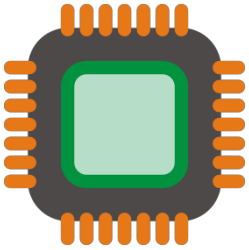
Topic 1.1

Systems Architecture

CPU

- Purpose of the CPU
- How common characteristics of CPUs affect their performance

Specification — Revised



- The purpose of the CPU is to carry out the processing of data on the computer system
- It performs the fetch-decode-execute cycle.
- The CPU fetches, decodes and executes instructions

The Performance of the CPU

Clock speed	Cache Size	Number of Cores
This determines the rate at which instructions are carried out each second The clock speed is measured in Hertz (Hz) A 3.6 Ghz processor carried out 3.6 billion calculations a second	Cache memory is a buffer that sits between the CPU and main memory. The CPU will check here first for instructions that have been fetched before The larger the cache the more space there is for instructions the CPU needs The cache has similar access speeds to the CPU and is therefore quicker to fetch instructions from	A core is an independent processor in the CPU A dual core has 2, quad core 4, hex core 6 processors working simultaneously The higher the number of cores the better performance of the computer as it can multitask

- Clock speed – number of fetch-decode-execute cycles a second
- Cache Size – high speed memory used by the CPU
- No of Cores – number of independent processors in the CPU working together

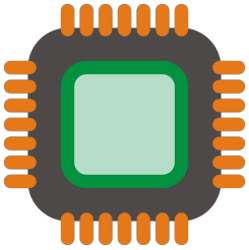
- The CPU contains **registers** which are temporary memory stores within the Cpu which have a specific purpose.

Systems Architecture

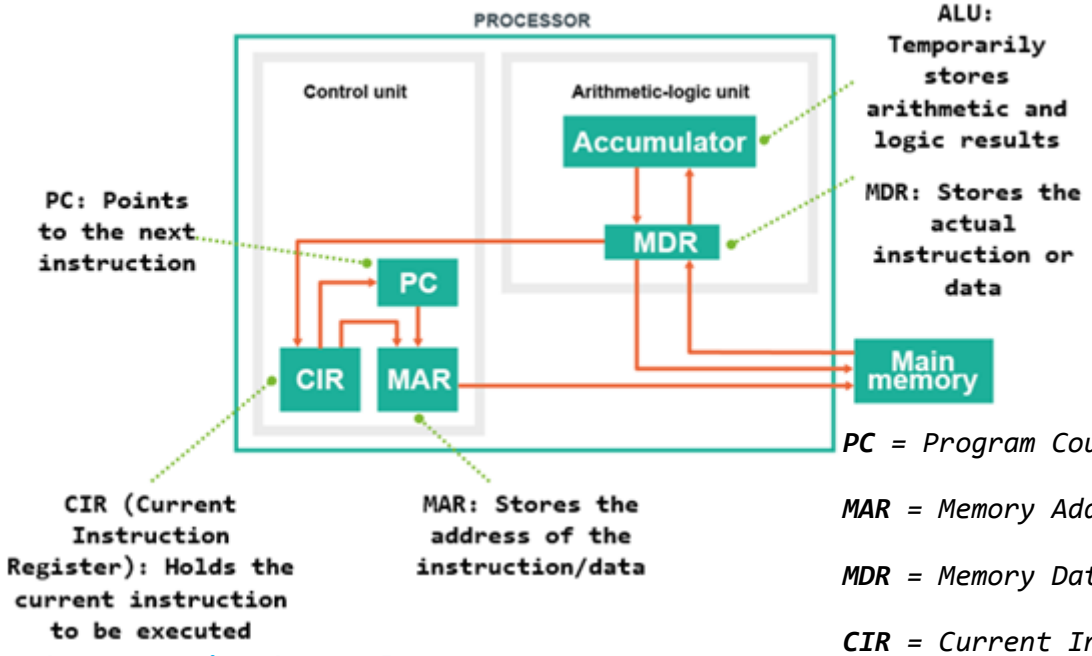
Von Neumann Architecture

- Von Neumann Architecture and the purpose of different registers

Specification — Revised



- The Von Neumann Architecture describes a system where data and programs/instructions are stored in the same main memory location
- The fetch-decode-execute cycle is the process of fetching these instructions from memory and executing them



***PC** = Program Counter*

***MAR** = Memory Address Register*

***MDR** = Memory Data Register*

***CIR** = Current Instruction Register*

The steps in the cycle:

Fetch	Decode	Execute
ⓘ Address from the PC is copied to the MAR ⓘ Instruction from MAR is fetched and copied to MDR ⓘ The instruction in the MDR is copied to the CIR ⓘ Increment the PC	ⓘ The instruction in CIR is decoded by the control unit ⓘ Data may be loaded into the MDR	ⓘ The instruction is performed ⓘ The ALU may be used for any logic or calculations ⓘ The result is stored in the accumulator

Systems Architecture

Embedded Systems

- Purpose of embedded systems
- Examples of embedded systems

Specification — Revised

- An embedded system is one which has a processor built in to another device
- A computer System that is made up of both Hardware and Software often known as Firmware
- Usually for very specialised tasks
- Doesn't usually contain an Operating System

Examples:

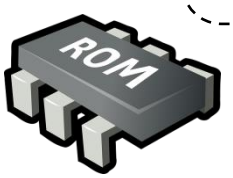
- Dishwasher
- Microwave
- Fridge
- Smart phone
- TV

Topic 1.2

Memory
RAM and ROM

- The difference between RAM and ROM
- The purpose of RAM in a computer system
- The purpose of ROM in a computer system
- The need for Virtual Memory
- Flash Memory

Specification — Revised



RAM and ROM

Random Access Memory (RAM)	Read Only Memory (ROM)
Purpose = Stores data and programs currently being used by the computer	**Purpose** = Stores instructions needed to start up the computer – contains the boot program
ⓘ Can be changed by the computer at any time	ⓘ Programmed during the computers manufacture and cannot normally be changed
ⓘ Volatile memory (data is lost when the power is turned off)	ⓘ Non-Volatile (data is not lost when the power is turned off)
ⓘ Larger memory (Starting at 4GB in most computers)	ⓘ Small (Only MB needed for the boot program)
ⓘ The more RAM the more programs that can be run at the same time. Allows for more multitasking.	ⓘ ROM is needed as it is always there to start the computer

Flash and Virtual Memory

Flash	Virtual
ⓘ Flash memory is type of non-volatile (ROM) memory that can be changed and does not need a power supply to keep its contents ⓘ There are no moving parts which make it fast and reliable ⓘ Examples of flash memory in use: ⓘ Memory cards in digital cameras. ⓘ Mini/Micro SD cards in Smartphones. ⓘ USB memory sticks. ⓘ Solid state drives	ⓘ Virtual memory is part of the hard drive used as an extension to RAM. If there is not enough RAM to hold all the data and run the programs needed then it will make use of some of the hard drive. ⓘ Access speeds from the hard drive are slower than from RAM. More RAM reduces the need for virtual memory which improves performance.

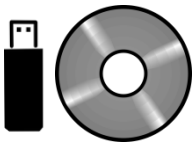
Topic 1.3

Storage

Secondary Storage

- The need for secondary storage
- Data capacity and calculation of data capacity requirements
- Common types of storage: Optical, Magnetic, Solid State

Specification — Revised



ⓘ Secondary storage is needed to store our files and programs when they are not in use. These commonly fall into three categories: Optical, Magnetic and Solid State.

ⓘ Secondary storage is long term, non-volatile storage

Optical	Magnetic	Solid State/Flash
ⓘ Lasers write data to the surface of a disk. ⓘ Optical media includes: CD, DVD, Blu-Ray ⓘ Excellent for distributing software ⓘ Good capacity ⓘ Low cost ⓘ Light and portable ⓘ Can get damaged over time ⓘ Slow access speed	ⓘ The magnetic tape is moved along a read-write head inside a disk drive ⓘ Examples: hard disk and tapes ⓘ Used for backups ⓘ High capacity ⓘ Cheap ⓘ Reliable ⓘ Slow to read due to moving parts	ⓘ No moving parts make solid state memory have a very fast access speeds. Most are a type of flash memory ⓘ Examples: USB drives, memory cards, solid state hard drives ⓘ Large capacity but less than magnetic tape ⓘ More expensive ⓘ Portable ⓘ Reliable and not affected by being moved around

Calculating Storage Requirements

ⓘ From the section on data representation we know how many bits are in a byte and how many bytes in a kilobyte etc. We also looked at how to calculate the size of an image.

ⓘ Using the knowledge we can calculate how much storage will be needed in different scenarios. For example:

A text file that contains 10000 characters. Give your answer in KB

We know that each character is 1 byte in ASCII. So 10000 x 1 = 10000. There are 1024 bytes in a kilobyte so 10000/1024 = 9.77kb

Storage

Secondary Storage

- Suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics:
  - Capacity
  - speed
  - portability
  - durability
  - reliability
  - cost

Specification — Revised

ⓘ When choosing an appropriate device for a given scenario you need to consider the following characteristics:

- **Capacity**: How much space there is to store files.
- **Access Speed**: How quickly the computer can read and write data to or from a storage device or write data to it.
- **Portability**: Can you easily unplug it and carry it away?
- **Durability**: How easily is it damaged? Will it survive being dropped?
- **Reliability**: Can the data always be accessed and be correct?
- **Cost**: How expensive is the storage device

Quick Comparison:

Access Speed

Slowest → Fastest: *Optical Disk* | *Magnetic Tape* | *HDD* | *USB/Flash* | *SDD*

Cost

Cheap → Expensive: *Magnetic Tape* | *Optical Disk* | *HDD* | *USB/Flash* | *SDD*

Capacity

Small → Large: *Optical Disk* | *USB/Flash* | *SDD* | *HDD* | *Magnetic Tape*

Portability

Difficult → Easy: *SDD* | *HDD* | *Magnetic Tape* | *Optical Disk* | *USB/Flash*

Durability/Reliability

Low → High: *Optical Disk* | *Magnetic Tape* | *USB/Flash* | *SDD* | *HDD*

Topic 2.4

Computational Logic

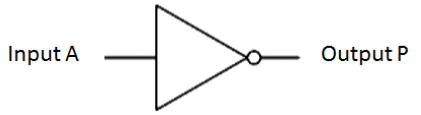
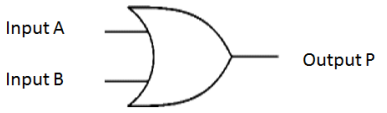
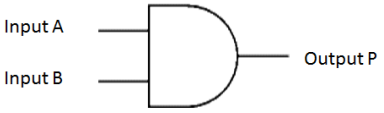
Logic Gates

- Why data is represented in a computer system in binary form
- Simple logic diagrams using the operations AND, OR and NOT
- Truth Table
- Combining Boolean operators using AND, OR and NOT to two levels
- Applying logical operators in appropriate truth tables to solve problems

Specification — Revised

- A binary number system means that only two digits can be used. These two digits are 0 and 1.
- Using only 0 and 1 makes it easier to design the electronic circuits that the computers will use.
- Memory and circuits in a computer are made by wiring millions of transistors together. They can make simple logic calculations such as are both inputs a 1. These simple circuits are called logic gates.

There are 3 main types of gates:

NOT	OR	AND
		
The NOT gate is a very simple gate – if 0 is input the it outputs 1 and if 1 is input then it outputs 0.	The OR gate tells us if one or both of the two inputs are 1 by outputting 1, otherwise output 0	The AND gate tells us if both inputs are 1, by outputting 1, otherwise it outputs 0.

Truth Tables

- We express this relationship between inputs and output as a truth table.
- We use A,B, C…for the inputs and P, Q, R … as outputs.
- Below are the truth tables for the NOT, AND and OR gates

NOT

A	P
0	1
1	0

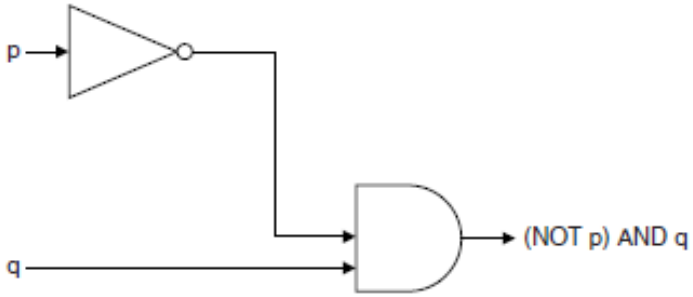
OR

A	B	P
0	0	0
0	1	1
1	0	1
1	1	1

AND

A	B	P
0	0	0
0	1	0
1	0	0
1	1	1

Gates can be combined together to form more complex gates. For example:



Here. We have joined together a NOT gate and an AND gate. The completed truth table would be:

p	q	(NOT p) AND q
0	0	0
1	0	0
0	1	1
1	1	0

You need to first work out the input p which is NOT p. This is in brackets. You do the brackets first

You then combine the result of this with q and show the result of going through the AND gate

Applying Computing Related Mathematics:

Operator	Purpose
+	Addition
-	Subtraction
/	Division
*	Multiplication
Exponentiation(^)	To the power of. E.g. 10^2
MOD	Gives the remainder after division. E.g. 5 MOD 2 = 1
DIV	Gives the integer result of the division. E.g. 5 DIV 2 = 2