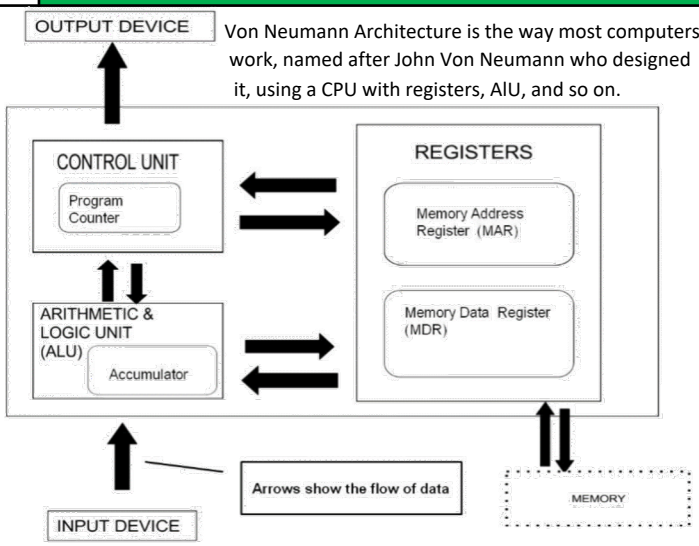
Knowledge Organiser: Yr 10 Computer Science;

Systems Architecture

1 Von Neumann Architecture



Von Neumann Architecture is the way most computers work, named after John Von Neumann who designed it, using a CPU with registers, AlU, and so on.

3 Why Von Neumann?

- CPU runs programs held in main memory (RAM)
- Computer can run multiple applications /programs at the same time.

4 Control Unit (CU)

-Overall control of the CPU.

-Executes instructions by following Fetch-Decode-Execute cycle.

-Controls the flow of data inside CPU (to main memory and input/output devices) ALU.

2 Fetch-Decode-Execute (FDE) Cycle

Fetch
- Copy memory address from the program counter to the MAR
- Copy the instruction stored in the MAR address to the MDR.
- Increment the program counter to point to the address of the next instruction, ready for the next cycle.

Decode
- Instruction in the MDR is decoded by the CU.
- The CU may prepare for the next step (eg load value into the MAR)

Execute
The instruction is performed, e.g.
- Load data from memory
- Write data to memory
- Do a calculation or logic operation (using ALU)
- Change the address in the PC
- Halt the program

5 Arithmetic Logic Unit (ALU)

- Does all the calculations.

- Addition and subtraction (multiplication / division are repeated add and subtracts).

- Performs logic operations AND, OR, NOT and binary shifts.

- Contains the accumulator register.

6 Cache Memory

- Very fast memory in the CPU.

- Slower than registers but faster than RAM

- Stores frequently used data so CPU can access it quickly. CPU checks cache for data first – then checks RAM

- Low capacity (not large) and more expensive than RAM

- Different levels of cache L1, L2, L3 (L1 quickest but lowest capacity)

Leave blank to allow students to glue.

Knowledge Organiser: Yr 10 Computer Science;
Systems Architecture

7 Registers

- Temporarily hold tiny bits of data needed by the CPU.

- Super quick to read/write to (quicker than any other form of memory).

- **Program Counter (PC)** - Keeps track of the memory address for the next instruction. Increments at each new cycle.

- **Accumulator** - Holds results of calculations made by the ALU temporarily.

- **Memory Address Register (MAR)** - Holds any memory address about to be used by the CPU. The address might point to data or an instruction.

- **Memory Data Register (MDR)** - Holds the actual data or instruction. This may have been fetched from memory or waiting to be written to memory.

8 How Common Characteristics of CPUs Affect Performance

Clock Speed

-A faster clock speed allows more instructions carried out (FDE cycles) per second and so instructions are executed more quickly.

Cache Size

- A larger cache gives more space for frequently used instructions.
- Because it is faster to fetch instructions from the cache inside the CPU than from the main memory/RAM outside the CPU, more cache means more instructions can be stored for fast access, meaning faster fetching of repeated instructions and so faster processing.

Number of Cores

- More cores allow more instructions carried out simultaneously.
- More cores allow the processor to process more instructions at the same time.
- Number of instructions (in billions) processed per second = clock speed (in GHz) x number of cores

Example

- For CPU 1, each core runs at 3.2GHz. That means 3.2 billion fetch execute cycles can be carried out per second. It is duel core which means it can process 2 instructions for each cycle, so 3.2 x 2 means 6.4 billion instructions can be executed per second by CPU1

- For CPU 2, each core runs at 2GHz. That means 2 billion fetch execute cycles can be carried out per second. It is quad core which means it can process 4 instructions for each cycle, so 2 x 4 means 8 billion instructions can be executed per second by CPU2

CPU 1	CPU 2
3.2 GHz	2 GHz
Duel Core	Quad Core
2MB Cache	2MB Cache

- In this example CPU 2 would perform better than CPU 1 for most modern software which can use multiple cores.

Leave blank to allow students to glue.

Knowledge Organiser: Yr 10 Computer Science;
Systems Architecture

9	Embedded Systems

This is where the instructions which run a system (software) are combined with the physical hardware of the device, usually by being stored in Read Only Memory (ROM) contained in a chip on a circuit board.

The instructions /software in an embedded system is often called the 'firmware'.

The purposes of embedded systems are usually:

- To run a device which has only one specific, pre-defined function.

- Cheaper than providing a full personal computer system.

- Doesn't include unnecessary features.

- Can be made much smaller than a personal computer system.

- Allows for a device to be automated / programmed.

- Means the user of the device does not need to install the software to use it, as it is already installed in the device's hardware.

A disadvantage of embedded systems is that if the software/instructions need to be updated or improved AFTER the device is sold to the user, some technical knowledge would be required to do this.

Examples of devices which use embedded systems might be:

- Dishwasher
- MP3 player
- Washing machine
- Sat nav
- Digital Camera

10	Other Factors Which Affect How Quickly a PC Runs (Performance)

- Amount of RAM
- Data transfer speed of the secondary storage (HDD/SSD)
- Graphics card capability

Leave blank to allow students to glue.

How do we use Knowledge Organisers in Computer Science?

How can we use knowledge organisers at home to help us?

- **Retrieval Practice**: Read over a section of the knowledge organiser, cover it up and then write down everything you can remember. Repeat until you remember everything.
- **Flash Cards**: Using the Knowledge Organisers to help on one side of a piece of paper write a question, on the other side write an answer. Ask someone to test you by asking a question and seeing if you know the answer.
- **Mind Maps**: Turn the information from the knowledge organiser into a mind map. Then reread the mind map and on a piece of paper half the size try and recreate the key phrases of the mind map from memory.
- **Sketch it**: Draw an image to represent each fact; this can be done in isolation or as part of the mind map/flash card.
- **Teach it**: Teach someone the information on your knowledge organiser, let them ask you questions and see if you know the answers.

How will we use knowledge organisers in Computer Science?

- **Revision:** We will access this knowledge organiser electronically as part of revision homework tasks, using the techniques above to help us revise prior learning.
- **Test:** We will do regular low stakes tests to check our ability to retrieve information from memory.
- **Mark our answers:** Once we have done a low stakes test we can mark our work using the knowledge organiser.
- **Improve our work:** Once we have finished a piece of work we might use this knowledge organiser to see if there is any information on it that we could add into an answer.
- **Pre-reading:** We may sometimes read sections of the knowledge organiser before we cover them in lesson as homework and note down specific questions we may have or specific concepts we are finding more challenging to understand ready to share in lesson.

Date	Section of knowledge organiser	Score