

Codification

The image displays a musical score for Violin I, consisting of five staves of music. The score is written in treble clef with a key signature of two sharps (F# and C#). The music features a variety of dynamics and articulations, including *f* (forte), *p* (piano), *sf* (sforzando), *dim.* (diminuendo), and *pdolce* (piano dolce). The notation includes slurs, accents, and fingerings (e.g., 1, 2, 3, 4). The score is a single melodic line with some double stops and trills.

Haydn: Streichquartett op 54.3 aus Largo, Violine I

Codification

Die Bahn 

Bitte auf A4 ausdrucken

OnlineTicket

IC/EC Fahrkarte

Gültigkeit: Hinfahrt ab 14.09.2007, Rückfahrt ab 16.09.2007

Sparpreis 50 (Hin- und Rückfahrt)

Klasse: 1

Erw: 1

Hinfahrt: Lüneburg → Bonn, mit IC/EC

Rückfahrt: Bonn → Lüneburg, mit IC/EC

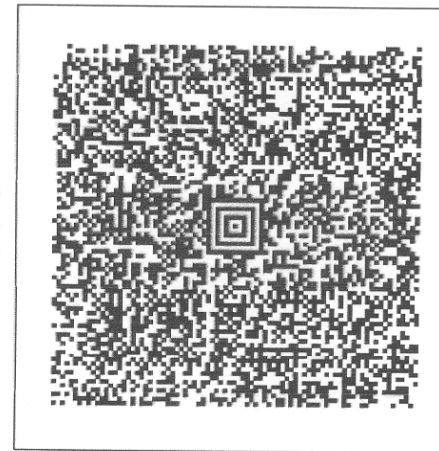
Über: H: NV*HH-Harb 11:57 IC2029 R: BonnHbf 15:22 IC2004/
MS-Hbf 17:55 IC1904/HH-Harb*NV

DB: Gilt nur in gebuchten Zügen gem. der im Abschnitt "Fahrkarte" angeg. Strecke u. Reisetagen sowie im NV (S/RB/RE/IRE) vor/nach den gebuchten Zügen. Besondere Konditionen für Umtausch/Erstattung beachten.

Zahlungspositionen und Preis

Kreditkartenzahlung		Positionen	
Betrag	EUR 116,00	Fahrkarte Hin- und Rückfahrt	1 EUR 113,00
Datum	11.09.2007	Reservierung Hinfahrt	1 EUR 1,50
Transaktions-Nr	329604	Reservierungen Rückfahrt	2 EUR 1,50
VU-Nr	4556695619	Summe	EUR 116,00
Gen-Nr	259428	Enthaltene MwSt. (D) 19%	EUR 18,52

Ihre Kreditkarte wurde mit dem oben genannten Betrag belastet. Die Buchung Ihres OnlineTickets erfolgte am 11.09.2007. DB Fernverkehr AG/DB Regio AG, Stephensonstr. 1, 60326 Frankfurt, Steuernummer: 045 231 28552.



Hinfahrt:
Zertifikat: 20C2 VNAR GDV
Gültig ab: 14.09.2007

Zangenabdruck

Rückfahrt:
Zertifikat: 20G1 ZRJX 3AV
Gültig ab: 16.09.2007

Zangenabdruck

Frau Prof. Dr. Dörte Haftendorn
Ausweis: BahnCard 0267
Auftragsnummer: ZF6P80

Ihre Reiseverbindung und Reservierung Hinfahrt am 14.09.2007

2

EAN Europäische Artikelnummer



Ziffern 1 und 2
codieren das
Hersteller-Land.

Ziffer 3 bis 12
codieren
Händler und
Ware.

**Die letzte
Ziffer ist
eine
Prüfziffer.**

EAN European article number

digits 1 and 2
encode the
country of the
producer

digits 3 and 12
encode the
merchant and
article

**The last
digit is a proof
number.**



EAN Europäische Artikelnummer

Außländische Produkte

• This is made by pupils,
12 year old

Holland 87

Österreich 90

Frankreich 31, 32, 33

USA 0...

Portugal 56

Chile 78

Deutschland 40, 42, 43

Italien 80

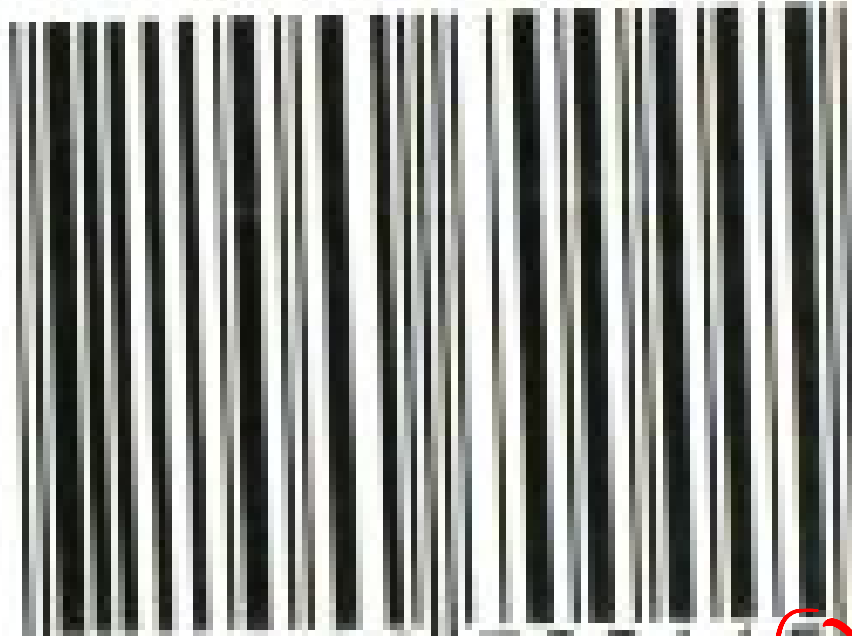
Spanien 84

Mexico 74

Großbritannien/Irland 50



EAN European article number



- The proof number has to be the complement to the next tenner.

• proof number

• ok

Prüfziffer passt



muß

$$8 + 2 \cdot 1 + 1 \cdot 3 + 5 + 0 + 0 + 2 \cdot 1 + 0 + 0 + 4 \cdot 3 + 5 = 75 + 5 = 80 = n \cdot 10$$



EAN

Switching of digits
is often detected by
the proof number

drehen

X	Y	Y · X
↓ · 1	↓ · 3	↓ · 1 ↓ · 3



EAN

Switching of digits is **often detected** by the proof number

•but not noticed are the pairs:

- Solution are
- $x-y=5$
- $x=y+5$

x	y
5	0
6	1
7	2
8	3
9	4

•twisted

$$\begin{array}{cc}
 x & y \\
 \downarrow \cdot 1 & \downarrow \cdot 3 \\
 x + 3y & \equiv y + 3x + \\
 \Leftrightarrow 2x - 2y \equiv_{10} 0 & \Leftrightarrow 2 \cdot (x - y) \equiv_{10} 0
 \end{array}$$

- In the last years one changed the type of the book number

ISBN out of date

- new: ISBN-13

Switching of digits is **always** detected by the proof number.



3 5 2 8 0 3 2 1 5 P

↓ ·10 ↓ ·9 ↓ 8 ↓ ·7 ↓ ·6 ↓ ·5 ↓ ·4 ↓ 3 ↓ ·2 ↓ ·1

old ISBN
without P

ISBN ist veraltet

- neu: ISBN-13
- sie ist eine EAN

Zifferndreher werden **immer** von der Prüfziffer gemerkt. Der Grund: 11 ist Primzahl



3 5 2 8 0 3 2 1 5 \boxed{p}
 $\downarrow \cdot 10 \downarrow \cdot 9 \downarrow \cdot 8 \downarrow \cdot 7 \quad \downarrow \cdot 5$

- Die alte Buchnummer ohne Prüfziffer
- 978 steht immer am Anfang der Buch-EAN

$$30 + 45 + 16 + 56 + 0 + 15 + 8 + 3 + 10 + p \equiv 0 \pmod{11}$$
$$183 + p \equiv 0 \pmod{11}$$

ISBN

out of date

- new: ISBN-13
- is a EAN

Switching of digits is **always** detected by the proof number. The reason: 11 is a prime



3 5 2 8 0 3 2 1 5 \boxed{p}

$\downarrow \cdot 10 \downarrow \cdot 9 \downarrow \cdot 8 \downarrow \cdot 7 \quad \downarrow \cdot 5$

$$30 + 45 + 16 + 56 + 0 + 15 + 8 + 3 + 10 + p \equiv 0 \pmod{11}$$

$$183 + p \equiv 0 \pmod{11}$$

- The old book number without proof number
- 978 always in front
- new EAN proof number

ISBN

out of date

- new: ISBN-13
- is a EAN

Switching of digits is **always** detected by the proof number. The reason: 11 is a prime



3 5 2 8 0 3 2 1 5

↓ ·10 ↓ ·9 ↓ 8 ↓ 7 ↓ 5

$$30 + 45 + 16 + 56 + 0 + 15 + 8 + 3 + 10 + p \equiv 0 \pmod{11}$$

- proof number must be: 4

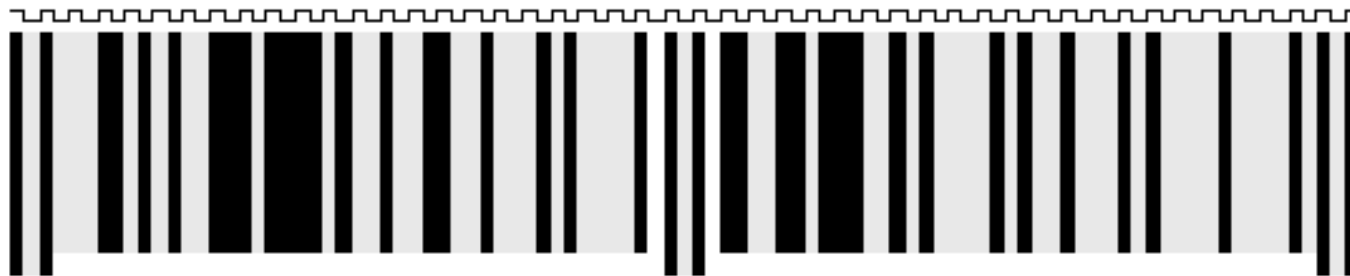
$$183 + 4 = 187 = 110 + 77$$

$$183 + p \equiv 0 \pmod{11}$$

EAN

4 0 0 3 2 7 3 1 0 3 8 6 3

4 0 0 3 2 7 3 1 0 3 8 6 3



•0001101 0100111 0111101 0010011! •parity 1 •parity 0 •parity 0

•Using the parity the reading direction of the bar code is identifiable.

The leading digit is encoded by the sequence in the first column.

4 -> ABA ABB

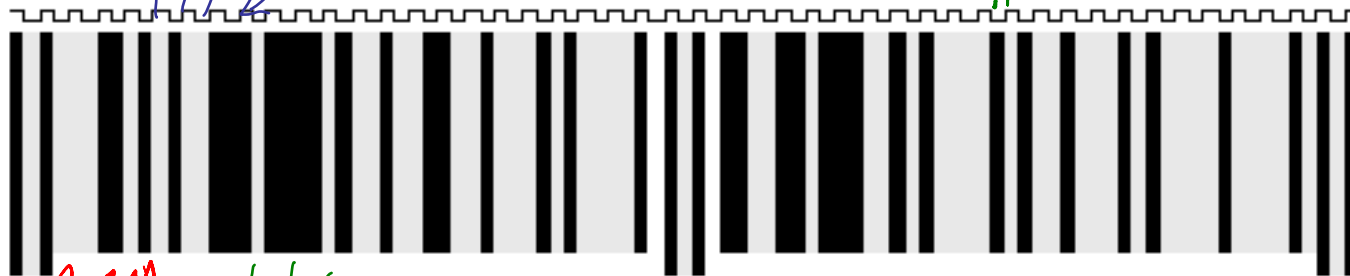
Z	Erste Ziffer	Code A	Code B	Code C
0	AAA AAA	0 00110 1	0 10011 1	1 11001 0
1	AAB AAB	0 01100 1	0 11001 1	1 10011 0
2	AAB BAB	0 01001 1	0 01101 1	1 10110 0
3	AAB BBA	0 11110 1	0 10000 1	1 00001 0
4	ABA ABB	0 10001 1	0 01110 1	1 01110 0
5	ABB AAB	0 11000 1	0 11100 1	1 00111 0
6	ABB BAB	0 10111 1	0 00010 1	1 01000 0
7	ABA BAB	0 11101 1	0 01000 1	1 00010 0
8	ABA BBA	0 11011 1	0 00100 1	1 00100 0
9	ABB ABA	0 00101 1	0 01011 1	1 11010 0

EAN

4 | A B A A B B | code a
 4 0 0 3 2 7 3 | 1 0 3 8 6 3

0 1 0 0

4 0 0 3 2 7 3 || 1 0 3 8 6 3



000
 11
 0
 1
 0
 1

Z	Erste Ziffer	Code A	Code B	Code C
0	AAA AAA	0 00110 1	0 10011 1	1 11001 0
1	AAB AAB	0 01100 1	0 11001 1	1 10011 0
2	AAB BAB	0 01001 1	0 01101 1	1 10110 0
3	AAB BBA	0 11110 1	0 10000 1	1 00001 0
4	ABA ABB	0 10001 1	0 01110 1	1 01110 0
5	ABB AAB	0 11000 1	0 11100 1	1 00111 0
6	ABB BAB	0 10111 1	0 00010 1	1 01000 0
7	ABA BAB	0 11101 1	0 01000 1	1 00010 0
8	ABA BBA	0 11011 1	0 00100 1	1 00100 0
9	ABB ABA	0 00101 1	0 01011 1	1 11010 0

The leading digit is encoded by the sequence in the first column.
 4 -> ABA ABB

parity = number of 1's mod 2

• parity 1 • parity 0 • parity 0

Fehlerkorrigierende Codes

- Wir betrachten **binäre Codewörter** aus 0 und 1
- Die **Parität** eines Codewortes ist
 - 0, wenn das Wort eine gerade Anzahl 1 hat
 - 1, wenn das Wort eine ungerade Anzahl 1 hat
- Der **Hammingabstand** zweier Codewörter ist die Anzahl der unterschiedlich besetzten Stellen.

0101111
0111011

0000101 $P =$
0010001 $P =$
 $h =$

0101111
0000101

1010000 $P =$
1000100 $P =$
 $h =$

Bug Corrective Codes

- We look at **binary code words** made of 0 and 1
- The **parity p** of a code word is
 - 0, when the word has an even number of 1's
 - 1 when the word has an odd number of 1's
- The **Hamming distance h** between two code words is the number of differently occupied positions.

0101111 $p = 1$
0111011 $p = 1$
 . . $h = 2$

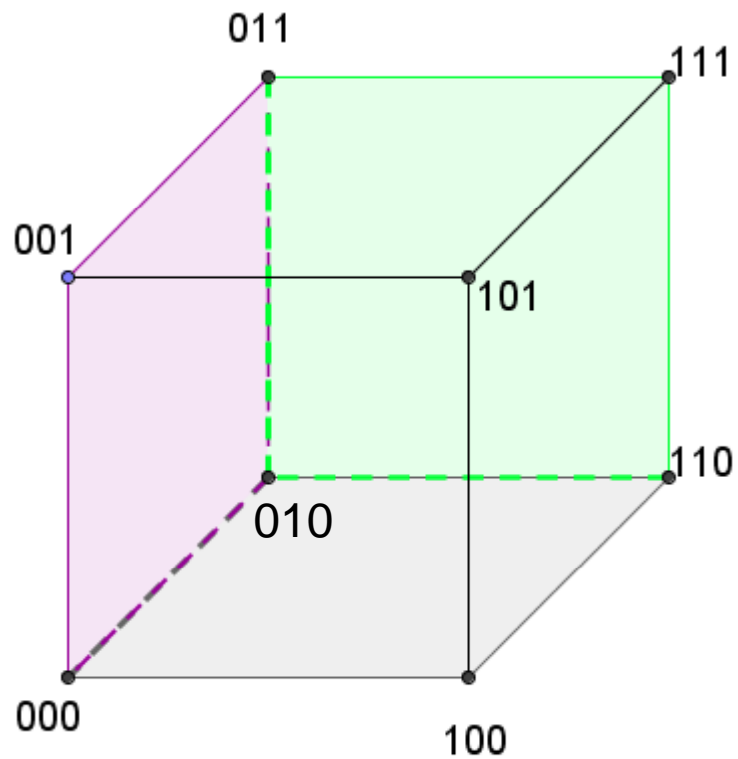
0000101 $p = 0$
0010001 $p = 0$
 . . $h = 2$

0101111 $p = 1$
0000101 $p = 0$
 . . . $h = 3$

1010000 $p = 0$
1000100 $p = 0$
 . . $h = 2$

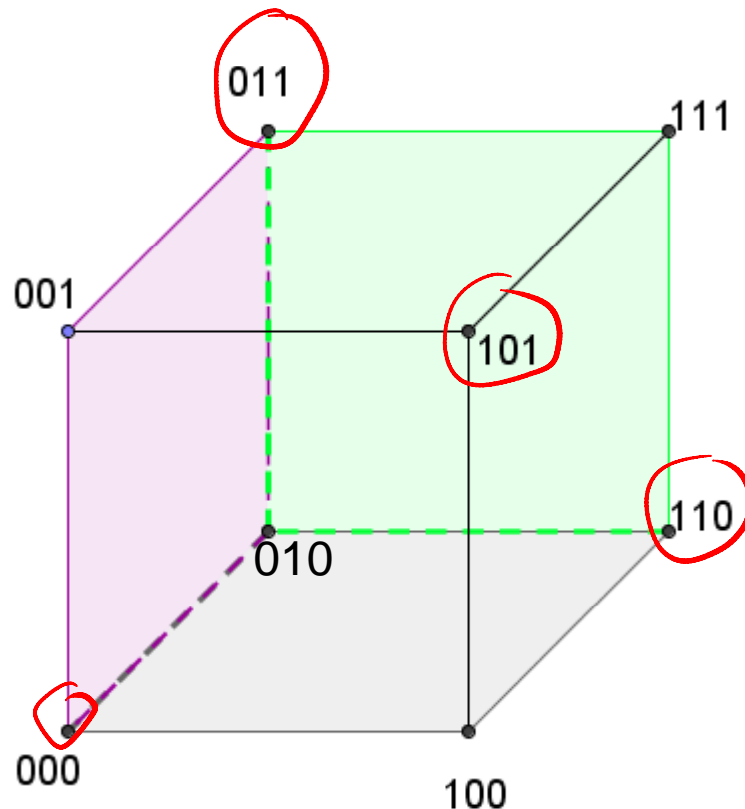
Fehler-erkennende Codes

- Wir betrachten binäre Codewörter mit 3 Bit.



Bug Corrective Codes

- We look at binary code words with 3 bits.



- Four words are allowed.
- $h=2$ for all pairs of words

The code with this four words is able to recognize single-bit bugs. But they can't be corrected.

Fehlerkorrigierende Codes

- Code mit Hammingabstand mindestens 3

• **11101101** **10011100** **10010011** Erlaubte Codewörter

- Nun tritt ein einziger Fehler auf.

• **10010100**

- Den Fehler entdeckt man dadurch, dass es kein zulässiges Wort ist.

- Die letzte Ziffer ist oft die Parität des davor stehenden Wortes.

Dann sieht man den Fehler an der falschen Parität.

- Das falsche Wort hat vom mittleren Codewort den Hammingabstand 1.

Von den anderen Codewörtern hat es den Hammingabstand mind. 2.

- Man nimmt dieses „nahe“ Codewort anstelle des falschen.

- **10011100** wird jetzt nach dem Fehler genommen

• **Nun ist alles wieder richtig.**

Bug Corrective Codes

- Code with Hamming distance at least 3

- **11101101** **10011100** **10010011** allowed code words

- Now a single-bit bug occurs

- **10010100**

- The bug is discovered, because this is not an allowed word..

- Often the last digit is the parity of the Word in front.

Then you find the bug, because the parity is false.

- The false word has Hamming distance 1 from the word in the middle (above). The Hamming distance to the other code words is at least 2.

- One takes this „nearest“ code word instead of the false word.

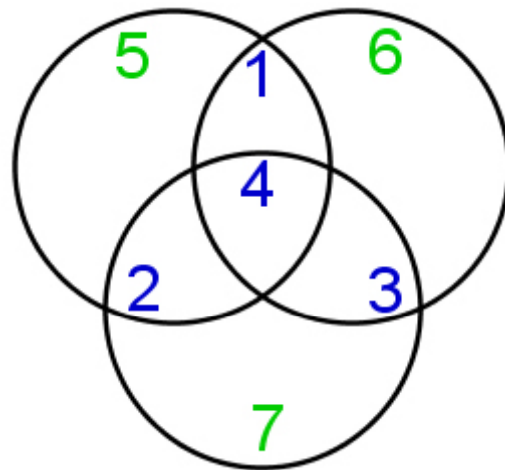
- **10011100** is now taken for further usage.

Now every thing is correct again.

Hamming-Code

- Richard Hammming fand 1948 als Erster einen fehlerkorrigierenden Code.
- Wir betrachten einen Code aus 4 Bits.
- Alle $2^4=16$ möglichen Wörter sind erlaubt.
- Die Nachricht:

IOII

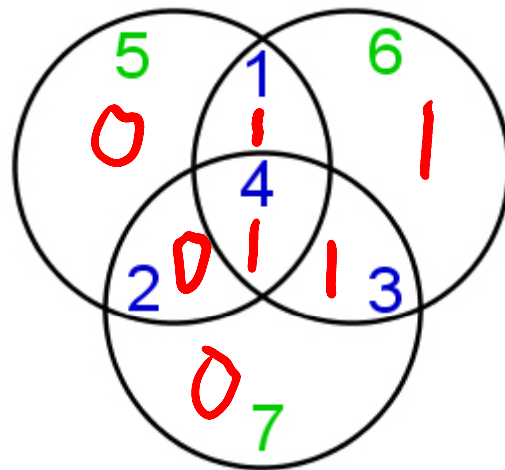


Gesendet wird:

Hamming Code

- Richard Hammming was in 19948 the first to invent a bug corrective code.
- We look at a code with 4 bits.
- All $2^4=16$ possible words are allowed.
- The message:

1011



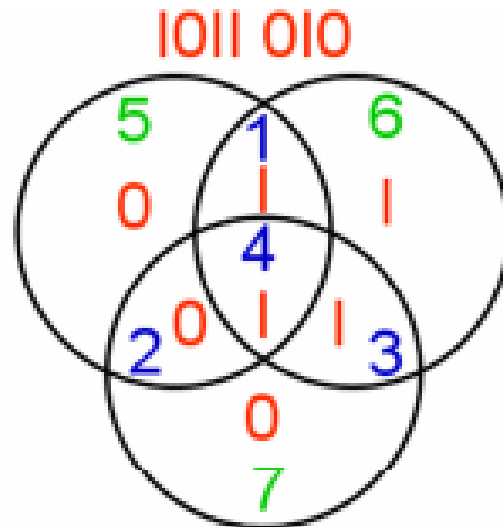
1011010 →

will be transmitted

Hamming-Code

- Zu je vier eigentlich zu sendenden Bits der Nachricht werden drei „Prüfbits“ berechnet und angehängt. Das Kreisbild verdeutlicht das Vorgehen:
- Schreibe die Nachricht in die blauen Felder 1,2,3,4.
- Schreibe in die grünen Felder die Parität der im zugehörigen Kreis stehenden Bits.
- Hänge die Bits der Felder 5,6,7 an die Nachricht an.
- Der Empfänger trägt die sieben Bits in die Felder ein und prüft, ob alles richtig ist.

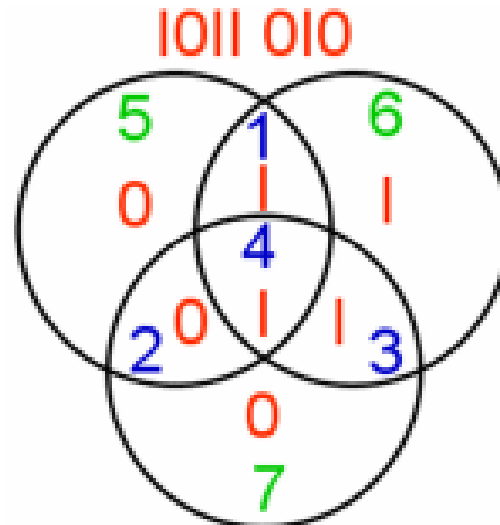
• Nachricht.
1011



• Gesendetes Wort
1011010

Hamming Code

- The message, which we will send, is splitted in packages of four bits. We calculate tree proof bits for every four message bits and attach them to the four bits. The following picture with the circles explains the method:
- The regions are numbered as shown. Write the four message bits in the blue inner regions 1, 2, 3, 4
- Write in the green regions 5, 6, 7 the parity of the bits in the correspondent circles.
- Attach the bits in regions 5, 6, 7 to the four message bits.
- The receiver puts all seven bits in the three circles and checks if all is correct.

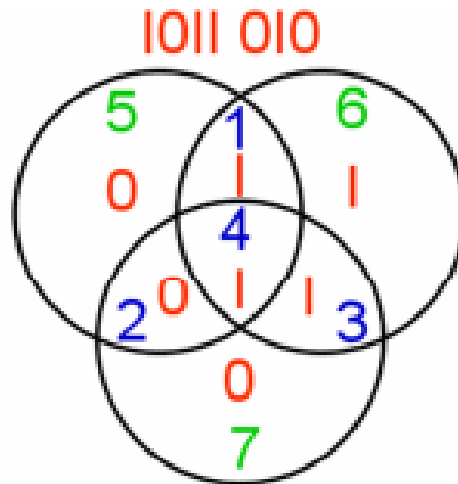


•message.
1011

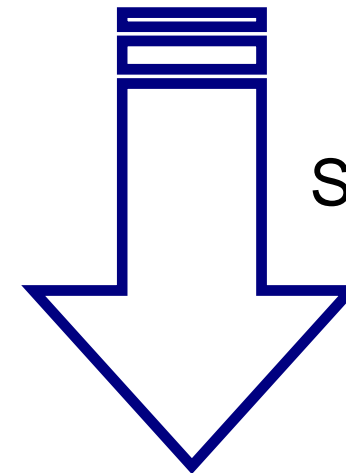
•transmitted word
1011010

Funktioniert der Hamming-Code ?

- Nachricht.
1011

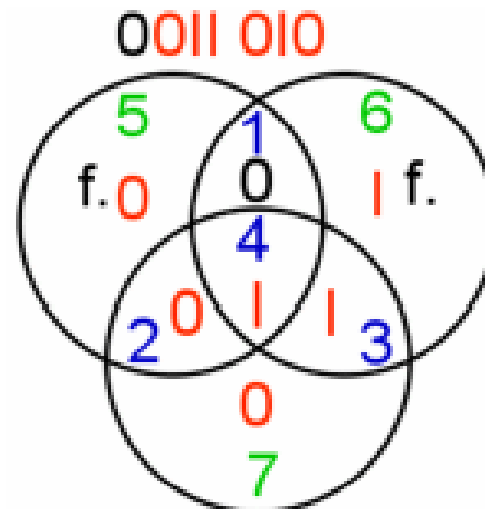


- Gesendetes Wort
1011010



Sendekanal

- Empfangenes Wort
0011010 mit Fehler

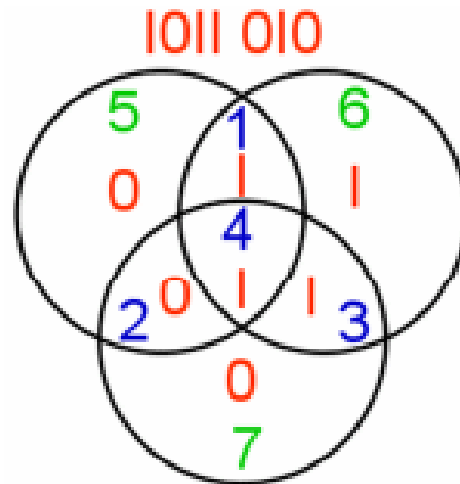


Wir betrachten
nur den Fall:
**genau ein
Fehler.**

Does the Hamming Code Work?

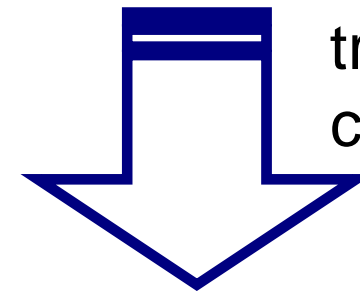
•message.

1011



•sent word

1011010



transmission
channel

• received word

0011010 with bug

• receiver detect:

5 and 6 are false.

• He changes the bit at the position 1.

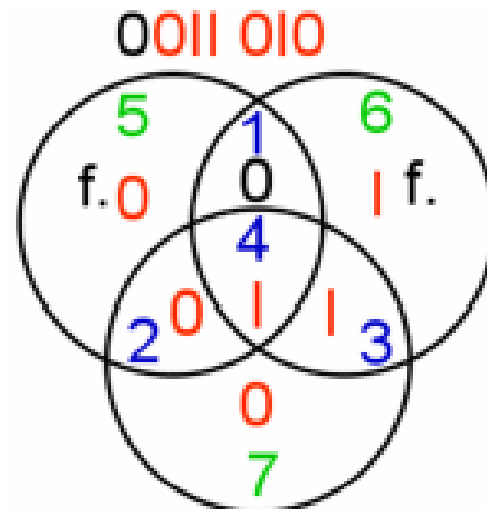
• He takes:

1011010

•all right

We only look at
the case:

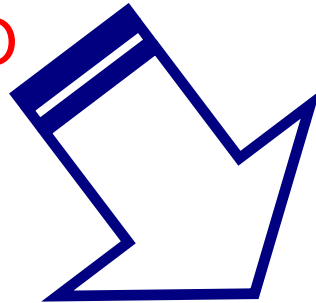
only one
single-bit bug.



Funktioniert der Hamming-Code ?

- Gesendetes Wort

1011010



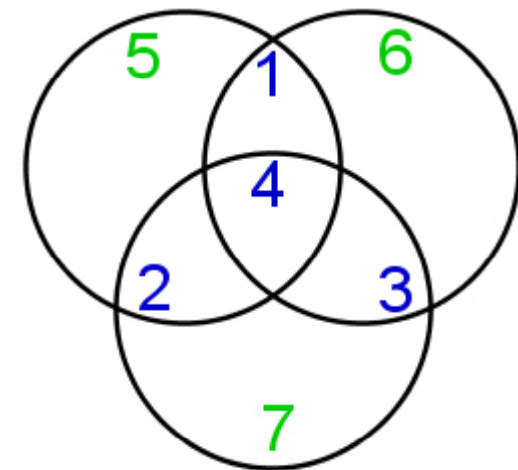
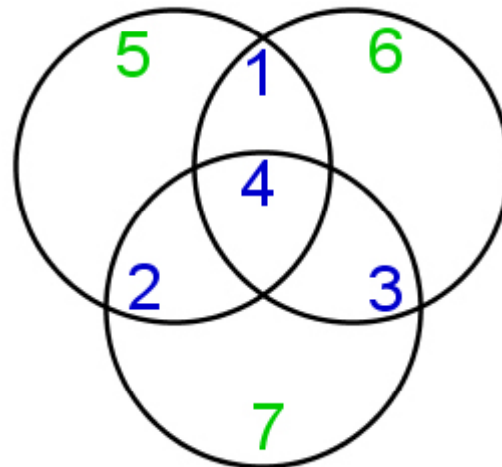
- Empfangenes Wort

- Fehler Platz 4

1010010

- Fehler Platz 5

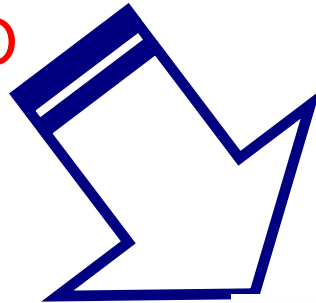
1011110



Aufgabe: Sie empfangen: 0101001 . Was ist das richtige Wort?

Does the Hamming Code Work?

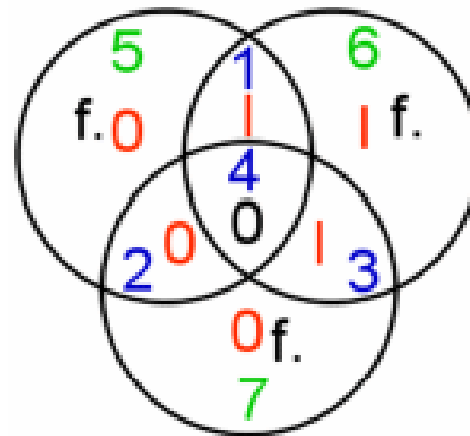
•sent word
1011010



•received word

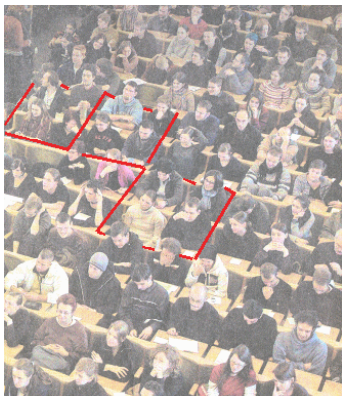
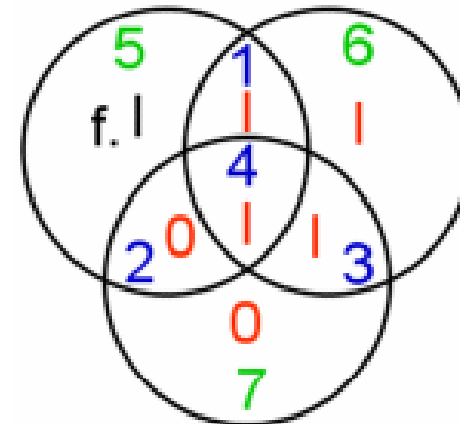
•bug at position 4

1010010
1010010



•bug at position 5

1011110
1011110



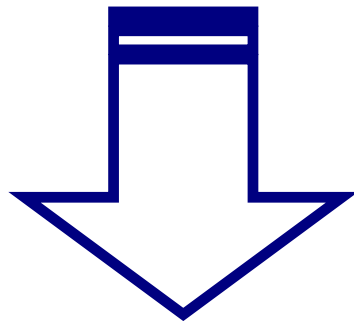
Problem: You receive: 0101001 . What is the correct word?

Funktioniert der Hamming-Code ?

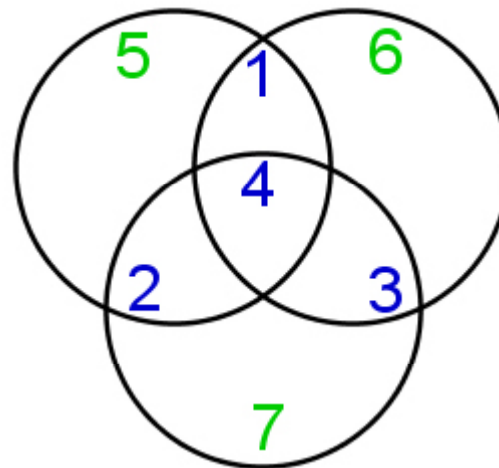
Aufgabe: Sie empfangen: 0101001 . Was ist das richtige Wort?

- Gesendetes Wort

???????



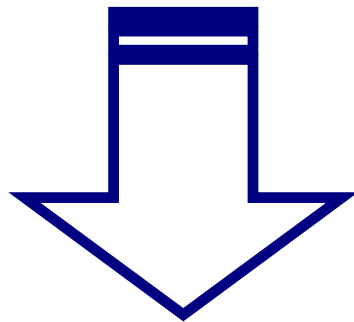
0101001



Does the Hamming Code Work?

sent word

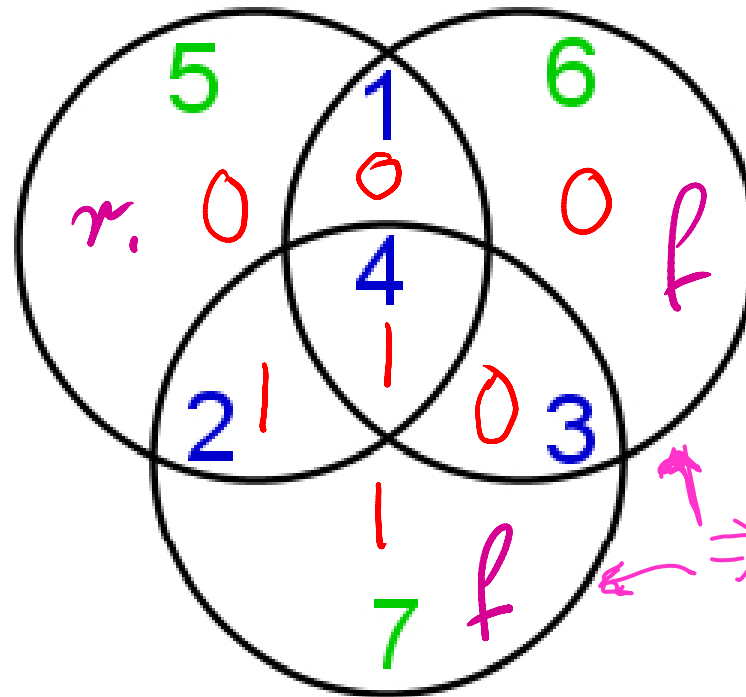
???????



0101001

Correct is

0111001



Yes, yet the old Hamming code can correct single-bit bugs automatically.

Today one has improved bug corrective codes.

Bug Corrective Codes or Why Does the CD not Crackle?



31

Fehlerkorrigierende Codes oder warum CD nicht knackt

