# CSC258 Week 1

# Instructor

Larry Zhang

Office: Virtual

Email: ylzhang@cs.toronto.edu

# Today's outline

- **Why** CSC258
- **What** is in CSC258
- **How** to do well in CSC258

- Start learning

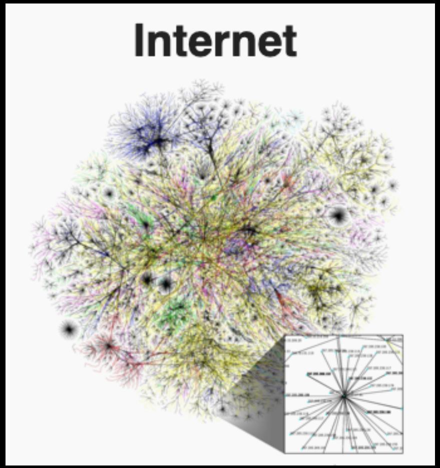
# **Why** take CSC258?

# Learning the Magic


magic

magic

magic




CSC258

CSC369

CSC358

# 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 "amongus.exe" file, what exactly happens when I double-click on it?
- How does the CPU run an if-statement, or for loop, or recursion?

**CSC258 has all the answers!**

# After learning CSC258…

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

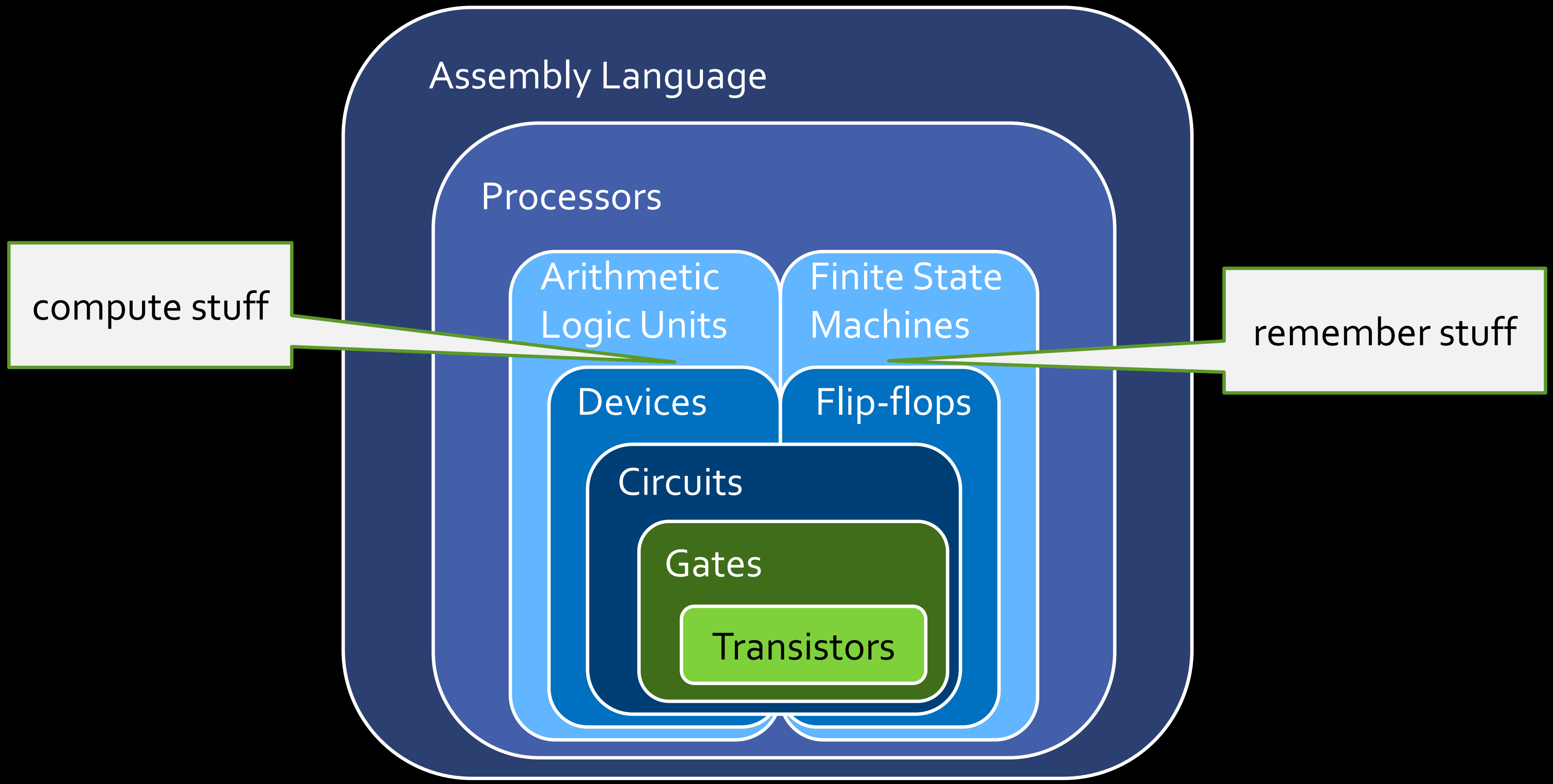
- With your hardware knowledge, you will be able to engineer the performance of your software like never before.

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

***-- Alan Kay***

# **What's** in CSC258?

# 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 **CSC258**

# 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 CSC258

# First of all …

Be interested

# Course website

https://mcs.utm.utoronto.ca/~258/

**All course materials are here.**

# Marking scheme

| Type | Description | Due Date | Weight |
|---|---|---|---|
| Lab | Weekly labs | On-going | 24% |
| Term Test | Midterm test | 2021-02-24 | 24% |
| Final Exam | Final Exam | TBA | 34% |
| Quiz | Weekly online quizzes | On-going | 9% |
| Assignment | Assembly Programming Project | 2021-04-09 | 9% |
| | | **Total** | 100% |

# Labs (starting from Week 2)

- Hands-on exercises.
- For credit: lab report submitted after each lab, **individually**.
  - Deadline: Tuesday at 10 PM, submitted on MarkUs.
    - one-hour lateness with 2% penalty
    - NOT accepted if more than one hour late
- Go to the lab section that you are registered to on ACORN. If you want to switch lab section, get permission from the instructor.

# Lab software

- We will use Logisim-Evolution
- The reference of the software has been posted on the course website.
- Task for this week: download the software, read the reference, and familiarize yourself with it.
- Note: I must be **Logisim-Evolution** downloaded at the link in the reference. Do NOT use the original Logisim or its other variations.

# Weekly Self-Assessment Quizzes

- We will use Quercus for online quizzes
- Starting in Week 2, posted on Friday after lectures.
- Deadline for quizzes is every Wednesday 10:00pm.
  - Late submissions are NOT accepted.
- You'll be told whether your answer is correct immediately, and you can try up to three times. We will record your highest score.
  - Note: you might get different versions of the question in different attempts.

# Assembly Programming Project

- Write a larger project in assembly
  - e.g., a game

# Tests

- Midterm
  - outside class!
  - Wed, Feb 24, 7:10pm~8:40pm (90 minutes)
  - **Let Larry know by Jan 31 if you have a conflict.**
- Final exam
  - Some time in April
  - Must get >= 40%

# Discussion board (Piazza)

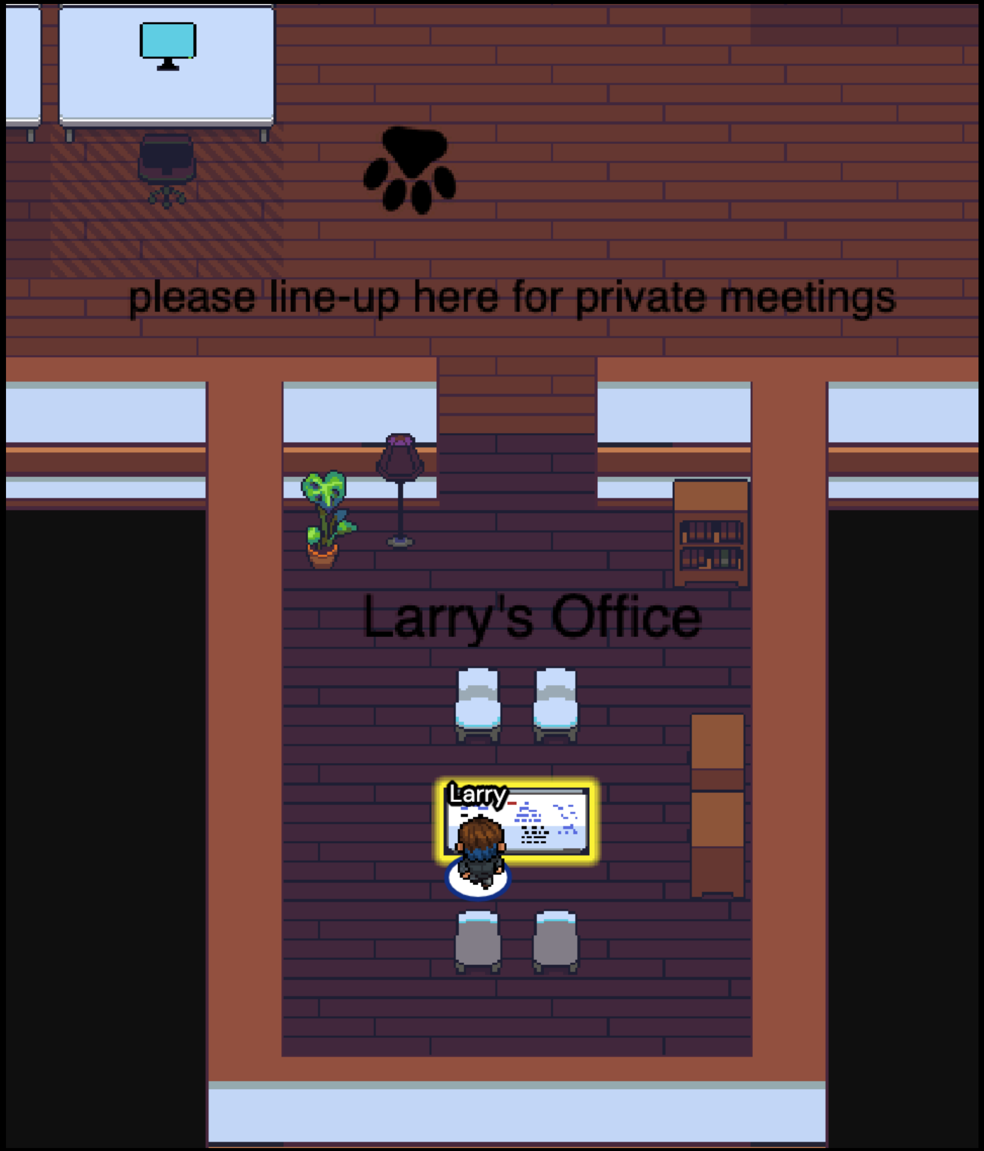
- Link on the course website
- All course announcements will be posted here.
- **Daily** reading is required.

# Office hours

Virtual on Gather Town
(link on course website)

Thursday 1:10-2:30 PM
Friday: 3:10-4:30 PM

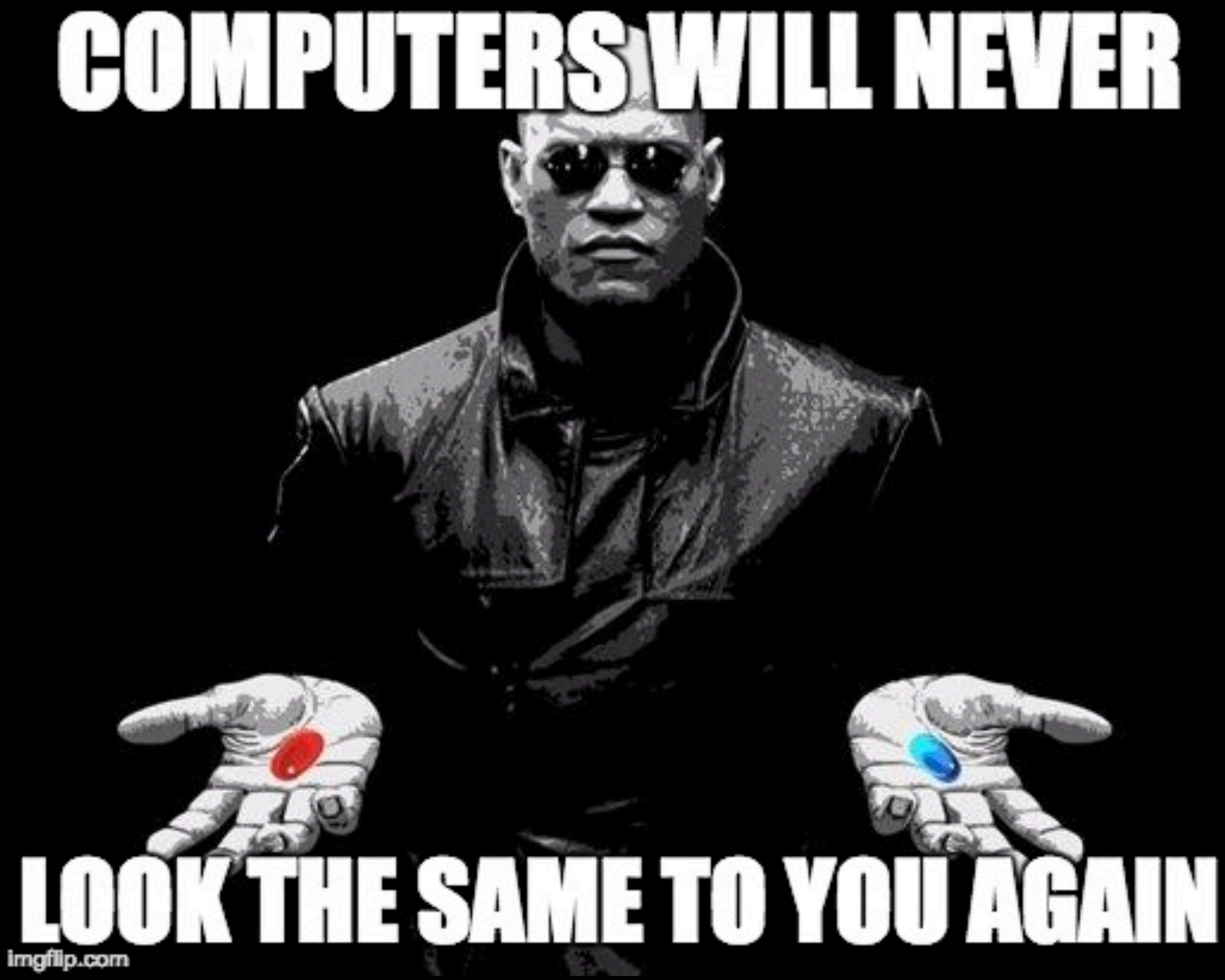
Office hours are helpful!



# A typical week of CSC258

- Monday: go to labs
- Tuesday by 10 PM, submit lab report
- Wednesday by 10 PM, submit quiz
- Wednesday: next lab’s handout posted, start working on it
- Thursday/Friday: lectures and office hours
- Friday: next quiz posted, start reviewing the content

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


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

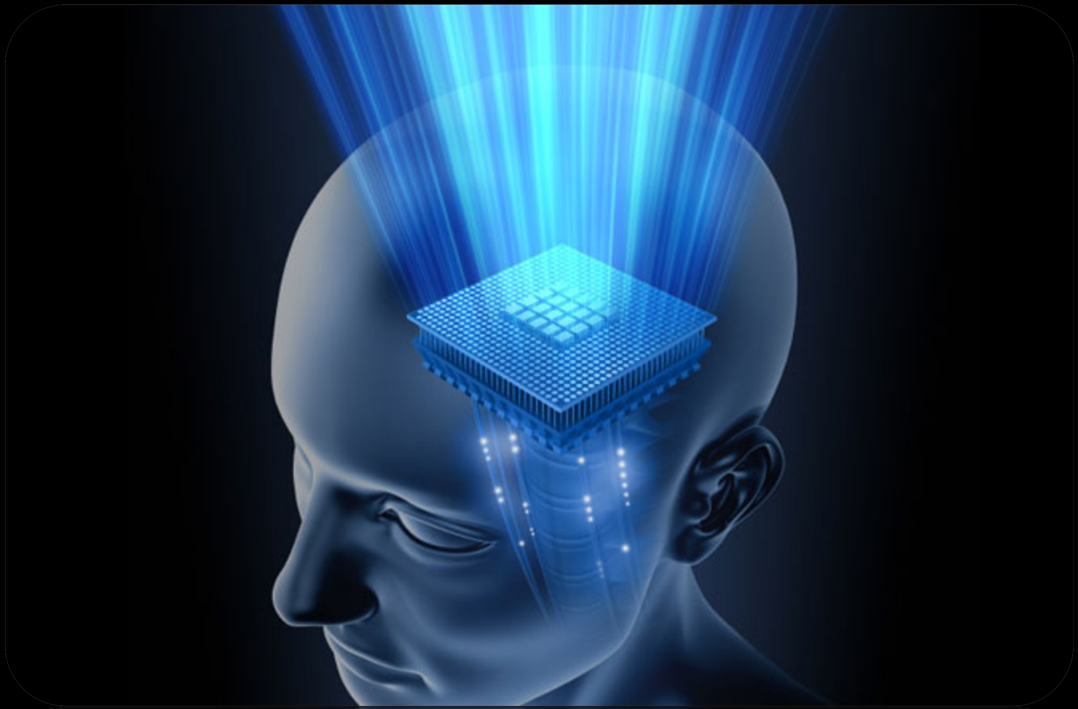
# Kahoot!

- In-class pop quizzes. To participate, you'll need:
  - be in the lecture
  - have access to a browser (on a phone, tablet or a laptop), or the Kahoot app
- has nothing (directly) to do with your course grade

# Let the learning begin

# Basic Logic Gates

# You already know something…

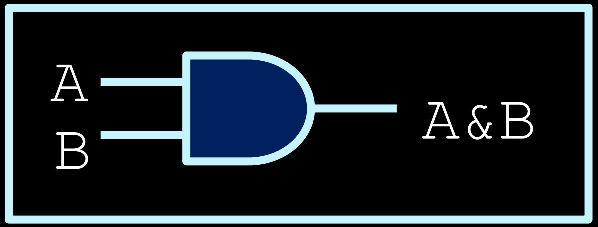


# Logic from math course

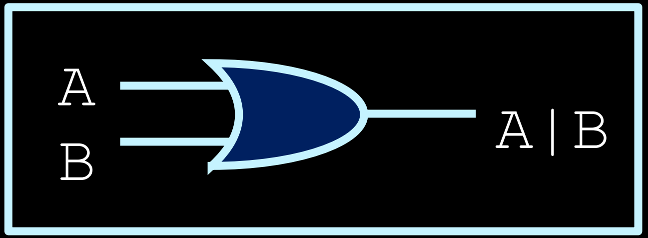
- Create an expression that is true iff the variables `A` and `B` are true, or `C` and `D` are true.

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

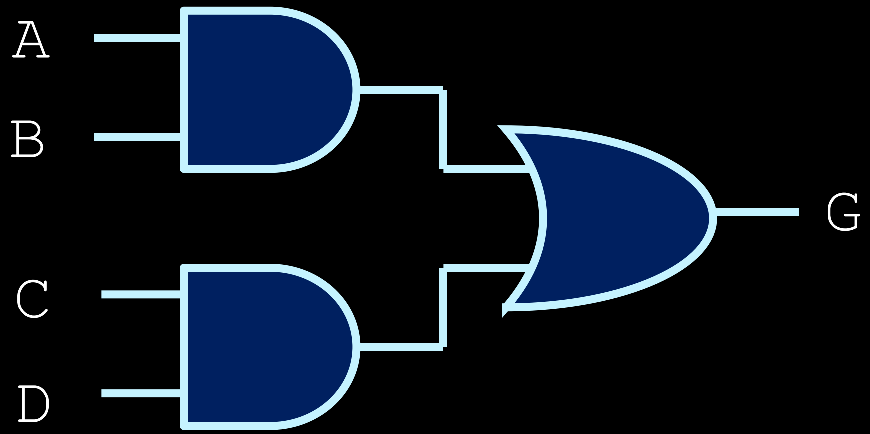
```
G = (A & B) | (C & D)
```




AND Gate




OR Gate




You just designed your first circuit in CSC258!

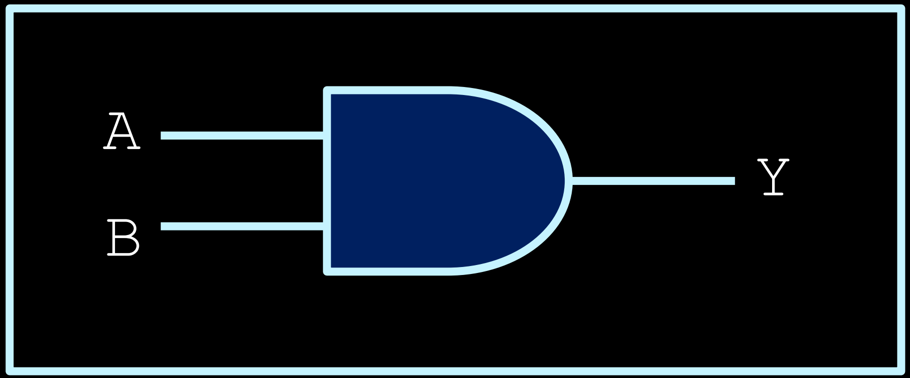
# Gates = Boolean logic

- 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.

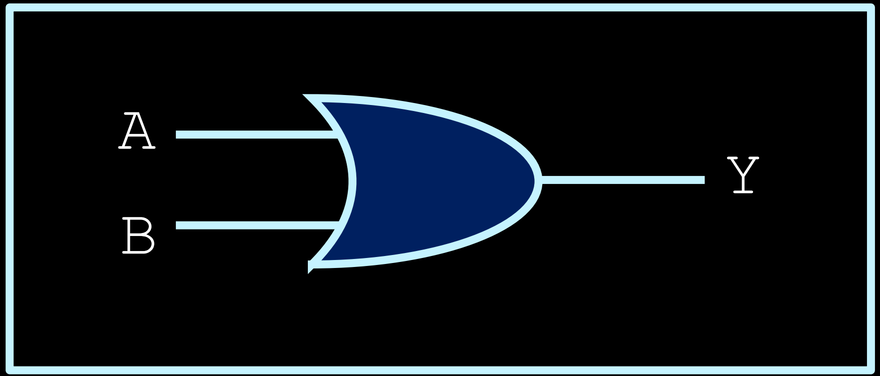
# AND Gates




Truth table

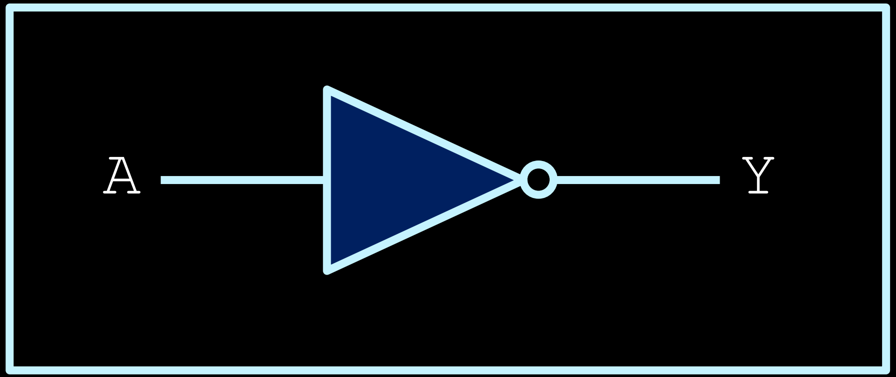
| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# OR Gates



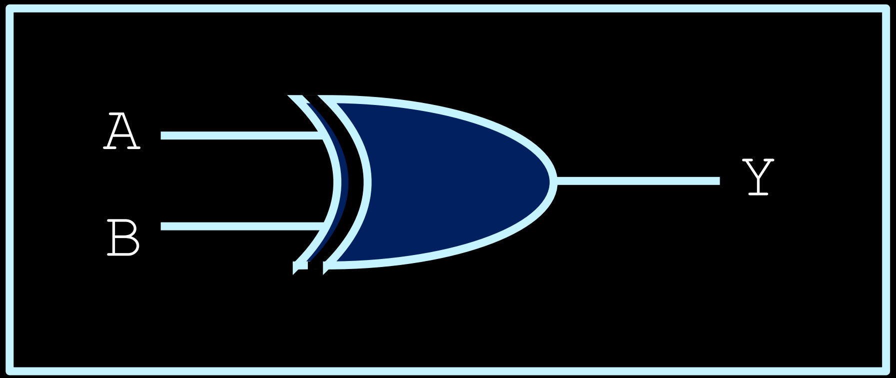

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# NOT Gates




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

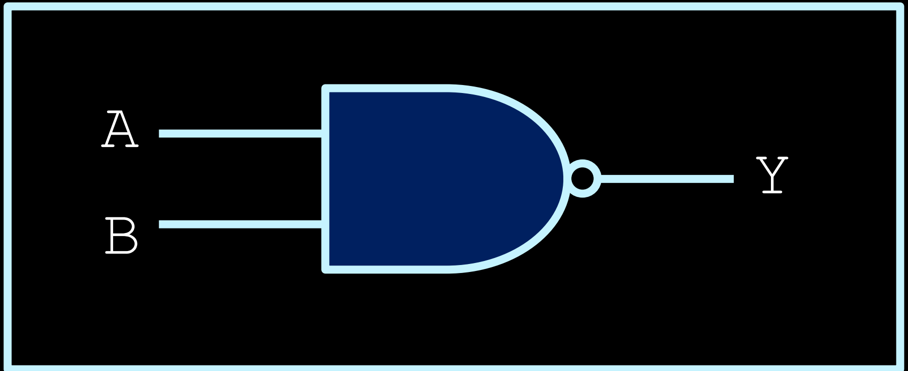
# XOR Gates




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

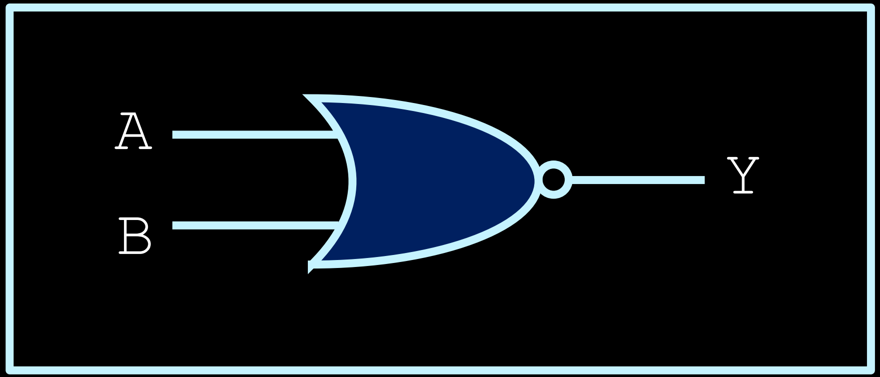
# Bill Gates

# NAND Gates



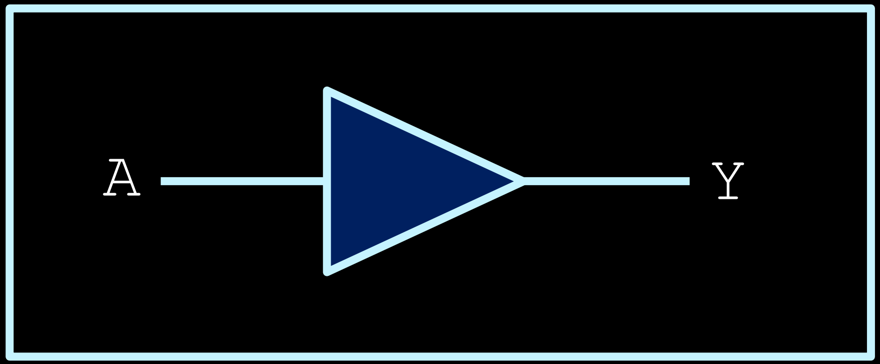

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# NOR Gates



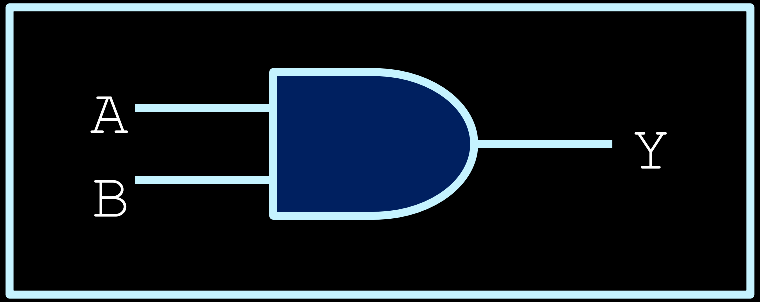

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

# Buffer

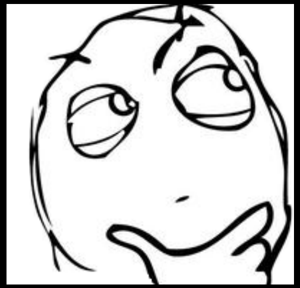



This is not as silly as you might think now, as we'll see later...

| 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?

# Transistors

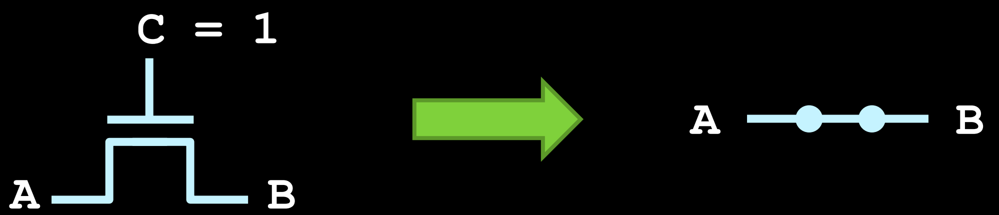
# One of the greatest inventions of the 20th century

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

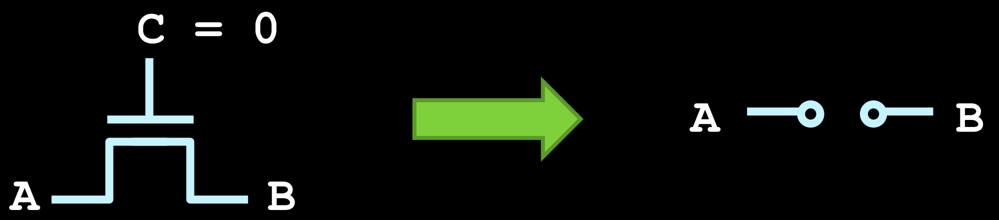
Building block for the hardware of all your computers and electronic devices.

# What do transistors do?

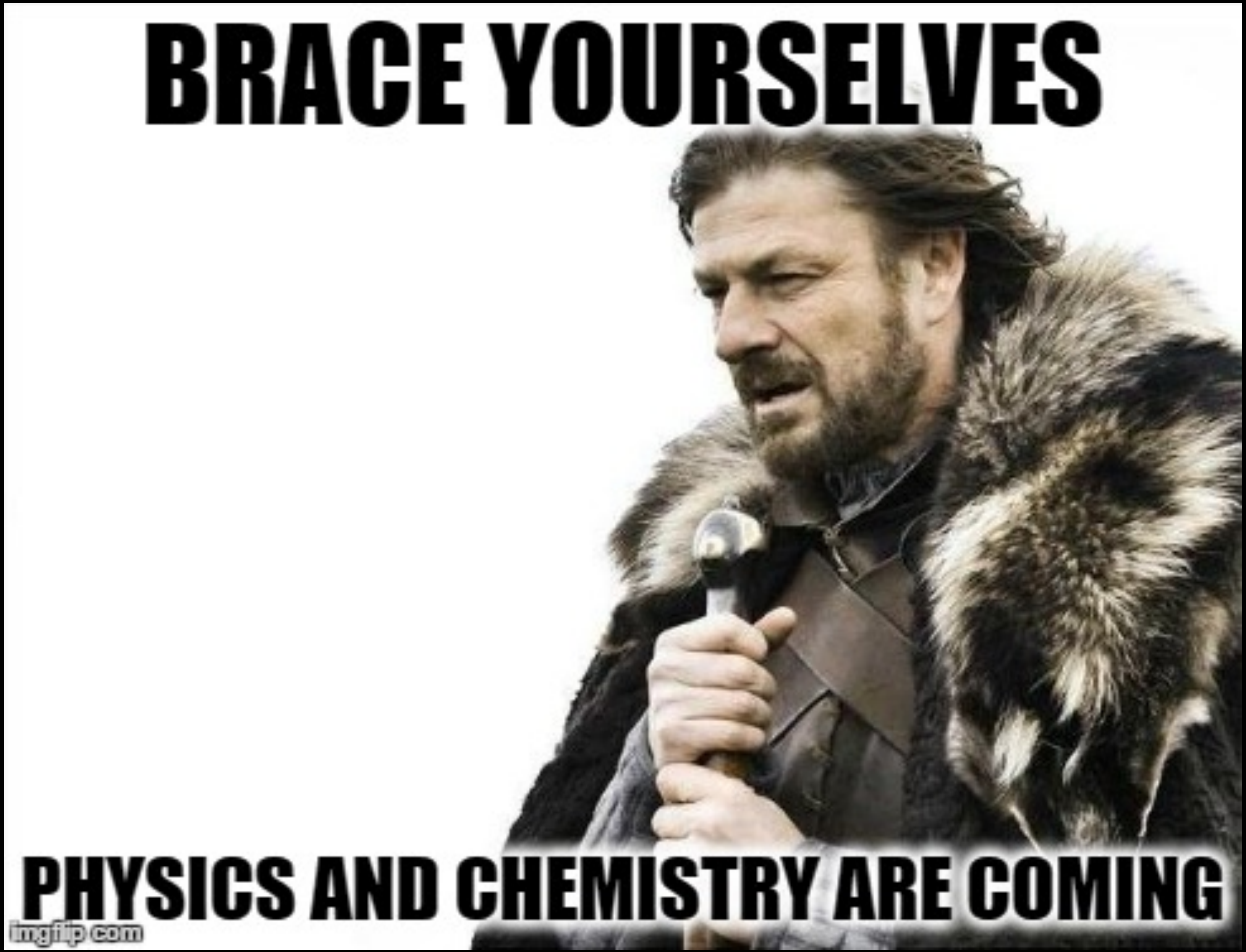
- Transistors connect Point A to Point B, based on the value at Point C.
  - If the value at Point C is high, A and B are connected.




  - 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

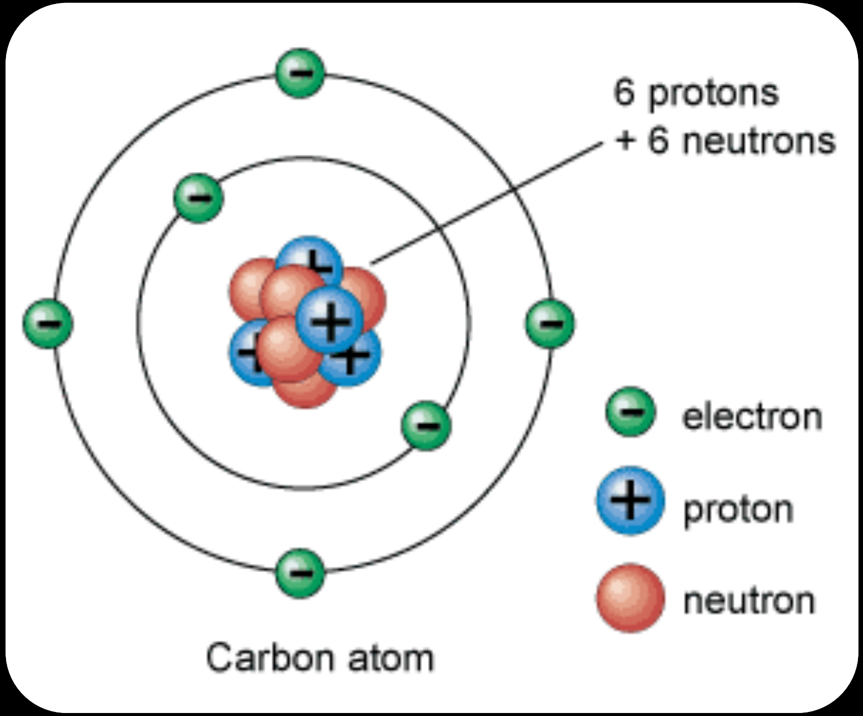
# Outline of the story

- 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

# Electricity Basics

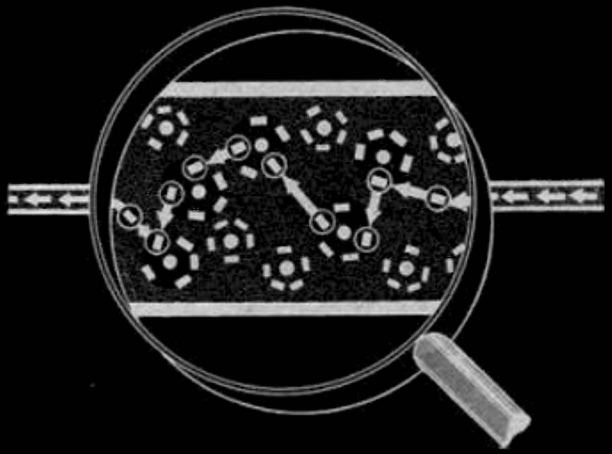
# Everything is made of atoms …

- Protons are big (hardly move) and positively charged.
- Electrons are small (easily move) and negatively changed.
- 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.



# Resistance

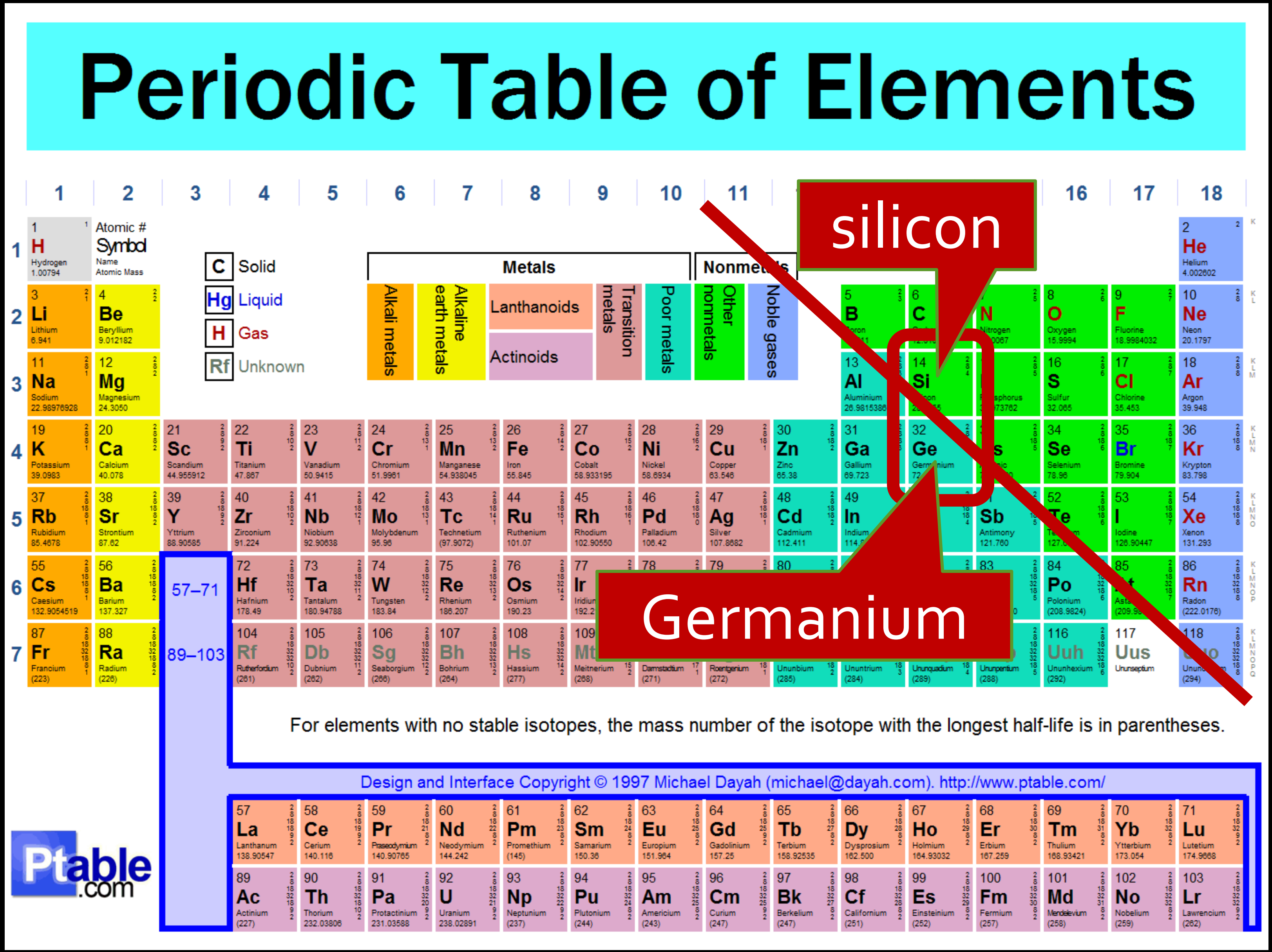
- Electrical resistance indicates how well a material allows electricity to flow through it:
  - High resistance (aka insulators) don't conduct electricity at all.
  - Low resistance (aka conductors) conduct electricity well and are generally used for wires.

- 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


Periodic Table of Elements
silicon
Germanium
Atomic #
Symbol
Name
Atomic Mass
C Solid
Hg Liquid
H Gas
Rf Unknown
Metals
Nonmetals
Alkali metals
Alkaline earth metals
Lanthanoids
Actinoids
Transition metals
Poor metals
Other nonmetals
Noble gases
57–71
89–103
For elements with no stable isotopes, the mass number of the isotope with the longest half-life is in parentheses.
Design and Interface Copyright © 1997 Michael Dayah (michael@dayah.com). http://www.ptable.com/
Ptable.com

# Conductivity of Semiconductors

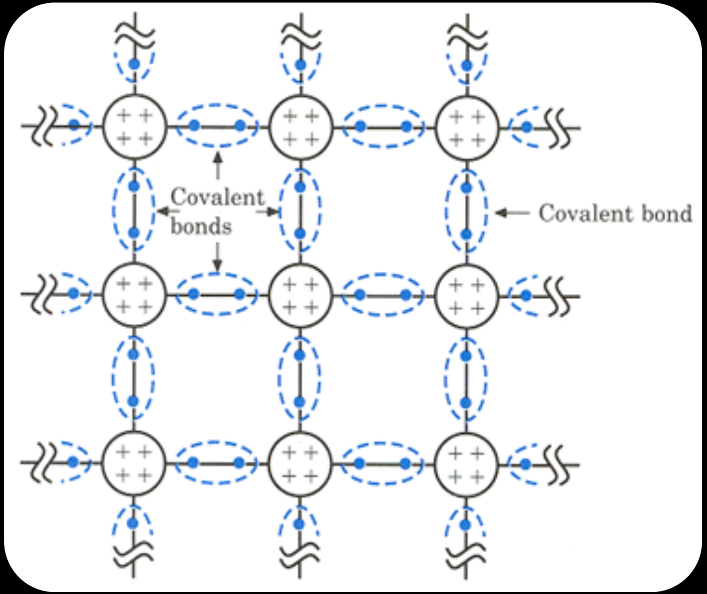
- Semiconductor materials (e.g., silicon and germanium) straddle the boundary between conductors and insulators, behaving like one or the other, depending on factors like temperature and impurities in the material.

# Impurity

# Pure semiconductor is pretty stable

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

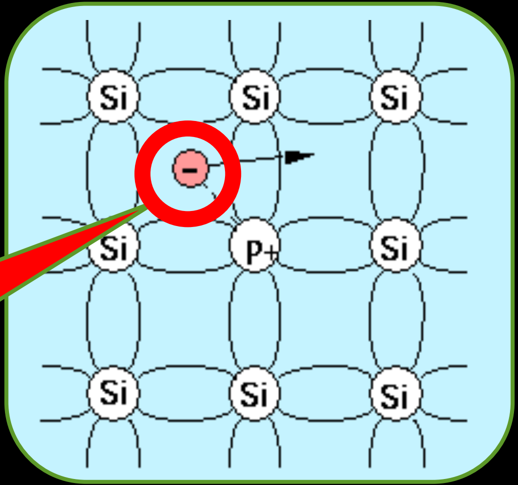
- At room temperature, very close to insulator.



# Encourage semiconductor's conductivity

**N-type:**

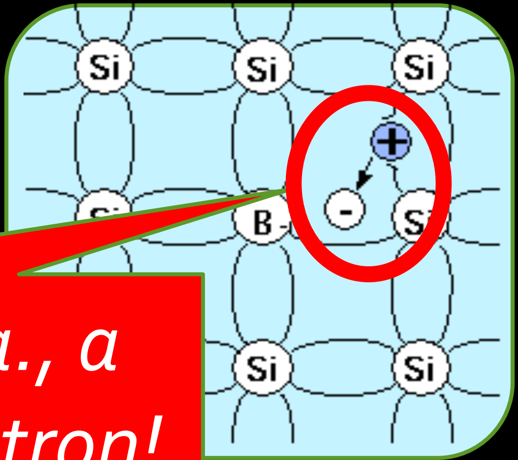
Add some atoms with **5** valence electrons, such as Phosphorus.




*An extra electron!*

**P-type:**

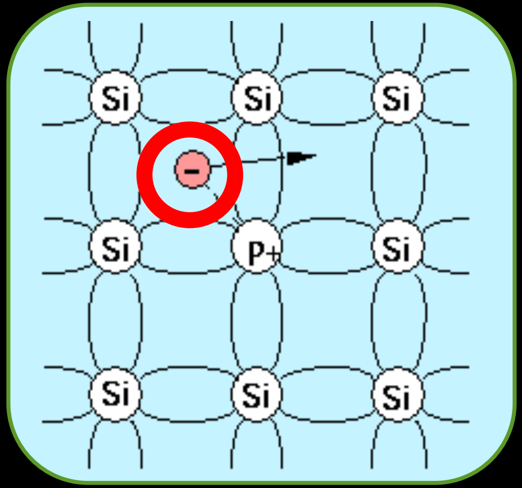
Add some atoms with **3** valence electrons, such as Boron.



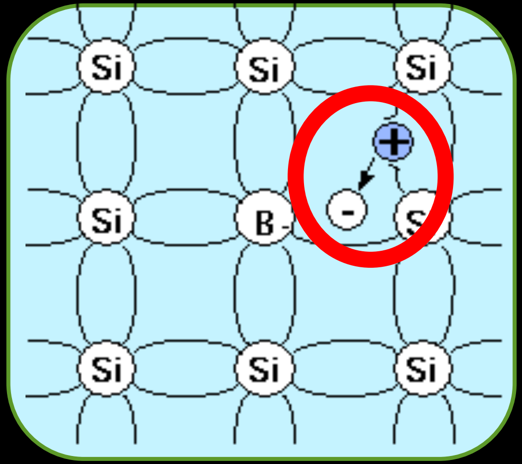

*A missing electron, a.k.a., a "hole", like a positive electron!*

# 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

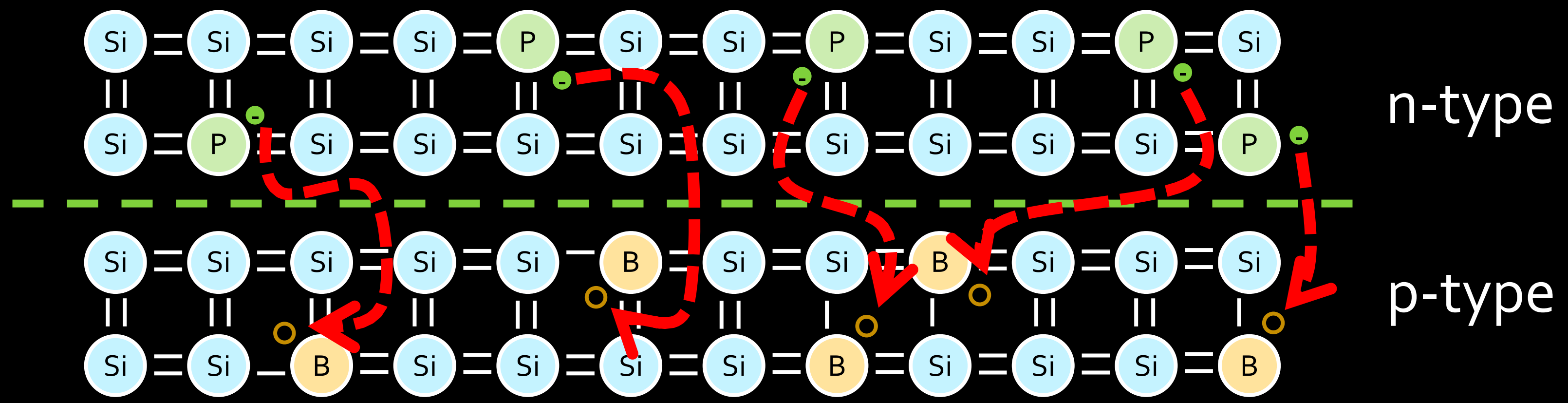
# Outline of the story

- 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

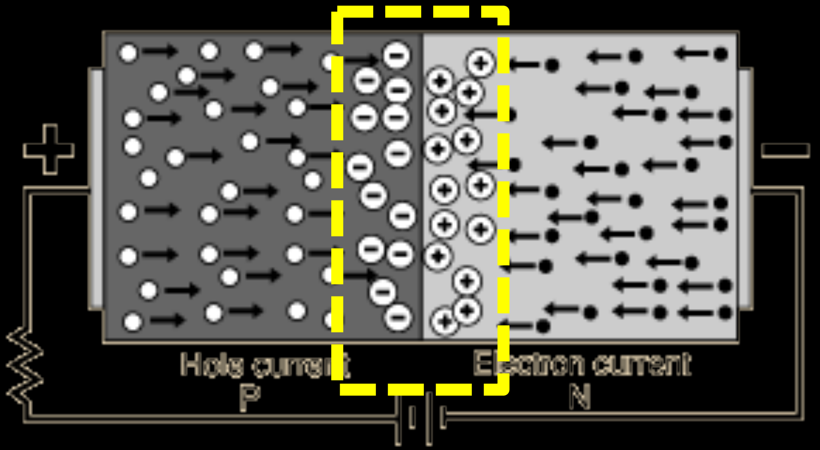
# Bringing p and n together

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



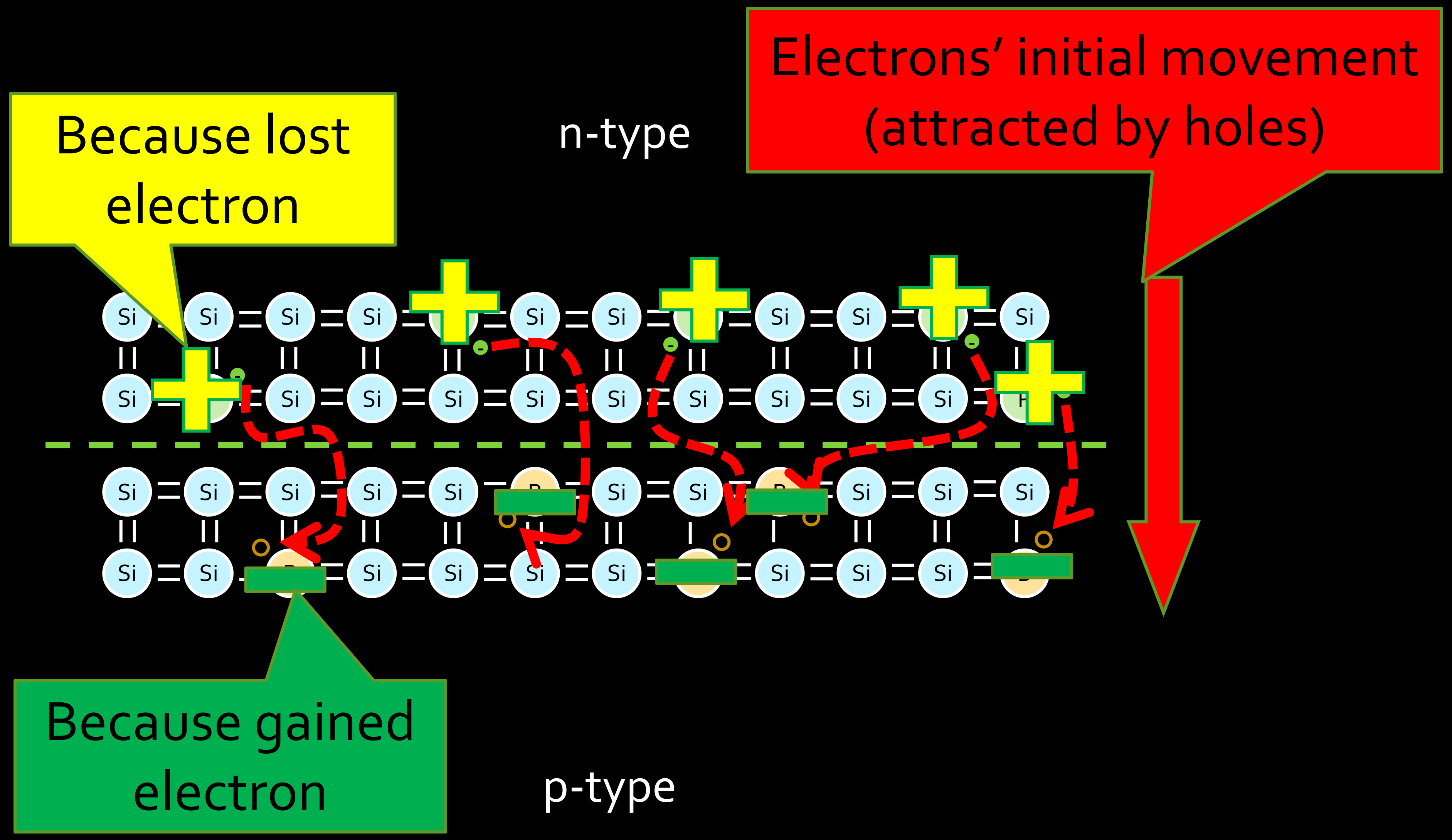

- The electrons at the surface of the n-type material are drawn to the holes in the p-type.

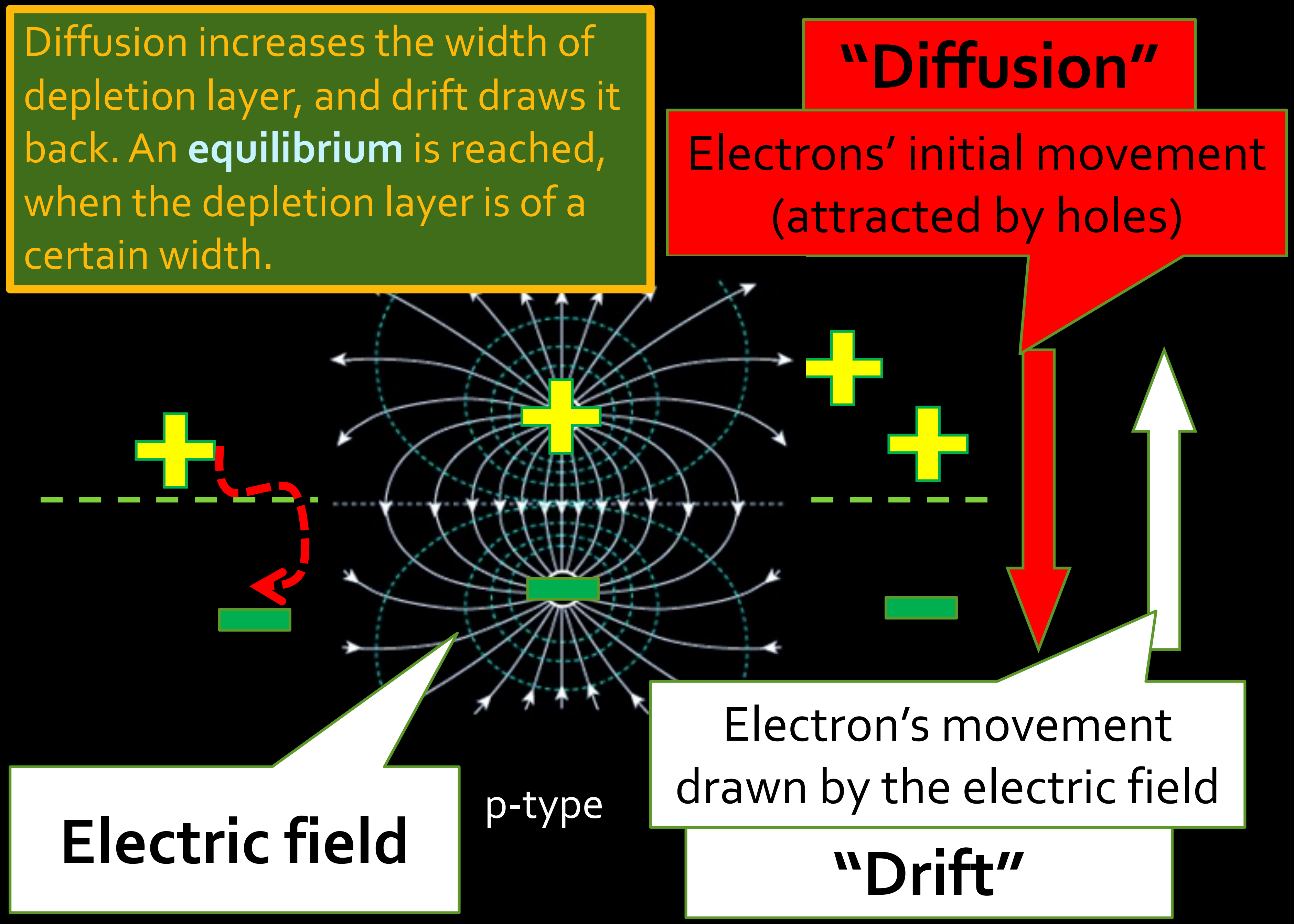
# p-n Junctions




- 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"**

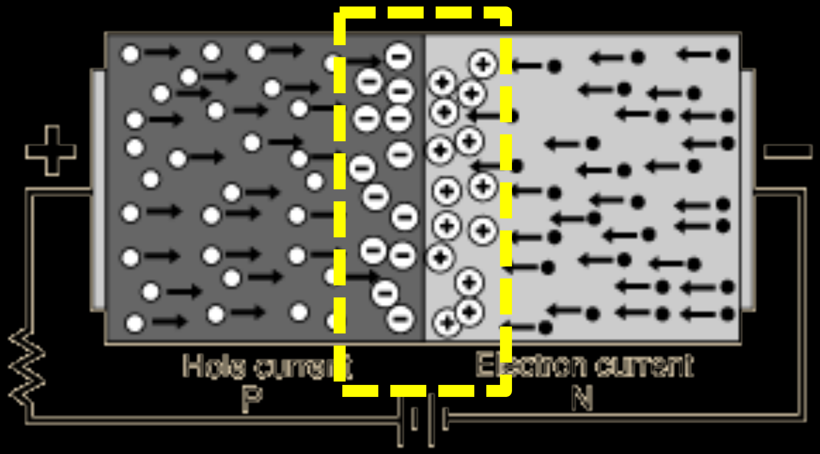
**"Drift"**

Attracted by gravity (hole)

Drawn by spring force field

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

# Summary of pn-junction

P  N



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

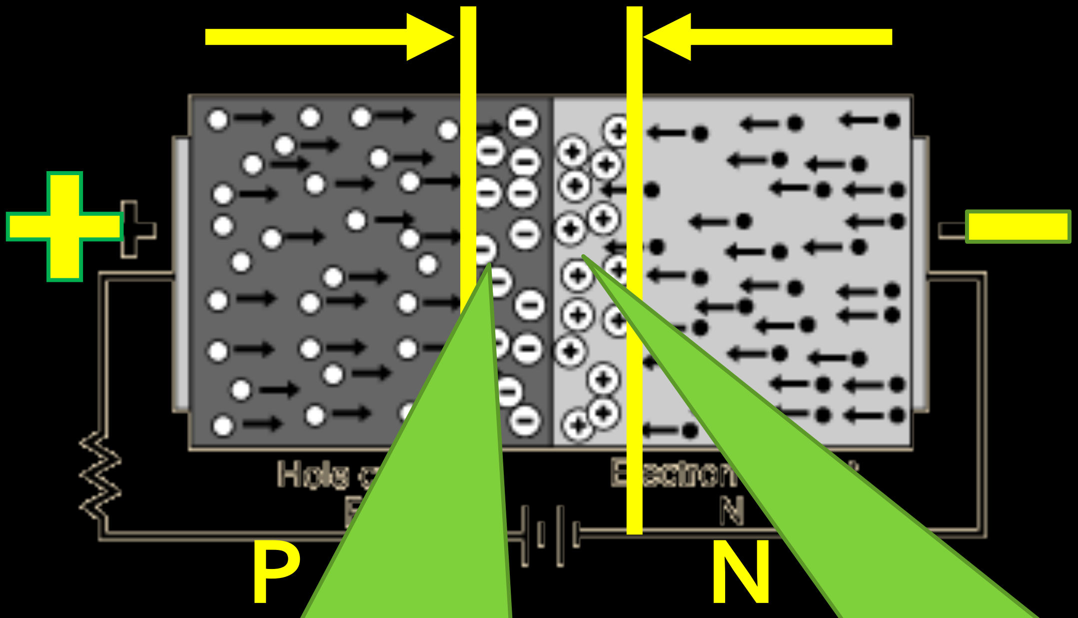
- Electricity, basic concepts
- Insulators, conductors, in between …, Semiconductors
- Impure semiconductors, p-type / n-type
- Put p-type and n-type together -- pn-junction
- Apply voltage to a pn-junction – principle of transistors
- A real-world manufacturing of transistor -- MOSFET

# 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)

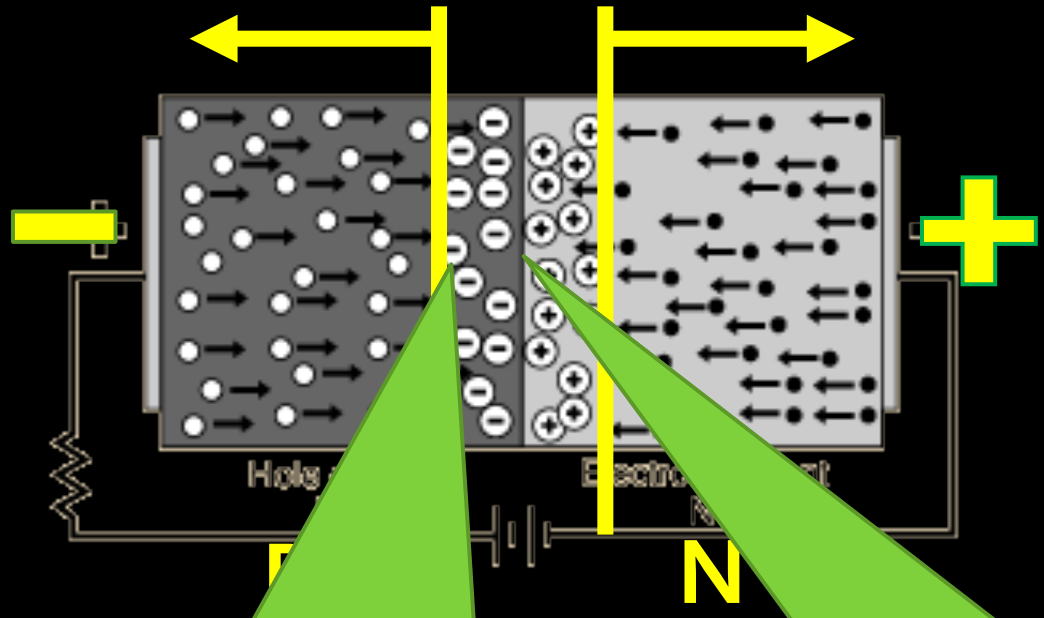



Negative charges sucked out of depletion layer

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!

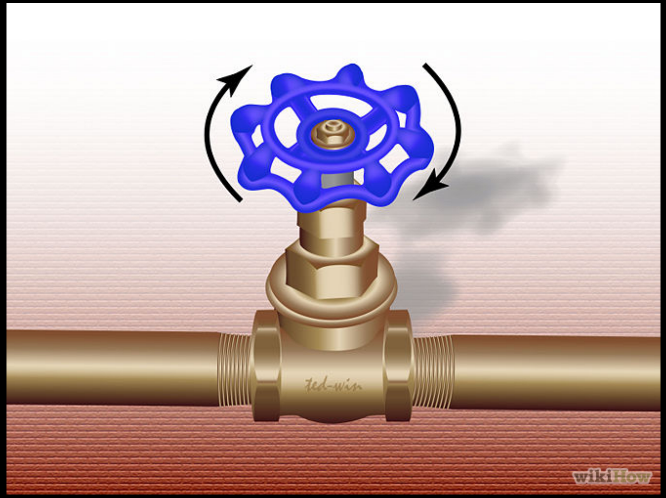
# Outline of the story

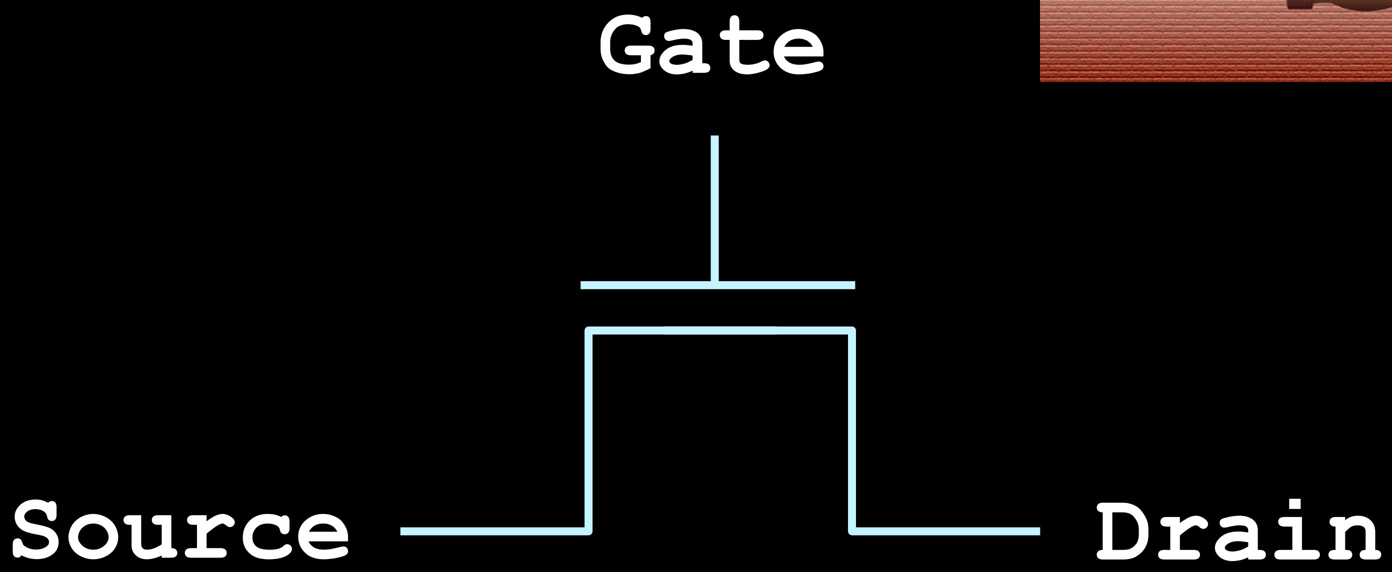
- Electricity, basic concepts
- Insulators, conductors, in between …, Semiconductors
- Impure semiconductors, p-type / n-type
- Put p-type and n-type together -- pn-junction
- Apply voltage to a pn-junction – principle of transistors
- ➡ A real-world manufacturing of transistor -- MOSFET

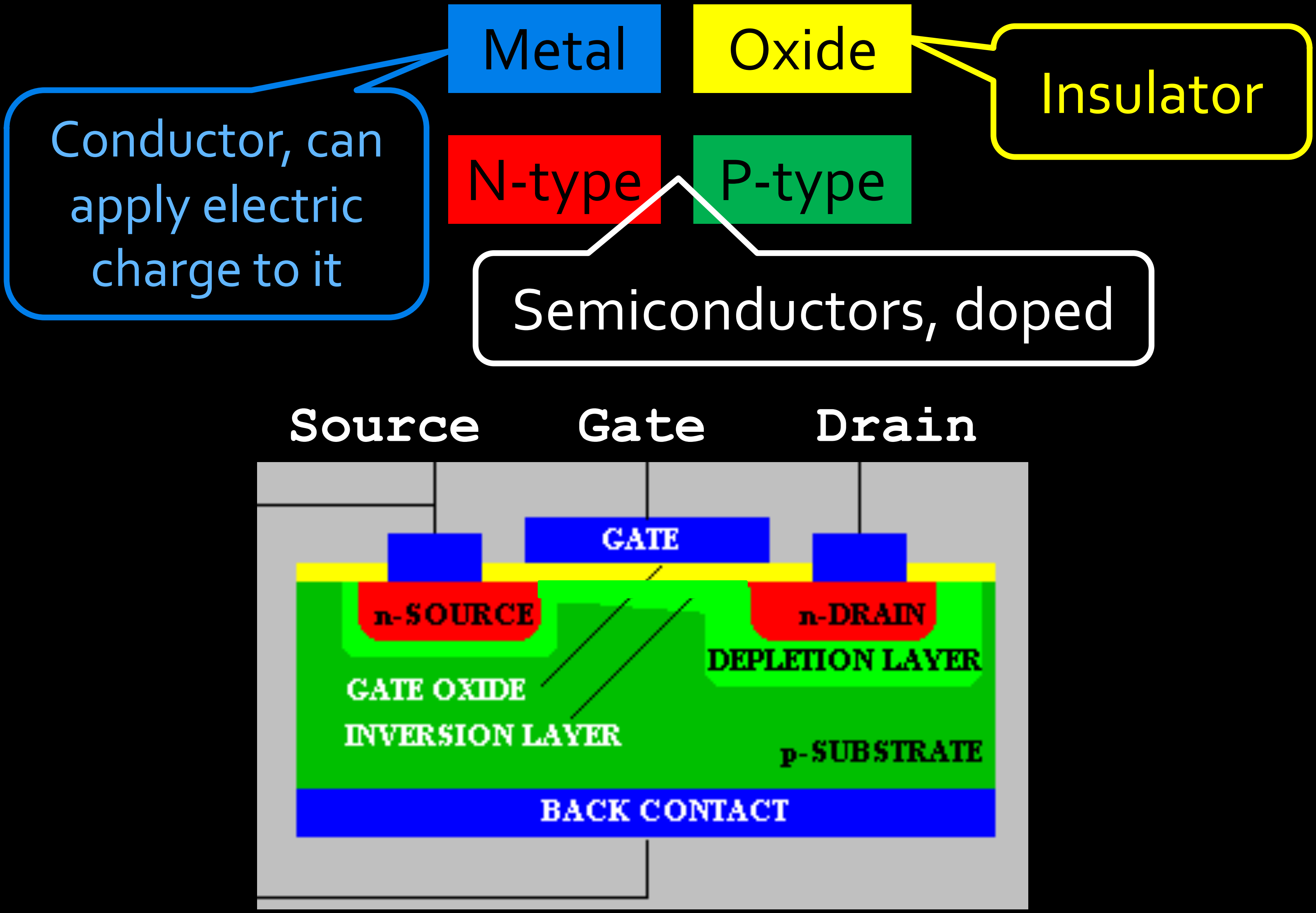
# Creating transistors

- 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.

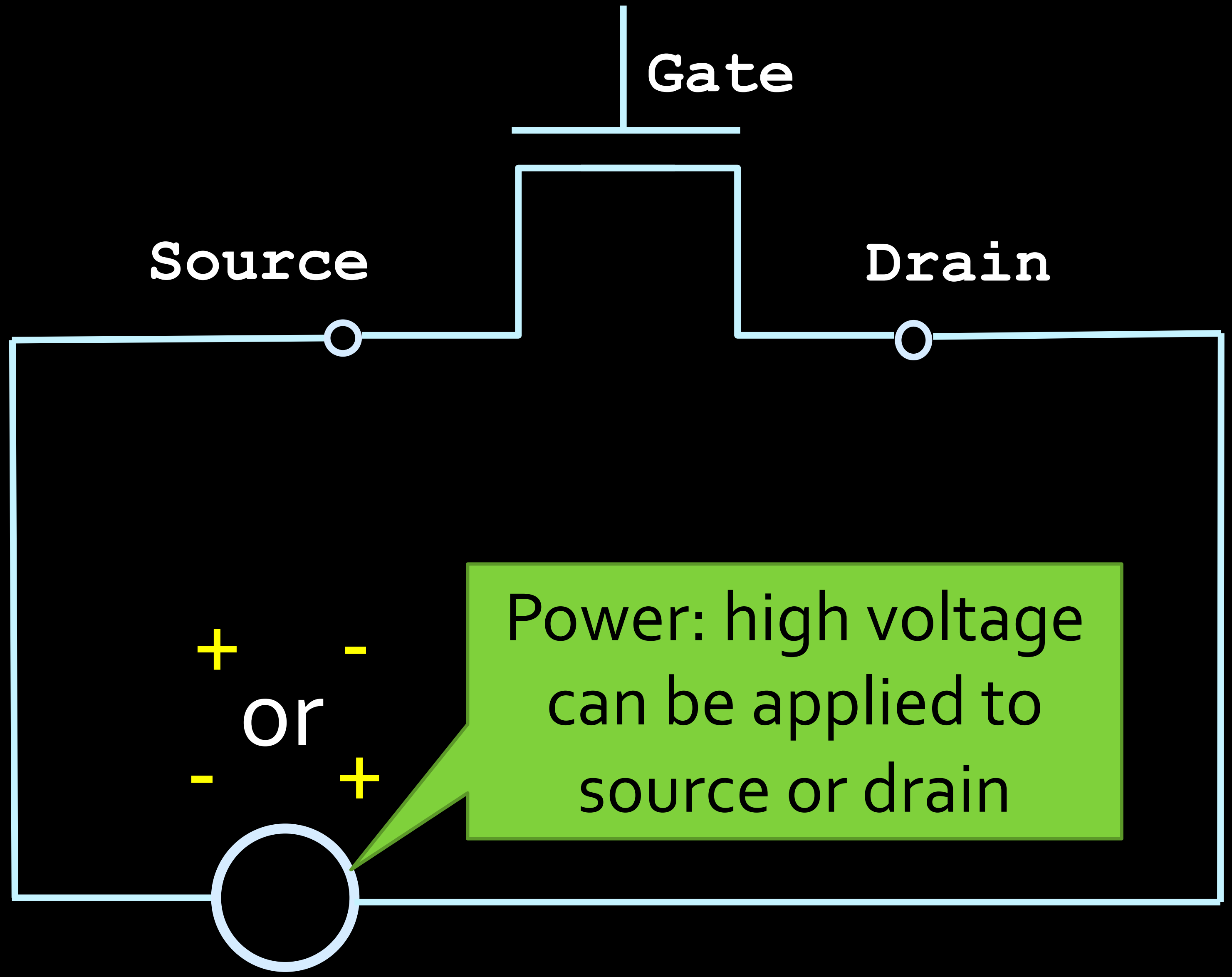
Metal Oxide Semiconductor
Field Effect Transistor


wikiHow

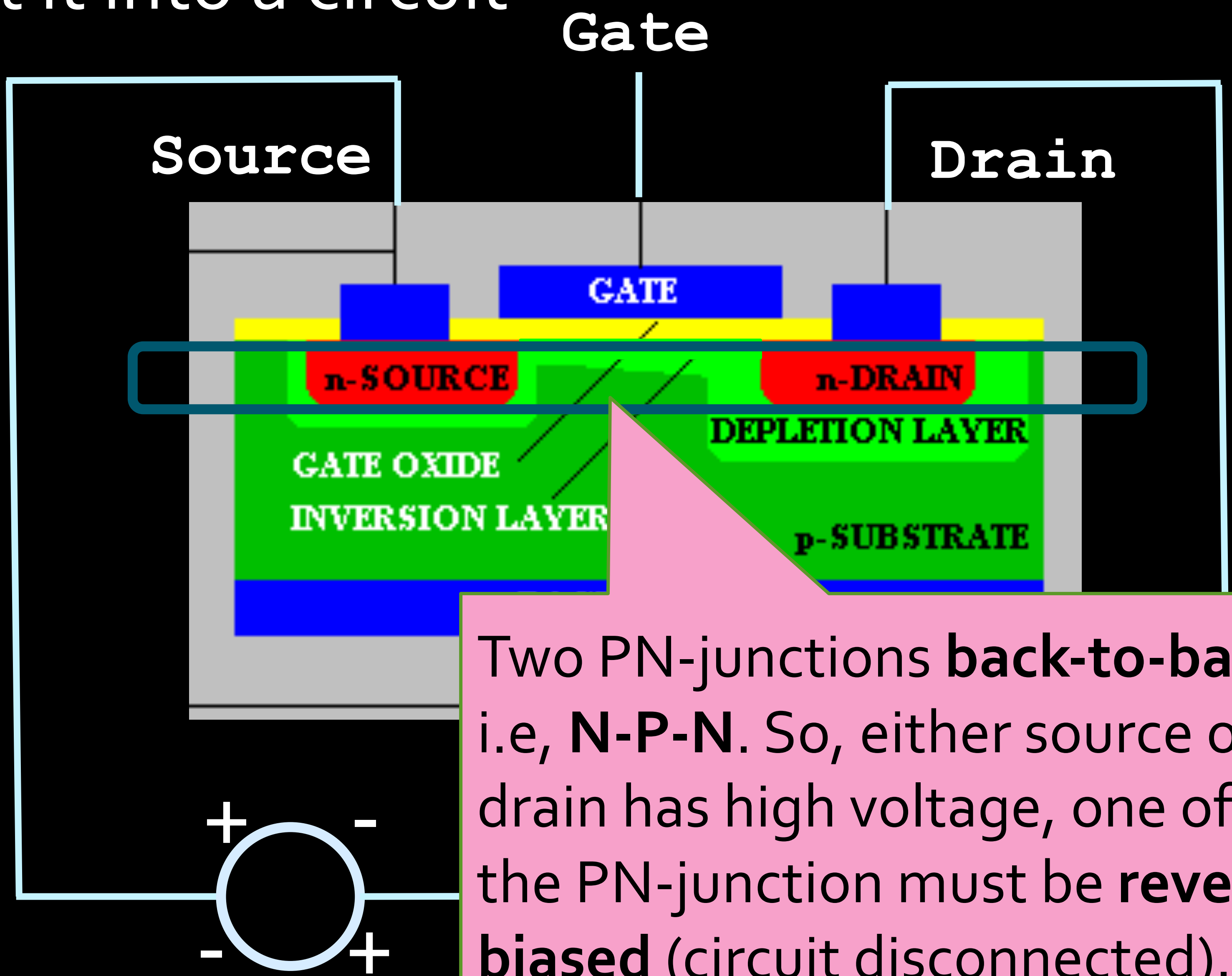

Gate
Source
Drain


Metal
Oxide
Insulator
Conductor, can apply electric charge to it
N-type
P-type
Semiconductors, doped
Source
Gate
Drain
GATE
n-SOURCE
n-DRAIN
DEPLETION LAYER
GATE OXIDE
INVERSION LAYER
p-SUBSTRATE
BACK CONTACT

# Put a MOSFET into a circuit



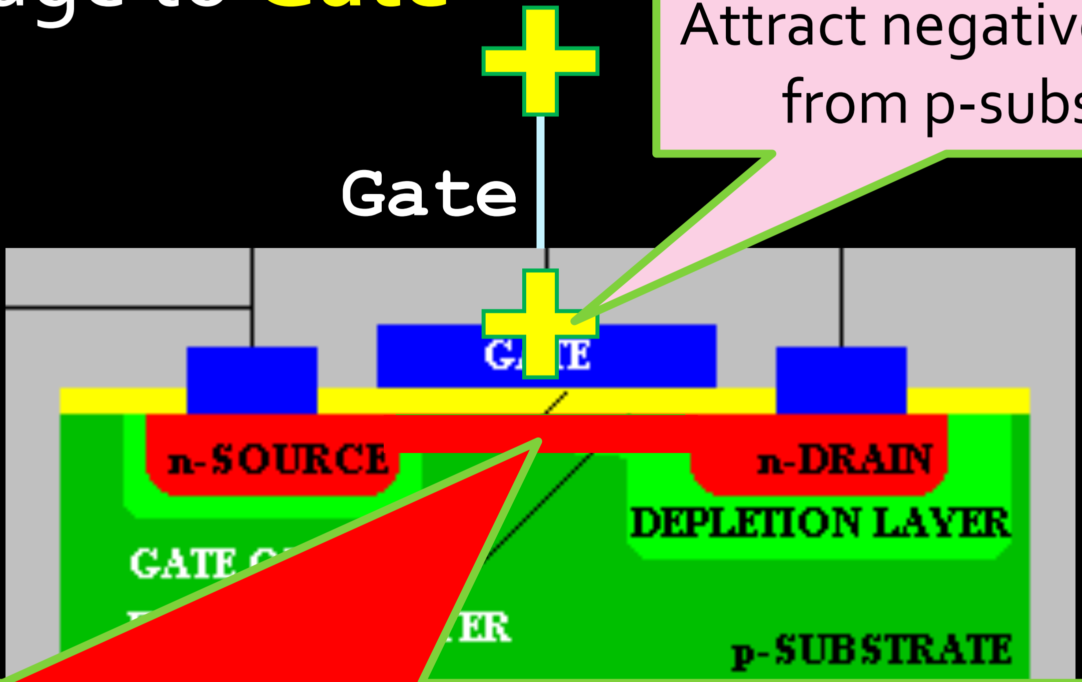
# Put it into a circuit




Two PN-junctions **back-to-back**, i.e, **N-P-N**. So, either source or drain has high voltage, one of the PN-junction must be **reverse biased** (circuit disconnected).

| Metal | Oxide | N-type | P-type |
| --- | --- | --- | --- |

But things change if we apply high voltage to **Gate**



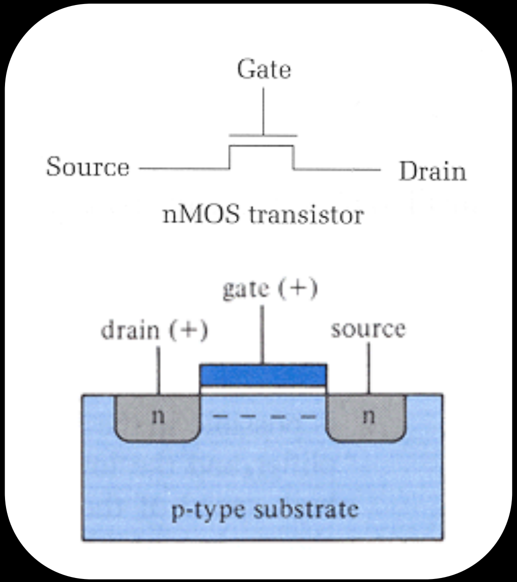

Attract negative charges from p-substrate

**Create n-type channel** between source and drain, **CIRCUIT CONNECTED**

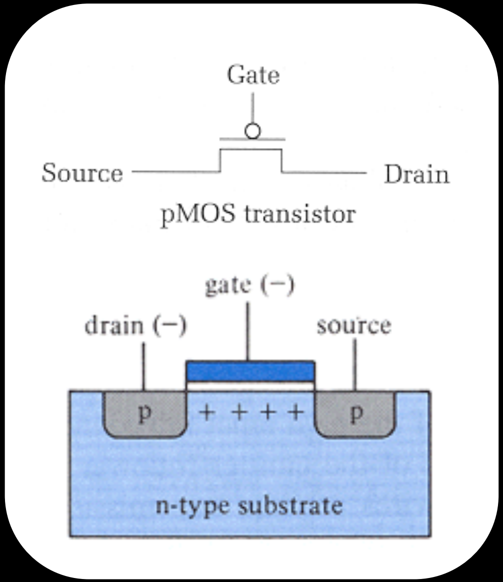
The wider the channel, the higher the current

# Two types of MOSFET

- **nMOS** (what we just describe)
  - N-P-N
  - Gate high, connected
  - Gate low, disconnected




- pMOS (opposite to nMOS)
  - P-N-P
  - Gate low, connected
  - Gate high, disconnected



# 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

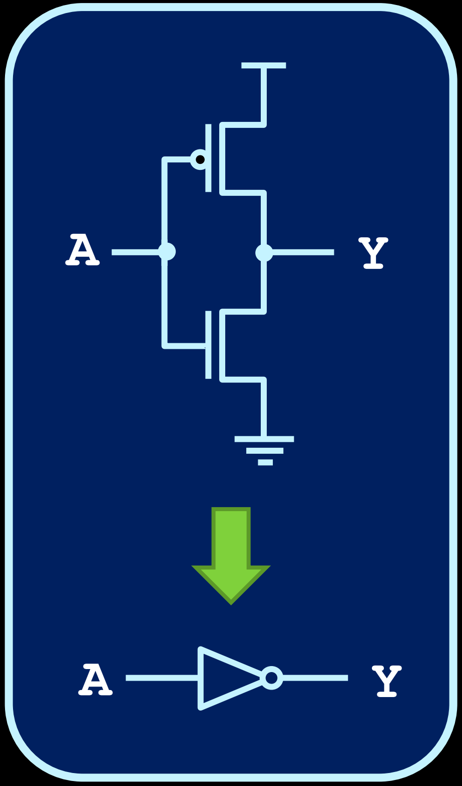
➡ Use transistors build Logic Gates

# Transistors to Logic Gates

# Create gates using a combination of transistors

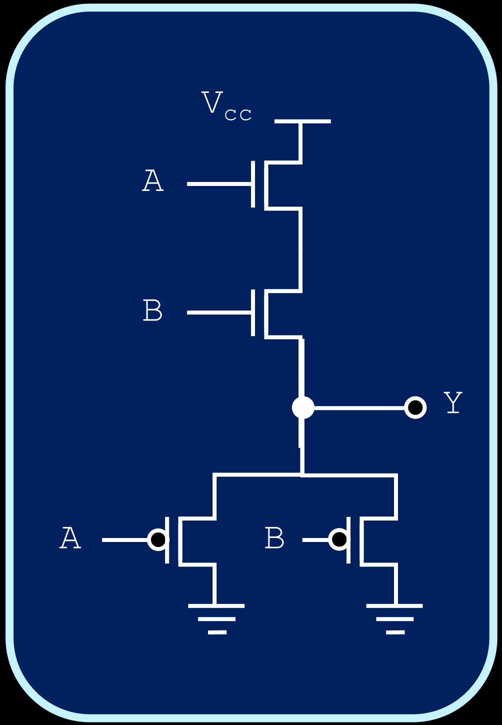
Physical data:

- "High" input = 5V
- "Low" input = 0V
- Switching time: ~20 picoseconds
- Switching interval ~10 ns

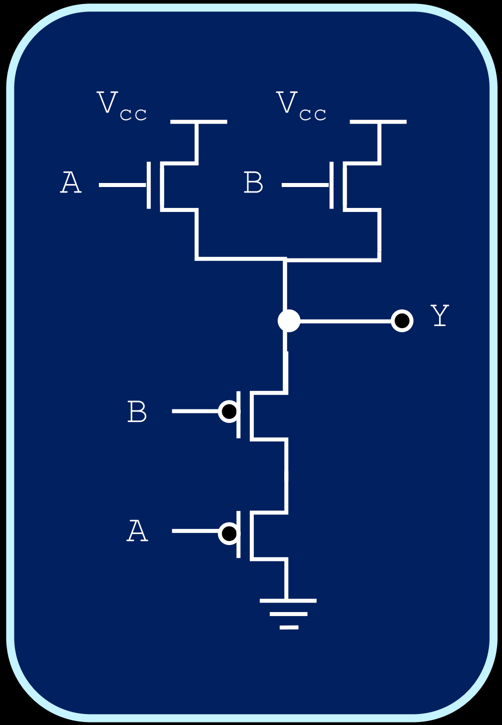



NOT Gate

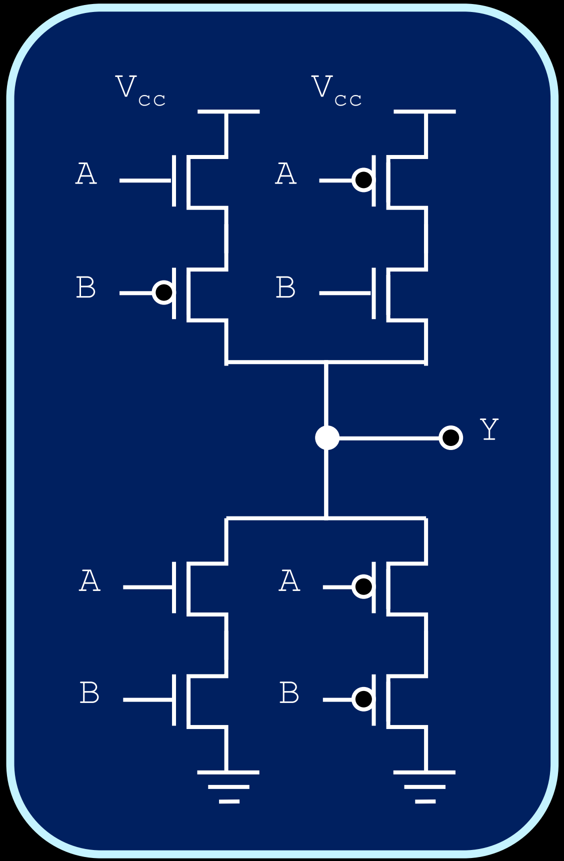
# Transistors into gates



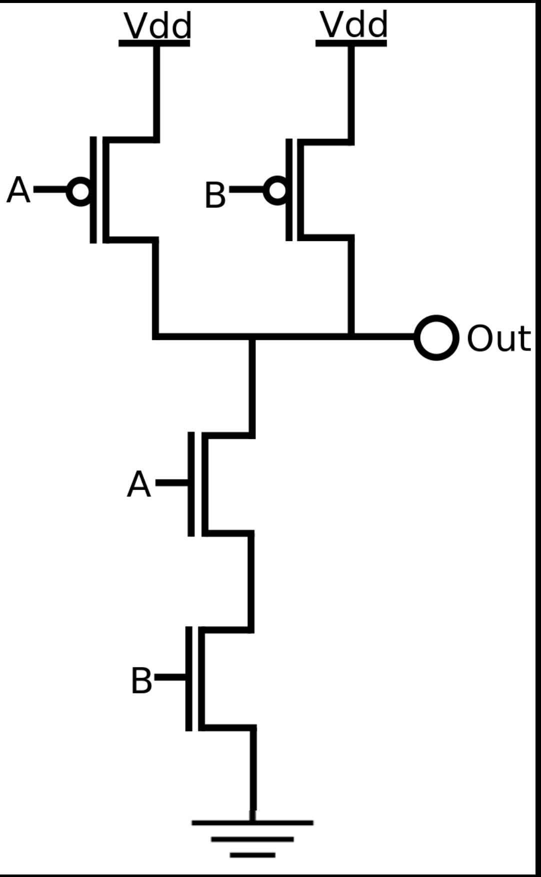

AND




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.

Next week:

- Circuit creation