# ERROR – CORRECTING CODES

## ROYAL INSTITUTION MASTERCLASS

## UNIVERSITY OF GLASGOW

## NOVEMBER 2013

Dr Ian Anderson
University of Glasgow

**Introduction.** Compact discs, dvds and mobile phones depend for their effectiveness upon the use of error-correcting codes. There are many other situations where error-correcting codes are invaluable, and at the end of this masterclass we shall see how they were used to transmit the first really good photographs of Mars back to Earth.
In the supermarket, other types of code are used: these are error-detecting, but not error-correcting. We will start by looking at such codes, seeing how check digits can be used in a variety of ways to spot when errors have been made.

# Error-detecting codes.

## 1(a) The EAN (European Article Number) system

A 500 gram packet of Kellog's cornflakes carries the number

5 0 0 0 1 2 7 0 1 2 0 9 7.

The 50 at the start indicates UK; 00127 indicates Kellogs; 01209 is the code given by Kellogs to this particular size of packet of cornflakes; and the 7 at the end is a check digit which is chosen so that if the numbers are given weights 1 and 3 alternately, the weighted sum is a multiple of 10:

5 0 0 0 1 2 7 0 1 2 0 9 7
1 3 1 3 1 3 1 3 1 3 1 3 1

5x1 + 0x3 + 0x1 + 0x3 + 1x1 + 2x3 + 7x1 + 0x3 + 1x1 + 2x3 + 0x1 + 9x3 + 7x1 = 60.

**Example 1** Simon Singh's Code Book has EAN number

9 7 8 1 8 5 7 0 2 8 7 9 y

where y is the check digit. Here 978 indicates 'bookland', and 185702 indicates the publisher (Fourth Estate). Find the digit y.

**Example 2** A particular bottle of Spanish wine has EAN

8410396z20406

where the digit z is unclear. Here 84 indicates Spain. Find z.

**Example 3** Sainsbury's uses an 8-digit version of EAN for their own products, with weights 1 and 3, but this time starting with 3 (why?). Find the check digit x for Sainsbury's drinking chocolate:

0 0 9 5 7 9 6 x.

**Example 4 Detecting transpositions.** A common error is the transposition of two adjacent digits, e.g. misreading 42869 as 42689. Does the EAN system detect such errors?

Suppose xy is misread as yx so that the weighted sum contribution 3x+y is replaced by 3y+x. The error will be spotted unless the difference between 3x+y and 3y+x is a multiple of 10.
When is 2x-2y a multiple of 10?

## 1(b) Bank account numbers

Each bank account has a 6-digit sort code followed by an 8-digit account number, giving 14 digits in all. The weights used depend on the bank.

**Example 5 The Glasgow City Mission** has a Bank of Scotland account

8 0 5 4 0 1 0 0 4 0 1 0 9 5

and the weights used for all accounts with sortcode 805401 are

0 0 1 8 2 6 3 7 9 5 8 4 2 1.

The checking property is that the weighted sum must be a multiple of 11.
Check that this holds here.

**Example 6** Santander 090128 accounts uses the weights

0 0 3 7 1 3 7 1 3 7 1 3 7 1

Can 09012849369138 be a valid Santander account number?

**1( c) The IBM system** used for Visa/Mastercard/GULibrary

This uses weights 2 1 2 1.....The problem here is that, for example, 3 and 8, when multiplied by 2, end in the same number. So the system has to distinguish between 1 and 6, 2 and 7, 3 and 8, and so on. So the rule used is:
The weighted sum plus the number of digits greater than or equal to 5 which were multiplied by 2 must be a multiple of 10.

**Example 7** A Glasgow University library book has code number

3 0 1 1 4 0 0 5 2 9 7 3 6 y

for some y. Find the value of y.
The weights give weighted sum
$2(3 + 1 + 4 + 0 + 2 + 7 + 6) + (0 + 1 + 0 + 5 + 9 + 3 + y) = 64 + y$.
Also there are two numbers 7,6 in the first of these sums which are ≥5; so
$64 + 2+y$ has to be a multiple of 10; so $y=4$

**Example 8** Find the check digit y in the following credit card number:

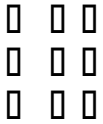
4547 3012 5478 103y

**Example 9** Find the missing number w in the following library number:

301141w7603811

## Error-correcting Codes

In 1971 wonderful pictures of Mars were sent back by the Mars Mariner 9 Spacecraft, over a distance of 84 million miles, using a 20-watt transmitter. How was it done?
Pictures are made up of many dots, each a different shade of darkness, ranging from white to black.
As a simple example, suppose there are only two shades, B and W. The picture



could be sent as 011010110
but if an error occurred it wouldn't be detected, and the wrong picture would arrive. We could repeat each digit and send

0011110011100111100

If we received 0010110011100111100
we would know that an error had occurred in 3rd or 4th place but we couldn't correct it. If we sent each digit thrice, then we could decode on a majority principle but we have tripled the length of the message just to correct one error.

Suppose there were 4 shades. We could use 00 01 10 and 11 to represent them. Again, can't detect errors. If we introduced a check digit to make all codewords have even sum, i.e. if we used 000, 011, 101, 110 as codewords, and received 010, then we would certainly detect an error, but we couldn't correct it – should it be 000 or 110 or 011 ? But suppose we used the four codewords

0000000 1110000 1010101 1111111.

Then each pair differ in at least 3 places, so one error would result in a binary sequence closer to the 'correct' word than to any other, and so can be corrected. For example,

1110101 is corrected to 1010101

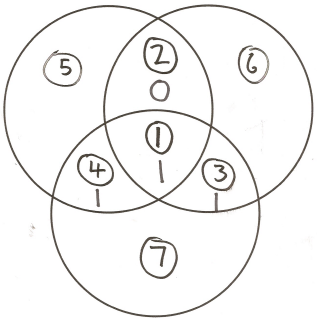
**Example 10** Find five binary words of length 7, differing in at least 3 places.

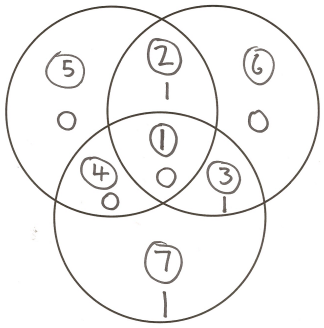
**Example 11 The Hamming Code.**

This code has 16 codewords of length 7, each pair differing in 3 places.

There are 16 4-digit binary words and we shall make each of them into a 7-digit word as follows.

The 7 regions of the Venn Diagram are labelled by the circled numbers 1 to 7. Let's consider the binary word 1011. Place the digits in the first 4 regions as shown in (a).




(a) (b)

To extend 1011 we look at the top left circle and put in region 5 whatever gives the circle an even number of 1s – namely 0 – and use it as the 5th digit of our codeword; for top right circle put 0 in region 6 and for bottom circle put 1. So extend 1011 to 1011001.
Similarly check that 0101 extends to 0101011 and 1101 extends to 1101100.
In this way we get *sixteen 7-digit codewords each pair of which differ in at least 3 places.*
**Error-correction.** As an example, suppose we receive the word 0110001. Put it in the regions of the Venn Diagram as

shown in (b). We see that the top right and bottom circles are fine, but the top left is wrong. So we have to change the symbol in the region not in the top left circle but not in the other two; so we change the 0 in region 5 to 1, and correct 0110001 to 0110101.

Example 12 In the Hamming Code, correct 1011111 and 1001111.

Richard Hamming invented his code in 1950 when he was working in the Bell Telephone Labs in New Jersey, USA. He died in 1998, at the age of 82.

## The Mars Mariner 9 Code

Each photograph was made up of more than 500,000 dots. Each dot was one of 512 shades of grey, each represented by a 9-digit binary number. So each photo was represented by a binary string of more than 500,000 x 9 bits, i.e. a binary sequence of length more than 4.5 million.
This long sequence is split into substrings of length 6. There are 64 such possible strings. Each will be encoded as a 32-digit codeword. So the photograph will now be encoded by a binary sequence of length greater than

$$32/6 \times 4.500{,}000$$

and so consisting of more than 20 million bits!

The 64 codewords of length 32 are to be chosen so that any two of them differ in at least 16 places. Then even if as many as 7 errors are made in each codeword of length 32, they can still be corrected. How are these codewords constructed? Follow the pattern below.

1 1
1 0 gives 2 codewords of length 2 differing in 1 place

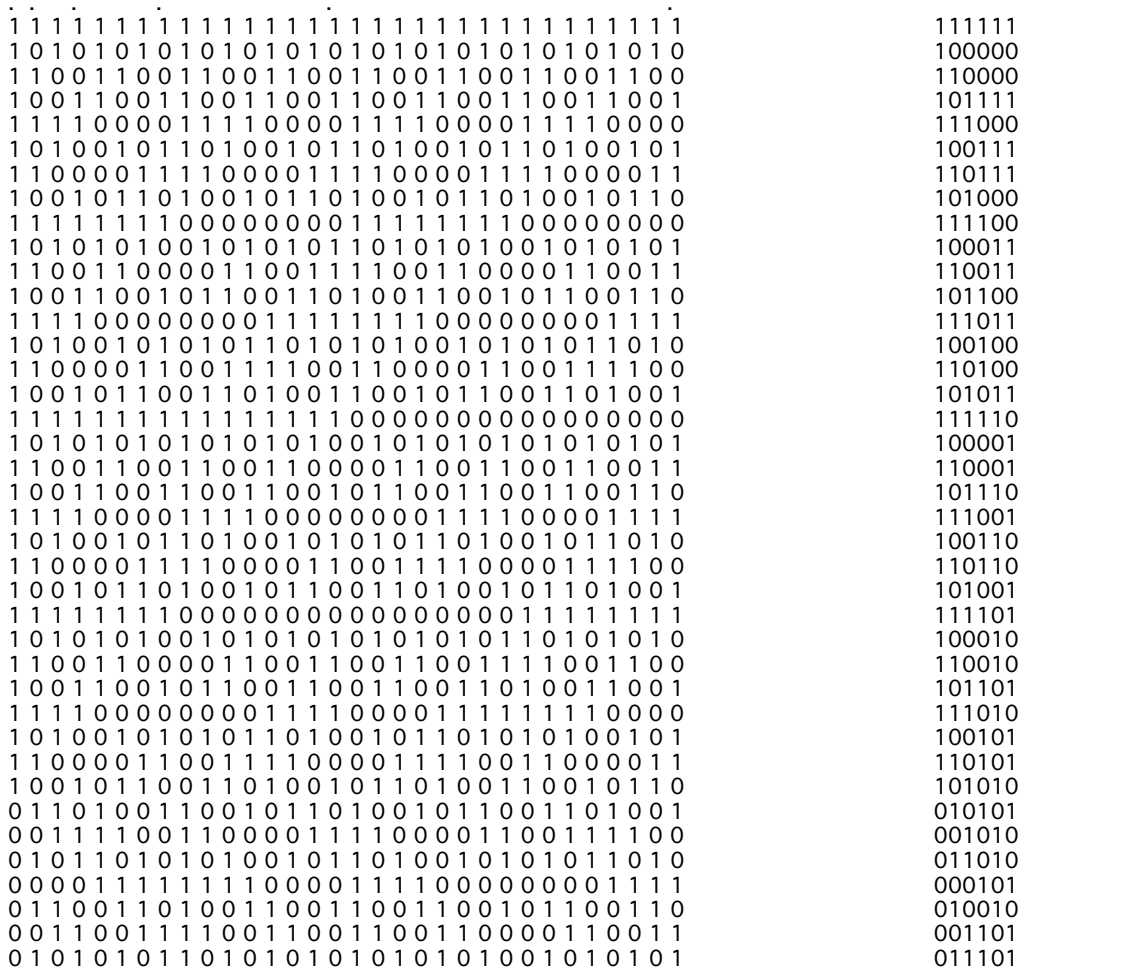
1 1 1 1
1 0 1 0
1 1 0 0

1 0 0 1 gives 4 codewords of length 4 differing in 2 places

1 1 1 1 1 1 1 1
1 0 1 0 1 0 1 0
1 1 0 0 1 1 0 0
1 0 0 1 1 0 0 1
1 1 1 1 0 0 0 0
1 0 1 0 0 1 0 1
1 1 0 0 0 0 1 1
1 0 0 1 0 1 1 0 gives 8 codewords of length 8 differing in 4 places.

Continuing in this way get 32 of length 32, differing in 16 places. Take these and the 32 'complements' (interchanging 0 and 1) to get the required 64 codewords.

## Mars Mariner 9 Hadamard Code



| | |
|---|---|
| 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 | 000010 |
| 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 | 010110 |
| 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 | 001001 |
| 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 | 011001 |
| 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 | 000110 |
| 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 | 010001 |
| 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 | 001110 |
| 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 | 011110 |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 000001 |
| 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 | 010100 |
| 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 | 001011 |
| 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 | 011011 |
| 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 | 000100 |
| 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 | 010011 |
| 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 | 001100 |
| 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 | 011110 |
| 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 000011 |
| 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 | 010111 |
| 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 | 001000 |
| 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 | 011000 |
| 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 | 000111 |
| 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 | 010000 |
| 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 | 001111 |
| 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 | 011111 |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 000000 |