4.5.5 Information Coding Systems

ASCII and Unicode

As computers can only process and store 0s and 1s there needs to be a binary representation to encode characters.

ASCII is a character set for representing characters using decimal encodings.

- The ASCII character set **uses 7 bits which can encode 128 different characters.**
- The extended ASCII character set uses 8 bits which can encode 256 different characters.
- This is restrictive and does not allow enough characters beyond the **Latin alphabet**.

Unicode extends the ASCII character set.

- It includes other characters such as **Asian language symbols and emojis.**
- Unicode can be used with **16-bits (UTF-16) or 32-bits (UTF-32)**.

Error Checking

When data is being transmitted over a network or internally in a computer there are many chances for data to be corrupted.

Bits can get flipped from 0 to 1 or from 1 to 0 due to issues such as electrical interference.

Error checking methods are used to try to detect errors.

They can attempt to detect if data was corrupted during transmission so it can be retransmitted or try to correct the error.

Four error checking methods are:

- Parity bits
- Majority voting
- Checksums
- Check digits

Parity Bits

A parity bit is a bit that is used to balance the number of 1s in a byte to an odd or even amount.

- Using **odd parity**, the parity bit is set to a value that makes the total number of 1s in the byte be odd.
- Using **even parity**, the parity bit is set to a value that makes the total number of 1s in the byte be even.

If an even number of bits are flipped during transmission, the parity check may fail to detect the error.

Example: Data is being transmitted and using an odd parity as an error detection method.

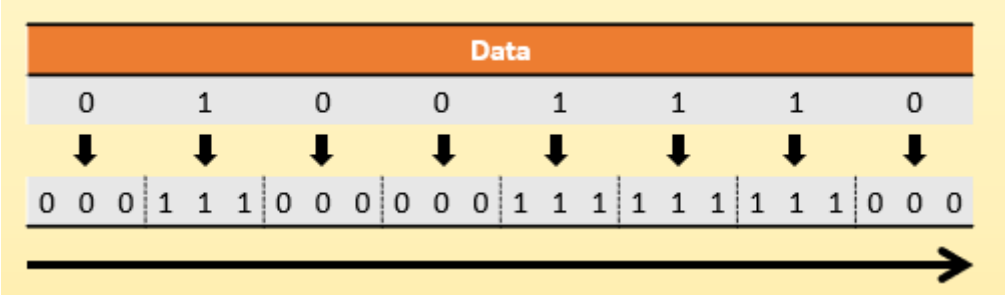
Parity bit	Data						
1	1	0	0	1	1	1	0

- There are four 1s in the data. Therefore, we set the parity bit to 1 so that the total number of 1s is odd.
- After the data is transmitted, the receiver can check the number of 1s in the byte.
- If the number of 1s was even when odd parity is being used, then the receiver can detect that the data was corrupted.

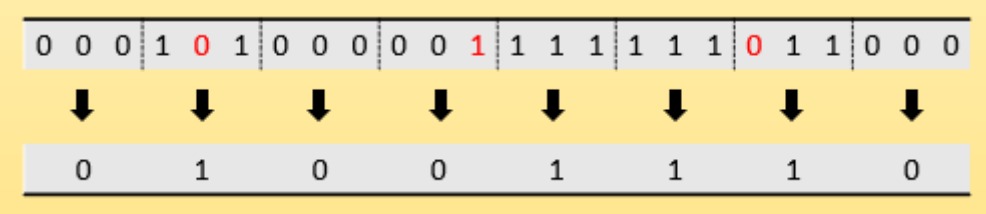
Majority Voting

Majority voting is an error detection method where each bit of the data is transmitted an odd number of times (at least three).

An odd number of copies ensures there can be a clear majority.



The receiver can take the most frequent bit in each group and assume that this value is the correct value.



Advantage: Unlike parity bits which can only detect errors, majority voting can correct errors.

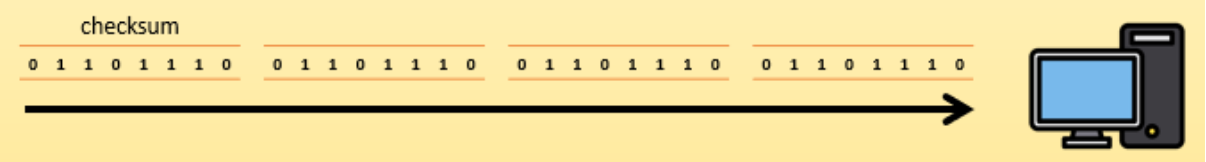
Disadvantage: As each bit is sent multiple times it uses more bandwidth and increases transmission times.

Checksum

A checksum is an additional value which is added to the end of a data transmission to check whether it became corrupted.

The sender uses an algorithm that takes the data to be sent and produces a value based on this input.

The checksum is then sent on the end of the block of data to the recipient.



The recipient takes the received data and applies the same algorithm to it to produce another value.

This value is compared to the checksum:

- If they are the same then the data was not corrupted during the transmission.
- If they differ then some of the bits must have been corrupted which gave a different output from the algorithm.

Check Digit

A check digit is a digit that is included at the end of a code or identification number.

Commonly, check digits are used on barcodes and ISBN numbers.

They work in a similar way to checksums but are much simpler; used more for human data entry than for communication systems.

The check digit is calculated using the other digits in the code.