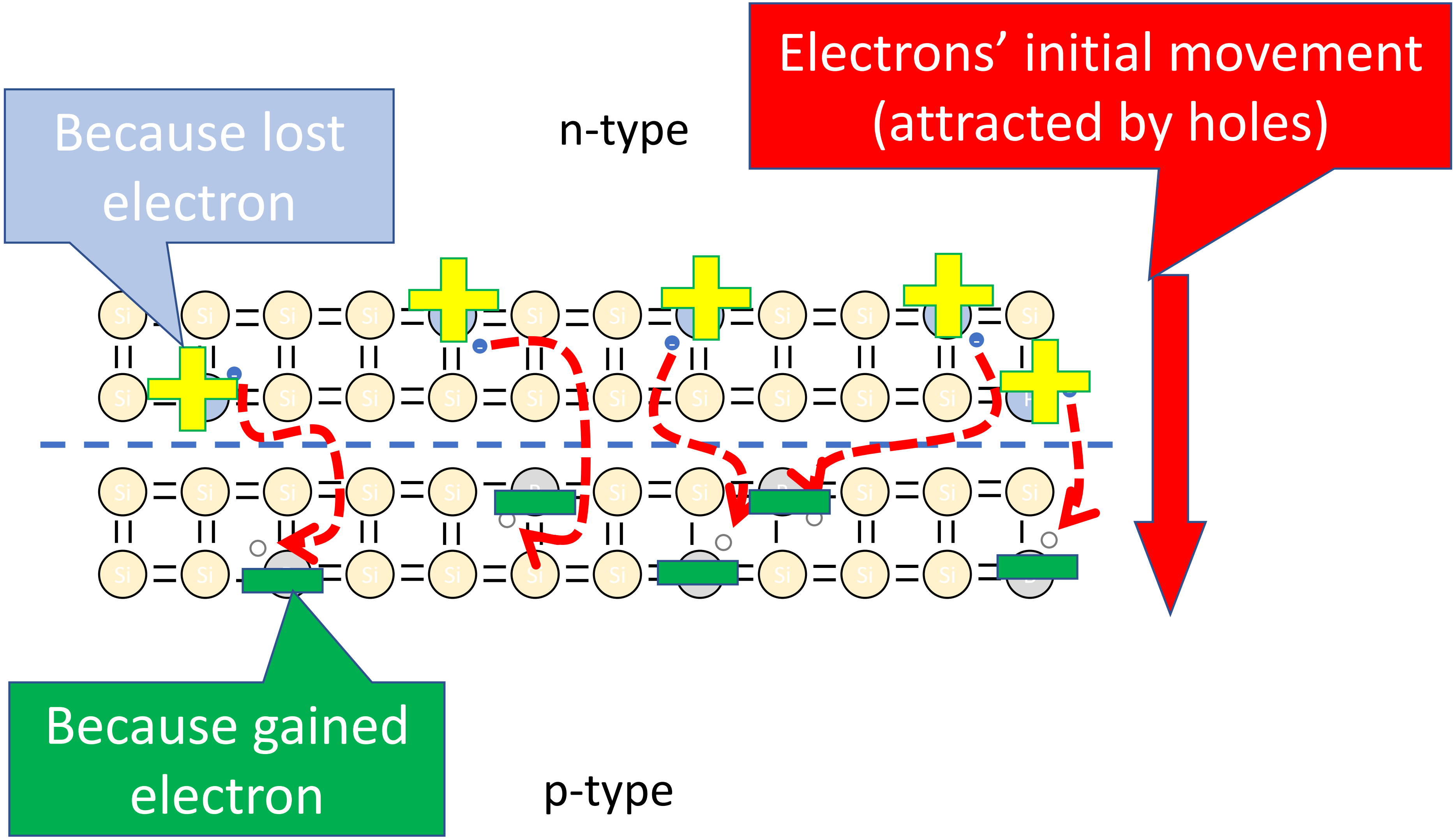

- The **electrons** at the surface of the n-type material are **drawn** to the **holes** in the p-type.

# p-n Junctions




- When left alone, the electrons from the n section of the junction will fill the holes of the p section, cancelling each other and create a section with no free carriers called the **depletion** layer.

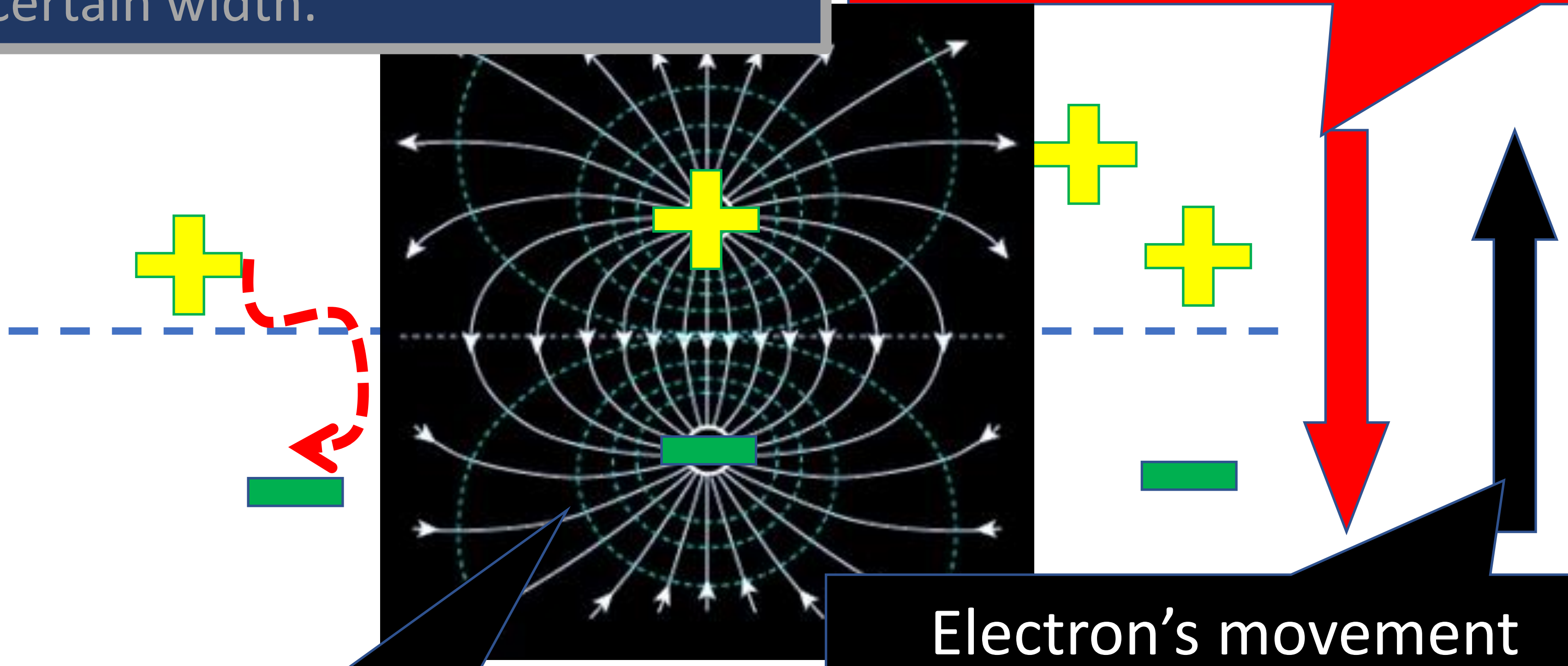
- Once this depletion layer is wide enough, the doping atoms that remain will create an **electric field** in that region.


Because lost electron
n-type
Electrons’ initial movement (attracted by holes)
Because gained electron
p-type

Diffusion increases the width of depletion layer, and drift draws it back. An **equilibrium** is reached, when the depletion layer is of a certain width.

**“Diffusion”**

Electrons’ initial movement (attracted by holes)



**Electric field**

p-type

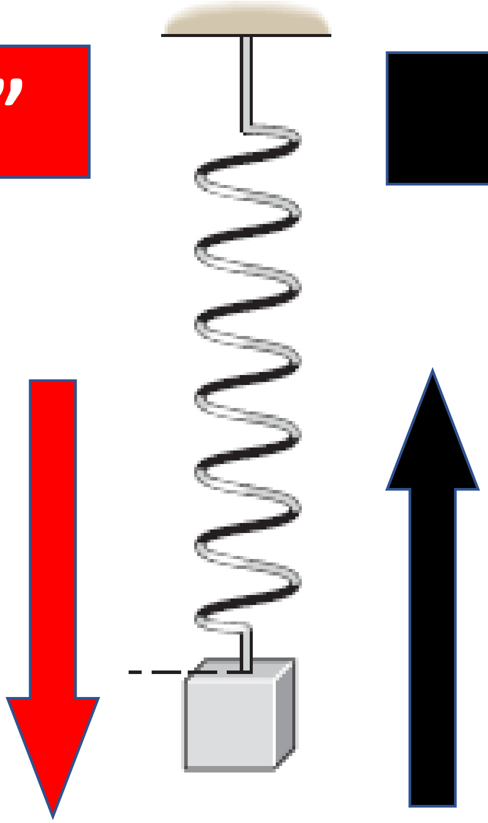
Electron’s movement drawn by the electric field

**“Drift”**

# Analogy: Spring with weight

“Diffusion”

Attracted by gravity (hole)

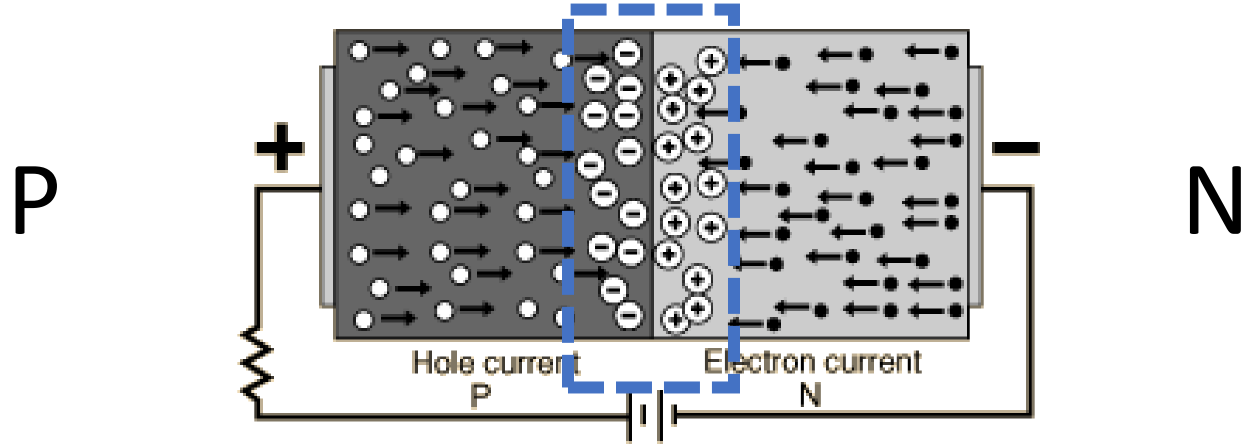


“Drift”

Drawn by spring force field

An equilibrium is reached when the spring is stretched by a certain length.

# Summary of pn-junction




When we put **p** and **n** together, they will form a depletion layer with electric field in it.

The depletion layer grows up to a certain **width**, until equilibrium is reached.

# Outline of the story

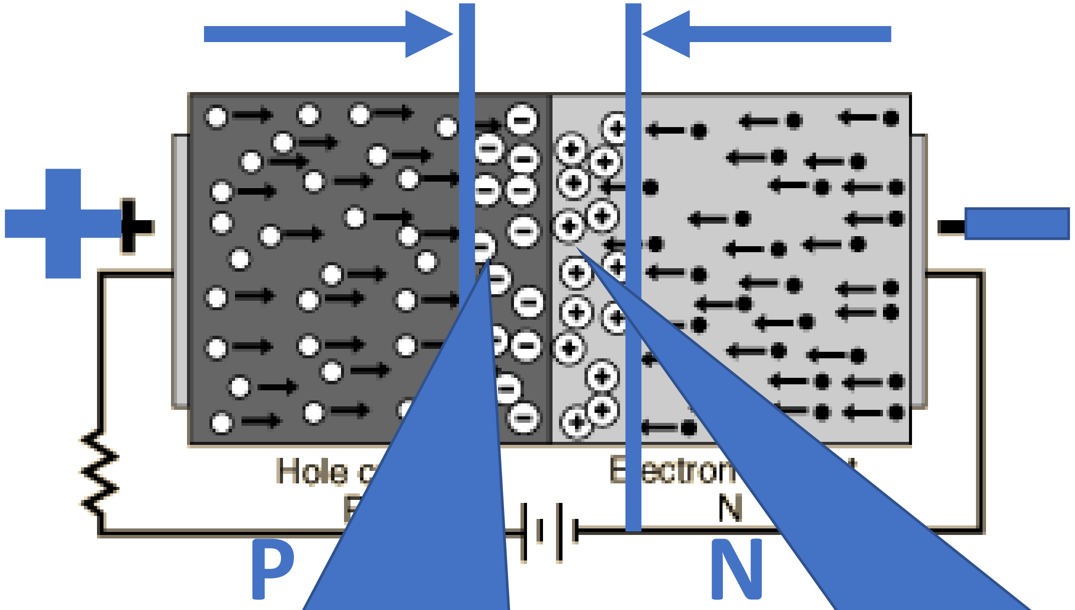
- Electricity, basic concepts
- Insulators, conductors, in between …, **Semiconductors**
- Impure semiconductors, **p-type / n-type**
- Put p-type and n-type together -- **pn-junction**
- Apply voltage to a pn-junction – **principle of transistors**
- A real-world manufacturing of transistor -- **MOSFET**

# Apply voltage to a PN-junction

It could be applied in two possible directions

- **P**ositive voltage to the **P** side
- **P**ositive voltage to the **N** side

# **Forward Bias** (Positive voltage to P)

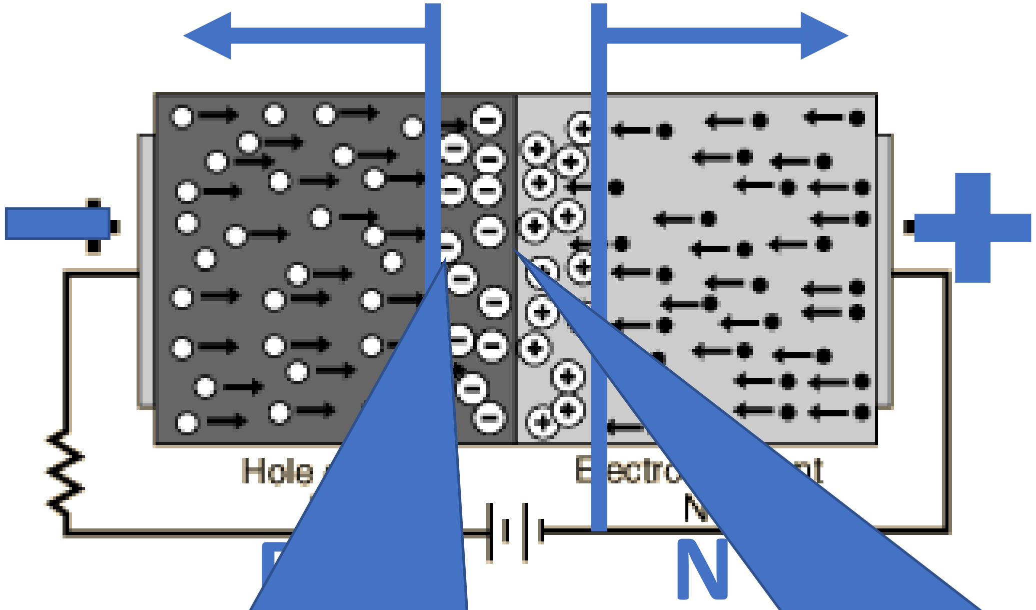



Negative charges sucked out of depletion layer

Positive charges sucked out of depletion layer

Depletion layer becomes **narrower**

# Reverse Bias (Positive voltage to N)




Negative charges **injected** into depletion layer

Positive charges **injected** into depletion layer

Depletion layer becomes **wider**

Apply forward bias

- Depletion layer narrower
- Easier to travel through
- Better conductivity
- Like switch **connected**

Apply reverse bias

- Depletion layer wider
- Harder to travel through
- Worse conductivity
- Like switch **disconnected**

That's how transistors work!

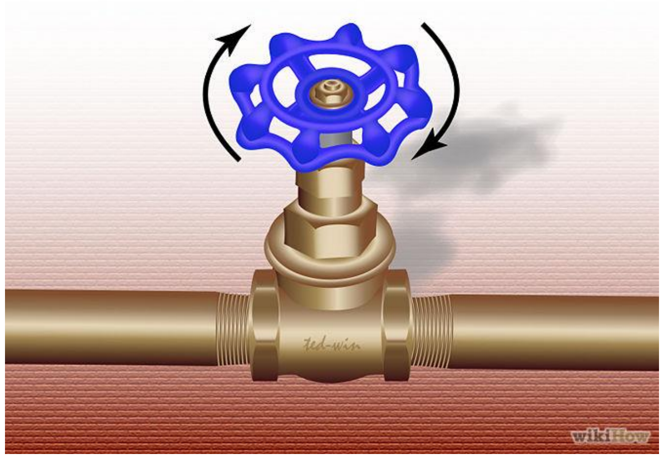
# Outline of the story

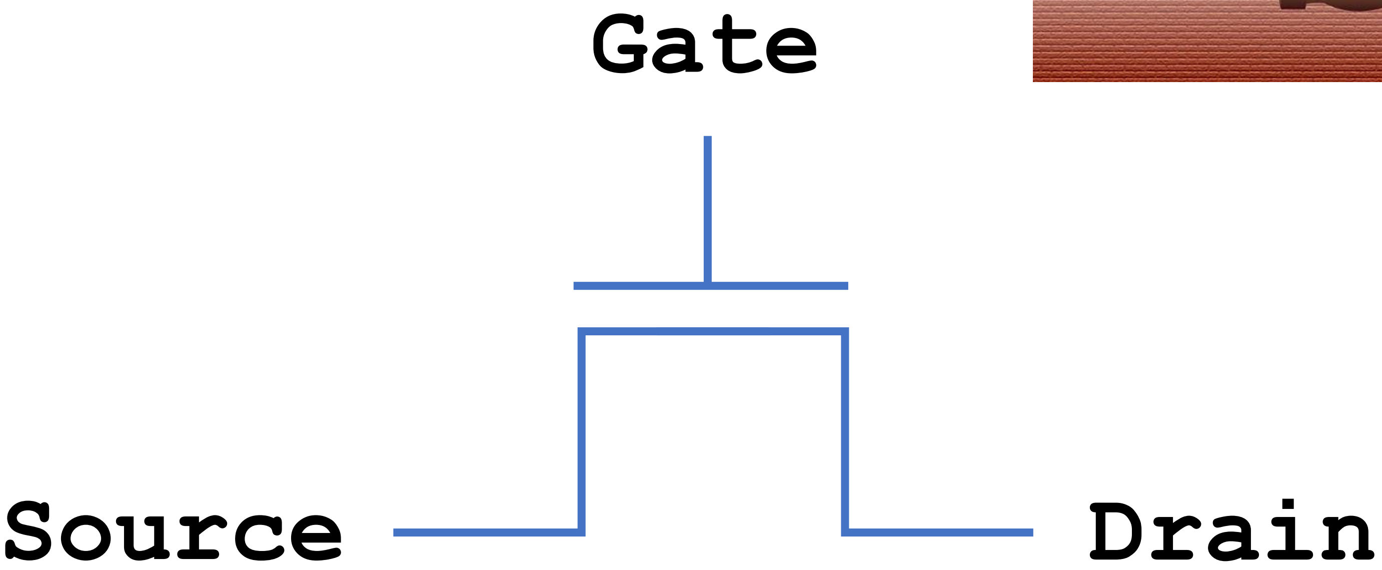
- Electricity, basic concepts
- Insulators, conductors, in between …, **Semiconductors**
- Impure semiconductors, **p-type / n-type**
- Put p-type and n-type together -- **pn-junction**
- Apply voltage to a pn-junction – **principle of transistors**
- A real-world manufacturing of transistor -- **MOSFET**

# Creating transistors

- Transistors use the characteristics of p-n junctions to create more interesting behaviour.
- Three main types:
  - Bipolar Junction Transistors (BJTs)
  - Metal Oxide Semiconductor Field Effect Transistor (MOSFET)
  - Junction Field Effect Transistor (JFET)
- The last two are part of the same family, but we'll only look at the MOSFET for now.

# **M**etal **O**xide **S**emiconductor **F**ield **E**ffect **T**ransistor

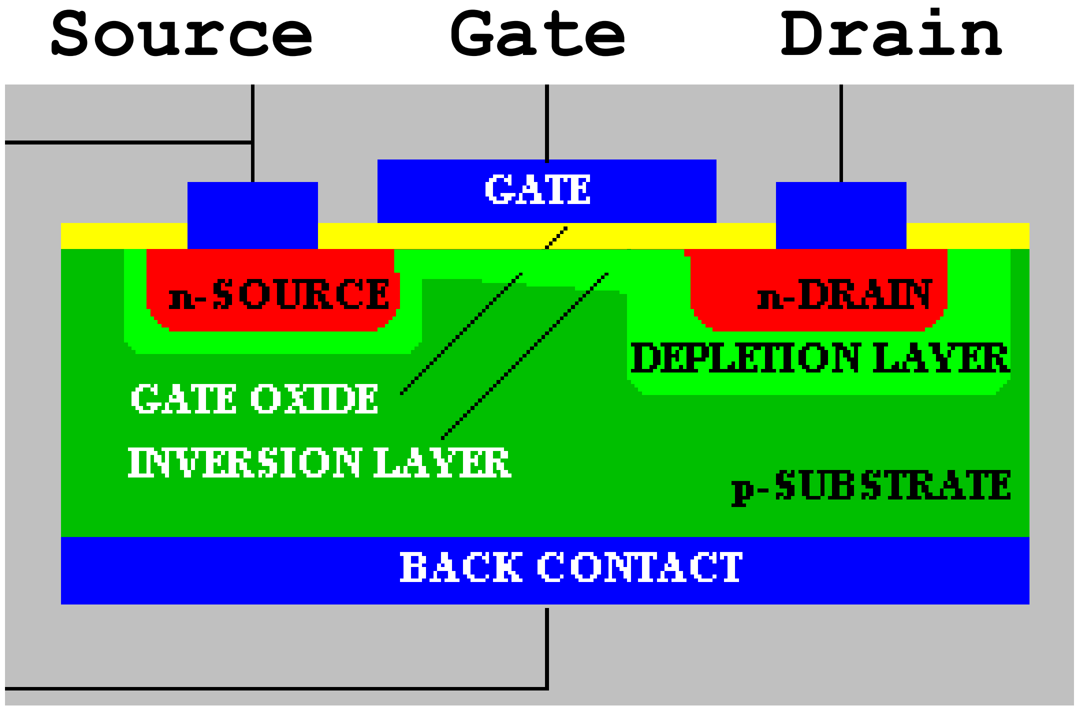



Gate
Source
Drain

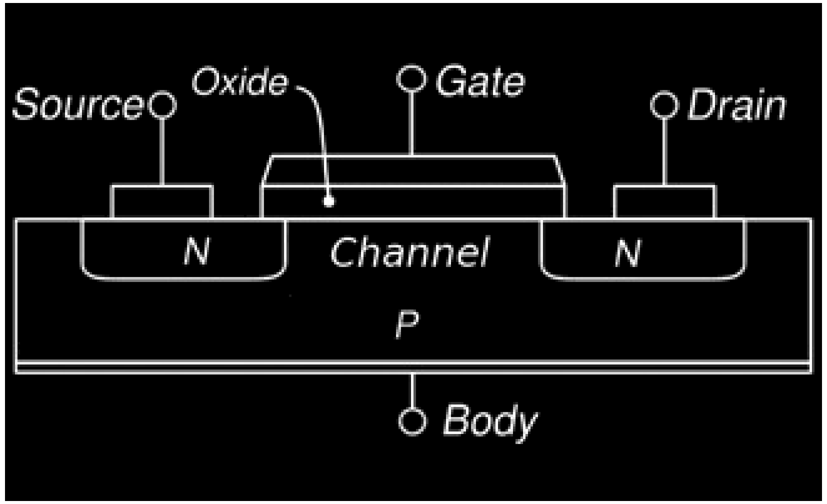
Metal — Conductor, can apply electric charge to it

Oxide — Insulator

N-type, P-type — Semiconductors, doped

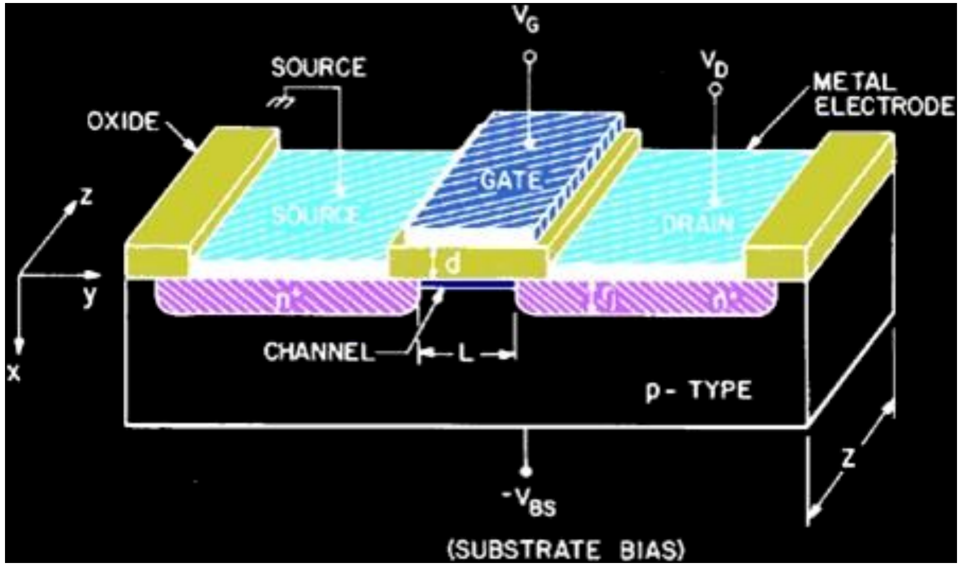


# The MO of MOSFETs



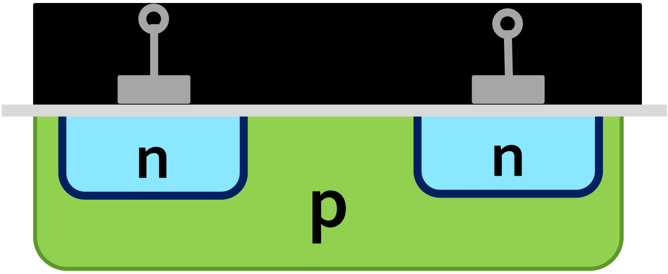

- Metal Oxide Semiconductor Field Effect Transistors are composed of a layer of semiconductor material, with two layers on top of the semiconductor:
  - An oxide layer that doesn’t conduct electricity,
  - A metal layer (called the gate), that can have an electric charge applied to it
  - These are the M and O components of MOSFETs.

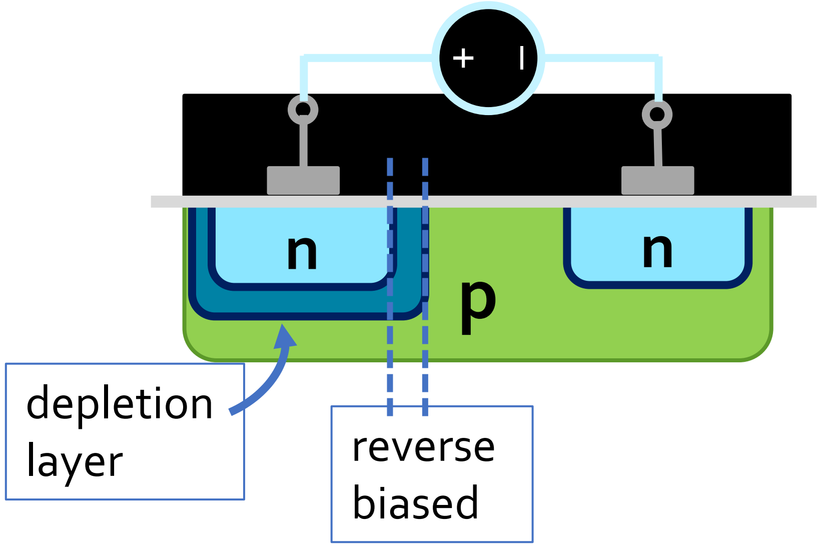
# The S of MOSFETs

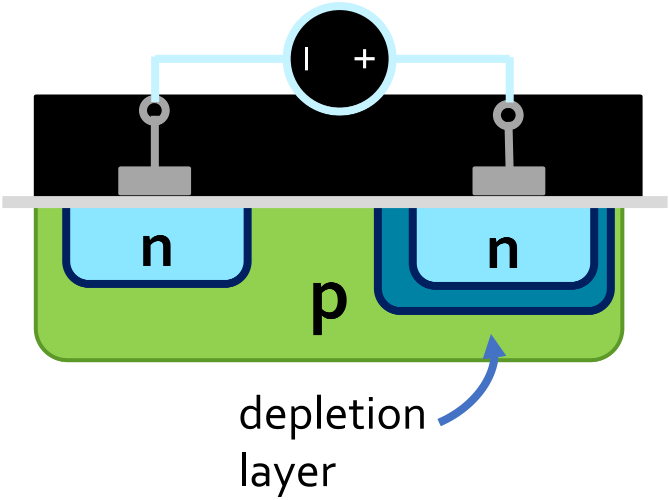



- The semiconductor sections are two pockets of n-type material, resting on a **substrate** layer type material.
- A voltage is applied across the two n-type sections, called the **drain** and the **source**. No current will pass between them though, because the p section in between creates at least one reverse-biased pn junction.

# Applying voltage to NPN

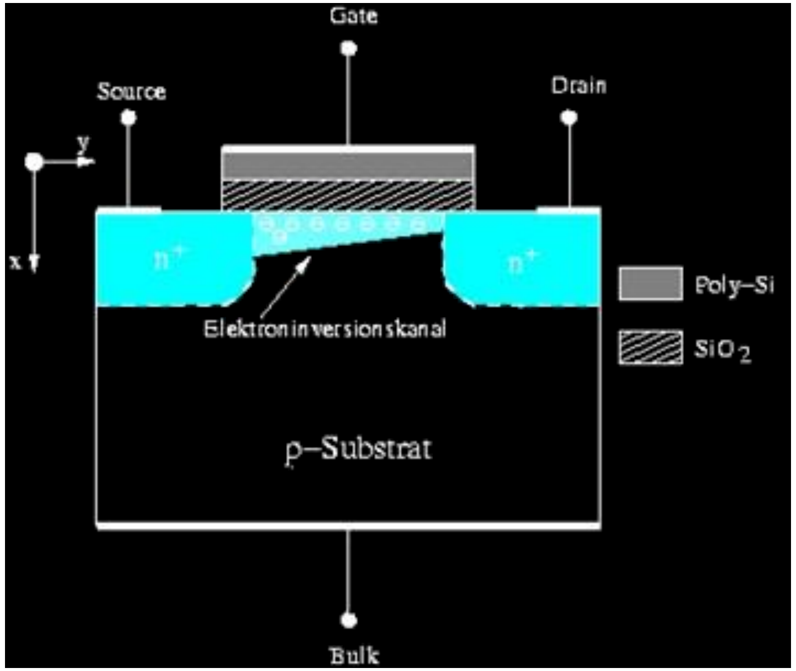


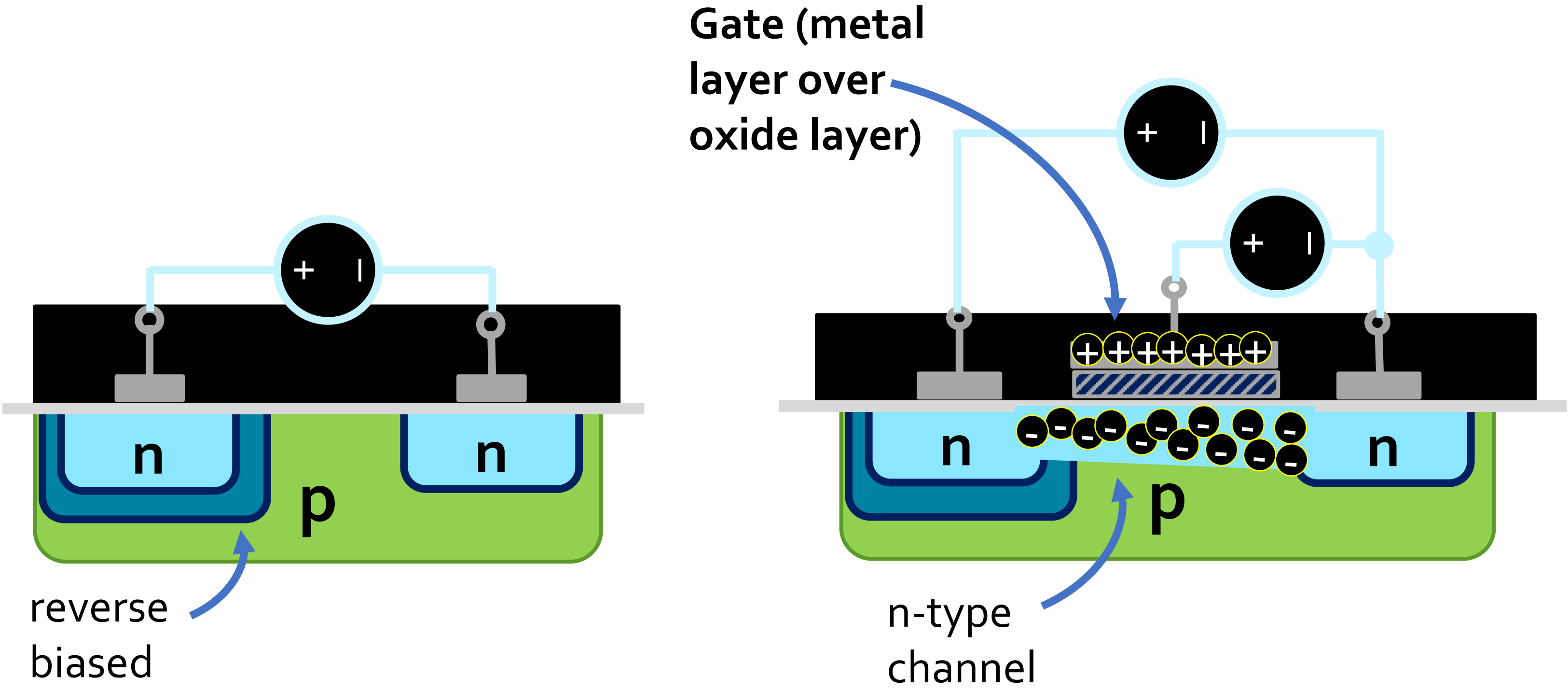
# n-channel MOSFETs

- However, when a voltage applied between the source and the metal plate (the **gate**), positive charges are built up in the metal layer, which attracts layer of negative charge to surface of the p-type material.




- This layer of electrons creates an n-type channel between the drain and the source, connecting the two and allowing current to flow between them.
  - the wider the channel, the higher the current

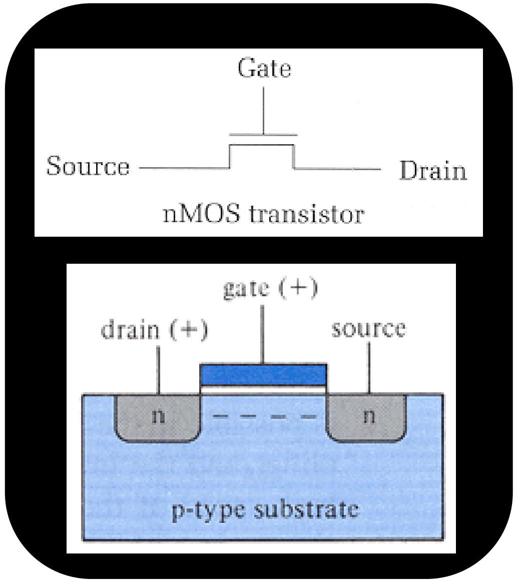
# Applying voltage to NPN

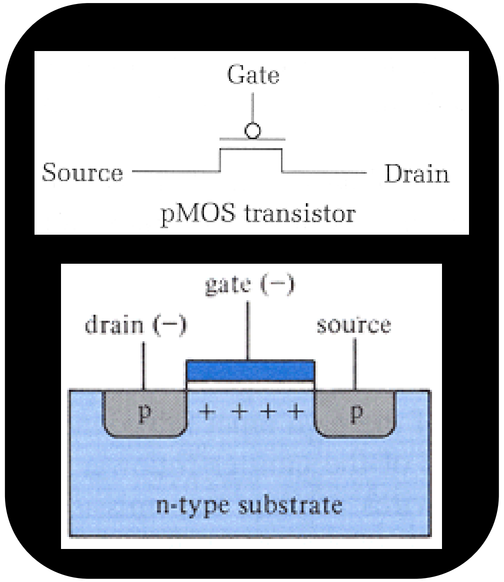



n-type channel creates path between source and drain for current to conduct!

# nMOS vs pMOS

- Two types of MOSFETs exist, based on the semiconductor type in the drain and source, and the channel formed.
  - nMOS transistors (the design described so far) conduct electricity when a positive voltage (5V) is applied to the gate.
  - pMOS transistors (indicated by a small circle above the gate) conduct electricity (i.e., act as a closed switch) when the gate voltage is logic-zero.



# Transistors to Logic Gates

# Transistors to Gates

- MOSFETs can make current flow, based on the voltage values in the gate and drain.
  - i.e. the truth table on the right.
- One final step: combining MOSFETS to create high and low voltage outputs, based on high and low voltage inputs.
  - General approach: create transistor circuits that make current flow to outputs from high or low voltage, based on transistor input values.

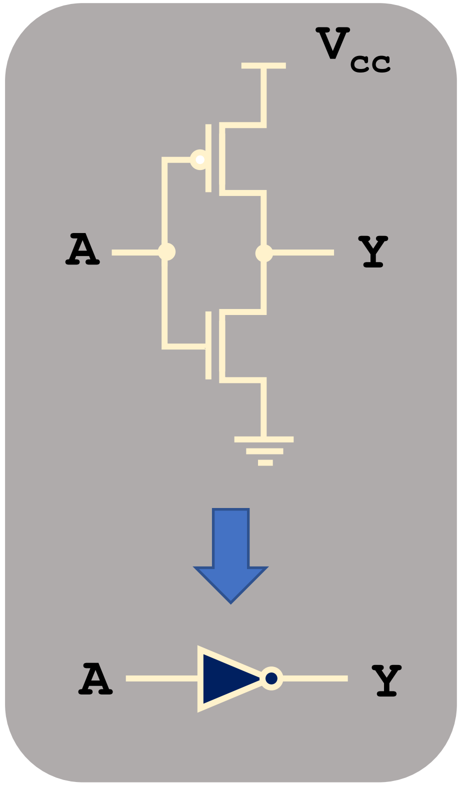
**MOSFET Truth Table**

| $V_{DS}$ | $V_{GS}$ | $I_{DS}$ |
|---|---|---|
| Low | Low | Low |
| Low | High | Low |
| High | Low | Low |
| High | High | High |

# Create gates using a combination of transistors

Physical data:

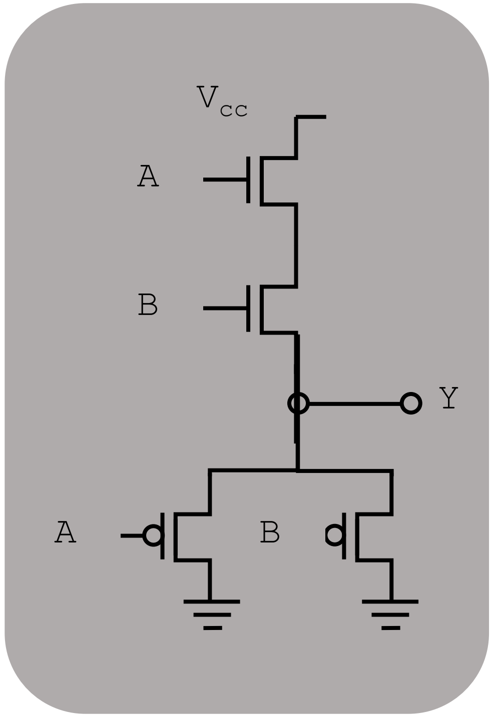
- “High” input = 5V (aka $V_{cc}$)
- “Low” input = 0V
- Switching time: ~20 picoseconds
- Switching interval ~10 ns



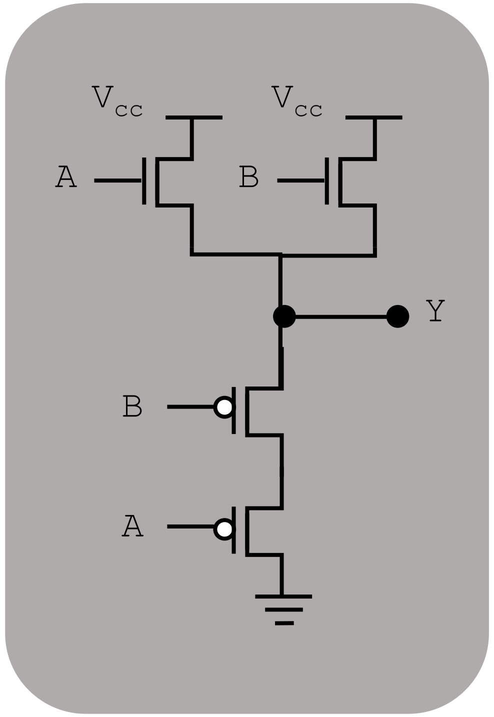

NOT Gate

# Transistors into gates

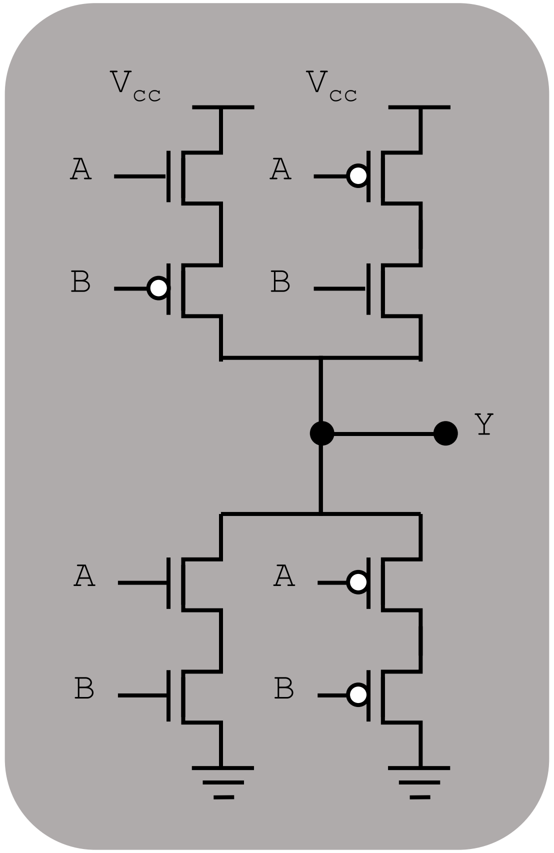
Note: `Vcc` = “Common Collector Voltage”
= high voltage (5 V)



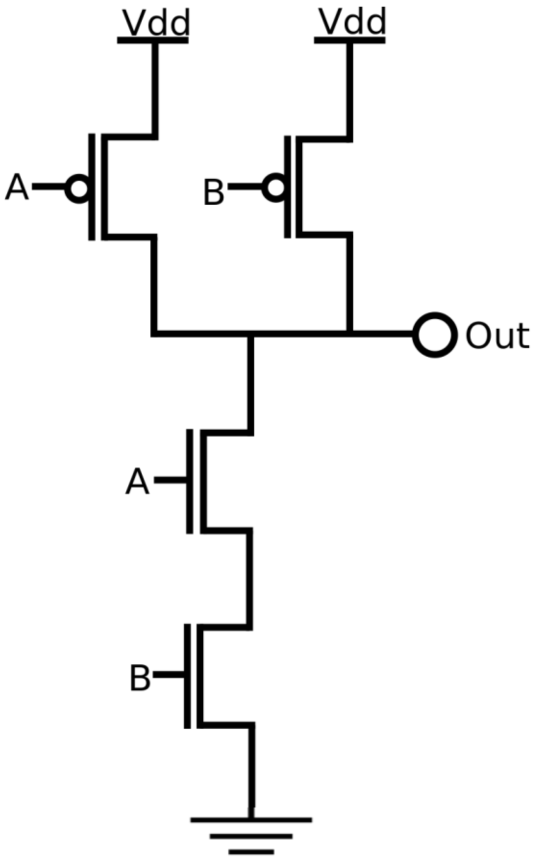

AND




OR




XOR



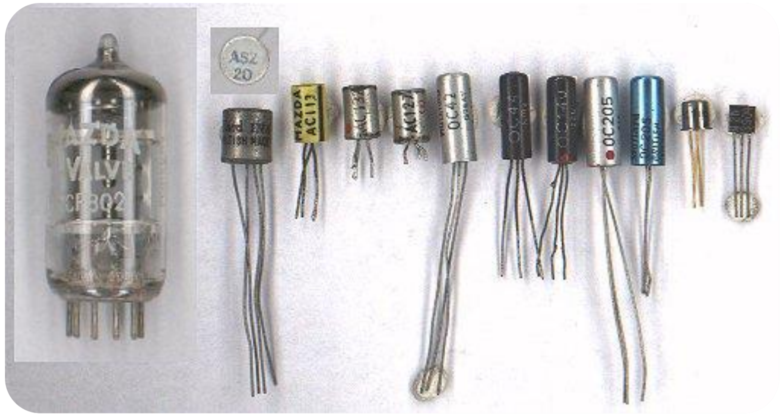

NAND is the most awesome logic gate

- It’s cheaper to build
- All other logic functions (AND, OR, …) can be implemented using only NAND, i.e., it is functionally complete

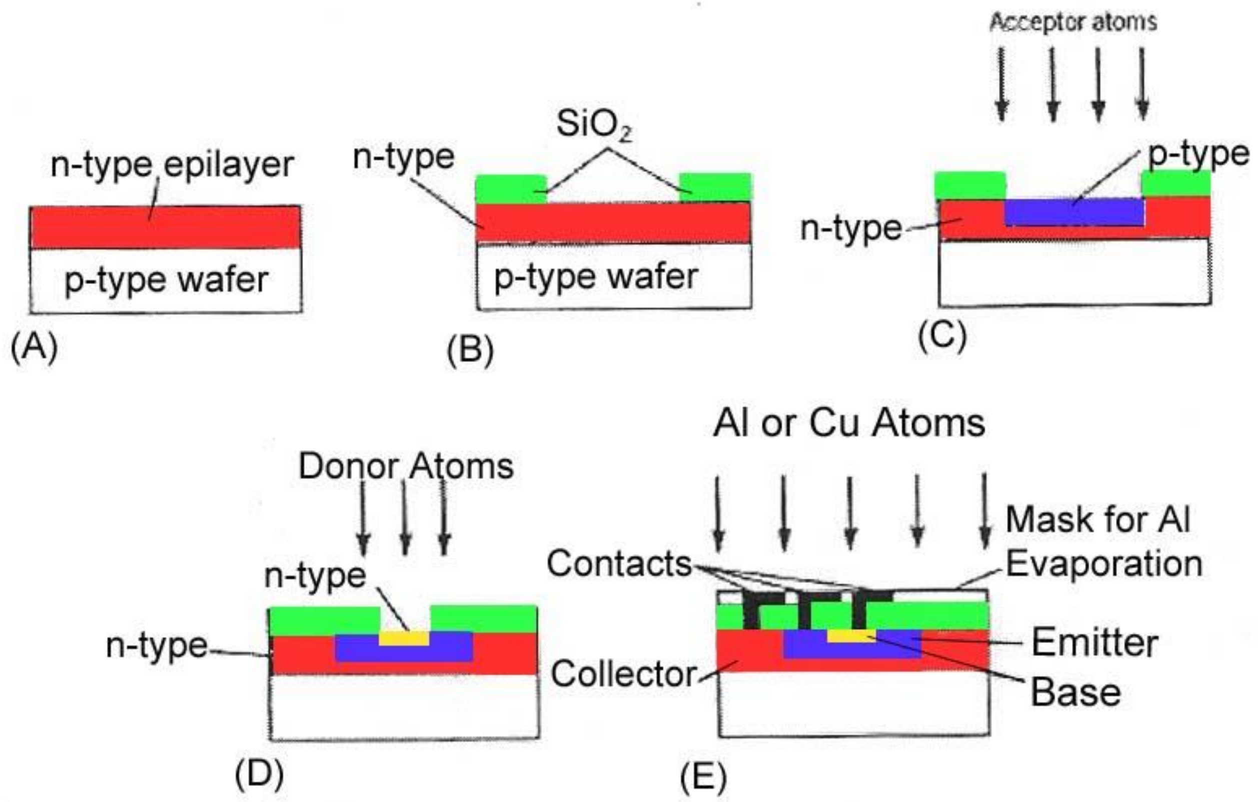
Challenge for home: implement AND, OR, NOT, XOR using only NAND.

# Transistor Fabrication

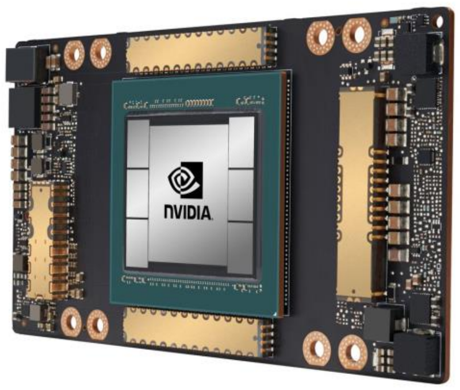
- Transistors are not formed by pushing large chunks of n- and p-type semiconductors together.
- Transistors are now made by bombarding silicon with doping substances to create the layers for each junction
  - Surface is oxidized in between stages to ensure that only the necessary sections are doped.



# Fabrication Process



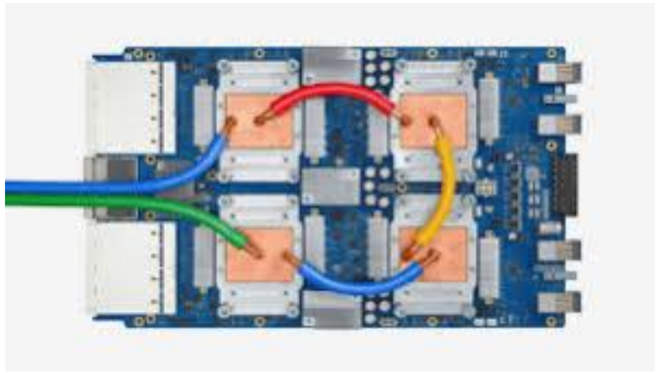
# CSCB58: Computer Organization




Prof. Gennady Pekhimenko

University of Toronto

Fall 2020



*The content of this lecture is adapted from the lectures of Larry Zheng and Steve Engels*