

The IGCSE Computer Science Revision Guide

Eason Qin (eason@ezntek.com)

February 2025

Contents

Introduction	4
What is this guide?	4
License Notice	5
Important Information	6
Notes	7
1 Data Representation	8
1.1 Converting between Binary, Denary and Hexadecimal	8
1.2 Applications of Hexadecimal	8
2 Data Transmission	10
2.1 Types of Data Transmission	10
2.2 USB	11
2.2.1 USB-A	11
2.2.2 Benefits and Drawbacks of USB-C	12
2.2.3 USB-C	12
2.2.4 Benefits and Drawbacks of USB-C	13
2.3 IP and MAC addresses	14
2.3.1 IP addresses	14
2.3.2 IPv4 Address Assignment	16
2.3.3 MAC addresses	17
2.3.4 Differences between IP and MAC addresses	18
2.3.5 NATs (Network Address Translation)	19

2.4 Packet Switching	20
2.4.1 Components of a packet	20
2.4.2 Packet Switching	21
2.5 Error Correction	22
2.5.1 Parity Blocks	22
2.5.2 Check Digits	23
2.5.3 Automatic Repeat Requests	25
2.6 Encryption	25
3 Hardware	28
3.1 Data Storage	28
3.1.1 RAM	28
3.1.2 ROM	31
3.1.3 Virtual Memory	32
3.1.4 HDDs (Hard disk drives)	32
3.1.5 SSDs (Solid state drives)	32
3.1.6 USB Mass Storage (Flash Drives)	32
3.1.7 Optical Media	32
3.1.8 QR Codes	32
3.2 The Von Neumann Architecture	32
3.2.1 The CPU	32
3.2.2 Components of a CPU	33
3.2.3 Buses	33
3.2.4 The Fetch-Decode-Execute Cycle	33
3.2.5 The Clock Speed	33
3.2.6 Increasing CPU speed	33
3.2.7 Cores	33
3.3 Input and Output Devices	34
3.4 Network Hardware	34

4 Software	35
4.1 The Operating System and Kernel	35
4.2 System Startup and Interrupts	35
4.3 Userspace Software	35
4.4 Programming Languages and Development Environments	36
4.5 Translators, Compilers and Interpreters	36

Introduction

What is this guide?

You are looking at the final IGCSE Computer Science revision guide (also referred to as the CSRG), this time covering the whole syllabus for the IGCSE mock and final examinations. The first ever CSRG was for the Comp Sci G1 semester 2 examinations at OFS, and the second was for the Comp Sci G2 semester 1 examinations at OFS.

This document aims to cover everything you need to know for the final IGCSE Computer Science 0478 examinations, for the 2023 2025 batch of IGCSE CS students. It aims to deliver the content in a concise yet informative form, short but with enough explanation to help develop an understanding for the content. If highlighting the guide helps you, you may do so.

This document is also prepared in \LaTeX , a high-quality typesetting system that is code-based. It is the de-facto standard for the communication and publication of scientific documents.¹

This is revision one of the guide.

NOTE: All references to "I", "Me", "Myself" and similar refer to the main author, Eason Qin.

For a quick tutorial as to how to use this guide, read [placeholder]

¹Taken from the \LaTeX website.

License Notice

The whole work, along with all code produced by all contributors and I are licensed under the Creative Commons Attribution-ShareAlike-NonCommercial (CC BY-SA-NC) 4.0 International License.

This means that you may do the following:

1. You must attribute me, i.e. state that the work was produced by us, the creators if you use it as a part of your work or teachings, or expand upon my work.
2. You may use the guide for any purpose, you may use it to teach yourself or teach others, whatever you like.
3. You may share the guide with anybody else with no restrictions.
4. If you want to create derivative works², you are allowed to do so, as long as if you put the exact same license on it. If it is not written in the text, it will be implied. If you would like the document in its raw, editable form, you may ask me.
5. You may then share it however you please. You can then add yourself to the authors list.
6. You must not make money off of it. Failure to comply means that I, and all other contributors may take any legal action on you if needed.

²Works based on this one.

Important Information

This is mostly targeted to certain individuals who carelessly read this guide while cramming.

1. The chapter and section markers ***will not correspond to the textbook directly!*** Some of the content is deemed too trivial to write about; for those sections I urge you to use your textbook.
- 2.

Notes

1. This revision text is authored by Eason Qin Luoja, with contributors listed on the cover page; including and not limited to Siddharth Harish and Karthik Sankar.
2. Some exercises for the Chapter Ten content on Logic Gates may be pulled from the textbook³, but some are also generated by the authors.
3. **Formal Citations and a bibliography are not provided**, as this is not an academic research document, but a reference booklet of notes from the IGCSE Computer Science 0478 course offered at my school, along with content from the textbook (as mentioned previously). Since the work is mostly produced from either directly pulling examples from the textbook (which will be annotated) or already synthesized information, no references for those points will be provided. If there is information that *must* be cited, including and not limited to extremely detailed data points, the source will be provided as a foot note. **In no case will MLA, APA, Harvard or any other form of formal academic referencing be used.**⁴
4. If there is an underlined portion of text, like so:
Chatbots have mostly been replaced by LLMs, simply go to the AI section below
is seen, and you have the **printed copy**, simply go to the section it says; do so via the .

³Cambridge IGCSE™ and O Level Computer Science, Second Edition, ISBN 9781398318281

⁴Legally, all licenses will be followed; i.e. if the document has a license that requires attribution, the attribution will be provided, etc.

Chapter 1

Data Representation

1.1 Converting between Binary, Denary and Hexadecimal

stub

1.2 Applications of Hexadecimal

There are several applications of the Hexadecimal number system in Computers, which include and are not limited to:

- Making binary easier to write/represent
- OS error codes
- MAC addresses
- IPv6 addresses
- Color codes

Making binary easier to write/represent

Given long binary sequences, such as **1011101010111110** Programmers may find it much easier to simply express the given example as **BABE**, which is 4 characters and not 16 when typing/writing. Converting between the two formats was covered in [this section](#).

OS error codes

stub

MAC addresses

stub

IPv6 addresses

stub

Color codes

stub

Chapter 2

Data Transmission

2.1 Types of Data Transmission

There exists several modes of data transmission:

- **Serial Transmission:** This is the process of sending data through one single channel, one bit at a time. It is slow, but reliable for small amounts of data. This is used by USB, the kind of connector that is used for flash drives, most modern peripherals like mice and keyboards, and other devices.
All generations of USB¹ that make use of the USB-A connector use serial exclusively.
- **Parallel Transmission:** This is the process of sending multiple bits at once, thorough multiple channels concurrently². This is faster, but it requires more computational resources, and can be prone to errors if one lane is congested³. The data may also become skewed⁴, and may cause unwanted issues.
- **Simplex:** This is when data can only be sent one way.
- **Half Duplex (HD):** This is when one device can send data at one time, like a walkie-talkie. The other device must wait until the data is fully sent; only then can they respond. This method is used in some radio communication protocols.
- **Full Duplex (FD):** This is when data can be sent both ways at the same time. This is like a phone call or video chat, where both people can talk simultaneously. This is used in video conferences, phone chats and real-time web applications.

¹There are multiple generations of USB, please check this section for more information.

²At the same time.

³Thought problem: If parallel uses many data channels to send 4 bits, and the second channel is clogged somehow, will the data send, and will there be an error? The solution is trivial and explains this property.

⁴Arrive out-of-order.

2.2 USB

USB is a protocol and interface by which data is transmitted between computers and computer peripherals. It can be seen in many places, like between your computer and a mouse, a cable to connect your phone to the computer, USB microphones, audio devices, hubs, and other possible peripherals.

2.2.1 USB-A

Usually referred to as just USB or USB Type-A apart from the name USB-A, the cross-section of the connector typically looks like a black rectangle with an empty section with a silver coating around it, with 2 holes. Inside the cable, there exists 4 wires:

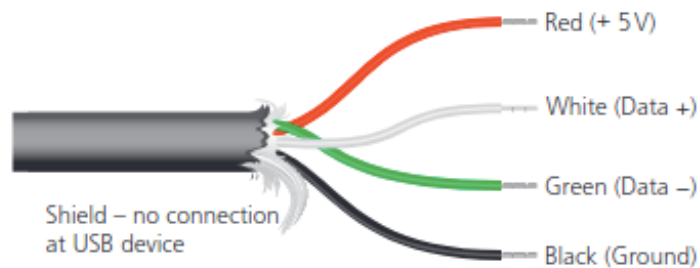


Figure 2.1: The inside of a USB cable, taken from the textbook

Figure 2.1 shows the 4 wires in a USB-A cable:

- A wire carrying **+5 volts**, usually in red,
- **Ground**, or the negative terminal usually in black
- **Green and white**, for serial communication (data- and data+).

And the corresponding port and connector looks like so:



Figure 2.2: The connector and port of USB-A

Since this connector is by far the most common connector used in computers, you may very likely have one on your device. Simply check the sides and or back for an example⁵.

⁵Unfortunately, if you possess a newer Apple computer, Apple took away the USB-A ports to make

2.2.2 Benefits and Drawbacks of USB-C

Note that one only needs to know a few advantages and disadvantages; some are quite specific.

Benefits include:

- USB-A is very common; seen on most computers and devices
- The connector itself is more durable due to its thickness
- Since the connector only fits in one way, incorrect connections cannot be made
- The USB protocol makes use of error correction⁶⁷
- If one needs more ports, they can choose to connect a *USB Hub* which splits one USB port into many, similar to an Ethernet switch)

Drawbacks include:

- Standard USB-A only supports cables up to 5 meters, requiring extensions for cables to go further.
- The connector is one way, making it difficult to insert without looking at the port.
- Very early USB standards, like USB1 may not be supported by the latest computers.
- By far the most common USB-A standard, USB2, found in the vast majority of devices from 2000 till now, only supports 480Mbps/s, meaning that ethernet adapters and other applications that may require higher speed data transfer cannot be done.

2.2.3 USB-C

USB-C, USB Type-C, or just Type-C is the most common USB connector on newer computing devices; all Apple devices after ~2016 have USB-C, and computers between around ~2016 til ~2020 have *only* USB-C ports.

This connection is called USB, however the connector looks quite different from standard USB-A:

As seen in Figure 2.3, USB-C takes on more of a rounded shape, smaller than USB-A and is a *Symmetrical Connector*, meaning it can be inserted both ways.

you purchase adapters. Look for another device or an adapter for an example in that case

⁶As it should be.

⁷If any errors are found, the data is to be re-transmitted.

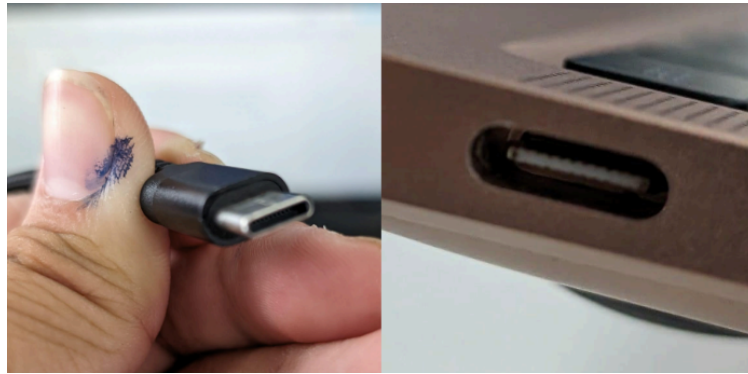


Figure 2.3: The connector and port of USB-C

Quirks

- USB-C can carry *A video signal*⁸, meaning external monitors and TVs can be connected.
- USB-C supports a standard called USB-C PD (Power Delivery). The textbook⁹ states USB-C can deliver a 20 volt 5 amp (therefore 100W) power signal to charge devices at a high wattage.
- USB-C supports extremely high data transfer rates. The textbook states that it can deliver up to 40 gigabits to second, however the latest standard¹⁰ supports up to 120 gigabits per second asymmetrically and 80Gbit/s symmetrically.
- USB-C is fully backwards compatible¹¹ with USB-A, one just needs a simple adapter with the correct lanes connected to make use of this feature.
- the USB-C connector is also used by Thunderbolt, which is a way to connect PCIe Express devices through USB¹². Professional audio devices or graphics cards can be connected through this port.

2.2.4 Benefits and Drawbacks of USB-C

Note that one only needs to know a few advantages and disadvantages; some are quite specific.

Benefits include:

- USB-C can carry more types of signals, such as a video signal, power delivery, and Thunderbolt.
- USB-C can deliver much faster speeds for high volume data transfers, like between film cameras and computers.

⁸Through the DisplayPort standard.

⁹Since USB-C standards change relatively rapidly, the textbook version should be better for the exams as that is in the syllabus

¹⁰USB4 Gen 2

¹¹This generally means that a newer thing fully supports all the features of the older thing

¹²This is very niche and may not come up in tests, but it is good to know

- It can be inserted into a port both ways; it makes connecting devices in less accessible locations more convenient.
- It supports far more ports and devices, like DisplayPort, through adapters.

Drawbacks include:

- It is relatively rare on older devices; only around 2016-2018 did computer manufacturers put USB-C ports on their devices. A lot of them, even to this day do not support data transfer through the port; only power delivery.
- Since USB-C has so many standards (Video, USB, PCI through Thunderbolt, different levels of Power Delivery and even no-data cables), it is confusing as some ports are incapable of delivering thunderbolt, while others are incapable of delivering video, etc.
- USB-A has colors (White/Black and Blue for USB2 and 3 respectively) to determine the generation; and for USB type B the connectors are physically different to determine the generation. USB-C does not have any way of differentiating, so one must memorize the capabilities of their USB-C device to know the supported generation.

2.3 IP and MAC addresses

2.3.1 IP addresses

IP stands for the Internet Protocol, and IP addresses are like street addresses; in the sense that they represent locations on a network. There are 2 main types, IPv4 and IPv6 addresses.

```
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noque
    link/ether 5c:c5:d4:e2:61:3e brd ff:ff:ff:ff:ff:ff
    altname wlx5cc5d4e2613e
    inet 10.71.118.252/18 brd 10.71.127.255 scope global dynamic
        valid_lft 2092sec preferred_lft 2092sec
    inet6 fe80::1540:dda:1da0:53b4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

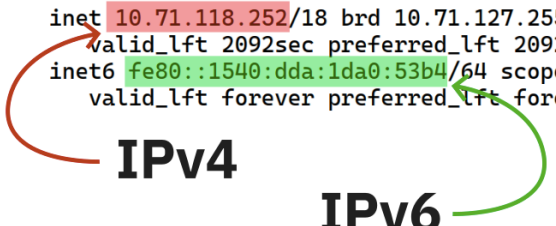


Figure 2.4: Output from the Linux `ip` utility, showing IPv4 and V6 addresses

IPv4 Addresses

Are **32-bit values** represented as 4 3-digit numbers with dots between them, like `192.168.0.1` with each value ranging between 0-255. There are 2 types of IP addresses, *Public* and *Private* IP addresses.

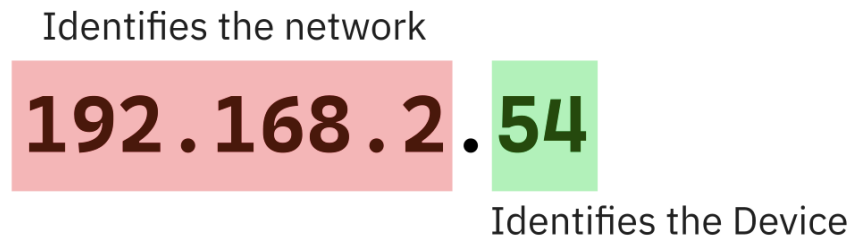


Figure 2.5: The two components of an IPv4 address

Figure 2.5 shows the two main components of IPv4 addresses; the first component determines the network itself, and the second component determines the device on the network uniquely.

In terms of private and public networks it will be covered more in **Chapter 4** and later sections about NATs.

Public IPv4 Addresses

These IP addresses denote the location of your home router or other routers on the public internet. All web servers, like the one in our school or your personal router has a public IP address accessible by everybody¹³.

The only rule they must follow is that their addresses cannot be in the range of **private IP addresses**.

Private IPv4 Addresses

Private IP addresses are like public IP addresses, but instead of it being publicly accessible to everybody on the internet, they are only accessible to users connected to a private network, like a phone hotspot or the network created by your router. Every device on, say, your home network like your phone or your mom's work laptop have a private IP. These addresses are then mapped to public IP addresses through NATs (Network Address Translators); more details in section 2.3.5.

In the IPv4 address range, addresses are reserved for private networks to use, and they are:¹⁴

Class	IP Range	Most typically seen in
A	10.0.0.0 to 10.255.255.255	Large networks
B	172.16.0.0 to 172.31.255.255	Medium-sized networks
C	192.168.0.0 to 192.168.255.255	Small networks

Figure 2.6: Table of private IP classes

¹³although they do not necessarily respond with data all the time

¹⁴Taken from <https://www.geeksforgeeks.org/private-ip-addresses-in-networking/>

Note that you do not need to memorize the class letters nor the exact ranges, you just need to identify them, especially the most common one, class C which is between 192.168.0.0 to 192.168.255.255.

These private IPs belong in a local network, where devices in the same approximate geographical location are connected to a router.

2.3.2 IPv4 Address Assignment

When you connect to the school network or your home WiFi network, you need to get an IP address in order for your device to communicate with the router, to transmit data to and from websites like Google. Without it, your device would be useless.

Static IP assignment

You must ask the router for an IP address; and there are 2 main methods. This one is called static IP assignment; this means that the IP address you have on the network does **not** change, and *your device chooses the IP address it has*. The IP you assign to yourself is now assigned to you for an indefinite¹⁵ time.

Pros include:

- It is good for servers, because your location on the network does not change. Computers on the network will be able to find you easier.
- The connection process is much faster, as you don't have to ask for one.
- You get faster upload/download speeds (only if you use a static NAT!¹⁶)

Cons include:

- on a large private network, your location on the network is more easily identifiable, as your IP doesn't change.
- you cannot tell if the IP you assigned to yourself is available or not.¹⁷
- it is expensive to maintain, as the device at the address must constantly be on to be available.

Dynamic IP assignment

This is like static IP assignment, as you are receiving an IP address to identify yourself. But here, you use a protocol called DHCP (Dynamic Host Configuration Protocol)

¹⁵not infinite, just an unknown amount of time

¹⁶You will learn about NATs in a later section. Here, it means that a private address is directly mapped to the public address of your router.

¹⁷You can always ping the IP address to see if a device at that IP responds. If there is no response, there is no device at the IP.

to talk to the router nicely to ask for an available IP address. Every time you connect to the network, your device talks to the router for an IP. Sometimes it is different, and sometimes it is the same as before.

This is by far the most common method.

Pros include:

- On large public networks, it tends to change more often, making it more secure.
- It is a lot more convenient, as the device does not have to set their own IP before connecting.

Cons include:

- If you're connected to a video call or voice call through VoIP¹⁸, if your IP address changes, it may disconnect the call¹⁹.
- If your device is old and does not support the DHCP protocol, you cannot use it. In short, it may not support all devices.
- Connection time takes longer, you must send a **DHCPLEASE** message to the router for the IP, the router must calculate an IP, and the router must send a **DHCPBACK** with your IP. This typically takes 5 seconds but may take up to 25 seconds.

2.3.3 MAC addresses

MAC addresses are 48 bit numbers that identify your device uniquely on any given network. They are typically represented as 3 hexadecimal numbers:

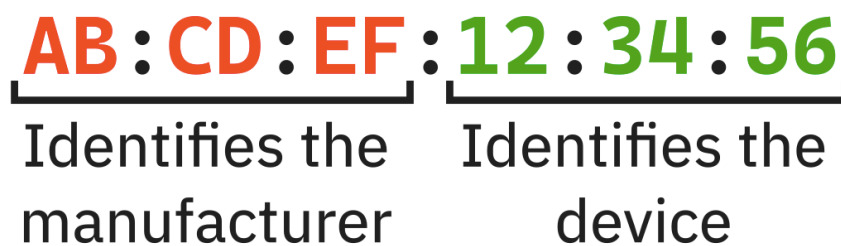


Figure 2.7: A breakdown of the portions of a MAC address.

The first 3 hexadecimal bytes (2 digits) identify the maker of the device²⁰, like Apple²¹. The last 3 digits uniquely identifies the device itself.

Figure 2.8 shows the MAC address of one of my computers. The first 3 components identify the manufacturer, and the last 3 uniquely identifies my device.²²

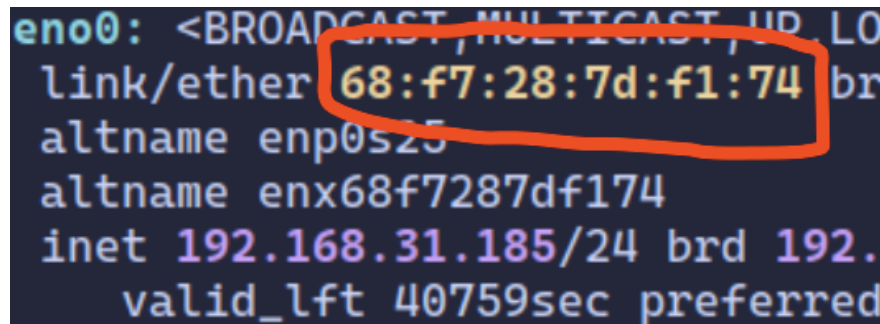
¹⁸Basically making phone or video calls through a network and not cell towers.

¹⁹It is not necessarily true, as your IP is not jumbled at regular intervals.

²⁰Technically the NIC or Network Interface Card itself

²¹Not necessarily, again, it identifies the maker of the NIC.

²²In this example, I have changed the default MAC address to a randomized one, so the manufacturer may be invalid if you try to look it up.



```
eno0: <BROADCAST,MULTICAST,UP,LO  
link/ether 68:f7:28:7d:f1:74 br  
altname enp0s25  
altname enx68f7287df174  
inet 192.168.31.185/24 brd 192.  
valid_lft 40759sec preferred
```

Figure 2.8: The MAC address of a computer, running Arch Linux.

Universally Administered

These addresses are by far the most common; they are the addresses assigned by manufacturers and universally identify the device. They are used to ensure the unique identification of the device throughout the globe, and avoids conflicts with other MAC addresses.

Locally Administered

These are a less commonly talked about form of MAC addresses, these are assigned to the computer by network administrators. These addresses have no guarantee of being completely unique from computer-to-computer, and are used in specific cases where addresses may conflict with each other.

Some organizations, such as mainframe or cluster computers, require MAC addresses to be in a certain format. LAAs allow for the MAC address to be changed, therefore allowing one to change the format. It may also be used to bypass a firewall, as some require the MAC addresses to be in a certain format.

Extra: LAA addresses also have a property where the 7th bit of the 1st byte of the MAC address is set to 1. UAA addresses have that bit set to 0.

2.3.4 Differences between IP and MAC addresses

MAC addresses	IP addresses
Identifies unique information about a device on a network	Identifies the location of something on a network
Unique for the device on the network	Only unique if it is a public IP. Private IPs are only unique to that network, and addresses across many private IPs may be the same, as they are powered by different routers.

48 bits	32 bits for IPv4, or 128 bits for IPv6
Can be locally or universally administered	Can be static or dynamic
Assigned by the manufacturer and is baked into the NIC	Static IPs are assigned by the connecting device. Dynamic IPs are assigned by the DHCP server, usually built into the router.

2.3.5 NATs (Network Address Translation)

By now you should be familiar with the concept of private versus public IPs. Private IP addresses only exist on internal private networks, and are separate from all public IPs. How do devices on private IPs then transmit data to websites on public networks?

NATs help one accomplish this. They allow private IP addresses and public IP addresses to be temporarily linked together to create a channel whereby data can flow. These are typically in your home router to map devices connected to it to the router's public IP address.

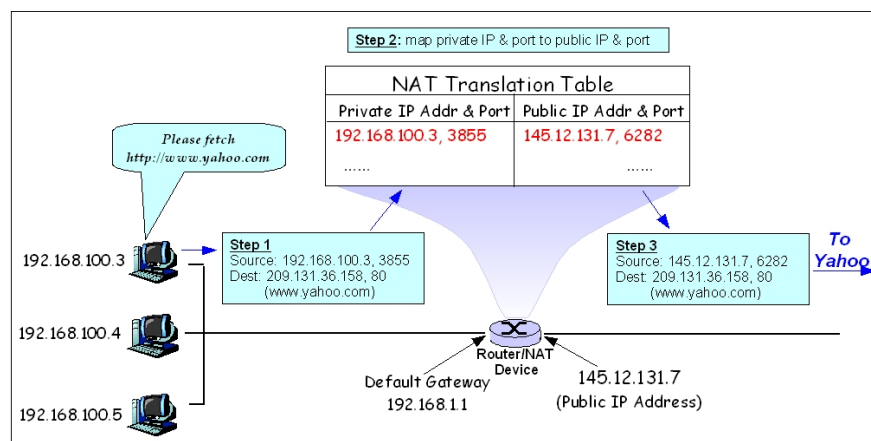


Figure 2.9: A diagram explaining how network addresses are mapped to public IPs through the NAT. Taken from Wikipedia.

As seen in figure 2.9, computers on the home network can request for websites (in this case, Yahoo, as the diagram is presumably very old), and the NAT in the router can add your device to a temporary table that shows where traffic from one private IP should go, and vice versa.²³

²³Do not worry about the port; this allows 65535 channels of data to be open on one IP address, as one single IP is quite limiting.

2.4 Packet Switching

Sending data across large distances is used all the time, like in video chats, voice chats, uploading homework or simply loading HTML²⁴. Typically, sending data like this is done through a stream of **data packets**, otherwise known as just packets.

A packet looks like the following:

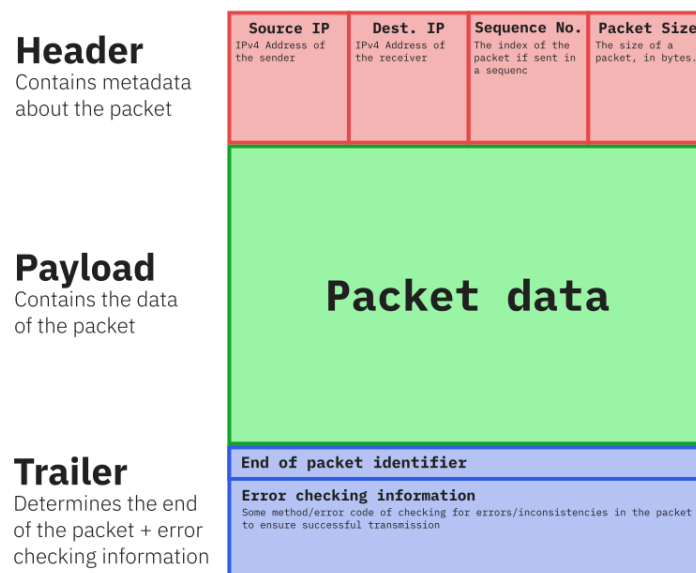


Figure 2.10: The components of a data packet.

2.4.1 Components of a packet

Here are they:

- The Header:
 - The IP²⁵ of the sender
 - The IP of the receiver
 - The sequence number; If a lot of data must be sent throughout multiple packets, the *sequence number* makes sure that the packets' payloads are re-assembled in the correct order.
 - The packet size, in order to make sure the packet received is of the correct size.
- The payload, which consists of the binary data to be transmitted via the buffer.
- The trailer, which consists of:

²⁴This markup language, commonly mistaken for a programming language is what lays out websites; it dictates the structure of the site, like the bricks for a house.

²⁵IP Address; in this context, always IPv4 unless if specified

- Some way of identifying the end of the packet. This is typically some special value like a null terminator²⁶. The algorithm can then scan the data until it hit that character to extract the payload.
- An error checking method. CRCs are used to check this.

2.4.2 Packet Switching

Packet switching is the next step after slicing up data into packets. This is done because sending data across large distances all at once is prone to failure (what happens if a slight portion of the data is corrupted? the data must be sent all over again, taking time).

This technique involves sending packets onto multiple paths that reach the same destination. They begin by being sent out onto the network; they travel separately but at the same time and when all the packets arrive at the destination, they are put back together via the sequence number. The role of the router on each node is to determine the next router the packet should go to.

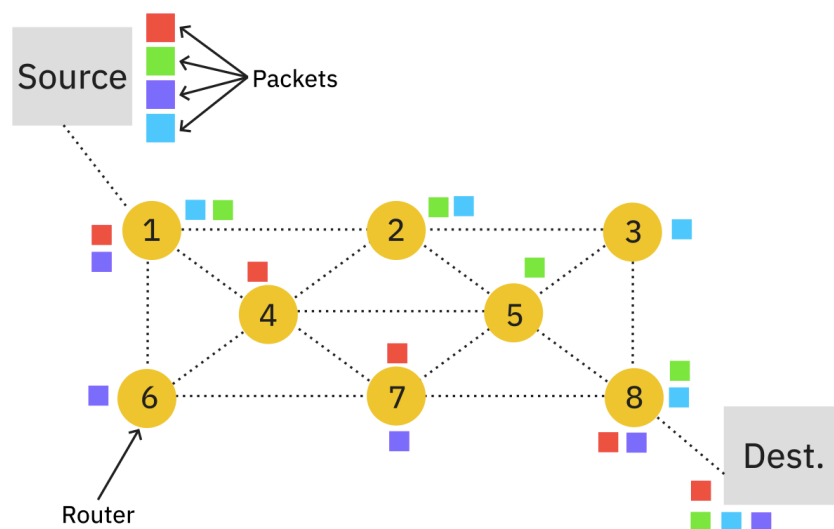


Figure 2.11: A diagram to illustrate packet switching.

Each yellow circle can be interpreted as an intersection on a road. The general term for these are nodes, but they are technically just routers.

Figure 2.11 has 4 packets that must be sent. The key points are:

- The path between the source and the destination are made up of the yellow circles; the nodes.
- Each packet takes a different path, i.e. the red packet takes 1-4-7-8, but the blue packet takes 1-2-3-8, etc.

²⁶In programming, this is referred to as a *sentinel value*, which just means a value that carries some special meaning. An example is like in the C programming language, to tell the end of a string, a sentinel value of 0 is used to denote the string finished.

- The router at each node will decide the next router the packet should go to. The shortest *available* route is always taken but not the *absolute shortest* route.
- When the packets arrive, they will be put back in order according to the sequence number (not depicted).

2.5 Error Correction

To make sure that data sent from a device is received on the other side correctly, without errors, error correction must be done to make sure the data is valid. Here are some methods:

- **Parity Checks:** This adds an extra bit (0 or 1) to a piece of data (usually a byte) based on the number of 1s already present. The number of 1s are totalled. Depending on if even or odd parity are agreed upon from the sender and receiver, if the number of 1s is even, the parity bit is 1, for odd parity the parity bit is 1 if the number of 1s is odd. If the receiver's calculated digit is the same as the digit sent, it indicates no error. However, this method can only detect single-bit errors and not multiple-bit errors.
- **Cyclic Redundancy Checks (Checksums, CRCs):** The data bits are added together, processed in a certain predetermined way, and the result (checksum) is appended to the data packet. The receiving device recalculates the checksum on the received data and compares it to the received checksum. If they match, the data is assumed to be error-free. This method is more robust than parity checks.
- **Echo Checks:** The sender sends the data packet and then waits for an exact copy of the packet back from the receiving device. This confirms successful data transfer but introduces a delay due to the back-and-forth communication.

2.5.1 Parity Blocks

These are like 2D parity checks. Each byte of data sent will have one bit used for parity, when the digit is calculated horizontally. Then, a whole other byte of data is also sent; with just parity bytes from parity checks from each vertical column.

▼ Table 2.3 Parity block showing nine bytes and parity byte

	Parity bit	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
Byte 1	1	1	1	1	0	1	1	0
Byte 2	1	0	0	1	0	1	0	1
Byte 3	0	1	1	1	1	1	1	0
Byte 4	1	0	0	0	0	0	1	0
Byte 5	0	1	1	0	1	0	0	1
Byte 6	1	0	0	0	1	0	0	0
Byte 7	1	0	1	0	1	1	1	1
Byte 8	0	0	0	1	1	0	1	0
Byte 9	0	0	0	1	0	0	1	0
Parity byte	1	1	0	1	0	0	0	1

Figure 2.12: From the textbook. "[The table] shows how the data arrived at the receiving end. It is now necessary to check the parity of each byte horizontally (bytes 1 to 9) and vertically (columns 1 to 8). Each row and column where the parity has changed from even to odd should be flagged."

Figure 2.12 from the textbook shows this. An extra byte of information is used as parity bits for each column, and each row has one data bit dedicated to storing horizontal parity data. This allows exact errors to be pinpointed (bit flips).

2.5.2 Check Digits

Check digits are another form of error correction; usually involving checking for errors in numbered codes, such as barcodes.



Figure 2.13: Barcodes consist of black and white bars that represent decimal numbers; written at the bottom.

typically, barcode readers that read these numbers may encounter stains on the physical code and such. Therefore, error correction is needed to make sure the data is not damaged to the point the barcode cannot be read.

ISBN-13

To calculate:

1. Add all odd digits
2. Add all even digits, and multiply the result by 3
3. Add the results from steps 1 and 2, and divide the result by 10
4. If the remainder is 0, use 0. Otherwise, subtract your result from step 3 from 10. This is your check digit.²⁷

To verify:

1. Add all odd digits
2. Add all even digits, and multiply the result by 3
3. Add the results from steps 1 and 2, and divide the result by 10
4. If the remainder is 0, there is no error.

Modulo-11

To calculate:

1. You must calculate a *Max Weighting*. Take the length of your number data, and add 1 to get your weighting.
2. Take your first digit, and assign it to your weighting. Assign the next weighting to the max weighting minus one. Assign the next weighting after that to the max weighting minus 2, until you run out of digits.
3. Multiply each digit by the weight.
4. Add all the products from step 3 together.
5. Divide the result from step 4 by 11, and note down the remainder.
6. Subtract your remainder from 11.²⁸ This is the check digit.

To verify:

1. perform steps 1-5.
2. If the remainder is 0, the check digit is correct.

²⁷As an example, if you got 6 for step 3, your check digit is $10 - 6$ or 4.

²⁸As an example, if you got 7 from step 5, your remainder is 4

2.5.3 Automatic Repeat Requests

In short, for small amounts of data, if the data received by the receiver is invalid, it automatically asks the sender to re-send the data.

This is a system used in networking to make sure the data received by the receiver over the network is valid. Before the data is sent, both sides agree on a timeout. The timeout determines the max time the data can take to be sent. The data is sent, and the timeout (like a stopwatch) begins. If the data is not received before the timeout is over, the data must be re-sent.

Acknowledgements are used for the sender and receiver to communicate about where the data is and the integrity of the data. If the data is received correctly, a *positive acknowledgement* is sent back to the sender. If the data is erroneous, a *negative acknowledgement* is sent. If there is no acknowledgement sent, the data is assumed to be lost in transit, and the data must be resent.

Example

- Tom's computer wants to send Jerry's computer a data packet. Tom's computer establishes a timeout of 10 seconds and sends out the packet.
- If Jerry does not send an acknowledgement in time, the communication is cancelled.
- If Jerry does get the data, a positive or negative acknowledgement is sent. The process is repeated until there's a positive acknowledgement from Jerry.

2.6 Encryption

Encryption is a process by which data is obfuscated in a certain way that only the sender and receiver can see the data in question, but onlookers along the way cannot. The process of obfuscating is completely reversible given a consensus on which method to use. Two of them exist. In what matters to us, keys are used to achieve this process.

Symmetric Encryption

This is when one key is used to encrypt and decrypt the data. The key is applied to the data to encrypt it, and applied to the data to decrypt it. Although the approach is simple, if a hacker gets hold of the key, all data encrypted with the key is compromised.

First, they go to a secure location and exchange their keys. Then, they can exchange data:

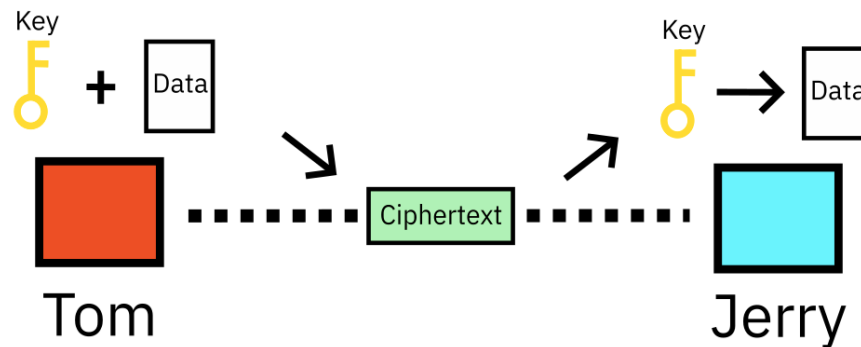


Figure 2.14: Tom and Jerry exchanges a message with the same key.

The same key is used both for encryption and decryption.

Asymmetric Encryption

This is when both the sender and the receiver have a pair of keys. One is used to decrypt data (the private key), and the other one is used to encrypt data (the public key). The public key is kept public, as anybody should be able to encrypt data for you. But only yourself can decrypt the data people garbled up for you. This means that only you can decrypt the data, which is the intent. If you want to encrypt data for someone else, you use their public key, and the other person has the private key to decrypt it.

Let's say Tom and Jerry now wants to make use of asymmetric encryption. To begin, they need each other's keys to encrypt data for one another. The process is called a key exchange (self explanatory).

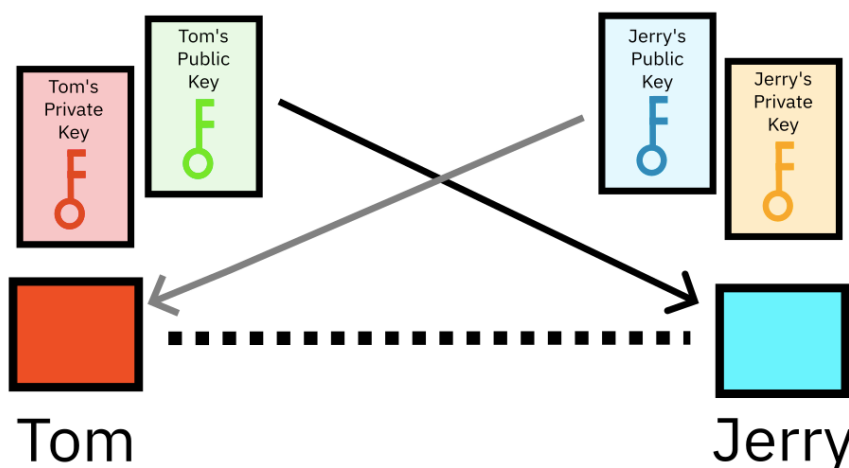


Figure 2.15: Tom and Jerry exchanges keys.

Then, if Tom wants to encrypt data for Jerry, Tom uses Jerry's public key, and Jerry decrypts it with his private key.

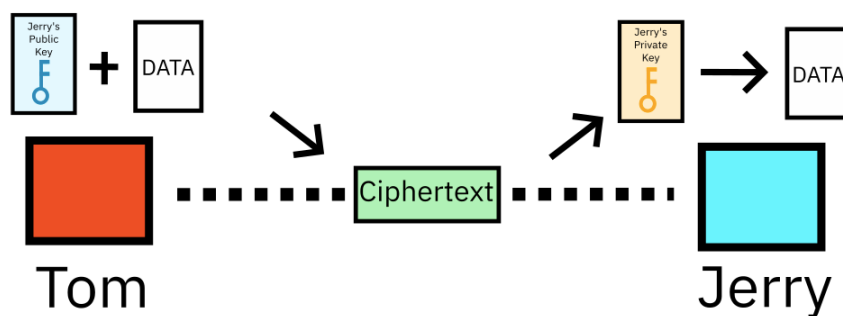


Figure 2.16: Tom sends data to Jerry.

And vice-versa.

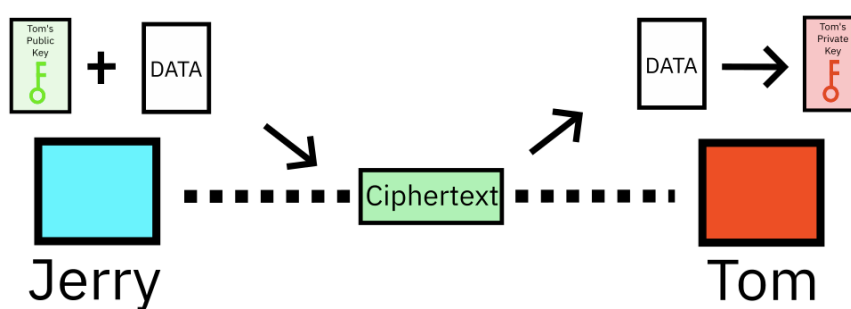


Figure 2.17: Jerry sends data to Tom.

Chapter 3

Hardware

3.1 Data Storage

Computers need to store data in order to function, like code, or data meaningful to you, like documents and images. Storage devices accomplish this. There are 3 main categories of storage devices, which will be covered here.

3.1.1 RAM

RAM or Random Access Memory; otherwise referred to as just Memory, stores *volatile data* that does not have to be on your hard drive. It only stores data that the computer uses when it is on. Examples of data that goes into memory include temporary values that programs must store; from simple integers to browser tabs.

It has a property where it is directly addressable by the CPU. Check section [3.2.1](#) for details on exactly what that is, but it is essentially the brains of your computer. This makes RAM very fast.



Figure 3.1: What a stick of LPDDR3 RAM (DRAM) looks like.

Some important information include:

- RAM can be freely written to, or read from. The data on RAM can be changed by the user, but not directly; through programs. As an example, opening a browser tab puts data into RAM, but you will not notice physical differences.
- RAM is **volatile**. All data stored in it disappears when power to the RAM is lost.
- RAM is critical to your computer's function; as all code and data used by code is stored in RAM.
- Increasing the amount of RAM will boost the speed of your system, as the computer is able to store more of this temporary data at once in a fast location.

There are 2 main kinds of RAM, SRAM (S for static), and DRAM (D for dynamic).

Dynamic RAM

DRAM chips (the black squares that you see on the image) consist of transistors and capacitors. It is by far the most common type of RAM in all computers. The access time for DRAM is ~60 milliseconds.

They consist of:

- Capacitors hold a bit of information (0, or 1)
- Transistors act as switches, which allows the chip's control circuitry to read or write to the capacitors.

This must be constantly refreshed, as the capacitors cannot hold data for very long.

Benefits of DRAM include:

- Much cheaper to make than SRAM
- Consume less power on average
- They hold a larger total capacity (typically)
- DRAM in computers are upgradable in many cases. Modern laptops do not allow you to do so for the most part, but all desktops and some older laptops allow you to remove and exchange DRAM easily for upgrades/repairs.

Drawbacks of DRAM include:

- It needs constant refreshing to keep the capacitors charged
- It is slower than SRAM, by more than 2 times. This limits its applications.

Static RAM

SRAM chips are made of flip-flops that hold a constant one bit value. They do not need to be constantly refreshed, therefore. SRAM is used when speed is required, such as the CPU's cache. The access time is ~25 milliseconds.

Benefits of SRAM include:

- Much faster than DRAM due to less latency¹
- Consumes less power
- Does not need to be constantly refreshed

Drawbacks of SRAM include:

- Much more expensive
- Very low capacity
- More complex and typically incompatible circuitry must be used to access the RAM.

¹Delay.

Differences between DRAM and SRAM

Dynamic RAM (DRAM)	Static RAM (SRAM)
DRAM is slower	SRAM is faster
Uses transistors to control the flow of electrons, and capacitors store the binary 1s and 0s	Uses flip-flop circuits
Must be constantly refreshed to make sure the capacitors have charge	Does not need refreshing
Cheaper to make	Much more expensive to make
In a computer, the main memory uses it	The CPU's cache uses it
Less overall power consumption	More overall power consumption
Higher capacity	Lower capacity

3.1.2 ROM

ROM is like RAM, however, it stands for *read-only memory*. Like RAM (see section 3.1.1), it stores data that is quickly accessible to the computer; i.e. directly addressable by the CPU. However, the *key difference* is that **ROM is NOT erased when the computer powers off**. This is part of the reason why ROM is purely read-only. ROM cannot be erased.²

The main use of ROM in computers is to store the BIOS/UEFI. This is code that the CPU executes immediately when the computer starts up. It does critical checks to make sure all your peripherals are working (like your keyboard), applies specific security patches to your CPU that the CPU maker sees fit, and loads the OS (see section . Increasing the size of ROM does nothing in most systems.

²Nowadays, computers do not use pure ROM chips anymore, as...they cannot be erased. Instead, they use EEPROM, or electrically-erasable and programmable ROM, which can be erased, but not as easily as RAM. You need a special device to erase an EEPROM, but it is doable. This means that manufacturing errors when making ROM can just be overwritten.

3.1.3 Virtual Memory

3.1.4 HDDs (Hard disk drives)

3.1.5 SSDs (Solid state drives)

3.1.6 USB Mass Storage (Flash Drives)

3.1.7 Optical Media

3.1.8 QR Codes

3.2 The Von Neumann Architecture

The von neumann architecture specifies a generic structure for CPUs and microprocessors to follow when they are designed. It dictates that the data used to store programs and the data used by the program (tempoary values, variables in memory) should co-exist in the same memory.

To begin, you must understand the main components of a computer.

3.2.1 The CPU

The CPU or the Central Processing Unit is the brains of your computer. It carries out all the instructions ever passed through your CPU, and is the center of your computer's activity. Modern CPUs have the ability to do many things at once, like playing music while doing word processing³

CPUs look like this:



Figure 3.2: An Laptop CPU in a socket, an Intel Core i7-4712MQ, a Laptop CPU directly soldered on the motherboard, an Intel Core i7-3520M, and a desktop Intel Core i7-14700KF.

Extra: The shiny part of the CPU is called the die. The die is filled with extremely dense transistors which are all semi-conductive. The die actually does the processing. For those

³A very, very, very oversimplified example.

who wonder, no, you cannot see the individual transistors and registers on the CPU; they are so incredibly small and the parts of the CPU are so incredibly small it is impossible to reverse-engineer its architecture; even with microscopes.

3.2.2 Components of a CPU

3.2.3 Buses

3.2.4 The Fetch-Decode-Execute Cycle

3.2.5 The Clock Speed

3.2.6 Increasing CPU speed

3.2.7 Cores

Cores are like Sub-CPU's in each CPU. Each CPU shown in figure 3.2 has more than 1 core. Each of them can carry out tasks independently of each other; despite how they can communicate together. For those who have already read this whole section, each core can carry out its own fetch-decode-execute cycle.

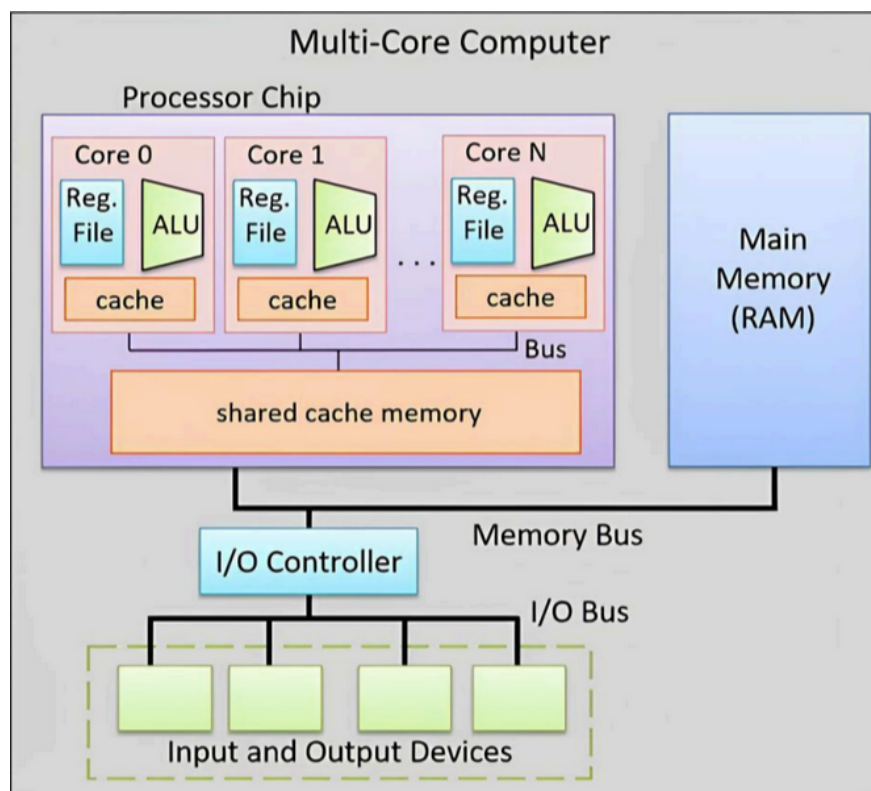


Figure 3.3: A diagram showing what cores look like in a CPU.

3.3 Input and Output Devices

Stub

stub

3.4 Network Hardware

Stub

stub

Chapter 4

Software

stub

4.1 The Operating System and Kernel

Stub

stub

4.2 System Startup and Interrupts

Stub

stub

4.3 Userspace Software

Stub

stub

4.4 Programming Languages and Development Environments

Stub

stub

4.5 Translators, Compilers and Interpreters

Stub

stub