Philosophy of Information Technology

Lecture 2

Mario Verdicchio

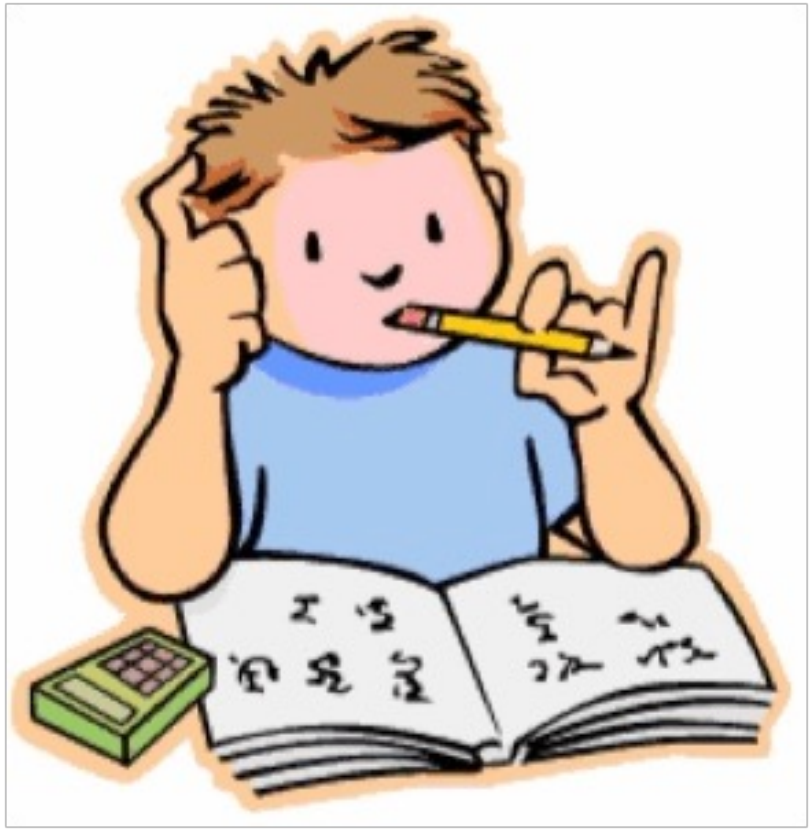
Università degli Studi di Bergamo

Academic Year 2025-2026

Now let's go inside the computer to study its main components and how they work.

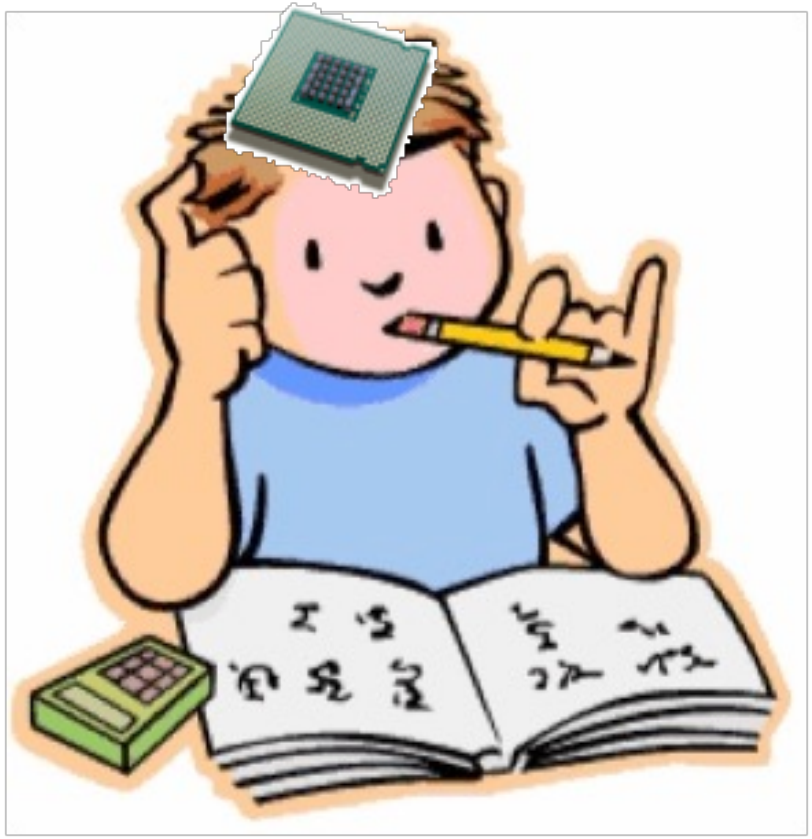
Hardware

Components of a computer



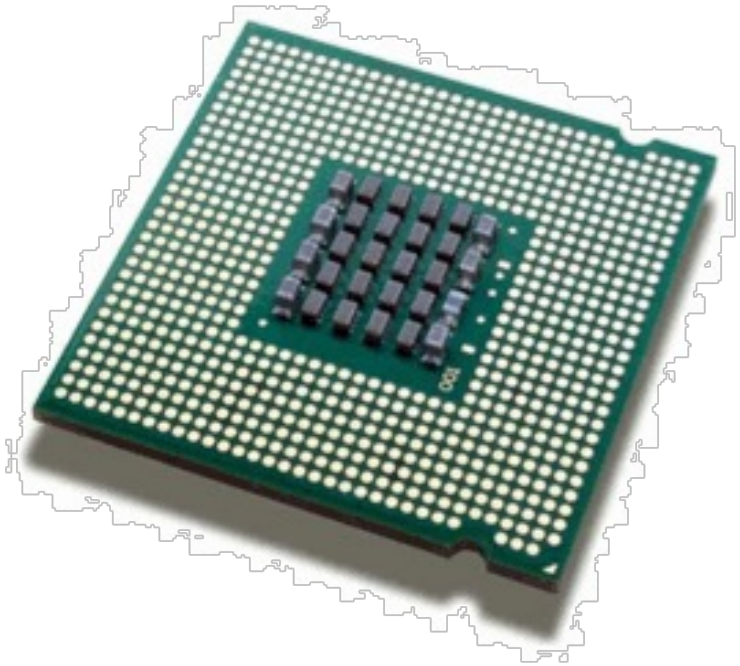
We will illustrate the components of a computer by comparison with what a human being uses to work on data.

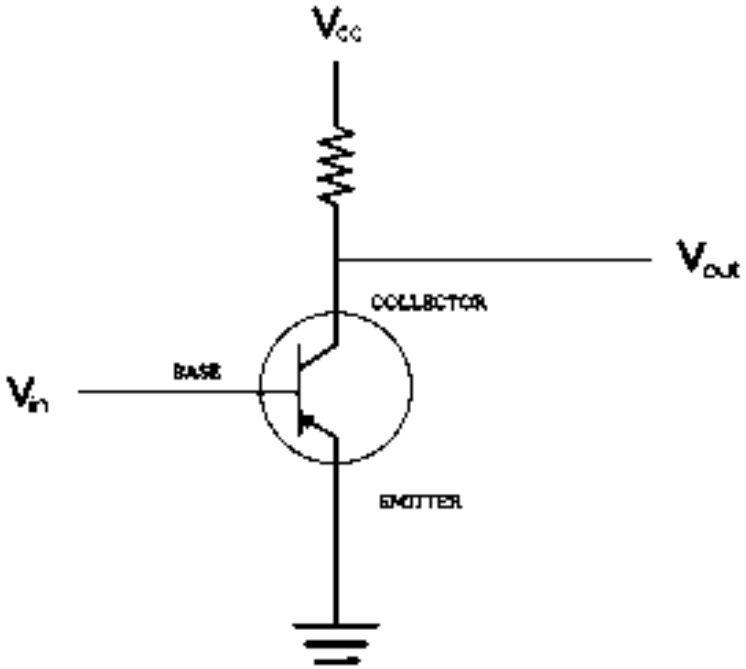
The most important component is the CPU (Central Processing Unit). In this human/computer comparison, the CPU corresponds to the brain, because it is the part of the computer that performs operations.



CPU

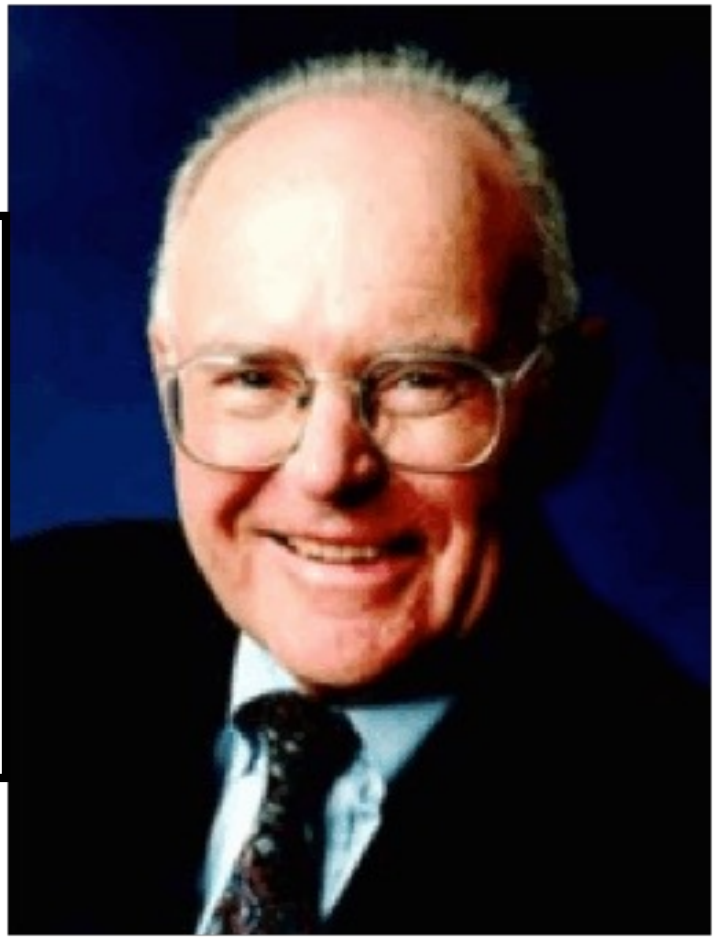
The CPU is an electronic circuit made up of a significant number of transistors. Each transistor is built in a way that it reacts to electrical input stimuli (V_{in} in the figure) with a specific output electrical voltage (V_{out} in the figure). When we say that the CPU performs operations, we mean that it receives electrical input signals that represent the data and indicate the operation to be performed.





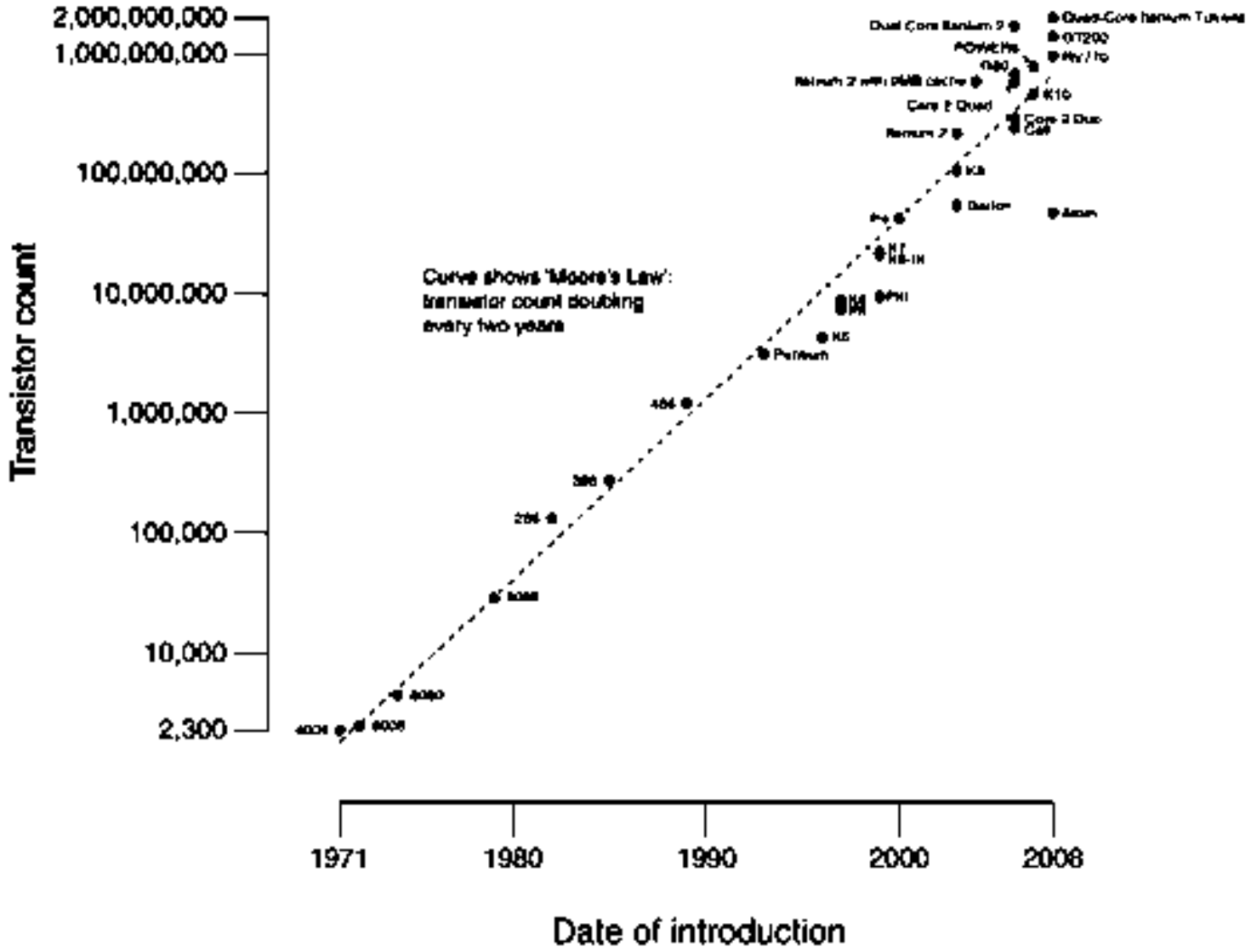
The CPU transistors react to stimuli and output other electrical signals, which represent the result of the operations performed.

The transistors in a CPU are very numerous (today they reach 2 billion per processor), and have continued to grow in number since the first CPUs were built. The trend is curiously constant, and predicts a doubling every year and a half. The first to notice this trend was Gordon Moore, co-founder of Intel, one of the largest CPU manufacturing companies. This phenomenon, known as “Moore's law”, is named after him.



Gordon E. Moore

CPU Transistor Counts 1971-2008 & Moore's Law

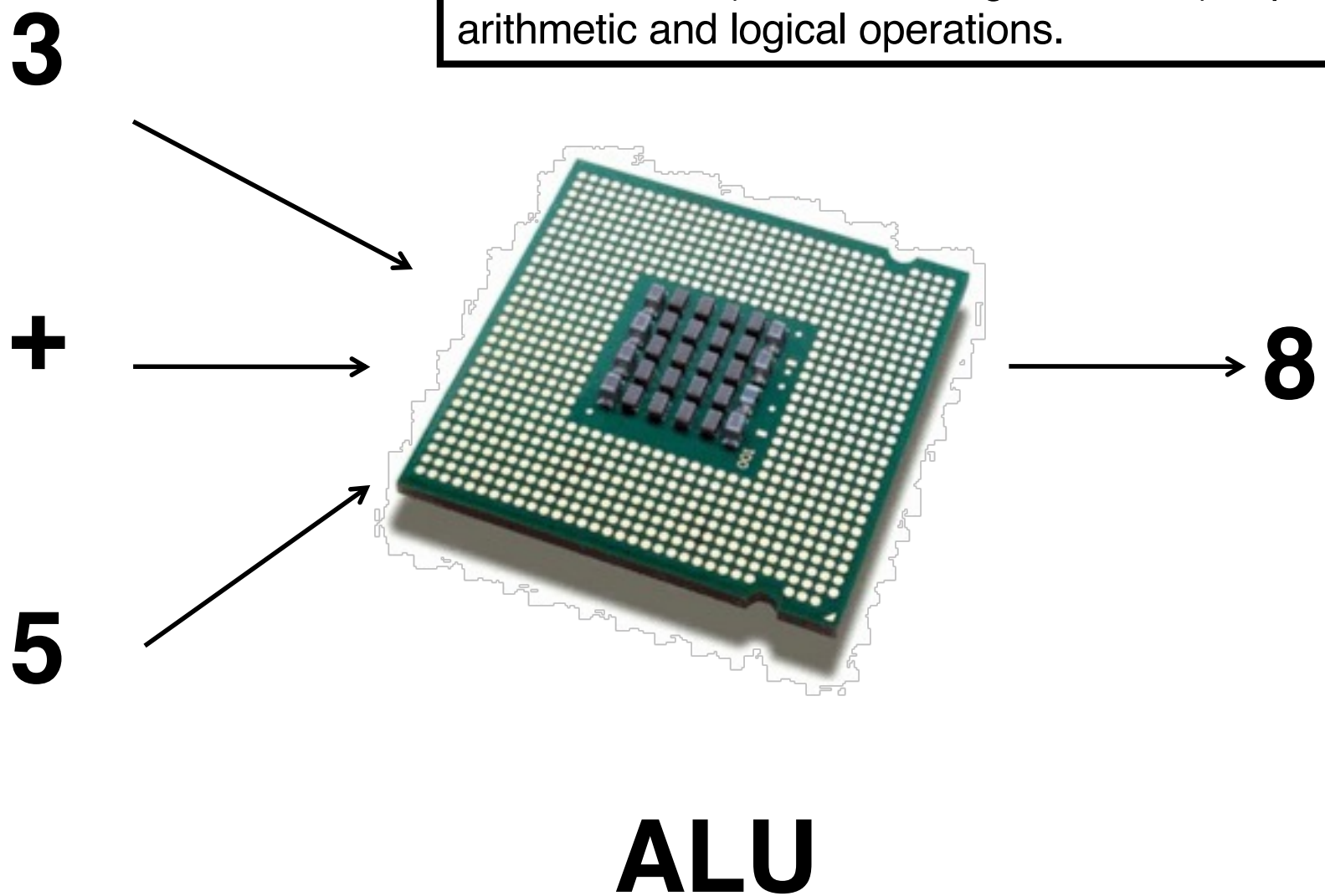


Despite some adjustments in the following decades, the pattern detected by Moore seems to hold. What can we make of this? There are at least two observations that should prevent us from jumping to apocalyptic or utopistic conclusions about the future of IT.

Firstly, there are physical limits to hardware given by the laws of physics all material entities are subject to. It is true that transistors can become smaller and smaller, but they cannot be smaller than one atom. The curve of the number of transistors per chip can be modeled as an exponential, but there is a cap. More generally, we must not take a mathematical model as a realistic depiction in all its parts, so even if Information Technology has shown exceptional growth in the past decades, this may not be the case in the future.

Secondly, we must not forget about the quantity vs quality (signs vs meanings; syntax vs semantics) dichotomy: an increase in transistor density surely leads to faster computing machines, which in turn means that a greater quantity of computational operations can be carried out per unit of time, but this does not entail that certain tasks will become amenable to machine operation.
There is a distinction between unfeasible and impossible tasks: an unfeasible task is one for which there is a computational solution, but it requires so many computational resources that it is not reasonable to tackle it; an impossible task is one for which there is no (known) computational solution.
Breaking a cryptography-based protection is currently unfeasible, but it may become much easier once quantum computing, i.e. computation exploiting quantum mechanics phenomena, becomes available thanks to a technological breakthrough.
Computing what happens in our brains when we understand a word is, instead, an impossible task, since we do not know how consciousness is produced in the brain, nor whether that mechanism can be obtained via computation. Increasing the number of operations that a computing machine performs in a unit of time will not change this.

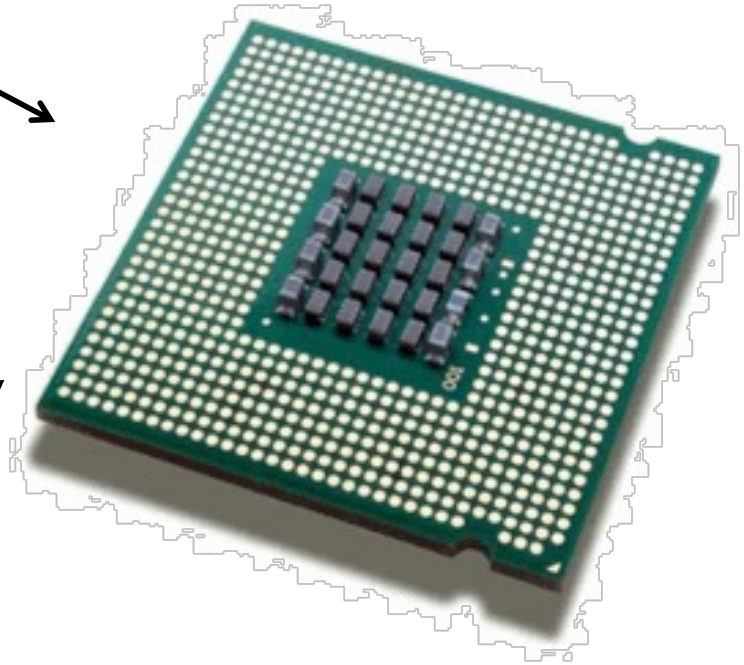
The CPU is actually not a unitary block, but we can distinguish two parts based on their function.
The ALU (Arithmetic-Logical Unit) performs arithmetic and logical operations.



ALU

The CU (Control Unit) is the part that controls the transfer of data from one part of the computer to another. In fact, everything that happens in the computer happens because electrical signals have been emitted by the CU: even the input of data into the ALU, for example, is controlled by the CU which executes the instructions written in the programs given to the computer.

print



“ciao”

CU

But where do the data that are processed by the CPU come from?

3

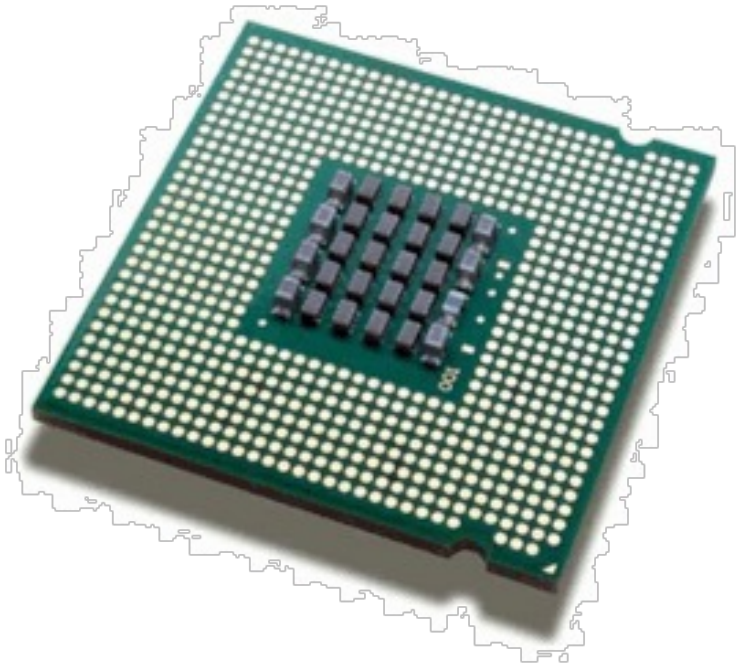
print

+

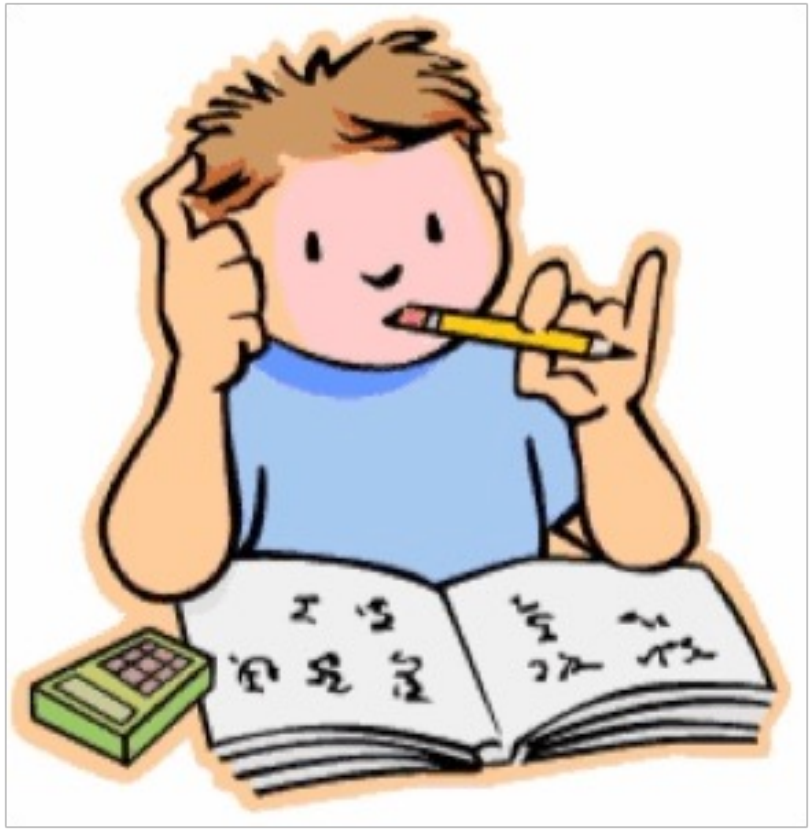
5

“ciao”

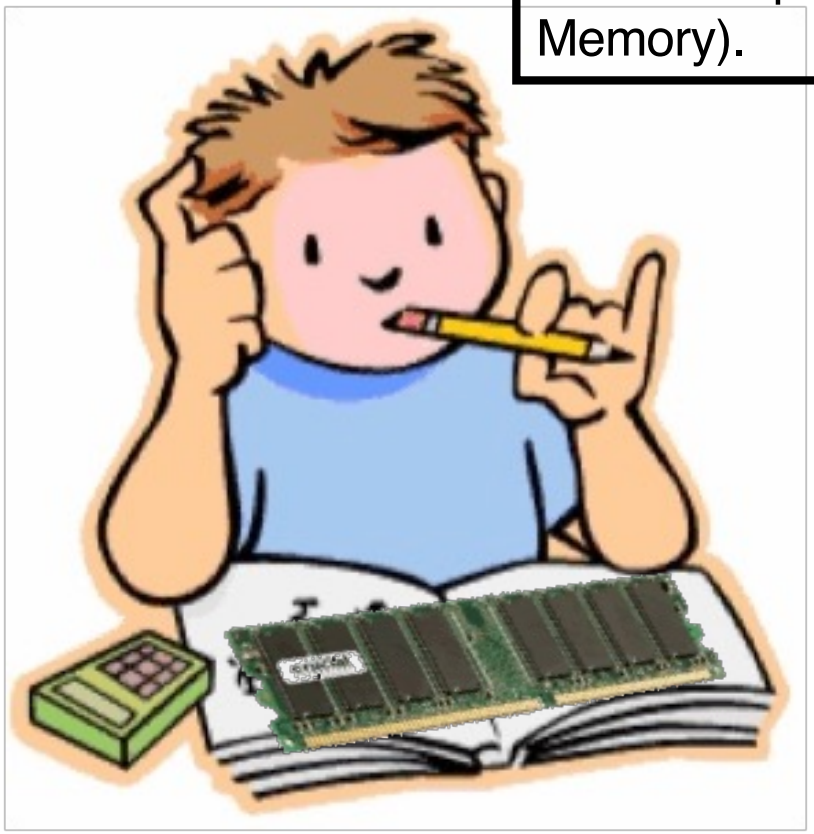
?



Let's go back to the comparison with the person who is processing data.
The data of a problem are written in their notebook.

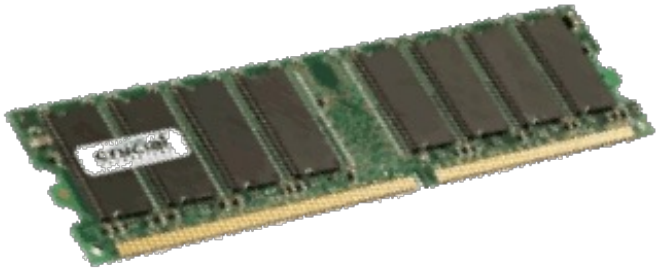


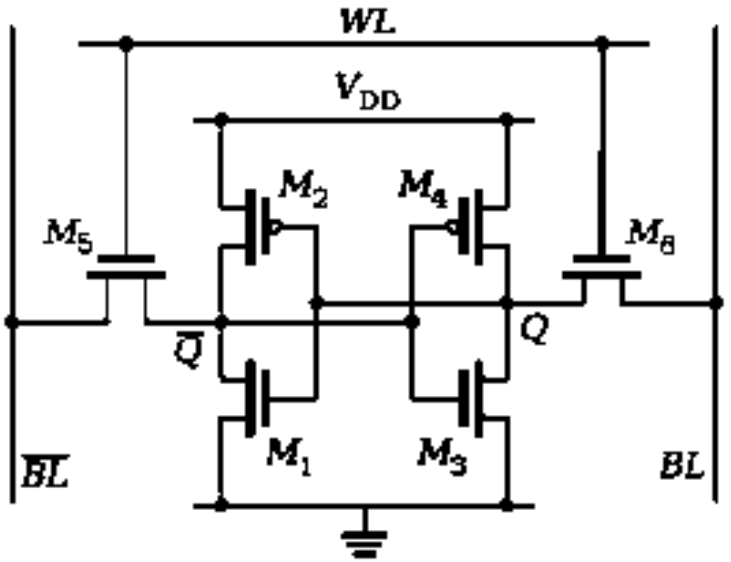
Here is another component of the computer, which corresponds to the notebook from which the person takes the data, and in which they write down the intermediate steps and the final results.
This component is called RAM (Random Access Memory).



RAM

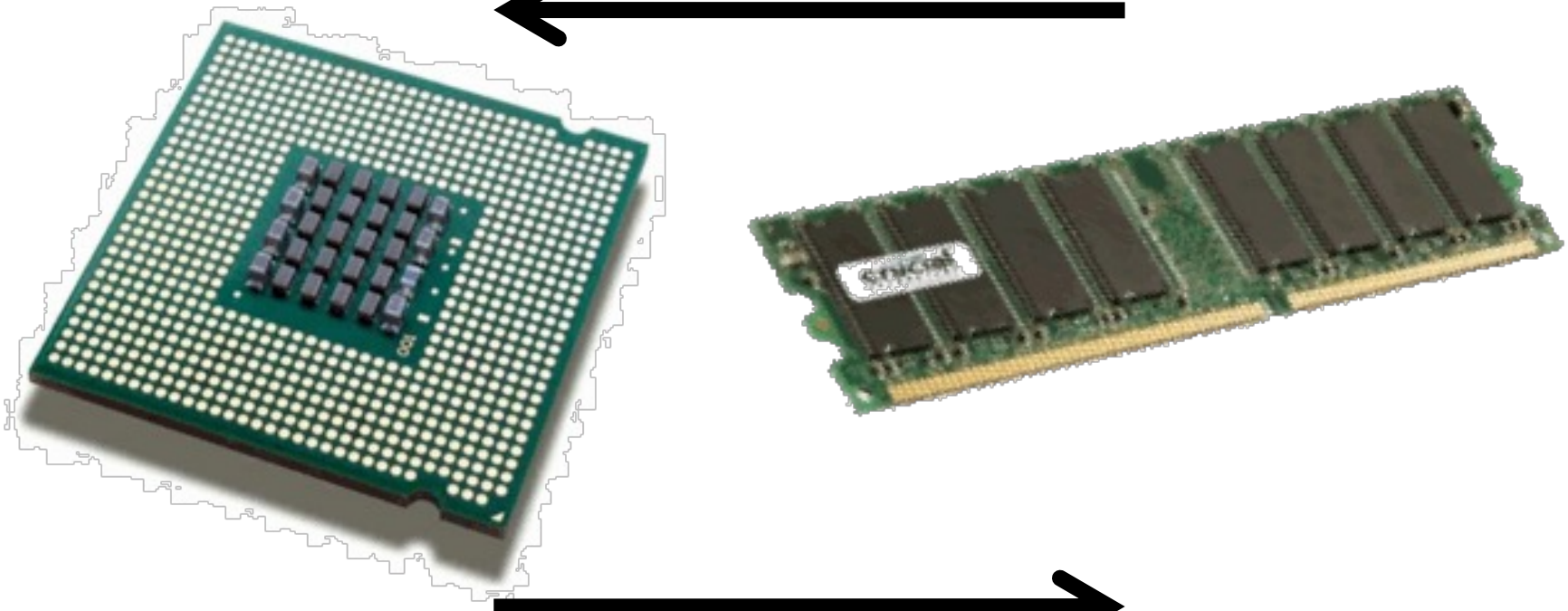
The RAM is also made up of electronic circuits with numerous transistors. The configuration appears more complex because the purpose of the RAM is not to respond to electrical stimuli by sending output signals, but to store data internally. Building a memory requires a more complex composition of transistors.





The name “RAM (Random Access Memory)” is meant to highlight a specific characteristic of this device: it allows access to data contained in any position of the device. You can choose a random access point, and the RAM is able to provide the data contained at that point. The name in Italian (“memoria centrale”), instead, highlights the central role played by the RAM inside the computer: the RAM works very closely with the CPU.

operations, operands



results

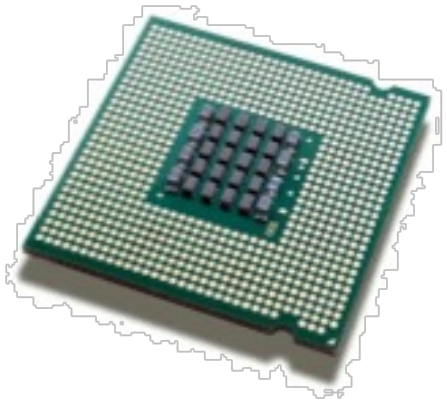
The operands and the operations that the CPU must process come from the RAM.
The results obtained in output from the CPU are, in turn, sent to and stored in the RAM.

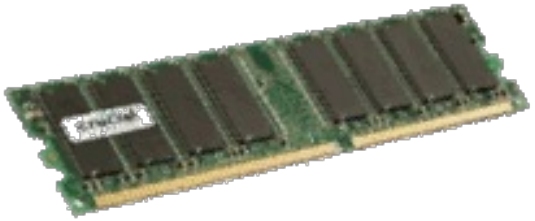
Let's now see the other components that participate in the computer's operations: clearly with only the CPU and RAM we cannot display the result of an operation on the screen.

3

+

5



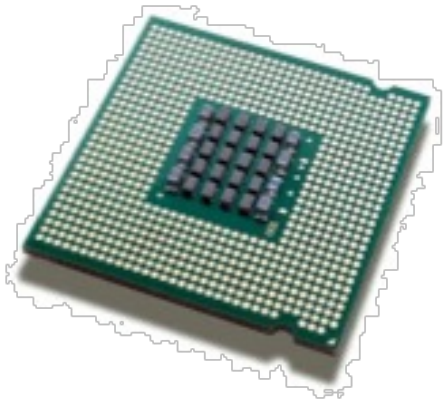


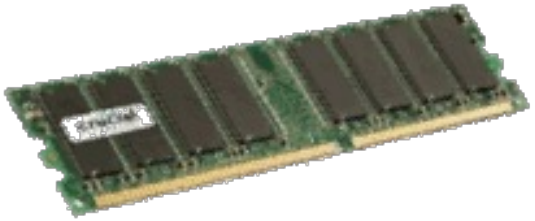
We said that the data and operations to be performed are stored in RAM.

3

+

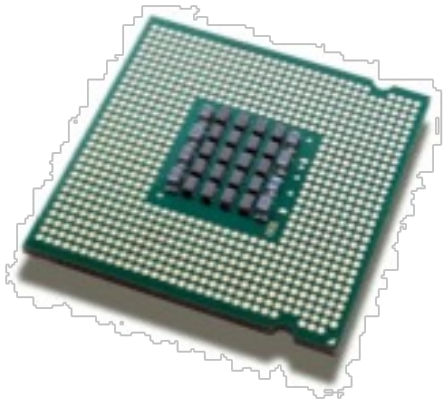
5

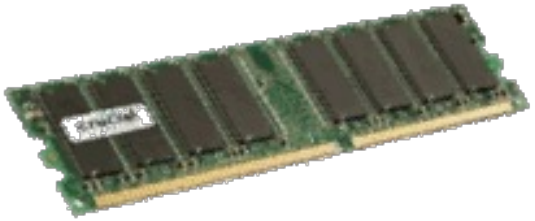




From the RAM, they are transferred to the CPU, by order of the CPU itself (or rather, of the CU) which takes instructions to be executed from the RAM continuously, from the moment the computer is turned on up to when it is switched off.

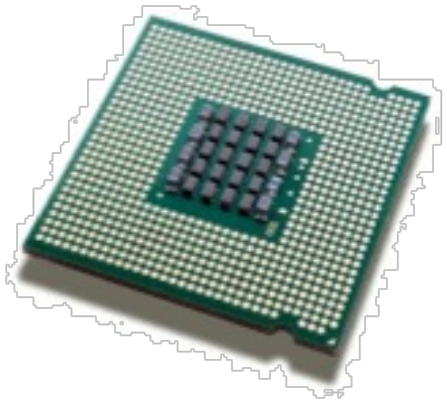
8

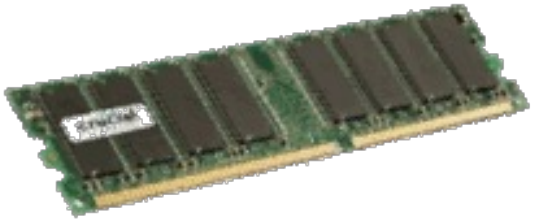




Once the output result from the ALU is obtained, it is sent to the RAM...

8

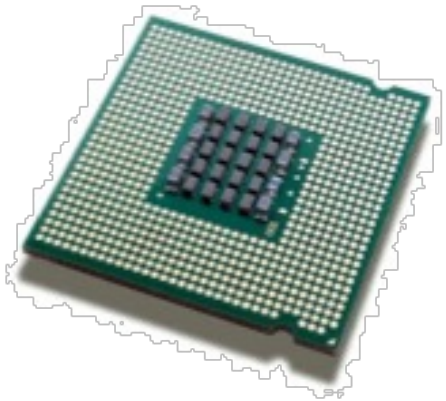
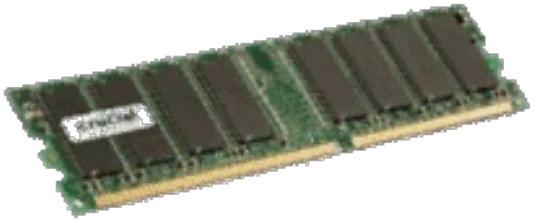




…where it is stored.

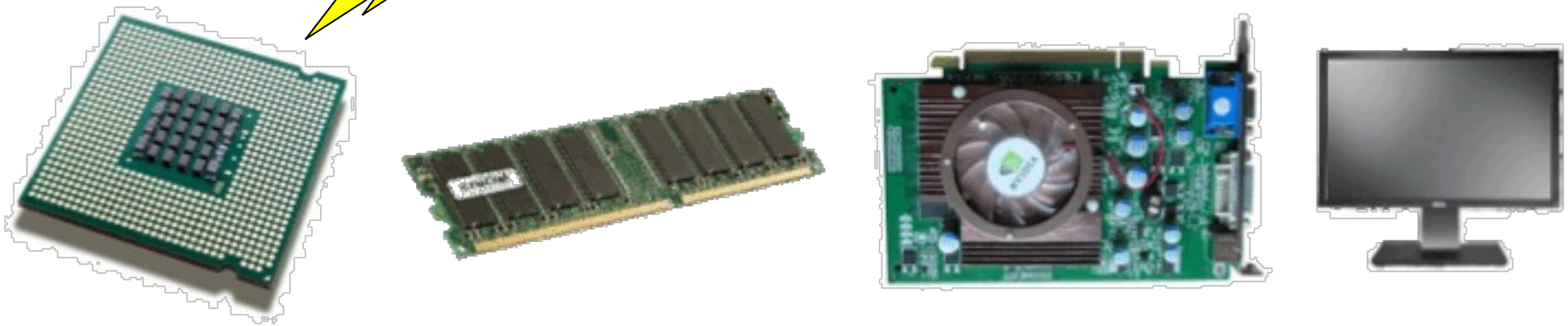
⟵ print

8

The CPU continues to take instructions from the RAM and let's suppose that the next instruction commands a printout of the result of the previously performed operation on the screen.

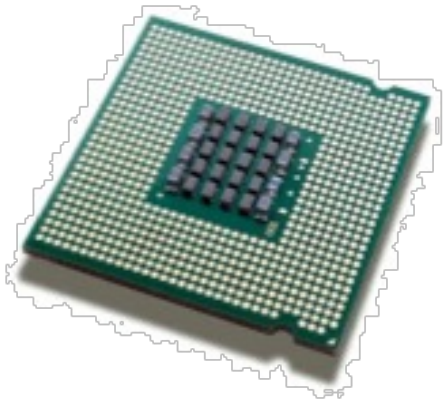
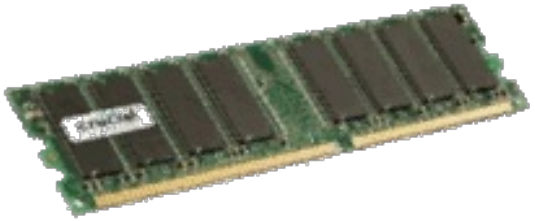
8



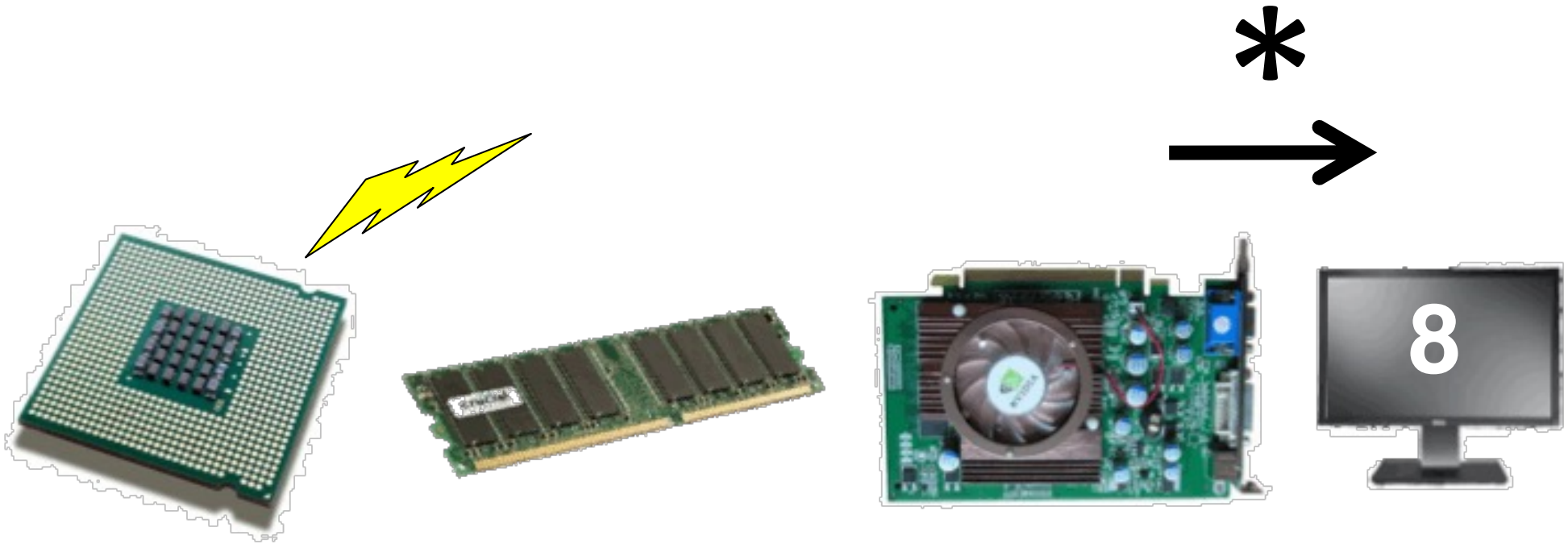
The CU in the CPU receives the command and sends appropriate signals to carry out the transfer of the result from the the RAM to the screen.

Graphics card

→ 8

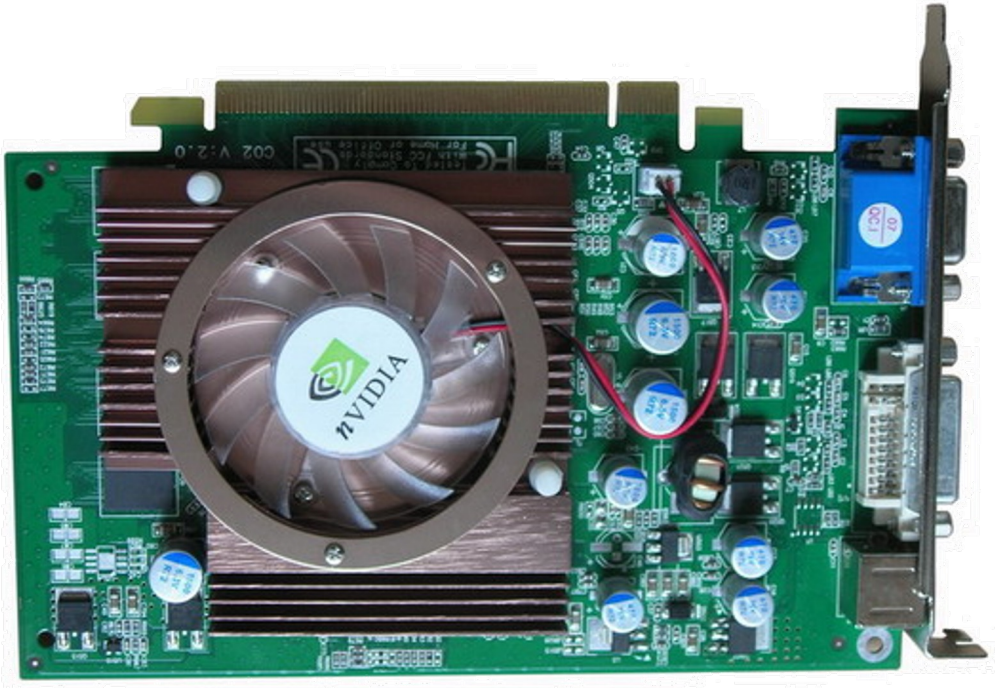



The signals from the CU ensure that the data is transferred from the RAM to the component that manages the signals to be sent to the screen: the graphics card...

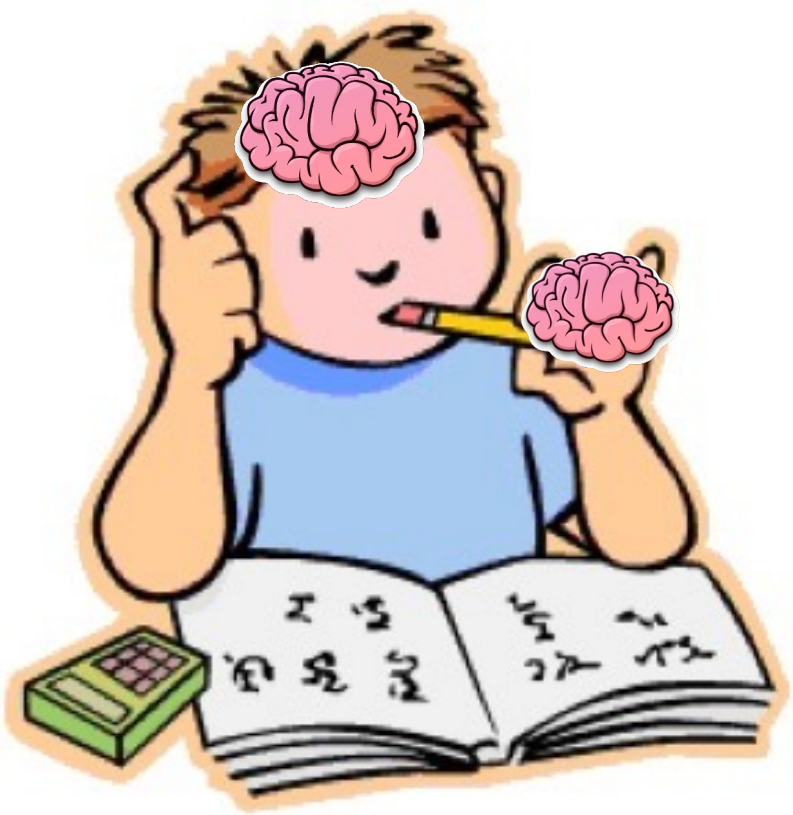


…and from the graphics card to the screen. This last transfer is more complex than the others because we have to shift from the data inside a computer to physical phenomena suitable for the human beings in front of the screen: no longer electrical signals, but rather light waves that represent a sign.

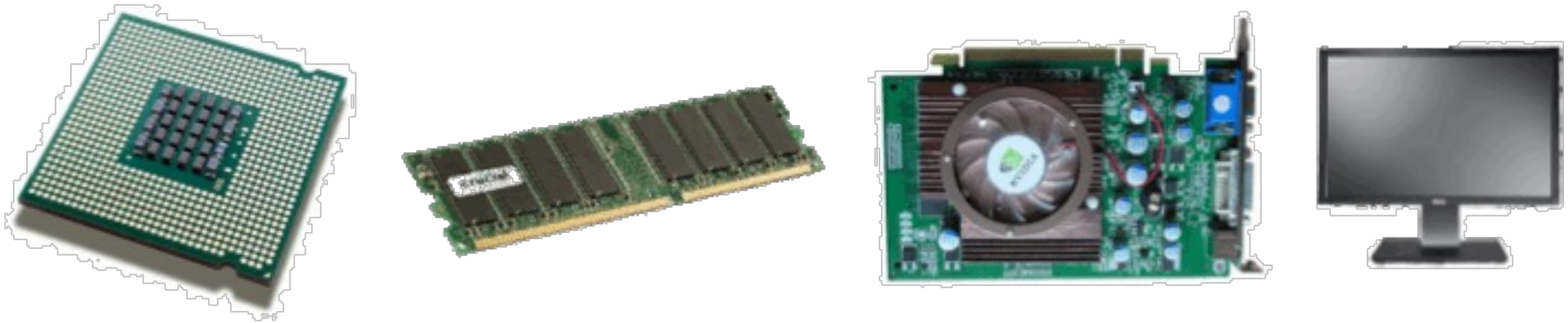
The graphics card shows that it has a fan. In reality, even the CPU is always accompanied by a fan, which turns on as soon as the computational work raises the temperature of the circuits, bringing it closer to an alert threshold. Without ventilation, each processor would melt due to its own activity (moving electrons from one side to the other produces heat). In fact, the graphics card has its own processor, the GPU (Graphical Processing Unit), to which the central CPU delegates the execution of instructions related to display of data on the screen. The CPU can handle displaying a simple '8' easily, but in the case of more demanding graphics applications such as electronic games, the CPU relies on the GPU to be able to work on other things.



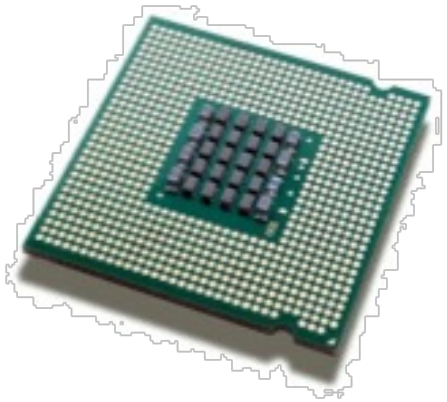
GPU

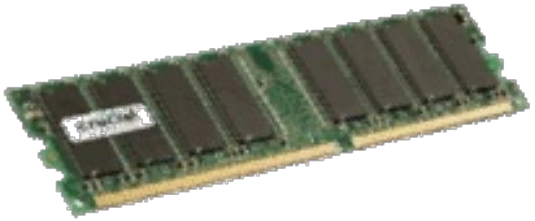


It is a bit as if the human’s hand had a small local brain that guides it in writing, without engaging the main brain.



There is another reason to place an asterisk on the transfer of data from the graphics card to the screen.





This is the first transfer that leaves the confinement of the box that encloses the computer's internal components.

Peripheral

The external components of the computer are called peripherals.
The screen is, in fact, a computer peripheral.

Input peripheals

Input peripherals are those that allow the flow of information from the outside to the inside of the computer.

Output peripherals

Output peripherals, vice versa, are those that allow the flow of information from inside the computer to the outside.

Did you notice that the Wii controller was in both peripheral groups? This is because there are peripherals that allow the flow of information in both directions. In the case of the Wii controller, it allows the user to move the cursor in the Wii menu or control the game (input), but, in certain game situations or when the cursor passes over a selectable menu item, it vibrates, providing information to the user (output).

In the following slide, the Wii controller is shown with other peripherals that are both input and output.

Wii controller (Nintendo)

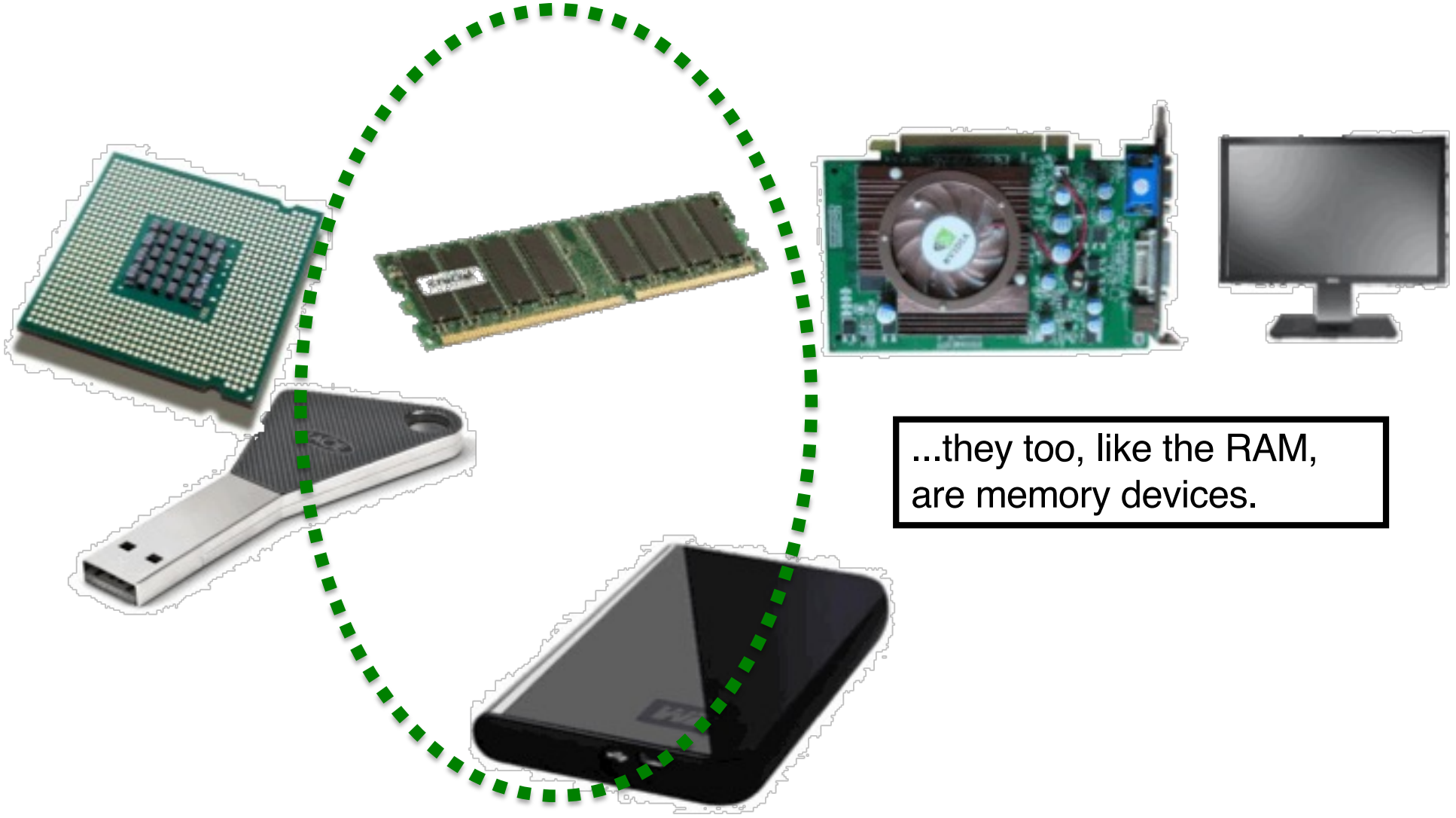
PS3 Dual Shock controller (Sony)

USB stick (Lacie)

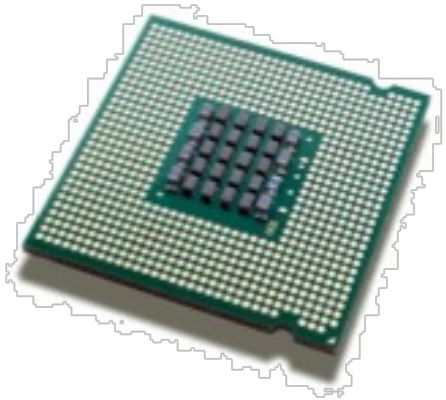
External Hard Disk (WD)

iPhone touch screen (Apple)

In particular, let's focus on the USB stick and the external hard disk...



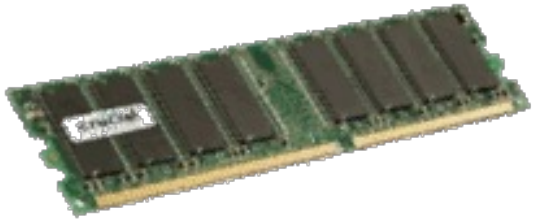
...they too, like the RAM,
are memory devices.



HD

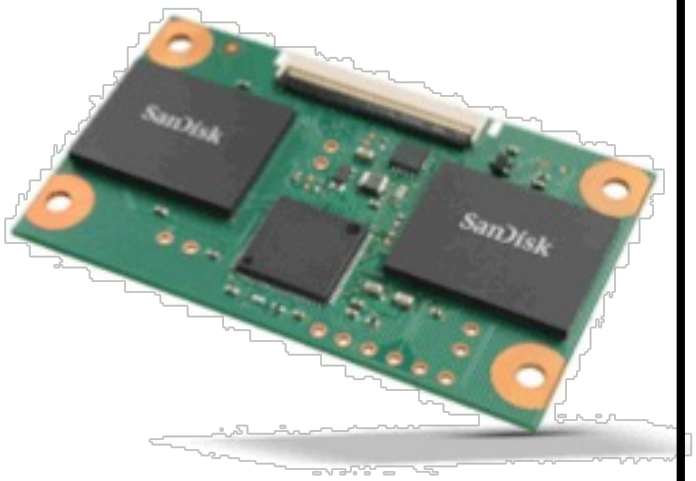
There is another memory device internal to the computer: the hard disk, alkso known as the hard drive.

Let's compare the RAM with the HD. They have complementary characteristics that make them both necessary for the proper functioning of a computer.

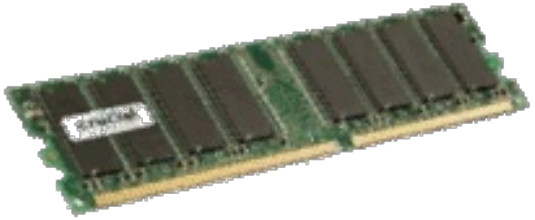


It is an electronic device made of transistors.	It is a magnetic device made of disks and a read-and-write head.
It is volatile: it retains its stored contents only while it is powered by electricity. Once the computer is turned off, it loses its contents without electricity	It stores data in the form of oriented magnetic domains in its disks. Like magnets, they never run out, and retain information even without electricity. It is non-volatile.
It is very fast and therefore manages to work in sync with the CPU.	It is very slow, so it is used to retain information when the computer is turned off.

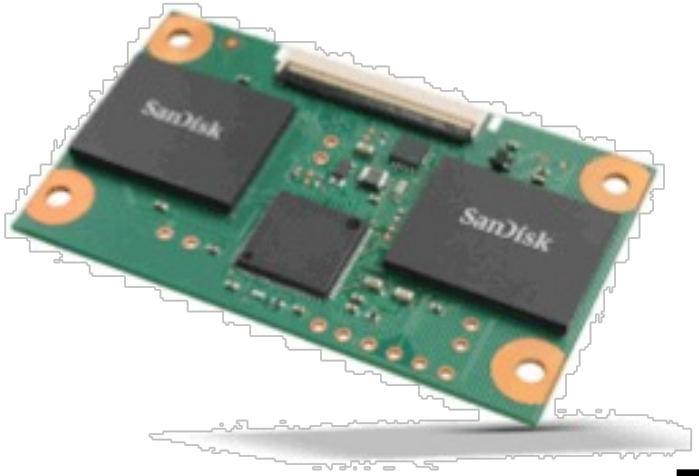
SSD



Recently, a new type of HD is available as an alternative to the classic magnetic HD, called SSD (Solid State Disk). It has the same non-volatility characteristic as the magnetic device, but it is based on electronic technology, therefore without mechanical parts (disks, heads) which can fail, and it is faster.

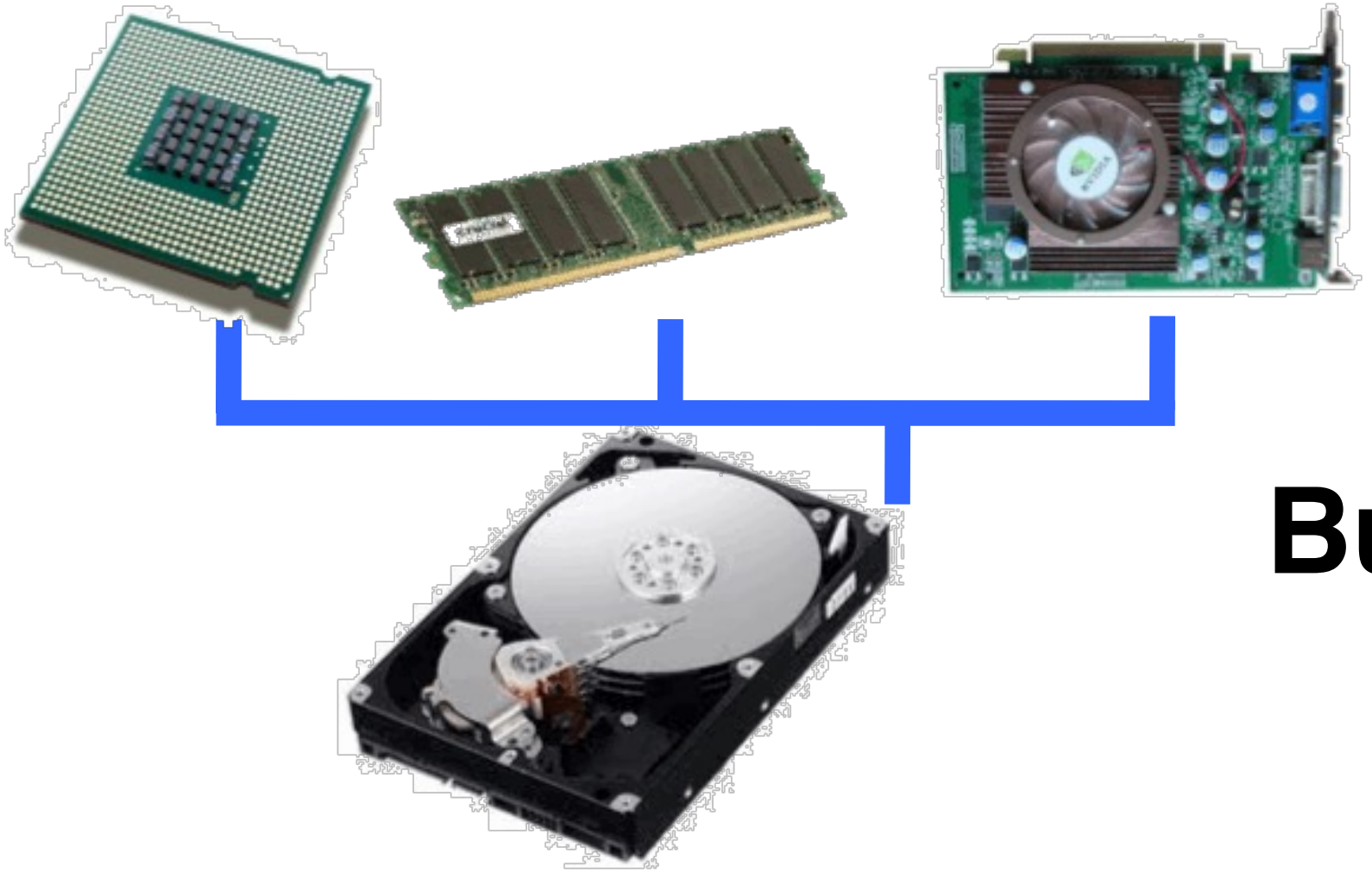


One wonders why the SSD is not proposed as an alternative to volatile RAM. In reality, the SSD, like other non-volatile electronic memories (USB sticks, "Secure Digital" SD cards typical of digital cameras), although faster than magnetic HD, are still much slower than RAM, and not they would be able to work in sync with the CPU (which can work at frequencies of 3 GHz, i.e. perform 3 billion operations per second).



The SSD (or the HD) is where all the information must be stored before the computer is turned off. When you “save” a document, what you are doing is ordering it to be stored on the SSD (or HD). The SSD (or HD) is the origin of the data and instructions that go from the RAM to the CPU. First of all they are transferred from the SSD (or HD) to the RAM. This is what happens when you turn on a computer: you have to wait for some time because this transfer is happening.

All components of a computer are connected to each other through a channel that carries electrical signals representing data from the CPU to the various parts of the computer. The channel is called bus. When you connect a peripheral to your computer (a printer or a “Universal Serial Bus” USB stick), you are actually connecting it to the bus to make it part of the system.



Tools for **data processing**

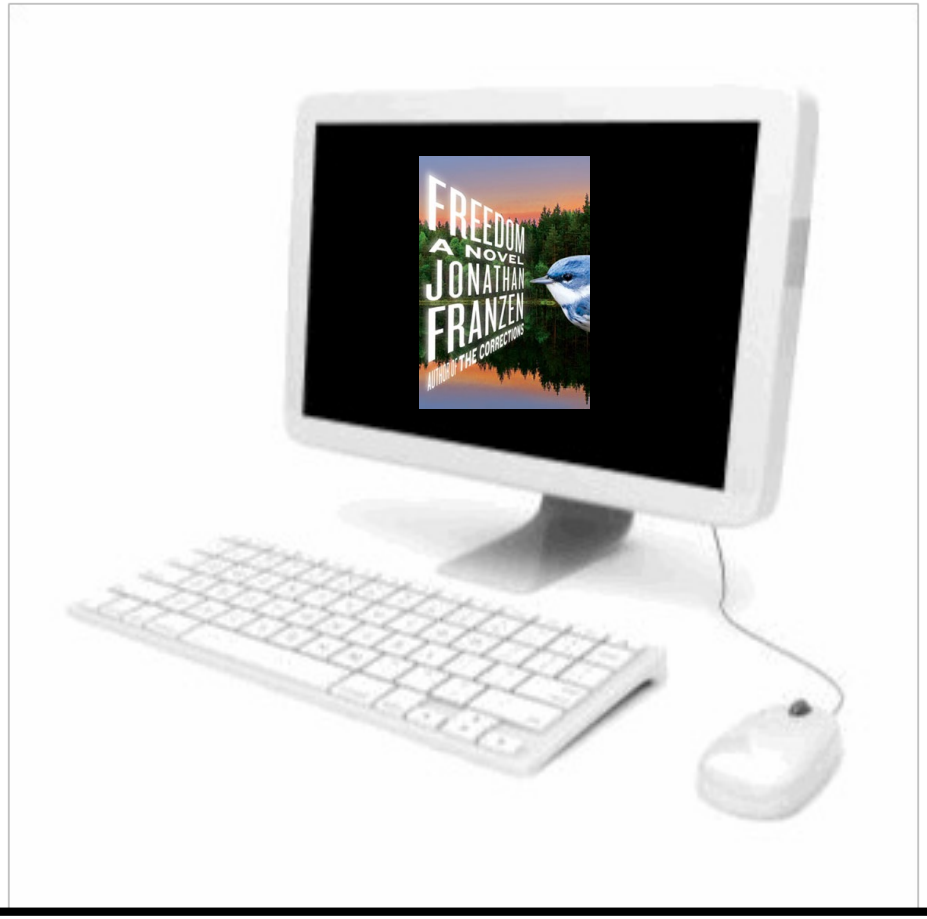
This lesson focuses on what data/symbols/signs can be processed using computers.

8

symbol **symbol**

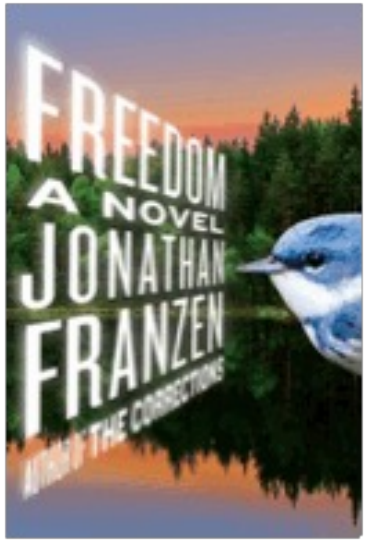
In computer science, the term “symbol” has a much weaker meaning than in the humanities: by “symbol” we mean any sign, such as a figure, without any associated meaning, as happens instead for the lion, to which we associate by symbolism the concept of “courage”.

Having simple numbers on a screen seems a bit limiting compared to what we usually see happening on the computers around us.



Authors, for example, write entire novels on their computer.

And, certainly, at least these three activities take place on the computers of the students of the course: social networks, listening to music, and watching films.

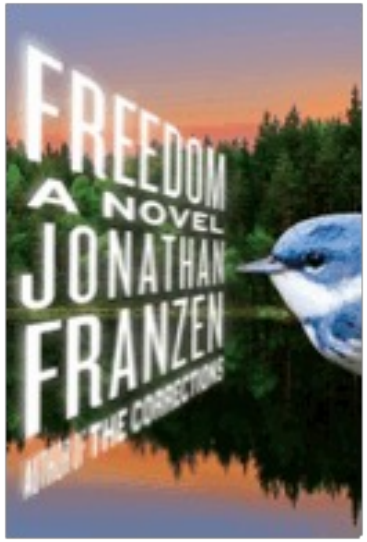


Let's think.
Premise 1: computers are nothing more than automatic executors of operations on symbols.
Premise 2: with computers we write novels, surf social networks, listen to music, watch films.

8

Conclusion:
writing novels, surfing social networks, listening to music, watching films are operations on symbols.

The reasoning is flawless, but the conclusion may leave you perplexed.



To better understand the conclusion of the previous reasoning, it is necessary to clarify the link between all these activities and the symbols processed by a computer.

8

encoding

This link is given by an encoding.

ENCODING

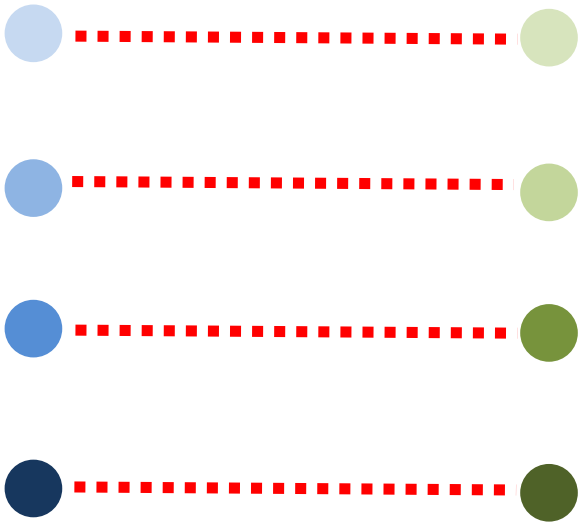
is

EVERYTHING*

*in digital technology

ENCODING [ɪnˈkəʊdɪŋ]: biunivocal correspondence between a set of entities of any kind and a set of natural numbers.

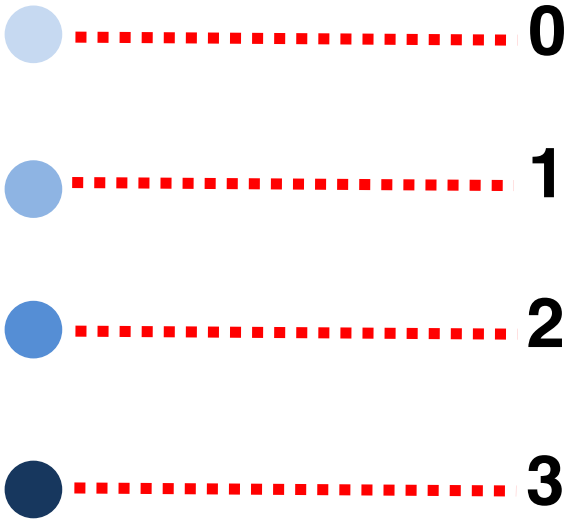
biunivocal correspondence



between a set of entities of any kind

and a set of natural numbers.

ENCODING

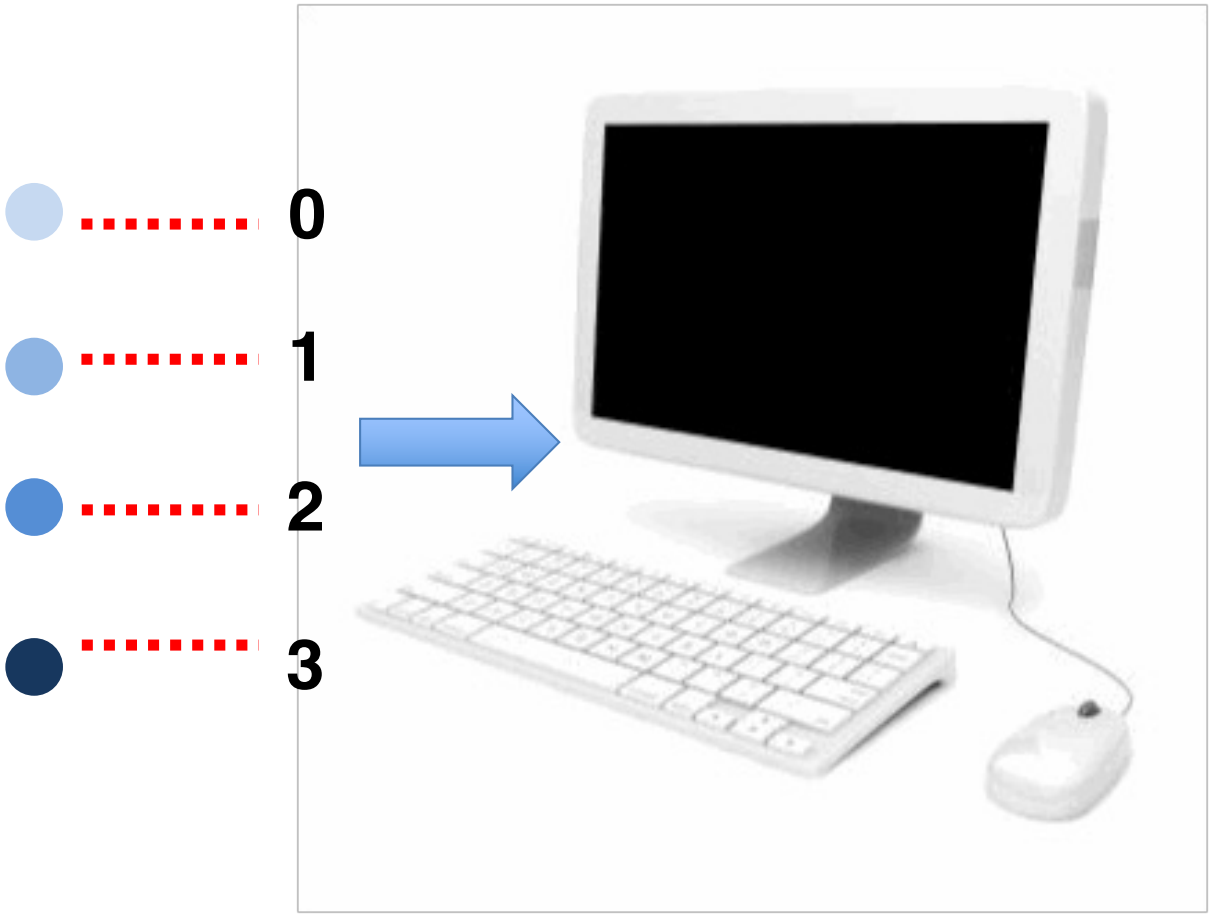


***from a conceptual perspective**

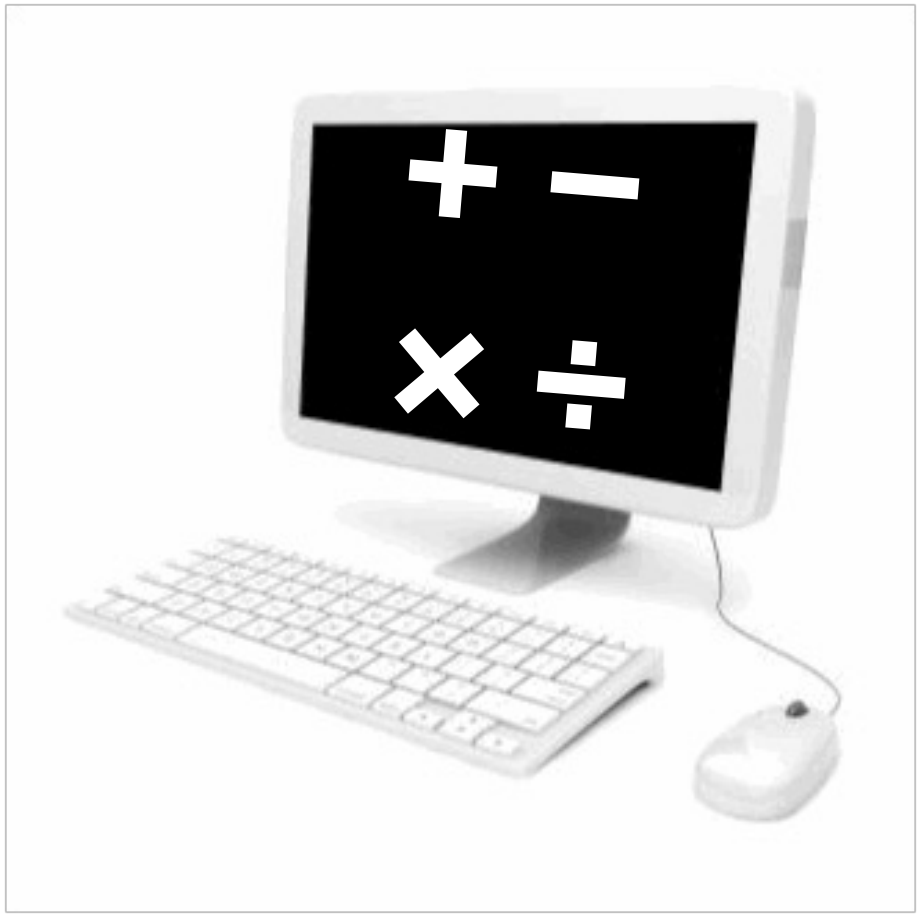
A computer.

It only works with numbers.

It cannot work with anything else.



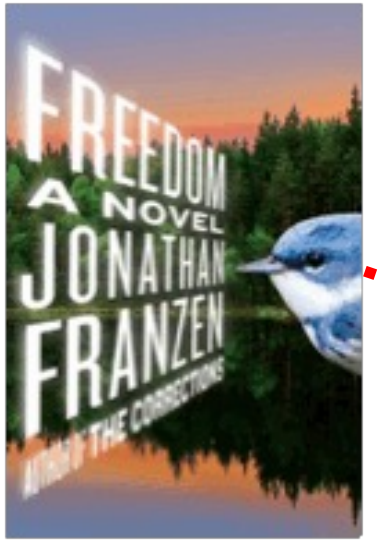
Input needs to be ENCODED.



The computer works.

Output needs to be DECODED.

COMPUTER SCIENCE



Let's see how we can create an encoding for texts.

8

The encoding of texts is based on the encoding of the characters that compose them: if I match each letter and each punctuation mark with a number, I obtain a one-to-one correspondence between letters and numbers. For this correspondence to be useful, it must be known and shared by all those who want to use a computer to exchange the texts thus encoded.
This table shows a well-known encoding: UTF-8

SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	007F
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

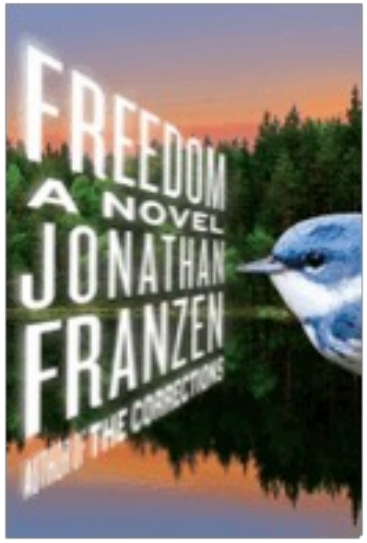
UTF-8

Universal Character Set Transformation Format – 8bit

The news about Walter Berglund wasn't picked up locally—he and Patty had moved away to Washington two years earlier and meant nothing to St. Paul now—but the urban gentry of Ramsey Hill were not so loyal to their city as not to read the *New York Times*. According to a long and very unflattering story in the *Times*, Walter had made quite a mess of his professional life out there in the nation's capital. His old neighbors had some difficulty reconciling the quotes about him in the *Times* ("arrogant," "high-handed," "ethically compromised") with the generous, smiling, red-faced 3M employee they remembered pedaling his commuter bicycle up Summit Avenue in February snow; it seemed strange that Walter, who was greener than Greenpeace and whose own roots were rural, should be in trouble now for conniving with the coal industry and mistreating country people. Then again, there had always been something not quite right about the Berglunds.

In order to be processed by a computer, the words of Jonathan Franzen's novel "Freedom" must be encoded in the form of a sequence of numbers. Each number corresponds uniquely to a character in the text.

1089273504563453476538450987530948570398456287650394175389457345893490583409584478262962042378623864378261781623727621304876276347801236402374628374676376437642087018287363475658056565827602786350827165082173658271365027813560821736565738027835623874602156060984652465745680380475693458309485630945630745603847563847650183746508731465013874650783456038475603748657834657465873450618374560183746501837645675840387657348018736457108857346578310458713465317845110387465 0



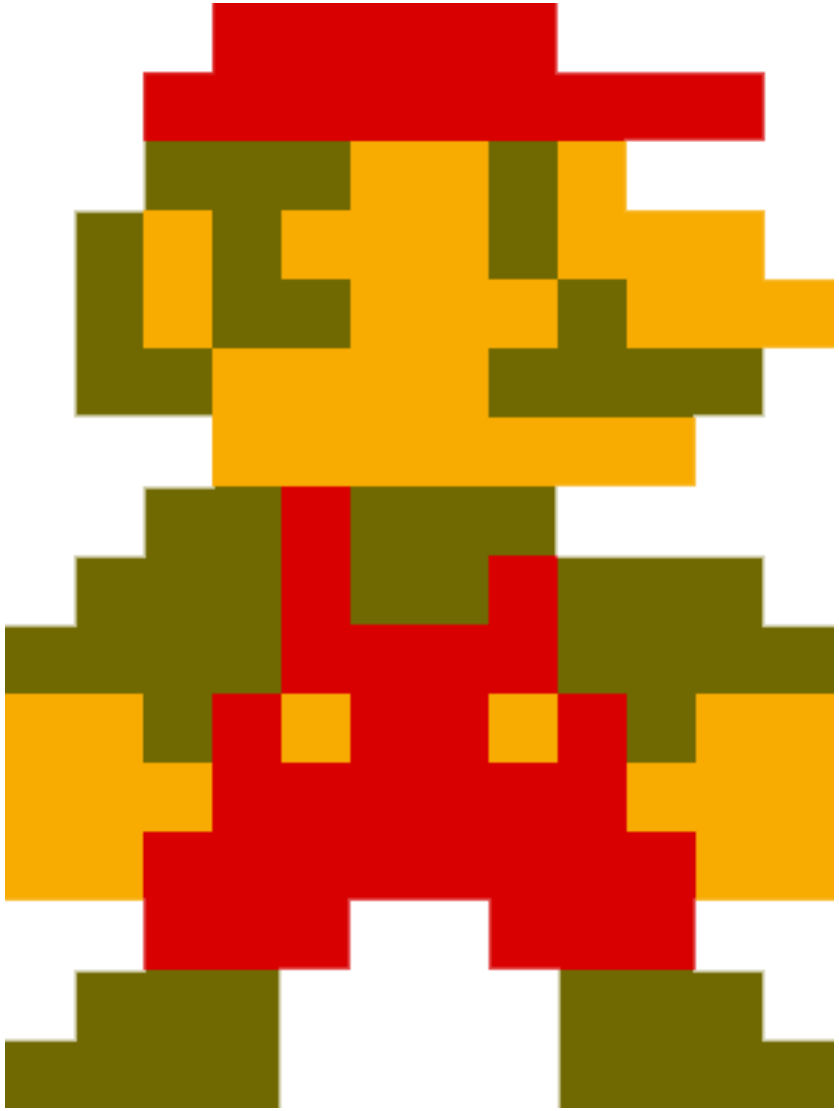
Let's see how social networks like Facebook can also be encoded.

8

What we see on our computer of a social network is nothing more than a set of texts and images.

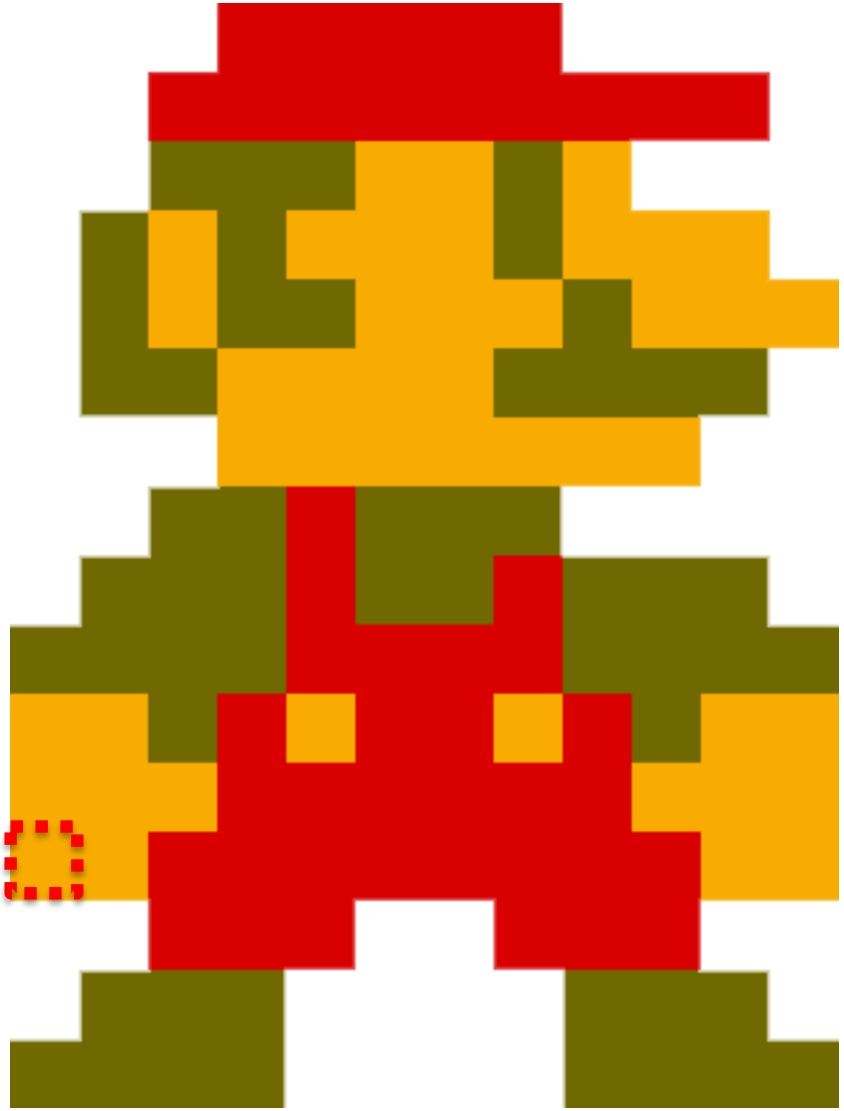
We've already dealt with texts, so now let's focus our attention on images.

Looking at a very detailed photo of sweets from Bergamo, it is difficult to think that there could be a correspondence with the natural numbers.



By looking at a low-resolution image of Mario, you can instead understand how a photo can be encoded.

pixel:
picture
element

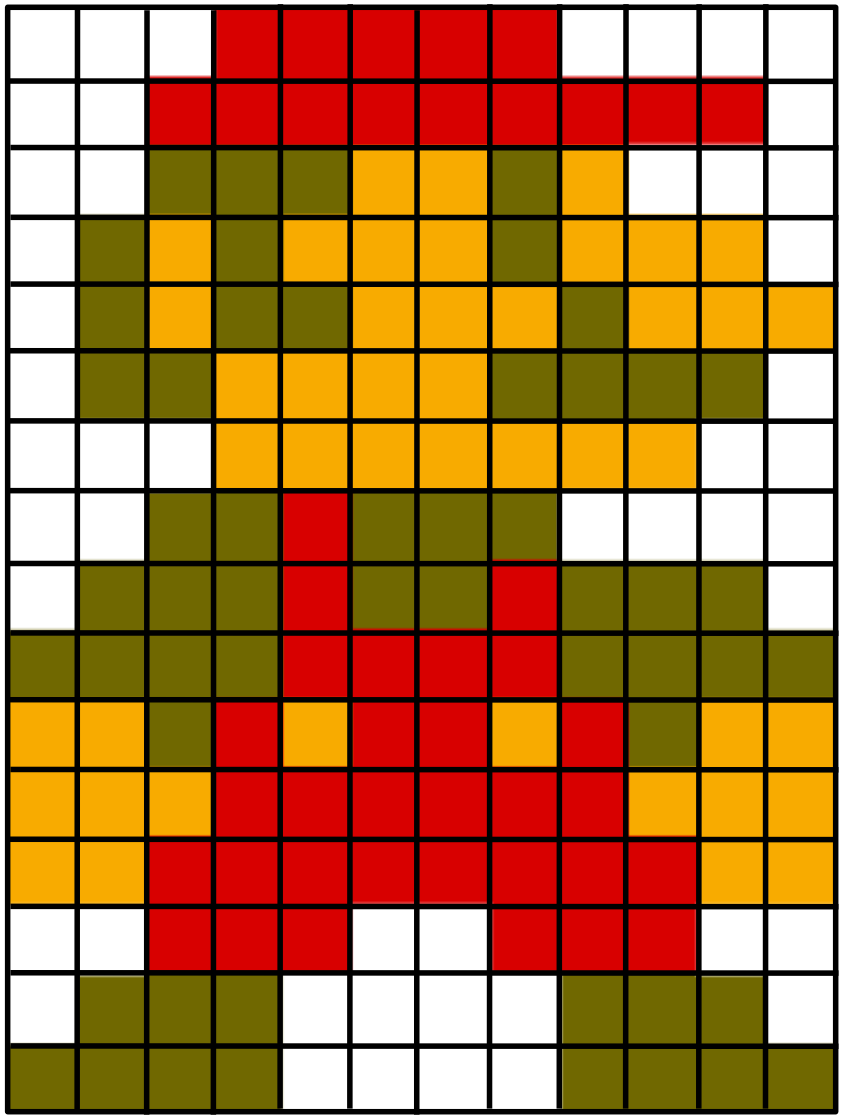


In the image of Mario, you can recognize basic square elements, characterized by the positions they occupy, and by their color: they are pixels.

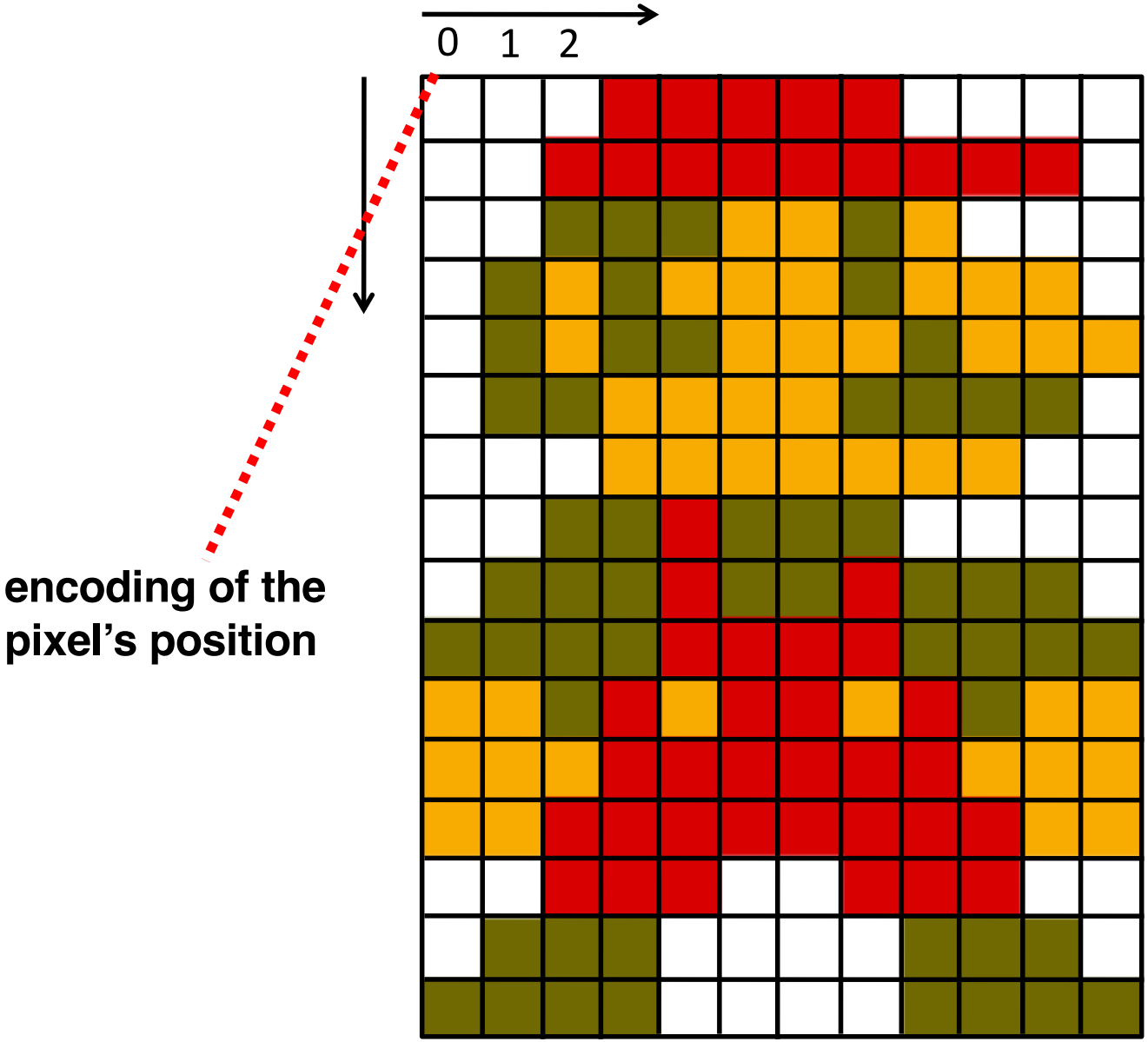
This photo is also made of pixels. We just don't see them because they are too small. In this image there are 12 million of them.

pixel (1/12000000)

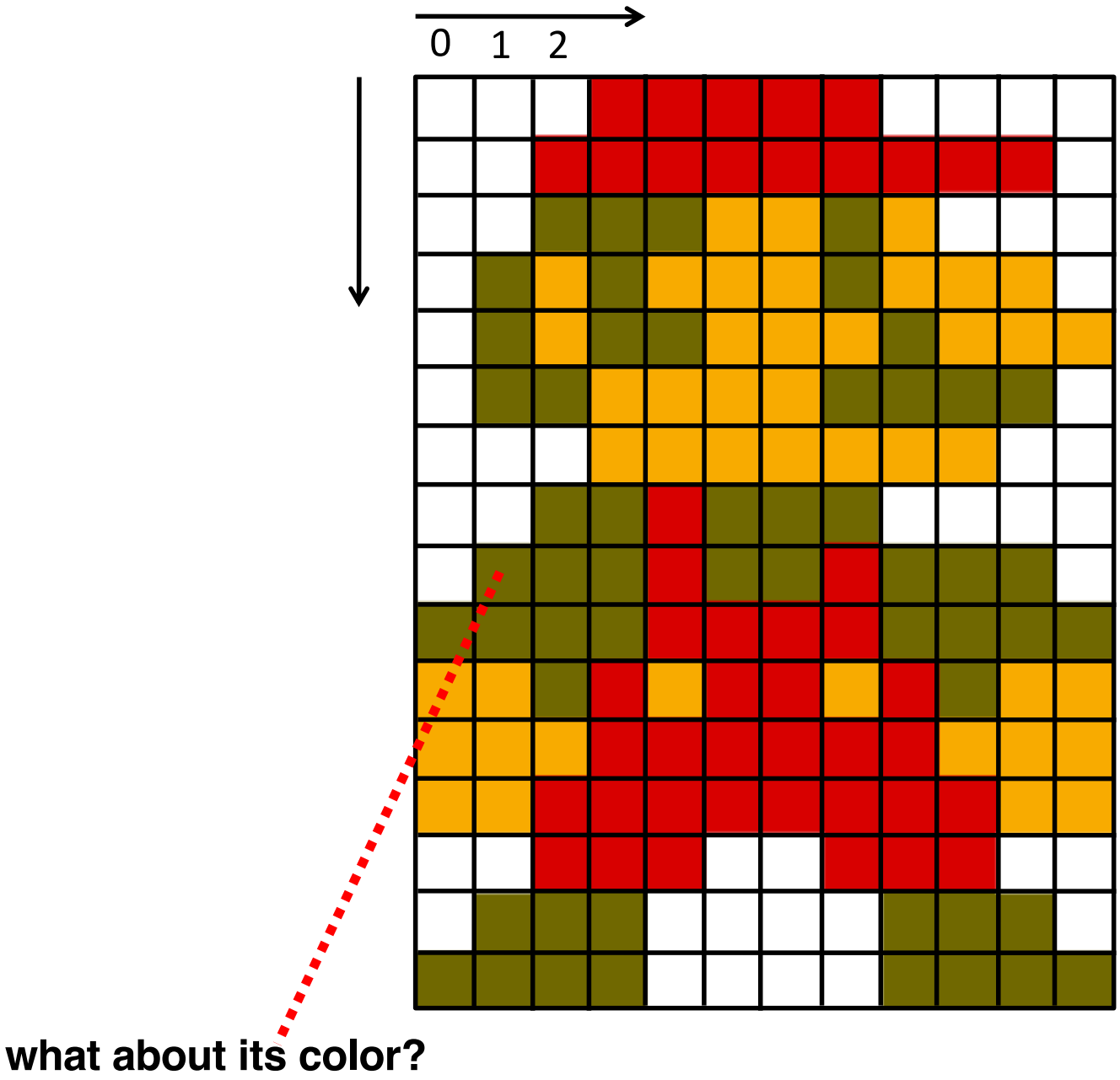
Image encoding is based on the pixels that compose it. In the same way as text is considered as a sequence of characters to be encoded…

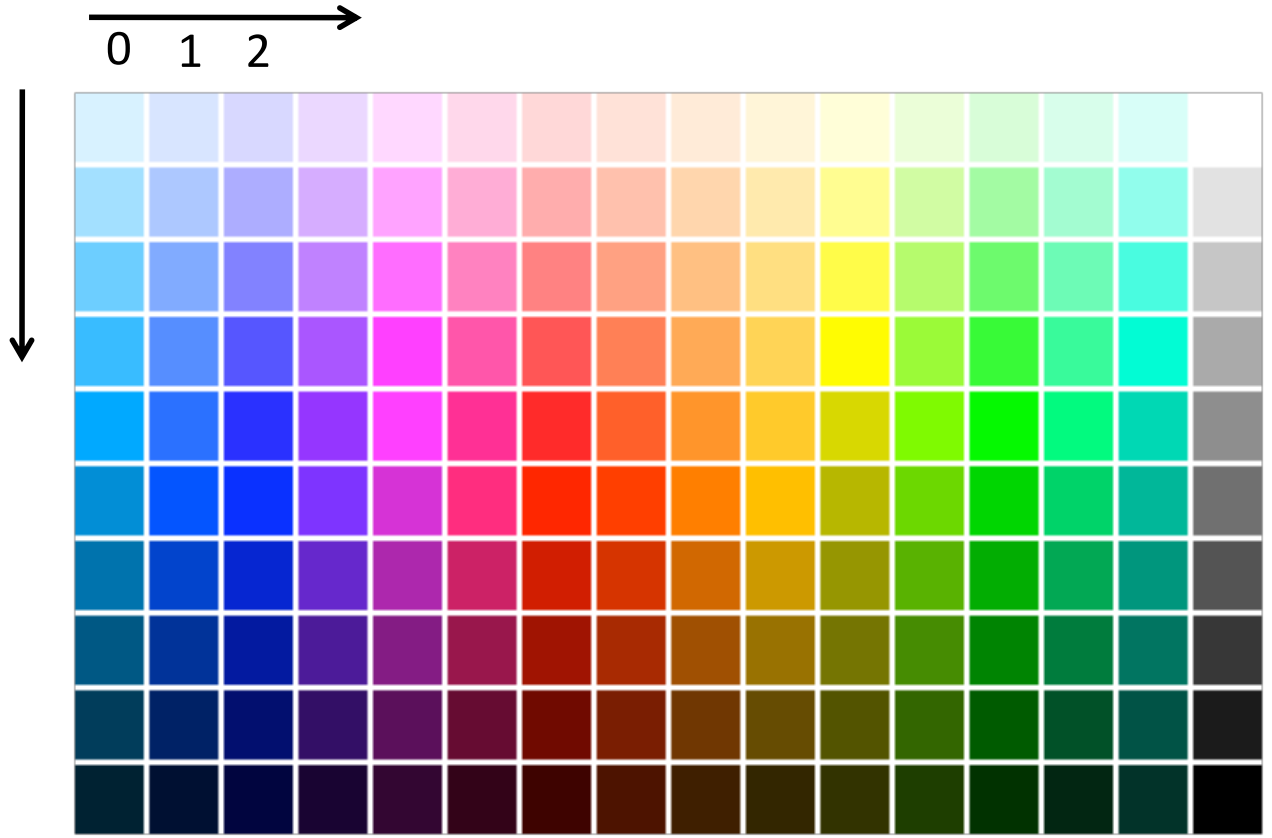


…so we consider a photo a sequence of pixels. If we can encode one pixel, we can encode an entire image.

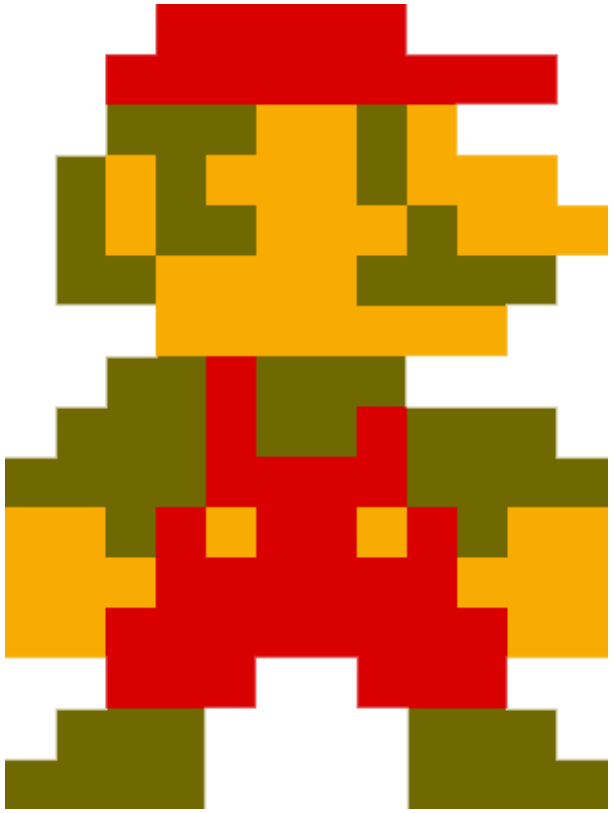


We said that the first characteristic of a pixel is its position. If we imagine inserting the image in a Cartesian plane, we are able to identify the position of the pixel via its numerical coordinates


0 1 2
what about its color?



If we imagine organizing colors in a table, in the same way as pixels we are able to associate a numerical value with each color, corresponding to the coordinates of the color in the table.

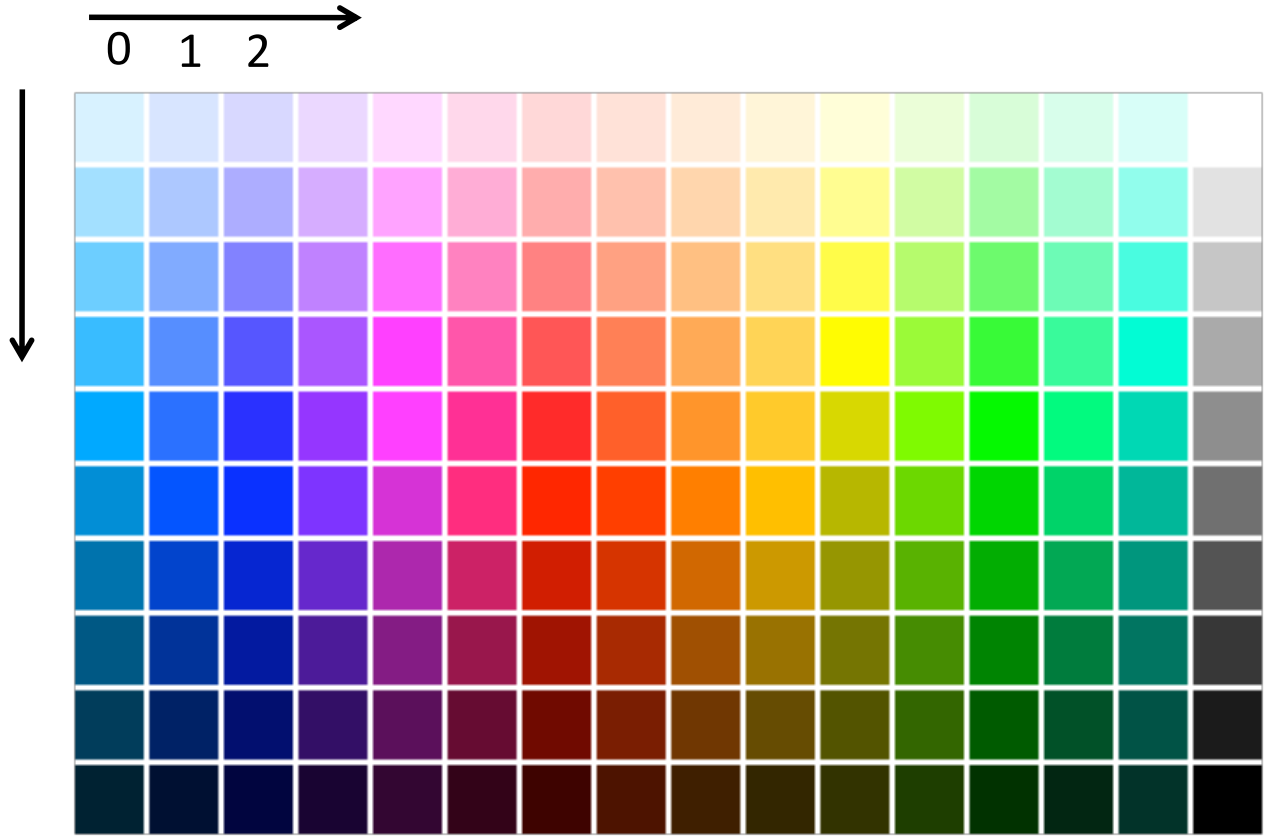


The pixels that make up Mario, therefore, can be coded based on their position and their color, allowing the image, a sequence of pixels, to be expressed in the form of a sequence of numbers.

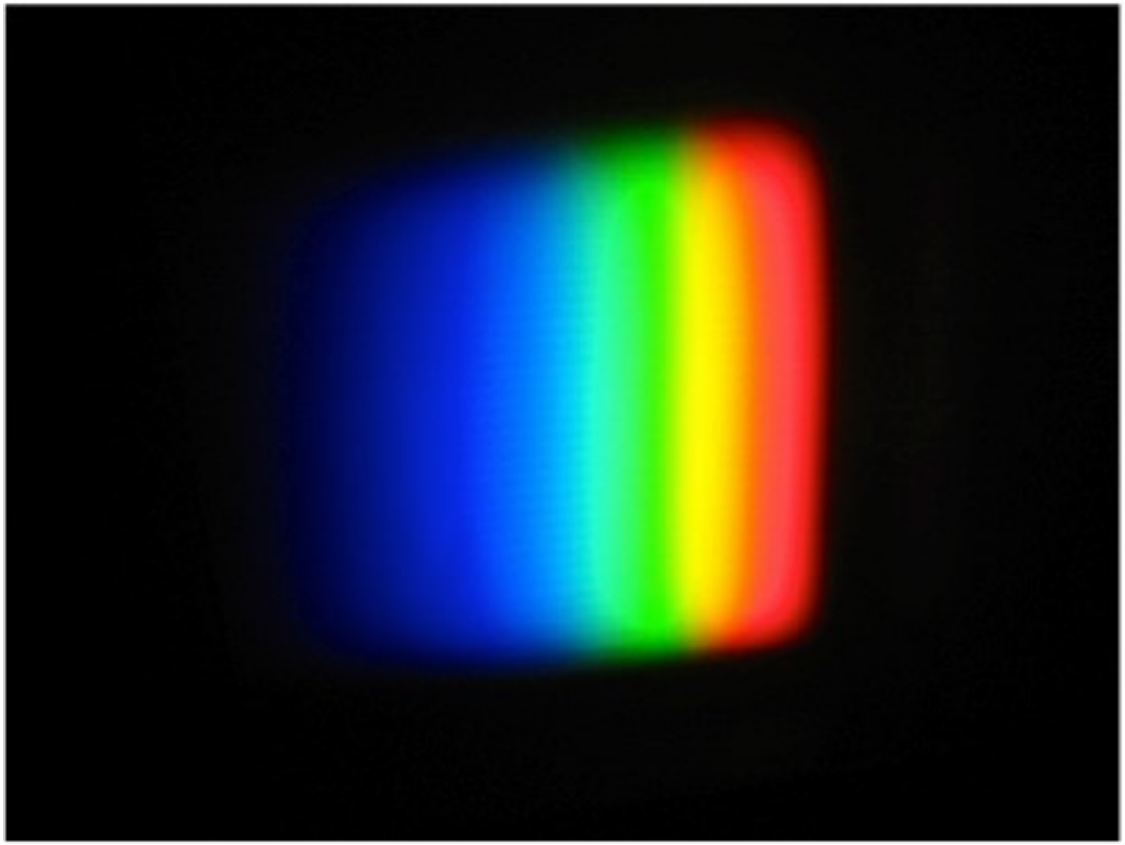
5678728729830939487489749874987484671011876278628767835463541376761291236213524136512438736149835023857467547790909394238472938472834782374923847238947283472983742836471651551556254654635463543764736584759847598247576715110198249284938493402839180917403754747834747364783647346378439401091348134094630745603847560384765018374650873146501387465078345603847560374865783465746587345061837456018374650183764567584038765734801873645710885734657831045871346531784511038746501837465087314650138746507834560384756037486578346574658734506183745601837465018376456758403876573480187364571088573465783104587134653178451103874650

We proceed in a similar way for the 12-million-pixel photo. It is no coincidence that more detailed photos take up more space in a computer memory: there are simply more encoded pixels.

1089273504563453476538450987530948570398456287650394175389457345893490583409584478262965673463746374633493489898101010294938483748734873948734817617647394873984739487384718937491837483748374837438147183974913847918347813471398471839473184781347183473874834783473847387487348734873847384738478347834738478347834791837498137483483138746507834560384756037486578346574658734506183745601837465018376456758403876573480187364571088573465783104587134653178451103874650



As you will have noticed, in the table that allows color coding, not all shades are present. Just take two adjacent squares and imagine taking the intermediate color: it is not clearly present in the table, so it is not an encoded color.



This means that only a small part of the colors in the spectrum can be encoded. Numerous nuances (infinite, to be precise) remain left out of the coding. On the other hand, it is unthinkable to have a table of infinite dimensions to accommodate the infinite shades of colors.

This is the problem of the conversion from physical phenomena to “digital” (i.e. made of digits) descriptions: the encoding of physical phenomena always involves a loss of information. Drawing a comparison with mathematics, physical phenomena are characterized by infinitesimal nuances similar to real numbers, while encodings are correspondences with natural numbers, which, despite being infinite, are still much smaller than real numbers and cannot act as references to the aforementioned nuances.

Physical [Analog]

vs.

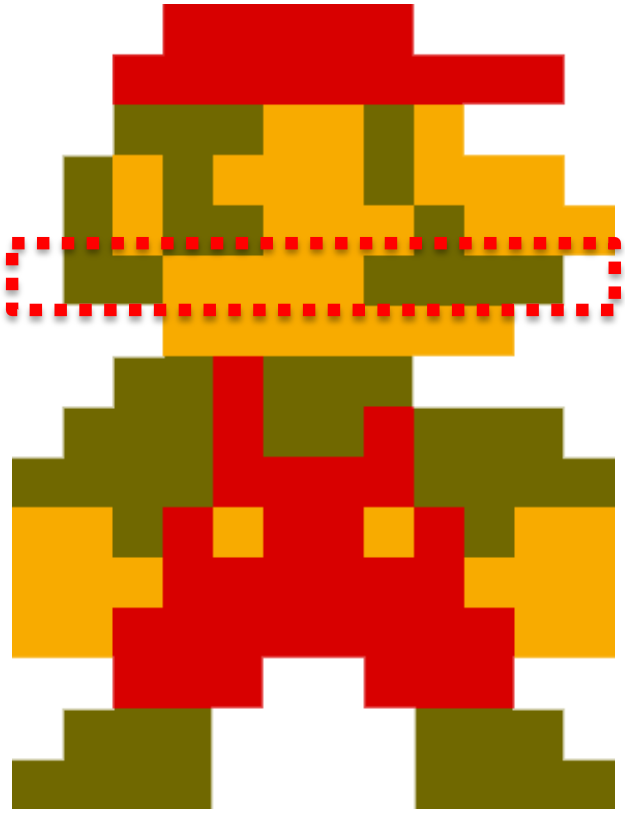
Digital

It is like when people say that looking at a photo can never be the same as seeing the real thing. In reality, three-dimensionality aside, current color encoding can capture millions of shades, many of which are indistinguishable to the human eye. Therefore, albeit with losses, encoding allows us to have more than good approximations of reality.

We mentioned before that the detailed photo of the sweets results in a much longer encoded sequence of numbers than the very simple drawing of Mario. The length of the encodings naturally depend on what is encoded: the greater the information to be encoded, the longer the result, and in fact the photo of the sweets (with all the details and nuances given by the 12 million pixels) certainly takes up more space in the computer than the Mario drawing (which, in fact, is a sequence of 16 x 12 pixels).

Compression

There are, however, different ways of describing the sequence of encoded pixels, and, if we choose wisely, we can obtain a shorter description of the same sequence, so that it takes up less space on the computer, and also saves time in transmitting it from one computer to another. We call “compression” a way to obtain sequences of symbols that describe a particular piece of information in a shorter way than the simple list of encodings of each element that composes it.

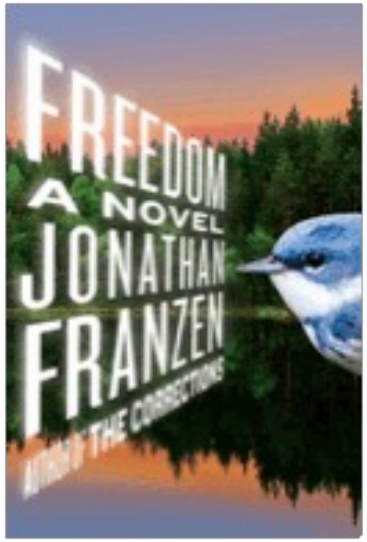


white pixel, green pixel, green pixel, yellow pixel, yellow pixel, yellow pixel, yellow pixel, green pixel, green pixel, green pixel, green pixel, white pixel

1 white pixel,
2 green pixels,
4 yellow pixels,
4 green pixels,
1 white pixel

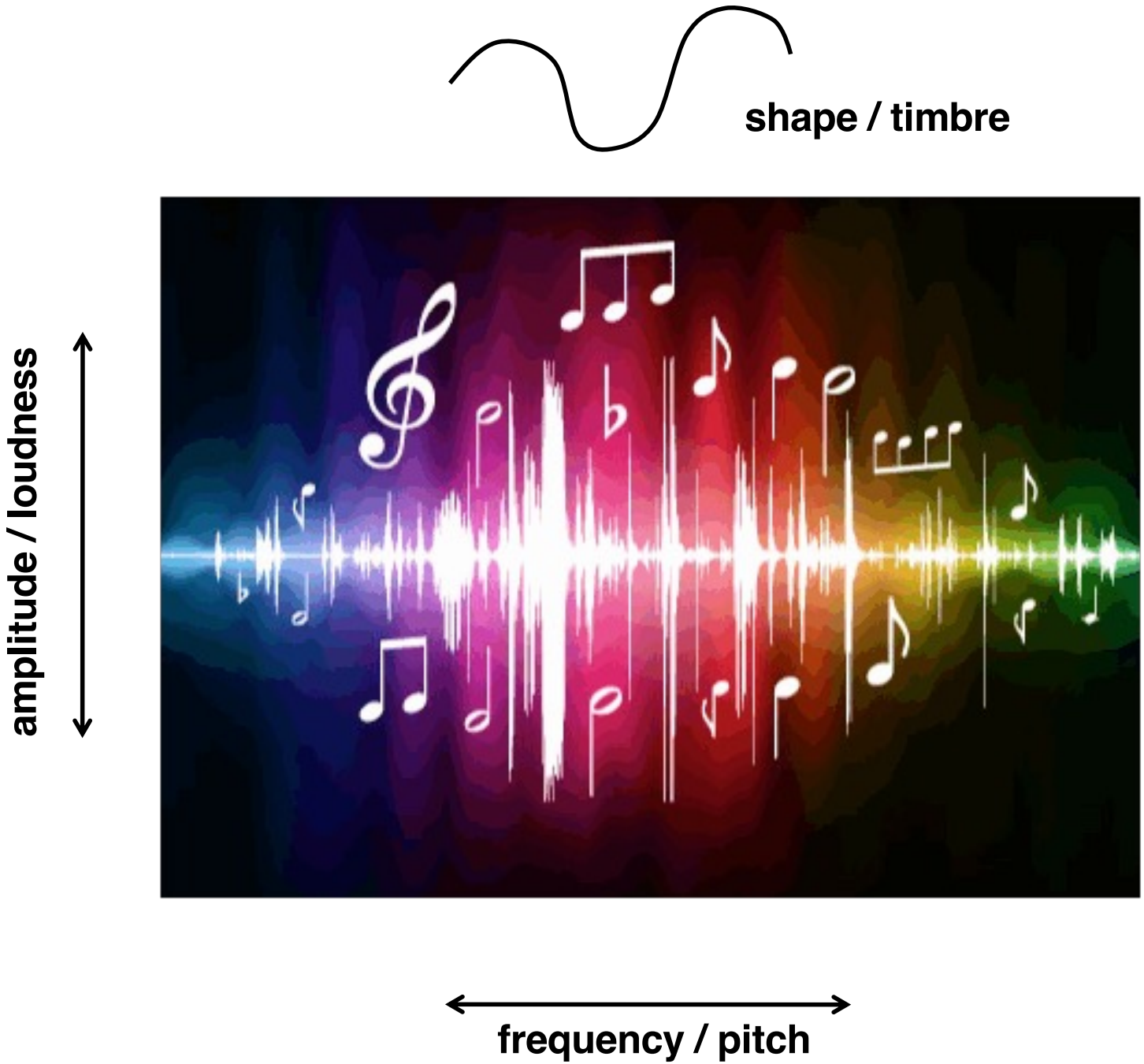
The description of the Mario image, for example, can be compressed by describing its pixels as shown in the second red box.

Any type of information that can be encoded can be compressed. The compression ratio, i.e. how much we can summarize the description, depends on the content of the information. (Imagine having to describe an all-white 16 x 12 rectangle. How would you do it?)



Let's now see how encoding allows you to have music on your computer.

8


shape / timbre
amplitude / loudness
frequency / pitch

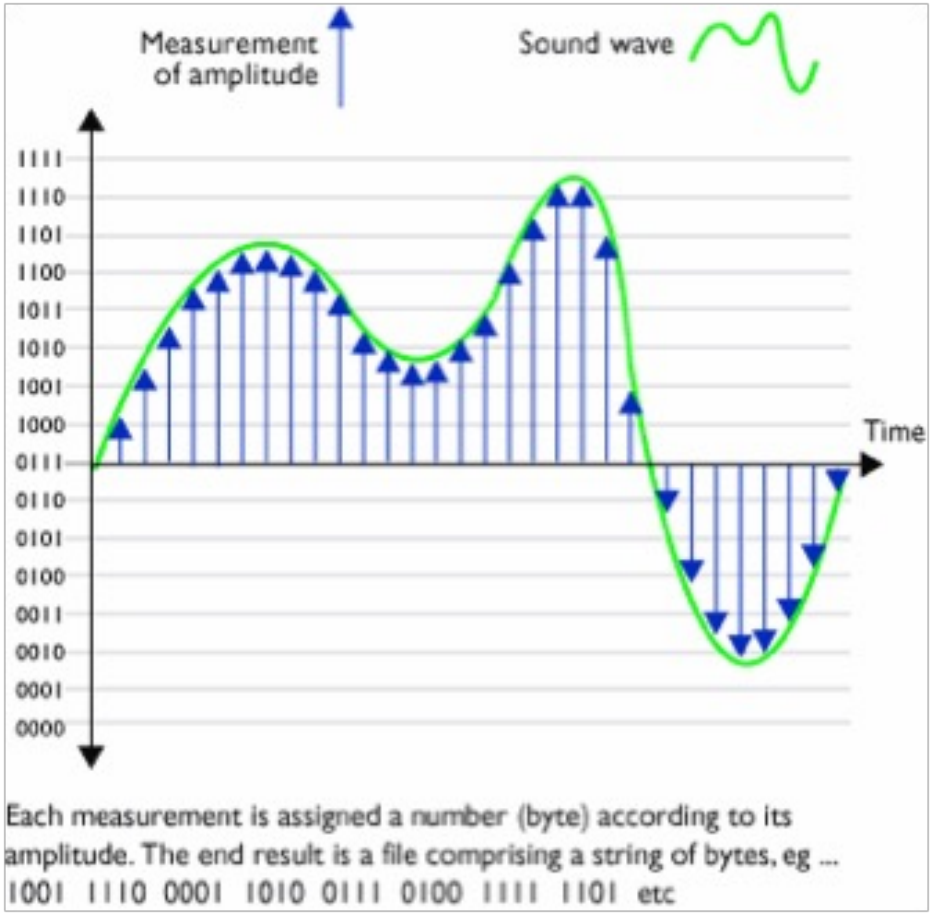
Sounds, and therefore also music, are due to vibrations of the air (or of any medium*) which propagate in the form of waves and which, hitting our eardrums, give rise to what we perceive with our ears. Sound waves are characterized by an amplitude (which determines the loudness of the sound: the wider the wave, the louder the sound), a frequency (which determines the pitch of the note: high notes correspond to high frequencies), and by a shape (which determines the timbre of the sound: my voice, your voice, piano, violin, cymbals, a doorbell, …)

*sounds do not propagate in a vacuum

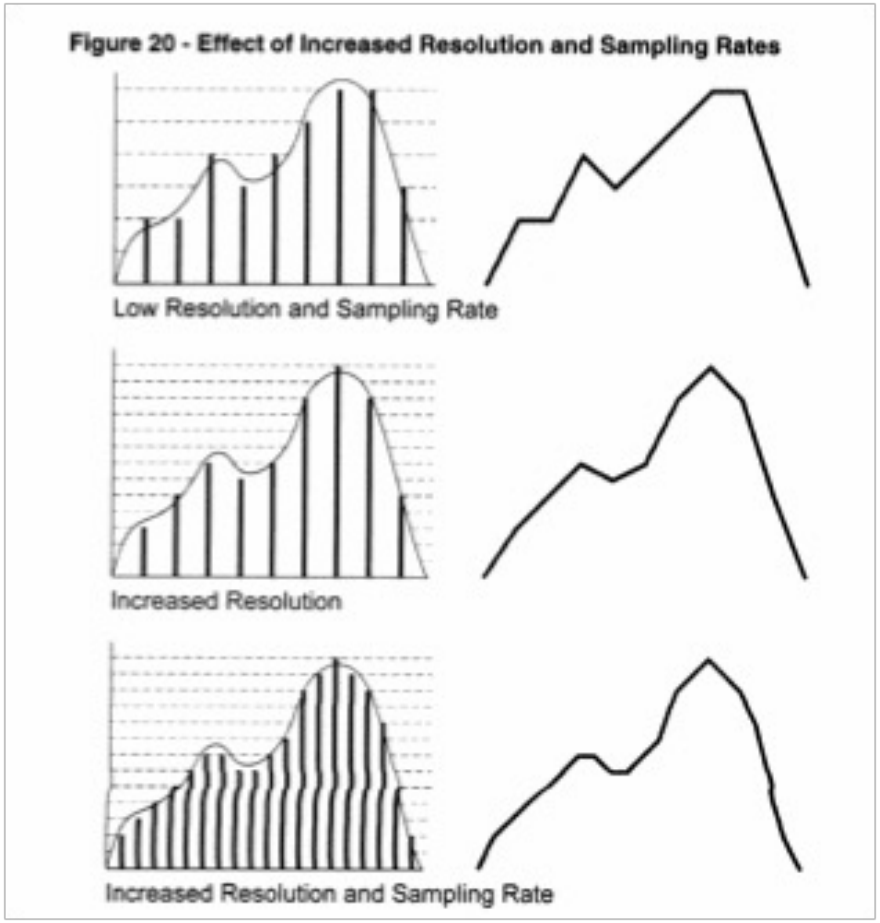
Sampling

The encoding of sounds is based on the encoding of the waves that produce them, which is in turn based on a procedure called sampling.

Sampling consists in considering the wave that constitutes the sound only in certain moments of time. Imagine describing the sound wave (with its amplitude, frequency, and shape) in a Cartesian plane, and considering only certain points of this curve.



These points correspond, in the Cartesian plane, to precise coordinates, whose numerical value is used as coding of the “sampled” sound. The encoded sequence of the samples constitutes the encoding of the entire sound data represented by the wave.

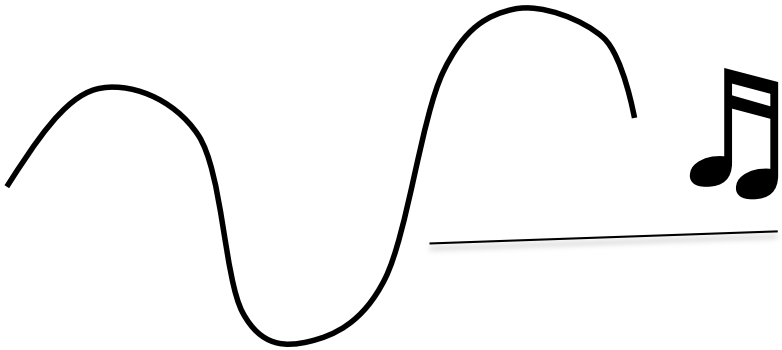


sampling frequency

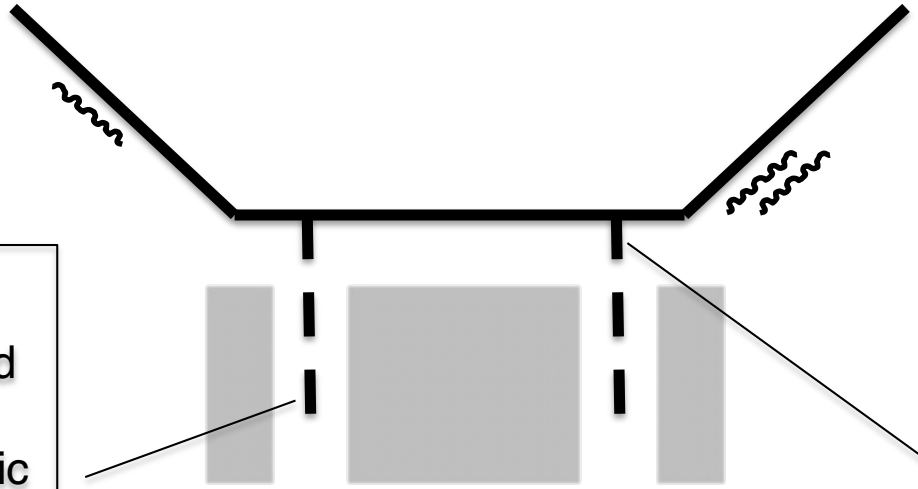
There is no universal criterion for establishing the distance between one sample and the next (or its inverse, known as “sampling frequency”). It is easy to imagine that samples that are closer to each other (higher sampling frequency) correspond to a longer encoding, and also a reconstruction that is more faithful to the original wave. Even in the case of sound and music encodings, there are compression methods to synthesize their description. The famous MP3 files are called this way because they take their name from a specific sound encoding technique with compression. Even in this context there are people who say that the sound of an MP3 on a computer will never equal the quality of a live concert: it’s all a question of approximations.

Encoding a sound is used to make it processable by a computer (for example, to allow the transfer of a song from the iTunes online store to our computer). However, even in the digital age, we continue to listen with our ears, and to make our listening possible, we need sound waves that propagate in the air. A reconversion from numerical coding to sound waves is therefore necessary: that is, we need speakers which, controlled by the electrical signals produced by the computer according to the numbers contained in the coding of the song, make membranes vibrate which produce waves that we perceive as sounds and music.

again, physical vs digital

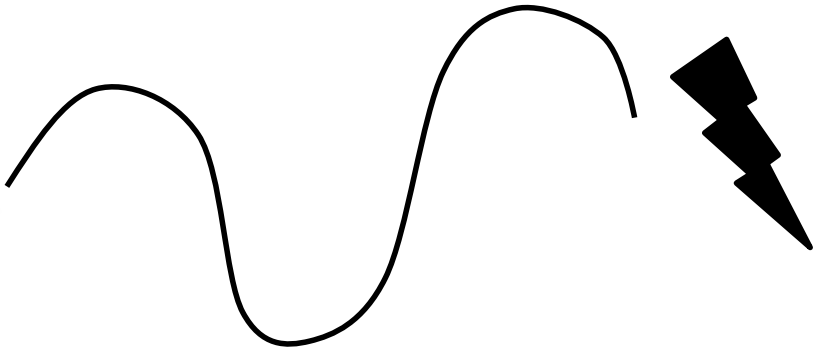


4. The vibrations of the membrane reproduce sounds and music

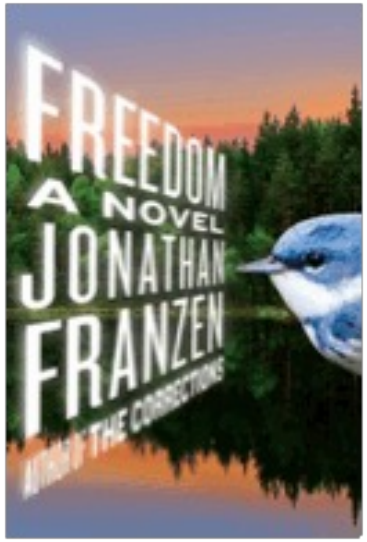


2. The signal goes through the solenoid and its changes determine a magnetic field in it.

3. The solenoid's field interacts with the magnet's and the solenoid moves, hitting the membrane



1. Electric signal built out of the numerical data in the computer



Finally, let's see how to proceed with video encoding.

8

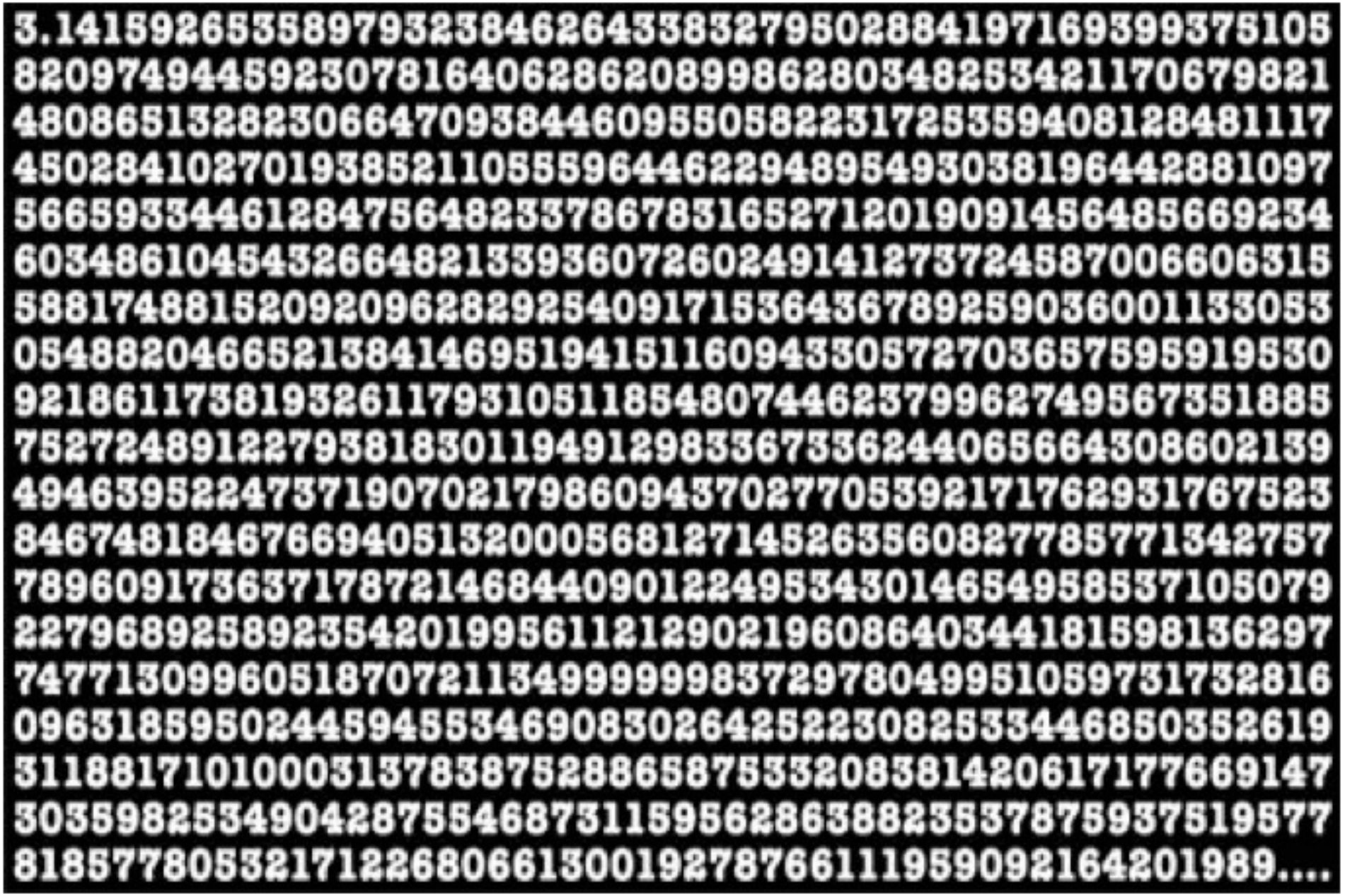
Actually, having image and sound encoding techniques available, it is easy to imagine that they can be combined to create video encoding.

=

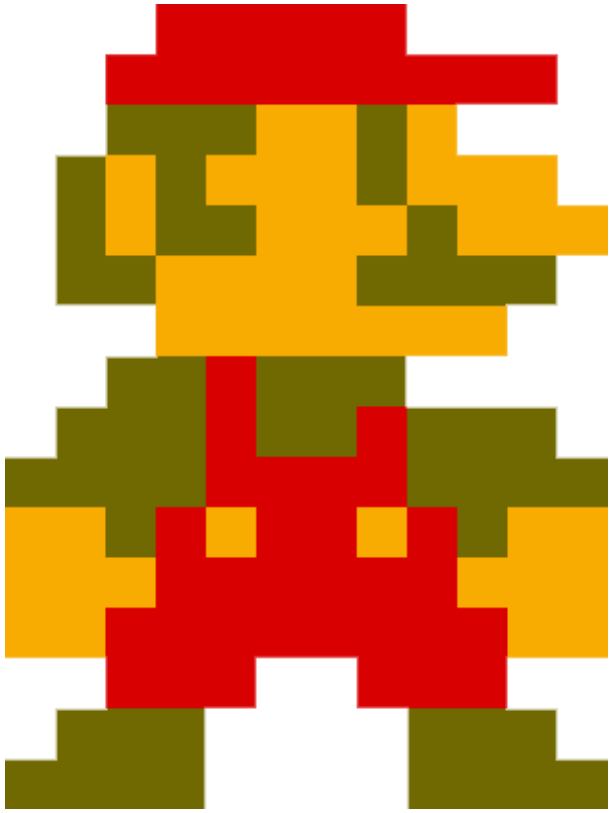
Additional techniques are needed to take into account the synchronization between images and sounds, and compression techniques based on the idea of not describing all the pixels of each image, but only describing the initial one and then focusing on the differences between an image and the next (better compression with very similar frames, worse compression with scene changes).

+

+ ...


3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196442881097566593344612847564823378678316527120190914564856692346034861045432664821339360726024914127372458700660631558817488152092096282925409171536436789259036001133053054882046652138414695194151160943305727036575959195309218611738193261179310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217986094370277053921717629317675238467481846766940513200056812714526356082778577134275778960917363717872146844090122495343014654958537105079227968925892354201995611212902196086403441815981362977477130996051870721134999999837297804995105973173281609631859502445945534690830264252230825334468503526193118817101000313783875288658753320838142061717766914730359825349042875546873115956286388235378759375195778185778053217122680661300192787661195909216420198 9....

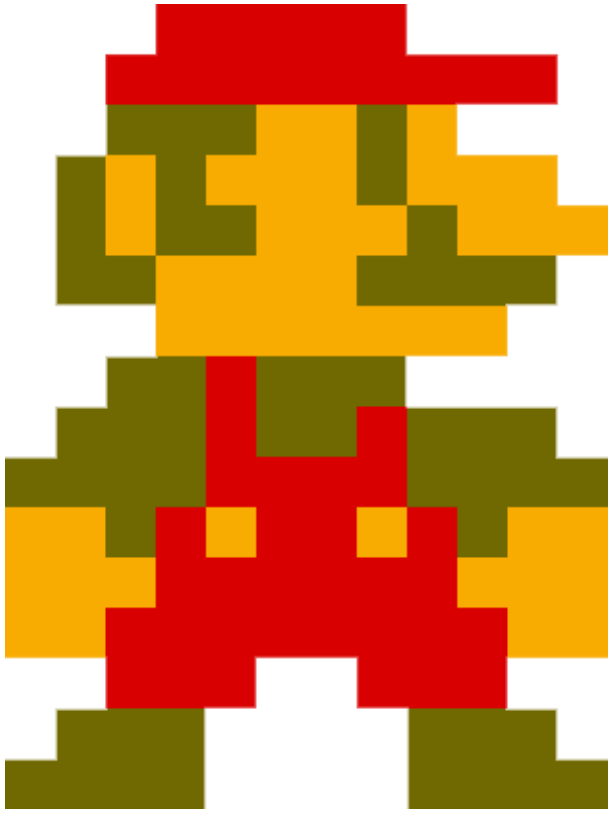
As already mentioned, the approximation that occurs when passing from physical phenomena such as colors and sound waves to numerical encoding suitable for a computer is reminiscent of the relationship between real numbers and natural numbers in mathematics. Perhaps the use of natural numbers to talk about the concept of encoding may seem limiting and may seem to invite the involvement of rational numbers, but the underlying argument does not change: even with rational numbers, the encodings are unable to faithfully reproduce a physical phenomenon, but they bring with them inevitable approximations. The number in the previous slide, for example, represents only a part of the decimal expansion of the number π. We can approximate it as we like, but part of the description of π will still be left out and cannot be included, since no calculator is capable of containing an infinite sequence of digits. The important thing is that the approximation reaches such detail that the reconstruction of the physical phenomenon is indistinguishable from the original to the eyes (or ears) of the human being who uses the computer to process it.



Let's consider again the concept of encoding with the example of the image of Mario. To be processed by a computer, it must be transformed into a sequence of digits.

5678728729830939487489749749874846710118762786287678354635413767612912362135241365124387361498350238574675477909093942384729384728347823749238472389472834729837428364716515515562546546354635437647365847598475982475767151101982492849349340283918091740375474783474736478364734637843940109134813409463074560384756038476501837465087314650138746507834560384756037486578346574658734506183745601837465018376456758403876573480187364571088573465783104587134653178451103874650

Actually, these digits must themselves be encoded to be processed by a computer.
This further encoding transforms them into a sequence of 0s and 1s.



01010101010101000100111010101010101010101
01110101000010101010101011101010101011010
10101001001111111101010101010101001010101
01010101011111100101000110101010001110101
01010101111010110101000101010011101001101
01010011100010101110101010001010101101010
10001101101010111010100101000101010100010
10111010101010111001010101010001010101010
10101010111010101010001010101011101010101
01011010101111101011100000111000111001101
10011010101110011100011100010101010111100
00001110000101010101011001010101111100000

What is a bit?

It is the basic data unit processed by a computer: it is 0 or 1.

bit = binary + digit.

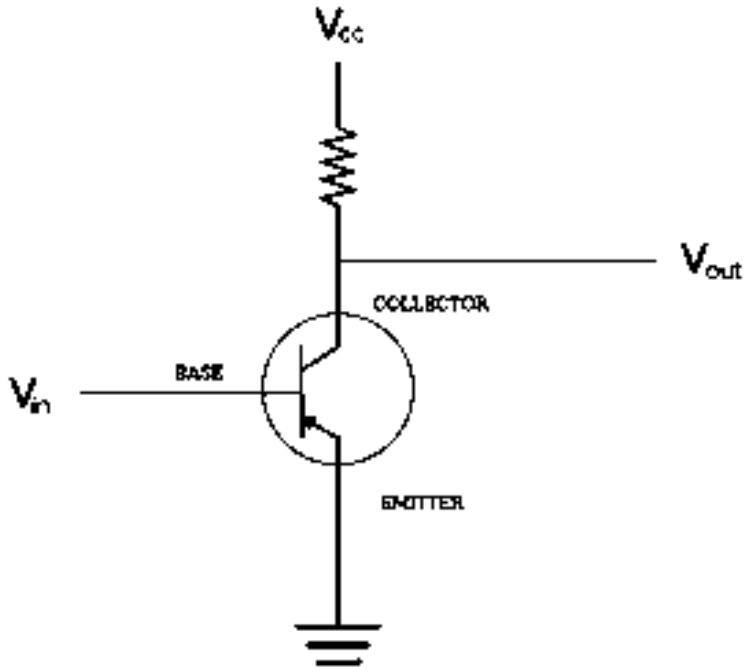
8 bit (b) = 1 Byte (B).

The encoding that uses only 0 and 1 is called binary encoding (because it only uses 2 digits).

Why 0 and 1?

THE MOST FUNDAMENTAL ENCODING OF THEM ALL

Why 0 and 1?



High electric voltage: 1
Low electric voltage: 0

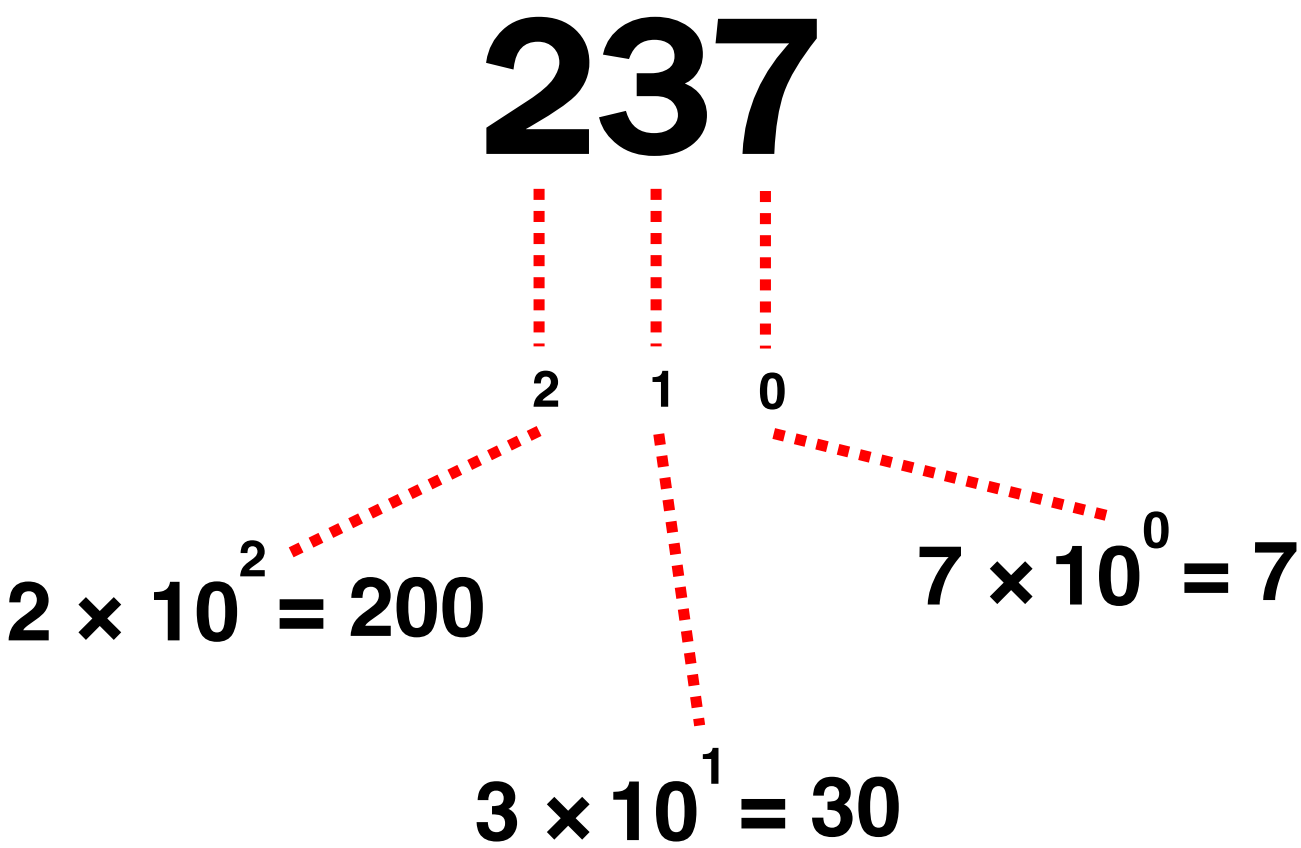
Gottfried Wilhelm Leibniz
(1646-1716)

THE BINARY SYSTEM

{0;1}

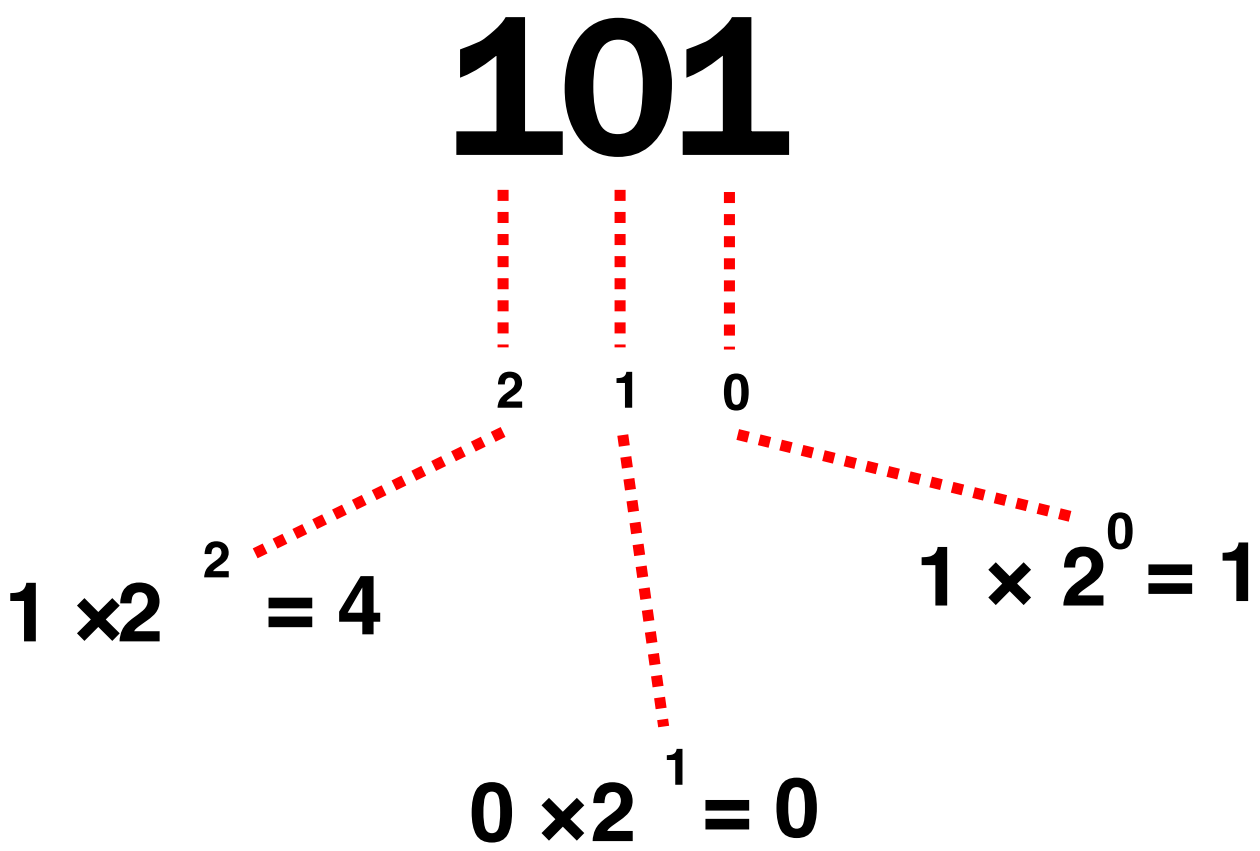
THE DECIMAL SYSTEM {0;1;2;3;4;5;6;7;8;9}

237


237
2
1
0
$2 \times 10^2 = 200$
$3 \times 10^1 = 30$
$7 \times 10^0 = 7$

THE BINARY SYSTEM

{0;1}

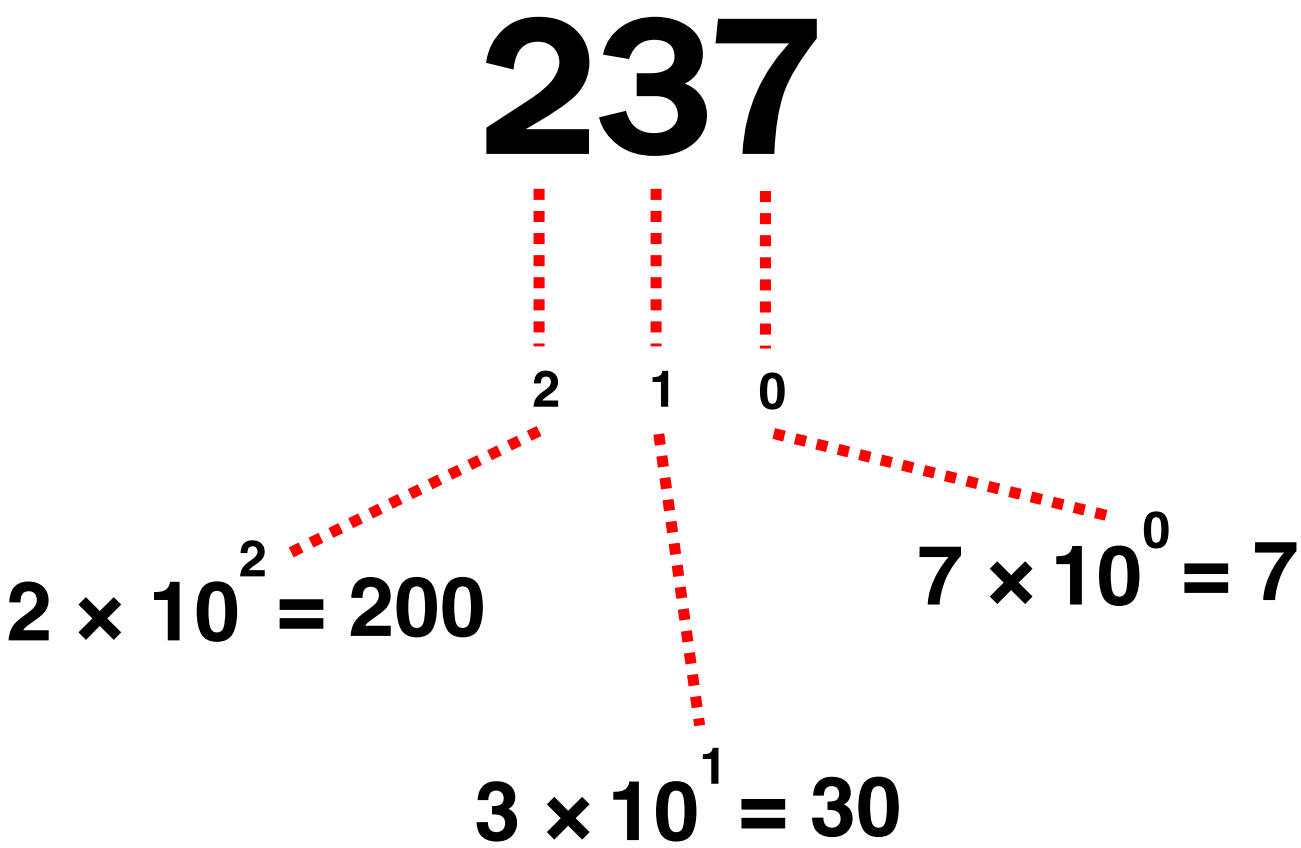

101
2
1
0
1 ×2 2 = 4
0 ×2 1 = 0
1 × 2 0 = 1

101
in binary
encodes
the number 5

$$101_2$$

$$=$$

$$5_{10}$$


237
2
1
0
$2 \times 10^2 = 200$
$3 \times 10^1 = 30$
$7 \times 10^0 = 7$

What is the binary encoding of 237?

$$237 =$$

$$128 + 64 + 32$$

$$+ 8 + 4 + 1$$

(look for powers of 2)

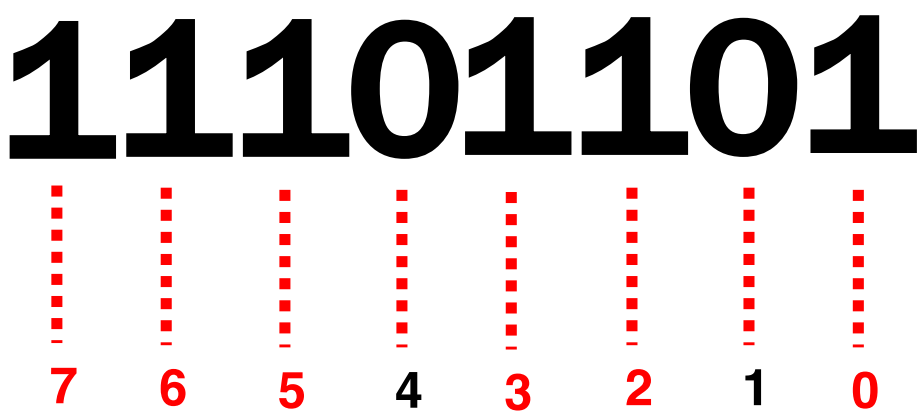
$$237 =$$
$$2^7 + 2^6 + 2^5$$
$$+ 2^3 + 2^2 + 2^0$$

$$237 =$$

$$2^7 + 2^6 + 2^5$$

$$+ 2^3 + 2^2 + 2^0$$

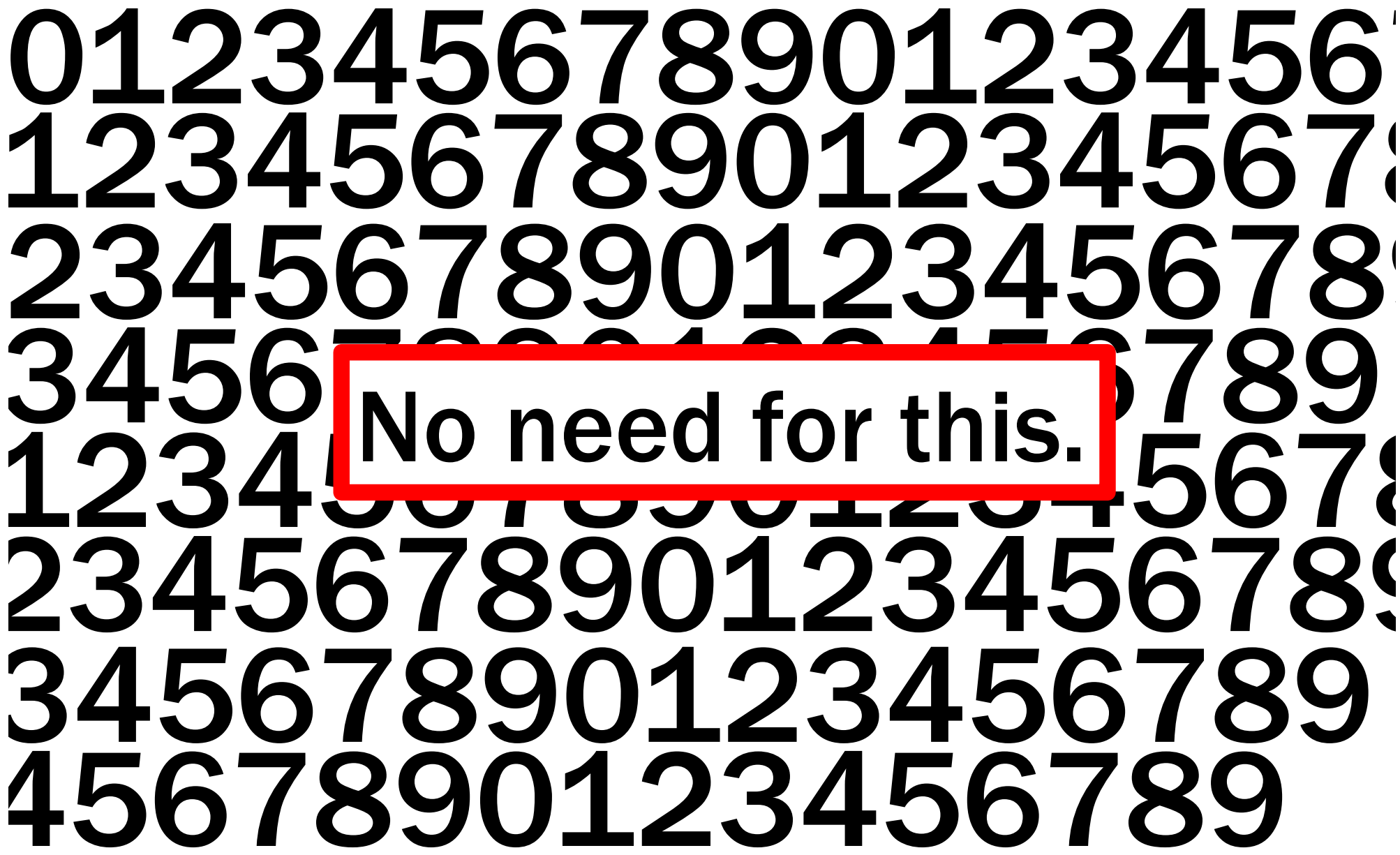
(put a 1 in the positions indicated by the exponents, 0 in the other positions)

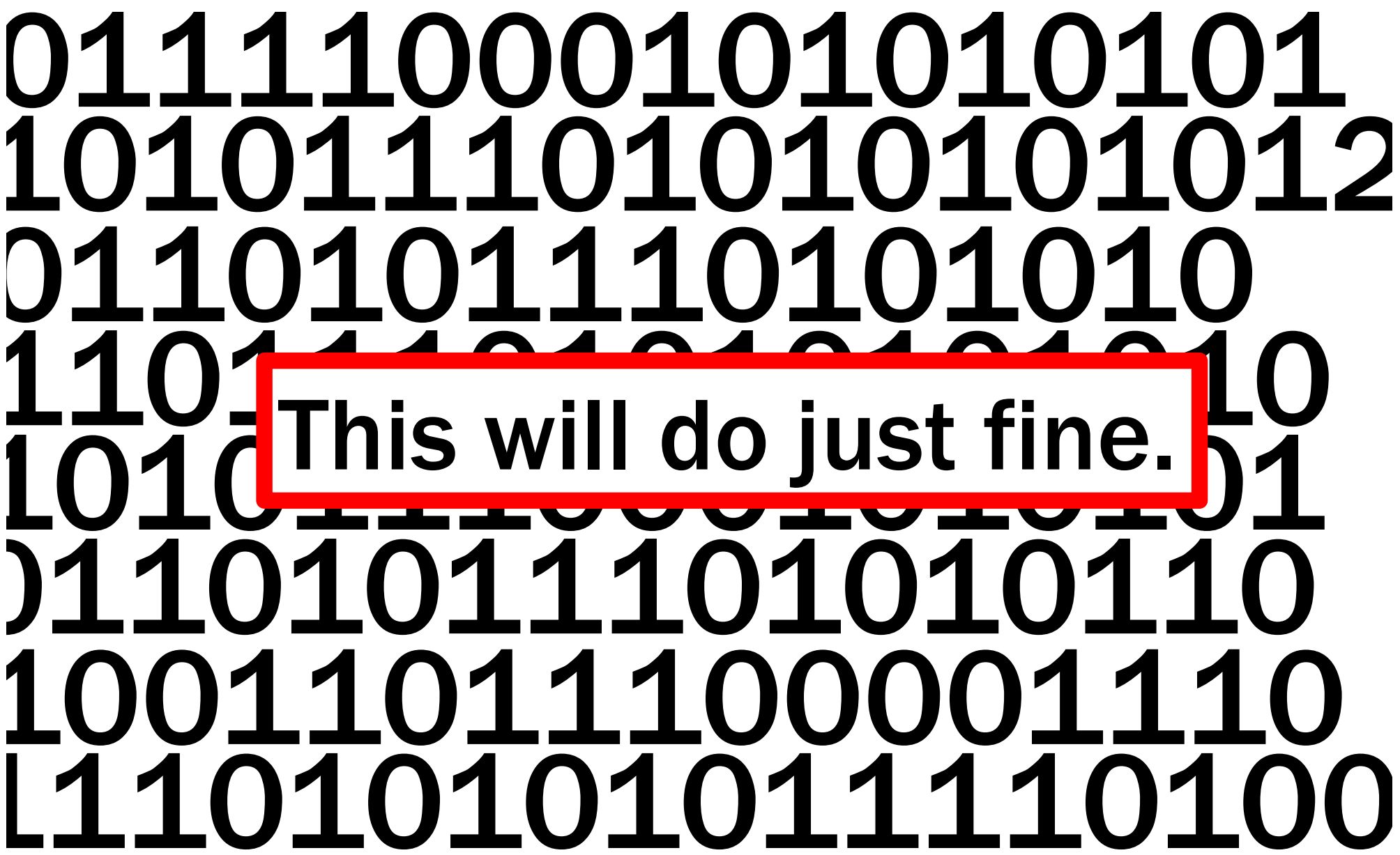

11101101
7 6 5 4 3 2 1 0

$$11101101_2$$

$$=$$

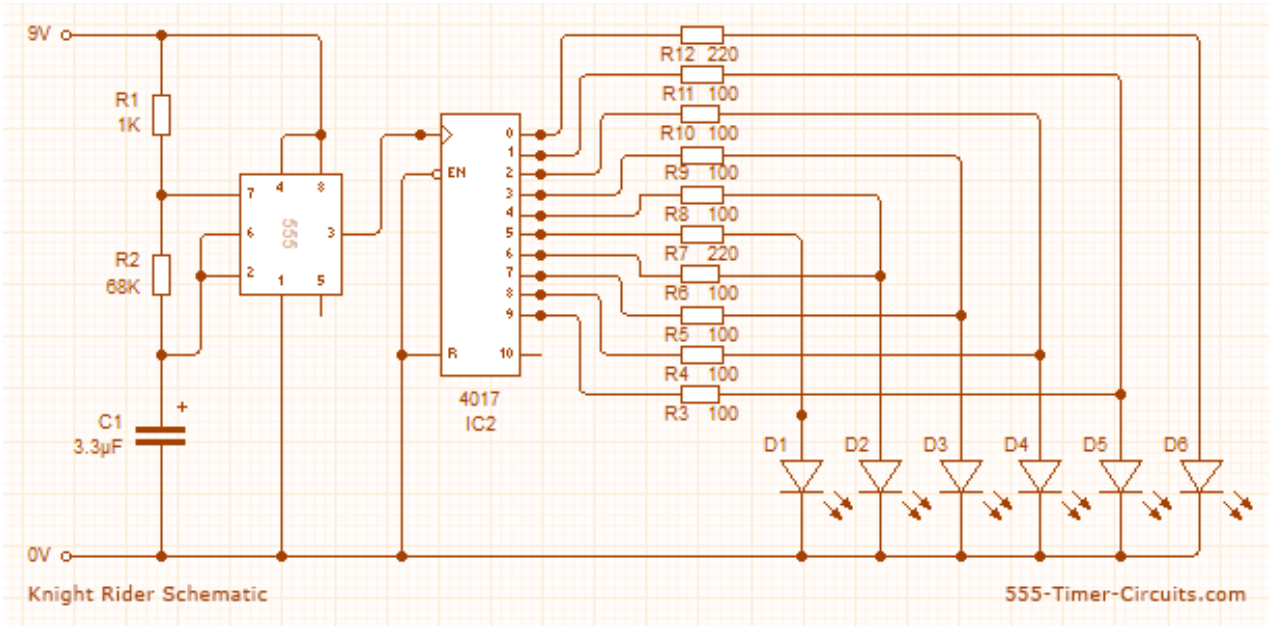
$$237_{10}$$

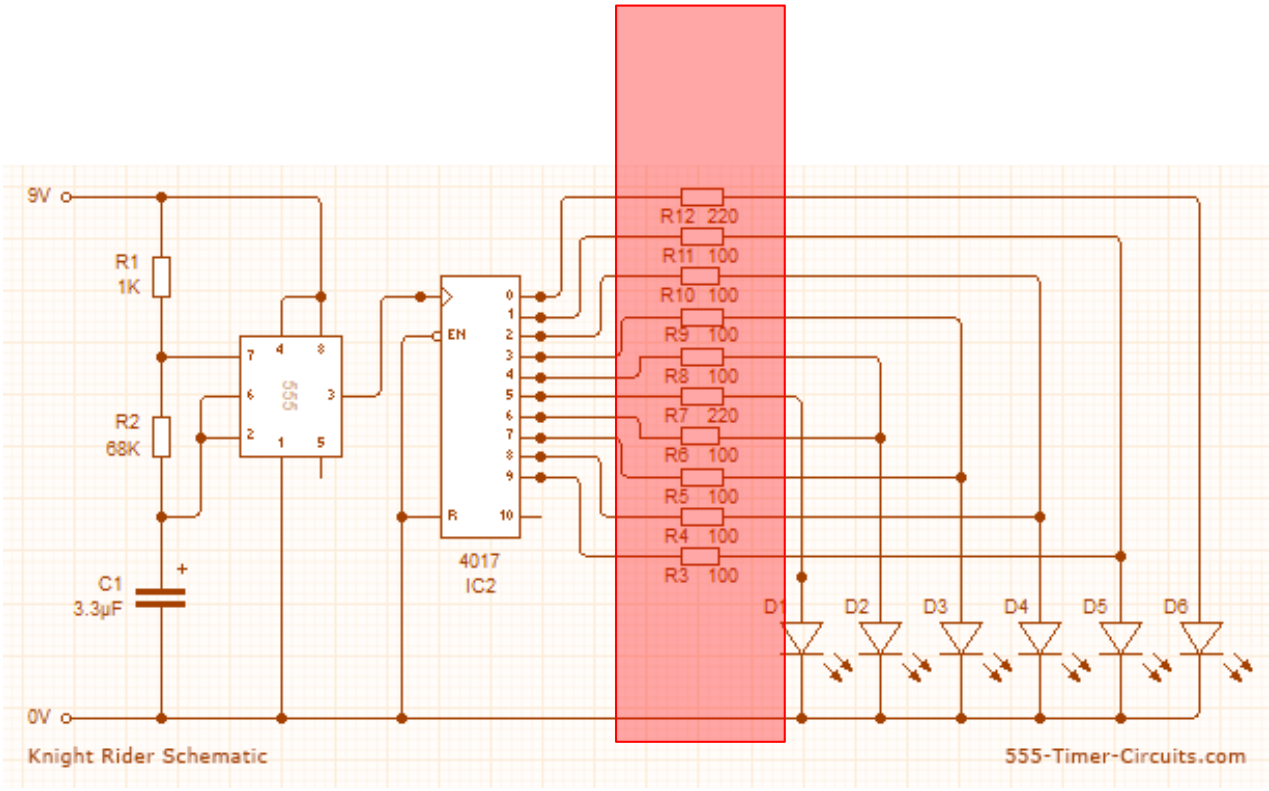

No need for this.

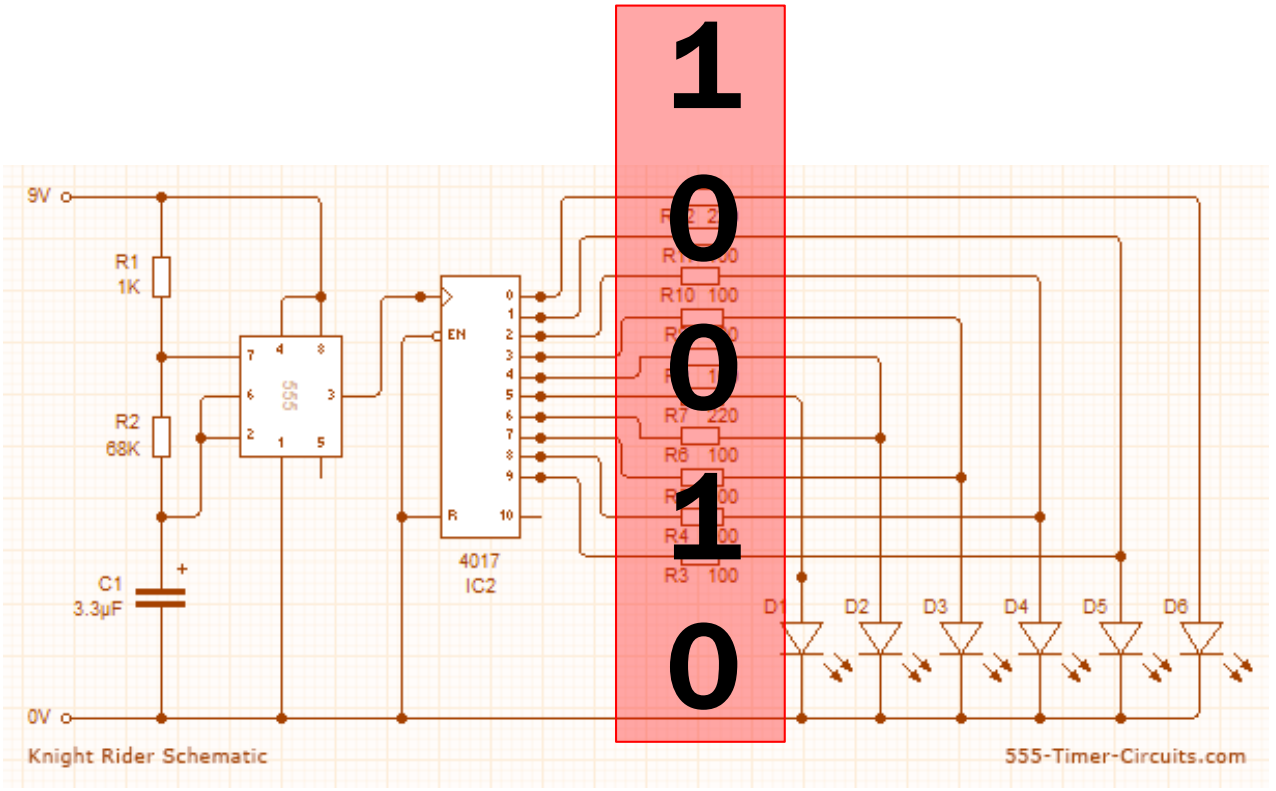

This will do just fine.

The electronic circuits that make up a computer are built to respond to high or low voltage electrical signals. High voltage is interpreted as a "1", while low voltage is interpreted as a "0". The restriction to only two values has no physical reason: it is possible to build circuits that respond with numerous voltage levels. For example, we could have 10 different ones, interpretable as the 10 digits (from "0" to "9") that are usually used in mathematics.

However, the advantages of having only two signals are numerous:
the circuits are simpler to make and cost less; furthermore, the output signals, even in the presence of disturbances due to natural causes, are easier to interpret, with less possibility of error. (e.g. 0.8 ≈ 1)


9V
R1
1K
R2
68K
C1
3.3µF
0V
555
EN
R
4017
IC2
R12 220
R11 100
R10 100
R9 100
R8 100
R7 220
R6 100
R5 100
R4 100
R3 100
D1
D2
D3
D4
D5
D6
Knight Rider Schematic
555-Timer-Circuits.com


9V
R1
1K
R2
68K
C1
3.3µF
0V
555
EN
R
4017
IC2
R12 220
R11 100
R10 100
R9 100
R8 100
R7 220
R6 100
R5 100
R4 100
R3 100
D1
D2
D3
D4
D5
D6
Knight Rider Schematic
555-Timer-Circuits.com


1
0
0
1
0
9V
R1
1K
R2
68K
C1
3.3µF
0V
555
EN
R
4017
IC2
R10 100
R7 220
R6 100
R3 100
D1
D2
D3
D4
D5
D6
Knight Rider Schematic
555-Timer-Circuits.com

ARE THE DIGITS REALLY THERE?

EYE OF THE BEHOLDER.

Binary encoding of numbers

Numbers with base 10:

$$215 = 2\times10^2 + 1\times10^1 + 5\times10^0$$

Numbers with base 2:

$$110010111 = 1\times2^8 + 1\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0$$

Exercise

Convert the following numbers from base 2 to base 10: 101, 1000, 11011.

Exercise

Convert the following numbers from base 10 to base 2: 8, 23, 144, 201.

While the definition of a binary system helps us solve the first exercise, we must find new methods to solve the second. There are two.

First method:
given the number n in base 10, we look for the largest power of 2 that is less than or equal to n.
If it is the same, we have solved the problem: we write a 1 in the position corresponding to that power of 2, followed by zeros.
For example, 8 is a power of 2: 2^3 to be precise, so its binary encoding will be 1000.
However, if the largest power of 2 that is less than or equal to n is less than n (let's call it k), let's set it aside and we calculate the difference n-k.
We repeat the same procedure with n-k, and look for the largest power of 2 that is less than or equal to it.
We continue until we are able to express n as a sum of powers of 2.
We take the list of powers and write a 1 in the corresponding positions, 0 in the others.
For example, the largest power of 2 contained in 23 is 16 (2^4). Their difference is 7, in which 4 (2^2) is contained. The difference is 3 where there is 2 (2^1), after which only 1 (2^0) remains.
Writing the powers of 2 present in order we get 10111.

Second method:
we divide the number by 2 and obtain quotient and remainder.
As long as the quotient is not 0, we take it as the new dividend and continue dividing. When we get zero quotient, we have to write the list of remainders in reverse order to get the binary encoding of the initial number.
Let's take 144 as an example:

144 : 2 = 72 with remainder 0
72 : 2 = 36 with remainder 0
36 : 2 = 18 with remainder 0
18 : 2 = 9 with remainder 0
9 : 2 = 4 with remainder 1
4 : 2 = 2 with remainder 0
2 : 2 = 1 with remainder 0
1 : 2 = 0 with remainder 1

144 in base 2 is 10010000 (these bits are the remainders in the reverse order of writing)