



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 38 416 T2** 2008.10.16

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 040 583 B1**

(21) Deutsches Aktenzeichen: **699 38 416.8**

(86) PCT-Aktenzeichen: **PCT/EP99/05338**

(96) Europäisches Aktenzeichen: **99 939 406.7**

(87) PCT-Veröffentlichungs-Nr.: **WO 2000/007300**

(86) PCT-Anmeldetag: **19.07.1999**

(87) Veröffentlichungstag

der PCT-Anmeldung: **10.02.2000**

(97) Erstveröffentlichung durch das EPA: **04.10.2000**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **26.03.2008**

(47) Veröffentlichungstag im Patentblatt: **16.10.2008**

(51) Int Cl.⁸: **H03M 13/00** (2006.01)
H03M 13/27 (2006.01)

(30) Unionspriorität:

98202511 **27.07.1998** **EP**

(73) Patentinhaber:

**Koninklijke Philips Electronics N.V., Eindhoven,
NL; Sony Corp., Tokio/Tokyo, JP**

(74) Vertreter:

Volmer, G., Dipl.-Ing., Pat.-Anw., 52066 Aachen

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE**

(72) Erfinder:

**VAN DIJK, Marten E., NL-5656 AA Eindhoven, NL;
TOLHUIZEN, Ludovicus M., NL-5656 AA
Eindhoven, NL; KAHLMAN, Josephus A., NL-5656
AA Eindhoven, NL; BAGGEN, Constant P.,
NL-5656 AA Eindhoven, NL; HATTORI, Masayuki,
NL-5656 AA Eindhoven, NL; YAMAMOTO, Kouhei,
NL-5656 AA Eindhoven, NL; NARAHARA, Tatsuya,
NL-5656 AA Eindhoven, NL; SENSHU, Susumu,
NL-5656 AA Eindhoven, NL**

(54) Bezeichnung: **KODIERUNG VON MEHRWORTINFORMATION MIT WORTWEISER VERSCHACHTELUNG**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

HINTERGRUND DER ERFINDUNG

[0001] Die Erfindung betrifft ein Verfahren gemäß dem Oberbegriff von Anspruch 1. Die US-Patentschriften 4,559,625 von Berlekamp et al. und US 5,299,208 von Blaum et al. offenbaren eine Decodierung von verschachtelten und fehlergeschützten Informationswörtern, wobei ein Fehlermuster, das in einem ersten Wort gefunden wurde, einen Hinweis (Clue) liefern kann, um Fehler in einem anderen Wort derselben Gruppe von Wörtern zu lokalisieren. In den zitierten Quellen wird ein Defektmodell mit mehrere Symbole enthaltenden Fehlerbündeln über mehrere Wörter hinweg verwendet. Ein Fehler in einem bestimmten Wort führt zu einer hohen Wahrscheinlichkeit dafür, dass ein Fehler an einer entsprechenden Symbolposition, auf die gezeigt wird, in einem nächsten Wort oder in nächsten Wörtern auftritt. Die Prozedur kann die Anzahl korrigierter Fehler erhöhen. Ein Hinweis kommt jedoch nur dann zur Realisierung, wenn das Hinweiswort vollständig korrigiert worden ist. Ferner speichert das Medium Informationen sowie Synchronisationsbitgruppen, welche eine beträchtliche Menge an Redundanz repräsentieren, die ebenfalls verwendet werden kann, um Fehler in den Zielwörtern zu signalisieren, so dass in gewissem Maße Hinweiswörter gegen Synchronisationsbitgruppen ausgetauscht werden können. Ein Teil der Fehler ist auf so genannte Bitslips (Bitschlupf) zurückzuführen, welche weiter entfernt von den Synchronisationsbitgruppen häufiger sind. Daher kann ein Verschachteln der Hinweiswörter zwischen Hinweisspalten und ebenso ein Verschachteln der Zielwörter zwischen Zielspalten den Fehlerschutz weiter verbessern.

[0002] Die Veröffentlichung von E. Paaske "Improved Decoding for a Concatenated Coding System Recommended By CCSDS", IEEE Transactions on Communications, Bd. 38, Nr. 8, August 1990, Seiten 1138–1144, XP000162505, offenbart ein Fehlerkorrekturverfahren, bei welchem Reed-Solomon-Codewörter in einer Spaltenrichtung angeordnet sind und welches den Schritt des Ableitens von Hinweisen auf Fehlerbündel in der Zeilenrichtung aus erfolgreich decodierten Codewörtern umfasst. Alle RS-Codewörter sind RS(255, 223) mit derselben Schutzwirkung.

[0003] Die Veröffentlichung von E. Paaske "Alternative to NASA's concatenated coding system for the Galileo mission", IEEE Proceedings on Communications, Bd. 141, August 1994, Seiten 229–232, XP006001668, offenbart ein Verfahren des Fehlerschutzes, bei welchem Reed-Solomon-Codewörter mit hoher und niedriger Schutzwirkung in einer Spaltenrichtung in einer alternierenden Art und Weise angeordnet sind.

KURZDARSTELLUNG DER ERFINDUNG

[0004] Demzufolge ist es unter anderem eine Aufgabe der Erfindung, ein Codierungsformat zu schaffen, das es ermöglicht, dass Hinweiswörter mit Synchronisationsbitgruppen in einem systematischen Format zusammenwirken, während außerdem mögliche negative Auswirkungen von Bitslips gleichmäßiger zwischen den verschiedenen Wörtern ausgeglichen werden. Daher ist die Erfindung gemäß einem ihrer Aspekte gekennzeichnet, wie in dem Kennzeichnungsteil von Anspruch 1 dargelegt. Ein Hinweis kann auf ein Lösch-Symbol zeigen. Zeigen kann eine Fehlerkorrektur leistungsfähiger machen. Tatsächlich werden viele Codes höchstens t Fehler korrigieren, wenn keine lokale (ortsbestimmende) Angabe für den Fehler bekannt ist. Sind die Lösstellen gegeben, kann im Allgemeinen eine größere Anzahl $e > t$ von Löschungen korrigiert werden. Auch der Schutz vor einer Kombination von Bündeln und zufälligen Fehlern kann sich verbessern. Die Erfindung kann sowohl für die Speicherung als auch für die Übertragung verwendet werden.

[0005] Vorteilhafterweise weist das Verfahren die Merkmale von Anspruch 2 auf. Dies ist eine relativ einfache Anordnung. Stattdessen kann die Anzahl von Synchronisationsspalten auch größer als eins sein, und die Anzahl von Hinweisspalten kann gerade sein, falls dies bevorzugt wird.

[0006] Vorteilhafterweise weist das Verfahren die Merkmale von Anspruch 3 auf. Bei einer Videoaufzeichnung beziehen sich Benutzerdaten auf das Bild und den begleitenden Ton, die einem Benutzer präsentiert werden sollen, während Systemdaten den Programmnamen, Zeit, Adressen, und verschiedene andere Parameter, welche von Nutzen sein können, angeben können, ohne Bezugnahme auf das eigentliche Video oder Audio. Dieses Merkmal ermöglicht einen schnellen Zugriff auf die Systemdaten ohne die Notwendigkeit, die Zielwörter zu decodieren.

[0007] Vorteilhafterweise weist das Verfahren die Merkmale von Anspruch 4 auf. Falls fehlerfrei, gibt dieses Merkmal unmittelbar einer Benutzereinrichtung an, ob der verarbeitete Informationsrahmen korrekt ist.

[0008] Vorteilhafterweise weist das Verfahren die Merkmale von Anspruch 5 auf. Dies ist eine geradlinige Organisation. Vorteilhafterweise weist das Verfahren die Merkmale von Anspruch 6 auf. Es hat sich erwiesen, dass dies den ungünstigsten Fall auf das mittlere Niveau anhebt, das für dieses Codeformat erreichbar ist. Vorteilhafterweise weist das Verfahren die Merkmale von Anspruch 7 auf. Optische Speicher haben sich als ein günstiges Medium erwiesen.

[0009] Die Erfindung betrifft außerdem ein Verfahren zum Decodieren von so codierten Informationen, eine Codier- und/oder Decodiereinrichtung zur Verwendung mit dem Verfahren und einen Träger, der so codierte Informationen enthält. Weitere vorteilhafte Aspekte der Erfindung sind in abhängigen Ansprüchen dargelegt.

KURZBESCHREIBUNG DER ZEICHNUNG

[0010] Diese und weitere Aspekte und Vorteile der Erfindung werden im Folgenden unter Bezugnahme auf die Beschreibung bevorzugter Ausführungsform ausführlicher erläutert, und insbesondere unter Bezugnahme auf die beigefügten Figuren. Es zeigen:

[0011] [Fig. 1a](#), [Fig. 1b](#) einen Aufzeichnungsträger;

[0012] [Fig. 2](#) eine Abspieleinrichtung;

[0013] [Fig. 3](#) eine Aufzeichnungseinrichtung;

[0014] [Fig. 4](#) ein System mit Codierer, Träger und Decodierer;

[0015] [Fig. 5](#) ein Codeformat-Prinzip;

[0016] [Fig. 6](#) eine schematische Darstellung eines physischen Clusters auf dem Träger;

[0017] [Fig. 7](#) einen Datenrahmen;

[0018] [Fig. 8](#) einen Aufbau eines Datensektors aus zwei Datenrahmen;

[0019] [Fig. 9](#) das Ummummern von Datenbytes und Bilden eines ECC-Sektors durch Hinzufügen von Paritäten;

[0020] [Fig. 10](#) einen Aufbau eines ECC-Clusters durch Multiplexing von 16 ECC- Sektoren;

[0021] [Fig. 11](#) einen unnummerierten ECC-Cluster vor dem Verschachteln;

[0022] [Fig. 12](#) einen verschachtelten ECC-Cluster;

[0023] [Fig. 13](#) ein Multiplexing des BIS-Clusters mit dem verschachtelten ECC- Cluster;

[0024] [Fig. 14](#) einen BIS-Block, der 24 BIS-Codewörter enthält;

[0025] [Fig. 14A](#) die Abbildung des BIS-Blockes in den BIS-Cluster;

[0026] [Fig. 15](#) die Rahmenstruktur zur Erläuterung der Verschachtelung der Hauptdaten;

[0027] [Fig. 16](#) die eigentliche Verschachtelung;

[0028] [Fig. 17](#) ein Beispiel der teilweisen Abbildung der BIS-Bytes auf die ersten acht Sektoren;

[0029] [Fig. 18](#) ein Beispiel der teilweisen Abbildung der BIS-Bytes auf die letzten acht Sektoren;

[0030] [Fig. 19](#) eine schematische Darstellung des Gesamtprozesses der Codierung.

AUSFÜHRLICHE BESCHREIBUNG BEVORZUGTER AUSFÜHRUNGSFORMEN

[0031] [Fig. 1a](#) zeigt einen plattenförmigen Aufzeichnungsträger **11** mit einer Spur **19** und einer zentralen Boh-

rung **10**. Die Spur **19** ist in einem spiralförmigen Muster von Windungen angeordnet, das im Wesentlichen parallele Spuren auf einer Informationsschicht bildet. Der Träger kann eine optische Platte mit einer beschreibbaren oder vorbeschriebenen Informationsschicht sein. CD-R, CD-RW und DVD-RAM sind beschreibbar, während eine Audio-CD ein Beispiel einer vorbeschriebenen Platte ist. Der vorbeschriebene Typ kann auf eine bekannte Weise hergestellt werden, indem zuerst eine Masterplatte beschrieben wird und danach Platten für den Verkauf gepresst werden. Auf einem beschreibbaren Träger ist die Spur durch eine vorgeprägte Spurstruktur angegeben, die während der Herstellung des Rohlings des Trägers geschaffen wird. Die Spur kann als eine Vorspur (Pregroove) **14** strukturiert sein, um einem Lese-Schreibkopf zu ermöglichen, der Spur **19** während des Abtastens zu folgen. Die Informationen sind auf der Informationsschicht mittels optisch erkennbarer Markierungen dargestellt, z. B. Pits (Täler) und Lands (Erhebungen).

[0032] **Fig. 1b** ist ein Querschnitt entlang der Linie b-b des Aufzeichnungsträgers **11** vom beschreibbaren Typ, wobei ein durchsichtiges Substrat **15** mit einer Aufzeichnungsschicht **16** und einer Schutzschicht **17** versehen ist. Die Vorspur **14** kann als Vertiefung oder eine Erhebung implementiert sein, oder als eine Materialeigenschaft, die von ihrer Umgebung abweicht.

[0033] Der Aufzeichnungsträger kann Benutzerinformationen tragen, welche im Interesse der Benutzerfreundlichkeit in kleinere Stücke unterteilt worden sind, welche jeweils eine Dauer von einigen Minuten haben können, z. B. Songs eines Albums oder Sätze einer Sinfonie. Zugriffsinformationen zum Identifizieren der Stücke können ebenfalls auf dem Träger vorgesehen sein, z. B. ein so genanntes Inhaltsverzeichnis (Table Of Contents, TOC) oder ein Dateisystem wie ISO 9660 für CD-ROM. Die Zugriffsinformationen können für jedes Stück Wiedergabezeit und Startadresse beinhalten, und weitere Informationen wie einen Songtitel. Solche Informationen können Systeminformationen darstellen. Die Informationen werden nach Analog-Digital-(A/D-)Wandlung auf eine digitale Weise aufgezeichnet.

[0034] **Fig. 2** zeigt ein Abspielgerät gemäß der Erfindung zum Lesen eines Aufzeichnungsträgers **11**, wie in **Fig. 1** dargestellt. Die Einrichtung weist einen Antrieb **21** auf, um den Aufzeichnungsträger **11** in Rotation zu versetzen, und einen Lesekopf **22** zum Abtasten der Spur auf dem Aufzeichnungsträger. Das Gerät weist Positionierungsmittel **25** für eine grobe radiale Positionierung des Lesekopfes **22** auf. Der Lesekopf umfasst ein optisches System eines bekannten Typs zum Erzeugen eines Strahls **24**, welcher durch optische Elemente hindurch geführt und auf einen Lichtpunkt **23** auf einer Spur der Informationsschicht fokussiert wird. Der Strahl **24** wird von einer bekannten Strahlungsquelle erzeugt. Der Lesekopf weist ferner einen Fokussierungs-Aktor (Focusing Actuator) zum Bewegen des Fokus des Strahls **24** entlang der optischen Achse des Strahls und einen Tracking-Aktor (Tracking Actuator) zur Feinpositionierung des Lichtpunktes **23** in einer radialen Richtung auf der Mitte der Spur auf. Der Tracking-Aktor kann Spulen umfassen, um ein optisches Element radial zu bewegen, oder kann dazu vorgesehen sein, den Winkel eines reflektierenden Elements zu verändern. Die von der Informationsschicht reflektierte Strahlung wird von einem Detektor eines üblichen Typs, z. B. einer Vier-Quadranten-Diode, im Lesekopf **22** detektiert, um ein Lesesignal und weitere Detektorsignale zu erzeugen, darunter ein Trackingfehler- und ein Fokusfehler-Signal, die mit dem besagten Tracking- bzw. Fokussierungs-Aktor gekoppelt sind. Das Lesesignal wird von Lesemitteln **27** verarbeitet, um die Daten wiederzugewinnen, wobei diese Lesemittel von einem üblichen Typ sind, der zum Beispiel einen Kanaldecoder und einen Fehlerkorrektor umfasst. Die wiedergewonnenen Daten werden zu einer Datenauswahlrichtung **28** gesendet, um gewisse Informationen aus den gelesenen Daten auszuwählen und zu dem Puffer **29** zu senden. Die Auswahl beruht auf Datentyp-Indikatoren, die auf dem Aufzeichnungsträger aufgezeichnet sind, z. B. Headern in einem Rahmenformat. Die komprimierten Informationen werden von dem Puffer **29** mittels eines Signals **30** zu einem Dekompressor **31** gesendet. Dieses Signal kann außerdem extern verfügbar sein. Der Dekompressor **31** decodiert die Daten, um die ursprünglichen Informationen am Ausgang **32** wiederzugeben. Der Dekompressor kann separat eingebaut sein, wie durch das Rechteck **33** in **Fig. 2** angegeben ist. Stattdessen kann der Puffer vor der Datenauswahl positioniert sein. Die Steuereinheit **20** empfängt ferner Steuerbefehle von einem Benutzer oder von einem Hostcomputer über Steuerleitungen **26** wie etwa einen Systembus, der den Antrieb **21**, Positionierungsmittel **25**, Lesemittel **27** und Datenauswahlmittel **28** und möglicherweise auch den Puffer **29** zur Füllstandsregelung verbindet. Zu diesem Zweck umfasst die Steuereinheit **20** Steuerschaltungen wie etwa einen Mikroprozessor, einen Programmspeicher und Steuergates oder eine Zustandsmaschine.

[0035] Kompression und Dekompression sind wohlbekannt. Während der Dekompression wird ein inverser Prozess angewendet, um das ursprüngliche Signal wiederherzustellen. Wenn das ursprüngliche digitalisierte Signal exakt wiederhergestellt wird, ist die (De-)Kompression verlustfrei, während bei einer verlustbehafteten (De-)Kompression manche Details des ursprünglichen Signals nicht wiedergegeben werden. Diese weggelassenen Details sind für das menschliche Ohr oder Auge im Wesentlichen nicht wahrnehmbar. Die meisten bekannten Systeme, wie etwa MPEG, verwenden eine verlustbehaftete Kompression für Audio und Video; eine

verlustfreie Kompression wird für das Speichern von Computerdaten angewendet.

[0036] Die Datenauswahl **28** ist dazu vorgesehen, Steuerinformationen aus den gelesenen Daten wiederzugewinnen und eventuelle Fülldaten zu verwerfen, die während des Aufzeichnens hinzugefügt wurden. Die Drehzahl kann unter Verwendung des durchschnittlichen Füllstandes des Puffers **29** eingestellt werden, z. B. indem die Drehzahl verringert wird, wenn der Puffer im Mittel mehr als zu 50% voll ist.

[0037] [Fig. 3](#) zeigt eine Aufzeichnungseinrichtung zum Schreiben von Informationen auf einem (wieder)beschreibbaren Aufzeichnungsträger **11**. Während eines Schreibvorgangs werden Markierungen, welche die Informationen repräsentieren, auf dem Aufzeichnungsträger ausgebildet. Die Markierungen können in einer beliebigen optisch lesbaren Form vorliegen, z. B. in der Form von Bereichen, deren Reflexionskoeffizient sich von dem ihrer Umgebung unterscheidet, durch Aufzeichnen in Materialien wie etwa Farbstoff, Legierung oder Phasenumwandlungsmaterial, oder in der Form von Bereichen mit einer Magnetisierungsrichtung, die von der ihrer Umgebung verschieden ist, wenn in einem magneto-optischen Material aufgezeichnet wird. Das Schreiben und Lesen von Informationen zum Aufzeichnen auf optischen Platten und die anwendbaren Regeln für das Formatieren, die Fehlerkorrektur und die Kanalcodierung sind in der Technik wohlbekannt, z. B. vom CD-System her. Die Markierungen können durch einen Lichtpunkt **23** gebildet werden, der auf der Aufzeichnungsschicht mittels eines elektromagnetischen Strahls **24** von einer Laserdiode erzeugt wird. Die Aufzeichnungseinrichtung umfasst ähnliche Grundelemente wie das Gerät zum Lesen, das oben in Verbindung mit [Fig. 2](#) beschrieben wurde, d. h. eine Steuereinheit **20**, Antriebsmittel **21** und Positionierungsmittel **25**, doch sie weist einen Schreibkopf **39** auf. Die Informationen werden dem Eingang von Kompressionsmitteln **35** zugeführt, welche in einem separaten Gehäuse angebracht sein können. Die mit variabler Bitrate komprimierten Informationen am Ausgang der Kompressionsmittel **35** werden zu einem Puffer **36** weitergegeben. Vom Puffer **36** werden die Daten zu Datenkombinationsmitteln **37** gesendet, um Fülldaten und weitere Steuerdaten hinzuzufügen. Der gesamte aufzuzeichnende Datenstrom wird Schreibmitteln **38** zugeführt. Der Schreibkopf **39** ist mit den Schreibmitteln **38** gekoppelt, welche einen Formstierer, einen Fehlercodierer und einen Kanalmodulator umfassen. Die Daten, die dem Eingang der Schreibmittel **38** zugeführt werden, werden entsprechend Formatierungs- und Codierungsregeln, die weiter unten erläutert werden, über logische und physikalische Sektoren verteilt und in ein Schreibsignal für den Schreibkopf **39** umgewandelt. Die Steuereinheit **20** ist dazu vorgesehen, den Puffer **36**, die Datenkombinationsmittel **37** und die Schreibmittel **38** über Steuerleitungen **26** zu steuern und den Positionierungsvorgang auszuführen, wie oben für das Lesegerät beschrieben wurde. Das Aufzeichnungsgerät kann durch die Merkmale eines Abspielgerätes und eines kombinierten Lese-Schreibkopfes ebenfalls für ein Lesen vorgesehen sein.

[0038] [Fig. 4](#) zeigt ein umfassendes System gemäß der Erfindung, das mit einem Codierer, einem Träger und einem Decodierer ausgestattet ist. Die Ausführungsform wird zum Codieren, Speichern und schließlich zum Decodieren einer Folge von Samples oder Multibit-Symbolen verwendet, die von einem Audio- oder Videosignal oder von Daten abgeleitet wurde. Das Endgerät **120** empfängt einen Strom, welcher Acht-Bit-Symbole aufweisen kann. Der Splitter **122** überträgt rekurrent und zyklisch erste Symbole, die für die Hinweiswörter (Clue Words) bestimmt sind, zu dem Codierer **124** und alle anderen Symbole zu dem Codierer **126**. In dem Codierer **124** werden die Hinweiswörter gebildet, indem die Daten zu Codewörtern eines ersten Multisymbol-Fehlerkorrekturcodes codiert werden. Dieser Code kann ein Reed-Solomon-Code, ein Produktcode, ein verschachtelter Code oder eine Kombination davon sein. Im Codierer **126** werden die Zielwörter durch Codieren zu Codewörtern eines zweiten Multisymbol-Fehlerkorrekturcodes gebildet. In [Fig. 5](#) haben alle Codewörter eine einheitliche Länge, doch dies ist keine Einschränkung. Die Hinweis-Wörter können einen wesentlich höheren Grad des Fehlerschutzes aufweisen. Dies kann durch eine größere Anzahl von Prüfsymbolen, durch eine kleinere Anzahl von Datensymbolen oder durch eine Kombination davon bewirkt werden.

[0039] In Block **128** werden die Codewörter zu einem oder mehreren Ausgängen übertragen, von welchen eine willkürliche Anzahl dargestellt wurde, so dass die Verteilung auf einem später zu erörternden Medium gleichmäßig wird. Block **130** symbolisiert das einheitliche Medium, wie etwa ein Band oder eine Platte, welches die codierten Daten empfängt. Diese kann sich auf ein direktes Schreiben in einer Kombination von Schreibmechanismus plus Medium beziehen. Stattdessen kann das Medium auch eine Kopie von einem als Master codierten Medium wie etwa einem Stempel sein. Die Speicherung kann optisch und vollständig seriell erfolgen, doch können auch andere Konfigurationen verwendet werden. In Block **132** werden die verschiedenen Wörter wieder von dem Medium gelesen. Danach werden die Hinweiswörter des ersten Codes zu einem Decodierer **134** gesendet und decodiert, basierend auf den ihnen innewohnenden Redundanzen. Ferner kann, wie nachfolgend bei der Erörterung von [Fig. 5](#) ersichtlich wird, diese Decodierung Hinweise auf die Stellen von Fehlern in anderen als diesen Hinweiswörtern liefern. Außerdem kann die Information aus Synchronisationsbitspalten in Bezug auf darin enthaltene Interferenzen analysiert werden, um unmittelbar zusätzliche Hinweise für die

Zielwörter zu erzeugen. Block **135** empfängt alle Hinweise und enthält ein Programm zur Verwendung einer oder mehrerer unterschiedlicher Strategien, um diese Hinweise in Löschstellen zu übersetzen. Die Zielwörter werden in einem Decodierer **136** decodiert. Mit Hilfe der Löschstellen wird der Fehlerschutz der Zielwörter auf ein höheres Niveau gehoben. Schließlich werden alle decodierten Wörter mittels eines Elements **138** zum Ausgang **140** demultiplext, so dass sie mit dem ursprünglichen Format konform sind. Der Kürze halber wurden elektromechanische Schnittstellen zwischen den verschiedenen Teilsystemen ignoriert.

[0040] [Fig. 5](#) zeigt ein einfaches Codeformat. Die codierten Informationen wurden als in einem 480 Symbole enthaltenden Block mit 15 horizontalen Reihen und 32 vertikalen Reihen angeordnet dargestellt. Die Speicherung auf einem Medium beginnt oben links und setzt sich entlang vertikaler Reihen fort. Der schraffierte Bereich enthält Prüfsymbole: die horizontalen Reihen 4, 8 und 12 weisen jeweils 8 Prüfsymbole auf und stellen Hinweiswörter dar. Die anderen Reihen enthalten jeweils 4 Prüfsymbole und stellen Zielwörter dar. Der gesamte Block weist 408 Informationssymbole und 72 Prüfsymbole auf. Die Letzteren können auf eine stärker über die jeweiligen Wörter verteilte Weise angeordnet sein. Zusätzlich zu Obigem enthält die obere horizontale Reihe eine Darstellung von Synchronisationsbitgruppen. Diese sind auf dem Medium vorhanden, um eine Leseeinrichtung mit dem Format zu synchronisieren, enthalten jedoch im Allgemeinen weder Systemdaten noch Benutzerdaten und haben ein vorgeschriebenes Format mit viel Redundanz. Daher ist es oft leicht, eine Interferenz zu detektieren, und das Auftreten einer einzelnen oder mehrerer gestörter Synchronisationsbitgruppen, welche physisch nahe beieinander oder bei verteilten Hinweissymbolen sind, kann verwendet werden, um das Auftreten eines Bündelfehlers zu signalisieren. Dies erzeugt dann Hinweise auf eine ähnliche Weise wie die Hinweiswörter.

[0041] Der Reed-Solomon-Code ermöglicht, in jedem Hinweiswort bis zu vier Symbolfehler zu korrigieren. Eigentliche Symbolfehler wurden durch Kreuze bezeichnet. Demzufolge können alle Hinweiswörter korrekt decodiert werden, insofern als sie nicht mehr als vier Fehler aufweisen. Insbesondere die Wörter 2 und 3 können jedoch nicht auf der Grundlage ihrer eigenen Redundanz allein decodiert werden. In der Figur stellen alle Fehler außer **62**, **66**, **68** Fehlerstrings (Fehlerzeichenketten) dar. Nur die Strings **52** und **58** durchqueren mindestens drei aufeinander folgende Hinweiswörter und werden als Fehlerbündel betrachtet, die zur Folge haben, dass mindestens alle dazwischen befindlichen Symbolstellen ein Lösch-Flag erhalten. Außerdem können Zielwörter unmittelbar vor dem ersten Hinweiswort-Fehler des Bündels und Zielwörter unmittelbar nach dem letzten Hinweiswort-Fehler des Bündels ein Lösch-Flag an der betreffenden Stelle erhalten, in Abhängigkeit von der verfolgten Strategie. String **54** wird nicht als ein Bündel betrachtet, da er zu kurz ist.

[0042] Infolgedessen erzeugen zwei der Fehler in Wort 4 ein Lösch-Flag in den zugehörigen vertikalen Reihen. Dies macht die Wörter 2 und 4 korrigierbar, jeweils mit einem Fehlersymbol und zwei Löschsymbolen. Jedoch stellen die zufälligen Fehler **62**, **68** und auch der String **54** keine Hinweise für die Wörter 5, 6, 7 dar, da jeder von ihnen nur ein Hinweiswort enthält. Manchmal verursacht eine Löschung ein Null-Fehler-Muster, da ein beliebiger Fehler in einem 8-Bit-Symbol eine Wahrscheinlichkeit von $1/256$ aufweist, wieder ein korrektes Symbol zu verursachen. Ebenso kann ein langes Bündel, das ein bestimmtes Hinweiswort durchquert, ein korrektes Symbol darin erzeugen. Durch eine Überbrückungsstrategie zwischen vorhergehenden und nachfolgenden Hinweissymbolen desselben Bündels wird dieses korrekte Symbol dann in das Bündel integriert und auf dieselbe Weise wie fehlerbehaftete Hinweissymbole in Löschwerte für entsprechende Zielsymbole übersetzt. Die obigen Entscheidungen können entsprechend der Decodierpolitik variieren.

[0043] Die Hinweise durch die Synchronisationsbitgruppen können in ähnlicher Weise verwendet werden wie diejenigen von den Hinweiswörtern.

[0044] Die Relevanz der vorliegenden Erfindung wird durch neuere Verfahren zur digitalen optischen Speicherung bewirkt. Ein besonderes Merkmal ist, dass im Falle eines Substrate-Incident Reading (Lesen durch ein durchsichtiges Substrat hindurch) die obere durchlässige Schicht nur 100 Mikrometer dünn ist. Die Kanalbites haben eine Größe von etwa 0,14 Mikrometer, so dass ein Datenbyte bei einer Kanalrate von $2/3$ dann eine Länge von nur 1,7 Mikrometern hat. An der oberen Fläche hat der Strahl einen Durchmesser von etwa 125 Mikrometern. Eine Plastikkassette oder Hülle für die Platte verringert dann die Wahrscheinlichkeit großer Bündel. Nicht konforme Partikel von weniger als 50 Mikrometern können kurze Defekte verursachen, obwohl die Erfindung auch gegen längere Defekte von Nutzen ist. Es wurde ein Defektmodell verwendet, bei dem Defekte von 50 Mikrometern durch Fehlerfortpflanzung zu Bündeln von 200 Mikrometern führen können, die etwa 120 Bytes entsprechen. Ein spezielles Modell weist Bündel mit fester Größe von 120B auf, welche zufällig mit einer Wahrscheinlichkeit pro Byte von $2,6 \cdot 10^{-5}$ beginnen, oder im Mittel ein Bündel pro Block von 32 kB. Die Erfindung wurde für optische Plattenspeicher entwickelt, jedoch würden auch andere Konfigurationen wie etwa mehrspurige Magnetbänder und andere Technologien wie etwa magnetische und magnetooptische von dem

verbesserten Ansatz profitieren.

Erörterung eines bevorzugten Formats der Information

[0045] Vor dem Aufzeichnen werden Benutzerdaten, die von einer Quelle empfangen wurden, welche eine Anwendung oder ein Host sein kann, in einer Anzahl von aufeinander folgenden Schritten formatiert, welche unter Bezugnahme auf [Fig. 19](#) eingehender erläutert werden, nämlich: Datenrahmen, Datensektoren, ECC-Sektoren, ein ECC-Cluster, ein BIS-Cluster, ein physischer Cluster und Aufzeichnungsrahmen.

[0046] Die Daten werden in Partitionen von 64k aufgezeichnet, die physische Cluster genannt werden und welche jeweils 32 Datenrahmen mit 2048 Bytes von Benutzerdaten enthalten. Ein physischer Cluster wird durch zwei Fehlerkorrekturmechanismen geschützt:

- erstens, einen Long-Distance-(LDS) (248,216,33) Reed-Solomon-(RS-)Fehlerkorrekturcode;
- zweitens, die Daten werden mit einem Burst Indicator Subcode (BIS) gemultiplext, welcher aus (62,30,33) Reed-Solomon-(RS-)Codewörtern besteht. Die Anzahl von Paritätssymbolen ist für die zwei Codekategorien gleich, was es ermöglicht, für beide Fälle dieselbe Decoder-Hardware zu verwenden. Der BIS-Code wird verwendet, um längere Bündelfehler anzuzeigen, wodurch der LDS-Code Löschkorrekturen effizienter durchführen kann. Alle Daten sind in einem Array angeordnet, wie in [Fig. 6](#) dargestellt. Es ist anzumerken, dass die horizontale und die vertikale Richtung in Bezug auf [Fig. 5](#) vertauscht worden sind. Das Array wird entlang einer horizontalen Richtung gelesen und wird nach dem Einfügen von Synchronisationsmustern und zusätzlichen d. c. Steuerbits sowie nach Modulation auf der Platte aufgezeichnet.

[0047] Die Fehlerkorrekturcodes werden in vertikaler Richtung angewendet, was ein gutes grundlegendes Aufbrechen von Bündelfehlern auf der Platte bewirkt. Zusätzlich sind die LDS-Codewörter in einer diagonalen Richtung verschachtelt worden. Zum Zwecke einer Adressierung wird ein ganzer physischer Cluster in 16 physische Sektoren unterteilt, die jeweils aus 32 aufeinander folgenden Reihen bestehen.

[0048] Ein Datenrahmen besteht aus 2052 Bytes: 2048 Benutzerdatenbytes, die mit d_0 bis d_{2047} nummeriert sind, und 4 Error Detection Code (EDC, Fehlerdetektionscode) Bytes, die mit e_{2048} bis e_{2851} nummeriert sind. Die Bytes e_{2048} bis e_{2851} enthalten einen Fehlerdetektionscode, der über 2048 Bytes des Datenrahmens berechnet wurde. Der Datenrahmen wird als ein Feld von einzelnen Bits betrachtet, beginnend mit dem höchstwertigen Bit des ersten Benutzerdatenbytes d_0 und endend mit dem niedrigstwertigen Bit des letzten EDC-Bytes e_{2851} . Das höchstwertige Bit ist b_{16415} und das niedrigstwertige Bit ist b_0 . Jedes Bit b_i des EDC ist wie folgt dargestellt, für $i = 0$ bis 31:

$$EDC(x) = \sum_{i=31}^0 b_i x^i \text{ mod } G(x)$$

wobei:

$$l(x) = \sum_{i=16415}^{32} b_i x^i, \quad G(x) = x^{32} + x^{31} + x^4 + 1 \quad (1)$$

[0049] Als nächstes werden zwei Datenrahmen (A, B) in einem Array von 19 Spalten mal 216 Zeilen, Daten-sektor genannt, angeordnet. Das Füllen des Arrays erfolgt Spalte für Spalte, beginnend am oberen Ende der ersten mit Byte $d_{0,A}$ und endend am unteren Ende der letzten Spalte mit Byte $e_{2051,B}$, siehe [Fig. 8](#).

[0050] Als nächstes werden in [Fig. 9](#) die Bytes in jeder Spalte des Datensektors wie folgt umnummeriert, beginnend am oberen Ende der Spalte: $d_{L,0}, d_{L,1} \dots d_{L,i} \dots$ bis $d_{L,215}$, wobei L die Spaltennummer (0 .. 18) ist. Der ECC-Sektor wird vervollständigt, indem jede Spalte um 32 Paritätsbytes eines (248,216,33) Long-Distance-RS-Codes verlängert wird. Die Paritätsbytes sind: $P_{L,216}, P_{L,217}, \dots P_{L,j} \dots$ bis $P_{L,247}$.

[0051] Der Long-Distance-RS-Code ist über dem finiten Feld $GF(2^8)$ definiert. Von null verschiedene Elemente des finiten Feldes $GF(2^8)$ werden durch ein primitives Element α erzeugt, welches eine Wurzel des primitiven Polynoms $p(X) = X^8 + X^4 + X^3 + X^2 + 1$ ist. Die Symbole von $GF(2^8)$ werden durch 8-Bit-Bytes dargestellt, unter Verwendung einer Polynombasis-Darstellung mit $(\alpha^7, \alpha^6, \alpha^5, \dots, \alpha^2, \alpha, 1)$ als Basis. Die Wurzel α wird als $\alpha = 00000010$ dargestellt. Jedes LDS-Codewort, dargestellt durch den Vektor $l_{ds} = (d_{L,0} \dots d_{L,i} \dots d_{L,215} P_{L,216} \dots P_{L,j} \dots P_{L,247})$, gehört zu einem Reed-Solomon-Code über $GF(2^8)$, mit 216 Informationsbytes und 32 Paritätsbytes. Ein solches Codewort kann durch ein Polynom $l_{ds}(x)$ vom Grad 247 dargestellt werden, welches einige Koeffizienten null haben kann, wobei die höchsten Grade dem Informationsteil des Vektors ($d_{L,0} \dots$ usw.) entsprechen und die niedrigsten Grade dem Paritätsteil ($P_{L,216} \dots$ usw.) entsprechen. Nun ist $l_{ds}(x)$ ein Vielfaches des Gene-

ratorpolynoms $g(x)$ des LDS-Codewortes. Das Generatorpolynom ist:

$$g(x) = \prod_{i=0}^{31} (x - \alpha^i)$$

[0052] Der LDS-Code ist systematisch: Die 216 Informationsbytes erscheinen unverändert in den Positionen höchster Ordnung jedes Codewortes. Die Paritätsprüfungsmatrix von Code lds ist so beschaffen, dass gilt: $HLDS \cdot lds^T = 0$ für alle LDS-Codewörter lds.

[0053] Die zweite Zeile $H_{LDS,2}$ der Paritätsprüfungsmatrix H_{LDS} ist gegeben durch $H_{LDS,2} = (\alpha^{247} \alpha^{246} \dots \alpha^2 \alpha 1)$ und entspricht der Nullstelle α des Generatorpolynoms $g(x)$, welche die Codewortpositionen definiert, die für Fehlerstellen zu verwenden sind.

[0054] Nach dem Erzeugen der LDS-Codewörter in den ECC-Sektoren werden 16 aufeinander folgende ECC-Sektoren zu einem ECC-Cluster kombiniert, indem die 16-19 Spalten der Höhe 248 in Zweiergruppen gemultiplext werden, einschließlich der Paritäten. Auf diese Weise werden 152 neue Spalten mit einer Höhe von 496 Bytes gebildet, wie in [Fig. 10](#) dargestellt. Die Nummerierung der Bytes erfolgt entsprechend $d_{L,M,N}$, wobei: $L = 0..18$ die LDS-Codewort-Nummer innerhalb des ECC-Sektors ist, $M = 0..247$ die Bytenummer innerhalb des LDS-Codewortes ist, $N = 0..15$ die ECC-Sektor-Nummer ist.

[0055] Um die Fähigkeiten zur Korrektur von Bündelfehlern noch weiter zu verbessern, wird eine zusätzliche Verschachtelung eingeführt, indem die Bytes in horizontaler Richtung durch alle Zeilen des ECC-Clusters hindurch unnummeriert werden, siehe [Fig. 11](#). Nun werden alle Zeilen eines ECC-Clusters in Zweiergruppen um $\text{mod}(k-3,152)$ Bytes nach links verschoben, beginnend ab Zeile 2 aufwärts, wobei die erste Zeile die Zeile 0; $k = \text{div}(\text{row-number}, 2)$ ist. Die Bytes, welche auf der linken Seite herausgeschoben werden, treten von der rechten Seite her wieder in das Array ein, siehe [Fig. 12](#). Nach diesem Prozess werden die Bytes noch einmal in horizontaler Richtung durch alle Zeilen hindurch unnummeriert, was die Nummerierung D_0 bis D_{75391} zum Ergebnis hat, die in [Fig. 6](#) angegeben ist. Die Umnummerierung der Bytes bewirkt eine ungleichförmige Abbildung von logischen Adressen auf physische Adressen. Die Folgen davon werden weiter unten erörtert.

[0056] Nach der Verschachtelung wird der ECC-Cluster in 4 Gruppen von jeweils 38 Spalten aufgespalten. Zwischen die 4 Gruppen werden 3 Spalten von jeweils einem Byte Breite eingefügt. Diese Spalten tragen die Adressinformationen, welche die Daten betreffen, die in dem verschachtelten ECC-Cluster enthalten sind. Sie bestehen aus (62,30,33) RS-BIS-Codewörtern mit 30 Bytes Information und 32 Bytes Parität. Aufgrund der hohen Fehlerschutz-Fähigkeiten und eines anspruchsvollen Verschachtelungsschemas können diese Spalten auch eine zuverlässige Anzeige von Bündelfehlern bieten.

[0057] Ein Array von 3 mal 496 Bytes, das von den 3 BIS-Spalten aus dem physischen Cluster gebildet wird, wird ein BIS-Cluster genannt. Der Inhalt des BIS-Clusters wird gebildet, indem alle BIS-Codewörter eines BIS-Clusters in den 24 Spalten eines Arrays von 24 mal 62 Bytes angeordnet werden, siehe [Fig. 14](#).

[0058] Der BIS-RS-Code ist über dem finiten Feld $GF(2^8)$ definiert. Die von null verschiedenen Elemente des finiten Feldes $GF(2^8)$ werden durch ein primitives Element α erzeugt, welches eine Wurzel des primitiven Polynoms $p(X) = X^8 + X^4 + X^3 + X^2 + 1$ ist. Die Symbole von $GF(2^8)$ werden durch 8-Bit-Bytes dargestellt, unter Verwendung der Polynombasis-Darstellung mit $(\alpha^7, \alpha^6, \dots, \alpha^2, \alpha, 1)$ als Basis. Die Wurzel α wird als $\alpha = 00000010$ dargestellt.

[0059] Jedes BIS-Codewort ist durch einen Vektor $bis = (b_{c,0} \dots b_{c,i} \dots b_{c,29}, Pb_{c,30} \dots Pb_{c,61})$ in einem Reed-Solomon-Code über $GF(2^8)$, mit 32 Paritätsbytes und 30 Informationsbytes. Ein solches Codewort kann durch ein Polynom $bis(x)$ vom Grad 61 dargestellt werden, welches einige Koeffizienten null haben kann, wobei die höchsten Grade den Informationsteil des Vektors ($b_{c,0} \dots$ usw.) darstellen und die niedrigsten Grade den Paritätsteil des Vektors ($Pb_{c,30} \dots$ usw.). Jedes Codewort ist ein Vielfaches des Generatorpolynoms $g(x)$ des BIS-Codewortes:

$$g(x) = \prod_{i=0}^{31} (x - \alpha^i)$$

[0060] Der BIS-Code ist systematisch: Die 30 Informationsbytes erscheinen unverändert in den Positionen höchster Ordnung jedes Codewortes. Die Paritätsprüfungsmatrix von Code bis ist so beschaffen, dass gilt:

$H_{\text{BIS}} \cdot \text{bis}^T = 0$ für alle BIS-Codewörter bis. Die zweite Zeile $h_{\text{BIS } 2}$ der Paritätsprüfungsmatrix H_{BIS} ist gegeben durch $h_{\text{BIS } 2} = (\alpha^{61} \alpha^{60} \dots \alpha^2, \alpha, 1)$. Sie entspricht der Nullstelle α des Generatorpolynoms $g(x)$ und definiert die Codewortpositionen, die für Fehlerstellen zu verwenden sind.

[0061] [Fig. 15](#) zeigt die Rahmenstruktur zur Erläuterung der Verschachtelung der Hauptdaten. Die verschiedenen physischen Sektoren und Spalten sind mit einer entsprechenden Nummerierung versehen. Es gibt 304·[248,216,33] LDS-Codewörter und 24·[62,30,33] BIS-Codewörter.

[0062] Ein logischer Sektor von 2kB enthält 9,5 LDS-Codewörter, in welchen 2048 Benutzerdatenbytes und 4 EDC-Bytes codiert worden sind, und ferner 22,5 BIS-Bytes zum Speichern eines Headers, 4ID + 2EDC Bytes, Copyright-Daten und 6 Copyright Management Information CPR_MAI. Ferner sind 10,5 BIS-Bytes für eine mögliche zukünftige Verwendung reserviert worden. Ein physischer Sektor von 4 kB besteht aus 31 Zeilen, wobei die Header-Bytes von zwei logischen Sektoren von 2 kB an physisch vorgeschriebenen Positionen gespeichert sind. [Fig. 16](#) zeigt die eigentliche Verschachtelung. Hierbei enthält das i -te LDS-Codewort, mit $0 \leq i \leq 303$, 248 Bytes $d_{i,j}$, wobei j von folgender Form ist: $j = [(i \bmod 2) + 2 \cdot a] \cdot 152 + \{[(i \text{ div } 2) - 3 \cdot a] \bmod 152\}$ für gewisse Werte $0 \leq a \leq 247$. Für die Anbringung der Paritäten sind verschiedene Stellen möglich, wie etwa das untere Ende, oder permutiert gemäß $a + 31 \cdot i$, $a \geq 22 \dots$

[0063] Als nächstes wird die Erzeugung eines BIS-Clusters erörtert. Nach dem Erzeugen der BIS-Codewörter wird der BIS-Block auf eine verschachtelte Art und Weise auf ein Array von $496 = (16 \times 31)$ Zeilen·3 Spalten abgebildet. Dieses neue Array wird ein BIS-Cluster genannt, dargestellt in [Fig. 14A](#). Die Positionierung der Bytes aus einem BIS-Block ([Fig. 14](#)) in einem BIS-Cluster wird zuerst durch mathematische Ausdrücke angegeben. Zu diesem Zweck wird der BIS-Cluster entsprechend den in [Fig. 6](#) dargestellten physischen Sektoren unterteilt. Die Sektoren werden mit $s = 1 \dots 15$ nummeriert, die Zeilen in einem solchen Sektor werden mit $r = 0 \dots 30$ nummeriert, und die Spalten werden mit $e = 0 \dots 2$ nummeriert, siehe [Fig. 14A](#). Nun bekommt ein Byte $b_{N,C}$ die folgende Position:

Sektornummer $s = \text{mod}\{[\text{div}(N, 2) + 8 - \text{div}(C, 3)], 8\} + 8 - \text{mod}(N, 2)$

Zeilennummer $r = \text{div}(N, 2)$

Spaltennummer $e = \text{mod}\{[C + \text{div}(N, 2)], 3\}$

[0064] Die Bytenummer m gibt die laufende Nummer B_m an, wenn der physische Cluster auf die Platte geschrieben wird, wie in [Fig. 6](#) dargestellt, entsprechend $m = (s \cdot 31 + r) \cdot 3 + e$. Die wesentlichen Elemente des Verschachtelungsschemas sind in den [Fig. 17](#), [Fig. 18](#) beispielhaft dargestellt und betreffen Folgendes:

Jede Zeile eines BIS-Blockes wird in 8 Gruppen von jeweils 3 Bytes aufgespalten, wobei diese Drei-Byte-Gruppen jeweils in einer entsprechenden Zeile des BIS-Clusters angeordnet werden.

[0065] Die geraden Zeilen des BIS-Blockes werden auf die Sektoren 0 bis 7 abgebildet, die ungeraden Zeilen des BIS-Blockes werden auf die Sektoren 8 bis 15 abgebildet. Die acht Drei-Byte-Gruppen aus einer geraden Zeile des BIS-Blockes werden jeweils auf dieselbe Zeile von acht aufeinander folgenden Sektoren gesetzt, wobei die Sektoren in einer Richtung verwendet werden, welche umgekehrt zu ihrer Nummerierung ist. Es wurde festgestellt, dass diese Umkehrung ein besseres Zerstreuen von Bündelfehlern bewirkt. Der Anfangssektor für jede Zeile des BIS-Blockes ist einen Sektor höher als für die vorhergehende Zeile.

– Zeile $N = 0$ des BIS-Blockes wird auf die Zeilen $r = 0$ der Sektoren 0, 7, 6, 5, ..., 2, 1 gesetzt.

– Zeile $N = 2$ des BIS-Blockes wird auf die Zeilen $r = 1$ der Sektoren 1, 0, 7, 6, ..., 3, 2 gesetzt.

– Zeile $N = 4$ des BIS-Blockes wird auf die Zeilen $r = 2$ der Sektoren 2, 1, 0, 7, ..., 4, 3 gesetzt.

– Diese Prozedur wird zyklisch wiederholt bis zur Zeile $N = 60$, welche auf die Zeilen $r = 30$ der Sektoren 6, 5, 4, ..., 0, 7 gesetzt wird.

[0066] Nun wird innerhalb jedes Sektors jede Zeile zyklisch um $\text{mod}(r, 3)$ Positionen nach rechts verschoben; so wird Zeile $r = 0$ überhaupt nicht verschoben, Zeile $r = 1$ wird um 1 verschoben, Zeile $r = 2$ wird um 2 verschoben, Zeile $r = 3$ wird nicht verschoben, Zeile $r = 4$ wird um 1 verschoben usw. Für die ungeraden Zeilen des BIS-Blockes wird nach einer entsprechenden Prozedur verfahren.

[0067] Diesbezüglich zeigt [Fig. 17](#) ein Beispiel der teilweisen Abbildung von BIS-Bytes auf die ersten acht Sektoren und [Fig. 18](#) ein Beispiel der teilweisen Abbildung von BIS-Bytes auf die letzten acht Sektoren, welche nach dem Obigen unmittelbar verständlich sind.

[0068] [Fig. 19](#) zeigt in Form einer Zusammenfassung eine schematische Darstellung des Gesamtprozesses der Codierung. Die Benutzerdaten, die von einer Quelle empfangen wurden, welche ein Host oder eine Anwendung sein kann, werden zuerst in Datenrahmen aufgeteilt, welche jeweils aus 2048+4 Bytes bestehen; wie

in Block **200** der Figur dargestellt, werden 32 von diesen Rahmen für den nächsten Codierungsschritt berücksichtigt. In Block **202** wird ein Datenblock gebildet und in 304 Spalten mit jeweils 216 Zeilen angeordnet. In Block **204** wird ein Long-Distance-Code-Block durch Hinzufügen von 32 Zeilen Parität gebildet. In Block **206** wird ein ECC-Cluster entsprechend 152 Spalten und 496 Zeilen angeordnet. Dieser wird so angeordnet, dass er die vier mit ECC bezeichneten Sektionen in dem physischen Cluster-Block **218** füllt, welcher die allumfassende Codeformat-Entität ist.

[0069] Die Adress- und Steuerdaten, die durch das Aufzeichnungssystem hinzugefügt werden, werden ebenfalls in aufeinander folgenden Schritten umgewandelt. Zuerst werden die logische Adresse und die Steuerdaten in Block **208** in 32·18 Bytes angeordnet. Die logischen Adressen sind diejenigen, welche zu Benutzerfunktionalitäten gehören, und können Aspekte angeben, welche eine Dauer der Wiedergabe eines Benutzerprogramms betreffen. Auch die physischen Adressen werden in Block **210** in 16·9 Bytes angeordnet. Die physischen Adressen betreffen physische Abstände auf dem Träger. Infolge der wiederholten Umnummerierung und Verschachtelung ging der Zusammenhang zwischen physischen und logischen Adressen verloren. Elemente, die in einem Programm nahe aufeinander folgen, können durch einen beträchtlichen physischen Abstand voneinander getrennt sein, und umgekehrt. Außerdem ist die Abbildung nicht gleichmäßig fortschreitend. In Block **212** werden die Adressen in einem Zugriffsblock von 24 Spalten mal 30 Zeilen kombiniert. In Block **214** werden 32 Zeilen Parität hinzugefügt. In Block **216** werden diese zu einem BIS-Cluster von 3 Spalten und 496 Zeilen angeordnet. Diese füllen die drei BIS-Spalten in Block **218**. Außerdem wird eine Spalte von Synchronisationsbitgruppen hinzugefügt, so dass ein physischer Cluster von 155 Spalten mal 496 Zeilen gebildet wird. Zusammen bilden diese 16 physische Sektoren, welche zu 496 Aufzeichnungsrahmen gruppiert sind, wie dargestellt.

Bezugszeichenliste

[Fig. 6](#)

words	Wörter
physical sector	Physischer Sektor
data stream an disc	Datenstrom auf Platte

[Fig. 7](#)

User Data Benutzerdaten

[Fig. 8](#)

19 columns	19 Spalten
216 rows with data	216 Zeilen mit Daten

[Fig. 9](#)

19 code words	19 Codewörter
216 rows with data	216 Zeilen mit Daten
32 rows with parity	32 Zeilen mit Parität

[Fig. 10](#)

16 ECS Sectors = 152 columns	16 ECS-Sektoren = 152 Spalten
432 rows with data	432 Zeilen mit Daten
64 rows with parity	64 Zeilen mit Parität

[Fig. 11](#)

16 ECC Sectors with 2 data frames each (152 columns)	16 ECC-Sektoren mit je 2 Datenrahmen (152 Spalten)
38 columns	38 Spalten
432 rows data	432 Zeilen Daten
64 rows parity	64 Zeilen Parität

[Fig. 12](#)

152 bytes 152 Bytes
shift Verschiebung

[Fig. 13](#)

BIS column BIS-Spalte
8 BIS codewords = 496 bytes 8 BIS-Codewörter = 496 Bytes

[Fig. 14](#)

codeword Codewort
1 BIS codeword = 62 bytes 1 BIS-Codewort = 62 Bytes
30 information 30 Information
32 parity 32 Parität

[Fig. 14A](#)

3 columns 3 Spalten
496 rows 496 Zeilen
31 rows 31 Zeilen
16 sectors 16 Sektoren

[Fig. 15](#)

BIS columns IS-Spalten
etc. usw.
Data stream an disk Datenstrom auf Platte
physical sector physischer Sektor

[Fig. 16](#)

row + 2 Zeile + 2
column - 3 mod 152 Spalte - 3 mod 152
column - 3 Spalte - 3
LDS code Word direction Richtung der LDS-Codewörter
Column numbers Spaltennummern

[Fig. 17\(1\), Fig. 17\(2\), Fig. 18](#)

sector s Sektor s
row r Zeile r
byte number N,C from BIS Block Bytenummer N,C aus BIS-Block
column e Spalte e
shift right (= mod(r,3)) nach rechts verschieben (= mod(r,3))
filling in upward direction Füllen in Aufwärtsrichtung
start of Block row Beginn von Block-Zeile
continuation of Block row Fortsetzung von Block-Zeile
end of Block row = Ende von Block-Zeile

[Fig. 19](#)

200	Benutzerdaten
	32 Rahmen-(2048 +4) Bytes
202	Datenblock
	304 Spalten·216 Zeilen
204	LDC-Block
	304 Spalten
216	Zeilen Daten
	32 Zeilen Parität
206	ECC-Cluster
	152 Spalten·496 Zeilen
208	Logische Adresse + Steuerdaten
	32·18 Bytes
210	Physische Adresse 16·9 Bytes
212	Zugriffsblock 24 Spalten·30 Zeilen
214	BIS Block
	24 Spalten
	30 Zeilen Daten
	32 Zeilen Parität
216	BIS Cluster = BIS-Cluster
	3 Spalten·496 Zeilen
496 Recording Frames	496 Aufzeichnungsrahmen
16 Physical Sectors	16 Physische Sektoren
Physical Cluster (155 columns * 496 rows)	Physischer Cluster (155 Spalten * 496 Zeilen)

Patentansprüche

1. Verfahren zur Codierung von Mehrwortinformation, welche auf Multibit-Symbolen beruht, die in relativer Nachbarschaft bezüglich eines Mediums angeordnet sind, wobei es wortweise Verschachtelungs- und wortweise Fehlerschutzcode-Fähigkeiten verschafft, um so Fehlerlokalisierungs-Hinweise über Mehrwortgruppen hinweg zu liefern, gekennzeichnet durch das Generieren von Hinweiswörtern mit hoher Schutzfähigkeit (BIS), welche unter Hinweisspalten verschachtelt sind, und außerdem von Synchronisationsspalten, die aus Synchronisationsbitgruppen gebildet werden, und Positionieren der Synchronisationsspalten, wo die Hinweisspalten vergleichsweise seltener angeordnet sind, zum Erzeugen solcher Hinweise sowohl in Hinweiswörtern mit hoher Schutzfähigkeit als auch in Synchronisationsspalten, wobei die Hinweise auf Zielwörter mit niedriger Schutzfähigkeit (LDS) gerichtet sind, welche in einer im Wesentlichen gleichförmigen Art und Weise unter Zielspalten verschachtelt sind, welche Spaltengruppen einheitlicher Größe zwischen periodischen Anordnungen von Hinweisspalten und Synchronisationsspalten bilden.

2. Verfahren nach Anspruch 1, und Anordnen der Information in physischen Cluster einheitlicher Größe, welche jeweils eine einzige Synchronisationsspalte und eine ungerade Anzahl von Hinweisspalten aufweisen.

3. Verfahren nach Anspruch 1, wobei Benutzerdaten ausschließlich den Zielspalten zugewiesen werden und Systemdaten wenigstens überwiegend den Hinweisspalten zugewiesen werden.

4. Verfahren nach Anspruch 1, wobei auf einer niedrigsten Ebene ein Mehrsymbol-Datenrahmen von Zielsymbolen eine Mehrsymbol-, jedoch bitorganisierte Fehlerdetektions-Bitgruppe EDC enthält.

5. Verfahren nach Anspruch 4, wobei auf einer nächsthöheren Ebene ein ECC-Sektor mehrere Datenrahmen zur Verteilung über mehrere Zielwörter durch Hinzufügung von Reed-Solomon-Redundanz enthält.

6. Verfahren nach Anspruch 5, wobei vor dem Verschachteln verschiedene Codewortblöcke sequenziell getrennt werden, als Maßnahme zur Beschleunigung im Hinblick auf ein späteres Decodieren ([Fig. 11](#)).

7. Verfahren nach Anspruch 1, wobei das Verschachteln einem zeilenweisen inkrementellen Rotieren von Zielsymbolen innerhalb ihres Clusters überlagert wird.

8. Verfahren nach Anspruch 1, angewendet auf eine Speicherung auf einem optischen Medium.

9. Verfahren nach Anspruch 1, wobei alle Hinweiswörter und Zielwörter eine einheitliche Menge an Redundanz tragen, Zielwörter jedoch mehr Datensymbole als Hinweiswörter aufweisen.
10. Verfahren nach Anspruch 1, wobei die Anzahl der Hinweiswörter eines umfassenden Physischen Speicher-Clusters ein Vielfaches (24/3) der Anzahl von Hinweisspalten in dem Physischen Cluster ist; wobei mit gleichen Nummern versehene Symbole der Hinweiswörter in Gruppen von so vielen Symbolen, wie Hinweisspalten in einem Physischen Cluster vorhanden sind, über gleichrangigen Aufzeichnungsrahmen verschiedener Physischer Sektoren des Physischen Clusters verteilt werden, in einem gestaffelten, jedoch ansonsten gleichmäßigen Verschachtelungsschema unter den verschiedenen Aufzeichnungsrahmen.
11. Verfahren nach Anspruch 10, wobei Hinweissymbolzeilen mit gerader Nummer einer ersten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren zugewiesen werden und Hinweissymbolzeilen mit ungerader Nummer der zweiten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren zugewiesen werden.
12. Verfahren nach Anspruch 10, wobei die Zeilen von Hinweisspaltensymbolen einer gestaffelten und systematischen Rotation unter den verschiedenen Hinweisspalten unterzogen werden.
13. Verfahren nach Anspruch 10, wobei den Hinweiswörtern sowohl logische als auch physische Adressdaten zugewiesen werden, die zu dem jeweiligen Physischen Cluster gehören.
14. Verfahren zur Decodierung von Mehrwortinformation, welche auf Multibit-Symbolen beruht, die in relativer Nachbarschaft bezüglich eines Mediums angeordnet sind, welches wortweise Verschachtelungs- und wortweise Fehlerschutzcode-Fähigkeiten aufweist und so Fehlerlokalisierungs-Hinweise über Mehrwortgruppen hinweg liefert, als eine vorausgehende Maßnahme vor dem eigentlichen Decodieren, gekennzeichnet durch das Ableiten solcher Hinweise sowohl aus Hinweiswörtern mit hoher Schutzfähigkeit, welche unter Hinweisspalten verschachtelt sind, als auch aus Synchronisationsspalten, die aus Synchronisationsbitgruppen gebildet werden, durch Zugreifen auf die Synchronisationsspalten, wo die Hinweisspalten vergleichsweise seltener angeordnet sind, wobei die Hinweise auf Zielwörter mit niedriger Schutzfähigkeit gerichtet sind, welche in einer im Wesentlichen gleichförmigen Art und Weise unter Zielspalten verschachtelt sind, welche Spaltengruppen einheitlicher Größe zwischen periodischen Anordnungen von Hinweisspalten und Synchronisationsspalten bilden.
15. Verfahren nach Anspruch 14, und Zugreifen auf die Information entsprechend physischen Clustern einheitlicher Größe, welche jeweils eine einzige Synchronisationsspalte und eine ungerade Anzahl von Hinweisspalten aufweisen.
16. Verfahren nach Anspruch 14, wobei Benutzerdaten ausschließlich aus den Zielspalten und Systemdaten wenigstens im Wesentlichen ausschließlich aus den Hinweisspalten abgeleitet werden.
17. Verfahren nach Anspruch 14, und Zugreifen auf einer niedrigsten Ebene innerhalb eines Mehrsymbol-Datenrahmens von Zielsymbolen auf eine Mehrsymbol-, jedoch bitorganisierte Fehlerdetektions-Bitgruppe, als vorausgehender Schritt für die Decodierung.
18. Verfahren nach Anspruch 17, wobei auf einer nächsthöheren Ebene innerhalb eines Datensektors auf mehrere jeweilige Datenrahmen, die über mehrere Zielwörter verteilt sind, durch Auswertung von hinzugefügter Reed-Solomon-Redundanz zugegriffen wird.
19. Verfahren nach Anspruch 14, wobei eine Korrektur bezüglich der Verschachtelung vorgenommen wird, die einem zeilenweisen inkrementellen Rotieren von Zielsymbolen innerhalb ihres Clusters überlagert ist.
20. Verfahren nach Anspruch 14, angewendet auf eine Speicherung auf einem optischen Medium.
21. Verfahren nach Anspruch 14, wobei alle Hinweiswörter und Zielwörter basierend auf einer einheitlichen Menge an Redundanz darin decodiert werden, wobei in Zielwörtern jedoch mehr Datensymbole als in Hinweiswörtern zugelassen werden.
22. Verfahren nach Anspruch 14, wobei die Anzahl der Hinweiswörter eines umfassenden Physischen Speicher-Clusters ein Vielfaches (24/3) der Anzahl von Hinweisspalten in dem Physischen Cluster ist; wobei mit gleichen Nummern versehene Symbole der Hinweiswörter von Gruppen von so vielen Symbolen, wie Hin-

weisspalten in einem Physischen Cluster vorhanden sind, über gleichrangigen Aufzeichnungsrahmen verschiedener Physischer Sektoren des Physischen Clusters abgeleitet werden, in einem gestaffelten, jedoch ansonsten gleichmäßigen Entschachtelungsschema unter den verschiedenen Aufzeichnungsrahmen.

23. Verfahren nach Anspruch 22, wobei Symbolzeilen mit gerader Nummer von einer ersten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren abgeleitet werden und Symbolzeilen mit ungerader Nummer von der zweiten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren abgeleitet werden.

24. Verfahren nach Anspruch 22, wobei die Zeilen von Hinweisspaltensymbolen von einer gestaffelten und systematischen Rotation unterzogenen Symbolen entlang der verschiedenen Hinweisspalten abgeleitet werden.

25. Verfahren nach Anspruch 22, wobei von Hinweiswörtern sowohl logische Adressdaten als auch physische Adressdaten abgeleitet werden, die zu dem jeweiligen Physischen Cluster gehören.

26. Einrichtung zur Codierung von Mehrwortinformation, welche auf Multibit-Symbolen beruht, die in relativer Nachbarschaft bezüglich eines Mediums angeordnet sind, durch Vorsehen von Mitteln zur wortweisen Verschachtelung und Mitteln zur wortweisen Fehlerschutzcodierung, um so Fehlerlokalisierungs-Hinweise über Mehrwortgruppen hinweg zu liefern, gekennzeichnet durch Mittel zum Generieren von Hinweiswörtern mit hoher Schutzfähigkeit, welche unter Hinweisspalten verschachtelt sind, und außerdem von Synchronisationsspalten, die aus Synchronisationsbitgruppen gebildet werden, Positionierungsmittel zum Positionieren der Synchronisationsspalten, wo die Hinweisspalten vergleichsweise seltener angeordnet sind, zum Erzeugen der Hinweise sowohl in Hinweiswörtern mit hoher Schutzfähigkeit als auch in Synchronisationsspalten, wobei alle Hinweise auf Zielwörter mit niedriger Schutzfähigkeit gerichtet sind, welche in einer im Wesentlichen gleichförmigen Art und Weise unter Zielspalten verschachtelt sind, welche Spaltengruppen einheitlicher Größe zwischen periodischen Anordnungen von Hinweisspalten und Synchronisationsspalten bilden.

27. Einrichtung nach Anspruch 26, die außerdem Anordnungsmittel zum Anordnen der Information in physischen Clustern einheitlicher Größe aufweist, welche jeweils eine einzige Synchronisationsspalte und eine ungerade Anzahl von Hinweisspalten aufweisen.

28. Einrichtung nach Anspruch 26, die für ein Zuweisen von Benutzerdaten ausschließlich zu den Zielspalten und von Systemdaten wenigstens im Wesentlichen ausschließlich zu den Hinweisspalten vorgesehen ist.

29. Einrichtung nach Anspruch 26, die Generatormittel aufweist, um auf einer niedrigsten Ebene in einem Mehrsymbol-Datenrahmen von Zielsymbolen außerdem eine Mehrsymbol-, jedoch bitorganisierte Fehlerdetektions-Bitgruppe zu generieren.

30. Einrichtung nach Anspruch 29, die dafür vorgesehen ist, auf einer nächsthöheren Ebene durch hinzugefügte Reed-Solomon-Redundanz einen Datensektor zu bilden, der mehrere Datenrahmen zur Verteilung über mehrere Zielwörter enthält.

31. Einrichtung nach Anspruch 26, die außerdem Überlagerungsmittel aufweist, um das Verschachteln einem zeilenweisen inkrementellen Rotieren von Zielsymbolen innerhalb ihres Clusters zu überlagern.

32. Einrichtung nach Anspruch 26, die außerdem Schnittstellenmittel zur Ankopplung an ein optisches Speichermedium aufweist.

33. Einrichtung nach Anspruch 26, wobei die Codierungsmittel dazu vorgesehen sind, allen Hinweiswörtern und Zielwörtern zuzuweisen, dass sie eine einheitliche Menge an Redundanz tragen, Zielwörtern jedoch eine größere Menge an Datensymbolen zuzuweisen als Hinweiswörtern.

34. Einrichtung nach Anspruch 26, wobei die Anzahl der Hinweiswörter eines umfassenden Physischen Speicher-Clusters ein Vielfaches ($24/3$) der Anzahl von Hinweisspalten in einem Physischen Cluster ist; wobei die Einrichtung Verteilungsmittel zum Verteilen von mit gleichen Nummern versehenen Symbolen der Hinweiswörter in Gruppen von so vielen Symbolen, wie Hinweisspalten in einem Physischen Cluster vorhanden sind, über gleichrangigen Aufzeichnungsrahmen verschiedener Physischer Sektoren des Physischen Clusters aufweist, in einem gestaffelten, jedoch ansonsten gleichmäßigen Verschachtelungsschema unter den verschiedenen Aufzeichnungsrahmen.

35. Einrichtung nach Anspruch 26, die Zuweisungsmittel zum Zuweisen von Symbolzeilen mit gerader Nummer zu einer ersten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren und von Symbolzeilen mit ungerader Nummer zu der zweiten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren aufweist.

36. Einrichtung nach Anspruch 26, die außerdem Rotationsmittel aufweist, um die Zeilen von Hinweisspaltensymbolen einer gestaffelten und systematischen Rotation unter den verschiedenen Hinweisspalten zu unterziehen.

37. Einrichtung nach Anspruch 26, die außerdem Adresszuweisungsmittel aufweist, um Hinweiswörtern sowohl logische als auch physische Adressmittel zuzuweisen, die zu dem jeweiligen Physischen Cluster gehören.

38. Einrichtung zur Decodierung von Mehrwortinformation, welche auf Multibit-Symbolen beruht, die in relativer Nachbarschaft bezüglich eines Mediums angeordnet sind, durch Bewirken einer wortweisen Entschachtelung und wortweise Fehlerschutzcodierungs-Fähigkeiten, und indem so Fehlerlokalisierungs-Hinweise über Mehrwortgruppen hinweg geliefert werden, dadurch gekennzeichnet, dass sie dafür vorgesehen ist, solche Hinweise sowohl aus Hinweiswörtern mit hoher Schutzfähigkeit, welche unter Hinweisspalten verschachtelt sind, als auch aus Synchronisationsspalten, die aus Synchronisationsbitgruppen gebildet werden, durch Zugriffsmittel zum Zugreifen auf die Synchronisationsspalten, wo die Hinweisspalten vergleichsweise seltener angeordnet sind, abzuleiten, wobei alle Hinweise auf Zielwörter mit niedriger Schutzfähigkeit gerichtet sind, welche in einer im Wesentlichen gleichförmigen Art und Weise unter Zielspalten verschachtelt sind, welche Spaltengruppen einheitlicher Größe zwischen periodischen Anordnungen von Hinweisspalten und Synchronisationsspalten bilden.

39. Einrichtung nach Anspruch 38, die außerdem Zugriffsmittel zum Zugreifen auf die Information entsprechend physischen Cluster einheitlicher Größe aufweist, welche jeweils eine einzige Synchronisationsspalte und eine ungerade Anzahl von Hinweisspalten aufweisen.

40. Einrichtung nach Anspruch 38, die für ein Ableiten von Benutzerdaten ausschließlich von Zielspalten und von Systemdaten wenigstens im Wesentlichen ausschließlich von Hinweisspalten vorgesehen ist.

41. Einrichtung nach Anspruch 38, die außerdem Zugriffsmittel aufweist, um auf einer niedrigsten Ebene innerhalb eines Datenrahmens auf eine Mehrsymbol-, jedoch bitorganisierte Fehlerdetektions-Bitgruppe zuzugreifen, um daraus ein Fehlerdetektionssignal abzuleiten.

42. Einrichtung nach Anspruch 41, wobei die Zugriffsmittel dafür vorgesehen sind, auf einer nächsthöheren Ebene innerhalb eines Datensektors auf mehrere jeweilige Datenrahmen, die über mehrere Zielwörter verteilt sind, durch Auswertung hinzugefügter Reed-Solomon-Redundanz zuzugreifen.

43. Einrichtung nach Anspruch 38, die Korrektormittel aufweist, um dem Verschachteln ein zeilenweises inkrementelles Zurückrotieren von Zielsymbolen innerhalb ihres Clusters zu überlagern.

44. Einrichtung nach Anspruch 38, die außerdem Schnittstellenmittel zur Ankopplung an ein optisches Speichermedium aufweist.

45. Einrichtung nach Anspruch 38, wobei die Decodierungsmittel dazu vorgesehen sind, alle Hinweiswörter und alle Zielwörter durch eine einheitliche Menge an Redundanz zu decodieren, aus Zielwörtern jedoch eine größere Menge an Datensymbolen abzuleiten als aus Hinweiswörtern.

46. Einrichtung nach Anspruch 38, wobei die Anzahl der Hinweiswörter eines umfassenden Physischen Speicher-Clusters ein Vielfaches ($24/3$) der Anzahl von Hinweisspalten in dem Physischen Cluster ist; und wobei sie dafür vorgesehen ist, mit gleichen Nummern versehene Symbole der Hinweiswörter als in Gruppen von so vielen Symbolen verteilt, wie Hinweisspalten in einem Physischen Cluster vorhanden sind, aus gleichrangigen Aufzeichnungsrahmen verschiedener Physischer Sektoren des Physischen Clusters abzuleiten, in einem gestaffelten, jedoch ansonsten gleichmäßigen Verschachtelungsschema unter den verschiedenen Aufzeichnungsrahmen.

47. Einrichtung nach Anspruch 46, die außerdem dafür vorgesehen ist, Symbolzeilen mit gerader Nummer aus einer ersten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren und Symbolzeilen mit

ungerader Nummer aus der zweiten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren abzuleiten.

48. Einrichtung nach Anspruch 46, die dafür vorgesehen ist, die Zeilen von Hinweisspaltensymbolen durch gestaffeltes und systematisches Zurückrotieren über die verschiedenen Hinweisspalten abzuleiten.

49. Einrichtung nach Anspruch 38, die dafür vorgesehen ist, sowohl logische als auch physische Adressdaten aus den Hinweisspalten als zu dem jeweiligen Physischen Cluster gehörend abzuleiten.

50. Einheitlicher Speicherträger, der durch ein Verfahren nach Anspruch 1 hergestellt wird und Mehrwortinformation speichert, welche auf Multibit-Symbolen beruht, die in relativer Nachbarschaft auf ihm angeordnet sind, mit wortweisen Verschachtelungs- und wortweisen Fehlerschutzcode-Fähigkeiten, welche Fehlerlokalisierungs-Hinweise über Mehrwortgruppen hinweg liefern, gekennzeichnet durch Hinweiswörter mit hoher Schutzfähigkeit, welche unter Hinweisspalten verschachtelt sind, und durch Synchronisationsspalten, die aus Synchronisationsbitgruppen gebildet werden, wobei die Synchronisationsspalten dort positioniert sind, wo die Hinweisspalten vergleichsweise seltener angeordnet sind, zum Erzeugen solcher Hinweise sowohl in Hinweiswörtern mit hoher Schutzfähigkeit als auch in Synchronisationsspalten, wobei alle Hinweise auf Zielwörter mit niedriger Schutzfähigkeit gerichtet sind, welche in einer im Wesentlichen gleichförmigen Art und Weise unter Zielspalten verschachtelt sind, welche Spaltengruppen einheitlicher Größe zwischen periodischen Anordnungen von Hinweisspalten und Synchronisationsspalten bilden.

51. Träger nach Anspruch 50, der die Information in physischen Cluster einheitlicher Größe angeordnet aufweist, welche jeweils eine einzige Synchronisationsspalte und eine ungerade Anzahl von Hinweisspalten aufweisen.

52. Träger nach Anspruch 50, der Benutzerdaten ausschließlich in Zielspalten enthalten und Systemdaten wenigstens im Wesentlichen ausschließlich in Hinweisspalten enthalten aufweist.

53. Träger nach Anspruch 50, wobei auf einer niedrigsten Ebene ein Mehrsymbol-Datenrahmen von Zielsymbolen außerdem eine Mehrsymbol-, jedoch bitorganisierte Fehlerdetektions-Bitgruppe enthält.

54. Träger nach Anspruch 53, wobei auf einer nächsthöheren Ebene ein Datensektor mehrere Datenrahmen enthält, die durch hinzugefügte Reed-Solomon-Redundanz über mehrere Zielwörter verteilt sind.

55. Träger nach Anspruch 54, bei dem die Verschachtelung einer zeilenweisen inkrementellen Rotation von Zielsymbolen innerhalb ihres Clusters überlagert ist.

56. Träger nach Anspruch 50, und basierend auf einem optischen Speichermedium.

57. Träger nach Anspruch 50, wobei alle Hinweiswörter und Zielwörter eine einheitliche Menge an Redundanz tragen, Zielwörter jedoch mehr Datensymbole als Hinweiswörter aufweisen.

58. Träger nach Anspruch 50, wobei die Anzahl der Hinweiswörter eines umfassenden Physischen Speicher-Clusters ein Vielfaches ($24/3$) der Anzahl von Hinweisspalten in dem Physischen Cluster ist; wobei mit gleichen Nummern versehene Symbole der Hinweiswörter in Gruppen von so vielen Symbolen, wie Hinweisspalten in einem Physischen Cluster vorhanden sind, über gleichrangigen Aufzeichnungsrahmen verschiedener Physischer Sektoren des Physischen Clusters verteilt sind, in einem gestaffelten, jedoch ansonsten gleichmäßigen Verschachtelungsschema unter den verschiedenen Aufzeichnungsrahmen.

59. Träger nach Anspruch 58, wobei Symbolzeilen mit gerader Nummer einer ersten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren zugewiesen sind und Symbolzeilen mit ungerader Nummer der zweiten zusammenhängenden Gruppe einer Hälfte der Physischen Sektoren zugewiesen sind.

60. Träger nach Anspruch 58, wobei die Zeilen von Hinweisspaltensymbolen als einer gestaffelten und systematischen Rotation unter den verschiedenen Hinweisspalten unterzogen angeordnet sind.

61. Träger nach Anspruch 58, wobei Hinweisspalten sowohl logische als auch physische Adressdaten enthalten, die zu dem jeweiligen Physischen Cluster gehören.

Es folgen 16 Blatt Zeichnungen

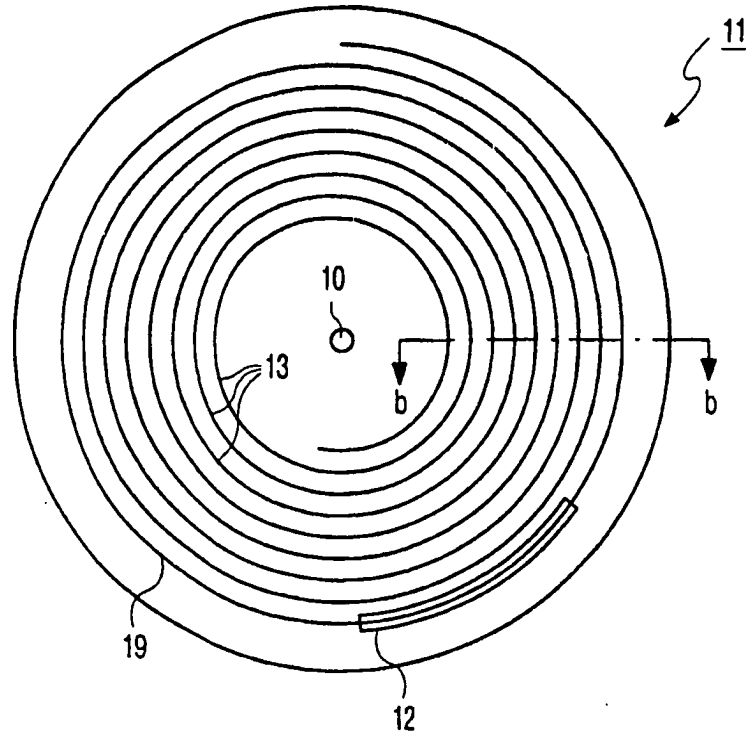


FIG. 1a

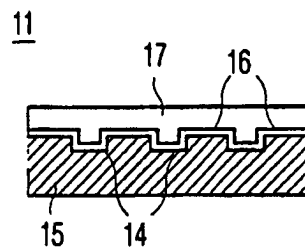


FIG. 1b

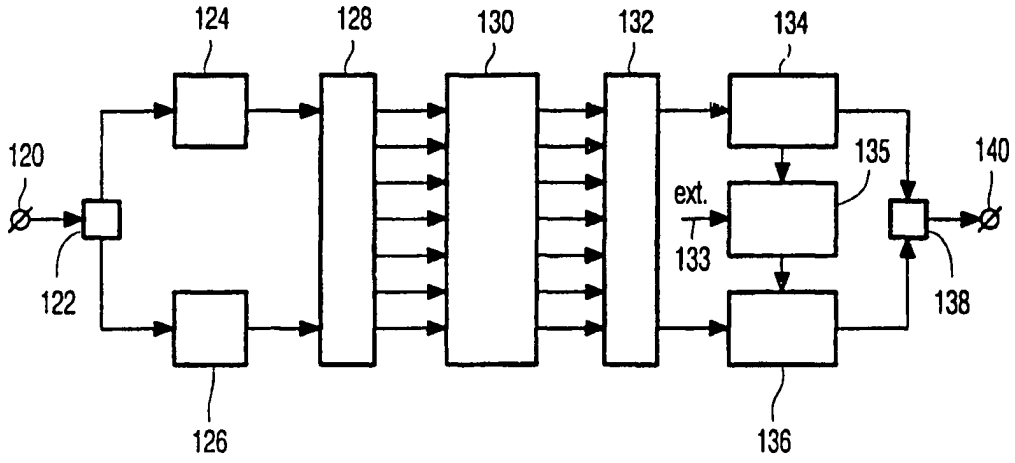


FIG. 4

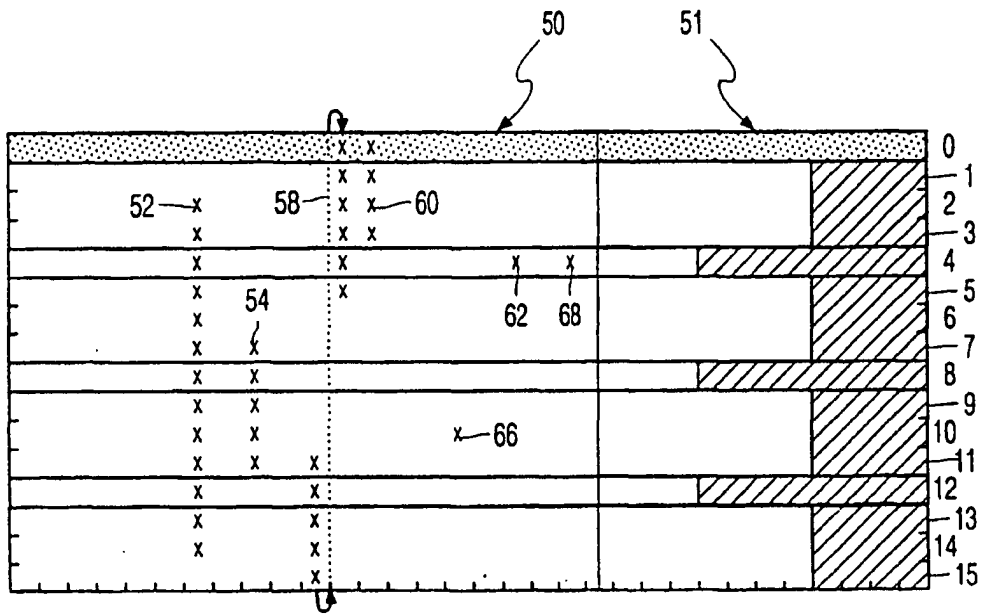


FIG. 5

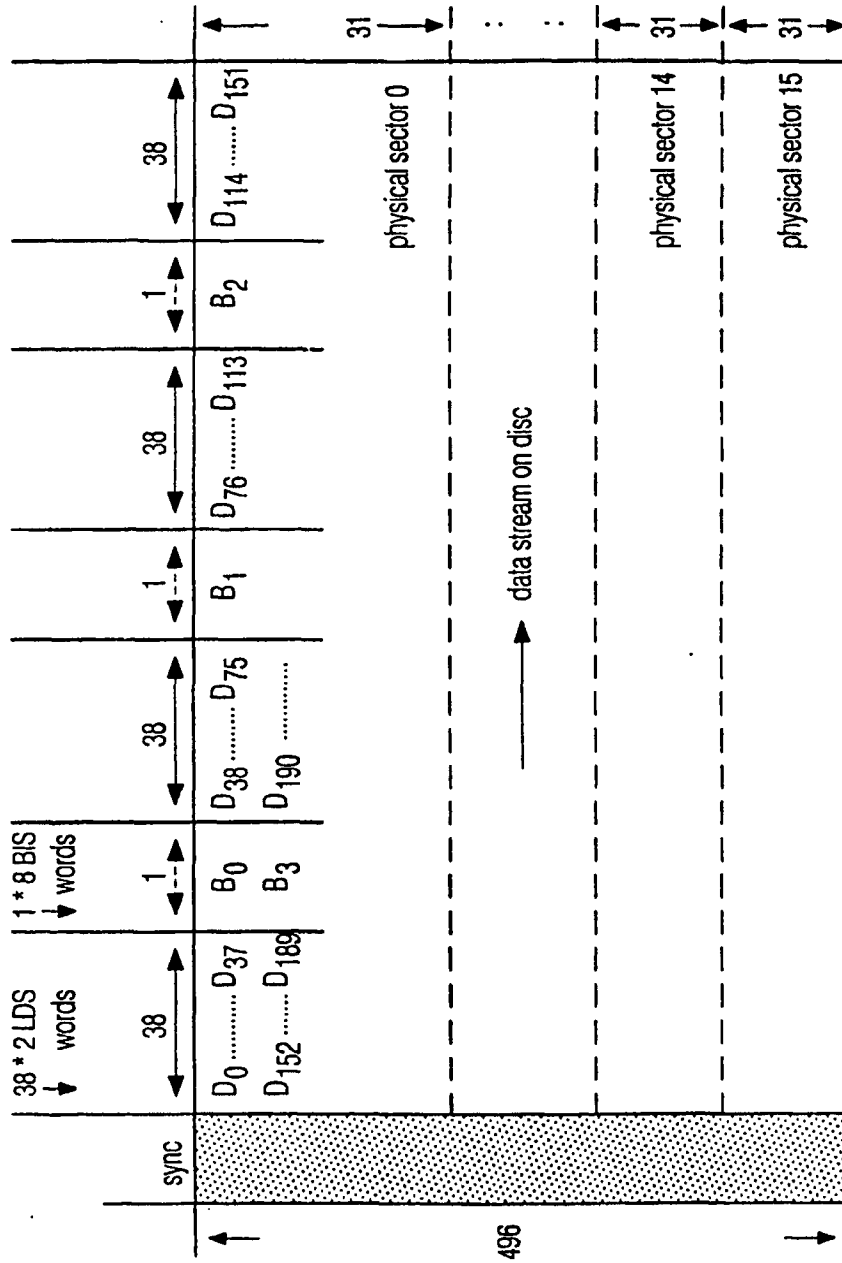


FIG. 6

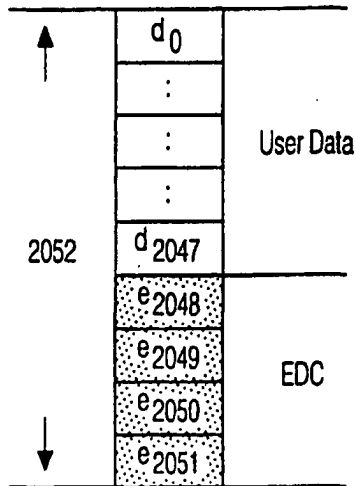


FIG. 7

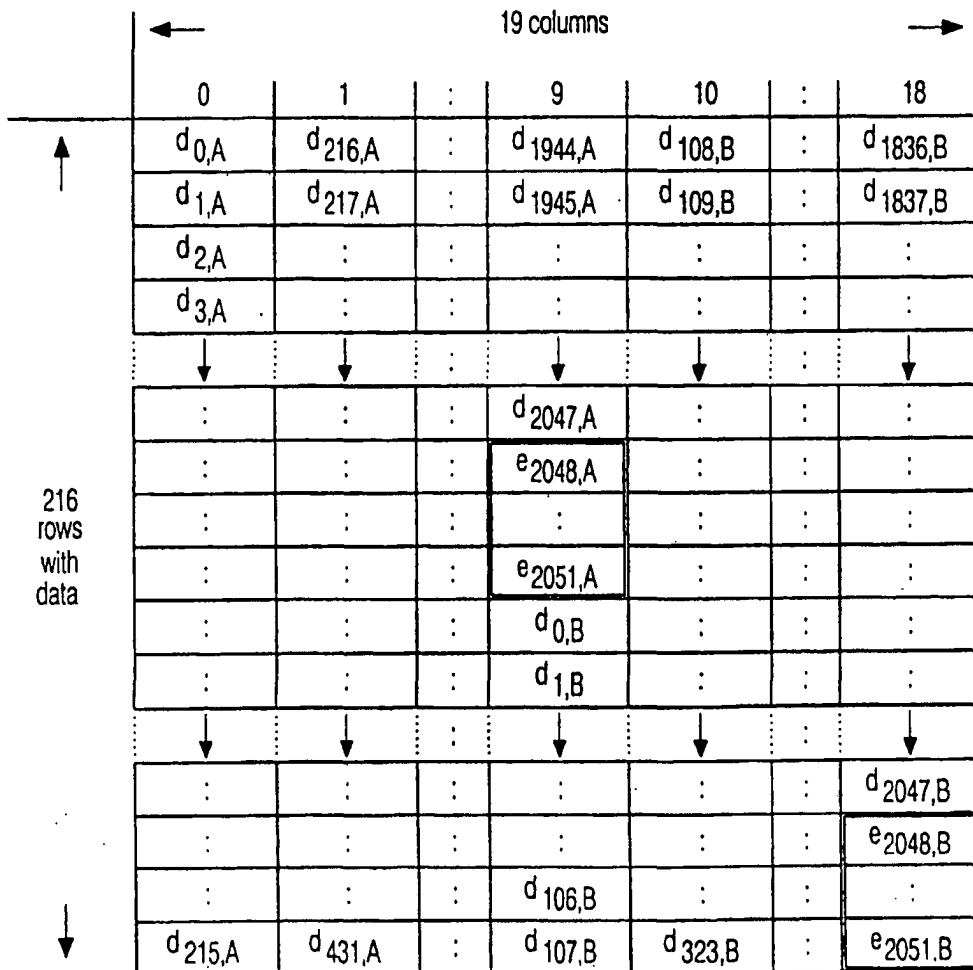


FIG. 8

		← 19 code words →						
		0	1	:	L	:	17	18
↑ 216 rows with data		$d_{0,0}$	$d_{1,0}$:	$d_{L,0}$:	$d_{17,0}$	$d_{18,0}$
		$d_{0,1}$	$d_{1,1}$:	$d_{L,1}$:	$d_{17,1}$	$d_{18,1}$
		$d_{0,2}$:	:	:	:	:	:
		$d_{0,3}$:	:	:	:	:	:
		:	:	:	:	:	:	:
		:	:	:	:	:	:	:
		:	:	:	:	:	:	:
		$d_{0,215}$	$d_{1,215}$:	$d_{L,215}$:	$d_{17,215}$	$d_{18,215}$
↑ 32 rows with parity		$P_{0,216}$	$P_{1,216}$:	$P_{L,216}$:	$P_{17,216}$	$P_{18,216}$
		$P_{0,247}$	$P_{1,247}$:	$P_{L,247}$:	$P_{17,247}$	$P_{18,247}$

FIG. 9

16 ECS Sectors
= 152 columns

	←	0	1	:	:	:	:	:	:	:	151	→	
↑	d _{0,0,0}	d _{16,0,N-1}	d _{1,0,N}	:	:	:	:	:	:	:	d _{17,0,N}	:	d _{17,0,15}
	d _{1,0,0}	d _{17,0,N-1}	d _{2,0,N}	:	:	:	:	:	:	:	d _{18,0,N}	:	d _{18,0,15}
	d _{0,1,0}	d _{16,1,N-1}	d _{1,1,N}	:	:	:	:	:	:	:	d _{17,1,N}	:	d _{17,1,15}
	d _{1,1,0}	d _{17,1,N-1}	d _{2,1,N}	:	:	:	:	:	:	:	d _{18,1,N}	:	d _{18,1,15}
	:	:	:	:	:	:	:	:	:	:	:	:	:
	:	:	:	:	:	:	:	:	:	:	:	:	:
↓	d _{0,215,0}	d _{16,215,N-1}	d _{1,215,N}	:	:	:	:	:	:	:	d _{17,215,N}	:	d _{17,215,15}
	d _{1,215,0}	d _{17,215,N-1}	d _{2,215,N}	:	:	:	:	:	:	:	d _{18,215,N}	:	d _{18,215,15}
↑	P _{0,216,0}	P _{16,216,N-1}	P _{1,216,N}	:	:	:	:	:	:	:	P _{17,216,N}	:	P _{17,216,15}
	P _{1,216,0}	P _{17,216,N-1}	P _{2,216,N}	:	:	:	:	:	:	:	P _{18,216,N}	:	P _{18,216,15}
	P _{0,217,0}	P _{16,217,N-1}	P _{1,217,N}	:	:	:	:	:	:	:	P _{17,217,N}	:	P _{17,217,15}
	P _{1,217,0}	P _{17,217,N-1}	P _{2,217,N}	:	:	:	:	:	:	:	P _{18,217,N}	:	P _{18,217,15}
	:	:	:	:	:	:	:	:	:	:	:	:	:
	:	:	:	:	:	:	:	:	:	:	:	:	:
↓	P _{0,247,0}	P _{16,247,N-1}	P _{1,247,N}	:	:	:	:	:	:	:	P _{17,247,N}	:	P _{17,247,15}
	P _{1,247,0}	P _{17,247,N-1}	P _{2,247,N}	:	:	:	:	:	:	:	P _{18,247,N}	:	P _{18,247,15}

FIG. 10

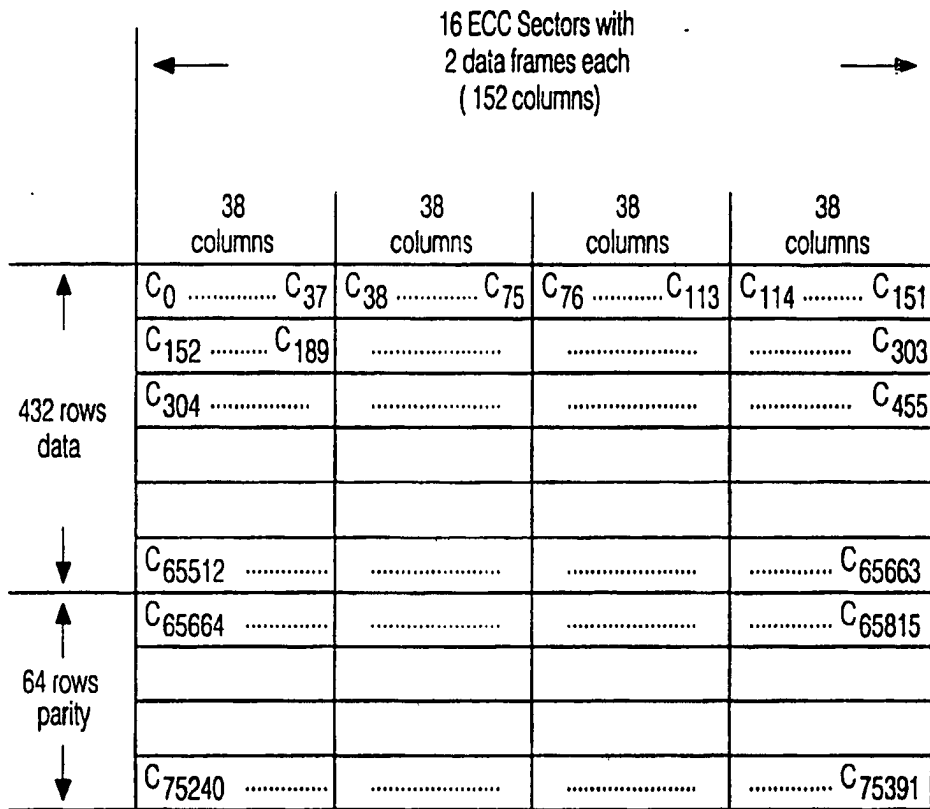


FIG. 11

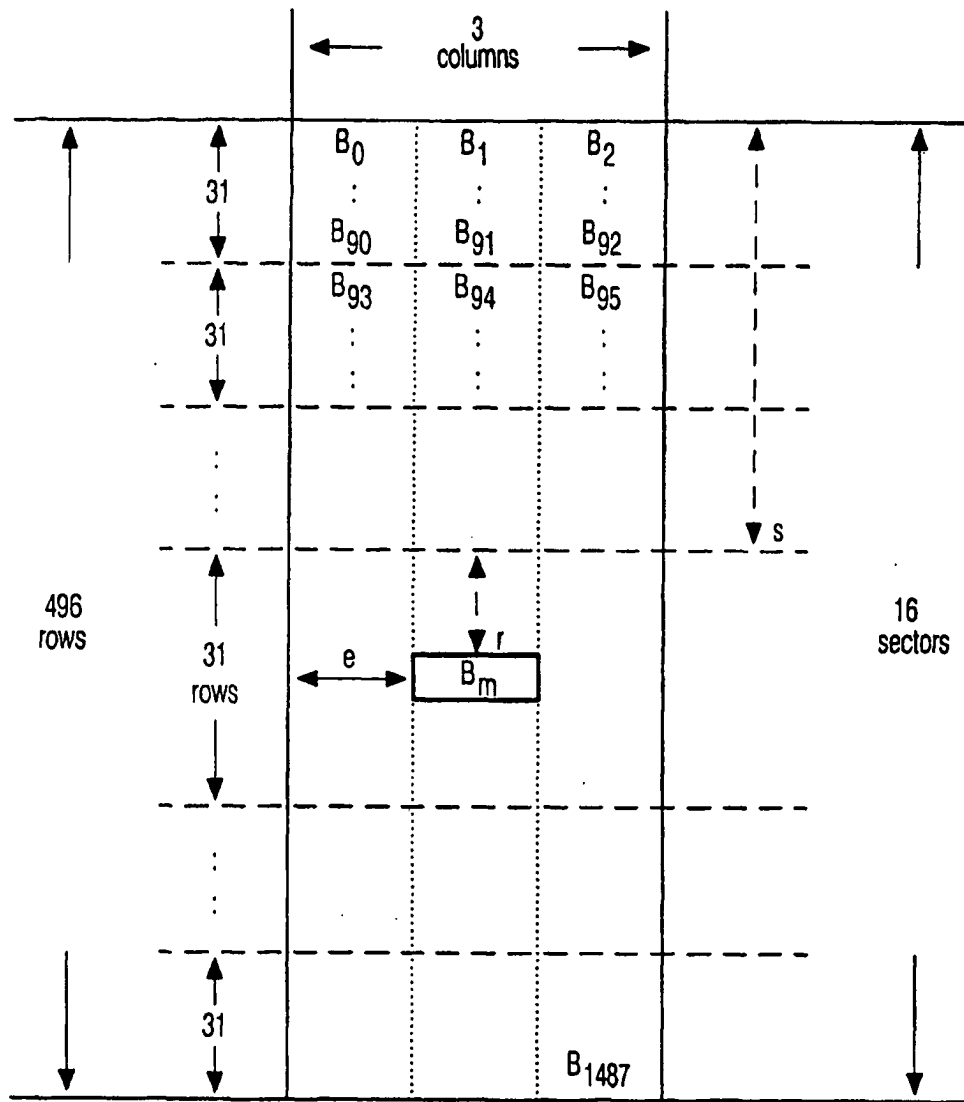


FIG. 14A

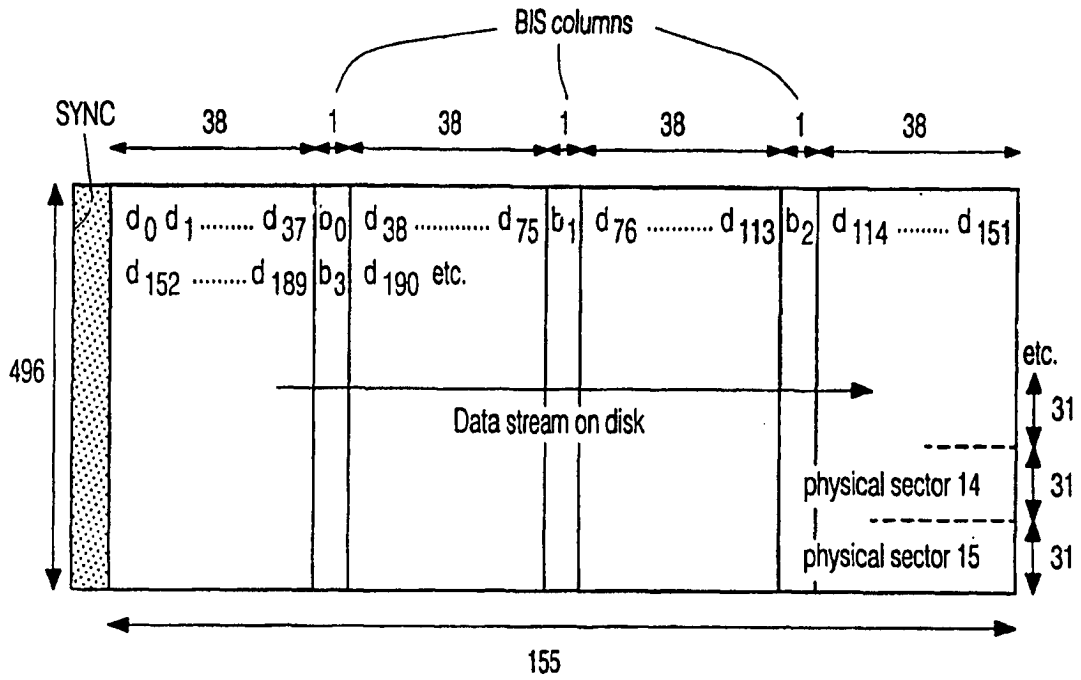


FIG. 15

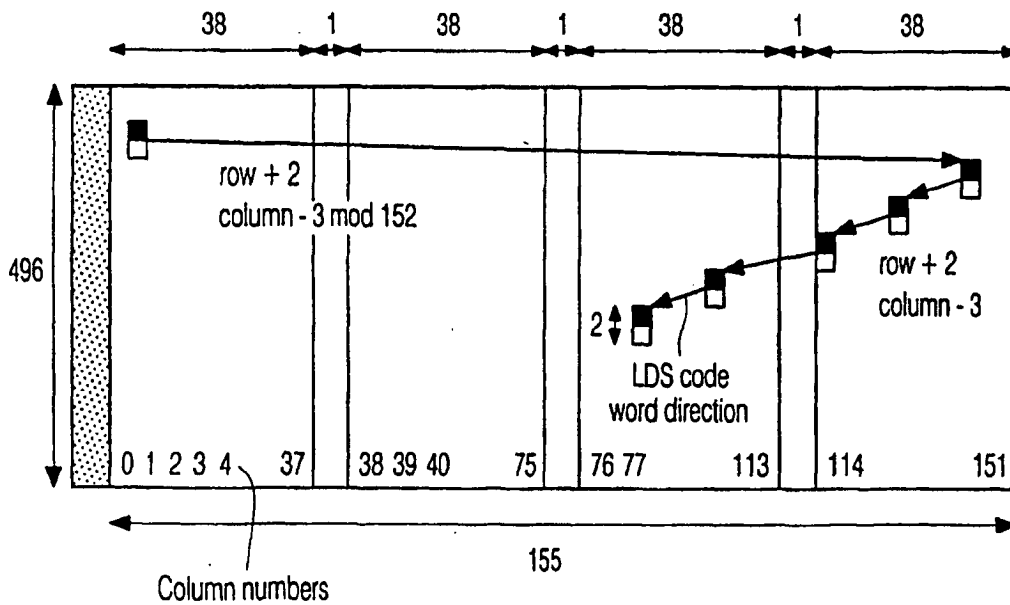


FIG. 16

sector s	row r	byte number N,C from BIS Block			shift right (= mod(r,3))	filling in upward direction
		0	column e 1	2		
0	0	0,0	0,1	0,2	0	start of Block row N = 0
	1	2,5	2,3	2,4	1	↑ continuation of Block row N = 2
	2	4,7	4,8	4,6	2	
	3	6,9	6,10	6,11	0	
	:					
	7	14,23	14,21	14,22	1	
	8	16,1	16,2	16,0	2	start of Block row N = 16
	:					
1	30	60,18	60,19	60,20	0	
	0	0,21	0,22	0,23	0	end of Block row N = 0
	1	2,2	2,0	2,1	1	start of Block row N = 2
	2	4,4	4,5	4,3	2	
	3	6,6	6,7	6,8	0	
2	:					
	0	0,18	0,19	0,20	0	
	1	2,23	2,21	2,22	1	end of Block row N = 2
	2	4,1	4,2	4,0	2	start of Block row N = 4
	3	6,3	6,4	6,5	0	
3	:					
	0	0,15	0,16	0,17	0	
	1	2,20	2,18	2,19	1	
	2	4,22	4,23	4,21	2	
	3	6,0	6,1	6,2	0	start of Block row N = 6
4	:					
	0	0,12	0,13	0,14	0	
	1	2,17	2,15	2,16	1	
	2					
	:					

FIG. 17(1)

5	0	0,9	0,10	0,11	0	
	1	2,14	2,12	2,13	1	
	2					
	:					
6	0	0,6	0,7	0,8	0	
	1	2,11	2,9	2,10	1	
	2	4,13	4,14	4,12	2	
	:					
7	0	0,3	0,4	0,5	0	↑ continuation of Block row N = 0
	1	2,8	2,6	2,7	1	↑ continuation of Block row N = 2
	2	4,10	4,11	4,9	2	
	:					
	7	14,2	14,0	14,1	1	start of Block row N = 14
	:					
	30	60,21	60,22	60,23	0	end of Block row N = 60

FIG. 17(2)

sector s	row r	byte number N,C from BIS Block			shift right (= mod(r,3))	filling in upward direction
		0	column e 1	2		
8	0	1,0	1,1	1,2	0	start of Block row N = 1
	1	3,5	3,3	3,4	1	
	2	5,7	5,8	5,6	2	
	3	7,9	7,10	7,11	0	
	:					
	8	17,1	17,2	17,0	2	start of Block row N = 17
	:					
	30	61,18	61,19	61,20		
9	0	1,21	1,22	1,23		end of Block row N = 1
10	0	1,18	1,19	1,20		
11	0	1,15	1,16	1,17		
12	0	1,12	1,13	1,14		
13	0	1,9	1,10	1,11		
14	0	1,6	1,7	1,8		
15	0	1,3	1,4	1,5	0	↑ continuation of Block row N = 1
	1	3,8	3,6	3,7	1	
	2	5,10	5,11	5,9	2	
	:					
	7	15,2	15,0	15,1	1	start of Block row N = 15
	:					
	30	61,21	61,22	61,23	0	end of Block row N = 61

FIG. 18

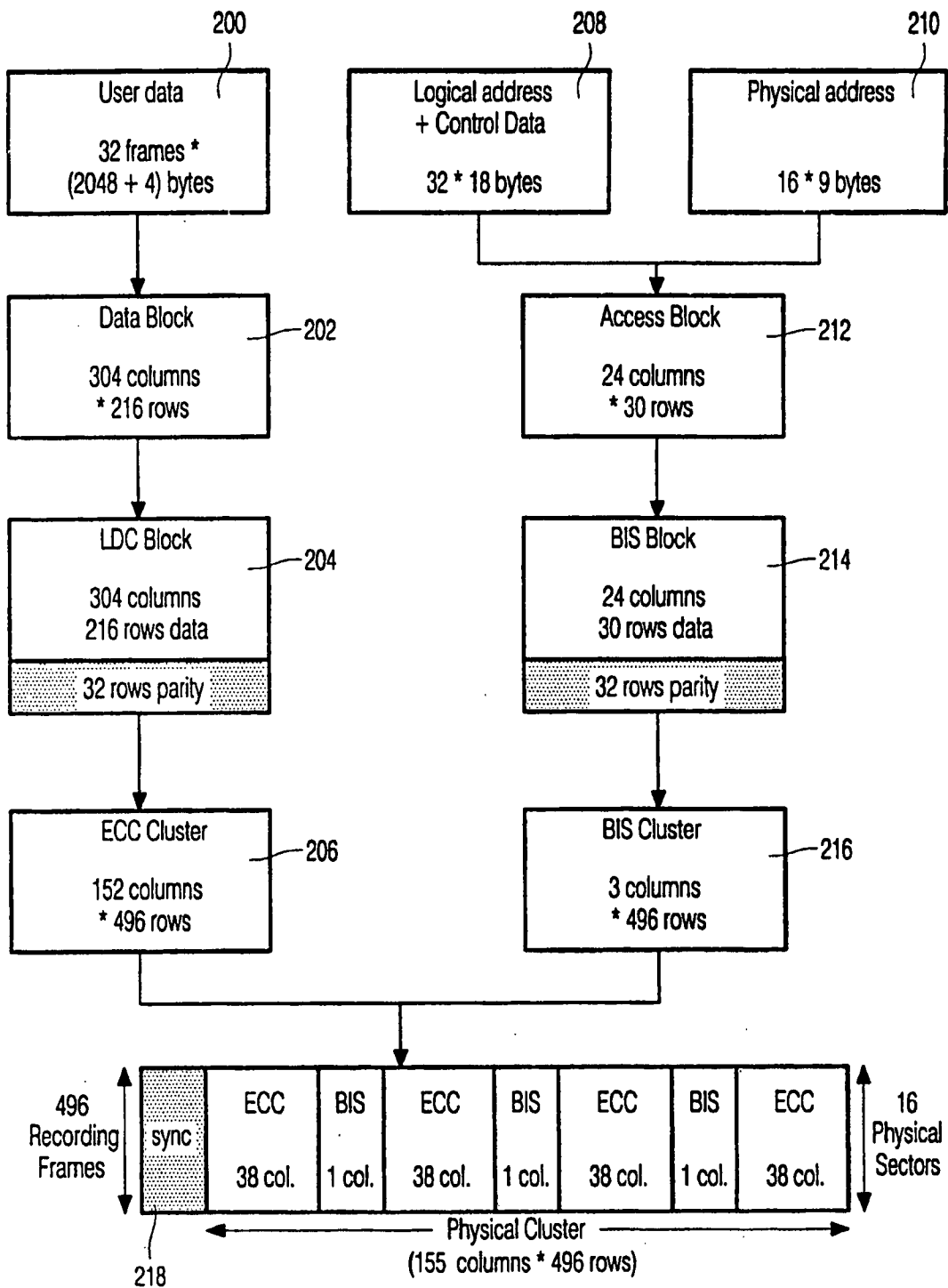


FIG. 19