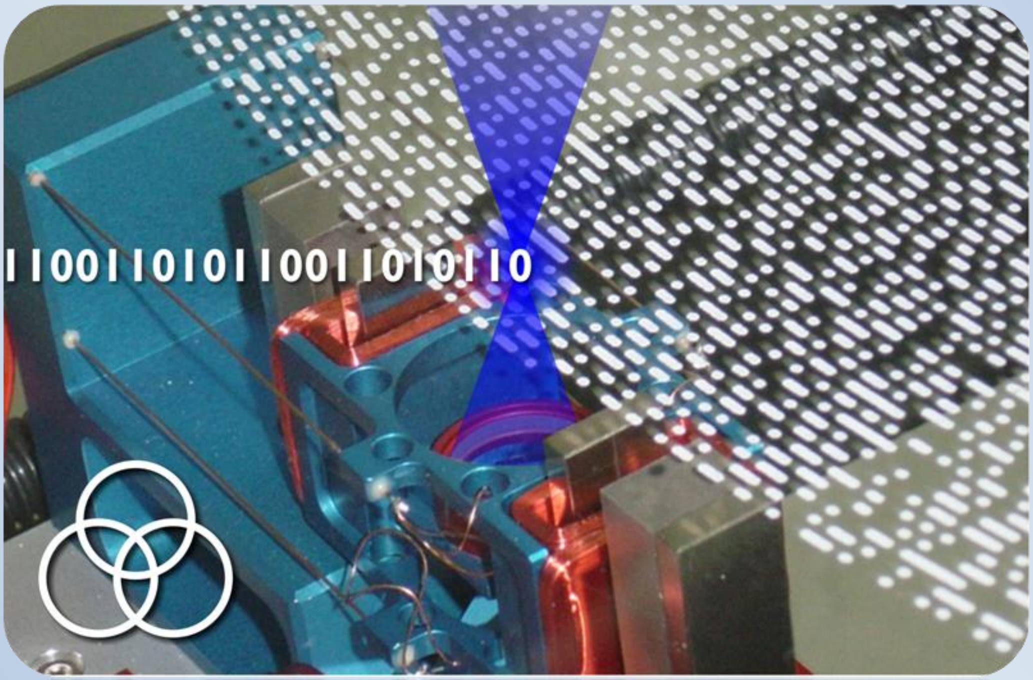
The Blu-ray Disc



An interdisciplinary assignment for secondary education
(physics, maths, electrical engineering, IT)

Jean Schleipen
Philips Research, Eindhoven, The Netherlands

European Schoolnet

inGenious

Many parties were involved in making this project available for schools:

This technology project was originally developed by Philips (The Netherlands) for the Dutch Jet-Net-project and incorporated in the EU 'Ingenious' project of European Schoolnet (EUN).

Jet-Net, the Dutch Youth and Technology Network, is a partnership between companies, education and government. The aim is to provide higher general secondary school (HAVO) and pre-university school (VWO) pupils with a true picture of science and technology and to interest them in a scientific-technological higher education course.

European Schoolnet (EUN) is a network of 30 Ministries of Education in Europe and beyond. EUN was created to bring innovation in teaching and learning to its key stakeholders: Ministries of Education, schools, teachers and researchers. The 'Ingenious' project is coordinated by European Schoolnet.

European Round Table of Industrialists

ERT is a forum bringing together around 45 chief executives and chairmen of multinational industrial and technological companies with European heritage with sales volume exceeding € 1,000 billion and thereby sustaining around 6.6 million jobs in Europe. ERT advocates policies at both national and European levels which help create conditions necessary to improve European growth and jobs. ERT was the initiating force for the EU Coordinating Body (ECB), now called 'Ingenious', to disseminate proven best practices of industry-school cooperation to stimulate interest in careers in science and technology throughout the European Member States.

This document is supported by the European Union's Framework Programme for Research and Development (FP7) – project ECB: European Coordinating Body in Maths, Science and Technology (Grant agreement Nº 266622). The content of this document is the sole responsibility of the Consortium Members and it does not represent the opinion of the European Union and the European Union is not responsible or liable for any use that might be made of information contained herein.

This is a publication of: Philips Human Resources Benelux / Jet-Net,
PO Box 80003, 5600 JZ Eindhoven, The Netherlands
Author: Jean Schleipen, Philips Research, Eindhoven, The Netherlands
Edition: English version 2.0, September 2012
for European Schoolnet 'Ingenious' programme 2-1-2012

CONTENTS

INTRODUCTION

People all over the world use optical data storage devices. You will of course be familiar with them: CD, DVD, Blu-ray Disc. The discs in the CD family are used for music, films, games, photos, computer files and so on. What happens if you put one of these discs into your gaming computer? Or into a DVD player or the disc drive of your computer?

In this teaching package you will be given a glimpse behind the scenes of optical data storage. How do you store music or a film on a disc and what is involved when one of these discs is played? We will start with a short history of data and how data has been transmitted and recorded over the years.

A short history

People have been recording data for a very long time. The oldest known cave drawings are around 40,000 years old. A lot has happened since those days. Information has developed from rock drawings to virtual environments; data carriers have developed from rocks to memory chips. Let's take a few examples.

Recording data

- *communications*:
  from picture writing in cave drawings to symbols such as cuneiform script, hieroglyphs, the modern alphabet, emoticons and the 0s and 1s by means of which our computers communicate;
- *audio*:
  from music box or organ book to gramophone, cassette and tape recorder, CD and MP3 player;
- *image*:
  from paintings and etchings to photography, film, video and 3D animations.

Storage media

- *communications*:
  from stone (*the Rosetta stone* for instance), clay tablets and paper to electronic media such as word processors, PDF and e-readers;
- *audio*:
  from metal (music box), cardboard (organ book) and vinyl (LP) to magnetic tape and discs, optical discs (CD, DVD and Blu-ray) and memory cards;

- *image*:
  from paper and canvas/linen to photographic film, magnetic and optical discs and memory cards.

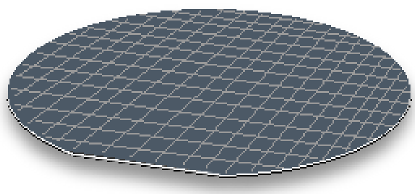
Transferring (and retrieving) data:

- from impossible (cave drawings);
- to passing from person to person (clay tablets, books, photos, …);
- to remote electronic communications (telegraph, telephone, satellite and fibre optics);
- to directly retrievable without any human intermediary with the advent of the Internet (YouTube, Google, iTunes, e-mail, *cloud computing*, …).

In the past the storage of images – such as paintings – was linked to paper or canvas; audio and film also had their own storage media. Modern digital storage media are suitable for all forms of information: communications, audio, image, ... The storage medium is no longer linked on a one-to-one basis to the nature of the data.

writing in stone

A lot has happened since rock drawings. Yet the majority of storage technology – on chips and memory cards – is still based on writing symbols (bits) in stone (silicon). The technique used for this is called lithography. This word is derived from the Greek: lithos=stone and grafein=drawing or writing.



Assignment 1

Make a timeline showing 10 examples of storage media. Make sure that at least half of your examples are from the last or present century.

Module
1

FROM ANALOGUE TO DIGITAL
(electrical engineering)

How do you record music, a film or a game on a small plastic disc? In this module you will learn how to convert a signal from the ordinary world, such as sound, to information that you can store on a CD, DVD or Blu-ray Disc. Before we do this we'll look at the terminology used in this field.

1.1 Terminology

In the computer world you come across many terms that are also applicable to CDs, DVDs and Blu-ray Discs. In order to avoid a confusion of tongues we will first discuss the most important terms.

Analogue

We call the signals in the world around us analogue. Light, for example, can be weak or very bright, with every gradation in between. Sound varies from quiet to loud. It is a sliding scale which – like a slide – can have all possible intermediate values. Our senses also work in an analogue fashion: our eyes and ears are able to perceive all those possible intensities of light and sound volume.

Digital

The analogue world can be in a virtually infinite number of states. However, to describe this world we use only a limited number of symbols, such as letters and numerals: the analogue information is digitised.
The word digital comes from the Latin (digita = fingers) and refers to counting with fingers. In a digital description information is expressed in a commonly agreed symbol language. A ***digit*** corresponds to a single symbol of this symbol language: digits are for instance the letters of the alphabet and the numerals (0 to 9) in our decimal system.

Binary

The word digital is often used in the context of computers. In that case a digit is the fundamental calculation unit for the computer. There are only two: the 0 (low voltage) and the 1 (high voltage). All information therefore has to be converted to a series of 0s and 1s.

This two-digit notation is called ***binary***. A digit in the binary system is called a ***bit***. A series of eight bits is called a ***byte***.

Digital or binary?

So we talk about digital when we're using symbol language (numerals, letters, …) to describe something and binary when that symbol language consists of only two symbols: 0 and 1. In everyday language usage, however, digital is virtually the only term used and then primarily in the meaning of binary. In this teaching module we will do the same. Confusing? Don't worry, we'll leave you in no doubt about what is meant.

Is a computer analogue or digital?

A computer calculates using 0s and 1s. So you'd think that a computer is binary. Strictly speaking that isn't true. A computer is also an analogue machine.
Nature and the world around us, so also including the workings of your computer, operate in an analogue manner. The voltage in a computer chip is an analogue voltage. It has been agreed that if the voltage applied to an electronic component is lower than a certain value we will call it '0' and if it is greater than a certain value we will call it '1'. So the alphabet for electronic calculation consists of two numerals: 0 and 1. The voltage in the computer chip can have many more values.

1.2 From analogue to digital sound

Sound is an analogue phenomenon. As you can see in Figure 1, an audio signal consists of pressure waves. In mathematical terms such a pressure wave can be described as a sine wave. These pressure waves impinge on our eardrums, which in turn start to vibrate. This is how we hear.

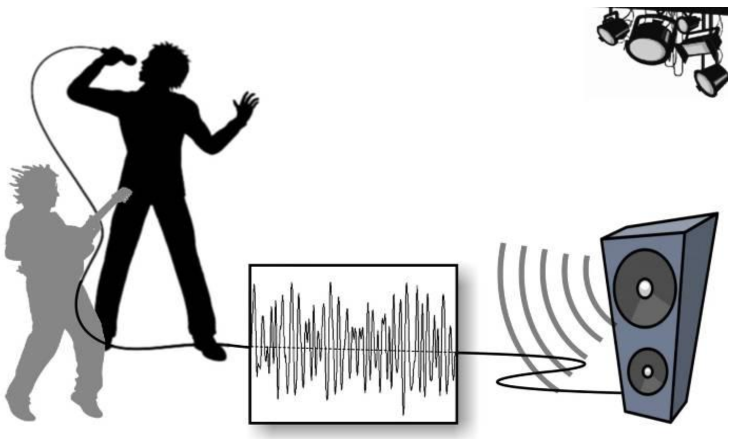


Figure 1. Sound is an analogue signal

If you want to record sound on a CD, this is not possible with sine waves. You can only write digital, binary signals to a CD: only ones and zeroes. So how do you convert an analogue audio signal to series of ones and zeroes?

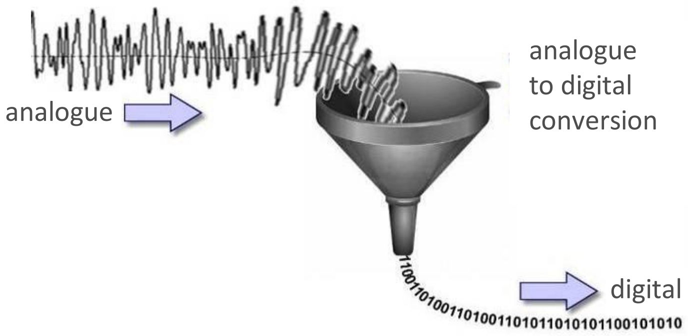


Figure 2. An analogue signal is converted to ones and zeroes

To digitise an analogue signal we perform three steps:

1. The analogue audio signal is first sampled: a snapshot of the signal is taken at regular intervals.
2. The values measured are then converted to a decimal number between zero and a previously agreed maximum value.
3. Finally, each decimal number is converted to a binary number.

We will focus on each of these steps in turn.

Step 1: Sampling a signal

Let's start with the electric signal from a sound recording. This signal consists of a large number of combined sine waves. A small piece of such a signal is shown in Figure 3. The voltage varies between two extreme values, in this case -4V and +4V. The red scale is time.

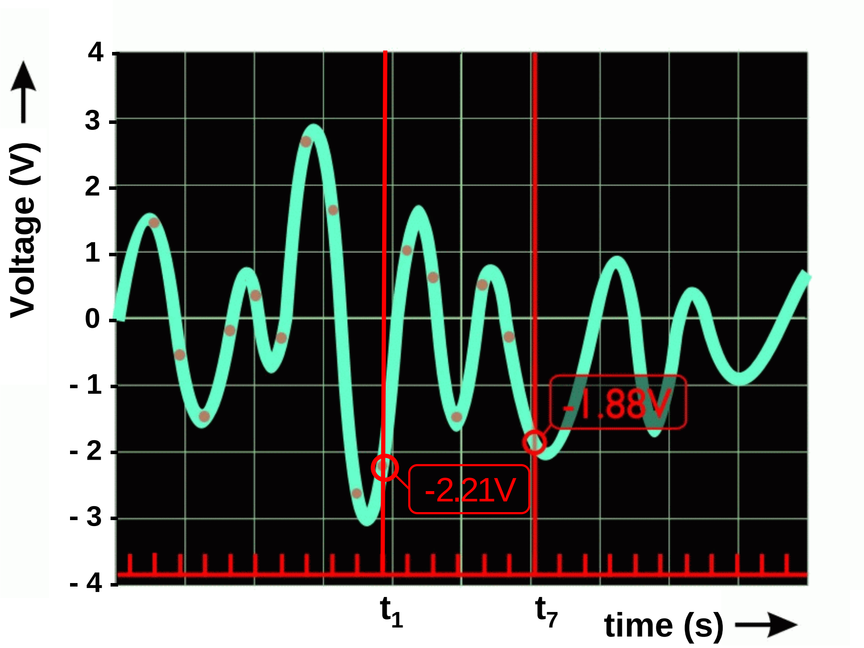


Figure 3. A small piece of an analogue music signal

To digitise the sound (make it binary), we measure the value of the voltage at regular intervals. At time t_1, for example, the voltage is -2.21 V. At t_7 the voltage is -1.88 V.
Performing these measurements at regular intervals is called ***sampling*** and the values measured are called ***samples***. A sample is taken at every red dot in the picture. The samples at t_1 to t_7 have the following values:

sample t_1 = -2.21 V
sample t_2 = 1.03 V
sample t_3 = 0.73 V
sample t_4 = -1.51 V
sample t_5 = 0.67 V
sample t_6 = -0.28 V
sample t_7 = -1.88 V

44100 samples per second

There is a rule that says that you have to sample a sine-shaped signal with frequency *f* with at least twice the frequency (2*f*) so as to correctly represent analogue information. This rule is called the *Nyquist theorem*.
The maximum frequency that we can hear is approximately 20 kHz. So for sufficient quality you need to sample at not less than 40 kHz. In 1980 Philips and Sony established a standard sampling frequency of 44.1 kHz for music CDs.
So to record music on a CD the voltage value of the music signal is measured 44100 times every second!

Step 2: From music signal to decimal number

We will now convert the voltage of a sample to an integer. As an example, take the sample at t_7. This has a value of -1.88 V. So we want to convert the value of -1.88 to a natural integer (0, 1, 2, 3, …). To do this we need the ***input range*** and the ***resolution***.

- *Input range*
  The input range is determined by the two extreme values of your music signal. In our example this range varies from -4 volts to +4 volts.

- *Resolution*
  The resolution is the precision with which you will shortly write the information to the CD. For instance: with 12-bit resolution, when you are using binary numerals made up of 12 digits, you can divide the range from -4 to +4 volts into 4096 steps (2 to the power of 12 = 2^{12}= 4096).

So you can convert all the values between -4 and +4 volts to a number from 0 to 4095:

-4V ⇔ 0
+4V ⇔ 4095

We can now convert the measured voltages to an integer.
To simplify the calculation we first shift all the values up by 4 volts. By doing this we can always perform calculations with numbers above zero. The values now vary between 0 and 8 volts. We then divide this into 4096 steps (0 to 4095):

0V ⇔ 0
+8V ⇔ 4095

Use the following steps to convert a voltage to a decimal integer:

1. Bring the value above 0
   add 4 volts
2. Calculate the relative point on the scale from 0 to 8 volts
   divide by 8
3. Multiply this value by the required resolution
   multiply by 4095 and round off to an integer

An example:

Volts scale of -4 to +4 V	**Volts** scale of 0 to 8 V	**Relative** scale of 0 to 1	**Decimal number**
-4	0	0	0
-3.5	0.5	0.0625	256
-3	1	0.1250	512
-2	2	0.2500	1024
-1	3	0.3750	1536
0	4	0.5000	2048
4	8	1.0000	4095

Step 3: From decimal to binary

As the final step we will convert the decimal values found to 0s and 1s: we will convert the samples to a ***binary*** number. We will first refresh our knowledge of binary numbers.

A binary number consists of ones and zeroes. A bit has a certain value depending on its place in the binary numeral: just like in our 'usual' decimal system, in which the digits correspond to ones, tens, hundreds, etc.

Binary numbers

In our decimal system the different places in a numeral correspond to powers of 10: 10^3 (thousands), 10^2 (hundreds), 10^1 (tens) and 10^0 (ones).

In the binary system this value is a power of 2. For a 4-bit binary number these powers of two have the following respective values: $\mathbf{2^3}$ $\mathbf{2^2}$ $\mathbf{2^1}$ $\mathbf{2^0}$.

So the first bit in a 4-bit numeral has a value of 8 ($\mathbf{2^3}$), the second a value of 4, the third 2 and the fourth 1.

So a binary numeral $b_3b_2b_1b_0$ corresponds in the decimal system to $b_3 x2^3 + b_2 x2^2 + b_1 x2^1 + b_0 x2^0$.

For example: **0110** $= 0x2^3 + 1x2^2 + 1x2^1 + 0x2$
$= 0x8 + 1x4 + 1x2 + 0x1$
$= 6$

Assignment 2

Verify that:
0101 = 5
0011 = 3
1000 = 8
1111 = 15
0000 = 0

Fill in:
1100 = …
0111 = …
… = 10
… = 4

Assignment 3

To digitise our audio fragment (represent it in binary form), you need to convert the measured voltage of the samples to a binary value. How precisely can this be done with a 4-bit binary number?

Fill in:
A 4-bit binary number can have …… different values.

The voltage of a sample is between +4V and -4V.
So we can divide this range into …… steps.

Each step then corresponds to …… volts.

With such large steps, of course, you cannot get a nice flowing sine wave if you want to write the values in binary form. In addition, you have to round off a value like -1.88V, resulting in a high degree of quality loss. So when converting an audio signal we use more bits.

Assignment 4

For an audio CD a resolution of 16 bits is used.

What is the largest power of 2 in a 16-bit number? ……….

How many different values can a 16-bit number have? ……………….

In the next assignment we will calculate using 12-bit binary numbers. The following table lists the decimal values of these binary numbers:

place	11	10	9	8	7	6	5	4	3	2	1	0
power of 2	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
value	2048	1024	512	256	128	64	32	16	8	4	2	1

Assignment 5

Verify for yourself that:

0010 0110 1100 = 512 + 64 + 32 + 8 + 4
= 620

1111 0000 1111 = 2048 + 1024 + 512 + 256 + 8 + 4 + 2 + 1
= 3855

Fill in:

1000 1000 1000 = ... + ... + ... = ...

0101 0101 0101 = ...

0000 0000 1010 = ...

1000 0000 0001 = ...

We can now decipher binary numbers as decimal numbers. What we need to do next is change decimal to binary. This is not quite so straightforward. Perform the following two steps to do this:

1. See which power of 2 is the largest that fits into your number.
2. Deduct this from your number and repeat step 1 with the remainder.

Repeat these steps until you have a remainder of zero.

An example:

You want to convert the decimal number 2643 to a 12-bit binary number. Keep noting down the largest power of 2 that fits into the remainder:

power of 2	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
value	2048	1024	512	256	128	64	32	16	8	4	2	1
binary	1	0	1	0	0	1	0	1	0	0	1	1

2643 – 2048 = 595
595 – 512 = 83
83 – 64 = 19
19 – 16 = 3
3 – 2 = 1
1 – 1 = 0

$2643 = 2^{11} + 2^{9} + 2^{6} + 2^{4} + 2^{1} + 2^{0}$
in binary form: **1010 0101 0011**

Assignment 6

Now practise making large binary numbers yourself. To do this you can use your calculator and the following auxiliary table.

4095 = _ _ _ _ _ _ _ _ _ _ _ _

3000 = _ _ _ _ _ _ _ _ _ _ _ _

512 = _ _ _ _ _ _ _ _ _ _ _ _

199 = _ _ _ _ _ _ _ _ _ _ _ _

1234 = _ _ _ _ _ _ _ _ _ _ _ _

Auxiliary table of binary numbers

2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
2048	1024	512	256	128	64	32	16	8	4	2	1

Once again, steps 1 to 3: From audio signal to binary number

So to convert an audio signal to a binary number we have performed the following steps:

1. sample the audio signal
2. convert measured value to a decimal number
3. convert decimal number to binary number

For example:

-4V	⇔	0	⇔	0000 0000 0000
+4V	⇔	4095	⇔	1111 1111 1111

Equipped with this knowledge we can now convert an audio signal to a series of binary numbers. These can then be written to a CD, DVD or Blu-ray Disc.

Assignment 7

Let's take the audio signal in Figure 3. Convert the values of the samples at t_1 to t_7 to a binary number. Use the following auxiliary table.

Volts scale of -4 to +4 V	**Volts** scale of 0 to 8 V	**Decimal number**	**Binary number**
-2.21	1.79	916	0011 1001 0100
1.03			
0.73			
-1.51			
0.67			
-0.28			
-1.88			

Auxiliary table of binary numbers

2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2048	1024	512	256	128	64	32	16	8	4	2	1

Assignment 8

You can also convert other data to binary numbers. Text, for instance. Fill in the following three steps in the table.

a. text: choose a three-letter word and write the three letters in the 3 columns in the following table;

b. from text to decimal: look up the corresponding decimal code for each letter (see 'Binary' worksheet in Annexe A);

c. from decimal to binary: convert the three decimal numbers to three 8-bit binary numbers.

text:			

text to decimal:			

decimal to binary:																								

Module
2

LOCATING AND CORRECTING ERRORS (maths, IT)

In Module 1 we saw how you can convert an audio signal to a series of zeroes and ones. These zeroes and ones are then written to a CD as pits. Greatly magnified, this looks as follows:

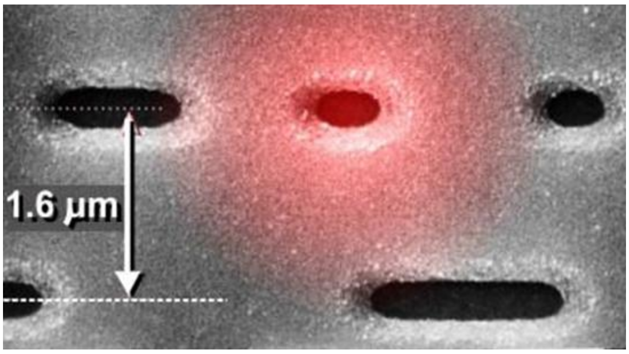


Figure 4. Magnified surface of a CD.

If you simply put the series of zeroes and ones on the CD, however, you can expect problems. If you look closely at a CD there are always some scratches, dust particles or finger-marks on it. So the chances of an error occurring while the pits are being read are enormous. And yet you don't notice this at all when you're watching a film on a DVD player or gaming with a gaming computer. How is that possible?

In this module you will be introduced to error encoding, a mathematical trick that locates and corrects errors on a CD. We will first take a look at how it works, then you can have a go yourself.

2.1 Hamming code

Say you play in a band and you want to record a CD of your favourite numbers. You all go off into the studio. The music is digitised after the recording. That great bass and that fast roulade are converted to a huge series of zeroes and ones (bits). A really tiny piece of music might look like this, for example:

1100 0110 1101 1100

If we write this series to a CD in exactly that form it is unlikely that we'll be able to play it back without errors. But if you add other bits in a clever way you can recognise and correct incorrect bits during playback. In this module we'll use the Hamming code to do this: this is a highly simplified little brother to the encoding technique used in a CD, DVD or Blu-ray Disc.

Encoding

For the Hamming code we split our digital music fragment into groups of four bits. These are the ***information bits*** (b3, b2, b1 and b0).
We then add three bits, known as ***parity bits*** (p3, p2 and p1), to each group of information bits:

1100*   0110***   1101***   1100*****

So our music fragment now consists of four numbers, each comprising seven bits b3, b2, b1, b0, p3, p2, p1. We call this kind of number a ***code word***.

To determine the value of the parity bits we use three circles and the following calculation rule:

- ***Each circle must contain an even number of ones***.

We fill in the circles as follows:

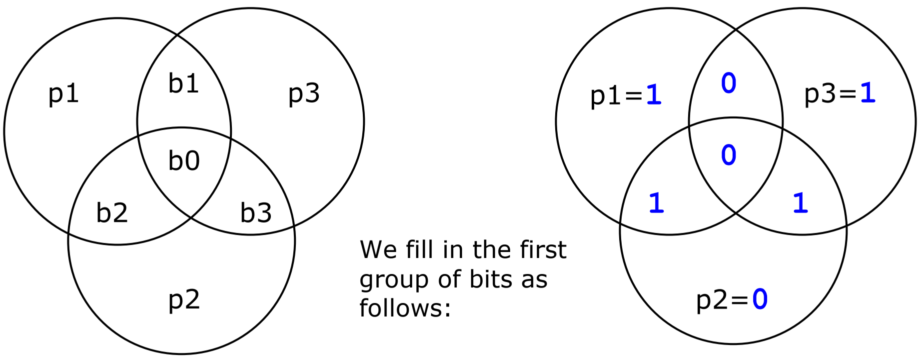


Figure 5. Determining the parity bits using the circle rule

The first code word in our example is therefore: **1100101**. Make sure the correct order is observed: b3, b2, b1, b0, p3, p2, p1. We write this code word to the CD.

Assignment 9

Determine the parity bits for the other three groups of bits in our example and write down all the code words for the music fragment here:

1100101   0110 _ _ _   _ _ _ _ _ _ _   _ _ _ _ _ _ _

Assignment 10

Check that 1010010 and 0001111 are valid code words (in other words: do they comply with the circle calculation rule or not?). Circle the correct answers:

1010010 is a valid / is not a valid code word
0001111 is a valid / is not a valid code word

Decoding

When we go to play the CD, we see that it has got dirty. Some small dust particles are preventing the pits from being read properly. We now get the following code words (compare them with the original code words that you determined in Assignment 9):

1000101   1110110   1101110   0100101

Using the three circles and the '*even number of ones in each circle*' rule, we can check whether an error has occurred during reading.

We fill in the first code word in the circles.

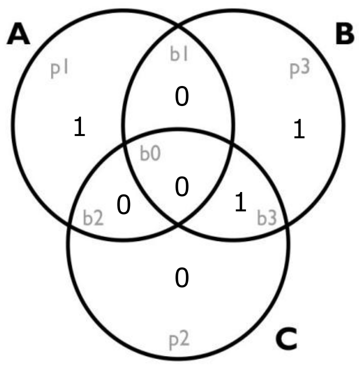


Figure 6. Checking a code word using the circle rule

We now see the following:

- Circle B complies with the rule that it must contain an even number of ones. So this circle does not contain an error.
- Circle A and circle C both contain an odd number of ones. So the error must be in the segment that is common to these two circles alone. This contains information bit b2. So we have to change the 0 that is currently in it into a 1.

	information bits				parity bits		
	b3	b2	b1	b0	p3	p2	p1
code word before correction	1	0	0	0	1	0	1
code word after correction	1	1	0	0	1	0	1

Assignment 11

Check the other three code words in the example in the same way and correct them if necessary. Which bits were incorrect? Circle the incorrect bits in the code words below (each code word contains not more than 1 incorrect bit).

Do you see the original code words from Assignment 9?

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

The code words we are using in our example comprise seven bits: four information bits and three parity bits. We call this a [7, 4, 3] code.

2.2 Interleaving

Using the [7,4,3] Hamming code you can locate and correct a maximum of one error in a code word. Can you work out why this is? If there is a scratch or a bit of dirt on a CD there is of course a good chance that an entire row of consecutive bits will be read incorrectly. A technique has therefore been developed in which the risk of errors is literally spread. With this technique the code words are first 'cut up' into separate parts and spread across a larger area on the disc during writing. This technique is called ***interleaving***. What this amounts to is that all the initial bits in a set of code words are first written in succession, then all the second bits, etc.

Our original code words:

1100101 0110110 1101010 1100101

are then written to the CD as follows:

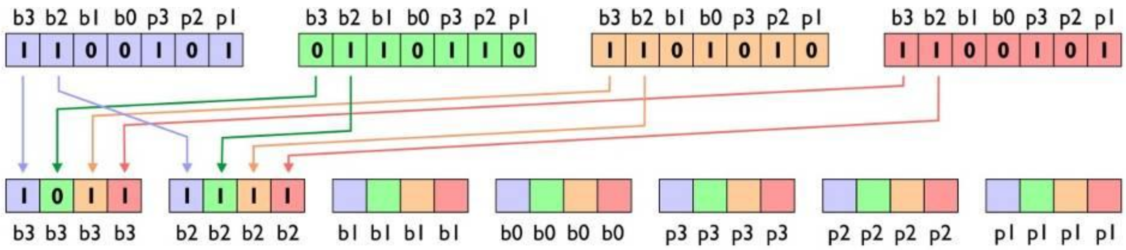


Figure 7. Writing code words by means of interleaving

Assignment 12

Write the code words from Assignment 9 using interleaving:

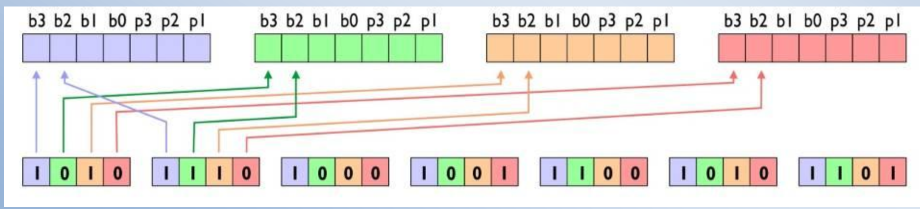
1011 1111 _

Assignment 13

The following series of bits has been recorded on a CD using interleaving. Write down the code words that make up this series. Use the following figure to do this.

1010 1110 1000 1001 1100 1010 1101

_ _ _ _ _ _ _   _ _ _ _ _ _ _   _ _ _ _ _ _ _   _ _ _ _ _ _ _



You have had some practice with the Hamming code and interleaving. We will now set to work seriously!

Assignment 14

'Make your own CD'

In this assignment you will yourself 'record' a CD, which will then be 'played' by a fellow pupil.

For this assignment use the 'Binary' and 'Parity table' worksheets and worksheets 1, 2 and 3 (Annexes A to E).

Part 1: Recording (note down your data in worksheet 1)

1. Think up a (secret) 5-letter message or word.
2. Digitise your message using the 'Binary' worksheet.
3. Split your message into 10 groups of 4 bits.
4. Now add the 3 parity bits to each group; to do this, use the Hamming code and the 'Parity table' worksheet.
5. Using interleaving, place all the initial bits one after the other, then all the second bits, etc. You then have a total of 70 bits.
6. Write the bits to the CD in the order they are now in (worksheet 3). Begin at START.

   NOTE: As you can see, there is a large scratch on the CD. **At the point where the scratch is, replace your information (0 or 1) by a 0.** Make sure that the original, correct bits are not visible. So don't scratch or rub off!

7. Cut out your CD and give it to a fellow pupil, who can then play your message. You can play a fellow pupil's CD yourself.

Assignment 14 - continued

Part 2: Playback (note down your data on worksheet 2)

8. Reconstruct the code words that were written to the CD using interleaving. You should now have 10 code words that are 7 bits in length.
9. Use the Hamming code to check whether there are any errors in the code words and correct them if necessary. Use the 'Parity table' worksheet to do this.
10. Remove the parity bits so that there are 10 groups of 4 bits remaining.
11. Keep combining two groups of 4 bits until there are 5 groups of 8 bits remaining.
12. Find the corresponding letter for each group in the 'Binary' worksheet.
13. What is the message that you have deciphered? Is it the same message that your fellow pupil encoded?
14. If you have any time left, or if you feel like doing this, try to decipher the message without correcting any incorrect bits! You will now get utter nonsense…

With this Hamming code you can only identify and correct one error per code word. The codes for error correction that are used nowadays are much more advanced and require much more calculation time. Your gaming equipment contains chips that have been specially designed to find and correct errors at lightning speed using one particular code. However, the principle behind this is the same as what you have already done in this module!

You can find this mathematical method of error encoding in all systems in which digital information is read or transferred and nowadays we could scarcely imagine daily life without it: Internet, Facebook, Skype, hard discs, computer memories, USB memory sticks, digital TV, online banking, iTunes, cloud computing, to mention just a few examples …

Module
3

READING AND WRITING WITH LIGHT (physics)

A music CD has a storage capacity of 750 MB (MB = Megabyte = 1024^2 bytes; 1 byte = 8 bits). A DVD can hold 4.7 GB and a Blu-ray Disc as much as 25 GB per layer and with a clever trick two or more layers can be written (GB = Gigabyte = 1024^3 bytes). In principle the discs are exactly the same size. So what determines the storage capacity of a disc?

In this module you will learn how to calculate the capacity of a disc. An important factor in this is the size of the laser spot that 'reads' the pits on the disc. How this has affected the development from CD to Blu-ray Disc will be discussed in a short piece of history.

3.1 The size of a laser spot

The size of the smallest possible structure that you can read using a laser beam is determined by the size of the focused laser spot. Figure 8 shows the magnified surface of a music CD. The red area corresponds to the focused light from the laser with which the CD is read.

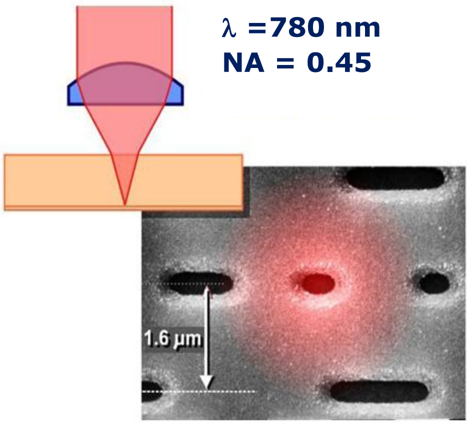


Figure 8. A highly magnified laser spot on the surface of a CD

Even if we could make a perfect lens that sends every 'light beam' through one point, the phenomenon of ***diffraction*** of light waves would still cause this 'point' to have a finite size. The amount of diffraction is determined by (i) a characteristic of the lens that is expressed in a dimensionless number (NA - numerical aperture), and (ii) the colour of the laser light used.

The diameter *w* of the focused, diffraction-limited laser spot is given by:

$$w = 1.22 \cdot \frac{\lambda}{NA}$$

where *w* = the diameter of the laser spot
λ = the wavelength of the laser light used
NA = the numerical aperture of the lens used

The wavelength of the light determines the colour that we perceive the light as: red light has a wavelength of approximately 650 nm, while for blue light this figure is 450 nm (1 nm = 0.001 µm = 10^{-9} m).

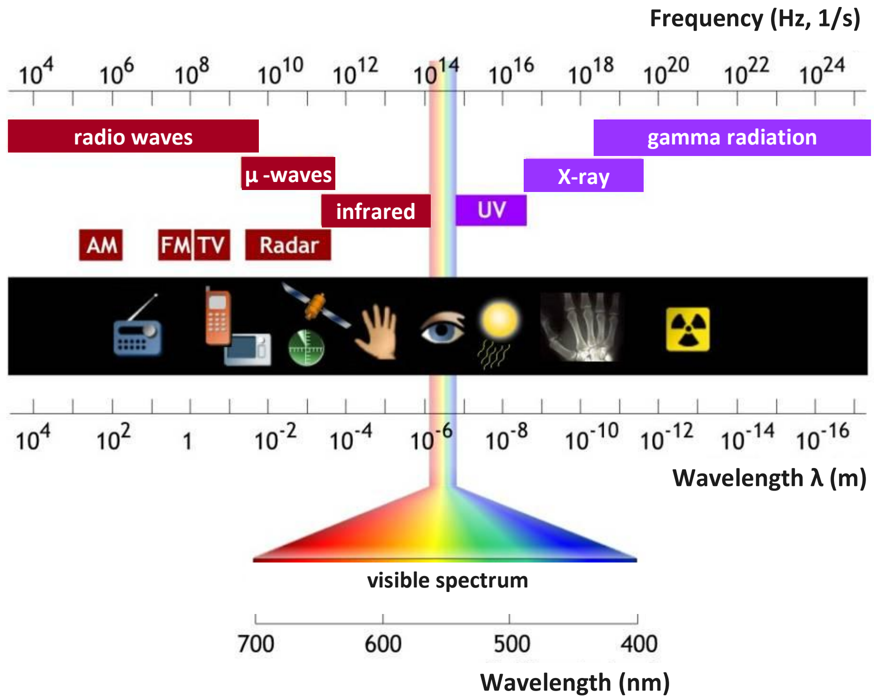
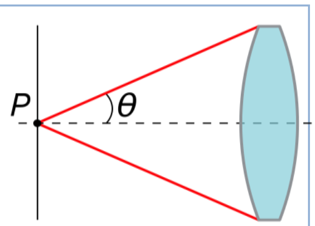


Figure 9. The electromagnetic spectrum: the spectrum with visible light is roughly in the centre

What is the maximum value of the NA?

The *numerical aperture* of a lens is equal to the sine of half the aperture angle θ of the focused beam: NA=sin(θ). The larger the angle θ, the larger the numerical aperture (NA). The maximum value for NA is 1, since the sine of an angle can never exceed 1.



If the beam of light is focused in a medium with *refractive index n*, then $NA = n \cdot \sin\theta$. This is the case for an immersion microscope. Here there is a thin layer of water (where n=1.33) or oil (where n>1.6) between the object and the objective lens. Aim: the NA can then be greater than 1, resulting in a smaller *w* and therefore larger resolution.

3.2 The development from CD to Blu-ray Disc

From CD ...

In the '70s of the last century, when the compact disc system was being developed, compact infrared semiconductor lasers were available with a wavelength of 780 nm. At that time it was possible to make inexpensive, diffraction-limited lenses with a numerical aperture of 0.45.
For microscopes objective lenses with a far larger NA had been available for a long time, but they were much too expensive for a consumer product. Typically, a lens for a CD, DVD or Blu-ray player must not cost more than €1!
With the available laser and lenses it was possible to achieve a storage capacity of 750 Mb.

Beethoven's 9th is standard for CDs

The storage capacity of a compact disc is 750 MBytes. This figure didn't just appear out of the blue. When the CD was being developed there were several important ambassadors for this new storage medium. One of them was Herbert von Karajan, at that time the conductor of the Berlin Philharmonic Orchestra and an icon in the world of classical music. Von Karajan wanted Beethoven's 9th symphony to fit on a single disc.

Beethoven's 9th lasts for about 74 minutes. To make it possible to record the dynamics of classical music (quiet and loud passages) precisely enough, the engineers at Philips and Sony opted for a 16-bit sampling method: every sample is represented by a 16-bit number. In addition, two channels had to be recorded (stereo). You can now calculate how many bits of memory capacity are needed for this. You're going to do this yourself shortly in the assignments.

… to DVD …

The music CD proved to be a success and then people also wanted to be able to put films on this shiny disc. But this required greater capacity than the 750 MBytes that a CD can hold. So in the '80s the large industries and research institutes searched assiduously for inexpensive semiconductor lasers with a shorter wavelength. When a red semiconductor laser came onto the market in the early '90s, development of the DVD system was quickly started. The manufacturing process for diffraction-limited lenses had by now been greatly improved, so that it was also possible to make lenses with a larger numerical aperture. In 1996 the DVD system came onto the market, with a laser wavelength of 660 nm and a numerical aperture of 0.6. This gave a storage capacity of 4.7 GBytes, enough for 1 hour of film, with an image quality that was far better than the VHS magnetic tape video recorders available at the time.

… to Blu-ray Disc

But this still wasn't enough. In the '90s at lot of work was done on improving the resolution of TV pictures. This resulted in HD TV, *high-definition TV*, the successor to the very outdated PAL TV system. At the same time better screens with a higher resolution came onto the market. HD TV images were five times sharper than the old PAL system. But this also meant that an HD TV film required five times more storage capacity.

So in the '90s the search for an inexpensive laser with an even shorter wavelength continued unabated. In 1996 a blue-violet laser with a wavelength of 405 nm was developed. By also increasing the NA from 0.6 to 0.85 it was possible to achieve a 500% increase in storage capacity. In 2006 the Blu-ray Disc system was introduced, with a laser wavelength of 405 nm, an NA of 0.85 and a storage capacity of 25 GBytes per layer.

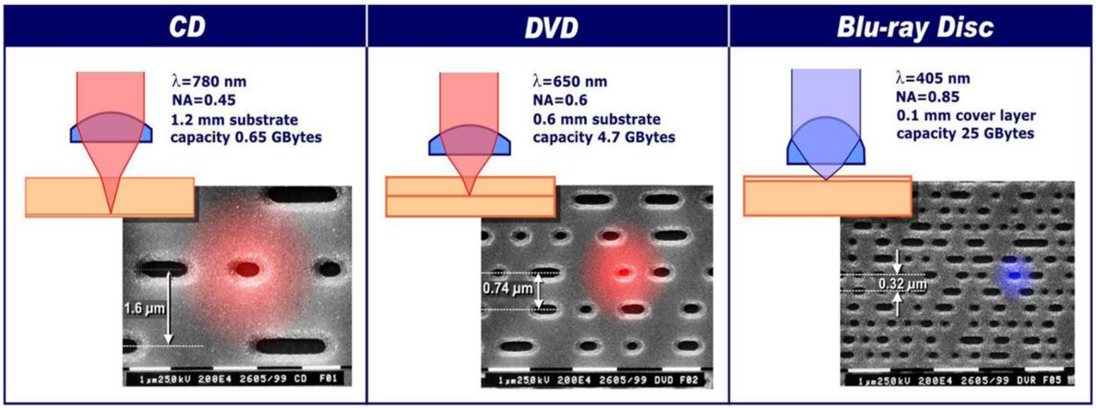


Figure 10. The CD, DVD and Blu-ray Disc with an ever smaller laser wavelength λ and an ever larger numerical aperture NA

Memory 'under the bonnet'

An audio CD can hold 750 MBytes of data. The first data CDs, such as CD-ROM and CD-RW, had a storage capacity of only 650 MBytes. This was due to a greatly improved *error correction* system that is required in order to correct errors during reading. But why do we need a more powerful error correction system for data than we need for music?

It transpires that reading errors in music are not so bad: to a certain extent they can be cleaned up by means of filters and interpolation techniques. With data, such as text, that isn't possible: a CD mechanism can't decide whether an incorrectly read word '*bun*' should really have been '*bus*'. *We* know that the sentence '*The bun stops at quarter past two*' is nonsense, but a CD player doesn't.

Assignment 15

In this assignment you will make an estimate of the amount of data that you can write to an optical disc.

a. Calculate the diameter *w* of the laser spot for the CD, DVD and Blu-ray Disc system. While doing this, pay attention to the units used: 1 nm $=10^{-3}$ μm = 1/1000 μm = 10^{-6} mm.

w_{CD} = ...
w_{DVD} = ...
$w_{Blu\text{-}ray}$ = ...

b. Calculate the surface area of the laser spots and make an *estimate* of the number of bits that a disc can hold. To do this, divide the surface area of the active part of the disc that contains the information (starting at radius 22 mm and ending at radius 58 mm of the disc) by the surface area of the laser spot.

number of bits on CD = ...
number of bits on DVD = ...
number of bits on Blu-ray = ...

Assignment 16

In the next fill-in exercise calculate what the capacity of a CD must be in order to store 74 minutes of **music**.

A music CD has a playback time of 74 minutes.	This is .... sec.
The music is sampled at 44100 samples/sec.	So for 74 min of music .... samples are required.
For stereo sound two channels are recorded simultaneously.	So altogether we have .... samples.
The resolution at which the samples are written to the disc is 16 bits/sample.	This gives a total of .... bits.
1 byte = 8 bits	This corresponds to .... bytes.
1 kbyte = 1024 bytes 1 Mbyte = 1024 kbytes (so 1 MB = 1 Mbyte = 1024^2 bytes)	Storing 74 minutes of music on a CD therefore requires .... MB of storage capacity.

Assignment 17

Now do the same fill-in exercise for 1 hour of **film**.

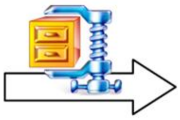
The film lasts for 1 hour.	This is .... sec.
A film is played back at 25 frames/sec.	This gives .... frames.
Each frame consists of 576 lines of 720 pixels each: the dimensions for the current PAL TV system.	In total, then, there are .... pixels.
The resolution at which the brightness of a single pixel on the disc is written is 8 bits/pixel.	This gives a total of .... bits.
For each pixel you have three colours: red, green and blue	In total, then, there are .... bits.
1 byte = 8 bits	This equals .... bytes.
1 kbyte = 1024 bytes 1 Mbyte = 1024 kbytes 1 Gbyte = 1024 Mbytes (so 1 GB =1 Gbyte = 1024^3 bytes)	Storing 1 hour of film requires .... GB of storage capacity.

Data compression

One hour of film thus requires much more storage capacity than the 4.7 Gbytes that a DVD disc can hold. To solve this problem ***data compression*** is used, a mathematical technique with which you can reduce the size of a digital file.

Several compression techniques exist. ZIP compression, for instance, is a well-known technique used in the computer world. For compressing video images (e.g. MPEG), *lossy compression* is used. With this method information that is not directly visible to the observer is lost and you can achieve large compression ratios. With the lossy compression used for DVD you can reduce digital video files by a factor of 30. A well-known lossy compression technique for music is mp3. With mp3, frequencies and frequency combinations which the human ear cannot perceive are removed from the music signal. The remaining signal is then digitised.

Music
5 minutes,
CD-quality
60 MBytes
WAV file

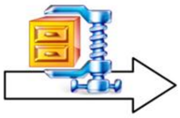


Music

4 MBytes

MP3 file

Photo
10 Mpixel camera
30 MBytes
BMP, RAW file

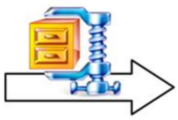


Photo

1 MByte

JPG file

Film
2 hours,
DVD quality
> 100 GBytes
AVI file



Film

4.7 GBytes

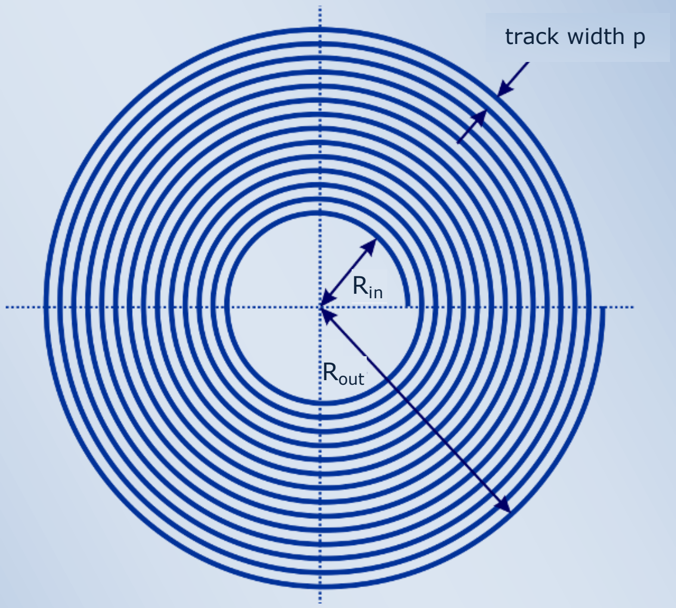
MPEG2 file

Assignment 18

Any library worth its salt will have at least around 100,000 books. A book contains an average of 300 pages and each page has around 4000 letters on it. We need 1 byte of storage memory per letter. We use ZIP compression, in which files can be reduced to 15% of their original size. How many Blu-ray Discs (25 GBytes) do we need to be able to store all these books?

Assignment 19

The individual bits on a CD, DVD or Blu-ray Disc are written along a spiral. The distance between two adjacent tracks in this spiral, the track width p, is p=1.6 µm for a CD. One µm is 10^{-3}=1/1000 of a millimetre.



a. How many tracks are adjacent to one another over a distance of one mm?

b. How long is the total track on a CD?

To get an idea of the total length ***L*** of the CD track, you can use a clever calculation trick. As we will show below, you can closely approximate the length ***L*** of the spiral by using the formula:

$$L = \frac{\pi \, (R_{uit}^{\;2} - R_{in}^{\;2})}{p}$$

The minimum and maximum radii of the spiral are R_{in} = 23 mm and R_{out} = 58 mm for CD, DVD and Blu-ray Disc respectively. The track widths p for the various systems are:
p_{CD} = 1.6 µm, p_{DVD} = 0.74 µm and $p_{\text{Blu-ray Disc}}$ = 0.32 µm.

Now calculate the total length of the data track for a CD, DVD and Blu-ray Disc (pay attention to the units used):

L$_{CD}$ = 3.14 x (………… - …………) / ………… = …………

L$_{DVD}$ = 3.14 x (………… - …………) / ………… = …………

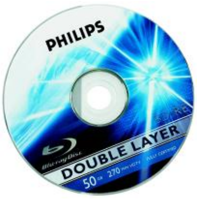
L$_{\text{Blu-ray Disc}}$ = 3.14 x (………… - …………) / ………… = …………

Assignment 20

How many bits now fit on a Blu-ray Disc? We can calculate this exactly as follows.

To do this, use the length of the track on a Blu-ray Disc that you calculated in the previous assignment.

Divide the total length of the track by the bit length. The physical length of 1 bit is 75 nm for the Blu-ray Disc system. The fact that this length is less than the size of the laser spot is due to the *channel coding*, which we will say more about shortly.



So a Blu-ray Disc can hold …………… / ……… = …………………….. bits.

N.B.: The number of bits calculated above is the total number of *channel bits* (see section 3.3)! So this includes all the extra information, such as bits required for error correction, bits for address information, for timing information, etc…

The length of a spiral, *approximately*

Regard the spiral as a collection of N concentric circles, with radii of R_1 to R_N. The distance between two adjacent circles is the same as the track width p. The circumference O_i of circle i with radius R_i is $2\pi R_i$.

The total length ***L*** of the spiral then equals the sum of the circumferences of all circles O_i, where i=1 to N:

$$L = \sum_{i=1}^{N} O_i = \sum_{i=1}^{N} 2\pi \cdot R_i$$

We can easily calculate this sum if we bear in mind that the sum of the circumferences of the first and last circles equals the sum of the circumferences of the 2nd and the last but one circles:

$$2\pi \cdot R_1 + 2\pi \cdot R_N = 2\pi \cdot (R_1 + p) + 2\pi \cdot (R_N - p)$$

and this equals the sum of the circumferences of the 3rd and the last but two circles, etc. So the sum of the circumferences of all the circles equals half the total number of circles times $(2\pi \cdot R_1 + 2\pi \cdot R_N)$, or:

$$L = \frac{1}{2} \cdot \frac{R_N - R_1}{p} \cdot 2\pi \cdot (R_1 + R_N) = \frac{\pi \cdot \left(R_N^2 - R_1^2\right)}{p}$$

In our case R_1 = 22 mm, R_N = 58 mm and p = 320 nm.

The length of a spiral, *exactly*

For the whizz-kids among us, and only if you've already covered the term integration, you can calculate the length of a spiral exactly by means of *integration*. As we move along the spiral the distance r from the track to the centre of the spiral will gradually increase as a function of the spiral angle ϕ (this is the definition of a spiral).

When we start (so when $\phi = 0$), r=R_1. For every complete revolution ($\phi \to \phi + 2\pi$) a distance p (the track width) will then be added:

$$r(\phi) = R_1 + \frac{\phi}{2\pi} \cdot p \; .$$

The total number of revolutions is $N_\phi = (R_N - R_1)/p$.

The length of a small arc segment at distance $r(\phi)$, with angle $\Delta\phi$, is $r \cdot \Delta\phi$. The total length of the spiral is then equal to the sum of all of these arc segments:

$$L = \sum_{\phi=0}^{N_\phi \cdot 2\pi} r(\phi) \cdot \Delta\phi = \int_{\phi=0}^{N_\phi \cdot 2\pi} r(\phi) \cdot d\phi = \int_{\phi=0}^{N_\phi \cdot 2\pi} \left[R_1 + \phi \cdot (p/2\pi)\right] \cdot d\phi$$

Calculate this integral and compare the answer with the approximate value for the spiral length.

3.3 Digital channel coding

You have calculated how many bits the various discs can hold. A trick can be used to put even more bits on a disc. We call this trick ***digital channel coding*** and it is used, for example, in optical telecommunications through a glass fibre (as is used for the Internet) and for transferring digital TV via satellite.
Shown below is the surface of a CD when burned without channel coding. Each bit is allocated a separate pit. Since we have so many ones and zeroes mixed together for a series of data, the surface has a very large number of small pits. This is not so cost-effective. It is also tricky from a technical point of view to read these small separate pits without errors.

Figure 11. Pits in the surface of a CD

With digital channel coding the information is converted to series of longer pits. These are not only easier to read, but can also hold more information than a row of separate pits. In addition, by looking very carefully at the *timing* of the pits that come past we can read and write to disc bits that are in fact smaller than the spot size of the focused laser beam. How does this work?

To be quite clear: the length of the shortest pit that we can read is still determined by the size of the laser spot. This is something we simply cannot get round.
If we call the smallest possible pit a, a single bit, **1** or **0**, would therefore correspond to a piece of data track with length a. A bit series such as **111** would then correspond to a piece of data track with length 3a. And for the bit series **0111** you would therefore need a length of 4a on the disc. For a Blu-ray Disc

$$a = 0.61 \cdot \frac{\lambda}{NA} = 0{,}29\ \mu m$$

But do we really have to write the bit series **111** as aaa? No! As long as the corresponding pit is larger than a (the *resolution* of the reading system), we can accurately measure the length of the pit.

N.B.: Attentive readers will have noticed that we are suddenly using a=0.61 x λ/NA, whereas previously we saw that the size of the laser spot is w=1.22 x λ/NA. This is due to the fact that the ***resolution*** of an optical system equals half the spot size.

Precise timing

We are now going to look carefully at the timing of the signals measured. With a sufficiently precise system it is best to make a distinction between a pit with a length a and one with a length of, say, 1½ a.

We can then agree as follows:

- we will write a single separate bit with length a;
- we will write two identical bits, for example **11** or **00**, as a pit with length (1+½)a=1½a;
- we will write three consecutive identical bits, **111** or **000,** as a pit with length (1+½+½)a=2a; etc.

This trick is called ***digital channel coding***.

What benefits does this give us? To get an idea of this, let us look at the following series of 40 bits that we are going to write to a disc:

1100010110101111000110000111000101111101

If we ***don't* use channel coding**, and so use a piece of data track with length a for every individual bit, we need a length of 40a on the disc in order to write all the bits. This is shown in the following figure. On the top row you see the series of bits, while on the bottom row you see the pits on the disc.

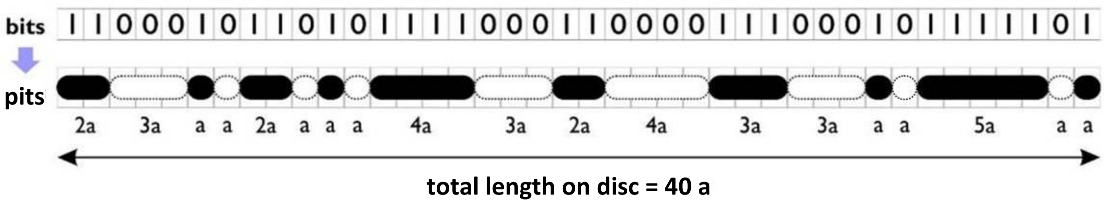


Figure 12. Writing binary numbers without channel coding

How much space do we need to write the same bit series ***with*** **channel coding?** To do this we first convert the series of bits to a series of numbers indicating how many identical bits occur consecutively, known as ***run lengths***.

First we have two **1**s: this gives a run length of 2;
then three **0**s: run length 3; etc.
The sum of all the run lengths for this bit series must of course be 40, the total number of bits. Just check.

To make it easier to calculate, we will first define the term ***T cycle***. One T cycle is the length of time that a single *channel bit* (about which more shortly) takes to pass under the laser spot. The smallest unit that we write to the disc, a pit with length a, is now assigned 2 T cycles:

$a := 2T$, or $T := \frac{1}{2}a$.

Next we apply the channel coding: we write a run length of 1 (1 single user bit) with the minimum permitted length of 1a = 2T (it cannot be smaller than this because the optical resolution is a). We sneakily make all run lengths that are greater than 1 slightly shorter using the following formula. The number of cycles N_T for a particular run length R is given by:

$$N_T(R) = R \cdot a - (R-1) \cdot \frac{1}{2}a \quad = \frac{1}{2}(1+R) \cdot a \quad = (1+R) \cdot T$$

So a run length of, say, R=5 now becomes 6T = 3a long. Without channel coding this same pit would be 5a long! We are therefore making a gain in storage capacity of 40%!

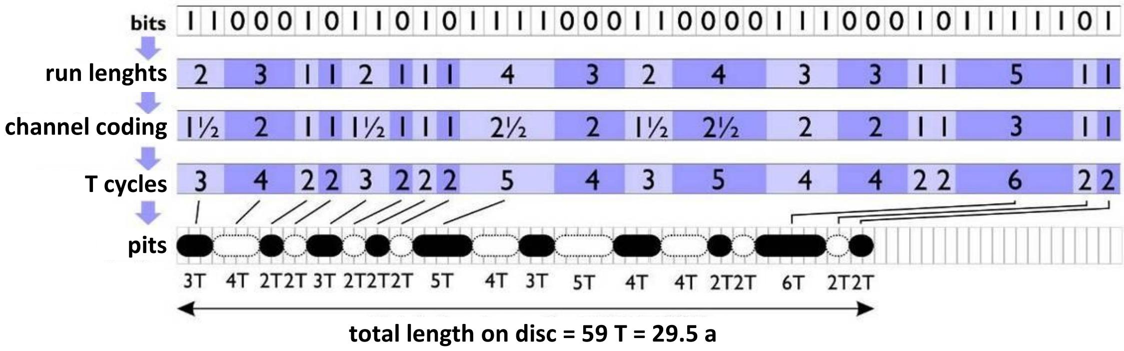


Figure 13. Writing binary numbers with channel coding:

Channel coding

This sounds very nice, but we're not there yet. A random pattern of user bits will largely consist of bit series such as '…0101010…'.
So run length R=1 occurs fairly frequently. But for run length R=1 we have no gain in storage capacity (since the series is not shorter and stays at a). To solve this problem, more advanced channel coding methods have been devised, in which the original series of user bits is first converted to a new series of bits, with an optimised distribution of run lengths.

For Blu-ray Discs it has been agreed that only run lengths will be used with lengths ranging from 2 cycles to 8 cycles, i.e.:
$2T \leq a \leq 8T$.

An upper limit has been set here for the maximum permitted run length so as to avoid an error being made during reading. To understand the reason for this, let's say the *precision* with which we can determine the length of the cycles is 10% of the T cycle. When reading a long pit of, for instance, 20 T cycles, a reading error of
20 x 10% x T = 2 T can occur. In other words: we have made an error of 2T! Where we really ought to have read a length of 20T we are measuring a length of 18T due to, say, noise. In order to avoid this type of error it has been agreed that the pits for Blu-ray Discs, etc., must not be longer than 8 cycles.

From user bits to channel bits

We will now convert the user bits to what are called channel bits. To do this we use a coding table. Below we give the coding table for what is is known as a 17-channel code (minimum of (1+**1**)T, maximum of (1+**7**)T).

User bits	Channel bits
00 00	101 000
00 01	100 000
10 00	001 000
10 01	010 000
00	101
01	100
10	001
11	010

How do you go about doing this?

1. Take our series of user bits and scan it slowly from left to right.
2. If you find a pattern in the user bits that is the same as a pattern in the left-hand column of the table, replace these user bits by the corresponding channel bits in the right-hand column.
3. Do the same for the remaining user bits until they have all been encoded into channel bits. If there are several possibilities, take the highest row in the table. For example: the user bits **10 01** are encoded as **010 000**, and not as **001 100**.

The new series of channel bits is interpreted as follows. Whenever a **1** comes past, the laser switches from reading to writing or from writing to reading. In other words: whenever a **1** comes past, we have a change from a pit to a non-pit or from a non-pit to a pit. You find the run length (in cycles) of the pits or non-pits by counting the number of zeroes between two consecutive **1**s in the series of channel bits and adding 1. You will find that regardless of which pattern of user bits you present, there will always be a series of channel bits with a minimum run length of 2T and a maximum run length of 8T.

Assignment 21

(i) Encode the following series of user bits using the above 17-channel coding table (the first part has already been done):

10-10-1001-01-100000010010111101000100101110010110

is encoded as:

001 – 001 - 010 000 – 100 - ………………………………………………

(ii) Calculate the run lengths in cycles for the channel coding calculated in (i). The run length is equal to 1 + the number of "0"s between two consecutive "1"s. Complete:

3T-3T-2T-5T- ………………………………………………

(iii) These run lengths ultimately determine the space required on the disc; as you know, T equals 1/2 a. For the bit series in (i), calculate the length required along the disc track. Do this also for the situation in which you do not use channel coding and every user bit corresponds to length a. And count your winnings!

Module 4

FOCUSING AND TRACKING

(measurement and control engineering, electrical engineering)

You see before you a large model of a real DVD player. Why is this a DVD player? The model uses the same red laser as in a real DVD player. And yes, this model really works, too! Using this scale model you will learn how a DVD player can read all those data on a disc so quickly.

As you know, a CD, DVD or Blu-ray Disc contains data written along a spiral in the form of pits. The laser recognises these pits because the laser beam is reflected onto a detector whenever a pit comes past. If no pit comes past, less light impinges on the detector.

4.1 Unwanted movements

The trick is of course to keep following the track of pits. And this track is already so narrow! It's just the same as with the old-fashioned record player: if the ‘needle’ goes outside the right groove it no longer sounds right.

Horizontal movement

To give you an idea of how precisely this track has to be followed, look at the following illustration:

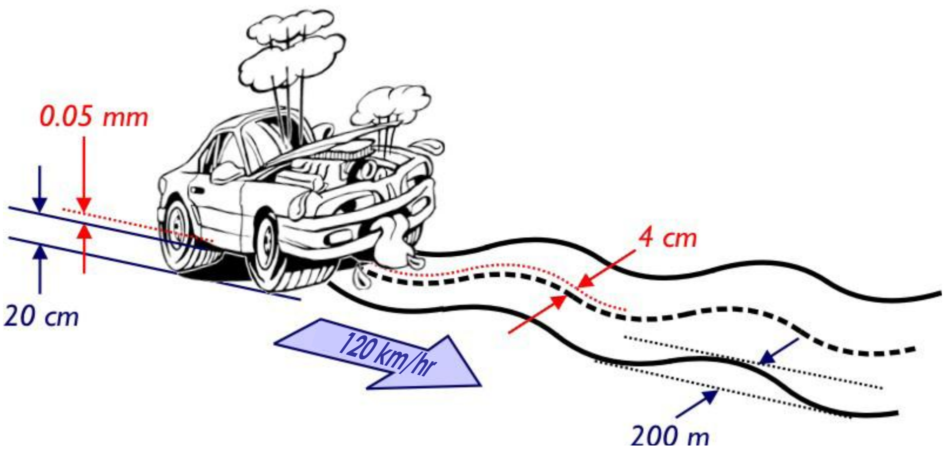


Figure 14. Comparison between the swerving motion of a car and a DVD player

Just compare the read unit in a Blu-ray player with a car doing 120 km/hour on a motorway. It must then be possible for it to swerve by 200 m while not moving more than 4 cm from the centre of this lane!!

This is necessary because the centre of the data spiral will never coincide precisely with the centre of the shaft of the motor that drives the disc round.
Or the disc or the player may vibrate, the disc might be slightly bent, etc. All reasons why the data track never precisely coincides with the laser spot. To solve this problem, CD, DVD and Blu-ray Disc players use active tracking or ***Radial Tracking***.

Vertical movement

A second problem is vertical vibration of the disc! Remember that the speed of a CD, DVD or Blu-ray Disc at the rim in a high-speed drive can be as high as 200 km/hour! Out-of-balance and air eddies will cause the disc to vibrate. Besides wobbling from left to right, the disc will then also wobble up and down. This affects the laser's 'focusing'.

The laser is focused by a lens to as sharp a dot as possible, which then impinges on the disc. You can also see this on the model. Move the disc up and down a little: what happens to the dot?

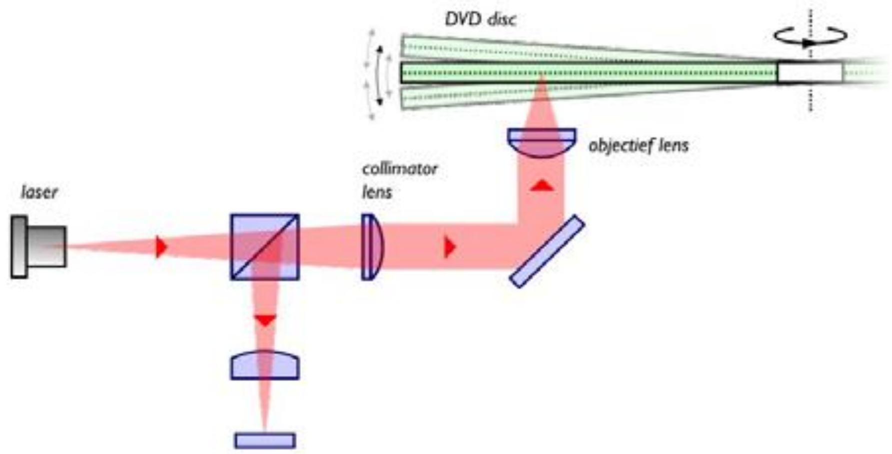


Figure 15. Projecting the laser beam onto the disc. If the disc moves up and down, the laser beam goes out of focus.
objectief lens → objective lens

Figure 15 shows how the laser is projected. You can also see that the disc is wobbling up and down. If you knew where the disc was at any given moment you could - by moving the objective lens - ensure that the laser dot was projected sharply onto the disc at all times. The system designed to do this is known as active ***focus tracking***.

4.2 Measurement and control

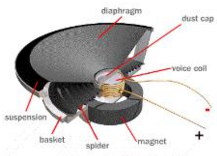
To make it possible – despite all kinds of wobbling movements by the disc – to follow the track containing pits properly, therefore, you need two tracking systems: radial tracking and focus tracking. These systems are combined in a single important component: the ***tracking actuator***.

Figure 16. A tracking actuator from a DVD player

A tracking actuator is constructed as follows:

- It consists of a holder in which the lens is suspended that focuses the laser beam;
- This holder is suspended loosely from four thin stalks and can therefore move freely in the horizontal and vertical directions.
- Several coils are attached to the holder and these are located in a magnetic field generated by two strong magnets. The magnets are also part of the actuator, but are attached to the 'fixed world'.

Magnets and coils in a loudspeaker



The same principle is also applied in loudspeakers: here too a moving coil is enclosed in a fixed magnetic field. Music (such as your iTunes file) is converted to an electric current that varies over time.
When a current is sent through a coil located in a magnetic field, a force is exerted on the loudspeaker, so that it starts to move and generates sound. In our case the lens holder in the actuator will start to move, so that the lens can be adjusted in the vertical and horizontal directions.

Using the Lorentz force

The force that makes the loudspeaker or actuator move is also known as the Lorentz force. This Lorentz force $F_{Lorentz}$ is proportional to the current I_{coil} and the magnetic field B. The larger the current, the larger the force exerted on the lens holder and hence the greater the deflection that the lens can make. By setting the current I_{coil} we can adjust the position of the lens very precisely and ensure that the laser spot is always exactly in the centre of the data track.

Error signal

How does a CD, DVD or Blu-ray Disc player know what current it has to send through the coils? It needs information about the current position of the disc in respect of the laser spot. We call this information the 'error signal', which is derived from the observed detector signal.

This kind of error signal is available to us both for focusing (vertical) and for tracking (horizontal). If for instance the focus error signal is positive, the disc is too high. If the focus error signal is negative, the disc is too low. If the tracking error signal is negative, the laser spot is to the left of the data track. And if the tracking error signal is positive, the laser spot is to the right of the track containing data to be read.
We can only read the information on the disc correctly if the laser spot coincides exactly with the disc (focus error signal = 0) and if the laser spot coincides exactly with the centre of the data track (tracking error signal = 0).

Controller

Once we know how large the error signal is, we need a 'controller'. This converts the error signal to an electric current, which is sent through the coils.

Cruise control

Just compare this with the 'cruise control' on a car. A controller is also used here to ensure that the speed always stays exactly the same as the set speed. The controller automatically attempts to reduce the error (current speed minus set speed) to zero. If you are driving downhill, the cruise control automatically reduces the acceleration. If you are driving uphill, the acceleration is increased a little. You don't have to do anything yourself to achieve this.

CD, DVD and Blu-ray Disc players contain several of these controllers. If the focus error signal is positive (disc is too high), the focus controller automatically provides a positive current that moves the actuator up and therefore the lens too. If the focus error signal is negative (disc is too low), the focus controller sends a negative current through the actuator coils, so that the actuator and lens move down. In this way the distance between the disc and the lens always remains unchanged.

The same happens for the tracking. The tracking controller ensures that the laser spot always coincides exactly with the centre of the track.

Control speed

When the focus and tracking controller are on, the laser spot is as it were 'stuck' to the disc. We can now start actually reading the data. You should be aware that a controller of this kind, which consists of some electronics, has a certain maximum control speed. The heavier the actuator, the lower this maximum control speed. This is because it takes more effort to move a heavy object quickly than a light object.

The controllers in a CD, DVD or Blu-ray Disc player also have a maximum control speed. This speed is also referred to as ***bandwidth***. So it may happen, if for instance the player takes a knock, that the controller cannot keep pace with this rapid change in the error signal. In that case ‘the control fails’ and pieces of data are not read or not read correctly. But don't worry: that's exactly why we invented error correction.

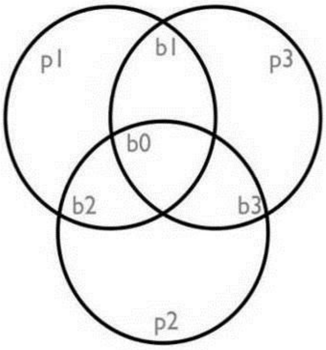
Just how all of this will work you'll find out using the large DVD player model!

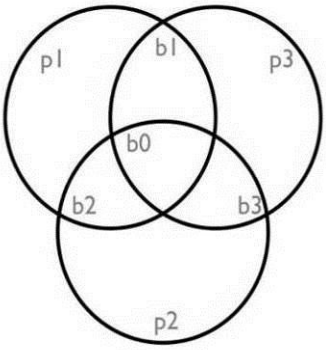
Annexe A: 'Binary' worksheet

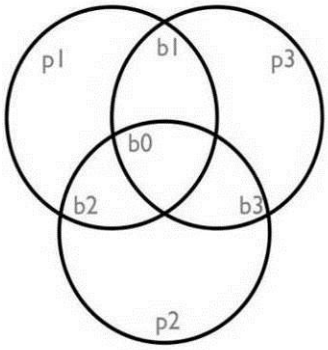
This table shows the letters of the alphabet, their ASCII codes and their binary equivalents. ASCII is the American Standard Code for Information Interchange and it links letters, numbers and other symbols to a numeral so that a computer can use it for 'calculating'.

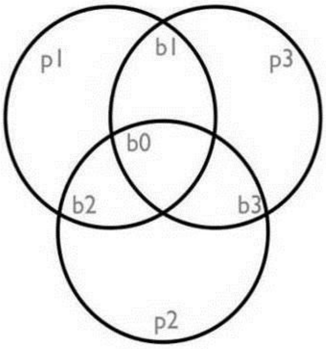
letter	ASCII code	binary
capital A	101	01000001
capital B	102	01000010
capital C	103	01000011
capital D	104	01000100
capital E	105	01000101
capital F	106	01000110
capital G	107	01000111
capital H	110	01001000
capital I	111	01001001
capital J	112	01001010
capital K	113	01001011
capital L	114	01001100
capital M	115	01001101
capital N	116	01001110
capital O	117	01001111
capital P	120	01010000
capital Q	121	01010001
capital R	122	01010010
capital S	123	01010011
capital T	124	01010100
capital U	125	01010101
capital V	126	01010110
capital W	127	01010111
capital X	130	01011000
capital Y	131	01011001
capital Z	132	01011010

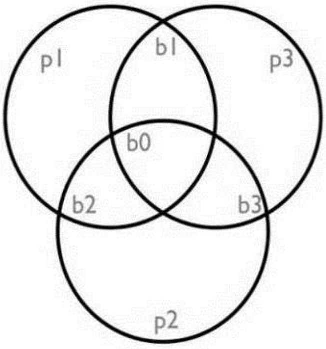
Annexe B: 'Parity table' worksheet

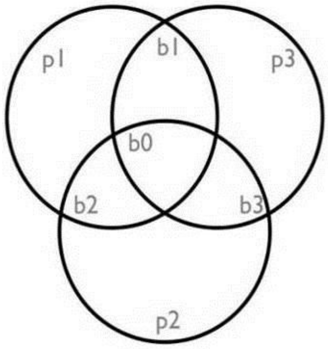


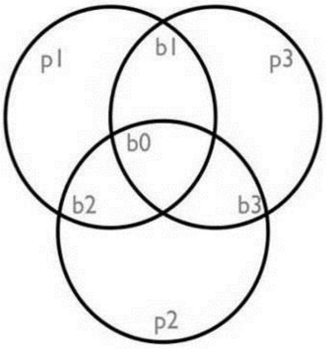


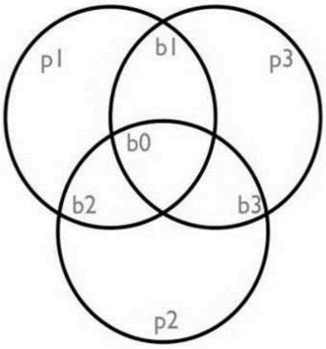


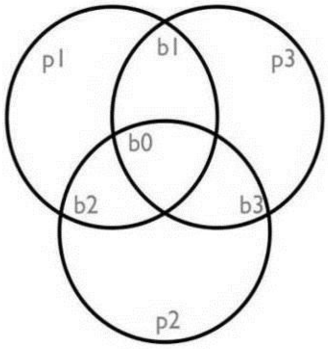


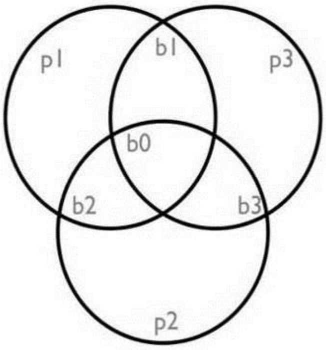


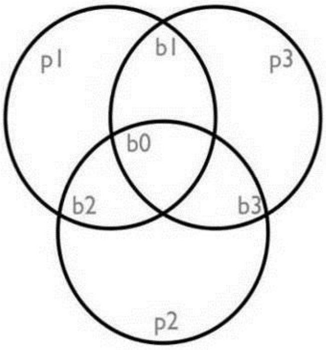


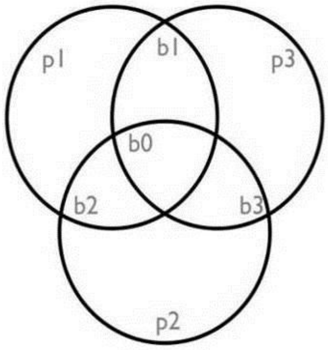












Annexe C: Worksheet 1 - 'Make your own CD'

Recording

Write your digitised (binary) message here:

			ASCII code		binary code
1st letter		=		=	CW1 / CW2
2nd letter		=		=	CW3 / CW4
3rd letter		=		=	CW5 / CW6
4th letter		=		=	CW7 / CW8
5th letter		=		=	CW9 / CW10

Write the code words here. The parity bits go in the grey boxes.
Cw1 stands for code word 1, cw2 stands for code word 2, etc.

1st letter	CW1		CW2	
2nd letter	CW3		CW4	
3rd letter	CW5		CW6	
4th letter	CW7		CW8	
5th letter	CW9		CW10	

Write here the series of ones and zeroes after you have applied interleaving. Pay attention to the order of the code words. The parity bits should go in the grey boxes.

	CW1	CW2	CW3	CW4	CW5	CW6	CW7	CW8	CW9	CW10
b3s										
b2s										
b1s										
b0s										
p3s										
p2s										
p1s										

Now write the bits to your CD (1 digit per box) line by line (from left to right and from top to bottom).

Begin at START and, **at the point where the scratch is, replace your own information by 0s.**

Annexe D: Worksheet 2 - 'Make your own CD'

Playback

Write here the series of ones and zeroes of the CD you're going to play.
Begin at START (first fill in the row with the b3s, then the row with the b2s, etc.)

	CW1	CW2	CW3	CW4	CW5	CW6	CW7	CW8	CW9	CW10
b3s										
b2s										
b1s										
b0s										
p3s										
p2s										
p1s										

Write here the reconstructed code words after 'de-interleaving' (this is the opposite operation of interleaving). The parity bits go in the grey boxes. Where necessary replace an incorrect bit after checking with the Hamming code.

1st letter	CW1								CW2							
2nd letter	CW3								CW4							
3rd letter	CW5								CW6							
4th letter	CW7								CW8							
5th letter	CW9								CW10							

Write the digital message here:

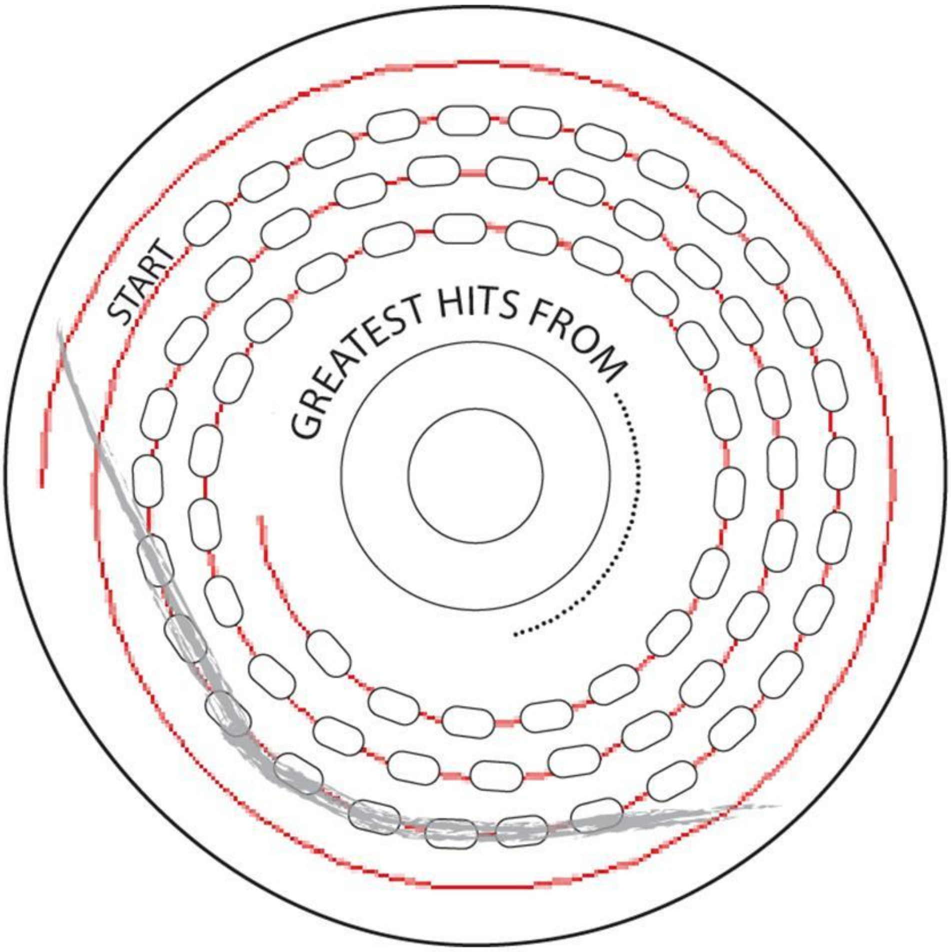
	binary code		ASCII code		letter
1st letter	← CW1 →← CW2 →	=		=	
2nd letter	← CW3 →← CW4 →	=		=	
3rd letter	← CW5 →← CW6 →	=		=	
4th letter	← CW7 →← CW8 →	=		=	
5th letter	← CW9 →← CW10 →	=		=	

The message is: _ _ _ _ _
Is this the same as the message that your fellow pupil encoded?

Annexe E: Worksheet 3 - 'Make your own CD'

Write the 0s and 1s in the boxes. You can also colour the boxes using two colours. In that case remember to stipulate which colour represents a 1 and which colour a 0.

At the point where the scratch is, replace your own information by 0s. **Note**: do **not** write your original, error-free bits here first, as otherwise they could be found by the team that will read this CD and that isn't the idea!



Make your own CD here.