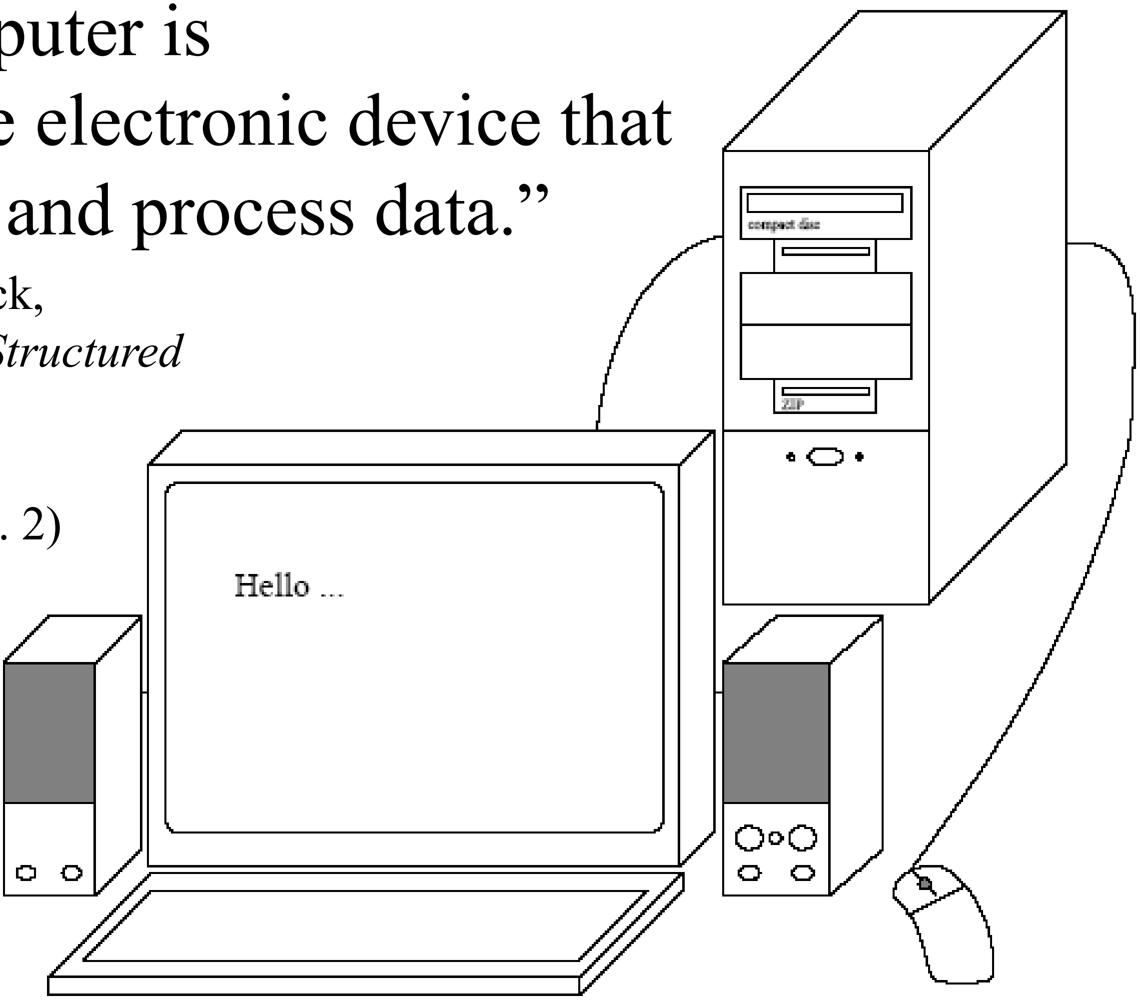
Hardware Outline

What is a Computer?

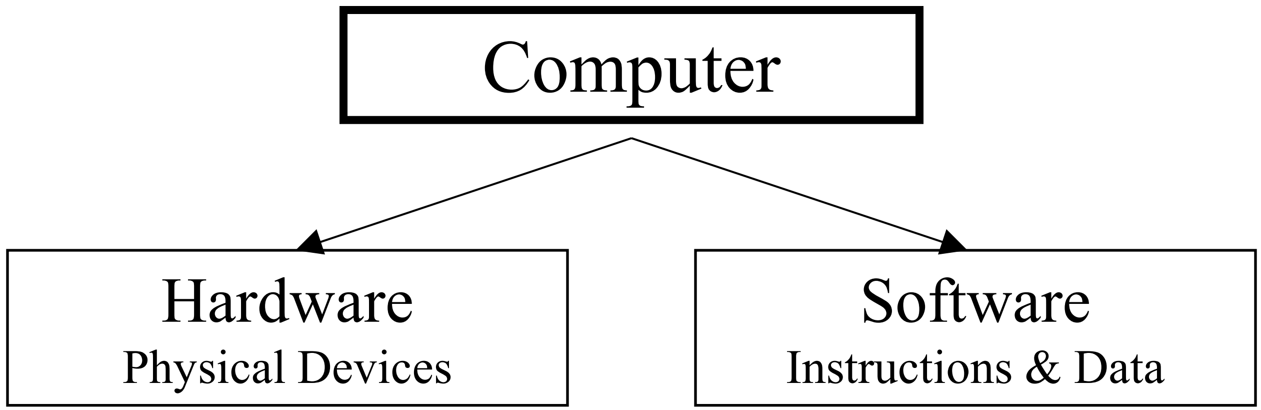
A computer is

"… [A] programmable electronic device that can store, retrieve and process data."

(N. Dale & D. Orshalick, *Introduction to PASCAL and Structured Design*, D.C. Heath & Co., Lexington MA, 1983, p. 2)



Components of a Computer



DON’T PANIC!

This discussion may be confusing at the moment;
it’ll make more sense after you’ve written a few programs.

Henry's Laptop

Dell Latitude 5430[4]

https://i.dell.com/is/image/DellContent/content/dam/ss2/product-images/dell-client-products/notebooks/latitude-notebooks/14-5430/pdp/laptop-latitude-14-5430-pdp-hero-504x350.psd?qlt=95&fit=constrain,1&hei=500&fmt=jpg&wid=570&hei=400

- Intel Core i5-1235U 1.30 GHz ("Alder Lake"), 10 cores:
  - 2 performance cores @ 4.4 GHz
  - 8 efficiency cores @ 3.3 GHz
- 32 GB 3200 MHz DDR4 SDRAM
- 512 GB SSD M.2
- 1 Gbps Ethernet Adapter, WiFi

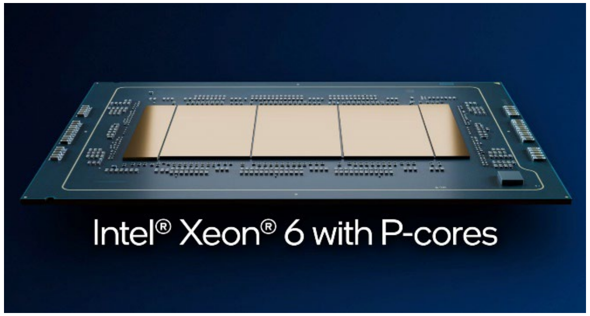
Categories of Computer Hardware

- ***Central Processing Unit*** (CPU)
- ***Storage***
  - ***Primary***: Cache, RAM
  - ***Secondary***: Hard disk, removable (e.g., USB thumb drive)
- ***I/O***
  - ***Input*** Devices
  - ***Output*** Devices

Central Processing Unit (CPU)

The ***Central Processing Unit*** (CPU), also called the ***processor***, is the "**brain**" of the computer.

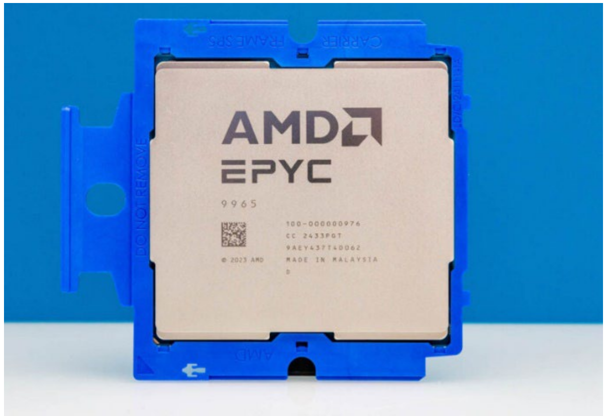
Intel Granite Rapids exterior
https://pbs.twimg.com/media/GYQMs6GakAMrvqN?format=jpg&name=4096x4096



AMD EPYC Turin exterior
https://www.servethehome.com/wp-content/uploads/2024/10/AMD-EPYC-9965-Front-2-800x551.jpg

Intel Emerald Rapids innards

https://substackcdn.com/image/fetch/f_auto,q_auto:good,fl_progressive:steep/https%3A%2F%2Fsubstack-post-media.s3.amazonaws.com%2Fpublic%2Fimages%2Ffe95ad57-d43e-4890-9499-20e2b808e608_6096x4986.jpeg



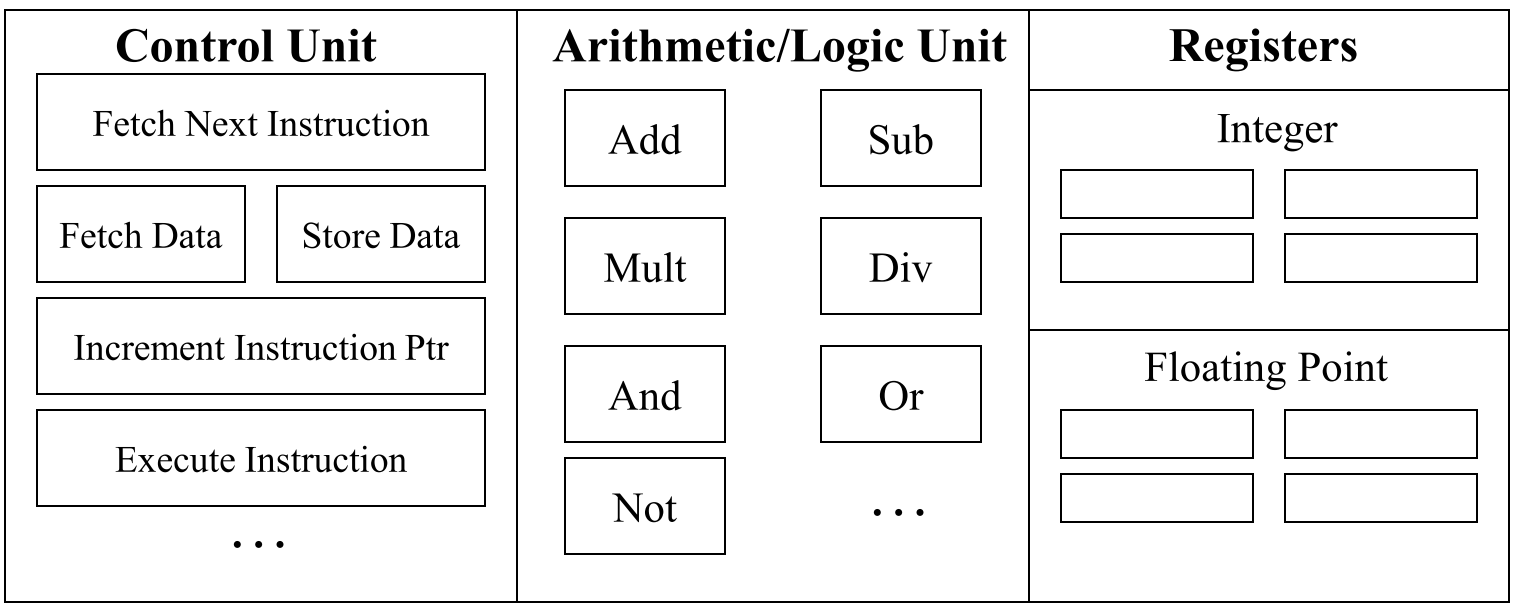
CPU Examples

- ***x86***: Intel Celeron/Pentium/Core/Xeon and AMD Ryzen/Threadripper/EPYC (and related models from smaller manufacturers) (Windows, MacOS and Linux PCs; some Android tablets)
  http://en.wikipedia.org/wiki/X86
  Market Share for PCs: Intel 78%, AMD 12.5%, ARM: Apple M2/M3 9%, other ARM 0.5%
  https://hardwaretimes.com/intel-pc-cpu-market-share/
- ***ARM*** (99% of smartphones, 9.5% of laptop/desktop PCs)
  http://en.wikipedia.org/wiki/ARM_processor
- ***IBM POWER10*** (servers)
  https://en.wikipedia.org/wiki/Power_ISA
  https://en.wikipedia.org/wiki/Power10

CPU Parts

The CPU consists of three main parts:

- Control Unit
- Arithmetic/Logic Unit
- Registers



CPU: Control Unit

The CPU's ***Control Unit*** decides what to do next.

For example:

- ***memory operations***: for example,
  - ***load*** data from ***main memory*** (RAM) into the ***registers***;
  - ***store*** data from the registers into main memory;
- ***arithmetic/logical operations***: e.g., add, multiply;
- ***branch***: choose among several possible courses of action.

CPU: Arithmetic/Logic Unit

The CPU's ***Arithmetic/Logic Unit*** (ALU) performs arithmetic and logical operations.

- ***Arithmetic* operations**: for example,
  add, subtract, multiply, divide, square root, cosine, etc.
- ***Logical* operations**: for example,
  compare two numbers to see which is greater,
  check whether both of a pair of true/false statements are true,
  etc.

CPU: Registers

Registers are memory-like locations inside the CPU where data and instructions reside that are **being used right now**.

That is, registers hold the operands being used by the current arithmetic or logical operation, and/or the result of the arithmetic or logical operation that was just performed.

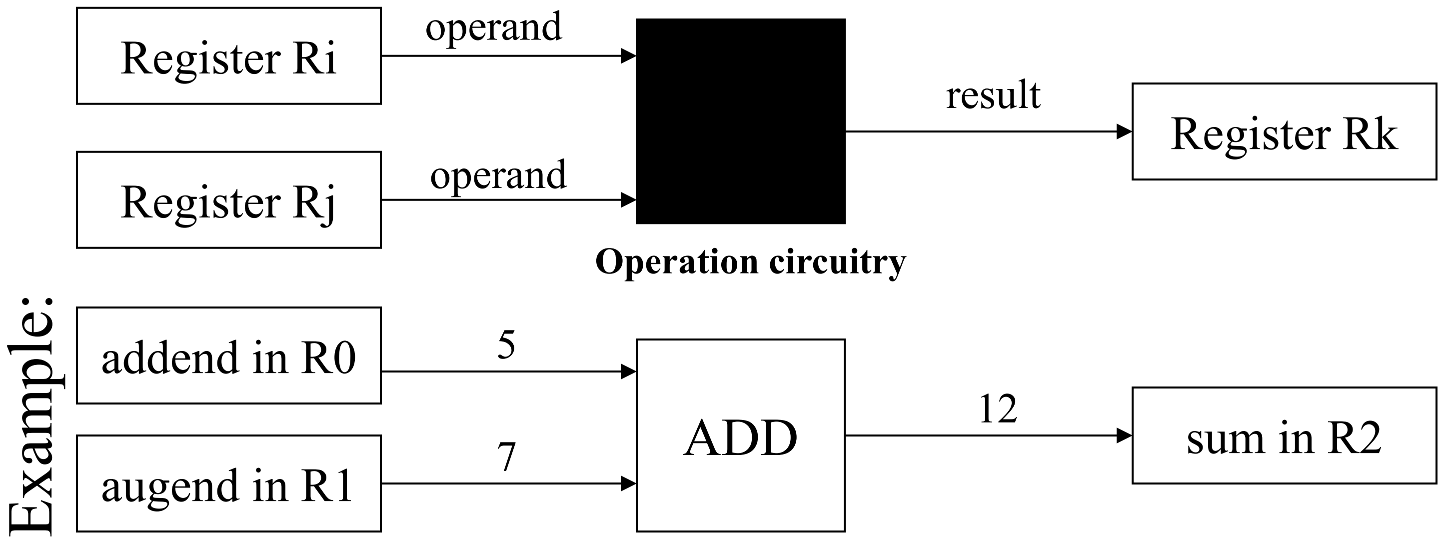
For example, if the CPU is adding two numbers, then

- the addend is in some register;
- the augend is in another register;
- after the addition is performed, the sum shows up in yet another register.

A typical CPU has only **a few hundred to a few thousand bytes** of registers.

How Registers Are Used

- Every arithmetic or logical operation has one or more operands and one result.
- Operands are contained in registers (“source”).
- A “black box” of circuits performs the operation.
- The result goes into a register (“destination”).



Multicore

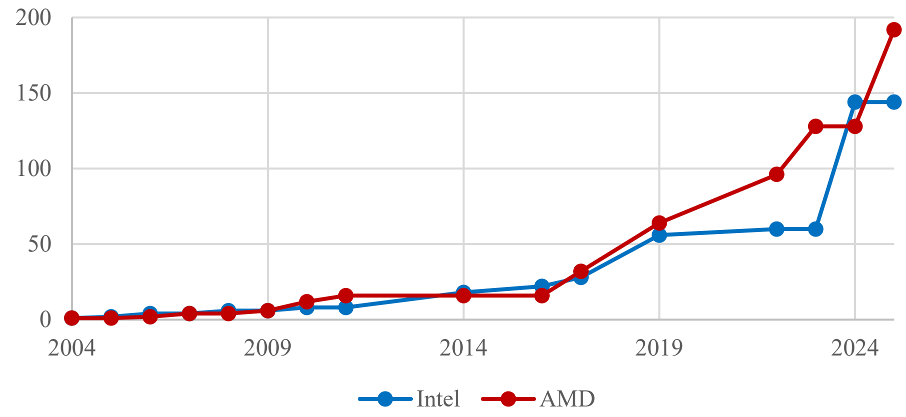
- A ***multicore*** CPU is a chip with multiple, independent "brains," known as ***cores***.
- These multiple cores can run completely separate programs, or they can cooperate together to work simultaneously in parallel on different parts of the same program.
- All of the cores share the same connection to memory – and the same ***bandwidth*** (memory speed).

Intel Emerald Rapids innards

https://substackcdn.com/image/fetch/f_auto,q_auto:good,fl_progressive:steep/https%3A%2F%2Fsubstack-post-media.s3.amazonaws.com%2Fpublic%2Fimages%2Ffe95ad57-d43e-4890-9499-20e2b808e608_6096x4986.jpeg

Multicore History (x86)

- Single core: November 1971 (Intel 4004)
- Dual core: October 2005 (Intel), March 2006 (AMD)
- Quad core: June 2006 (Intel), Sep 2007 (AMD)
- Hex core: Sep 2008 (Intel), June 2009 (AMD)
- Oct core (Intel & AMD): March 2010
- 12-core (AMD only): March 2010
- 16-core: Nov 2011 (AMD only)
- 18-core: Sep 2014 (Intel only)
- 22-core: March 2016 (Intel only)
- 28-core: July 2017 (Intel only)
- 32-core: June 2017 (AMD only)
- 56-core: Apr 2019 (Intel only)
- 64-core: Aug 2019 (AMD only)
- 96-core: Nov 2022 (AMD only)
- 128-core: June 2023 (AMD only)
- 144-core: June 2024 (Intel only)
- 192-core: Oct 2024 (AMD only)



Note that this is only for x86 – other processor families (for example, POWER) introduced multicore earlier.

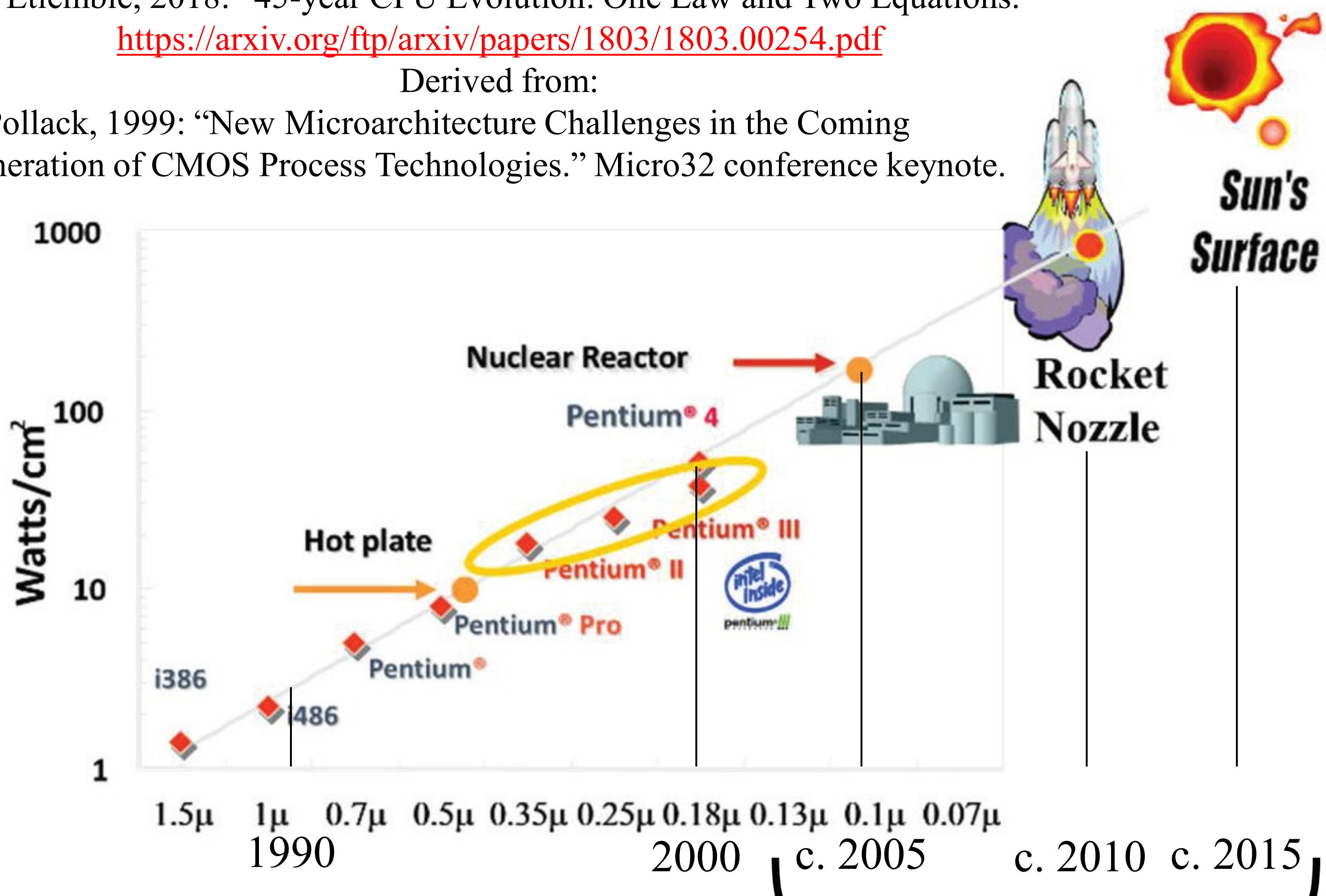
http://www.intel.com/pressroom/kits/quickreffam.htm (dual core, quad core)
http://ark.intel.com/products/family/34348/Intel-Xeon-Processor-7000-Sequence#@Server (hex core)
http://ark.intel.com/ProductCollection.aspx?familyID=594&MarketSegment=SRV (oct core)
http://en.wikipedia.org/wiki/Intel_Nehalem_(microarchitecture) (oct core)
http://en.wikipedia.org/wiki/AMD_Opteron (12-core)
https://en.wikipedia.org/wiki/Broadwell_(microarchitecture) (22-core)
https://en.wikipedia.org/wiki/Skylake_(microarchitecture) (28-core)
https://en.wikipedia.org/wiki/Cascade_Lake_(microarchitecture) (56-core)
https://en.wikipedia.org/wiki/Sierra_Forest (144-core)
https://en.wikipedia.org/wiki/Epyc (32-core, 64-core, 96-core, 128-core, 192-core)

Why Multicore? #1

- In the golden olden days (through about 2005), the way to speed up a CPU was to increase its "clock speed."
  - Every CPU has a little crystal inside it that vibrates at a fixed frequency (for example, 1 GHz = 1 billion vibrations per second).
  - Each operation (add, subtract, multiply, divide, etc) requires a specific number of clock ticks to complete.
- But, the power density (watts per square centimeter) of a CPU chip is proportional to the **square** of the clock speed.
- So, continuing to increase the clock speed would have been, quite literally, a dead end, because by now such CPU chips would have already exceeded the power density of the sun.

Why Multicore? #2

D. Etiemble, 2018: “45-year CPU Evolution: One Law and Two Equations.”
https://arxiv.org/ftp/arxiv/papers/1803/1803.00254.pdf
Derived from:
F. Pollack, 1999: “New Microarchitecture Challenges in the Coming Generation of CMOS Process Technologies.” Micro32 conference keynote.



Storage

There are two major categories of storage:

- ***Primary***
  - Cache
  - Main memory (RAM)
- ***Secondary***
  - Hard disk
  - Removable (e.g., thumb drive, CD, floppy)

Primary Storage

Primary storage is where data and instructions reside when they're **being used by a program that is currently running**.

- Typically is ***volatile***: The data disappear when the power is turned off.
- Typically comes in two subcategories:
  - Cache
  - Main memory (RAM)

Cache

Cache memory is where data and instructions reside when they are **going to be used very very soon**, or have just been used.

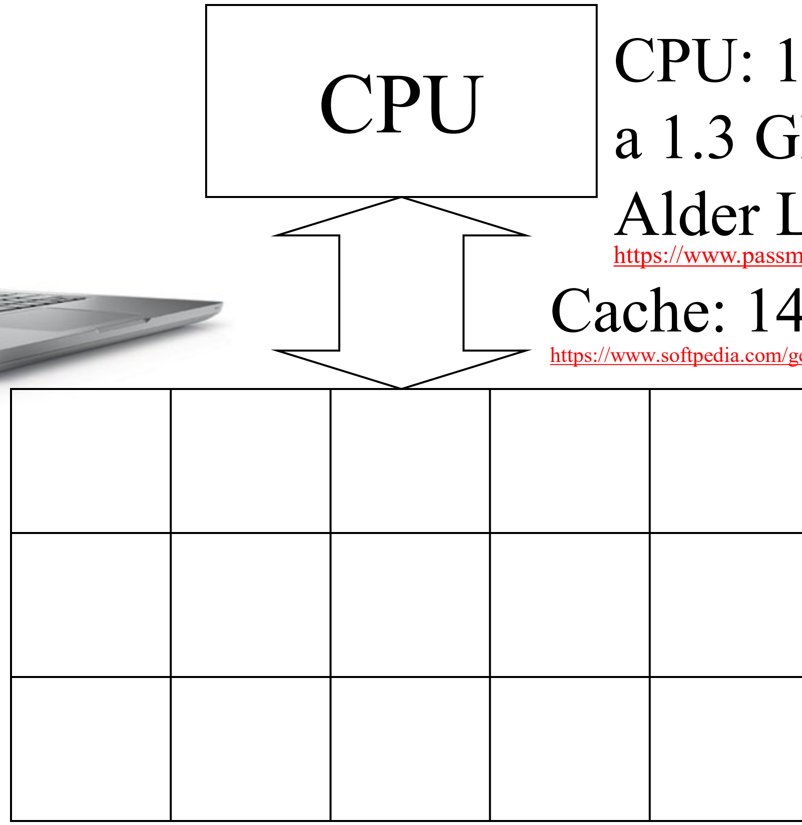
- Cache is **very fast** (typically multiple percent of the speed of the registers) compared to RAM (< 1% of the speed of the registers).
- Therefore, it's **very expensive** (e.g., $2.34 per MB):
  https://www.amd.com/en/products/processors/desktops/ryzen/5000-series/amd-ryzen-7-5800x3d.html ($449 MSRP, 96 MB)
  https://www.amd.com/en/support/downloads/drivers.html/processors/ryzen/ryzen-5000-series/amd-ryzen-7-5700x.html ($299 MSRP, 32 MB)
  https://en.wikipedia.org/wiki/List_of_AMD_Ryzen_processors

  Therefore, cache is **very small** (from less than 1 MB to 1152 MB)
  https://en.wikipedia.org/wiki/Epyc

  … but still **much bigger than registers**
  (which range from less than 1 KB to a few KB).

From Cache to the CPU

https://i.dell.com/is/image/DellContent/content/dam/ss2/product-images/dell-client-products/notebooks/latitude-notebooks/14-5430/pdp/laptop-latitude-14-5430-pdp-hero-504x350.psd?qlt=95&fit=constrain,1&hei=500&fmt=jpg&wid=570&hei=400



CPU: 13,517 GB/sec on a 1.3 GHz Intel i5-1235U Alder Lake
https://www.passmark.com/products/performancetest/

Cache: 142 GB/sec (1%)
https://www.softpedia.com/get/System/Benchmarks/BenchMem.shtml

Cache

Typically, data move between cache and the CPU at speeds closer to that of the CPU performing calculations.

Main Memory (RAM)

Main memory (RAM) is where data and instructions reside when a **program that is currently running** is going to use them at some point during the run (whether soon or not).

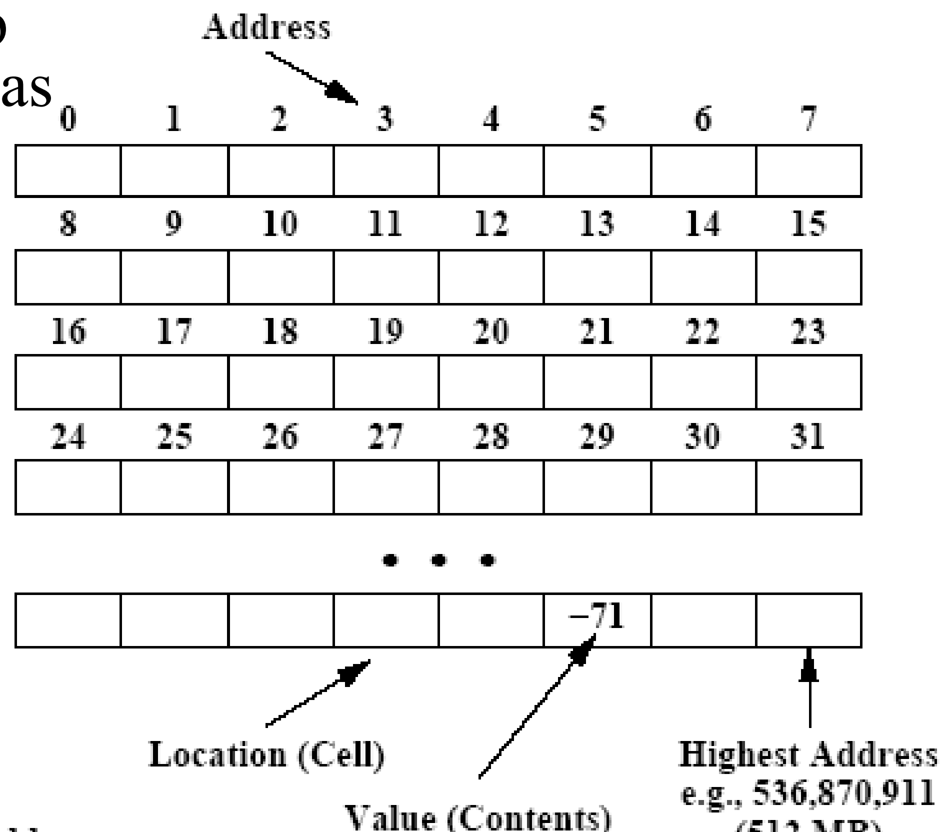
- **Much slower** than cache
  (e.g., less than 1% of CPU speed for RAM, versus
  ~1-100% of CPU speed for cache)
- Therefore, **much cheaper** than cache
  (e.g., ~$0.005/MB for RAM versus $2.34/MB for cache).
  http://www.pricewatch.com/, http://www.ebay.com/,
  http://www.crucial.com/usa/en/compatible-upgrade-for/Dell/latitude-e5540
- Therefore, **much larger** than cache – for example,
  1 GB (1024 MB) to 32 TB (~32M MB) for RAM,
  versus under 1 MB to 1152 MB for cache.
  https://en.wikipedia.org/wiki/Epyc

Main Memory Layout

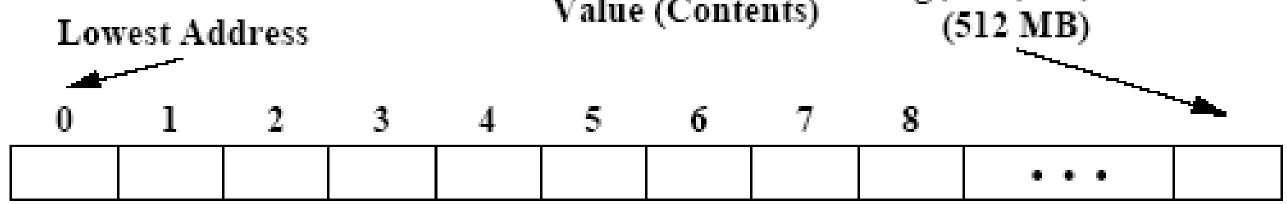
Main memory is made up of ***locations***, also known as ***cells***.

Each location has a unique integer ***address*** that never changes.

Each location has a ***value*** – also known as the ***contents*** – that the CPU can look at and change.



We can think of memory as one ***contiguous*** line of cells. →



RAM versus ROM

RAM: **Random Access Memory**

- Memory that the CPU can look at and change arbitrarily (i.e., can load from or store into any location at any time, not just in a sequence).
- We often use the terms **Main Memory**, **Memory** and **RAM** interchangeably.
- Sometimes known as ***core*** memory, named for an older memory technology. (Note that this use of the word "core" is unrelated to "multi-core.")

ROM: **Read Only Memory**

- Exactly like RAM, except no one can change its values.

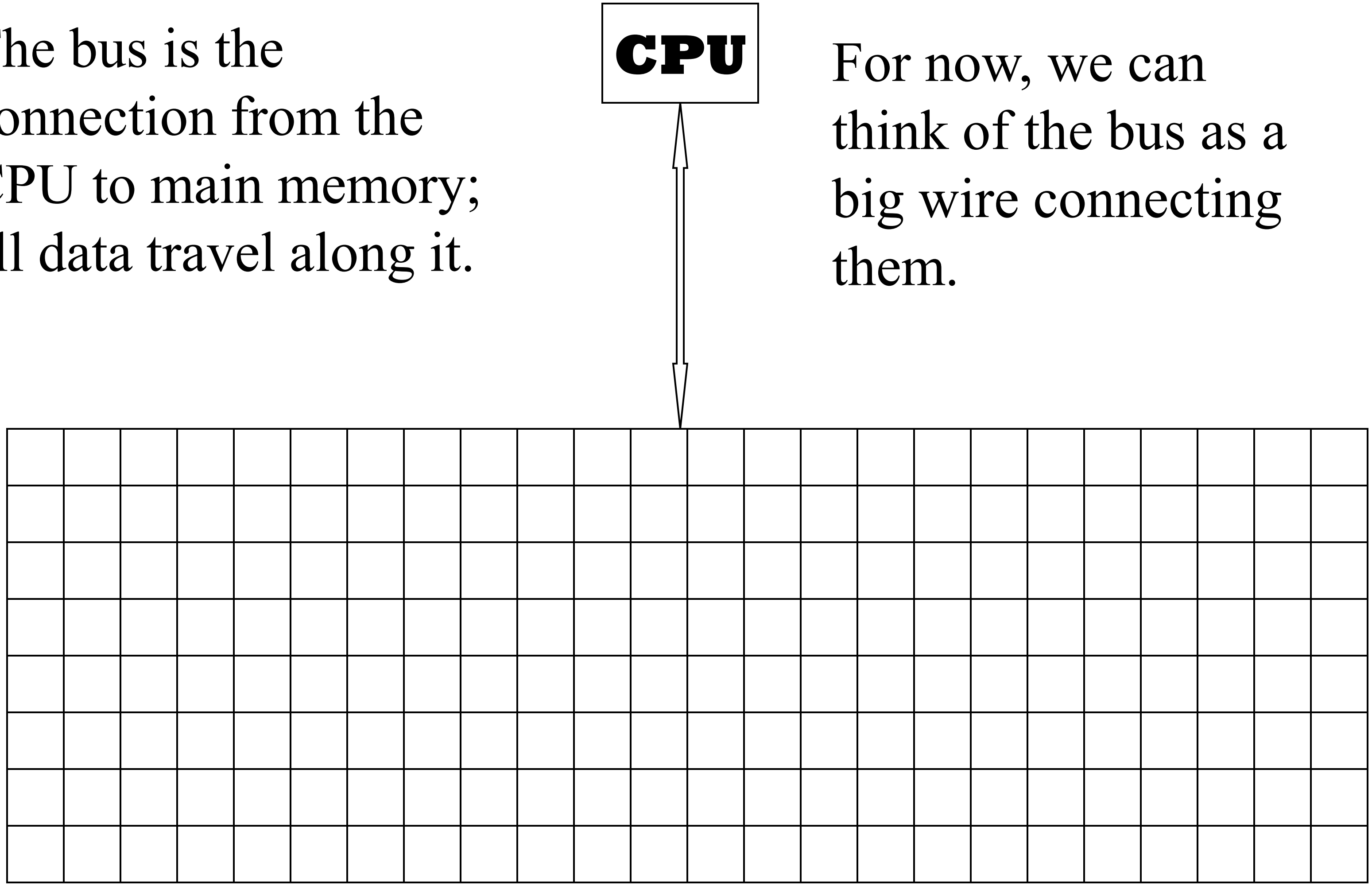
Speed => Price => Size

- **Registers** are **VERY fast**, because they are etched directly into the CPU.
- **Cache** is also **very fast**, because it's also etched into the CPU, but it isn't directly connected to the Control Unit or Arithmetic/Logic Unit in the same way as registers. Cache operates at speeds similar to registers, but cache is **MUCH bigger** than the collection of registers (typically on the order of 1,000 to 100,000 times as big).
- **Main memory (RAM)** is **much slower** than cache, because it isn't part of the CPU; therefore, it's **much cheaper** than cache, and therefore it's **much bigger** than cache (for example, 1000 times as big).

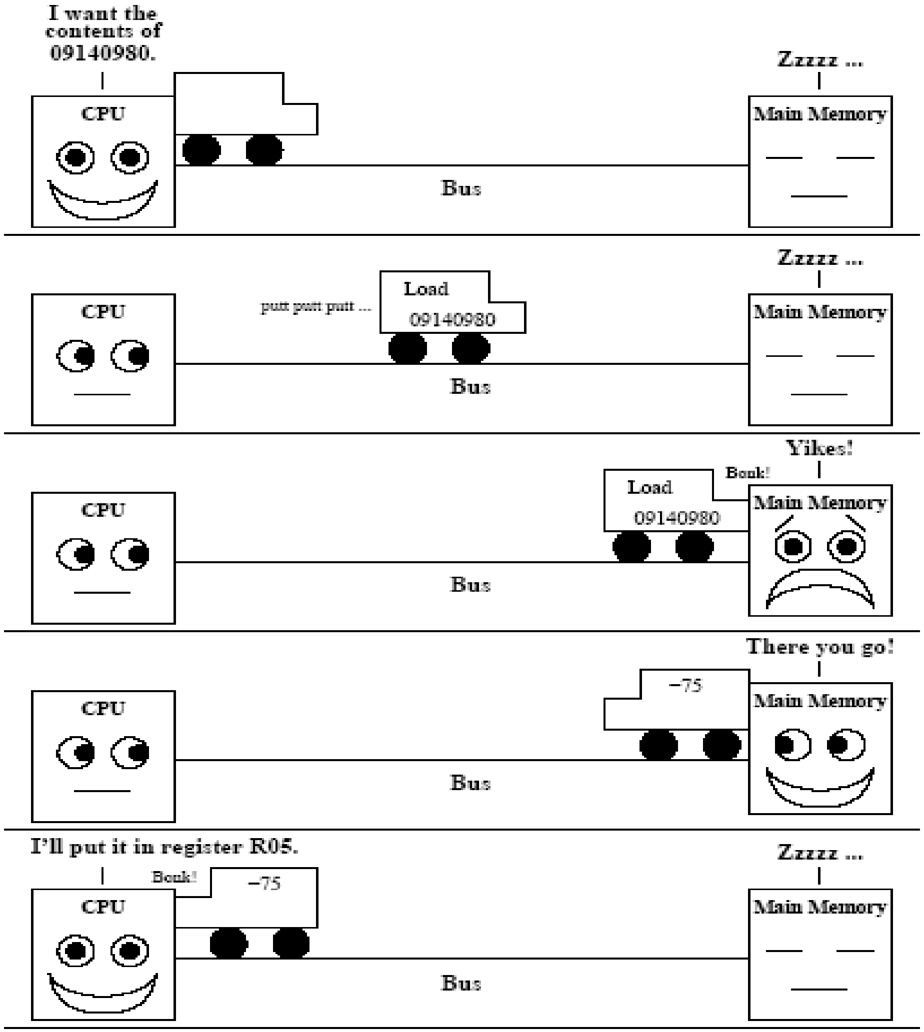
How Data Travel Between RAM and CPU

The bus is the connection from the CPU to main memory; all data travel along it.

For now, we can think of the bus as a big wire connecting them.

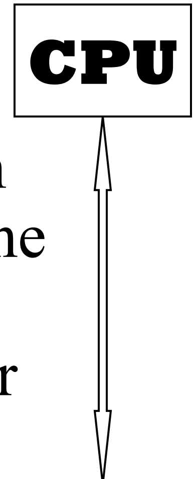


Loading Data from RAM into the CPU



RAM is Slow

The speed of data transfer between Main Memory and the CPU is much slower than the speed of calculating, so the CPU spends most of its time waiting for data to come in or go out.



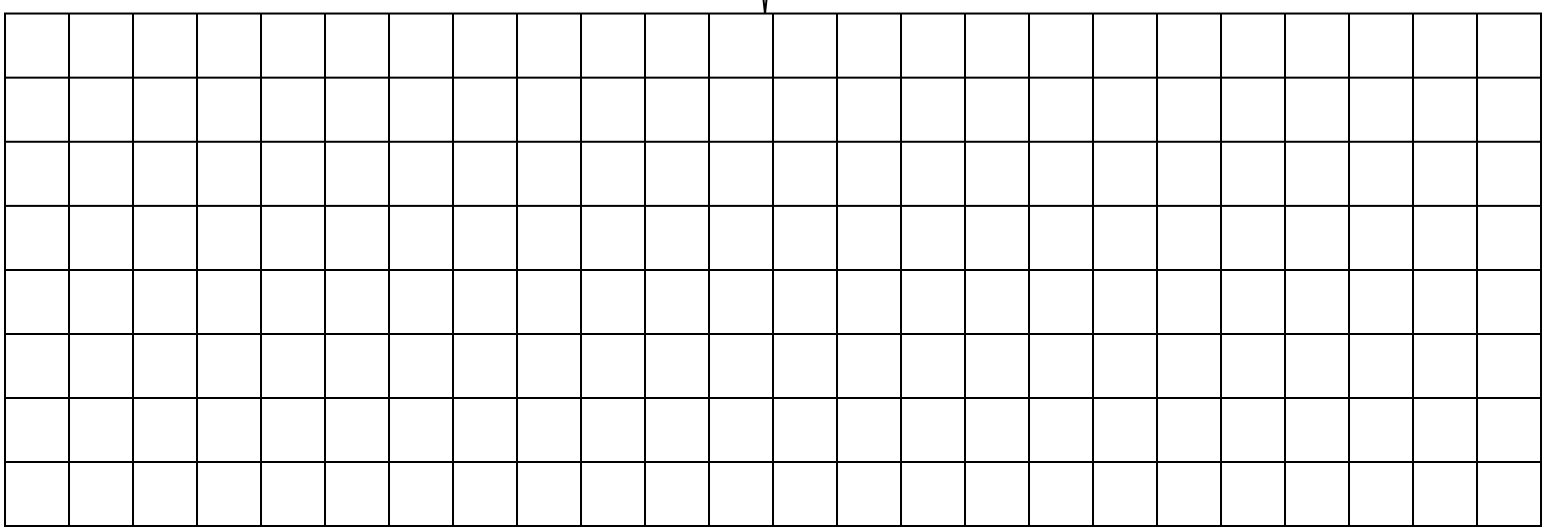
CPU: 13,517 GB/sec on a 1.3 GHz Intel i5-1235U Alder Lake

http://www.aida64.com

Bottleneck

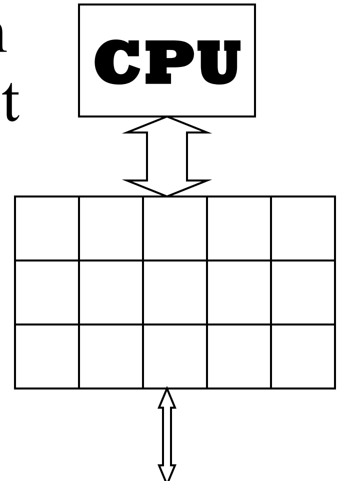
RAM: 21 GB/sec (0.02%)

http://www.aida64.com



Why Have Cache?

Cache is much faster than RAM, so the CPU doesn't have to wait nearly as long for stuff that's already in cache: it can do more operations per second!



CPU: 13,517 GB/sec on a 1.3 GHz Intel i5-1235U Alder Lake

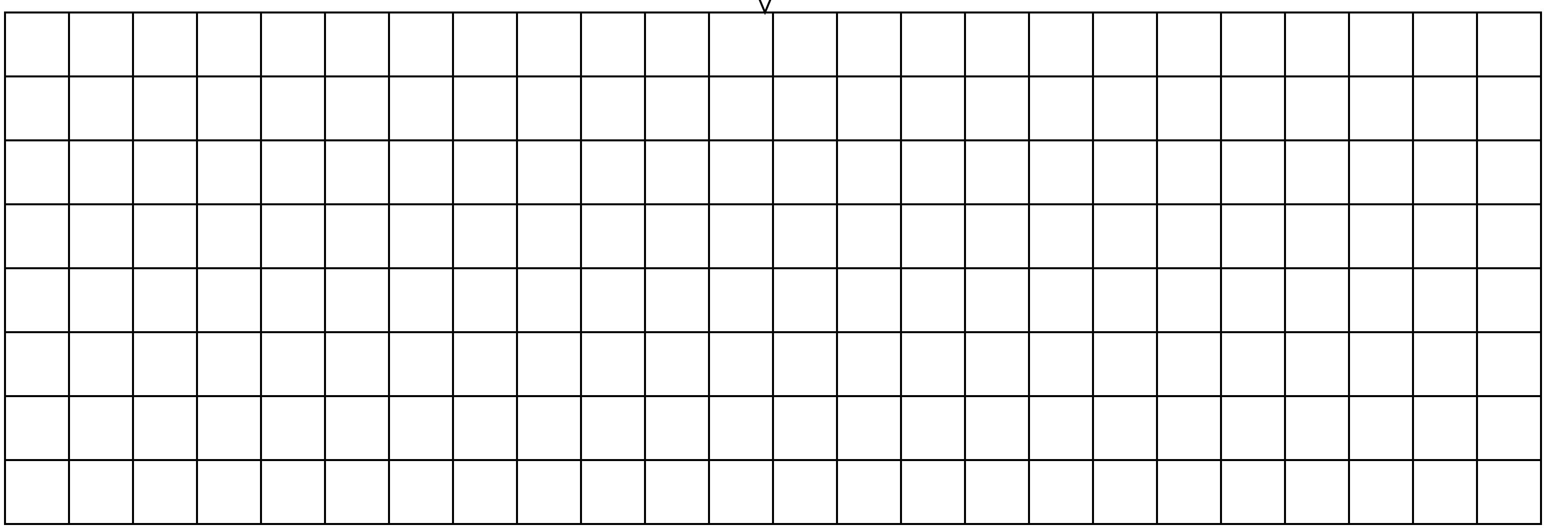
http://www.aida64.com

Cache: 142 GB/sec (1%)

http://www.aida64.com

RAM: 21 GB/sec (0.02%)

http://www.aida64.com



Multiple Levels of Cache

- Nowadays, most CPUs have multiple levels of cache.
- For example, Henry's laptop has:
  - Registers (for comparison): 13,517 GB/sec
  - L1 cache: 48 KB/P-core, 32KB/E-core, 1052 GB/sec (7.8% of register speed)
  - L2 cache: 1,280 KB per core, 171 GB/sec (1.3% of register speed)
  - L3 cache: 12,288 KB shared by all cores, 142 GB/sec (1.0% of register speed)
  - RAM (for comparison): 21 GB/sec (0.02% of register speed)

https://en.wikipedia.org/wiki/Alder_Lake
http://www.aida64.com

- So, the goal is to get the data you need into the fastest (but therefore the smallest) cache by the time you need it.

Secondary Storage

- Where data and instructions reside that are going to be used **<u>in the future</u>**
- ***<u>Nonvolatile</u>***: data don't disappear when power is turned off.
- **<u>Much slower</u>** than RAM, therefore **<u>much cheaper</u>**, therefore **<u>much larger.</u>**
- Other than the internal disk drive, most are ***<u>portable</u>***: they can be easily removed from your computer and taken to someone else's.

Media Types

- ***Solid State*** (for example, flash drive)
  - Always can be read
  - Always can be written and rewritten multiple times
  - Contents don't degrade much over time
  - Can't be erased by magnets
- ***Magnetic*** (for example, spinning disk drive)
  - Always can be read
  - Always can be written and rewritten multiple times
  - Contents degrade relatively rapidly over time
  - Can be erased by magnets
- ***Optical*** (for example, DVD)
  - Always can be read
  - Some can be written only once, some can be rewritten multiple times
  - Contents degrade more slowly than magnetic media
  - Can't be erased by magnets
- **Paper**: forget about it!

Speed, Price, Size

Medium	Speed (MB/sec)	Size (MB)	Media Type	Can write to it?	Port-able?	Pop-ular?	Drive cost ($)	Media cost ($/MB)
Cache	145,600	12	L1/L2/L3	Y	N	Req'd		$2.340000
RAM	21,000	33,554,432	DDR4	Y	N	Req'd		$0.003000
Solid State Drive	500	100,000,000	Solid	Y	Y	Y		$0.000100
USB 3 Thumb	380	2,000,000	Solid	Y	Y	Y		$0.000300
Hard Disk	100	32,000,000	Mag	Y	N	Y		$0.000014
Blu-ray	72	50,000	Opt	Y	Y	N	$70	$0.000038
DVD±RW	32	4,700	Opt	Y	Y	N	$13	$0.000100
CD-RW	7.8	700	Opt	Y	Y	N	$12	$0.000900
Mag tape	400	18,000,000	Mag	Y	Y	N	$4455	$0.000004
Floppy	0.03	1.44	Mag	Y	Y	N	$17	$0.900000
Audio Cassette	<< 1	<< 1	Mag	Y	Y	Historical		
Paper tape	<< 1	<< 1	Paper	Y	Y	Historical		
Punch card	<< 1	<< 1	Paper	Y	Y	Historical		

* Maximum among models commonly available for PCs

Note: All numbers are approximate as of Aug 2021 (amazon.com, bestbuy.com, cendyne.com, creativelabs.com, dell.com, ebay.com, floppydisk.com, nextag.com, nimbusdata.com, pcworld.com, rakuten.com, seagate.com, sony.com, toshiba.com, walmart.com, wikipedia.org). Tape is LTO-9.

Why Is Tape So Much Cheaper Than Thumb?

Magnetic tape and thumb drives (USB 3 flash) have roughly the same speed (400 MB/sec).

So why is tape so much cheaper per MB?

There are actually two measures of storage (and network) speed:

- ***Bandwidth***: bits per second
- ***Latency***: the time it takes for the first bit to arrive at the destination

Magnetic tape and thumb drives have the same **bandwidth**.

But for **latency**, magnetic tape averages about a minute (rewind/fast forward), whereas a thumb drive typically takes less than a millisecond.

CD-ROM/DVD-ROM/BD-ROM

When a CD or DVD or Blu-ray holds data instead of music or a movie, it acts very much like Read Only Memory (ROM):

- it can only be read from, but not written to;
- it’s nonvolatile;
- it can be addressed essentially arbitrarily (it’s not actually arbitrary, but it’s fast enough that it might as well be).

CD-ROM/DVD-ROM/BD-ROM: Disadvantage

Disadvantage of CD-ROM/DVD-ROM/BD-ROM compared to ROM:

- **Speed**: CD-ROM/DVD-ROM/BD-ROM are **much slower** than ROM:
  - CD-ROM is 7.8 MB/sec (peak);
    DVD-ROM is 32 MB/sec;
    BD-ROM is 72 MB/sec.
  - Most ROM these days is 10-300 GB/sec (hundreds or thousands of times as fast as secondary storage).

CD-ROM & DVD-ROM: Advantages

Advantages of CD-ROM/DVD-ROM compared to ROM:

- **Price**: CD-ROM and DVD-ROM are **much cheaper** than ROM.
  - Blank BD-REs are roughly $0.00004 per MB (rewritable BluRay);
    blank DVD-RWs are roughly $0.00010 per MB;
    blank CD-RWs are roughly $0.00090 per MB.
  - ROM is even more expensive than RAM (which is ~$0.005/MB), because it has to be programmed special (with "firmware").
- **Size**: CD-ROM and DVD-ROM are **much larger** – they can have **arbitrary amount of storage** (on many CDs or DVDs);
  ROM is limited to a few GB.

Floppy Disk

https://www.boredpanda.com/blog/wp-content/uploads/2023/05/920406104911233024-png__700.jpg

Why Are Floppies So Expensive Per MB?

BD-REs (rewritable blank Blurays) cost roughly $0.00004 per MB, but floppy disks cost about $0.90 per MB, more than 20,000 times as expensive per MB as Blu-Ray. **Why?**

Well, an individual BD-RE has **much greater capacity** than an individual floppy (25-100 GB versus 1.44 MB), and the costs of manufacturing the actual physical objects are similar.

And, because floppies are much less popular than CDs, they aren't manufactured in high quantities – so it's harder to *amortize* the high fixed costs of running the factory.

So, the cost of a floppy **per MB** is much higher.

I/O

We often say ***I/O*** as a shorthand for **"Input/Output."**

I/O: Input Devices

We often say ***I/O*** as a shorthand for **"Input/Output."**

Input Devices transfer data into computer (e.g., from a user into memory).

For example:

- Keyboard
- Mouse
- Scanner
- Microphone
- Touchpad
- Joystick

I/O: Output Devices

We often say ***I/O*** as a shorthand for **"Input/Output."**

Output Devices transfer data out of computer (for example, from memory to a user).

For example:

- Monitor
- Printer
- Speakers

NOTE: A device can be **both** input and output – for example, a touchscreen.

Bits

Bit (**B**inary dig**IT**)

- Tiniest possible piece of memory.
- Made of teeny tiny transistors wired together (the most recent are smaller than 10 nanometers)
- Has 2 possible values that we can think of in several ways:
  - **Low or High**: Voltage into transistor
  - **Off or On**: Conceptual description of transistor state
  - **False or True**: ***Boolean*** value for symbolic logic
  - **0 or 1**: Integer value
- Bits aren't individually ***addressable***:
  the CPU can't load from or store to an individual bit of memory.

Bytes

Byte: a sequence of 8 contiguous bits (typically)

- On most ***platforms*** (kinds of computers),
  a byte is the smallest ***addressable*** piece of memory: typically, the CPU can load from, or store into, an individual byte.
- Possible integer values: 0 to 255 or -128 to127
  (to be explained later)
- Can also represent a character
  (for example, letter, digit, punctuation; to be explained later)

Contiguous: one after the other, abutting.

Words

Word: a sequence of 4 or 8 contiguous bytes (typically);
that is, 32 or 64 contiguous bits

- Standard size for storing a **number** (integer or real)
- Standard size for storing an **address** (special kind of integer)

Contiguous: One after the other, abutting.

Putting Bits Together

1 bit: $2^1 = 2$ possible values: 0 or 1

2 bits: $2^2 = 4$ possible values

0 0	0 1	1 0	1 1

3 bits: $2^3 = 8$ possible values

0 0 0	0 0 1	0 1 0	0 1 1
1 0 0	1 0 1	1 1 0	1 1 1

Putting Bits Together (cont'd)

4 bits: 2^4 = 16 possible values

…

8 bits: 2^8 = 256 possible values

…

10 bits: 2^{10} = 1024 possible values

…

16 bits: 2^{16} = 65,536 possible values

…

32 bits: 2^{32} = 4,294,967,296 possible values

(typical size of an integer in most computers today)

Powers of 2

2^0 =	1	2^{11} =	2,048
2^1 =	2	2^{12} =	4,096
2^2 =	4	2^{13} =	8,192
2^3 =	8	2^{14} =	16,384
2^4 =	16	2^{15} =	32,768
2^5 =	32	2^{16} =	65,536
2^6 =	64	2^{17} =	131,072
2^7 =	128	2^{18} =	262,144
2^8 =	256	2^{19} =	524,288
2^9 =	512	2^{20} =	1,048,576 (about a million)
2^{10} =	1,024 (about a thousand)		

Powers of 2 versus Powers of 10

A rule of thumb for comparing powers of 2 to powers of 10:

$2^{10} \approx 10^3$ (that is, $1024 \approx 1000$)

So:

- $2^{10} \approx 1{,}000$ (thousand)
- $2^{20} \approx 1{,}000{,}000$ (million)
- $2^{30} \approx 1{,}000{,}000{,}000$ (billion)
- $2^{40} \approx 1{,}000{,}000{,}000{,}000$ (trillion)
- $2^{50} \approx 1{,}000{,}000{,}000{,}000{,}000$ (quadrillion)
- $2^{60} \approx 1{,}000{,}000{,}000{,}000{,}000{,}000$ (quintillion)

The fastest supercomputer in the world today can do ~2.75 quintillion calculations per second (top500.org).
(OU's supercomputer can do ~2.1 quadrillion calculations/second.)

KB, MB, GB, TB, PB

Kilobyte (KB): 2^{10} bytes, which is approximately 1,000 bytes (thousand)

Megabyte (MB): 2^{20} bytes, which is approximately 1,000,000 bytes (million)

Gigabyte (GB): 2^{30} bytes, which is approximately 1,000,000,000 bytes (billion)

Terabyte (TB): 2^{40} bytes, which is approximately 1,000,000,000,000 bytes (trillion)

Petabyte (PB): 2^{50} bytes, which is approximately 1,000,000,000,000,000 bytes (quadrillion)

Kilo, Mega, Giga, Tera, Peta

Kilobyte (KB): 2^{10} bytes = 1,024 bytes $\simeq$ 1,000 bytes

Approximate size: one e-mail (plain text)

Desktop Example: TRS-80 w/4 KB RAM (1977)

Megabyte (MB): 2^{20} bytes = 1,048,576 bytes $\simeq$ 1,000,000 bytes

Approximate size: 30 phonebook pages

Desktop Example: IBM PS/2 PC w/1 MB RAM (1987)

Gigabyte (GB): 2^{30} bytes = 1,073,741,824 bytes $\simeq$ 1,000,000,000 bytes

Approximate size: 15 copies of the OKC white pages

Desktop: c. 1997

Terabyte (TB): 2^{40} bytes = 1,099,511,627,776 bytes $\simeq$ 1,000,000,000,000 bytes

Approximate size: 5,500 copies of a phonebook listing everyone in the world

Desktop: Example: Dell T630 workstation (2014)

Petabyte (PB): 2^{50} bytes $\simeq$ 1,000,000,000,000,000 bytes

Desktop: ???

(Largest individual server today: 32 TB = 1/32 of a PB) https://www.supermicro.com/en/products/system/mp/6u/sys-681e-tr

EB, ZB, YB

- ***Exabyte*** (EB): 2^{60} bytes, which is approximately 1,000,000,000,000,000,000 bytes (quintillion)
  (Global annual Internet traffic reached ~1 EB in 2001, monthly in 2004, daily in 2013.)
- ***Zettabyte*** (ZB): 2^{70} bytes, which is approximately 1,000,000,000,000,000,000,000 bytes (sextillion)
  (By 2012, total worldwide digital data was ~1 ZB;
  by 2016, annual Internet traffic was ~1 ZB.)
- ***Yottabyte*** (YB): 2^{80} bytes, which is approximately 1,000,000,000,000,000,000,000,000 bytes (septillion)
  (At current growth rates, by 2030, total worldwide digital data will be ~1 YB;
  1 YB ≈ 1400 metric tons of DNA.)

http://en.wikipedia.org/wiki/Exabyte
https://www.iea.org/reports/digitalisation-and-energy
https://en.wikipedia.org/wiki/Zettabyte_Era
https://biwintechnology.com/the-coming-yottabyte-era/
http://www.extremetech.com/extreme/134672-harvard-cracks-dna-storage-crams-700-terabytes-of-data-into-a-single-gram

10^{3x} versus 2^{10x}

In theory, there are different words for 10^{3x} versus 2^{10x}, but in practice approximately nobody uses them:

- Kilobyte (KB): 10^3 bytes
- Megabyte (MB): 10^6 bytes
- Gigabyte (GB): 10^9 bytes
- Terabyte (TB): 10^{12} bytes
- etc

- Kibibyte (KiB): 2^{10} bytes
- Mebibyte (MiB): 2^{20} bytes
- Gibibyte (GiB): 2^{30} bytes
- Tebibyte (TiB): 2^{40} bytes
- etc

Just about everybody uses KB/MB/GB/TB for both 10^{3x} and 2^{10x}, so you have to use context to tell which they mean:

- Primary storage is usually expressed as 2^{10x} (but called GB/TB/PB).
- Secondary storage is usually expressed as 10^{3x}.

https://en.wikipedia.org/wiki/Byte#Multiple-byte_units

Moore's Law

Moore's Law: Computing speed and capacity double every 24 months.

In 1965 Gordon Moore (Chairman Emeritus, Intel Corp) observed the "doubling of transistor density on a manufactured die every year."

People have noticed that computing speed and capacity are roughly proportional to transistor density.

Moore's Law is usually hedged by saying that computing speed doubles every 24 months.

See:

http://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html
http://www.intel.com/pressroom/kits/quickreffam.htm
http://en.wikipedia.org/wiki/Transistor_count
http://en.wikipedia.org/wiki/Beckton_%28microprocessor%29#6500.2F7500-series_.22Beckton.22
https://en.wikipedia.org/wiki/List_of_Intel_Nehalem-based_Xeon_microprocessors#"Beckton"_(45_nm)

Implication of Moore's Law

If computing speed and capacity double every 24 months, what are the implications in our lives?

Well, the average undergrad student is – to one significant figure – about 20 years old.

And the average lifespan in the US – to one significant figure – is about 80 years.

So, the average undergrad student has 60 years to go.

So how much will computing speed and capacity increase during the time you have left?

Double, double, …

60 years / 2 years = 30 doublings

What is 2^{30}?

Consider the computer on your desktop today, compared to the computer on your desktop the day you die.

How much faster will it be?

Can we possibly predict what the future of computing will enable us to do?