Übungen zur Grundlagen der Technischen Informatik

Übung 2 – Fehlererkennung, Fehlerkorrektur und Huffman

Florian Frank

Friedrich-Alexander-Universität Erlangen-Nürnberg

Wintersemester 2018/19

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Was machen wir heute?

Aufgabe 1 – Hamming-Distanz

Was machen wir heute?

Aufgabe 1 – Hamming-Distanz

Aufgabe 2 – Fehlererkennung

Was machen wir heute?

Aufgabe 1 – Hamming-Distanz

Aufgabe 2 – Fehlererkennung

Aufgabe 3 – Blocksicherung

Was machen wir heute?

Aufgabe 1 – Hamming-Distanz

Aufgabe 2 – Fehlererkennung

Aufgabe 3 – Blocksicherung

Aufgabe 4 – Fehlerkorrektur

design

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG TECHNISCHE FAKULTÄT

Was machen wir heute?

Aufgabe 1 – Hamming-Distanz

Aufgabe 2 – Fehlererkennung

Aufgabe 3 – Blocksicherung

Aufgabe 4 – Fehlerkorrektur

Aufgabe 5 – Huffman-Code

Aufgabe 1 – Hamming-Distanz

design

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Aufgabe 1 – Hamming-Distanz (Grundlagen)

Definition 1 (Hamming-Distanz)

Seien x und y zwei gleich lange Wörter der Länge n. Der Hamming-Abstand $\Delta(x, y)$ ist definiert als:

$$\Delta(x, y) := \left|\{j \in \{1, ..., n\} \mid x_j \neq y_j\}\right|$$

Aufgabe 1 – Hamming-Distanz (Grundlagen)

Definition 1 (Hamming-Distanz)

Seien x und y zwei gleich lange Wörter der Länge n. Der Hamming-Abstand $\Delta(x, y)$ ist definiert als:

$$\Delta(x, y) := \left|\{j \in \{1, ..., n\} \mid x_j \neq y_j\}\right|$$

In einfacheren Worten: Die Hamming-Distanz gibt die Anzahl an **unterschiedlichen** Binärstellen in zwei **gleichlangen** Codewörtern an.

Aufgabe 1 – Hamming-Distanz (Grundlagen)

Definition 1 (Hamming-Distanz)

Seien x und y zwei gleich lange Wörter der Länge n. Der Hamming-Abstand $\Delta(x, y)$ ist definiert als:

$$\Delta(x, y) := \left|\{j \in \{1, ..., n\} \mid x_j \neq y_j\}\right|$$

In einfacheren Worten: Die Hamming-Distanz gibt die Anzahl an **unterschiedlichen** Binärstellen in zwei **gleichlangen** Codewörtern an.

Definition 2 (Minimale Hamming-Distanz HD_{min})

Die minimale Hamming-Distanz HD_{min} ist wie folgt definiert:

$$HD_{min} = \min \{\Delta(x_i, x_j) \mid \forall x_i, x_j \in \mathcal{C}\}$$

Aufgabe 1 – Hamming-Distanz (Grundlagen)

Definition 1 (Hamming-Distanz)

Seien x und y zwei gleich lange Wörter der Länge n. Der Hamming-Abstand $\Delta(x, y)$ ist definiert als:

$$\Delta(x, y) := \left|\{j \in \{1, ..., n\} \mid x_j \neq y_j\}\right|$$

In einfacheren Worten: Die Hamming-Distanz gibt die Anzahl an **unterschiedlichen** Binärstellen in zwei **gleichlangen** Codewörtern an.

Definition 2 (Minimale Hamming-Distanz $\mathrm{HD}_{\mathrm{min}}$)

Die minimale Hamming-Distanz $\mathrm{HD}_{\mathrm{min}}$ ist wie folgt definiert:

$$\mathrm{HD}_{\mathrm{min}} = \min\left\{\Delta(x_i, x_j) \mid \forall x_i, x_j \in \mathcal{C}\right\}$$

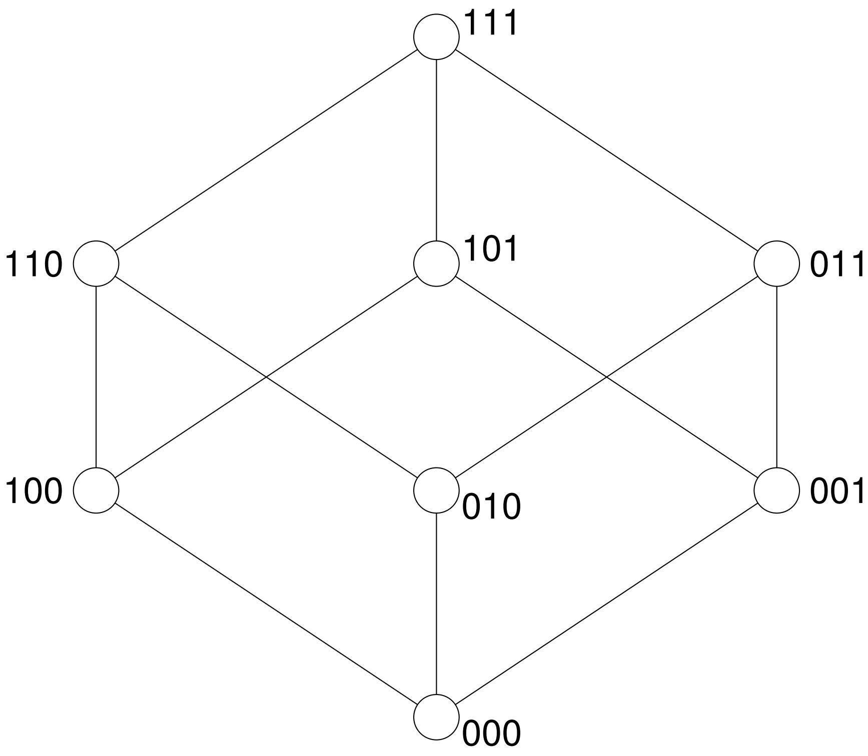
In einfacheren Worten: Die minimale Hamming-Distanz ist das Minimum der Hamming-Distanzen zwischen **allen Codewörtern** eines Codes.

Definition 3 (N-fach Fehler)

Wenn sich durch externe Störeinflüsse bei einem vorher korrekten Binärwort $\mathcal{BW}$ N unterschiedliche Stellen ändern, so spricht man von einem N-fach Fehler.

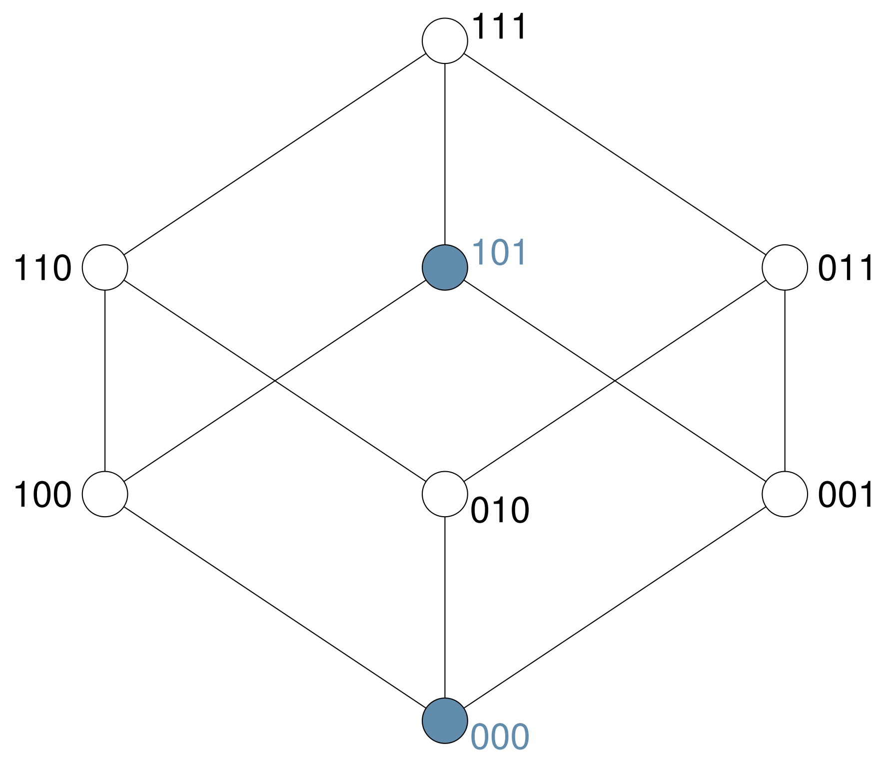
Aufgabe 1 – Hamming-Distanz

Diagramm der Nachbarschaftsbeziehungen für einen Kode mit drei Binärstellen.



Aufgabe 1 – Hamming-Distanz

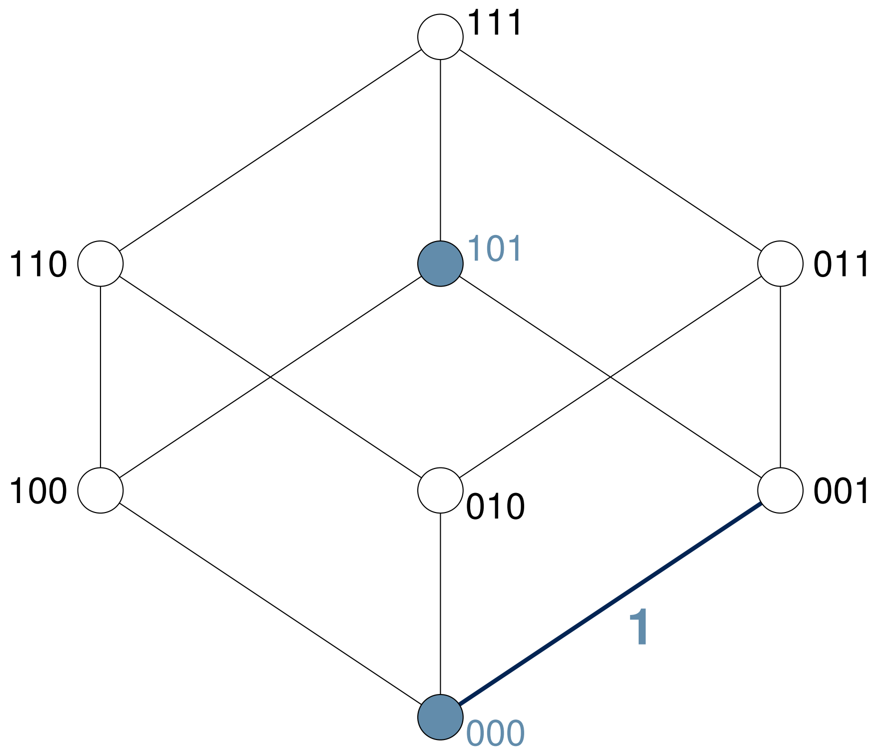
Diagramm der Nachbarschaftsbeziehungen für einen Kode mit drei Binärstellen.



Welche Hamming-Distanz weisen die Kodewörter `000` und `101` auf?

Aufgabe 1 – Hamming-Distanz

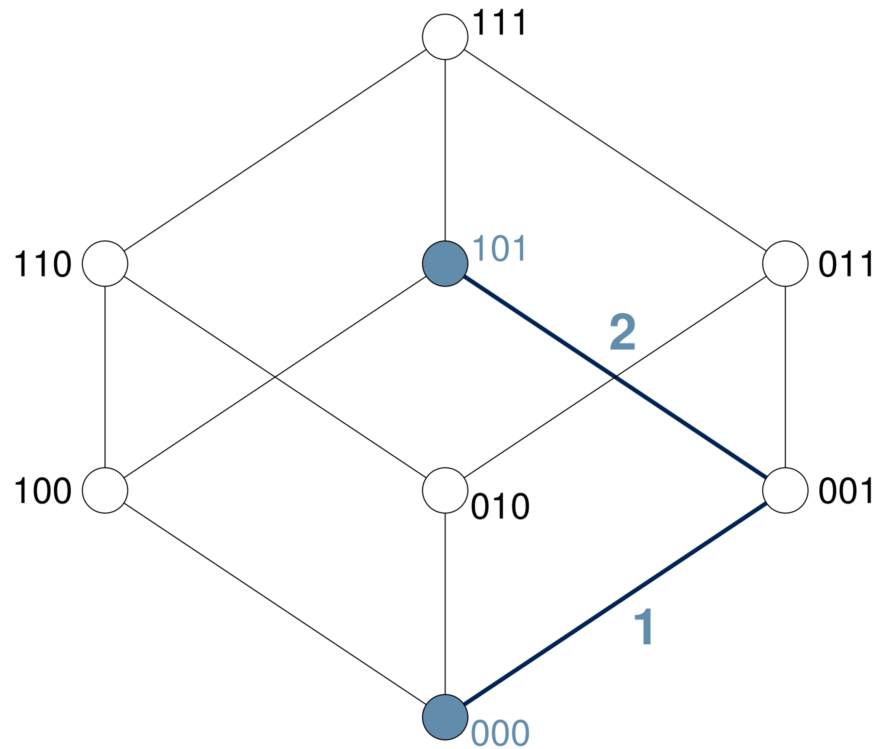
Diagramm der Nachbarschaftsbeziehungen für einen Kode mit drei Binärstellen.



Welche Hamming-Distanz weisen die Kodewörter `000` und `101` auf?

Aufgabe 1 – Hamming-Distanz

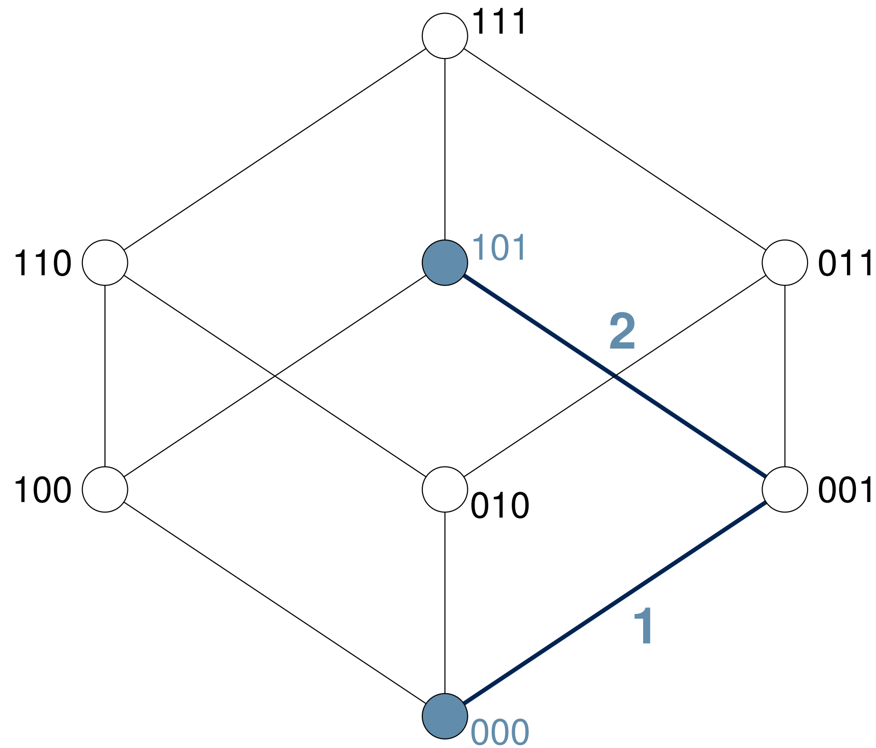
Diagramm der Nachbarschaftsbeziehungen für einen Kode mit drei Binärstellen.



Welche Hamming-Distanz weisen die Kodewörter `000` und `101` auf?

Aufgabe 1 – Hamming-Distanz

Diagramm der Nachbarschaftsbeziehungen für einen Kode mit drei Binärstellen.

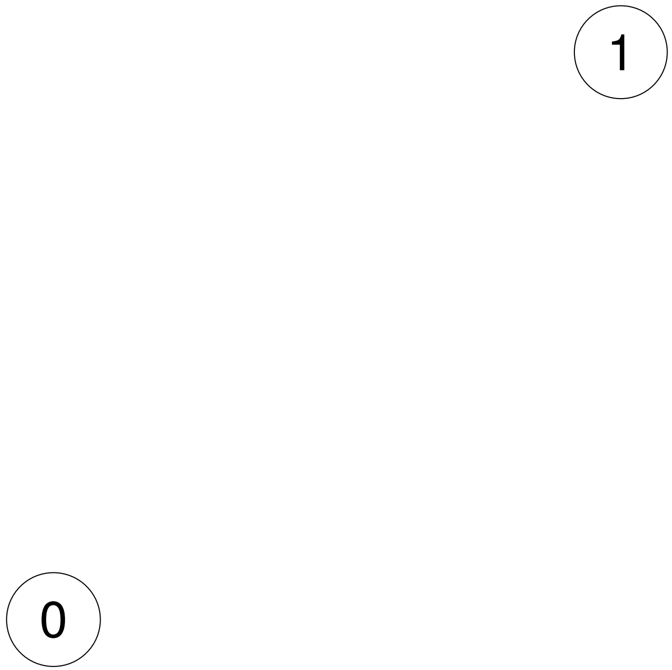


Welche Hamming-Distanz weisen die Kodewörter `000` und `101` auf?

HD(`000`,`101`) = 2

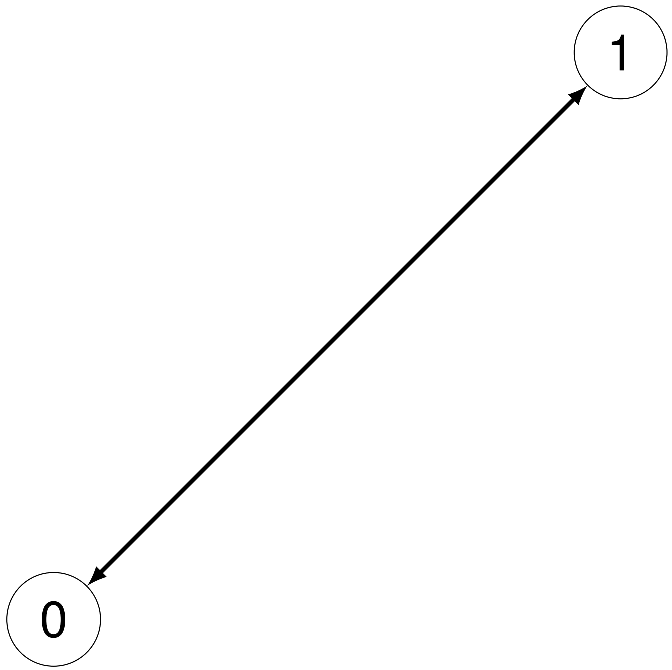
Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



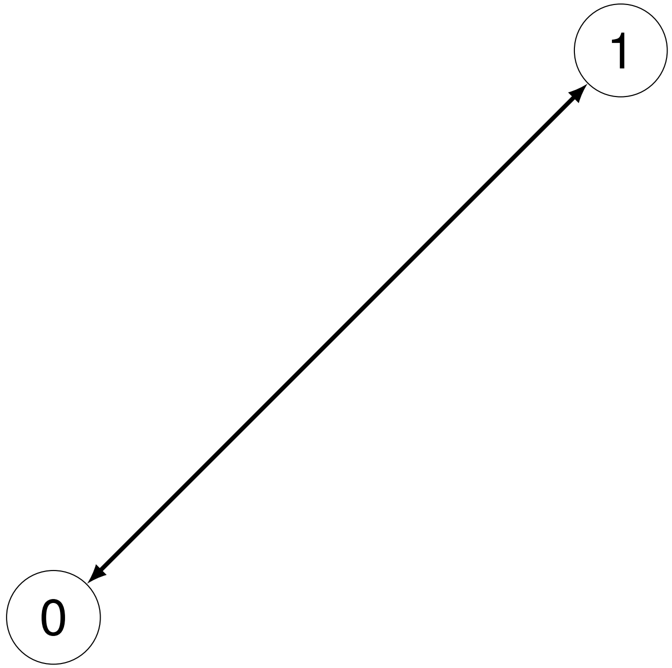
Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



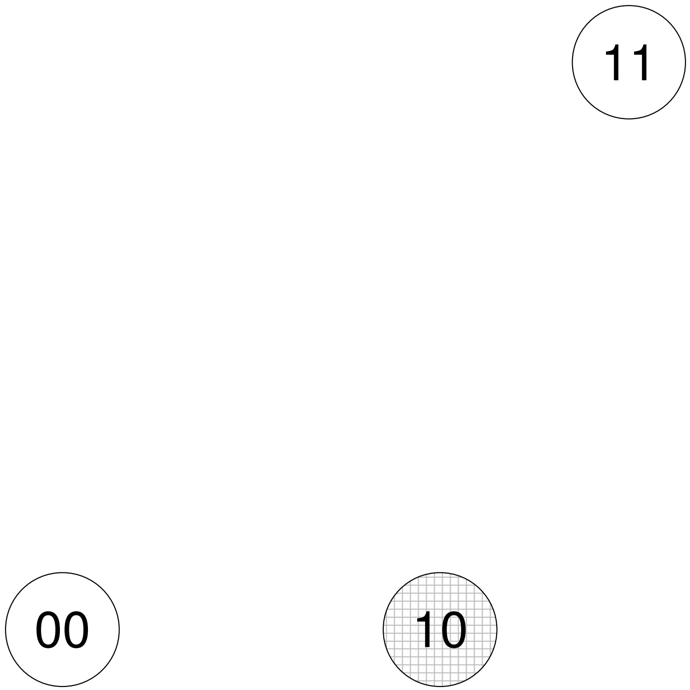
HD(0 , 1) = 1

Problem

Durch Bitfehler kommt man wieder zu einem gültigen Kodewort $\longrightarrow$ Bitfehler sind nicht erkennbar.

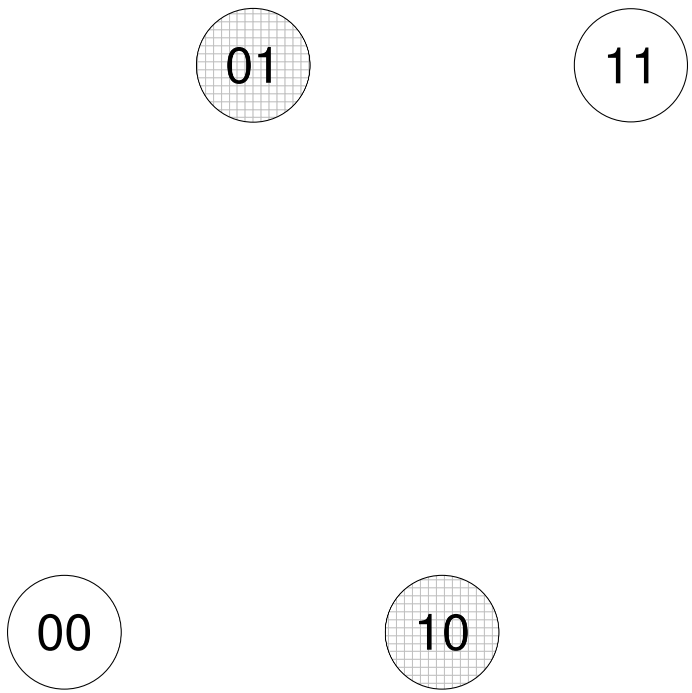
Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



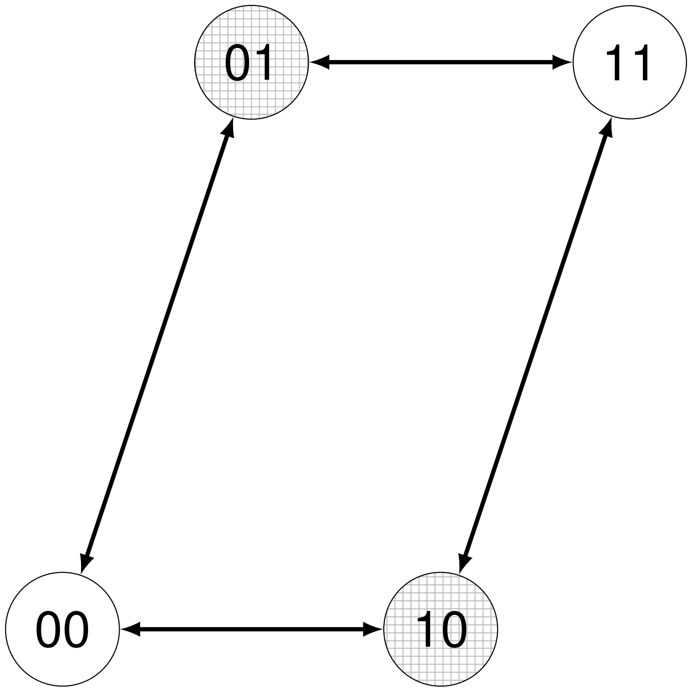
Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



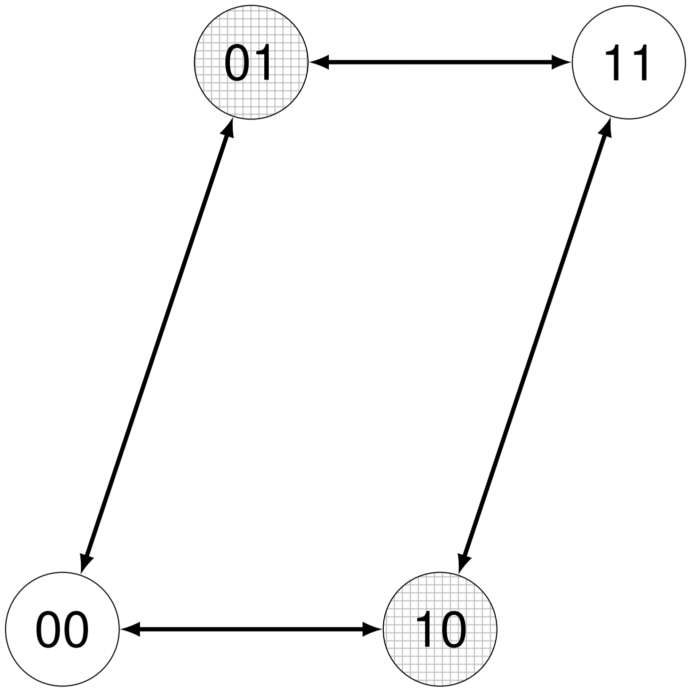
Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?



HD(`00`, `11`) = 2

Erfolg

Einzelfehler sind nun erkennbar, da sie von den gültigen Kodewörtern auf ungültige führen (in der Darstellung gekreuzelte).

Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?
Sie müssen eine minimale Hamming-Distanz von 2 aufweisen.
Allgemein:

Erkennung von Fehlern

Um einen n-fach Fehler bei der Übertragung zu erkennen, ist eine minimale Hamming-Distanz von $n+1$ erforderlich.
In anderen Worten: Sei d die minimale Hamming-Distanz, so kann der Kode $d-1$-fach Fehler erkennen, jedoch **nicht** korrigieren.

Wie viele Zeichen können so mit drei Binärstellen maximal kodiert werden?
Wir betrachten das Startwort mit 3 Stellen.

Aufgabe 1 – Hamming-Distanz

a) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler erkannt werden können?
Sie müssen eine minimale Hamming-Distanz von 2 aufweisen.
Allgemein:

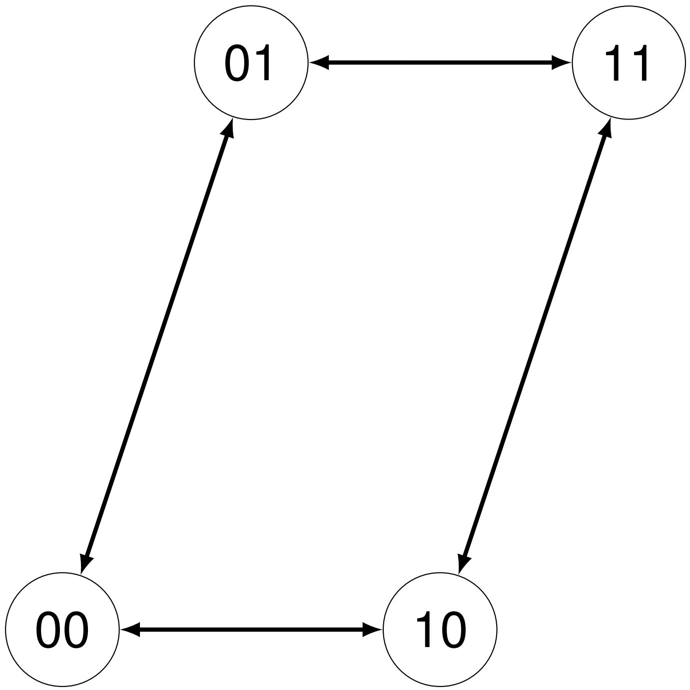
Erkennung von Fehlern

Um einen n-fach Fehler bei der Übertragung zu erkennen, ist eine minimale Hamming-Distanz von $n+1$ erforderlich.
In anderen Worten: Sei d die minimale Hamming-Distanz, so kann der Kode $d-1$-fach Fehler erkennen, jedoch **nicht** korrigieren.

Wie viele Zeichen können so mit drei Binärstellen maximal kodiert werden?
Wir betrachten das Startwort mit 3 Stellen. Die Möglichkeiten bei drei Binärstellen zwei Binärstellen zu verändern liegen bei $\binom{3}{2} = 3$.
Zusammen mit dem Ausgangswort macht das 4 Wörter.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



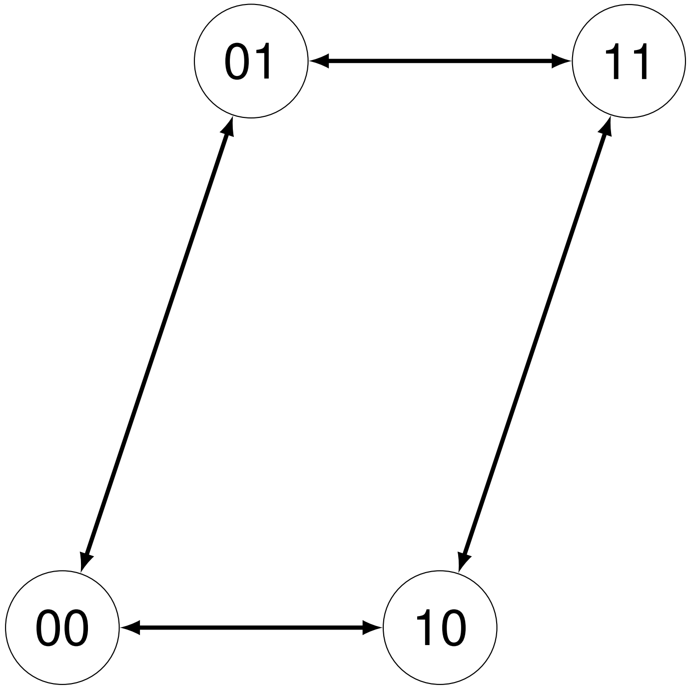
HD(`00`, `11`) = 2

Problem

Einzelfehler sind nun zwar erkennbar, jedoch noch nicht korrigierbar.

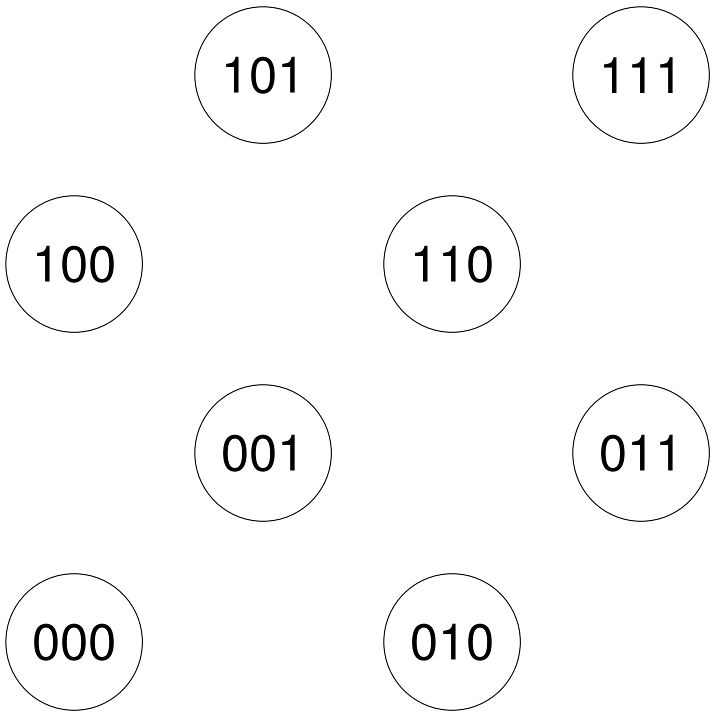
Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



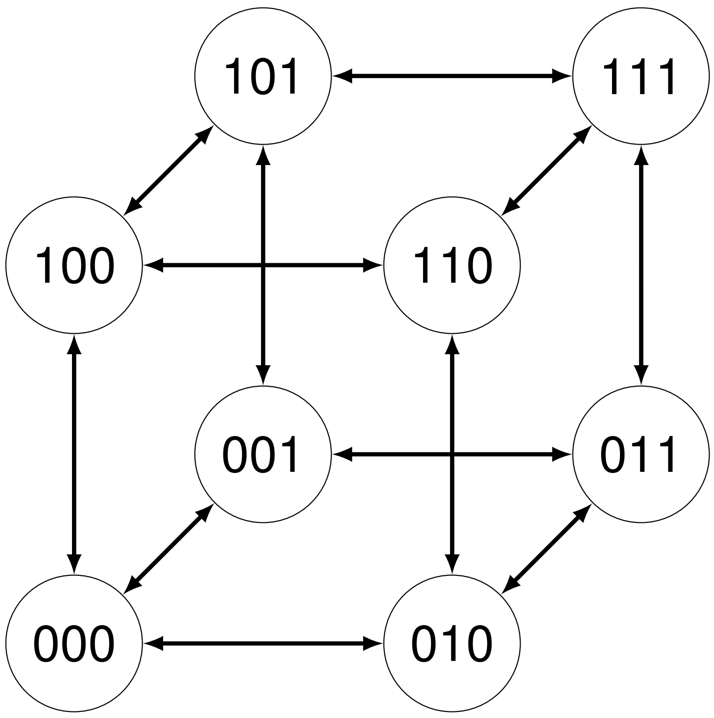
Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



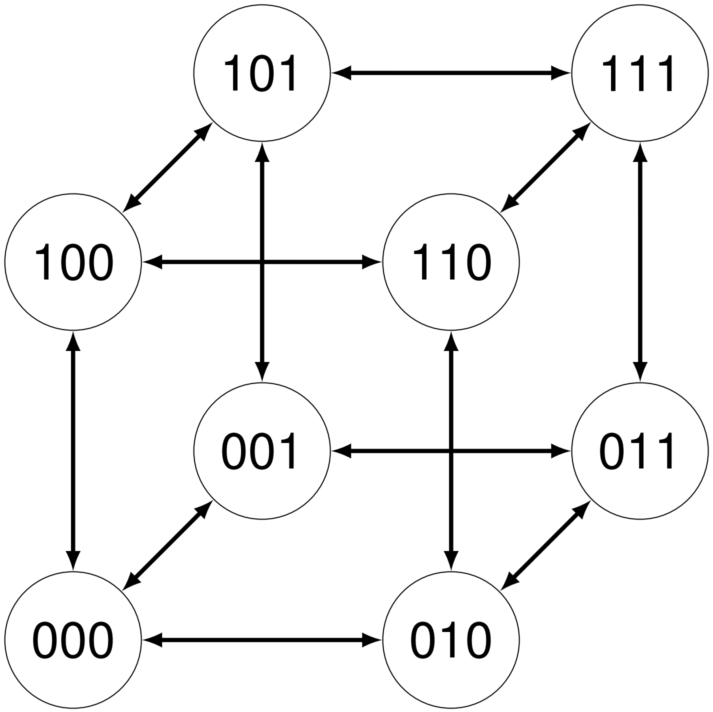
Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



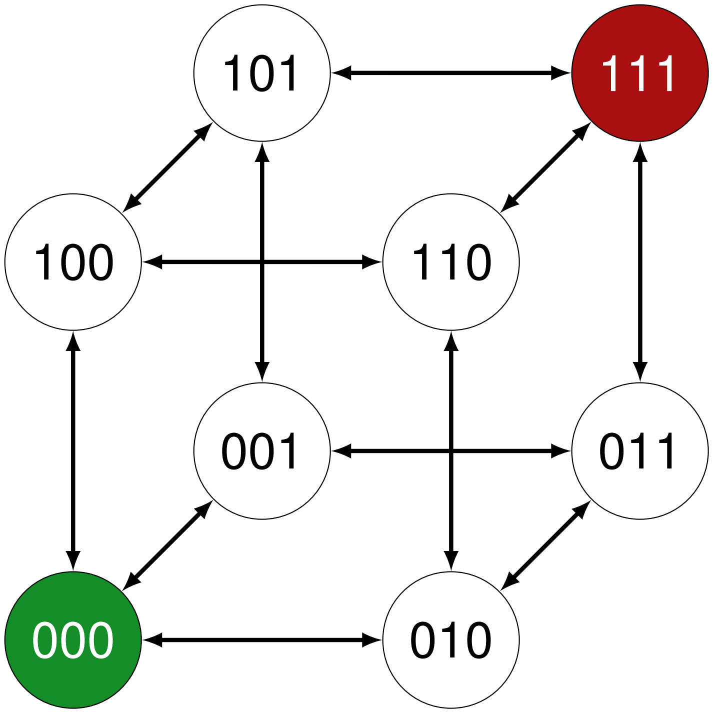
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



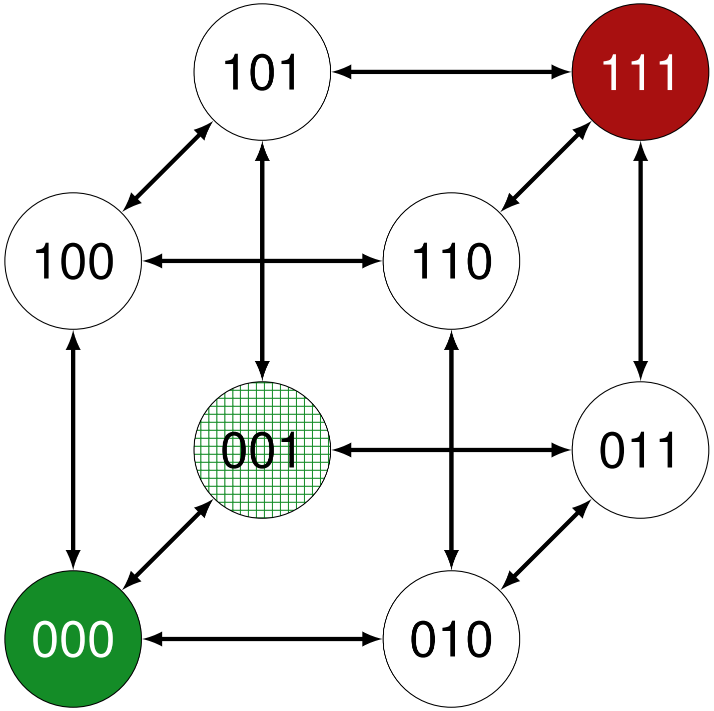
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



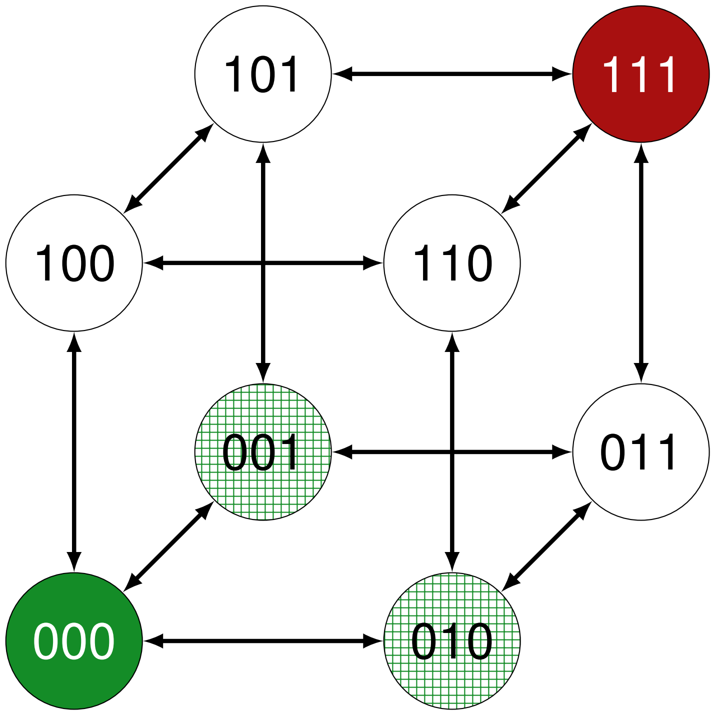
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



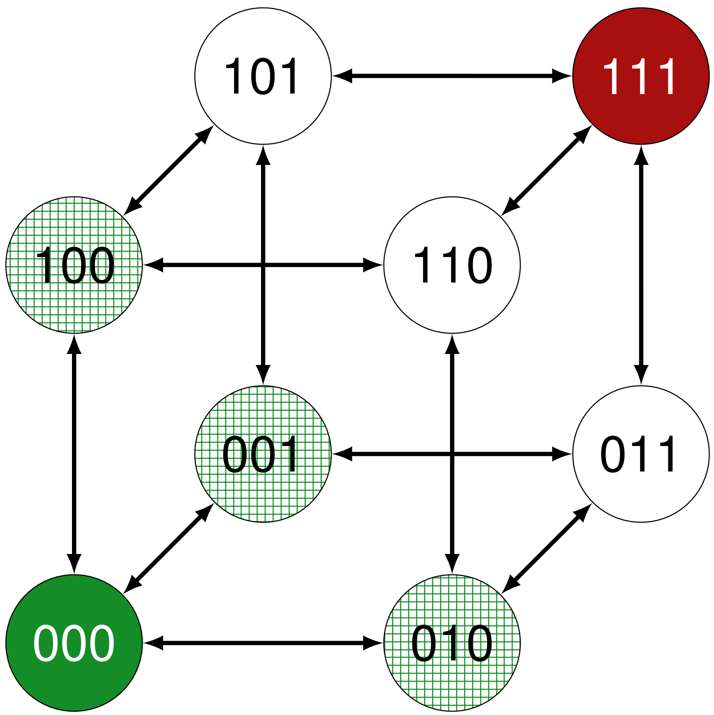
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



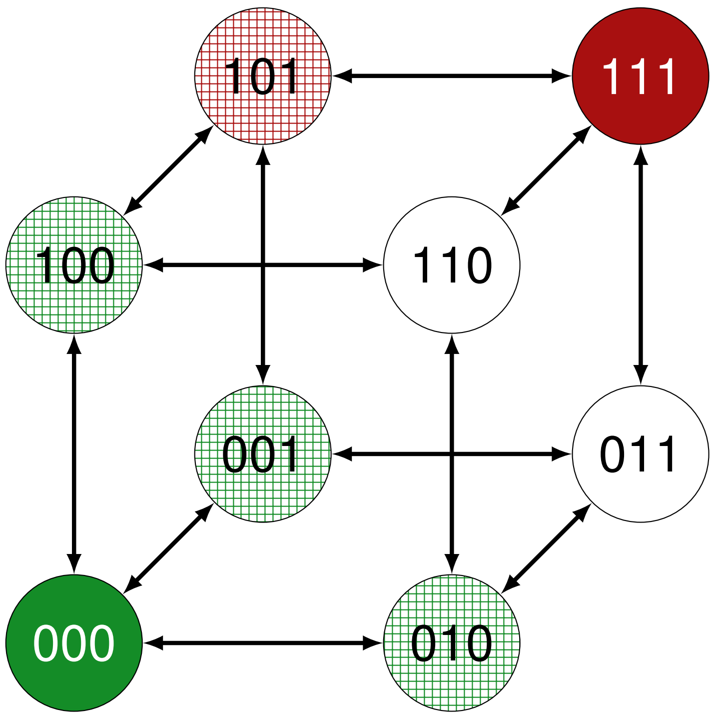
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



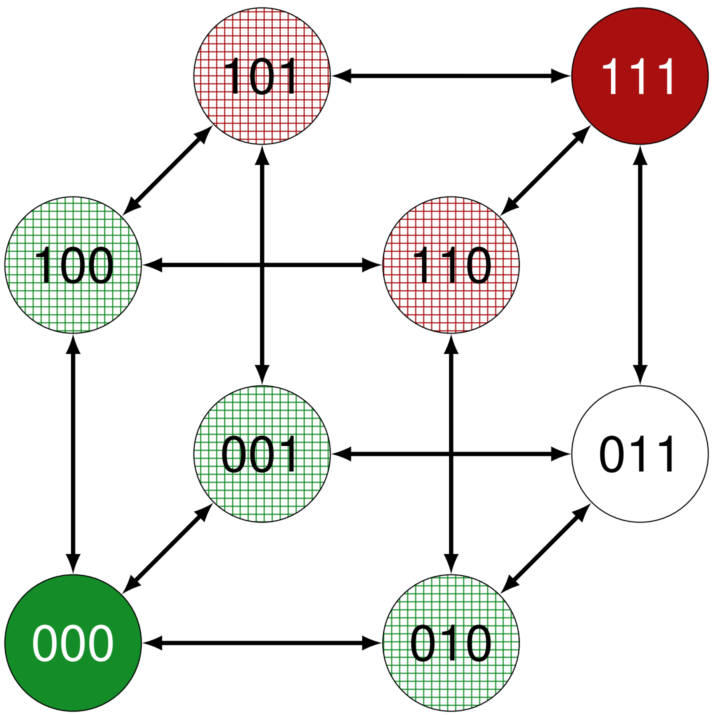
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



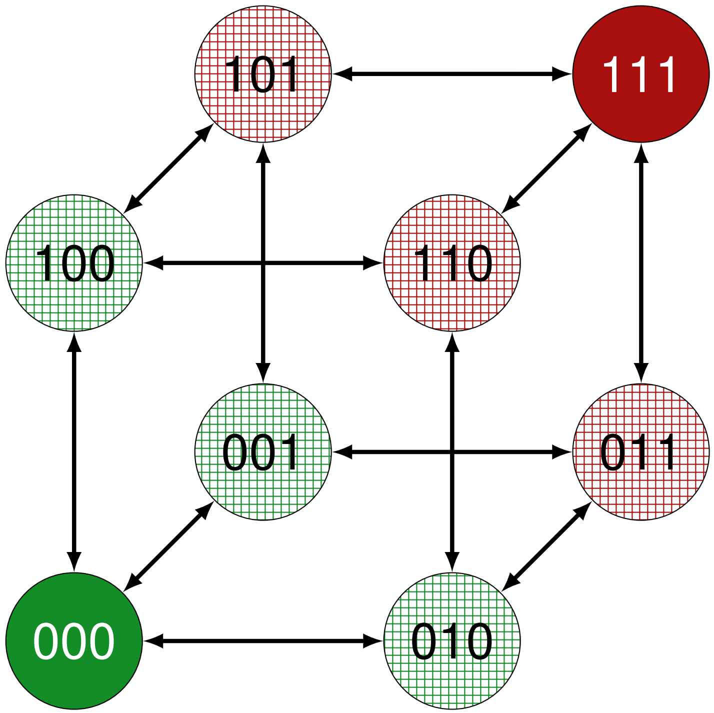
HD(000,111) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?



HD(`000`,`111`) = 3

Erfolg

Einzelfehler sind nun sogar korrigierbar, da man sie jeweils einem gültigen Kodewort zuordnen kann.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?
Sie müssen eine minimale Hamming-Distanz von 3 aufweisen.
Allgemein:

Korrektur von Fehlern

Um einen n-fach Fehler bei der Übertragung zu korrigieren, ist eine minimale Hamming-Distanz von $2n + 1$ erforderlich.
In anderen Worten: Sei d die minimale Hamming-Distanz, so kann der Kode $\left\lfloor \frac{d-1}{2} \right\rfloor$-fach Fehler korrigieren.

Wie viele Zeichen können so mit drei Binärstellen maximal kodiert werden?
Wir betrachten das Startwort mit 3 Stellen.

Aufgabe 1 – Hamming-Distanz

b) Welche (minimale) Hamming-Distanz müssen die gültigen Kodewörter aufweisen, damit Einzelfehler korrigiert werden können?
Sie müssen eine minimale Hamming-Distanz von 3 aufweisen.
Allgemein:

Korrektur von Fehlern

Um einen n-fach Fehler bei der Übertragung zu korrigieren, ist eine minimale Hamming-Distanz von $2n + 1$ erforderlich.
In anderen Worten: Sei d die minimale Hamming-Distanz, so kann der Kode $\left\lfloor \frac{d-1}{2} \right\rfloor$-fach Fehler korrigieren.

Wie viele Zeichen können so mit drei Binärstellen maximal kodiert werden?
Wir betrachten das Startwort mit 3 Stellen. Die Möglichkeiten bei drei Binärstellen zwei Binärstellen zu verändern liegen bei $\binom{3}{3} = 1$.
Zusammen mit dem Ausgangswort macht das 2 Wörter.

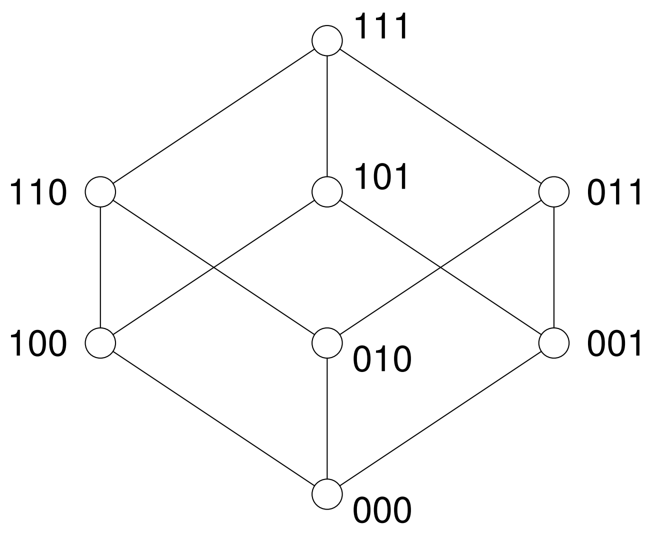
Aufgabe 1 – Hamming-Distanz

c) Die beiden Zeichen *A* und *B* sollen so kodiert werden, dass Einzelfehler korrigierbar sind. Wie viele Lösungen sind für die Kodierung der beiden Zeichen mit drei Binärstellen möglich? Geben Sie eine Lösung an.

d) Bei der Datenübertragung mit einer Kodierung nach c) wurde genau eine Binärstelle falsch übertragen. Die folgenden Daten wurden empfangen:

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

Korrigieren Sie den Fehler.



Aufgabe 2 – Fehlererkennung

Aufgabe 2 – Fehlererkennung

Am Ende einer längeren Übertragungsstrecke wird die folgende Nachricht im ASCII-Code empfangen:

ASCII-Code		Zeichen
0	1 0 0 0 1 1 1	G
1	1 0 1 0 1 0 0	T
1	1 0 0 1 0 0 1	I
1	0 1 0 0 1 0 0	$
0	1 1 0 1 0 0 1	i
1	1 1 1 0 0 1 1	s
0	1 1 1 0 1 0 0	t
1	0 1 0 0 0 0 0	␣
1	1 1 1 0 1 1 0	v
0	1 1 0 1 1 1 1	o
0	1 1 0 1 1 0 0	l
0	1 1 0 1 1 0 0	l

Es ist bekannt, dass der Sender die sieben Bits des ASCII-Codes um ein sogenanntes Paritätsbit (ganz links) ergänzt hat.

a) Welches Zeichen wurde offensichtlich falsch übertragen?

b) Das letzte Wort lautete vor der Übertragung „toll“ und nicht „voll“. Warum ist der von der Übertragungsstrecke verursachte Fehler nicht erkennbar?

Aufgabe 2 – Fehlererkennung: Begriffsklärung (I)

Fehlererkennung mit Paritätsbit

Bei der Fehlererkennung mit Paritätsbits werden zu übertragende Codewörter mit einem zusätlichen Bit gesichert.

Es gibt einen Unterschied zwischen ...

- ... **gerader Parität**: Die Einsen werden auf gerade Anzahl ergänzt. (Ver-XOR-e jede Stelle miteinander).

$$p_B = x_0 \oplus x_1 \oplus \ldots \oplus x_n = \bigoplus_{i=1}^{n} x_i$$

- ... **ungerader Parität**: Die Einsen werden auf ungerade Anzahl ergänzt. (Ver-XNOR-e jede Stelle miteinander).

$$p_B = \overline{x_0 \oplus x_1 \oplus \ldots \oplus x_n} = \overline{\bigoplus_{i=1}^{n}} x_i$$

Aufgabe 2 – Fehlererkennung: Begriffsklärung (II)

ASCII-Kodierung

ASCII $\equiv$ **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange

- Der ASCII-Code kodiert mit 7 Bit insgesamt **128 Zeichen** und reicht damit weitestgehend für die englische Sprache aus (Vorsicht: Keine Sonderzeichen $\rightarrow$ Unicode oder Extended ASCII)
- Die Bitgruppen werden zusammengefasst in die **MSB** (**M**ost **S**ignificant **B**its) und **LSB** (**L**east **S**ignificant **B**its).

$$\underbrace{b_7 b_6 b_5}_{\textbf{MSB}}\ \underbrace{b_4 b_3 b_2 b_1}_{\textbf{LSB}}$$

Aufgabe 2 – Fehlererkennung: Begriffsklärung (II)

MSBs / LSBs	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	NUL	DLE	SP	0	@	P	‘	p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	"	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
0 1 1 1	BEL	ETB	’	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	j	z
1 0 1 1	VT	ESC	+	;	K	[	k	{
1 1 0 0	FF	FS	,	<	L	\	l	|
1 1 0 1	CR	GS	—	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	_	o	DEL

Tabelle 1: ASCII-Kodierung

Aufgabe 2 – Fehlererkennung

Am Ende einer längeren Übertragungsstrecke wird die folgende Nachricht im ASCII-Code empfangen:

ASCII-Code	Zeichen
0 1 0 0 0 1 1 1	G
1 1 0 1 0 1 0 0	T
1 1 0 0 1 0 0 1	I
1 0 1 0 0 1 0 0	$
0 1 1 0 1 0 0 1	i
1 1 1 1 0 0 1 1	s
0 1 1 1 0 1 0 0	t
1 0 1 0 0 0 0 0	␣
1 1 1 1 0 1 1 0	v
0 1 1 0 1 1 1 1	o
0 1 1 0 1 1 0 0	l
0 1 1 0 1 1 0 0	l

Es ist bekannt, dass der Sender die sieben Bits des ASCII-Codes um ein sogenanntes Paritätsbit (ganz links) ergänzt hat.

a) Welches Zeichen wurde offensichtlich falsch übertragen?

b) Das letzte Wort lautete vor der Übertragung „toll“ und nicht „voll“. Warum ist der von der Übertragungsstrecke verursachte Fehler nicht erkennbar?

Aufgabe 3 – Blocksicherung

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Aufgabe 3 – Blocksicherung

Es sollen wichtige Daten im ASCII-Code mit einer Blocksicherung geschützt werden, die gerade Parität für die Prüfbits verwendet. Die folgende Tabelle zeigt die empfangenen Daten, welche offensichtlich nicht alle korrekt übermittelt wurden:

	Binärcode		Prüfbit	ASCII
	1 0 1	0 0 1 0	1	R
Block	1 0 0	1 0 0 1	1	I
1	1 0 1	0 0 1 1	1	S
	0 1 0	1 0 0 0	0	H
Prüfbits	1 1 1	0 0 0 0	1	

	Binärcode		Prüfbit	ASCII
	1 0 1	0 0 0 0	1	P
Block	1 0 0	1 0 0 1	1	I
2	1 0 0	0 1 1 1	0	G
	0 1 0	0 0 0 1	0	!
Prüfbits	1 1 1	1 0 1 1	0	

a) Welche Fehler (Anzahl, Einfach-/Mehrfachfehler) sind korrigierbar?

b) Die aufgetretenen Fehler seien korrigierbar. Korrigieren Sie die entsprechenden Binärstellen in der Tabelle. Bestimmen Sie für die korrigierten Codewörter das zugehörige ASCII-Zeichen.

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.

Die Nachricht wird in Blöcke von je n Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Betrachten wir ein Beispiel mit gerader Parität:

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & \\ 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 1 & \\ 1 & 0 & 0 & 0 & \\ \hline & & & & \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$
\begin{array}{cccc|c}
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & \\
0 & 0 & 0 & 1 & \\
1 & 0 & 0 & 0 & \\
\hline
 & & & & \\
\end{array}
$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$
\begin{array}{cccc|c}
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & \\
1 & 0 & 0 & 0 & \\
\hline
 & & & &
\end{array}
$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

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

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je n Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Betrachten wir ein Beispiel mit gerader Parität:

$$
\begin{array}{cccc|c}
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
\hline
 & & & &
\end{array}
$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$
\begin{array}{cccc|c}
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
\hline
1 & & & &
\end{array}
$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je n Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & & & \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je n Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & & \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je n Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (I)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je n Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.
Betrachten wir ein Beispiel mit gerader Parität:

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar.
Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Wie erkenne ich einen Fehler?

Durch die Schnittpunkte von fehlerhaften Zeilen- und Spaltenparitäten.

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Wie erkenne ich einen Fehler?

Durch die Schnittpunkte von fehlerhaften Zeilen- und Spaltenparitäten.

0	1	0	1	**0**
0	1	0	0	1
0	0	1	1	1
1	0	0	0	1
1	0	0	0	1

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Wie erkenne ich einen Fehler?

Durch die Schnittpunkte von fehlerhaften Zeilen- und Spaltenparitäten.

$$\begin{array}{cccc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & \mathbf{0} & 0 & 0 & 1 \end{array}$$

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Wie erkenne ich einen Fehler?

Durch die Schnittpunkte von fehlerhaften Zeilen- und Spaltenparitäten.

0	1	0	1	0
0	1	0	0	1
0	0	1	1	**1**
1	0	0	0	1
1	0	**0**	0	1

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Wie erkenne ich einen Fehler?

Durch die Schnittpunkte von fehlerhaften Zeilen- und Spaltenparitäten.

0	1	0	1	0
0	1	0	0	1
0	0	1	1	**1**
1	0	0	0	1
1	0	**0**	0	1

Aufgabe 3 – Blocksicherung: Begriffsklärung (II)

Fehlerkorrektur durch Blocksicherung

Eine Erweiterung der normalen Paritätssicherung stellt die Blocksicherung dar. Die Nachricht wird in Blöcke von je *n* Codewörtern mit Paritätsbit eingeteilt. **Zusätzlich** wird am Ende eines jeden Blocks ein weiteres Codewort eingefügt, das alle Paritätsbits der Spalten enthält.

Wie erkenne ich einen Fehler?

Durch die Schnittpunkte von fehlerhaften Zeilen- und Spaltenparitäten.

0	1	0	1	0
0	1	0	0	1
0	0	1	1	**1**
1	0	0	0	1
1	0	**0**	0	1

Konklusion

Es sind nur Einfachfehler pro Block korrigierbar!

Aufgabe 3 – Blocksicherung

Es sollen wichtige Daten im ASCII-Code mit einer Blocksicherung geschützt werden, die gerade Parität für die Prüfbits verwendet. Die folgende Tabelle zeigt die empfangenen Daten, welche offensichtlich nicht alle korrekt übermittelt wurden:

	Binärcode	Prüfbit	ASCII
	1 0 1 0 0 1 0	1	R
Block	1 0 0 1 0 0 1	1	I
1	1 0 1 0 0 1 1	1	S
	0 1 0 1 0 0 0	0	H
Prüfbits	1 1 1 0 0 0 0	1	

	Binärcode	Prüfbit	ASCII
	1 0 1 0 0 0 0	1	P
Block	1 0 0 1 0 0 1	1	I
2	1 0 0 0 1 1 1	0	G
	0 1 0 0 0 0 1	0	!
Prüfbits	1 1 1 1 0 1 1	0	

b) Die aufgetretenen Fehler seien korrigierbar. Korrigieren Sie die entsprechenden Binärstellen in der Tabelle. Bestimmen Sie für die korrigierten Codewörter das zugehörige ASCII-Zeichen.

Aufgabe 4 – Fehlerkorrektur

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Aufgabe 4 – Fehlerkorrektur

Sei ein nicht fehlertolerantes Kommunikationssystem gegeben, das in der Lage ist, einstellige Hexadezimalzahlen zu übertragen. Es soll nun dahingehend erweitert werden, dass es mittels eines Hamming-Codes Zweifachfehler erkennen oder Einfachfehler korrigieren kann.

a) Welche Hamming-Distanz wird benötigt, um die geforderte Fehlertoleranz zu erreichen?

Aufgabe 4 – Fehlerkorrektur

Sei ein nicht fehlertolerantes Kommunikationssystem gegeben, das in der Lage ist, einstellige Hexadezimalzahlen zu übertragen. Es soll nun dahingehend erweitert werden, dass es mittels eines Hamming-Codes Zweifachfehler erkennen oder Einfachfehler korrigieren kann.

b) Wie viele Bits werden benötigt, um die jeweiligen Informationen und die Paritätsbits nach Hamming zu codieren? Wie lang wird das gesamte zu übertragende Codewort?

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (I)

Hamming-Code

Der Hamming-Code ist ein Beispiel für einen Code mit $HD_{min} = 3$. Dabei werden Prüfsummen nur auf Teilwörtern generiert.

Ein Algorithmus für die Erzeugung eines Hamming-Codes:

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.
   Äquivalent dazu ($x_k(i)$ bezeichnet das $i$te Bit des Wortes x_k, d ist die Anzahl an Datenbits):

$$y_i = \bigoplus_{\substack{k=1 \\ x_k(i)=1}}^{d} x_k$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile
4
3
2
1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	1
4	0
3	0
2	0
1	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	2	1
4	0	0
3	0	0
2	1	0
1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	3	2	1
4	0	0	0
3	0	0	0
2	1	1	0
1	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	4	3	2	1
4	0	0	0	0
3	1	0	0	0
2	0	1	1	0
1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.

Zeile	5	4	3	2	1
4	0	0	0	0	0
3	1	1	0	0	0
2	0	0	1	1	0
1	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.

Zeile	6	5	4	3	2	1
4	0	0	0	0	0	0
3	1	1	1	0	0	0
2	1	0	0	1	1	0
1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.

Zeile	7	6	5	4	3	2	1
4	0	0	0	0	0	0	0
3	1	1	1	1	0	0	0
2	1	1	0	0	1	1	0
1	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	8	7	6	5	4	3	2	1
4	1	0	0	0	0	0	0	0
3	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.

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

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.

Zeile	10	9	8	7	6	5	4	3	2	1
4	1	1	1	0	0	0	0	0	0	0
3	0	0	0	1	1	1	1	0	0	0
2	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	11	10	9	8	7	6	5	4	3	2	1
4	1	1	1	1	0	0	0	0	0	0	0
3	0	0	0	0	1	1	1	1	0	0	0
2	1	1	0	0	1	1	0	0	1	1	0
1	1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.

Zeile	12	11	10	9	8	7	6	5	4	3	2	1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.

Zeile	12	11	10	9	8	7	6	5	4	3	2	1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 =$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$

$$y_2 =$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$

$$y_2 = x_1$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 =$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2 \oplus x_3$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2 \oplus x_3 \oplus x_4 \oplus x_8$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: $(12, 8)$ Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2 \oplus x_3 \oplus x_4 \oplus x_8$$
$$y_4 =$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$
\begin{aligned}
y_1 &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7 \\
y_2 &= x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7 \\
y_3 &= x_2 \oplus x_3 \oplus x_4 \oplus x_8 \\
y_4 &= x_5
\end{aligned}
$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$
\begin{aligned}
y_1 &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7 \\
y_2 &= x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7 \\
y_3 &= x_2 \oplus x_3 \oplus x_4 \oplus x_8 \\
y_4 &= x_5 \oplus x_6
\end{aligned}
$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2 \oplus x_3 \oplus x_4 \oplus x_8$$
$$y_4 = x_5 \oplus x_6 \oplus x_7$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (II)

Beispiel: (12, 8) Hamming-Code

3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Zeile	x_8	x_7	x_6	x_5	y_4	x_4	x_3	x_2	y_3	x_1	y_2	y_1
4	1	1	1	1	1	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0	0	0
2	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1

$$y_1 = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7$$
$$y_2 = x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7$$
$$y_3 = x_2 \oplus x_3 \oplus x_4 \oplus x_8$$
$$y_4 = x_5 \oplus x_6 \oplus x_7 \oplus x_8$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (III)

Hamming-Code

Aus dem Bildungsalgorithmus folgt damit unmittelbar:

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (III)

Hamming-Code

Aus dem Bildungsalgorithmus folgt damit unmittelbar:
Bei m Datenbits x_i werden k Prüfbits y_i zur Bildung des Hamming-Codes benötigt und es gilt im Allgemeinen:

$$2^k - k - 1 \geqslant m$$

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (III)

Hamming-Code

Aus dem Bildungsalgorithmus folgt damit unmittelbar:
Bei m Datenbits x_i werden k Prüfbits y_i zur Bildung des Hamming-Codes benötigt und es gilt im Allgemeinen:

$$2^k - k - 1 \geqslant m$$

$\rightsquigarrow$ Denn es gilt für die maximale Gesamtzahl der Bits $n = 2^k - 1$.

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (III)

Hamming-Code

Aus dem Bildungsalgorithmus folgt damit unmittelbar:
Bei m Datenbits x_i werden k Prüfbits y_i zur Bildung des Hamming-Codes benötigt und es gilt im Allgemeinen:

$$2^k - k - 1 \geqslant m$$

$\rightsquigarrow$ Denn es gilt für die maximale Gesamtzahl der Bits $n = 2^k - 1$.

$\rightsquigarrow$ Da man aber noch die k Prüfbits braucht, gilt für die Datenbits m die obere Schranke $2^k - \mathbf{k} - 1$.

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (III)

Hamming-Code

Aus dem Bildungsalgorithmus folgt damit unmittelbar:
Bei m Datenbits x_i werden k Prüfbits y_i zur Bildung des Hamming-Codes benötigt und es gilt im Allgemeinen:

$$2^k - k - 1 \geqslant m$$

- ⇝ Denn es gilt für die maximale Gesamtzahl der Bits $n = 2^k - 1$.
- ⇝ Da man aber noch die k Prüfbits braucht, gilt für die Datenbits m die obere Schranke $2^k - \mathbf{k} - 1$.
- ⇝ Weil wir möglichst mit der geringsten Bitanzahl übertragen wollen, suchen wir die kleinste obere Schranke, damit das kleinste k für das die Ungleichung erfüllt ist!

Aufgabe 4 – Fehlerkorrektur: Begriffsklärung (IV)

Hamming-Code

In algorithmischer Form:

Algorithmus 1 : Anzahl an Prüfbits aus Anzahl an Datenbits berechnen

Eingabe : Anzahl an Datenbits m

Ausgabe : Minimale Anzahl an Prüfbits k

int $k \leftarrow 0$;

solange $2^k - k - 1 < m$ **tue**

  $k \leftarrow k + 1$;

Gib k **aus**;

Aufgabe 4 – Fehlerkorrektur

Sei ein nicht fehlertolerantes Kommunikationssystem gegeben, das in der Lage ist, einstellige Hexadezimalzahlen zu übertragen. Es soll nun dahingehend erweitert werden, dass es mittels eines Hamming-Codes Zweifachfehler erkennen oder Einfachfehler korrigieren kann.

b) Wie viele Bits werden benötigt, um die jeweiligen Informationen und die Paritätsbits nach Hamming zu codieren? Wie lang wird das gesamte zu übertragende Codewort?

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	1
1	1
2	0
3	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	2	1
1	0	1
2	1	0
3	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	3	2	1
1	1	0	1
2	1	1	0
3	0	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	4	3	2	1
1	0	1	0	1
2	0	1	1	0
3	1	0	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	5	4	3	2	1
1	1	0	1	0	1
2	0	0	1	1	0
3	1	1	0	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	6	5	4	3	2	1
1	0	1	0	1	0	1
2	1	0	0	1	1	0
3	1	1	1	0	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	7	6	5	4	3	2	1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 =$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 =$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller x = 1-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 =$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$

$$y_2 = x_1 \oplus x_3 \oplus x_4$$

$$y_3 = x_2$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis n binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe i.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3$$

Aufgabe 4 – Fehlerkorrektur – Beispiel

Algorithmus zum Erstellen eines Hamming-Code

1. Schreibe die natürlichen Zahlen von 1 bis *n* binär in absteigender Reihenfolge.
2. Die Spalten mit genau einer 1 (also die 2-er Potenzen) werden zu Paritätsbits, der Rest zu Datenbits.
3. Jedes Paritätsbit y_i ist das Ergebnis der Ver-XOR-ung aller $x = 1$-Komponenten seiner Reihe *i*.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

Zeile	x_4	x_3	x_2	y_3	x_1	y_2	y_1
1	1	0	1	0	1	0	1
2	1	1	0	0	1	1	0
3	1	1	1	1	0	0	0

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0				0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0				0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0			0	0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0			0	0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0		0	0	0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0		0	0	0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$
$$y_3 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$
$$y_3 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1			1	1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$

$$y_2 = x_1 \oplus x_3 \oplus x_4$$

$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1			1	1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

$$y_2 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1		1	1	1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

$$y_2 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1		1	1	1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$
$$y_2 = 1 \oplus 0 \oplus 0 = 1$$
$$y_3 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$
$$y_2 = 1 \oplus 0 \oplus 0 = 1$$
$$y_3 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0				2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0			1	2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0			1	2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 0 = 1$$

$$y_2 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0		0	1	2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 0 = 1$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0		0	1	2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 0 = 1$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$
$$y_3 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 0 = 1$$
$$y_2 = 0 \oplus 0 \oplus 0 = 0$$
$$y_3 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1				3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1			0	3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1			0	3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$

$$y_2 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1		1	0	3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$

$$y_2 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1		1	0	3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$
$$y_2 = 1 \oplus 0 \oplus 0 = 1$$
$$y_3 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$
$$y_2 = 1 \oplus 0 \oplus 0 = 1$$
$$y_3 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7,4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0				4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0			0	4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0			0	4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0		1	0	4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0		1	0	4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$
$$y_3 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$
$$y_3 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1				5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1			1	5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1			1	5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

$$y_2 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1		0	1	5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$

$$y_2 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1		0	1	5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$
$$y_2 = 1 \oplus 1 \oplus 0 = 0$$
$$y_3 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 0 = 1$$
$$y_2 = 1 \oplus 1 \oplus 0 = 0$$
$$y_3 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0				6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0			1	6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0			1	6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0		1	1	6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$

$$y_2 = x_1 \oplus x_3 \oplus x_4$$

$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$

$$y_2 = 0 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0		1	1	6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$
$$y_3 = 1 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 0 = 0$$
$$y_2 = 0 \oplus 1 \oplus 0 = 1$$
$$y_3 = 1 \oplus 1 \oplus 0 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1				7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1			0	7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1			0	7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$
$$y_2 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1		0	0	7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$
$$y_2 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1		0	0	7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$
$$y_2 = 1 \oplus 1 \oplus 0 = 0$$
$$y_3 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1	0	0	0	7

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 0 = 0$$
$$y_2 = 1 \oplus 1 \oplus 0 = 0$$
$$y_3 = 1 \oplus 1 \oplus 0 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0				8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0			1	8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7,4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0			1	8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0		1	1	8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0		1	1	8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$
$$y_3 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7,4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0		1	1	8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$
$$y_3 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0		1	1	8
1	0	0	1				9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1			0	9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1			0	9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1		0	0	9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7,4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1		0	0	9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$
$$y_3 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$
$$y_3 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0				A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0			0	A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0			0	A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 1 = 0$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0		1	0	A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 1 = 0$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0		1	0	A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 1 = 0$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$
$$y_3 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 1 \oplus 1 = 0$$
$$y_2 = 0 \oplus 0 \oplus 1 = 1$$
$$y_3 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1				B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1			1	B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1			1	B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1		0	1	B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1		0	1	B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$
$$y_3 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 0 \oplus 1 = 0$$
$$y_3 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0				C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0			1	C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0			1	C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0		0	1	C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0		0	1	C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$
$$y_3 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$
$$y_3 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1				D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1			0	D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1			0	D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1		1	0	D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1		1	0	D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$
$$y_3 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 0 \oplus 1 = 0$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$
$$y_3 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0				E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0			0	E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0			0	E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0		0	0	E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0		0	0	E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$
$$y_3 = 1 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7,4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 0 \oplus 0 \oplus 1 = 1$$
$$y_2 = 0 \oplus 1 \oplus 1 = 0$$
$$y_3 = 1 \oplus 1 \oplus 1 = 0$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1				F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1			1	F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1			1	F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1		1	1	F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.

Lösung: Wir brauchen einen (7, 4)-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1		1	1	F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$
$$y_3 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \ldots x_1 y_k \ldots y_1$.

Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1	1	1	1	F

$$y_1 = x_1 \oplus x_2 \oplus x_4$$
$$y_2 = x_1 \oplus x_3 \oplus x_4$$
$$y_3 = x_2 \oplus x_3 \oplus x_4$$

$$y_1 = 1 \oplus 1 \oplus 1 = 1$$
$$y_2 = 1 \oplus 1 \oplus 1 = 1$$
$$y_3 = 1 \oplus 1 \oplus 1 = 1$$

Aufgabe 4 – Fehlerkorrektur

c) Erstellen Sie nun den Hamming-Code und ordnen Sie den Codewörtern die entsprechenden Hexadezimalwerte zu, die der Wertigkeit der Informationsstellen entsprechen sollen. Der Aufbau der Codewörter soll wie folgt aussehen: $x_m \dots x_1 y_k \dots y_1$.
Lösung: Wir brauchen einen $(7, 4)$-Hamming-Code

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1	0	0	0	7

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1	1	1	1	F

Aufgabe 4 – Fehlerkorrektur

Bei einer Übertragung mit diesem Kommunikationssystem wurde folgende Binärfolge empfangen:

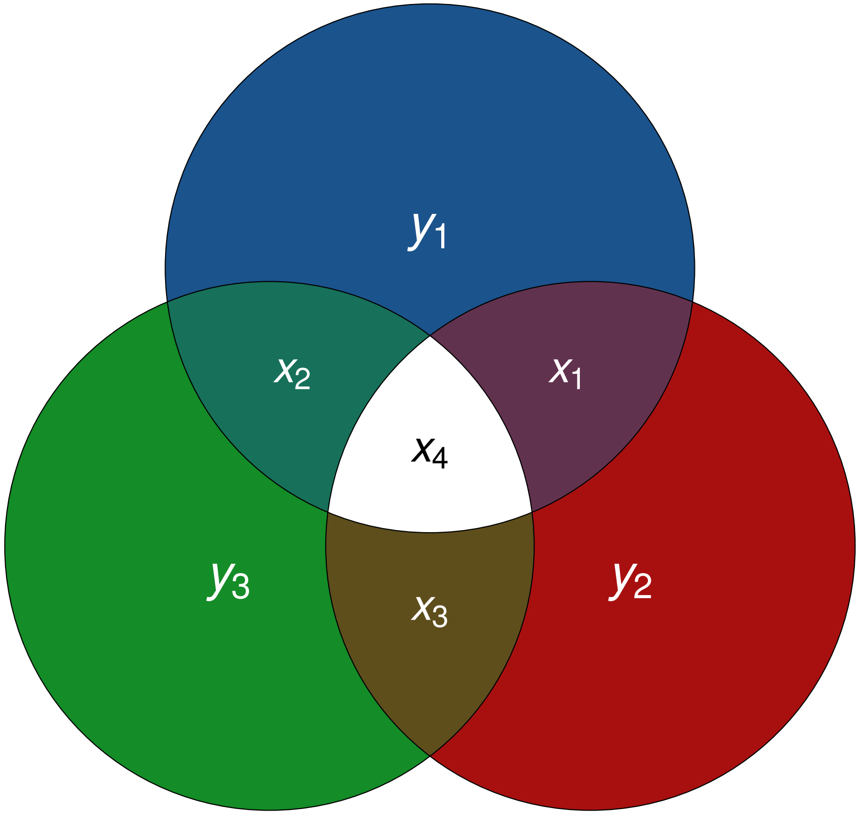
0 1 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 0 1 0 0

d) Überprüfen Sie anhand Ihrer Code-Tabelle, ob der Empfang der Codewörter fehlerfrei erfolgt ist, und führen Sie falls notwendig eine Korrektur durch.

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	2
0	0	1	1	1	1	0	3
0	1	0	0	1	1	0	4
0	1	0	1	1	0	1	5
0	1	1	0	0	1	1	6
0	1	1	1	0	0	0	7

x_4	x_3	x_2	x_1	y_3	y_2	y_1	HEX
1	0	0	0	1	1	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	1	0	A
1	0	1	1	0	0	1	B
1	1	0	0	0	0	1	C
1	1	0	1	0	1	0	D
1	1	1	0	1	0	0	E
1	1	1	1	1	1	1	F

Aufgabe 4 – Fehlerkorrektur: Hamming-Code im Venndiagramm



Aufgabe 5 – Huffman-Code

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Aufgabe 5 – Huffman-Code

a) Erstellen Sie einen Huffman-Codierungsbaum für die folgende Zeichenkette:

`ABRAKADABRASIMSALABIM`

b) Wieviel Bits werden durch diese Codierung im Vergleich zu einer Codierung mit einer festen Codewort-Länge eingespart? (Das Codebuch ist zu vernachlässigen.)

c) Wieviel Bits sind minimal nötig (optimale Codierung)? Wieviel Prozent schlechter ist der Huffman-Code?

Aufgabe 5 – Huffman-Code

Definition 4 (Optimale Codes)

Wir bezeichnen einen Code als optimal bzgl. einer Wahrscheinlichkeitsverteilung p, wenn die durchschnittliche Codewortlänge minimal sind.
Die **durchschnittliche Codewortlänge** $\overline{m}$ berechnet sich durch:

$$\overline{m} = \sum_{i=1}^{n} p(x_i) \cdot m(x_i)$$

Der Idealwert eines optimalen Codes bezeichnen wir als **Entropie** und berechnet sich durch:

$$H = \sum_{i=1}^{n} p(x_i) \cdot I(x_i)$$

Aufgabe 5 – Huffman-Code

Definition 5 (Huffman-Code)

Der Huffman-Code stellt einen nahezu optimalen, präfixfreien Code dar, bei dem eine Codierung mit variabler Bitlänge verwendet wird.

Huffman-Kodierungs-Algorithmus[a]:

Schritt 1: Sortiere die vorkommenden Zeichen der zu codierenden Nachricht N **aufsteigend** nach ihrer Häufigkeit ($\equiv$ Liste Q).

Schritt 2: Finde aus der sortierten Liste Q die beiden Minimas z_l und z_r.

Schritt 3: Verschmelze z_l und z_r zu einem neuen Element z. Die Häufigkeit von z ist die Summe der Häufigkeiten von z_l und z_r.

Schritt 4: Sortiere z in Q gemäß seiner Häufigkeit ein.

Schritt 5: Ist nur noch ein Element in Q vorhanden, so ist dieses Element die Wurzel des Codierungsbaums, breche den Algorithmus ab.

Schritt 6: Sonst: Springe zu Schritt 2.

[a] nach Folie 09-16f.

Aufgabe 5 – Huffman-Code

a) Erstellen Sie einen Huffman-Codierungsbaum für die folgende Zeichenkette:

`ABRAKADABRASIMSALABIM`

Aufgabe 5 – Huffman-Code

Sei folgende Kodierung nun gegeben:

i	Zeichen x_i	Code	Anzahl N_i
1	A	11	7
2	B	101	3
3	I	000	2
4	M	001	2
5	R	010	2
6	S	011	2
7	L	1000	1
8	K	10010	1
9	D	10011	1

b) Wieviel Bits werden durch diese Codierung im Vergleich zu einer Codierung mit einer festen Codewort-Länge eingespart? (Das Codebuch ist zu vernachlässigen.)

Aufgabe 5 – Huffman-Code

Seien die Häufigkeiten nun nochmal gegeben:

Buchstabe	A	B	I	M	R	S	L	K	D
Anzahl	7	3	2	2	2	2	1	1	1

c) Wieviel Bits sind minimal nötig (optimale Codierung)? Wieviel Prozent schlechter ist der Huffman-Code?

Aufgabe 5 – Huffman-Code

Seien die Häufigkeiten nun nochmal gegeben:

Buchstabe	A	B	I	M	R	S	L	K	D
Anzahl	7	3	2	2	2	2	1	1	1

c) Wieviel Bits sind minimal nötig (optimale Codierung)? Wieviel Prozent schlechter ist der Huffman-Code?

Optimale Codierung

Die theoretisch minimale Anzahl an Bits zur Codierung eines Zeichens x entspricht dessen Informationsgehalt

$$I(x) = -\,ld\left(\frac{\text{Anzahl}(x)}{\text{Gesamtzeichenanzahl}}\right)$$