(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 698 38 869 T2** 2008.12.04

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 944 249 B1**
(21) Deutsches Aktenzeichen: **698 38 869.0**
(86) PCT-Aktenzeichen: **PCT/JP98/04497**
(96) Europäisches Aktenzeichen: **98 945 621.5**
(87) PCT-Veröffentlichungs-Nr.: **WO 1999/018720**
(86) PCT-Anmeldetag: **05.10.1998**
(87) Veröffentlichungstag der PCT-Anmeldung: **15.04.1999**
(97) Erstveröffentlichung durch das EPA: **22.09.1999**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **19.12.2007**
(47) Veröffentlichungstag im Patentblatt: **04.12.2008**

(51) Int Cl.[8]: ***H04N 5/268*** (2006.01)
H04N 7/24 (2006.01)

(30) Unionspriorität:
27127597 03.10.1997 JP

(84) Benannte Vertragsstaaten:
BE, DE, ES, FR, GB, IT, NL

(73) Patentinhaber:
Sony Corp., Tokio/Tokyo, JP

(74) Vertreter:
Mitscherlich & Partner, Patent- und Rechtsanwälte, 80331 München

(72) Erfinder:
TAHARA, Katsumi, Tokyo 141-0001, JP; YASUDA, Mikita, Tokyo 141-0001, JP; NEGISHI, Shinji, Tokyo 141-0001, JP

(54) Bezeichnung: **Vorrichtung und Verfahren zum Spleißen von codierten Datenströmen sowie Vorrichtung und Verfahren zur Erzeugung von codierten Datenströmen**

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

Technisches Gebiet

[0001] Die Erfindung bezieht sich auf eine Codierstrom-Spleißeinrichtung, ein Codierstrom-Spleißverfahren, eine Codierstrom-Erzeugungseinrichtung und ein Codierstrom-Erzeugungsverfahren, welche bei einem digitalen Rundfunksystem verwendet werden. Insbesondere bezieht sie sich auf eine Codierstrom-Spleißeinrichtung, ein Codierstrom-Spleißverfahren, eine Codierstrom-Erzeugungseinrichtung und ein Codierstrom-Erzeugungsverfahren, welche eingerichtet sind, einen saumlosen Spleißstrom zu erzeugen, indem zwei codierte Ströme bei einem Strompegel gespleißt werden.

Technischer Hintergrund

[0002] Fig. 1 zeigt ein aktuelles Fernsehrundfunksystem. Bei dem aktuellen Fernsehrundfunksystem weisen Rundfunkstationen zum Verteilen von Fernsehprogrammen an jeden Haushalt eine Schlüsselstation (oder Hauptstation) S_K auf, um Fernsehprogramme nationalweit zu erzeugen, und mehrere lokale (örtliche) Stationen (Zweige) S_A, S_B und S_C, welche mit der Schlüsselstation verbunden sind, um spezifische lokale Fernsehprogramme zu erzeugen. Die Schlüsselstation S_K ist eine Sendestation zum Erzeugen gemeinsamer nationaler Fernsehprogramme und zum Übertragen der erzeugten Fernsehprogramme an die lokalen Stationen. Die lokalen Stationen sind Rundfunkstationen, um die ursprünglichen Fernsehprogramme, welche von der Schlüsselstation gesendet werden, über eine Zwischenstationsübertragung an Haushalte der örtlichen Bereiche, und die Fernsehprogramme, welche durch Editieren eines Teils der ursprünglichen Fernsehprogramme erzeugt werden, in spezifischen örtlichen Versionen zu verteilen. Wie beispielsweise in Fig. 1 gezeigt ist, ist die örtliche Station S_A eine Station zum Erzeugen von Fernsehprogrammen, welche zu Haushalten in einem Rundfunkbereich E_A übertragen werden. Die örtliche Station S_B ist eine Station, um Fernsehprogramme, welche an Haushalte in einem Rundfunkbereich E_B übertragen werden, zu erzeugen. Die örtliche Station S_C ist eine Station, um Fernsehprogramme, welche an Haushalte in einem Rundfunkbereich E_C übertragen werden, zu erzeugen. Die Editierverarbeitung, welche in jeder örtlichen Station ausgeführt wird, ist beispielsweise die Verarbeitung, ein spezifisches lokales Wettervorhersageprogramm in ein Nachrichtenprogramm einzufügen, welches von der Schlüsselstation übertragen wird, oder eine Verarbeitung, um eine örtliche Werbung in ein Programm, beispielsweise in einen Film oder ein Drama einzufügen.

[0003] Fig. 2A bis Fig. 2C zeigen die Editierverarbeitung in jeder örtlichen Station. Fig. 2A zeigt ein ursprüngliches Fernsehprogramm PG_{OLD}, welches in der Schlüsselstation erzeugt wird. Fig. 2B zeigt ein Ersatzfernsehprogramm PG_{NEW} für örtliche Zuschauer, welches in einer örtlichen Station erzeugt wird. Fig. 2C zeigt ein Fernsehprogramm PG_{EDIT}, welches in einer örtlichen Station editiert wird. Das Editierverarbeitungsbeispiel, welches in Fig. 2A bis Fig. 2C gezeigt ist, ist ein Beispiel einer Editierverarbeitung zum Ersetzen einer Werbung CM1, eines Programms 3 und einer Werbung CM3 der ursprünglichen Fernsehprogramme, welche von der Schlüsselstation übertragen werden, durch eine Werbung CM1', ein Programm 3' und eine Werbung CM3', welche in der örtlichen Station für örtliche Zuschauer erzeugt werden. Als Ergebnis dieser Editierverarbeitung in der örtlichen Station werden die Fernsehprogramme für örtliche Zuschauer erzeugt, in denen die Fernsehprogramme, welche in der Schlüsselstation erzeugt werden (d. h., ein Programm 1, ein Programm 2, eine CM2 und ein Programm 4) und die Fernsehprogramme, welche in der örtlichen Station erzeugt werden (d. h., die Werbung CM1', das Programm 3' und die CM3') gemeinsam existieren.

[0004] Da das aktuelle Fernsehrundfunksystem analogen Rundfunk verwendet, um analoge Basisband-Fernsehsignale an jeden Haushalt zu verteilen, werden seit einiger Zeit Versuche durchgeführt, das analoge Rundfunksystem durch ein Rundfunksystem einer nächsten Generation zu ersetzen, wobei ein Digitalverfahren verwendet wird. Das digitale Rundfunksystem ist ein System, um Videodaten und Audiodaten unter Verwendung eines Kompressionscodierverfahrens, beispielsweise MPEG2 (Moving Picture Experts Group Phase 2) kompressions-zu-codieren, und um die codierten Ströme an jeden Haushalt und an andere Stationen über Bodenwellen oder Satellitenwellen zu übertragen. Insbesondere ist unter den Rundfunkverfahren, welche als digitales Rundfunksystem vorgeschlagen wurden, der DVB-Standard (digitaler Videofernsehstandard), der als Rundfunksystem der nächsten Generation in Europa vorgeschlagen ist, das maßgeblichste Verfahren. Dieser DVB-Standard ist zum De-Fakto-Standard geworfen.

[0005] Mit Hilfe von Fig. 3 wird nun eine typisches digitales Übertragungssystem zum Übertragen eines Programms, welches Videodaten und Audiodaten umfasst, von Seiten eines Übertragungssystems zur Seite eines Empfangssystems unter Verwendung des MPEG-Standards beschrieben.

[0006] Bei dem typischen digitalen Übertragungssystem hat die Seite des Übertragungssystems **10** einen MPEG-Videocodierer **11**, einen MPEG-Audiocodierer **12** und einen Multiplexer **13**. Ein empfangsseitiges System **20** hat einen Demultiplexer **21**, einen MPEG-Videodecoder **22** und einen MPEG-Audiodecoder **23**.

[0007] Der MPEG-Videocodierer **11** codiert Basisband-Quellenvideodaten V auf Basis des MPEG-Standards und gibt den Codierstrom als elementaren Videostrom ES aus. Der MPEG-Audiocodierer **12** codiert die Basisband-Quellenaudiodaten A auf Basis des MPEG-Standards und gibt den Codierstrom als elementaren Audiostrom ES aus. Der Multiplexer **13** empfangt den elementaren Videostrom vom MPEG-Videocodierer **11** und den elementaren Audiostrom vom MPEG-Audiocodierer **12**. Der Multiplexer **13** setzt dann die Ströme in die Form von Transportstrompaketen um, um ein Transportstrompaket, welches den elementaren Videostrom aufweist, und ein Transportstrompaket zu erzeugen, welches den elementaren Audiostrom aufweist. Der Multiplexer **13** multiplext die Transportstrompakete so, dass das Transportstrompaket, welches den elementaren Videostrom aufweist, und das Transportstrompaket, welches den elementaren Audiostrom aufweist, gemeinsam existieren, um somit einen Transportstrom zu erzeugen, der zum Empfangssystem **20** übertragen wird.

[0008] Der Demultiplexer **21** empfangt den Transportstrom, der über eine Übertragungsleitung übertragen wird, und demultiplext den Transportstrom in das Transportstrompaket, welches den elementaren Videostrom aufweist, und das Transportstrompaket, welches den elementaren Audiostrom aufweist. Der Demultiplexer **21** erzeugt dann den elementaren Videostrom für das Transportstrompaket, welches den elementaren Videostrom aufweist, und erzeugt den elementaren Audiostrom vom Transportstrompaket, der den elementaren Audiostrom aufweist. Der MPEG-Videodecoder **22** empfängt den elementaren Videostrom vom Demultiplexer **21** und decodiert diesen elementaren Videostrom auf Basis des MPEG-Standards, wodurch die Basisband-Videodaten V erzeugt werden. Der MPEG-Audiodecoder **23** empfangt den elementaren Audiostrom vom Demultiplexer **21** und decodiert diesen elementaren Audiostrom auf Basis des MPEG-Standards, wodurch die Basisband-Audiodaten A erzeugt werden.

[0009] In dem Fall, wo das herkömmliche analoge Rundfunksystem durch ein digitales Rundfunksystem unter Verwendung eines Verfahrens des digitalen Übertragungssystems ersetzt werden soll, sind Videodaten der Fernsehprogramme, welche zur örtlichen Station von der Schlüsselstation übertragen werden, ein Codierstrom, der auf Basis des MPEG2-Standards kompressions-codiert ist. Um die Editierverarbeitung in der örtlichen Station auszuführen, um einen Teil des ursprünglichen Codierstroms, der von der Schlüsselstation übertragen wird, durch einen Codierstrom zu ersetzen, der in der örtlichen Station erzeugt wird, muss daher der Codierstrom einmal decodiert werden, um die Basisband-Videodaten wiederherzustellen, bevor die Editierverarbeitung ausgeführt wird. Der Grund dafür ist folgender. Das heißt, da die Vorhersagerichtung jedes Bilds, welches im Codierstrom enthalten ist, gemäß dem MPEG-Standard mit der Vorhersagerichtung der vorhergehenden und nachfolgenden Bilder in Bezug steht, kann ein Codierstrom nicht an einer beliebigen Position des Stroms verbunden werden. Wenn zwei Codierströme zwangsweise verbunden werden, werden die Daten am Saum der Codierströme unstetig und können nicht genau decodiert werden.

[0010] Um die Editierverarbeitung, wie mit Hilfe von Fig. 2A bis Fig. 2C beschrieben wurde, zu realisieren, ist es daher notwendig, eine Decodierverarbeitung, um einmal sowohl einen ursprünglichen Codierstrom, der von der Schlüsselstation geliefert wird, als auch einen Codierstrom, der für örtliche Zuschauer erzeugt wird, zu decodieren, um die Basisband-Videosignale wiederherzustellen, die Editierverarbeitung, um zwei Basisband-Videodaten zu editieren, um Videodaten zu erzeugen, welche zum Senden editiert werden, und die Codierverarbeitung auszuführen, um wiederum die editierten Videodaten zu codieren, um Codiervideoströme zu erzeugen. Da jedoch die Codier-/Decodierverarbeitung auf Basis des MPEG-Standards keine 100%ige umkehrbare Codier-/Decodierverarbeitung ist, besteht ein Problem dahingehend, dass die Bildqualität verschlechtert wird, wenn die Decodierverarbeitung und die Codierverarbeitung wiederholt werden.

[0011] Somit ist seit einiger Zeit ein Verfahren erforderlich, welches das Editieren im Zustand codierter Ströme ermöglicht, ohne zugeführte Codierströme zu decodieren. Das Verfahren zum Verbinden von zwei unterschiedlichen Codierbitströmen bei dem Pegel codierter Bitströme, um somit verbundene Bitströme zu erzeugen, wird als "Spleißen" bezeichnet. Kurz ausgedrückt bedeutet Spleißen Editierung und Verbindung mehrerer Ströme im Zustand codierter Ströme.

[0012] Die Realisierung dieser Spleißverarbeitung hat jedoch die folgenden beiden Probleme. Das erste Problem wird nun beschrieben.

[0013] Gemäß dem MPEG-Standard, der für den oben beschriebenen MPEG-Videocodierer **11** und den MPEG-Videodecoder **22** verwendet wird, wird ein bidirektionales Vorhersagecodiersystem als Codiersystem verwendet. Bei diesem bidirektionalen Vorhersagecodiersystem werden drei Codierarten, d. h., Intra-Rahmencodierung, Inter-Rahmen-Vorwärtsvorhersagecodierung und bidirektionale Vorhersagecodierung ausgeführt. Bilder, welche durch die jeweiligen Codierarten erlangt werden, werden als I-Bild (intra-codiertes Bild), P-Bild (vorhersage-codiertes Bild) und B-Bild (bidirektionales vorhersage-codiertes Bild) bezeichnet. Die I-, P- und B-Bilder werden passend kombiniert, um eine GOP (Bildergruppe) als eine Einheit für den Zufallszugriff zu bilden. Allge-

mein hat das I-Bild die größte Menge erzeugter Codes, das P-Bild hat die zweitgrößte Menge erzeugter Codes, das B-Bild hat die kleinste Menge erzeugter Codes.

[0014] Bei dem Codierverfahren, bei dem die Menge erzeugter Bits für jedes Bild, wie beim MPEG-Standard variiert, muss, um erlangte codierte Bitströme (anschließend einfach als Ströme bezeichnet) durch den Videodecoder genau zu decodieren, um ein Bild zu erlangen, die Datenbelegungsmenge in einem Eingangspuffer des Videodecoders **22** durch den Videocodierer **11** erfasst werden. Somit wird gemäß dem MPEG-Standard ein virtueller Puffer, der als "VBV-Puffer" (Video Buffering Verifier) bezeichnet wird, als ein Puffer angenommen, der dem Eingangspuffer des Videodecoders **22** entspricht, und es ist vorgeschrieben, dass der Videocodierer **11** die Codierverarbeitung durchführt, um keinen Zusammenbruch des VBV-Puffers zu bewirken, d. h., einen Überlauf oder Unterlauf. Beispielsweise wird die Kapazität des VBV-Puffers gemäß dem Standard von zu übertragenden Signale festgelegt. Im Fall von Standardvideosignalen des Hauptprofils und des Hauptpegels (MP@ML) hat der VBV-Puffer eine Kapazität von 1,75 Mbits. Der Videocodierer **11** steuert die Menge erzeugter Bits jedes Bilds, um somit keinen Überlauf oder Unterlauf des VBV-Puffers zu bewirken.

[0015] Der VBV-Puffer wird anschließend mit Hilfe von Fig. 4A bis Fig. 4C beschrieben.

[0016] Fig. 4A zeigt einen ursprünglichen Strom ST_{OLD}, welcher durch einen Videocodierer erlangt wird, indem ursprüngliche Fernsehprogramme, welche ein Programm 1 umfassen, und eine Werbung CM1, welche in der Schlüsselstation erzeugt werden, codiert werden, und die Ortskurve der Datenbelegungsmenge im VBV-Puffer, welche dem ursprünglichen Strom ST_{OLD} entspricht. Fig. 4B zeigt einen Ersatzstrom ST_{NEW}, der durch einen Videocodierer in einer örtlichen Station erlangt wird, indem eine Werbung CM1', welche für örtliche Zuschauer erzeugt wird, codiert wird, um den Teil der Werbung CMI der ursprünglichen Fernsehprogramme zu ersetzen, und die Ortskurve der Datenbelegungsmenge im VBV-Puffer, welche dem Ersatzstrom ST_{NEW} entspricht. In der folgenden Beschreibung wird, da ein Teil des Stroms, der durch Codieren der ursprünglichen Fernsehprogramme erlangt wird, welche von der Schlüsselstation zur örtlichen Station übertragen werden, durch einen neuen Strom ersetzt wird, der ursprüngliche Strom, der durch Codieren der ursprünglichen Fernsehprogramme erlangt wird, ausgedrückt als "ST_{OLD}", was einen alten Strom bezeichnet, und der Ersatzstrom wird, um einen Teil des ursprünglichen Stroms ST_{OLD} neu zu ersetzen, ausgedrückt als "ST_{NEW}". Fig. 4C zeigt einen gespleißten Strom ST_{SPL}, der durch Spleißen des Ersatzstroms ST_{NEW} in Bezug auf den ursprünglichen Strom ST_{OLD} im Spleißpunkt SP erlangt wird, und die Ortskurve der Datenbelegungsmenge im VBV-Puffer, welche dem gespleißten Strom ST_{SPL} entspricht.

[0017] In Fig. 4A bis Fig. 4C drücken rechte Aufwärtsbereiche (schräge Bereiche) der Ortskurve der Datenbelegungsmenge im VBV-Puffer die Übertragungsbitrate aus, und die vertikal abfallenden Bereiche drücken die Menge an Bits aus, welche aus dem Decoderpuffer durch den Videodecoder herausgeführt werden, um jedes Bild wiederzugeben. Der zeitliche Ablauf, mit dem der Videodecoder Bits vom Decoderpuffer herausführt, wird gemäß der Information bestimmt, welche als Decodierzeitstempel (DTD) bezeichnet wird. In Fig. 4A bis Fig. 4C zeigen I, P und B das I-Bild, das P-Bild bzw. das B-Bild.

[0018] Der ursprüngliche Codierstrom ST_{OLD} ist ein Strom, der durch den Videocodierer der Schlüsselstation codiert wurde, und der Ersatzstrom ST_{NEW} ist ein Strom, der durch den Videocodierer der örtlichen Station codiert wurde. Der ursprüngliche codierte Strom ST_{OLD} und der Ersatzstrom ST_{NEW} sind durch ihre entsprechenden Videocodierer individuell codiert. Da der Videocodierer der örtlichen Station die Decodierverarbeitung durchführt, um spezifisch den Ersatzstrom ST_{NEW} zu erzeugen, ohne die Ortskurve der Datenbelegungsmenge im VBV-Puffer des Videocodierers der Schlüsselstation überhaupt zu kennen, sind daher die Datenbelegungsmenge VBV_{OLD} des ursprünglichen Stroms ST_{OLD} im VBV-Puffer am Spleißpunkt und die Datenbelegungsmenge VBV_{NEW} des Ersatzstroms ST_{NEW} im VBV-Puffer am Spleißpunkt voneinander verschieden.

[0019] Kurz ausgedrückt muss, um eine Diskontinuität der Ortskurve der Datenbelegungsmenge im VBV-Puffer im Bereich des Spleißpunkts SP des Spleißstroms ST_{SPL} zu verhindern, der Anfangspegel der Datenbelegungsmenge des Ersatzstroms ST_{NEW} des Spleißstroms ST_{SPL} im VBV-Puffer gleich dem Anfangspegel der Datenbelegungsmenge VBV_{OLD} im VBV-Puffer sein. Als Ergebnis erzeugt, wie in Fig. 4A bis Fig. 4C gezeigt ist, wenn der Wert der Datenbelegungsmenge VBV_{NEW} des Ersatzstroms ST_{NEW} im VBV-Puffer kleiner ist als der Wert der Datenbelegungsmenge VBV_{OLD} des ursprünglichen Stroms ST_{OLD} im VBV-Puffer, der VBV-Puffer einen Überlauf im Teil des Ersatzstroms ST_{NEW} des Spleißstroms ST_{SPL}. Wenn dagegen der Wert der Datenbelegungsmenge VBV_{NEW} des Ersatzstroms ST_{NEW} im VBV-Puffer größer ist als der Wert der Datenbelegungsmenge VBV_{OLD} des ursprünglichen Stroms ST_{OLD} im VBV-Puffer, erzeugt der VBV-Puffer einen Unterlauf im Teil des Ersatzstroms ST_{NEW} des Spleißstroms ST_{SPL}.

[0020] Das zweite Problem wird nun beschrieben.

[0021] In einem Kopf eines Stroms, der auf Basis des MPEG-Standards codiert ist, werden verschiedene Datenelemente und Flags, welche codierte Information zeigen, beschrieben. Der Codierstrom wird unter Verwendung dieser Datenelemente und Flags decodiert.

[0022] Die Programme 1, 2, 3 und 4, welche den Hauptbereich der ursprünglichen Fernsehprogramme bilden, welche in Fig. 2A bis Fig. 2C gezeigt sind, sind nicht notwendigerweise aus Fernsehsignalen des NTSC-Systems gebildet, welche eine Rahmenrate von 29,97 Hz (ungefähr 30 Hz) haben, welche durch eine Videokamera oder dgl. aufgezeichnet sind, und können aus Fernsehsignalen bestehen, welche von einem Kinomaterial umgesetzt wurden, das eine Rahmenrate von 24 Hz (24 Rahmen pro Sekunde) hat. Allgemein wird die Verarbeitung zum Umsetzen eines Kinomaterials aus 24 Hz in Fernsehsignale von 29,97 Hz als "2:3-Herunterziehverarbeitung" bezeichnet, da sie das Verarbeiten zum Umsetzen von zwei Teilbildern des ursprünglichen Materials in drei Teilbilder einer vorgegebenen Sequenz umfasst.

[0023] Fig. 5 zeigt diese 2:3-Herunterziehverarbeitung. In Fig. 5 bezeichnen T1 bis T8 Kopffelder eines Kinofilmmaterials, welches eine Rahmenfrequenz von 24 Hz hat, und B1 bis B8 zeigen Bodenfelder des Filmmaterials, welches eine Rahmenfrequenz von 24 Hz hat. Die Ellipsen und die Dreiecke, die in Fig. 5 gezeigt sind, zeigen die Strukturen von Rahmen, welche durch Kopffelder und Bodenfelder gebildet sind.

[0024] Insbesondere wird bei der 2:3-Herunterziehverarbeitung die Verarbeitung zum Einfügen von vier Wiederholungsfelder in das Filmmaterial (acht Kopffelder T1 bis T8 und acht Bodenfelder B1 bis B8), welche eine Rahmenfrequenz von 24 Hz haben, ausgeführt. Die vier Wiederholungsfelder umfassen ein Wiederholungsfeld B2', welches durch Wiederholen des Bodenfelds B2 erzeugt wird, ein Wiederholungsfeld T4', welches durch Wiederholen des Kopffelds T4 erzeugt wird, ein Wiederholungsfeld B6', welches durch Wiederholen des Bodenfelds B6 erzeugt wird, und ein Wiederholungsfeld T8', welches durch Wiederholen des Kopffelds T8 erzeugt wird. Als Ergebnis werden durch diese 2:3-Herunterziehverarbeitung Fernsehsignale, welche eine Rahmenfrequenz von 29,97 Hz haben, von dem Kinofilmmaterial, welches eine Rahmenfrequenz von 24 Hz hat, erzeugt.

[0025] Im MPEG-Codierer werden die Fernsehsignale, welche durch 2:3-Herunterziehverarbeitung erlangt werden, nicht unmittelbar durch den Videocodierer codiert, sondern sie werden codiert, nachdem die Wiederholungsfelder von den 2:3-herunterzieh-verarbeiteten Fernsehsignalen entfernt sind. In dem in Fig. 5 gezeigten Beispiel werden die Wiederholungsfelder B2', T4', B6' und T8' von dem 2:3-herunterzieh-verarbeiteten Fernsehsignale entfernt. Der Grund, um die Wiederholungsfelder vor der Codierverarbeitung zu entfernen, ist der, dass die Wiederholungsfelder redundante Felder sind, welche im Zeitpunkt der 2:3-Herunterziehverarbeitung eingefügt sind und keine Verschlechterung der Bildqualität bewirken, sogar, wenn sie gelöscht werden, um die Kompressionscodierwirksamkeit zu verbessern.

[0026] Gemäß dem MPEG-Standard ist außerdem vorgeschrieben, dass ein Flag "erstes Wiederholungsfeld", welches zeigt, ob ein Wiederholungsfeld erzeugt werden sollte oder nicht, indem eines von zwei Feldern, welches einen Rahmen bildet, wiederholt wird, bei dem Decodieren eines codierten Stroms beschrieben wird. Insbesondere erzeugt beim Decodieren eines Codierstroms, wenn das Flag "erstes Wiederholungsfeld" im Codierstrom gleich "1" ist, der MPEG-Decoder ein Wiederholungsfeld. Wenn das Flag "erstes Wiederholungsfeld" im Codierstrom gleich "0" ist, erzeugt der MPEG-Decoder kein Wiederholungsfeld.

[0027] In dem in Fig. 5 gezeigten Beispiel ist das "erste Wiederholungsfeld" eines Stroms, der durch Codieren des Rahmens erlangt wird, der durch das Kopffeld T1 und das Bodenfeld B1 gebildet wird, gleich "0", und das "erste Wiederholungsfeld" eines Stroms, der durch Codieren des Rahmens erlangt wird, der durch das Kopffeld T2 und das Bodenfeld B2 gebildet wird, gleich "1". Das Flag das "erste Wiederholungsfeld" eines Stroms, der durch Codieren des Rahmens erlangt wird, der durch das Kopffeld T3 und das Bodenfeld B3 gebildet wird, ist "0", und das "erste Wiederholungsfeld" eines Stroms, der durch Codieren des Rahmens erlangt wird, der durch das Kopffeld T4 und das Bodenfeld B4 gebildet wird, ist "1". Daher wird beim Decodieren des Codierstroms des Rahmens, der durch das Kopffeld T2 und das Bodenfeld B2 gebildet wird, das Wiederholungsfeld B2' erzeugt. Beim Decodieren des Codierstroms des Rahmens, der durch den Kopfrahmen T4 und den Bodenrahmen B4 gebildet wird, wird das Wiederholungsfeld T4' erzeugt.

[0028] Gemäß dem MPEG-Standard ist zusätzlich vorgeschrieben, dass ein Flag "erstes Kopffeld", welches zeigt, ob das erste Feld von zwei Feldern, welche einen Rahmen bilden, ein Kopffeld oder ein Bodenfeld ist, in einem Codierstrom beschrieben. Wenn insbesondere das "erste Kopffeld" gleich "1" ist, zeigt dies eine Rahmenstruktur, bei dem das Kopffeld zeitlich dem Bodenfeld vorhergeht. Wenn das "erste Kopffeld" gleich "0" ist, zeigt dies eine Rahmenstruktur, bei dem das Kopffeld zeitlich dem Bodenfeld folgt.

[0029] Im Beispiel von Fig. 5 ist das "erste Kopffeld" des Codierstroms des Rahmens, der durch das Kopffeld T1 und das Bodenfeld B1 gebildet wird, gleich

"0", und das "erste Kopffeld" des Codierstroms des Rahmens, der durch das Kopffeld T2 und das Bodenfeld B2 gebildet wird, ist "0". Das Flag "erstes Kopffeld" des Codierstroms des Rahmens, der durch das Kopffeld T3 und das Bodenfeld B3 gebildet wird, ist "1", und das "erste Kopffeld" des Codierstroms des Rahmens, der durch das Kopffeld T4 und das Bodenfeld B4 gebildet wird, ist "1".

[0030] In Bezug auf Fig. 6A bis Fig. 6C wird ein Problem, welches in Bezug auf die Flags erzeugt wird, beispielsweise "erstes Kopffeld" und "erstes Wiederholungsfeld", welche gemäß dem MPEG-Standard definiert sind, wenn der Codestrom gespleißt wird, beschrieben.

[0031] Fig. 6A zeigt die Rahmenstruktur des ursprünglichen Stroms ST_{OLD}, der durch Codieren der ursprünglichen Fernsehprogramme erlangt wird, welche in der Schlüsselstation erzeugt werden. Fig. 6B zeigt die Rahmenstruktur des Ersatzstroms ST_{NEW}, der durch Codieren der Werbung CM1' für örtliche Zuschauer erlangt wird, welche in der örtlichen Station erzeugt wird. Fig. 6C zeigt die Rahmenstruktur des Spleißstroms ST_{SPL}, der durch Spleißverarbeitung erlangt wird.

[0032] Das Programm 1 und das Programm 2 im ursprünglichen Strom ST_{OLD} sind Codierströme, welche durch die 2:3-Herunterziehverarbeitung erlangt werden, und jeder Rahmen der Werbung CM1 des Hauptbereichs ist ein Codierstrom, der die Rahmenstruktur hat, bei dem das "erste Kopffeld" gleich "0" ist. Die örtliche Werbung CM1', welche in Fig. 6B gezeigt ist, ist ein Codierstrom, um die Werbung CM1 in den ursprünglichen Fernsehprogrammen zu ersetzen, und hat die Rahmenstruktur, bei der das "erste Kopffeld" gleich "1" ist. Der Spleißstrom ST_{SPL}, der in Fig. 6C gezeigt ist, ist ein Strom, der durch Spleißen des Ersatzstroms ST_{NEW} erzeugt wird, der dem ursprünglichen Strom ST_{OLD} folgt, der durch das Programm 1 gezeigt wird, und dann das Spleißen des ursprünglichen Stroms ST_{OLD}, der durch das Programm 2 gezeigt wird, im Anschluss an den Ersatzstrom ST_{NEW}. Kurz ausgedrückt ist der Spleißstrom ST_{SPL} ein Strom, der erlangt wird, indem die örtliche Werbung CM1' anstelle der Hauptwerbung CM1 des ursprünglichen Stroms ST_{OLD} eingefügt wird.

[0033] Die Werbung CM1, welche in der Schlüsselstation, welche in **Fig.** 6 gezeigt ist, erzeugt wird, ist ein Codierstrom mit jedem Rahmen, der die Rahmenstruktur hat, bei dem das "erste Kopffeld" gleich "0" ist. Die Werbung CM1', welche in der örtlichen Station erzeugt wird, ist ein Codierstrom, welcher die Rahmenstruktur hat, bei dem das "erste Kopffeld" gleich "1" ist.

[0034] In dem Fall, wo die Rahmenstruktur der Werbung CM1 und die Rahmenstruktur der Ersatzwerbung CM1', um die Werbung CM1 zu ersetzen, voneinander verschieden sind, wie in Fig. 6A und Fig. 6B gezeigt ist, wird, wenn der Strom der Werbung CM1' nachfolgend in den Strom des Programms 1 im Spleißpunkt SP1 im ursprünglichen Strom ST_{OLD} gespleißt wird, eine Feldlücke im Spleißstrom ST_{SPL} erzeugt. Die Feldlücke bedeutet einen Ausfall des Bodenfelds B6 am Spleißpunkt SP1 vom Spleißstrom ST_{SPL}, wodurch eine Diskontinuität im Wiederholungsmuster des Kopffelds und des Bodenfelds bewirkt wird, wie in Fig. 6C gezeigt ist.

[0035] Der Codierstrom, in welchem die Feldlücke erzeugt wird, um Diskontinuität im Feldmuster zu erzeugen, ist ein Codierstrom, der für den MPEG-Standard unbequem ist und kann durch einen üblichen MPEG-Decoder nicht normal decodiert werden.

[0036] Wenn dagegen der Strom des Programms 2 nachfolgend in die Werbung CM1' am Spleißpunkt SP2 im ursprünglichen Strom ST_{OLD} gespleißt wird, wie in Fig. 6A und Fig. 6B gezeigt ist, wird eine Feldvervielfältigung im Spleißstrom ST_{SPL} erzeugt. Diese Feldvervielfältigung bedeutet das Vorhandensein eines Bodenfelds b12 und eines Bodenfelds B12 am Spleißpunkt SP2 im gleichen Anzeigezeitpunkt, wie in Fig. 6C gezeigt ist.

[0037] Der Codierstrom, in welchem die Feldvervielfältigung erzeugt wird, ist, um Diskontinuität im Feldmuster zu bewirken, ein Codierstrom, der für den MPEG-Standard ungünstig ist und der durch einen üblichen MPEG-Decoder nicht normal decodiert werden kann.

[0038] Wenn kurz ausgedrückt die Spleißverarbeitung vereinfachend ausgeführt wird, wird das Feldmuster oder das Rahmenmuster unstetig, und ein Spleißstrom in Übereinstimmung mit dem MPEG-Standard kann nicht erzeugt werden.

[0039] Im Dokument XP002098561 ist ein System beschrieben, um einen komprimierten Videostrom zu editieren, der von zwei komprimierten Videosignalen hergeleitet wird, indem entsprechende decodierte Signale umgeschaltet werden. In der EP-A 0 755 157 ist die Einfügung einer Nullstopfinformation in einem codierten Bitstrom beschrieben, der durch Spleißen zweier komprimierter Videosignal erlangt wird.

[0040] In der EP-A 0 720 371 ist die Einfügung von ungültigen Rahmen bei einer 3:2-Herunterzieh-Signalverarbeitung beschrieben, um Synchronisation mit dem Rahmensynchronisationssignal zu erzielen.

Offenbarung der Erfindung

[0041] Es besteht die Notwendigkeit, eine Codierstrom-Spleißeinrichtung bereitzustellen, um saumlose Spleißverarbeitung zu realisieren, bei der eine ste-

tige Ortskurve der Datenbelegungsmenge eines Spleißstrom im VBV-Puffer und kein Zusammenbruch des VBV-Puffers erzeugt wird. Es besteht außerdem die Notwendigkeit, eine Codierstrom-Spleißeinrichtung bereitzustellen, um saumlose Spleißverarbeitung zu realisieren, wobei Diskontinuität in der Stromstruktur eines Codierstroms im Bereich eines Spleißpunkts verhindert wird.

[0042] Gemäß einem ersten Gesichtspunkt liefert die vorliegende Erfindung eine Codierstrom-Spleißeinrichtung zum Spleißen eines ersten Codierstroms und eines zweiten Codierstroms an einem Spleißpunkt, wobei die Vorrichtung aufweist:
eine Änderungseinrichtung zum Ändern zumindest eines Codierparameters des zweiten Codierstroms auf Basis eines entsprechenden Codierparameters des ersten Codierstroms, so dass der erste Codierstrom und der zweite Codierstrom saumlos am Spleißpunkt gespleißt werden; und
eine Spleißeinrichtung zum Spleißen des ersten Codierstroms und des zweiten Codierstroms, der die geänderten Codierparameter hat;
wobei der Codierparameter, der durch die Änderungseinrichtung geändert wird, zumindest einer von einem ersten Wiederholungsfeldflag ist, welches zeigt, ob oder nicht ein Wiederholungsfeld im Decodierzeitpunkt erzeugt wird, und
ein erstes Kopffeldflag, welches zeigt, ob das erste Feld ein Kopffeld oder ein Bodenfeld ist.

[0043] Damit kann saumlose Spleißverarbeitung, bei der eine fortlaufende Ortskurve der Datenbelegungsmenge eines Spleißstroms in einem VBV-Puffer erzeugt wird und wobei kein Zusammenbruch des VBV-Puffers erzeugt wird, realisiert werden. Gemäß von zumindest Ausführungsformen der Erfindung kann eine Spleißverarbeitung, welche die Erzeugung eines saumlosen Stroms ermöglicht, welche eine Konsistenz mit der der Stromstruktur des Codierstroms im Bereich des Spleißpunkts hat, welcher nicht stetig ist, realisiert werden.

[0044] Gemäß einem zweiten Merkmal liefert die vorliegende Erfindung ein Codierstrom-Spleißverfahren zum Spleißen eines ersten Codierstroms und eines zweiten Codierstroms an einem Spleißpunkt, wobei das Verfahren aufweist:
einen Änderungsschritt zum Ändern zumindest eines Codierparameters des zweiten Codierstroms auf Basis eines entsprechenden Codierparameters des ersten Codierstroms, so dass der erste Codierstrom und der zweite Codierstrom saumlos am Spleißpunkt gespleißt werden; und
einen Spleißschritt zum Spleißen des ersten Codierstroms und des zweiten Codierstroms, der den geänderten Codierparameter hat;
wobei der Codierparameter, der durch die Änderungseinrichtung geändert wird, zumindest einer von einem ersten Wiederholfeldflag ist, der zeigt, ob ein Wiederholfeld im Decodierzeitpunkt zu erzeugen ist, und
ein erstes Kopffeldflag, welches zeigt, ob das erste Feld ein Kopffeld oder ein Bodenfeld ist.

[0045] Damit kann saumlose Spleißverarbeitung, bei der eine fortlaufende Ortskurve der Datenbelegungsmenge eines Spleißstroms in einem VBV-Puffer erzeugt wird und wobei kein Zusammenbruch des VBV-Puffers erzeugt wird, realisiert werden. Gemäß von zumindest Ausführungsformen der vorliegenden Erfindung kann außerdem die Spleißverarbeitung, welche die Erzeugung eines saumlosen Stroms ermöglicht, welche Konsistenz mit der des Stormaufbaus des codierten Stroms im Bereich des Spleißpunkts hat, welcher nicht stetig ist, realisiert werden.

Kurzbeschreibung der Zeichnungen

[0046] Fig. 1 zeigt ein typisches Fernsehrundfunksystem;

[0047] Fig. 2A bis Fig. 2C zeigen die Editierverarbeitung von Fernsehprogrammen in einer örtlichen Station;

[0048] Fig. 3 zeigt ein typisches digitales Übertragungssystem, bei dem der MPEG-Standard verwendet wird;

[0049] Fig. 4A bis Fig. 4C zeigen eine Steuerung eines VBV-Puffers;

[0050] Fig. 5 zeigt die 2:3-Herunterziehverarbeitung:

[0051] Fig. 6A bis Fig. 6C zeigen Probleme, welche beim Spleißen eines Codierstroms erzeugt werden;

[0052] Fig. 7 ist ein Blockdiagramm, welches den Gesamtaufbau eines digitalen Rundfunksystems zeigt, welches eine Codierstrom-Spleißeinrichtung gemäß einer Ausführungsform der vorliegenden Erfindung aufweist;

[0053] Fig. 8 ist ein Blockdiagramm, um ausführlich den Aufbau eines MPEG-Codierblocks **35** einer Schlüsselstation **30** und eines Codierblocks einer örtlichen Station **40** zu erläutern;

[0054] Fig. 9A bis Fig. 9D zeigen die Verarbeitung zum Erzeugen eines Transportstroms von einem elementaren Strom in einer Stromumsetzungsschaltung;

[0055] Fig. 10 zeigt die Syntax einer Sequenz von MPEG-Videoelementarströmen;

[0056] Fig. 11 zeigt die Syntax eines Sequenzkopfs (Sequenzkopf);

[0057] Fig. 12 zeigt die Syntax einer Sequenzerweiterung;

[0058] Fig. 13 zeigt die Syntax der Erweiterung und Benutzerdaten (Erweiterung- und Benutzerdaten);

[0059] Fig. 14 zeigt die Syntax eines Gruppenbildkopfs (Gruppenbild-Kopf);

[0060] Fig. 15 zeigt die Syntax des Bildkopfs (Bildkopf);

[0061] Fig. 16 zeigt die Syntax der Bildcodiererweiterung (Bildcodiererweiterung);

[0062] Fig. 17 zeigt die Syntax von Bilddaten;

[0063] Fig. 18 zeigt die Syntax eines Transportstrompakets;

[0064] Fig. 19 zeigt die Syntax eines Adaptionsfelds;

[0065] Fig. 20 ist ein Blockdiagramm, um den einfachen Aufbau einer örtlichen Station und den Aufbau einer Stromspleißeinrichtung zu erläutern;

[0066] Fig. 21A bis Fig. 21C zeigen die Verarbeitung bezogen auf einen VBV-Puffer der Stromspleißeinrichtung;

[0067] Fig. 22A bis Fig. 22C zeigen ein erstes Verarbeitungsbeispiel, welches auf "erstes Kopffeld" und "erstes Wiederholungsfeld" der Stromspleißeinrichtung bezogen ist;

[0068] Fig. 23A bis 23C zeigen ein zweites Verarbeitungsbeispiel, welches auf "erstes Kopffeld" und "erstes Wiederholungsfeld" der Stromspleißeinrichtung bezogen ist.

Bevorzugte Ausführungsart

[0069] Fig. 7 zeigt den Aufbau eines digitalen Rundfunksystems, welches eine Codierstromspleißeinrichtung aufweist, gemäß einer Ausführungsform der vorliegenden Erfindung.

[0070] Wie in Fig. 7 gezeigt ist, weist ein digitales Rundfunksystem allgemein eine Schlüsselstation **30** und eine lokale Station **40** auf, welche mit dieser Schlüsselstation verbunden ist.

[0071] Die Schlüsselstation **30** ist eine Rundfunkstation, um gemeinsame Fernsehprogramme zu erzeugen und zu den verbundenen örtlichen Stationen zu übertragen. Die Schlüsselstation **30** weist eine Rundfunksystemsteuerung **31**, einen Materialserver **32**, einen CM-Server **33**, einen Matrixschalterblock **34**, einen MPEG-Codierblock **35**, einen Multiplexer **36** und eine Modulationsschaltung **37** auf.

[0072] Die Rundfunksystemsteuerung **31** ist ein System, um alle Einheiten und Schaltungen, welche in der Rundfunkstation vorgesehen sind, beispielsweise den Materialserver **32**, den CM-Server **33**, den Matrixschalterblock **34**, den MPEG-Codierblock **35**, den Multiplexer **36** und die Modulationsschaltung **37** umfassend zu verwalten und zu steuern. In dieser Rundfunksystemsteuerung **31** ist eine Programmeditiertabelle registriert, um die Sendezeit aller Materialien zu verwalten, beispielsweise der Programmmaterialien, welche von einer Programmversorgungsgesellschaft geliefert werden, und der Programmmaterialien und CM-Materialien, welche in der Schlüsselstation selbst erzeugt werden. Gemäß der Programmeditiertabelle steuert die Sendesystemsteuerung **31** die oben beschriebenen Einheiten und Schaltungen. Die Programmeditiertabelle besteht beispielsweise aus einer Ereignisinformationsdatei, in welcher die Sendeprogrammtabelle für jede Stunde oder jeden Tag aufgezeichnet ist, und einer Betriebsinformationsdatei, in welcher der Zeitplan der Sendeprogramme für alle 15 Sekunden aufgezeichnet ist, usw..

[0073] Der Materialserver **32** ist ein Server, um Videodaten und Audiodaten von Filmprogrammen, Sportprogrammen, Unterhaltungsprogrammen und Nachrichtenprogrammen zu speichern, welche als Hauptbereiche von Fernsehprogrammen zu senden sind, und um ein Programm, welches durch die Rundfunksystemsteuerung **31** bestimmt wird, in dem Zeitpunkt konform mit dem Zeitplan in Bezug auf die Programmeditiertabelle auszugeben. Das Filmprogramm umfasst Videodaten, welche durch Umsetzen eines Filmmaterials, welches eine Rahmenfrequenz von 24 Hz hat, in Fernsehsignale, welche eine Rahmenfrequenz von 30 Hz haben, durch 2:3-Herunterziehverarbeitung zu erlangen, wie oben beschrieben. Die Videodaten und die Audiodaten, welche als Programm des Hauptbereichs vom Materialserver **32** ausgegeben werden, werden zu einem Matrixschalterblock **34** geliefert. In dem in Fig. 2A bis Fig. 2C gezeigten Beispiel werden das Programm 1, das Programm 2, das Programm 3 und das Programm 4 in diesem Materialserver **32** aufgezeichnet. Die Videodaten und die Audiodaten, welche im Materialserver **32** gespeichert sind, sind Basisband-Videodaten und Audiodaten, welche nicht kompressions-codiert sind.

[0074] Der CM-Server **33** ist ein Server, um Werbung, welche zwischen den Programmen des Hauptbereichs, die vom Materialserver **32** wiedergegeben werden, eingefügt wird, zu speichern, und um eine Werbung, welche durch die Rundfunksystemsteuerung **31** bezeichnet wird, mit dem zeitlichen Ablauf konform mit dem Zeitschema auf der Programmeditiertabelle auszugeben. Die Videodaten und die Audiodaten, welche als Werbung vom CM-Server **33** ausgegeben werden, werden zum Matrixschalterblock

34 geliefert. In dem in Fig. 2A bis Fig. 2C gezeigten Beispiel sind die Werbung CM1, die Werbung CM2 und Werbung CM3 in diesem CM-Server **33** aufgezeichnet. Die Videodaten und die Audiodaten, welche im CM-Server **33** gespeichert sind, sind Basisband-Videodaten und Audiodaten, welche nicht kompressions-codiert sind.

[0075] Der Matrixschalterblock **34** hat eine Matrixschaltung, um Liveprogramme beispielsweise Live-Sport-Rundfunk- und Nachrichtenprogramme, die Hauptprogramme, welche vom Materialserver **32** ausgegeben werden, und die Werbungsprogramme, welche vom CM-Server **33** ausgegeben werden, weiter zu leiten. Der Matrixschalterblock **34** hat außerdem eine Schaltschaltung, um die Hauptprogramme, welche vom Materialserver **32** geliefert werden und die Werbungsprogramme, welche vom CM-Server **33** geliefert werden, zu verbinden und zu schalten, mit dem Zeitablauf konform mit dem Zeitplan auf der Programmeditiertabelle, welche durch die Rundfunksystemsteuerung bestimmt wird. Durch Umschalten der Hauptprogramme und der CM-Programme unter Verwendung dieser Schaltschaltung kann das Übertragungsprogramm PG_{OLD}, welches in Fig. 2A gezeigt ist, erzeugt werden.

[0076] Der MPEG-Codierblock **35** ist ein Block, um die Basisband-Videodaten und Audiodaten, welche vom Matrixschalterblock ausgegeben werden, auf der Basis des MPEG2-Standards zu codieren, und hat mehrere Videocodierer und Audiocodierer.

[0077] Der Multiplexer **36** multiplext 9 Kanäle der Transportströme, welche vom MPEG-Codierblock **35** ausgegeben werden, wodurch ein gemultiplexter Transportstrom erzeugt wird. Daher existieren im gemultiplexten Transportstrom die Transportstrompakete, welche die codierten Videoelementarströme von 9 Kanälen aufweisen, und die Transportstrompakete, welche die codierten Audioelementarströme von 9 Kanälen aufweisen, in einer gemischten Weise.

[0078] Die Modulationsschaltung **37** führt die QPSK-Modulation des Transportstroms durch und überträgt die modulierten Daten zur örtlichen Station **40** und zu einem Haushalt **61** über die Übertragungsleitungen.

[0079] Der Gesamtaufbau der örtlichen Station **40** wird nun mit Hilfe von Fig. 7 beschrieben.

[0080] Die örtliche Station **40** ist eine Sendestation, um gemeinsame Fernsehprogramme, welche von der Schlüsselstation für örtliche Zuschauer übertragen werden, zu editieren, und um die Fernsehprogramme, welche für örtliche Zuschauer editiert werden, zu jedem Haushalt zu senden. Die örtliche Station **40** umfasst eine Rundfunksystemsteuerung **41**, eine Demodulationsschaltung **42**, einen Demultiplexer **43**, eine Stromumsetzungsschaltung **44**, einen Materialserver **46**, einen CM-Server **47**, einen Codierblock **48**, einen Stromserver **49**, einen Stromspleißeinrichtung **50**, einen Stromprozessor **51**, einen Multiplexer **52** und eine Modulationsschaltung **53**.

[0081] Die Rundfunksystemsteuerung **41** ist ein System, um umfassend alle Einheiten und Schaltungen, welche in der örtlichen Station vorgesehen sind, zu verwalten und zu steuern, beispielsweise die Demodulationsschaltung **42**, den Demultiplexer **43**, die Stromumsetzungsschaltung **44**, den Materialserver **46**, den CM-Server **47**, den Codierblock **48**, den Stromserver **49**, die Stromspleißeinrichtung **50**, den Stromprozessor **51**, den Multiplexer **52** und die Modulationsschaltung **53**, ähnlich wie die Sendesystemsteuerung **31** der Schlüsselstation **30**. In der Sendesystemsteuerung **31** ist ähnlich wie bei der Sendesystemsteuerung **31** der Schlüsselstation **30** eine Programmeditiertabelle registriert, um die Sendezeit von editierten Fernsehprogrammen zu verwalten, welche erlangt werden, indem Programme und CMs, welche in der örtlichen Station erzeugt werden, in das Übertragungsprogramm eingefügt werden, welches von der Schlüsselstation **30** geliefert wird. Gemäß dieser Programmeditiertabelle steuert die Sendesystemsteuerung **41** die oben beschriebenen Einheiten und Schaltungen.

[0082] Die Demodulationsschaltung **42** führt die QPSK-Demodulation des Übertragungsprogramms aus, welches von der Schlüsselstation **30** über die Übertragungsleitung übertragen wird, wodurch ein Transportstrom erzeugt wird.

[0083] Der Demultiplexer **43** demultiplext den Transportstrom, der von der Demodulationsschaltung **42** ausgegeben wird, um Transportströme von 9 Kanälen zu erzeugen, und gibt den Transportstrom jedes Kanals an die Stromumsetzungsschaltung **44** aus. Kurz ausgedrückt führt dieser Demultiplexer **43** die Umkehrverarbeitung der Verarbeitung des Multiplexers **36** der Schlüsselstation **30** durch.

[0084] Die Stromumsetzungsschaltung **44** ist eine Schaltung, um die Transportströme, welche vom Demultiplexer **43** geliefert werden, in die Form von Elementarströmen umzusetzen.

[0085] Der Materialserver **46** ist ein Server, der Videodaten und Audiodaten von Unterhaltungsprogrammen, Nachrichtenprogrammen und dgl. speichert, welche als Fernsehprogramme für örtliche Zuschauer zu senden sind. Der CM-Server **47** ist ein Server, um Videodaten und Audiodaten örtlicher Werbungen zu speichern, welche zwischen die Hauptprogramme einzufügen sind, welche von der Schlüsselstation **30** zu liefern sind. Die Videodaten und die Audiodaten, welche im Materialserver **46** und

im CM-Server **47** gespeichert sind, sind Basisband-Videodaten und Audiodaten, welche kompressions-codiert sind.

[0086] Der Codierblock **48** ist ein Block, um die Videodaten mehrerer Kanäle und die Audiodaten mehrerer Kanäle zu codieren, welche vom Materialserver **46** und vom CM-Server **47** geliefert werden, und hat mehrere Videocodierer und mehrere Audiocodierer entsprechend den mehreren Kanälen. Der Unterschied zwischen dem Codierblock **48** und dem MPEG-Codierblock **35** der Schlüsselstation ist der, dass der Codierblock **48** der örtlichen Station **40** die elementaren Ströme ausgibt, während der MPEG-Codierblock **35** der Schlüsselstation **30** den Transportstrom ausgibt. Die wesentliche Funktion und die Verarbeitung des Codierblocks **48** sind jedoch gleich denen des MPEG-Codierblocks **35** der Schlüsselstation **30**. Unter den Elementarströmen der mehreren Kanäle, welche vom Codierblock **48** ausgegeben werden, werden die Elementarströme von drei Kanälen zum Stromserver **49** geliefert, und die Elementarströme der verbleibenden Kanäle werden zur Stromspleißeinrichtung **50** geliefert.

[0087] Der Stromserver **49** empfängt die Elementarströme der drei Kanäle, welche vom Codierblock geliefert werden. Der Stromserver **49** zeichnet dann die Elementarströme im Zustand von Strömen auf einem zufallsmäßig zugreifbaren Aufzeichnungsmedium auf, und gibt die Elementarströme vom zufallsmäßig zugreifbaren Aufzeichnungsmedium unter der Steuerung der Sendesystemsteuerung **41** wieder.

[0088] Die Stromspleißeinrichtung **50** ist ein Block, um die mehreren Elementarströme, welche vom Codierblock **48** und vom Stromserver **49** geliefert werden, weiterzuleiten, gibt dann die geleiteten Elementarströme an eine vorgegebene Ausgangsleitung aus und spleißt die Elementarströme, welche von Schlüsselstation **30** geliefert werden, und die Elementarströme, welche in der örtlichen Station **40** erzeugt werden, beim Strompegel. Die Verarbeitung durch die Stromspleißeinrichtung **50** wird später ausführlich beschrieben.

[0089] Die Stromumsetzungsschaltung **51** ist eine Schaltung, um die Elementarströme, welche als Spleißströme von der Stromspleißeinrichtung **50** ausgegeben werden, zu empfangen, und um die Elementarströme in Transportströme umzusetzen.

[0090] Der Multiplexer **52** multiplext die Transportströme von 9 Kanälen, welche von der Stromumsetzungsschaltung ausgegeben werden, wodurch ein gemultiplexter Strom erzeugt wird, ähnlich dem Multiplexer **36** der Schlüsselstation **30**.

[0091] Die Modulationsschaltung **53** führt die QPSK-Modulation des Transportstroms aus und verteilt die modulierten Daten auf jeden Haushalt **62** über eine Übertragungsleitung.

[0092] Fig. 8 ist ein Blockdiagramm, um ausführlich den Aufbau des MPEG-Codierblocks **35** der Schlüsselstation **30** und des Codierblocks **48** der örtlichen Station **40** zu erläutern. Da der MPEG-Codierblock **35** der Schlüsselstation **30** und der Codierblock **48** der örtlichen Station **40** im Wesentlichen den gleichen Aufbau haben, wird der Aufbau und die Funktion des MPEG-Codierers **35** der Schlüsselstation als typisches Beispiel beschrieben.

[0093] Der MPEG-Codierblock **35** hat eine Codiersteuerung **350**, um alle Schaltungen des MPEG-Codierblocks **35** zentral zu steuern, mehrere MPEG-Videocodierer **351-1V** bis **351-V**, um die gelieferten Videodaten mehrerer Kanäle zu codieren, und MPEG-Audiocodierer **351-1A** bis **351-9A**, um mehrere Audiodaten entsprechend den Videodaten auf Basis des MPEG2-Standards zu codieren.

[0094] Der MPEG-Codierblock **35** hat außerdem Stromumsetzungsschaltungen **352-1V** bis **352-9V**, um die codierten Elementarströme (ES), welche von den Videocodierern **351-1V** bis **351-9V** ausgegeben werden, in Transportströme umzusetzen, und Stromumsetzungsschaltungen **352-1A** bis **352-9A**, um die codierten Elementarströme (ES), welche von den Audiocodierern **351-1A** bis **351-9A** ausgegeben werden, in Transportströme umzusetzen. Der MPEG-Codierblock **35** hat außerdem einen Multiplexer **353-1**, um den Transportstrom, der den Videoelementarstrom eines ersten Kanals (1ch) und den Transportstrom, der den Audioelernentarstrorn des ersten Kanals (1ch) umfasst, für jedes Transportstrompaket zu multiplexen, einen Multiplexer **353-2**, um den Transportstrom, der den Videoelementarstrom eines zweiten Kanals (2cH) und den Transportstrom, der den Audioelementarstrom des zweiten Kanals (2ch) umfasst, für jedes Transportstrompaket zu multiplexen, ..., und einen Multiplexer **353-9**, um den Transportstrom, der den Videoelementarstrom eines neunten Kanals (9ch) und den Transportstrom, der den Audioelementarstrom des neunten Kanals (9ch) umfasst, für jedes Transportstrompaket zu multiplexen.

[0095] Der in Fig. 8 gezeigte MPEG-Codierblock **35** hat einen Aufbau, um Übertragungsprogramme von neun Kanälen zu codieren. Es ist natürlich klar, dass die Anzahl der Kanäle nicht auf neun begrenzt ist, und diese eine beliebige Anzahl sein kann.

[0096] Der MPEG-Codierblock **35** führt die Steuerung eines statistischen Multiplexens zum dynamischen Ändern der Übertragungsrate des Übertragungsprogramms eines jeden Kanals gemäß dem Muster der zu codierenden Videodaten aus. Gemäß diesem statistischen Multiplexverfahren werden in

dem Fall, wo das Muster eines Bilds eines Übertragungsprogramms eines Kanals relativ einfach ist und lediglich eine kleine Anzahl von Bits zum Codieren erfordert, während das Muster des Bilds eines anderen Programms relativ kompliziert ist und eine große Anzahl von Bits erfordert, während das Muster des Bilds eines anderen Programms relativ kompliziert ist und eine große Anzahl von Bits zum Codierer erfordert, Bits zum Codieren des Bilds eines Kanals den Bits zugeteilt, um das Bild des anderen Kanals zu codieren, wodurch eine effiziente Übertragungsrate der Übertragungsleitung realisiert wird. Das Verfahren zum dynamischen Ändern der Codierrate jedes Videocodierers wird kurz anschließend beschrieben.

[0097] Die Videocodierer **351-1V** bis **351-9V** erzeugen zunächst Schwierigkeitsdaten (D1 bis D9), welche jeweils die Menge an Bits zeigen, welche zum Codieren eines Bilds als Codierziel erforderlich sind, von der statistischen Menge, beispielsweise dem Bewegungskompensationsrest und einer Intra-AC, welche als Ergebnis der Bewegungskompensation vor der Codierverarbeitung erlangt wird. Die Schwierigkeitsdaten sind Information, welche die Codierschwierigkeit zeigen. Ein großer Schwierigkeitswert zeigt, dass das Muster eines Bilds als Codierziel kompliziert ist, und ein kleiner Schwierigkeitswert zeigt, dass das Muster eines Bilds als Codierziel einfach ist. Die Schwierigkeitsdaten können grob auf Basis der statistischen Menge eingeschätzt werden, beispielsweise Intra-AC und dem Bewegungskompensationsrest (ME-Rest), die im Zeitpunkt der Codierverarbeitung durch die Videocodierer verwendet werden.

[0098] Die Codiersteuerung **350** empfangt die Schwierigkeitsdaten D1 bis D9, welche von den Videocodierern **351-1V** bis **351-9V** entsprechend ausgegeben werden, und berechnet Zielbitraten R1 bis R9 für die Videocodierer **351-1V** bis **351-9V** auf Basis der Schwierigkeitsdaten D1 bis D9 entsprechend. Insbesondere kann die Codiersteuerung **350** die Zielbitdaten R1 bis R9 herausfinden, indem die gesamte Übertragungsrate 'Totalrate' der Übertragungsleitung unter Verwendung der Schwierigkeitsdaten D1 bis D9 proportional verteilt wird, wie in der folgenden Gleichung (1) gezeigt ist:

$$Ri = (Di/\Sigma Dk) \times \text{Totalrate} \quad (1)$$

[0099] In der Gleichung (1) zeigt "Ri" die Zielbitrate des Übertragungsprogramms des Kanals "i", und "Di" zeigt die Schwierigkeitsdaten, um das Bild des Übertragungsprogramms des Kanals "i" zu codieren. "F" zeigt die Summe der Schwierigkeitsdaten von k = 1 bis 9 Kanälen.

[0100] Die Codiersteuerung **350** liefert die Zielbitraten R1 bis R9, welche auf Basis der Gleichung (1) berechnet wurden, zu den entsprechenden Videocodierern **351-1V** bis **351-9V**. Die Zielbitraten R1 bis R9 können für jedes Bild oder für jede GOP berechnet werden.

[0101] Die Videocodierer **351-1V** bis **351-9V** empfangen die Zielbitraten R1 bis R9, welche von der Codiersteuerung **350** geliefert werden, und führen die Codierverarbeitung entsprechend den Zielbitraten R1 bis R9 durch. Durch dieses dynamische Ändern der Zielbitrate des Codierstroms, welche von dem Videocodierer ausgegeben wird, auf Basis der Schwierigkeitsdaten, welche die Codierschwierigkeit des zu codierenden Bilds zeigen, kann die optimale Menge an Bits in Bezug auf die Codierschwierigkeit des zu codierenden Bilds zugeordnet werden, und es wird verhindert, dass die Gesamtmenge der Bitraten, welche von den Videocodierern ausgegeben wird, die Gesamtübertragungsrate 'Totalrate' der Übertragungsleitung überläuft.

[0102] Die Stromumsetzungsschaltung **352-1V** bis **352-9V** und die Stromumsetzungsschaltungen **352-1A** bis **352-9A** sind Schaltungen, um Elementarströme in Transportströme umzusetzen.

[0103] In Bezug auf Fig. 9A bis Fig. 9D wird ein Prozess zum Erzeugen eines Transportstroms von elementaren Videoströmen beschrieben, wobei ein Beispiel verwendet wird, bei dem gelieferte Quellenvideodaten durch den Videocodierer **351-1V** codiert werden, um einen elementaren Videostrom zu erzeugen, der dann in einen Transportstrom durch die Stromumsetzungsschaltung **3521-V** umgesetzt wird.

[0104] Fig. 9A zeigt Quellenvideodaten, welche zum Videocodierer **351-1V** geliefert werden, und Fig. 9B zeigt elementare Videoströme (ES), welche vom Videocodierer **351-1V** ausgegeben werden. Fig. 9C zeigt einen paketierten Elementarstrom (PES), und Fig. 9D zeigt einen Transportstrom (TS).

[0105] Die Datenmenge der Elementarströme, welche gemäß der MPEG2-Standard codiert sind, beispielsweise die Ströme V1, V2, V3 und V4, welche in Fig. 9B gezeigt sind, variiert in Abhängigkeit von der Bildart eines Videorahmens (I-Bild, P-Bild oder B-Bild) und des Vorhandenseins/Nichtvorhandenseins von Bewegungskompensation. Der paketierte Elementarstrom (PES), der in Fig. 9C gezeigt ist, wird durch Paketieren der mehreren Elementarströme und durch Addieren eines PES-Kopfs an seinem Anfang erzeugt. Beispielsweise weist der PES-Kopf einen 24-Bit-Paketstartcode auf, der den Start des PES-Pakets zeigt, eine 8-Bit-Strom-ID, welche die Art der Stromdaten zeigt, welche in einem realen Datenteil des PES-Pakets untergebracht sind (beispielsweise Video- und Audio-Art), eine 16-Bit-Paketlänge, welche die Länge von nachfolgenden Daten zeigt, Codedaten, welche einen Wert "10" zeigen, ein Flagsteuerteil, in welchem verschie-

dene Flaginformation gespeichert ist, eine 8-Bit-PES-Kopflänge, welche die Datenlänge eines Bedingungscodierungsteils zeigt, und ein Bedingungscodeteil einer variablen Länge, in welchem Wiedergabeausgabe-Zeitinformation, welche als PTS (Präsentationszeitstempel) bezeichnet ist, und die Zeitverwaltungsinformation im Decodierzeitpunkt, welche als DTS (Decodierzeitstempel) bezeichnet ist, oder Stopf-Bytes, um die Datenmenge einzustellen, gespeichert sind.

[0106] Der Transportstrom (TS) ist eine Datengruppe von Transportstrompaketen, von denen jedes aus einem 4-Byte-TS-Kopf und einem Nutzinformationsteil besteht, in welchem 184-Byte-Realdaten aufgezeichnet sind. Um dieses Transportstrompaket (TS-Paket) zu erzeugen, wird zunächst ein Datenstrom eines PS-Pakets für jeweils 184-Bytes verwandelt, und die 184-Byte-Realdaten werden in das Nutzinformationsteil eines TS-Pakets eingefügt. Dann wird ein 4-Byte-TS-Kopf den 184-Byte-Nutzinformationsdaten hinzugefügt, womit ein Transportstrompaket erzeugt wird.

[0107] Anschließend werden die Syntax und die Struktur des Elementarstroms unter Bezug auf Fig. 10 bis Fig. 17 beschrieben, und die Syntax und die Struktur des Transportstroms werden ausführlich unter Bezug auf Fig. 18 und Fig. 19 beschrieben.

[0108] Fig. 10 zeigt die Syntax der MPEG-Videoelementarströme. Die Videocodierer **351-1V** bis **351-9V** im Videocodierblock **35** erzeugen codierte Elementarströme gemäß der Syntax, welche in Fig. 10 gezeigt ist. In der Syntax, welche später beschrieben wird, werden Funktionen und Bedingungssätze durch reguläre Zeichen ausgedrückt, und Datenelemente werden durch Zeichen im Klammem ausgedrückt. Datenposten werden als Name, Bit-Länge und beschrieben, welche die Art und die Übertragungsreihenfolge zeigen.

[0109] Zunächst werden die Funktionen, welche in der Syntax, welche in Fig. 10 gezeigt ist, beschrieben. In der Praxis ist die in Fig. 10 gezeigte Syntax eine Syntax, welche verwendet wird, um Daten einer vorher festgelegten Bedeutung von den übertragenen Codierströmen auf Seiten des Videodecoders zu extrahieren. Die auf Seiten des Videocodierers verwendete Syntax ist eine Syntax, welche durch Auslassen von Wenn-Klauseln und While-Klauseln von der Syntax von Fig. 10 erlangt werden.

[0110] Eine Nächst-Start-CodeO-Funktion, welche an der ersten Stelle in der Videosequenz() beschrieben ist, ist eine Funktion, um nach einem Startcode zu suchen, der im Bitstrom beschrieben ist. Im Codierstrom, der gemäß der in Fig. 10 gezeigten Syntax erzeugt wird, werden zunächst die Datenelemente, welche durch eine Sequenzkopf-()-Funktion und eine Sequenz-Erweiterungs-()-Funktion definiert sind, beschrieben. Diese Sequenz-Kopf()-Funktion ist eine Funktion, um Kopfdaten einer Sequenzebene des MPEG-Bitstroms zu definieren. Die Sequenz-Erweiterungs()-Funktion ist eine Funktion, Erweiterungsdaten der Sequenzebene des MPEG-Bitstroms zu definieren.

[0111] Eine do {}-While-Sentenz, welche im Anschluss an die Sequenz-Erweiterungs-Q-Funktion angeordnet ist, ist eine Sentenz, welche zeigt, dass, das Datenelement, welches auf Basis der Funktion in {} der do-Klausel beschrieben ist, im Codierdatenstrom beschrieben ist, während der Zustand, der durch die While-Klausel definiert ist, gilt. Eine Nächst-Bit()-Funktion, welche in die While-Klausel verwendet wird, ist eine Funktion, um ein Bit oder eine Bitgruppe, welche im Bitstrom beschrieben ist, mit einem Referenzdatenelement zu vergleichen. Im Beispiel der in Fig. 10 gezeigten Syntax vergleicht die Nächst-Bit()-Funktion eine Bitgruppe im Bitstrom mit dem Sequenz-End-Code, der das Ende der Videosequenz zeigt. Wenn die Bitgruppe im Bitstrom und der Sequenz-End-Code nicht miteinander übereinstimmen, gilt der Zustand der While-Klausel. Somit zeigt do {}-While-Sentenz, welche in Anschluss an die Sequenz-Erweiterungs()-Funktion angeordnet ist, dass das Datenelement, welches durch die Funktion in der do-Klausel definiert ist, im Codierbitstrom beschrieben ist, während Sequenz-End-Code, welcher das Ende der Videosequenz zeigt, im Bitstrom nicht erscheint.

[0112] Im Codierbitstrom folgt auf jedes Datenelement, welches durch die Sequenz-Erweiterungs()-Funktion definiert ist, ein Datenelement, welches definiert ist durch eine Erweiterungs- und Benutzerdaten(0)-Funktion. Diese Erweiterungs- und Benutzerdaten(0)-Funktion ist eine Funktion, um Erweiterungsdaten und Benutzerdaten in der Sequenzebene des MPEG-Bitstroms zu definieren.

[0113] Eine do {}-While-Sentenz, welche im Anschluss an die Erweiterungs- und Benutzerdaten(0)-Funktion angeordnet ist, ist eine Funktion, welche zeigt, dass das Datenelement, welches auf Basis der Funktion in {} der do-Klausel beschrieben ist, im Bitstrom beschrieben ist. Eine Nächst-Bit()-Funktion, welche in der While-Klausel verwendet wird, ist eine Funktion, um die Übereinstimmung zwischen einem Bit oder einer Bitgruppe, welche im Bitstrom erscheint, und dem Bildstartcode oder dem Gruppenstartcode zu bestimmen. Wenn das Bit oder die Bitgruppe, welche im Bitstrom auftritt, und der Bildstartcode oder der Gruppenstartcode miteinander übereinstimmen, gilt der Zustand, der durch die While-Klausel definiert ist. Daher zeigt diese do {}-While-Sentenz, dass, wenn der Bildstartcode oder Gruppenstartcode im Codierbitstrom erscheint, der Code des Datenelements, der durch die Funktion in

der do-Klausel definiert ist, im Anschluss an den Startcode beschrieben ist.

[0114] Eine Wenn-Klausel, die an der ersten Stelle dieser do-Klausel beschrieben ist, zeigt einen solchen Zustand, dass der Gruppenstartcode im Codierbitstrom erscheint. Wenn der Zustand dieser Wenn-Klausel gilt, werden Datenelemente, welche durch eine Gruppenbild-Kopf()-Funktion und eine Erweiterungs- und Benutzerdaten(1)-Funktion definiert sind, sequentiell im Anschluss an den Gruppenstartcode im Codierbitstrom beschrieben.

[0115] Diese Gruppenbild-Kopf()-Funktion ist eine Funktion, um Kopfdaten einer GOP-Ebene des MPEG-codierten Bitstroms zu definieren. Die Erweiterungs- und Benutzerdaten(1)-Funktion ist eine Funktion, um Erweiterungsdaten und Benutzerdaten der GOP-Ebene des MPEG-codierten Bitstroms zu definieren.

[0116] Außerdem folgen bei diesem Codierbitstrom den Datenelementen, welche durch die Gruppenbild-Kopf()-Funktion und die Erweiterungs- und Benutzerdaten(1)-Funktion definiert sind, Datenelemente, welche durch eine Bildkopf()-Funktion und eine Bildcodier-Erweiterungs()-Funktion definiert sind. Natürlich werden, wenn der oben beschriebene Zustand der Wenn-Klausel nicht gilt, die Datenelemente, welche durch die Gruppenbild-Kopf()-Funktion und die Erweiterungs- und Benutzerdaten(1)-Funktion definiert sind, beschrieben. Daher werden im Anschluss an das Datenelement, welches durch die Erweiterungs- und Benutzerdaten(0)-Funktion definiert ist, Datenelemente, welche durch die Bildkopf()-Funktion, die Bildcodier-Erweiterungs()-Funktion und die Benutzerdaten(2)-Funktion definiert sind, beschrieben.

[0117] Diese Bildkopf()-Funktion ist eine Funktion, um Kopfdaten einer Bildebene des MPEG-codierten Bitstroms zu definieren. Die Bildcodier-Erweiterungs()-Funktion ist eine Funktion, um erste Erweiterungsdaten der Bildebene des MPEG-codierten Bitstroms zu definieren. Die Erweiterungs- und Benutzerdaten(2)-Funktion ist eine Funktion, um Erweiterungsdaten und Benutzerdaten der Bildebene des MPEG-codierten Bitstroms zu definieren. Die Benutzerdaten, welche durch die Erweiterungs- und Benutzerdaten(2)-Funktion definiert sind, sind Daten, welche in der Bildebene beschrieben sind und für jedes Bild beschrieben werden können. Daher wird bei der vorliegenden Erfindung die Zeitcodeinformation als Benutzerdatenelement beschrieben, definiert durch die Erweiterungs- und Benutzerdaten(2)-Funktion.

[0118] Im codierten Bitstrom folgt auf die Benutzerdaten der Bildebene ein Datenelement, welches durch eine Bilddaten()-Funktion definiert ist. Diese Bilddaten/)-Funktion ist eine Funktion, um Datenelemente bezogen auf eine Slice-Ebene und einer Makroblockebene zu beschreiben.

[0119] Eine While-Klausel, der im Anschluss an die Bilddaten()-Funktion beschrieben wird, ist eine Funktion, um den Zustand der nächsten Wenn-Klausel zu bestimmen, während der Zustand, der durch diese While-Klausel beschrieben wird, gilt. Eine Nächst-Bit()-Funktion, welche in While-Klausel verwendet wird, ist eine Funktion, um zu bestimmen, ob der Bildstartcode oder der Gruppenstartcode in dem Codierstrom beschrieben ist oder nicht. Wenn der Bildstartcode oder der Gruppenstartcode im Bitstrom beschrieben ist, gilt der Zustand, der durch diese While-Klausel definiert ist.

[0120] Die nächste Wenn-Klausel ist eine Bedingungssentenz, um zu bestimmen, ob der Sequenz-End-Code im Codierbitstrom beschrieben ist oder nicht. Diese zeigt, dass, ob der Sequenz-End-Code nicht beschrieben ist, Datenelemente, welche durch eine Sequenzkopf()-Funktion und eine Sequenz-Erweiterungs()-Funktion definiert sind, beschrieben sind. Da der Sequenz-End-Code ein Code ist, der das Ende der Sequenz des Codiervideostroms zeigt, werden die Datenelemente, welche durch die Sequenzkopf()-Funktion und die Sequenz-Erweiterungs-()-Funktion definiert sind, beschrieben, wenn der Codierstrom nicht endet.

[0121] Die Datenelemente, welche durch die Sequenzkopf()-Funktion und die Sequenz-Erweiterungs()-Funktion definiert sind, sind perfekt die gleichen wie die Datenelemente, welche durch die Sequenzkopf()-Funktion und die Sequenz-Erweiterungs()-Funktion definiert sind, welche am Anfangsende der Videostromsequenz beschrieben sind. Die gleichen Daten werden somit im Strom aus dem folgenden Grund beschrieben. Das heißt, wenn die Bitstrom-Empfangseinrichtung mit dem Empfangen eines Punkts auf halben Wege des Datenstroms (beispielsweise eines Bitstrombereichs entsprechend der Bildebene) beginnt, können die Daten der Sequenzebene nicht empfangen werden und der Strom kann nicht decodiert werden. Der Grund zum Beschreiben der gleichen Daten ist, eine derartige Situation zu verhindern.

[0122] Im Anschluss an die Datenelemente, welche durch die letzte Sequenzkopf()-Funktion und die Sequenz-Erweiterungs()-Funktion definiert sind, d. h., am letzten Teil des Datenstroms, wird ein 32-Bit-Sequenz-End-Code, der das Ende der Sequenz zeigt, beschrieben.

[0123] Die Sequenzkopf()-Funktion, die Sequenz-Erweiterungs()-Funktion, die Erweiterungs- und Benutzerdaten()-Funktion, die Bildgruppe-Kopf()-Funktion und die Erweiterungs- und Benutzerdaten(1)-Funktion werden nun ausführlich be-

schrieben.

[0124] Fig. 11 zeigt die Syntax der Sequenzkopf()-Funktion. Die Datenelemente, welche durch die Sequenzkopf()-Funktion definiert sind, umfassen den Sequenzkopfcode, das Sequenzkopf-Augenblicks-Flag, den Horizontalgrößenwert, den Vertikalgrößenwert, die Bildseitenverhältnisinformation, den Rahmenratencode, den Bitratenwert, das Markerbit, den VBV-Puffergrößenwert, das Zwangs-Parameterflag, die Last-Intra-Quantisier-Matrix, die Intra-Quantisier-Matrix, den Nichtlast-Intra-Quantisier-Matrix und die Nicht-Intra-Quantisier-Matrix.

[0125] Der Sequenzkopfcode ist ein Datenelement, welches einen Startsynchronisationscode der Sequenzebene ausdrückt. Das Sequenzkopf-Augenblicks-Flag ist ein Datenelement, welches zeigt, ob die Daten im Sequenzkopf gültig sind oder nicht. Der Horizontalgrößenwert ist ein Datenelement, welches aus unteren 12 Bits einer Anzahl von Horizontalpixeln des Bilds besteht. Der Vertikalgrößenwert ist ein Datenelement, welches aus unteren 12 Bits der Anzahl vertikaler Zeilen des Bilds besteht. Die Bildseitenverhältnisinformation ist ein Datenelement, welches das Bildseitenverhältnis von Pixeln oder das Bildseitenverhältnis des Anzeigebildschirms ausdrückt. Der Rahmenratencode ist ein Datenelement, welches den Anzeigezyklus des Bilds zum Ausdruck bringt. Der Bitratenwert ist ein Datenelement unterer 18 Bits (nach oben für jeweils 400 bps aufgerundet) der Bitrate zur Begrenzung in Bezug auf die Menge erzeugter Bits. Der Marker-Bit ist ein Bitdatenelement, welches eingefügt wird, um Startcode-Emulation zu verhindern. Der Puffer VBV-Puffergrößenwert ist ein Datenelement unterer 10 Bits eines Werts, der die Größe des virtuellen Puffers (Videopufferverifizierer) bestimmt, um die Menge erzeugter Codes zu steuern. Das Zwangsparameterflag ist ein Datenelement, welches zeigt, dass jeder Parameter innerhalb der Grenze liegt. Die Last-Intra-Quantisier-Matrix ist ein Datenelement, welches die Existenz von Quantisierungs-Matrixdaten für Intra-MB zeigt. Die Intra-Quantisierungs-Matrix ist ein Datenelement, welches den Wert der Quantisiermatrixdaten für Intra-MB zeigt. Die Nichtlast-Intra-Quantisier-Matrix ist ein Datenelement, welches die Existenz von Quantisierungsmatrixdaten für Nicht-Intra-MB zeigt. Die Nicht-Intra-Quantisier-Matrix ist ein Datenelement, welches den Wert der Quantisiermatrix für Nicht-Intra-MB ausdrückt.

[0126] Fig. 12 zeigt die Syntax der Sequenz-Erweiterungs()-Funktion. Die Datenelemente, welche durch die Sequenz-Erweiterungs()-Funktion definiert sind, umfassen den Erweiterungsstartcode, den Erweiterungsstartcode-Identifizierer, das Sequenz-Erweiterungs-Augenblicks-Flag, die Profil- und Pegelanzeige, die progressive Sequenz, das Chroms-Format, die Horizontalgrößenerweiterung, die Vertikalgrößenerweiterung, die Bitratenerweiterung, die VBV-Puffergrößenerweiterung, Niedrig-Verzögerung, Rahmenratenerweiterung-n und die Rahmenratenerweiterung-d.

[0127] Der Erweiterungsstartcode ist ein Datenelement, welches das Ausdrücken eines Startsynchronisationscodes von Erweiterungsdaten zeigt. Der Erweiterungsstartcode-Identifizierer ist ein Datenelement, welches zeigt, welche Erweiterungsdaten zu übertragen sind. Das Sequenz-Erweiterungs-Augenblicks-Flag ist ein Datenelement, welches zeigt, ob Daten in der Sequenzerweiterung gültig sind oder nicht. Die Profil- und Pegelanzeige ist ein Datenelement zum Bestimmen des Profils und des Pegels der Videodaten. Die Progressivsequenz ist ein Datenelement, welches zeigt, dass die Videodaten sequentiell abgetastete Daten sind. Das Chroma-Format ist ein Datenelement, um das Farbdifferenzformat der Videodaten zu bestimmen. Die Horizontalgrößenerweiterung ist ein Datenelement von oberen zwei Bits, welches dem Horizontalgrößenwert des Sequenzkopfs hinzugefügt ist. Die Vertikalgrößenerweiterung ist ein Datenelement von oberen zwei Bits, welches dem Vertikalgrößenwert des Sequenzdatenkopfs hinzugefügt ist. Die Bitratenerweiterung ist ein Datenelement von oberen 12 Bits, welche dem Bitratenwert des Sequenzdatenkopfs hinzuzufügen sind. Die VBV-Puffergrößenerweiterung ist ein Datenelement von oberen 8 Bits, welche dem VBV-Puffergrößenwert des Sequenzkopfs hinzuzufügen sind. Die Niedrig-Verzögerung ist ein Datenelement, welches zeigt, dass kein B-Bild enthalten ist. Die Rahmenratenerweiterung-n ist ein Datenelement, um eine Rahmenrate in Kombination mit dem Rahmenratencode des Sequenzkopfs zu erlangen. Die Rahmenratenerweiterung-d ist ein Datenelement, um eine Rahmenrate in Kombination mit dem Rahmenratencode des Sequenzkopfs zu erlangen.

[0128] Fig. 13 zeigt die Syntax der Erweiterungs- und Benutzerdaten(i)-Funktion. Wenn (i) dieser Erweiterungs- und Benutzerdaten(i)-Funktion nicht 2 ist, wird lediglich das Datenelement, welches durch die Benutzerdaten()-Funktion definiert ist, beschrieben, ohne das Datenelement zu beschreiben, welches durch die Erweiterungsdaten()-Funktion definiert ist. Daher beschreibt die Erweiterungs- und Benutzerdaten(0)-Funktion lediglich das Datenelement, welches durch die Benutzerdaten()-Funktion definiert ist.

[0129] Fig. 14 zeigt die Syntax der Gruppe der Bildkopf()-Funktion. Die Datenelemente, welche durch die Gruppe der Bildkopf()-Funktion definiert sind, umfassen den Gruppenstartcode, das Bildgruppen-Kopf-Augenblicks-Flag, den Zeitcode, die geschlossene GOP und die unterbrochene Verknüpfung.

[0130] Der Gruppenstartcode ist ein Datenelement, welches einen Startsynchronisationscode der GOP-Ebene zeigt. Das Bildgruppen-Kopf-Augenblicks-Flag ist ein Datenelement, welches zeigt, ob die Datenelement im Bildgruppen-Kopf gültig ist oder nicht. Der Zeitcode ist ein Zeitcode, der die Zeit vom Anfang der Sequenz des Anfangsbilds der GOP zeigt. Die geschlossene GOP ist ein Flagdatenelement, welches zeigt, dass das Bild in der GOP unabhängig von der anderen GOP wiedergegeben werden kann. Die unterbrochene Verknüpfung ist ein Flagdatenelement, welches zeigt, dass das Anfangs-B-Bild in der GOP nicht genau wiedergegeben werden kann, um dies zu editieren und dgl..

[0131] Die Erweiterungs- und Benutzerdaten(1)-Funktion ist eine Funktion, um lediglich das Datenelement zu beschreiben, welches durch die Benutzerdaten()-Funktion definiert ist, ähnlich wie die Erweiterungs- und Benutzerdaten(0)-Funktion.

[0132] Unter Bezug auf Fig. 15 bis Fig. 17 werden nun die Bildkopf()-Funktion, die Bildcodier-Erweiterungs()-Funktion, die Erweiterungs- und Benutzerdaten(2)-Funktion und die Bilddaten()-Funktion zum Beschreiben der Datenelemente bezogen auf die Bildebene des Codierstroms beschrieben.

[0133] Fig. 15 zeigt die Syntax der Bildkopf()-Funktion. Die Datenelemente, welche durch diese Bildkopf()-Funktion definiert sind, umfassen den Bildstartcode, temporäre Referenz, die Bildcodierart, VBV-Verzögerung, Voll-Pel-Vorwärtsvektor, Vorwärts-f-Code, Voll-Pel-Rückwärtsvektor, Rückwärts-f-Code, Extra-Bit-Bild und Extra-Informations-Bild.

[0134] Insbesondere ist der Bildstartcode ein Datenelement, welches einen Startsynchronisationscode der Bildebene zum Ausdruck bringt. Die temporäre Referenz ist ein Datenelement, welches am Anfang der GOP mit einer Zahl, welche die Anzeigereihenfolge des Bilds zeigt, zurückgesetzt wird. Die Bildcodierart sind Daten, welche den Bildtypus zeigen.

[0135] Die VBV-Verzögerung ist ein Datenelement, welches den Anfangszustand des VBV-Puffers zeigt und für jedes Bild eingestellt wird. Das Bild des Codierelementarstroms, welches von dem übertragungsseitigen System auf das empfangsseitige System übertragen wird, wird durch den VBV-Puffer gepuffert, der auf Seiten des Empfangssystems vorgesehen ist. Das Bild wird vom VBV-Puffer herausgeführt (gelesen) und zum Decoder in dem Zeitpunkt geliefert, der durch den DTS (Decodierzeitstempel) bestimmt wird. Die Zeit, welche durch VBV-Verzögerung bestimmt wird, ist die Zeit, von der, wenn das Bild eines Decodierziels damit beginnt, vom VBV-Puffer herausgeführt zu werden, bis das Bild eines Codierziels vom VBV-Puffer herausgeführt ist, d. h., bis zu der Zeit, welche durch DTS bestimmt wird. In der Codierstrom-Spleißeinrichtung der vorliegenden Erfindung wird saumloses Spleißen, so dass die Datenbelegungsmenge im VBV-Puffer nicht unstetig wird, unter Verwendung der VBV-Verzögerung, welche im Bildkopf gespeichert ist, realisiert. Dieses Merkmal wird später ausführlich beschrieben.

[0136] Der Voll-Pel-Vorwärtsvektor ist ein Datenelement, welches zeigt, ob die Genauigkeit eines Vorwärtsbewegungsvektors auf einer ganzzahligen Einheit oder auf einer Halbpixeleinheit basiert. Der Vorwärts-f-Code ist ein Datenelement, welches den Vorwärtsbewegungsvektor-Suchbereich zum Ausdruck bringt. Der Voll-Pel-Rückwärtsvektor ist ein Datenelement, welches zeigt, ob die Genauigkeit eines Rückwärtsbewegungsvektors auf der ganzzahligen Einheit oder einer Halbpixeleinheit basieren. Der Rückwärts-f-Code ist ein Datenelement, welches den Rückwärtsbewegungsvektor-Suchbereich zum Ausdruck bringt. Das Extra-Bit-Bild ist ein Flag, welches die Existenz nachfolgender Zusatzinformation zeigt. Wenn dieses Extra-Bit-Bild gleich "1" ist, wird die Extra-Informations-Bild anschließend kommen. Wenn dieses Extra-Bit-Bild gleich "0" ist, folgen keine Daten. Die Extra-Bild-Information ist Information, welche gemäß dem Standard reserviert ist.

[0137] Fig. 16 zeigt die Syntax der Bildcodier-Erweiterungs()-Funktion. Die Datenelemente, welche durch diese Bildcodier-Erweiterungs()-Funktion definiert sind, umfassen den Erweiterungsstartcode, den Erweiterungsstart-Codeidentifizierer, den f-Code [0][0], den f-Code [0][1], den f-Code [1][0], den f-Code [1][], die Intra-dc-Präzisierung, die Bildstruktur, das erste Kopffeld, den Rahmenvorhersagerahmen-dct, Verbergungsbewegungsvektoren, q-Skalenart, Intra-VLC-Format, Alternativ-Abtastung, erstes Wiederholungsfeld, Chroms-420-Typus, Progressiv-Rahmen, zusammengesetztes Anzeigeflag, v-Achse, Feldsequenz, Hilfsträger, Burst-Amplitude und Hilfsträgerphase.

[0138] Der Erweiterungsstandcode ist ein Startcode, der den Start von Erweiterungsdaten der Bildebene zeigt. Der Erweiterungsstartcode-Identifizierer ist ein Code, der zeigt, welche Erweiterungsdaten zu übertragen sind. Der f-Code[0][0] ist ein Datenelement, welches den Suchbereich für einen Vorwärtshorizontal-Bewegungsvektor zum Ausdruck bringt. Der f-Code [0][1] ist ein Datenelement, welches den Suchbereich des vertikalen Vorwärtsbewegungsvektors zum Ausdruck bringt. Der f-Code [1][0] ist ein Datenelement, welches den Suchbereich eines Horizontalrückwärtsbewegungsvektors zeigt. Der f-Code [1][1] ist ein Datenelement, welches den Suchbereich des vertikalen Rückwärtsbewegungsvektors zum Ausdruck bringt. Die Intra-dc-Präzision ist ein Datenelement, welches die Präzision eines DC-Koeffizienten zum Ausdruck bringt. Die Bildstruktur ist ein Da-

tenelement, welches zeigt, ob die Bildstruktur eine Rahmenstruktur oder eine Feldstruktur ist. Im Fall der Feldstruktur zeigt dies auch, ob das Feld ein oberes Feld oder ein unteres Feld ist.

[0139] Das erste Kopffeld ist ein Flag, welches zeigt, ob das erste Feld ein Kopffeld oder ein Bodenfeld im Fall der Rahmenstruktur ist. Die Rahmenvorhersage-Rahmen-DCT ist ein Datenelement, das zeigt, dass die Vorhersage der Rahmenmodus-DCT lediglich der Rahmenmodus im Fall der Rahmenstruktur ist. Die Verbergungsbewegungsvektoren sind Daten, welche zeigen, dass ein Bewegungsvektor zum Verbergen eines Übertragungsfehlers im Intra-Makroblock vorgesehen ist. Der q-Skalen-Typus ist ein Datenelement, welches zeigt, ob eine lineare Quantisierungsskala oder eine nichtlineare Quantisierungsskala verwendet wird. Das Intra-VLC-Format ist ein Datenelement, welches zeigt, ob ein anderer zweidimensionaler VLC für den Intra-Makroblock zu verwenden ist. Die Alternativ-Abtastung ist ein Datenelement, welches die Auswahl zeigt, ob eine Zickzack-Abtastung oder eine abwechselnde Abtastung verwendet wird.

[0140] Das erste Wiederholungsfeld ist ein Flag, welches zeigt, ob ein Wiederholungsfeld im Zeitpunkt der Decodierung zu erzeugen wird oder nicht. Bei einer Verarbeitung im Decodierzeitpunkt wird, wenn dieses erste Wiederholungsfeld gleich "1" ist, ein Wiederholungsfeld erzeugt. Wenn dieses Wiederholungsfeld gleich "0" ist, wird ein Wiederholungsfeld nicht erzeugt. Der Chroms-420-Typus ist ein Datenelement, welches den gleichen Wert wie den nächsten Progressivrahmen in dem Fall zeigt, wo das Signalformat gleich 4:2:0 ist, und 0 in dem Fall zeigt, wo das Signalformat nicht 4:2:0 ist. Der Progressivrahmen ist ein Datenelement, das zeigt ob dieses Bild sequentiell abgetastet ist oder nicht. Das zusammengesetzte Anzeigeflag ist ein Datenelement, das zeigt, ob ein Quellensignal ein zusammengesetztes Signal ist oder nicht. Die v-Achse ist ein Datenelement, welches in dem Fall verwendet wird, wo das Quellensignal vom PAL-System ist. Die Feldsequenz ist ein Datenelement, welches in dem Fall verwendet wird, wo das Quellensignal das vom PAL-System ist. Der Hilfsträger ist ein Datenelement, welches in dem Fall verwendet werden, wo das Quellensignal vom PAL-System ist. Die Burst-Amplitude ist ein Datenelement, welches in dem Fall verwendet wird, wenn das Quellensignal vom PAL-System ist. Die Hilfsträgerphase Daten, welche in dem Fall verwendet werden, wo das Quellensignal vom PAL-System ist.

[0141] Die Erweiterungs- und Benutzerdaten(2)-Funktion beschreibt das Datenelement, welches durch die Erweiterungsdaten()-Funktion definiert ist, wenn der Erweiterungsstartcode im codierten Bitstrom existiert, wie in Fig. 13 gezeigt ist. Wenn jedoch der Erweiterungsstartcode im Bitstrom nicht existiert, wird das Datenelement, welches durch die Erweiterungsdaten()-Funktion definiert wird, im Bitstrom nicht beschrieben. Wenn der Benutzerdatenstartcode im Bitstrom existiert, wird das Datenelement, welches durch die Benutzerdaten()-Funktion definiert ist, im Anschluss an das Datenelement, welches durch die Erweiterungsdaten()-Funktion definiert ist, beschrieben.

[0142] Fig. 17 zeigt die Syntax der Bilddaten()-Funktion. Das Datenelement, welches durch diese Bilddaten()-Funktion definiert ist, ist ein Datenelement, welches durch eine Slice()-Funktion definiert ist. Wenn der Slice-Startcode, welcher den Startcode der Slice()-Funktion zeigt, jedoch nicht im Bitstrom existiert, wird das Datenelement, welches durch die Slice()-Funktion definiert ist, im Bitstrom nicht beschrieben.

[0143] Die Slice()-Funktion ist eine Funktion, um Datenelemente bezogen auf die Slice-Ebene zu beschreiben. Insbesondere ist sie eine Funktion, um Datenelemente, beispielsweise den Slice-Startcode, den Slice-Quantisier-Skalencode, das Intra-Slice-Flag, Intra-Slice, Reserve-Bits, Extra-Bits-Slice, Extra-Informations-Slice und Extra-Bits-Slice, und Datenelemente zu beschreiben, welche durch eine Makroblock()-Funktion definiert sind.

[0144] Der Slice-Startcode ist der Startcode, der den Start der Datenelemente zeigt, welche durch die Slice()-Funktion definiert sind. Der Slice-Quantisier-Skalencode ist ein Datenelement, welches die Quantisierungsschrittgröße zeigt, welche für einen Makroblock gesetzt wird, welcher in dieser Slice-Ebene existiert. Wenn jedoch der Quantisier-Skalencode für jeden Makroblock gesetzt wird, werden die Daten des Makroblock-Quantisier-Skalencodes, welcher für jeden Makroblock gesetzt sind, vorzugsweise verwendet. Das Intra-Slice-Flag ist ein Flag, welches zeigt, ob das Intra-Slice und Reserve-Bits im Bitstrom existieren oder nicht. Das Intra-Slice ist ein Datenelement, welches zeigen, ob ein Nicht-Intra-Makroblock in der Slice-Ebene existiert oder nicht. Wenn einer der Makroblöcke in der Slice-Ebene ein Nicht-Intra-Makroblock ist, ist das Intra-Slice gleich "0". Wenn alle Makroblöcke in der Slice-Ebene Nicht-Intra-Makroblicke sind, ist das Intra-Slice gleich "1". Die Reserve-Bits sind 7-Bit-Daten und haben einen Wert "0". Das Extra-Bit-Slice ist ein Flag, welches zeigt, dass Zusatzinformation als Codierstrom existiert. Wenn das Extra-Informations-Slice im Anschluss an das Extra-Bit-Slice existiert, wird dieses Flag auf "1" gesetzt. Wenn keine Zusatzinformation existiert, wird das Flag auf "0" gesetzt.

[0145] Die Makroblock()-Funktion ist eine Funktion, um Datenelemente bezogen auf die Makroblockebene zu beschreiben. Insbesondere ist sie eine Funkti-

on, um Datenelemente zu beschreiben, beispielsweise Makroblock-Codeumschaltung, Makroblock-Adress-Inkrement und Makroblock-Quantisier-Skalencode, und Datenelemente, welche durch einen Makroblock-Modus()-Funktion und eine Makroblock-Vektor(Vektoren)-Funktion definiert sind.

[0146] Die Makroblock-Codeumschaltung ist eine feste Bitgruppe, welche zeigt, ob die horizontale Differenz zwischen einem Referenz-Makroblock und einem vorherigen Makroblock nicht kleiner ist als 34. Wenn die horizontale Differenz zwischen dem Referenz-Makroblock und dem vorherigen Makroblock nicht kleiner ist als 34, wird 33 dem Wert des Makroblock-Adress-Inkrements hinzugefügt. Das Makroblock-Adress-Inkrement ist ein Datenelement, welches die Horizontaldifferenz zwischen dem Referenz-Makroblock und dem vorherigen Makroblock zeigt. Wenn eine Makroblock-Codeumschaltung vor diesem Makroblock-Adress-Inkrement existiert, wird der Wert, der durch Hinzufügen von 33 zum Wert des Makroblock-Adress-Inkrements erlangt wird, zu einem Datenelement, welches die aktuelle horizontale Differenz zwischen dem Referenz-Makroblock und dem vorherigen Makroblock zeigt. Der Makroblock-Quantisier-Skalencode zeigt die Quantisierungsschrittgröße, welche für jeden Makroblock festgelegt ist. In jeder Slice-Ebene wird der Slice-Quantisier-Skalencode, welcher die Quantisierungsschrittgröße der Slice-Ebene zeigt, festgelegt. Wenn jedoch der Makroblock-Quantisier-Skalencode für den Referenzmakroblock festgelegt wird, wird diese Quantisierungsschrittgröße ausgewählt.

[0147] Unter Bezug auf Fig. 18 und Fig. 19 wird der Aufbau des Transportstrompakets und die Syntax des Transportstrompakets ausführlich beschrieben.

[0148] Das Transportstrompaket besteht aus einem 4-Byte-Kopf und einem 184-Byte-Nutzinformationsteil, um verschiedene Daten und Datenelemente zu speichern.

[0149] Das Kopfteil des Transportstrompakets umfasst verschiedene Felder, beispielsweise Synchronisationsbyte, Transportfehlerindikator, Nutzinformationseinheits-Startindikator, Transportpriorität, PID, Transport-Verwürfelung, Steuerung, Adaption, Fehlsteuerung, Kontinuitätszähler und Adaptionsfeld.

[0150] Das Synchronisationsbyte ist ein festes 8-Bit-Feld, um ein Synchronisationsmuster von dem Bitstrom zu ermitteln. Dessen Wert wird durch einen festen Wert aus "01000111" (0×47) definiert. Durch Ermitteln dieses Bitmusters im Strom kann die Synchronisation ermittelt werden.

[0151] Der Transportfehlerindikator ist ein 1-Bit-Flag. Wenn dieses Flag auf "1" gesetzt wird, zeigt dies, dass ein nicht korrigierbarer Bitfehler von zumindest einem Bit im Transportstrompaket existiert.

[0152] Der Nutzinformations-Einheitsstart-Indikator ist ein 1-Bit-Flag. Dies ist ein Datenelement, welches vorgeschriebene Bedeutungen für Elementardaten hat, beispielsweise für Video-/Audiodaten oder ein Transportstrompaket zum Übertragen von Programmspezifikationsinformation (PSI). Wenn die Nutzinformation des Transportstrompakets Elementardaten aufweist, hat der Nutzinformations-Einheitsstart-Indikator die nachfolgenden Bedeutungen. Wenn der Nutzinformations-Einheitsstart-Indikator gleich "1" ist, zeigt dies, dass Elementardaten am Anfang der Nutzinformation des Transportstrompakets eingefügt sind. Wenn der Nutzinformation-Einheitsstart-Indikator gleich "0" ist, sind Elementardaten am Anfang der Nutzinformation des Transportstrompakets nicht eingefügt. Wenn der Nutzinformation-Einheitsstart-Indikator auf "1" gesetzt ist, zeigt dies, dass das lediglich ein PSI-Paket mit einem willkürlichen Transportstrompaket beginnt. Wenn dagegen die Nutzinformation des Transportstroms PSI-Daten aufweist, hat der Nutzinformations-Einheitsstart-Indikator die folgenden Bedeutungen. Wenn das Transportstrompaket das erste Byte des PSI-Abschnitts überträgt, wird der Nutzinformation-Einheitsstart-Indikator zu "1". Wenn das Transportstrompaket das erste Byte des PSI-Abschnitts nicht überträgt, wird der Nutzinformations-Einheitsstart-Indikator zu "0". Wenn außerdem das Transportstrompaket ein Nullpaket ist, wird der Nutzinformation-Einheitsstart-Indikator zu "0".

[0153] Die Transportpriorität ist ein 1-Bit-Identifizierer, der die Priorität des Transportstrompakets zeigt. Wenn die Transportpriorität auf "1" gesetzt ist, zeigt dies, dass dieses Transportstrompaket Priorität über einem Paket hat, welches die gleiche Paketidentifizierer-PDI hat, und die Transportpriorität hat, welche nicht "1" ist. Durch Setzen dieses Paketidentifizierer des Transportpriorität kann beispielsweise Priorität einem beliebigen Paket in einem Elementarstrom gegeben werden.

[0154] Die Transportverwürfelungssteuerung sind 2-Bit-Daten, welche den Verwürfelungsmodus des Transportstrompakets zeigen. Der Verwürfelungsmodus ist ein Modus, um zu zeigen, ob Daten, welche im Nutzinformationsteil gespeichert sind, verwürfelt sind oder nicht, sowie die Art der Verwürfelung. Der Transportstrom-Paketkopf und das Adaptionsfeld sind standardisiert, die durch einen Verwürfelungsschlüssel Ks nicht verwürfelt werden. Somit kann durch diese Transportverwürfelungssteuerung bestimmt werden, ob Daten, welche im Nutzinformationsteil des Transportstrompakets gespeichert sind, verwürfelt sind oder nicht.

[0155] Die Adaptionsfeldsteuerung ist ein 2-Bit-Da-

tenelement, welches zeigt, dass das Adaptionsfeld und/oder die Nutzinformation im Paketkopf dieses Transportstrompakets angeordnet sind. Wenn insbesondere lediglich die Nutzinformationsdaten im Paketkopf angeordnet sind, wird diese Adaptionsfeldsteuerung zu "01". Wenn lediglich das Adaptionsfeld im Paketkopf angeordnet ist, wird diese Adaptionsfeldsteuerung zu "10". Wenn das Adaptionsfeld und die Nutzinformation im Paketkopf angeordnet sind, wird diese Adaptionsfeldsteuerung zu "11".

[0156] Der Kontinuitätszählwert ist ein Datenelement, welches zeigt, ob ein Teil der Pakete, welche die gleiche PDI haben, welche stetig übertragen werden, verloren ist oder ausgegeben ist. Insbesondere ist der Kontinuitätszählwert ein 4-Bit-Feld, welches für jedes Transportstrompaket ansteigt, welches die gleiche PDI hat. Wenn jedoch dieser Kontinuitätszählwert gezählt wird, ist das Adaptionsfeld im Paketkopf angeordnet.

[0157] Das Adaptionsfeld() ist ein Feld, um Zusatzinformation bezogen auf einen Individualstrom oder Stopfbytes als Option einzufügen. Durch dieses Adaptionsfeld kann die gesamte Information bezogen auf dynamische Änderungen des Zustands eines Individualstroms zusammen mit den Daten übertragen werden.

[0158] Fig. 19 zeigt die Syntax des Adaptionsfelds(). Dieses Adaptionsfeld() weist verschiedene Felder auf, beispielsweise die Adaptionsfeldlänge, den Diskontinuitätszähler, den Zufallszugriffsindikator, den Elementarstrom-Prioritätsindikator, das OPCR-Flag, das Spleißpunkt-Flag, das Transport-Privatdaten-Flag, das Adaptionsfeld-Erweiterungsflag, die Programmtaktreferenz (PCR), die Ursprungsprogramm-Taktreferenz (OPCR), die abschließende Spleiß-Vorbereitung, die Transport-Privat-Datenlänge, die Privatdaten, die Adaptionsfeld-Erweiterungslänge, das LTW-Flag (Legalzeit-Fensterflag), das Stückraten-Flag und das Saumlos-Spleiß-Flag.

[0159] Die Adaptionsfeldlänge ist ein Datenelement, welches die Anzahl von Bytes eines Adaptionsfelds im Anschluss an diese Adaptionsfeldlänge zeigt. Wenn die Adaptionsfeldsteuerung gleich "11" ist, beträgt die Adaptionsfeldlänge 0 bis 182 Bits. Wenn die Adaptionsfeldsteuerung gleich "10" ist, ist die Adaptionsfeldlänge gleich 183 Bits. Wenn der Elementarstrom, der ausreichend ist, die Nutzinformation des Transportstroms zu füllen, nicht existiert, ist die Stopfverarbeitung für Füllbits erforderlich.

[0160] Der Diskontinuitätszählwert ist ein Datenelement, das zeigt, ob eine Systemtaktreferenz (SCR) in einem Bereich auf halbem Weg von mehreren Paketen zurückgesetzt ist oder nicht, welche die gleiche PID haben und daher nicht stetig wurde. Wenn die Systemtaktreferenz nicht stetig ist, ist dieser Diskontinuitätszählwert gleich "1". Wenn die Systemtaktreferenz stetig ist, ist dieser Diskontinuitätszählwert gleich "0". Diese Systemtaktreferenz ist Referenzinformation, um den Wert des Systemzeittakts auf Seiten des Decoders auf einen Zeitablauf, der auf der Codiererseite beabsichtigt ist, im MPEG-Decoder einzustellen, um Video- und Audiodaten zu decodieren.

[0161] Der Zufallszugriffs-Indikator ist ein Datenelement, welches den Beginn eines Videosequenzkopfes oder eines Audiorahmens zeigt. Kurz ausgedrückt ist dieser Zufallszugriffs-Indikator ein Datenelement, welches einen Video- oder Audiozugriffspunkt (den Start eines Rahmens) beim Ausführen eines Zufallszugriffs auf die Datenelemente zeigt.

[0162] Der Elementarstrom-Prioritäts-Indikator ist ein Datenelement, welches die Priorität der Elementarstromdaten zeigt, welche in der Nutzinformation dieses Transportstrompakets zu übertragen sind, in Bezug auf die Pakete, welche die gleiche PID haben. Wenn beispielsweise der Elementarstrom Videodaten aufweist, welche intra-codiert sind, wird der Elementarstrom-Prioritäts-Indikator auf "1" gesetzt. Dagegen wird der Elementarstrom-Prioritäts-Indikator des Transportstroms, der intercodierte Videodaten aufweist, auf "0" gesetzt.

[0163] Das PCR-Flag ist ein Datenelement, das zeigt, ob PCR-Daten (Programmtaktreferenz-Daten) im Adaptionsfeld existieren oder nicht. Wenn PCR-Daten im Adaptionsfeld existieren, wird das PCR-Flag auf "1" gesetzt. Wenn PCR-Daten im Adaptionsfeld nicht existieren, wird das PCR-Flag auf "0" gesetzt. Diese PCR-Daten sind Daten, die verwendet werden, um den Zeitablauf der Decodierverarbeitung übertragener Daten im Decoder der Empfangseinheit zu erlangen.

[0164] Das OPCR-Flag ist ein Datenelement, das zeigt, ob OPCR-Daten (Ursprungsprogramm-Taktreferenz-Daten) im Adaptionsfeld existieren oder nicht. Wenn OPCR-Daten im Adaptionsfeld existieren, wird das OPCR-Flag auf "1" gesetzt. Wenn OPCR-Daten im Adaptionsfeld nicht existieren, wird das OPCR-Flag auf "0" gesetzt. Diese OPCR-Daten sind Daten, die verwendet werden, wenn ein Transportstrom von mehreren Ursprungstransportströmen durch Spleißverarbeitung rekonstruiert wird. Diese OPCR-Daten sind Daten, welche PCR-Daten eines bestimmten ursprünglichen Transportstroms zum Ausdruck bringen.

[0165] Das Spleißpunkt-Flag ist ein Datenelement, welches zeigt, ob Spleiß-Vorbereitungen zum Anzeigen eines Editierpunkts (Spleißpunkt) beim Transportpegel im Adaptionsfeld existieren oder nicht. Wenn Spleiß-Vorbereitungen im Adaptionsfeld existieren, ist dieses Spleißpunkt-Flag gleich "1". Wenn

Spleiß-Vorbereitungen im Adaptionsfeld nicht existieren, ist dieses Spleißpunkt-Flag gleich "0".

[0166] Das Transport-Privatdaten-Flag ist ein Datenelement, das zeigt, ob Privatdaten zum Beschreiben von beliebigen Benutzerdaten im Adaptionsfeld existieren oder nicht. Wenn Privatdaten im Adaptionsfeld existieren, wird dieses Transport-Privatdaten-Flag auf "1" gesetzt. Wenn keine Privatdaten im Adaptionsfeld existieren, wird dieses Transport-Privatdaten-Flag auf "0" gesetzt.

[0167] Dieses Adaptionsfeld-Erweiterungsflag ist ein Datenelement, das zeigt, ob ein Erweiterungsfeld im Adaptionsfeld existiert oder nicht. Wenn ein Erweiterungsfeld im Adaptionsfeld existiert, wird dieses Adaptionsfeld-Erweiterungsflag auf "1" gesetzt. Wenn kein Erweiterungsflag im Adaptionsfeld existiert, wird dieses Adaptionsfeld-Erweiterungsflag auf "0" gesetzt.

[0168] Die Programmtaktreferenz (PCR) ist ein Referenztakt, der darauf bezogen ist, um die Phase des Takts der Empfangsseite mit der Phase des Takts auf der Übertragungsseite zu synchronisieren. Diese PCR-Daten speichern die Zeit, wenn das Transportpaket erzeugt wird. Die PCR-Daten bestehen aus der 33-Bit-Programmtakt-Referenzbasis und der 9-Bit-Programmtakt-Referenzerweiterung, d. h., aus 42 Bits insgesamt. Der Systemtakt wird von 0 bis 299 durch die Programmtakt-Referenzerweiterung gezählt, und ein Bit wird der Programmtakt-Referenzbasis durch einen Träger beim Rücksetzen von 299 auf 0 hinzugefügt. Somit können 24 Stunden gezählt werden.

[0169] Die Ursprungsprogramm-Taktreferenz (OPCR) ist ein Datenelement, welches zum Rekonstruieren eines Transportstroms eines einzelnen Programms von einem bestimmten Transportstrom verwendet wird. Wenn ein einzelner Programmtransportstrom vollständig rekonstruiert ist, wird diese Ursprungsprogramm-Taktreferenz in die Programmtaktreferenz kopiert.

[0170] Die Spleiß-Vorbereitungen sind Daten, welche die Anzahl von Paketen bis zu einem Punkt zeigen, wo das Editieren (Spleiß-Verarbeitung) auf dem Transportstrom-Paketpegel ausgeführt werden kann, in Bezug auf Transportstrompakete, welche die gleiche PD haben. Daher ist in einem Transportstrompaket an einem Spleißpunkt, wo das Editieren ausgeführt werden kann, die Spleiß-Vorbereitung gleich "0". Im Transportpaket, welches die Spleiß-Vorbereitung von "0" hat, kann die Spleiß-Verarbeitung ausgeführt werden, indem veranlasst wird, dass das letzte Byte die Transportstrompaket-Nutzinformation das letzte Byte des codierten Bilds ist.

[0171] Diese Spleiß-Verarbeitung ist die Verarbeitung, um zwei verschiedene Elementarströme beim Transportpegel miteinander zu verbinden, um somit einen neuen Transportstrom zu erzeugen. Die Spleiß-Verarbeitung kann in saumloses Spleißen klassifiziert werden, welche keine Diskontinuität der Codes erzeugt, und nichtsaumloses Spleißen, welches Diskontinuität der Codes erzeugt. "Nichterzeugung von Diskontinuitätscodes" bedeutet das Nichtvorhandensein eines Widerspruchs zwischen der Decodierzeit einer Zugriffseinheit eines neu verbundenen nachfolgenden Stroms und der Decodierzeit einer Zugriffseinheit eines alten Stroms vor dem Spleißen. "Erzeugen von Diskontinuität von Codes" bedeutet die Erzeugung eines Widerspruchs zwischen der Decodierzeit einer Zugriffseinheit eines neu verbundenen nachfolgenden Stroms und der Decodierzeit einer Zugriffseinheit eines alten Stroms vor dem Spleißen.

[0172] Die Transport-Privatdatenlänge ist ein Datenelement, welches die Anzahl von Bytes von Privatdaten im Adaptionsfeld zeigt.

[0173] Das Privatdatenelement ist ein Feld, welches durch den Standard nicht besonders vorgeschrieben ist und kann beliebige Benutzerdaten im Adaptionsfeld beschreiben.

[0174] Die Adaptionsfeld-Erweiterungslänge ist ein Datenelement, welches die Datenlange der Adaptionsfeld-Erweiterung im Adaptionsfeld zeigt.

[0175] Das LTW-Flag (Legalzeit-Fensterflag) ist ein Datenelement, welches zeigt, ob ein LTW-Offset, der den Offsetwert eines Anzeigefensters zeigt, im Adaptionsfeld existiert oder nicht.

[0176] Das Stückweis-Ratenflag ist ein Datenelement, welches zeigt, ob eine stückweise Rate im Adaptionsfeld existiert oder nicht.

[0177] Das Saumlos-Spleißflag ist ein Datenelement, welches zeigt, ob der Spleißpunkt ein normaler Spleißpunkt oder ein saumloser Spleißpunkt ist. Wenn dieses Saumlos-Spleißflag gleich "0" ist, zeigt dies, dass der Spleißpunkt ein normaler Spleißpunkt ist. Wenn dieses Saumlos-Spleißflag gleich "1" ist, zeigt dies, dass der Spleißpunkt ein saumloser Spleißpunkt ist. Der normale Spleißpunkt existiert am Ende des PES-Pakets. Das Spleißpaket unmittelbar vor diesem Spleißpunkt endet mit einer Zugriffseinheit, und ein Transportpaket, welches die gleiche PD hat, beginnt am Kopf des PES-Pakets. Dagegen existiert der saumlose Spleißpunkt an einem Punkt auf halben Weg im PES-Paket. Um einen Widerspruch zwischen der Decodierzeit einer Zugriffseinheit eines neu verbundenen nachfolgenden Stroms und der Decodierzeit einer Zugriffseinheit eines alten Stroms vor dem Spleißen zu verhindern, wird ein Teil der Kenndaten des alten Stroms als Kenndaten des

neuen Stroms verwendet.

[0178] Die Spleiß-Verarbeitung zum Spleißen eines Stroms ST_{OLD}, der von der Schlüsselstation **30** übertragen wird, und eines Stroms ST_{NEW}, der in der örtlichen Station **40** erzeugt wird, wird nun mit Hilfe von Fig. 20 bis 23 beschrieben.

[0179] In Fig. 20 ist lediglich ein Kanal von mehreren Kanälen gezeigt, und die anderen Kanäle sind weggelassen, um die Steuerung der örtlichen Station **40**, welche in Fig. 7 beschrieben wurde, zu verdeutlichen. Bei der vorliegenden Erfindung sind drei Ausführungsformen einer Spleiß-Verarbeitung vorgesehen. Die erste, zweite und dritte Ausführungsform der Spleiß-Verarbeitung wird nun der Reihe nach beschrieben.

[0180] Bei der ersten Ausführungsform der Spleiß-Verarbeitung wird die Spleiß-Verarbeitung in dem Fall ausgeführt, wo ein Codierstrom ST_{NEW} der Werbung CM', der neu einzufügen ist, schon erzeugt wurde, bevor ein Codierstrom ST_{OLD} des Übertragungsprogramms von der Schlüsselstation **30** übertragen wird. Das heißt, der Strom der Werbung CM1', der vorher codiert wurde, wird in den Codierstrom ST_{OLD} der Werbung CM des Übertragungsprogramms eingefügt. Normalerweise wird eine Werbung wiederholt mit einer bestimmten Häufigkeit gesendet. Daher ist es nicht wirksam, die Videodaten der Werbung zu codieren, jedes Mal, wenn diese gesendet werden. Somit werden die Videodaten der örtlichen Werbung CM1' codiert, und dessen Codierstrom ST_{NEW} wird im Stromserver **49** gespeichert. Wenn dann der Codierstrom ST_{OLD} der Werbung CM1, der zu ersetzen ist, von der Schlüsselstation **30** übertragen wird, wird der Codierstrom ST_{NEW} der örtlichen Werbung CM1' vom Stromserver **49** wiedergegeben, wodurch die Verarbeitung zum wiederholten Codieren der gleichen Werbung mit einer bestimmten Häufigkeit wegfällt. In einem solchen Fall wird die erste Spleiß-Verarbeitung durchgeführt, wie ausführlich anschließend beschrieben wird.

[0181] Zunächst wird die Anfangsverarbeitung in der örtlichen Station **40** zum Codieren der örtlichen Werbung C1', um die Werbung CM1 des Übertragungsprogramms zu ersetzen, und zum Speichern des Codierstroms ST_{NEW} im Stromserver **49** beschrieben. Die Sendesystemsteuerung **41** steuert den CM-Server **47**, um die Videodaten der Werbung CM1' wiederzugeben, um die Werbung CM1 des Übertragungsprogramms zu ersetzen. Dann empfängt ein Codierer **481** die Basisband-Videodaten, welche vom CM-Server **47** wiedergegeben werden, und liefert die Codierschwierigkeit Di jedes Bilds dieser Videodaten zu einer Codierersteuerung **480**. Die Codierersteuerung **480** liefert ähnlich wie die Codierersteuerung **350**, welche in Fig. 8 beschrieben wurde, eine Zielbitrate Ri zum Codierer **481**, so dass der Codierer **481** passende codierte Bits erzeugt. Der Codierer **481** führt die Codierverarbeitung auf Basis der Zielbitrate Ri aus, welche von der Codierersteuerung **480** geliefert wird, und kann somit den codierten Elementarstrom ST_{NEW} der optimalen Bitrate erzeugen. Der codierte Elementarstrom ST_{NEW}, der vom Codierer **481** ausgegeben wird, wird zum Stromserver **49** geliefert. Der Stromserver **49** zeichnet den codierten Elementarstrom in dem Zustand des Stroms auf einem zufallsmäßig zugreifbaren Aufzeichnungsmedium auf. Damit ist die Anfangsverarbeitung zum Speichern des Codierstroms ST_{NEW} im Stromserver **49** beendet.

[0182] Anschließend wird die Spleiß-Verarbeitung zum Spleißen des Codierstroms ST_{OLD} des Übertragungsprogramms, welches von der Schlüsselstation übertragen wird, und des Codierstroms ST_{NEW}, der im Stromserver **49** durch die oben beschriebene Anfangsverarbeitung gespeichert ist, beschrieben.

[0183] Der Codierstrom ST_{OLD}, der von der Schlüsselstation **30** übertragen wird, wird in der Form des Transportstroms in die Form des Elementarstroms durch die Stromumsetzungsschaltung **44** umgesetzt. Der Codierstrom ST_{OLD}, welcher in die Form des Elementarstroms umgesetzt ist, wird zum Strom-Spleißer **50** geliefert.

[0184] Der Strom-Spleißer **50** weist eine Spleiß-Steuerung **500**, eine Umschaltschaltung **501**, eine Stromanalyseschaltung **502**, einen Stromprozessor **503** und eine Spleiß-Schaltung **504** auf, wie in Fig. 20 gezeigt ist.

[0185] Bei der ersten Ausführungsform der Spleiß-Verarbeitung schaltet die Spleiß-Steuerung **500** einen Eingangsanschluss der Umschaltschaltung **501** auf "a" um, um den Elementarstrom ST_{NEW}, der vom Stromserver **49** geliefert wird, zur Stromanalyseschaltung **502** zu liefern.

[0186] Die Stromanalyseschaltung **502** ist eine Schaltung, um die Syntax des Codierstroms ST_{OLD} und des Codierstroms ST_{NEW} zu zergliedern und zu analysieren. Insbesondere sucht die Stromanalyseschaltung **502** den 32-Bit-Bildstartcode, der im Codierstrom ST_{OLD} beschrieben ist, um dadurch die Stelle im Strom zu erlernen, wo Information bezogen auf den Bildkopf beschrieben wurde, wie aus der Syntax des Codierstroms, der in Fig. 10 und Fig. 15 gezeigt ist, verstanden wird. Die Stromanalyseschaltung **502** findet dann die 3-Bit-Bildcodierart, welche 11 Bits nach dem Bildstartcode beginnt, wodurch die Bildart erlernt wird. Außerdem erlernt die Stromanalyseschaltung **502** die VBV-Verzögerung des codierten Bilds von der 16-Bit-VBV-Verzögerung, welche im Anschluss an diese 3-Bit-Bildcodierart beschrieben ist.

[0187] Außerdem sucht die Stromanalyseschaltung

502 den 32-Bit-Erweiterungsstartcode, der im Codierstrom ST_{OLD} beschrieben ist, und den Codierstrom ST_{NEW}, um dadurch die Stelle im Strom zu erlernen, wo die Information bezogen auf die Bildcodeerweiterung beschrieben ist, wie aus der Syntax des Codierstroms verstanden wird, welche in Fig. 10 und Fig. 15 gezeigt ist. Dann sucht die Stromanalyseschaltung **502** das erste 1-Bit-Kopffeld, welches 25 Bits nach dem Bildstartcode beschrieben ist, und das erste Wiederholungsfeld, welches 6-Bits nach dem ersten Kopffeld beschrieben ist, um dadurch die Rahmenstruktur des codierten Bilds zu erlernen. Wenn beispielsweise das "erste Kopffeld" des codierten Bilds gleich "1" ist, zeigt dies eine Rahmenstruktur, bei der das Kopffeld zeitlich früher ist als das Bodenfeld. Wenn "das erste Kopffeld" gleich "0" ist, zeigt dies eine Rahmenstruktur, bei der das Kopffeld zeitlich später ist als das Bodenfeld. Wenn das Flag "erstes Kopffeld" im Codierstrom gleich "1" ist und das "erste Wiederholungsfeld" gleich "1" ist, zeigt dies eine Bildstruktur, wo das Wiederholungsfeld vom Kopffeld im Zeitpunkt der Decodierung erzeugt wird. Wenn das Flag "erstes Kopffeld" im Codierstrom "0" ist und das "erste Widerholungsfeld" gleich "1" ist, zeigt dies die Bildstruktur, wo das Wiederholungsfeld vom Bodenfeld im Zeitpunkt der Decodierung erzeugt wird.

[0188] Die oben beschriebene Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld werden aus dem Codierstrom für jedes Bild extrahiert und zur Spleiß-Steuerung **500** geliefert. Der Elementarstrom ST_{OLD} und der Elementarstrom ST_{NEW}, welche zur Stromanalyseschaltung **502** geliefert werden, werden unmittelbar zum Stromprozessor **503** als Elementarstrom ST_{OLD} und als Elementarstrom ST_{NEW} geliefert.

[0189] Außerdem hat die Stromanalyseschaltung **502** einen Zähler, um die Anzahl von Bits des gelieferten Stroms ST_{OLD} und des Stroms ST_{NEW} zu zählen. Auf Basis des Zählwerts und der Menge erzeugter Bits in jedem Bild simuliert die Stromanalyseschaltung **502** die verbleibende Datenmenge im VBV-Puffer in Bezug auf jedes Bild. Die verbleibende Datenmenge im VBV-Puffer in Bezug auf jedes Bild, welche durch die Stromanalyseschaltung **502** wird, wird ebenfalls zur Spleiß-Steuerung **500** geliefert.

[0190] Der Stromprozessor **503** ist eine Schaltung, die Stromstruktur, die Datenelemente und die Flags des Stroms ST_{OLD} und des Stroms ST_{NEW} zu ändern, so dass ein Spleiß-Strom ST_{SPL}, der durch Spleißen des Stroms ST_{OLD} und des Stroms ST_{NEW} erzeugt wird, zu einem saumlosen Strom wird. Eine spezifische Verarbeitung, welche durch den Stromprozessor **503** ausgeführt wird, wird mm mit Hilfe von Fig. 21A bis Fig. 21B beschrieben.

[0191] Fig. 21A zeigt den ursprünglichen Strom ST_{OLD}, welcher von der Schlüsselstation **30** geliefert wird, und die Ortskurve der Datenbelegungsmenge des Stroms ST_{OLD} im VBV-Puffer. Fig. 21B zeigt den Ersatzstrom ST_{NEW}, der im Stromserver **49** gespeichert ist, und die Ortskurve der Datenbelegungsmenge des Stroms ST_{NEW} im VBV-Puffer. Fig. 21C zeigt den Spleiß-Strom ST_{SPL}, der durch den Strom ST_{OLD} und den Strom ST_{NEW} erlangt wird, bei Spleißpunkten SP1 und SP2, und die Ortskurve der Datenbelegungsmenge des Spleiß-Stroms ST_{SPL} im VBV-Puffer. In Fig. 21A ist der DTS (Decodierzeitstempel) gezeigt. SP1vbv zeigt den ersten Spleißpunkt auf der Ortkurve der Datenbelegungsmenge im VBV-Puffer. SP2vbv zeigt den zweiten Spleißpunkt auf der Ortskurve der Datenbelegungsmenge im VBV-Puffer. VO(I6) zeigt die Datenmenge eines Bilds I6, welche im VBV-Puffer gepuffert ist, wenn ein Bild B5 vom VBV-Puffer herausgeführt wird. GP(I6) zeigt die Menge erzeugter Bits des Bilds I6. VD(I6) zeigt den Wert der VBV-Verzögerung des Bilds I6. VO(B7) zeigt die Datenmenge eines Bilds B7, welches auf dem VBV-Puffer gepuffert ist, wenn das Bild I6 aus dem VBV-Puffer herausgeführt ist. GP(B11) zeigt die Menge erzeugter Bits eines Bilds B11. VD(I12) zeigt den Wert der VBV-Verzögerung eines Bilds I12. VO(I12) zeigt die Datenmenge des Bilds I12, welches im VBV-Puffer gepuffert ist, wenn das Bild B11 aus dem VBV-Puffer herausgeführt wird. In Fig. 21B zeigt GB(I6') die Menge erzeugter Bits eines Bilds I6'. VD(I6') zeigt den Wert der VBV-Verzögerung des Bilds I6'. VO(I6') zeigt die Datenmenge des Bilds I6', welches im VBV-Puffer gepuffert ist, am ersten Spleißpunkt SP1vbv im VBV-Puffer. GB(B11') zeigt die Menge erzeugter Bits eines Bilds B11'. VO(I12') zeigt die Datenmenge eines Bilds B12', welches im VBV-Puffer gepuffert ist, am zweiten Spleißpunkt ST2vbv im VBV-Puffer. In Fig. 21C zeigt GB(I6'') die Menge erzeugter Bits eines Bilds I6'', welches verarbeitet wurde, so dass der Spleiß-Strom ST_{SPL} zu einem saumlosen Strom wird. VD(I6'') zeigt den Wert der VBV-Verzögerung des Bilds I6''. GP(B11'') zeigt die Menge erzeugter Bits eines Bilds B11'', welches so verarbeitet ist, dass der Spleiß-Strom ST_{SPL} zu einem saumlosen Strom wird.

[0192] Der ursprüngliche Strom ST_{OLD} ist ein Strom, der in der Schlüsselstation **30** codiert ist, und der Ersatzstrom ST_{NEW} ist ein Strom, welcher in der örtlichen Station **40** codiert ist. Daher sind der Strom ST_{OLD} und der Strom ST_{NEW} Ströme, welche durch ihre entsprechenden Videocodierer separat codiert sind. Das heißt, der Wert VD(I6) der VBV-Verzögerung des ersten Bilds I6 im Strom ST_{OLD} und der Wert VD(I6') der VBV-Verzögerung des ersten Bilds **16'** im Strom ST_{NEW} haben nicht den gleichen Wert. In einem solchen Fall ist die Datenbelegungsmenge VO(I6) des ursprünglichen Stroms ST_{OLD} im VBV-Puffer und die Datenbelegungsmenge VO(I6') des Ersatzstroms ST_{NEW} im VBV-Puffer bei dem Zeitpunkt des Strom-Spleiß-Punkts SP1vbv im VBV-Puffer vonein-

ander verschieden.

[0193] Wie beim Stand der Technik der vorliegenden Erfindung beschrieben, wird, wenn der Strom ST_{OLD} und der Strom ST_{NEW} vereinfachend am Spleißpunkt SP1 gespleißt werden, die Datenbelegungsmenge des vereinfachend gespleißten Stroms im VBV-Puffer unstetig und erzeugt einen Überlauf/Unterlauf.

[0194] Somit führt im Strom-Spleißer **50** der Stromprozessor **503** die Stromverarbeitung in Bezug auf die Stromstruktur des zugeführten Stroms ST_{OLD} und des Stroms ST_{NEW} aus, so dass der Spleiß-Strom ST_{SPL} am Spleißpunkt saumlos wird, auf Basis der Datenelemente, welche vom Strom ST_{OLD} und vom Strom ST_{NEW} durch die Stromanalyseschaltung **502** extrahiert werden. Diese Verarbeitung wird anschließend beschrieben.

[0195] Die Spleiß-Steuerung **500** empfangt von der Stromanalyseschaltung **502** Information, beispielsweise die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld für jedes Bild, die Menge erzeugter Bits in jedem Bild und die Datenbelegungsmenge im VBV-Puffer in Bezug auf jedes Bild als die Datenelemente bezogen auf den Strom ST_{OLD}. In Fig. 21A bis Fig. 21C wird der Wert von VBV-Verzögerung des Bilds I6 ausgedrückt als VD(I6), und die Menge erzeugter Bits des Bilds I6 wird ausgedrückt als GB(I6). Die Datenbelegungsmenge des Bilds I6 im VBV-Puffer wird ausgedrückt als VO(I6).

[0196] Anschließend wird die Verarbeitung, welche durch die Spleiß-Steuerung **500** und den Stromprozessor **503** in Bezug auf den VBV-Puffer am Spleißpunkt SP1 ausgeführt wird, beschrieben.

[0197] Zunächst instruiert, wenn die Spleiß-Steuerung **500** bestimmt hat, dass der Wert VD(I6) der VBV-Verzögerung des Bilds I6 des ursprünglichen Stroms ST_{OLD} und der Wert VD(I6') der VBV-Verzögerung des Bilds I6' des Ersatzstroms ST_{NEW} am Spleißpunkt SP1 voneinander verschieden sind, die Spleiß-Steuerung **50** den Stromprozessor **503**, den Wert der VBV-Verzögerung des Bilds I6', der im Ersatzstrom ST_{NEW} beschrieben wurde, von VD(I6') auf VD(I6) umzuschreiben.

[0198] Gemäß der Instruktion von der Spleiß-Steuerung **500** schreibt der Stromprozessor **503** den Wert der VBV-Verzögerung **16** Bits, der im Bildkopf des Ersatzstroms ST_{NEW} beschrieben ist, von VD(I6') auf VD(I6) um.

[0199] Wenn in diesem Fall der Wert der VBV-Verzögerung im Ersatzstrom ST_{NEW} einfach von VD(I6') auf VD(I6) umgeschrieben wird, um somit einen Bitstrom vom VBV-Puffer gemäß dem Umschreiben der VBV-Verzögerung auszugeben, ist die Menge erzeugter Bits des Bilds I6' unzureichend und bewirkt somit einen Unterlauf des VBV-Puffers. Somit führt die Spleiß-Steuerung **500** die Verarbeitung durch, um Stopfbytes in das Bild I6' des Ersatzstroms ST_{NEW} einzufügen, so dass die Menge erzeugter Bits GP(I6') des Bilds I6' des Ersatzstroms ST_{NEW} zur Menge erzeugter Bits GB(I6") des Bilds I6" des saumlos-gespleißten Stroms ST_{SPL} wird. Die Stopfbytes sind Daten, welche aus Dummy-Bits "0" bestehen.

[0200] Um die Verarbeitung zum Einfügen von Stopfbytes auszuführen, berechnet die Spleiß-Steuerung **500** die Datenmenge der Stopfbytes, welche eingefügt werden sollte, indem die Menge erzeugter Bits GB(I6) und die Datenbelegungsmenge VO(I6) im VBV-Puffer, welche als Information empfangen wird, bezogen auf das Bild I6 und das Bild B7 im Strom ST_{OLD}, und die Menge erzeugter Bits GB(I6') und die Datenbelegungsmenge VO(I6') im VBV-Puffer verwendet wird, welche als Information empfangen wird, welche auf das Bild I6' im Strom ST_{NEW} bezogen ist. Insbesondere werden die Stopfbytes SP1 [Byte] auf Basis der folgenden Gleichung (2) berechnet:

$$SB1\ [Byte] = \{GB(I6") - GB(I6')\}/8 = \{GB(I6) - GB(I6') + VO(B7) - VO(B7')\}/8 \quad (2)$$

[0201] Die Spleiß-Steuerung **500** steuert den Stromprozessor **503**, um die Stopfbytes SP1, welche gemäß der Gleichung (2) berechnet wurden, in den Strom ST_{NEW} einzufügen.

[0202] Gemäß dem Befehl von der Spleiß-Steuerung **500** beschreibt der Stromprozessor **503** die Stopfbytes SP1 im Strom ST_{NEW}. Als Position, wo die Stopfbytes in den Strom geschrieben werden sollten, wird eine Position vor dem Startcode des Bildkopfs des Bilds I6 im Codierstrom ST_{NEW} am meisten gewünscht, wobei jedoch eine Position vor einem anderen Startcode ebenfalls verwendet werden kann.

[0203] Die oben beschriebene Verarbeitung ist die Steuerung, welche durch die Spleiß-Steuerung **500** und den Stromprozessor **503** in Bezug auf den VBV-Puffer am Spleißpunkt ST1 ausgeführt wird.

[0204] Anschließend wird die Steuerung, welche durch die Spleiß-Steuerung **500** und den Stromprozessor **503** in Bezug auf den VBV-Puffer am Spleißpunkt SP2 ausgeführt wird, beschrieben.

[0205] Wenn der Strom ST_{NEW} und der Strom ST_{OLD} vereinfachend am Spleißpunkt ST2 gespleißt werden, ist die Menge erzeugter Bits GB(B11') des letzten Bilds I11' des Stroms ST_{NEW} unzureichend und daher nicht mit der Ortskurve der Datenbelegungsmenge im VBV-Puffer in Bezug auf das erste Bild I12 des Stroms ST_{NEW} fortlaufend. Als Ergebnis wird ein Unterlauf oder Überlauf des VBV-Puffers erzeugt.

[0206] Damit führt die Spleiß-Steuerung **500** die Verarbeitung durch, um Stopfbytes in den Strom ST_{NEW} einzufügen, so dass die Menge erzeugter Bits GB(I11') des letzten Bilds I11' des Stroms ST_{NEW} zur Menge erzeugter Bits GB(I11") des Bilds I11" von Fig. 21C wird, um somit eine stetige Ortskurve des VBV-Puffers am Spleißpunkt SP2vbv im VBV-Puffer zu realisieren.

[0207] Um die Verarbeitung zum Einfügen der Stopfbytes auszuführen, berechnet die Spleiß-Steuerung **500** die Datenmenge der Stopfbytes, welche eingefügt werden sollte, wobei VO(I12), welche als Information bezogen auf das Bild I12 im Strom ST_{OLD} empfangen wird, die Menge erzeugter Bits GB(B11') des letzten Bilds B11' des Stroms ST_{NEW}, und die Datenbelegungsmenge VO(I12') im VBV-Puffer in Bezug auf das Bild I12' des Stroms ST_{NEW} verwendet wird. Insbesondere werden die Stopfbytes SB2 [Byte] auf Basis der folgenden Gleichung (3) berechnet:

$$SB2\ [Byte] = \{GB(B11") - GB(B11')\}/8 = \{VO(I12) - VO(I12)\}/8 \quad (3)$$

[0208] Anders ausgedrückt ist die Datenbelegungsmenge VO(I12') die Datenbelegungsmenge im VBV-Puffer in Bezug auf den Strom ST_{NEW}, nachdem das letzte Bild B11' aus dem VBV-Puffer herausgeführt ist. Diese Datenbelegungsmenge VO(I12') kann durch die Stromanalyseschaltung **502** leicht ermittelt werden, welche die Ortskurve des VBV erfasst hat, indem die Anzahl von Bits des Stroms STNEW gezählt wird.

[0209] Die Spleiß-Steuerung **500** steuert den Stromprozessor **503**, um die Stopfbytes SB2, welche gemäß der Gleichung (3) berechnet wurden, in den Strom ST_{NEW} einzufügen.

[0210] Gemäß dem Befehl von der Spleiß-Steuerung **500** beschreibt der Stromprozessor **503** die Stopfbytes SB2 als Information bezogen auf das Bild B11' des Stroms ST_{NEW}. Als Position, wo die Stopfbytes im Strom beschrieben werden sollten, ist eine Position vor dem Startcode des Bildkopfs des Bilds B11' im Codierstrom ST_{NEW} am meisten erwünscht.

[0211] Die oben beschriebene Verarbeitung ist die Steuerung, welche durch die Spleiß-Steuerung **500** und den Stromsprozessor **503** in Bezug auf den VBV-Puffer am Spleißpunkt SP2 ausgeführt wird.

[0212] Unter Bezug auf Fig. 22A bis Fig. 22C wird nun ein erstes Beispiel einer Verarbeitung in Bezug auf die Flags, beispielsweise das erste Kopffeld und das erste Wiederholungsfeld durch die Spleiß-Steuerung **500** und den Stromprozessor **503** am Spleißpunkt SP1 beschrieben.

[0213] Fig. 22A zeigt die Rahmenstruktur eines Fernsehprogramms PG_{OLD}, welches aus einem Programm 1, einer Werbung CM1 und einem Programm 2, welche in der Schlüsselstation **30** erzeugt werden, besteht, und einen Codierstrom ST_{OLD}, der durch Codieren des Fernsehprogramms PG_{OLD} erlangt wird. **Fig.** 22 zeigt die Rahmenstruktur einer Ersatzwerbung CM1', welche in der örtlichen Station **40** erzeugt wird, und eines Codierstroms ST_{NEW}, welcher durch Codieren der Ersatzwerbung CM1' erlangt wird. Fig. 22C zeigt einen Spleiß-Strom ST_{SPL}, der durch Spleißen des ursprünglichen Stroms ST_{OLD} und des Ersatzstroms ST_{NEW} erzeugt wird, und die Rahmenstruktur, welche durch Decodieren des Spleiß-Stroms ST_{SPL} erlangt wird.

[0214] Die Spleiß-Steuerung **500** vergleicht das erste Kopffeld jedes Bilds der Werbung CM1 im Strom ST_{OLD}, welches von der Stromanalyseschaltung **502** geliefert wird, mit dem ersten Kopffeld der Werbung CM1' im Ersatzstrom ST_{NEW}. Wenn das erste Kopffeld des Stroms ST_{OLD} und das erste Kopffeld des Ersatzstroms STNEW miteinander übereinstimmen, wird die gleiche Feldstruktur verwendet, und daher ist die Verarbeitung in Bezug auf die Flags, beispielsweise das erste Kopffeld und das erste Wiederholungsfeld nicht notwendig. Wenn jedoch das erste Kopffeld der ursprünglichen Werbung CM1 gleich "0" ist und das erste Kopffeld der Ersatzwerbung CM1' gleich "1" ist, wie in Fig. 22A bis Fig. 22C gezeigt ist, werden Probleme einer Diskontinuität und einer Vervielfältigung von Feldern, wie in **Fig.** 6 beschrieben, erzeugt.

[0215] Somit schreibt der Strom-Spleißer **50** nach der vorliegenden Erfindung das erste Kopffeld und das erste Wiederholungsfeld der Bilder in der Nähe des Spleißpunktes um, um somit nicht einen Strom zu erzeugen, der für den MPEG-Standard ungünstig ist, bei dem Feldausfall und Überlappung durch die Spleiß-Verarbeitung erzeugt werden.

[0216] In dem in Fig. 22A bis Fig. 22C gezeigten Beispiel steuert die Spleiß-Steuerung **500** den Stromprozessor **503**, um das erste Wiederholungsfeld eines Bilds P3, welches einen Rahmen hat, der durch ein Kopffeld T4 und ein Bodenfeld B4 gebildet wird, von 0 auf 1 umzuschreiben. Zusätzlich steuert die Spleiß-Steuerung **500** den Stromprozessor **503**, um das erste Wiederholungsfeld eines Bilds P9', welches einen Rahmen hat, der durch ein Kopffeld t10 und ein Bodenfeld B11 gebildet ist, von 0 auf 1 umzuschreiben, um somit einen Strom zu erzeugen, der am Spleißpunkt SP2 saumlos ist. Da außerdem die Werbung CM1' um die Zeit eines Rahmens von der ursprünglichen Werbung CM1 durch Umschreiben des ersten Kopffelds des Bilds P9' verschoben wird, steuert die Spleiß-Steuerung **500** den Stromprozessor **503**, um vom Strom ST_{OLD} ein Bild B13 zu löschen, welches zu allererst Bilder des Programms 2 auf der Anzeige angezeigt wird.

[0217] Gemäß der Instruktion von der Spleiß-Steuerung **500** sucht der Stromprozessor **503** den ursprünglichen Strom ST_{OLD} für den Startcode der Bildcodiererweiterung bezogen auf das Bild P3 und schreibt den Wert des ersten Wiederholungsfelds darin von 0 auf 1 um. Das somit ein Wiederholungsfeld B4' durch Decodieren des Bilds P3 erzeugt wird, bei dem der Wert des ersten Wiederholungsfelds umgeschrieben ist, werden die Felder am Spleißpunkt SP1 stetig. Ähnlich sucht der Stromprozessor **503** den Ersatzstrom ST_{NEW} für den Startcode der Bildcodiererweiterung bezogen auf das Bild P9' und schreibt den Wert des ersten Wiederholungsfelds darin von 0 auf 1 um. Das somit ein Wiederholungsfeld t10' durch Decodieren des Bilds P9' erzeugt wird, bei dem der Wert des ersten Wiederholungsfelds umgeschrieben ist, werden die Felder am Spleißpunkt SP2 stetig. Der Stromprozessor **503** löscht außerdem einen Bereich, wo das Datenelement bezogen auf das Bild B13 im Ursprungsstrom ST_{OLD} beschrieben ist oder ersetzt den Bereich durch Nulldaten.

[0218] Fig. 23A bis 23C zeigen ein weiteres Beispiel der Verarbeitung in Bezug auf die Flags, beispielsweise das erste Kopffeld und das erste Wiederholungsfeld, welche in Fig. 22A bis Fig. 22C beschrieben sind. In Bezug auf Fig. 23A bis 23C wird nun ein zweites Beispiel einer Verarbeitung in Bezug auf die Flags, beispielsweise das erste Kopffeld und das erste Wiederholungsfeld durch die Spleiß-Steuerung **500** und den Stromprozessor **503** an den Spleißpunkten SP1 und SP2 beschrieben.

[0219] In dem Beispiel, welches in Fig. 23A bis 23C gezeigt ist, steuert die Spleiß-Steuerung **500** den Stromprozessor **503**, um das erste Kopffeld eines Bilds B7', welches durch ein Kopffeld t5 und ein Bodenfeld B6 gebildet ist, von 1 auf 0 umzuschreiben, und um das erste Wiederholungsfeld des Bilds B7' von 0 auf 1 umzuschreiben, so dass die Felder am Saum zwischen dem Programm 1 und der Werbung CM1' am Spleißpunkt SP1 stetig werden. Zusätzlich steuert die Spleiß-Steuerung **500** den Stromprozessor **503**, um das erste Kopffeld eines Bilds B13, welche durch ein Kopffeld T11 und ein Bodenfeld B11 gebildet wird, von 1 auf 0 umzuschreiben, so dass die Felder am Saum zwischen der Werbung CM1' und dem Programm 2 am Spleißpunkt SP2 stetig werden. Außerdem steuert die Spleiß-Steuerung **500** den Stromprozessor **503**, um das erste Kopffeld eines Bilds B14, welches durch ein Kopffeld T12 und ein Bodenfeld B12 gebildet wird, von 1 auf 0 umzuschreiben, und um das erste Wiederholungsfeld von 1 auf 0 umzuschreiben.

[0220] Unter der Steuerung der Spleiß-Steuerung **500** sucht der Stromprozessor **503** den Ersatzstrom ST_{NEW} für den Startcode der Bildcodiererweiterung bezogen auf das Bild B7'. Der Stromprozessor **503** schreibt dann das erste Kopffeld im Strom von 1 auf 0 um und schreibt das erste Wiederholungsfeld von 0 auf 1 um. Somit wird durch Decodieren des Bilds B7', bei dem die Werte des ersten Kopffelds und des ersten Wiederholungsfelds umgeschrieben sind, die Anzeigezeit des Bodenfelds b6 um einen Rahmen verschoben, und ein Wiederholungsfeld b6' wird erzeugt. Daher werden die Felder am Spleißpunkt SP1 stetig. Ähnlich sucht der Stromprozessor **503** den ursprünglichen Strom ST_{OLD} für den Startcode der Bildcodiererweiterung bezogen auf das Bild B13 und schreibt das erste Kopffeld darin von 1 auf 0 um. Zusätzlich schreibt der Stromprozessor **503** das erste Kopffeld bezogen auf das Bild B14 im ursprünglichen Strom ST_{OLD} von 1 auf 0 um, und schreibt das erste Wiederholungsfeld von 1 auf 0 um. Da somit die Anzeigezeit der Bodenfelder B11 und B12 um einen Rahmen verschoben ist, werden, indem die Bilder B13 und B14 decodiert werden, bei denen die Werte des ersten Kopffelds und des ersten Wiederholungsfelds umgeschrieben sind, die Felder am Spleißpunkt SP2 stetig.

[0221] Das erste Verarbeitungsbeispiel, welches in Fig. 22A bis Fig. 22C gezeigt ist, und das zweite Verarbeitungsbeispiel, welches in Fig. 23A bis 23C gezeigt ist, werden verglichen. Wie man aus Fig. 22C ersehen kann, wird, da das Bild B7, welches am Anfang der Ersatzwerbung CM1' angezeigt wird, um ein Feld vom Bild B7 verschoben ist, welches am Anfang der ursprünglichen Werbung CM1 angezeigt wird, die Verzögerungszeit für die Ersatzwerbung CM1' um ein Feld verzögert. Die Verzögerung der Anzeige um ein Feld kann kaum durch die menschlichen Augen erkannt werden. Da jedoch die Sendestation von Sendewerbungen von Kundengesellschaften profitiert, könnte es für die Sendestation erforderlich sein, Werbungen genau zu senden, ohne irgendeine Verzögerung zu haben, statt das Hauptprogramm, beispielsweise das Programm 1 zu senden. In dem Fall, wo eine solche genaue Anzeigezeit erforderlich ist, ist das zweite Verarbeitungsbeispiel, welches in Fig. 23A bis 23C gezeigt ist, wirksam. Durch Umschreiben der Werte des ersten Kopffelds und des ersten Wiederholungsfelds des Bilds B7' wie beim zweiten Verarbeitungsbeispiel, welches in Fig. 23A bis 23C gezeigt ist, kann das erste Bild B7' der Ersatzwerbung CM1' genau angezeigt werden, ohne irgendeine Verzögerung gegenüber dem ersten angezeigten Bild der ursprünglichen Werbung CM1 zu haben.

[0222] Kurz ausgedrückt steht die Ortskurve der Datenbelegungsmenge im VBV-Puffer des Stroms ST_{NEW}, der vom Stromprozessor **503** ausgegeben wird, in Einklang mit der Ortskurve der Datenbelegungsmenge des Stroms ST_{OLD} im VBV-Puffer, und sie steht im Einklang in Bezug auf das Feldmuster/Rahmenmuster. Daher wird durch Steuern des Schaltbetriebs der Spleiß-Schaltung **504** auf Basis eines Steuersignals und der Spleiß-Steuerung **500**

der Spleiß-Strom ST_{SPL} erzeugt, indem der Strom STNEW nach dem Strom ST_{OLD} am Spleißpunkt verbunden wird, und indem der Strom ST_{OLD} nach dem Strom ST_{NEW} am Spleißpunkt SP2 verbunden wird. Bei den Spleißpunkten SP1 und SP2 ist in diesem Spleiß-Strom ST_{SPL} die Ortskurve der Datenbelegungsmenge im VBV-Puffer stetig, und das Feldmuster/Rahmenmuster ist ebenfalls stetig.

[0223] Es wird nun die zweite Ausführungsform der Spleiß-Verarbeitung beschrieben. Die zweite Ausführungsform der Spleiß-Verarbeitung bezieht sich auf die Spleiß-Verarbeitung, welche in dem Fall ausgeführt wird, wo eine neu einzufügende Werbung CM1' codiert wird, um einen Codierstrom ST_{NEW} zu erzeugen, wenn ein Codierstrom ST_{OLD} eines Übertragungsprogramms von der Schlüsselstation **30** übertragen wird. Das heißt, der Codierstrom ST_{OLD} des Übertragungsprogramms, welches von der Schlüsselstation **30** übertragen wird, wird analysiert, und die neu einzufügende Werbung CM1' wird auf Basis des Analyseergebnisses codiert.

[0224] Zunächst wird der Codierstrom ST_{OLD}, welcher von der Schlüsselstation **30** übertragen wird, von der Form des Transportstroms in die Form des Elementarstroms durch die Stromsetzungsschaltung **44** umgesetzt. Der Codierstrom ST_{OLD}, der in die Form des Elementarstroms umgesetzt ist, wird zur Stromanalyseschaltung **502** des Strom-Spleißers **50** geliefert.

[0225] Die Stromanalyseschaltung **502** des Strom-Spleißers **50** ist eine Schaltung, um die Stromsyntax des Codierstroms ST_{OLD} zu analysieren. Bei der zweiten Spleiß-Verarbeitungs-Ausführungsform analysiert die Stromanalyseschaltung **502** lediglich die Syntax des Codierstroms ST_{OLD}, und sie analysiert nicht die Syntax des Ersatzstroms ST_{NEW}.

[0226] Insbesondere sucht die Stromanalyseschaltung **502** nach dem 32-Bit-Bildstartcode, der im ursprünglichen Strom ST_{OLD} beschrieben ist, um dadurch den Ort zu erlernen, wo Information bezogen auf den Bildkopf im Strom beschrieben ist, wie man aus der Syntax des Codierstroms, der in Fig. 10 und Fig. 15 gezeigt ist, versteht. Danach findet die Stromanalyseschaltung **502** den 3-Bit-Bildcodiertypus, der 11 Bits nach dem Bildstartcode beginnt, um dadurch die Bildart zu erlernen. Außerdem erlernt die Stromanalyseschaltung **502** die VBV-Verzögerung des codierten Bilds von der 16-Bit-VBV-Verzögerung, welche anschließend beschrieben wird, welche im Anschluss an diese 3-Bit-Bildcodierart beschrieben wird.

[0227] Zusätzlich sucht die Stromanalyseschaltung **502** nach dem 32-.Bit-Erweiterungsstartcode, der im Codierstrom ST_{OLD} beschrieben ist, um dadurch die Stelle im Strom zu erlernen, wo Information bezogen auf die Bildcodiererweiterung beschrieben ist, wie man aus der Syntax des Codierstroms, die in Fig. 10 und Fig. 15 gezeigt ist, versteht. Danach sucht die Stromanalyseschaltung **502** nach dem 1-Bit-Kopffeld, welches 25 Bits nach dem Bildstartcode beschrieben ist, und nach dem ersten Wiederholungsfeld, welches 6 Bits nach diesem ersten Kopffeld beschrieben ist, um dadurch die Rahmenstruktur des codierten Bilds zu erlernen.

[0228] Die Stromanalyseschaltung **502** liefert die Datenelemente, beispielsweise die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld, welche für jedes Bild aus dem ursprünglichen Strom ST_{OLD} extrahiert werden, zur Sendesystemsteuerung **41** über die Spleiß-Steuerung **500**. Es ist nicht notwendig, die Datenelemente aller Bilder des ursprünglichen Stroms ST_{OLD} zu übertragen. Es mag genügen, lediglich die Datenelemente zu senden, beispielsweise die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld der Bilder entsprechend der Werbung CM1 des Übertragungsprogramms.

[0229] Die Sendesystemsteuerung **41** steuert den CM-Server **47**, um Videodaten der Werbung CM1' wiederzugeben, um die Werbung CM des Übertragungsprogramms zu ersetzen. Außerdem liefert die Sendesystemsteuerung **41** die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld, welche vom ursprünglichen Strom ST_{OLD} extrahiert werden, zur Codierersteuerung **480** des Codierblocks **48**.

[0230] Die Codiersteuerung **480** steuert den Codierer **481**, um die Basisband-Videodaten der Ersatzwerbung CM1' unter Verwendung der Bildcodierart, der VBV-Verzögerung, des ersten Kopffelds und des ersten Wiederholungsfelds, welche von der Sendesystemsteuerung **41** geliefert werden, zu codieren. Das heißt, die Ersatzwerbung CM1' wird so codiert, dass die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld des Codierstroms ST_{OLD} der ursprünglichen Werbung CM1 und die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld des Stroms $_{STNEW,}$ welche durch Codieren der Ersatzwerbung CM1' erlangt werden, zueinander perfekt identisch werden. Als Ergebnis wird der Codierstrom ST_{NEW}, der die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld hat, welche perfekt identisch mit der Bildcodierart, der VBV-Verzögerung, dem ersten Kopffeld und dem ersten Wiederholungsfeld des Codierstroms ST_{OLD} der ursprünglichen Werbung CM1 sind, erzeugt.

[0231] Die Spleiß-Steuerung **500** schaltet den Eingangsanschluss der Umschaltschaltung **501** auf "b", um den Elementarstrom ST_{NEW}, der vom Codierer **481** ausgegeben wird, zur Stromanalyseschaltung

502 zu liefern. Bei dieser zweiten Ausführungsform einer Spleiß-Verarbeitung wird, da die Stromanalyseschaltung **502** lediglich die Syntax des Codierstroms ST_{OLD} analysiert und nicht die Syntax des Ersatzstroms ST_{NEW} analysiert, der Strom ST_{NEW} unverändert ausgegeben, ohne durch die Stromanalyseschaltung **502** analysiert zu werden.

[0232] Bei der zweiten Ausführungsform einer Spleiß-Verarbeitung führt, da die Stromverarbeitung zum Ändern der Datenelemente im Stroms ST_{OLD} und im Strom ST_{NEW}, welche von der Stromanalyseschaltung **502** ausgegeben werden, nicht notwendig ist, der Stromprozessor **503** lediglich eine Synchronisationsverarbeitung (oder Rahmensynchronisation) durch, um eine Rahmensynchronisation zwischen dem Strom ST_{OLD} und dem Strom ST_{NEW} zu realisieren. Insbesondere hat der Stromprozessor **503** einen FIFO-Puffer für mehrere Rahmen, und er puffert den Strom ST_{OLD} im FIFO-Puffer, bis der Ersatzstrom ST_{NEW} vom Codierer **481** ausgegeben wird. Damit kann die Rahmensynchronisation zwischen dem Strom ST_{OLD} und dem Strom ST_{NEW} realisiert werden. Der Strom ST_{OLD} und der Strom ST_{NEW}, welche zur Rahmensynchronisation verarbeitet werden, werden zur Spleiß-Schaltung **504** geliefert.

[0233] Die Spleiß-Steuerung **500** steuert das Umschalten der Spleiß-Schaltung **504**, so dass der Strom ST_{NEW} nach dem Strom ST_{OLD} am Spleißpunkt SP1 verbunden wird, und so dass der Strom ST_{OLD} nach dem Strom ST_{NEW} am Spleißpunkt ST2 verbunden wird. Als Ergebnis wird ein Spleiß-Strom ST_{SPL} von der Spleiß-Schaltung **504** ausgegeben.

[0234] Obwohl der Strom ST_{OLD} und der Strom ST_{NEW} einfach durch die Spleiß-Schaltung **504** umgeschaltet werden, ist die Ortskurve der Datenbelegungsmenge des Spleiß-Stroms ST_{SPL} im VBV-Puffer stetig, und das Rahmenmuster an den Spleißpunkten ist ebenfalls stetig. Der Grund dafür ist folgender. Das heißt, da der Strom ST_{NEW} auf Basis des Stromanalyseergebnisses der Syntax des ursprünglichen Stroms ST_{OLD} codiert wird, wird der Strom ST_{NEW}, der mit dem ursprünglichen Strom ST_{OLD} im Eingang steht, erzeugt. Daher ist die Ortskurve des Spleiß-Stroms ST_{SPL} im VBV-Puffer perfekt die gleiche wie die Ortskurve der Datenbelegungsmenge des ursprünglichen Stroms ST_{OLD} im VBV-Puffer, und die Rahmenstruktur des erzeugten Spleiß-Stroms ST_{SPL} ist perfekt identisch mit der Rahmenstruktur des ursprünglichen Stroms ST_{OLD}.

[0235] Damit wird bei der zweiten Ausführungsform die Syntax des ursprünglichen Codierstroms ST_{OLD}, welche von der Schlüsselstation übertragen wird, analysiert, und die Ersatzwerbung SM1' wird gemäß dem Analyseergebnis so codiert, dass sie die gleiche Stromstruktur und den Codierparameter wie des Codierstroms ST_{OLD} haben. Daher kann beim Spleißen des Codierstroms ST_{OLD} und des Codierstroms ST_{NEW}, welcher separat erzeugt werden, die Übereinstimmung zwischen dem Codierstrom ST_{OLD} und dem Codierstrom ST_{NEW} leicht realisiert werden. Als Ergebnis kann der saumlos-gespleißte Strom ST_{SPL} in Einklang stehend mit dem MPEG-Standard erzeugt werden.

[0236] Die dritte Ausführungsform der Spleiß-Verarbeitung wird nun beschrieben. Bei der dritten Ausführungsform der Spleiß-Verarbeitung werden Codierparameter zum Codieren eines Codierstroms ST_{OLD} einer ursprünglichen Werbung CM1 und eines Codierstroms ST_{NEW} einer Ersatzwerbung CM1' vorher bestimmt, bevor der Codierstrom ST_{OLD} der ursprünglichen Werbung CM1 und der Codierstrom ST_{NEW} der Ersatzwerbung CM1' erzeugt werden, und die ursprüngliche Werbung CM1 und die Ersatzwerbung CM1' werden auf Basis der vorher festgelegten Codierparameter codiert. Beispielsweise sind die Codierparameter Information, welche durch Bildcodierart, VBV-Verzögerung, erste Kopffeld, erstes Wiederholungsfeld und die Menge erzeugter Bits, wie schon beschrieben, gezeigt wird.

[0237] Zunächst werden als Codierparameter zum Codieren der ursprünglichen Werbung CM1 die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld in der Schlüsselstation **30** bestimmt. Die Sendesystemsteuerung **31** der Schlüsselstation **30** liefert die Codierparameter zur Codiersteuerung **350** des MPEG-Codierblocks **35**, und sie liefert außerdem die Codierparameter zur Sendesystemsteuerung **41** jeder örtlichen Station **40** über eine Kommunikationsleitung.

[0238] Die Codiersteuerung **350** steuert den Videocodierer **351-1V**, um die Videodaten der ursprünglichen Werbung CM1 zu codieren, indem die Codierparameter verwendet werden, beispielsweise die Codierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld, welche von der Sendesystemsteuerung **31** geliefert werden. Das heißt, der Codierstrom ST_{OLD}, der vom Videocodierer **351-1V** ausgegeben wird, ist ein Strom auf Basis der Codierparameter, beispielsweise Bildcodierart, VBV-Verzögerung, erstes Kopffeld und erstes Wiederholungsfeld.

[0239] Der Codierstrom ST_{OLD}, welcher vom Videocodierer **351-1V** ausgegeben wird, wird über den Multiplexer **36** und die Modulationsschaltung **37** zur örtlichen Station **40** geliefert.

[0240] Dagegen liefert die örtliche Station **40** die Codierparameter, beispielsweise Bildcodierart, VBV-Verzögerung, erstes Kopffeld und erstes Wiederholungsfeld, welche von der Sendesystemsteuerung **31** der Schlüsselstation **30** geliefert werden, zur Codiersteuerung **480** des Codierblocks **48**.

[0241] Die Codiersteuerung **480** steuert den Codierer **481**, um die Basisband-Videodaten der Ersatzwerbung CM1' zu codieren, wobei die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld, welche von der Sendesystemsteuerung **41** geliefert werden, verwendet werden. Das heißt, der Strom STNEW, der die Bildcodierart, die VBV-Verzögerung, das erste Kopffeld und das erste Wiederholungsfeld hat, welche perfekt identisch mit der Bildcodierart, der VBV-Verzögerung, dem ersten Kopffeld und dem ersten Wiederholungsfeld des Codierstroms ST_{OLD} der ursprünglichen Werbung CM1 sind, wird erzeugt.

[0242] Die Spleiß-Steuerung **500** schaltet den Eingangsanschluss der Umschaltschaltung **501** auf "b", um somit den Elementarstrom ST_{NEW}, der vom Codierer **481** ausgegeben wird, zur Stromanalyseschaltung **502** zu liefern. Bei dieser dritten Ausführungsform einer Spleiß-Verarbeitung führt die Stromanalyseschaltung **502** nicht die Analyse der Syntax des Codierstroms ST_{OLD} und des Codierstroms ST_{NEW} aus.

[0243] Bei der dritten Ausführungsform der Spleiß-Verarbeitung führt, da die Stromverarbeitung zum Ändern der Datenelemente im Strom ST_{OLD} und im Strom ST_{NEW}, welche von der Stromanalyseschaltung **502** ausgegeben werden, nicht notwendig ist, der Stromprozessor **503** lediglich die Synchronisationsverarbeitung (oder Rahmensynchronisation) durch, um Rahmensynchronisation zwischen dem Strom ST_{OLD} und dem Strom ST_{NEW} zu realisieren. Insbesondere hat der Stromprozessor **503** einen FIFO-Puffer für mehrere Rahmen und puffert den Strom ST_{OLD} im FIFO-Puffer, bis der Ersatzstrom ST_{NEW} vom Codierer **481** ausgegeben wird. Somit kann Rahmensynchronisation zwischen dem Strom ST_{OLD} und dem Strom ST_{NEW} realisiert werden. Der Strom ST_{OLD} und der Strom ST_{NEW}, welche zur Rahmensynchronisation verarbeitet werden, werden zur Spleiß-Schaltung **504** geliefert.

[0244] Die Spleiß-Steuerung **500** steuert das Umschalten der Spleiß-Schaltung **504** durch, so dass der Strom ST_{NEW} nach dem Strom ST_{OLD} beim Spleißpunkt SP1 verbunden wird, so dass der Strom ST_{OLD} nach dem Strom ST_{NEW} am Spleißpunkt SP2 verbunden wird. Als Ergebnis wird ein Spleiß-Strom ST_{SPL} von der Spleiß-Schaltung **504** ausgegeben.

[0245] Obwohl der Strom ST_{OLD} und der Strom ST_{NEW} lediglich durch die Spleiß-Schaltung **504** umgeschaltet werden, ist die Ortskurve der Datenbelegungsmenge des Spleiß-Stroms ST_{SPL} im VBV-Puffer stetig, und das Rahmenmuster an den Spleißpunkten ebenfalls stetig. Der Grund dafür ist der, dass die ursprüngliche Werbung CM1 und die Ersatzwerbung CM1' unter Verwendung der Codierparameter codiert sind, beispielsweise der Bildcodierart, der VBV-Verzögerung, des ersten Kopffelds und des ersten Wiederholungsfelds, welche vorher durch die Sendesystemsteuerung **31** der Schlüsselstation **30** bestimmt wird.

[0246] Somit werden bei der dritten Ausführungsform die Codierparameter zwischen der Schlüsselstation und der örtlichen Station bestimmt. Dann wird der Codierstrom ST_{OLD} der ursprünglichen Werbung SM1 in der Schlüsselstation auf Basis der vorher festgelegten Codierparameter erzeugt, und der Codierstrom ST_{NEW} der Ersatzwerbung CM1' wird in der Schlüsselstation auf Basis der vorher festgelegten Codierparameter erzeugt. Daher kann beim Spleißen des Codierstroms ST_{OLD} und des Codierstroms ST_{NEW}, welche separat erzeugt werden, die Übereinstimmung zwischen dem Codierstrom ST_{OLD} und des Codierstroms ST_{NEW} leicht realisiert werden. Als Ergebnis kann der saumlos-gespleißte Strom ST_{SPL} im Einklang stehend mit dem MPEG-Standard erzeugt werden.

Patentansprüche

1. Codierstrom-Spleißeinrichtung zum Spleißen eines ersten Codierstroms und eines zweiten Codierstroms an einem Spleißpunkt, wobei die Vorrichtung aufweist:
eine Änderungseinrichtung (**500**) zum Ändern zumindest eines Codierparameters des zweiten Codierstroms auf Basis eines entsprechenden Codierparameters des ersten Codierstroms, so dass der erste Codierstrom und der zweite Codierstrom saumlos am Spleißpunkt gespleißt werden; und
eine Spleißeinrichtung (**503**) zum Spleißen des ersten Codierstroms und des zweiten Codierstroms, der die geänderten Codierparameter hat;
wobei der Codierparameter, der durch die Änderungseinrichtung geändert wird; zumindest einer von einem ersten Wiederholungsfeldflag ist, welches zeigt, ob oder nicht ein Wiederholungsfeld im Decodierzeitpunkt erzeugt wird, und
ein erstes Kopffeldflag, welches zeigt, ob das erste Feld ein Kopffeld oder ein Bodenfeld ist.

2. Codierstrom-Spleißeinrichtung nach Anspruch 1, welche eine Stromanalyseeinrichtung zum Analysieren der Syntax des ersten Codierstroms aufweist, um den entsprechenden Codierparameter des ersten Codierstroms zu extrahieren, der durch die Änderungseinrichtung verwendet wird.

3. Codierstrom-Spleißeinrichtung nach Anspruch 1, wobei der zumindest eine Codierparameter, der durch die Änderungseinrichtung geändert wird, einen vbv_Verzögerungsparameter aufweist, der einen Anfangszustand eines VBV-Puffers zeigt.

4. Codierstrom-Spleißeinrichtung nach Anspruch 3, wobei die Änderungseinrichtung (**500**) den

vbv_Verzögerungswert des ersten Bilds nach dem Spleißpunkt im zweiten Codierstrom auf den vbv_Verzögerungswert des ersten Bilds nach dem Spleißpunkt im ersten Codierstrom umschreibt.

5. Codierstrom-Spleißeinrichtung nach Anspruch 4, wobei die Änderungseinrichtung (**500**) Stopfbytes in den zweiten Codierstrom einfügt, so dass die Menge erzeugter Bits des ersten Bilds nach dem Spleißpunkt im zweiten Codierstrom zur Bitmenge wird, welche dem umgeschriebenen vbv_Verzögerungswert entspricht.

6. Codierstrom-Spleißeinrichtung nach Anspruch 5, wobei die Änderungseinrichtung (**500**) die Datenmenge der Stopfbytes auf Basis der Menge erzeugter Bits des ersten Bilds nach dem Spleißpunkt im ersten Codierstrom und der Menge erzeugter Bits des ersten Bilds nach dem Spleißpunkt im zweiten Codierstrom berechnet.

7. Codierstrom-Spleißeinrichtung nach Anspruch 6, wobei die Änderungseinrichtung (**500**) die Datenmenge der Stopfbytes auf Basis der Datenbelegungsmenge des VBV-Puffers am Spleißpunkt des ersten Codierstroms und der Datenbelegungsmenge des VBV-Puffers am Spleißpunkt des zweiten Codierstroms berechnet.

8. Codierstrom-Spleißeinrichtung nach Anspruch 1, wobei die Änderungseinrichtung (**500**) den Wert des ersten Wiederholfelds eines Bilds vor dem Spleißpunkt im ersten Codierstrom oder den Wert des ersten Wiederholfelds eines Bilds vor dem Spleißpunkt im zweiten Codierstrom ändert, so dass die Rahmenstruktur des Bilds vor dem Spleißpunkt im ersten Codierstrom und die Rahmenstruktur des Bilds vor dem Spleißpunkt im zweiten Codierstrom miteinander folgerichtig werden.

9. Codierstrom-Spleißeinrichtung nach Anspruch 1, wobei die Änderungseinrichtung (**500**) den Wert des ersten Kopffelds eines Bilds vor dem Spleißpunkt im Codierstrom oder den Wert des ersten Kopffelds eines Bilds vor dem Spleißpunkt im zweiten Codierstrom ändert, so dass die Rahmenstruktur des Bilds vor dem Spleißpunkt im ersten Codierstrom und die Rahmenstruktur des Bilds vor dem Spleißpunkt im zweiten Codierstrom miteinander folgerichtig werden.

10. Codierstrom-Spleißverfahren zum Spleißen eines ersten Codierstroms und eines zweiten Codierstroms an einem Spleißpunkt, wobei das Verfahren aufweist:
einen Änderungsschritt zum Ändern zumindest eines Codierparameters des zweiten Codierstroms auf Basis eines entsprechenden Codierparameters des ersten Codierstroms, so dass der erste Codierstrom und der zweite Codierstrom saumlos am Spleißpunkt gespleißt werden; und
einen Spleißschritt zum Spleißen des ersten Codierstroms und des zweiten Codierstroms, der den geänderten Codierparameter hat;
wobei der Codierparameter, der durch die Änderungseinrichtung geändert wird, zumindest einer von einem ersten Wiederholfeldflag ist, der zeigt, ob ein Wiederholfeld im Decodierzeitpunkt zu erzeugen ist, und
ein erstes Kopffeldflag, welches zeigt, ob das erste Feld ein Kopffeld oder ein Bodenfeld ist.

11. Codierstrom-Spleißverfahren nach Anspruch 10, wobei der zumindest eine Codierparameter, der durch die Änderungseinrichtung geändert wird, einen vbv_Verzögerungsparmeter aufweist, der einen Anfangszustand eines VBV-Puffers zeigt.

12. Codierstrom-Spleißverfahren nach Anspruch 11, wobei im Änderungsschritt der vbv_Verzögerungswert des ersten Bilds nach dem Spleißpunkt im zweiten Codierstrom auf den vbv_Verzögerungswert des ersten Bilds nach dem Spleißpunkt im ersten Codierstrom umgeschrieben wird.

13. Codierstrom-Spleißverfahren nach Anspruch 12, wobei im Änderungsschritt Stopfbytes in den zweiten Codierstrom eingefügt werden, so dass die Menge erzeugter Bits des ersten Bilds nach dem Spleißpunkt im zweiten Codierstrom zur Bitmenge wird, welche dem umgeschriebenen vbv_Verzögerungswert entspricht.

14. Codierstrom-Spleißverfahren nach Anspruch 13, wobei im Änderungsschritt die Datenmenge der Stopfbytes auf Basis der Menge erzeugter Bits des ersten Bilds nach dem Spleißpunkt im ersten Codierstrom und der Menge erzeugter Bits des ersten Bilds nach dem Spleißpunkt im zweiten Codierstrom berechnet wird.

15. Codierstrom-Spleißverfahren nach Anspruch 14, wobei im Änderungsschritt die Datenmenge der Stopfbytes auf Basis der Datenbelegungsmenge des VBV-Puffers im Spleißpunkt des ersten Codierstroms und der Datenbelegungsmenge des VBV-Puffers im Spleißpunkt des zweiten Codierstroms berechnet wird.

16. Codierstrom-Spleißverfahren nach Anspruch 10, wobei im Änderungsschritt der Wert des ersten Wiederholfelds eines Bilds vor dem Spleißpunkt im ersten Codierstrom oder der Wert des ersten Wiederholfelds eines Bilds vor dem Spleißpunkt im zweiten Codierstrom geändert wird, so dass die Rahmenstruktur des Bilds vor dem Spleißpunkt im ersten Codierstrom und die Rahmenstruktur des Bilds vor dem Spleißpunkt im zweiten Codierstrom miteinander folgerichtig werden.

17. Codierstrom-Spleißverfahren nach Anspruch 10, wobei im Änderungsschritt der Wert des ersten Kopffelds eines Bilds vor dem Spleißpunkt im ersten Codierstrom oder der Wert des ersten Kopffelds eines Bilds vor dem Spleißpunkt im zweiten Codierstrom geändert wird, so dass die Rahmenstruktur des Bilds vor dem Spleißpunkt im ersten Codierstrom und die Rahmenstruktur des Bilds vor dem Spleißpunkt im zweiten Codierstrom miteinander folgerichtig werden.

Es folgen 20 Blatt Zeichnungen

Anhängende Zeichnungen

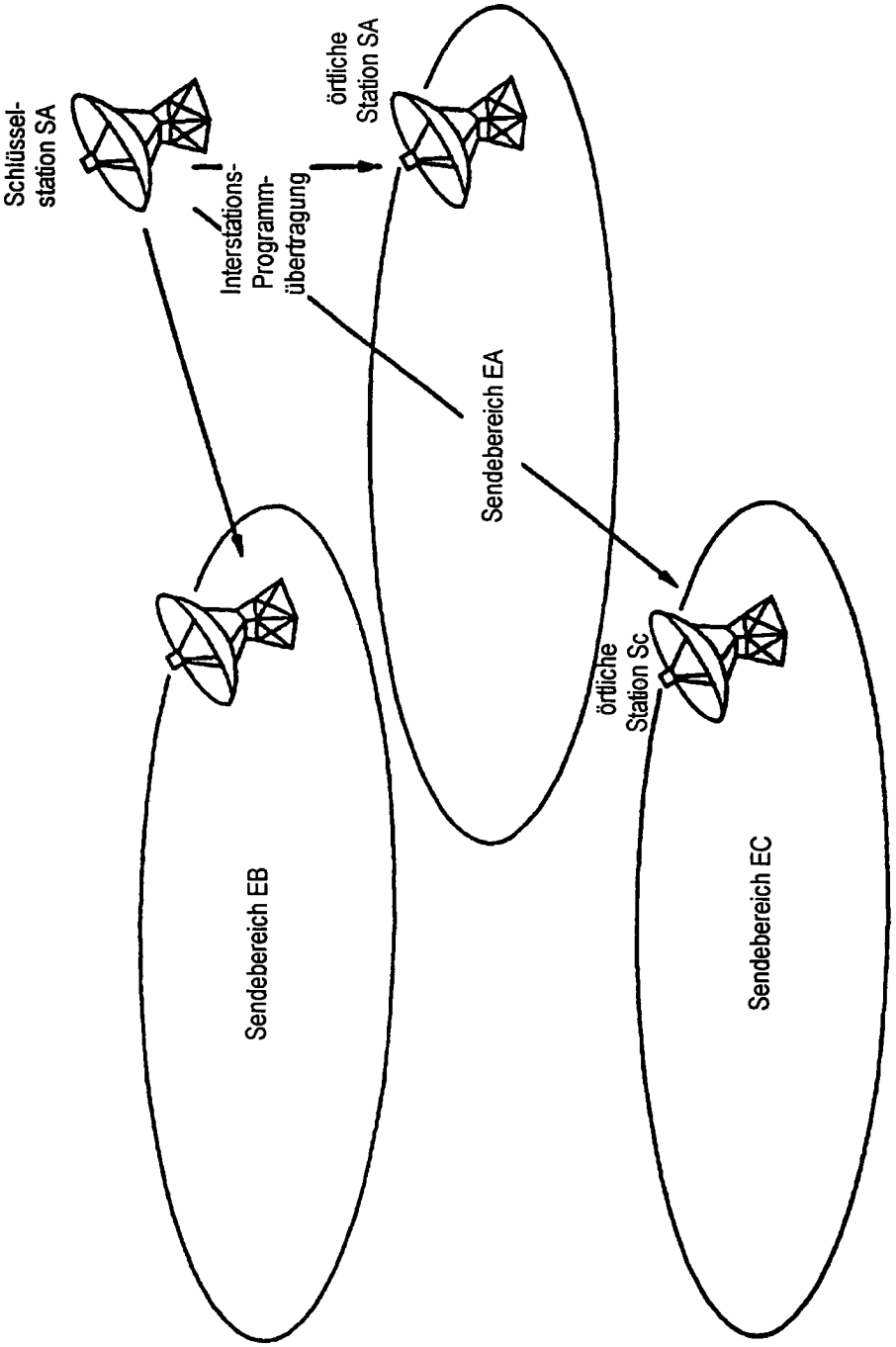


FIG.1

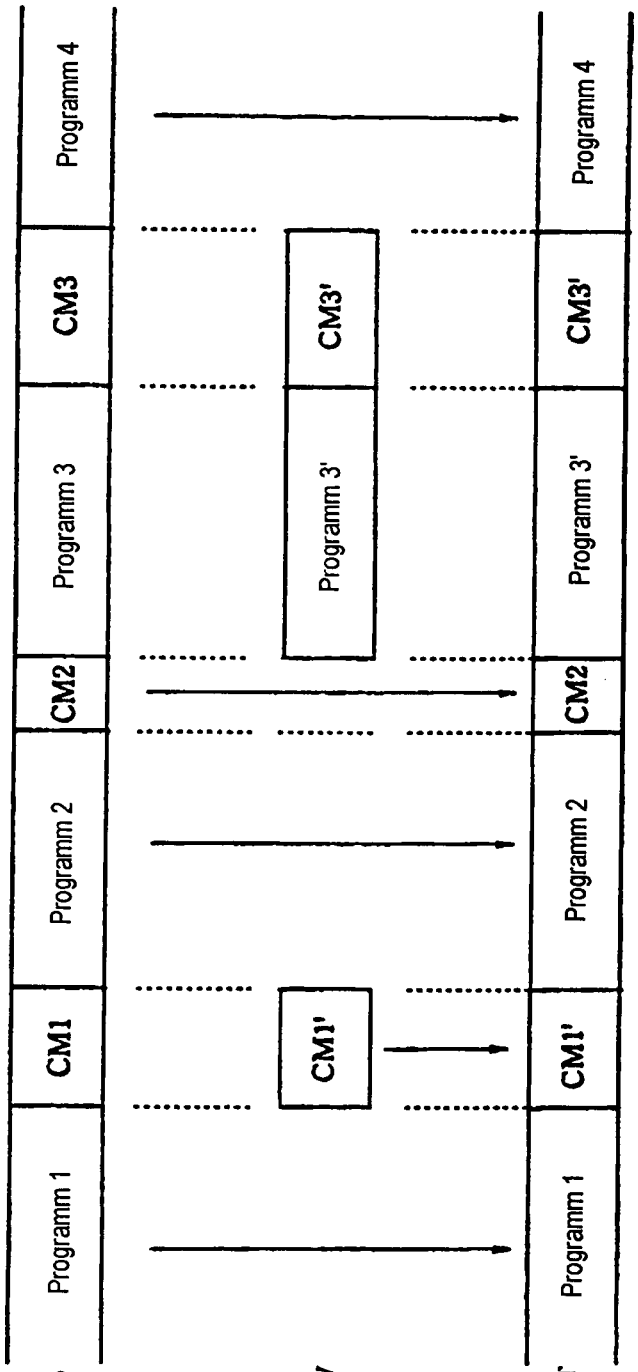


FIG.2A

FIG.2B

FIG.2C

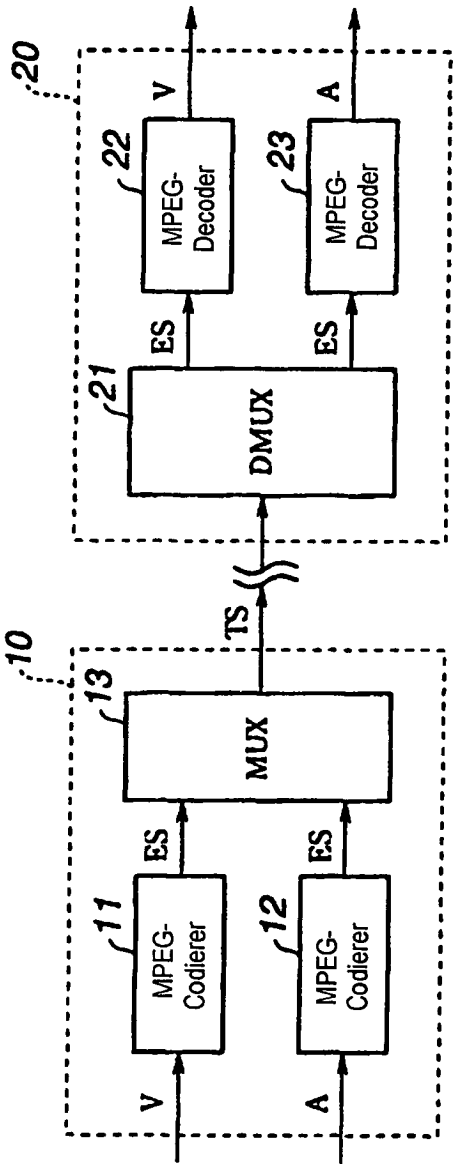

20
V
A
22
MPEG-
Decoder
23
MPEG-
Decoder
ES
ES
21
DMUX
TS
10
13
MUX
ES
ES
11
MPEG-
Codierer
12
MPEG-
Codierer
V
A

FIG.3

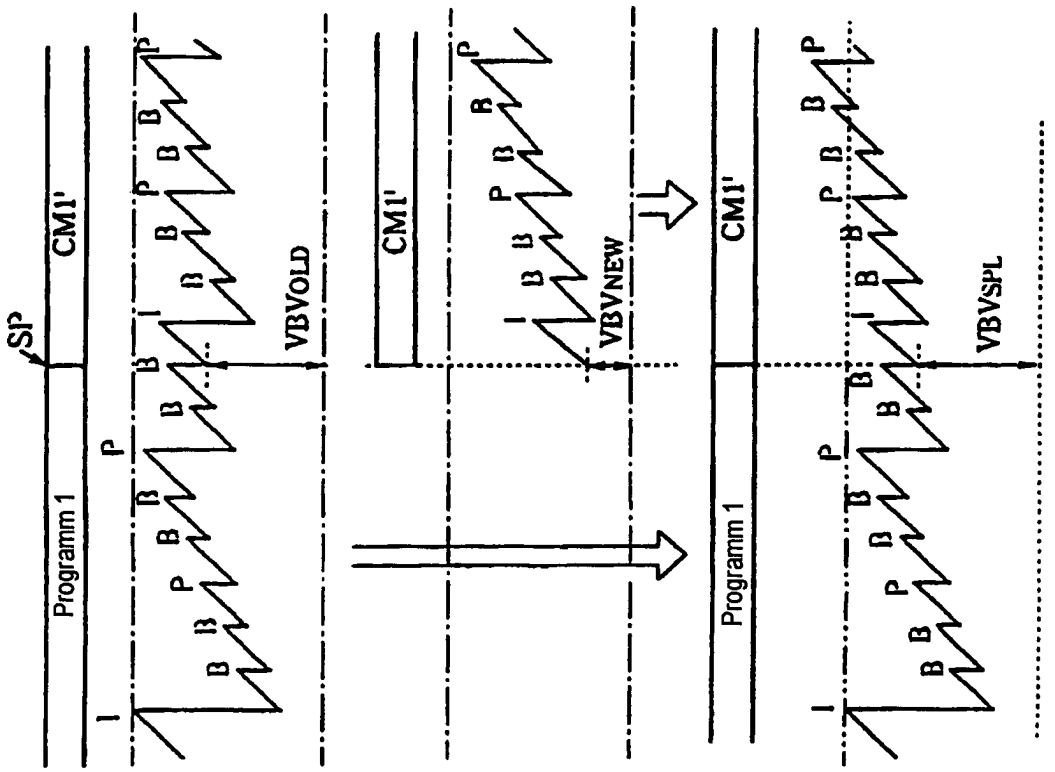


FIG.4A

FIG.4B

FIG.4C

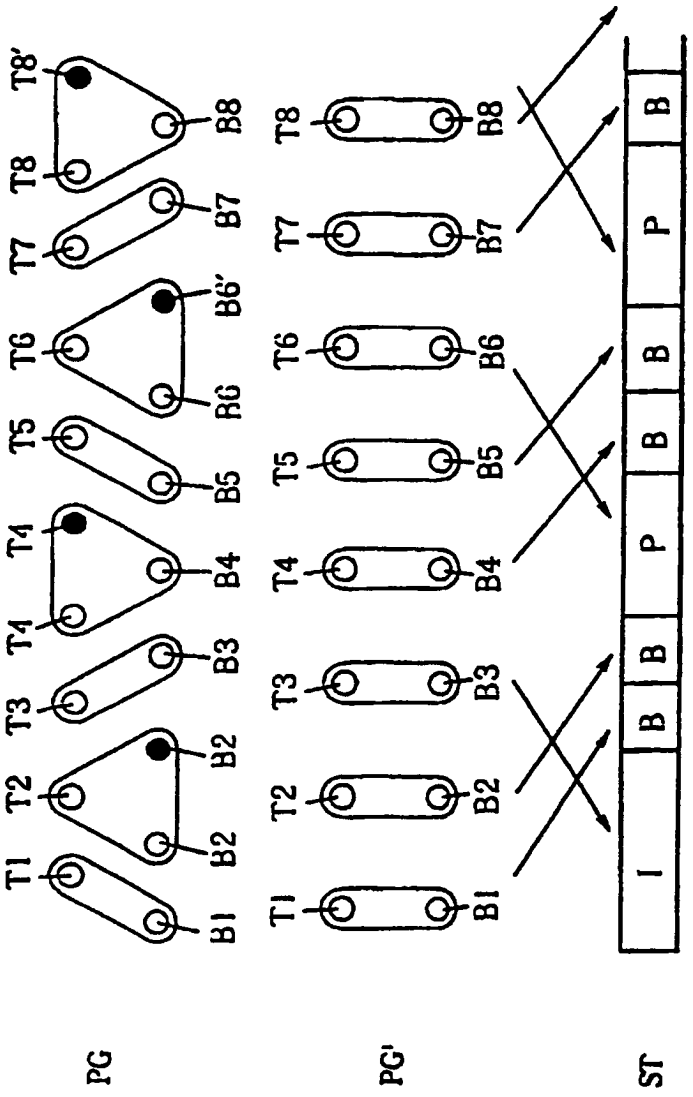

T1
T2
T3
T4
T4
T5
T6
T7
T8
T8'
B1
B2
B2
B3
B4
B5
B6
B6'
B7
B8
PG
T1
T2
T3
T4
T5
T6
T7
T8
B1
B2
B3
B4
B5
B6
B7
B8
PG'
I
B
B
P
B
B
P
B
ST

FIG.5

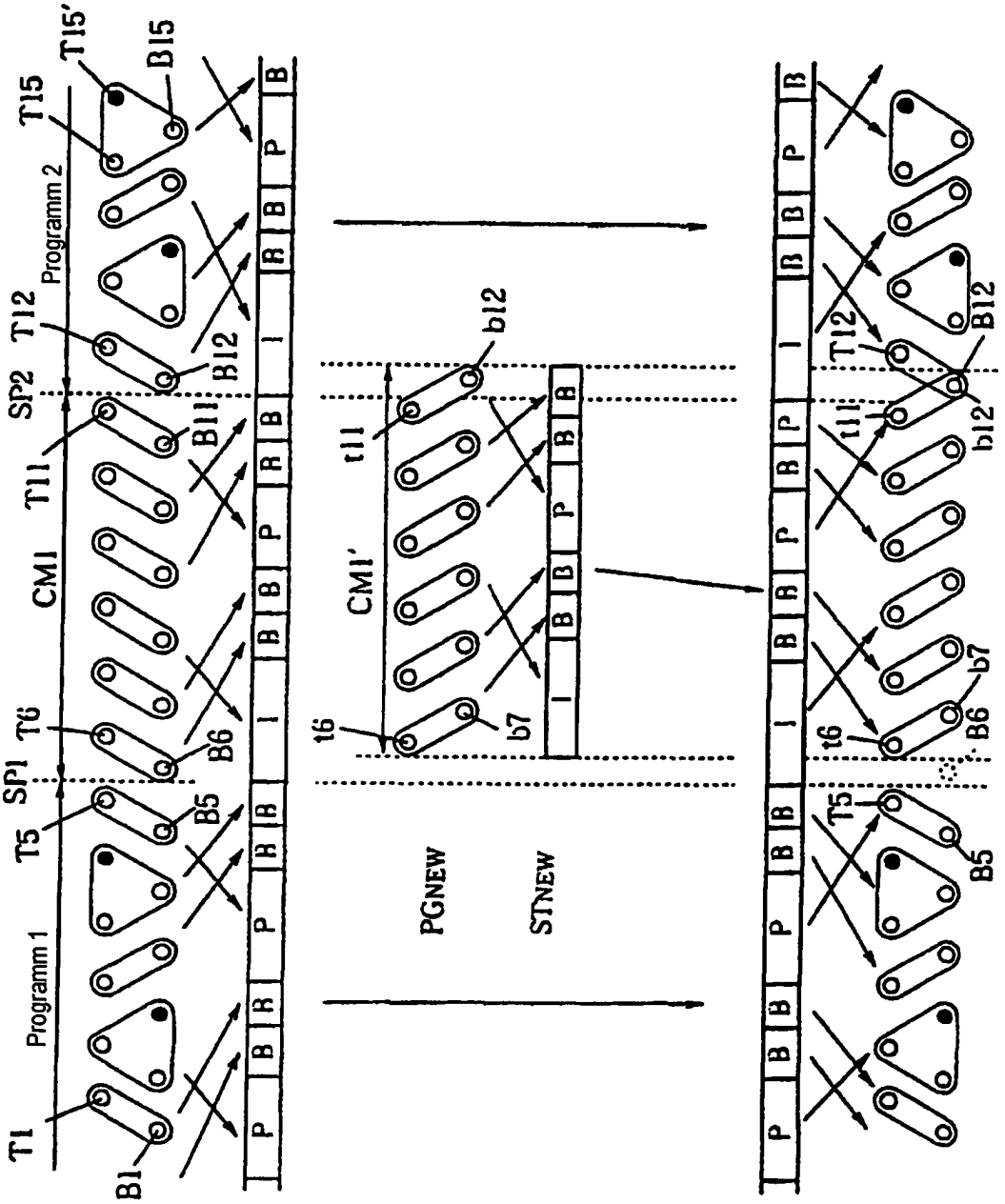


FIG.6A

FIG.6B

FIG.6C

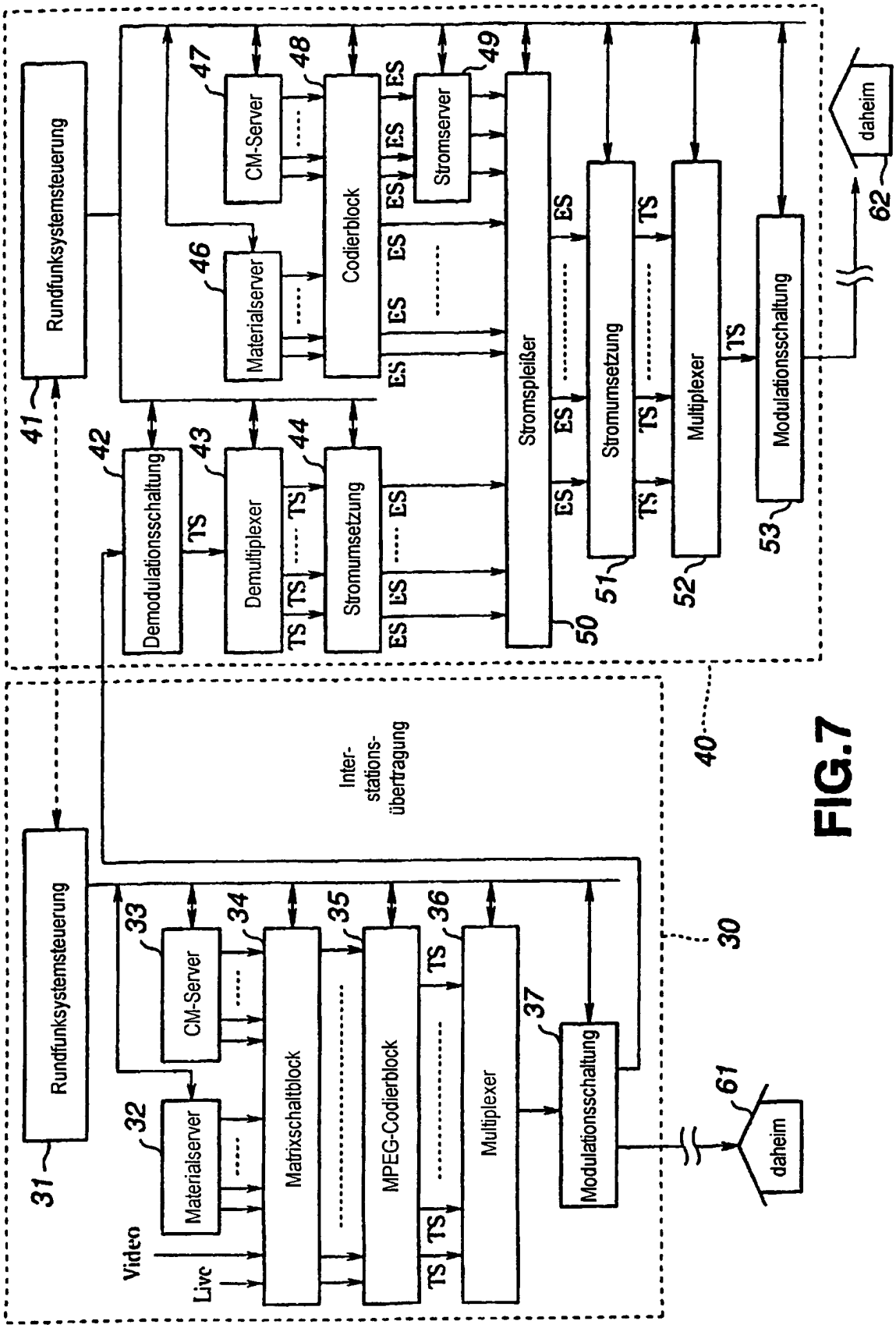

Rundfunksystemsteuerung
41
Demodulationsschaltung
42
Demultiplexer
43
Stromumsetzung
44
CM-Server
47
Materialserver
46
Codierblock
48
Stromserver
49
Stromspleißer
50
Stromumsetzung
51
Multiplexer
52
Modulationsschaltung
53
daheim
62
40
Inter-stations-übertragung
Rundfunksystemsteuerung
31
CM-Server
33
Materialserver
32
Matrixschaltblock
34
MPEG-Codierblock
35
Multiplexer
36
Modulationsschaltung
37
30
daheim
61
Video
Live
ES
TS

FIG.7

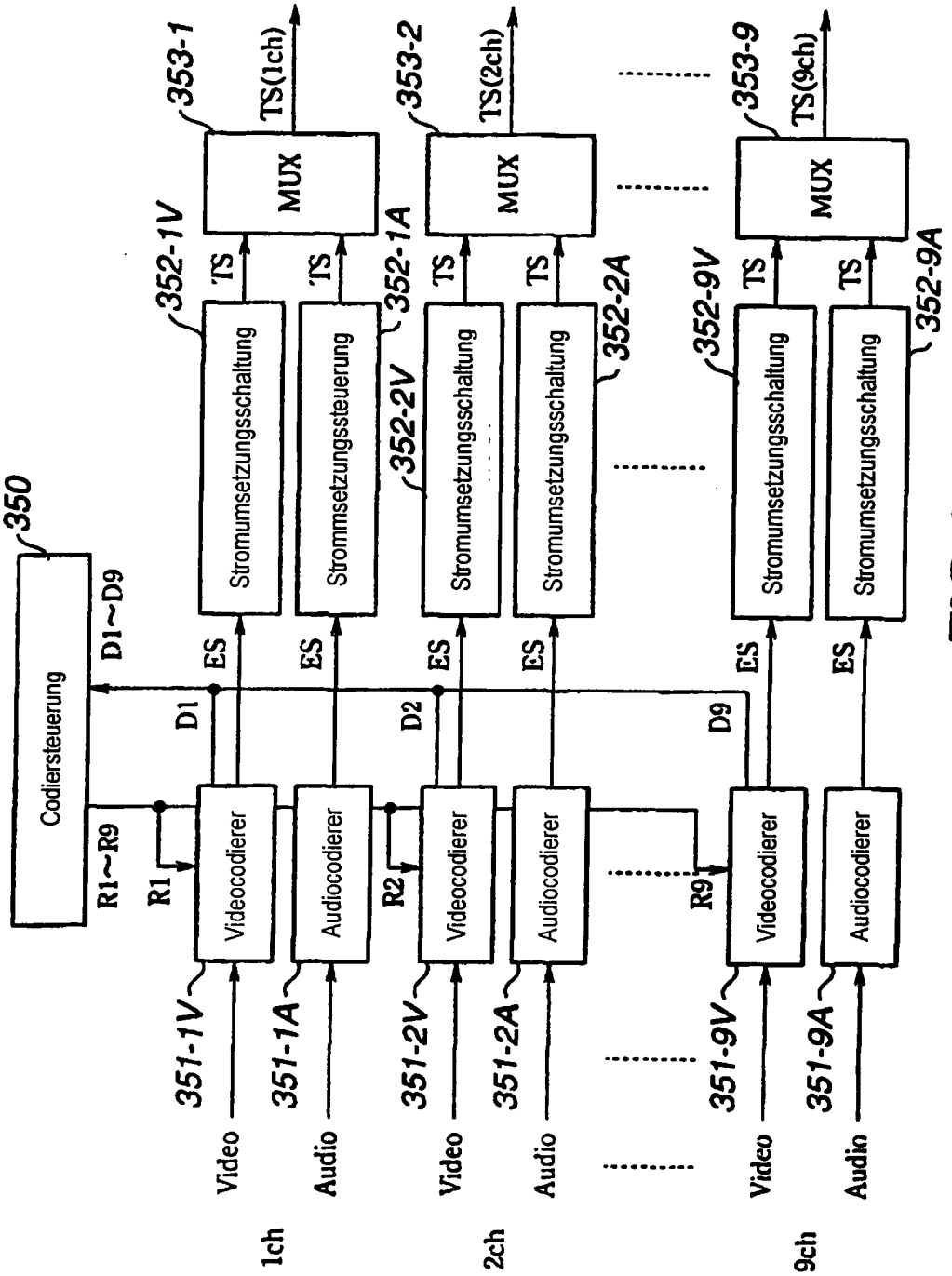

350
Codiersteuerung
D1~D9
R1~R9
R1
R2
R9
D1
D2
D9
ES
Videocodierer
Audiocodierer
351-1V
351-1A
351-2V
351-2A
351-9V
351-9A
Video
Audio
1ch
2ch
9ch
352-1V
352-1A
352-2V
352-2A
352-9V
352-9A
Stromumsetzungsschaltung
Strommsetzungssteuerung
TS
MUX
353-1
353-2
353-9
TS(1ch)
TS(2ch)
TS(9ch)

FIG.8

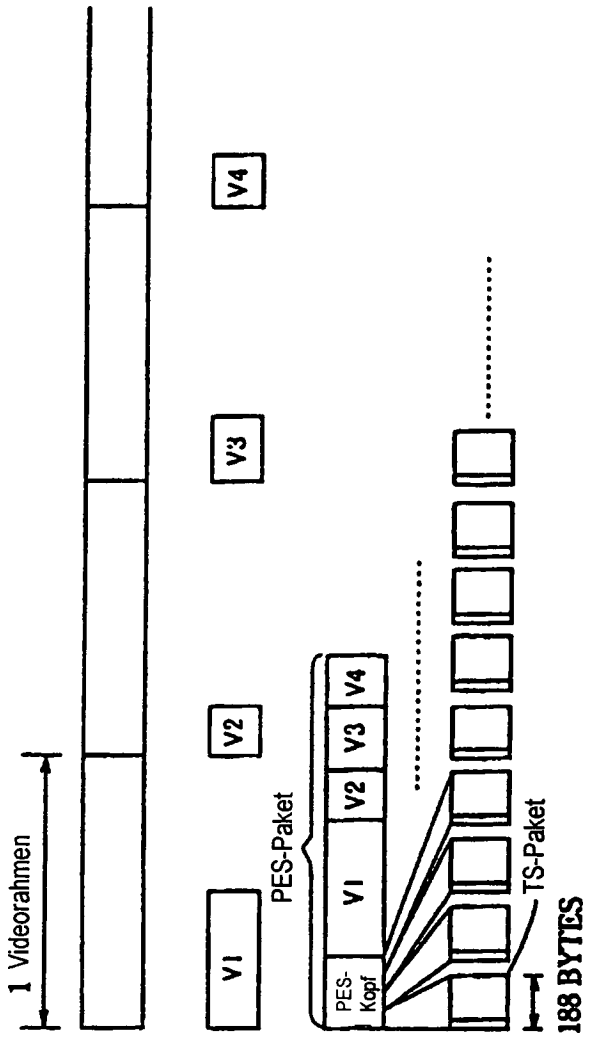


FIG.9A

FIG.9B

FIG.9C

FIG.9D

video_sequence ( ) {	Anzahl der Bits	Mnemonik
next_start_code ( )		
sequence_header ( )		
sequence_extension ( )		
do {		
extension_and_user_data (0)		
do {		
if (nextbits ( ) == group_start_code) {		
group_of_pictures_header ( )		
extension_and_user_data ( )		
}		
picture_header ( )		
picture_coding_extension ( )		
extension_and_user_data (2)		
picture_data ( )		
} while ((nextbits ( ) == picture_start_code)||		
(nextbits ( ) == group_start_code))		
if (nextbits ( ) != sequence_end_code) {		
sequence_header ( )		
sequence_extension ( )		
}		
} while (nextbits ( ) != sequence_end_code)		
sequence_end_code	32	bslbf
}		

FIG.10

Sequenzkopf

sequence_header ( ) {	Anzahl der Bits	Mnemonik
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	"1"
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantise_matrix	1	
if (load_intra_quantiser_matrix)		
intra_quantiser_matrix [64]	8 * 64	uimsbf
load_non_intra_quantiser_matrix	1	
if (load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8 * 64	uimsbf
next_start_code ( )		
}		

FIG.11

Sequenzerweiterung

sequence_extension ( ) {	BITS	Mnemonik
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indentifier	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bsalbf
vbv_buffer_size_extension	8	uimsbf
low_dalay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code ( )		
}		

FIG.12

Erweiterungsdaten
und Benutzerdaten

extension_and_user_data (i) {	Anzahl der Bits	Mnemonic
while (((il=1) its ( ) && (nextbits ( ) extension_start_code)) ll		
(nextbits ( ) ==user_data_start_code)) {		
if (nextbitS ( ) ==extension_start_code)		
extension_data (i)		
if (nextbits ( ) =user_data_start_code)		
user_data ( )		
}		
}		

FIG.13

Bildgruppen-
kopf

group_of_picture_header ( ) {	Anzahl der Bits	Mnemonik
group_start_code	32	bslbf
time_code	25	bslbf
cloused_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code ( )		
}		

FIG.14

Bildkopf

picture header ( ) {	Anzahl der Bits	Mnemonik
picture_start_code	32	bslbf
tenporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if (picture_coding_type==2||picture_coding__type==3) {		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if (picture_coding_type==3) {		
full_pel_backward_vector	1	
backward_f_code	3	uimsbf
}		
while (nextbits ( )==' 1') {		
extra_bit_picture/ * with the value " 1" * /	1	uimsbf
extra_information_picture	8	
}		
extra_bit_picture/ * with the value " 0" * /	1	uimsbf
next_start_code ( ) t		
}		

FIG.15

Bildcode-
erweiterung

picture_coding_extension ( ) {	Anzahl der Bits	Mnemonik
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code [0] [0] / * forward horizontal * /	4	uimsbf
f_code [0] [1] / * forward vertical * /	4	uimsbf
f_code [1] [0] / * backward horizontal * /	4	uimsbf
f_code [1] [1] / * backward vertical * /	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top__field__first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if (composite_display_flag) {		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code ( )		
}		

FIG.16

Bilddaten

picture_data ( ) {	Anzahl der Bits	Mnemonik
do {		
slice ( )		
} while (nextbits ( )==slice_start_code)		
next_start_code ( )		
}		

FIG.17

Syntax	Anzahl der Bits	Mnemonik
transport packet ( ) {		
sync_byte	8	bslbf
transport_error_indicator	1	bslbf
payload_unit_start_indicator	1	bslbf
transport_priority	1	bslbf
PID	13	uimsbf
transport_scrambling_control	2	bslbf
adaptaion_field_control	2	bslbf
continuity_counter	4	uimsbf
if (adaptation_field_control=='‖ adaptation_field_control=='11') {		
adaptation field ( )		
}		
if (adaptation_field_control=='‖ adaptation_field_control=='11') {		
for (i=0 ; 1<N ; i+t) {		
data_byte	8	bslbf
}		
}		
}		

FIG.18

Syntax	Anzahl der Bits	Mnemonik
adaptation_field ( ) {		
adaptation_field_length	8	uimbf
if (adaptation_field_length > 0) {		
discontinuity_indicator	1	bslbf
random access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if (PCR_flag=='1') {		
program_clock_refernce_base	33	uimsbf
reserved	6	bslbf
progrqam_clock_reference_extension	9	uimsbf
}		
if (OPCR)_flag=='1') {		
original_program_clock_reference_base	33	uimsbf
reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if (splicing_point_flag=='1') {		
splice_countdown	8	tcimsbf
}		
if (transport_private_data_flag=='1') {		
transport_private_data_length	8	uimsbf
for (i=0 ; i< transport_private_data_length ; i+t) {		
provate_data_byte	8	bslbf
}		
}		
if (adaptation_field_extension_flag=='1') {		
adaptation_field_extension_length	8	uimsbf
ltw_flag	1	bslbf
piecewise_rate_flag	1	bslbf
seamless_splice_flag	1	bslbf
reserved	5	bslbf
if (ltw_flag=='1') {		
ltw_valid_flag	1	bslbf
ltw_offset	15	uimsbf
}		
if (piecewise_rate_flag=='1') {		
reserved	2	bslbf
piecewise_rate	22	uimsbf
}		
if (seamless_splice_flag=='1') {		
splice_type	4	bslbf
DTS_next_AU[32..30]	3	bslbf
marker_bit	1	bslbf
DTS_next_AU[29..15]	15	bslbf
marker_bit	1	bslbf
DTS_next_AU[14..0]	15	bslbf
maker_bit	1	bslbf
}		
for (i=0 ; i<N ; i++) {		
reserved	8	bslbf
}		
for (i=0 ; i<N ; i++) {		
stuffing_byte	8	bslbf
}		
}		
}		

FIG.19

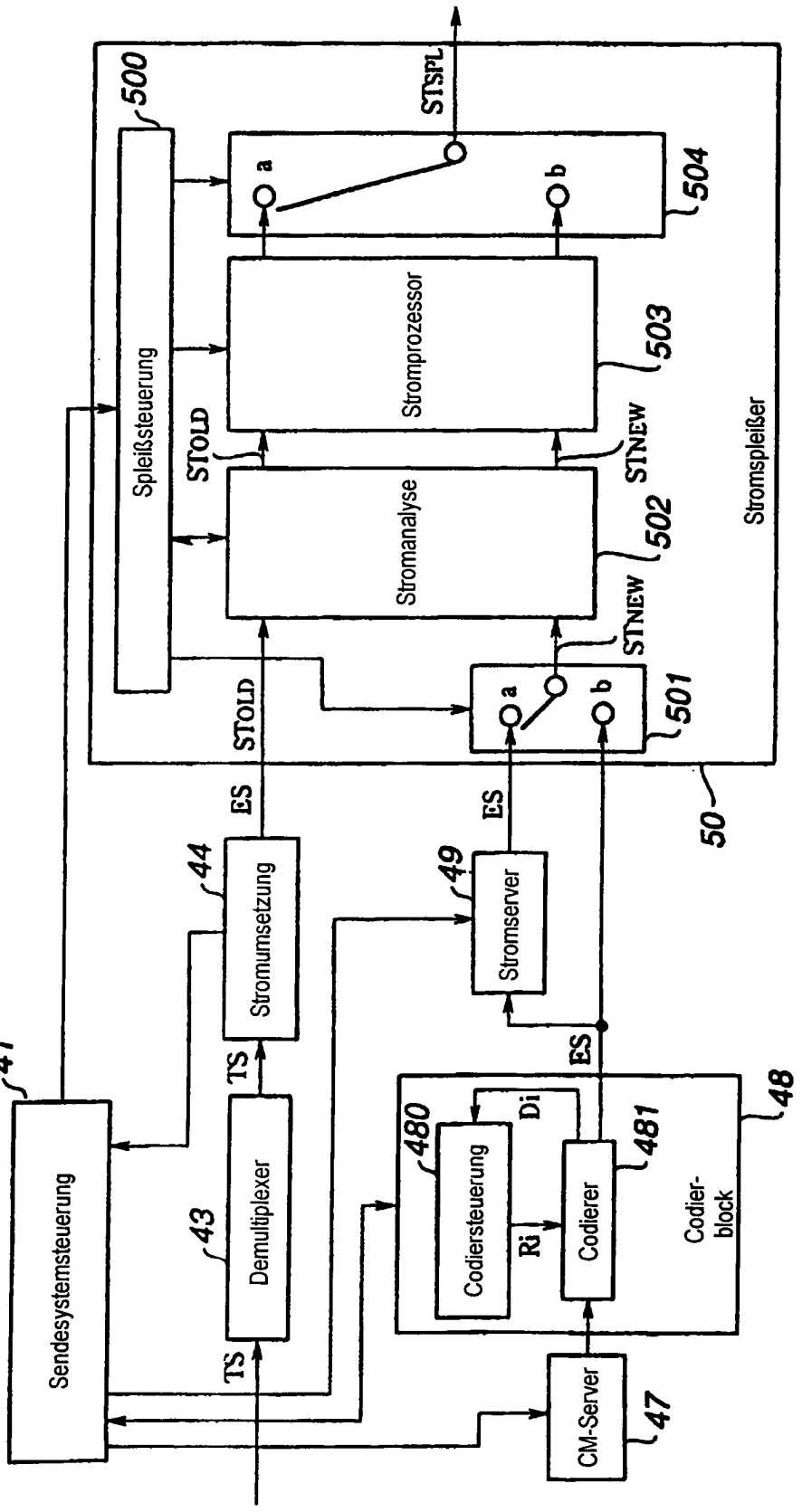

Sendesystemsteuerung
41
Demultiplexer
43
Stromumsetzung
44
Stromserver
49
CM-Server
47
Codiersteuerung
480
Codierer
481
Codier-block
48
Di
Ri
TS
TS
ES
ES
Spleißsteuerung
500
Stromanalyse
502
Stromprozessor
503
Stromspleißer
50
STOLD
STOLD
STNEW
STNEW
STSPL
a
b
501
504

FIG.20

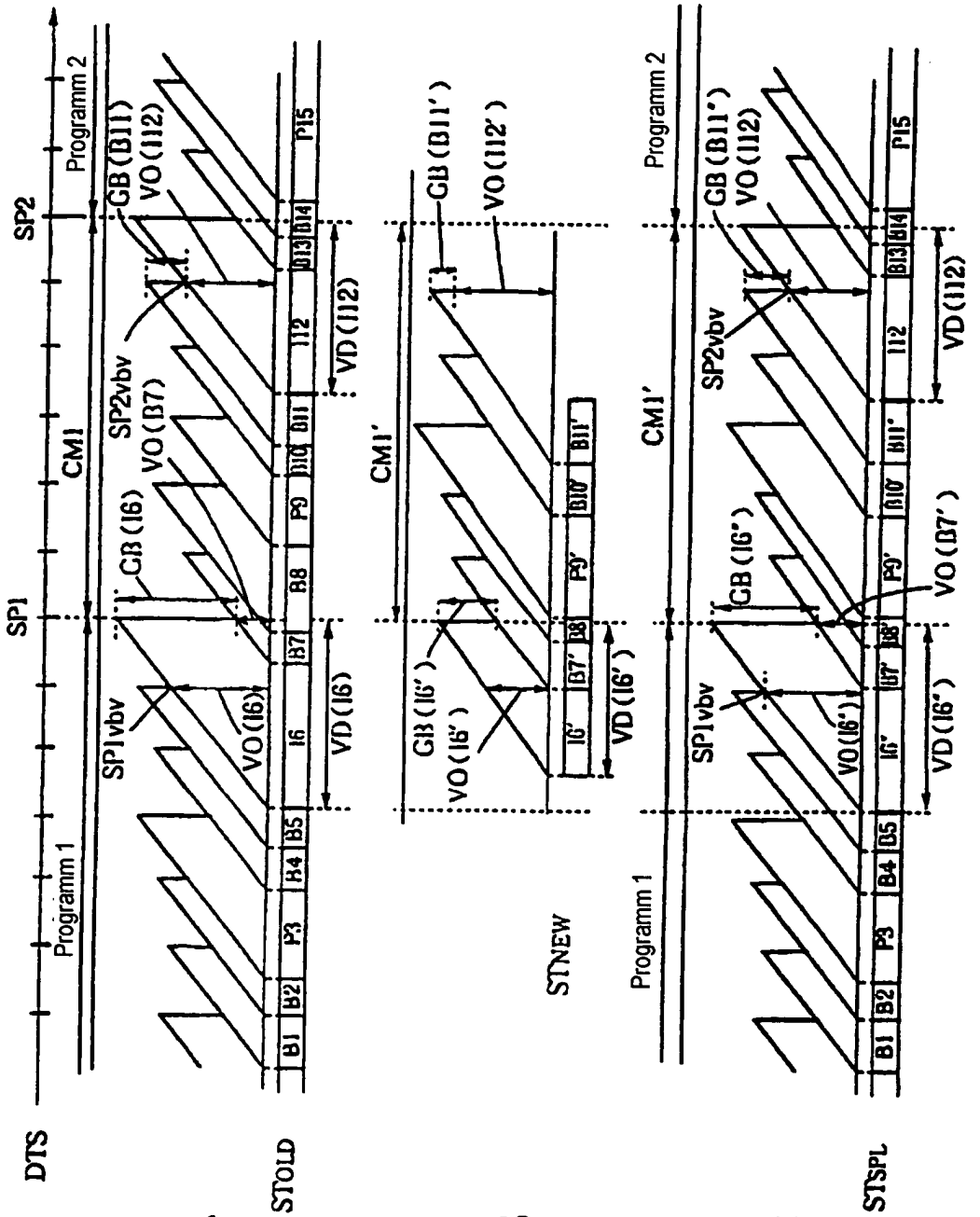


FIG.21A

FIG.21B

FIG.21C

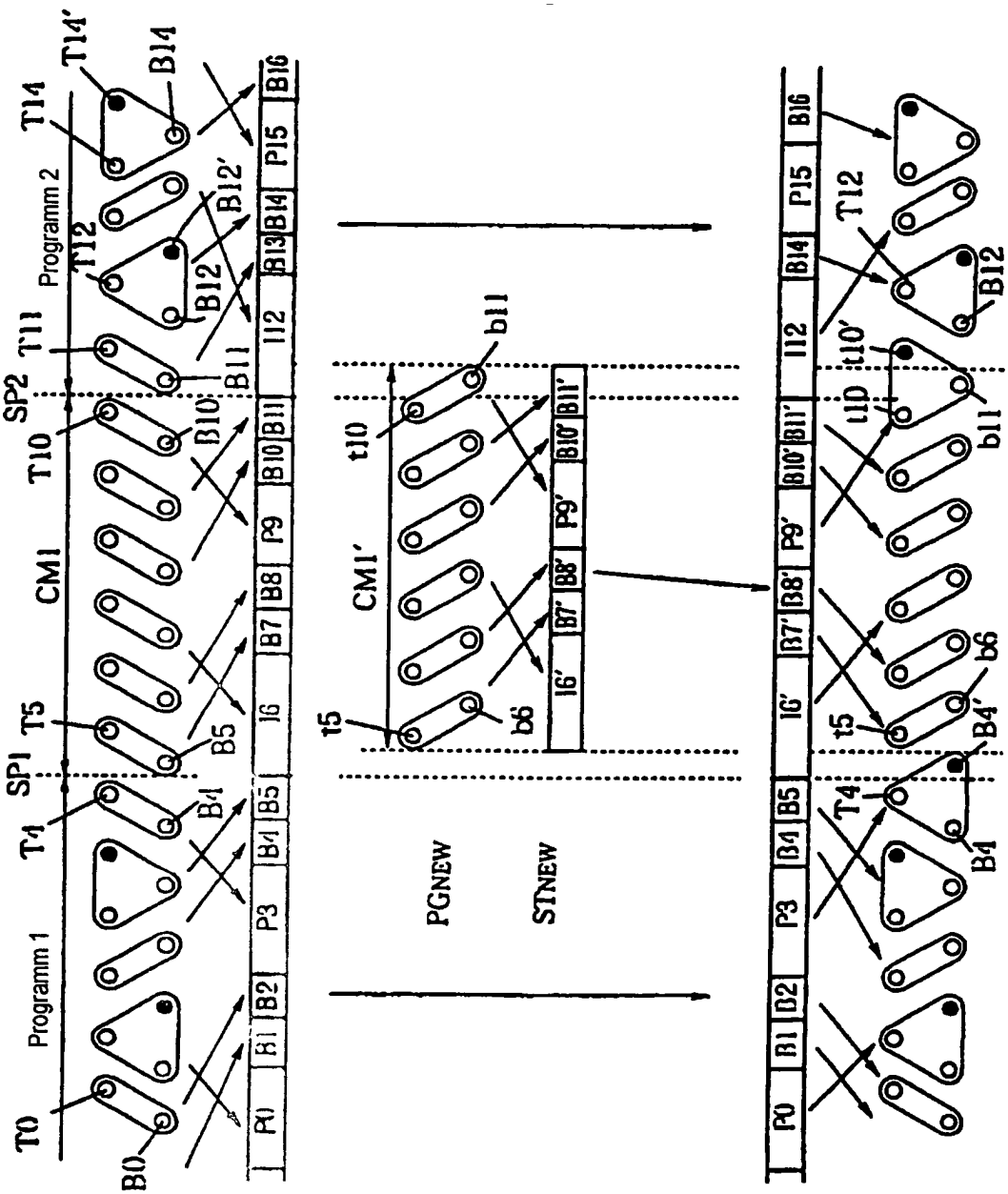


FIG.22A

FIG.22B

FIG.22C

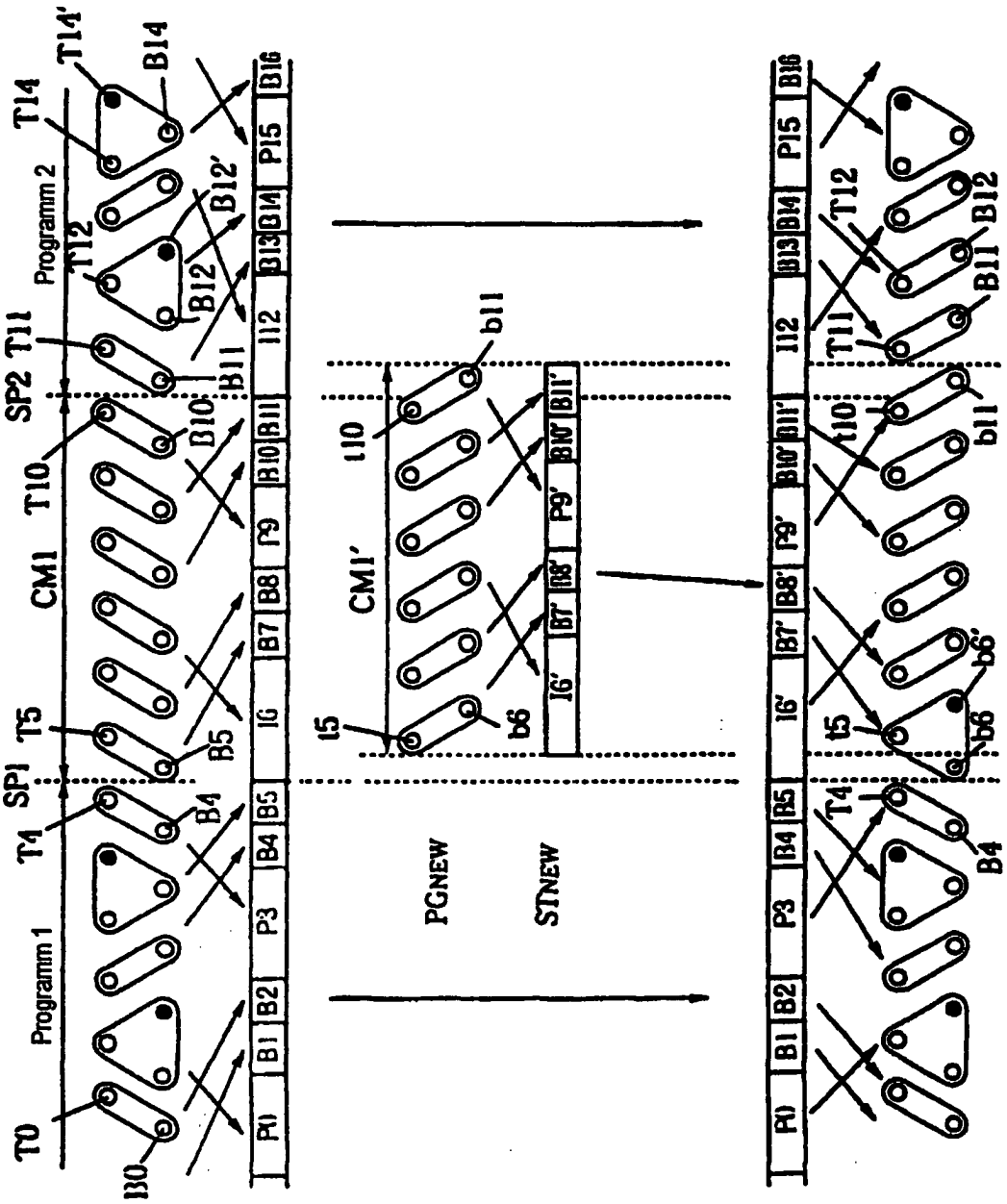


FIG.23A

FIG.23B

FIG.23C