Signals & Systems for Speech & Hearing:
Week 10

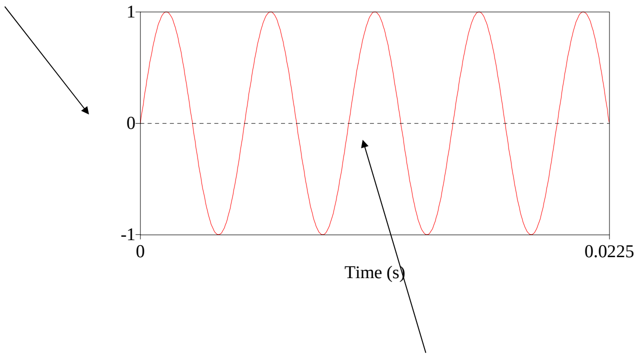
Digital Signals & Systems

The essential issue:
How can we store a waveform on a computer?

- In some sense, the only thing the computer can store is a list of numbers, so …
  - a waveform has to be converted in some way into such a list
- But …
  - Analogue waveforms exist at infinite points in time
  - Analogue waveforms have amplitude values with an infinite number of decimal places
  - What do we do?!
  - *Sample* and *quantise*

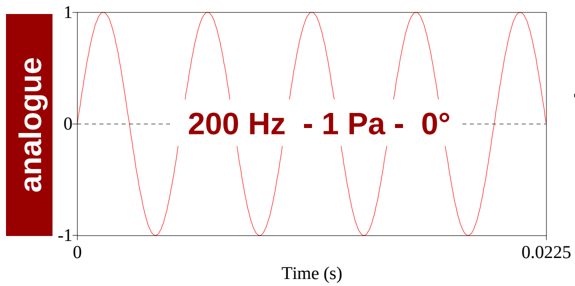
Analogue → Digital

Convert y-axis into numbers:
quantisation

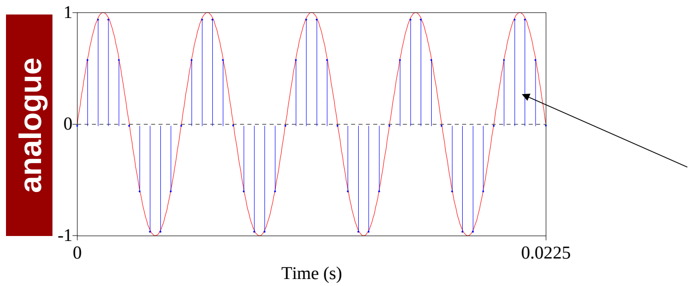


Convert x-axis into numbers:
sampling

Sampling

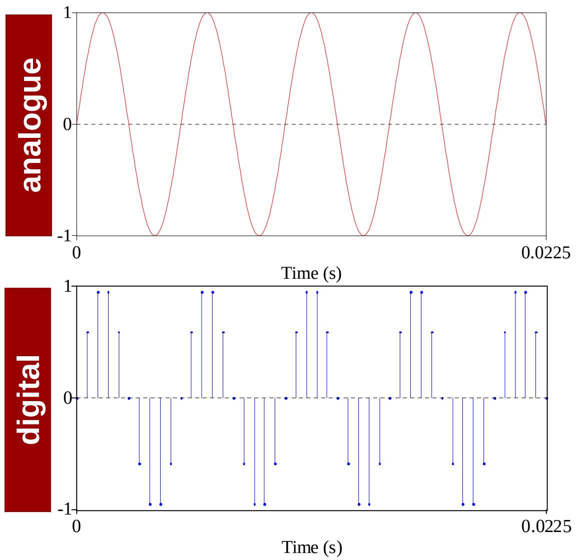


The analogue signal is continuous in time



taking discrete samples

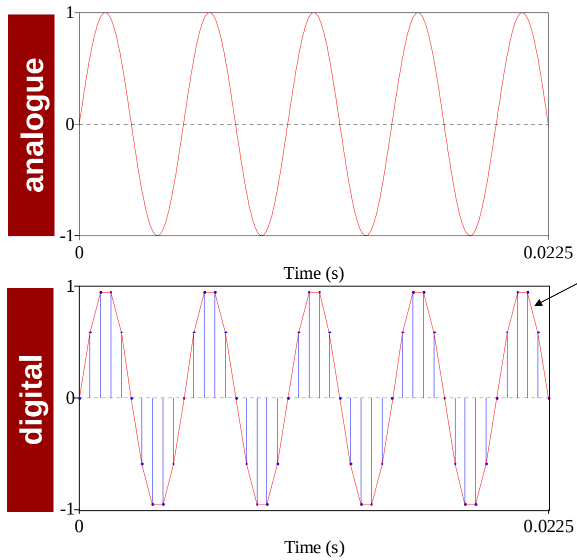
Sampling



continuous

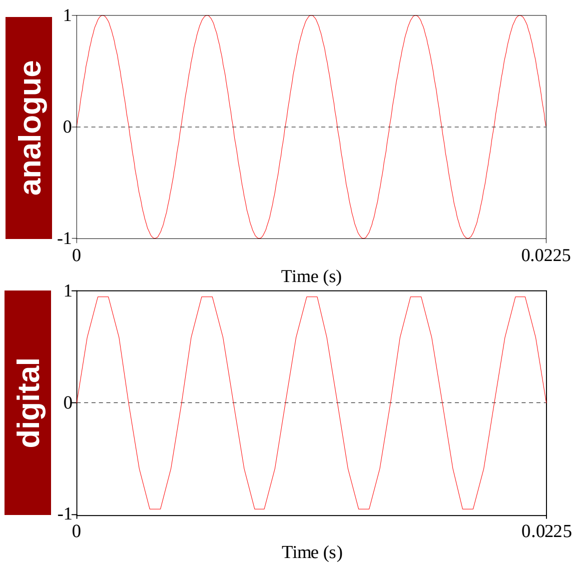
discrete

Sampling



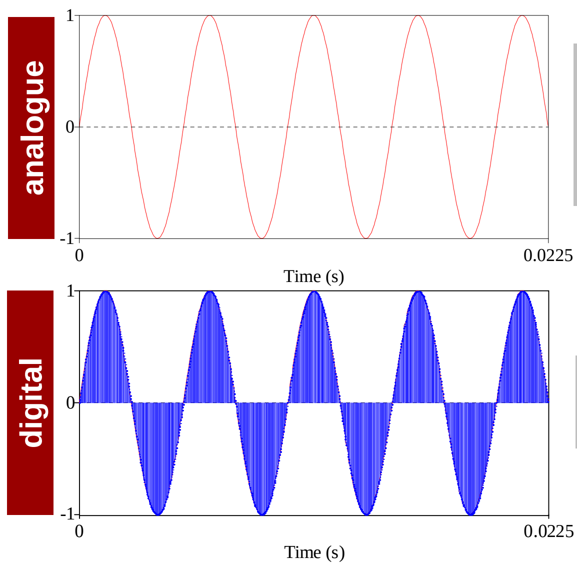
interpolating samples

Sampling



visual digital waveform on the screen

Sampling

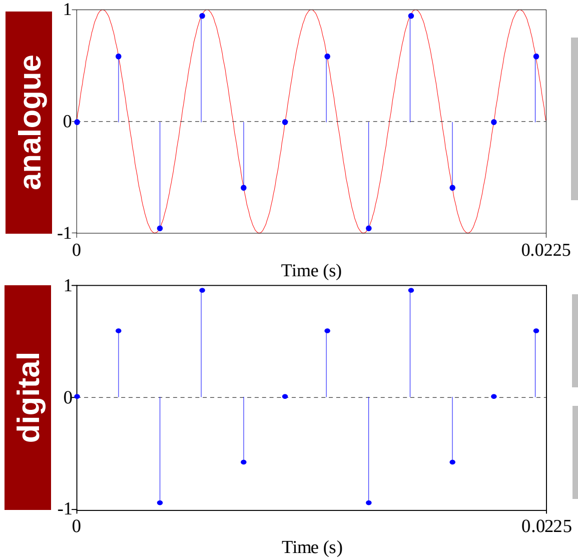


taking more samples:

e.g. 44100 samples per second

gives a more precise output

Sampling



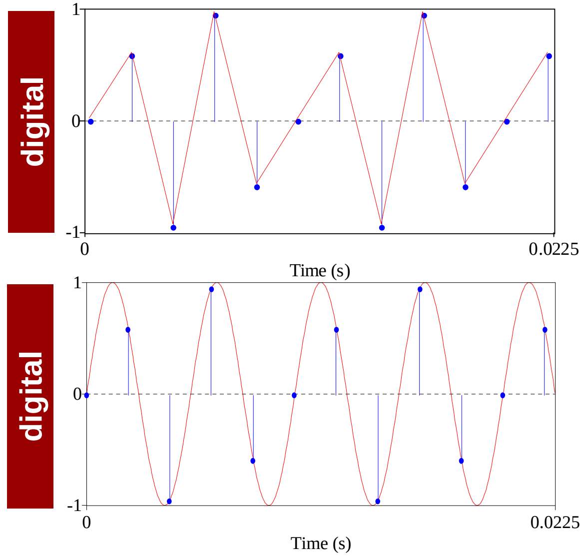
taking less samples:

e.g. 500 samples per second

gives a less precise output...

... but still each cycle is represented

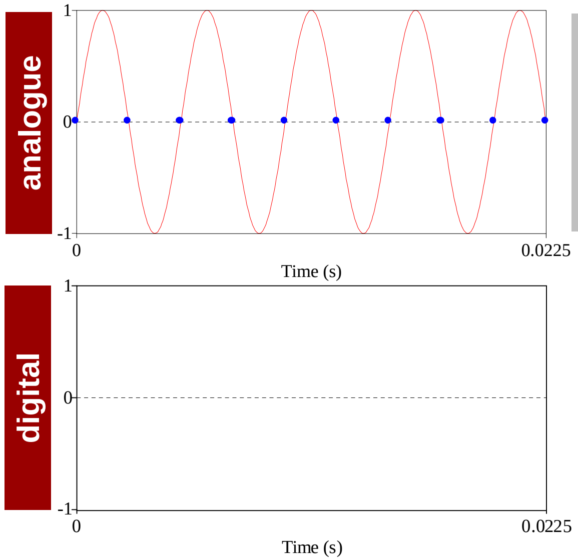
Sampling



This is what you see ...

... but this is what you hear (with a special kind of interpolation)!

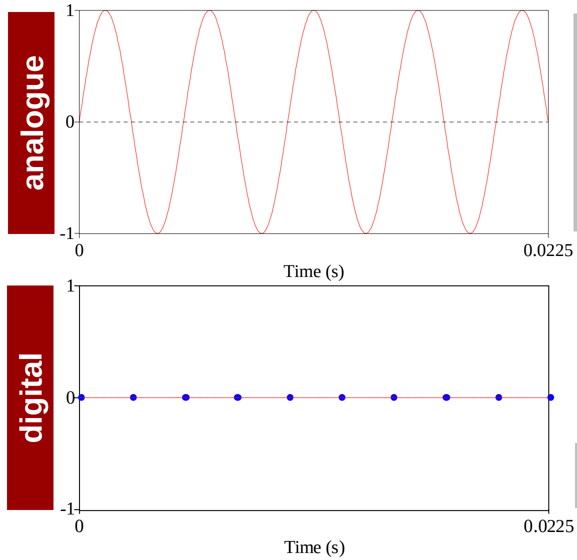
Sampling



What could happen when the sampling rate is exactly twice the frequency of the waveform?

In this case it would be 400 samples/ second.

Sampling



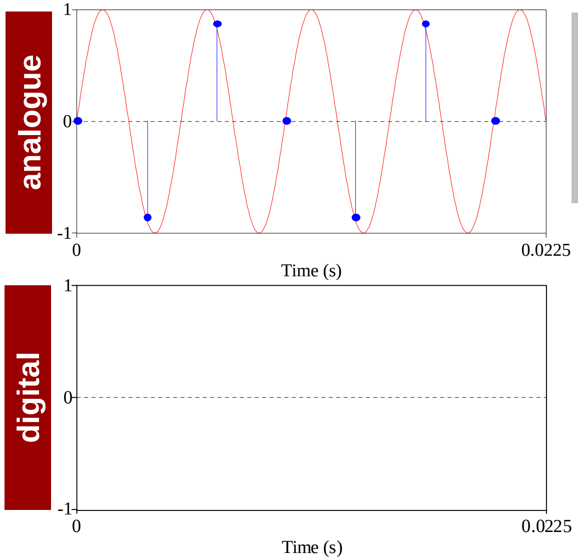
What could happen when the sampling rate is exactly double as high as the frequency of the waveform?

In this case it would be 400 samples/ second.

The digital output may be silence.

This frequency is called the Nyquist frequency

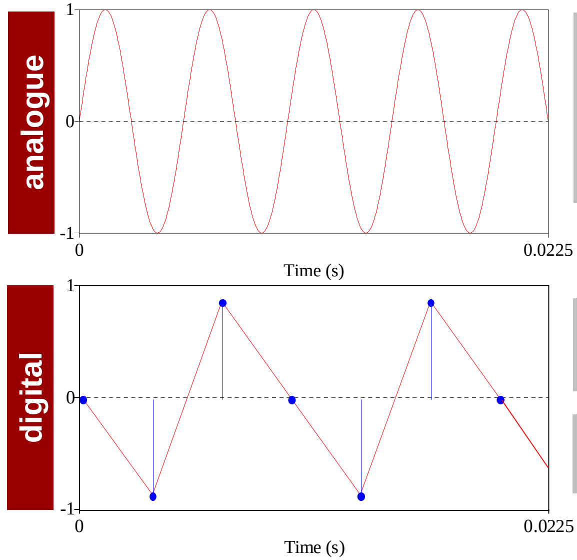
Sampling



What happens when the sampling rate is lower than the Nyquist frequency?

e.g. 300 samples per second

Sampling



What happens when the sampling rate is lower than the Nyquist frequency?

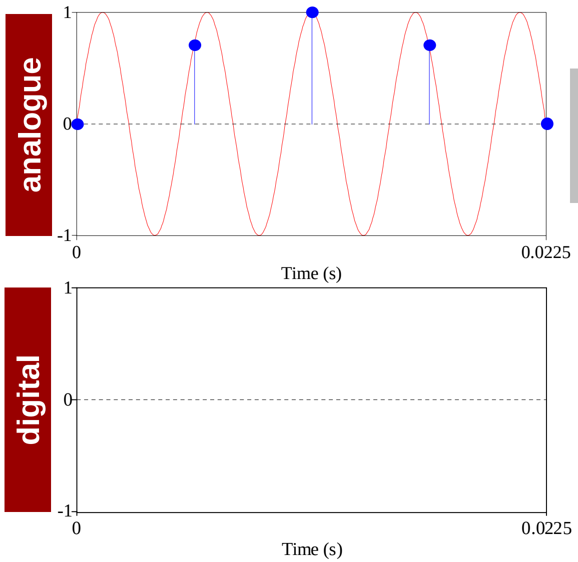
e.g. 300 samples per second

The waveform changes its period, i.e. its frequency.

This process is called:

aliasing

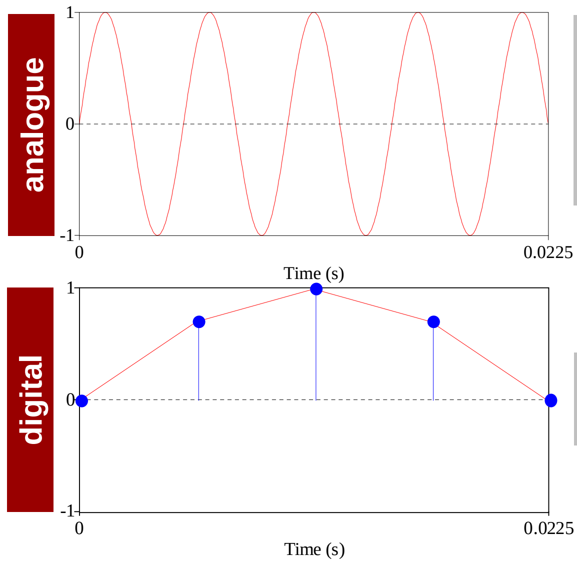
Sampling



Another example of aliasing:

A sampling rate of 177.7 samples/ second.

Sampling



What happens when the sampling rate is lower than the Nyquist frequency?

Let's say e.g. 177.7 samples/ second.

The aliased waveform has a frequency of 22.22 Hz (1/0.045)

Sampling

Review:

The Nyquist rate is the sampling rate below which the frequency of a sine wave is no longer represented acurately when converted to digital form.

To represent the frequency of an analogue sinusoid accurately in a digital form, the sampling rate has to be higher than the Nyquist frequency. That is, the sampling rate has to be more than twice the frequency that is to be represented in the digital signal.

This is referred to as the *sampling theorem*

All sampling rates below the Nyquist frequency will result in aliasing, i.e. sine waves appear at a different frequency in the digital signal.

For complex signals, sampling must be at a rate higher than twice the highest frequency component in the signal.

Sampling

Problem:

A standard hifi sampling rate (e.g. for used on CDs) is 44100 samples per second.

Up to which frequency will an analogue signal be represented accurately after having been converted to a digital signal?

Answer: 22049.9999999999... Hz

Is that enough for speech?

Answer: Yes, by far. It is in frequency regions that are higher than the human ear is able to perceive.

But there may be much higher frequencies in the analogue signal which will be aliased to lower frequencies and will thus appear disturbingly in the signal to be digitised. How can we get rid of these frequencies?

Answer: By low-pass filtering the signal prior to sampling. This process is refered to as *anti-aliasing filtering*.

Quantisation and Storing numbers on a computer

- How does a computer store numbers at all?
  - As a series of 0s and 1s
- How do we transform even a single number (like 125) into 0s and 1s?
  - By using the *binary* system

Number systems

The *decimal system* has 10 symbols (0-9) from which all other numbers are created.

$$125_{10} = 100 + 20 + 5 = 1\cdot 10^2 + 2\cdot 10^1 + 5\cdot 10^0$$

The *binary system* has 2 symbols (0 & 1) from which all other numbers are created (one binary digit is called a *bit*).

$$6_{10} = 4 + 2 + 0 = 1\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 = 110_2$$

$$125_{10} = 64 + 32 + 16 + 8 + 4 + 1$$

$$125_{10} = 1\cdot 2^6 + 1\cdot 2^5 + 2\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 2\cdot 2^1 + 1\cdot 2^0$$

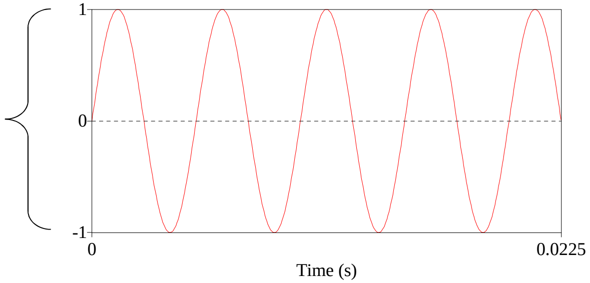
$$125_{10} = 1111101_2$$

Quantisation

Making the amplitude axis discrete:

QUANTISATION

Split amplitude axis
into a number of
discrete values



For quantisation computers use the binary counting system since one position of a binary number represents one bit on a computer.
One bit can have two values: 0 & 1

***All* data on a computer is at the lowest level represented as a series of bits.**

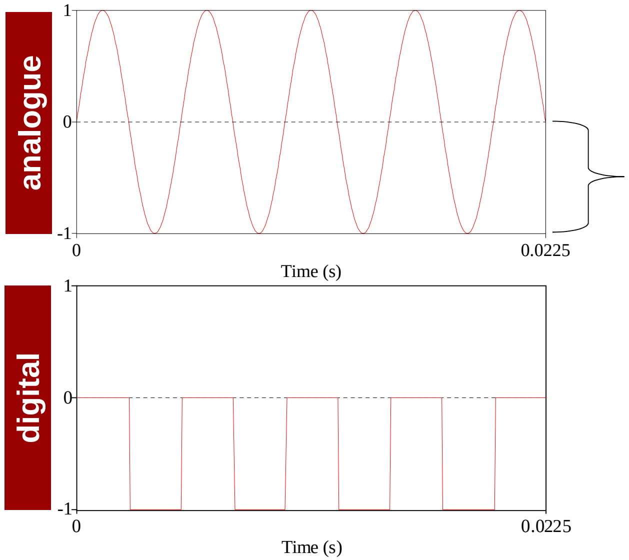
Quantisation

The number of values that can be captured using different numbers of bits:

1 bit = 2^1 = 2 values
2 bits = 2^2 = 4 values
3 bits = 2^3 = 8 values
4 bits = 2^4 = 16 values
8 bits = 2^8 = 256 values
12 bits = 2^{12} = 4096 values
16 bits = 2^{16} = 65536 values

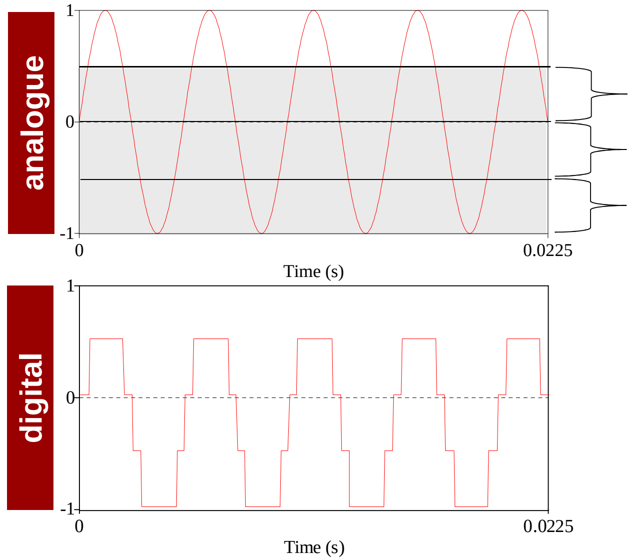
Digital Signals

QUANTISATION



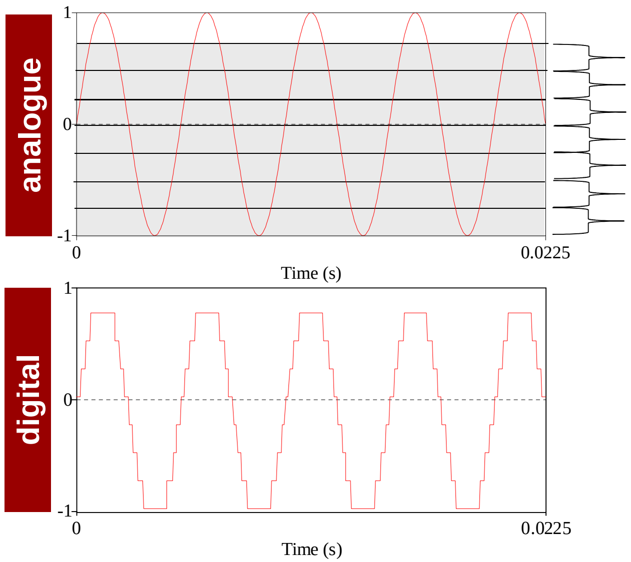
1 bit quantisation

Quantisation



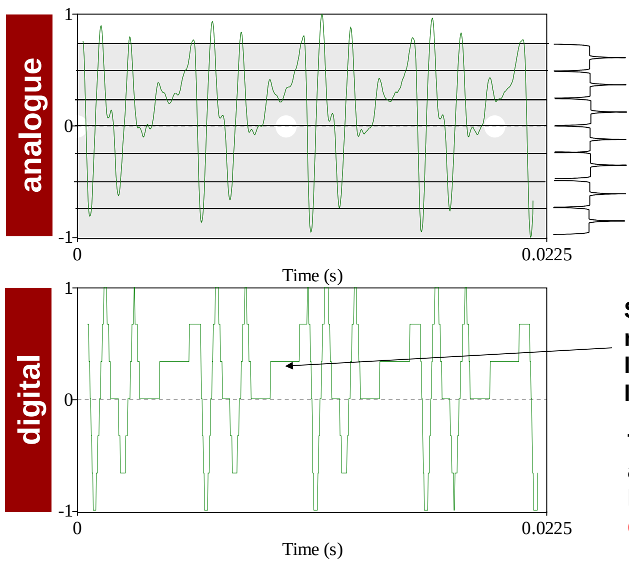
2 bit =
4 quantisation levels

Quantisation



3 bit =
8 quantisation levels

Quantisation



3 bit =
8 quantisation levels

Some values will fall randomly in higher or lower quantisation levels.

This causes low amplitude noise known as **quantisation noise**

Quantisation

16 bit quantisation is standard today

(i.e the amplitude is divided into 65536 quantisation levels)

So the highest possible amplitude range that can be captured by this would reach from 1 to 65536.

How big is that range in dB?

Assume that your measured value is 65536 and your reference is 1.

Then you get:

$$x \text{ dB} = 20 \times \log_{10}(65536/1) = 96.33 \text{ dB}$$

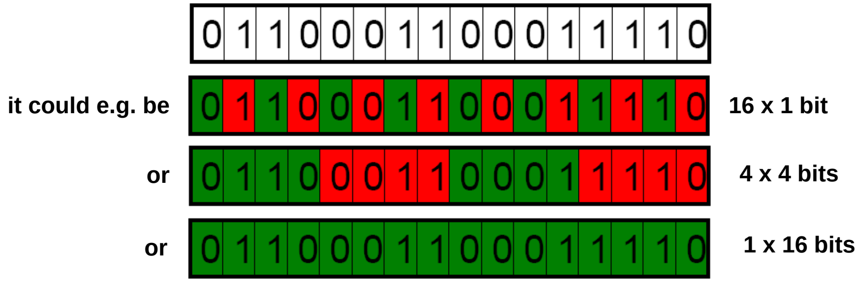
That means 16bit soundcards allow a dynamic range of ~ 96 dB

An essential difference between sampling and quantisation

- Sampling faster than the Nyquist rate means more samples to store, but no improvement in signal quality
- But no such limit exists for quantisation
  - More bits gives more information but …
  - 16 bits is probably enough.
- Summary
  - After sampling and quantisation, a waveform is simply a list of *numbers.*

Storing digital data

Once a computer receives a series of bits like the following:



The information about how to read a series of bits in a sound file is stored in the file header.

Sound files without a header are called raw sound files.

The Microsoft sound file standard .wav always has a header.

Storing digital data

How much disc space does a .wav file need?

This depends on the sampling rate and the number of quantisation levels (bits):

file size = sampling rate x number of bits

8 bits = 1 byte → 16 bits = 2 bytes

→ a file with a sampling rate of 44100 samples/second has: 44100 x 2 byte per second

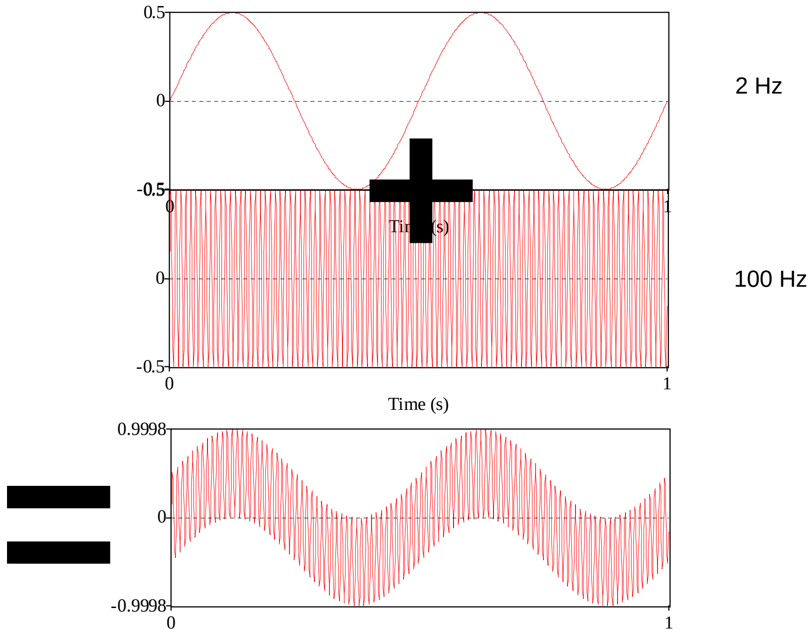
= 88200 byte
or 88.2 kbyte
or 0.0882 MB

How much space does the signal take if it is recorded in stereo?

Storing digital data

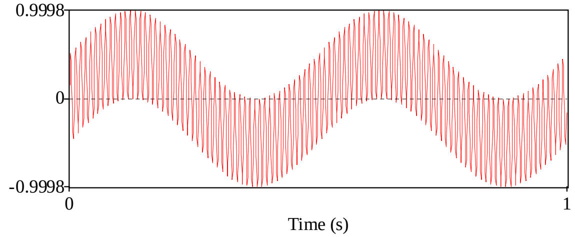
- .wav files take up a lot of space!
- MP3: Moving Picture Experts Group (MPEG) Audio Layer-III
  - a standard format for compressing digital audio, but it is a *lossy* compression
  - results in files about 10% of their original size, so that a 74 min CD can hold about 12 hours of music
- 'Perceptual encoding' …
  - uses a psychoacoustic model to remove parts of the signal that can't be heard, and then applies standard lossless data compression techniques.

Digital Systems

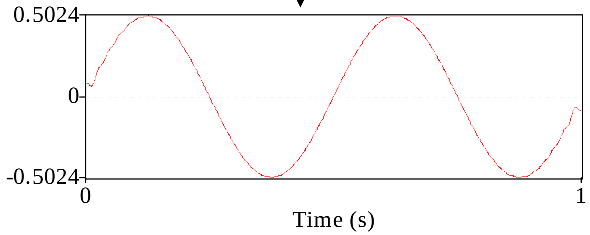


Digital Systems

MODIFYING DIGITAL SIGNALS

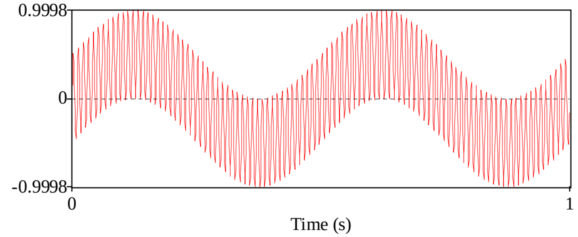


a.) Take ten samples
b.) Average samples
c.) Put average as first sample in out wave
d.) Move one sample forward



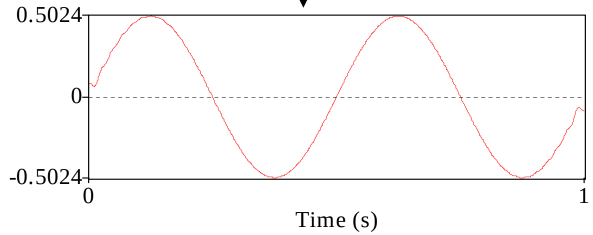
Digital Systems

MODIFYING DIGITAL SIGNALS

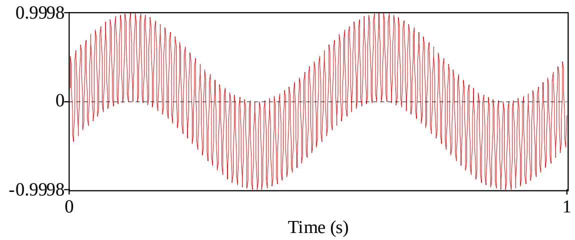


What system is this?

a.) Take three samples
b.) Average samples
c.) Put average as first sample in out wave
d.) Move one sample forward

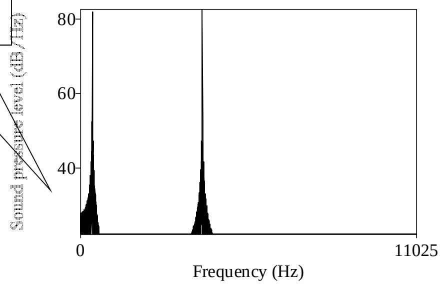


Fourier Transform of a digital signal



Fourier Transform

This is a discrete series of numbers too!



Fourier Transform

What you should remember:

DFT = Discrete Fourier Transformation

A special kind of Fourier transformation that is applied to discrete/digital signals

FFT = Fast Fourier Transformation

A fast version of the DFT, so generally used nowadays for the Fourier transformation in digital signal processing