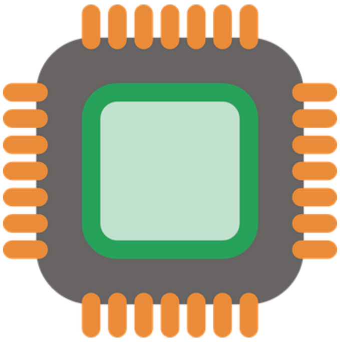
Topic 7 Fundamentals of Computer Organisation and Architecture

The Processor



The processor executes program instructions to run applications. The processor may also be called the CPU (Central Processing Unit).

Arithmetic Logic Unit (ALU)
As the name suggests, the ALU performs logic (such as AND or XOR) and arithmetic (such as addition) operations.

Control Unit (CU)
The control unit controls the different processor components and runs the fetch-execute cycle.

Registers
Registers are small, fast storage locations which hold data for a short time. General purpose registers can hold any data which is required, but there are also special pupil registers:

- PC - Program Counter - Holds the memory address of the next instruction in the fetch-execute cycle.
- CIR - Current Instruction Register - Holds the instruction which is currently being executed.
- MAR - Memory Address Register - Stores the memory address of the location in memory where data should be read from or written to.
- MBR - Memory Buffer Register - Also known as the MDR (Memory Data Register) it holds the contents of a memory location which has been read from or needs to be stored.
- SR - Status Register - Contains a number of bits which are changed to indicate an interrupt.

Clock
Every processor contains a clock, which generates a timing signal at regular frequencies. This signal synchronises communications between the processor components and the rest of the system.

Main Memory

The memory contains program instructions and frequently used data; it is much faster than secondary storage allowing instructions to be quickly executed. Main memory can be RAM (random access memory) or ROM (read only memory).

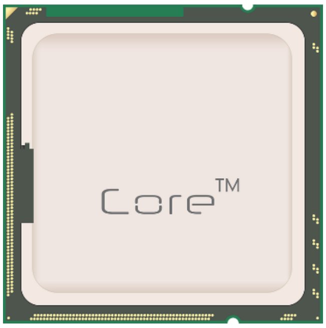
Busses

A bus is a series of parallel wires connecting the internal components of a computer system, allowing signals to be passed between the components. The number of wires within the bus is called its width, which affects the amount of data which can be transmitted at a time.

Address Bus
This bus transports memory addresses, which tell the system where in memory data should be sent or retrieved from. Increasing the width of the bus allows it to specify more addresses, and in doing so increases the amount of addressable memory. Each wire added doubles the number of addressable memory locations.

Data Bus

The data bus transfers data and instructions between different components of the system. Increasing the bus width allows it to transfer more instructions at once.



Control Bus

This bus carries the computer's clock signal, along with control signals which are used to regulate the system.

I/O Controllers

Input / Output (I/O) controllers control communication between the processor and external devices such as a monitor or keyboard.

The Stored Program Concept

Early computers could run one specific program. The stored program concept allows one set of instructions to be easily changed for another, allowing computers to run different applications.

The official definition of the concept is "machine code instructions stored in main memory are fetched and executed serially by a processor that performs arithmetic and logical operations".

- Machine code instructions - code formed from 1s and 0s which the processor can directly execute.
- Main memory - ROM or RAM where instructions and frequently used data are stored.
- Fetched - instructions are retrieved from main memory.
- Executed - the instructions are carried out by the processor.
- Serially - instructions are fetched and executed in a specific order.
- Arithmetic - mathematical operations such as addition and subtraction.
- Logical - operations involving logic gates such as AND and NOT.

System Architectures

Harvard Architecture

Processors using the Harvard Architecture use separate locations in memory for instructions and data, allowing each location to be given different characteristics, such as making instructions read only. This architecture is widely used in embedded systems.

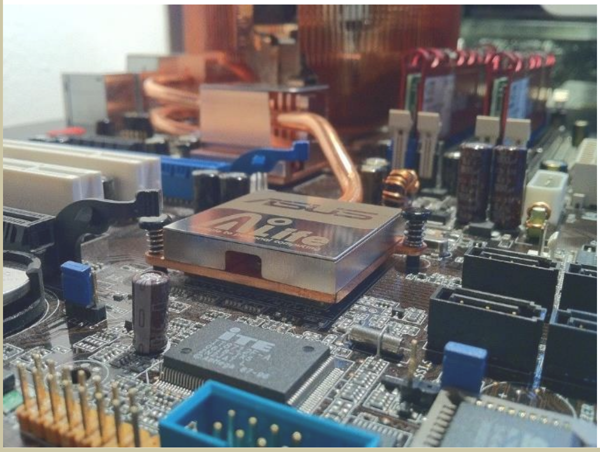
Von Neumann Architecture

Here, instructions and data are stored together in the same memory location. This leads to poorer performance than with Harvard architecture because busses have to be shared when fetching data and instructions. This architecture is used for general purpose computers such as laptops and desktops.

The Fetch-Execute Cycle

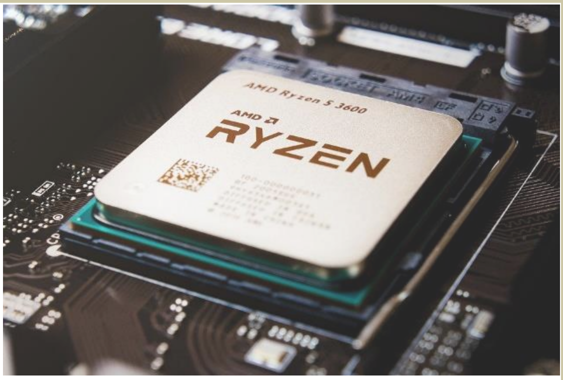
This cycle is continuously performed by the processor and consists of three stages.

Fetch



This stage fetches the next instruction to be executed from main memory.

1. The content of the PC is copied to the MAR
2. The content of the MAR is transferred to main memory by the address bus
3. The instruction is sent from main memory to the MBR by the data bus
4. The PC is incremented by one
5. The content of the MBR is copied to the CIR



Decode

This stage decodes the instruction which has just been fetched.

1. The content of the CIR is decoded by the control unit.
2. The decoded instruction is split into two parts, opcode and operands.

Execute

This stage executes the instruction.

1. Any data which the instruction needs but is not present is fetched.
2. The instruction is carried out.
3. The results of any calculations are stored in main memory or a general purpose register.

Check for Interrupts

Between each stage of the cycle, the content of the status register is checked for any changes which would signal an interrupt.

Interrupts

An interrupt is a signal sent from a component to the processor requesting attention, for example, an I/O controller may need to inform the processor that a key has been pressed on the keyboard. Interrupts can also be sent by software, such as when an unexpected error has occurred.

Interrupts are detected as a change in the content of the status register, which is checked between the execute and fetch stages of the fetch-execute cycle. Interrupts are handled using the vectored input method. When an interrupt is detected, the processor stops executing the current program and places the content of its registers into the system stack. This is known as saving the volatile environment.

With the progress saved to the system stack, the processor loads a specific series of instructions appropriate to the type of interrupt, known as an interrupt service routine. When the processor has finished executing the interrupt service routine, it restores the volatile environment from the system stack and resumes executing the program it was running before the interrupt.

The Processor Instruction Set

The instruction set is the group of instructions which a processor can carry out, and each type of processor has its own set of instructions. This means that instructions for one processor may not be compatible with another.

These instructions are usually stored in machine code, consisting of opcode along with one or more operands.

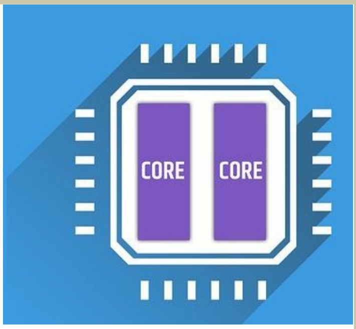
Opcode is the type of operation which should be executed, while operands are the pieces of data on which the opcode should be performed.

Addressing Modes

A bit in machine code instructions can be assigned to either immediate or direct addressing mode.

Immediate addressing treats the value in operand as the actual value, while direct addressing treats the operand as an address in memory and executes the opcode on the data stored at that address.

Factors Affecting Processor Performance



Number of Cores

Each processor core performs its own fetch-execute cycle independently, allowing different applications to be allocated to different cores. This means that the number of cores directly affects processor performance, with more cores allowing more instructions to be executed at the same time, giving faster performance. It is common for processors to be quad core (four cores) or octa core (eight cores).

Cache Memory

Cache memory is incredibly fast and built directly into the processor. It is used to store information, which is frequently used by the processor, meaning it does not need to be fetched from main memory over and over again. The more cache memory a processor has, the more information it can store, and therefore the less time is spent fetching instructions from main memory, resulting in better performance.

Word Length

A word is a group of bits treated by the processor as a single unit, words are used to represent both data and instructions. The length of the word is the number of bits assigned to it. Higher word lengths allow more bits to be transferred and manipulated in one go, leading to better performance.

Address Bus Width

Increasing the width of the address bus increases the range of addresses it can specify, and with that increases the addressable memory, increasing performance.

Data Bus Width

Increasing the width of the data bus increases the volume of data which can be transferred across the bus at any one time. The wider the bus is, the more data can be fetched from main memory during one cycle of the fetch-execute cycle. This reduces the number of cycles needed to fetch data and improves performance.

Logical Shifts

A logical shift shifts all the bits in a binary number a set number of positions to the left or right. This has the effect of doubling or halving the number. Shifting one digit left doubles the number and shifting one digit to the right halves the number.

In the example below, we perform a logical shift by three places on the number 10110101, which has the effect of doubling the number 3 times.

			1	0	1	1	0	1	0	1
1	0	1	1	0	1	0	1	0	0	0

AQA Assembly Language

AQA use a specific assembly language in their examinations, which you need to be able to write and interpret.

Operation	AQA Assembly Language	Description
Load	LDR Rx, <memory reference>	Load the value stored in the memory location specified by <memory reference> into register x.
Store	STR Rx, <memory reference>	Store the value that is in register x into the memory location specified by <memory reference> .
Add	ADD Rx, Ry, <operand>	Add the value specified by <operand> to register y and store the result in register x
Subtract	SUB Rx, Ry, <operand>	Subtract the value specified by <operand> from the value in register y and store the result in register x.
Move	MOV Rx, <operand>	Copy the value specified by <operand> into register x.
Compare	CMP Rx, <operand>	Compare the value stored in register x with the value specified by <operand> .
Branch unconditional	B <label>	Always branch to the instruction at the position specified by <label> in the program.
Branch conditional	B<condition> <label>	Branch to the instruction at position <label> if the last comparison met the criterion specified by <condition> . Possible values for <condition> are: ● EQ : equal to ● NE : not equal to ● GT : greater than ● LT : less than
Logical AND	AND Rx, Ry, <operand>	Perform a bitwise logical AND operation between the value in register y and the value specified by <operand> and store the result in register x.
Logical OR	ORR Rx, Ry, <operand>	Perform a bitwise logical OR operation between the value in register y and the value specified by <operand> and store the result in register x.
Logical XOR	EOR Rx, Ry, <operand>	Perform a bitwise logical XOR (exclusive or) operation between the value in register y and the value specified by <operand> and store the result in register x.
Logical NOT	MVN Rx, <operand>	Perform a bitwise logical NOT operation on the value specified by <operand> and store the result in register x.
Logical shift left	LSL Rx, Ry, <operand>	Logically shift left the value stored in register y by the number of bits specified by <operand> and store the result in register x.
Logical shift right	LSR Rx, Ry, <operand>	Logically shift right the value stored in register y by the number of bits specified by <operand> and store the result in register x.
Halt	HALT	Stops the execution of the program

Input and Output Devices

Barcodes and Barcode Readers

Barcodes are printed diagrams made from light and dark areas and are read using a barcode reader.

2D barcodes can contain more information in the same space as 1D barcodes but need more processing in order to read and extract the information.

Barcodes readers use a laser light source, lens, photodiodes and a mirror to read the barcode. The mirror directs the laser's light onto the barcode. The reflected light then passes through the lens onto the photodiode, which turns it into electrical charges. The charge is measured to form a digital signal which represents the data held in the barcode. Lighter portions of the barcode reflect more light than darker sections, and this is represented as a pattern of 1s and 0s.

Barcodes may use error detection and prevention such as parity bits or check digits to tell whether the barcode has been read correctly. If the barcode fails to scan correctly, the reader continues to scan until the barcode is correctly read.

1D barcodes: **2D barcodes:**

12345678

Digital Cameras

Digital cameras use a lens to focus light onto a sensor. The path of the light from the lens to the sensor is controlled by a sensor.

The most common types of sensors used are CMOS (complementary metal oxide semiconductor) and CCD (Charge Coupled Device). Both convert light into an electrical charge. The charge builds up in cells within the sensor, with each cell representing a pixel. Once the photograph has been taken, the charge in each cell is measured and converted into a digital value, which is processed and stored as a digital image.

Colour cameras use multiple cells for each pixel, each of which uses a filter to allow only certain wavelengths of light. This allows the camera to build up a separate image for each colour of light, with these separate images being combined to form a full photograph.

A base filter is a special colour filter which has the same number of green filters as it does red and blue combined. This produces an image which is closer to what the human eye would see, since human eyes are more sensitive to green light.

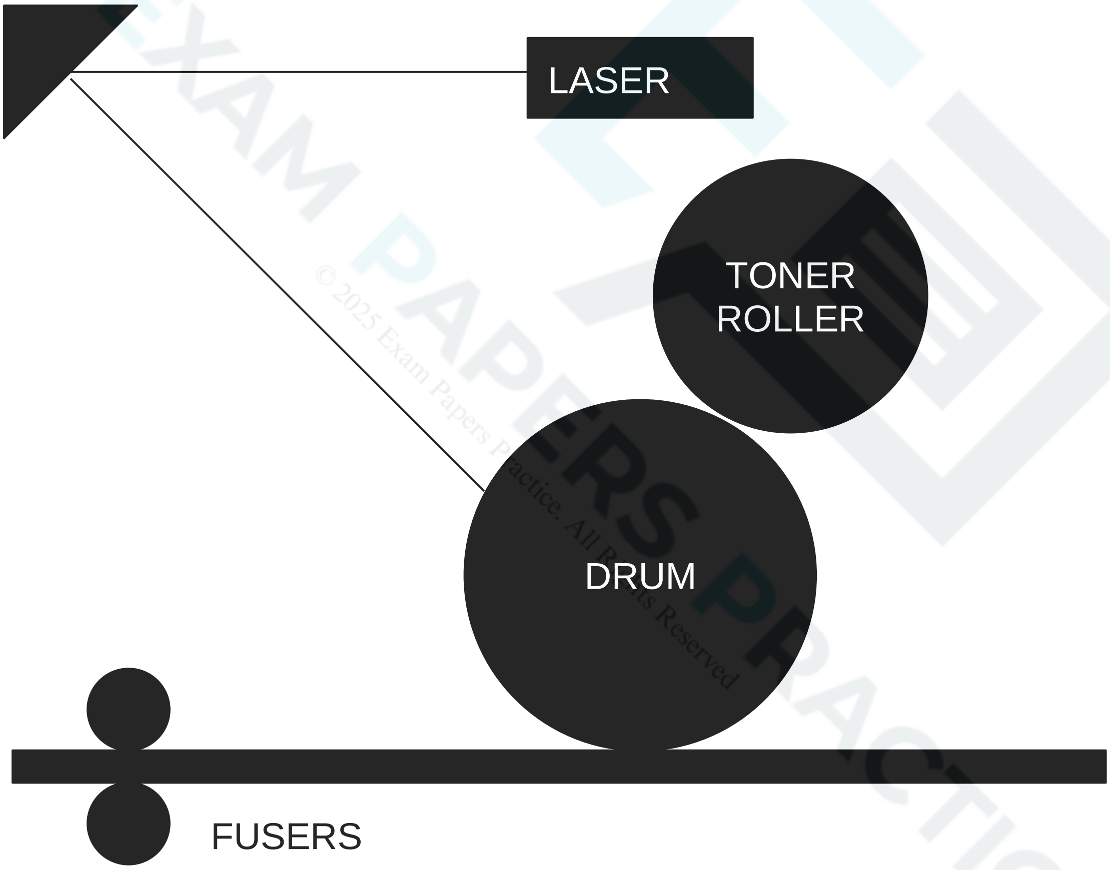
Laser Printers

Laser printers produce a printed image on paper from a digital signal.

To print a document:

1. The drum is positively charged.
2. The laser is used to “draw” an image of the page onto the drum in negative charge.
3. The toner is negatively charged and dispensed by the toner roller. The toner sticks to the positively charged areas of the drum.
4. The paper rolls across the drum, transferring the toner to it.
5. The fuser is used to heat the toner, fixing it onto the page.

Black and white laser printers run through this process once using black toner, whilst colour printers use four different toner colours, cyan, yellow, magenta and black (CYMK) to give colour prints.



RFID

Radio Frequency Identification (RFID) allows information to be wirelessly transmitted between a “tag” and a reader. It is used in hotel room cards, and with contactless credit and debit cards. A chip inside the tag contains a small amount of memory and is attached to a wire coil which functions as an antenna.

Most tags are passive, meaning they rely on induced power from a reader to operate the chip. However, some are active, which contain a small battery and power supply. These active tags have a much greater range.

To read a tag:
1. The reader emits radio waves which are picked up by the antenna in the tag.
2. The power induced from these waves powers the chip.
3. The chip uses its own antenna to emit a radio signal in reply.
4. This signal contains the information held in the chip and is picked up by the reader.
5. The reader decodes the information and returns it to the computer.

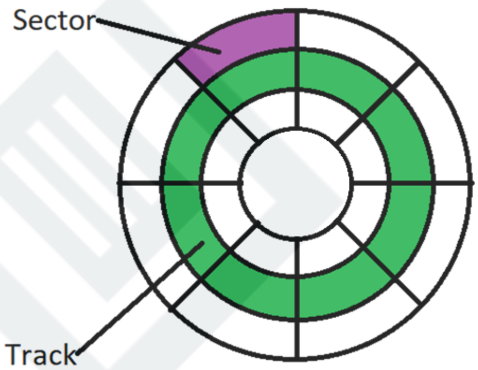
Secondary Storage

Computers use main memory such as RAM and ROM as their primary storage, however, this storage is expensive and in the case of RAM is lost whenever the computer is turned off. Secondary storage allows files and applications to be cheaply and conveniently stored whether the computer is turned on or off.

Hard Disk Drives (HDDs)

Hard disk drives are made up of one or more circular magnetic platters. Each platter has an arm hovering above it, on which is mounted the read/write head. The arm moves the head to the area of the disk which needs to be read or written, and the head changes the polarity to write data or reads it to read data.

Each disk is divided into rings called tracks and each track is divided into sectors. The platter spins thousands of times a minute, allowing good read and write speeds. Hard disk drives are usually available in capacities of 500mb to 5TB. The more platters the drive has, and the narrower the tracks are, the more data it can hold.



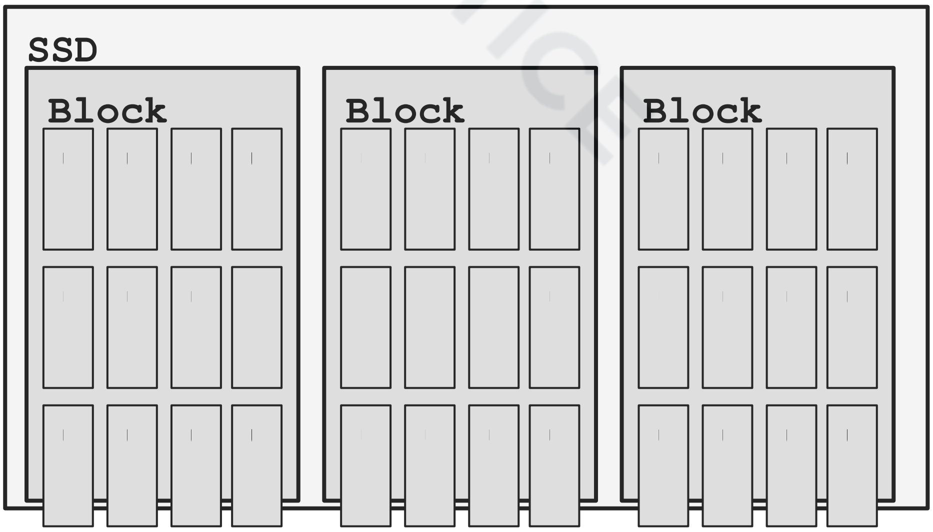
The high number of moving parts in these drives means they can be easily damaged due to movement, making them not well suited for use in portable devices.

Solid State Drives (SSDs)

SSDs are made of NAND flash memory cells and a controller. They are non volatile (meaning data is not lost when power is not applied). Each memory cell is made of floating gate transistors, which trap electrical charge to store information.

Data on SSDs is stored in pages, which are combined into blocks.

Data on a page cannot be overwritten, and the controller must completely erase the page before new data can be written. SSDs have far higher read write times than HDDs and have no moving parts, making them more robust.



Optical Disks

CDs, DVDs and Blu-Rays are all optical disks. They can be read only, recordable or rewritable and store data which can be read optically by a laser.

The disk's plastic surface contains pits and lands. Pits are dips burnt into the surface, whilst lands are the remaining “high” areas. Optical disks have one continuous track in a spiral from the centre of the disk outwards.

A laser is used to read the disk, with the light being reflected onto a photodiode. When the beam passes over a land the light is reflected. When it hits a pit, the light is scattered rather than being reflected back. This is used to form a digital signal of 0s and 1s.

On recordable and rewritable disks, an opaque dye is applied to the disk’s surface rather than permanently deforming the disk’s surface. This allows the dye pattern to be changed by a high powered laser multiple times but be unaffected by the low powered laser used to read the disk.

	Hard Disk Drive	Solid State Drive	Optical Disk
Read/Write Speeds	Good	High	Low
Latency	High	Low	High
Capacity	High	Low	Very Low
Portability	Low	High	High (although surface can be easily damaged)
Power Consumption	High	Low	High