Structure and Function of the Processor (1.1.1)

Control Unit (CU):

- Decodes instructions and handles execution
- Sends memory read/write requests to RAM on the control bus
- Coordinates and communicates with all parts of the CPU and the rest of the computer by directing the flow of data

Program Counter (PC): Holds the address of the next instruction to be executed. Memory Address Register (MAR): Holds the address of the data or instruction to be fetched or to which data is to be written. Memory Data Register (MDR): Holds the data which is read from or about to be written to memory. Current Instruction Register (CIR): Holds current instruction being executed, Instruction = opcode + operand. ALU: carries out Arithmetic and logical operations on data; Bitwise shifts left and right. ACC: Stores data and the results of ALU calculations. A CPU may have many GPRs for temporarily storing data while instructions or calculations are being carried out. More GPRs, faster operations.

Address Bus: Carries memory addresses; a one-way bus (MAR to RAM). Data Bus: Carries data; two-way bus (MDR / RAM). Control Bus: Carries command and control signals (CU / rest of CPU and computer)

Fetch

- The address stored in the PC is copied into the MAR.
- It is sent along the address bus to RAM where it waits for a signal from the control bus
- The CU sends a read signal along the control bus to RAM
- The contents of the specified memory location are sent along the data bus to the MDR
- If the MDR now contains data, it is copied into the ACC
- If the MDR now contains instructions, it is copied into the CIR
- The PC is incremented by 1

Decode: The instruction in the CIR is decoded by the decode unit. Execute: Carry out the Instruction.

Clock Speed (in Hz): Number of FDE cycles per second.

Cache Size: Temporarily stores frequently used data and instructions, data that has just been used and data that is about to be used. Much quicker to access than RAM. Therefore, we can access the cache instead of RAM, saving time.

CPU core: A complete copy of a CPU. Each with its own registers, ALU etc. Can execute multiple instructions at the same time. However, doubling the number of cores does not double the performance, since CPU cores must communicate with each other (taking up time). Many programs are not designed for multiple cores.

Pipelining:

- Very efficient way of speeding up the FDE cycle. Works best with multiple cores.
- The next instruction is fetched while another is decoded, while another is executed.
- Otherwise, while the CPU is fetching data, the ALU will be idle, wasting time.

Flushing: The computer may need to branch elsewhere. Once this occurs, the pipe will need to be flushed (restarted). A program with lots of branch instructions may not benefit much from the effects of pipelining. The width of the address bus determines the maximum possible memory capacity of the system.

Von Neumann Architecture:

- Shared memory space for instructions and data (which are stored in the same format)
- Process one instruction/item of data at a time (No parallel processing)
- One set of buses
- Advantages: One set of buses: simpler for the CU to manage, cheaper to manufacture. All memory can be used for either instructions or data. Thus, memory usage is more efficient.

Harvard Architecture:

- Instructions and data are stored in separate memory units

Structure and Function of the Processor (1.1.1)

- Each has its own set of buses
- Reading and writing data can be done at the same time as fetching an instruction
- Advantages: Processes can be carried out in parallel, improving performance.
- Disadvantages: Each has its own set of buses: More complex, more expensive, takes longer to manufacture. Memory can be wasted as free instruction memory cannot be used by the program's data.

Contemporary / Modern architectures:

- SIMD (Single Instruction Multiple Data): Carries out a single instruction on multiple data items at the same time (GPUs)
- MIMD (Multiple Instructions Multiple Data): Carries out multiple instructions on multiple data items at the same time, across multiple cores.
- Distributed Computing: Multiple computers on a shared network each take on part of a large problem

Von Newmann architecture is used for the CPU and the RAM. Harvard architecture is used in the cache.

A Decode Unit contains an instruction set: A list of instructions that the Control Unit can understand and can perform on the data.

CISC	RISC
Each instruction may take multiple FDE cycles to complete a task	Each Instruction requires one FDE cycle to complete a task
Large Instruction Set	Small Instruction Set
Complicated processor design, expensive to design	Simple processor design, cheaper to design
Because it is just one instruction (which may represent several steps), it requires little memory to be stored.	Complex tasks require several instructions, therefore requiring lots of memory to store a program
Instruction format varies	Instruction format is the same

Multicore processors have more than one core. GPUs can carry out multiple jobs simulteaneously (only with software that allows this).

Graphics Processing Unit (GPU): Traditionally used to process large blocks of visual data very quickly. Consists of thousands of cores; can handle multiple processes at once. GPUs use SIMD. Now, many other jobs are taken on by the GPU. It takes on some of the more computer-intensive jobs freeing up more capacity for the CPU.