

Computer Science 478 Notes '24

★ Data Representation

1.1 Number Systems

Analogue data is a continuous stream of data that is processed by humans.

Digital data is discrete data that only uses values 0 and 1 (aka binary).

Denary is a base-10 number system that uses values 0-9.

- Computers CANNOT process hexadecimal or denary numbers, so they must be converted to binary.

Hexadecimal is a base-16 number system that uses values 0-9 and characters A-F.

Denary - 0, 1-9, 10, 11, 12, 13, 14, 15

Hexadecimal - 0, 1-9, A, B, C, D, E, F

-> Hexadecimal is a more concise number system, it's easier to write, read, and understand. Debugging and locating are also easier with Hexadecimal.

Converting Binary to Denary

- To calculate a binary number like 10101000, place it in columns of base 2 numbers
- Add from left to right, writing 1 if the base-2 number is less than or equal to your denary number
- Then add all the base 2 numbers

128	64	32	16	8	4	2	1
1	0	1	0	1	0	0	0

$$128 + 32 + 8 + 1 = 168$$

Converting Denary to Binary

- To calculate a denary number like 84, set up the columns of base 2 numbers.

128	64	32	16	8	4	2	1
0	1	0	1	0	1	0	0

01010100

Converting Binary to Hexadecimal

- Separate into two nibbles (4 bits)
- Convert nibbles to denary
- Change denary numbers to hex if needed (e.g. 10 = A)

8	4	2	1
0	1	1	0

=6

8	4	2	1
0	0	0	1

=1

Hexadecimal value is 61

Binary addition rules

- > $0 + 0 = 0$
- > $1 + 0 = 1$
- > $1 + 1 = 10$ (10 is 2 in binary)
- > $1 + 1 + 1 = 11$ (11 in binary is 3)

What is an **overflow error**? It is a type of error that occurs when a number larger than a register can store is generated.

Logical binary shifts

- << A logical shift to the left multiplies the number by 2
- >> A logical binary shift to the right divides the number by 2

- A problem with logical shifts is that data can lose its integrity as vital bits can be removed from the register.

Negative binary numbers (Two's complement)

* The leftmost bit represents the sign of the number (0 for positive and 1 for negative)

E.x. -43

-> Start by representing 43 in 8-bit binary form

128	64	32	16	8	4	2	1
0	0	1	0	1	0	1	1

-> Invert all the bits

1	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---

-> Add 1

1	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---

1.2 Text, Sound and Images

Character sets contain all the characters that are in the character set and the binary value that is assigned to each character.

ASCII? American standard code for information interchange

Metadata? Additional data that is stored with an image can provide information such as the dimensions of the image and the time and date the image was taken.

Text is converted to binary using a character set such as ASCII or Unicode.

Images are converted to binary by taking each pixel and providing it with a colour code that is converted to a binary value. The larger the resolution and the colour depth of an image, the greater the amount of data that will need to be stored. Increases the quality of the image.

Sound is converted to binary by sampling the sound at a set time interval. The value of each sound sample is converted to binary. The greater the sample rate and resolution of the sound, the greater the amount of data that will need to be stored in the sound file. This also increases the accuracy of the recording.

1.3 Data storage and compression

Unit	Description
Bit	Smallest measurement
Nibble	4 bits in a nibble
Byte	8 bits in a byte
KibiByte {KiB}	1024 bytes in a KiB
MebiByte {MiB}	1024 KiB in a MiB
GigiByte {GiB}	1024 MiB in a GiB
TebiByte {TiB}	1024 GiB in a TiB
PebiByte {PiB}	1024 TiB in a PiB
ExbiByte {EiB}	1024 PiB in an Eib

Calculating the size of a file

- Number of images
- Resolution of the image
- Colour depth of the image

-> Width of image x Height of image x Colour depth of image x No. of files in the image

Calculating the size of a sound file

- Length of audio
- Sample rate of the audio
- Sample resolution of the audio

-> Sample rate x Sample resolution x Length of soundtrack

Data compression

-> Less storage space needed to store files, takes less time to transmit the files, download and upload the file and less bandwidth is needed to transmit the file over the internet

Lossy	Lossless
A compression method that reduces the size of a file by permanently removing data.	A compression method that reduces the size of a file by permanently removing data.
Used for images and sound files.	Finds repeating data and groups it. Used in text , images and sound files .
For images, it reduces colour depth and resolution. For sound files, it removes sounds that cannot be heard by a human and reduces sample rate/resolution.	Data is no longer stored individually, instead, it is stored in lookup tables (p.28)
Perpetual Music Shaping is a process that is used in lossy compression that removes sounds that are not normally heard by the human ear.	Run Length Encoding (RLE): An algorithm that groups repeating patterns and indexes them.

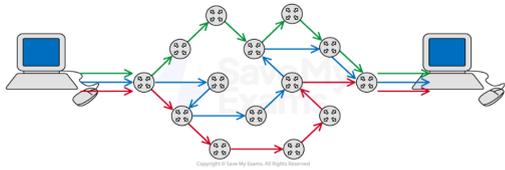
★ Data transmission

2.1 Types and methods of data transmission

Packets are small units of data. Each packet contains three different sections:

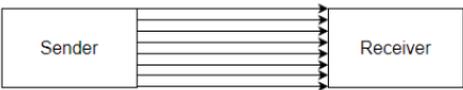
Packet header	Section of the packet of data that contains information about the contents of the packet and its destination. -> Destination address, packet number and originator address (IP)
Payload	The actual data that the user is sending to the receiver.
Trailer	Section of the packet of data that contains information about any error-checking methods that may be used.

Packet switching is a method of transmitting data across a network. Each data packet can take an individual pathway across the network.



-> A router is a network component that examines a data packet to obtain its destination address and then forwards the packet to this address.

Methods of data transmission

Method	Explanation	Example
Serial	<p>A transmission method where data is sent one bit at a time, down a single wire.</p> <p>-> Data arrives sequentially, is less likely to have errors and is cheap as the cost of wire is inexpensive.</p> <p>-> Transmission is slow, serial transmission over very long distances is expensive as the cost of wire dramatically increases.</p>	<p>Connecting an external hard drive to a computer.</p> 
Parallel	<p>A transmission method where data is sent multiple bits at a time, down multiple wires.</p> <p>-> Parallel transmission is fast as large quantities of data are transmitted</p> <p>-> Expensive as multiple wires are needed, delays can be caused as data may arrive asynchronously.</p>	<p>Transmitting data from a computer to a printer using a multi-wire connector.</p> 
Simplex	<p>A transmission method where data is transmitted in a single direction only.</p> <p>-> Cheap</p> <p>-> Slow and requires two sets of wires for bidirectional transmission, increasing the cost</p>	<p>Transmitting data from a computer to a printer.</p>
Half-duplex	<p>A transmission method where data is transmitted in both directions, but only one direction at a time.</p> <p>-> Cheaper than simplex for bidirectional transmission.</p> <p>-> Transmission is slow as data only travels one bit at a time, in one direction</p>	<p>Phone conversations where only one person needs to speak at a time (like a walkie-talkie)</p>
Full-duplex	<p>A transmission method where data is transmitted in both directions at the same time.</p> <p>-> Faster, as data is transmitted at the same time, both directions</p>	<p>Phone conversations where both people can talk and be heard at the same time allowing them to interrupt each other.</p>

	-> Expensive as the wire technology is harder to implement	
--	--	--

USB (universal serial bus) is an industry standard used to transmit data.

- USB ports are sockets that are a part of a device or computer that enables you to insert a USB cable.
- USB cable is a type of transmission media that uses the USB port to transfer data.
- USB connections are a collective name for using a USB cable plugged into a USB port to transfer data from one device to another.
- USB devices plug into a USB port on a computer.

Advantages	Disadvantages
Simple interface = fewer errors Very high speed allowing data to be transferred fast The universal standard for connecting different devices Can be used to power the device Automatically detected	The length of the cable is limited to 5 meters Transmission speeds for USBs are high but not as high as connections like Ethernet.

2.2 Methods of error detection

[Check digit](#)

[Parity check, checksum, echo](#)

[ARQ](#)

In the process of transmitting data, interference can occur. This causes data to be lost, gained, or changed.

Parity check

A type of error detection that adds an additional bit to each byte to create an odd or even sum.

- Every byte transmitted has one of its bits allocated as a **parity bit**
- The sender and receiver must agree before transmission whether they are using odd or even parity
 - If odd parity is used then there must be an odd number of 1's in the byte, including the parity bit

Odd parity bit	Byte
1	0 0 1 0 0 0 1

- If even parity is used then there must be an even number of 1's in the byte, including the parity bit

The value of the parity bit is determined by counting the number of 1's in the byte, including the parity bit. **If the number of 1's does not match the agreed parity then an error has occurred** (won't reveal where the error is).

Checksum

A type of error detection that performs a calculation on the data to create a checksum value. Checksum values are compared after transmission to see if they match.

- A value is calculated from the data that will be transmitted (e.g modulus 11)
- It is added to the data to be transmitted.
- After transmission, the receiving device uses the same method to calculate a value from the received data. If the values match, there are no errors. If they **don't match**, an error has occurred.

Echocheck

A type of error detection that sends a copy of the transmitted data back to the sender to be compared with the original data sent.

- Echo checks involve transmitting the received data back to the sender. The **sender then checks the data to see if any errors occurred during transmission.**
- This method isn't reliable as an error could have occurred when the sender transmits the data or when the receiver transmits the data. Neither will know when the error occurred.
- If an error does occur the sender will retransmit the data

Check digit

A type of error detection that is used for data entry. A calculation is performed on the data entered to create a value. Check digit values are compared to see if the data entered is correct.

- Data is sent in blocks and an additional check digit value is added at the end of the block
- Check Digits are custom user-created algorithms that perform mathematical calculations on data

An example of a check digit is the ISBN value on books:

- Each book has a unique ISBN that identifies the book
- A standard ISBN may be ten digits, for example, 965-448-765-9
- The **check digit value is the final digit** (9 in this example). This number is chosen specifically so that when the algorithm is completed the result is a whole number (an integer) with no remainder parts

- A check digit algorithm is performed on the ISBN. If the result is a whole number then the ISBN is valid

ARQ (Automatic Repeat Request)

A type of error detection that uses acknowledgement and timeout to see if data has arrived correctly after transmission.

When the receiver receives transmitted data it must check for the presence of any errors. **Errors can usually be detected but not always pinpointed.**

- If an error is detected the receiver sends a **negative acknowledgement** transmission to indicate the data is corrupted.
- If no error is detected the receiver sends a **positive acknowledgement** transmission meaning the data is correct.
- If the receiver does not send any acknowledgement transmission then the sender waits for a certain time period known as a **time-out** before automatically resending the data.

This process is repeated until all data has been received and acknowledged.

2.3 Encryption

Encryption is a method of securing data for storage or transmission that scrambles it and makes it meaningless.

-> Encryption does not stop hackers from stealing data that is transmitted, it only makes the data **meaningless**.

Symmetric	Asymmetric
<p><i>Symmetric encryption uses the same key to encrypt and decrypt</i></p> <ul style="list-style-type: none"> - Plain text is encrypted into cipher text using an encryption key. - The cipher text and key are sent separately to the receiving device. - The same key is used then to decrypt the cipher text back into its plain text form. <p>The simplest method of encryption is used for low-profile data.</p> <p>-> Many people were worried that it was far too easy for hackers to intercept both packets and get access to the key and cipher, hence this method of encryption is not as widely used.</p>	<p><i>Asymmetric encryption uses two different keys to encrypt and decrypt data.</i></p> <ul style="list-style-type: none"> - Plain text is encrypted into cipher text using the public key. - The cipher text is transmitted to the receiving device - The cipher text is decrypted using the private key (cannot be decrypted using public key) <p>When we use this algorithm, we will have both a private key and a public key. Anyone who wants to send you encrypted data will use your public key. This encrypted data can only be decrypted using your private key.</p> <p>-> Can work in reverse as well</p>

★ Hardware

3.1 Computer architecture

Von Neumann architecture

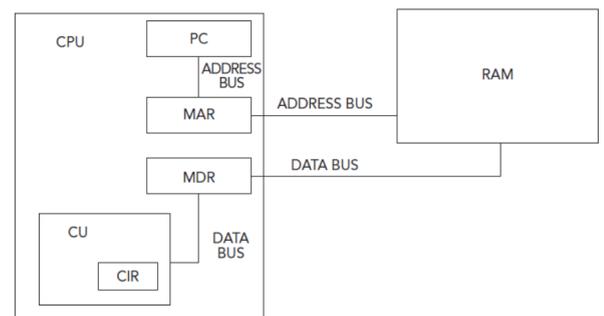
The **CPU** is a component in a computer system that processes data and instructions. A microprocessor is an integrated circuit that can perform many of the functions of a CPU.

Fetch - Decode - Execute

The cycle through which data and instructions are processed.

1. Fetch

- This is the stage where the instruction is fetched into the CPU. The memory address of the instruction to be fetched is stored in the Memory Address Register and is sent down the address bus. The data/instruction at the memory address is transferred back to the CPU, via the data bus, where it is stored in the Memory Data Register. The instruction is copied into the Current Instruction Register and the Program Counter increments.



2. Decode

- This is the stage where the instruction are decoded within the CPU. This begins when the instructions are received by the CU. The control unit is the component of the CPU that controls all the operations in the CPU. To decode an instruction the CU uses an instruction set. This is a set of all the commands that are understood and can be processed by the CPU. They are in machine code.

3. Execute

- This is the stage where the instruction is executed by the CPU. The instruction is executed by the Arithmetic Logic Unit and the opcode is performed upon the operand. The result is stored in the accumulator or written to a memory location within memory.

Special purpose register	Definition
Program Counter (PC)	Stores the address of the next instruction to be fetched from memory Memory Address Registers (MAR)
Memory Address Registers (MAR)	Stores the address of the instruction or data to be fetched from or written to memory

Memory Data Register (MDR)	Stores the data that has been fetched from memory or being written to memory Data from MDR is sent to ALU to be executed
Current Instruction Register (CIR)	Stores the instruction the CPU is currently decoding or executing
Accumulator	Temporarily stores the results of the calculations performed by the arithmetic and logic unit
Arithmetic Logic Unit (ALU)	Performs all mathematical and logical questions when processing

[RAM -> CPU -{PC -> MAR}> RAM -{MDR -> CIR}>] [CU] -> [{ALU -> ACC} -> Execute]
 Address bus Data bus Control bus

Performance factors

There are three main factors that can affect the performance of a CPU and this is the number of cores it contains, the clock speed and the size of the cache.

The **core** is the part of the CPU that contains all the components that are used to perform the fetch-decode-execute cycle. **Clock speed** is the number of FDE cycles that can be performed in a second. Lastly, **cache** is a type of storage that is built into the CPU to store the most frequently used data and instruction. The more cores, the higher clock speed, and the bigger cache; the better performance.

Speed is measured in Hz = cycles per second (cycle = time between two)

3.2 Input and output devices

An **input device** is any device that allows data to be entered into a computer system.

- Cameras, keyboards, QR code scanners.

Human data entry devices - require data to be entered manually into the computer (using keyboards, etc.)

Source entry data devices - these devices enter the source directly into the computer (audio, video, scanner, etc.).

An **output device** is any device that allows the result of the data processing to be seen or heard. A device that can be used to obtain the results of data that has been processed.

- Printers, LED screens, speakers

Hard copy devices - outputted in tangible form can be used without a computer.

Soft copy devices - intangible output allows corrections to be made, stored and sent digitally.

A sensor is a type of input device that is used to capture data from its immediate environment.

An automated system is a system designed to operate without the need for any human intervention.

3.3 Data storage

Primary storage - a type of storage that is directly accessed by the CPU

RAM (random access memory)	ROM (read-only memory)
It is volatile/temporary storage. This means that it requires power to store data. If the power is turned off, the contents of the RAM are lost.	It is non-volatile/permanent storage. This means that if power is removed, the contents of the ROM are retained.
Stores data and programs that are currently in use by the computer.	Stores programs that initially boot the computer, such as the bootstrap and the BIOS.
Contents are constantly being changed as data is processed by the CPU.	Contents are fixed and do not normally change. It is possible to reprogram, but not normally attempted and is complex.
Possible to increase the amount of RAM a computer has by adding further RAM components.	It is not easy to alter the ROM available in a computer system.

Secondary storage - a type of storage that is designed to store data permanently, until it is deleted by the user.

[Save my exams further explanation](#)

- Magnetic storage uses the properties of magnetism to store data.
 - Circular plates that spin allowing data to be read. Magnetised -> 1, demagnetised -> 0
- Optical storage uses lasers to store data, the lasers read and write data (CDs and DVDs), and pits are burned into it and arranged in a spiral form)
- Solid-state storage uses transistors to store data. Uses logic gates to control flow of electrons.

Virtual Memory is a part of secondary storage which acts as RAM when RAM is full. Data that needs to be accessed by the CPU directly is switched by the operating system from virtual memory into RAM and an inactive page from RAM replaces it in virtual memory. This is called page switching.

- Segmentation is a second approach that uses segments not in use in memory. These are moved to virtual memory space on the hard drive.

Cloud storage is the concept of storing data on remote servers, that can be accessed via the internet, instead of on a local storage device. Data is owned by a third party and accessed by the user using the internet.

- No hardware costs, less responsibility, travel friendly and inc/dec the amount of storage with ease.

3.4 Network Hardware

NIC(Network Interface Card)? The component in a computer that is used to connect it to a network.

-> **MAC address?** Media Access Control is a unique address given to the NI by the manufacturer.

Transmission media? cables used to transfer data in a network.

Static IP

An IP address that does not change every time the device connects to a network.

Dynamic IP

An IP address that can change each time the device connects to a network.

★ Software

4.1 Types of software and interrupts

System software is a type of software that manages the hardware and the software in a computer. The **operating system(OS)** is a program that manages the computer, allows the user to communicate with the computer and allows software to be run.

Utility programs are system software that performs housekeeping activities.

Application software is software that allows the user to perform a useful task. This includes things like **word processors, spreadsheets, databases, and web browsers.**

Application software runs on OS which runs on firmware loaded by bootstrap.

The **OS** has many uses, this includes, providing an interface, managing files, managing peripherals and drivers, managing memory, managing multitasking, managing interrupts, providing a platform for running applications and managing user accounts.

An interrupt is a signal sent to the processor to tell it that its attention is required.

Software	Hardware
Division by 0	Data input, i.e. key pressed on a keyboard, mouse click.
Two processes attempting to access the same location in memory	Error from hardware, i.e. printer out of paper

Program request for input	Hardware failure
Output required	Hard drive signal that it has read data
Data required from memory	A new hardware device connected

Interrupt handler (IH) is a program that organises interrupts into an order based on priorities. An **Interrupt Service Routine (ISR)** is a program that retrieves an interrupt and performs the required actions(s).

4.2 Types of programming language, translators and IDEs

High-level languages are programming languages that use human-language-style words. Programs written with high-level languages can be run on different types and manufacturers of computers, such as **Python, Java, and VB.net**.

Low-level languages are closer to the machine's language. This could be assembly language or binary code. **Machine/binary code** is 0s and 1s. This means it is non-portable and cannot be run on different types of computers. **Assembly code** is written in mnemonics that allow direct manipulation of the hardware. It still has to be converted to binary to run.

High-level language	Low-level language
Easier for users to understand, read, write, amend.	More difficult for users to understand, read, write, and amend.
Easier for users to debug code	Harder for users to debug code
Portable, machine-independent	Not portable, machine-dependent
It must be converted to low-level language before it can be run.	It doesn't need converting so it is faster.
One statement can represent many low-level instructions	Several instructions are required for each high-level language statement
Cannot directly manipulate hardware	Can manipulate hardware, making programs more efficient in terms of speed and memory usage.

A **translator** is a type of software that converts code written in one programming language into another, usually a high-level language into a low-level language.

-> **Assemblers** convert assembly language into machine code

-> **Interpreters** are a type of translator that converts a high-level language into a low-level language, by checking one line of code and then executing it before checking the next.

-> A **compiler** is a translator. It checks all code before running the program.

Interpreter	Compiler
Translated one line of code into machine code and runs it	Translates all lines of code before running the program

Reports a syntax error as soon as it's found, and stops the program.	Reports all syntax errors at the same time
Useful when writing the program	Useful when a program is finished and ready for testing
Code must be reprogrammed each time it is translated	Code doesn't need to be re-translated each time the program is run
Doesn't produce an exe file	Produces an exe file
Source code is required	Source code not required
Interpreter software required	No other software required
Can test part of a program without completing the code	Need to finish a section of code before testing it

An **IDE** is a piece of software that allows a user to write, test, and run program code. It includes an editor which has features such as **auto-completion**, **auto-correction**, **pretty print** (helps spot keywords) and **block minimising**. IDEs also include run-time environments which allow a program to be run and lets the user interact with the program. Debugging features include **breakpoints**, **variable watch windows** and **stepping**.

★ The Internet and its uses

(this entire topic is mostly physical notes bc I'm lazy and this shit is ez)

5.1 The Internet and the World Wide Web

The **Internet** is a very large global network that allows users to access the World Wide Web. The **World Wide Web** is a collection of all the web pages that have been made available.

(in-depth notes in notebook bc I understood this topic through programming an actual website, so I don't want to re-cover all that lol)

5.2 Digital currency

Cryptocurrency is a currency that exists electronically rather than physically.

Blockchain is a method that is used to track all the transactions made with crypto. A

digital ledger is a database that is a list of all the transactions recorded by the use of blockchain.

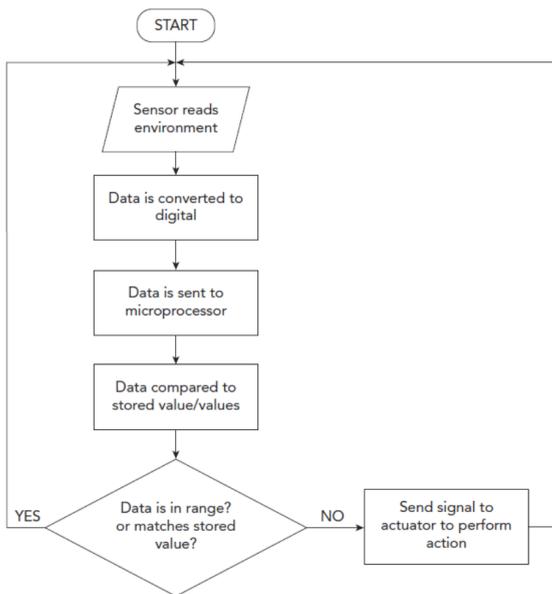
5.3 Cyber security

p106-122

[cyber security threats](#)

[keeping data safe](#)

★ Automated and emerging technologies



6.1 Automated systems

An **automated system** performs actions without interaction with humans. They are used in small systems such as the heating or air-conditioning in buildings, all the way up to flying aeroplanes and building items such as cars.

They usually include the following features-

Sensors (used to capture data from its immediate environment), **Microprocessors** (to process data from the sensors and determine whether action needs to be taken), and **Actuators** (to create movement).

6.2 Robotics

Advantages	Disadvantages
Increased productivity: robots can work continuously without breaks or fatigue	High initial investment: purchasing and setting up robots can be expensive
Consistency and accuracy: robots can perform tasks with a high level of precision and accuracy	Maintenance costs: robots require regular maintenance and repairs, which can add to the overall cost.
Safety: robots can perform tasks that are dangerous for humans	Lack of flexibility: robots are programmed for specific tasks and cannot adapt easily to new situations or tasks.
Cost-effective: robots can help reduce labour costs in the long run	Unemployment: increased use of robots can lead to job loss in certain industries.

6.3 Artificial Intelligence

Artificial Intelligence is a branch of computer science that involves creating computer systems that can perform tasks that would normally require human intelligence.

Things like image recognition, speech recognition, natural language, computer games and diagnosis systems.

(physical notes for components of AI, expert systems and machine learning :))