Improving PC performance: The CPU

Clock Speed

The clock speed is the number of fetch-execute-decode cycles the CPU can run per second. It is measured in Hz, MHz or GHz.

The CPU is controlled by an internal 'clock' signal. The clock is an a-stable operation (alternating between logic 1 and logic 0, i.e. on/off) that cycles over a specific time.

Increasing clock speed:

The **faster the clock**, the faster the **fetch-decode-execute or machine cycle** is. This means that the CPU can **calculate and manipulate data in less time** so the computer performance is increased.

Overclocking = running a computer's processor at a higher clock speed **than intended** by the manufacturer.

Compromises to increasing clock speed:

- **Heat**: Every time the **clock ticks power is consumed**. Consequently, **heat** is produced. The more power the more heat. Increasing The clock speed will **increase the cycles per second** so **more power** is used so **more heat** is produced within the chip.

- **Reliability** of the chip: As a higher clock results in a higher machine cycle frequency, more power is consumed so more heat is produced. **The hotter the chip** gets, the **lower the average life** of the chip since **stress on components low down** is **caused by temperature**. A **slight increase in temperature** can have an **exponential decrease in chip life expectancy.**

- **Damage to the chip**: For decades increasing clock speed has meant an increase in performance. Today **much beyond 3.5Ghz** uses too much power and **too much heat** that can cause **the chip to melt damaging** the integrated components. More **efficient coolers are required**.

- **Battery life**: Increasing the clock speed results in the machine cycle running at a higher frequency so there are more ticks which will **increase the power consumption** of the CPU. The CPU runs **less efficiently** in electrical terms as **more heat is produced**. This becomes a big problem with **portable computer systems that rely on a battery** as the battery life is decreased.

  In portable computers manufactures have to make a **compromise** by not using clock speeds too high as this will **degrade the battery life**.

- **Von Neumann Bottleneck**: Generally a higher clock speed will mean that the CPU has to spend more time idle while waiting for the retrieval of data over the busses. Since a single bus can only access one of the two classes of memory at the same time, throughput is lower than the rate at which h the CPU functions. With only one shared data bus for instructions & data, instruction fetches and data transfers are scheduled and don't run simultaneously.

Adaptive clock speeds (dynamic frequency scaling)

To increase the reliability of the CPU and reduce the power consumption, modern chips are designed to decrease the clock speed when handling less strenuous tasks and speed up when it is busier. This is also the case with thermal throttling where the CPU gets too hot the clock speed decreases to release less heat and protect the chip although it slows work rate down.

CPU Cores

Multiple core processor is the term given to a CPU containing two or more independent actual processing units called cores.
Manufacturers integrate multiple cores onto a single integrated circuit what is known as a CMP (chip-multi-processor).

Having multiple cores means that a CPU can undertake multiple tasks simultaneously and so improve the efficiency. The operating system can allocate different applications being run to separate cores provided they are largely independent. This means a given workload can be fetched-executed-decoded in a quicker time period.

Having multiple cores means there are more processors running at the given clock speed within the chip. Whenever each core's clock ticks power is consumed and so power consumption increases. This can prove an issue where battery life is important (mobile computers).

Compromises

The performance increase, however, is not linear since the CPU cores have to communicate through channels. This process is allocated some of the clock speed.
More complicated programs and operating systems are required to be coded to make use of the extra resources of the multiple cores the CPU offers.
Many programs can't keep up with the technologies and so can't make use of vast amounts of cores.

A dual-core CPU is a CPU that has four independent processing cores within the chip.
A quad-core CPU is a CPU that has four independent processing cores within the chip.
An octo-core CPU is a CPU that has eight independent processing cores within the chip.

Cache

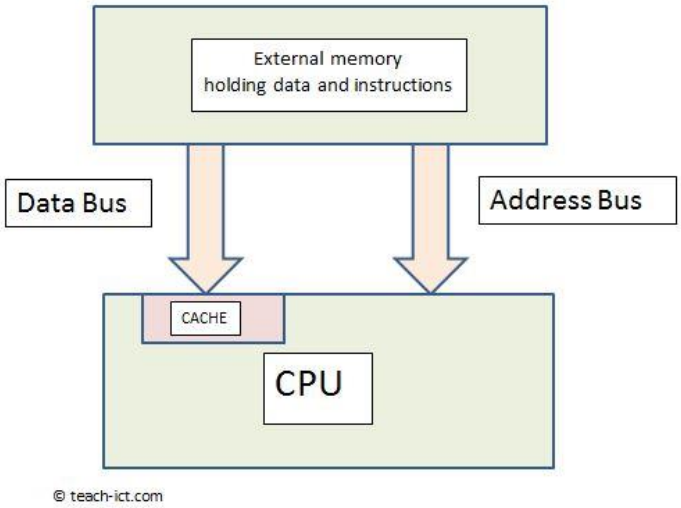
RAM
Both data and instructions of programs are allocated locations in the volatile memory (RAM) external to the CPU. The data and instructions in the memory have to be sent to the CPU via a data bus in order to be handled by the CPU. The transfer rate across this is far slower than the clock speed of the CPU as so the CPU has to remain stationary waiting for the instructions and data. This is known as the von Neumann bottleneck.

Cache is a high-speed memory located in the CPU (or close to it on a separate chip). This allows the instructions and data that needed to be frequently retrieved to be readily stored within the CPU so it does not have to travel across the data bus from external memory.

The advantage of cache memory is that the CPU does not have to use the motherboard's system bus for data transfer. Whenever data must be passed through the system bus, the data transfer speed slows to the motherboard's capability. The CPU can process data much faster by avoiding the bottleneck created by the system bus.

Code Optimisation

Sophisticated programmers can make use of the CPU's cache by coding for instructions that frequently run to be stored in the cache so there is no need to use the slow data bus.



Levels of Cache

Within a CPU there are separate levels of caches. With multiple caches, they can be arranged in a hierarchy depending on speed, with layers closer to the CPU faster than the ones feeding it. Level 1 cache is the fastest however since cache memory is expensive a compromise is made so it is often very little. Higher levels of cache, further from the CPU have more capacity but are slower.
Instructions of a higher priority are stored in the faster lower level caches closer to the CPU.

Modern CPUs unusually have 3 layers of cache. They vary for different CPU manufacturers.

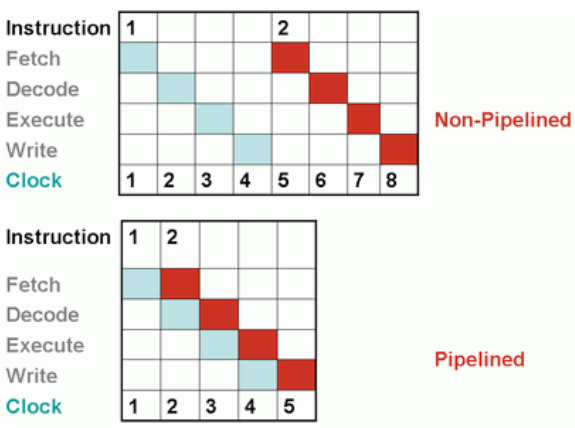
Memory	Size	Latency	Physical Location
L1 cache	32KB	4 cycles	within each core
L2 cache	256KB	11 cycles	beside each core
L3 cache	6 MB	~ 21 cycles	shared with all cores
L4 E-cache	128MB	~ 58 cycles	separate chip
RAM	4GB +	~ 117 cycles	motherboard ram
Swap File	100GB +	~ 10,000 cycles	hard disk

Disadvantages: Very expensive to implement in chips, limited capacity.
Advantages: Faster access speeds, stores temporary data, stores programs that can be executed quickly.

Pipelining

Pipelining is a technique that CPUs can use which allows them to run more efficiently. It allows one instruction to be processed even when a previous one hasn't completely finished it's fetch-decode-execute (machine) cycle.

The resources within the CPU are used more effectively as multiple instructions can be split into different stages of their fetch-execute decode cycle, as one instruction is being e.g. decoded another can be fetched. This means that more instructions can be completed with fewer clock cycles in a given time any clock speed.



We can see on the left how the non-pipelined CPU took more clock cycles and so longer to complete the two instructions as making efficient use of its resources (some of the stages are idle and should be used).

Advantages:

- *The amount of instructions that can be completed simultaneously is increased
- *The delay between completed instructions and proceeding ones is decreased (the throughput).
- *Pipelined CPUs function at higher frequencies than the clock of the main memory.

Disadvantages:

*If a CPU has pipelining, its arithmetic unit can be more complex to manufacture.

- *Pipeling does not decrease the amount of time taken to complete each individual function (i.e. latency)

- *In some cases it can increase the completion time (latency) if there is a branch in an instruction and the pipeline is `flushed' it is cleared of previous instructions. This results in a delay before the other instruction that was being worked on simultaneously can be completed.
- *Pipeliend CPUs are more expensive to design and manufacture.
- *When a coder writes in assembly code and assumes that each instruction is being executed consecutively, hazards can arise as this may not be the case with a pipeline architecture. Hazards can prevent the program working as intended.