(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
25. Juli 2002 (25.07.2002)

PCT

(10) Internationale Veröffentlichungsnummer
WO 02/058051 A2

(51) Internationale Patentklassifikation[7]: G10L 19/00

(21) Internationales Aktenzeichen: PCT/EP02/00295

(22) Internationales Anmeldedatum:
14. Januar 2002 (14.01.2002)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
101 02 154.2 18. Januar 2001 (18.01.2001) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E. V.** [DE/DE]; Leonrodstrasse 54, 80636 München (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder *(nur für US)*: **SPERSCHNEIDER, Ralph** [DE/DE]; Donato-Polli-Strasse 42, 91056 Erlangen (DE). **TEICHMANN, Bodo** [DE/DE]; Hirschenstrasse 31, 90762 Fürth (DE). **LUTZKY, Manfred** [DE/DE]; Heinrich-von-Brentano-Strasse 9, 90427 Nürnberg (DE).

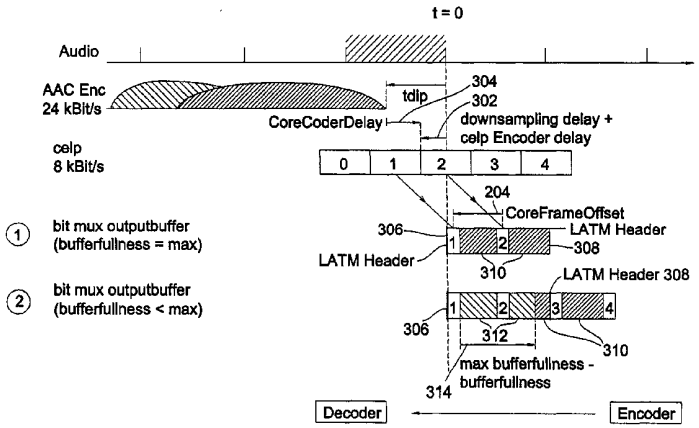
(74) Anwälte: **SCHOPPE, Fritz** usw.; Schoppe, Zimmermann, Stöckeler & Zinkler, Postfach 71 08 67, 81458 München (DE).

(81) Bestimmungsstaaten *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU,

[Fortsetzung auf der nächsten Seite]

(54) **Title:** METHOD AND DEVICE FOR PRODUCING A SCALABLE DATA STREAM, AND METHOD AND DEVICE FOR DECODING A SCALABLE DATA STREAM WHILE TAKING A BIT BANK FUNCTION INTO ACCOUNT

(54) **Bezeichnung:** VERFAHREN UND VORRICHTUNG ZUM ERZEUGEN EINES SKALIERBAREN DATENSTROMS UND VERFAHREN UND VORRICHTUNG ZUM DECODIEREN EINES SKALIERBAREN DATENSTROMS UNTER BERÜCKSICHTIGUNG EINER BITSPARKASSENFUNKTION



(57) **Abstract:** The invention relates to a method for producing a scalable data stream from one or more blocks of output data of a first encoder and from one or more blocks of output data of a second encoder. According to this method, a destination data block (306) is written for a current section of an input signal. In addition, output data (312) of the second decoder that represents a preceding section of the input signal is written, in a direction of transmission, from an encoder to a decoder behind the destination data block (306). When the output data (312) of the second encoder is written for a preceding section of the input signal, the output data (310) of the second encoder that represents the current section of the input signal is written. In order to indicate where the output data of the second encoder for the preceding section ends, and where the output data of the second encoder for the current section begins, buffer information (314) is written into the scalable data stream. In order to provide that output data of a preceding section follows a destination data block for the current section, a bit bank function can be implemented in the scalable encoder and can be easily indicated in the bit stream.

[Fortsetzung auf der nächsten Seite]

WO 02/058051 A2

SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Bestimmungsstaaten *(regional)*: ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI-Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(57) Zusammenfassung: Verfahren und Vorrichtung zum Erzeugen eines skalierbaren Datenstroms und Verfahren und Vorrichtung zum Decodieren eines skalierbaren Datenstroms unter Berücksichtigung einer Bitsparkassenfunktion ZusammenfassungIn einem Verfahren zum Erzeugen eines skalierbaren Daten-stroms aus einem oder mehreren Blöcken von Ausgangsdaten ei-nes ersten Codierers und aus einem oder mehreren Blöcken von Ausgangsdaten eines zweiten Codierers, wird ein Bestimmungs-datenblock (306) für einen aktuellen Abschnitt eines Ein-gangssignals geschrieben. Darüber hinaus werden Ausgangsda-ten (312) des zweiten Codierers, die einen vorhergehenden Abschnitt des Eingangssignals darstellen, in Übertragungs-richtung von einem Codierer zu einem Decodierer hinter den Bestimmungsdatenblock (306) geschrieben. Wenn die Ausgangs-daten (312) des zweiten Codierers für einen vorhergehenden Abschnitt des Eingangssignals geschrieben sind, werden die Ausgangsdaten (310) des zweiten Codierers, die den aktuellen Abschnitt des Eingangssignals darstellen, geschrieben. Um zu signalisieren, wo die Ausgangsdaten des zweiten Codierers für den vorhergehenden Abschnitt enden und die Ausgangsdaten des zweiten Codierers für den aktuellen Abschnitt beginnen, wer-den Pufferinformationen (314) in den skalierbaren Daten-strom geschrieben. Dadurch, daß Ausgangsdaten eines vorhergehenden Abschnitts einem Bestimmungsdatenblock für den ak-tuellen Abschnitt folgen, kann eine Bitsparkassenfunktion im skalierbaren Codierer implementiert und im Bitstrom einfach signalisiert werden.

Verfahren und Vorrichtung zum Erzeugen eines skalierbaren Datenstroms und Verfahren und Vorrichtung zum Decodieren eines skalierbaren Datenstroms unter Berücksichtigung einer Bitsparkassenfunktion

Beschreibung

Die vorliegende Erfindung bezieht sich auf skalierbare Codierer und Decodierer und insbesondere auf das Erzeugen von skalierbaren Datenströmen, durch die eine Bitsparkasse signalisiert werden kann.

Skalierbare Codierer sind in der EP 0 846 375 B1 gezeigt. Allgemein versteht man unter der Skalierbarkeit die Möglichkeit, einen Teilsatz eines Bitstroms, der ein codiertes Datensignal, wie z.B. ein Audiosignal oder ein Videosignal, darstellt, in ein nutzbares Signal zu decodieren. Diese Eigenschaft ist insbesondere dann gewünscht, wenn z.B. ein Datenübertragungskanal nicht die nötige vollständige Bandbreite zur Übertragung eines vollständigen Bitstroms zur Verfügung stellt. Andererseits ist eine unvollständige Decodierung auf einem Decodierer mit niedrigerer Komplexität möglich. Allgemein werden in der Praxis verschiedene diskrete Skalierbarkeitsschichten definiert.

Ein Beispiel für einen skalierbaren Codierer, wie er im Subpart 4 (General Audio) des Parts 3 (Audio) des MPEG-4 Standards (ISO/IEC 14496-3:1999 Subpart 4) definiert ist, ist in Fig. 1 gezeigt. Ein zu codierendes Audiosignal s(t) wird eingangsseitig in den skalierbaren Codierer eingespeist. Der in Fig. 1 gezeigte skalierbare Codierer enthält einen ersten Codierer 12, der ein MPEG-Celp-Codierer ist. Der zweite Codierer 14 ist ein AAC-Codierer, der eine hochqualitative Audiocodierung liefert und im Standard MPEG-2 AAC (ISO/IEC 13818) definiert ist. Der Celp-Codierer 12 liefert über eine Ausgangsleitung 16 eine erste Skalierungsschicht, während der AAC-Codierer 14 über eine zweite Aus-

gangsleitung 18 eine zweite Skalierungsschicht zu einem Bitstrom-Multiplexer (BitMux) 20 liefert. Ausgangsseitig gibt der Bitstrom-Multiplexer dann einen MPEG-4-LATM-Bitstrom 22 aus (LATM = Low-Overhead MPEG-4 Audio Transport Multiplex). Das LATM-Format ist im Abschnitt 6.5 des Parts 3 (Audio) der ersten Ergänzung zum MPEG-4 Standard (ISO/IEC 14496-3:1999/AMD1:2000) beschrieben.

Der skalierbare Audiocodierer umfaßt ferner einige weitere Elemente. Zunächst existiert eine Verzögerungsstufe 24 im AAC-Zweig und eine Verzögerungsstufe 26 im Celp-Zweig. Durch beide Verzögerungsstufen kann eine optionale Verzögerung für den jeweiligen Zweig eingestellt werden. Der Verzögerungsstufe 26 des Celp-Zweigs ist eine Downsampling-Stufe 28 nachgeschaltet, um die Abtastrate des Eingangssignals s(t) an die von dem Celp-Codierer geforderte Abtastrate anzupassen. Dem Celp-Codierer 12 nachgeschaltet ist ein inverser Celp-Decodierer 30, wobei das Celp-codierte/decodierte Signal einer Upsampling-Stufe 32 zugeführt wird. Das upgesampelte Signal wird dann einer weiteren Verzögerungsstufe 34, die im MPEG-4-Standard mit "Core Coder Delay" bezeichnet ist, zugeführt.

Die Stufe CoreDoderDelay 34 hat folgende Funktion. Ist die Verzögerung auf Null eingestellt, so verarbeiten der erste Codierer 14 und der zweite Codierer 16 in einem sogenannten Superframe exakt dieselben Abtastwerte des Audioeingangssignals. Ein Superframe kann beispielsweise aus drei AAC-Frames bestehen, die zusammen eine gewisse Anzahl von Abtastwerten Nr. x bis Nr. y des Audiosignals darstellen. Der Superframe umfaßt ferner z. B. 8 CELP-Blöcke, die im Falle von CoreCoderDelay = 0 dieselbe Anzahl von Abtastwerten und auch dieselben Abtastwerte Nr. x bis Nr. y darstellen.

Ist dagegen ein CoreCoderDelay D als Zeitgröße ungleich Null eingestellt, so stellen die drei Blöcke von AAC Frames dennoch die gleichen Abtastwerte Nr. x bis Nr. y dar. Die acht Blöcke von CELP-Frames stellen dagegen Abtastwerte Nr. x -

Fs D bis Nr. y - Fs D dar, wobei Fs die Abtastfrequenz des Eingangssignals ist.

Die aktuellen Zeitabschnitte des Eingangssignals in einem Superframe für die AAC-Blöcke und die CELP-Blöcke können somit entweder identisch sein, wenn CoreCoderDelay D = 0 ist, oder aber im Falle von D ungleich Null um CoreCoderDelay zueinander verschoben sein. Für die nachfolgenden Ausführungen wird jedoch aus Einfachheitsgründen ohne Einschränkung der Allgemeinheit ein CoreCoderDelay = 0 angenommen, so daß der aktuelle Zeitabschnitt des Eingangssignals für den ersten Coder und der aktuelle Zeitabschnitt für den zweiten Coder identisch sind. Allgemein besteht für einen Superframe jedoch lediglich die Anforderung, daß der/die AAC-Block/Blöcke und der/die CELP-Blöcke in einem Superframe dieselbe Anzahl von Abtastwerten darstellen, wobei die Abtastwerte an sich nicht unbedingt die identischen sein müssen, sondern auch um CoreCoderDelay zueinander verschoben sein können.

Es sei angemerkt, daß der Celp-Codierer einen Abschnitt des Eingangssignals s(t) je nach Konfiguration schneller verarbeitet als der AAC-Codierer 14. In dem AAC-Zweig ist der Optionalverzögerungsstufe 24 eine Blockentscheidungsstufe 26 nachgeschaltet, die u. a. feststellt, ob zum Fenstern des Eingangssignals s(t) kurze oder lange Fenster zu verwenden sind, wobei für stark transiente Signale kurze Fenster zu wählen sind, während für weniger transiente Signale lange Fenster vorgezogen werden, da bei ihnen das Verhältnis zwischen Nutzdatenmenge und Seiteninformationen besser als bei kurzen Fenstern ist.

Durch die Blockentscheidungsstufe 26 wird im vorliegenden Beispiel eine feste Verzögerung um z. B. das 5/8-fache eines Blocks durchführt. Dies wird in der Technik als Look-Ahead-Funktion bezeichnet. Die Blockentscheidungsstufe muß bereits um eine gewisse Zeit vorausschauen, um überhaupt feststellen zu können, ob in der Zukunft transiente Signale sind, die

mit kurzen Fenstern codiert werden müssen. Hierauf wird sowohl das entsprechende Signal im Celp-Zweig als auch das Signal im AAC-Zweig einer Einrichtung zum Umsetzen der zeitlichen Darstellung in eine spektrale Darstellung zugeführt, welche in Fig. 1 mit MDCT 36 bzw. 38 bezeichnet ist (MDCT = Modified Discrete Cosine Transform = Modifizierte Diskrete Cosinus-Transformation). Die Ausgangssignale der MDCT-Blöcke 36, 38 werden dann einem Subtrahierer 40 zugeführt.

An dieser Stelle müssen zeitlich zusammengehörige Abtastwerte vorliegen, d. h. das Delay muß in beiden Zweigen identisch sein.

Der darauffolgende Block 44 stellt fest, ob es günstiger ist, das Eingangssignal an sich dem AAC-Codierer 14 zuzuführen. Dies wird über den Umgehungszweig 42 ermöglicht. Wenn jedoch festgestellt wird, daß das Differenzsignal am Ausgang des Subtrahierers 40 z.B. energiemäßig kleiner ist als das von dem MDCT-Block 38 ausgegebene Signal, so wird nicht das ursprüngliche Signal, sondern das Differenzsignal genommen, um durch den AAC-Codierer 14 codiert zu werden, um schließlich die zweite Skalierungsschicht 18 zu bilden. Dieser Vergleich kann bandweise durchgeführt werden, was durch eine frequenzselektive Schalteinrichtung (FSS) 44 angedeutet ist. Die näheren Funktionen der einzelnen Elemente sind in der Technik bekannt und beispielsweise im MPEG-4-Standard sowie in weiteren MPEG-Standards beschrieben.

Ein wesentliches Merkmal beim MPEG-4-Standard bzw. auch bei anderen Codierer-Standards ist, daß die Übertragung des komprimierten Datensignals über einen Kanal mit konstanter Bitrate erfolgen soll. Alle High-Quality-Audiocodecs arbeiten blockbasiert, d.h. sie verarbeiten Blöcke von Audiodaten (Größenordnung 480-1024 Samples) zu Stücken eines komprimierten Bitstroms, welche auch als Frames bezeichnet werden. Das Bitstromformat muß dabei so aufgebaut sein, daß ein Decodierer ohne A-Priori-Informationen, wo ein Frame beginnt,

in der Lage ist, den Anfang eines Frames zu erkennen um mit einer möglichst geringen Verzögerung die Ausgabe der decodierten Audiosignaldaten zu beginnen. Daher beginnt jeder Header oder Bestimmungsdatenblock eines Frames mit einem bestimmten Synchronisationswort, nach dem in einem kontinuierlichen Bitstrom gesucht werden kann. Weitere übliche Bestandteile im Datenstrom neben dem Bestimmungsdatenblock sind die Hauptdaten oder "Payload Data" der einzelnen Layer, in denen die eigentlichen komprimierten Audiodaten enthalten sind.

Fig. 4 zeigt ein Bitstromformat mit fester Framelänge. In diesem Bitstromformat werden die Header oder Bestimmungsdatenblöcke äquidistant in den Bitstrom eingefügt. Die zu diesem Header zugehörigen Seiteninformationen ("Side Information") und Hauptdaten (Main Data) folgen unmittelbar dahinter. Die Länge, d.h. Bitanzahl, für die Hauptdaten ist in jedem Frame gleich. Ein solches Bitstromformat, wie es in Fig. 4 gezeigt wird, wird beispielsweise bei MPEG-Layer 2 oder MPEG-CELP verwendet.

Fig. 5 zeigt ein anderes Bitstromformat mit einer festen Framelänge und einem Backpointer oder Rückwärtszeiger. Bei diesem Bitstromformat sind der Header und die Seiteninformationen wie bei dem Format, das in Fig. 4 gezeigt ist, äquidistant angeordnet. Der Beginn der zugehörigen Hauptdaten erfolgt allerdings nur im Ausnahmefall unmittelbar im Anschluß an einen Header. In den meisten Fällen ist der Beginn in einem der vorherigen Frames. Die Anzahl an Bits, um die der Beginn der Hauptdaten im Bitstrom verschoben ist, wird durch die Seiteninformations-Variable Backpointer übertragen. Das Ende dieser Hauptdaten kann in diesem Frame liegen oder in einem vorherigen Frame. Die Länge der Hauptdaten ist damit nicht mehr konstant. Somit kann die Anzahl der Bits, mit denen ein Block codiert wird, an die Eigenschaften des Signals angepaßt werden. Gleichzeitig kann jedoch eine konstante Bitrate erreicht werden. Diese Technik wird "Bitsparkasse" genannt und vergrößert das theoretische

Delay in der Übertragungskette. Ein solches Bitstromformat wird beispielsweise bei MPEG Layer 3 (MP3) eingesetzt. Die Technik der Bitsparkasse ist ebenfalls in dem Standard MPEG Layer 3 beschrieben.

Allgemein gesagt stellt die Bitsparkasse einen Buffer von Bits dar, die eingesetzt werden können, um zum Codieren eines Blocks von zeitlichen Abtastwerten mehr Bits zur Verfügung zu stellen, als eigentlich durch die konstante Ausgangsdatenrate erlaubt sind. Die Technik der Bitsparkasse trägt der Tatsache Rechnung, daß manche Blöcke von Audioabtastwerten mit weniger Bits als durch die konstante Übertragungsrate vorgegeben codiert werden können, so daß sich durch diese Blöcke die Bitsparkasse füllt, während wieder andere Blöcke von Audioabtastwerten psychoakustische Eigenschaften haben, die keine so große Kompression erlauben, so daß für diese Blöcke zum störungsarmen bzw. störungsfreien Codieren die zur Verfügung stehenden Bits eigentlich nicht ausreichen würden. Die benötigten überzähligen Bits werden aus der Bitsparkasse genommen, so daß sich die Bitsparkasse bei solchen Blöcken leert.

Ein solches Audiosignal könnte jedoch auch, wie es in Fig. 6 gezeigt ist, durch ein Format mit variabler Framelänge übertragen werden. Bei dem Bitstromformat "Variable Framelänge", wie es in Fig. 6 dargestellt ist, wird die feste Reihenfolge der Bitstromelemente Header, Seiteninformationen und Hauptdaten wie bei der "Festen Framelänge" eingehalten. Da die Länge der Hauptdaten nicht konstant ist, kann auch hier die Bitsparkassentechnik eingesetzt werden, es werden jedoch keine Backpointer wie in Fig. 5 benötigt. Ein Beispiel für ein Bitstromformat, wie es in Fig. 6 dargestellt ist, ist das Transportformat ADTS (Audio Data Transport Stream), wie es im Standard MPEG 2 AAC definiert ist.

Es sei darauf hingewiesen, daß die vorher genannten Codierer alle keine skalierbaren Codierer sind, sondern lediglich einen einzigen Audiocodierer umfassen.

In MPEG 4 ist die Kombination verschiedener Codierer/Decodierer zu einem skalierbaren Codierer/Decodierer vorgesehen. So ist es möglich und sinnvoll, einen Celp-Sprachcodierer als ersten Codierer mit einem AAC-Codierer für die weitere bzw. die weiteren Skalierungsschichten zu kombinieren und in einem Bitstrom zu verpacken. Der Sinn dieser Kombination besteht darin, daß die Möglichkeit offen steht, entweder alle Skalierungsschichten oder Layer zu decodieren und damit eine bestmögliche Audioqualität zu erreichen, oder auch Teile davon, unter Umständen auch nur die erste Skalierungsschicht mit der entsprechenden eingeschränkten Audioqualität. Gründe für die alleinige Decodierung der untersten Skalierungsschicht können sein, daß wegen zu kleiner Bandbreite des Übertragungskanals der Decodierer nur die erste Skalierungsschicht des Bitstroms erhalten hat. Deswegen werden bei der Übertragung die Anteile der ersten Skalierungsschicht im Bitstrom gegenüber der zweiten und den weiteren Skalierungsschichten bevorrechtigt, wodurch bei Kapazitätsengpässen im Übertragungsnetz die Übertragung der ersten Skalierungsschicht sichergestellt wird, während die zweite Skalierungsschicht eventuell ganz oder teilweise verloren geht.

Ein weiterer Grund kann darin liegen, daß ein Decodierer ein möglichst geringes Codec-Delay erreichen möchte und deswegen nur die erste Skalierungsschicht decodiert. Es sei darauf hingewiesen, daß das Codec-Delay eine Celp-Codecs im allgemeinen signifikant kleiner als das Delay des AAC-Codecs ist.

In MPEG 4 Version 2 ist das Transportformat LATM standardisiert, welches unter anderem auch skalierbare Datenströme übertragen kann.

Im nachfolgenden wird auf Fig. 2a Bezug genommen. Fig. 2a ist eine schematische Darstellung der Abtastwerte des Eingangssignals s(t). Das Eingangssignal kann in verschiedene aufeinanderfolgende Abschnitte 0, 1, 2, 3 eingeteilt werden, wobei jeder Abschnitt eine bestimmte feste Anzahl von zeit-

lichen Abtastwerten hat. Üblicherweise verarbeitet der AAC-Codierer 14 (Fig. 1) einen gesamten Abschnitt 0, 1, 2 oder 3, um für diesen Abschnitt ein codiertes Datensignal zu liefern. Der Celp-Codierer 12 (Fig. 1) verarbeitet jedoch üblicherweise eine geringere Menge an zeitlichen Abtastwerten pro Codierungsschritt. So ist in Fig. 2b beispielhaft gezeigt, daß der Celp-Codierer bzw. allgemein gesagt der erste Codierer oder Coder 1 eine Blocklänge hat, die ein Viertel der Blocklänge des zweiten Codierers beträgt. Es sei darauf hingewiesen, daß diese Aufteilung völlig willkürlich ist. Die Blocklänge des ersten Codierers könnte auch halb so groß sein, könnte jedoch auch ein Elftel der Blocklänge des zweiten Codierers betragen. Somit wird der erste Codierer aus dem Abschnitt des Eingangssignals vier Blöcke erzeugen (11, 12, 13, 14), aus denen der zweite Codierer einen Block von Daten liefert. In Fig. 2c ist ein übliches LATM-Bitstromformat gezeigt.

Ein Superframe kann verschiedene Verhältnisse von Anzahl von AAC-Frames zu Anzahl von CELP-Frames haben, wie es in MPEG 4 tabellarisch dargelegt ist. So kann ein Superframe z. B. einen AAC Block und 1 bis 12 CELP-Blöcke, 3 AAC-Blöcke und 8 CELP-Blöcke aber auch z. B. mehr AAC-Blöcke als CELP-Blöcke je nach Konfiguration aufweisen. Ein LATM-Frame, der einen LATM-Bestimmungsdatenblock hat, umfaßt einen Superframe oder auch mehrere Superframes.

Es wird beispielhaft die Erzeugung des durch den Header 1 eröffneten LATM-Frames beschrieben. Zunächst werden die Ausgangsdatenblöcke 11, 12, 13, 14 des Celp-Codierers 12 (Fig. 1) erzeugt und zwischengespeichert. Parallel dazu wird der Ausgangsdatenblock des AAC-Codierers, der in Fig. 2c mit "1" bezeichnet ist, erzeugt. Dann, wenn der Ausgangsdatenblock des AAC-Codierers erzeugt ist, wird erst der Bestimmungsdatenblock (Header 1) geschrieben. Je nach Konvention kann dann unmittelbar hinter den Header 1 der als erstes erzeugte Ausgangsdatenblock des ersten Codierers, der in Fig. 2c mit 11 bezeichnet ist, geschrieben, d. h.

übertragen, werden. Es wird üblicherweise (in Anbetracht geringer erforderlicher Signalisierungsinformationen) zum weiteren Schreiben bzw. Übertragen des Bitstroms ein äquidistanter Abstand der Ausgangsdatenblöcke des ersten Codierers gewählt, wie es in Fig. 2c dargestellt ist. Dies bedeutet, daß nach dem Schreiben bzw. Übertragen des Blocks 11 der zweite Ausgangsdatenblock 12 des ersten Codierers, dann der dritte Ausgangsdatenblock 13 des ersten Codierers und dann der vierte Ausgangsdatenblock 14 des ersten Codierers in äquidistanten Abständen geschrieben bzw. übertragen werden. Der Ausgangsdatenblock 1 des zweiten Codierers wird während der Übertragung in die verbleibenden Lücken eingefüllt. Dann ist ein LATM-Frame fertig geschrieben, d. h. fertig übertragen.

Nachteilig an den in den Figuren 4 bis 6 dargestellten bekannten Bitstromformaten ist die Tatsache, daß dieselben nicht für skalierbare Datenströme geeignet sind.

Ein weiterer Nachteil der bekannten Bitstromformate besteht darin, daß kein Bitstromformat für einen skalierbaren Datenstrom existiert, so daß die Bitsparkassenfunktion für skalierbare Datenströme mit Ausgangsdaten von Codierern mit unterschiedlicher Zeitbasis, insbesondere für die Kombination AAC-Codierer und CELP-Codierer einer skalierbaren Codiervorrichtung derzeit nicht nutzbar gemacht werden kann. Da jedoch eine konstante Übertragungsrate gefordert wird, der AAC-Codierer jedoch Blöcke unterschiedlicher Länge abhängig von den Eigenschaften des codierten Signals ausgibt, kann durchaus der Fall auftreten, daß der AAC-Codierer zur Codierung eines Abschnitts des Zeitsignals mehr Bits als durch die Übertragungsrate vorgegeben, benötigt, während er für einen anderen Abschnitt wieder weniger Bits als durch die Ausgangsdatenrate vorgegeben fordert. Damit werden im letzteren Fall dem AAC-Codierer der skalierbaren Codiervorrichtung die Bits ausgehen, während der AAC-Codierer der skalierbaren Codiervorrichtung im ersteren Fall, um die konstante Ausgangsdatenrate einzuhalten, nicht

umhin kommt, hörbare Störungen in das codierte und wieder decodierte Signal einzuführen.

Die Aufgabe der vorliegenden Erfindung besteht darin, ein Verfahren und eine Vorrichtung zum Erzeugen eines skalierbaren Datenstroms zu schaffen, das dazu geeignet ist, daß eine Bitsparkassenfunktion für eine Skalierungsschicht eingesetzt werden kann.

Diese Aufgabe wird durch ein Verfahren nach Patentanspruch 1 oder durch eine Vorrichtung nach Patentanspruch 9 gelöst.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, ein Verfahren zum Decodieren eines skalierbaren Datenstroms zu schaffen.

Diese Aufgabe wird durch ein Verfahren nach Patentanspruch 10 oder durch eine Vorrichtung nach Patentanspruch 11 gelöst.

Der vorliegenden Erfindung liegt die Erkenntnis zugrunde, daß von dem bekannten, in Fig. 2c dargelegten Konzept weggegangen werden muß, das darin besteht, daß sämtliche Daten eines Ausgangsdatenblocks des zweiten Codierers zwischen zwei aufeinanderfolgenden LATM-Headern angeordnet sind. Statt dessen wird es zugelassen, daß auch Ausgangsdaten des zweiten Codierers, die einen vorausgehenden Zeitabschnitt des Eingangssignals darstellen, nach einem Bestimmungsdatenblock für den aktuellen Zeitabschnitt geschrieben werden, wobei diese Tatsache bzw. wieviel Daten noch in Übertragungsrichtung hinter dem Bestimmungsdatenblock geschrieben werden, durch spezielle ebenfalls zu übertragende Pufferinformationen einem Decodierer signalisiert werden.

Der Decodierer kann dann ohne weiteres, ausgehend von einem Bestimmungsdatenblock und unter Verwendung der Pufferinformationen feststellen, wo die Ausgangsdaten des zweiten Codierers enden, und wo dann die Ausgangsdaten des zweiten

Codierers für den aktuellen Zeitabschnitt beginnen, so daß der Decodierer in der Lage ist, die korrespondierenden Ausgangsdatenblöcke des ersten Codierers mit korrespondierenden Ausgangsdatenblöcken des zweiten Codierers in Verbindung zu bringen, um das Signal in allen Schichten wieder zu decodieren, wobei sich der Ausdruck "korrespondierend" darauf bezieht, daß die entsprechenden Daten des ersten und des zweiten Codierers auf den selben Abschnitt des Eingangssignals im Falle von CoreCoderDelay gleich Null (siehe Fig. 1) oder auf um Core Coder Delay verschobenen aktuelle Abschnitte für den ersten und den zweiten Codierer bezogen sind.

Bei einem erfindungsgemäßen Verfahren zum Erzeugen eines skalierbaren Datenstroms aus einem oder mehreren Blöcken von Ausgangsdaten eines ersten Codierers und aus einem oder mehreren Blöcken von Ausgangsdaten eines zweiten Codierers wird daher ein Bestimmungsdatenblock für einen aktuellen Abschnitt des Eingangssignals geschrieben. Darüber hinaus werden die Ausgangsdaten des zweiten Codierers, die einen vorhergehenden Abschnitt des Eingangssignals darstellen, in Übertragungsrichtung von einem Codierer zu einem Decodierer hinter den Bestimmungsdatenblock geschrieben. Die Ausgangsdaten des zweiten Codierers, die sich auf den aktuellen Abschnitt des Eingangssignals beziehen, also die zu dem Bestimmungsdatenblock eigentlich gehören, können dann geschrieben werden, wenn die Ausgangsdaten des zweiten Codierers für den vorhergehenden Abschnitt vollständig geschrieben sind. Darüber hinaus werden Pufferinformationen in den skalierbaren Datenstrom geschrieben, wobei die Pufferinformationen anzeigen, wie weit sich die Ausgangsdaten des zweiten Codierer für den vorausgehenden Abschnitt hinter dem Bestimmungsdatenblock für den aktuellen Abschnitt erstrecken. Die Ausgangsdaten des ersten Codierers können entweder äquidistant oder nicht in den skalierbaren Datenstrom geschrieben werden, wobei es jedoch, aus Delaygründen, um eine verzögerungsarme Decodierung der ersten Skalierungsschicht alleine, also lediglich der Ausgangsdatenblöcke des ersten Codierers zu ermöglichen, wünschenswert ist, diese

Datenblöcke äquidistant und delayoptimiert zu schreiben.

Üblicherweise wird eine Bitsparkasse u. a. durch die maximale Größe der Bitsparkasse definiert, wobei dieser Wert in Fig. 3 mit "Max Bufferfullness" bezeichnet wird. Dieser Wert ist fest und dem Decodierer bekannt. Darüber hinaus wird im Datenstrom der aktuelle Wert der Belegung der Bitsparkasse, der mit "Bufferfullness" bezeichnet wird, übertragen. Die Differenz aus der Variablen Max Bufferfullness und Bufferfullness liefert dann, wenn die vorliegende Erfindung auf einen MPEG 4-Codierer angewendet wird, die Pufferinformationen, wobei, wie es später dargelegt werden wird, in diesem Fall zu berücksichtigen ist, daß unter Umständen in den AAC-Blöcken eingestreute Celp-Blöcke oder Daten anderer Skalierungsschichten nicht berücksichtigt werden dürfen, um den genauen Wert des Beginns der Ausgangsdaten des zweiten Datenblocks hinter dem LATM-Bestimmungsdatenblock zu finden.

Unabhängig von der Funktionalität der Bitsparkasse ermöglicht es das erfindungsgemäße Format jedoch auch, in einem äquidistanten Raster von Bestimmungsdatenblöcken, Ausgangsdatenblöcke variierender Länge des zweiten Codierers zu übertragen. So kann es sinnvoll sein, das Raster für die Bestimmungsdatenblöcke und das Raster für die Ausgangsdatenblöcke des ersten Codierers äquidistant zu wählen, und insbesondere so zu wählen, daß einem Bestimmungsdatenblock immer ein Ausgangsdatenblock des ersten Codierers folgt. Der Ausgangsdatenblock des zweiten Codierers wird dann in die verbleibenden Lücken geschrieben, wobei durch die Pufferinformationen signalisiert wird, wieviel Daten des zweiten Codierers hinter einem Bestimmungsdatenblock zu dem Zeitabschnitt, auf den der Bestimmungsdatenblock hinweist, gehören, oder noch zu dem vorausgehenden zeitlichen Abschnitt des Eingangssignals zu zählen sind, damit der Decodierer eindeutig und zweifelsfrei eine Zuordnung zwischen Ausgangsdatenblöcken des ersten Codierers und einem Ausgangsdatenblock des zweiten Codierers für einen Zeitabschnitt des Eingangssignals schaffen kann.

Ein Vorteil der vorliegenden Erfindung besteht ferner darin, daß das Signalisieren des Ausgangsdatenblocks hinter dem Bestimmungsdatenblock ohne weiteres mit einem Signalisieren von Ausgangsdatenblöcken des ersten Codierers vor dem Bestimmungsdatenblock für den aktuellen Zeitabschnitt kombiniert werden kann, um eine verzögerungsarme Decodierung lediglich der ersten Skalierungsschicht zu ermöglichen.

Der erfindungsgemäße skalierbare Datenstrom ist besonders für Echtzeitanwendungen von Nutzen, kann jedoch genauso auch für Nicht-Echtzeitanwendungen eingesetzt werden.

Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend bezugnehmend auf die beiliegenden Zeichnungen detailliert erläutert. Es zeigen:

Fig. 1 einen skalierbaren Codierer gemäß MPEG 4;

Fig. 2a eine schematische Darstellung eines Eingangssignals, das in aufeinanderfolgende Zeitabschnitte eingeteilt ist;

Fig. 2b eine schematische Darstellung eines Eingangssignals, das in aufeinanderfolgende Zeitabschnitte eingeteilt ist, wobei das Verhältnis der Blocklänge des ersten Codierers zu der Blocklänge des zweiten Codierers dargestellt ist;

Fig. 2c eine schematische Darstellung eines skalierbaren Datenstroms mit hoher Verzögerung bei der Decodierung der ersten Skalierungsschicht;

Fig. 2d eine schematische Darstellung eines skalierbaren Datenstroms mit niedriger Verzögerung bei der Decodierung der ersten Skalierungsschicht;

Fig. 2e ein Bitstromformat gemäß der vorliegenden Erfin-

dung, in dem hinter dem Bestimmungsdatenblock für einen aktuellen Abschnitt noch Ausgangsdaten des zweiten Codierers aus einem vorhergehenden Zeitabschnitt angeordnet sind;

Fig. 3 eine detaillierte Darstellung des erfindungsgemäßen skalierbaren Datenstromes am Beispiel eines Celp-Codierers als erster Codierer und eines AAC-Codierers als zweiter Codierer mit Bitsparkassenfunktion.

Fig. 4 ein Beispiel für ein Bitstromformat mit fester Framelänge;

Fig. 5 ein Beispiel für ein Bitstromformat mit fester Framelänge und Back-Pointer; und

Fig. 6 ein Beispiel eines Bitstromformats mit variabler Framelänge.

Im nachfolgenden wird auf Fig. 2d im Vergleich zu Fig. 2c eingegangen, um einen Bitstrom mit niedriger Verzögerung für die erste Skalierungsschicht zu erläutern. Genauso wie in Fig. 2c enthält der skalierbare Datenstrom aufeinanderfolgende Bestimmungsdatenblöcke, die als Header 1 und Header 2 bezeichnet sind. Bei MPEG 4 sind die Bestimmungsdatenblöcke LATM-Header. In Übertragungsrichtung von einem Encoder zu einem Decodierer, die in Fig. 2d mit einem Pfeil 202 dargestellt ist, findet sich hinter dem LATM-Header 200 die von links oben nach rechts unten schraffierten Teile des Ausgangsdatenblocks des AAC-Codierers, die in verbleibende Lücken zwischen Ausgangsdatenblöcken des ersten Codierers eingetragen sind.

Ferner finden sich im Unterschied zu Fig. 2c nun jedoch in dem durch den LATM-Header 200 begonnenen Frame nicht mehr nur Ausgangsdatenblöcke des ersten Codierers, die in diesen Frame gehören, wie z.B. die Ausgangsdatenblöcke 13 und 14,

sondern auch die Ausgangsdatenblöcke 21 und 22 des nachfolgenden Abschnitts von Eingangsdaten. Anders ausgedrückt sind bei dem in Fig. 2d gezeigten Beispiel die beiden Ausgangsdatenblöcke des ersten Codierers, die mit 11 und 12 bezeichnet sind, in Übertragungsrichtung (Pfeil 202) vor dem LATM-Header 200 im Bitstrom vorhanden. Bei dem in Fig. 2d gezeigten Beispiel deuten die Offset-Informationen 204 auf einen Offset der Ausgangsdatenblöcke des ersten Codierers von zwei Ausgangsdatenblöcken hin. Wenn Fig. 2d mit Fig. 2c verglichen wird, so ist zu erkennen, daß der Decodierer bereits die unterste Skalierungsschicht genau um eine diesem Offset entsprechende Zeit früher decodieren kann als im Fall von Fig. 2c, wenn der Decodierer lediglich an der ersten Skalierungsschicht interessiert ist. Die Offset-Informationen, die z. B. in Form eines "Core Frame Offset" signalisiert werden können, dienen dazu, die Position des ersten Ausgangsdatenblocks 11 im Bitstrom zu bestimmen.

Für den Fall von Core Frame Offset = Null ergibt sich der in Fig. 2c bezeichnete Bitstrom. Ist jedoch Core Frame Offset > Null, so wird der entsprechende Ausgangsdatenblock des ersten Codierers 11 um die Anzahl Core Frame Offset an Ausgangsdatenblöcken des ersten Codierers früher übertragen. Anders ausgedrückt ergibt sich das Delay zwischen dem ersten Ausgangsdatenblock des ersten Codierers nach dem LATM-Header und dem ersten AAC-Frame aus Core Coder Delay (Fig. 1) + Core Frame Offset x Core-Blocklänge (Blocklänge des Coders 1 in Fig. 2b). Wie aus dem Vergleich von Fig. 2c und 2d deutlich wird, werden für Core Frame Offset = Null (Fig. 2c) nach dem LATM-Header 200 die Ausgangsdatenblöcke 11 und 12 des ersten Codierers übertragen. Durch die Übertragung von Core Frame Offset = 2 können die Ausgangsdatenblöcke 13 und 14 nach dem LATM-Header 200 folgen, wodurch die Verzögerung bei reiner Celp-Decodierung, also Decodierung der ersten Skalierungsschicht, um zwei Celp-Blocklängen verringert wird. Optimal wäre im Beispiel ein Offset von drei Blöcken. Ein Offset von einem oder zwei Blöcken bringt jedoch ebenfalls bereits einen Delayvorteil.

Durch diesen Bitstromaufbau ist es möglich, daß der Celp-Codierer den erzeugten Celp-Block unmittelbar nach dem Codieren übertragen kann. In diesem Fall wird dem Celp-Codierer kein zusätzliches Delay durch den Bitstrommultiplexer (20) zugefügt. Somit wird für diesen Fall zu dem Celp-Delay kein zusätzliches Delay durch die skalierbare Kombination hinzugefügt, so daß das Delay minimal wird.

Es wird darauf hingewiesen, daß der in Fig. 2d gezeigte Fall lediglich beispielhaft ist. So sind verschiedene Verhältnisse der Blocklänge des ersten Codierers zu der Blocklänge des zweiten Codierers möglich, die z. B. von 1:2 bis zu 1:12 variieren können oder aber auch andere Verhältnisse einnehmen können, wobei Verhältnisse größer oder kleiner Eins auftreten können.

Dies heißt im Extremfall (1:12 für MPEG 4 CELP/AAC), daß für denselben Zeitabschnitt des Eingangssignals, für den der AAC-Codierer einen Ausgangsdatenblock erzeugt, der Celp-Codierer zwölf Ausgangsdatenblöcke erzeugt. Der Verzögerungs-Vorteil durch den Datenstrom, der in Fig. 2d gezeigt ist, gegenüber dem Datenstrom, der in Fig. 2c gezeigt ist, kann in diesem Fall durchaus in Größenordnungen von einer viertel bis zu einer halben Sekunde kommen. Dieser Vorteil wird sich um so mehr erhöhen, je größer das Verhältnis zwischen Blocklänge des zweiten Codierers und Blocklänge des ersten Codierers wird, wobei im Falle des AAC-Codierers als zweiter Codierer eine möglichst große Blocklänge aufgrund des dann günstigeren Verhältnisses zwischen Nutzinformationen zu Seiteninformationen angestrebt wird, wenn es das zu codierende Signal zuläßt.

Im nachfolgenden wird auf Fig. 2e Bezug genommen. Im Unterschied zu Fig. 2d, in der bereits die Offset-Funktion, also die Verschiebung der Ausgangsdatenblöcke des ersten Codierers bezüglich eines Bestimmungsdatenblocks dargestellt sind, wird in Fig. 2e die erfindungsgemäße Verschiebung der

Ausgangsdatenblöcke des zweiten Codierers bezüglich des durch die Bestimmungsdatenblöcke gegebenen Rasters dargestellt. Die Anordnung der Ausgangsdatenblöcke des ersten Codierers, die mit 11, 12, 13, 14, 21, 22, 23, 24, 31 in Fig. 2e bezeichnet sind, ist gegenüber Fig. 2d unverändert. Während in Fig. 2d keine Bitsparkassenfunktion möglich ist, bzw., wenn die Bestimmungsdatenblöcke in einem festen Raster sein sollen, keine Ausgangsdatenblöcke veränderlicher Länge für den zweiten Codierer eingesetzt werden können, ist dies bei Fig. 2e nunmehr gemäß der vorliegenden Erfindung möglich.

Hierzu werden die Daten des Ausgangsdatenblocks des zweiten Codierers des vorausgehenden Abschnitts, der mit "0" in den Figuren 2a bis 2e bezeichnet ist, in Übertragungsrichtung von einem Codierer zu einem Decodierer hinter den LATM-Header 200 geschrieben, bis der skalierbare Codierer sämtliche Daten des vorausgehenden Abschnitts in den Bitstrom geschrieben hat. Erst dann wird an einer Übergangsgrenze 220 damit begonnen, die Ausgangsdaten des zweiten Codierers für den aktuellen Abschnitt des Eingangssignals in den Bitstrom zu schreiben. So kann die Übergangsgrenze 220 mit einer Grenze eines Celp-Datenblocks zusammenfallen oder auch nicht. Je nach Signalisierung kann entweder der Abstand vom Ende des Bestimmungsdatenblocks bis zur Übergangsgrenze 220 oder der Abstand vom Anfang des Bestimmungsdatenblocks bis zur Übergangsgrenze 220 oder aber der Abstand von der hinteren Grenze des Celp-Blocks 13 bis zur Übergangsgrenze 220 mit oder ohne Länge der Celp-Blöcke 13, 14 und/oder der Länge des Bestimmungsdatenblocks als Pufferinformationen signalisiert werden. Die letztere Variante wird bezugnehmend auf Fig. 3 noch näher dargestellt.

Erfindungsgemäß wird es im Fall der Anwendung auf einen skalierbaren Codierer bevorzugt, keine eigenen Seiteninformationen zur Signalisierung der Pufferinformationen vorzusehen, sondern hierzu den bereits ohnehin im Bitstrom übertragenen Wert Bufferfullness zu verwenden, wobei die Länge des

mit "Pufferinformationen" in Fig. 2e bezeichneten Zeigers, der in Fig. 3 mit dem Bezugszeichen 314 gekennzeichnet ist, genau gleich der Differenz zwischen Max Bufferfullness und Bufferfullness ist, wenn die Länge der Bestimmungsdatenblöcke und die Länge eventuell vorhandener Celp-Blöcke sowie evtl. vorhandener weiterer Skalierungsschichten unberücksichtigt bleiben, wie es bezugnehmend auf Fig. 3 durch den unterbrochen gezeichneten Pfeil dargestellt ist.

Im nachfolgenden wird auf Fig. 3 eingegangen, welche zu Fig. 2 ähnlich ist, jedoch die besondere Implementierung am Beispiel von MPEG 4 darstellt. In der ersten Zeile ist wieder ein aktueller Zeitabschnitt schraffiert gezeigt. In der zweiten Zeile ist die Fensterung, die beim AAC-Codierer verwendet wird, schematisch dargestellt. Wie es bekannt ist, wird ein Overlap-And-Add von 50 % verwendet, so daß ein Fenster üblicherweise die doppelte Länge von zeitlichen Abtastwerten hat wie der aktuelle Zeitabschnitt, der in der obersten Zeile von Fig. 3 schraffiert dargestellt ist. In Fig. 3 ist ferner die Verzögerung tdip eingezeichnet, die dem Block 26 von Fig. 1 entspricht und die im gewählten Beispiel eine Größe von 5/8 der Blocklänge hat. Typischerweise wird eine Blocklänge des aktuellen Zeitabschnitts von 960 Abtastwerten verwendet, so daß die Verzögerung tdip von 5/8 der Blocklänge 600 Abtastwerte beträgt. Beispielsweise liefert der AAC-Codierer einen Bitstrom von 24 kBit/s, während der darunter schematisch dargestellte Celp-Codierer einen Bitstrom mit einer Rate von 8 kBit/s liefert. Dies resultiert in einer Gesamtbitrate von 32 kBit/s.

Wie es aus Fig. 3 ersichtlich ist, entsprechen die Ausgangsdatenblöcke Null und Eins des Celp-Codierers dem aktuellen Zeitabschnitt des ersten Codierers. Der Ausgangsdatenblock mit der Nummer 2 des Celp-Codierers entspricht bereits dem nächsten Zeitabschnitt. Dasselbe trifft für den Celp-Block mit der Nummer 3 zu. In Fig. 3 ist ferner die Verzögerung der Downsampling-Stufe 28 und des Celp-Codierers 12 durch einen Pfeil eingezeichnet, der mit dem Bezugszeichen 302

dargestellt ist. Daraus ergibt sich als die Verzögerung, die durch die Stufe 34 eingestellt werden muß, damit an der Subtrahierstelle 40 von Fig. 1 gleiche Verhältnisse vorliegen, die Verzögerung, die durch Core Coder Delay bezeichnet ist und mit einem Pfeil 304 in Fig. 3 veranschaulicht ist. Diese Verzögerung kann alternativ auch durch Block 26 erzeugt werden. So gilt beispielsweise:

Core Coder Delay =

= tdip - Celp Encoder Delay - Downsampling Delay =

= 600 - 120 - 117 = 363 Abtastwerte.

Für den Fall ohne Bitsparkassenfunktion bzw. für den Fall, daß die Bitsparkasse (Bit Mux Outputbuffer) voll ist, was durch die Variable Bufferfullness = Max angezeigt ist, ergibt sich der in Fig. 2d gezeichnete Fall. Im Unterschied zu Fig. 2d, bei der vier Ausgangsdatenblöcke des ersten Codierers entsprechend einem Ausgangsdatenblock des zweiten Codierers erzeugt werden, wird bei Fig. 3 für einen Ausgangsdatenblock des zweiten Codierers, welcher in den beiden letzten Zeilen von Fig. 3 schwarz gezeichnet ist, zwei Ausgangsdatenblöcke des Celp-Codierers, die mit "0" und "1" bezeichnet sind, erzeugt. Erfindungsgemäß wird nun jedoch hinter einen ersten LATM-Header 306 nicht mehr der Ausgangsdatenblock des Celp-Codierers mit der Nummer "0" geschrieben, sondern der Ausgangsdatenblock des Celp-Codierers mit der Nummer "Eins", zumal der Ausgangsdatenblock mit der Nummer "Null" bereits zum Decodierer übertragen worden ist. In dem für die Celp-Datenblöcke vorgesehenen äquidistanten Rasterabstand folgt dann dem Celp-Block 1 der Celp-Block 2 für den nächsten Zeitabschnitt, wobei dann zur Fertigstellung eines Frames der Rest der Daten des Ausgangsdatenblocks des AAC-Codierers in den Datenstrom geschrieben wird, bis wieder ein nächster LATM-Header 308 für den nächsten Zeitabschnitt folgt.

Die vorliegende Erfindung kann, wie es in der letzten Zeile von Fig. 3 dargestellt ist, einfach mit der Bitsparkassenfunktion kombiniert werden. Für den Fall, daß die Variable "Bufferfullness", die die Füllung der Bitsparkasse anzeigt, kleiner als der maximale Wert ist, bedeutet dies, daß der AAC-Frame für den unmittelbar vorhergehenden Zeitabschnitt mehr Bits als eigentlich zulässig benötigt hat. Dies bedeutet, daß hinter dem LATM-Header 306 die Celp-Frames wie vorher geschrieben werden, daß jedoch zunächst der Ausgangsdatenblock oder die Ausgangsdatenblöcke des AAC-Codierers aus vorhergehenden Zeitabschnitten in den Bitstrom geschrieben werden müssen, bevor mit dem Schreiben des Ausgangsdatenblocks des AAC-Codierers für den aktuellen Zeitabschnitt begonnen werden kann. Aus dem Vergleich der beiden letzten Zeilen von Fig. 3, die mit "1" und "2" gekennzeichnet sind, ist zu sehen, daß die Bitsparkassenfunktion unmittelbar auch zu einer Verzögerung im Codierer für den AAC-Frame führt. So sind die Daten für den AAC-Frame des aktuellen Zeitabschnitts, die in Fig. 3 mit 310 bezeichnet sind, zwar genau zum gleichen Zeitpunkt wie im Fall "1" vorhanden, können jedoch erst dann in den Bitstrom geschrieben werden, nachdem die AAC-Daten 312 für den unmittelbar vorhergehenden Zeitabschnitt in den Bitstrom geschrieben worden sind. In Abhängigkeit von dem Bitsparkassenstand des AAC-Codierers verschiebt sich somit die Anfangsposition des AAC-Frames.

Der Bitsparkassenstand wird gemäß MPEG 4 im Element StreamMuxConfig durch die Variable "Bufferfullness" übertragen. Die Variable Bufferfullness berechnet sich aus der Variablen Bitreservoir geteilt durch das 32fache der gerade vorhandenen Kanalanzahl der Audiokanäle.

Es sei darauf hingewiesen, daß es sich bei dem Zeiger, der in Fig. 3 mit dem Bezugszeichen 314 gekennzeichnet ist, und dessen Länge = max Bufferfullness - Bufferfullness ist, um einen Vorwärtszeiger handelt, der gewissermaßen in die Zukunft zeigt, während es sich bei dem in Fig. 5 gezeichneten Zeiger um einen Rückwärtszeiger handelt, der ge-

wissermaßen in die Vergangenheit zeigt. Dies liegt daran, daß gemäß vorliegendem Ausführungsbeispiel der LATM-Header immer dann in den Bitstrom geschrieben wird, nachdem der aktuelle Zeitabschnitt durch den AAC-Codierer verarbeitet worden ist, obgleich ggf. noch AAC-Daten aus vorherigen Zeitabschnitten in den Bitstrom zu schreiben sind.

Es sei ferner darauf hingewiesen, daß der Zeiger 314 absichtlich unterhalb des Celp-Blocks 2 unterbrochen gezeichnet ist, da er die Länge des Celp-Blocks 2 genauso wie die Länge des Celp-Blocks 1 nicht berücksichtigt, da diese Daten selbstverständlich nichts mit der Bitsparkasse des AAC-Codierers zu tun haben. Ferner werden keinerlei Header-Daten und Bits von gegebenenfalls vorhandenen weiteren Layern berücksichtigt.

Im Decodierer wird zunächst aus dem Bitstrom eine Extraktion der Celp-Frames vorgenommen, was ohne weiteres möglich ist, da dieselben beispielsweise äquidistant angeordnet sind und eine feste Länge haben.

Im LATM-Header können jedoch ohnehin Länge und Abstand aller CELP-Blöcke signalisiert werden, so daß in jedem Fall eine unmittelbare Decodierung möglich ist.

Damit werden die gewissermaßen durch den Celp-Block 2 getrennten Teile der Ausgangsdaten des AAC-Codierers des unmittelbar vorhergehenden Zeitabschnitts wieder aneinandergefügt, und der LATM-Header 306 rückt gewissermaßen an den Beginn des Zeigers 314, so daß der Decodierer unter Kenntnis der Länge des Zeigers 314 weiß, wann nunmehr die Daten des unmittelbar vorhergehenden Zeitabschnitts zu Ende sind, um dann, wenn diese Daten vollständig eingelesen sind, den unmittelbar vorhergehenden Zeitabschnitt zusammen mit den für denselben vorhandenen Celp-Datenblöcken mit voller Audioqualität decodieren zu können.

Im Gegensatz zu dem in Fig. 2c gezeigten Fall, bei dem einem

LATM-Header sowohl die Ausgangsdatenblöcke des ersten Codierers als auch der Ausgangsdatenblock des zweiten Codierers folgt, kann nun einerseits durch die Variable Core Frame Offset eine Verschiebung von Ausgangsdatenblöcken des ersten Codierers nach vorne im Bitstrom erfolgen, während durch den Pfeil 314 (max Bufferfullness - Bufferfullness) eine Verschiebung des Ausgangsdatenblocks des zweiten Codierers nach hinten im skalierbaren Datenstrom erreicht werden kann, so daß die Bitsparkassenfunktion auch im skalierbaren Datenstrom auf einfache und sichere Art und Weise implementiert werden kann, während das Grundraster des Bitstroms durch die aufeinanderfolgende LATM-Bestimmungsdatenblöcke beibehalten wird, die immer dann geschrieben werden, wenn der AAC-Codierer einen Zeitabschnitt codiert hat, und die daher als Bezugspunkt dienen können, auch wenn, wie es in Fig. 3 in der letzten Zeile gezeigt ist, ein Großteil der Daten in dem durch einen LATM-Header bezeichneten Frame einerseits vom nächsten Zeitabschnitt stammen (hinsichtlich der Celp-Frames) oder aber von unmittelbar vorhergehenden Zeitabschnitten stammen (hinsichtlich des AAC-Frames), wobei die jeweiligen Verschiebungen jedoch durch die zwei im Bitstrom zusätzlich zu übertragenden Variablen einem Decodierer mitgeteilt werden.

Patentansprüche

1. Verfahren zum Erzeugen eines skalierbaren Datenstroms aus einem oder mehreren Blöcken von Ausgangsdaten eines ersten Codierers (12) und aus einem oder mehreren Blöcken von Ausgangsdaten eines zweiten Codierers (14), wobei der eine oder die mehreren Blöcke von Ausgangsdaten des ersten Codierers (12) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den ersten Codierer darstellen, die einen aktuellen Abschnitt des Eingangssignals für den ersten Codierer bilden, und wobei der eine Block oder die mehreren Blöcke von Ausgangsdaten des zweiten Codierers (14) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den zweiten Codierer darstellen, wobei die Anzahl von Abtastwerten für den zweiten Codierer einen aktuellen Abschnitt des Eingangssignals für den zweiten Codierer bildet, wobei die Anzahl von Abtastwerten für den ersten Codierer und die Anzahl von Abtastwerten für den zweiten Codierer gleich sind, und wobei die aktuellen Abschnitte für den ersten und den zweiten Codierer identisch sind oder um eine Zeitdauer (34) zueinander verschoben sind, mit folgenden Schritten:

Schreiben eines Bestimmungsdatenblocks (306) für den aktuellen Abschnitt des Eingangssignals für den ersten oder den zweiten Codierer;

Schreiben von Ausgangsdaten (312) des zweiten Codierers, die einen vorhergehenden Abschnitt des Eingangssignals für den zweiten Codierer darstellen, in Übertragungsrichtung von einem Codierer zu einem Decodierer hinter den Bestimmungsdatenblock (306);

Schreiben von Ausgangsdaten (310) des zweiten Codierers, die den aktuellen Abschnitt des Eingangssignals für den zweiten Codierer darstellen, wenn die Ausgangsdaten des zweiten Codierers für den vorhergehenden Abschnitt des Eingangssignals geschrieben sind;

Schreiben von Pufferinformationen (314) in den skalierbaren Datenstrom, wobei die Pufferinformationen anzeigen, wie weit sich die Ausgangsdaten des zweiten Codierers für den vorausgehenden Abschnitt für den zweiten Codierer hinter den Bestimmungsdatenblock erstrecken; und

Schreiben des einen oder der mehreren Blöcke der Ausgangsdaten des ersten Codierers (12) in den skalierbaren Datenstrom.

2. Verfahren nach Anspruch 1,

bei dem die Längen der Blöcke von Ausgangsdaten des zweiten Codierers für gleichlange Abschnitte des Eingangssignals unterschiedlich sind, wobei die Längen der Blöcke von Ausgangsdaten von Signaleigenschaften des Eingangssignals abhängen,

bei dem der eine oder die mehreren Blöcke der Ausgangsdaten des ersten Codierers für gleichlange Abschnitte des Eingangssignals gleich lang sind, und

bei dem die Übertragungsrate des Bitstroms konstant ist.

3. Verfahren nach Anspruch 1 oder 2,

bei dem der zweite Codierer (14) eine Bitsparkassenfunktion aufweist, wobei die maximale Größe der Bitsparkasse durch Maximal-Puffergrößeninformationen gegeben ist, und wobei der aktuelle Stand der Bitsparkasse durch Aktuell-Pufferinformationen gegeben ist,

bei dem die Pufferinformationen (314) die Aktuell-Pufferinformationen sind, und

bei dem die Größe, wie weit sich die Ausgangsdaten des

zweiten Codierers für den vorhergehenden Zeitabschnitt hinter den Bestimmungsdatenblock (306) erstrecken, aus der Differenz zwischen den Maximal-Puffergrößeninformationen und den Aktuell-Pufferinformationen ableitbar ist.

4. Verfahren nach einem der vorhergehenden Ansprüche,

bei dem das Schreiben von Ausgangsdaten des ersten Codierers so durchgeführt wird, daß ein Block von Ausgangsdaten des ersten Codierers unmittelbar hinter einem Bestimmungsdatenblock (306) angeordnet ist, und

bei dem die Länge dieses Bestimmungsdatenblocks (306) sowie die Länge von vorhandenen Ausgangsdatenblöcken des ersten Codierers sowie gegebenenfalls vorhandene Daten weiterer Skalierungsschichten bei der Bestimmung der Größe, wie weit sich die Ausgangsdaten des zweiten Codierers hinter den Bestimmungsdatenblock erstrecken, unter Verwendung der Aktuell-Pufferinformationen und der Maximal-Puffergrößeninformationen ignoriert werden.

5. Verfahren nach einem der vorhergehenden Ansprüche,

bei dem die Einrichtung (20) zum Schreiben des einen oder der mehreren Blöcke von Ausgangsdaten des ersten Codierers ausgebildet ist, um die Blöcke von Ausgangsdaten des ersten Codierers äquidistant in den skalierbaren Datenstrom zu schreiben.

6. Verfahren nach einem der vorhergehenden Ansprüche,

bei dem der erste Codierer (12) ein Celp-Codierer ist,

bei dem der zweite Codierer (14) ein AAC-Codierer ist, und

bei dem der Bestimmungsdatenblock ein LATM-Header gemäß MPEG 4 ist.

7. Verfahren nach einem der vorhergehenden Ansprüche, bei dem der zumindest eine Block von Ausgangsdaten des zweiten Codierers (14) und der zumindest einen Block von Ausgangsdaten des ersten Codierers (12) Nutzdaten in einem Superframe sind, der neben den Nutzdaten genau einen Bestimmungsdatenblock aufweist.

8. Verfahren nach einem der vorhergehenden Ansprüche,

   bei dem im Schritt des Schreibens der Blöcke von Ausgangsdaten des ersten Codierers zumindest ein Block von Ausgangsdaten des ersten Codierers für den aktuellen Abschnitt des Eingangssignals für den ersten Codierer in Übertragungsrichtung vor dem Bestimmungsdatenblock für den aktuellen Zeitabschnitt geschrieben wird.

9. Vorrichtung zum Erzeugen eines skalierbaren Datenstroms aus einem oder mehreren Blöcken von Ausgangsdaten eines ersten Codierers (12) und aus einem oder mehreren Blöcken von Ausgangsdaten eines zweiten Codierers (14), wobei der eine oder die mehreren Blöcke von Ausgangsdaten des ersten Codierers (12) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den ersten Codierer darstellen, die einen aktuellen Abschnitt des Eingangssignals für den ersten Codierer bilden, und wobei der eine Block oder die mehreren Blöcke von Ausgangsdaten des zweiten Codierers (14) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den zweiten Codierer darstellen, wobei die Anzahl von Abtastwerten für den zweiten Codierer einen aktuellen Abschnitt des Eingangssignals für den zweiten Codierer bildet, wobei die Anzahl von Abtastwerten für den ersten Codierer und die Anzahl von Abtastwerten für den zweiten Codierer gleich sind, und wobei die aktuellen Abschnitte für den ersten und den zweiten Codierer identisch sind oder um eine Zeitdauer (34) zueinander verschoben sind, mit folgenden Merkmalen:

einer Einrichtung zum Schreiben eines Bestimmungsdatenblocks (306) für den aktuellen Abschnitt des Eingangssignals für den ersten oder den zweiten Codierer;

einer Einrichtung zum Schreiben von Ausgangsdaten (312) des zweiten Codierers, die einen vorhergehenden Abschnitt des Eingangssignals für den zweiten Codierer darstellen, in Übertragungsrichtung von einem Codierer zu einem Decodierer hinter den Bestimmungsdatenblock (306);

einer Einrichtung zum Schreiben von Ausgangsdaten (310) des zweiten Codierers, die den aktuellen Abschnitt des Eingangssignals für den zweiten Codierer darstellen, wenn die Ausgangsdaten des zweiten Codierers für den vorhergehenden Abschnitt des Eingangssignals geschrieben sind;

einer Einrichtung zum Schreiben von Pufferinformationen (314) in den skalierbaren Datenstrom, wobei die Pufferinformationen anzeigen, wie weit sich die Ausgangsdaten des zweiten Codierers für den vorausgehenden Abschnitt für den zweiten Codierer hinter den Bestimmungsdatenblock erstrecken; und

einer Einrichtung zum Schreiben des einen oder der mehreren Blöcke der Ausgangsdaten des ersten Codierers (12) in den skalierbaren Datenstrom.

10. Verfahren zum Decodieren eines skalierbaren Datenstroms aus einem oder mehreren Blöcken von Ausgangsdaten eines ersten Codierers (12) und aus einem oder mehreren Blöcken von Ausgangsdaten eines zweiten Codierers (14), wobei der eine oder die mehreren Blöcke von Ausgangsdaten des ersten Codierers (12) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den ersten Codierer darstellen, die einen aktuellen Abschnitt des Eingangssignals für den ersten Codierer bilden, und wobei der eine Block oder die mehreren Blöcke von Ausgangsdaten des zweiten Codierers (14) zusammen eine Anzahl von Ab-

tastwerten des Eingangssignals für den zweiten Codierer darstellen, wobei die Anzahl von Abtastwerten für den zweiten Codierer einen aktuellen Abschnitt des Eingangssignals für den zweiten Codierer bildet, wobei die Anzahl von Abtastwerten für den ersten Codierer und die Anzahl von Abtastwerten für den zweiten Codierer gleich sind, und wobei die aktuellen Abschnitte für den ersten und den zweiten Codierer identisch sind oder um eine Zeitdauer (34) zueinander verschoben sind, wobei der skalierbare Datenstrom einen Bestimmungsdatenblock für den aktuellen Abschnitt für den ersten oder zweiten Codierer, Ausgangsdaten des zweiten Codierers für einen vorhergehenden Abschnitt des Eingangssignals in Übertragungsrichtung hinter dem Bestimmungsdatenblock und Pufferinformationen aufweist, die anzeigen, wie weit sich die Ausgangsdaten des zweiten Codierers für den vorausgehenden Abschnitt hinter den Bestimmungsdatenblock erstrecken, mit folgenden Schritten:

Lesen des Bestimmungsdatenblocks (306) für den aktuellen Abschnitt des Eingangssignals für den ersten oder zweiten Codierer;

Lesen der Ausgangsdaten des ersten Codierers für den aktuellen Abschnitt des ersten Codierers (12);

Lesen der Pufferinformationen (314);

Lesen der Ausgangsdaten (310) des zweiten Codierers für den aktuellen Abschnitt ausgehend von einer durch die Pufferinformationen (314) angezeigten Stelle im skalierbaren Datenstrom; und

Decodieren der Ausgangsdaten (310) des zweiten Codierers und der Ausgangsdaten des ersten Codierers, um ein decodiertes Signal zu erhalten.

11. Vorrichtung zum Decodieren eines skalierbaren Daten-

stroms aus einem oder mehreren Blöcken von Ausgangsdaten eines ersten Codierers (12) und aus einem oder mehreren Blöcken von Ausgangsdaten eines zweiten Codierers (14), wobei der eine oder die mehreren Blöcke von Ausgangsdaten des ersten Codierers (12) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den ersten Codierer darstellen, die einen aktuellen Abschnitt des Eingangssignals für den ersten Codierer bilden, und wobei der eine Block oder die mehreren Blöcke von Ausgangsdaten des zweiten Codierers (14) zusammen eine Anzahl von Abtastwerten des Eingangssignals für den zweiten Codierer darstellen, wobei die Anzahl von Abtastwerten für den zweiten Codierer einen aktuellen Abschnitt des Eingangssignals für den zweiten Codierer bildet, wobei die Anzahl von Abtastwerten für den ersten Codierer und die Anzahl von Abtastwerten für den zweiten Codierer gleich sind, und wobei die aktuellen Abschnitte für den ersten und den zweiten Codierer identisch sind oder um eine Zeitdauer (34) zueinander verschoben sind, wobei der skalierbare Datenstrom einen Bestimmungsdatenblock für den aktuellen Abschnitt für den ersten oder zweiten Codierer, Ausgangsdaten des zweiten Codierers für einen vorhergehenden Abschnitt des Eingangssignals in Übertragungsrichtung hinter dem Bestimmungsdatenblock und Pufferinformationen aufweist, die anzeigen, wie weit sich die Ausgangsdaten des zweiten Codierers für den vorausgehenden Abschnitt hinter den Bestimmungsdatenblock erstrecken, mit folgenden Merkmalen:

einem Bitstromdemultiplexer, der ausgebildet ist, um folgende Schritte durchführen zu können:

Lesen des Bestimmungsdatenblocks (306) für den aktuellen Abschnitt des Eingangssignals für den ersten oder zweiten Codierer;

Lesen der Ausgangsdaten des ersten Codierers für den aktuellen Abschnitt des ersten Codierers (12);

Lesen der Pufferinformationen (314);

Lesen der Ausgangsdaten (310) des zweiten Codierers für den aktuellen Abschnitt ausgehend von einer durch die Pufferinformationen (314) angezeigten Stelle im skalierbaren Datenstrom; und

einer Einrichtung zum Decodieren der Ausgangsdaten (310) des zweiten Codierers und der Ausgangsdaten des ersten Codierers, um ein decodiertes Signal zu erhalten.

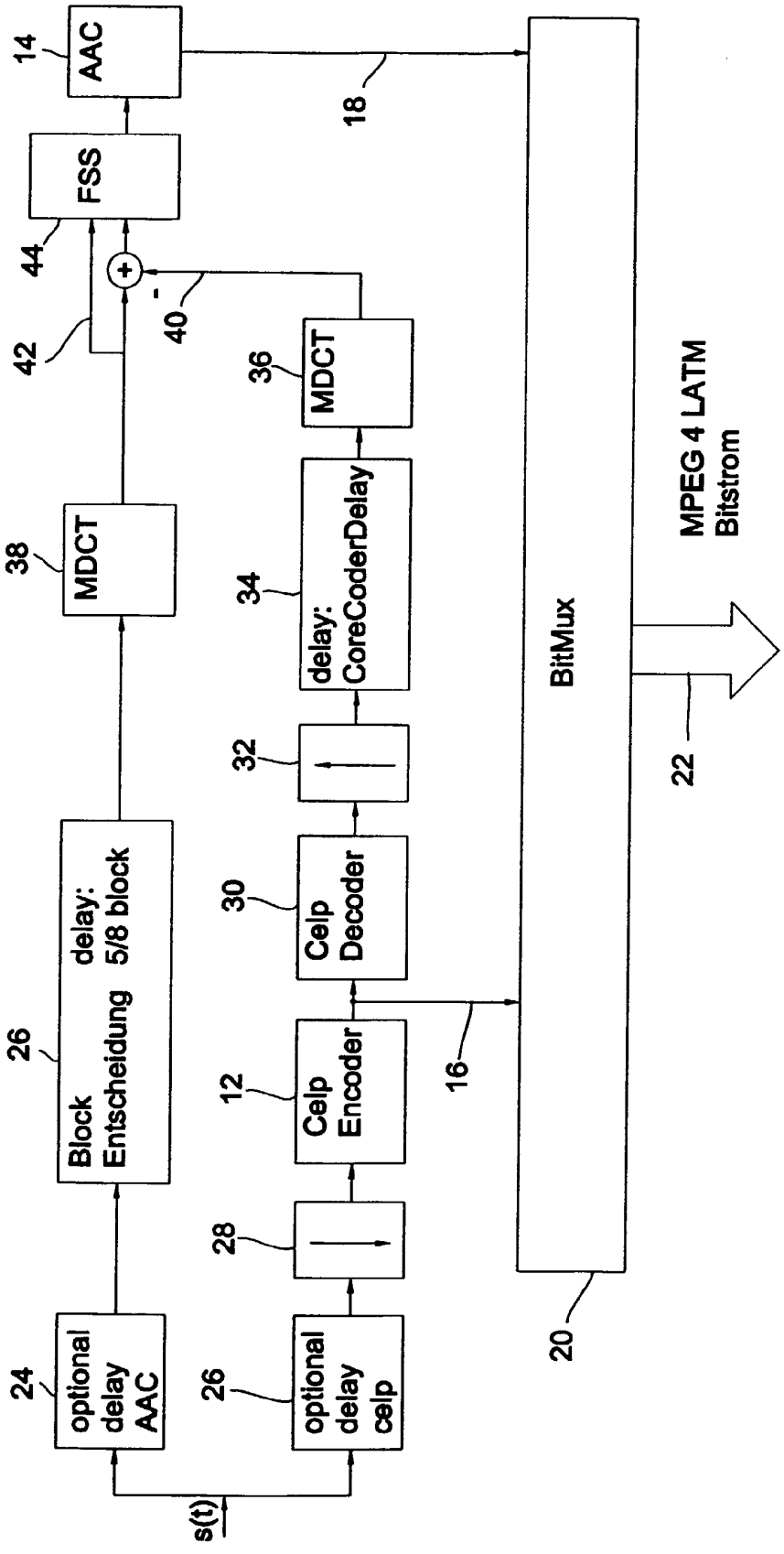

14
AAC
18
FSS
44
42
40
36
MDCT
38
MDCT
34
delay:
CoreCoderDelay
32
30
Celp
Decoder
26
Block
Entscheidung
delay:
5/8 block
12
Celp
Encoder
16
28
24
optional
delay
AAC
26
optional
delay
celp
s(t)
BitMux
20
22
MPEG 4 LATM
Bitstrom

Fig. 1

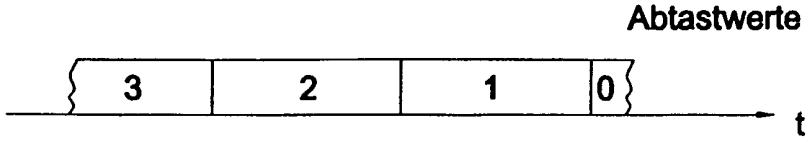

Abtastwerte
3
2
1
0
t

Fig. 2a

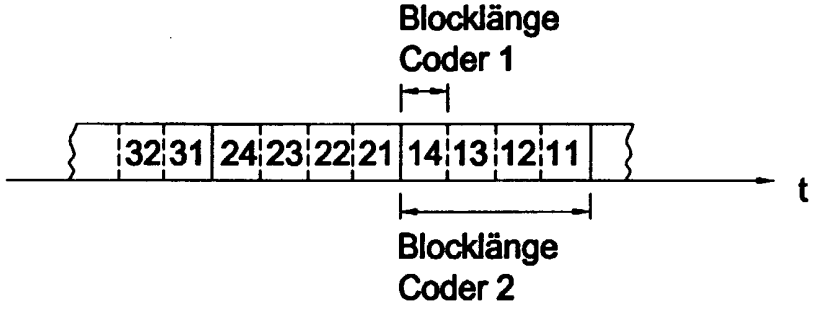

Blocklänge
Coder 1
32
31
24
23
22
21
14
13
12
11
t
Blocklänge
Coder 2

Fig. 2b

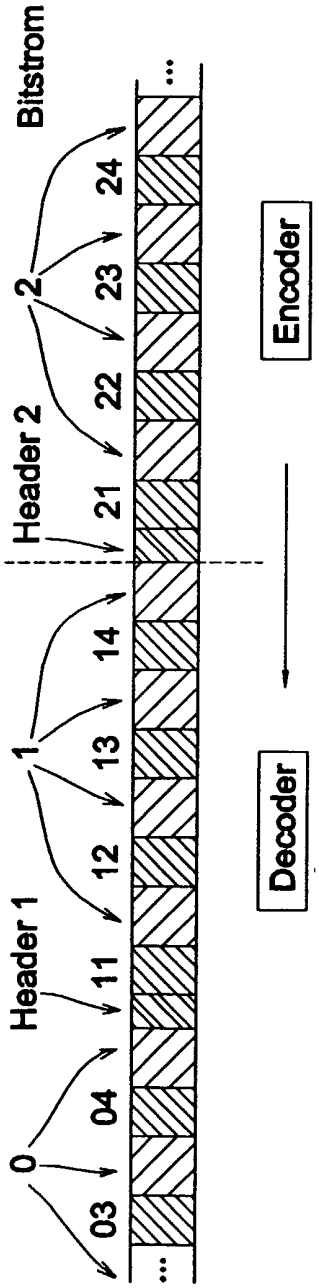


Fig. 2c (Stand der Technik)

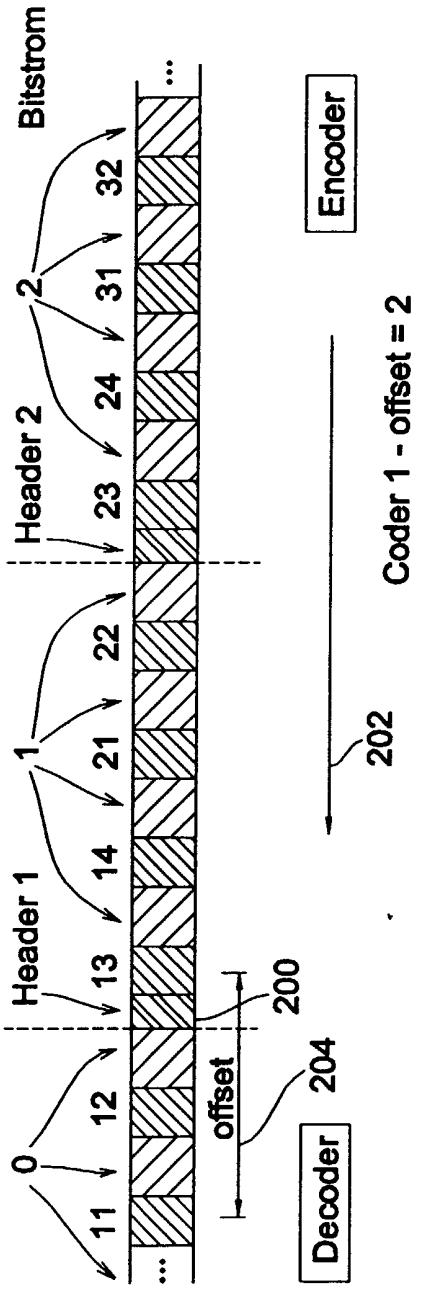


Fig. 2d

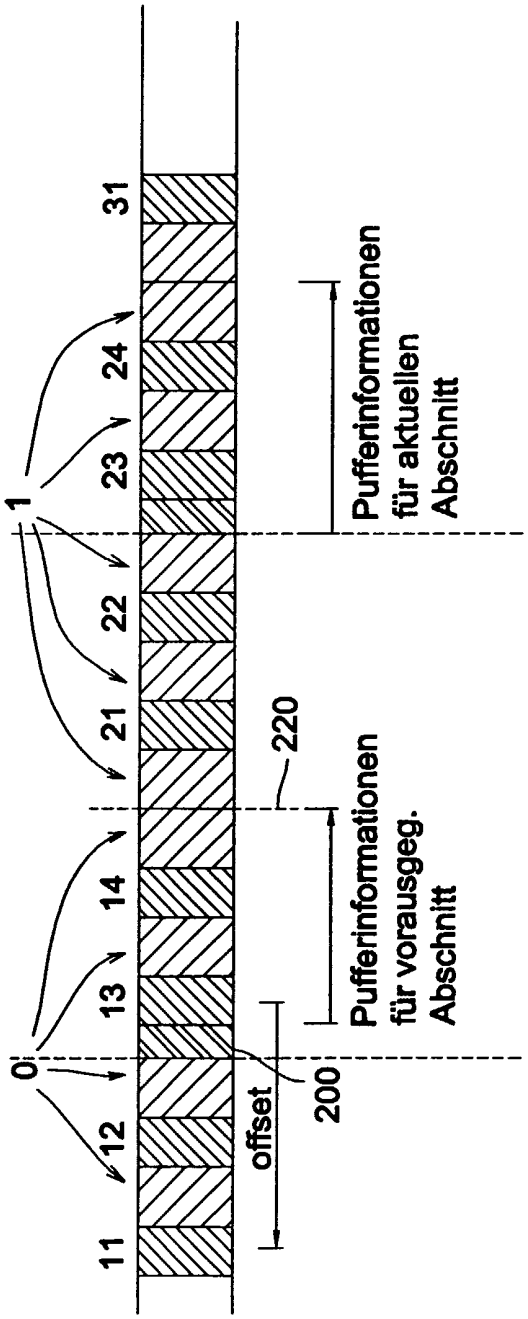

0
1
11
12
13
14
21
22
23
24
31
offset
200
220
Pufferinformationen für vorausgeg. Abschnitt
Pufferinformationen für aktuellen Abschnitt

Fig. 2e

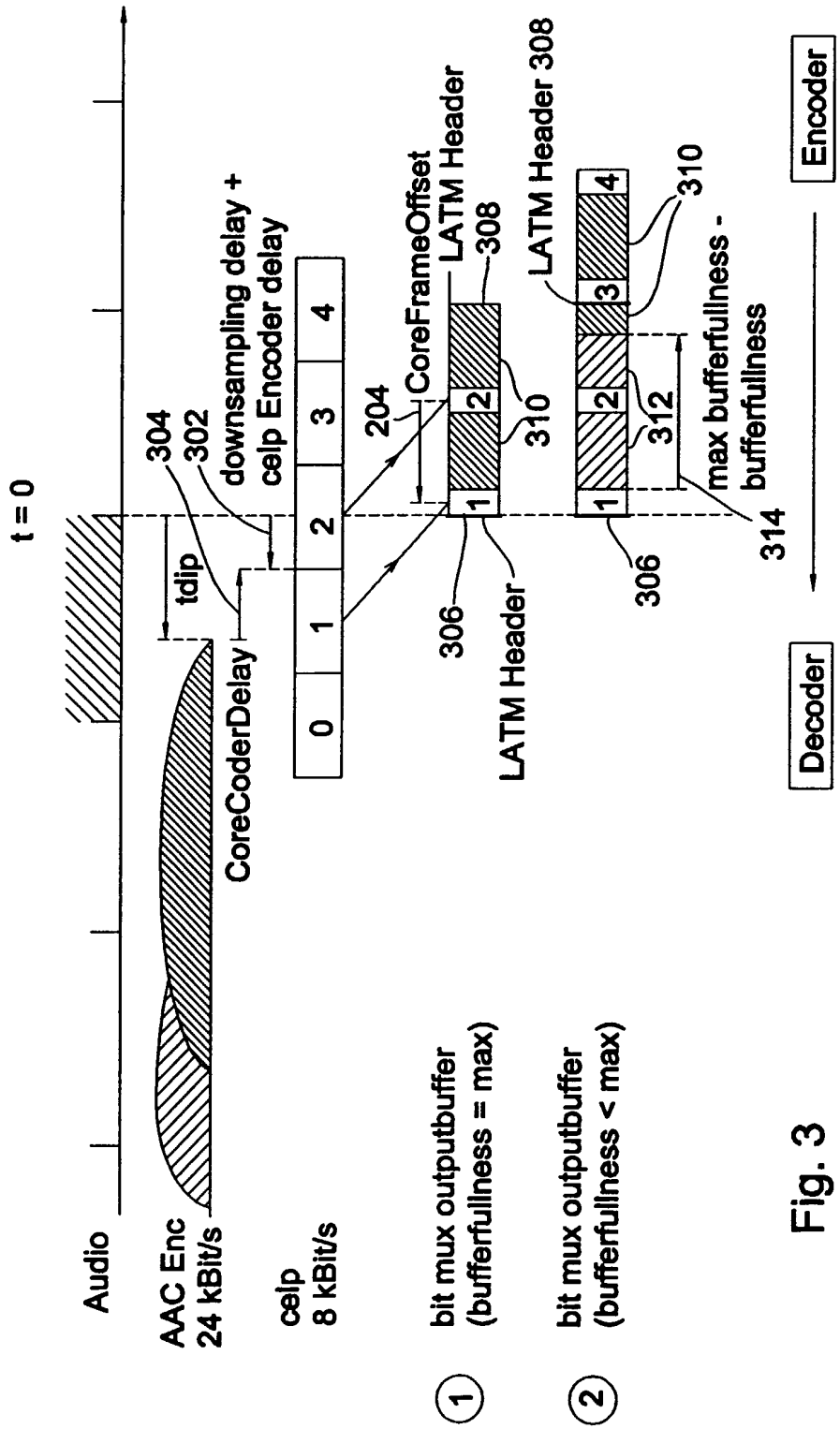

t = 0
Audio
AAC Enc
24 kBit/s
celp
8 kBit/s
CoreCoderDelay
tdip
304
302
downsampling delay +
celp Encoder delay
0
1
2
3
4
204
CoreFrameOffset
LATM Header
308
310
306
LATM Header
LATM Header 308
312
max bufferfullness -
bufferfullness
314
bit mux outputbuffer
(bufferfullness = max)
bit mux outputbuffer
(bufferfullness < max)
Decoder
Encoder

Fig. 3

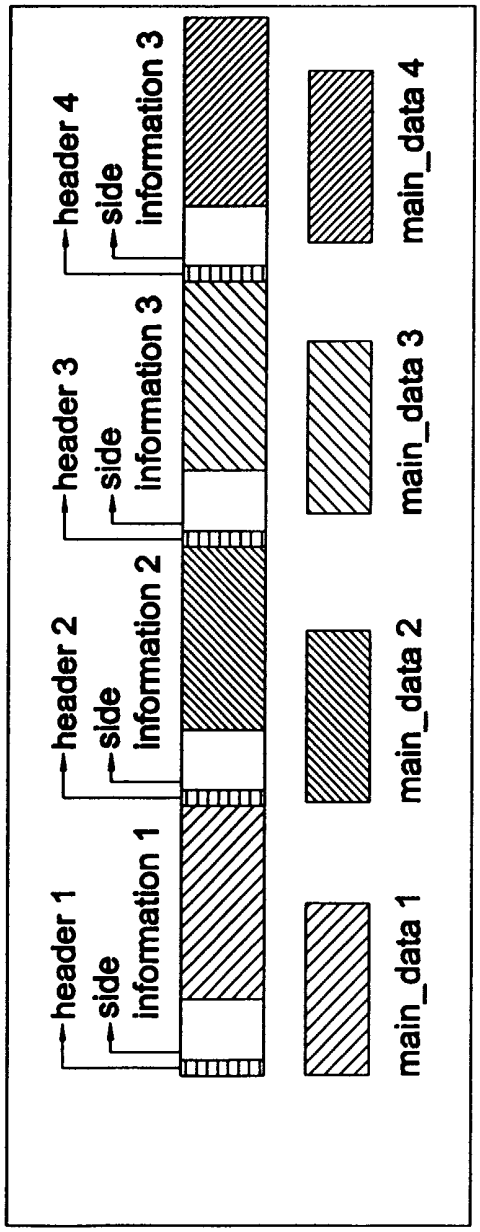

header 1
side information 1
header 2
side information 2
header 3
side information 3
header 4
side information 3
main_data 1
main_data 2
main_data 3
main_data 4

Fig. 4

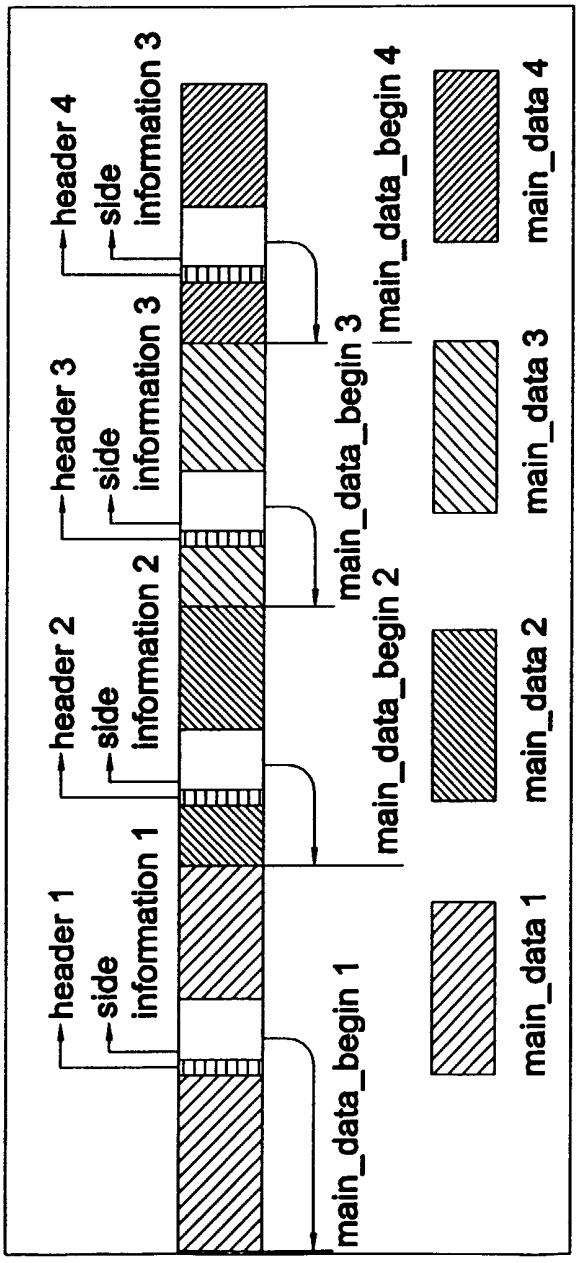

header 1
side information 1
header 2
side information 2
header 3
side information 3
header 4
side information 3
main_data_begin 1
main_data_begin 2
main_data_begin 3
main_data_begin 4
main_data 1
main_data 2
main_data 3
main_data 4

Fig. 5

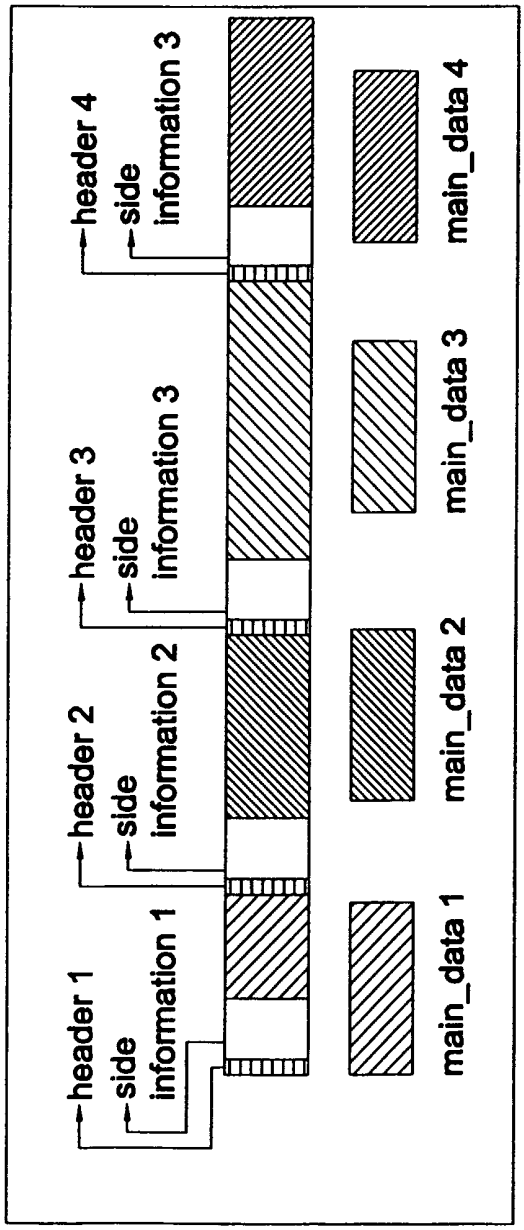

header 1
side information 1
header 2
side information 2
header 3
side information 3
header 4
side information 3
main_data 1
main_data 2
main_data 3
main_data 4

Fig. 6