

8 Externspeicher

8.1 Überblick

8.2 Magnetplatten

8.3 Codierung

8.4 Platten-Scheduling

8.5 RAID

8.6 CD-ROM

8.7 Illustration zur Fehlererholung auf der CD-ROM

Externe Datenträger:

Lochstreifen (historisch)

Lochkarten (historisch)

Papier:

Ausgabe über Drucker

Eingabe über Scanner

Magnetische Medien:

Disketten ("Floppies")

(Beispiel: 3,5" mit 1,44 MB formatiert)

Festplatten

(übliche Ausführung: Winchester)

Bewegliche Magnetplatten

(Beispiele: ZIP, 100MB; JAZ, 1GB)

Magnetische Bänder

(Beispiel: Kapazität eines DDS-3 Bandes:

12 GB unkomprimiert,

24 GB komprimiert)

Optische Medien:

CD-ROM (Compact Disc - Read Only Memory)

CD-R (CD - Recordable)

CD-RW (CD - Rewritable)

DVD (= Digital Video Disc oder

= Digital Versatile Disc,

Kapazität: 4,7 – 17 GB)

HD-DVD (= High Density DVD;

Kapazität etwa 15 GB pro Lage)

Blu-ray Disc (Kapazität: 27 GB – 200 GB)

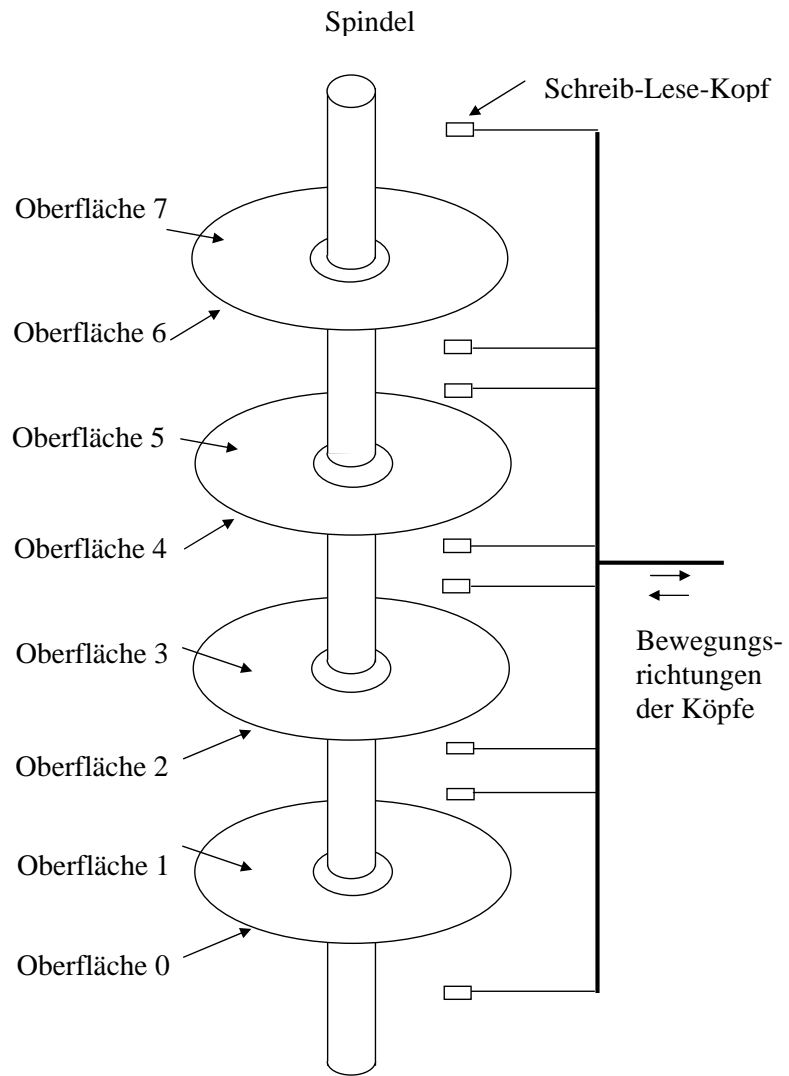
**Ausgewählte Kenndaten der Festplatte
Barracuda ES 320 GB (ST3320620NS):**

Abmessungen:	Höhe: 26,11 mm
	Breite: 101,6 mm
	Tiefe: 146,99 mm
Gewicht:	0,72 kg
Kapazität:	320 GB
Byte/Sektor:	512
Rotationsgeschwindigkeit:	7.200 U/min
Suchzeiten:	
Spur zu Spur:	Lesen: 0,8 ms
	Schreiben: 1,0 ms
durchschnittlich:	Lesen: 8,5 ms
	Schreiben: 9,5 ms
Puffergröße:	16 MiB
Externe Transferrate:	3,0 Gb/s
Maximale	
Interne Transferrate:	1030 Mb/s
Maximale dauerhafte	
Interne Transferrate:	72 MB/s
Schutzcode:	10 Bit
Nichtkorrigierbare	
Lesefehler pro gelesene Bit:	1 Sektor pro 10¹⁴

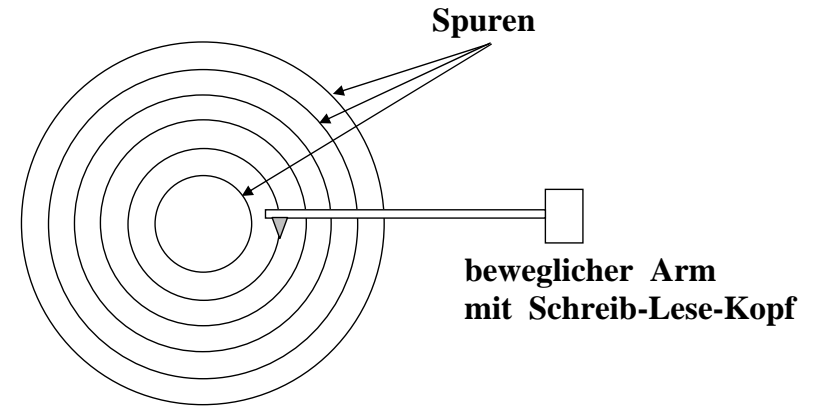
**Ausgewählte Kenndaten der Festplatte
Barracuda ES.2 1TB (ST31000340NS):**

Abmessungen:	Höhe: 26,11 mm
	Breite: 101,6 mm
	Tiefe: 146,99 mm
Gewicht:	0,677 kg
Kapazität:	1 TB
Byte/Sektor:	512
Rotationsgeschwindigkeit:	7.200 U/min
Suchzeiten:	
Spur zu Spur:	Lesen: 0,8 ms
	Schreiben: 1,0 ms
durchschnittlich:	Lesen: 8,5 ms
	Schreiben: 9,5 ms
Puffergröße:	32 MiB
Externe Transferrate:	3,0 Gb/s
Maximale	
Interne Transferrate:	1287 Mb/s
Maximale dauerhafte	
Interne Transferrate:	105 MB/s
Schutzcode:	10 Bit
Nichtkorrigierbare	
Lesefehler pro gelesene Bit:	1 Sektor pro 10¹⁵

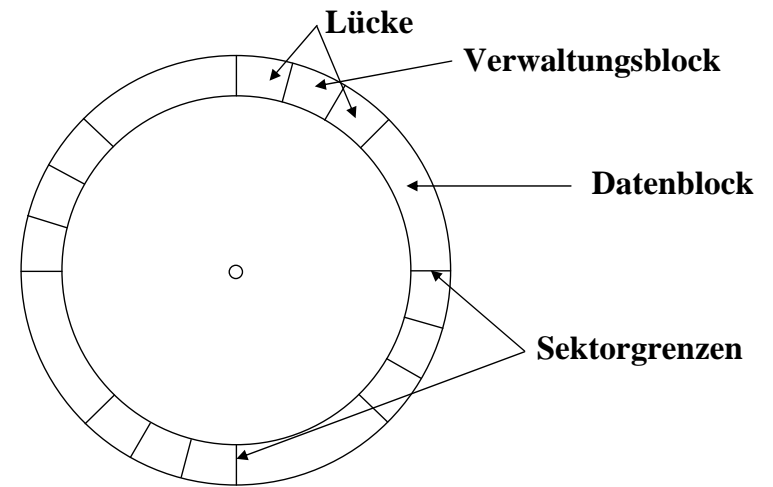
Seitensicht einer Platte:



Aufsicht auf Plattenoberfläche:



Aufbau einer Spur:



**Aufbau eines physikalischen Sektors
der Platte ST506 (Fall 1):**

Gesamtgröße: 315 Byte

Lücke 1: 14 Byte

Adreßfeld:

ID AM: 2 Byte

Zylindernummer: 1 Byte

Kopfnummer: 1 Byte

Sektornummer: 1 Byte

CRC: 2 Byte

Lücke 2: 16 Byte

Datenfeld:

Data AM: 2 Byte

Daten: 256 Byte

CRC: 2 Byte

Lücke 3: 18 Byte

$$\text{CRC-Polynom} = x^{16} + x^{12} + x^5 + 1$$

**Aufbau eines physikalischen Sektors
der Platte ST506 (Fall 2):**

Gesamtgröße: 600 Byte

Lücke 1: 17 Byte

Adreßfeld:

Synchronisation: 1 Byte

Zylindernummer: 2 Byte

Kopfnummer: 1 Byte

Sektornummer: 1 Byte

CRC: 2 Byte

Lücke 2: 41 Byte

Datenfeld:

Synchronisation: 1 Byte

Daten: 512 Byte

CRC: 2 Byte

Lücke 3: 20 Byte

$$\text{CRC-Polynom} = x^{16} + x^{12} + x^5 + 1$$

Aufbau einer PC-Diskettenspur bei MFM-Codierung:

Spurbeginn:

Lücke	80 Byte	$80 * 0x4e$
Synchronisation	12 Byte	$12 * 0x00$
Indexadreibmarke	4 Byte	$0xc2c2c2fc$
Lücke	50 Byte	$50 * 0x4e$

Sektor:

Synchronisation	12 Byte	$12 * 0x00$
Sektoradreibmarke	4 Byte	$0xa1a1a1fe$
Sektoridentifikator	4 Byte	Spur Kopf Sektor Sektorgröße

CRC	2 Byte	
Lücke	22 Byte	$22 * 0x4e$
Synchronisation	12 Byte	$12 * 0x00$
Datenadreibmarke	4 Byte	$0xa1a1a1fb$ oder $0xa1a1a1f8$
Daten	512 Byte	
CRC	2 Byte	
Lücke	80 Byte	$80 * 0x4e$

Spurende:

Lücke	Restliche Byte	$xx * 0x4e$
-------	----------------	-------------

Bemerkungen:

- (i) CRC-Polynom = $x^{16} + x^{12} + x^5 + 1$
- (ii) Die Größe der Lücke nach einem Sektor ist die in einem Formatbefehl verwendete.

Schutz durch Polynomdivision:

Bitketten lassen sich als Polynome über dem $GF(2)$ interpretieren.

Seien:

$$g(x) = x^k + g_{k-1}x^{k-1} + \dots + g_1x + 1$$

ein Polynom vom Grade $k > 0$,

$d(x)$ die zu schützende Bitkette der Länge n .

Verfahren:

1. Bilde $a(x) = d(x) * x^k$.
2. Berechne den Rest $r(x)$ mit
 $a(x) = q(x) * g(x) + r(x)$
und $\text{Grad}(r) < \text{Grad}(g)$.
3. Bilde $s(x) = a(x) - r(x)$.

Dann gilt: $s(x)$ wird durch $g(x)$ ohne Rest geteilt.

Bemerkungen:

1. Man nennt dies Schutzverfahren ein CRC-Verfahren. (CRC = Cyclic Redundancy Check.)
2. Häufig benutzte Schutzpolynome sind:
CRC-12: $x^{12} + x^4 + x^3 + x^2 + x + 1$
CRC-16: $x^{16} + x^{15} + x^2 + 1$
CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$
CD-ROM: $(x^{16} + x^{15} + x^2 + 1) * (x^{16} + x^2 + x + 1)$

Beispiel einer Polynomdivision:

Datenbitfolge: 1101011011
Divisionspolynom: 10011

Durchführung der Division:

$$\begin{array}{r}
 x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4 : x^4 + x^1 + x^0 = x^9 + x^8 + x^3 + x^1 \\
 \underline{x^{13} + x^{10} + x^9} \\
 \phantom{x^{13} +} x^{12} + x^9 + x^8 \\
 \underline{\phantom{x^{13} +} x^{12} + x^9 + x^8} \\
 \phantom{x^{13} + x^{12} +} x^7 + x^5 + x^4 \\
 \underline{\phantom{x^{13} + x^{12} +} x^7 + x^4 + x^3} \\
 \phantom{x^{13} + x^{12} + x^7 +} x^5 + x^3 \\
 \phantom{x^{13} + x^{12} + x^7 +} \underline{x^5 + x^2 + x^1} \\
 \phantom{x^{13} + x^{12} + x^7 + x^5 +} x^3 + x^2 + x^1
 \end{array}$$

Rest = $x^3 + x^2 + x^1$

damit CRC-Schutzcode = 1110

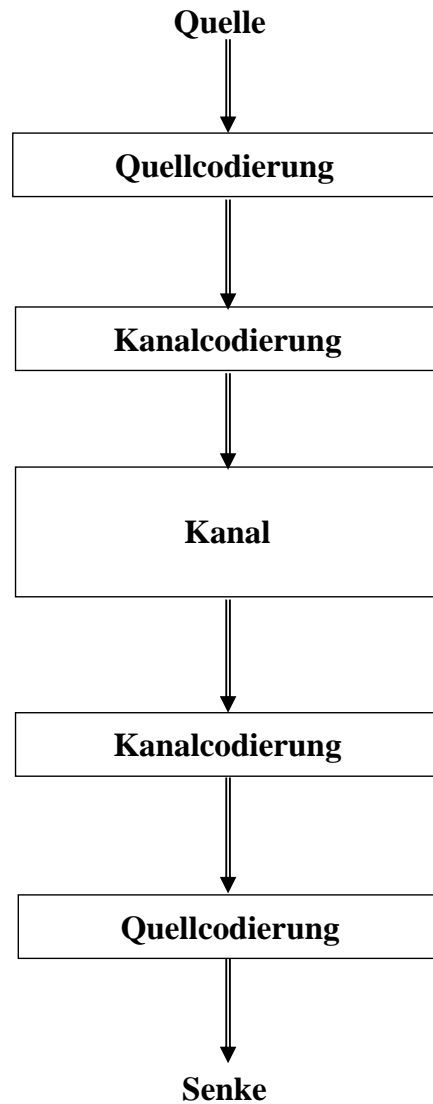
damit geschützte Daten = 11010110111110

Zoneneinteilung der 3,8 GB Quantum Fireball (1996):

6.810 Spuren pro Oberfläche

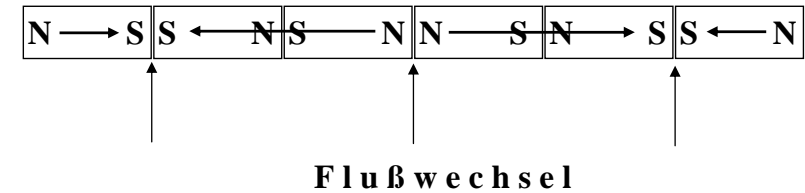
Zone	Spuren in Zone	Sektoren pro Spur	Datentransfer-rate (Mbit/s)
0	454	232	92,9
1	454	229	91,7
2	454	225	90,4
3	454	221	89,2
4	454	214	85,8
5	454	205	82,1
6	454	195	77,9
7	454	185	74,4
8	454	180	71,4
9	454	170	68,2
10	454	162	65,2
11	454	153	61,7
12	454	142	57,4
13	454	135	53,7
14	454	122	49,5

Codierungsmodell:



Ausgewählte Aufzeichnungsverfahren für magnetische Träger:

Die Aufzeichnung der digitalen Information erfolgt in Form von Flußwechseln.



Man kennt drei einfache Verfahren:

FM = "Frequency Modulation"

MFМ = "Modified Frequency Modulation"

RLL = "Run Length Limited"

Codierungsvorschriften:

FM-Verfahren:

Bit 0: Flußwechsel in Bitzellenmitte,

Bit 1: Kein Flußwechsel in Bitzelle,

zusätzlich zur Takterhaltung:

ein Flußwechsel zu Beginn jeder Bitzelle.

MFМ-Verfahren:

Bit 0: Kein Flußwechsel in Bitzelle,

Bit 1: Flußwechsel in Bitzellenmitte,

zusätzlich zur Takterhaltung:

ein Flußwechsel zu Beginn einer 0-Bitzelle, die auf eine 0-Bitzelle folgt.

Beispiel:

Bitkette: 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1
FM: F FFFFF F F FFFFF FFF FFFFFFFF
MFm: F F F F F F F F F F F

Bemerkung: Die Zahl der Flußwechsel hat sich im MFm-Verfahren gegenüber dem FM-Verfahren halbiert.

RLL (x, y):

Die Parameter x und y beschränken die Lauflänge eines Binärsymbols, im allgemeinen des 0-Symbols; zwischen dem Auftreten zweier 1-Symbole müssen mindestens x 0-Symbole und dürfen höchstens y 0-Symbole liegen.

Betrachtet man die Flußwechsel als 1-Symbole, dann gilt:

FM = RLL (0, 1) und MFm = RLL (1, 3).

Ein RLL-Verfahren wird definiert durch eine Codierungstabelle, z. B. für einen RLL (2, 7) – Code durch

Bitfolge	Flußwechsel
10	kFkk
11	Fkkk
000	kkkFkk
010	FkkFkk
011	kkFkkk
0010	kkFkkFkk
0011	kkkkFkkk

(Legende: F = Flußwechsel, k = kein Flußwechsel)

Fortführung Beispiel:

Bitkette: 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1
F F F F F F F

Beispiel für einen RLL (1, 7) – Code:

Bitfolge	Flußwechsel
0000	FkFkkk
0001	Fkkkkk
1000	kkFkkk
1001	kFkkkk
00	FkF
01	Fkk
10	kkF
11	kFk

Die Codierungstabelle ist von oben nach unten zu lesen.

Beispiel:

Eingangsbittfolge: 11 00 11 11 00 01 01 11
10 01 01 10 01 00 00 10

Folge der Flußwechsel: kFk FkF kFk kFk Fkk
kkk kFk kkk kkF kFk
Fkk kFk kkk FkF kkk
kkF

Zeiten beim Datentransfer von oder zu einer Platte:

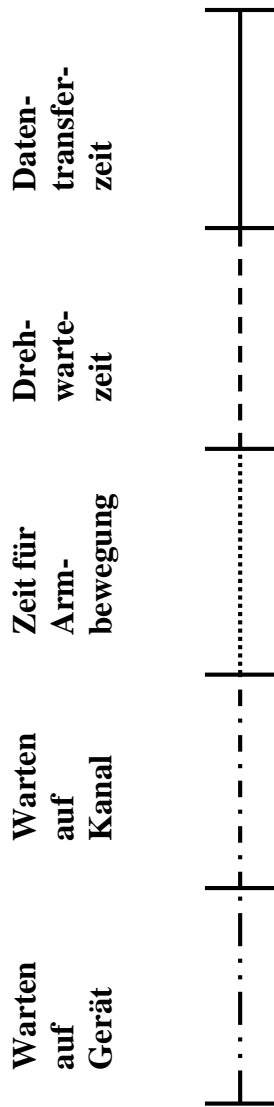


Tabelle:

Umdrehungszahl	Drehwartezeit
3.600 U/min	8,33 ms
5.400 U/min	5,56 ms
7.200 U/min	4,17 ms
10.000 U/min	3 ms
15.000 U/min	2 ms

Zeit für einen Plattentransfer

$$= \text{Suchzeit} + \text{Drehwartezeit} + \text{Datentransferzeit}$$

Näherungsformeln:

$$\text{Geschätzte Suchzeit} = m * n + s$$

mit m = Gerätekonstante

n = Zahl der überquerten Spuren

s = Anlaufzeit

$$\text{Datentransferzeit} = b / (r * N)$$

mit b = Zahl der zu übertragenden Byte

r = Rotationsgeschwindigkeit

N = Zahl der Datenbyte auf Spur

Bemerkung: Die berechnete Zeit für einen Plattentransfer liefert nur einen groben Anhalt. Außer für r und b kennt man für die anderen Größen nur Schätzwerte. Auch erschweren die Datenpuffer in den Plattengeräten jede genaue Messung.

Scheduling der Abarbeitung von Platten-Aufträgen gemäß der Zylindernummer:

Einfache Verfahren:

1. FCFS ("First Come, First Served"):
Einfach und fair.
2. SSTF ("Shortest Seek Time First"):
Im Mittel kürzeste Wartezeit, wobei einzelne Aufträge sehr lange Wartezeiten erfahren können.
3. Fahrstuhlalgorithmus:
Vereinigt die Vorteile von 1 und 2.
4. Einseitiger Fahrstuhlalgorithmus:
Die Wartezeiten sind gleichmäßiger verteilt als bei 3.

Beispiel:

Position des Kopfes: 53
Spurnummern: 98, 183, 37, 22, 14, 65, 67

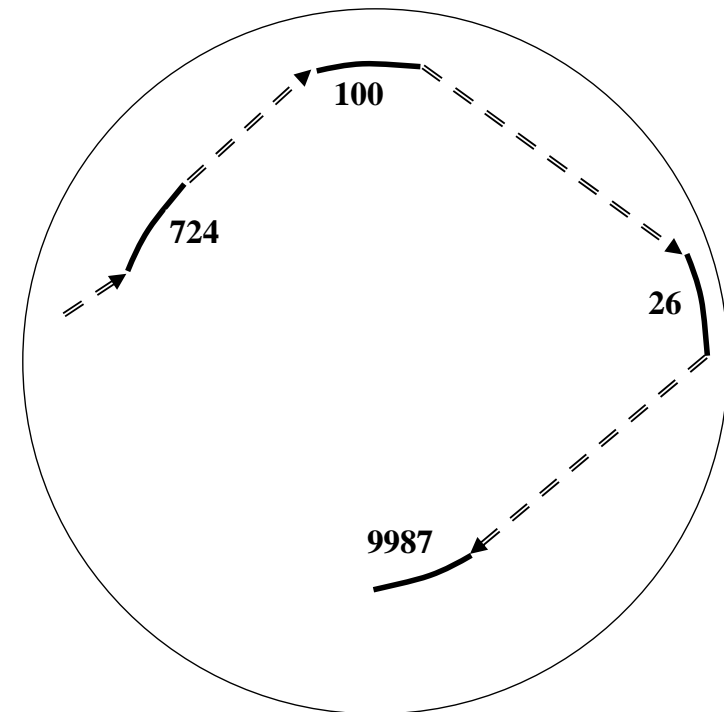
ad 1: 98, 183, 37, 22, 14, 65, 67
ad 2: 65, 67, 37, 22, 14, 98, 183
ad 3: 65, 67, 98, 183, 37, 22, 14
ad 4: 65, 67, 98, 183, 14, 22, 37

Festplatten sind gekapselte Einheiten. Die Scheduling-Entscheidungen bezüglich der Abarbeitung einer Auftragsfolge sollte man der Festplattensoftware überlassen.

Beispiel von Anderson:

Die angegebenen Nummern sind logische Sektoradressen.

Auftragsreihenfolge: 724, 100, 9987, 26
Sortierte Folge: 26, 100, 724, 9987
Beste Abarbeitungsfolge gemäß Plattengeometrie:
724, 100, 26, 9987

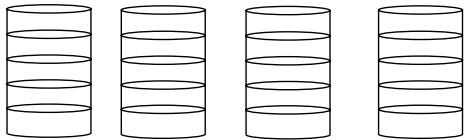


RAID:

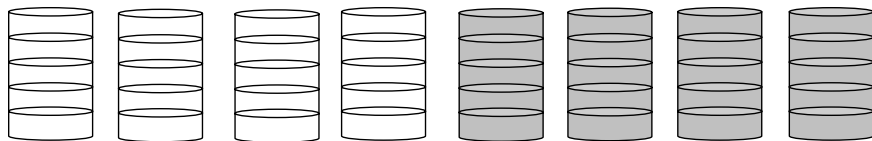
RAID = Redundant Array of Inexpensive Disks
oder = Redundant Array of Independent Disks

Man unterscheidet mehrere Ebenen, die sieben bekanntesten sind:

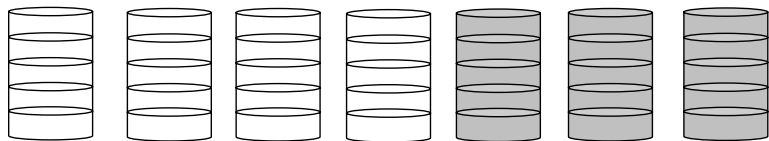
RAID 0: Keine Redundanz



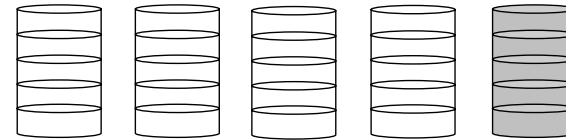
RAID 1: Spiegelung von Einzelplatten



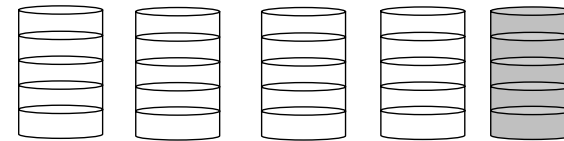
RAID 2: Einsatz fehlerkorrigierender Bitcodes



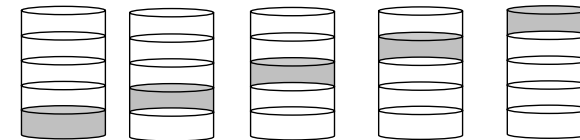
RAID 3: Bit-Parität



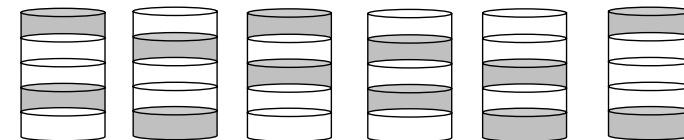
RAID 4: Block-Parität



RAID 5: Rotierende Block-Parität



RAID 6: Doppelte Redundanz

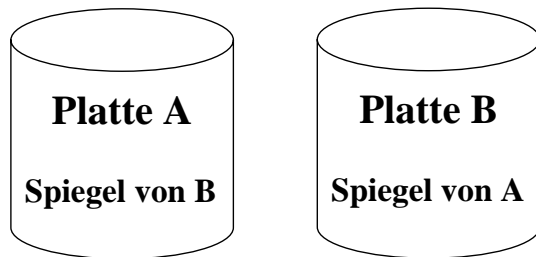


Bemerkungen:

RAID 0: Die Datenblöcke werden gleichmäßig auf mehrere unabhängige Platten verteilt. Hierdurch erreicht man Lastglättung.

Block 0	Block 1	Block 2	Block 3
Block 4	Block 5	Block 6	Block 7
Block 8	Block 9	Block 10	Block 11
Block 12	Block 13	Block 14	Block 15

RAID 1: Jeder Platteninhalt ist zweimal vorhanden. Bei Ausfall einer Platte nutzt man das Duplikat und erstellt eine neue Kopie auf einer Reserveplatte. Hierbei muß jeder Schreibzugriff doppelt ausgeführt werden. Lesezugriffe werden nur einmal ausgeführt.



RAID 2: Man verwendet einen fehlerkorrigierenden Bitcode. Ein einfacher ist der (7, 4) Hamming-Code. Die Originalbit und die Redundanzbit werden auf verschiedene Platten verteilt. Ein Datenzugriff erfordert viele synchronisierte Plattenoperationen. Im Fall des (7, 4)-Codes gehören zu einem Original-Quadrupel drei Korrekturbit, man benötigt also sieben Platten. Insgesamt erscheint der Aufwand nicht gerechtfertigt.

Gleichungssystem: $k_2 = x_2 + x_1 + x_0 \pmod{2}$
 $k_1 = x_3 + x_1 + x_0 \pmod{2}$
 $k_0 = x_3 + x_2 + x_0 \pmod{2}$

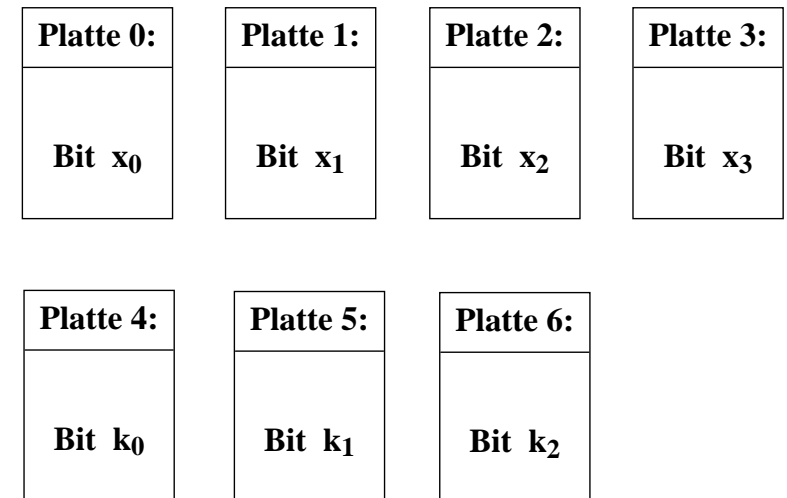
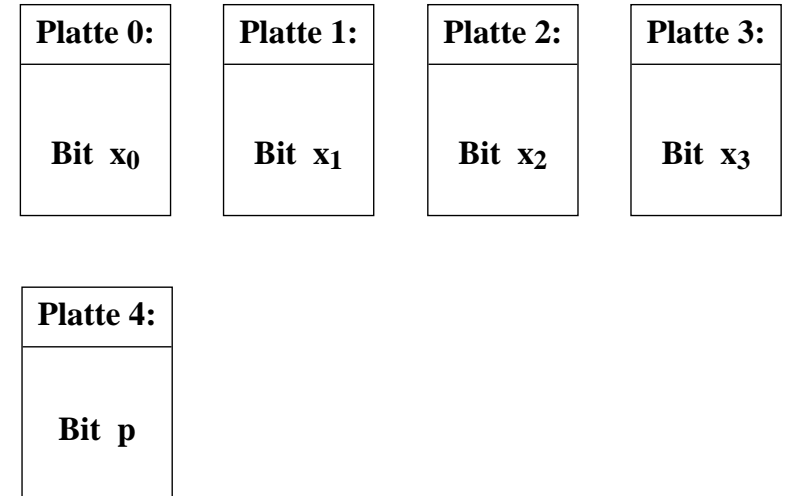


Tabelle eines Hamming-Codes:

Datum				Schutzbit		
x_3	x_2	x_1	x_0	k_2	k_1	k_0
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	1	0	0	1	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	1	0	0
1	0	1	0	1	0	1
1	0	1	1	0	1	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	1	0	0	0	0
1	1	1	1	1	1	1

RAID 3: Um gegen den Totalausfall einer Platte gewappnet zu sein, genügt der Einsatz einer Paritätsplatte. Hat man die Originaldaten auf vier Platten verteilt, dann bildet man die Modulo-2-Summe über entsprechende Bits der Originaldaten.



Formel: $p = x_0 \text{ xor } x_1 \text{ xor } x_2 \text{ xor } x_3$

Bei Ausfall einer der Platten 0 bis 4 garantiert die Eigenschaft der Modulo-2-Addition die Restaurierbarkeit des Inhalts der ausgefallenen Platte.

RAID 4: Dies ist ähnlich zu RAID 3. Die Paritätsinformation nutzt man nur bei Totalausfall einer Platte. Daher führen nur Schreibzugriffe zu einer erhöhten Last. Jeder Schreibzugriff erfordert zwei Lese- und zwei Schreibzugriffe.

RAID 5: Um den Flaschenhals der Paritätsplatte zu mildern, verteilt man die Redundanzinformation zyklisch über alle Platten.

RAID 6: Hier erhöht man die Datensicherheit gegenüber RAID 5 durch Bildung mehrerer unabhängiger Schutzinformationen. Ein einfaches Verfahren zur Bildung unabhängiger Schutzinformationen besteht in der Nutzung Vandermondescher Gleichungssysteme. Um das Problem des Zahlen-Überlaufs zu meiden, rechnet man in endlichen Körpern, vorzugsweise im Galois-Körper $GF(2^8)$.

Optisches Speichermedium CD-ROM:
(CD-ROM = Compact Disc Read Only Memory)

Die CD-ROM ist eine Weiterentwicklung der Audio-CD, die 1982 auf den Markt kam. Die Daten werden in einer Spirale von über 5 km Länge aufgezeichnet. Die Speicherkapazität einer 74-Minuten-CD-DA beträgt etwa 783 Millionen Nutzbyte. Für die CD-ROM im engeren Sinne sinkt dieser Wert auf 650 MB. Die Standards für Audio-CD und CD-ROM sind IEC 908 ("red book") und ISO/IEC 10149 ("yellow book").

Bemerkenswert ist das Aufzeichnungsverfahren für die Rohdaten. Jeweils 24 8-Bit-Datenbyte werden 4 Q-Paritäts-Byte und 4 P-Paritäts-Byte und ein Subcode-Byte hinzugefügt. Jedes Byte wird durch 17 Kanalbit codiert. Die teils technisch bedingte aufwendige Codierung, wobei noch eine geschickte Verschachtelung der Datenbyte hinzukommt, führt zu einer Wahrscheinlichkeit von 10^{-8} für nicht korrigierbare Lesefehler auf der Audio-Ebene. Für Daten ist dies zu gering. Zu jeweils 2.048 Datenbyte berechnet man zusätzlich 276 Byte Fehlerkorrekturcode. Damit sinkt die Wahrscheinlichkeit eines nicht korrigierbaren Lesefehlers von 10^{-8} auf der Audio-Ebene auf 10^{-12} auf der Datenebene. Der Nutzen der 276 zusätzlichen Korrekturbyte wird mittels der letzten Folien im Anhang dieses Kapitels demonstriert.

Einige Merkmale der Audio-CD:

maximale Spieldauer:	74 Minuten, 33 Sekunden
konstante Lesegeschwindigkeit:	1,2 m/s - 1,4 m/s
Spurabstand:	1,6 μm
Durchmesser:	120 mm
Dicke:	1,2 mm
Zentrumsdurchmesser:	15 mm
Aufzeichnungsbereich:	46 mm - 117 mm
Datenbereich:	50 mm - 116 mm
Minimale "Pit"-Länge:	0,833 μm (1,2 m/s) bis 0,972 μm (1,4 m/s)
Maximale "Pit"-Länge:	3,05 μm (1,2 m/s) bis 3,56 μm (1,4 m/s)
"Pit"-Tiefe:	\sim 0,11 μm
"Pit"-Breite:	\sim 0,5 μm
Standard Wellenlänge:	780 nm
Brechungsindex des Materials:	1,55
Zahl der Kanäle:	2 (4 zulässig)
Quantelung:	16 Bit linear
Abtastfrequenz:	44.100 Hz
Kanal-Bit-Rate:	4,3218 Mbit/s
Daten-Bit-Rate:	2,0338 Mbit/s
Verhältnis Datenbit zu Kanalbit:	8 : 17
Fehlerkorrekturcode:	Verschachtelter Reed-Solomon Code (25% Redundanz)
Modulationssystem:	Eight to fourteen Modulation (EFM)

Ein Audio-Rahmen:

Synchronisation	24 Bit
Subcode	14 Bit
6 * 2 * 2 * 14 Datenbit	336 Bit
8 * 14 Paritätsbit	112 Bit
34 * 3 Pufferbit	102 Bit
<hr/>	
Summe:	588 Bit

Bemerkungen:

1. 192 Datenbit werden in 588 Kanalbit codiert.
2. Das Subcode-Byte enthält 8 Bit, die den Subkanälen P, Q, R, S, T, U, V, W zugeordnet sind. Nur die Subkanäle P und Q werden von der Audio-disc genutzt.
3. Die Fehlerbehandlung erfolgt in drei Schritten:
 - a) Zunächst wird versucht, die fehlerhaften Byte zu korrigieren.
 - b) Unkorrigierbare Bytewerte versucht man durch Interpolation zu gewinnen.
 - c) Läßt sich auch die Interpolation nicht durchführen, dann wird der entsprechende Musikteil durch Stille ersetzt.

Beispiel zur Codeverschränkung:

Länge der Nachricht = 103

Auslesen der teilweise zerstörten Nachricht:

Ein Buch mit dem Titel "Das Geheimnis meiner XXXXXXXXX" kann nichts anderes als leere Seiten enthalten.

Speicherung der gleichen Nachricht mit Verschränkungszahl 11:

E ltnsmi aeialet etSnhnlilhai ene so ann
di iBXXXXXXXXXmuleDtermeincenaer ncihe"
sneMT hs r .siiet mekG

Rücktransformation:

Ein Buch mit dXm Titel XDas Geheimnis me
inXr MillioXen" kann Xichts anXeres alsX
leere SeiXen enthaXten.

Nun ist die ursprüngliche Nachricht ohne großen Aufwand rekonstruierbar.

Bemerkung: Mittels Codeverschränkung werden Bündelfehlern in Einzelfehler transformiert.

Konversion von 8-Bit-Byte zu "pits and lands"

Daten: 11101000 11100010 10111010 11101011

Umsetzung in 14-Bit-Darstellung:

10000100000010 10000100010010 10010000001001 000010010000010

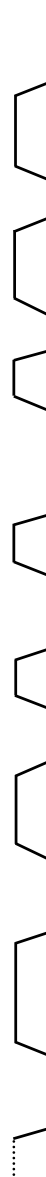
Einfügen von Pufferbits:

10000100000010 000 10000100010010 000 10010000001001 001 000010010000010

Bitfolge:

10000100000010000100001000100100000100100000010010010000010010000010

"Pits and Lands":



Bemerkung: Der Wechsel zwischen "pit" und "land" wird durch eine 1 codiert.

**Ausschnitt aus der EFM-Tabelle:
(EFM = Eight to Fourteen Modulation)**

0:	00000000	01 001000 100000
1:	00000001	10 000100 000000
2:	00000010	10 010000 100000
3:	00000011	10 001000 100000
4:	00000100	01 000100 000000
5:	00000101	00 000100 010000
6:	00000110	00 010000 100000
7:	00000111	00 100100 000000
8:	00001000	01 001001 000000
9:	00001001	10 000001 000000
10:	00001010	10 010001 000000
11:	00001011	10 001001 000000
12:	00001100	01 000001 000000
13:	00001101	00 000001 000000
14:	00001110	00 010001 000000
15:	00001111	00 100001 000000
16:	00010000	10 000000 100000
17:	00010001	10 000010 000000
18:	00010010	10 010010 000000
19:	00010011	00 100000 100000
20:	00010100	01 000010 000000
21:	00010101	00 000010 000000
22:	00010110	00 010010 000000
23:	00010111	00 100010 000000
24:	00011000	01 001000 010000
25:	00011001	10 000000 010000
26:	00011010	10 010000 010000
27:	00011011	10 001000 010000
28:	00011100	01 000000 010000
29:	00011101	00 001000 010000
30:	00011110	00 010000 010000
31:	00011111	00 100000 010000

Aufbau eines MODE-1-Blocks einer Daten-CD:

Byte-Nummer	Inhalt
0	0000 0000
1 - 10	1111 1111
11	0000 0000
12	Minuten (74 max.)
13	Sekunden (59 max.)
14	Blocknummer innerhalb Sekunde (75 Blöcke pro Sekunde)
15	01 ("Modus")
16 – 2063	Nutzdaten
2064 – 2067	4 Byte Fehlererkennungscode
2068 – 2075	8 Byte 0000 0000
2076 - 2247	172 Byte P-Parität
2248 - 2351	104 Byte Q-Parität

Illustration zur CD-ROM-Fehlerkorrektur (1): 13 Einzelfehler in einem Halbrahmen.

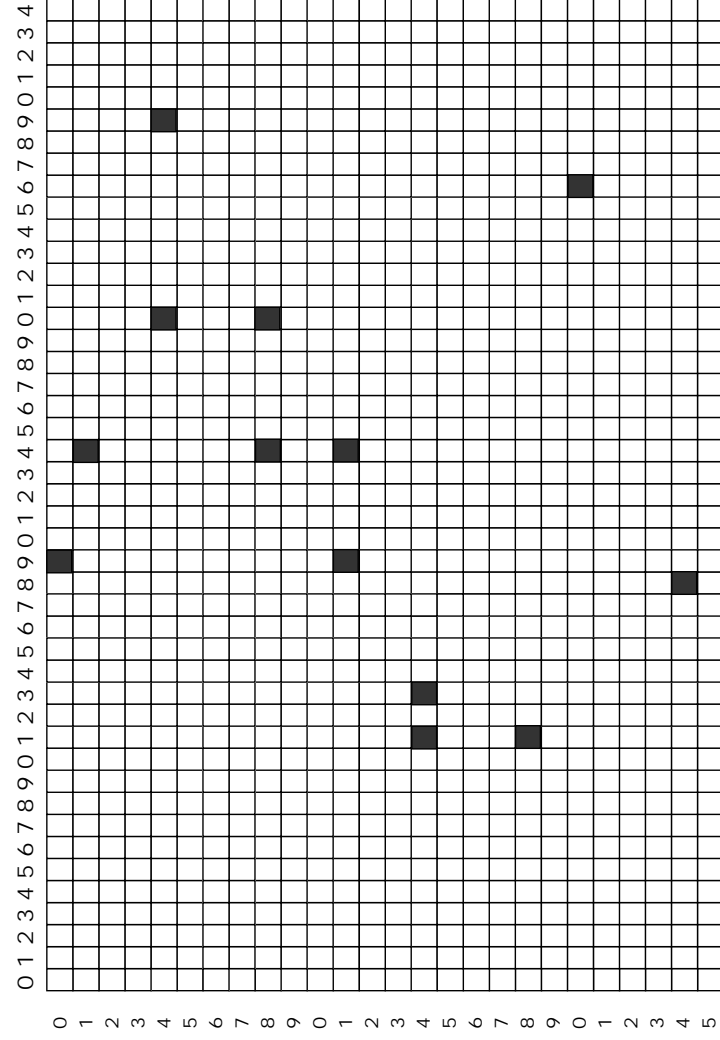


Illustration zur CD-ROM-Fehlerkorrektur (2): 5 Einzelfehler in Zeilen korrigiert.

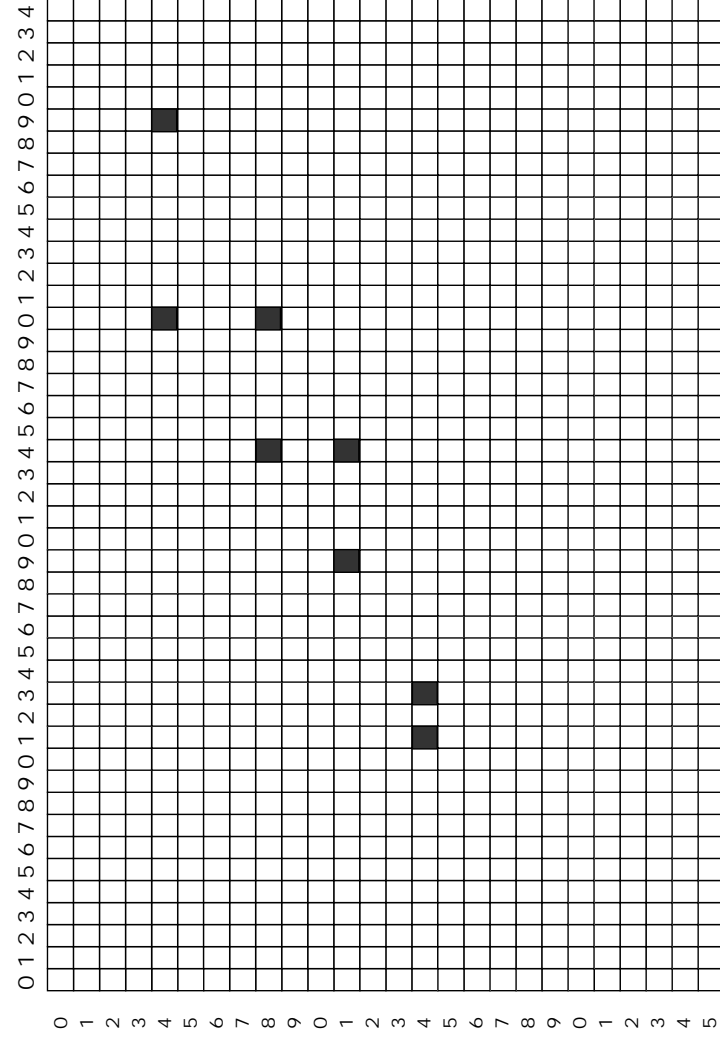


Illustration zur CD-ROM-Fehlerkorrektur (3): 4 Einzelfehler in Spalten korrigiert.

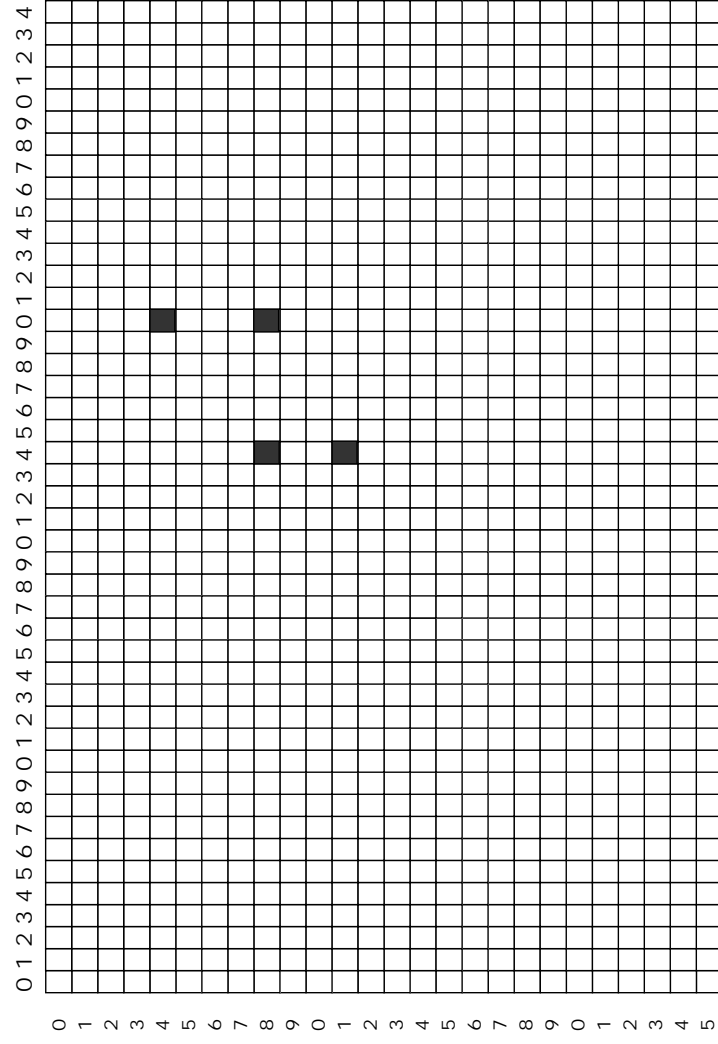


Illustration zur CD-ROM-Fehlerkorrektur (4): 2 Einzelfehler in Zeilen korrigiert.

