WHAT IS A COMPUTER AND HOW DOES IT WORK ?

What is a Computer ?

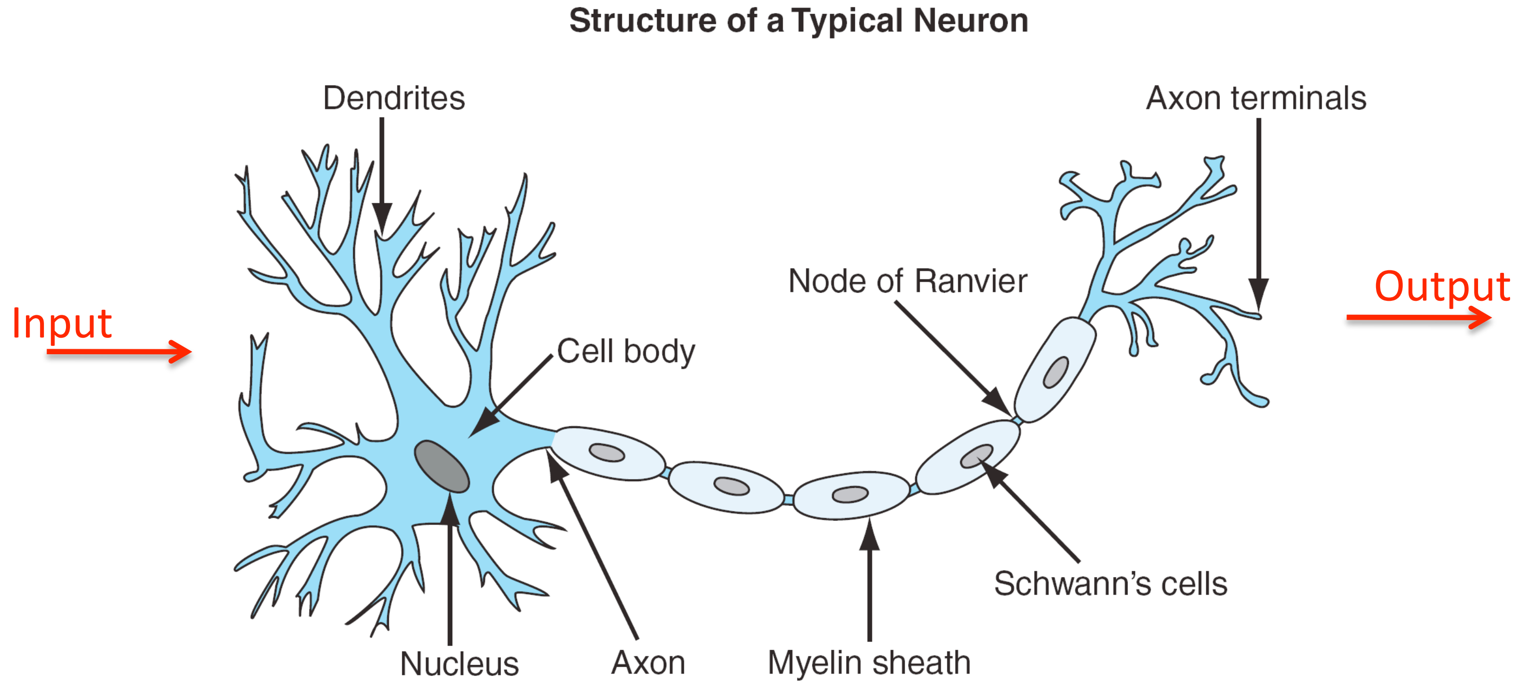
- Kind of obvious, but a computer is … **something that does computation.**
- What is interesting in it is what is going to be computed.
- In the 1960's, when computers were becoming popular, they were commonly called **"electronic brains."**
- However, human brain has very little in common with computers (except that they both do "computation").

The Human Brain

- The **human brain** consists of approximately 100 billion (10^{11}) small cells called **neurons**.
- Neurons have a **simple task**: to act as a kind of **switch**.

How neurons work

- A neuron sends a signal (“**fires**”) **when enough chemical transmitters accumulate at its dendrites**.



- When it fires, the signal, as a chemical/electrical **impulse moves down the axon**
- When the signal reaches the terminals, **more chemical transmitters are released to other neurons.**

How neurons work

- **Signal transmission is very slow**
  – Millions of times slower than a computer
- Neuron's require **"recovery time"** before they can fire again.
- What causes a firing depends on the **number of signals a neuron receives** at its dendrites and how fast they arrive.
  – that is, it *depends on the number of connections*.

People !

Computers used to refer to *people*

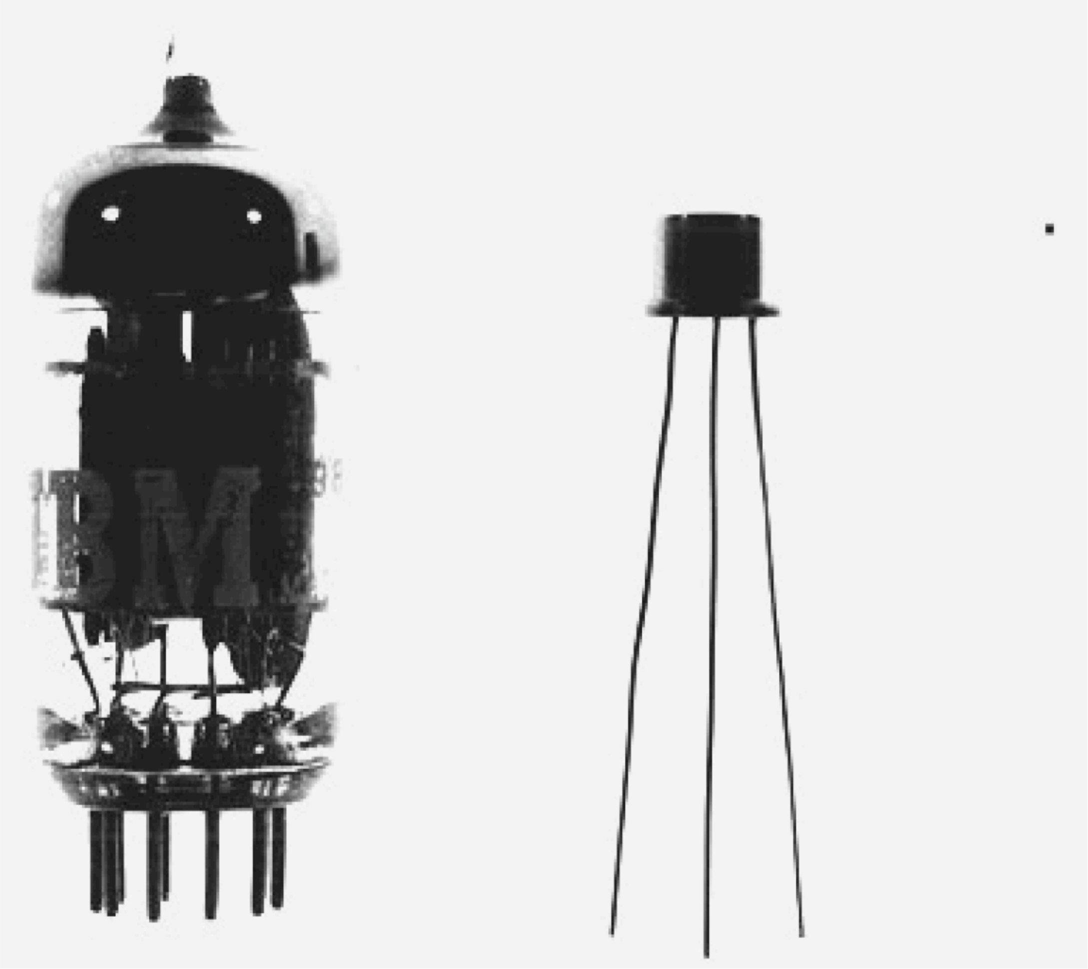
In WWII, computers were people who did **difficult calculations** ***by hand,*** for things like ballistic tables.

The importance of being … a switch

- As in the brain, the basic component of most digital circuitry is nothing more complicated than a simple **switch**.
- A **switch's function** is pretty obvious, said in a number of different ways
  - On or Off
  - True or False
  - 1 or 0

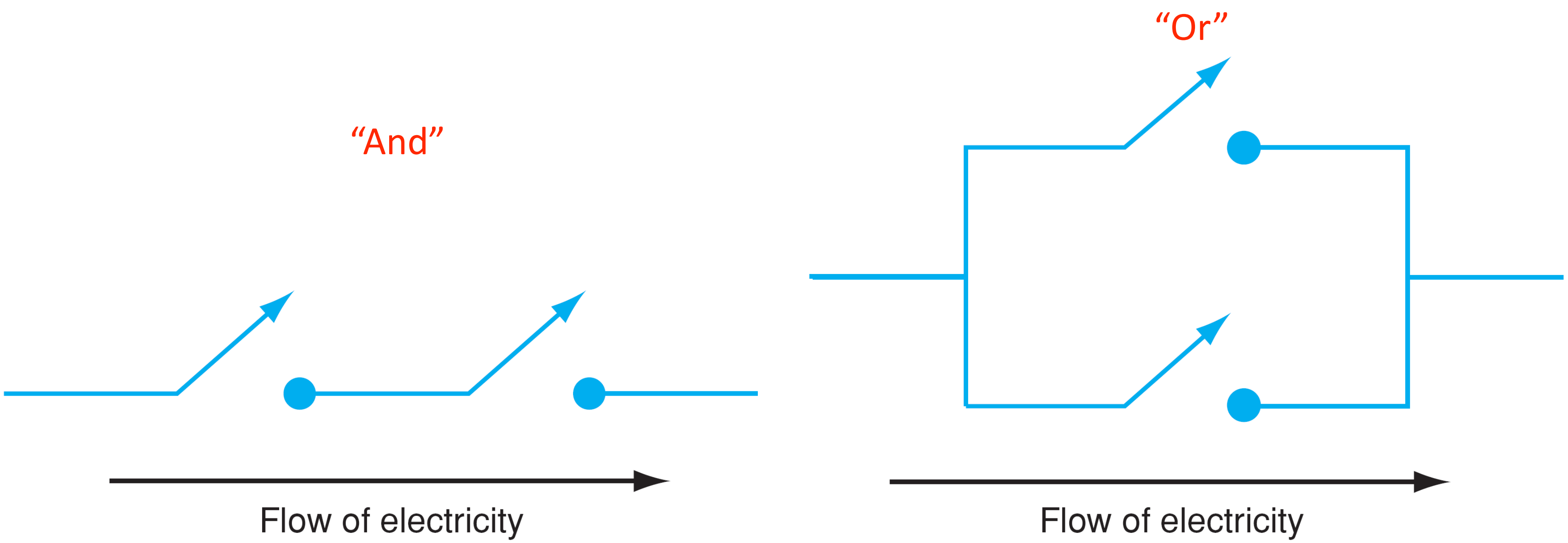
Electronic switches

- Early computers used vacuum tubes as switches
- Later, transistors were used as substitutes



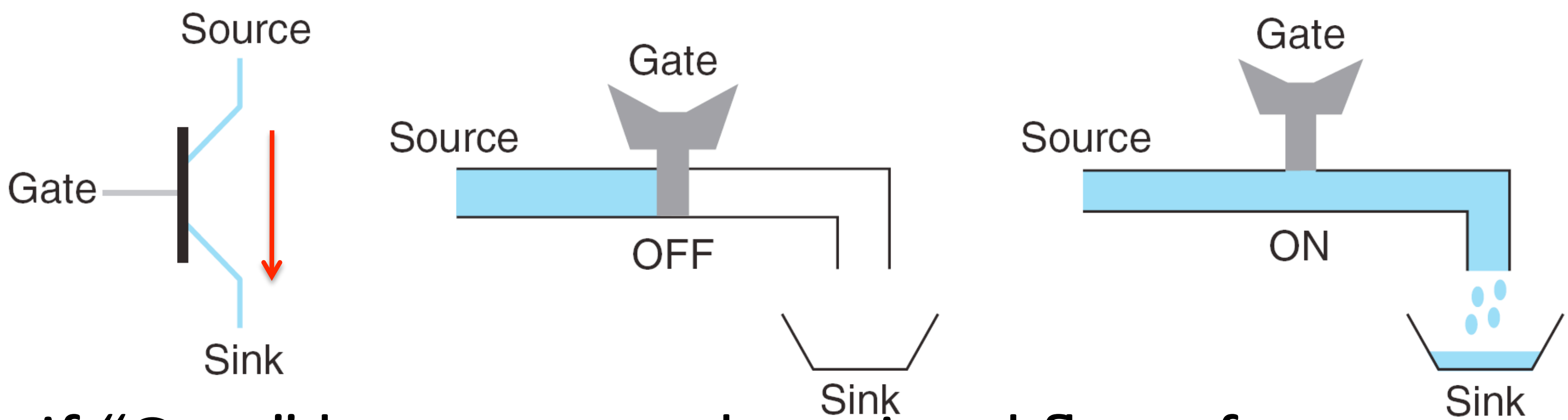
Switches for Boolean circuits

- Switches can be used to construct more complicated functions, such as Boolean circuits (**and** on left, **or** on right)



Transistors

The transistor is an electronic switch:



- if "Gate" has current, then signal flows from "Source" to "Sink"
- if gate has no current, no flow

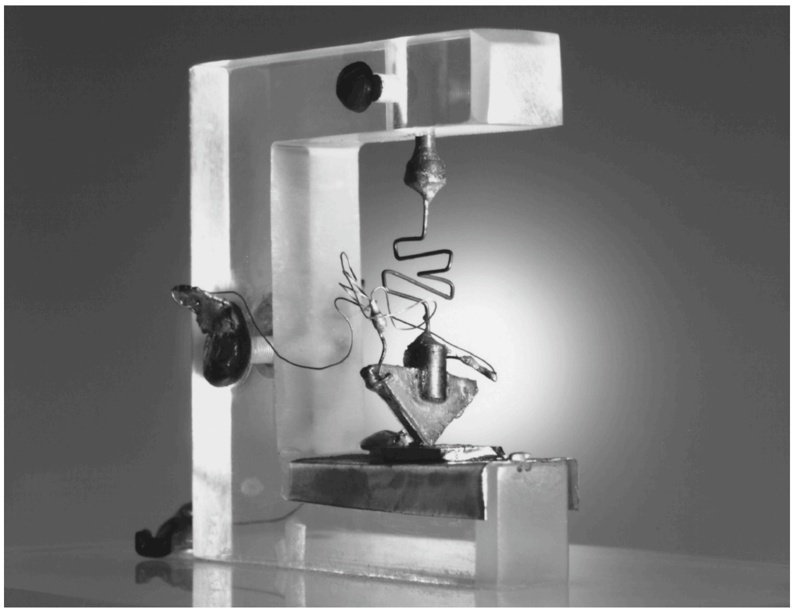
Interesting properties of transistors

Transistors have three interesting "features" that have made them the **fundamental element of the computer revolution**:

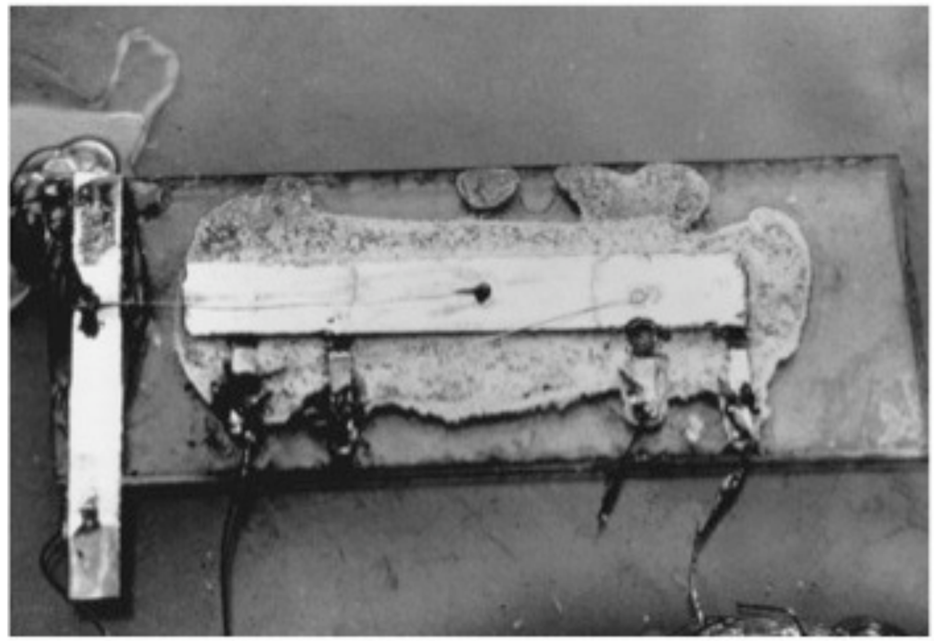
- size
- quantity
- speed

Size of transistors

Originally very large



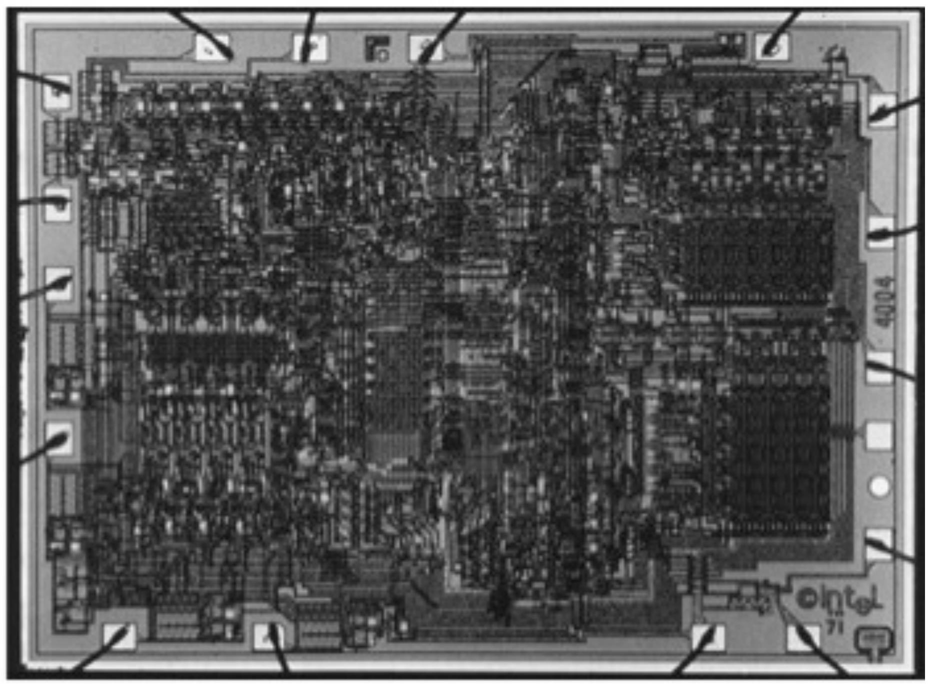
Shockley transistor



Kilby integrated circuit

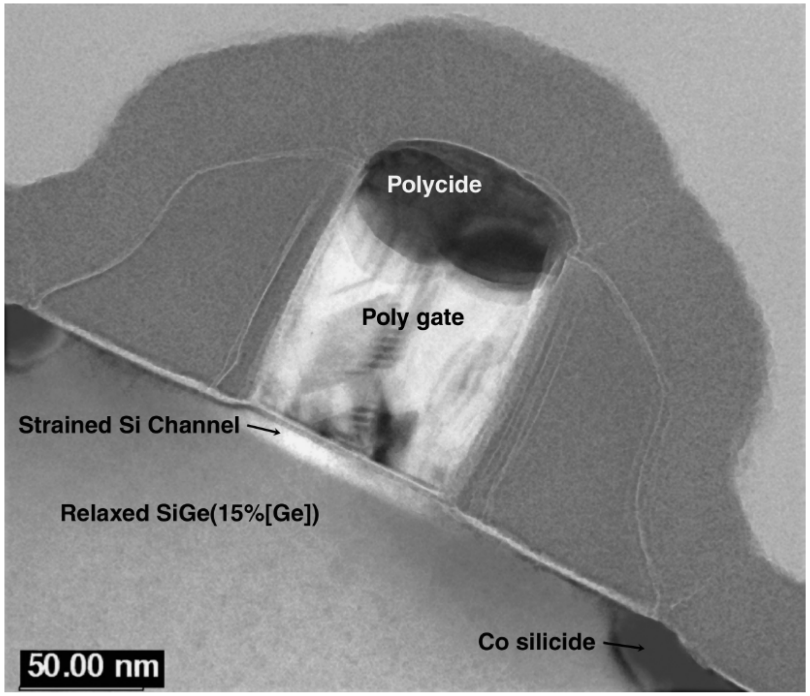
Intel's first CPU

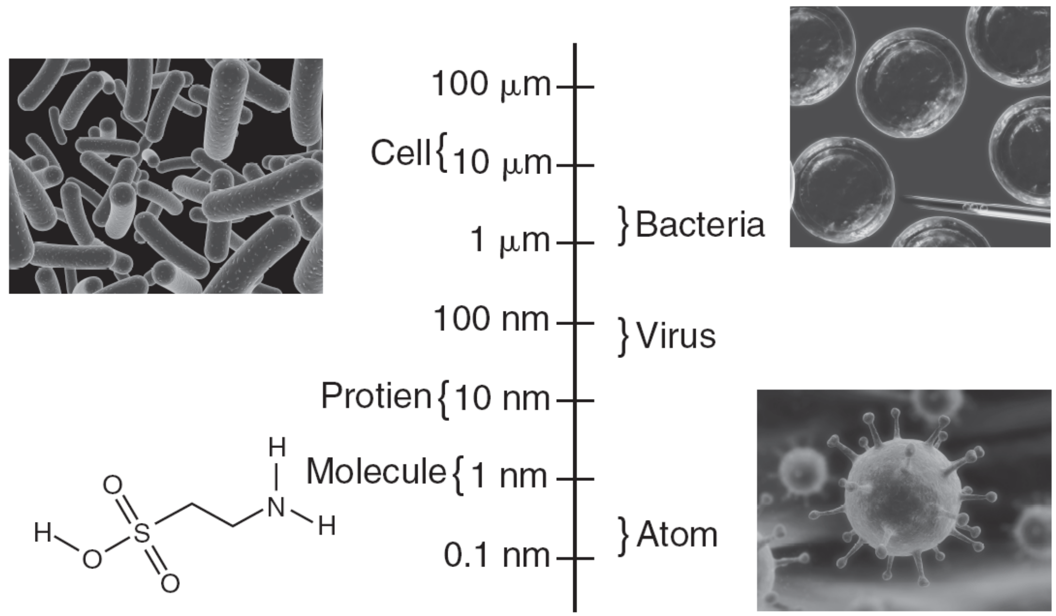
By 1971, Intel had created a "computer on a chip," the 4004 microprocessor, the size of a fingernail with 2300 transistors.



But now… *really* small !

Chips now have gates measured in billionths of a meter (nanometers, nm)





Smaller Means More

Name	Count	Year	Co	Size
Intel 4004	2,300	1971	Intel	10 µm
Intel 8008	3,500	1972	Intel	10 µm
Intel 8080	4,500	1974	Intel	6 µm
Intel 8088	29,000	1979	Intel	3 µm
Intel 80286	134,000	1982	Intel	1.5 µm
Intel 80386	275,000	1985	Intel	1.5 µm
Intel 80486	1,180,000	1989	Intel	1 µm
Pentium	3,100,000	1993	Intel	0.8 µm
AMD K5	4,300,000	1996	AMD	0.5 µm
Pentium II	7,500,000	1997	Intel	0.35 µm
AMD K6	8,800,000	1997	AMD	0.35 µm
Pentium III	9,500,000	1999	Intel	0.25 µm
AMD K6-III	21,300,000	1999	AMD	0.25 µm
AMD K7	22,000,000	1999	AMD	0.25 µm
Pentium 4	42,000,000	2000	Intel	180 nm
Atom	47,000,000	2008	Intel	45 nm
Barton	54,300,000	2003	AMD	130 nm
AMD K8	105,900,000	2003	AMD	130 nm
Itanium 2	220,000,000	2003	Intel	130 nm
Cell	241,000,000	2006	Sony/	90 nm
Core 2 Duo	291,000,000	2006	Intel	65 nm
AMD K10	463,000,000	2007	AMD	65 nm
Itanium 2	592,000,000	2004	Intel	130 nm
Core i7 (Quad)	731,000,000	2008	Intel	45 nm
POWER6	789,000,000	2007	IBM	65 nm
6-Core Opteron	904,000,000	2009	AMD	45 nm
2-Core Itanium 2	1,700,000,000	2006	Intel	90 nm
6-Core Xeon 7400	1,900,000,000	2008	Intel	45 nm
4-Core Tukwila	2,000,000,000	(future	Intel	65 nm
8-Core Nehalem-E)	2,300,000,000	(future	Intel	45 nm

GPUs too !

Processor	count	Date	Co.	Process	Area
G80	681,000,000	2006	NVIDIA	90 nm	480 mm^2
RV770	956,000,000	2008	AMD	55nm	
GT200	1,400,000,000	2008	NVIDIA	55 nm	576 mm^2
HD5800	2,154,000,000	2009	AMD	40 nm	334 mm^2
GF100	2,900,000,000	(future)	NVIDIA	40 nm	

"Moore's law"

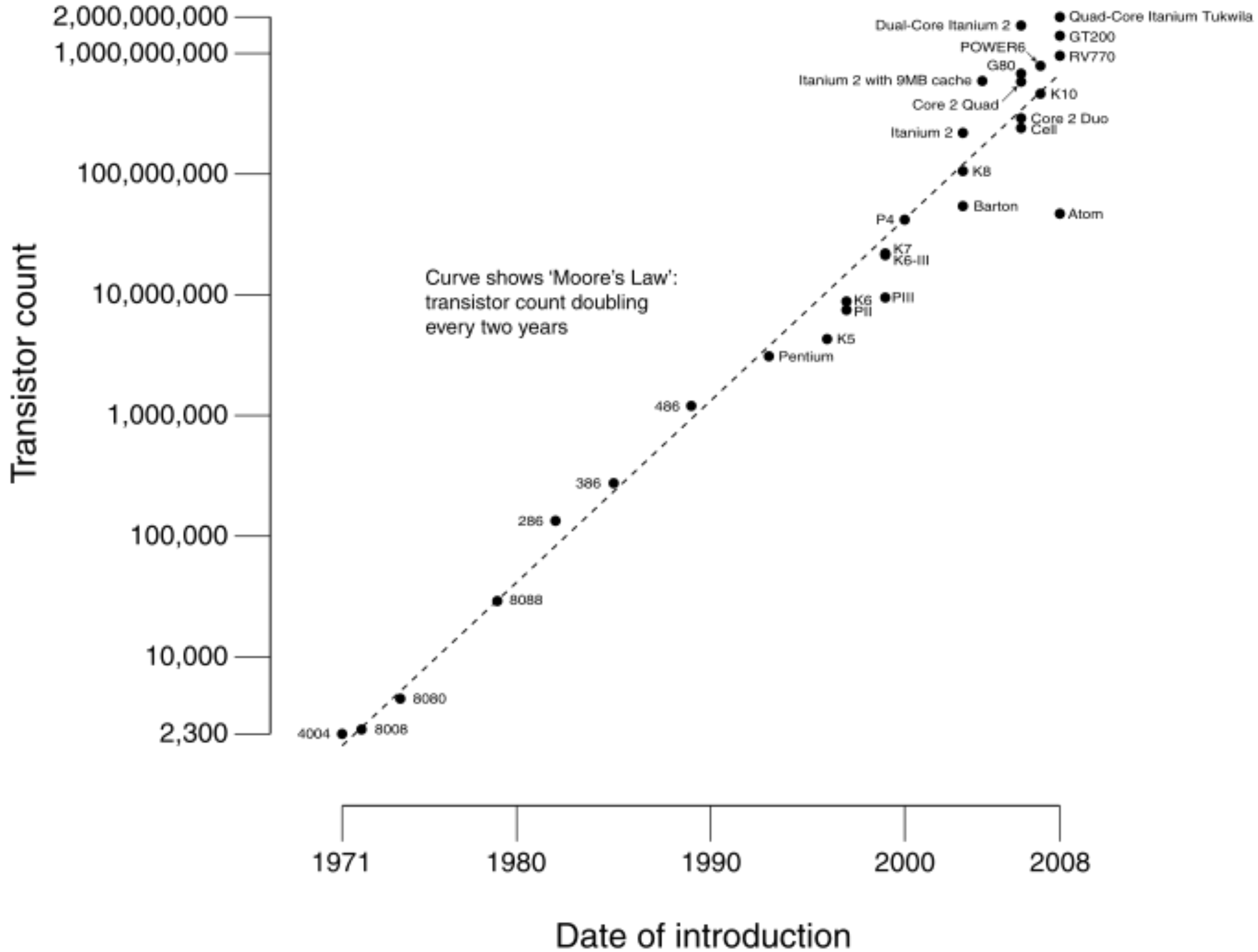
- Gordon Moore is one of the founders of the chip maker Intel
- in 1965, he has observed (over that last 15 years or so) the **growth rate of the number of transistors** in a circuit
- Made a famous prediction…

The prediction

"The complexity for minimum component costs has increased at a **rate of roughly a factor of two per year** ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer"

Electronics Magazine 19 April 1965

CPU Transistor Counts 1971-2008 & Moore's Law



What it means

- Roughly, since 1965, the **number of transistors on a chip doubles every 18 months** for approximately the same cost
- Often quoted as the **speed of a cpu doubling** every 18 months for the same cost
- speed and density were related

Speed of CPUs against physic's limits

- Typical processor this day runs at **GHz speeds**. That is **1 billion "operations" a second**.
- Ghz is not a speed, but a frequency…
- In fact the "clock" is like a drummer in the band. The faster it beats, the faster the operations.
- In **1 clock tick (at 1 GHz), electricity can travel 1 ft.** At 2GHz, 6 inches. At 4GHz, 3 inches

Nanoseconds … in meters

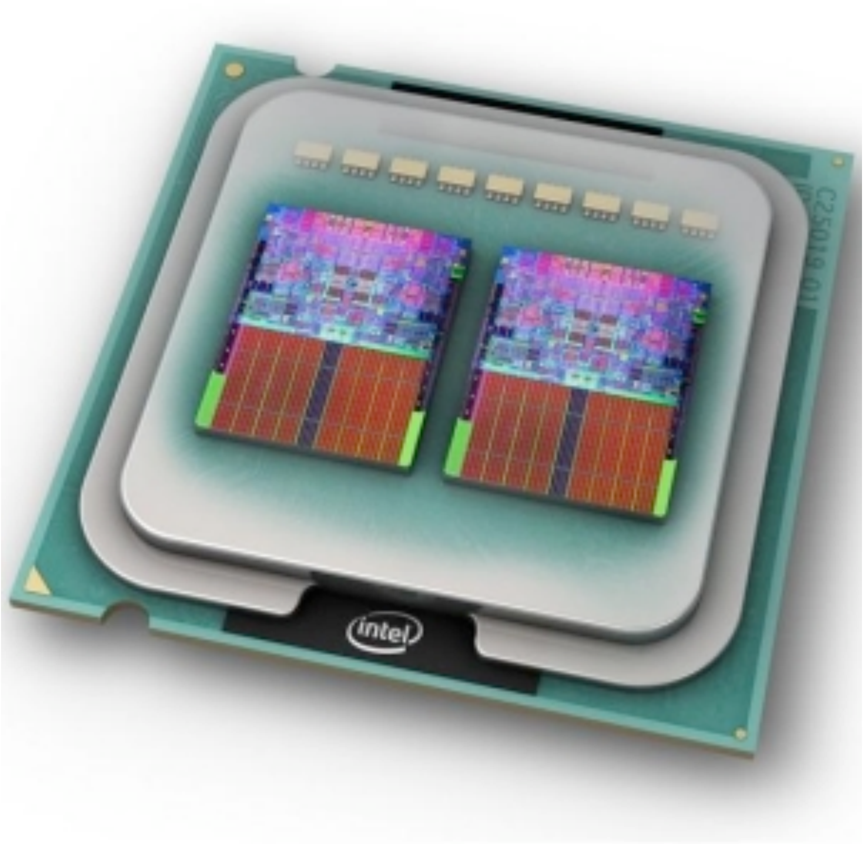
- Speed of light =186,000 miles/sec, **$3x10^8$ meters/sec** (actually 299,792,458)
- nanosecond = **10^{-9} seconds**
- thus **30 centimeters** or 11.8 inches (**in a nanosecond**)
- For example… at 4GHz only 2.95 inches!
- … so it becomes very **difficult to get electricity to “travel” any distance in such a short time**!

How to get around physics ?

- Since 2000, clock speed has not been increasing that much!
- So if physics is getting in the way (and it is), you find a way to get around it.
- If **you can't make processors faster**, what do you do?

More CPUs

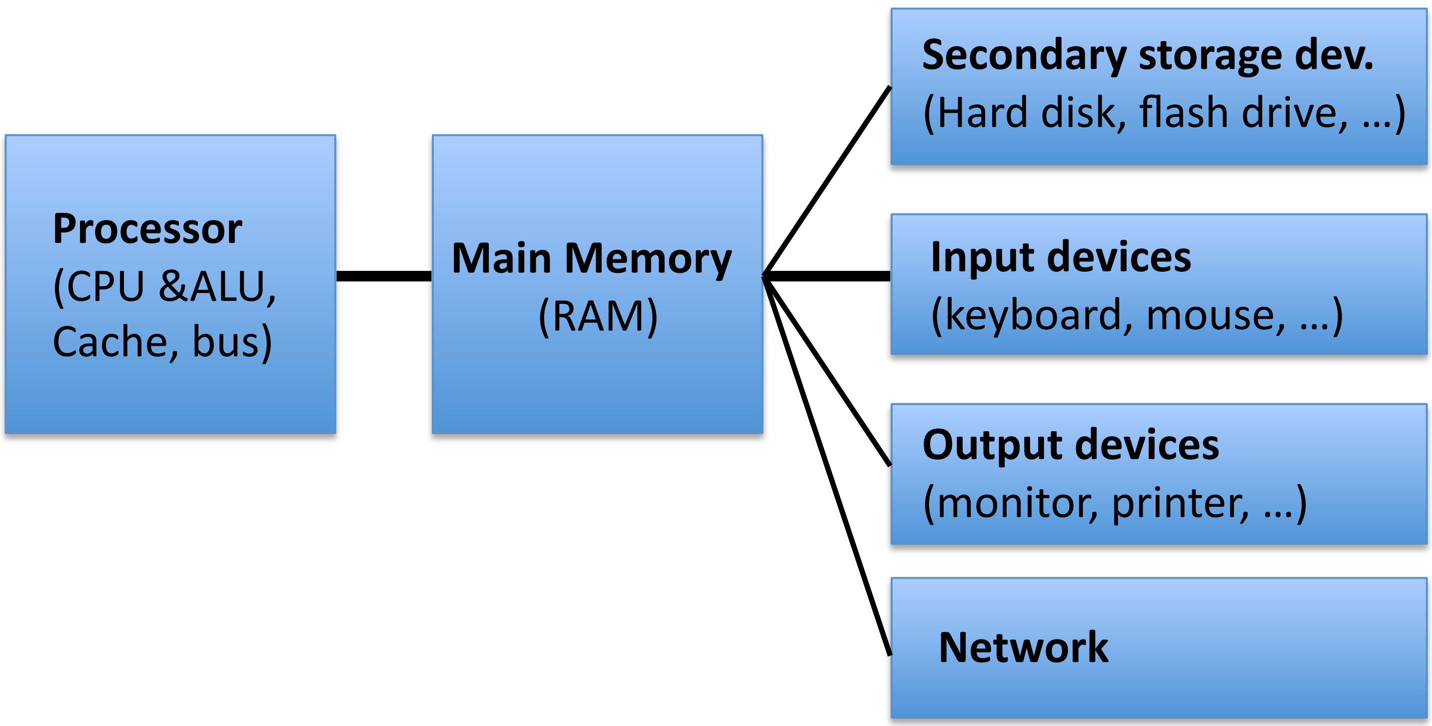
Now more transistors mean more CPUs on a single chip. Below a quad-core CPU



Computers: Main Components

- People!!!
- **Hardware**
  - Physical Devices: processor, memory, keyboard, monitor, mouse, etc.
- **Software**
  - Executable Programs: word processor, spread sheet, internet browser, etc.

Hardware: a typical computer architecture



The Processor

- The processor is the “brain” of a computer.
- The processor **controls the other devices** as well as **performing calculations**
- It is often called **CPU** (Central Processing Unit)
- CPU has inside the **ALU** (Arithmetic and Logic Unit), and local fast storage (**cache**).
- Connections happens thorough the **bus**

Main Memory

- stores instructions and data for current program(s)
  (e.g., textures of characters in a 3-D videogame)
- other names: primary or main memory, **RAM** (Random Access Memory)
- **RAM is "volatile"** so it requires power to retain information
- often hundreds of Megabytes (million-bytes)

Peripheral Devices

- Secondary storage devices
  - *disk (hard & floppy), tape, usb drives, flash drives, etc.*
- Input devices
  - *keyboard, mouse, camera, mic, etc.*
- Output devices
  - *monitor, printer, speaker, etc.*
- Network
  - *wireless, bluetooth, ethernet, etc.*

Secondary Storage Devices

- nonvolatile -- information is recorded magnetically so power is not needed
- disks hold Gigabytes (billions of bytes)
- cheap, but slow
  - RAM access is a hundred CPU clock ticks
  - disk access is a million CPU clock ticks
- not directly accessed by CPU

Software

- The **programs** available for execution
- simple classification
  - system software
  - application software

System Software

- **operating system**: manages system resources, e.g. DOS, UNIX
- **user interface**: interface with operating system, e.g. Windows, X
- **combined**: NT, MacOS

Application Software

- programs which **perform specific tasks** for the user (and use the operating system to interact with the hardware)
- examples: word processor, spreadsheet, internet browser, etc.

What is a program ?

- A **sequence of instructions** written in machine language that tells the CPU to take certain **actions** in a specific **order**
- In this course we will learn to create programs

Storage

- **Machine language instructions** are encoded as **bit patterns** (binary, remember our transistors)
- memory can only hold **binary info**.
- a bit is represented by two-states, e.g. L-R magnetism, high-low voltage
- it **takes many <u>bits</u> to <u>represent reasonable amounts of information</u>**