ECS404U
Computer Systems and Networks

Week 2
Computer Architecture

Week 1

- Recap on week 1: what is a computer?: basic components: how they are put together
    - microprocessors and microcontrollers and where to find them
    - standard components: cpu, short and long-term memory, motherboard, IO
    - the von Neumann architecture and how it relates to modern pc’s
    - quick look at modern laptops, phones, servers

Week 1

- We saw that pc's, laptops, phones and servers have a lot in common

- They also have a lot in common with: wireless routers, televisions, game consoles,…

Week 2: Basic Operation

- Bits and how they are represented
- Computation with bits: Logic circuits and how they are built
    - transistors
    - gates
- Internal comms
    - buses
- How computers operate: a simple dynamics

Part 1: Computation and semiconductor storage

- We'll go through various levels
- Physics of semiconductors (not examinable)
- How transistors work
- How transistors are put together to make gates
- Flip flops and semiconductor storage

Bits
and
how they are represented

Bits

- Computers work in binary

- A single **b**inary dig**it** is either 0 or 1, and is called a bit

Bits

- Any real piece of information will be encoded using lots of bits:
  - single character: 8 bits
  - one pixel on the screen: 24 bits
  - single number: 32 or 64 bits
  - one second of sound: about a million bits.

Bits

- Computers are physical devices.

- So they need to represent bits using actual physical quantities.

Bit: optical drive (CD, DVD)

- A pit in a reflective medium on the drive
- Governs whether a laser is reflected back or not.

Bit: magnetic disk

- Old-style hard disk drive
- Large disks on laptops/servers
- Bit is small magnetised area on disk. Can be magnetised up (1), or down (0).

Bit: optical fibre

- Message consists of flashes of light down the fibre.
- Bit can be light on or off.
- But there are more sophisticated encodings.

Bit: semiconductor

- Examples: cpu or semiconductor memory.
- Bit is voltage (+/- 5v or +/- 3v)
- High represents 1, low represents 0.

Computation

The aim of this part is to see how microprocessors work.

State

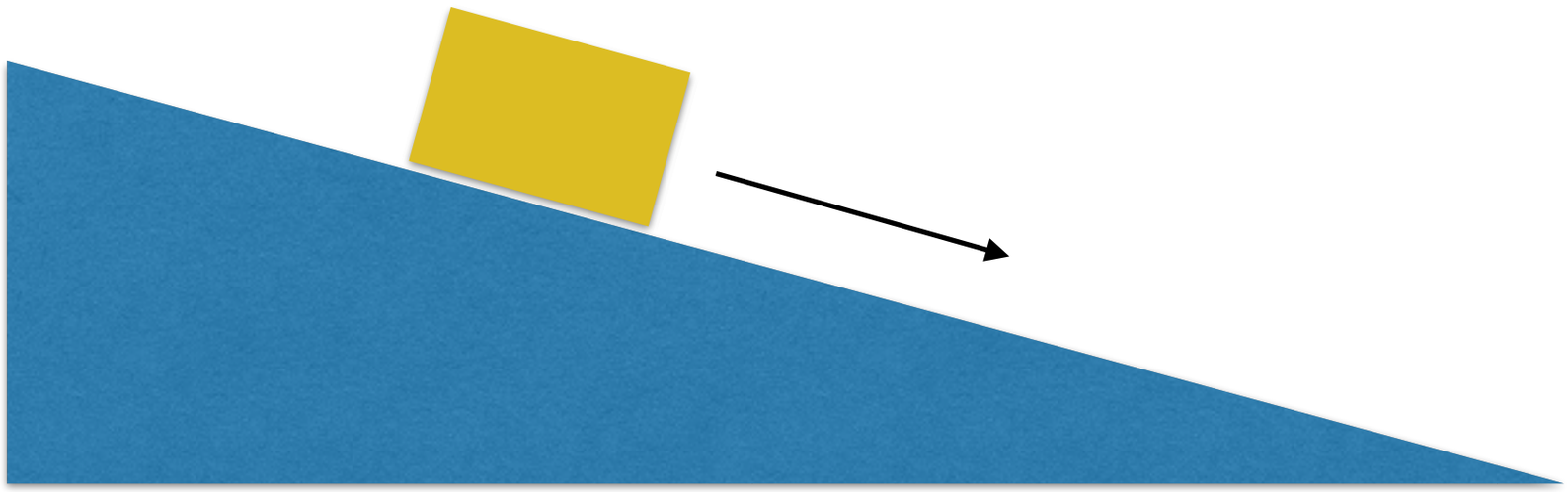
- The **state** of a system is a description of the configuration it is in.

- The state should contain the information you need to know in order to say how the system will evolve.

- The state describes what changes with time.

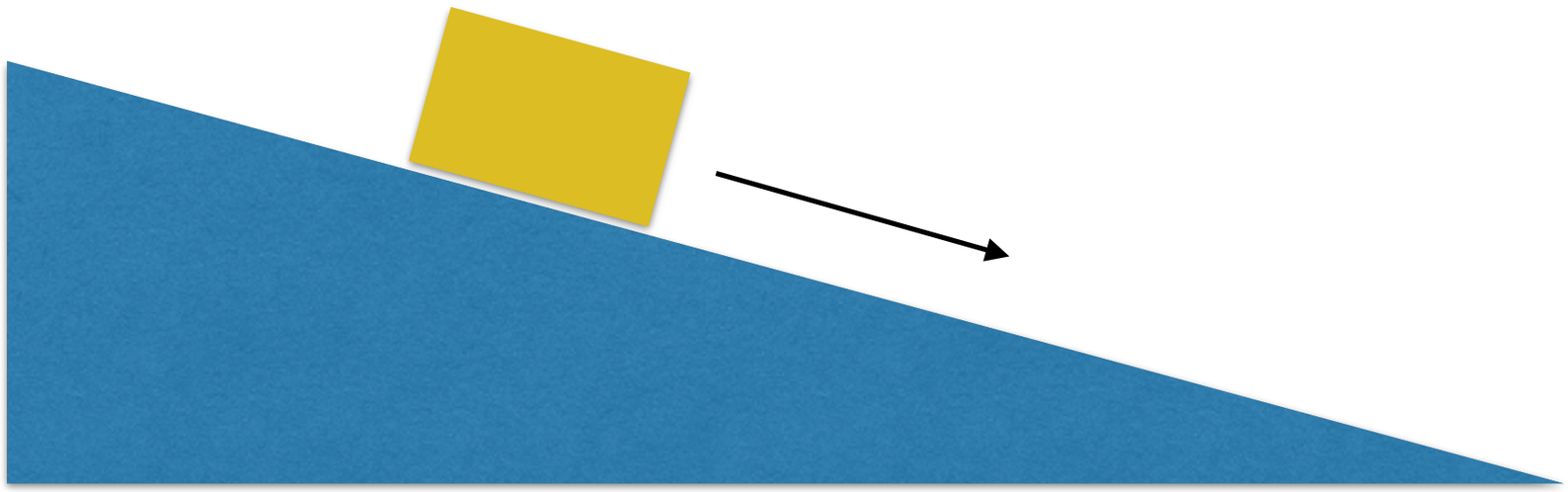
State

- State can be abstract.
- Block sliding down an inclined plane.



State

- State is position of the block and its speed.
- You calculate how these change with time.



(State) Transition Systems

- The abstract systems we work with tend to have discrete transitions.

- They jump from one state to another.

- Example: cpu's are clocked, so we are primarily interested in what they are doing at clock ticks, not between them.

Integrated Circuit State

- The state of an integrated circuit (cpu, memory) is given by knowing the voltages at all of the features.

- In general, voltage is a continuous quantity varying continuously over time.

- For a cpu we abstract this and make it discrete:

  - in time (only at clock ticks)

  - in voltage level (voltages can be high, low or possibly mid/unknown/don’t care)

Computation

- happens through charge moving around the system.
- The system behaves differently with different inputs and different programs. So there has to be some kind of switching mechanism to do this.
- In fact, you can think of the whole cpu as a big switching mechanism, and how you make the switches is critical.
- In semi-conductor based cpu's, the switches are **transistors**, which are built from a combination of **n** and **p semiconductors**.

Transistors and Gates

Transistors and Gates

- What we're going to do now is build up to an account of how transistors work and how they are put together to form devices that do computation and can store data.

- Some of this is going to get bumpy.

- You should pay attention to the way "gates" are built and put together.

- The stuff about semiconductors and transistors is useful background.

Semiconductors

- Computer chips (cpu's, memory, the chips that go into IO devices) are made out of semiconductors.

- The semiconductor is almost always Silicon.

Semiconductors

- Pure semiconductors are not good conductors of electricity (example: silicon).

- So semiconductors are **doped** to make them conductors.

- They can be doped to be either **negative** or **positive** (**n or p semiconductors**).

- An **n-semiconductor** has **negative electrons** that are free to travel (dopant example: arsenic).

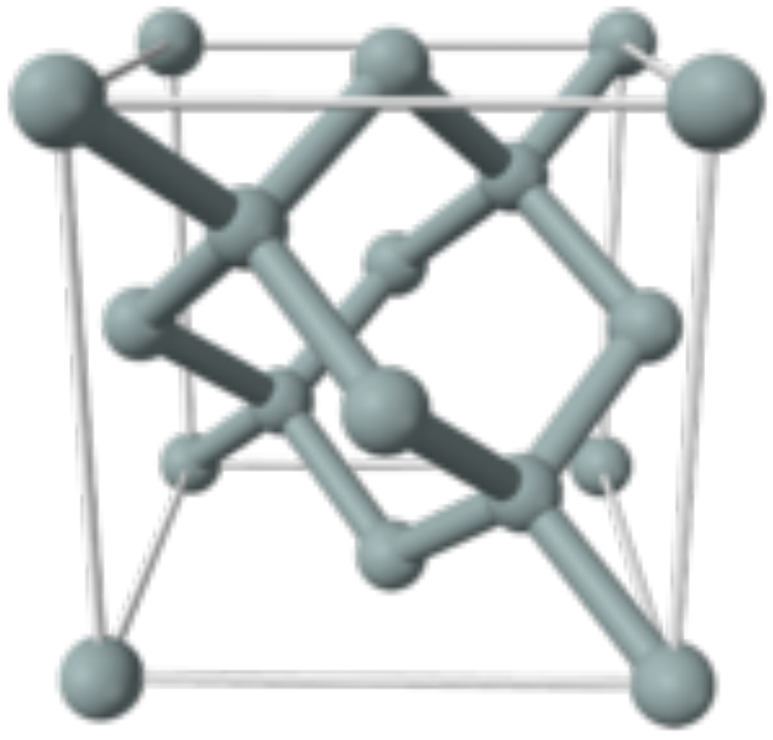
- An **p-semiconductor** has **positive holes** that are free to travel (dopant example: boron).

Electrical Conduction in general and in semiconductors

- Understanding this properly is complicated.
- It takes understanding some quantum physics, electron energy levels, the Pauli exclusion principle, statistical mechanics,…
- So a chunk of a second year physics course.
- We’re not going to do that. This is a simplified picture…

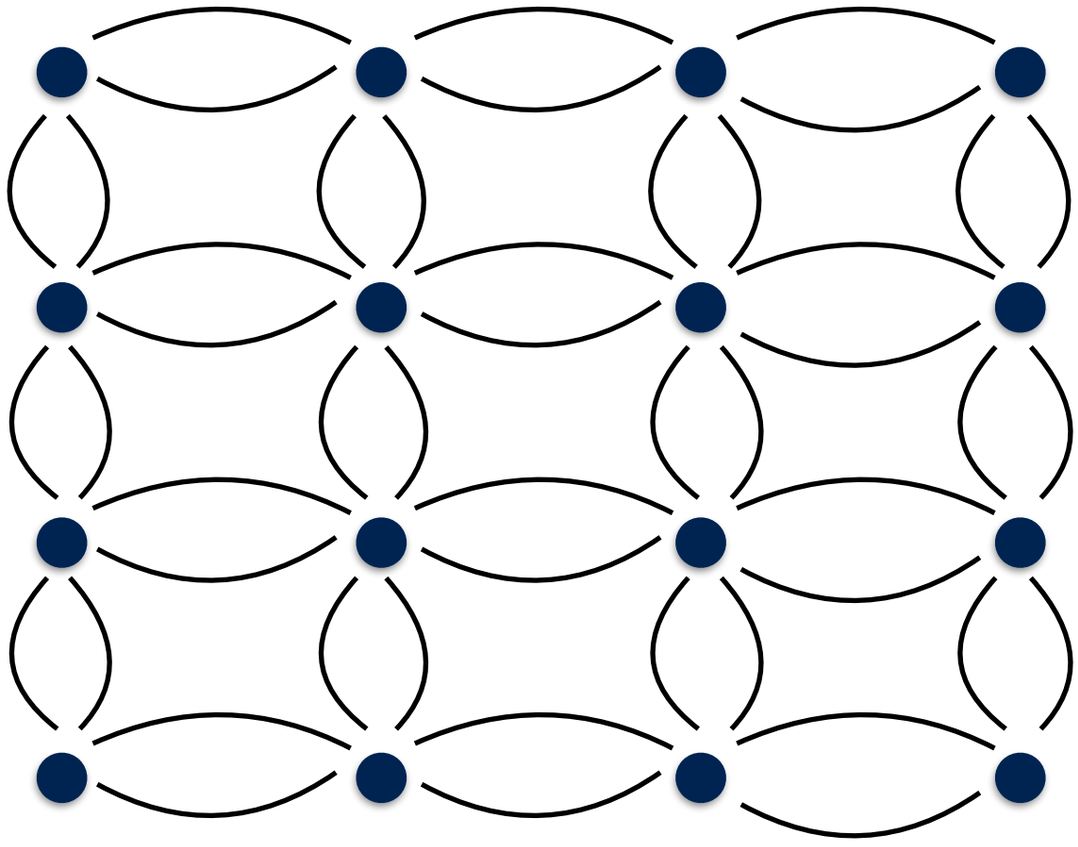
Basic Silicon Structure

- Silicon is a crystal in which each silicon atom is connected to four others (like diamond).



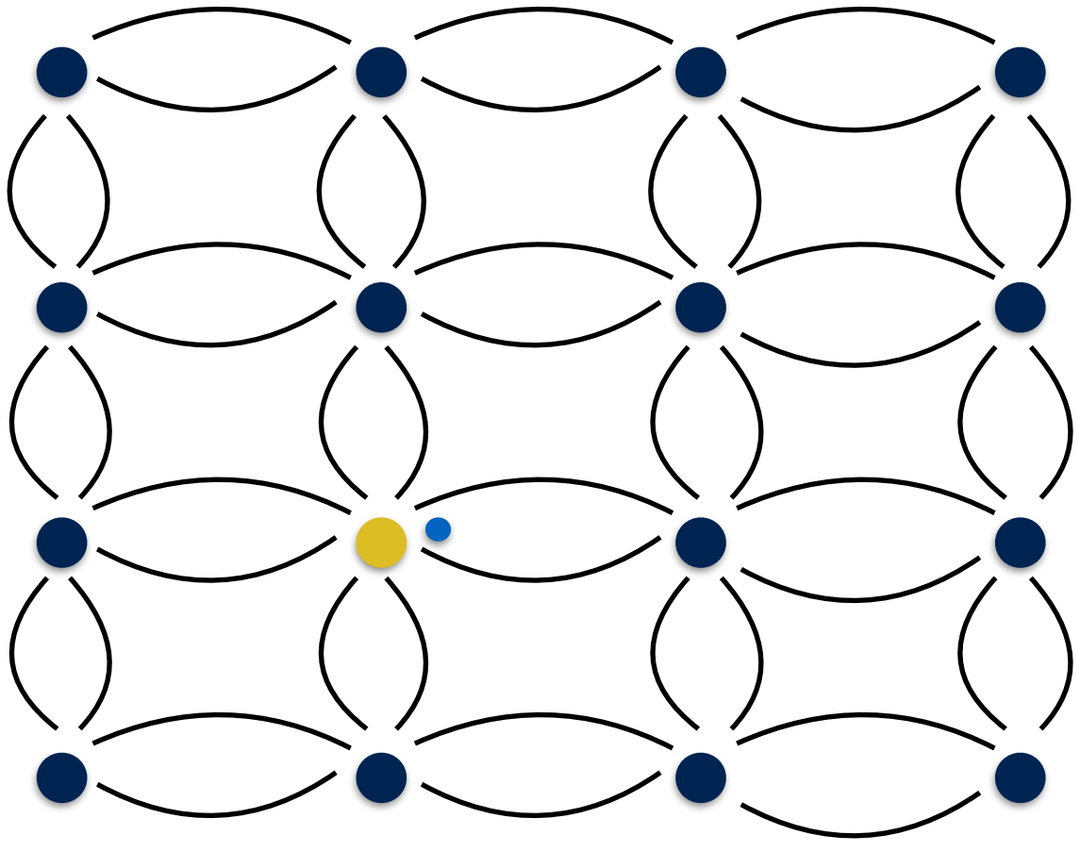
Basic Silicon Structure

- We can flatten this out… violating the structure a bit.



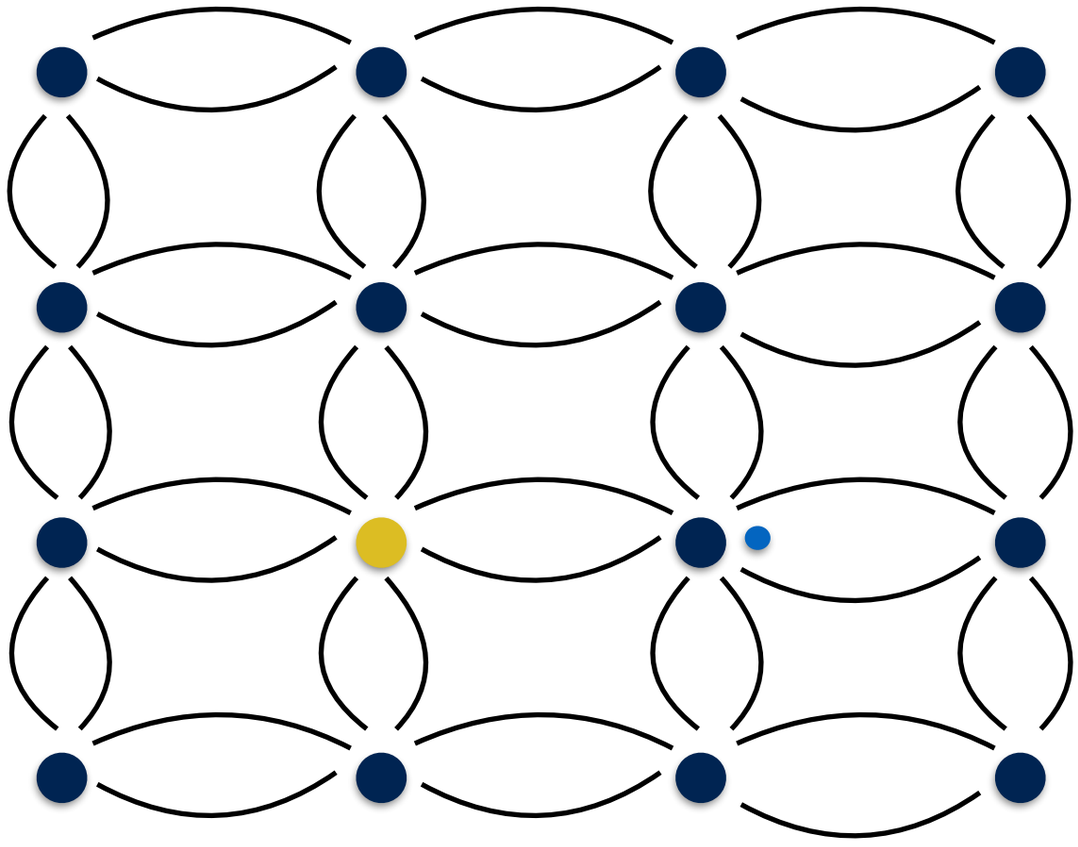
n-semiconductor structure

- If we replace a Silicon atom with an Arsenic atom, then that Arsenic atom has an extra electron in its outer shell.



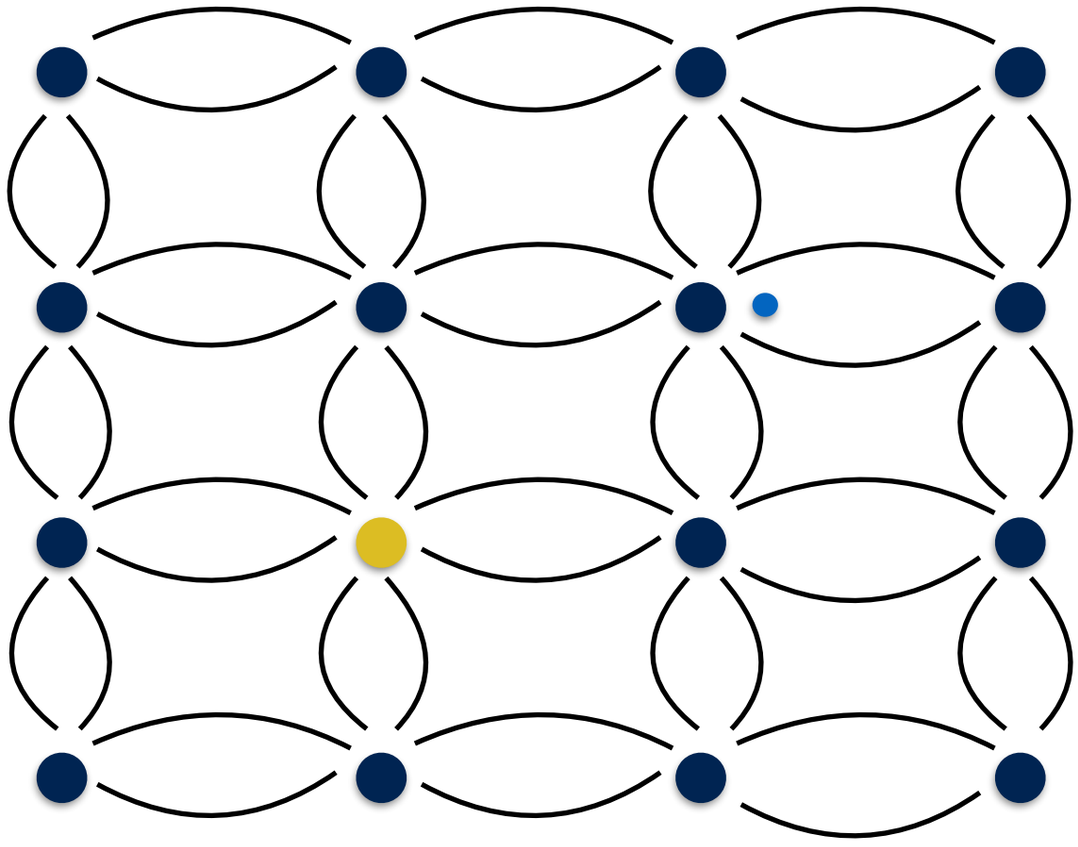
n-semiconductors

- But that means the shell is overfilled, and the electron can wander through the structure.



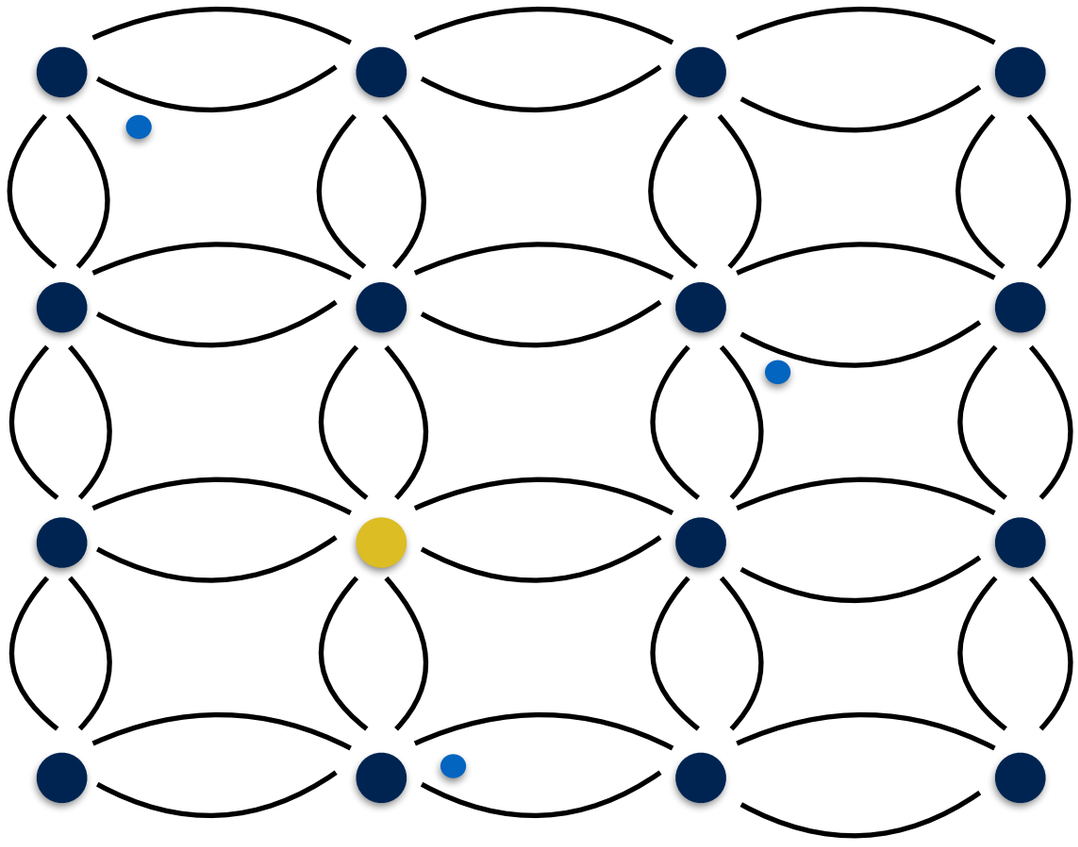
n-semiconductors

- This allows the semiconductor to conduct electricity (electric current is moving charge)



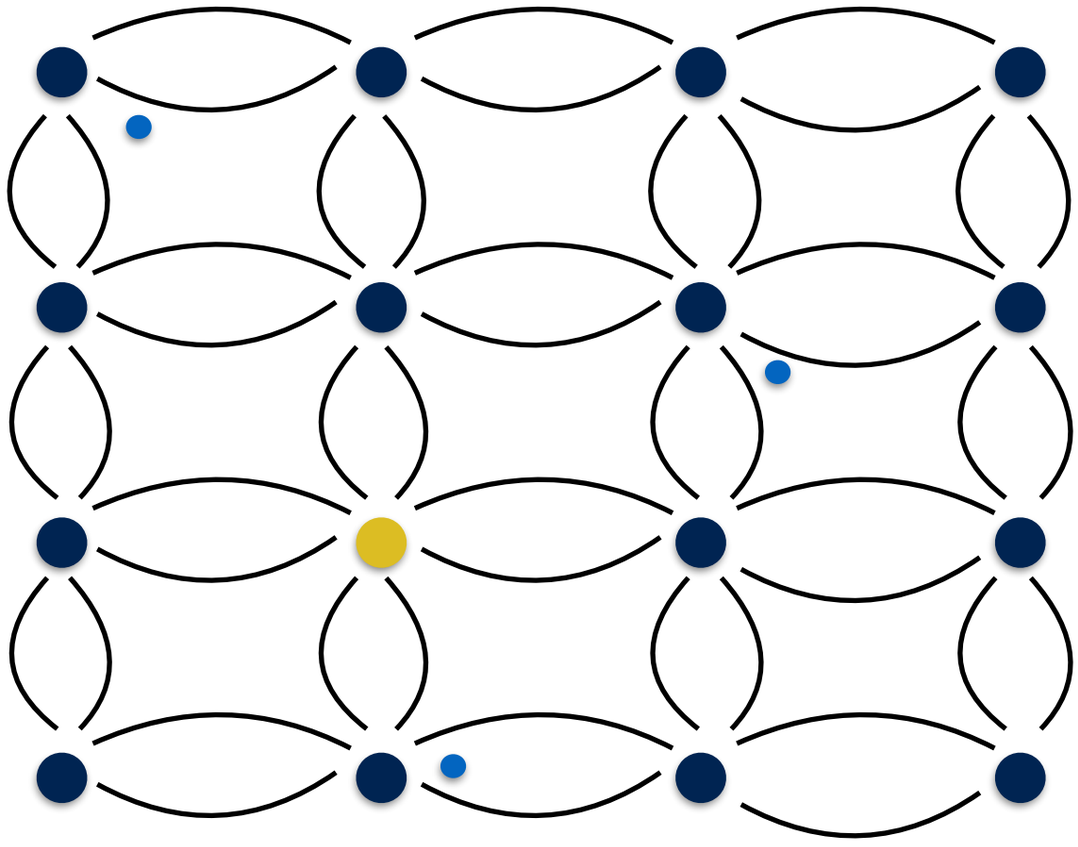
n-semiconductors

- The physics is complicated, but the upshot is that these electrons behave as an ideal gas.



n-semiconductors

- The semiconductor is filled with a cloud of electrons, and that cloud behaves like a gas.



n-semiconductors

- Semiconductor (Silicon) doped with a substance that has extra electrons in the outer shell (Arsenic)

- Substance is electrically neutral, but

- Extra electrons can move through crystal structure

- Collectively behaving as a (charged) gas cloud inside the solid semiconductor.

p-semiconductors

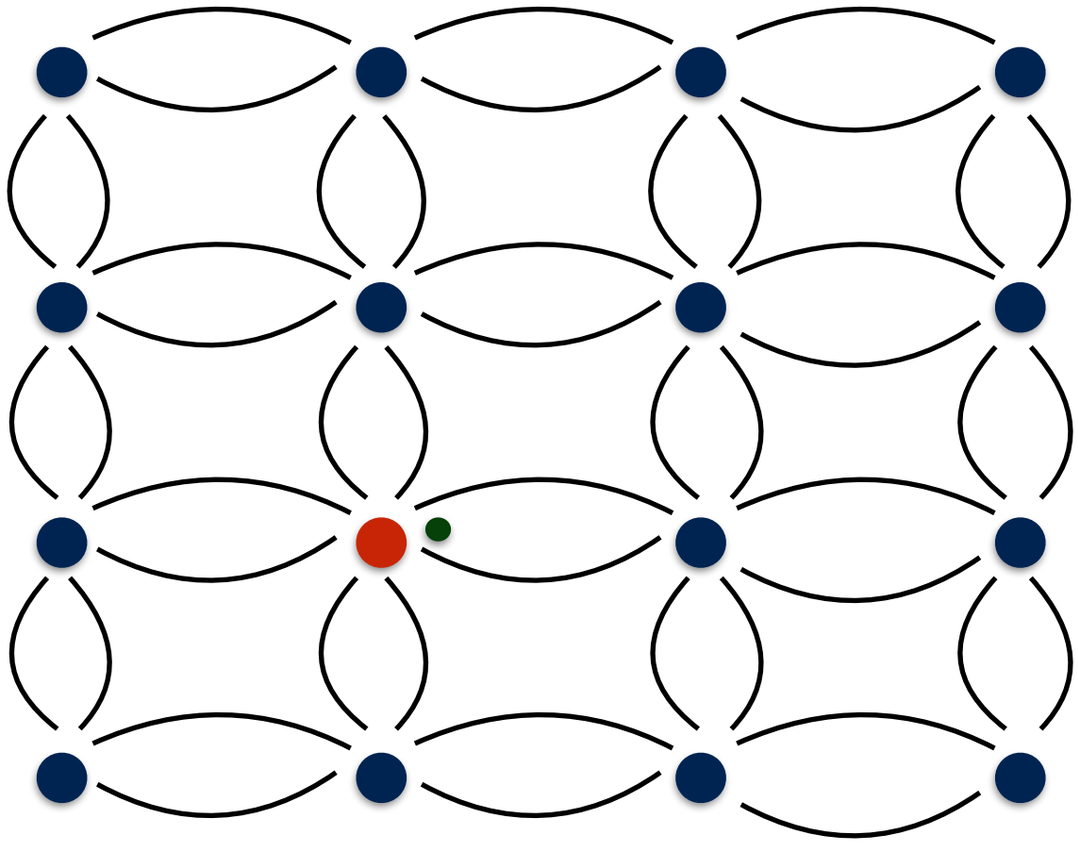
- Semiconductor (Silicon) doped with a substance that has fewer electrons in the outer shell (Boron)

- Instead of there being an extra electron, there is an electron-shaped “hole” where the crystal structure expects there to be an electron.

- This hole can drag in an electron from one of the neighbours, creating a hole elsewhere.

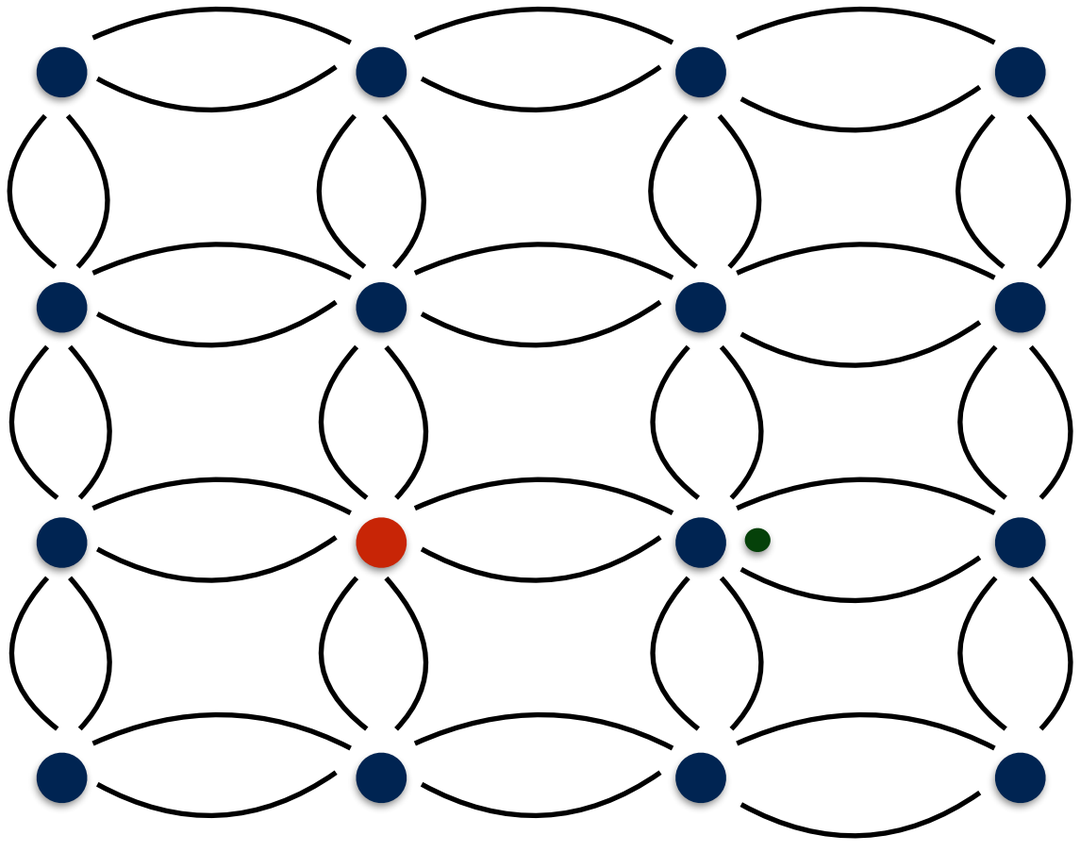
p-semiconductors

- The Boron atom has an electron missing from its outer shell. This makes a “hole”.



p-semiconductors

- The hole can drag in electrons from nearby, effectively moving the hole.



p-semiconductors

- There is a symmetry between **holes** and **electrons**.
- Electrons are real particles. Holes are not, but they behave as if they were.
- The mathematics of their behaviour is the same, except that holes are positively charged.

p-semiconductors

- Semiconductor (Silicon) doped with a substance that has fewer electrons in the outer shell (Boron)

- Substance is electrically neutral, but
  - holes can move through crystal structure
  - collectively behaving as a (positively charged) gas cloud.

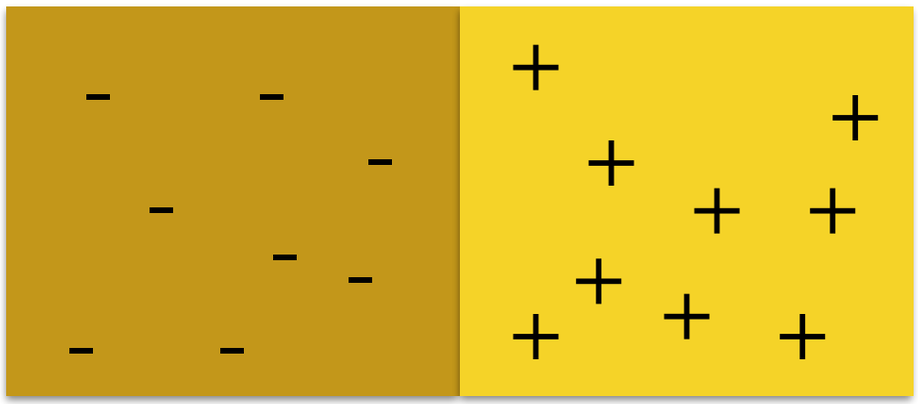
semiconductors

- The electron-hole account is the one you usually find in the literature.

- Here is an alternative:
  - n-semiconductor is filled with an electron gas at positive pressure
  - p-semiconductor is filled with a hole gas at positive pressure, which equates with an electron gas a negative pressure.

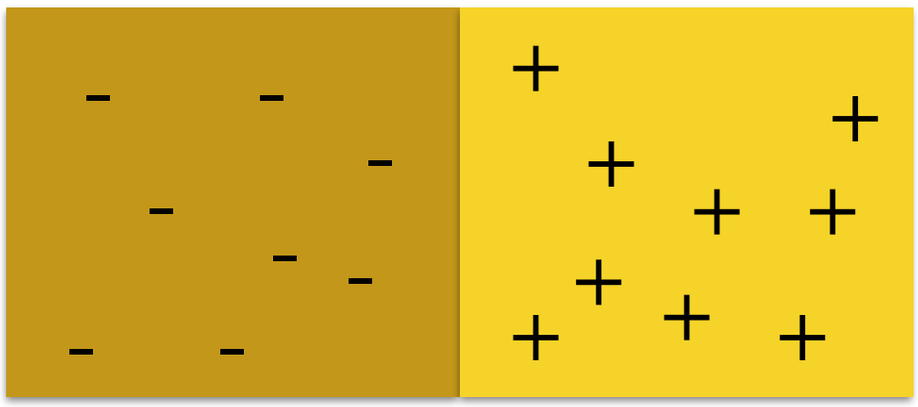
Diodes

- Interesting things start to happen when you put a p-semiconductor next to an n-semiconductor.



Diodes

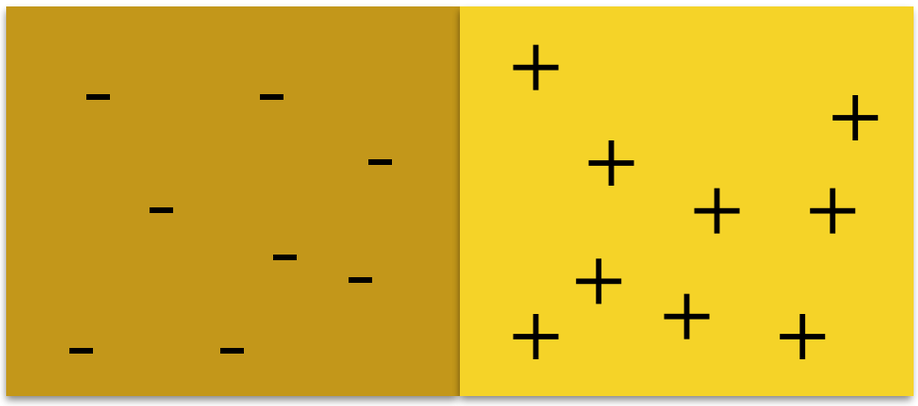
- It's easy to get electron flow from left to right (n to p).
- Connect a low (ie negative) potential at the left, and it pushes the electrons across the border and to the right .



- Think of this as saying its easy to get electrons to move from an area of high electron pressure to an area of low electron pressure.

Diodes

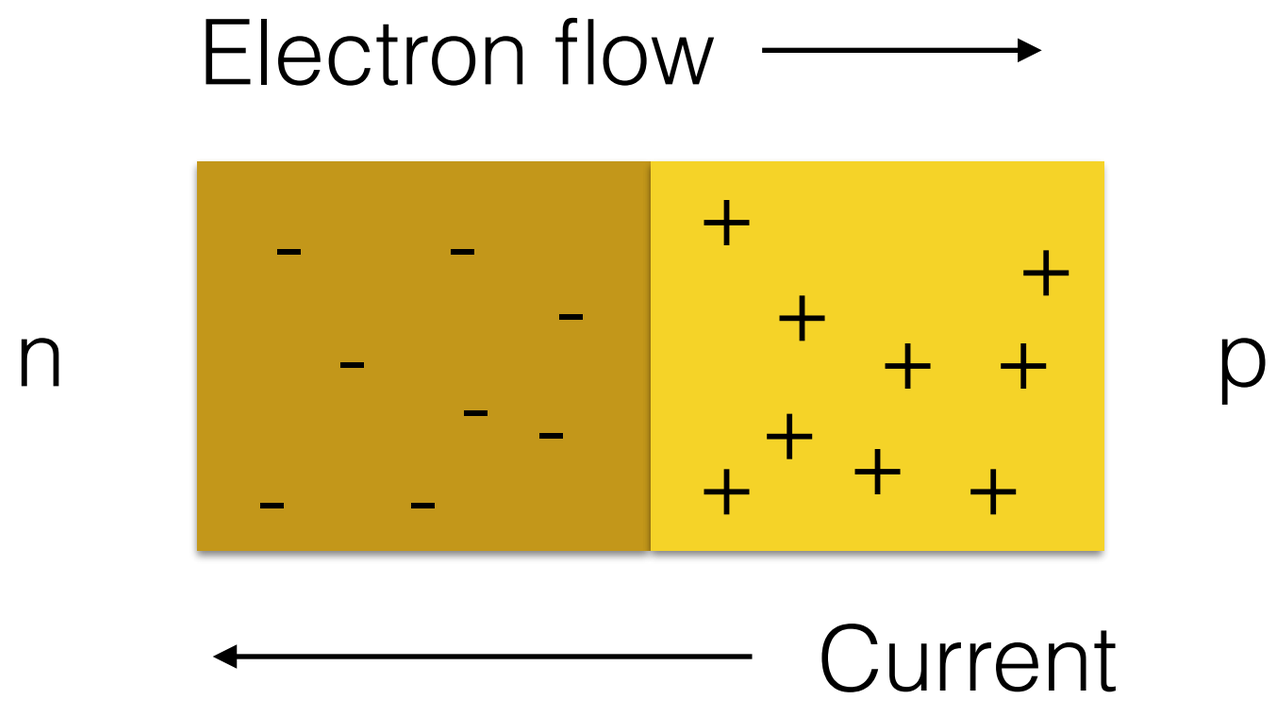
- But it's hard to get electron flow from right to left (p to n).
- Connect a low (ie negative) potential at the right, and it tends to drag the holes toward it, but does not do much at the boundary.



- Think of this as saying it's hard to get electrons to move from an area of low electron pressure to an area of high electron pressure.

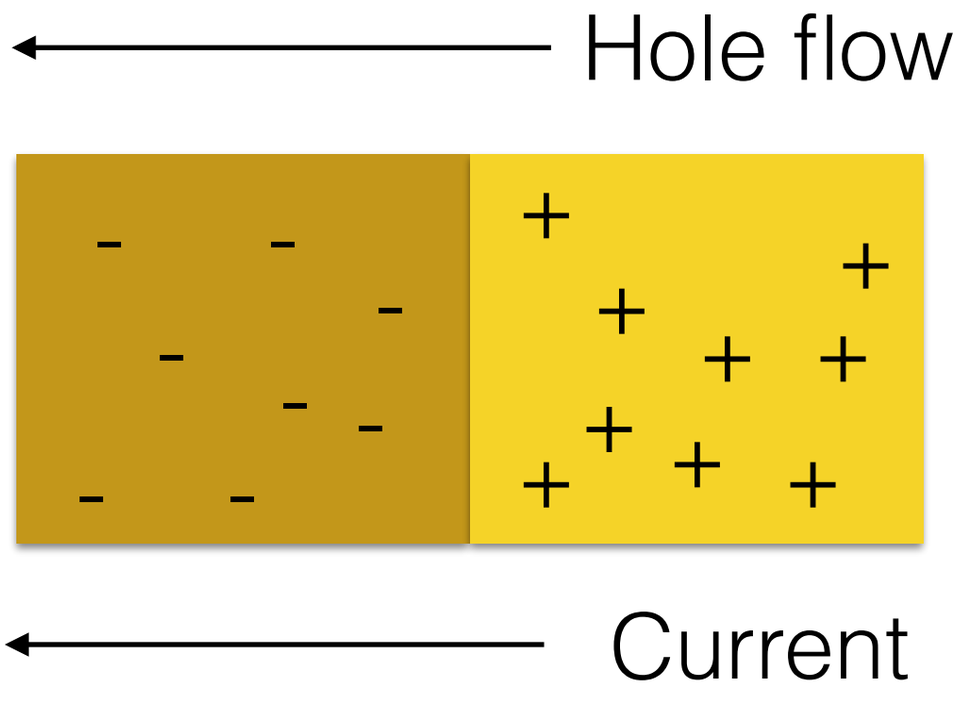
Diodes

- Electrons flow easily n to p but not p to n.
- This means it's easy to get a current from p to n, hard to get one from n to p.



Diodes

- Symmetrically, holes flow easily p to n but not n to p.
- This again means it's easy to get a current from p to n, hard to get one from n to p.



Transistors

- Transistors are the electrical components we're aiming at.

- We'll just discuss them as switches, but they can also be used as amplifiers.

- There are several types, but the particular type we will look at is a "Field Effect Transistor"

- This is the type used in chips.

- But the physical layout is now different.

mosfet

- Transistors on chips are described as **MOSFET**
- **MOS** = **metal oxide semiconductor**
- **FET** = **field effect transistor**
- Standard chip technology is **CMOS** = **complementary MOS**
- This has to do with the symmetric way that circuits are implemented on the chips.

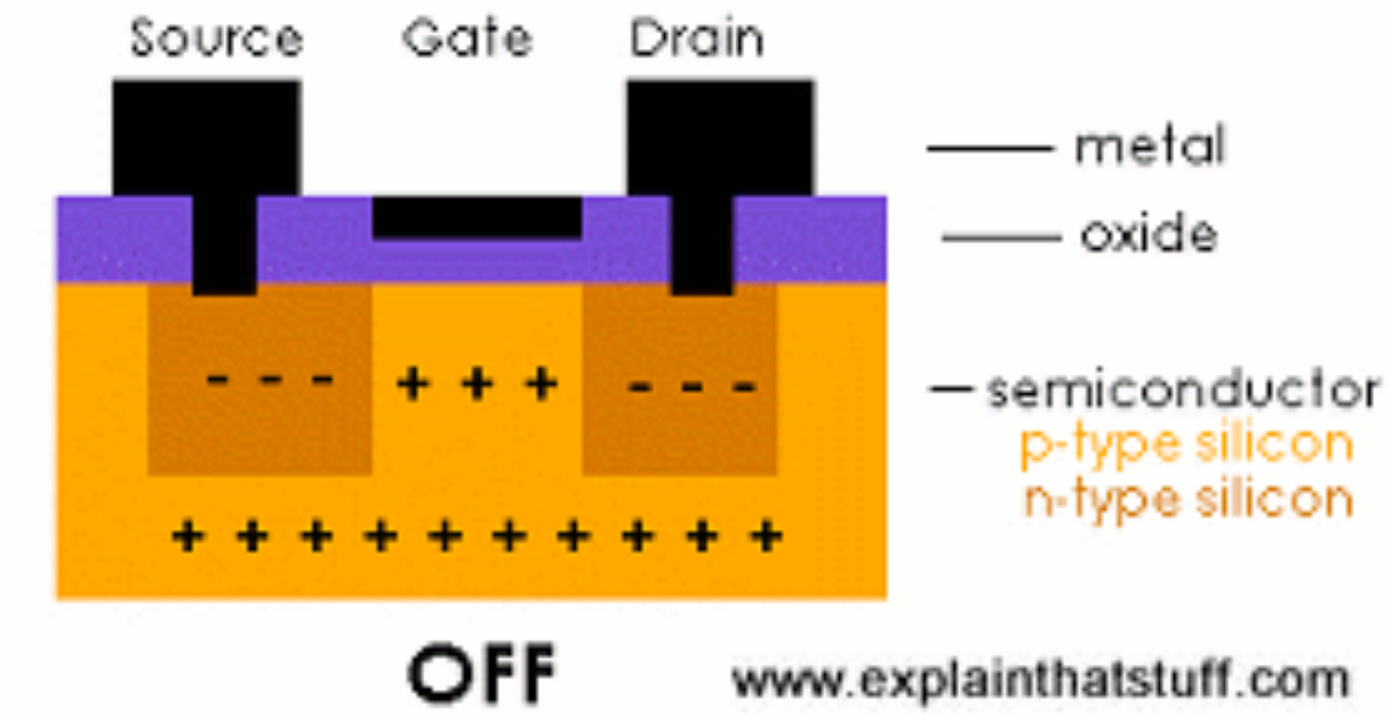
Transistors

- have three connections:
  - **gate**
  - **source**
  - **drain**
- use the gate to switch the connection between source and drain on and off

Transistors

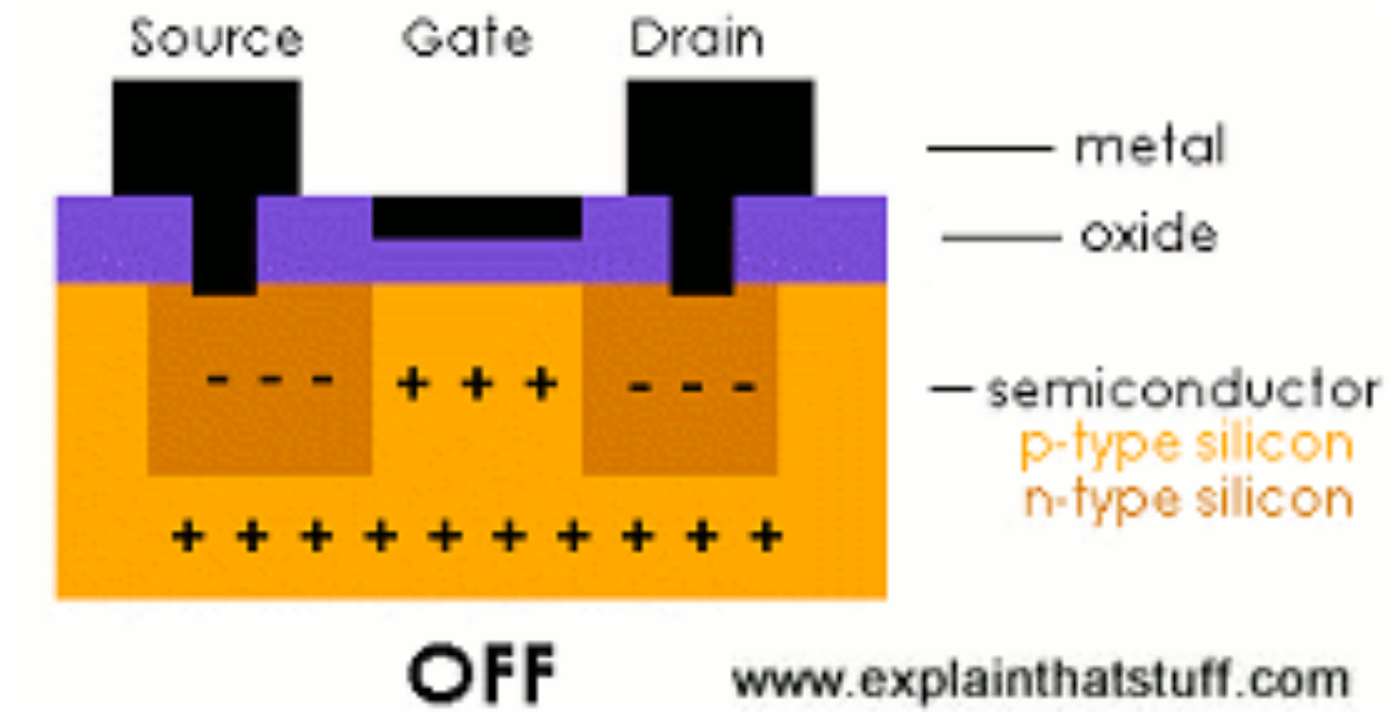
- have three connections:
  - **gate** - controls on and off
  - **source** - potential in
  - **drain** - potential out
- use the gate to switch the connection between source and drain on and off

Field Effect Transistor (npn type)



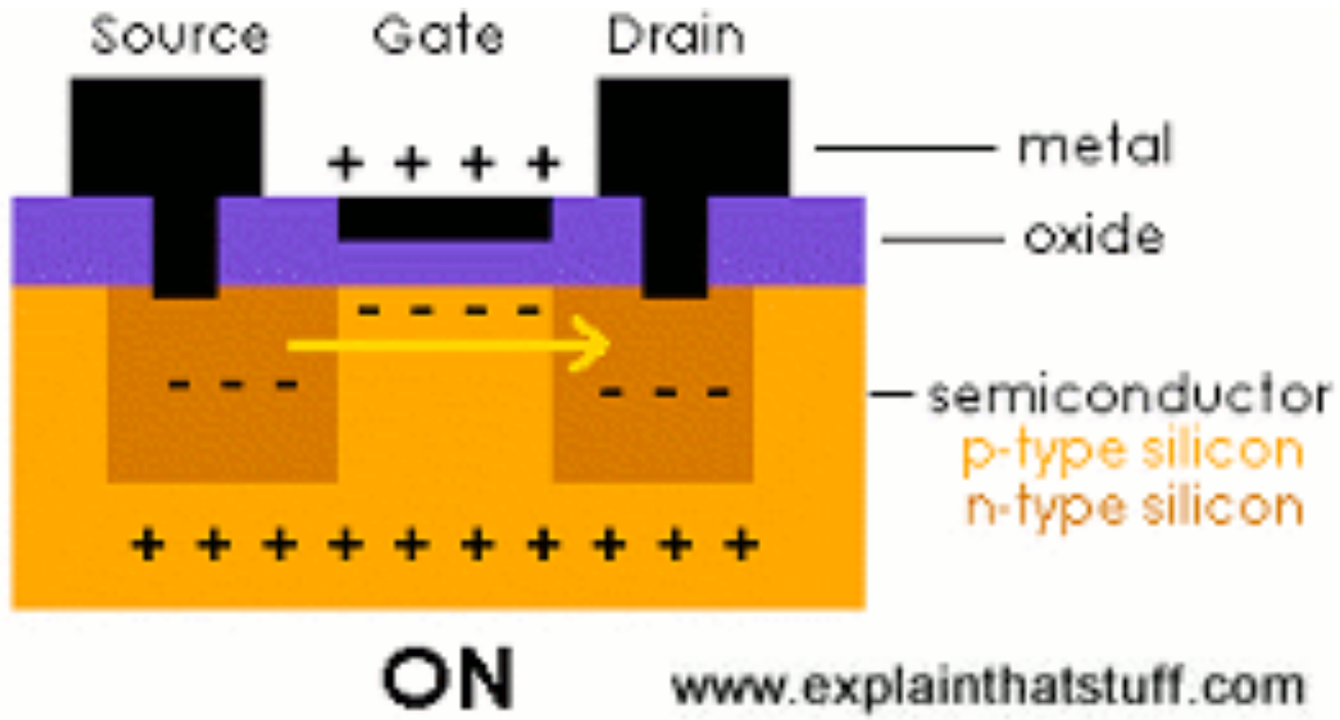
Oxide is an insulator: charge at the gate goes nowhere.

Field Effect Transistor (npn type)



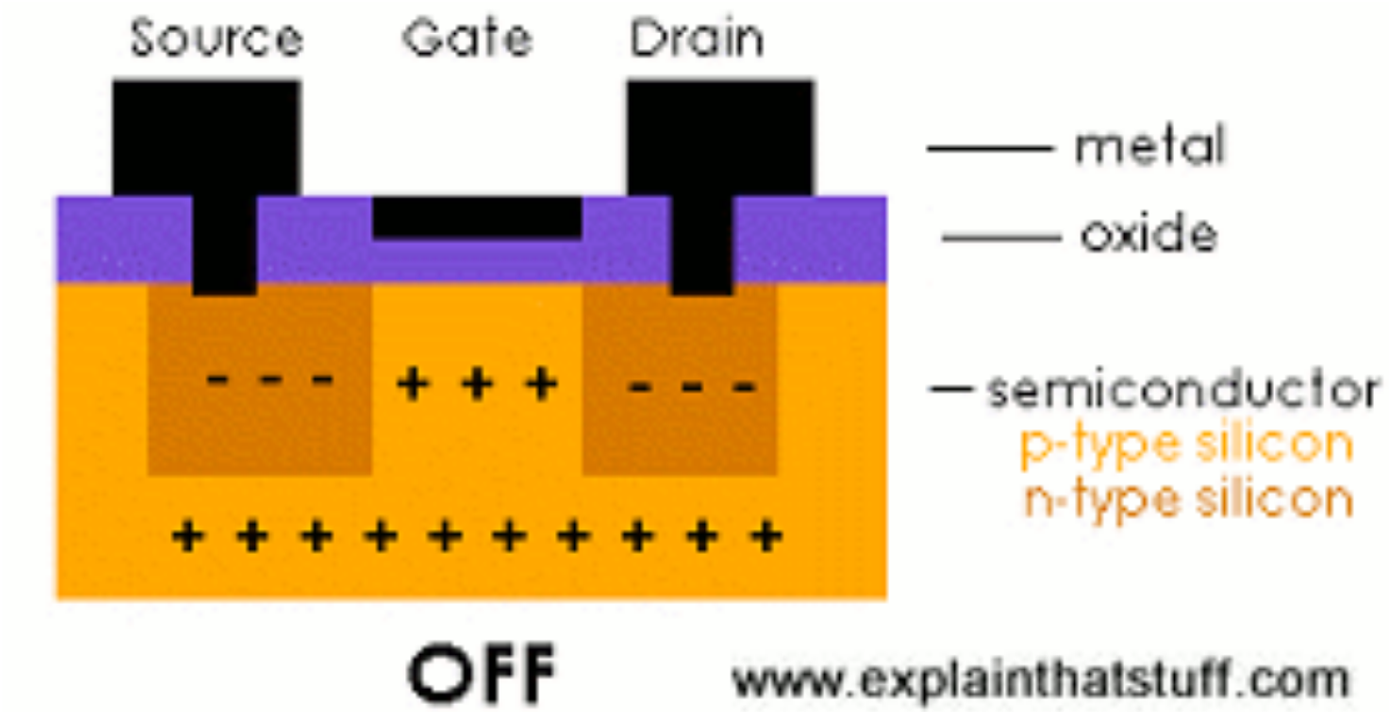
Gate is neutral or negative.
Transistor is off.
Current from source to drain would have to pass through an n to p boundary.

npn Field Effect Transistor



Gate is positive.
Electrostatic charge pushes holes away, and creates a s
corridor of electrons near the gate.
Current passes along the electron corridor.
Transistor is on.

Field Effect Transistor (pnp type): not pictured



Not pictured: if you exchange the p and n semiconductors then you get another transistor

This switches on when the gate is negative.

On a computer chip..

- These things are tiny (40nm = $40*10^{-9}$m)
- There are billions of them
- They now use a different geometry
- Here is what Intel has to say…

Video Animation: Mark Bohr Gets Small: 22 nm Explained

Tags: Architecture & Silicon | 22nm Process Technology

Rel

solut

with

http://www.intel.com/content/www/us/en/silicon-innovations
standards-22nm-explained-video.html

Talking Points

- Moore's Law (come back to that next week)
- Key point of the talk is that they're introducing a different transistor design (different shape for the semiconductors).
- Advantages: area, power, switching speed

Area

Switching speed

Power

Talking Points

- Size: Bohr shrinks himself by a factor of 20,000 twice.
- At the end of the day, what factor has he shrunk by?
- Note that after the first stage he is on the same scale as the diameter of a human hair. He then shrinks himself again.

Brief interlude: scales

- Suppose your phone uses the Intel 22nm technology, and we blow you and your phone up in size so that each feature on your chip is now a normal size for us: say 1cm. What size would you be?

1. About the size of this lecture theatre?

2. About the size of London?

3. About the size of the UK?

4. About the size of the planet?

Brief interlude: scales

- That's physical scale, what about the timescale?
- Switching speed relates to the clock speed on the chip.
- Clock speeds are about 2GHz.
- This means the chip's little heart beats 2 billion times a second.

Brief interlude: scales

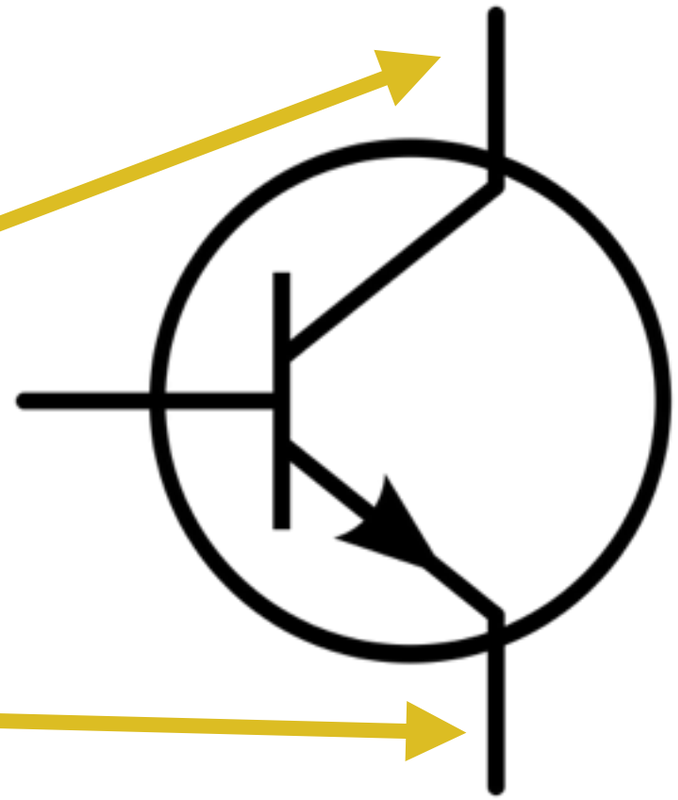
- How long does it take your little heart to beat 2 billion times:

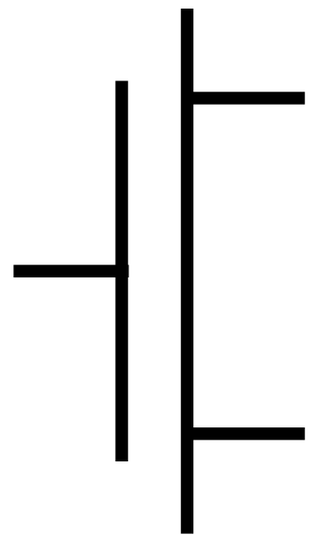
1. a few weeks
2. half a lifetime
3. thousands of years
4. the entire history of the planet

End of interlude

Transistors

- three connections:
  - gate
  - source
  - drain





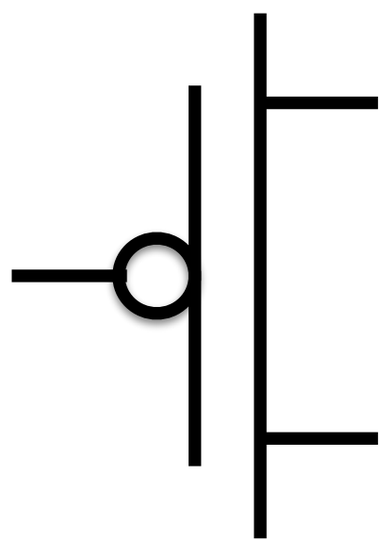
npn transistor stable configurations

Gate	Source	Drain
+	+	+
+	-	-
-	Any	Any

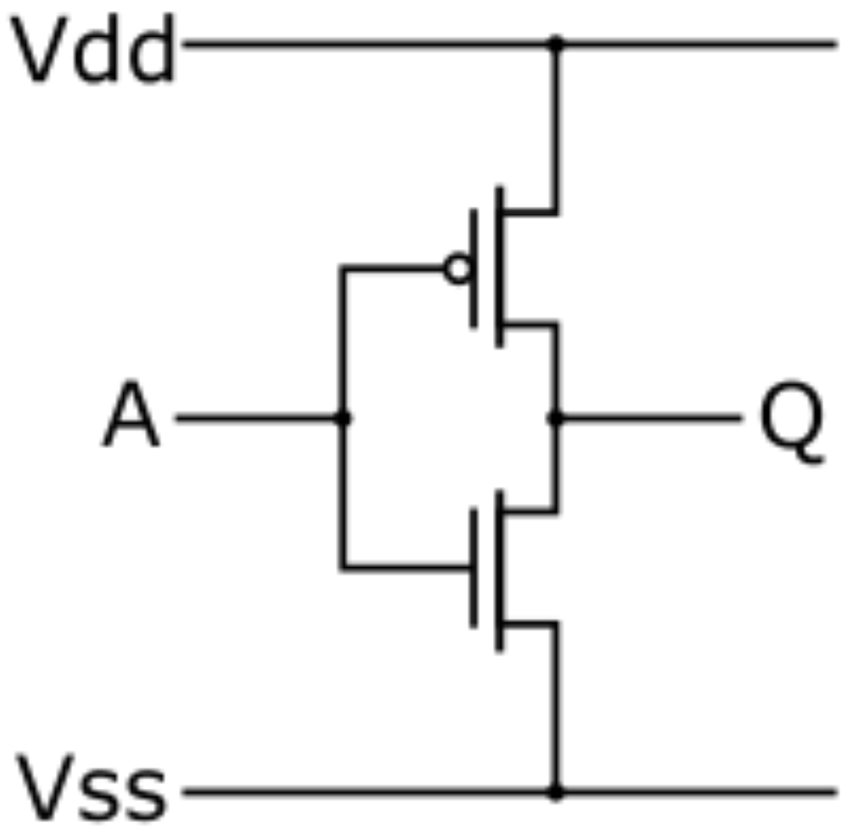
pnp transistor

- Reverse n and p areas, and we get a pnp transistor.

- Switches on if gate is n. (0)

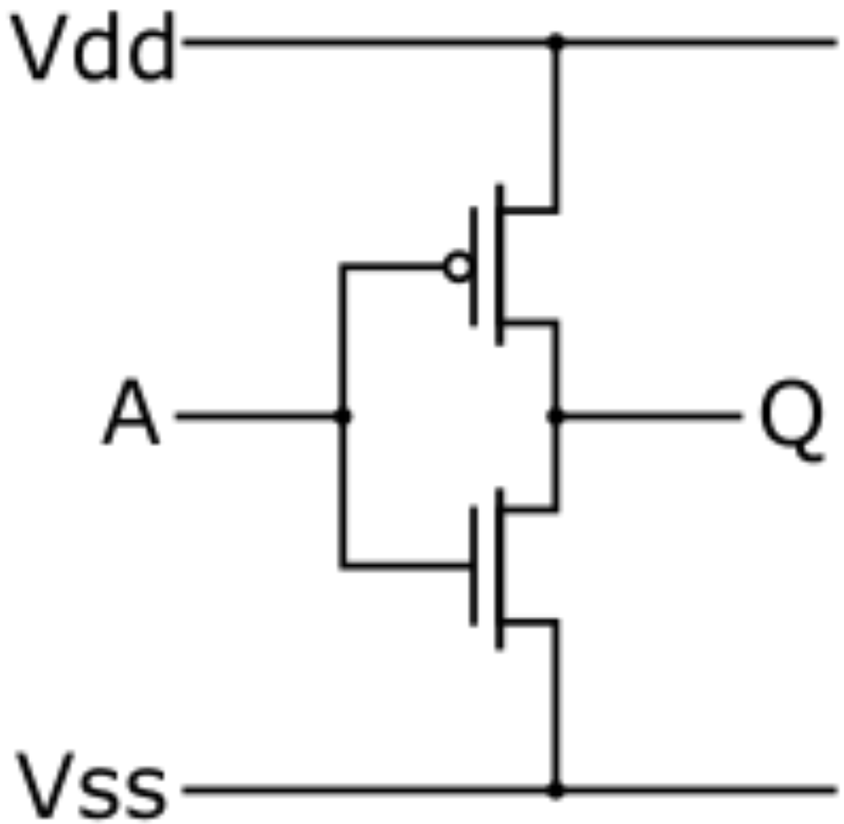


not



not

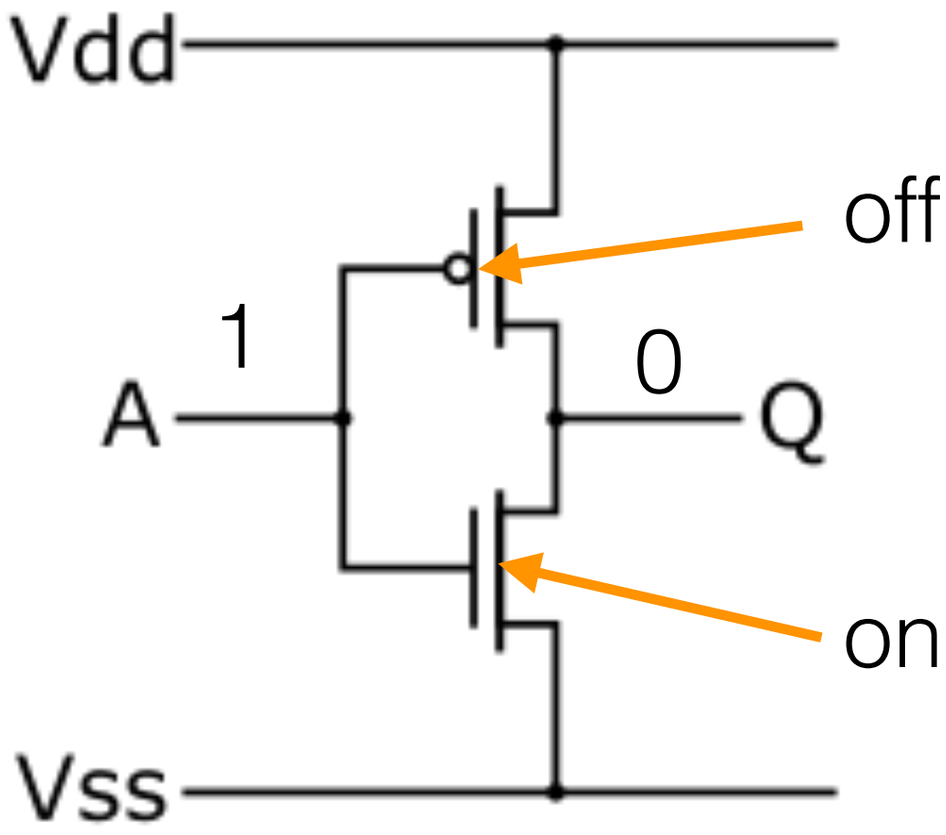
Constant High: 1



Constant Low: 0

not

Constant High: 1

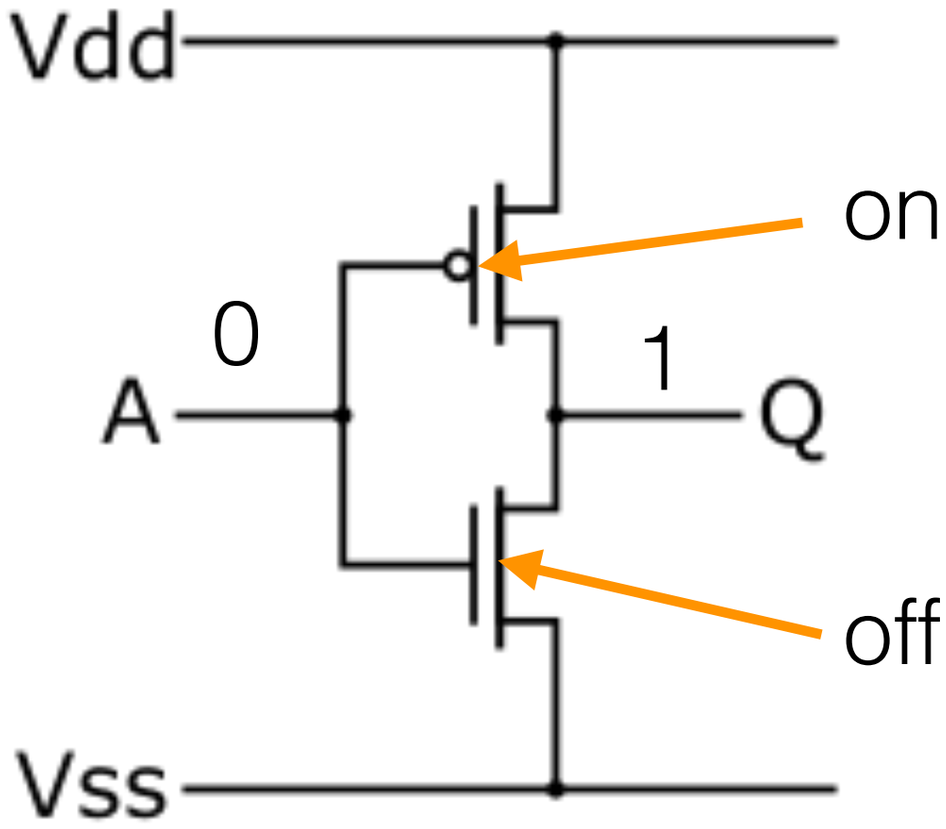


Constant Low: 0

If A=1 then Q is disconnected from Vdd=1,
but connected to Vss=0, so Q=0

not

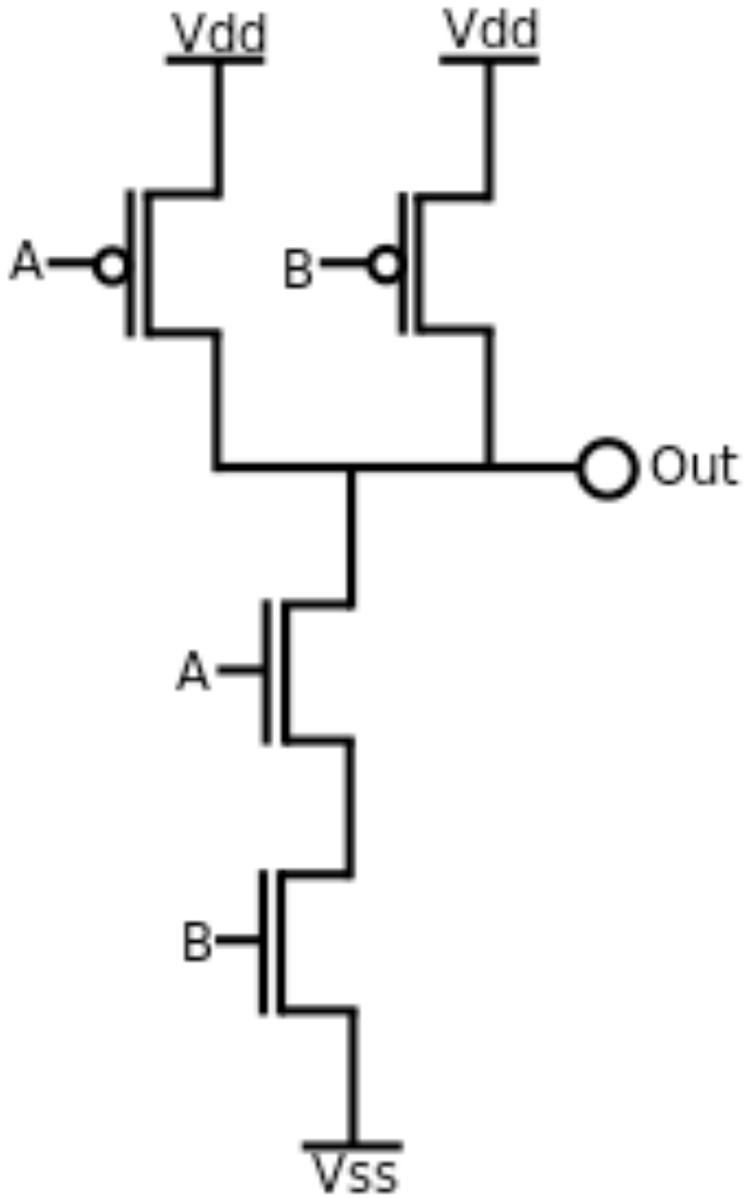
Constant High: 1



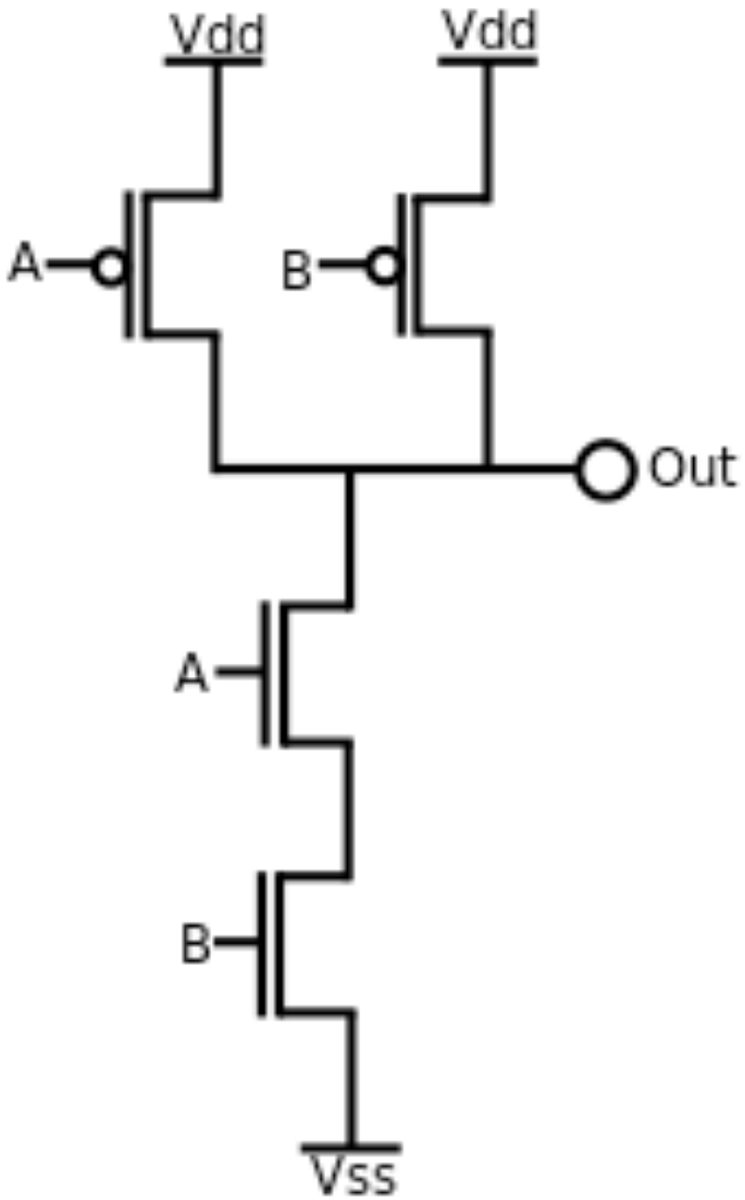
Constant Low: 0

If A=1 then Q is disconnected from Vss=0,
but connected to Vdd=1, so Q=1

nand (not and)



nand (not and)



Analyse as before
to show that
behaviour is given
by following table of
stable configurations

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

nand (not and)

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

The value of Out depends **functionally** on the values of A and B, so we can write this as a truth table.

nand	A=0	A=1
B=0	1	1
B=1	1	0

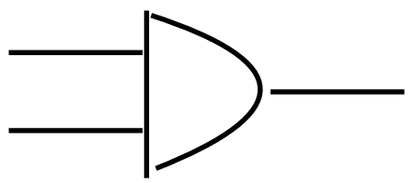
Boolean logic

- Small circuits like these that compute simple boolean functions are called **gates.**

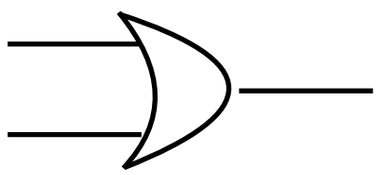
- Gates are named after the function they compute: so the last slide showed a **nand gate**.

- Gates can be plugged together to compute any given **boolean function**.

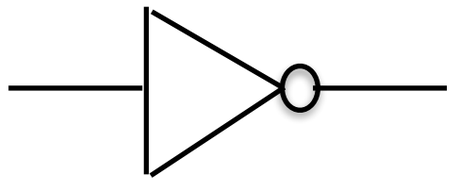
Symbols used



And gate



Or gate



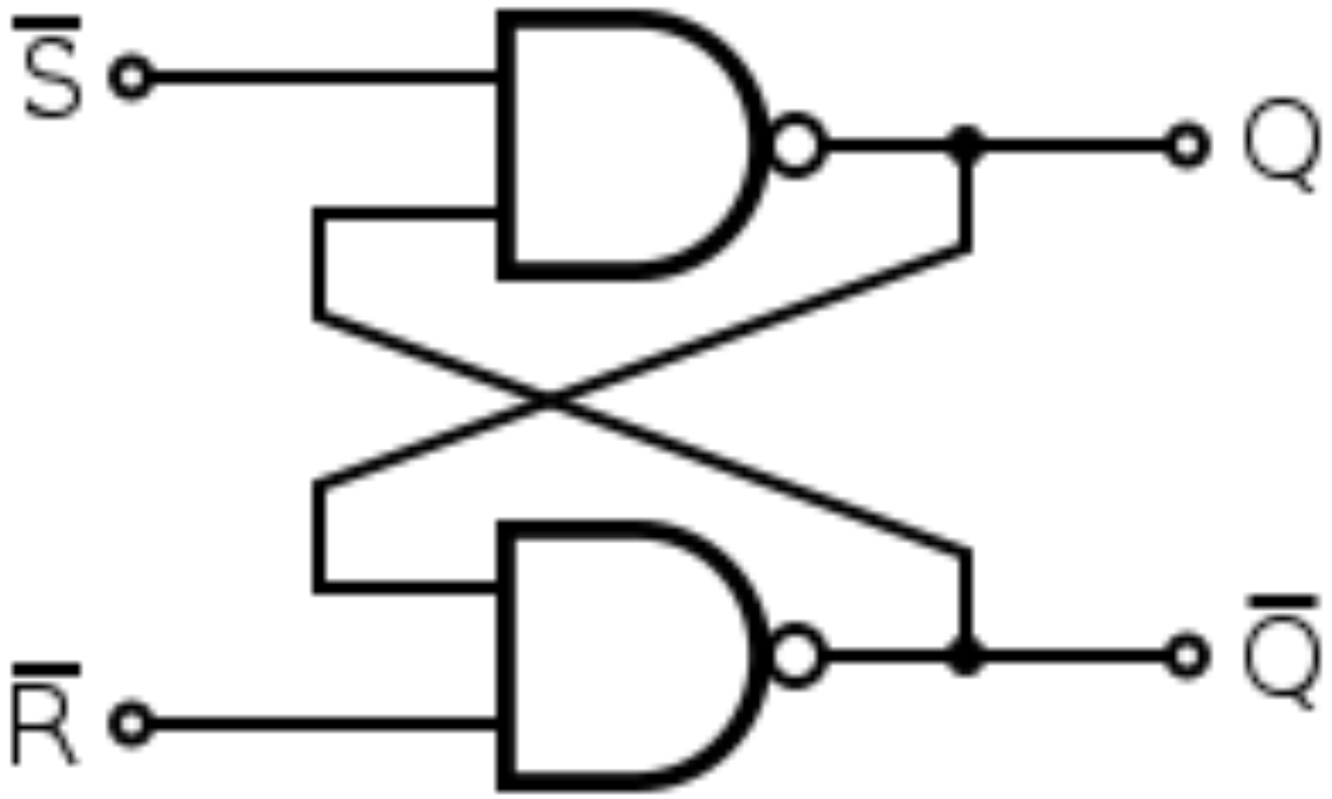
Inverter (circle indicates “not”)

Flip Flops

- The circuits for computing boolean logic (and the circuits for computing addition, etc) are simple and directional. Values come in at one end, flow through, and come out at the other.

- Now we look at circuits that include **feedback**.

Flip Flop

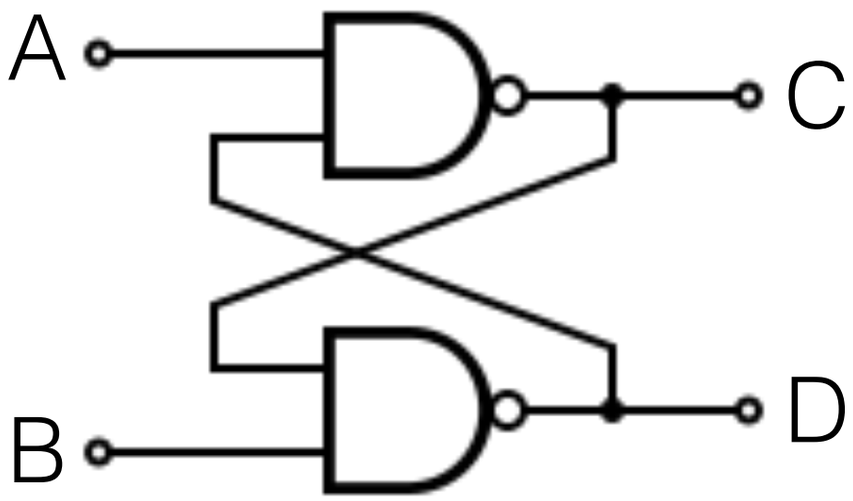


Two nand gates.
What are the stable configurations?

Nand gate

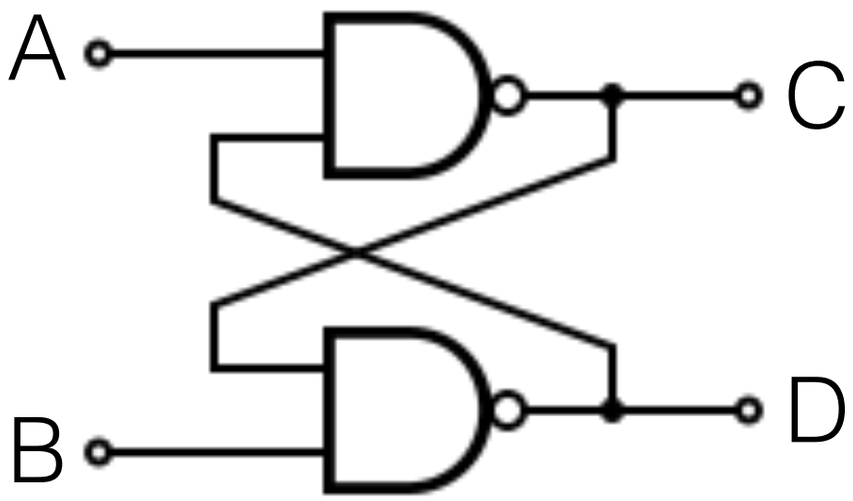
- Output is 1 if either input is 0
- Output is 0 if both inputs are 1
- Note that if you increase an input, you decrease the output (or leave it the same).

Flip Flop



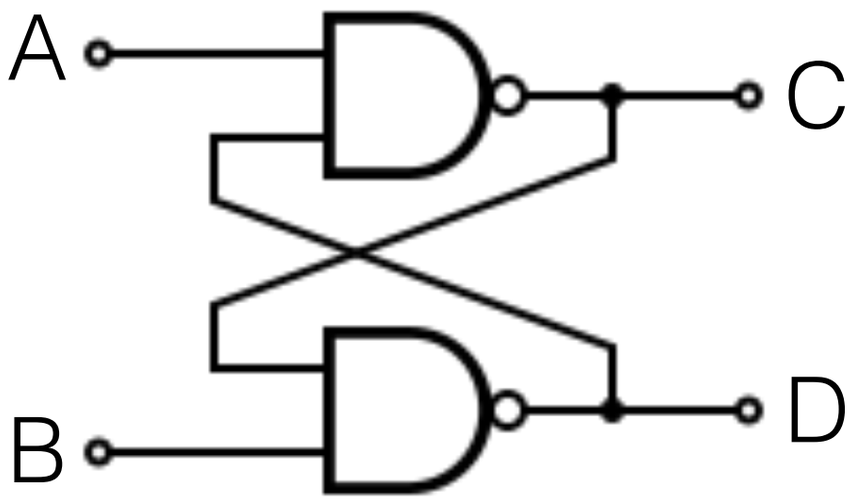
A	B	C	D
0	0		
0	1	1	
1	0		
1	1		

Flip Flop



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

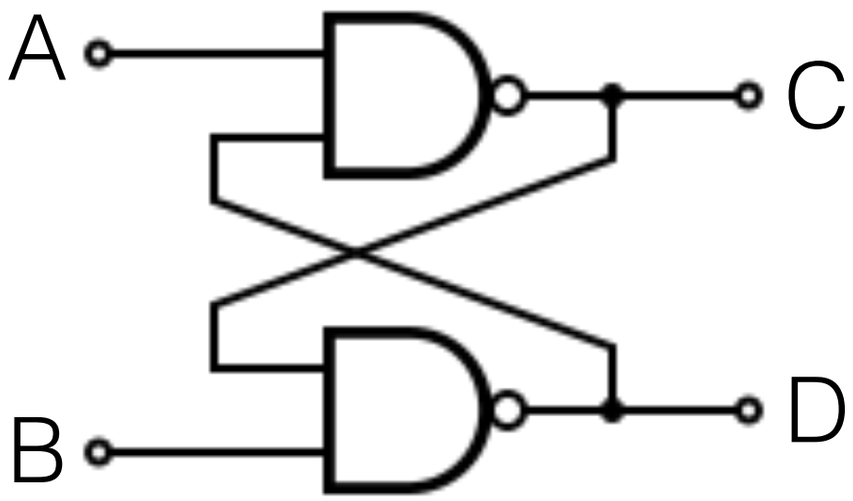
Flip Flop



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

Can’t have (1 1 0 0) or (1 1 1 1)

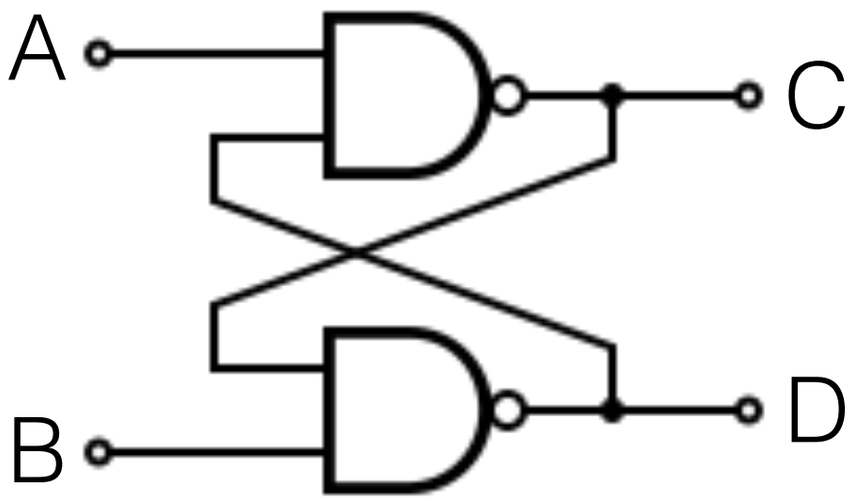
Flip Flop



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

So if A and B are both 1, which do we get?

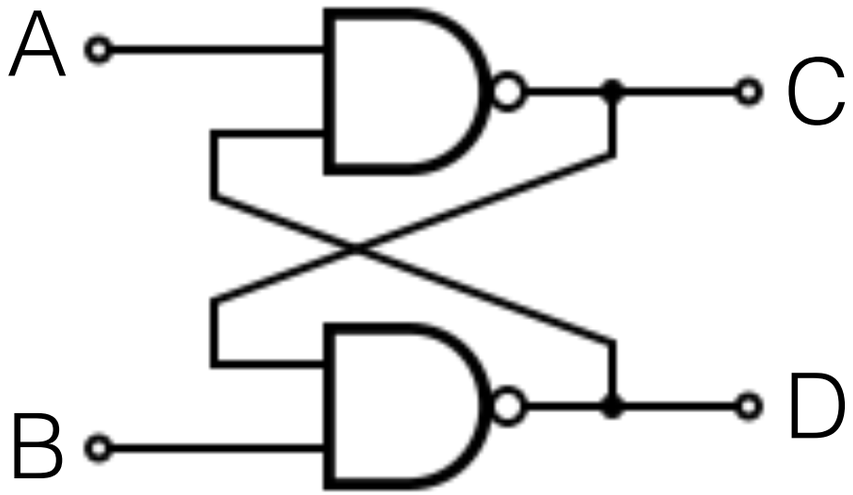
Flip Flop



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

So if A and B are both 1, which do we get?
Ans: the one we had before. The charge distribution in the feedback loop keeps the circuit stable.

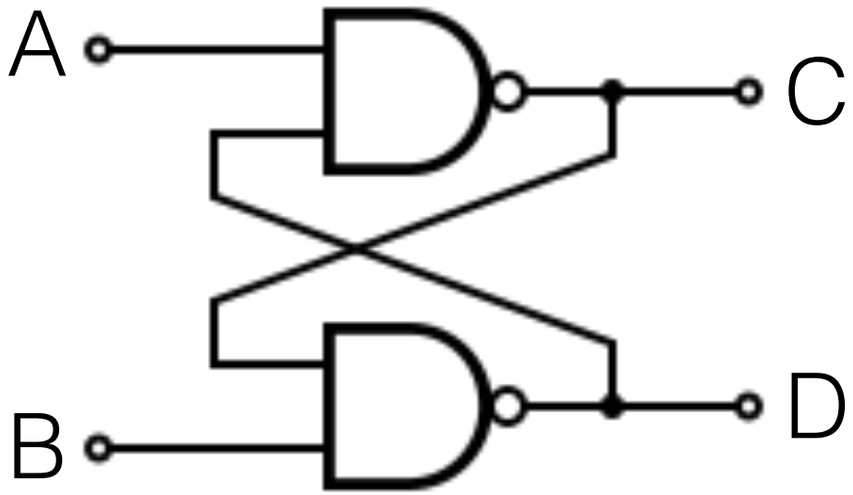
Memory



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

This means that if (A B) = (1 1) then the flip flop “remembers” whether they were previously (1 0) or (0 1).
In other words it stores one bit of information.

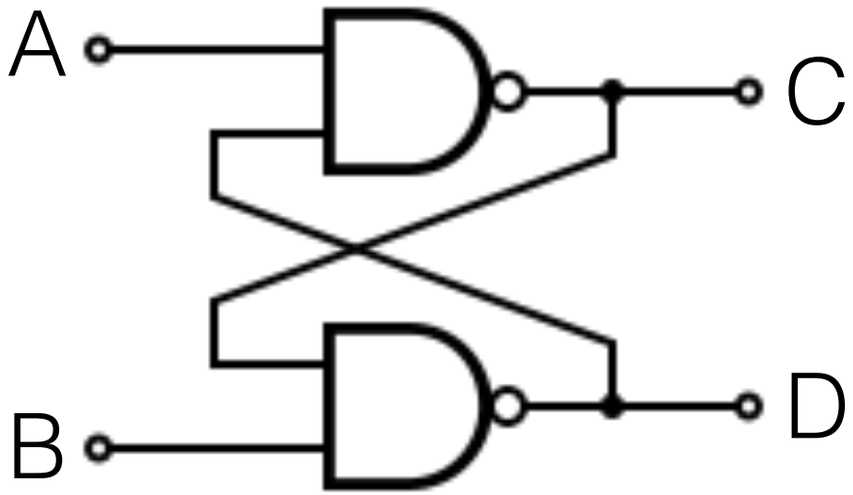
Flip Flop as Memory



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

A & B are controls
D is the read line

Flip Flop as Memory



A	B	C	D
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0
1	1	0	1

A = 0, B = 1: store 0
A= 1, B = 0: store 1
A = 1, B = 1: hold previous
A = 0, B = 0: not allowed
D is the read line

Memory

- .. and that is enough.
- If we can store one bit, then we can build whole computer memories.
- WARNING: real memory circuits are more complex.

Exercises

- In this week's lab you will be (amongst other things):
  - producing transistor designs for other gates
  - going a bit further to show how the flip flop can be used to store one bit of data

Key Learning Points

- How field effect transistors work in terms of the electron/hole gas model.
- How field effect transistors operate as switches
- Basic structure of simple logic gates.
- Use of feedback to implement memory.

Part 2: Comms

- Last part was about how chips work.
- Concentrated on how things were done with single bits.
- This section is about how data is moved from component to component.
- Covers both: between separate chips and between subcomponents of the same chip

Bus

- A comms link that can carry more than one bit of information at a time.

- NB a single wire carries a single bit at a time.

Buses

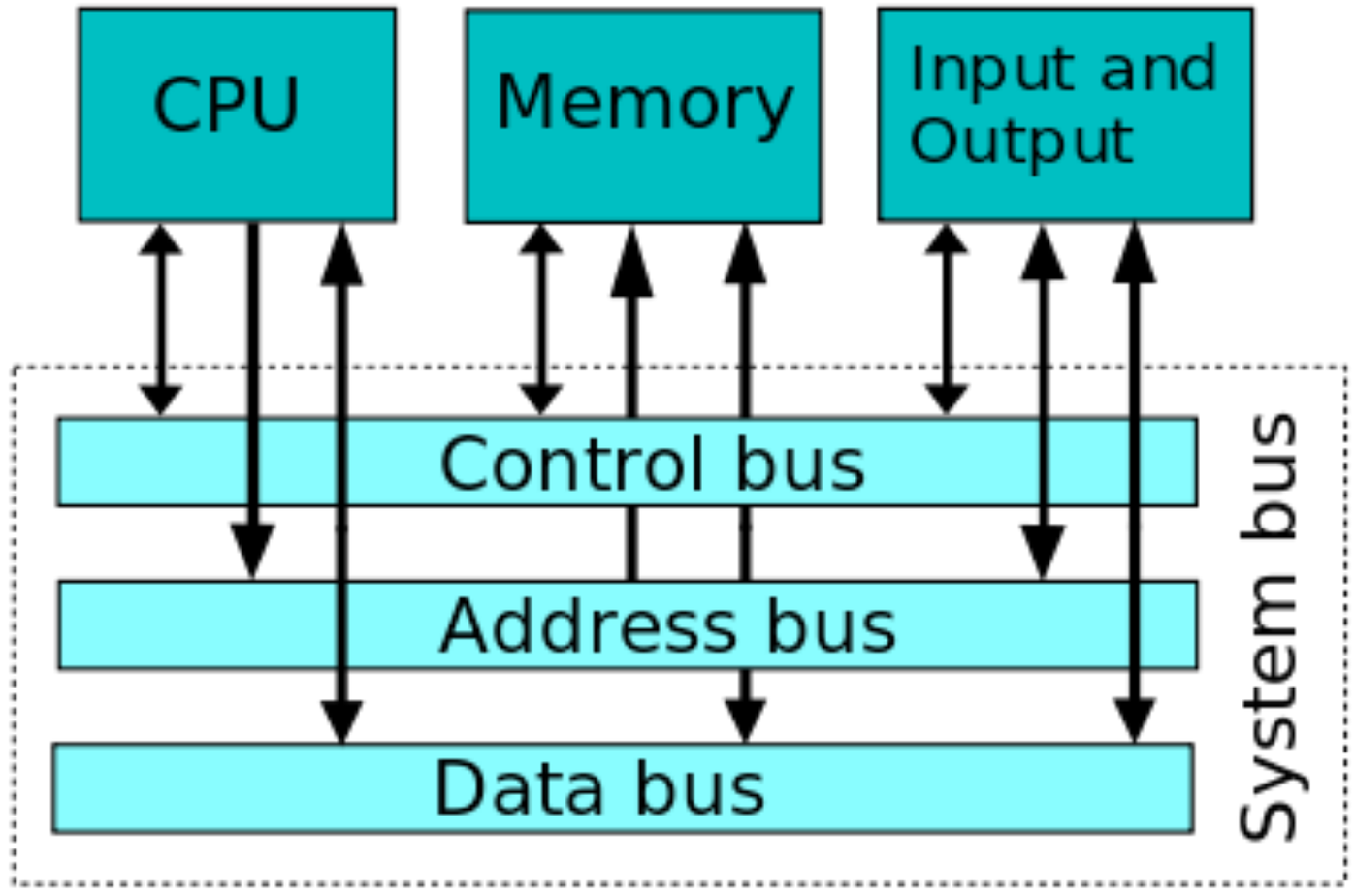
- Buses are the main means of communication inside a computer.

- Traditionally focus is on bus connecting cpu and main memory (known as the front side bus, or just the system bus).

- But cpu's also include internal buses.

Buses

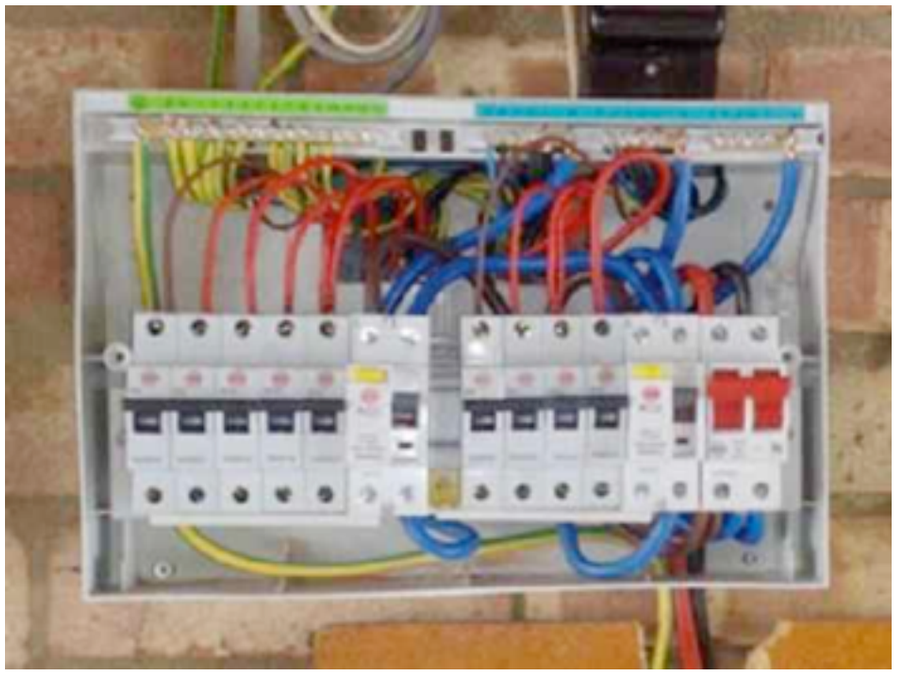
- A simplified picture



Don’t think

Think

- The lines that carry the power to the different circuits in a fusebox are called busbars.

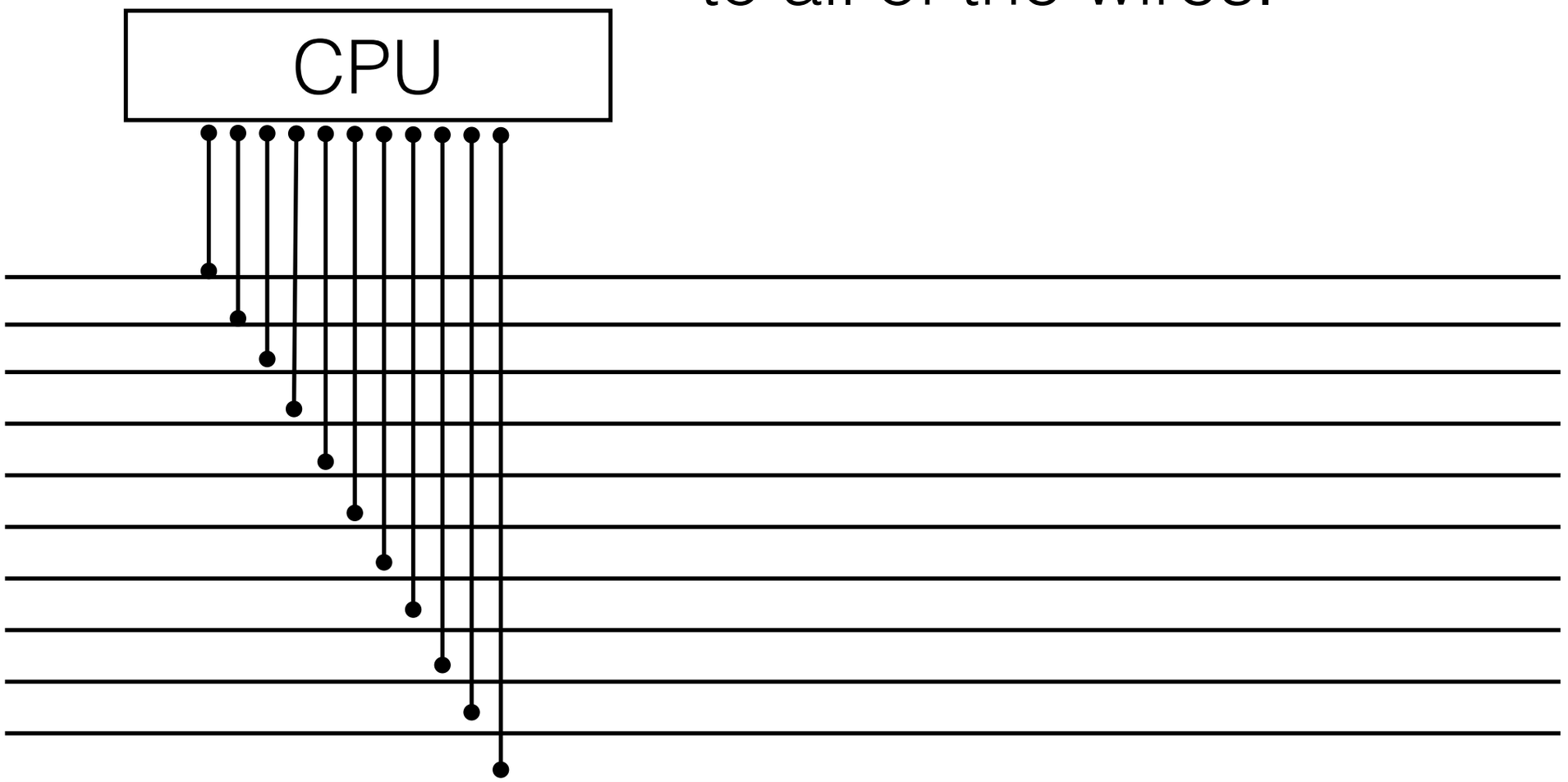


Basic picture

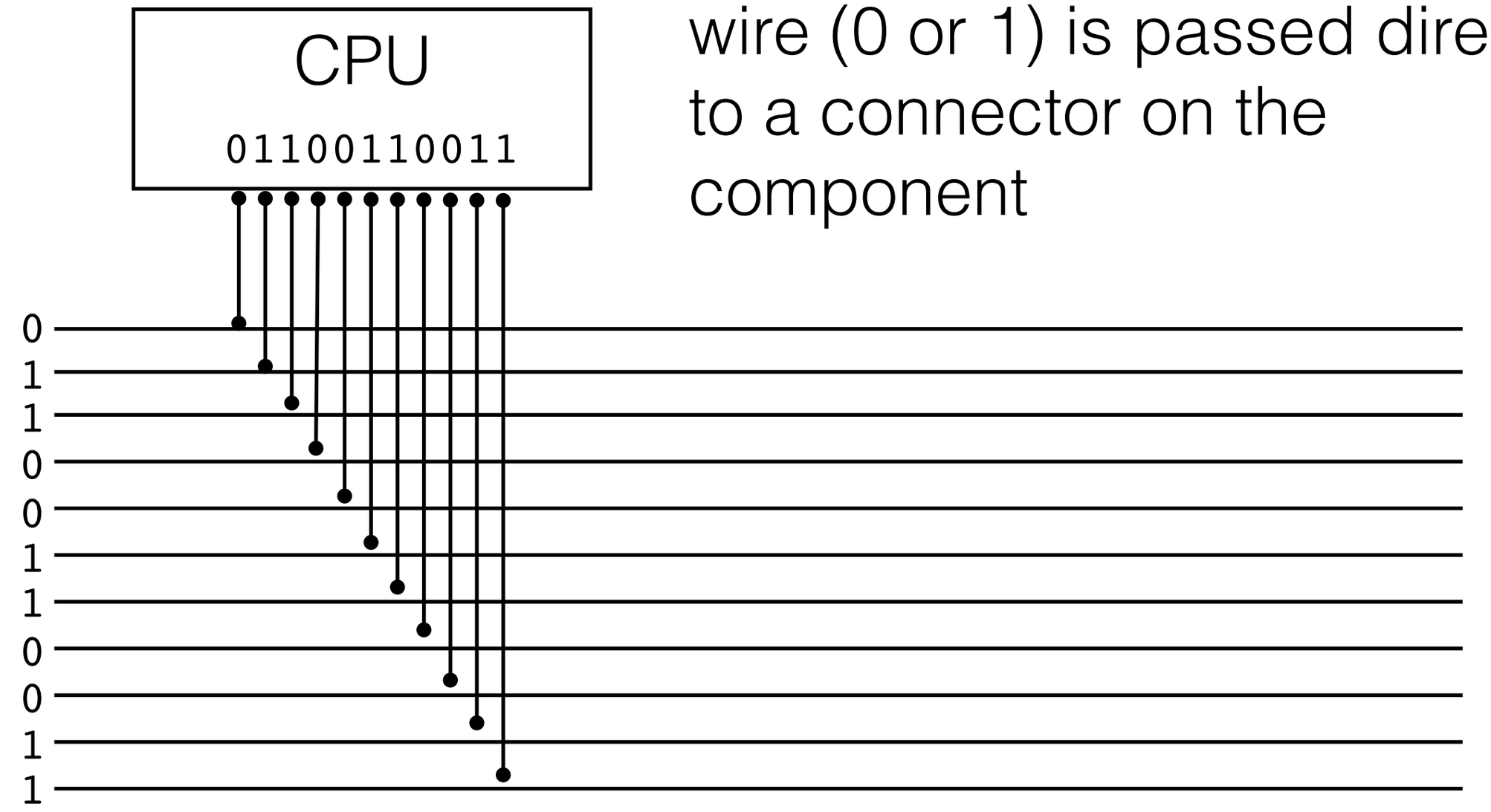
Think of the bus as a whole load of parallel wires.
This is the way it used to be.

Basic buses

Each of the components on the bus is connected to all of the wires.

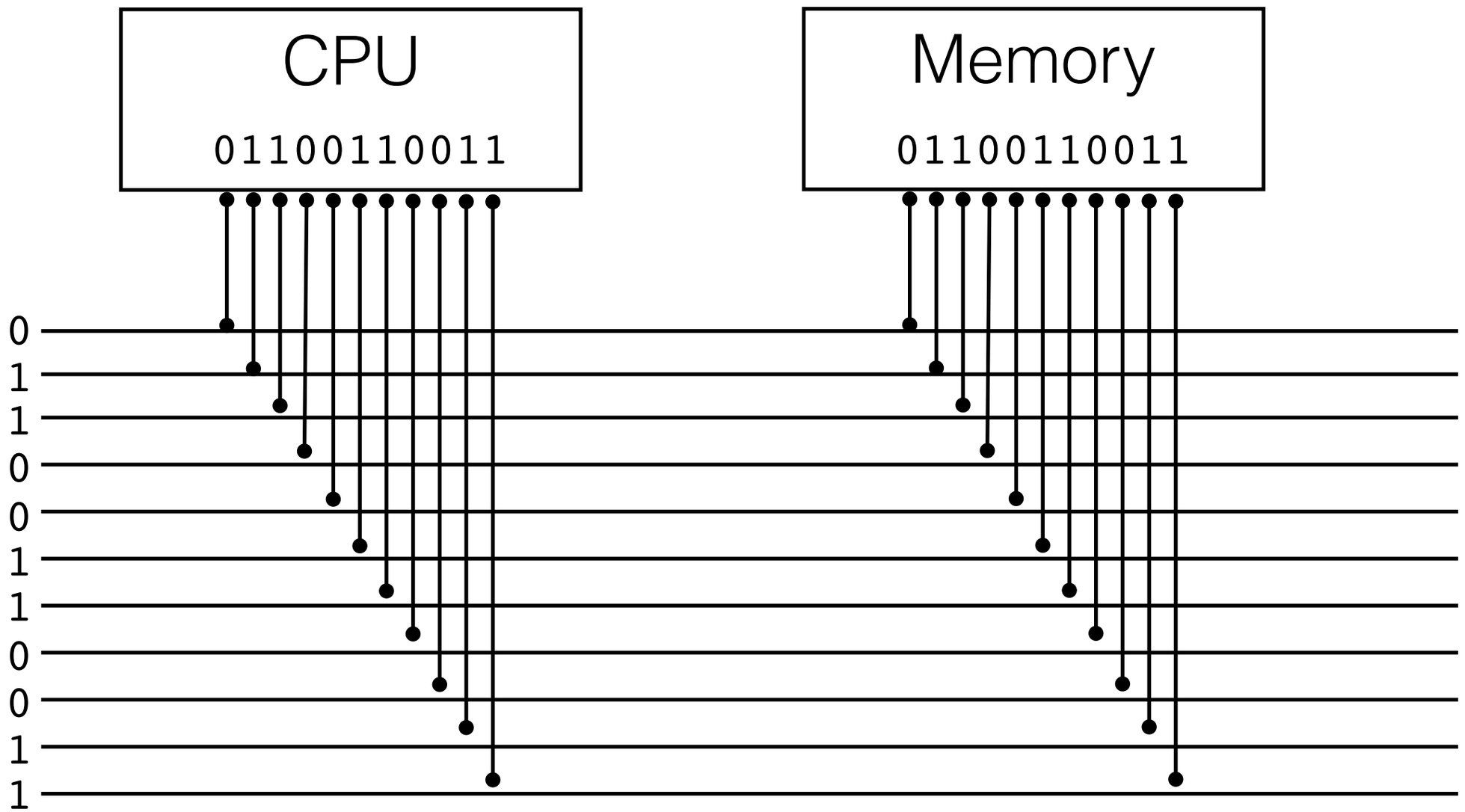


Basic buses

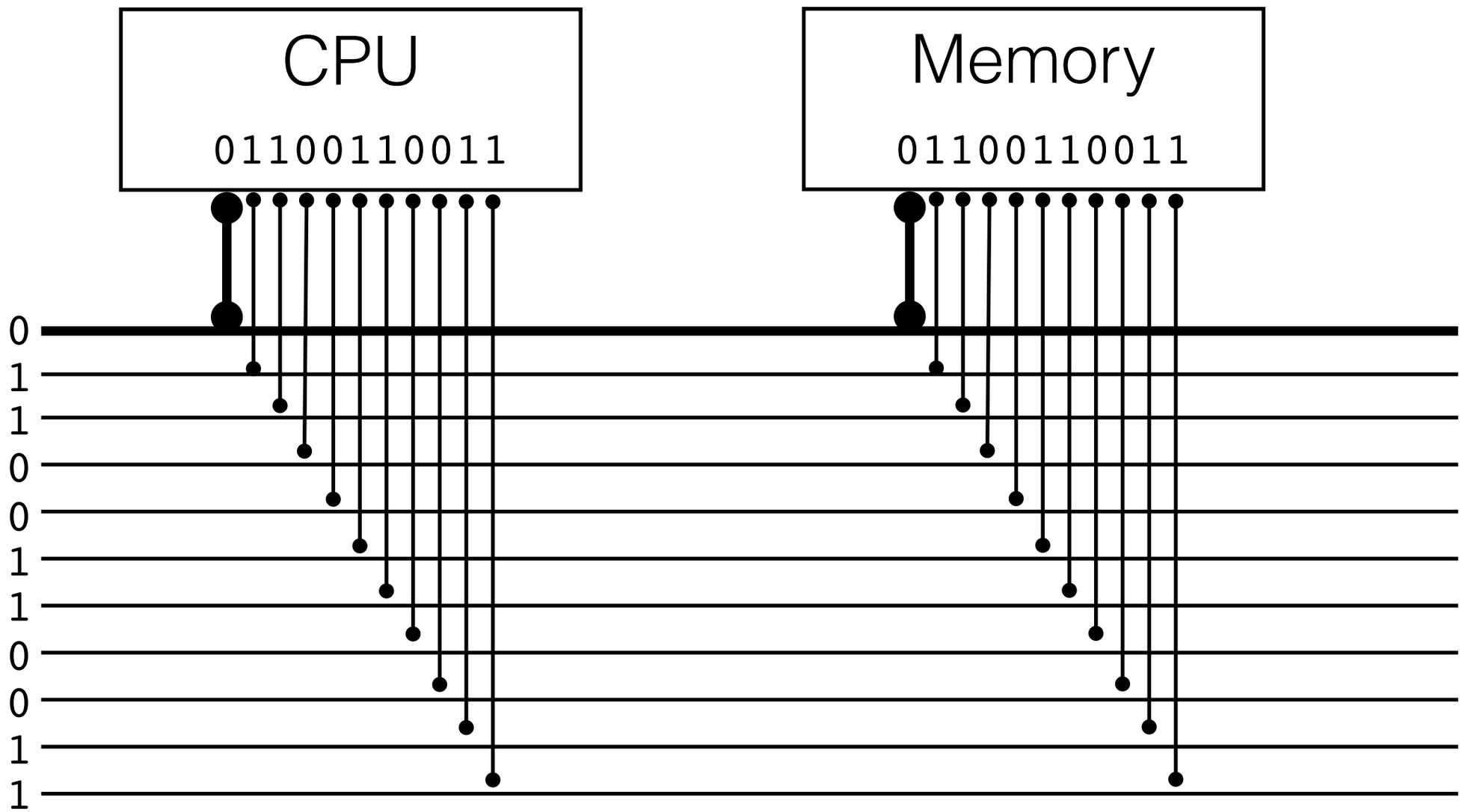


So a high or low voltage on a wire (0 or 1) is passed directly to a connector on the component

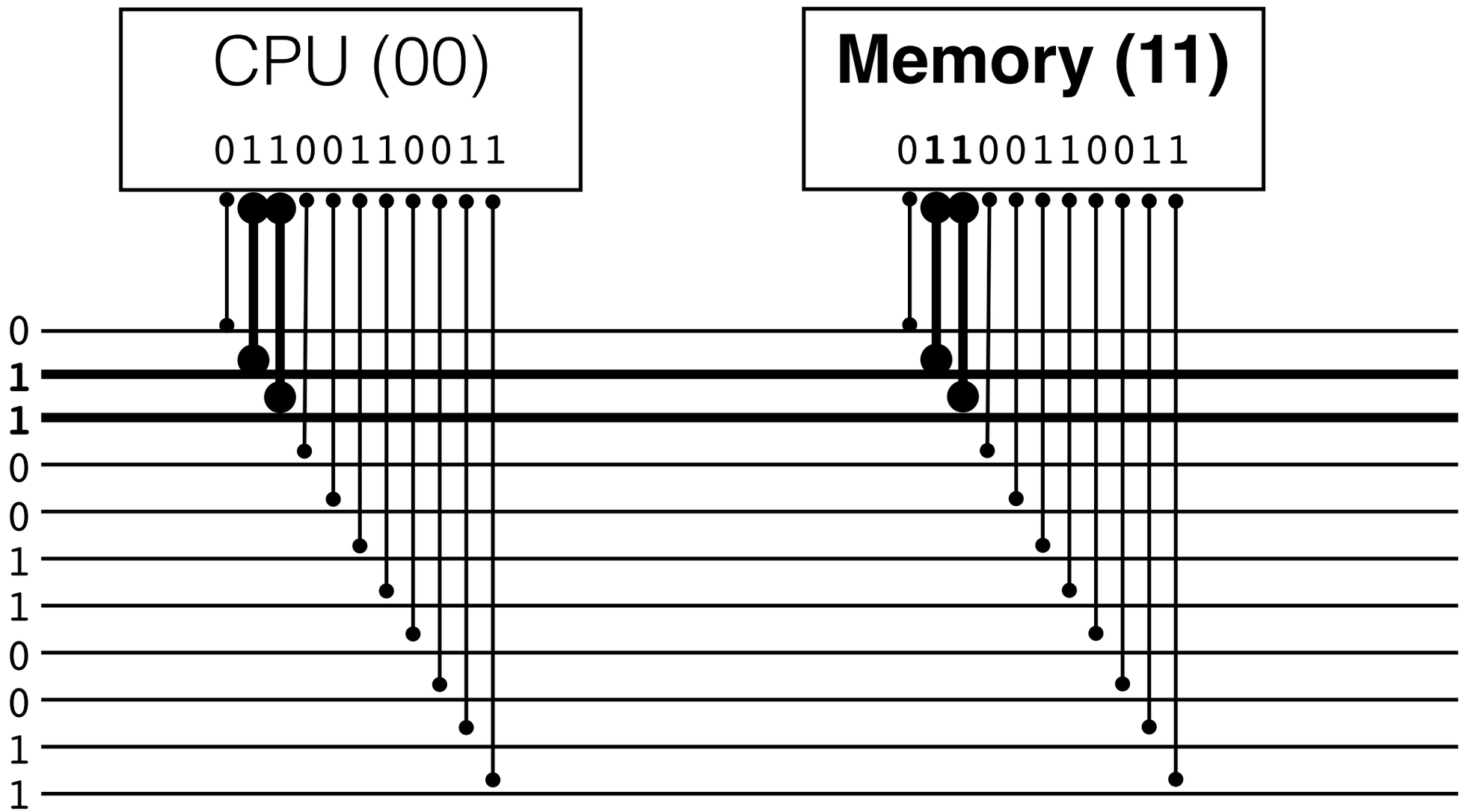
Everything on the bus sees the same thing.



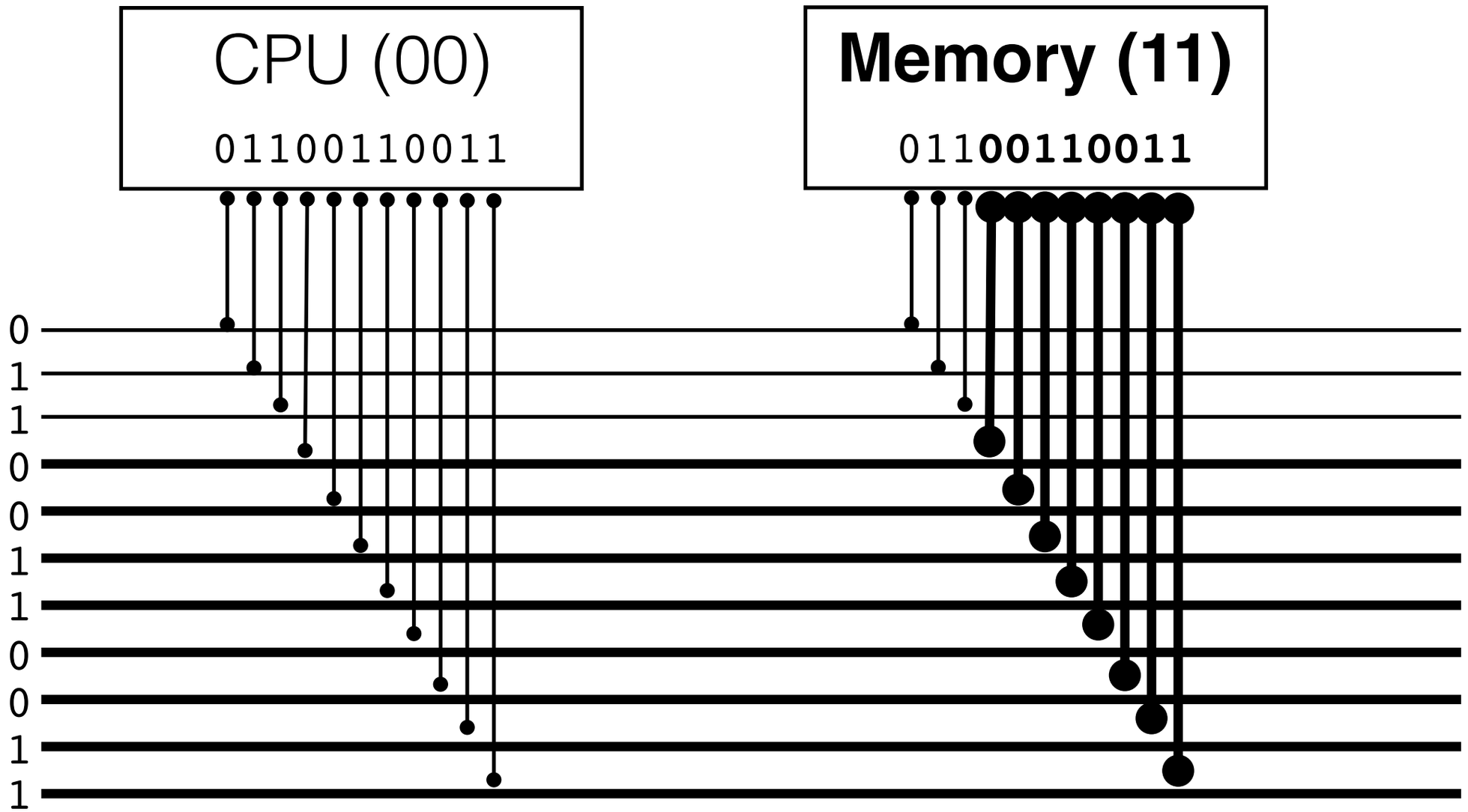
Some of the wires are used for timing and synchronisation.



Some of the wires tell the components which one the current data is for: each component has a number.

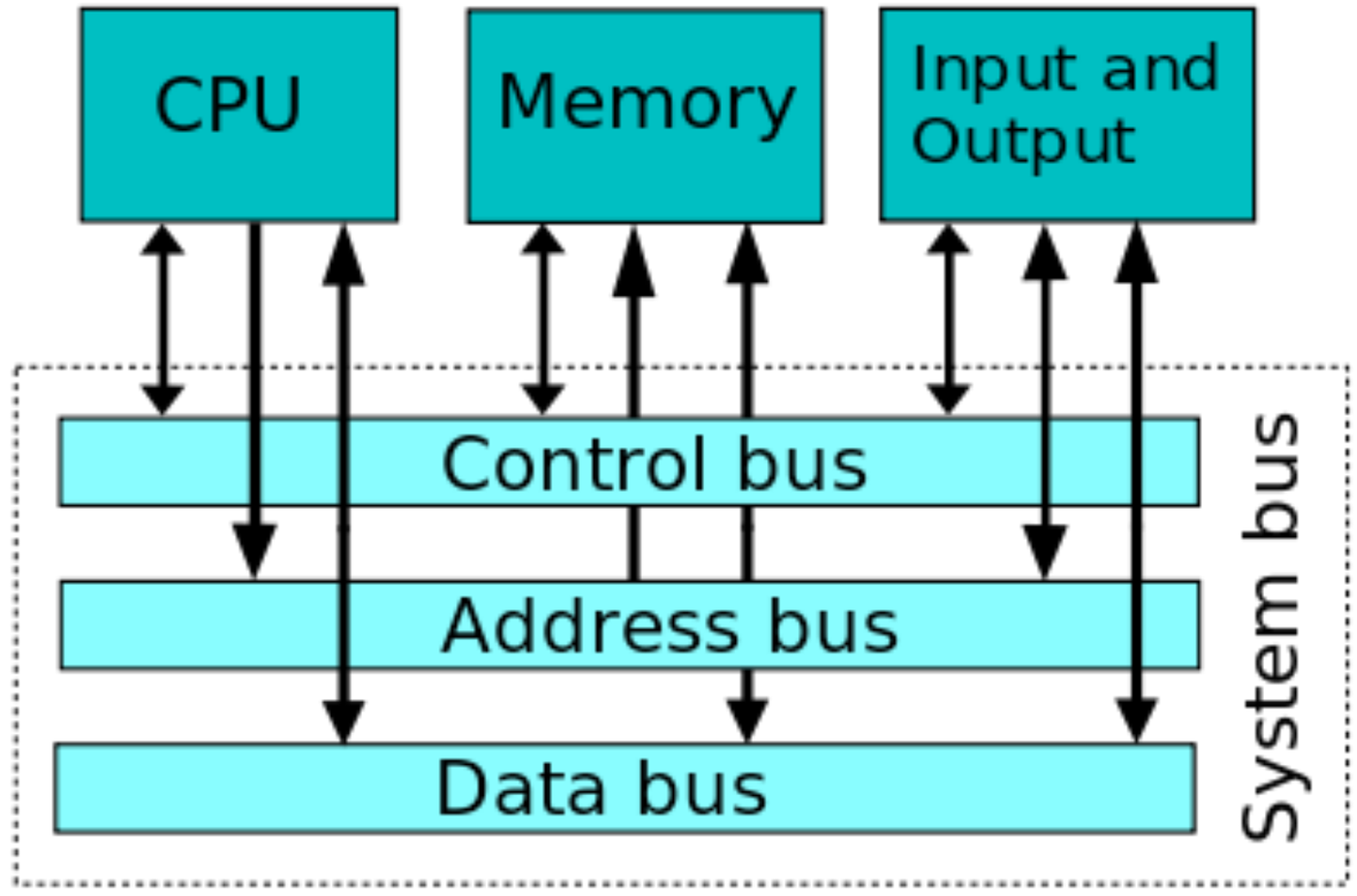


The rest are data



Buses

- This is summarised here.



Modern buses

- The bus we drew has 8 data bits.
- This fits with very early microprocessors.
- We now use 64 bit architectures.
- But for separate peripherals using 64+ wires in parallel does not make sense.
- So more complex designs using fewer actual wires are now used.

Timing

- CPU clock cycle 0.5 ns
- Execute a CPU instruction ≥ 1 ns
- RAM access 100 ns
- Disk access ≥ 1 ms

- But RAM access takes a fixed time, while disk access can vary hugely, as can accessing any network storage.

Key communication concepts

- Communication type: Synchronous and asynchronous communication

- Communication speed: bandwidth and latency

Communication channels can be **Synchronous** or **Asynchronous**

- **Synchronous:** All parties are synchronised so that when one party sends a message the others can be ready to receive it. Works for eg CPU and RAM. Makes communication designs simple.

- **Asynchronous:** Parties are not synchronised and messages may have to be held over in a buffer. Makes communication designs more complex.

Bandwidth and Latency

- **Bandwidth:** measures **rate** of data transfer. The amount of data the link can carry per unit time. It is measured in bits per second.

- **Latency:** measures responsiveness. The time it takes to receive a response. It is measured in seconds.

Latency and Bandwidth

- This design of bus is very low **latency** (everything on it sees the same thing at once).

- But the **bandwidth** is crucial.

- A 400 MHz bus means the bus can carry 400 million pieces of data per second.

- For a 2 GHz quad-core processor, this is at most one piece of data in one direction per core every 20 cpu clock cycles.

- It is easy for the bandwidth of the front-side bus to be too small for jobs that involve a lot of communication between cpu and memory.

Bandwidth and Latency

- I’ve presented this as a bandwidth issue.
- But obviously if your network is always busy because it does not have enough bandwidth, then it will be less responsive and you will have a latency issue as well.

To be continued..

- We will continue with this next week.

Key learning points (Comms)

- Main communication devices inside computers are buses.
- Buses communicate more than one bit at a time and have control, address and data sections.
- Bus speeds may govern the speed of certain calculations, even the overall speed of a computer.
- Comms can be synchronous or asynchronous.
- Speed of communication governed by bandwidth and latency.