- 4. 3. CPU (Central Processing Unit) -

- INDEX -

- 4. 3. 1. The Microprocessor and the Motherboard -

The microprocessor

The main component of the most important part of a computer is the processor, or so called microprocessor, since it has been miniaturized so much that it can be no longer be seen without a microscope.

The processor can be considered the brain of the computer. It performs all the operations, executes all the instructions and makes all the calculations and decisions necessary to solve problems.

Therefore, the microprocessor is the true computer. All the rest is just addons that allow it to communicate with the user.

If the microprocessor is the brain, the motherboard is the spinal cord. The motherboard establishes a link between the microprocessor and other parts such as the memory and peripheral devices. It acts like a fast data highway that transports the information from one component to another.

The Microprocessor is the result of encapsulating a Central Process Unit (CPU) in a microchip.

The Central Process Unit is sometimes confused with the computer case, probably because the CPU is inside the case.

The main parts of the Microprocessor are the following:

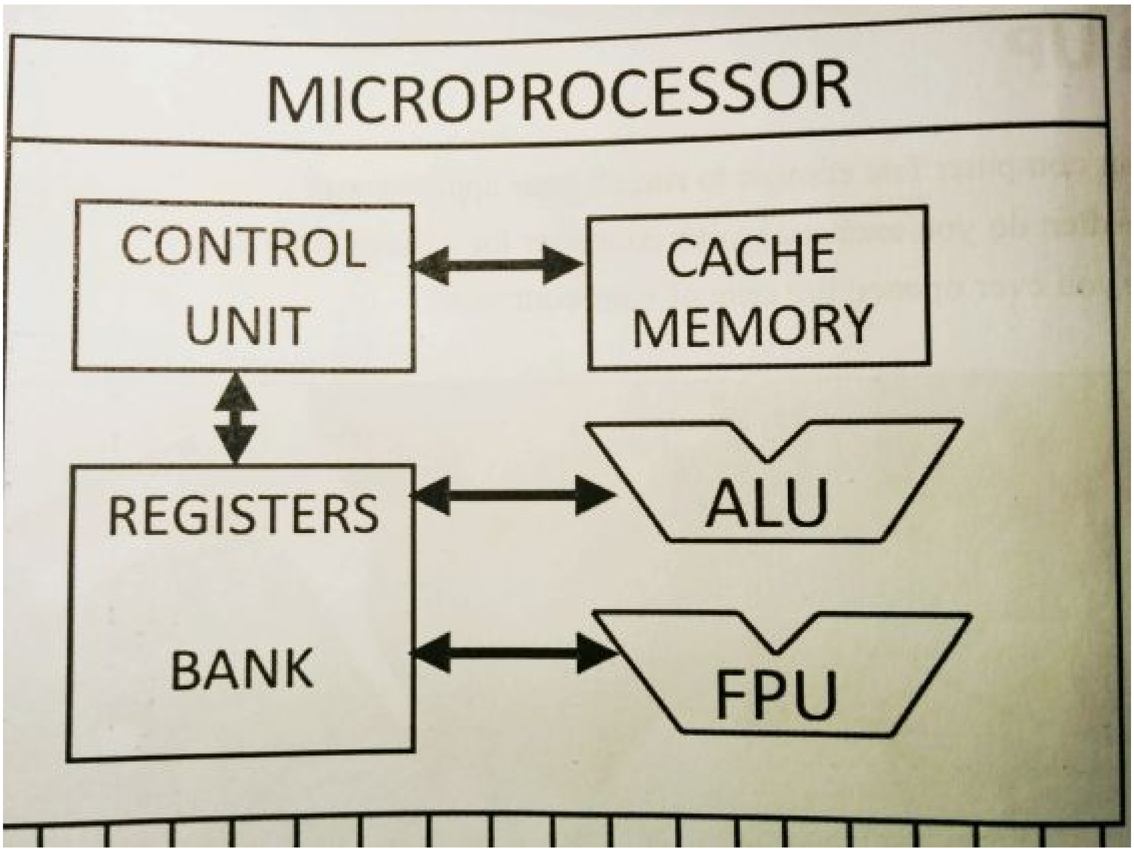
• Control Unit (CU). Its mission is to fetch a program instruction from the memory and execute it.

• Arithmetic-Logic Unit (ALU). It is a pretty fast calculator for integers. It is capable of performing basic operations such as adding, subtracting, multiplying and dividing; not to mention, logic operations such as OR, AND, XOR, etc.

• Floating Point Unit (FPU). It is an ALU for real numbers. A computer can represent a finite range of infinite real numbers. Thanks to the Floating Point Unit, a computer can approximate real numbers in the range (-10^38..10^38) for single precision and (-10^308..10^308) for double precision.

• Registers bank. A register is a memory cell, the smallest and fastest memory that computers have. These registers are connected directly to the ALU and the FPU. They are used to store the operands and results of executed instructions.

• Cache memory. This memory is faster than RAM, but slower than Registers. It is used to store recently executed instructions and data. Most of them are predictive because they are able to predict and get the instructions which are about to be executed. That memory helps the computer to improve its performance.



Software advances very quickly.

Because of this, more powerful microprocessors are needed to support new technologies.

Performance improvement has become a tricky problem to solve.

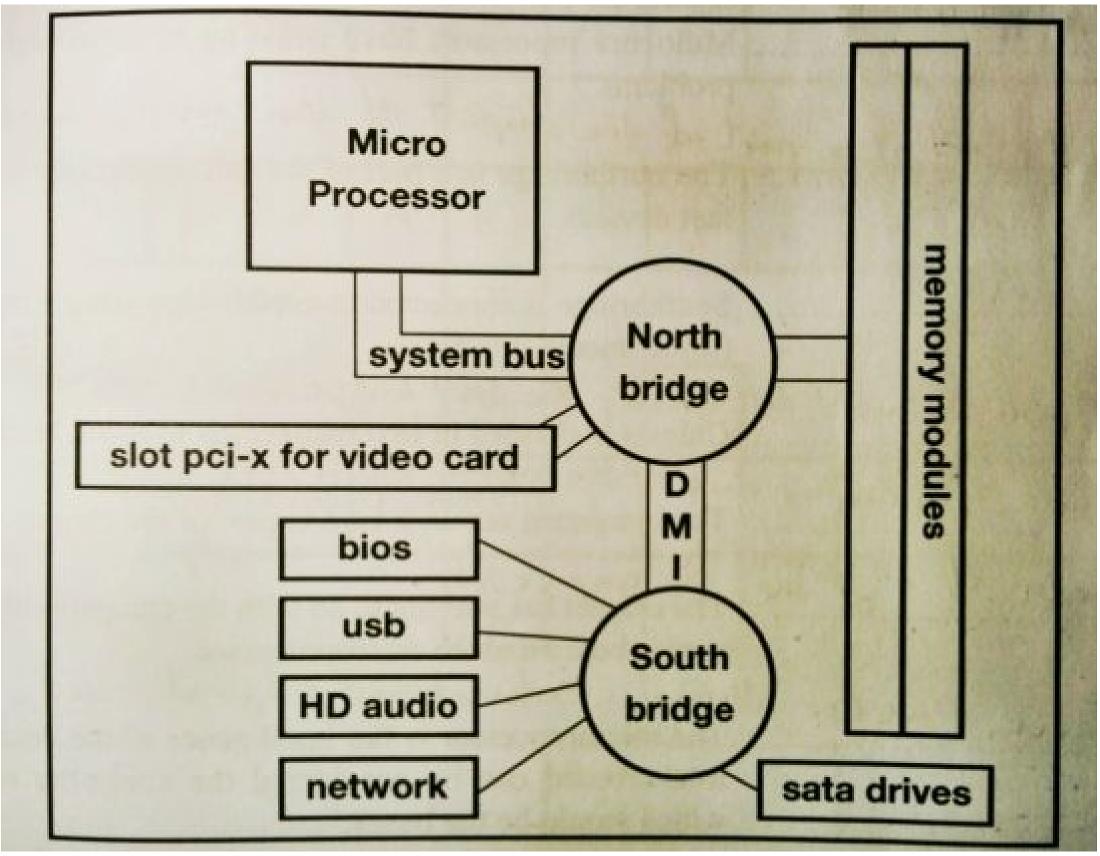
Traditionally, this problem has been solved through miniaturization, in other words: making the components smaller, grouping thousands of millions of transistors in the same amount of space. Another way to obtain faster processors is to increase the clock speed, although clock speeding causes a great increase of temperature that is difficult to cool.

For that reason, manufacturers paved the way for multicore technology: in order to improve microprocessor's performance, the best option is adding two or more processors in the same chip.

The motherboard

The motherboard contains a socket to house the microprocessor. This socket has enough lines to communicate the rest of the computer with the processor. These lines are grouped in units called buses and they transport information to the memory, the graphics card and the peripheral devices.

All these buses are controlled by two main chips. On the one hand, the North Bridge chip, which controls the faster links, like the link between the processor and the memory. On the other hand, the South Bridge chip is responsible for slower communications like the link between the processor and serial ports. It is directly connected to the North Bridge through a special bus called Direct Media Interface (DMI).



This organization improves the computer's performance by avoiding bottlenecks caused by transmissions from the slowest devices which are not fast enough to travel through the North Bridge.

There are also other important chips included in the motherboard like the BIOS and other integrated devices such as the network card, USB controllers, etc.

- 4. 3. 2. Exercises - The Microprocessor and the Motherboard -

A) Answer the following questions:

1. Why are computer processors called microprocessors?
2. Which functions does a processor carry out?
3. Which is the main function of the motherboard?
4. What is the difference between registers and cache memory?
5. Which operations can a FPU perform that an ALU cannot?
6. What is the main problem when increasing the frequency of the clock of the processor?
7. What is the purpose of the motherboard buses?

B) Are the following statements true or false? Justify your answers:

1. Performance improvement is almost impossible to achieve.
2. Multicore processors have come up to solve high temperature problems.
3. The northbridge is a part of the microprocessor which controls fast devices.
4. The southbridge is connected to the northbridge using a bus called DMI (Direct Media Interface).
5. The chipset is divided in two main chips to avoid bottlenecks.
6. The integrated network card is part of the chipset.
7. The chipset has nothing to do with the compatibility between the motherboard and the microprocessor.
8. The microprocessor is the intelligence of the computer and the motherboard can be considered the controller of the nerves, which would be the buses.
9. The best way to improve performance is to group millions of transistors in a very short amount of space.
10. The reason why manufacturers release new hardware very quickly is the continuing advance of software technologies.

- 4. 3. 3. CPU -

CPU

A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.

Traditionally, the term “CPU” refers to a processor, more specifically to its processing unit and control unit (CU), distinguishing these core elements of a computer from external components such as main memory and I/O circuitry.

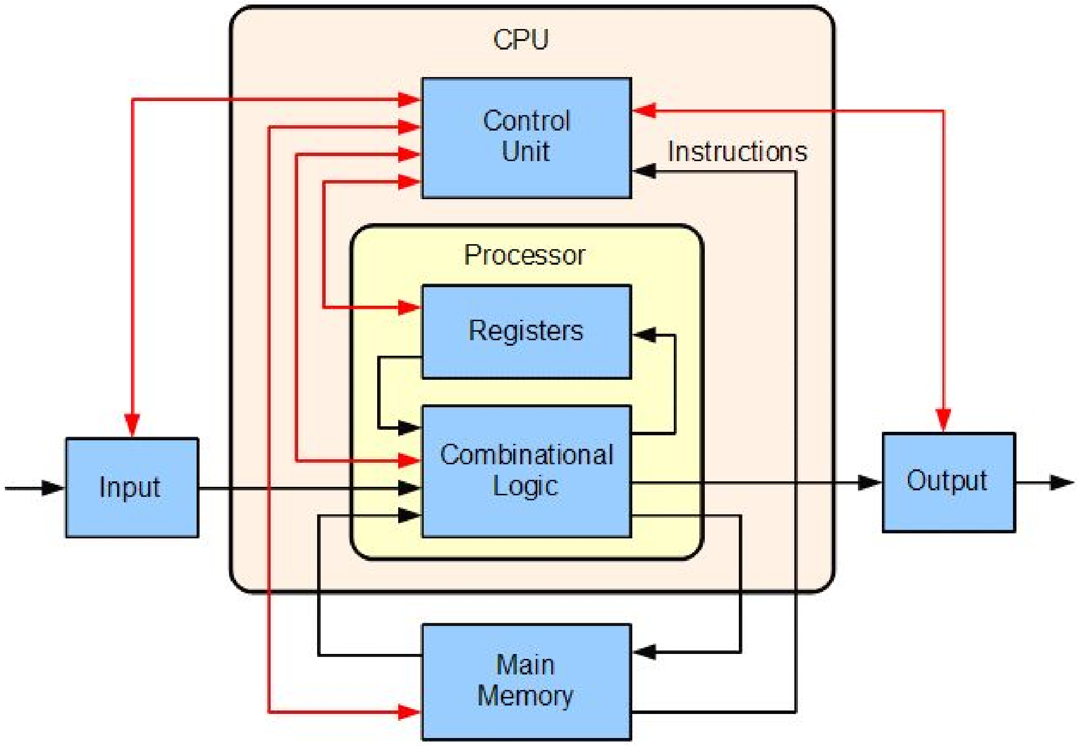
The form, design and implementation of CPUs have changed over the course of their history, but their fundamental operation remains almost unchanged.

Principal components of a CPU include the arithmetic logic unit (ALU) that performs arithmetic and logic operations, processor registers that supply operands to the ALU and store the results of ALU operations, and a control unit (CU) that fetches instructions from memory and “executes” them by directing the coordinated operations of the ALU, registers and other components.

Most modern CPUs are microprocessors, meaning they are contained on a single integrated circuit (IC) chip.

An IC that contains a CPU may also contain memory, peripheral interfaces, and other components of a computer; such integrated devices are variously called microcontrollers or systems on a chip (SoC).

Some computers employ a multi-core processor, which is a single chip containing two or more CPUs called “cores”; in that context, single chips are sometimes referred to as “sockets”. Array processors or vector processors have multiple processors that operate in parallel, with no unit considered central.



- 4. 3. 4. Video - Inside your computer -

Watch the following video:

https://www.youtube.com/watch?v=AkFi90IZmXA

and answer to these questions:

1. The mouse has a couple of buttons and a system for ...
2. When your mouse's click is received, it's handled by the ...
3. Basically, the BIOS provides a way for the computer to ...
4. The BIOS also acts as a ...
5. Which subsystem decides if your mouse's click is pretty important to interrupt the CPU?
6. The CPU is the ... of the whole computer.
7. The CPU's job is ...
8. Your computer's CPU is one heck of a ...
9. There are programs for everything that ...

10. Each program was iniatilly written by ...
11. Human programs take up ...
12. Just the simple task of clicking your mouse means ...

- 4. 3. 5. The CPU Operation Cycle -

Operation

The fundamental operation of most CPUs, regardless of the physical form they take, is to execute a sequence of stored instructions that is called a program.

The instructions to be executed are kept in some kind of computer memory.

Nearly all CPUs follow the fetch, decode and execute steps in their operation, which are collectively known as the instruction cycle.

After the execution of an instruction, the entire process repeats, with the next instruction cycle normally fetching the next-in-sequence instruction because of the incremented value in the program counter.

If a jump instruction was executed, the program counter will be modified to contain the address of the instruction that was jumped to and program execution continues normally.

In more complex CPUs, multiple instructions can be fetched, decoded, and executed simultaneously.

Some instructions manipulate the program counter rather than producing result data directly; such instructions are generally called “jumps” and facilitate program behavior like loops, conditional program execution (through the use of a conditional jump), and existence of functions.

In some processors, some other instructions change the state of bits in a “flags” register. These flags can be used to influence how a program behaves, since they often indicate the outcome of various operations. For example, in such processors a “compare” instruction evaluates two values and sets or clears bits in the flags register to indicate which one is greater or whether they are equal; one of these flags could then be used by a later jump instruction to determine program flow.

Fetch

The first step, fetch, involves retrieving an instruction (which is represented by a number or sequence of numbers) from program memory.

The instruction's location (address) in program memory is determined by a program counter (PC), which stores a number that identifies the address of the next instruction to be fetched.

After an instruction is fetched, the PC (Program Counter) is incremented by the length of the instruction so that it will contain the address of the next instruction in the sequence.

Often, the instruction to be fetched must be retrieved from relatively slow memory, causing the CPU to stall while waiting for the instruction to be returned. This issue is largely addressed in modern processors by caches and pipeline architectures.

Decode

The instruction that the CPU fetches from memory determines what the CPU will do.

In the decode step, performed by the circuitry known as the instruction decoder, the instruction is converted into signals that control other parts of the CPU.

The way in which the instruction is interpreted is defined by the CPU's instruction set architecture (ISA).

Often, one group of bits (that is, a "field") within the instruction, called the opcode, indicates which operation is to be performed, while the remaining fields usually provide supplemental information required for the operation, such as the operands. Those operands may be specified as a constant value (called an immediate value), or as the location of a value that may be a processor register or a memory address, as determined by some addressing mode.

In some CPU designs the instruction decoder is implemented as a hardwired, unchangeable circuit.

In others, a microprogram is used to translate instructions into sets of CPU configuration signals that are applied sequentially over multiple clock pulses.

In some cases the memory that stores the microprogram is rewritable, making it possible to change the way in which the CPU decodes instructions.

Execute

After the fetch and decode steps, the execute step is performed.

Depending on the CPU architecture, this may consist of a single action or a sequence of actions.

During each action, various parts of the CPU are electrically connected so they can perform all or part of the desired operation and then the action is completed, typically in response to a clock pulse.

Very often the results are written to an internal CPU register for quick access by subsequent instructions.

In other cases results may be written to slower, but less expensive and higher capacity main memory.

For example, if an addition instruction is to be executed, the arithmetic logic unit (ALU) inputs are connected to a pair of operand sources (numbers to be summed), the ALU is configured to perform an addition operation so that the sum of its operand inputs will appear at its output, and the ALU output is connected to storage (e.g., a register or memory) that will receive the sum. When the clock pulse occurs, the sum will be transferred to storage and, if the resulting sum is too large (i.e., it is larger than the ALU's output word size), an arithmetic overflow flag will be set.

- 4. 3. 6. Videos - The CPU Operation Cycle: Fetch, Decode, Execute -

Watch the following videos:

https://www.youtube.com/watch?v=jFDMZpkUWCw

https://www.youtube.com/watch?v=Ne8mqikvFh4

- 4. 3. 7. Exercises - The CPU Operation Cycle: Fetch, Decode, Execute -

Click on the following link to go to the "Simple CPU Simulator".

You have to make exercises 1 to 19 following carefully the given instructions.

Open the Google Document with the exercises of this sub-unit and write there your answers to exercises 1 to 19.

Remember that in some exercises you have to give 2 answers: "Expected Results" and "Actual Results".

- 4. 3. 8. Clock Rate & Parallelism -

Clock rate

Most CPUs are synchronous circuits, which means they employ a clock signal to pace their sequential operations. The clock signal is produced by an external oscillator circuit that generates a consistent number of pulses each second in the form of a periodic square wave.

The frequency of the clock pulses determines the rate at which a CPU executes instructions and, consequently, the faster the clock, the more instructions the CPU will execute each second.

Higher clock rates in increasingly complex CPUs make it more difficult to keep the clock signal in phase (synchronized) throughout the entire unit.

This has led many modern CPUs to require multiple identical clock signals to be provided to avoid delaying a single signal significantly enough to cause the CPU to malfunction.

Another major issue, as clock rates increase dramatically, is the amount of heat that is dissipated by the CPU.

The constantly changing clock causes many components to switch regardless of whether they are being used at that time.

In general, a component that is switching uses more energy than an element in a static state. Therefore, as clock rate increases, so does energy consumption, causing the CPU to require more heat dissipation in the form of CPU cooling solutions.

Parallelism

Parallel computing is a type of computation in which many calculations are carried out simultaneously, or the execution of processes are carried out simultaneously.

Large problems can often be divided into smaller ones, which can then be solved at the same time.

When referring to parallelism in CPUs, two terms are generally used to classify these design techniques:

- Instruction-Level Parallelism (ILP), which seeks to increase the rate at which instructions are executed within a CPU (that is, to increase the utilization of on-die execution resources).
- Task-Level Parallelism (TLP), which purposes to increase the number of threads or processes that a CPU can execute simultaneously.

Each methodology differs both in the ways in which they are implemented, as well as the relative effectiveness they afford in increasing the CPU's performance for an application.

A) Instruction-Level Parallelism

One of the simplest methods used to accomplish increased parallelism is to begin the first steps of instruction (fetching and decoding) before the prior instruction finishes executing.

This is the simplest form of a technique known as instruction pipelining, and is utilized in almost all modern general-purpose CPUs.

Pipelining allows more than one instruction to be executed at any given time by breaking down the execution pathway into discrete stages.

This separation can be compared to an assembly line, in which an instruction is made more complete at each stage until it exits the execution pipeline and is retired.

Pipelining does, however, introduce the possibility for a situation where the result of the previous operation is needed to complete the next operation; a condition often termed data dependency conflict.

To cope with this, additional care must be taken to check for these sorts of conditions and delay a portion of the instruction pipeline if this occurs.

B) Task-Level Parallelism

Another strategy of achieving performance is to execute multiple threads or processes in parallel.

This area of research is known as parallel computing.

One technology used for this purpose was Multi-Processing (MP).

The initial flavor of this technology is known as Symmetric Multi-Processing (SMP), where a small number of CPUs share a coherent view of their memory system.

In this scheme, each CPU has additional hardware to maintain a constantly up-to-date view of memory.

By avoiding stale views of memory, the CPUs can cooperate on the same program and programs can migrate from one CPU to another.

To increase the number of cooperating CPUs beyond a handful, schemes such as Non-Uniform Memory Access (NUMA) and directory-based coherence protocols were introduced in the 1990s.

SMP (Symmetric Multi-Processing) systems are limited to a small number of CPUs, while NUMA (Non-Uniform Memory Access) systems have been built with thousands of processors.

Initially, Multi-Processing was built using multiple discrete CPUs and boards to implement the interconnect between the processors.

When the processors and their interconnect are all implemented on a single chip, the technology is known as Chip-level Multi-Processing (CMP) and the single chip as a multi-core processor.

It was later recognized that finer-grain parallelism existed with a single program.

A single program might have several threads (or functions) that could be executed separately or in parallel.

Some of the earliest examples of this technology implemented input/output processing such as direct memory access as a separate thread from the computation thread.

A more general approach to this technology was introduced in the 1970s when systems were designed to run multiple computation threads in parallel.

This technology is known as Multi-Threading (MT).

This approach is considered more cost-effective than multiprocessing, as only a small number of components within a CPU is replicated to support MT (Multi-Threading) as opposed to the entire CPU in the case of MP (Multi-Processing).

In Multi-Threading, the execution units and the memory system including the caches are shared among multiple threads.

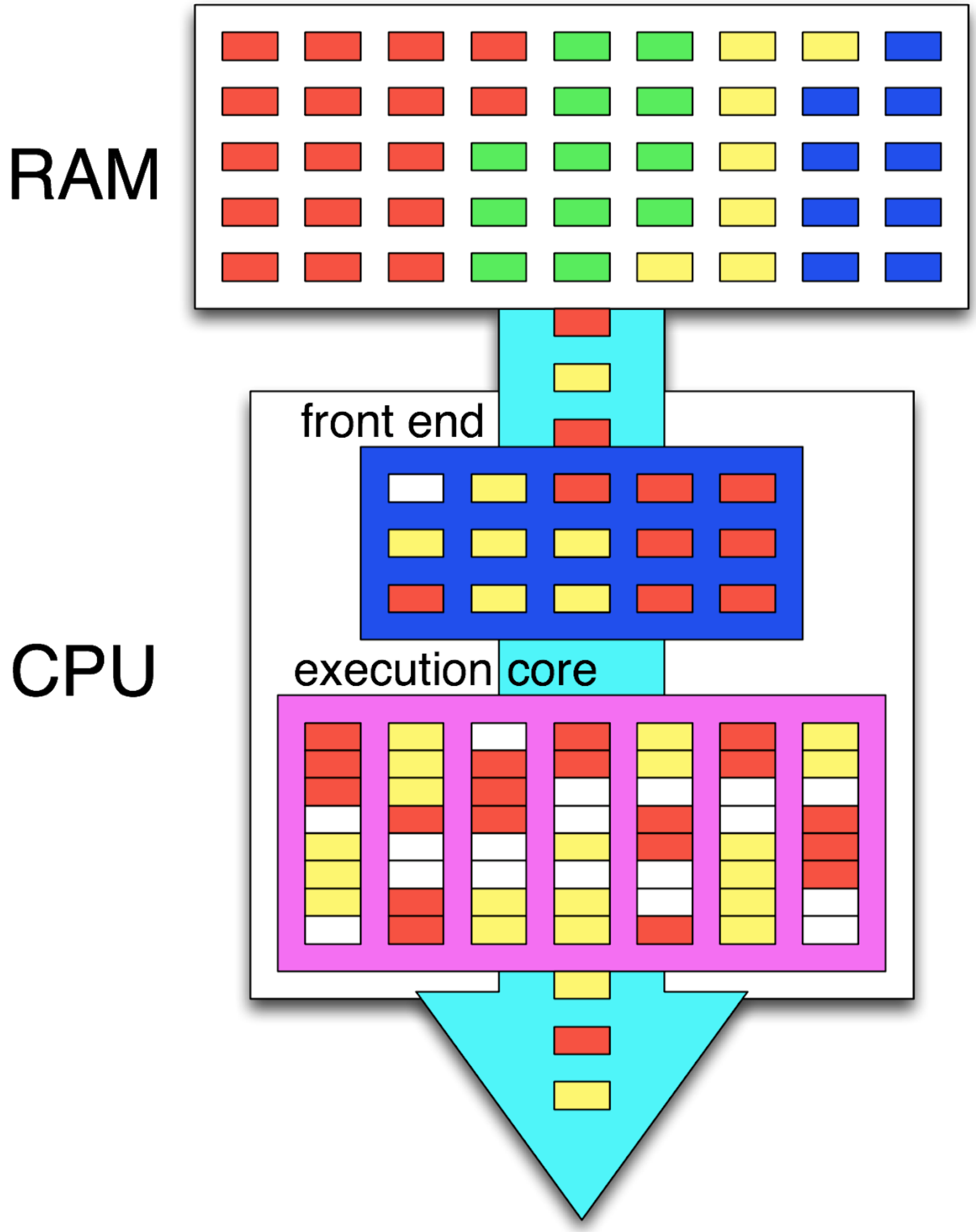
The downside of Multi-Threading is that the hardware support for Multi-Threading is more visible to software than that of Multi-Processing and thus supervisor software like operating systems have to undergo larger changes to support MT.

- 4. 3. 9. Hyper-Threading -

Hyper-Threading is Intel's proprietary simultaneous multithreading (SMT) implementation used to improve parallelization of computations (doing multiple tasks at once) performed on x86 microprocessors.

For each processor core that is physically present, the operating system addresses two virtual (logical) cores and shares the workload between them when possible.

The main function of hyper-threading is to increase the number of independent instructions in the pipeline; it takes advantage of superscalar architecture, in which multiple instructions operate on separate data in parallel.



With Hyper-Threading technology, one physical core appears as two processors to the operating system, allowing concurrent scheduling of two processes per core.

In addition, two or more processes can use the same resources: if resources for one process are not available, then another process can continue if its resources are available.

In addition to requiring simultaneous multithreading (SMT) support in the operating system, hyper-threading can be properly utilized only with an operating system specifically optimized for it.

Furthermore, Intel recommends Hyper-Threading technology to be disabled when using operating systems unaware of this hardware feature.

Architecturally, a processor with Hyper-Threading technology consists of two logical processors per core, each of which has its own processor architectural state.

Each logical processor can be individually halted, interrupted or directed to execute a specified thread, independently from the other logical processor sharing the same physical core.

Unlike a traditional dual-processor configuration that uses two separate physical processors, the logical processors in a hyper-threaded core share the execution resources.

These resources include the execution engine, caches, and system bus interface.

The sharing of resources allows two logical processors to work with each other more efficiently, and allows a logical processor to borrow resources from a stalled logical core (assuming both logical cores are associated with the same physical core).

A processor stalls when it is waiting for data it has sent for so it can finish processing the present thread.

The degree of benefit seen when using a hyper-threaded or multi core processor depends on the needs of the software, and how well it and the operating system are written to manage the processor efficiently.

This technology is transparent to operating systems and programs.

The minimum that is required to take advantage of hyper-threading is symmetric multiprocessing (SMP) support in the operating system, as the logical processors appear as standard separate processors.

- 4. 3. 10. Video - Hyper-Threading -

Watch the following video:

https://www.youtube.com/watch?v=zloLAGKr3b4

and answer to these questions:

1. True or false? Hyper-Threading means doubling the physical cores of a processor.
2. True or false? Hyper-Threading magically increases core count.
3. Hyper-Threading is a technology that tries to … processes.
4. When the processor uses Hyper-Threading technology … logical cores comprise a single physical core.
5. By implementing Hyper-Threading on a single core … logical cores are detected by … .
6. The scheduler in charge will more appropiately divide the data between the … logical cores in an effort to maximise … .
7. Only … pipeline at a time can send … through to be processed.
8. You will need … to process data simultaneously.
9. If only one logical core exists per physical core, the scheduler … prepare the data while sending data, resulting in …, … and … .
10. Intel Core i5-6400 … Hyper-Threading.
11. Intel Core i7-6700K … Hyper-Threading.
12. In the case of heavy … , Hyper-Threading can come in … .
13. The next time you are in hunt for a new processor, ask yourself whether or not your … will benefit from … .

- 4. 3. 11. Multi-Core -

A **multi-core** processor is an integrated circuit (IC) to which 2 or more processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks.

A single integrated circuit is used to package or hold these processors.

These single integrated circuits are known as a die.

Multicore architecture places multiple processor cores and bundles them as a single physical processor.

The objective is to create a system that can complete more tasks at the same time, thereby gaining better overall system performance.

The concept of multicore technology is mainly centered on the possibility of parallel computing, which can significantly boost computer speed and efficiency by including 2 or more central processing units (CPUs) in one single chip.

This reduces the system's heat and power consumption.

This means much better performance with less or the same amount of energy.

The architecture of a multicore processor enables communication between all available cores to ensure that the processing tasks are divided and assigned accurately.

At the time of task completion, the processed data from each core is delivered back to the motherboard by means of a single shared gateway.

This technique significantly enhances performance compared to a single-core processor of similar speed.

Multicore technology is very effective in challenging tasks and applications, such as encoding, 3-D gaming and video editing.

Multicore-based processors are used in mobile devices, desktops, workstations and servers.

A **CPU cache** is a hardware cache used by the central processing unit (CPU) of a computer to reduce the average cost (time or energy) to access data from the main memory (RAM).

A cache is a smaller, faster memory, closer to a processor core, which stores copies of the data from frequently used main memory locations.

Most CPUs have different independent caches, including instruction and data caches, where the data cache is usually organized as a hierarchy of more cache levels (L1, L2, L3).

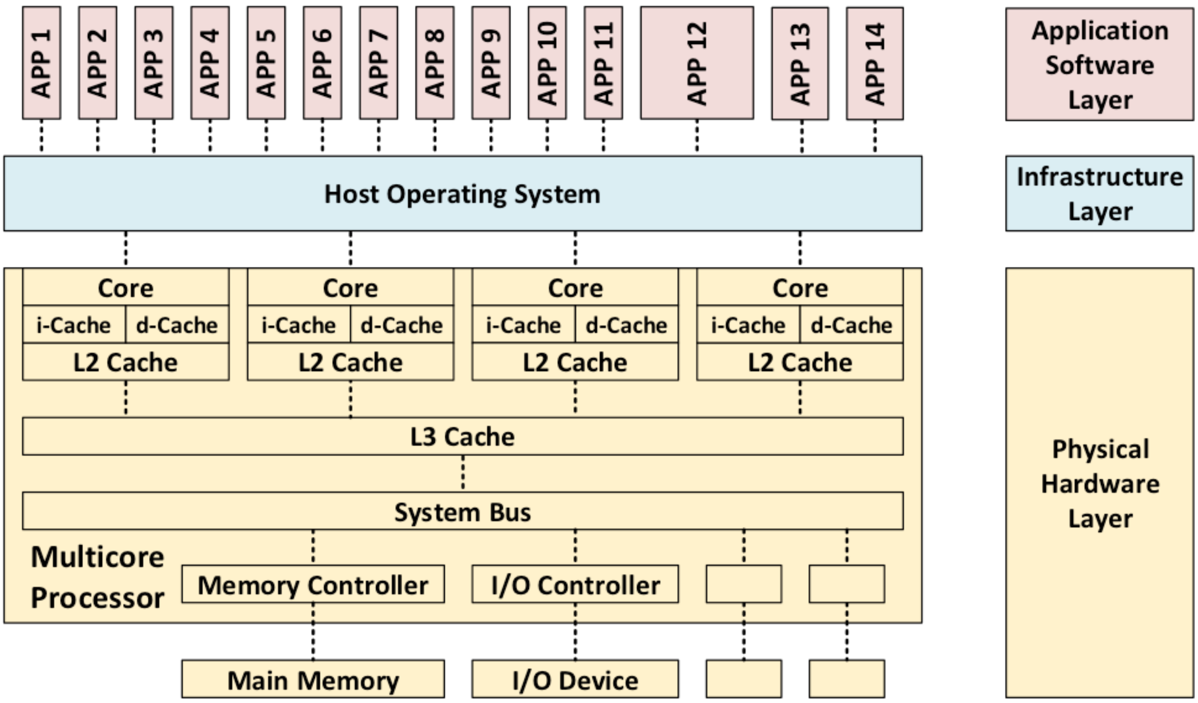
Almost all current CPUs with caches have a split L1 cache, one L1 cache for data and other L1 cache for instructions.

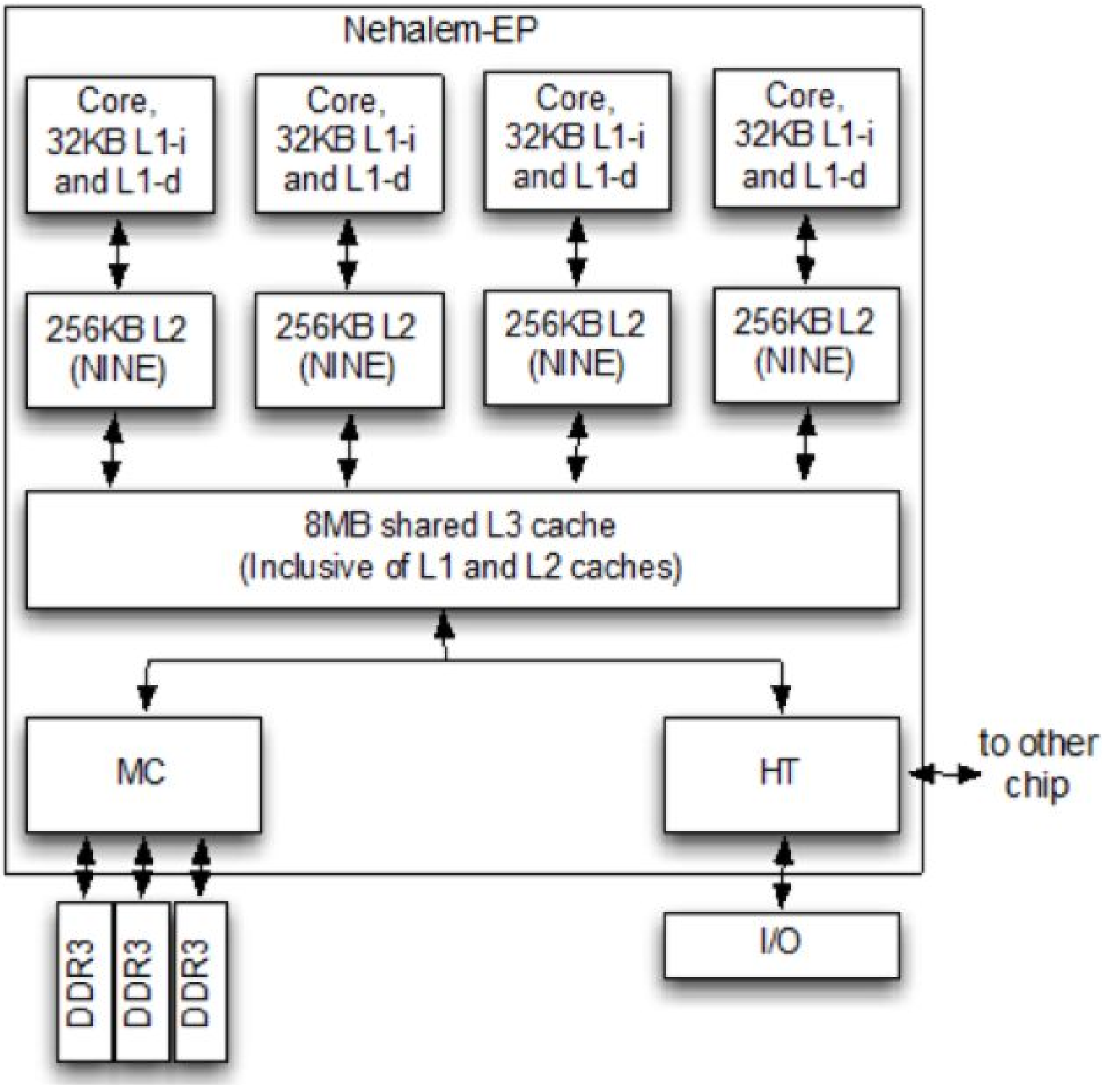
CPUs also have L2 caches and, for larger processors, L3 caches as well.

The L2 cache is usually not split and acts as a common repository for the already split L1 cache.

Every core of a multi-core processor has a dedicated L2 cache and is usually not shared between the cores.

The L3 cache is shared between the cores and are not split.


APP 1
APP 2
APP 3
APP 4
APP 5
APP 6
APP 7
APP 8
APP 9
APP 10
APP 11
APP 12
APP 13
APP 14
Application Software Layer
Host Operating System
Infrastructure Layer
Core
i-Cache
d-Cache
L2 Cache
L3 Cache
System Bus
Multicore Processor
Memory Controller
I/O Controller
Main Memory
I/O Device
Physical Hardware Layer



- 4. 3. 12. Video - Multi-Core -

Watch the following video:

https://www.youtube.com/watch?v=S3I5WNHbnJ0

and answer to these questions:

1. Higher core counts suggest more ...
2. Super high-end CPUs with many cores or Hyper-Threading ... in games.
3. Parallelization means ...
4. Can all programs take advantage of multiple CPU cores?

5. Which types of programs do lots of calculations that can be parallelized easily?
6. Are GPUs designed for parallel processing?
7. Which other important aspects of games are still handled by the CPU?
8. Which things are much harder for programmers to parallelize and split the workload?
9. If you are just using your PC to game, getting ...

- 4. 3. 13. Video - Intel Processor Generations -

Watch the following video:

https://www.youtube.com/watch?v=FPks89DvOms

and answer to these questions:

1. When did Intel introduce its first processor?
2. The 8086 was a ... bit processor.
3. The x86 instruction set remains ...
4. What is Moore's Law?
5. The first 32-bit Intel processor was ... and was introduced in ... .
6. Intel's 386 processor was capable of addressing ... of system RAM, a limitation that actually wasn't left until almost ... years later.
7. The Pentium generation added ...
8. The Celeron 300A was notable for ...
9. The Pentium III generation introduced SpeedStep, the ability ...
10. The Pentium III was Intel's first CPU to include ... and to break the ... GHz barrier.
11. The Pentium IV introduced ... technology in ... and the ... type socket in ... .
12. The first Intel processor to support Dual Core technology was ...
13. 2006 marks the ends of the ... war, because ...
14. The very next CPU sockets where: ... for the ..., and ... for the ... .
15. Which technologies replaced the Front Side Bus?
16. The memory controller moved from the ... to the ... .
17. The mainstream Dual Core models introduced ... and is a trend that continues today.
18. In the last decade ... has happened in the Intel's processors evolution.
19. Updated I/O options in the last years include: ... .
20. Cool features in the last years include: ... and ... .

- 4. 3. 14. Video - Intel's 8th Generation CPUs -

Watch the following video:

https://www.youtube.com/watch?v=EJOnwF8mgXc

and answer to these questions:

1. Which 2 CPUs are reviewed in this video?
2. Which is the codename for Intel's 8th generation CPUs?
3. These CPUs are built on the ...... process.
4. True or false? Intel's 8th generation CPUs fit in the same socket as Intel's 6th and 7th generation.
5. Intel's 8th generation CPUs and the 300 chipset series motherboards that go with them ...... backwards compatible.
6. Which is Intel's "excuse" for that?
7. AMD's history is to support a socket for a long time but ......
8. People who invested in fancy Z270 chipset motherboards are ......
9. Which are the codenames for Intel's previous generations?
10. Which is the compatible socket for Intel's 8th generation CPUs?
11. True or false? Intel's 8th generation are getting a new HD graphics chip.
12. With Intel's 8th generation CPUs we are finally getting a ...... Intel chip on the mainstream platform.
13. Intel's 8th generation CPUs are getting ...... cache at ...... TDP.
14. In this video the performance comparison is done between Intel's 8th generation CPUs, ... , ... , ... , ... , ... and ....
15. In the 1080p performance tests, which Intel's 8th generation CPU was better?
16. In the productivity tests, which Intel's 8th generation CPU was better easily surpasing ...... and even approaching ...... ?
17. Themal performance is ...... for the Intel Core i7 8700K.
18. Power consumption is ...... than the last generation.
19. You can overclock the Intel Core i7 8700K at ....
20. In the value – gaming tests, the best overall option is ...... , and it's going to be really disruptive for ...... or ....
21. In the value – productivity tests, the best overall option is ....
22. In the value – overall tests, Intel's 8th generation CPUs ....
23. The release of Intel's 8th generation CPUs is a major ...... for the consumer.

- 4. 3. 15. Video - Intel's 10th Generation CPUs -

Watch the following video:

https://www.youtube.com/watch?v=Kfq-hWajeH4

and answer to these questions:

1. Which is Intel's codename for its desktop 10th generation processors?
2. How many cores / threads does Intel's i9 line of processors have?
3. Which processors of Intel's i9 line of processors have integrated GPU (Graphical Processing Unit)?
4. Which maximum peak speed can you get with Intel's i9 line of processors?
5. Which is the price range for Intel's i9 line of processors?
6. How many cores / threads does Intel's i7 line of processors have?
7. Which processors of Intel's i7 line of processors have integrated GPU (Graphical Processing Unit)?
8. Which maximum peak speed can you get with Intel's i7 line of processors?
9. Which is the price range for Intel's i7 line of processors?
10. In terms of competition, Intel's i7-10700KF is going up against AMD's ...
11. Now we have a situation where Intel could be offering similar ... at a similar ...
12. Intel's competitor to AMD's Ryzen 7 3700X is ...
13. How many cores / threads does Intel's i5 line of processors have?
14. Which maximum peak speed can you get with Intel's i5 line of processors?
15. Which is the range price for Intel's i5 line of processors?
16. All that really needed to happen was for Intel to ...
17. Intel's i3 line of processors pricing is a ...
18. How many cores / threads does Intel's i3 line of processors have?
19. Which maximum peak speed can you get with Intel's i3 line of processors?
20. Which is the range price for Intel's i3 line of processors?
21. In terms of other platform features, the 10th generation lineup will require a new ... socket, so these CPUs are not ...
22. The compatible motherboards for Intel's 10 generation CPUs can have the ... chipsets.
23. Intel's 10 generation CPUs have the unique ability to disable ... on a ...
24. Intel claims that the ... is the fastest gaming processor.
25. Overall, in the ... and the ... ranges in particular, the offerings here are ... especially considering ...
26. Intel couldn't continue offering ... cores at nearly ... and expect to be ... , but now that that's looking like a ... to ... parts, that's a lot better.
27. If you want to use one of Intel's 10th generation processors you will need to buy ...

- 4. 3. 16. Exercise: Oral - Processor -

Working individually, you have to prepare a presentation about a processor.

You have to choose a real processor and prepare a presentation file (using Google Drive Presentations, OpenOffice Impress, Microsoft PowerPoint...), where you will explain the following issues:

- Model name.
- Number of cores.
- Hyper-Threading (Intel) or SMT (AMD).
- Base clock speed.
- Boost availability.
- Cache levels.
- TDP.
- Socket.
- Chipset.
- Onboard graphics availability.
- Architecture.
- Target market.
- Price.
- Final thoughts.
- Any other information that you can find.

After you have prepared the presentation, you must show it to the teacher.

Finally, you must represent the scene to the whole class, without reading any papers (you can show your presentation on the screen).