INFO 654 – Information Technologies

Week 2: Hardware & Software

Week 2 - Hardware & Software

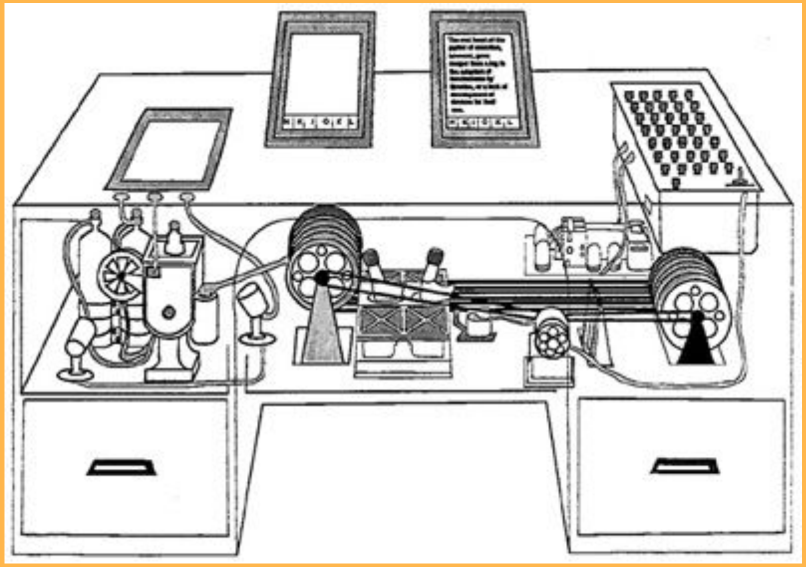
I. Announcements, updates
II. The evolution of technology
III. What is a computer?
IV. Representing information digitally: bits/bytes and digitization

<break>

I. Discuss assignments: sign-up sheet for Current Events Presentation + overview of Careers & IT Post due next week
II. Software (incl. proprietary and open-source software)

THE EVOLUTION OF TECHNOLOGY

How did we get to the present?

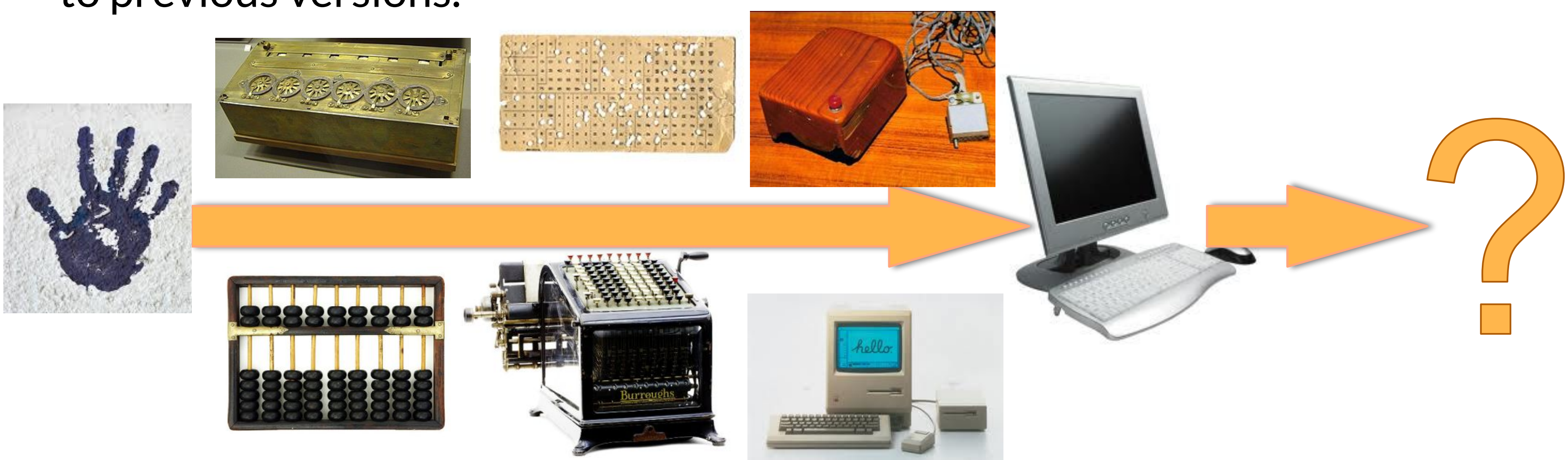


The Evolution of Technology

No technology suddenly springs from the mind of a genius inventor; it evolves over time based on two factors:

- The shortcomings of existing technology
- The social, cultural, and technological context of the time (including funding factors)

Advancements are often incremental, with only slight modifications to previous versions.




25 Years Difference


40 Years Difference

Vannevar Bush's Vision

Vannevar Bush was an American engineer, inventor and science administrator

- During World War II, he headed the U.S. Office of Scientific Research and Development (OSRD)

In 1945, Vannevar Bush laid out his vision for the future of computing in his essay "As We May Think";

- He didn’t get everything right, but his ideas were well before his time!

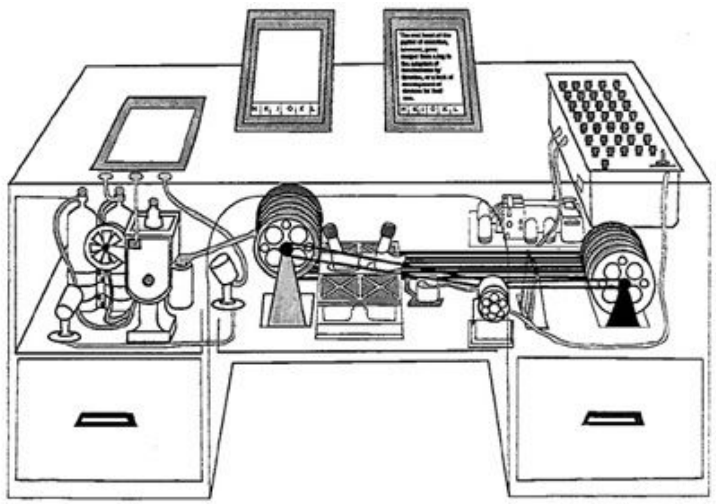
Bush described a device called the “memex.”

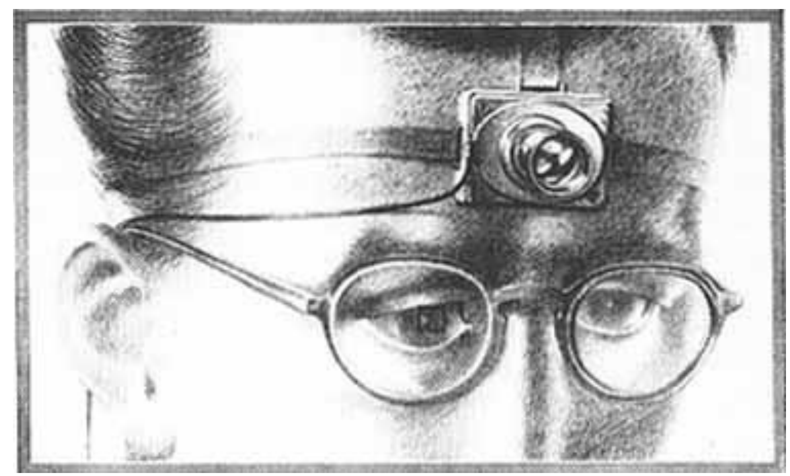
- The machine was never actually built but his vision was highly influential to the computing field

Source: Bush, V. (July 1945). As We May Think. *The Atlantic Monthly*. Available at: http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/3881/?single_page=true

As We May Think

What modern technologies did Bush predict?





"Consider a future device … in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

Media Archaeology Lab

https://www.mediaarchaeologylab.com/

1/25

POLAROID SUN 600 LMS LAND INSTANT CAMERA

Accession Number : 2019.10.01
Medium: Audio-Visual
Product Name: Polaroid Sun 600 LMS
Manufacturer: Polaroid
Model: Sun 600 LMS
Year: 1981
Compatibility: 600 film cartidge
Accession Number: 2019.10.01

PRINTED MATTER	SOFTWARE	HARDWARE	AUDIO-VISUAL

What metadata are they collecting? Are there attributes you would want to add in your ideal media archaeology archive? Are there ways you would change the website/digital visitor experience?

Activity

Media Archaeology Lab posts!

- Go to the course Canvas site to the Discussion where your colleagues have posted favorites from the Media Archaeology Lab
- Read through all of the posts (if you haven't already before class)
- Pick one and (1) view the hardware and software it mentions on the Lab's site and (2) reply to their post, commenting on either the hardware, the software or something specific responding to their original comment
- We'll spend about 15 mins

WHAT IS A COMPUTER?

Computer = a multipurpose device + Input, Process, Output, Store

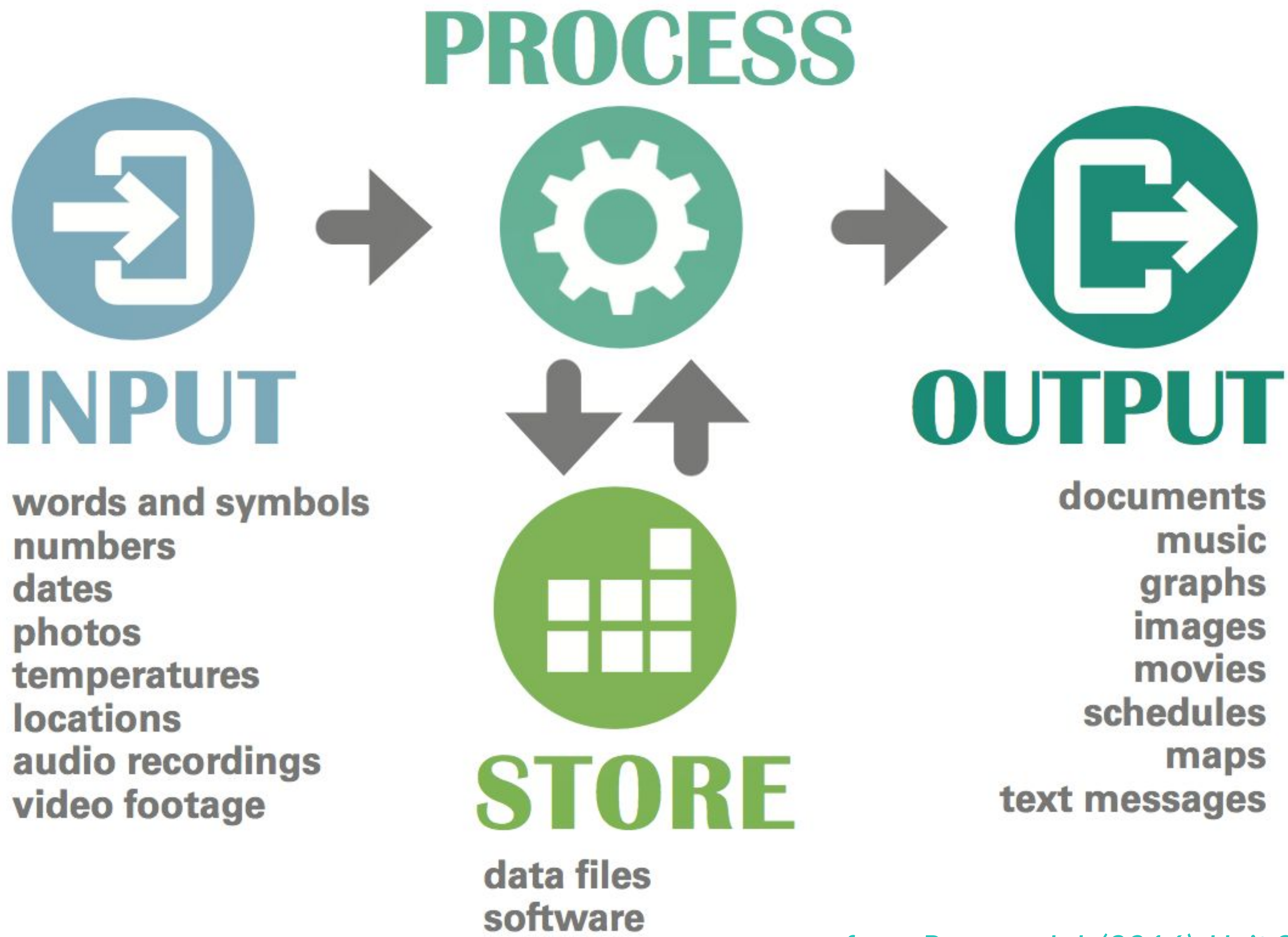
Input is whatever is typed, submitted, or transmitted to a computer. **Output** is the result produced by a computer. Input and output can be handled by components contained within the computer or by **peripheral devices**, such as keyboards and printers, that are attached to the computer with cables or connected wirelessly.

Computers **process data** by performing calculations, modifying documents and pictures, drawing graphs, and sorting lists of words or numbers. Processing is handled by the computer's **central processing unit** (CPU). The CPU of most modern computers is a **microprocessor**, which is an electronic component that can be programmed to process data.

Computers store data and the software that processes data. Most computers have temporary holding areas called **memory** in addition to long-term **storage** housed on hard disks or flash drives. Figure 2-1 illustrates the IPOS (input, process, output, store) activities characteristic of computers.

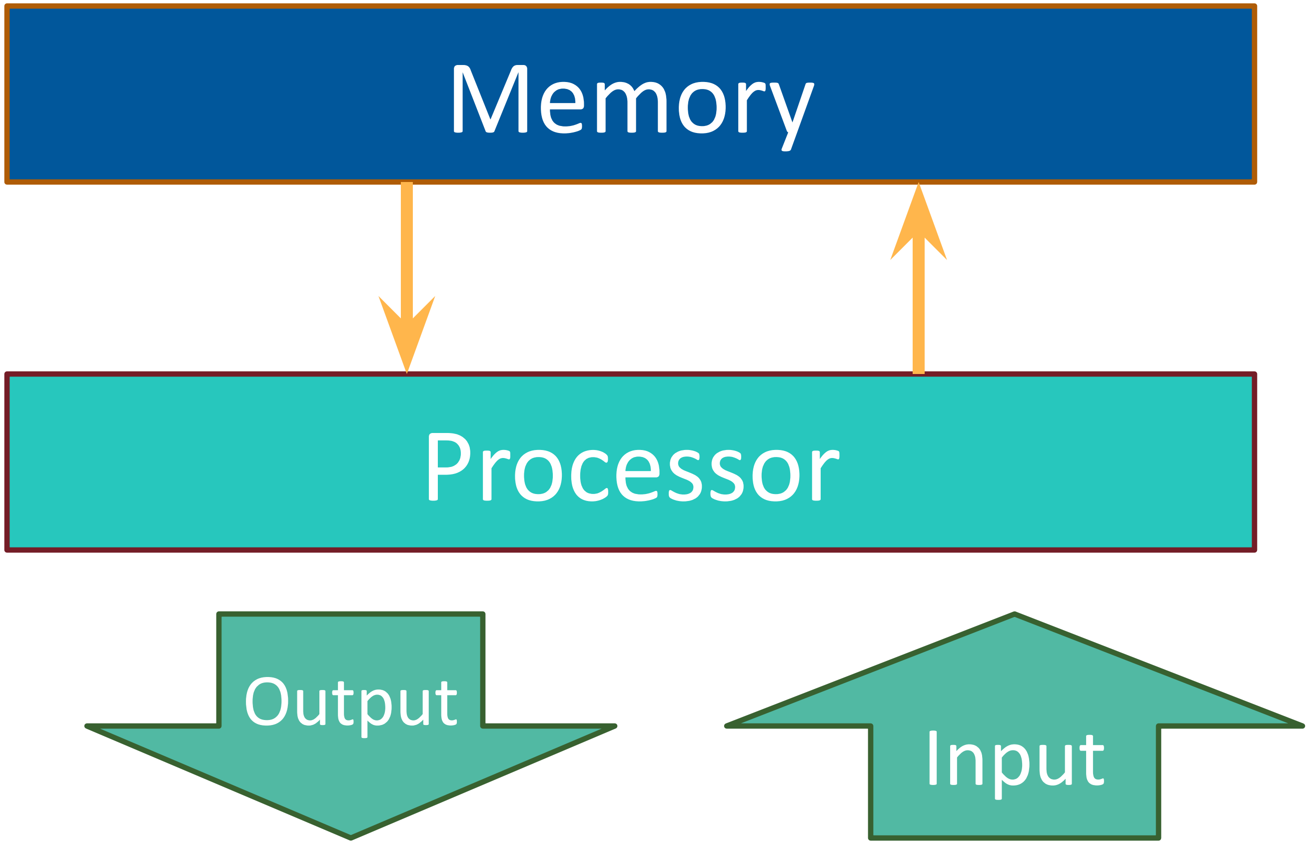
from: Parsons, J. J. (2016). Unit 2: Digital Devices

Computers, defined:
Input, Process, Output, Store (IPOS)



from: Parsons, J. J. (2016). Unit 2: Digital Devices

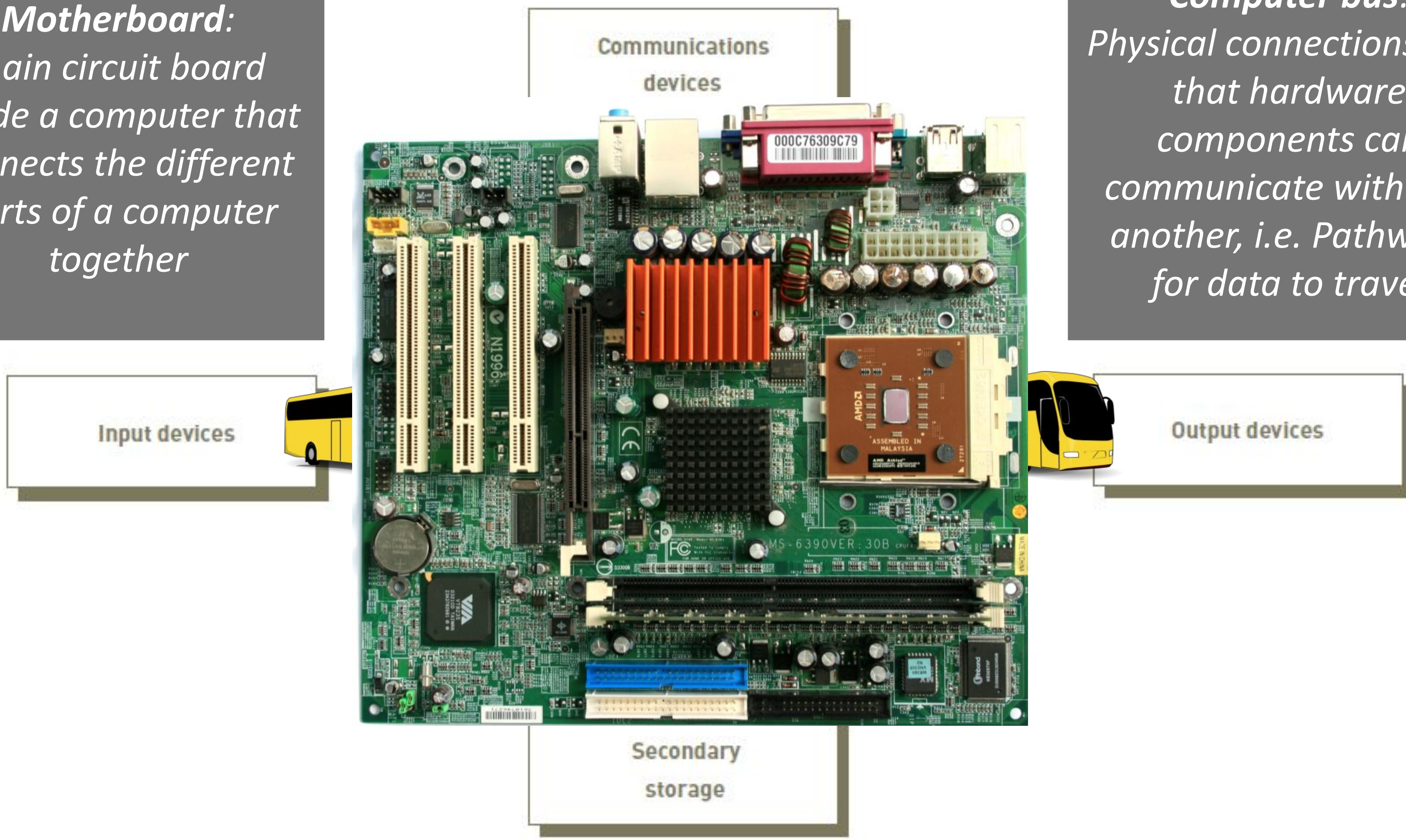
The Processing Cycle

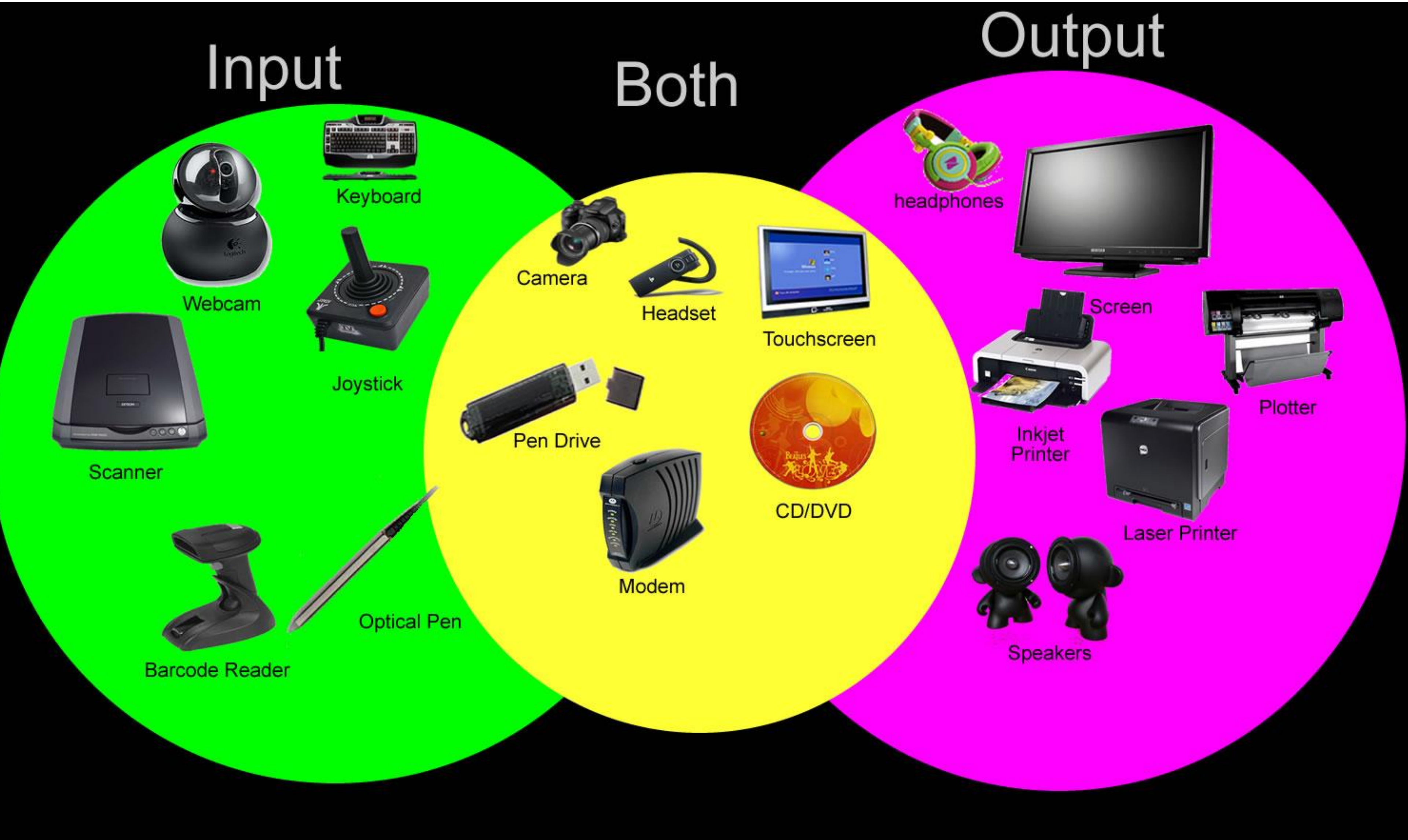


A High-Level View of Hardware

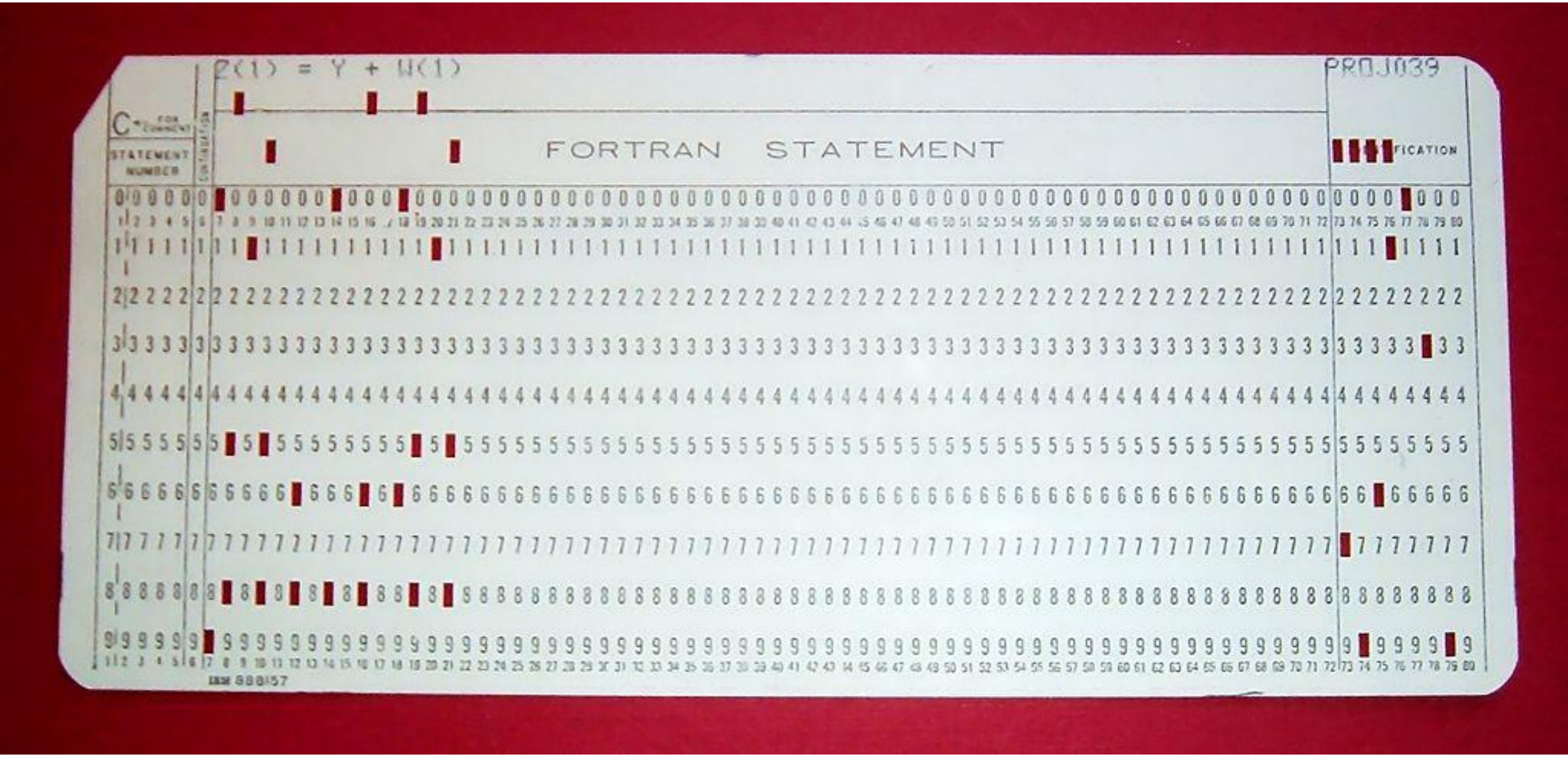
***Motherboard**: main circuit board inside a computer that connects the different parts of a computer together*

***Computer bus**: Physical connections such that hardware components can communicate with one another, i.e. Pathways for data to travel*





Input and Output Devices

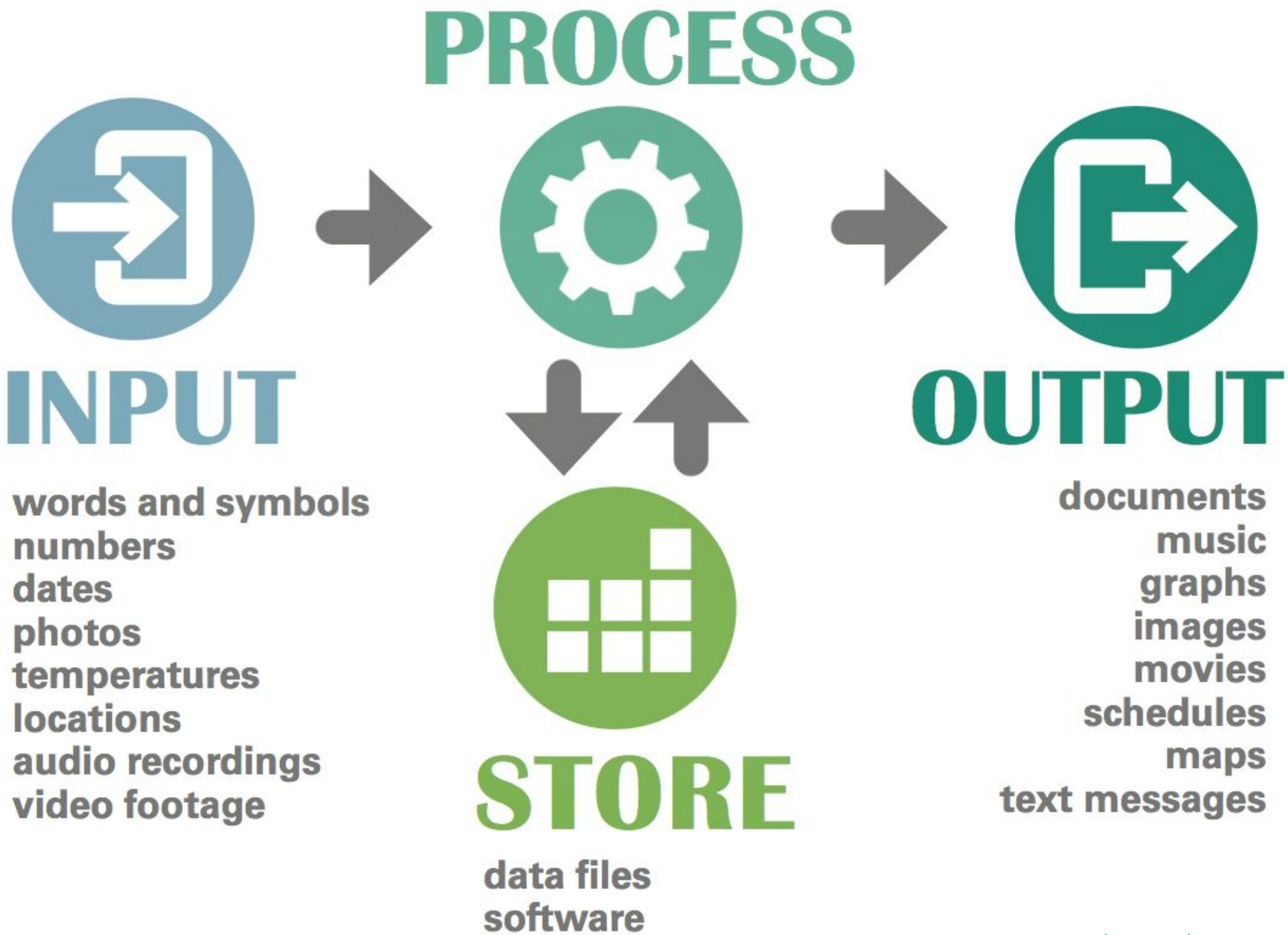


Input --

early programming with a FORTRAN punchcard
the punches indicate characters, numbers, and special attributes
a large computer would be fed a stack of punchcards as commands

Source: Wikimedi ta Commons

Computers, defined:
Input, Process, Output, Store (IPOS)



from: Parsons, J. J. (2016). Unit 2: Digital Devices

Processing

Hardware components:

- Central processing unit (CPU):
  - Consists of the arithmetic/logic unit, the control unit, and the register areas

- Primary storage:
  - Also called **main memory**
  - Closely associated with the CPU

The Processor (CPU)

1. **Arithmetic/logic unit (ALU):**
   - Performs mathematical calculations and makes logical comparisons
2. **Control unit:**
   - Sequentially accesses program instructions, decodes them, and coordinates the flow of data in and out of the ALU, registers, primary storage, and even secondary storage and various output devices
3. **Registers:**
   - High-speed storage areas
   - Used to temporarily hold small units of program instructions and data

Processing in Action

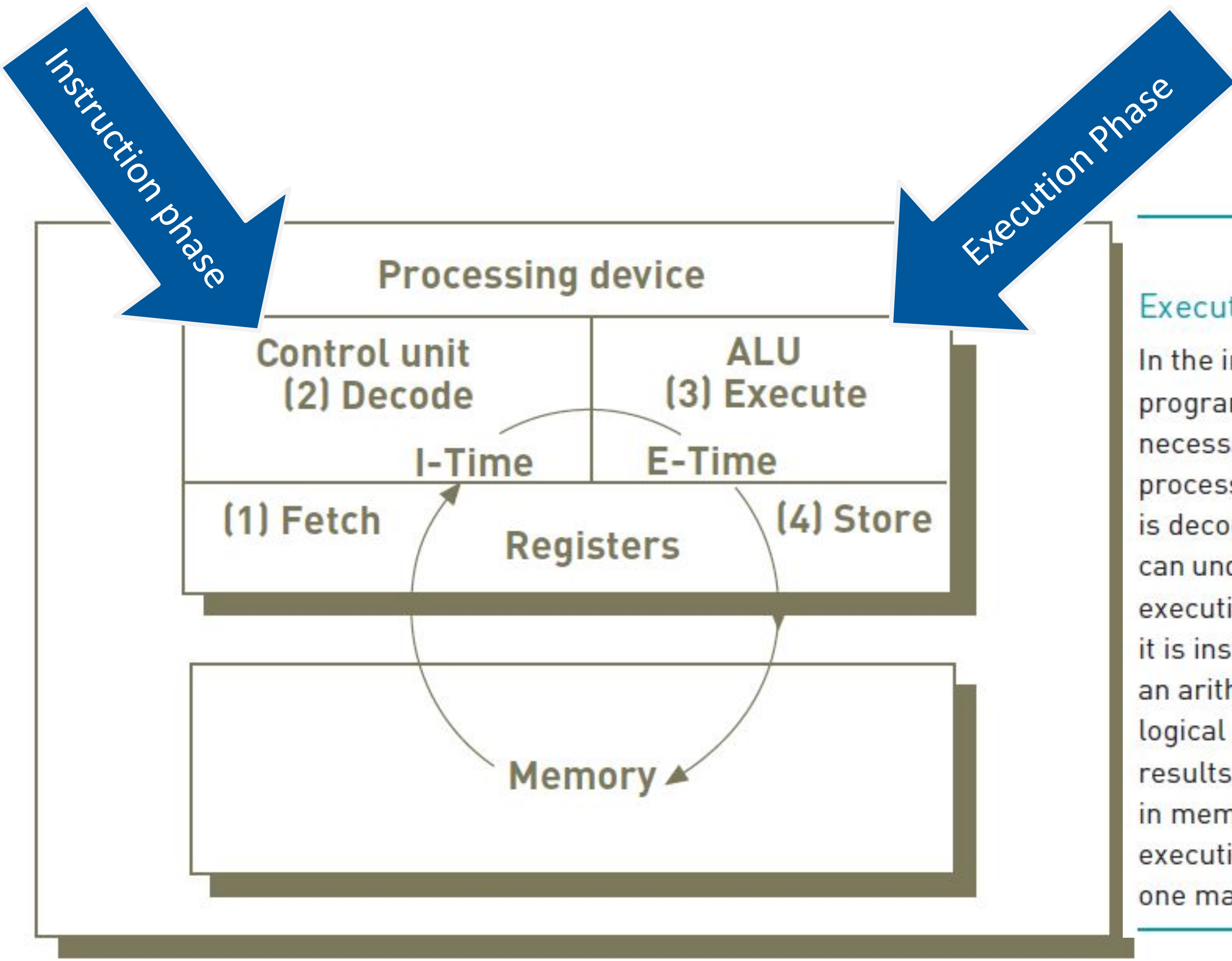


Figure 3.2

Execution of an Instruction

In the instruction phase, a program's instructions and any necessary data are read into the processor (1). Then the instruction is decoded so the central processor can understand what to do (2). In the execution phase, the ALU does what it is instructed to do, making either an arithmetic computation or a logical comparison (3). Then the results are stored in the registers or in memory (4). The instruction and execution phases together make up one machine cycle.

Machine Cycles

Machine cycle time is measured in:

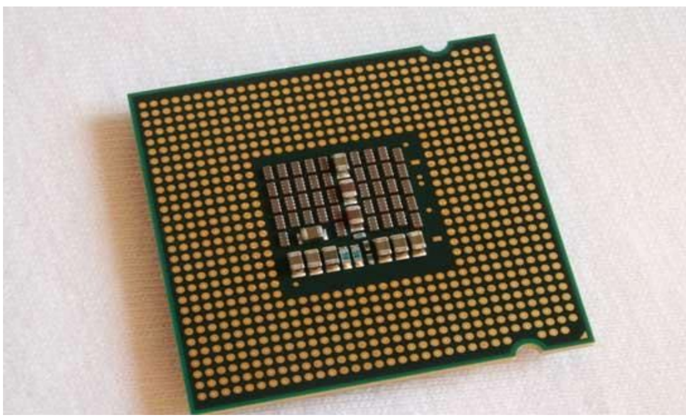
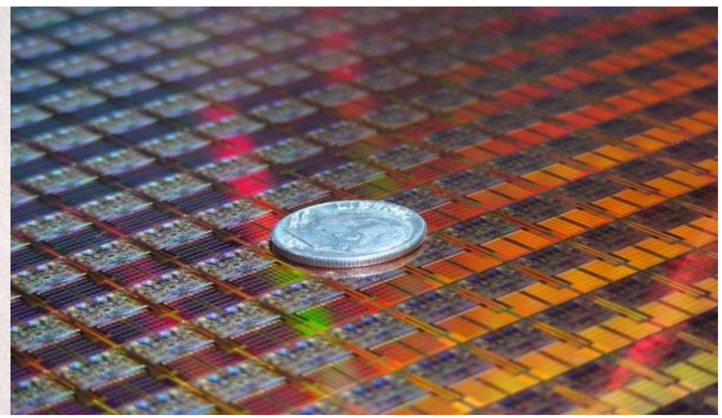
- Nanoseconds (1 billionth of a second)
- Picoseconds (1 trillionth of a second)
- MIPS (millions of instructions per second)

Clock speed:

- Series of electronic pulses produced at a predetermined rate that affects machine cycle time
- Often measured in:
  - Megahertz (MHz): millions of cycles per second
  - Gigahertz (GHz): billions of cycles per second

Physical Characteristics Of The CPU

Most CPUs are collections of **digital circuits** imprinted on silicon wafers, or **chips**, each no bigger than the tip of a pencil eraser

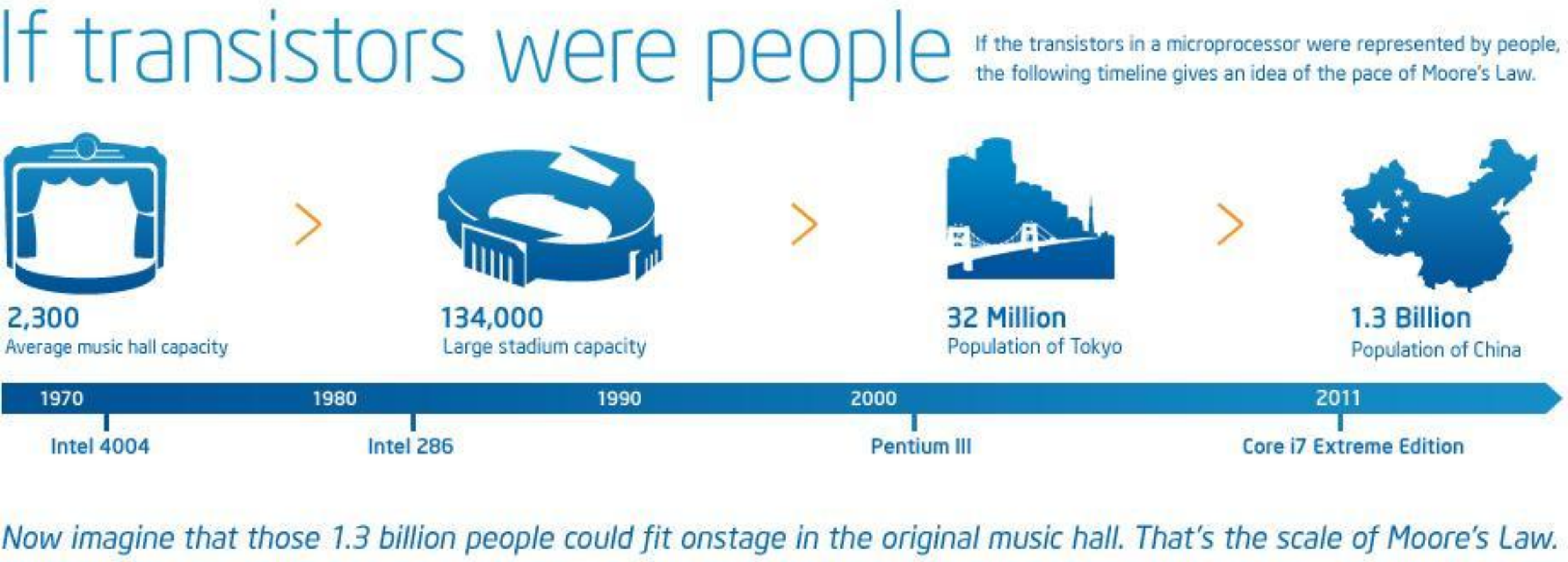



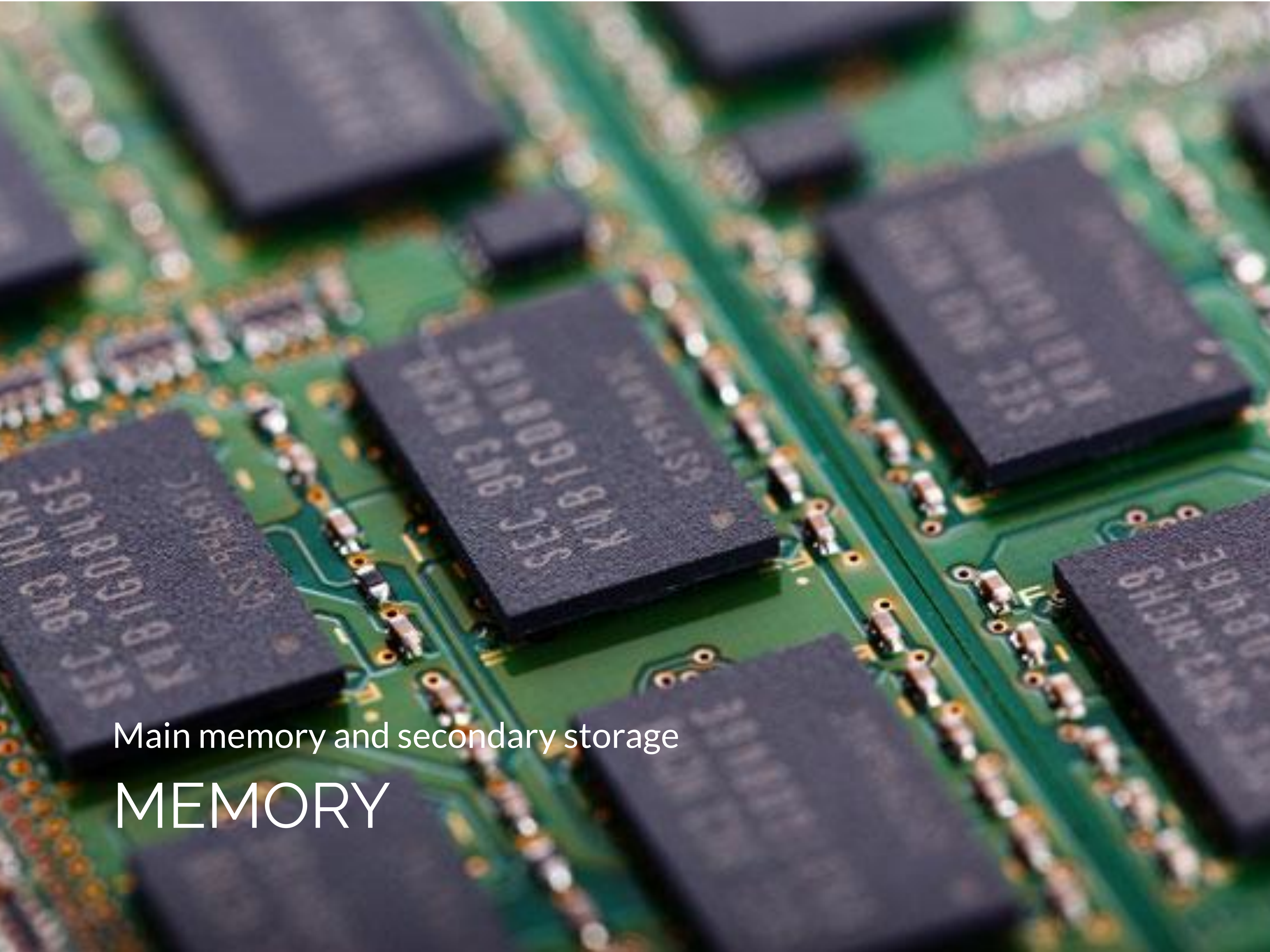
Moore's Law?

Hypothesis stating that transistor *densities* on a **single chip** will double every two years

Rate has slowed slightly in recent years, but proposition remains highly influential in hardware development

Transistor Density Over Time





Main memory and secondary storage

MEMORY

Overview of Computer Memory

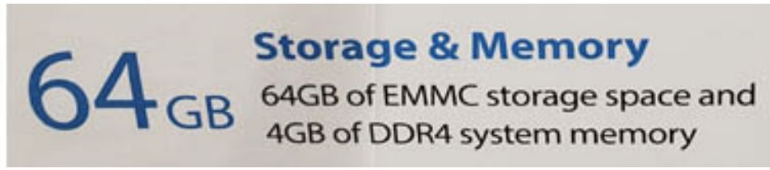
Computers store data using *various methods*

Each of which have inherent tradeoffs!

1. **Main memory** aka **primary storage** aka **RAM**
    - Provides the CPU with a working storage area for programs and data
    - Connected *directly* to the CPU via the memory bus; rapidly provides data and instructions to the CPU

2. **Secondary storage**
    - Hard drives are considered secondary storage since they are not connected directly to the CPU
        - Data must be moved to main memory for use
        - non-volatile
    - Storage is slower, but (more) permanent and capacity is greater

Main Memory/RAM and Secondary Storage in Hardware Specs



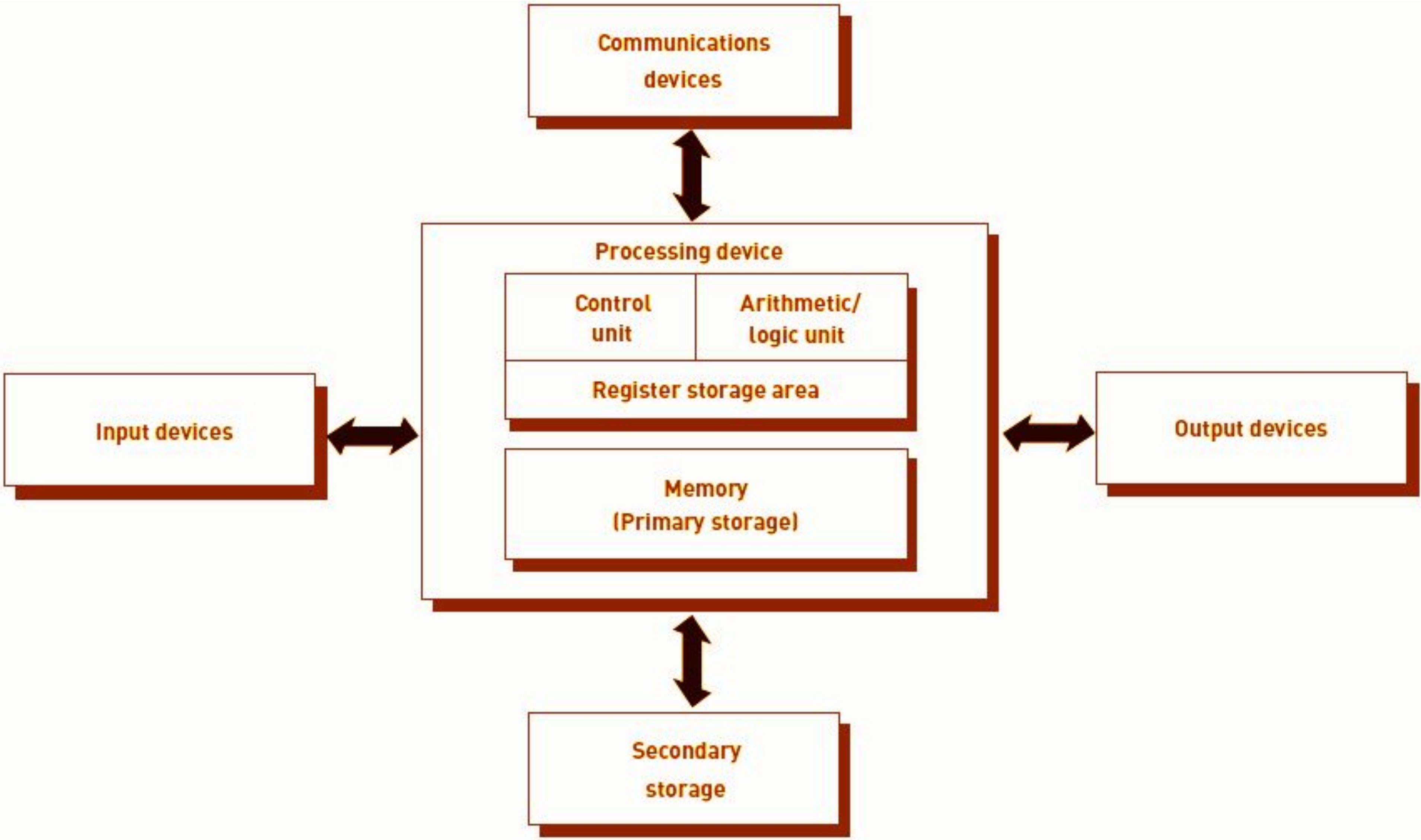
from: PC Mag Encyclopedia Online,
https://www.pcmag.com/encyclopedia/term/storage-vs-memory

RAM (Random Access Memory)

How does RAM work? In RAM, microscopic electronic parts called **capacitors** hold the bits that represent data. You can visualize the capacitors as microscopic lights that can be turned on or off. A charged capacitor is "turned on" and represents a "1" bit. A discharged capacitor is "turned off" and represents a "0" bit. A RAM address on each bank helps the computer locate data, as needed, for processing.

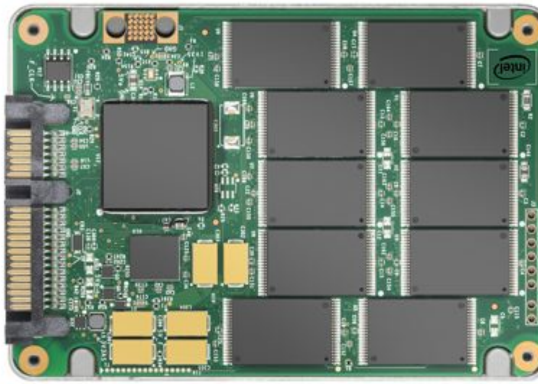
from: Parsons, J. J. (2016). Unit 2: Digital Devices, p107

Overview of Computer Memory, continued



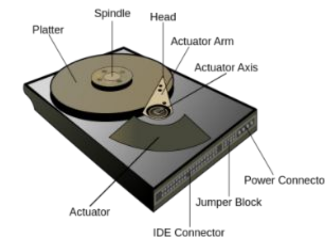
Secondary Storage

- Compared with main memory, offers the advantages of non-volatility, greater capacity, and greater economy
- **Magnetic (hard) disk drives:**
  - Circular rotating "plate" of magnetic material called a *platter* with read/write arm
  - Billions of tiny sections which are individually magnetized to store 0 or 1
- **Solid State Drives (SSD):**
  - Store data in memory chips which have few moving parts, so are less fragile than hard disk drives
  - High cost and lower capacity compared to current hard drives



Other Secondary Storage Devices

- **Magnetic tapes**:
    - Primarily for storing backups of critical organizational data

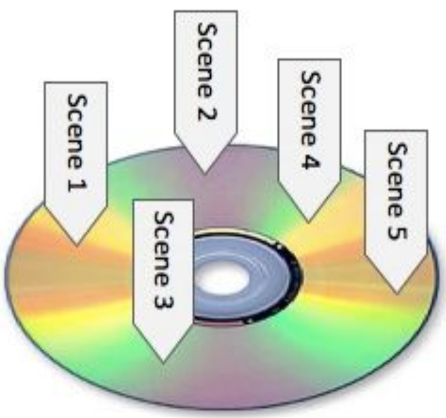


- **Optical secondary storage devices**:
    - Compact disc read-only memory (CD-ROM):
        - Storage capacity is 740 MB
    - Digital video disc (DVD):
        - 6 x capacity of CD, 4.7 GB
    - Blue-ray high-definition video disk:
        - 3 x capacity of DVD, 27 to 54 GB!


VERBATIM
CD-R 52X

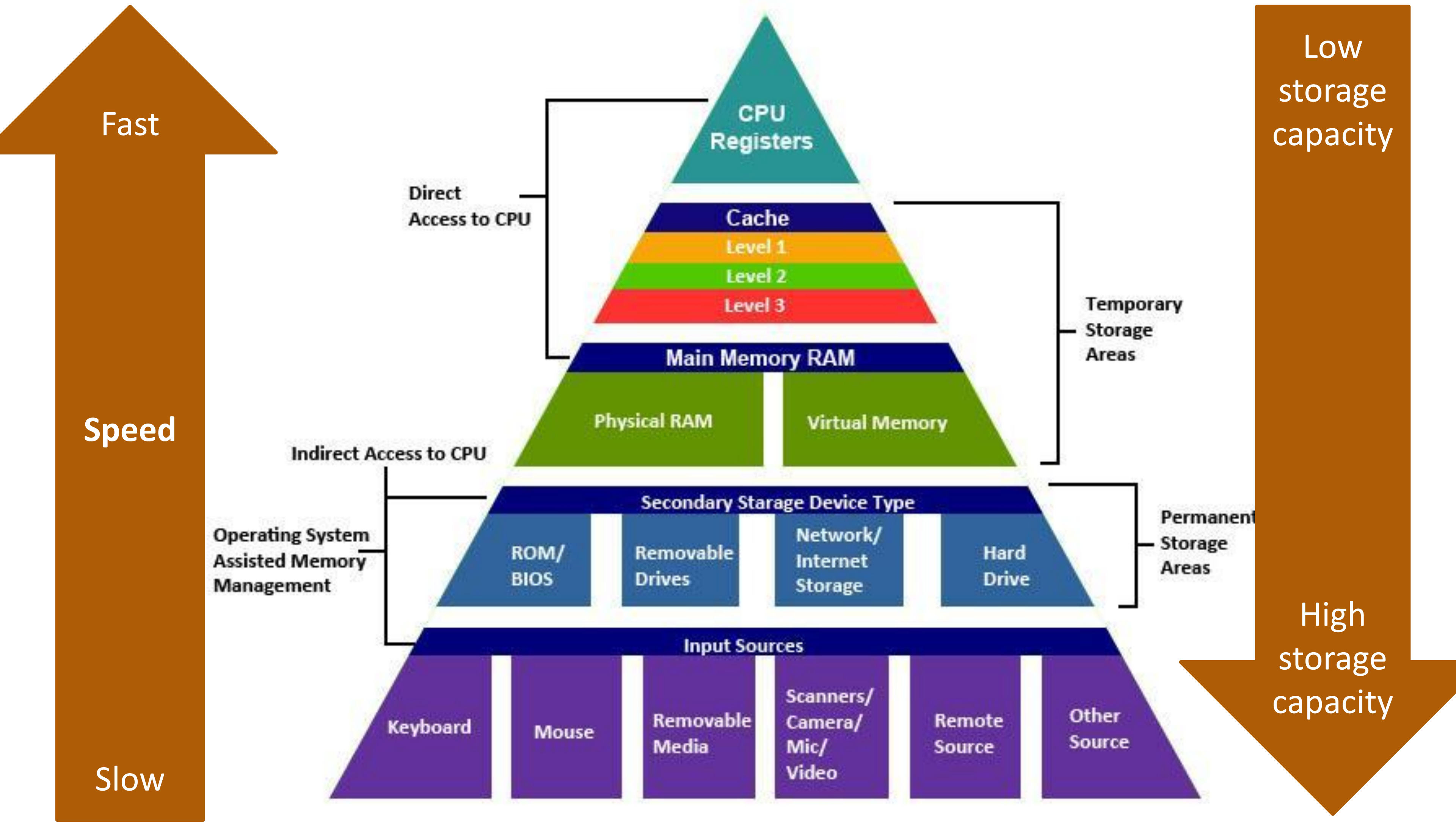
Access Methods

- **Sequential access:**
  - Data must be retrieved in the order in which it is stored
  - Devices used called sequential access storage devices (SASDs)
- **Direct access:**
  - Records can be retrieved in any order
  - Devices used are called direct access storage devices (DASDs)



Which is sequential access and which is direct access?

Types of Memory Compared



System On A Chip (SoC)

- A **system on a chip** or **system on chip (SoC or SOC)** is an integrated circuit (IC) that integrates *all* components of a computer into a single chip
  - Reduces complexity, cost, power consumption, and saves space!

- Common in smartphones, tablets, wearables, etc.

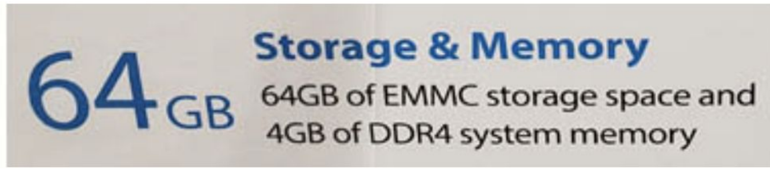
Computer Storage Units

Name	Abbreviation	Number of Bytes
Byte	B	1
Kilobyte	KB	2^{10} or approximately 1,024 bytes
Megabyte	MB	2^{20} or 1,024 kilobytes (about 1 million)
Gigabyte	GB	2^{30} or 1,024 megabytes (about 1 billion)
Terabyte	TB	2^{40} or 1,024 gigabytes (about 1 trillion)
Petabyte	PB	2^{50} or 1,024 terabytes (about 1 quadrillion)
Exabyte	EB	2^{60} or 1,024 petabytes (about 1 quintillion)

Which common now?

Powers of 2!

Main Memory/RAM and Secondary Storage in Hardware Specs



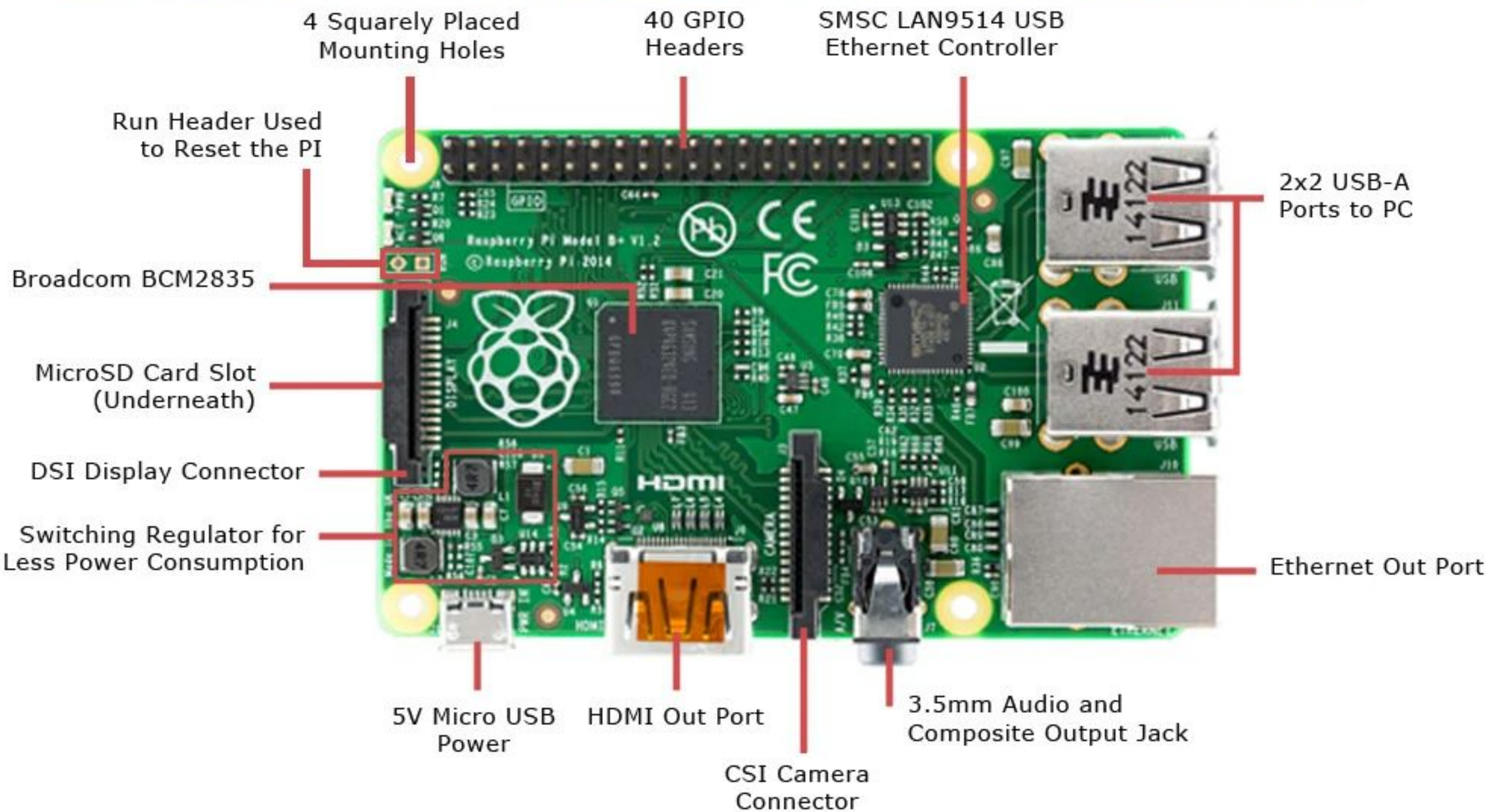
Storage	256GB SSD Configurable to 1TB Fusion Drive
Memory	8GB of 2133MHz DDR4 memory Configurable to 16GB

from: PC Mag Encyclopedia Online,
https://www.pcmag.com/encyclopedia/term/storage-vs-memory

Memories of Memory:

The Commodore 64 (1982) w/64 KB of RAM

Example hardware: Raspberry Pi (3 model B+)



Example: Raspberry Pi (3 model B+)

Specifications

Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
1GB LPDDR2 SDRAM
2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
Extended 40-pin **GPIO** header
Full-size HDMI
4 USB 2.0 ports
CSI camera port for connecting a Raspberry Pi camera
DSI display port for connecting a Raspberry Pi touchscreen display
4-pole stereo output and composite video port
Micro SD port for loading your operating system and storing data
5V/2.5A DC power input
Power-over-Ethernet (PoE) support

What are some examples of computers in this room / that you have near you right now…?

let’s name some examples

Class Activity

"Check the Specs"

- Form a group of 2-3 people
- Pick a device (or several) of your team-mates
  - This can be a laptop, tablet, lab desktop, smartphone, etc.
- Determine the device's:
  - Processor & speed
  - Memory – primary & secondary storage
  - Input & output devices

  ***Note – you may have to do some research as to how to find out these specifications on your device! (and your book reading for this week has hints for several device types)*
- Take notes to report back on
- You'll have 10-15 minutes to discuss, then we'll all combine and look at our results

report-back in group:

- *any "outliers" in your group of devices? (very large, very small, very powerful, very not-powerful, etc.)*
- *unusual input or output?*
- *what are the specs for the Room 606 computers?*

Representing Information Digitally

(a brief introduction)

What do we mean by the term digital?

Analog vs. Digital

- "Analog" comes from the same root as "analogous”
  - It is meant to convey the idea of values that change smoothly.
  - Much of what we deal with in the physical world is analog!
- How are these controls operated? Which of the below would you consider analog?

- Digital systems deal with *discrete values*; there are only a limited number of possible values

Digitizing Discrete Information

- The dictionary definition of *digitize* is to **represent information with digits.**

- Digitizing uses whole numbers to ***stand for*** *things*.
  - What kinds of *things*?

So Why Digital?

Digital data is easy to work with

- It can be stored, transported, and processed in many ways regardless of its original source, which is not true for analog information
- Digital information can be compressed by squeezing out redundant or unimportant information.
- It can be encrypted for security and privacy
- It can be merged with other data
- It can be copied without error
- It can be shipped anywhere via the Internet
- It can be stored in an endless variety of devices

Much of this is simply not possible or is to cumbersome with analog information

Fundamental Information Representation

In the *physical world*, the most fundamental form of information is the **presence** or **absence** of a physical phenomenon

From a *digital* information point of view, the amount of a phenomenon is not important as long as it is reliably detected

Whether there is some information or none;

i.e. whether it is **present** or **absent**

1 **0**

The PandA Representation

- **PandA** is the name used for two fundamental patterns of digital information:
  - Presence
  - Absence
- PandA is the mnemonic for "***Presence and Absence***"
  - A key property of PandA is that the phenomenon is either present or not
- The presence or absence can be viewed as "true" or "false"
- The PandA encoding has two patterns: present and absent
  - *Two patterns* make it a ***binary system***
    0 1

How many units do we have?

The PandA *unit* is known as a **bit**

- **Bit** is a contraction for "**binary digit**"
- Only two values can be represented for each bit (0 and 1 in combination)

Bit sequences can be interpreted as binary numbers

- *Groups of bits* form **symbols**
- the two patterns are combined into multiple sequences to create enough symbols to encode information

PATTERNS of **on-off, present-absent, 1-0**

- `00`
- `01`
- `11`
- `10`

can then be used in groups of bits to form longer chains of patterns to encode information

Bits in Computer Memory

- Computing devices use electronic circuits called **two-state**
  - Only two states are possible, either **ON** (usually represented by a 1) or **OFF** (represented by a 0)

- **Bits** are typically represented by **an electrical state or voltage**

- Memory is arranged inside a computer in a *very, very, very, very, very long sequence of bits*
  - This means places where the physical phenomenon encoding the information can be set and detected, i.e. determine **presence or absence**

Combining Bit Patterns

- The two-bit patterns gives *limited* resources for digitizing information
  - *Only two values* can be represented for each **bit**
- However, we can use **groups of bits** to form **symbols**
  - The two patterns must be *combined into sequences to create enough symbols to encode the intended information*
- Example – how many possibilities are there with two coin tosses?
  - 00
  - 01
  - 11
  - 10

Encoding in Binary

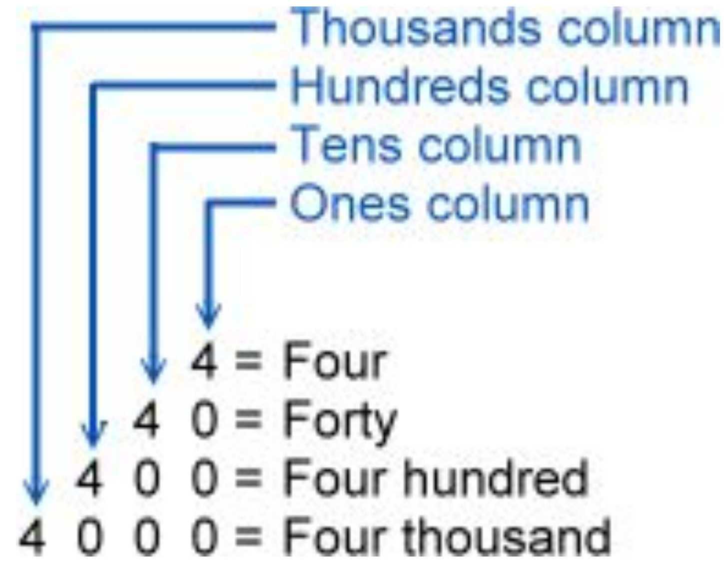
- We **encode** the binary data by mapping the bits to the symbols that need to be represented, e.g. characters, numbers

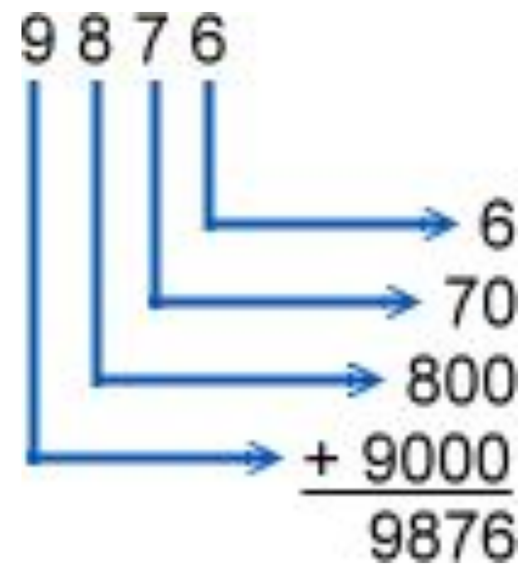
- The two earliest uses of PandA were to:
  - Encode numbers
  - Encode keyboard characters

- Representations for sound, images, video, and other types of information also became important!

Encoding Numbers

- Binary numbers are limited to two digits, 0 and 1
- The number of digits is the base of the numbering system
  - We are used to working with base 10, e.g. ones, tens, hundreds, etc.
- With binary numbers, it is the same idea, but with higher powers of 2!





Encoding Numbers *continued*

Power	Decimal	Binary
0	$1 = 10^0$	$1 = 2^0$
1	$10 = 10^1$	$2 = 2^1$
2	$100 = 10^2$	$4 = 2^2$
3	$1000 = 10^3$	$8 = 2^3$
4	$10{,}000 = 10^4$	$16 = 2^4$
. . .	. . .	. . .

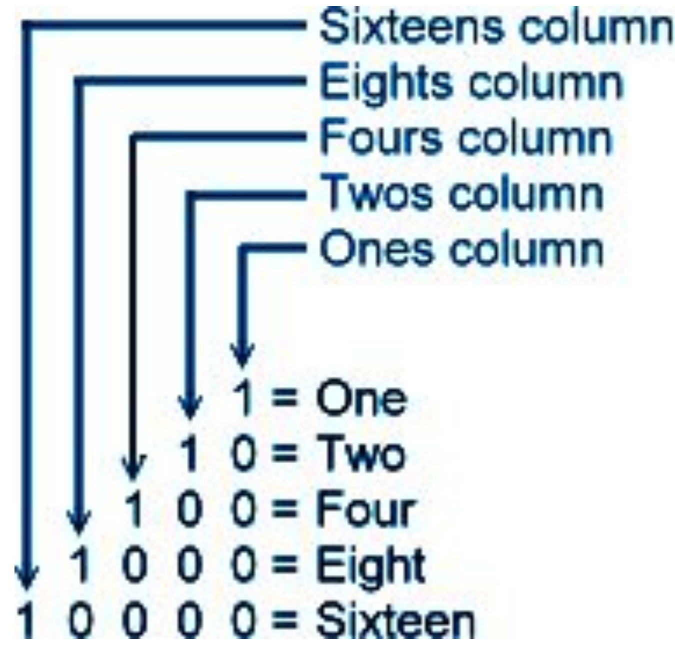
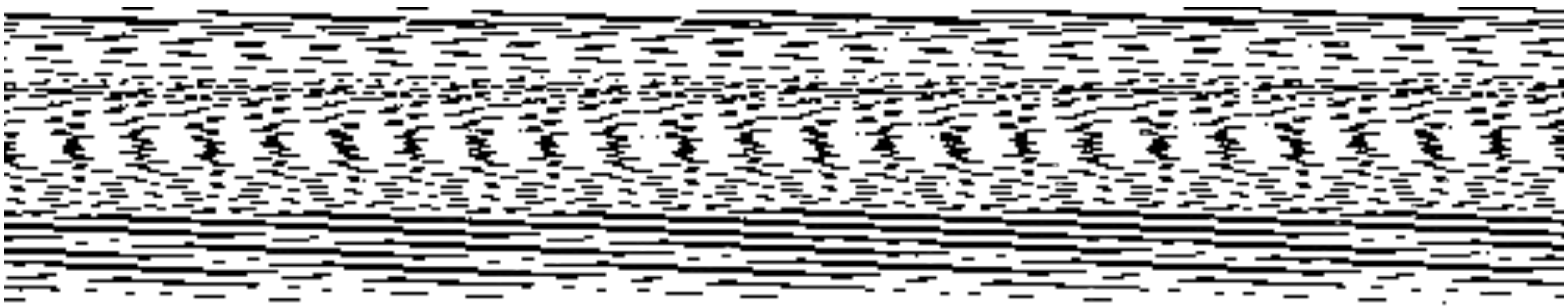


Table 7.5 The binary number 1010, representing the decimal number ten = 8 + 2

2^3	2^2	2^1	2^0	Binary Place Values
1	0	1	0	Bits of Binary Number
1×2^3	0×2^2	1×2^1	0×2^0	Multiply place bit by place value
8	0	2	0	and add to get a decimal 10

Counting in a Binary System!



http://britton.disted.camosun.bc.ca/jbbinary.htm

A programming joke ....

"There are only 10 kinds of people in the world-those who understand binary numbers and those who don't."

Binary number *1 0* =

$1*2^1 + 0*2^0$ =

2+0 =

2

Exercise

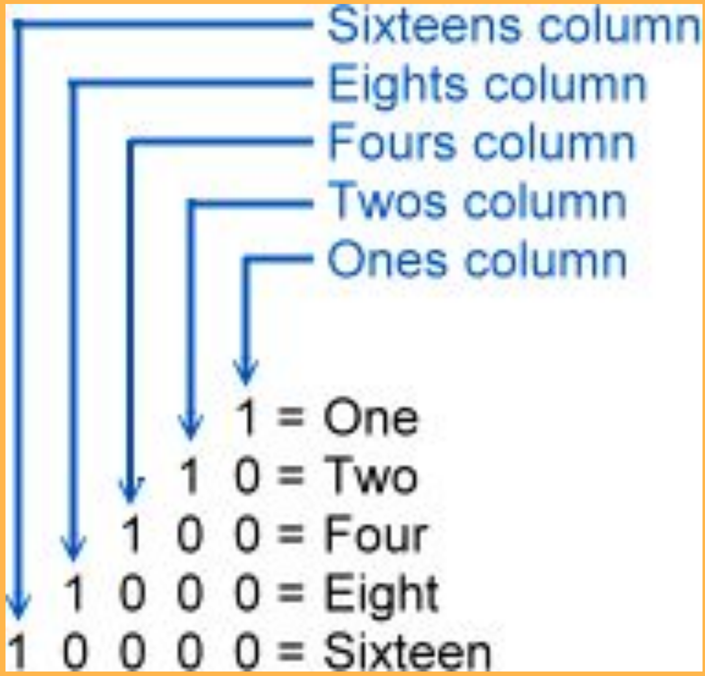
Take a few minutes to do the two conversions using the reference

1. Convert the decimal value below to *binary*:

   14

2. Convert the binary value below to *decimal*:

   11101



Exercise - Answer

1. Convert the decimal value below to *binary*:

   14

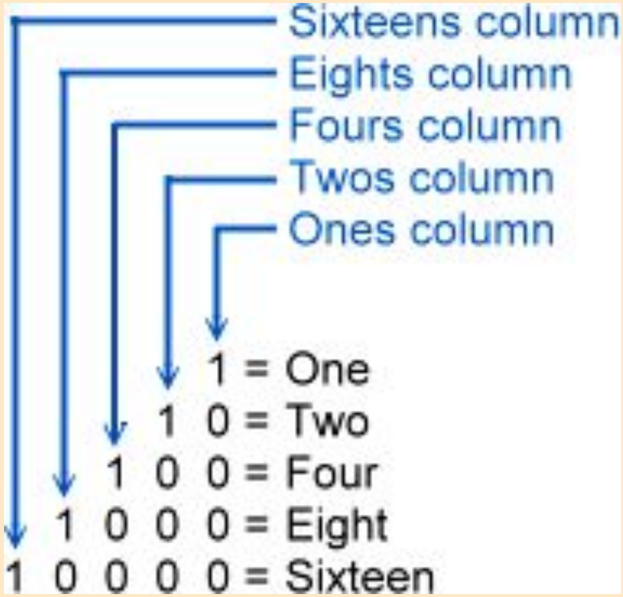
   1 1 1 0

   8 + 4 + 2 + 0 = 14

2. Convert the binary value below to *decimal*:

   1 1 1 0 1

   16 + 8 + 4 + 0 + 1 = 29



Encoding Text

- The number of bits determines the number of symbols available for representing values:
    - *n bits in sequence yield 2^n symbols*

- The more characters you want encoded, the more symbols you need!

- Roman letters, Arabic numerals, and about a dozen punctuation characters are about the minimum needed to digitize **English** text

Encoding Text, continued

- We need to represent:
  - 26 uppercase, 26 lowercase letters, 10 numerals,
  - 20 punctuation characters,
  - 10 useful arithmetic characters,
  - <u> 3 other characters (new line, tab, and backspace)</u>
  - 95 symbols…enough for English but not to adequately represent many other languages
- To represent **95 distinct symbols**, how many bits do we need? **7 bits give 2^7 = 128 symbols**
  - 6 bits gives only 2^6 = 64 symbols
  - 128 symbols is ample for the 95 different characters needed for English characters

Assigning Symbols

- ***ASCII*** stands for American Standard Code for Information Interchange
  - ASCII is a widely used 7-bit (2^7) code
- The advantages of a "standard" are many:
  - Computer parts built by different manufacturers can be connected
  - Programs can create data and store it so that other programs can process it later, and so forth
- 7-bit ASCII is not enough, **it cannot represent text from non-English languages**
  - The next larger set of symbols, the ***8***-bit symbols (2^8), i.e. Extended ASCII; Eight bits produce 2^8 = 256 symbols

For further discussions and reflection beyond in-class time:

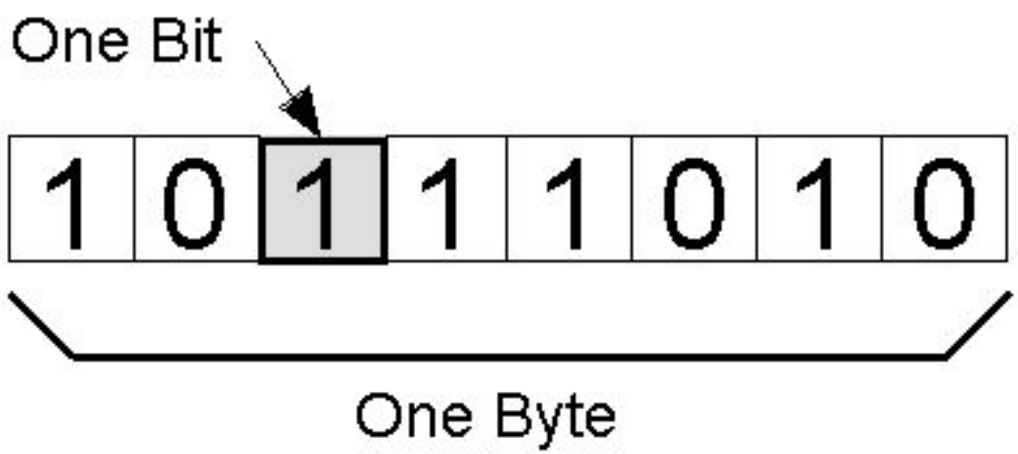
How are computer coding languages embedded in specific histories? How may they reflect colonial and neocolonial practices or mirror economic and political power structures of the specific times in which they were produced? How do those dynamics live on in the present (and can they transform in the future...)?

We will touch on this briefly in our weeks on Programming in October

Extended ASCII: An 8-Bit Code

IBM gave 8-bit sequences a special name, ***byte***

It is a standard unit for computer memory



First letter of my name is an "R", which is:

01010010

ASCII	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NU	SH	SX	EX	ET	EQ	AK	BL	BS	HT	LF	YT	FF	CR	SO	SI
0001	DL	D1	D2	D3	D4	NK	SY	EΣ	CN	EM	SB	EC	FS	GS	RS	US
0010		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DT
1000	80	81	82	83	IN	NL	SS	ES	HS	HJ	YS	PD	PV	RI	S2	S3
1001	DC	P1	P2	SE	CC	MM	SP	EP	Q8	QQ	QA	CS	ST	OS	PM	AP
1010	AO	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure 7.3 ASCII, the American Standard Code for Information Interchange.

Note: The original 7-bit ASCII is the top half of the table; the whole table is known as Extended ASCII (ISO-8859-1). The 8-bit symbol for a letter is the four row bits followed by the four column bits (e.g., A = 0100 0001, while z = 0111 1010). Characters shown as two small letters are control symbols used to encode nonprintable information (e.g., $^{B}{}_{S}$ = 0000 1000 is backspace). The bottom half of the table represents characters needed by Western European languages, such as Icelandic's eth (ð) and thorn (þ).

Class Activity: ASCII Initials!

- Using the same process, **convert the first letter of your first/preferred name and first letter of your last/famly name to binary OR convert the first two letters of your pet's name to binary** (your choice!)
- Write it down on the piece of paper **(**ONLY write the binary version**)**
- Pass it to the person on your left
- Decode their initials back to two characters and compare together to see if you got it right.
  Was it their intials or their pet's initials? Were any adjustmennts made to fit the limitations of ASCI? (accented letters changed, etc.)

DIGITIZING MULTIMEDIA

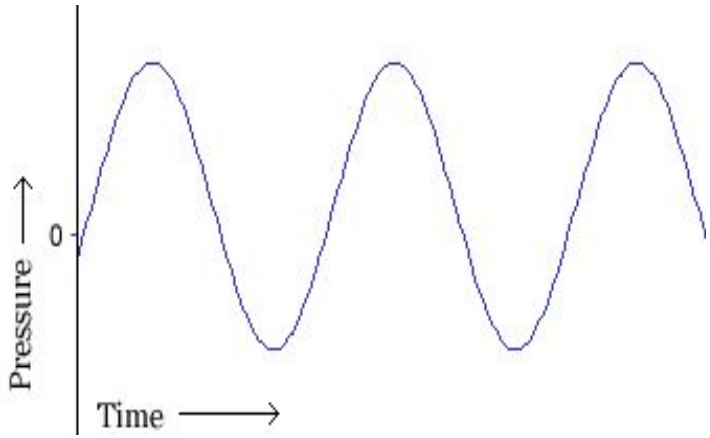
A very very brief summary/example set

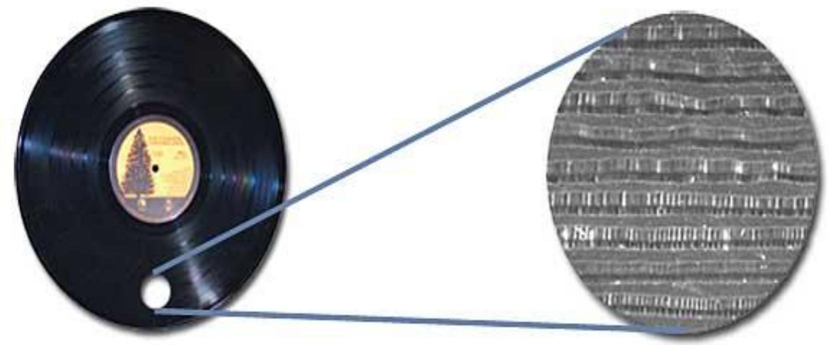
Recording Analog Music

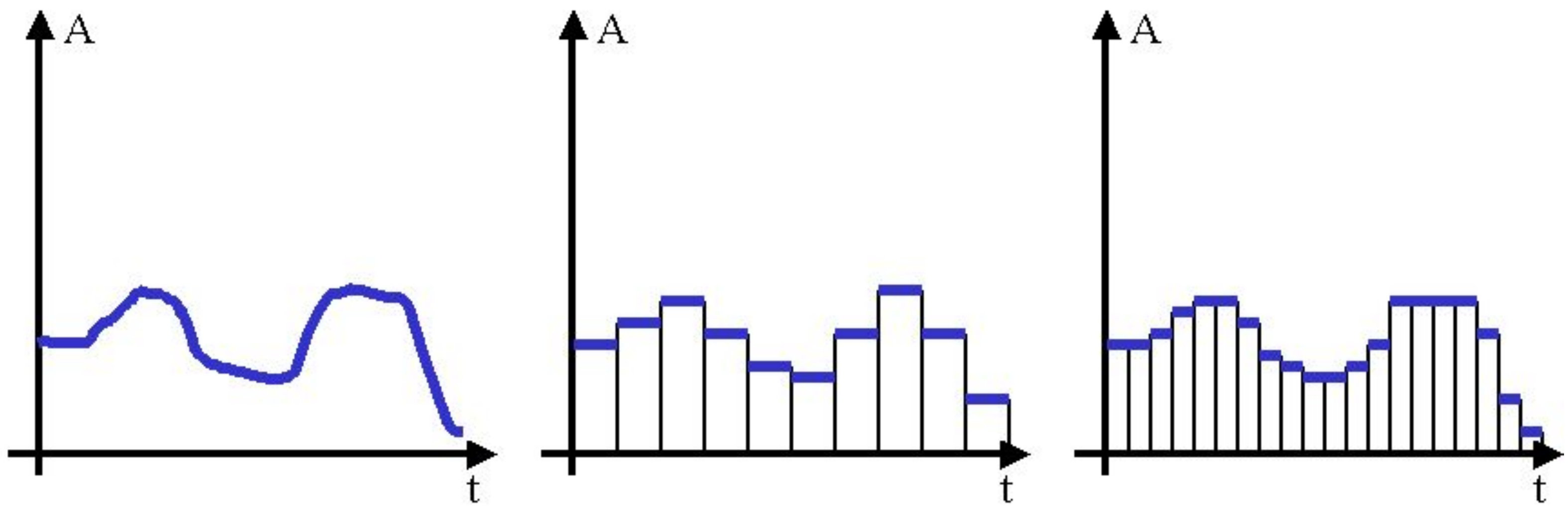
- A sound source creates **fluctuations in air pressure** by vibration or other rapid motion
  - Our ears **convert** the pressure changes into neural activity that our brains interpret as "sound."
- In the 1870s, Thomas Edison built a device that he called a "phonograph”
  - **Converted the fluctuations into an analogous pattern of grooves** in a wax cylinder that could be used later to recreate the air pressure fluctuations.
- Converting a sound into a pattern of grooves was "recording“
  - Converting from the pattern to fluctuations in air pressure was "playback."

Digitizing Music:

generating a series of numbers that describe/represent an analog sound/recording







Analog signal – continuously varying

Digital signal – large time divisions

Digital signal – small time divisions

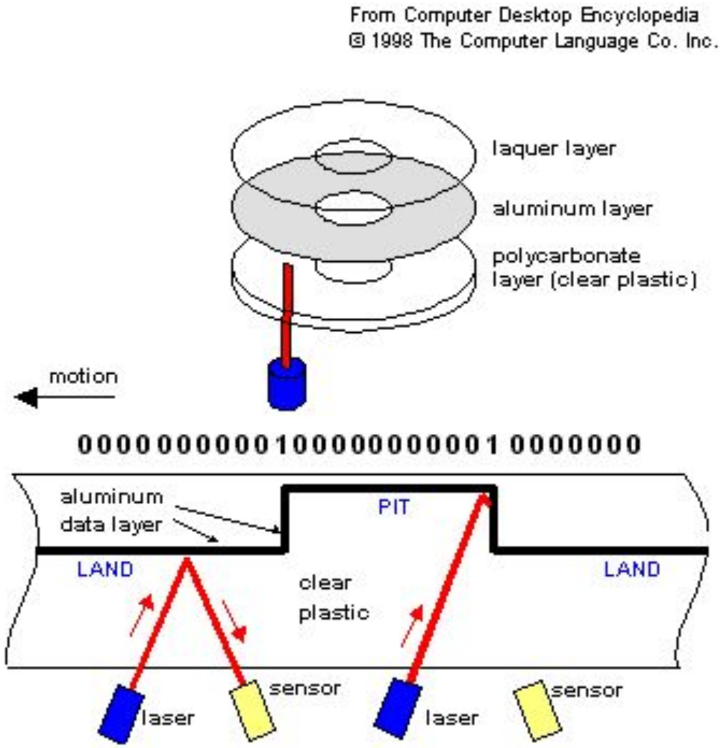
What’s on a CD?

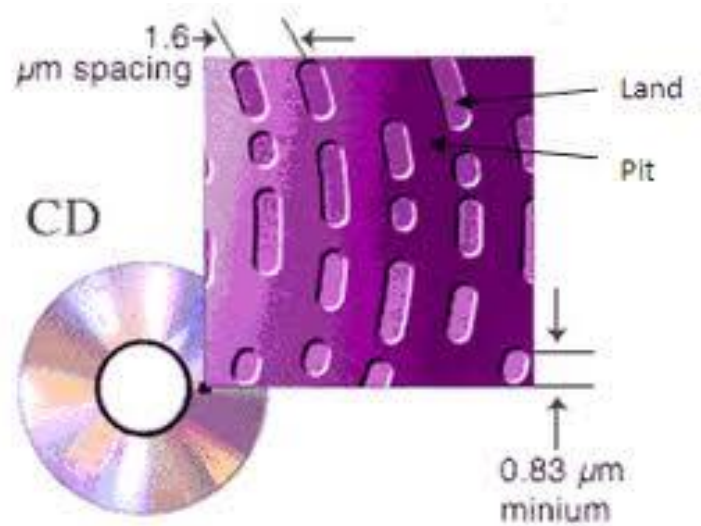
A CD records *numbers* in a long spiral track on one side of the disk

- The surface at each point along the track either is smooth or has a tiny pit
- These pitted or smooth spots are used to encode the numeric values of the wave

Each spot is a single bit

- A sequence of bits represents the numeric value in a binary encoding





<break!>

looking ahead: Assignments

Current Events Presentation:

- an opportunity to read more about a current topic and share with all of us!
- potential news sources list here
(but feel free to use your own)
- present between weeks 5 and 14, on the date you sign up for -- you'll present at the beginning of class (10-15 mins each presentation; max 2 presentations each week)
- between today and next week: add your name to a date in the sign-up sheet
- see full assignment description via link in Canvas under the first module

Current Events Presentation:

- For your presentation, you should find a current event that relates to information technology and has a current (or potential) impact on the field of information.
- In your presentation (10 to 15 minutes) you should:
  - Provide a summary of the current event
  - Describe and explain the technologies involved in this event, using appropriate technical terminology
  - Explain the impact or relevance to the field of information, in the present and/or future
  - You may include short audio-visual materials as desired

Careers & Information Technology Post:

- due next week, **Tuesday, September 13** by class start time
- submit via Discussion Post on Canvas, following instructions in assignment description for content (note this is our only formal Discussion post)
- see full assignment description via link in Canvas under next week's date

Careers & Information Technology Post:

In your discussion post discuss the following:

- What technologies or skills are listed as required for this position? How do these technologies help complete the tasks and responsibilities of this role?
- What other IT roles might this position need to collaborate with?
- What decision-making involving IT might this position be involved in?
- What technology skills necessary to this position do you feel that you currently possess? What technologies / skills would you like to learn to fulfill this role?

Within your post do the following:

- Include a link to the job listing(s) that you have used in your research
- Tag your post with several keywords appropriate to the content, e.g. HTML, Asset Management, LMS, etc.
- Incorporate visuals and/or additional external links, if possible

SOFTWARE

The computer programs that govern the operation of the computer

Machine Code

- A CPU understands a low level "**machine code**" language
    - Represented in **binary**
- The language of the machine code is hardwired into the design of the CPU hardware
    - Each family of compatible CPUs has its own, idiosyncratic machine code

- Programs, e.g. Microsoft Excel, Firefox, etc. consist of *millions* of simple machine code instructions
    - Programmers write ***source code*** that is compiled, or translated, into machine code

Types of Software

Systems Software

- Set of programs that coordinates the activities and functions of hardware and other programs
- Computer system *platform*: Combination of a hardware configuration and systems software
- Each type of systems software is designed for a specific CPU and class of hardware

Application Software

- Helps users solve particular problems
- May esides on the computer's hard disk
- Or may be accessed via the Web as a rich Internet application, e.g. Google Docs

Systems Software

Controls the operations of computer hardware

Operating system:

- Set of programs that controls computer hardware and acts as an interface with application programs
- What are some examples?

The Operating System

OPERATING SYSTEM BASICS

An operating system gives your digital device a personality. It controls key elements of the **user interface**, which includes the visual experience as well as the keyboard, mouse, microphone, or touchscreen that collects user commands. Behind the scenes, the operating system is busy supervising critical operations that take place within a device.

from: Parsons, J. J. (2016). Unit 6: Software, p383

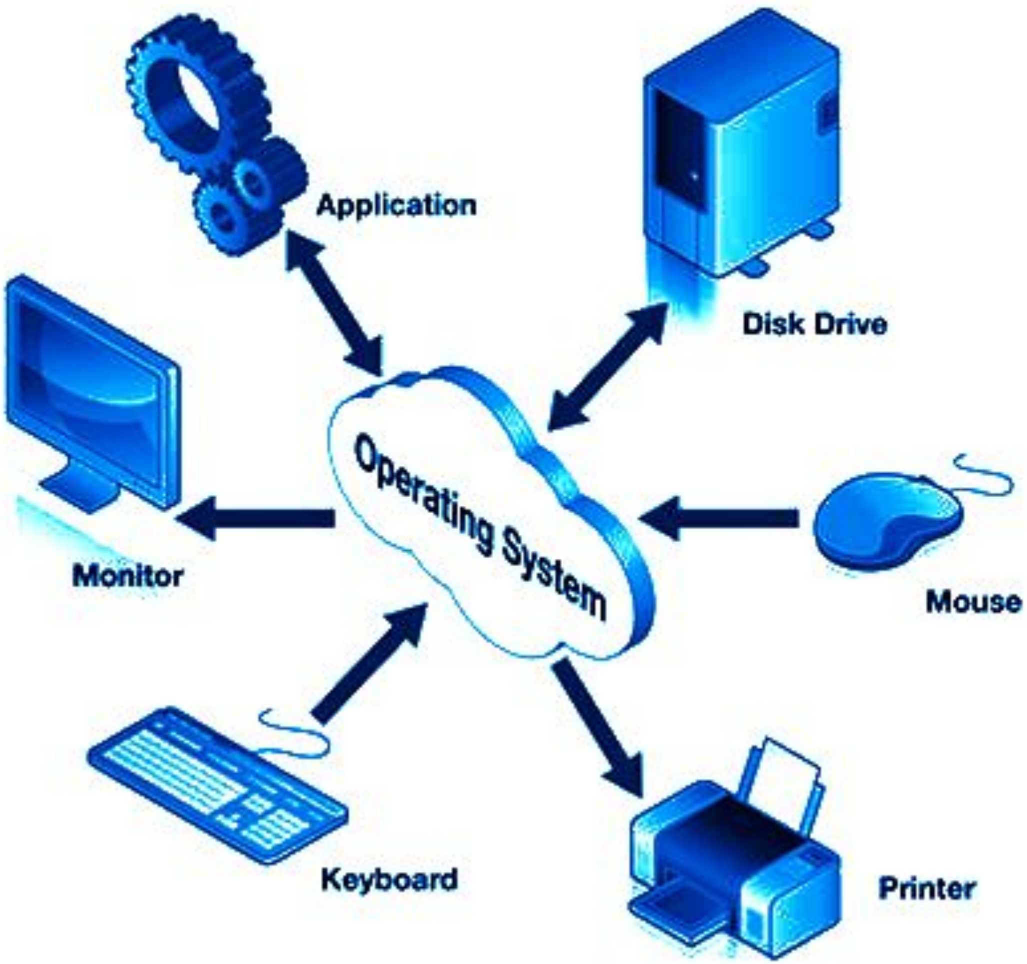
The Operating System

Performs basic tasks:

- Recognizing input from the keyboard
- Sending output to the display screen
- Keeping track of files and directories on the disk
- Controlling peripheral devices such as disk drives and printers

Ensures that different programs and users running at the same time do not interfere with each other

Provides **a software platform on top of which other programs** (i.e., application software) **can run**



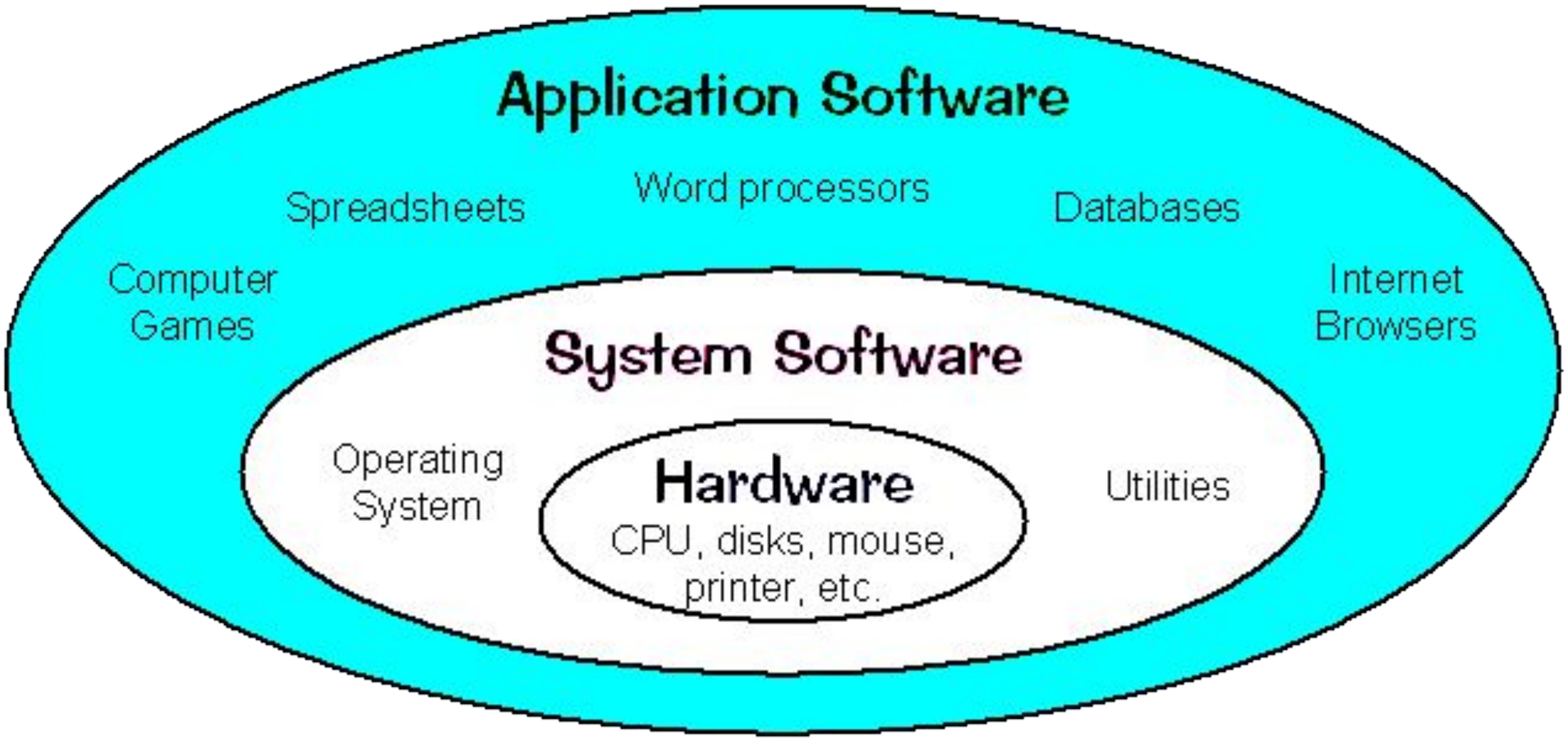
The Operating System, continued

- **User interface (UI)** allows individuals to access and command the computer system
- **Command-based user interface (e.g., commands in Terminal):**
  - Requires that text commands be given to the computer to perform basic activities
- **Graphical user interface (GUI):**
  - Uses icons and menus displayed on screen to send commands to the computer system
- **Natural user interface:**
  - Touch, voice, etc.

Application Software

Application software consists of programs designed to perform specific tasks for users

- *System software* serves as the interface between a user, the application software, and the computer's hardware



VS

OPEN SOURCE SOFTWARE

PROPRIETARY SOFTWARE

A Few Basic Definitions

- What is a **program**?
    - An organized list of *instructions* that, when executed, causes the computer to behave in a predetermined manner.
- What is **source code**?
    - Program instructions in their original, human-readable form.
- What is **object/executable code (binaries)**?
    - Program instructions in a form that can be directly executed by a computer. A compiler takes source code and generates executable code, i.e. machine code.

Proprietary Software

- Distributed in machine-readable **binaries** *only*
  - Generally not possible to reconstruct the original code in human-readable form
  - Source code is intentionally kept *secret*
- Users pay for some form of *license*
  - Grants *certain* usage rights
  - Typically major restrictions on copying, further distribution, modification, etc.
- Analogy: buying a car…
  - With the hood welded shut
  - That only you can drive
  - That you can't change the rims on

Open Source Principles

- In early computing, most software was free and open source – shared by researchers and computer scientists
  - Later on, proprietary software became the predominant (and lucrative!) model as commercial computing expanded
- Over the past 20+ years, the open source movement has emerged again as a major computing paradigm facilitated by the Internet
- The distribution terms of open-source software must comply with the following criteria:
  - Free distribution
  - Source code distribution
  - Derived works allowed
  - Full list at https://opensource.org/osd

Examples of Open Source Software

You likely use a combination of proprietary and open source technologies already

	Proprietary	Open Source
Operating system	Windows, MacOS	Linux
Office suite	MS Office	OpenOffice
Image editor	Adobe Photoshop	GIMP
Web browser	MS Edge	Mozilla Firefox
Web server	IIS	Apache
Database	MS Access, FileMaker	MySQL

Open Source

Pros

- Peer-reviewed code is high-quality and transparent
- Rapid bug fixes and iterative releases
- Released by engineers, not marketing people
- No vendor lock-in
- Simplified licensed management

Cons

- Dead-end, fragmented software
- Developed by engineers, often for engineers, i.e. not end users
- Difficulty in managing community development model
- Inability to point fingers
- Sometimes: Lack of well-written user Help docs

Open Source

What does it mean for something to be "open source"?

- the technical meaning of "open source"
- alignment with the value-based/moral/ethical aspects of "open source"
- **"FOSS" = free and open source software**
  (different things but many software are both together)

Defining:

- "Proprietary" software
- Free software
- Open source software
- Public domain software

Open Source as a means of visibility into projects:
https://www.openhub.net/

Open Source in Information Professions

- Used very, very, very (!) frequently in museums, archives, libraries and educational organizations
    - List of examples: Open Source Software for Libraries
        - http://www.libsuccess.org/index.php?title=Open_Source_Software

- Open source software runs a large percentage of the web itself
    - E.g. Linux servers, Apache web server, Firefox, etc.

- Most (if not all) information professionals will be in involved in open source software in some way

Open Source - examples

What comes to mind when you hear "open source"?

- Code is accessible
- Changeable / adaptable
- Code is reusable
- Often volunteer work
- Uses an open license
- Generally free, donation-based, or low cost
- Part of a movement -- part of an ethics of software development

What are some examples of Open Source projects? at different scales:

- Infrastructure
- Enterprise
- Personal computing
- Other?

Different models for obtaining paid software

One-time purchase. The traditional way to obtain software is through a one-time purchase in which the consumer pays a set amount to license and use the software without an expiration date. The advantage of the one- time purchase pricing model is that there are no additional fees, and with the exception of a few updates, the software remains basically the same as when it was purchased. There are no surprise changes to the way the software looks or works during its lifetime.

Subscription. The subscription pricing model is an emerging trend in which consumers pay a monthly or an annual fee to use the software. Consumers benefit because updates and upgrades are usually included in the pricing. Consumers must remain alert while using subscription services. When a subscription lapses, the software may cease to function. Some vendors allow former customers to launch the software and view files but do not allow those files to be further modified. Credit card information is stored on the vendor's site and may be vulnerable to hacking.

Trial. A third type of pricing model offers consumers the use of a software product during a free trial period. The trial version may be fully functional or it may be limited in functionality. When the trial period ends, payment in the form of a one-time purchase or subscription is required. This pricing model is common for software applications, such as antivirus utilities, games, and weather apps, that are preinstalled on new devices.

Freemium. Another popular pricing model for software provides free use of a stripped-down or basic version of the product but requires payment for upgraded features.

TCO

Total Cost of Ownership

Total Cost of Ownership, known as “TCO” is an approach used to gauge the total investment that a particular purchase or implementation will require across its “lifespan”, rather than focusing on its initial purchase price alone.

For example, there may be an initial purchase price (e.g. proprietary software) or no initial cost (FOSS software), yet either will likely require other cost and/or time investments for things like training, installation, and technical support. For example, adopting a particular hardware may require adopting new software to support using it effectively. For example, proprietary software may require regular upgrades or monthly user fees.

$ --> time --> $$$$

INFORMATION LABOR / TECH LABOR / TIME

Time and labor as factors in software/hardware implementation and maintenance

Labor is always a part of implementing software and hardware. It may be a greater or lesser amount and it may be more or less technical, but it should always be factored in to organizational planning. It can be conceived of as a kind of "cost" related to TCO.

What kinds of labor from which staff might be involved in:

- evaluating and selecting and purchasing hardware/software
- installing hardware/software
- training staff or developing help materials for staff/clients/customers/visitors
- maintaining and reworking hardware/software as organizational needs change over time

“BUY IN”

The soft skills (and sometimes hard realities...) of software/hardware implementation

Aside from the relatively measurable concerns of costs and labor, software and hardware implementations involve the question of adoption, or “buy in,” on the part of users.

The tools are nothing without the users using them!

Organizations have internal cultures and that culture needs to shift to include new tools.

Activity

Explore Examples of Open Source Software

- Use the links in Canvas to navigate to one or more open source software tools potentially of use to information professionals *(see next slide)*
- Imagine that you are evaluating it for potential use by your team/organization -- Can you find their statements on being open source and anything about their history and support? Do they have demo examples that you could evaluate? What other factors would you look at to determine if they were a good choice for your organization's needs…?

Explore examples of open source software:

choose 1 or 2 of these open source software to explore, imagining that you were evaluating it for potential use by your team/organization -- Can you find their statements on being open source and anything about their history and support? Do they have demo examples that you could evaluate? What other factors would you look at to determine if they were a good choice for your organization's needs…?

Omeka

https://omeka.org/

collections management

CiviCRM

https://civicrm.org/

contacts, funding drives, and more

Collective Access

https://www.collectiveaccess.org/

collections management

Blender

https://www.blender.org/

drawing, animation, 3D

see you next week!
and I'm here after class in 607 if you have Qs
(or email: rdaniell@pratt.edu)