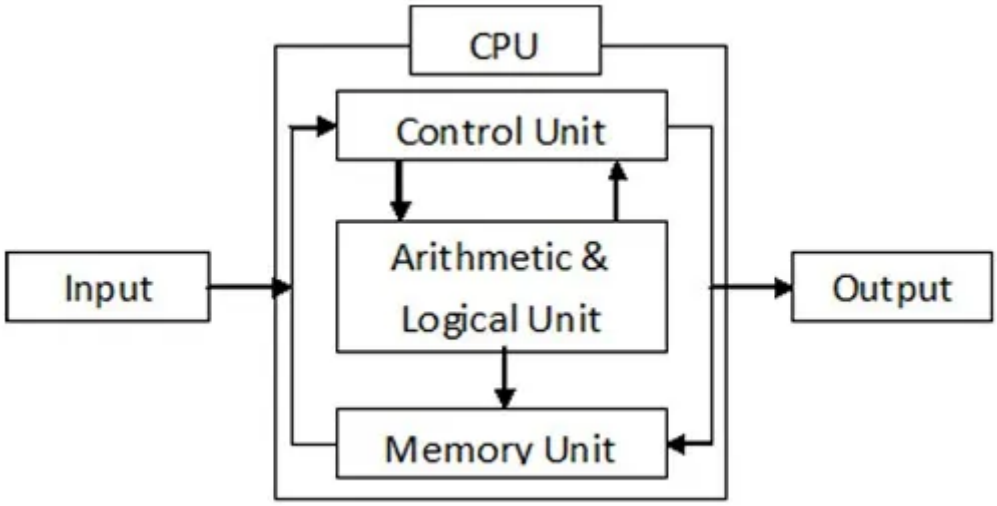
BASIC HARDWARES

ARIHARASUDHAN

HARDWARE COMPONENTS

1. Basic Components



2. CPU vs. GPU

CPU : Central Processing Unit | The Processor | Performs Arithmetic, Logic and IO Operations | Program forwards the instructions to the Processor to be executed | "Brain", or "Heart" of The Computer | The CPU processor is known as the core. The older CPUs were single-core CPUs, but the modern CPUs have cores between 2 to 28. The single core CPUs was focused on a single task. But latest multi-core CPUs can handle multitasking easily. It is for serial instruction processing.

GPU : Graphics Processing Unit | To render the images mainly in computer games | Provides high-throughputs to allow faster performance in gaming | A GPU requires more ALU units than a CPU so we need a dedicated GPU | The GPU contains it's own RAM called VRAM (Video RAM). It is for parallel instruction processing. Although they're best known for their capabilities in gaming, GPUs are becoming more popular for use in creative production and artificial intelligence (AI).

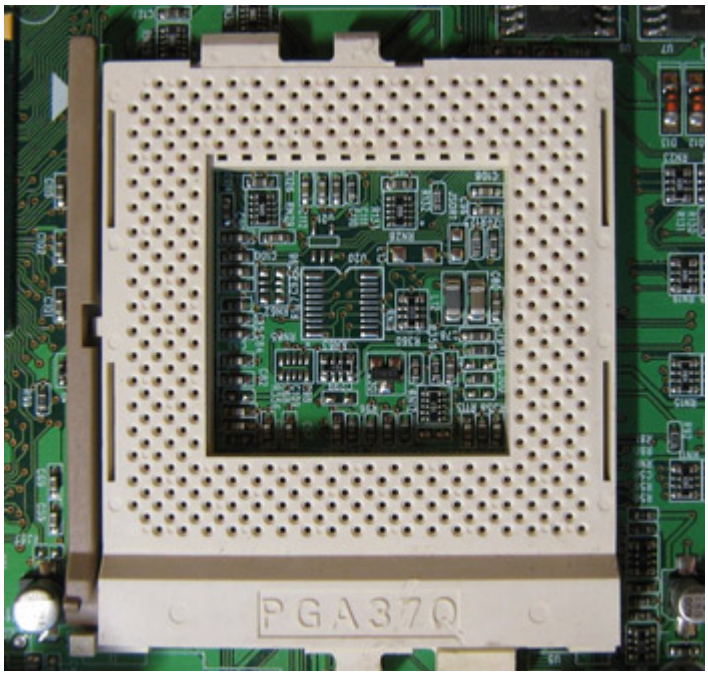
A GPU is like a specialized CPU and is particularly well suited for multitasking. In fact, CPUs used to do the job that GPUs do

today. Rather than processing tasks serially (in order) like a CPU, a GPU breaks up tasks and runs them in parallel. GPUs have many more cores than CPUs, although they are smaller.

3. CORE & SOCKET

CORE : CPU core, is the "brain" of a CPU. It receives instructions, and performs calculations, or operations, to satisfy those instructions. A CPU can have multiple cores. A processor with two cores is called a dual-core processor ; with four cores, a quad-core ; six cores, hexa-core ; eight cores, octa-core.

SOCKET : When referring to a processor, a CPU socket or processor socket is a connection that allows a computer processor to connect to a motherboard. For example, the Socket 370 is an example of such a socket. The picture shows an example of what a socket may look like on a motherboard. Although there were computers that used the slot processor, most computers today and in the past have used socket processors. The processor socket helps determine what computer processors your computer motherboard can accept.



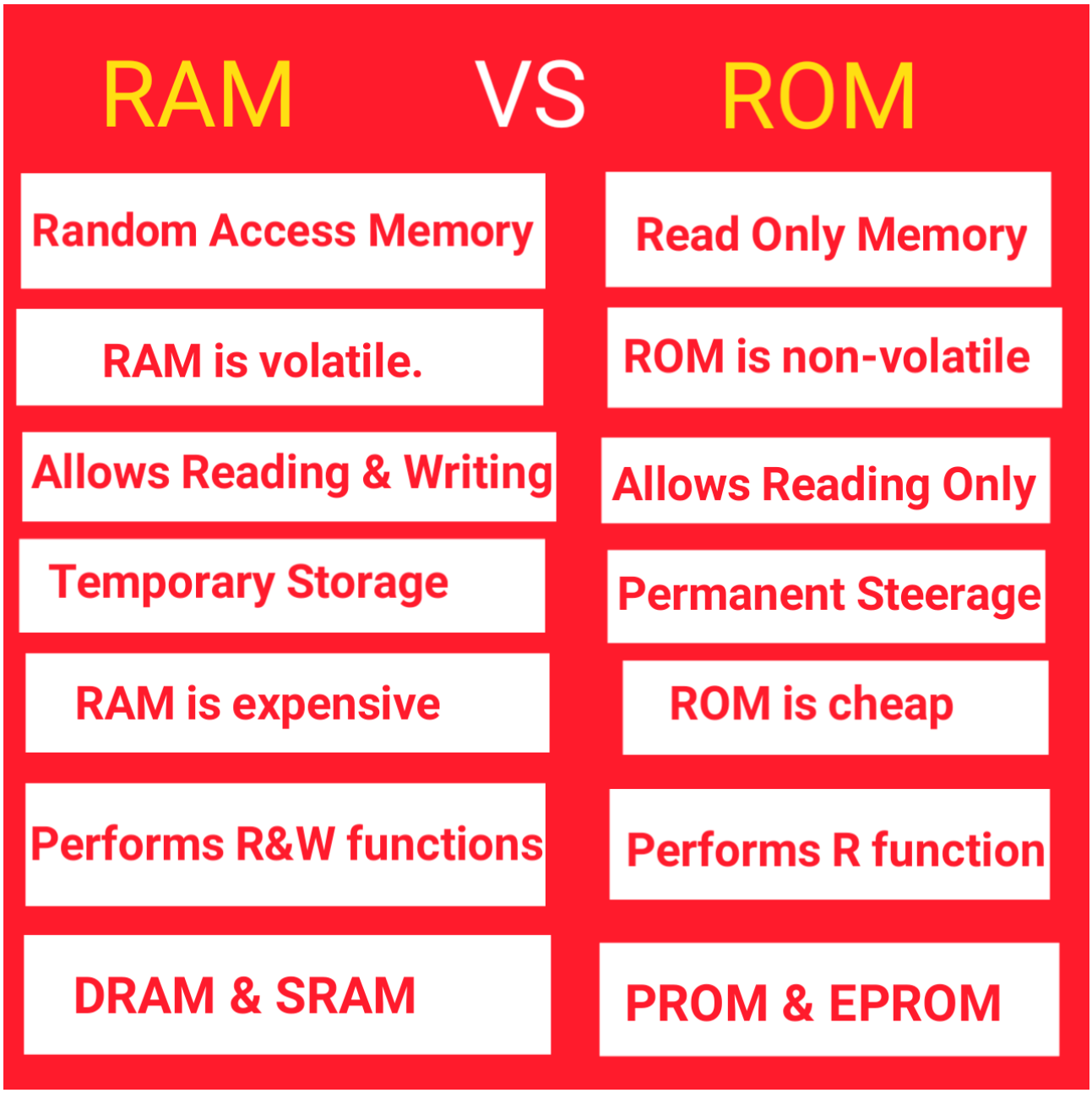
4. CPU FREQUENCY

How many cycles a CPU can execute in a second ? Alternatively called clock rate and processor speed, clock speed is the speed that the microprocessor executes each instruction or each vibration of the clock. The CPU requires a fixed number of clock ticks, or cycles, to execute each instruction. *CPU cycle* Usually, the time required for the execution of one simple processor operation such as an addition. The higher the frequency of the CPU's clock, the more operations it can perform per second. So, as the frequency of the CPU's clock increases, the time required to perform tasks decreases. Clock speeds are measured in MHz, 1 MHz representing 1 million cycles per second, or in GHz, 1 GHz representing 1 thousand million cycles per second. The higher the CPU speed, the better a computer performs, in a general sense. Other components like RAM, hard drive, motherboard, and the number of processor cores (e.g., dual-core or quad-core) can also improve the computer speed. The CPU speed determines how many calculations it can perform in one second of time. The higher the speed, the more calculations it can perform, thus making the computer faster. While several brands of computer processors are available, including Intel and AMD, they all use the same CPU speed standard, to determine what speed each of their processors run. If a processor has dual or quad-cores, the computer's performance can increase even if the CPU speed remains the same. A dual-core 3.0 GHz processor would be capable of performing double the number of calculations as a single-core 3.10 GHz processor.

Why is the frequency of CPUs in super-computers lower than in desktops/servers/workstations? Is it supposed to be higher?

Because of POWER and PRICE.

5. RAM and ROM



6. SSD

Short for solid-state drive, an SSD is a storage medium that uses non-volatile memory to hold and access data. Unlike a hard drive, an SSD has no moving parts, which gives it advantages, such as faster access time, noiseless operation, higher reliability, and lower power consumption.

7. Process vs. Thread

A Process is a computer program under execution. A thread is a lightweight process. A process can do more than one unit of work concurrently by creating one or more threads.

8. Fork vs. Clone

In an operating system, a fork is a Unix or Linux system call to create a new process from an existing running process. The new process is a child process of the calling parent process. `clone()` is a new, versatile system call which can be used to create a new thread of execution.

9. Parallelism vs. Concurrency

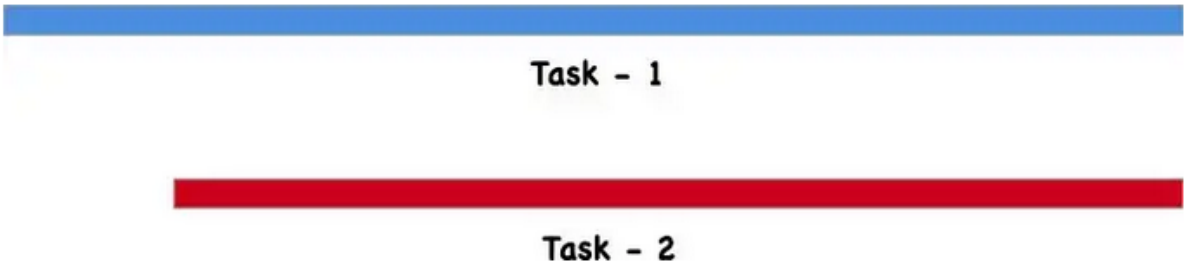
Consider you are given a task of singing and eating at the same time. At a given instance of time either you would sing or you would eat as in both cases your mouth is involved. So in order to do this, you would eat for some time and then sing and repeat this until your food is finished or song is over. So you performed your tasks concurrently.

Concurrency means executing multiple tasks at the same time but not necessarily simultaneously. In a concurrent application, two tasks can start, run, and complete in overlapping time periods i.e Task-2 can start even before Task-1 gets completed.
In the computer science world, the way how concurrency is achieved in various processors is different. In a single core environment (i.e your processor is having a single core), concurrency is achieved via a process called context-switching. If its a multi-core environment, concurrency can be achieved through parallelism.

Execution of tasks in a single core environment. Tasks are context switched between one another.

Consider you are given two tasks of cooking and speaking to your friend over the phone. You could do these two things simultaneously. You could cook as well as speak over the phone. Now you are doing your tasks parallelly.

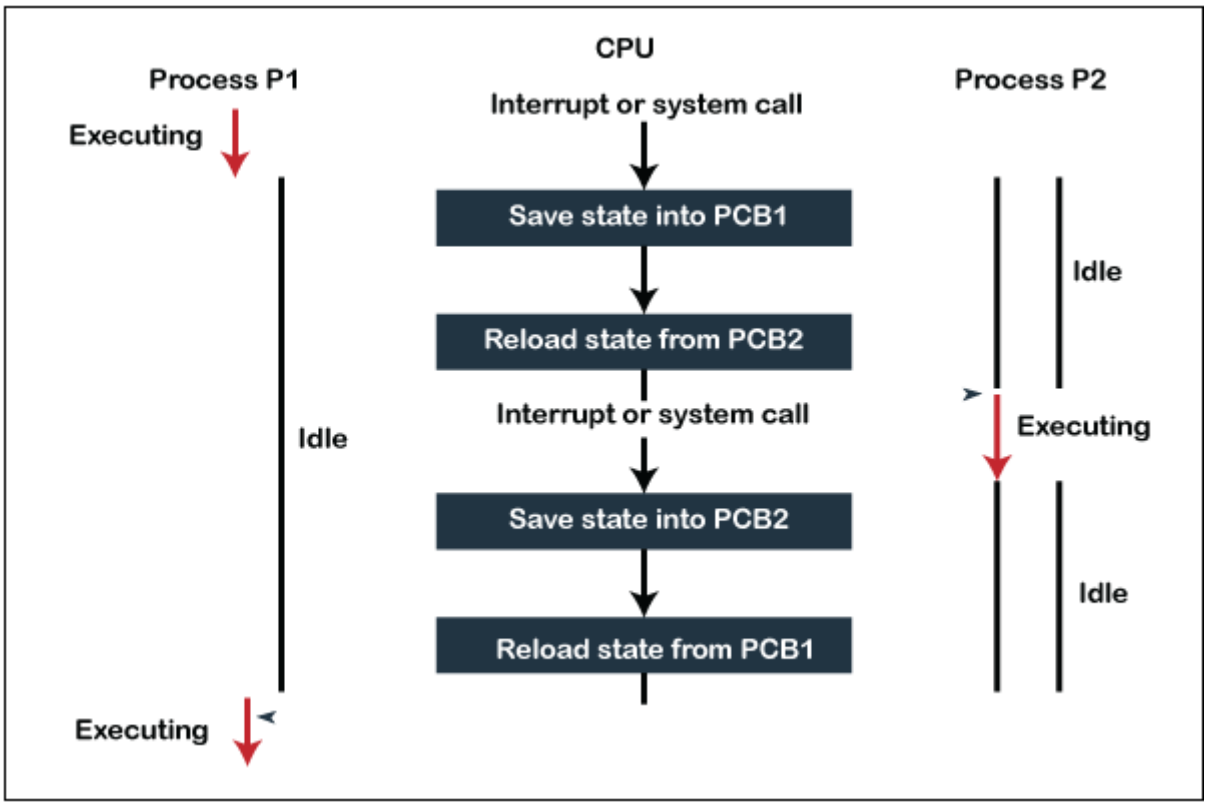
Parallelism means performing two or more tasks simultaneously. Parallel computing in computer science refers to the process of performing multiple calculations simultaneously.



Two tasks are being performed simultaneously over the same time period.

10. Contextual Switching

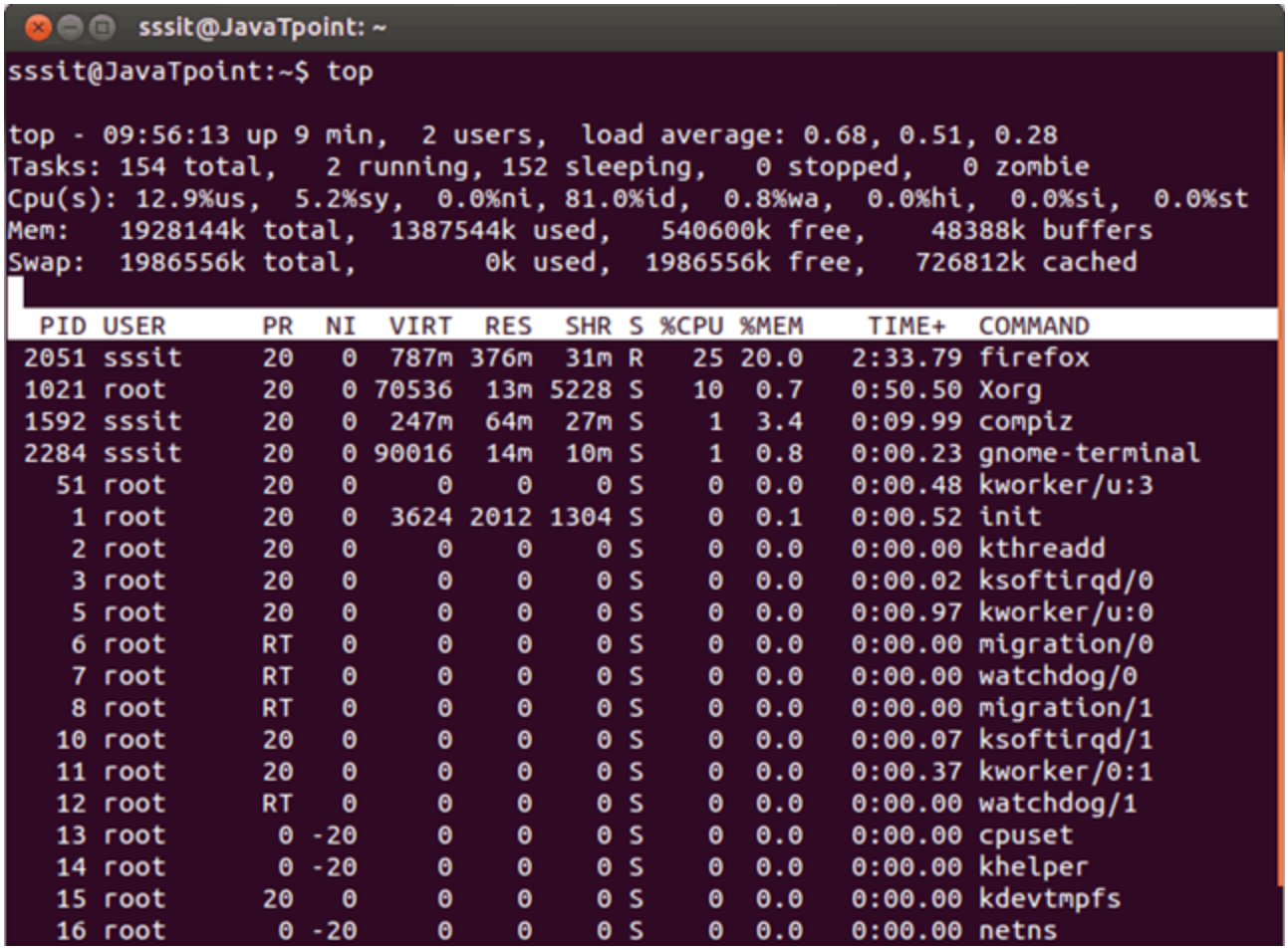
The Context switching is a technique or method used by the operating system to switch a process from one state to another to execute its function using CPUs in the system. When switching perform in the system, it stores the old running process's status in the form of registers and assigns the CPU to a new process to execute its tasks. While a new process is running in the system, the previous process must wait in a ready queue. The execution of the old process starts at that point where another process stopped it. It defines the characteristics of a multitasking operating system in which multiple processes shared the same CPU to perform multiple tasks without the need for additional processors in the system.



Interrupts ----- Multitasking ----- Kernel/User Switch

11. The top and htop Commands

top command : TABLE OF PROCESSES | The top command displays all the running process within the environment of your system. It helps in monitoring system usage and performances. It is mainly used to detect load on the server by system administrators.



htop command : HISHAM'S TOP ; htop uses color and gives visual information about processor, swap and memory status. htop can also display the processes as a tree.

12. DIMM

In computing, a memory module or RAM (random-access memory) stick is a printed circuit board on which memory integrated circuits are mounted. SIMM (Single Inline Memory Module) is a small printed circuit board that plugs into a socket on a personal computer and increases the available random-access memory (RAM). Early modules had one row of electrical contacts and were called SIMMs (single inline memory modules), whereas later modules had two rows and were called DIMMs (dual inline memory modules). They are the predominant method for adding memory into a computer system.

13. Data Rate

Data Rate is defined as the amount of data transmitted during a specified time period.

SDR : Single Data Rate

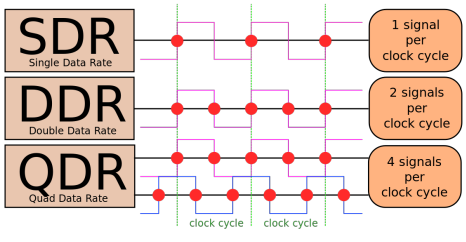
A feature available in Intel devices that transfers data on either the rising edge or the falling edge of the clock signal.

DDR : Double Data Rate

In computing, a computer bus operating with double data rate (DDR) *transfers data on both the rising and falling edges of the clock signal*.

QDR : Quad Data Rate / Quad Pumping

Quad data rate (QDR, or quad pumping) is a communication signaling technique wherein data are transmitted at four points in the clock cycle: on the rising and falling edges, and at two intermediate points between them.



14. Memory Controller

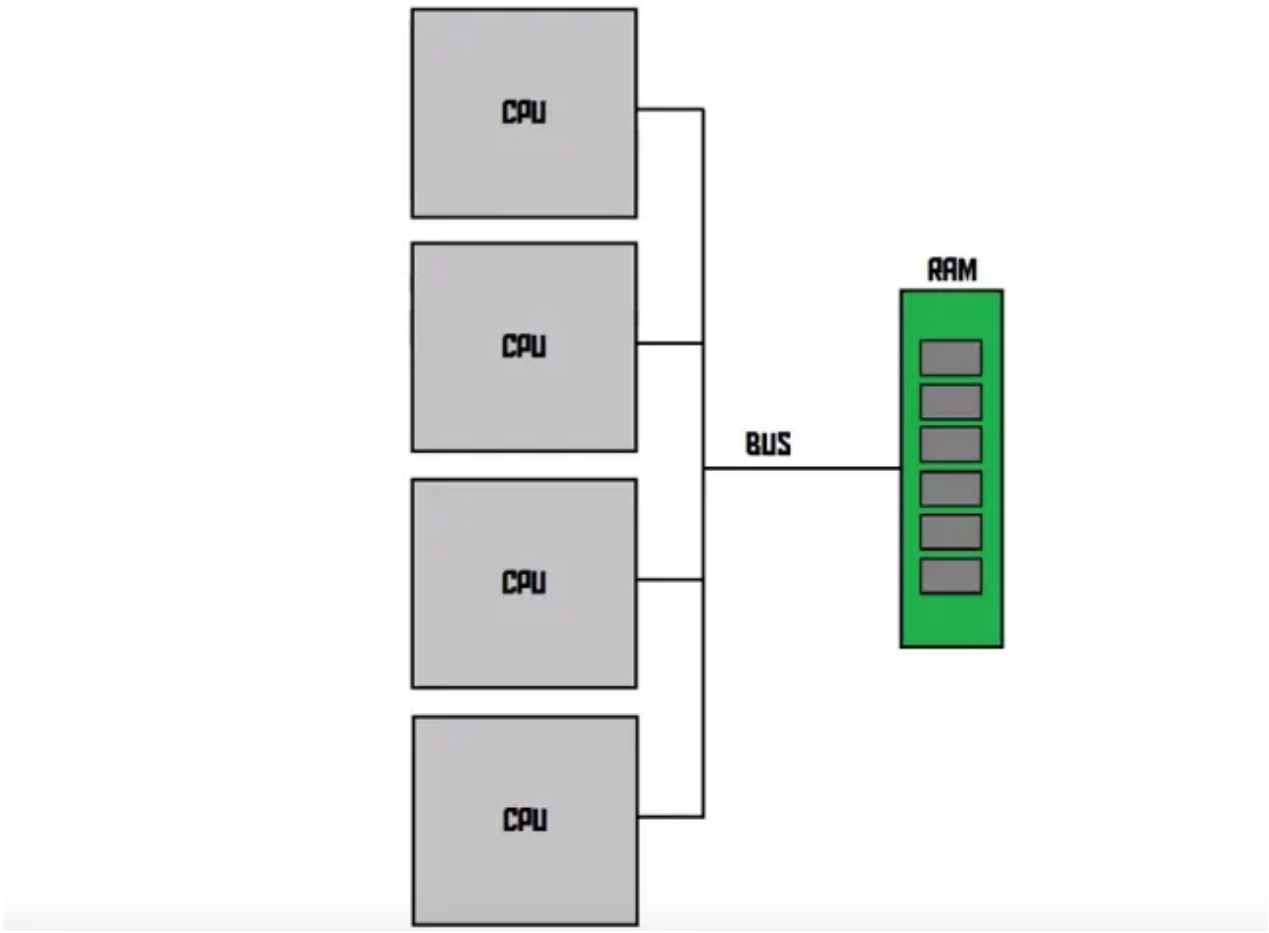
The memory controller is a digital circuit that manages the flow of data going to and from the computer's main memory. A memory controller can be a separate chip or integrated into another chip, such as being placed on the same die or as an integral part of a microprocessor; in the latter case, it is usually called an integrated memory controller (IMC). A memory controller is sometimes also called a memory chip controller (MCC) or a memory controller unit (MCU). Memory controllers contain the logic necessary to read and write.

15. Memory Channel

It is a way to connect CPU to RAM. Basically your standard motherboard has either 2 or 4 RAM slots. And it can support single channel memory, dual channel memory, as well as quad channel memory. Single channel will get you the rated speed of RAM, say 4GB/s , that's the bandwidth between CPU and RAM. With dual channel memory, if you have 2 sticks of RAM that support dual channel, and they are both rated for speeds of 4GB/s, you can basically get 8GB/s bandwidth. It works by taking the bandwidth of both RAM sticks at the same, basically doubling the speed.

16. UMA vs. NUMA

UMA **is an abbreviation for** ***"Uniform Memory Access"*****. It is a multiprocessor shared memory architecture. In this model, all of the processors in the multiprocessor system use and access the same memory with the aid of the interconnection network.**

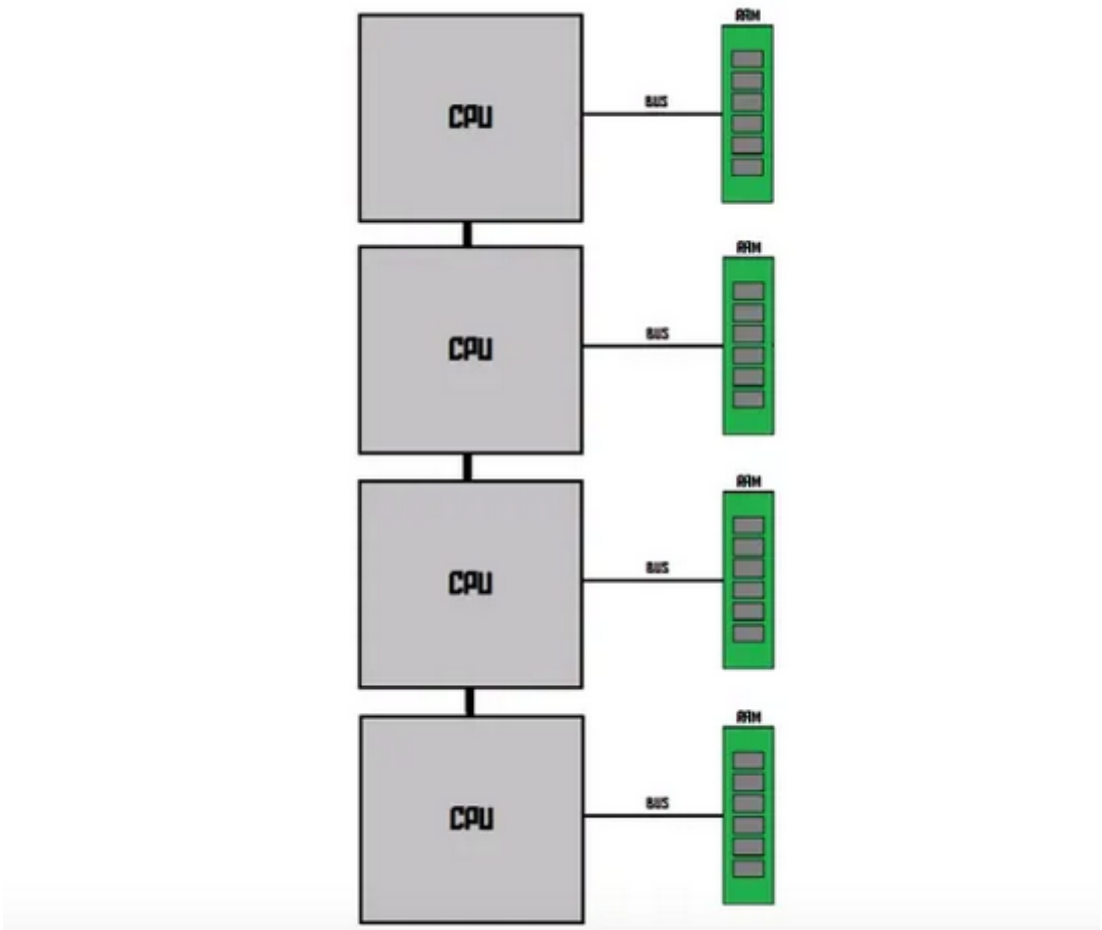


NUMA **is an abbreviation for** ***"Non-uniform Memory Access"*****. It is also a multiprocessor model with dedicated memory attached to each CPU. But these small memory components come together to form a single address space. Memory access time is determined by the distance between the CPU and the memory, resulting in varied memory access times. It provides access to any memory place using the physical address.**

The UMA (Uniform Memory Access) contains a single memory controller. In contrast, the NUMA (Non-Uniform Memory

Access) may utilize several memory controllers to access the memory.

1. **The memory accessing time for each CPU in UMA is the same. In contrast, the memory accessing time in NUMA varies with the distance of memory from the CPU.**
2. **In terms of bandwidth, the UMA architecture has limited bandwidth. On the other hand, the NUMA has higher bandwidth than UMA.**
3. **One memory controller in UMA and many in NUMA**



17. Registers

MAR : The central processor unit frequently uses the memory address register to read and store any sort of data. The memory address register keeps track of the address so that data can be retrieved quickly. The memory address register is primarily utilised for memory reading and writing operations.

MDR : A memory data register is used to hold data that will be stored or fetched from the computer memory, also known as random-access memory (RAM). The memory data register serves primarily as a buffer, storing everything that can be copied from the computer memory and used by the processor for subsequent tasks.

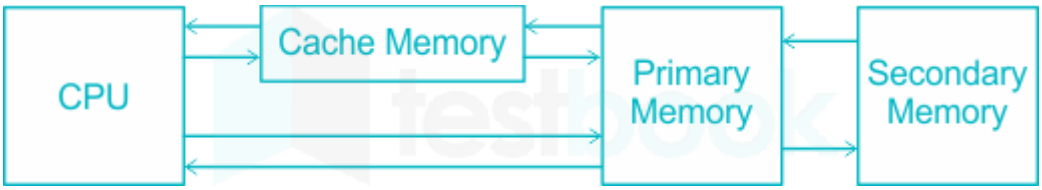
MBR : In addition to the memory data register, there is MBR, which stands for memory buffer register. Data and information that can be read or written into the computer memory are stored in the memory buffer register.

PC : PC stands for programme counter register. The instruction address register (IAR) or instruction counter register (IC) is another term for the programme counter register (instruction counter). The register which stores the address of the next instruction is called the program counter..

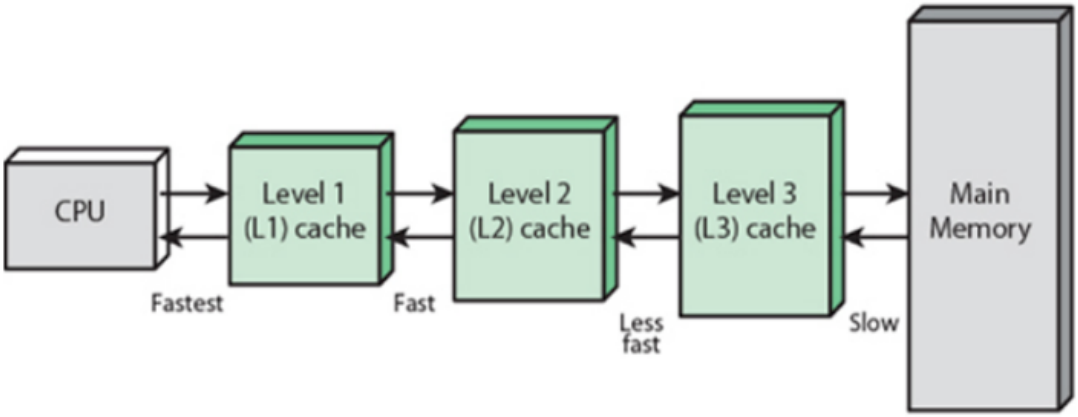
ACCUMULATOR : Another form of central processing unit register that is commonly used for storing logic or intermediate outcomes is the accumulator. The accumulator register is critical. If it is not present, all intermediate results must be kept in the main memory, which increases the memory cost, because then, the number of unnecessary read and write operations will increase.

IR : In computing, the instruction register (IR) or current instruction register (CIR) is the part of a CPU's control unit that holds the instruction currently being executed or decoded.

18. Caches



The cache memory is required to balance the speed mismatch between the main memory and the CPU. The clock of the processor is very fast, while the main memory access time is comparatively slower. Hence, the processing speed depends more on the speed of the main memory.



This is to do with the physical size on the die. Each bit in a cache is held by one or more transistors, so if you want a lot of cache you need a lot of transistors. The speed of the cache is essentially inversely proportional to locality to the unit that wants to access it - in tiny devices such as this, communication gets slower when your signal path gets longer. This is partially to do with substrate impedance, but at this level there are some more complex physics phenomena involved.

If we want to include a large singular cache, it has to be within a very short distance of the MMU, ALU, etc. at the same time. This makes the physical design of the processor quite difficult, as a large cache takes up a lot of space. In order to make the cache "local" to these subunits, you have to sacrifice the

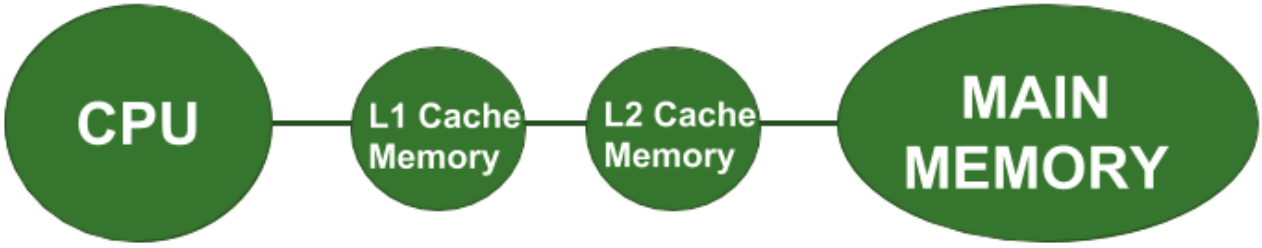
locality of the subunits to one-another. By using a small, fast local cache (L1) we can maximise locality and speed, but we lose size. So we then use a secondary cache (L2) to keep larger amounts of data, with a slight locality (and therefore speed) sacrifice. This gives us the best of both worlds - we can store a lot of data, but still have a very fast local cache for processor subunits to use. In multicore processors, the L3 cache is usually shared between cores. In this type of design, the L1 and L2 caches are built into the die of each core, and the L3 cache sits between the cores. This gives reasonable locality to each core, but also allows for a very large cache. The functionality of caches in modern processors is very complicated, so I'll not even attempt a proper description, but a very simplistic process is that target addresses are looked for in the L1, then the L2, then the L3, before resorting to a system memory fetch. Once that fetch is done, it's pulled back up through the caches.

MultiLevel Caches

Multilevel Caches is one of the techniques to improve Cache Performance by reducing the "*MISS PENALTY*". Miss Penalty refers to the extra time required to bring the data into cache from the Main memory whenever there is a "*miss*" in the cache.

Here the CPU at first checks whether the desired data is present in the Cache Memory or not i.e. whether there is a "*hit*" in cache or "*miss*" in the cache. Suppose there is *3* miss in Cache Memory then the Main Memory will be accessed only 3 times. We can see that here the miss penalty is reduced because the Main Memory is accessed a lesser number of times than that in the previous case. The Cache performance is optimized further by introducing multilevel Caches. As shown in the above figure, we are considering 2 level Cache Design. Suppose there is *3 miss* in the L1 Cache Memory and out of these 3 misses there is *2 miss* in the L2 Cache Memory then the Main Memory will be accessed only 2 times. It is clear that here

the Miss Penalty is reduced considerably than that in the previous case thereby improving the Performance of Cache Memory.



19. Cache-lines

When the processor accesses a part of memory that is not already in the cache it loads a chunk of the memory around the accessed address into the cache, hoping that it will soon be used. The chunks of memory handled by the cache are called cache lines. The size of these chunks is called the cache line size.

20. Hyper-Threading

Hyper-threading, on the other hand, is Intel's version of simultaneous multithreading (SMT). SMT splits each CPU core into two virtual cores (called threads). These two virtual cores are able to process instructions simultaneously (if the program allows for it), meaning that multi-threading effectively doubles the number of cores that the CPU has. So if you have a quad-core CPU with hyperthreading, you have eight virtual cores. Hyperthreading improves CPU performance by doubling the number of processes that the CPU can handle simultaneously. This means that the CPU can handle more demanding applications much easier.

21. Pipe-Lining

Alternatively called instruction pipelining or pipeline, pipelining is an advanced processing technique that handles multiple stages of different computer instructions at once. For example, without pipelining a processor may process the instructions fetch, decode, and execute from three different instructions in three different clock cycles. However, with pipelining the processor could combine the different instructions into one cycle and effectively do three different things in one cycle instead of three.

Hazards : Data Hazard | Resource Hazard | Control Hazard

22. Branch Prediction

Branch prediction is a technique used in CPU design that attempts to guess the outcome of a conditional operation and prepare for the most likely result. A digital circuit that performs this operation is known as a branch predictor. It is an important component of modern CPU architectures, such as the x86. When a conditional operation such as an if…else statement needs to be processed, the branch predictor "speculates" which condition most likely will be met. It then executes the operations required by the most likely result ahead of time. This way, they're already complete if and when the guess was correct. At runtime, if the guess turns out not to be correct, the CPU executes the other branch of operation, incurring a slight delay. But if the guess was correct, speed is significantly increased.

23. AVX

Intel® Advanced Vector Extensions 512 (Intel® AVX-512) is a set of new instructions that can accelerate performance for workloads and usages such as scientific simulations, financial analytics, artificial intelligence (AI)/deep learning, 3D modeling and analysis, image and audio/video processing, cryptography and data-mining.

FMA : Fused Multiply Add

FMADD → r=(x*y)+z

FMSUB → r=(x*y)-z

FNMADD → r=-(x*y)+z

FNMSUB → r=-(x*y)-z

24. Cache Coherence

For higher performance in a multiprocessor system, each processor will usually have its own cache. Cache coherence refers to the problem of keeping the data in these caches consistent. The main problem is dealing with writes by a processor. There are two general strategies for dealing with writes to a cache:

Write-through : all data written to the cache is also written to memory at the same time.

Write-back : During write operation, only the cache location is updated in the write-back method. Then, the location is marked by a flag so that it is later copied to the main memory when the word is removed from the cache. For the write-back method, the reason is that during the time a word remains in the cache, it can be updated multiple times. Thus, as long as the word remains in the cache, it does not matter if the copy in the main cache. This is only when the word is displaced from the cache which needs an exact copy that is rewritten into main memory.

25. USEFUL LINUX Commands

Flags : Flags modify the operation of a command and are sometimes called *options*.

$ lshw

-memory

description: System memory

physical id: 0

size: 15GiB

$ lsblk

```
NAME            MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINTS
nvme0n1         259:0    0 476.9G  0 disk
├─nvme0n1p1     259:1    0     2G  0 part  /boot
├─nvme0n1p2     259:2    0     2G  0 part  /boot/efi
├─nvme0n1p3     259:3    0 143.1G  0 part
│ └─nvme0n1p3_crypt 253:0    0   143G  0 crypt /
└─nvme0n1p4     259:4    0 329.9G  0 part
  └─nvme0n1p4_crypt 253:1    0 329.9G  0 crypt /home
```

$ df -h

```
Filesystem              Size  Used Avail Use% Mounted on
tmpfs                   1.6G  2.2M  1.6G   1% /run
/dev/mapper/nvme0n1p3_crypt  144G   29G  114G  21% /
tmpfs                   7.7G   28K  7.7G   1% /dev/shm
tmpfs                   5.0M  4.0K  5.0M   1% /run/lock
/dev/nvme0n1p1          2.0G  253M  1.6G  14% /boot
/dev/nvme0n1p2          2.0G   46M  2.0G   3% /boot/efi
/dev/mapper/nvme0n1p4_crypt  330G  2.0G  326G   1% /home
tmpfs                   1.6G  172K  1.6G   1% /run/user/618255811
```

$ numactl --hardware

```
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5
node 0 size: 15629 MB
node 0 free: 5716 MB
node distances:
node   0
  0:  10
```

$ lscpu

```
Architecture:          x86_64
  CPU op-mode(s):      32-bit, 64-bit
  Address sizes:       39 bits physical, 48 bits virtual
  Byte Order:          Little Endian
CPU(s):                6
  On-line CPU(s) list: 0-5
Vendor ID:             GenuineIntel
  Model name:          Intel(R) Core(TM) i7-9850H CPU @ 2.60GHz
    CPU family:        6
    Model:             158
    Thread(s) per core: 1
    Core(s) per socket: 6
    Socket(s):         1
    Stepping:          13
    CPU max MHz:       4600.0000
    CPU min MHz:       800.0000
    BogoMIPS:          5199.98
    Flags:             fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mc
                       a cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss
                       ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
                        arch_perfmon pebs bts rep_good nopl xtopology nonstop_
                       tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cp
                       l vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid ss
                       e4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes
                       xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_f
                       ault epb invpcid_single ssbd ibrs ibpb stibp ibrs_enhan
                       ced tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsba
                       se tsc_adjust sgx bmi1 avx2 smep bmi2 erms invpcid mpx
```

Virtualization features:

Virtualization: VT-x

Caches (sum of all):

L1d: 192 KiB (6 instances)

L1i: 192 KiB (6 instances)

L2: 1.5 MiB (6 instances)

L3: 12 MiB (1 instance)

NUMA:

NUMA node(s): 1

NUMA node0 CPU(s): 0-5

Vulnerabilities:

Itlb multihit: KVM: Mitigation: VMX disabled

L1tf: Not affected

Mds: Not affected

Meltdown: Not affected

Mmio stale data: Mitigation; Clear CPU buffers; SMT disabled

Retbleed: Mitigation; Enhanced IBRS

Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp

Spectre v1: Mitigation; usercopy/swapgs barriers and __user pointer sanitization

Spectre v2: Mitigation; Enhanced IBRS, IBPB conditional, RSB fillin g, PBRSB-eIBRS SW sequence

Srbds: Mitigation; Microcode

Tsx async abort: Mitigation; TSX disabled$ numactl --hardware

available: 1 nodes (0)

node 0 cpus: 0 1 2 3 4 5

node 0 size: 15629 MB

node 0 free: 5716 MB