(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2009-500688
(P2009-500688A)
(43)公表日　平成21年1月8日(2009.1.8)

(51)Int.Cl.		F I		テーマコード（参考）
G10L 19/14	(2006.01)	G10L 19/14	400Z	
G10L 19/00	(2006.01)	G10L 19/00	250	

審査請求 有　予備審査請求 未請求　（全 37 頁）

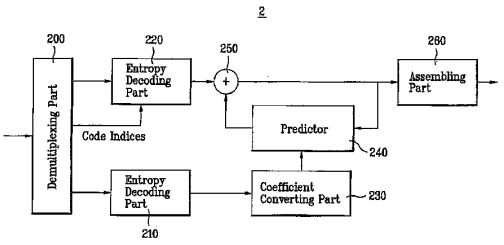
(21)出願番号　特願2008-521313 (P2008-521313)
(86)(22)出願日　平成18年7月10日 (2006.7.10)
(85)翻訳文提出日　平成20年3月10日 (2008.3.10)
(86)国際出願番号　PCT/KR2006/002685
(87)国際公開番号　WO2007/008007
(87)国際公開日　平成19年1月18日 (2007.1.18)
(31)優先権主張番号　60/697,551
(32)優先日　平成17年7月11日 (2005.7.11)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　PCT/KR2005/002290
(32)優先日　平成17年7月16日 (2005.7.16)
(33)優先権主張国　韓国 (KR)
(31)優先権主張番号　PCT/KR2005/002291
(32)優先日　平成17年7月16日 (2005.7.16)
(33)優先権主張国　韓国 (KR)

(71)出願人　502032105
エルジー　エレクトロニクス　インコーポレイティド
大韓民国，ソウル　１５０－７２１，ヨンドゥンポ－ク，ヨイド－ドン，２０
(74)代理人　100099759
弁理士　青木　篤
(74)代理人　100092624
弁理士　鶴田　準一
(74)代理人　100102819
弁理士　島田　哲郎
(74)代理人　100114018
弁理士　南山　知広
(74)代理人　100108383
弁理士　下道　晶久

最終頁に続く

(54)【発明の名称】オーディオ信号の処理装置及び方法

(57)【要約】
【課題】オーディオ信号の処理装置及び方法を提供する。
【解決手段】一実施形態において、サンプル情報及びフレーム長情報はオーディオ信号から取得される。サンプル情報はオーディオ信号内の各チャンネルに対するオーディオデータサンプルの総数を示し、フレーム長情報は各チャンネルのフレーム内のサンプル数を示す。サンプル情報がフレーム長情報により除算される場合、余りが存在するかどうかが判定され、余りが存在すれば、オーディオ信号内のフレーム数はフレーム情報により除算されたサンプル情報の整数値に１を加算した値として判定される。また、最後のフレームの長さは、前記判定ステップにおいて余りが存在すると判定した場合に、その余りに基づいて設定可能である。データサンプルは、判定されたフレーム数に基づいてデコーディングされる。
【選択図】図2



【特許請求の範囲】
【請求項１】
オーディオ信号の処理方法であって、
前記オーディオ信号内の各チャンネルに対するオーディオデータサンプルの総数を示すサンプル情報を前記オーディオ信号から取得するステップと、
各チャンネルのフレーム内のサンプル数を示すフレーム長情報を前記オーディオ信号から取得するステップと、
前記サンプル情報が前記フレーム長情報により除算される場合、余りが存在するかどうかを判定するステップと、
前記判定ステップにおいて余りが存在すると判定した場合、前記オーディオ信号内のフレーム数を、前記フレーム情報により除算された前記サンプル情報の情報値に１を加算した値として判定するステップと、
前記判定ステップにおいて余りが存在すると判定した場合、前記余りに基づいて最後のフレームの長さを設定するステップと、
前記判定されたフレーム数に基づいて、前記データサンプルをデコーディングするステップと、
を有することを特徴とする方法。
【請求項２】
前記設定ステップは、前記余りに等しい最後のフレームの長さを設定する、請求項１に記載の方法。
【請求項３】
前記判定ステップにおいて余りが存在しないと判定した場合、前記オーディオ信号内のフレーム数を、前記フレーム情報により除算された前記サンプル情報の整数値として判定するステップをさらに有する、請求項１に記載の方法。
【請求項４】
前記判定ステップにおいて余りが存在しないと判定した場合、前記フレーム情報に等しい最後のフレームの長さを設定するステップをさらに有する、請求項３に記載の方法。
【請求項５】
前記サンプル情報取得ステップは、前記オーディオ信号の構成情報内のサンプルフィールドから前記サンプル情報を取得する、請求項１に記載の方法。
【請求項６】
前記フレーム長取得ステップは、前記オーディオ信号の構成情報内のフレーム長フィールドから前記フレーム長情報を取得する、請求項１に記載の方法。
【請求項７】
オーディオ信号の処理方法であって、
前記オーディオ信号内の各チャンネルに対するオーディオデータのトータル数を示すサンプル情報を前記オーディオ信号に付加するステップと、
各チャンネルのフレーム内のサンプル数を示すフレーム長情報を付加するステップと、
前記オーディオ信号内のオーディオデータをエンコーディングして、（ｉ）前記サンプル情報が前記フレーム長情報により除算された場合に、余りが存在すれば、前記オーディオ信号内のフレーム数が、前記フレーム情報に１を加算した値で除算された前記サンプル情報の整数値としてデコーダにおいて判定され、（ii）前記余りが存在すれば、前記余りに基づいて、最後のフレームの長さが前記デコーダにおいて設定されるエンコーディングステップと、
を有することを特徴とする方法。
【請求項８】
オーディオ信号を処理する装置であって、
前記オーディオ信号内の各チャンネルに対するオーディオデータサンプルのトータル数を示すサンプル情報を前記オーディオ信号から取得し、各チャンネルのフレーム数を示すフレーム長情報を取得するデコーダを有し、

前記デコーダは、前記サンプル情報が前記フレーム長情報により除算された場合に、余りが存在するかどうかを判定し、余りが存在すれば、前記フレーム情報に１を加算した値で除算された前記サンプル情報の整数値として前記オーディオ信号内のフレーム数を判定し、
前記デコーダは、余りが存在すれば、前記余りに基づいて最後のフレームの長さを設定し、
前記デコーダは、前記判定されたフレーム数に基づいて、前記データサンプルをデコーディングする、
ことを特徴とする装置。
【請求項９】
オーディオ信号の処理装置であって、
前記オーディオ信号内の各チャンネルに対するオーディオデータのトータル数を示すサンプル情報を前記オーディオ信号に付加し、各チャンネルのフレーム内のサンプル数を示すフレーム長情報を付加するエンコーダを有し、
前記エンコーダは、前記オーディオ信号内のオーディオデータをエンコーディングして、（ｉ）前記サンプル情報が前記フレーム長情報により除算された場合に、余りが存在すれば、前記オーディオ信号内のフレーム数が、前記フレーム情報に１を加算した値で除算された前記サンプル情報の整数値としてデコーダにおいて判定され、（ii）前記余りが存在すれば、前記余りに基づいて最後のフレームの長さが前記デコーダにおいて設定される、
ことを特徴とする装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明はオーディオ信号を処理する方法に係り、特に、オーディオ信号をエンコーディングし且つデコーディングする方法及び装置に関する。
【背景技術】
【０００２】
従来には、オーディオ信号の保存及び再生は異なる方式により行われていた。例えば、音楽及び音声は蓄音機技術（例えば、レコードプレーヤ）、磁気技術（例えば、カセットテープ）、及びデジタル技術（例えば、コンパクトディスク）により記録され且つ保存されていた。オーディオ保存技術の進歩に伴い、オーディオ信号の品質及び保存性を最適化させるために多くの問題点を克服する必要がある。
【０００３】
音楽信号のアーカイブ及び広帯域送信において、無損失再構成は、ＭＰ３またはＡＡＣなどのＭＰＥＧ標準において定義された概念的なコーディングによる圧縮に際して、高効率よりもなお一層重要な特徴となっている。たとえ、ＤＶＤオーディオ及びスーパＣＤオーディオが独自仕様の無損失圧縮スキームを含んでいるとしても、コンテンツ保有者及び放送業体に開放的で且つ一般的な圧縮スキームが求められるようになった。このような要求に応えて、新規な無損失コーディングスキームがＭＰＥＧ－４オーディオ標準に対する拡張として考慮される。無損失オーディオコーディングは原信号の完全な再構成に起因する品質の損失なしにデジタルオーディオデータを圧縮可能にする。
【発明の開示】
【発明が解決しようとする課題】
【０００４】
本発明はオーディオ信号の処理方法に関する。
【課題を解決するための手段】
【０００５】
一実施形態において、サンプル情報及びフレーム長情報はオーディオ信号から取得される。サンプル情報はオーディオ信号内の各チャンネルに対するオーディオデータサンプル

の総数を示し、フレーム長情報は各チャンネルのフレーム内のサンプル数を示す。サンプル情報がフレーム長情報により除算される場合に、余りが存在するかどうかを判定し、余りが存在すれば、オーディオ信号内のフレーム数は、フレーム情報により除算されたサンプル情報の整数値に１を加算した値として判定する。また、最後のフレームの長さは、前記判定ステップにおいて余りが存在すると判定した場合、その余りに基づいて設定可能である。データサンプルは、判定されたフレーム数に基づいてデコーディングされる。

【０００６】

一実施形態において、最後のフレームの長さは余りに等しく設定される。

他の実施形態において、オーディオ信号内のフレーム数は、前記判定ステップにおいて余りが存在しないと判定した場合、フレーム情報により除算された整数値として判定される。

【発明を実施するための最良の形態】

【０００７】

以下、添付図面に基づき、本発明の好適な実施形態について説明する。なお、図中、同じ構成要素にはできる限り同じ符号を付してある。

【０００８】

本発明を記述する前に、本発明において開示されたほとんどのターミノロジーは公知の汎用用語であるが、一部の用語は必要に応じて選択されて次の説明において用いられるところに留意すべきである。よって、出願人により定義された用語は本発明においてその意味に基づいて理解すべきものである。

【０００９】

無損失オーディオコーディング方法において、エンコーディングプロセスが情報の損失無しに完全な可逆性を有する必要があるため、エンコーダ及びデコーダの両方の種々の構成要素は決定的な方式により実現できなければならない。

【００１０】

コーデック構造

図１は、本発明によるエンコーダ１の例示図である。

【００１１】

区画部１００は、入力オーディオデータをフレームに区画する。１フレーム内において、各チャンネルはさらなる処理のためにオーディオサンプルにサブ分割される。バッファ１１０は、区画部１００により区画されたブロック及び／またはフレームサンプルを保存する。

【００１２】

係数推定部１２０は、各ブロックに対する最適な組の係数値を推定する。係数の数、すなわち、予測器の順序は適宜に選択可能である。係数推定部１２０は、デジタルオーディオデータのブロックに対してパーコール値の組を演算する。パーコール値（Ｐａｒｃｏｒｖａｌｖｅ）は、予測係数を示すパーコールを示す。量子化部１３０は、パーコール値組を量子化する。

【００１３】

第１のエントロピー符号化部１４０は、パーコール値からオフセット値を差し引いてパーコール余り値を演算し、エントロピーパラメータにより定義されたエントロピー符号化を用いてパーコール余り値をエンコーディングする。ここで、オフセット値及びエントロピーパラメータは最適なテーブルから選択される。最適なテーブルは、デジタルオーディオデータのブロックのサンプリングレートに基づいて複数のテーブルから選択される。複数のテーブルは、送信用のデジタルオーディオデータの最適な圧縮のために、複数のサンプリング範囲のそれぞれに対して予め定義される。

【００１４】

係数変換部１５０は、量子化されたパーコール値を線形予測符号化（ＬＰＣ：Ｌｉｎｅａｒ Ｐｒｅｄｉｃｔｉｖｅ Ｃｏｄｉｎｇ）係数に変換する。予測器１６０は、ＬＰＣ係数を用いてバッファ１１０内に保存されている以前の原サンプルから現在の予測値を推定

する。減算器１７０は、バッファ１１０に保存されているデジタルオーディオデータのオリジナル値を用いてデジタルオーディオデータのブロックの余り予測及び予測器１６０により推定された予測値の演算を行う。

【００１５】

第２のエントロピー符号化部１８０は、他のエントロピーコードを用いて余り予測をコーディングし、コードインデックスを生成する。選択されたコードのインデックスは、補助情報として送信される。第２のエントロピー符号化部１８０は、異なる複雑性を有する２つの代案的なコーディング技術のうちいずれか一方を用いて余り予測をコーディングする。一方のコーディング技術は公知のゴロム－ライス（Ｇｏｌｏｍｂ－Ｒｉｃｅ）コーディング（以下、「ライスコーディング」と略す。）方法であり、他方のコーディング技術は公知のブロックギルバート－ムーアコード（以下、「ＢＧＭＣ」と略す。）方法である。ライスコードは複雑性は低いものの、効率的である。ＢＧＭＣ算術コーディングスキームは、ライスコードに比べてやや増加して複雑性を犠牲にして遥かに良好な圧縮を提供する。

【００１６】

結局として、マルチプレックス部１９０は、コーディングされた余り予測、コードインデックス、コーディングされたパーコール余り値、及び他の付加情報をマルチプレックスして圧縮されたビットストリームを形成する。また、エンコーダ１は、デコーディングされたデータを検証するためにデコーダにおいて主として提供される巡回冗長検査（ＣＲＣ：Ｃｙｃｌｉｃ Ｒｅｄｕｎｄａｎｃｙ Ｃｈｅｃｋ）チェックサムを提供する。エンコーダ側上において、ＣＲＣは圧縮されたデータが無損失にてデコーディング可能であることを保証するのに用いられる。

【００１７】

付加的なエンコーディングオプションは、フレキシブルブロックスイッチングスキーム、ランダムアクセス及びジョイントチャンネルコーディングを含む。エンコーダ１は、種々の圧縮レベルに異なる複雑度を与えるためにこれらのオプションを用いる。ジョイントチャンネルコーディングは、ステレオチャンネルまたはマルチチャンネル信号同士の従属性を活用するのに用いられる。これは、２つのチャンネル間の差分が原チャンネルにおいてより効率よくコーディング可能なセグメントにおけるこのような差分をコーディングして達成することができる。このようなエンコーディングオプションの詳細については、本発明による例示的なデコーダの説明後に行う。

【００１８】

図２は、本発明によるデコーダ２の例示図である。特に、図２は、エンコーダよりもやや複雑性の低い無損失オーディオ信号デコーダを示し、その理由は、いかなる適用も行われないためである。

【００１９】

デマルチプレックス部２００は、オーディオ信号を受信し、デジタルオーディオデータのブロックのコーディングされた余り予測、コードインデックス、コーディングされたパーコール余り値、及び他の付加情報をデマルチプレックスする。第１のエントロピー復号化部２１０は、エントロピーパラメータにより定義されたエントロピーコードを用いてパーコール余り値をデコーディングし、デコーディングされたパーコール余り値にオフセット値を加えてパーコール値の組を演算する。ここで、オフセット値及びエントロピーパラメータはデジタルオーディオデータのブロックのサンプルレートに基づいて複数のテーブルからデコーダにより選択されたテーブルから選択される。第２のエントロピー復号化部２２０は、コードインデックスを用いてデマルチプレックスされたコーディングされた余り予測をデコーディングする。係数変換部２３０は、エントロピー復号化されたパーコール値をＬＰＣ係数に変換する。予測器２４０は、ＬＰＣ係数を用いてデジタルオーディオデータのブロックの予測余りを推定する。加算器２５０は、デコーディングされた予測余りを推定された予測余りに加えてデジタルオーディオデータの原ブロックを取得する。アセンブリング部２６０は、デコーディングされたブロックデータをフレームデータにアセ

ンブリングする。

【００２０】

この後、デコーダ２は、コーディングされた予測余り及びパーコール余り値をデコーディングし、パーコール余り値をＬＰＣ係数に変換し、逆予測フィルタを適用して無損失再構成信号を演算する。デコーダ２の計算量は、エンコーダ１により選択された予測順序による。ほとんどの場合、リアルタイムなデコーディングはロウエンドシステムにおいてさえ行うことが可能である。図３は、本発明による複数のチャンネル（例えば、Ｍチャンネル）を含む圧縮オーディオ信号のビットストリーム構造の例示図である。ビットストリームは、複数のチャンネル（例えば、Ｍチャンネル）を含む少なくとも１枚のオーディオフレームから構成される。ビットストリーム構成シンタックス（テーブル６参照）の「チャンネル」フィールドは、チャンネル数を示す。各チャンネルは、後述するように、本発明によるブロックスイッチングスキームを用いて複数のブロックにサブ分割される。各サブ分割されたブロックは異なるサイズを有し、図１のエンコーディングによりコーディングデータを含む。例えば、サブ分割されたブロック内のコーディングデータは、コードインデックス、予測順序Ｋ、予測係数、及びコーディングされた余り値を含む。チャンネル対同士のジョイントコーディングが用いられる場合、ブロック区画は両チャンネルにおいて同じであり、ブロックはインタリーブ方式により保存される。ビットストリーム構成シンタックス（テーブル６）内の「ｊｓ＿ｓｔｅｒｅｏ」フィールドは、ジョイントステレオ（チャンネル差分）がオンであるか、またはオフであるかを示し、フレーム＿データシンタックス（テーブル７）内の「ｊｓ＿ｓｗｉｔｃｈ」フィールドは、ジョイントステレオ（チャンネル差分）が選択されているかどうかを示す。そうでなければ、各チャンネルに対するブロック区画は独立的である。

【００２１】

以下、添付図面に基づき、ブロックスイッチング、ランダムアクセス、区画、及び上述したエントロピー符号化オプションについてシンタックスを参考として詳述する。

【００２２】

ブロックスイッチング

本発明の一態様は、実際にコーディングスキームを用いる前に各チャンネルを複数のブロックにサブ分割することと関連する。以下、本発明によるブロック区画（または、サブ分割）方法は「ブロックスイッチング方法」として言及される。

【００２３】

階層ブロックスイッチング

図４は、本発明による階層ブロックスイッチング方法の概念の例示図である。例えば、図４は、１チャンネルを３２ブロックに階層的にサブ分割する方法を示す。複数のチャンネルが単一のフレームに与えられるとき、各チャンネルは３２個のブロックまでサブ分割（または、分割）可能であり、各チャンネルに対するサブ分割されたブロックは１フレームを構成する。このため、本発明によるブロックスイッチング方法は、図１に示す区画部１００により行われる。さらに、上述したように、予測及びエントロピー符号化はサブ分割されたブロック単位で行われる。

【００２４】

一般に、従来のオーディオ無損失符号化（ＡＬＳ：Ａｕｄｉｏ Ｌｏｓｓｌｅｓｓ Ｃｏｄｉｎｇ）は、比較的に単純なブロックスイッチングメカニズムを含む。Ｎサンプルの各チャンネルは１つの完全長さのブロック（N_B＝Ｎ）または４個ブロックの長さN_B＝Ｎ／４（例えば、１：４スイッチング）を用いてエンコーディングされ、ここで、同じブロック分割がすべてのチャンネルに適用される。一部の状況下においては、このようなスキームは一部の制限を有する。例えば、単に１：１または１：４スイッチングのみが可能であることがあり、他のスイッチング（例えば、１：２、１：８及びこれらの組み合わせ）の方が一部の場合にはさらに有効であることがある。また、従来のＡＬＳにおいて、スイッチングは他のチャンネルが他のスイッチング（チャンネルが相関しない場合には特に真である）から利得を得るとしても、全てのチャンネルに対して同様に行われる。

【００２５】
このため、本発明の実施形態によるブロックスイッチング方法は比較的にフレキシブルなブロックスイッチングスキームを提供し、１フレームの各チャンネルは複数のブロックに階層的にサブ分割される。例えば、図４は、３２個のブロックに階層的にサブ分割可能な１チャンネルを示す。N_B＝Ｎ、Ｎ／２、Ｎ／４、Ｎ／８、Ｎ／１６、及びＮ／３２とのブロックの任意の組み合わせは、各ブロックが２重長さの上位ブロックのサブ分割から生じる限り、この実施形態によりチャンネル内において可能でありうる。例えば、図４に例示するように、Ｎ／４＋Ｎ／４＋Ｎ／２への分割は可能であるが、Ｎ／４＋Ｎ／２＋Ｎ／４への分割は不可である（例えば、図５（ｅ）及び図５（ｆ）に示すブロックスイッチング例は後述する）。換言すると、チャンネルは複数のブロックに分割されて、各ブロックがｉ＝１、２、...ｐに対するＮ／（m^i）のうちいずれかと同じ長さを有する。ここで、Ｎはチャンネルの長さ、ｍは２以上の整数、ｐはサブ分割階層のレベル数を示す。

【００２６】
このため、本発明の実施形態において、ビットストリームは、ブロックスイッチングレベルを示す情報と、ブロックスイッチング結果を示す情報と、を含む。ここで、ブロックスイッチングと関連する情報は、後述するように、デコーディングプロセスにおいて用いられるシンタックスに含まれる。

【００２７】
例えば、ブロックスイッチングプロセス後に生成された最大ブロックサイズがN_B＝Ｎ／３２になるように設定が行われる。しかしながら、このような設定は、本発明の説明を簡略化するための例に過ぎない。このため、本発明による設定はこのような設定に制限されない。

【００２８】
特に、最大ブロックサイズがN_B＝Ｎ／３２の場合、ブロックスイッチングプロセスが、レベル５ブロックスイッチングとして言及される５回に亘って階層的に行われる。代案的に、最大のブロックサイズがN_B＝Ｎ／１６の場合、ブロックスイッチングプロセスが、レベル４ブロックスイッチングとして言及される４回に亘って階層的に行われる。同様に、最大のブロックサイズがN_B＝Ｎ／８の場合、ブロックスイッチングプロセスは、レベル３ブロックスイッチングとして言及される３回に亘って階層的に行われる。最大のブロックサイズがN_B＝Ｎ／４の場合、ブロックスイッチングプロセスは、レベル２ブロックスイッチングとして言及される２回に亘って階層的に行われる。最大のブロックサイズがN_B＝Ｎ／２の場合、ブロックスイッチングプロセスは、レベル１ブロックスイッチングとして言及される１回に亘って階層的に行われる。結局として、最大のブロックサイズがN_B＝Ｎの場合、階層的なブロックスイッチングプロセスは行われることなく、これは、レベル０ブロックスイッチングとして言及される。

【００２９】
本発明の実施形態において、ブロックスイッチングを示す情報は、第１のブロックスイッチング情報として言及される。例えば、第１のブロックスイッチング情報は、後述するプロセスにおいて説明されるように、テーブル６に示すシンタックス内の２ビット「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ」フィールドにより表現される。特に、「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝００」はレベル０を、「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝０１」はレベル１から３のうち少なくともいずれか１つを、「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝１０」はレベル４を、そして「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝１１」はレベル５を示す。

【００３０】
また、上述したブロックスイッチングレベルによる、各階層的なレベルに対して行われたブロックスイッチングの結果を示す情報は、実施形態において第２のブロックスイッチング情報として言及される。以下、第２のブロックスイッチング情報は、テーブル７に示すシンタックス内の８ビット、１６ビット及び３２ビットのうち少なくともいずれか１つにより表現される。特に、「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝０１」の場合（レベル１

から３のいずれかを示す場合）、「ｂｓ＿ｉｎｆｏ」は８ビットにより表現される。「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝１０」の場合（レベル４を示す場合）、「ｂｓ＿ｉｎｆｏ」は１６ビットにより表現される。すなわち、ブロックスイッチング結果の４レベルまでは１６ビットを用いて表現される。さらに、「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝１１」の場合（レベル５を示す場合）、「ｂｓ＿ｉｎｆｏ」は３２ビットにより表現される。すなわち、ブロックスイッチング結果の５レベルまでは３２ビットを用いて表現される。結局として、「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝００」の場合（ブロックスイッチングが行われない旨を示す場合）、「ｂｓ＿ｉｎｆｏ」は送信されない。これは、１チャンネルが１ブロックを構成することを示す。

【００３１】

第２のブロックスイッチング情報に割り当てられたビットの総数は、第１のブロックスイッチング情報のレベル値に基づいて決まる。これは、最終的なビットレートを減らす結果を招く。第１のブロックスイッチング情報と第２のブロックスイッチング情報との間の関係は、以下のテーブル１に簡略に説明する。

【００３２】

【表１】

テーブル１：ブロックスイッチングレベル

最大のレベル数	最小N_B	「ｂｓ＿ｉｎｆｏ」に対するバイト数
０（「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝００」）	Ｎ	０
１（「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝０１」）	Ｎ／２	１（＝８ｂｉｔｓ）
２（「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝０１」）	Ｎ／４	１（＝８ｂｉｔｓ）
３（「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝０１」）	Ｎ／８	１（＝８ｂｉｔｓ）
４（「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝１０」）	Ｎ／１６	２（＝１６ｂｉｔｓ）
５（「ｂｌｏｃｋ＿ｓｗｉｔｃｈｉｎｇ＝１１」）	Ｎ／３２	４（＝３２ｂｉｔｓ）

【００３３】

以下、第２のブロックスイッチング情報ｂｓ＿ｉｎｆｏ内の各ビットを構成（または、マッピング）する方法の実施形態について説明する。

【００３４】

ｂｓ＿ｉｎｆｏフィールドは、上述した実施形態により４バイトまでを含む。レベル１から５に関するビットのマッピングは［（０）１２２３３３３　４４４４４４４４　５５５５５５５５５５５５５５５］でありうる。第１のビットは、独立／同期ブロックスイッチングのセクションにおいて詳述する独立的なまたは同期化ブロックスイッチングを示すように保有される。図５（ａ）から図５（ｆ）は、レベル３ブロックスイッチングが行われるチャンネルに対する他のブロックスイッチング例を示す。このため、これらの例において、最大のブロック長はN_B＝Ｎ／８であり、ｂｓ＿ｉｎｆｏは１バイトからなる。最大のブロック長N_B＝Ｎから始まるとすれば、ｂｓ＿ｉｎｆｏはブロックがさらにサブ分割される場合に設定される。例えば、図５（ａ）においてはサブ分割がなくて、「ｂｓ＿ｉｎｆｏ」は（０）０００　００００である。図５（ｂ）において、フレームは（（０）１...）にサブ分割され、長さＮ／２の第２のブロックは、（（０）１０１...）をＮ／４長さの２つのブロックに分割して、「ｂｓ＿ｉｎｆｏ」は（０）１０１０　０００）であ

る。図５（ｃ）において、フレームは（（０）１...）にサブ分割され、Ｎ／２長さの第１のブロックだけが（（０）１１０...）をＮ／４長さの２つのブロックに分割して、「ｂｓ＿ｉｎｆｏ」は（０）１１００　００００となる。図５（ｄ）において、フレームは（（０）１...）にサブ分割され、Ｎ／２長さの第１及び第２のブロックが（（０）１１１...）をＮ／４長さの２つのブロックにさらに分割し、長さＮ／４の第２のブロックだけが（（０）１１１０１...）を長さＮ／８の２つのブロックにさらに分割して、「ｂｓ＿ｉｎｆｏ」は（０）１１１　０１００となる。

【００３５】

上述したように、図５（ｅ）及び図５（ｆ）の例は、図５ＥのＮ／２ブロックと図５Ｆの第１のＮ／４ブロックが以前のレベルのブロックをサブ分割して取得できないために許容されないブロックスイッチングの場合を示す。

【００３６】

独立／同期化ブロックスイッチング

図６（ａ）から図６（ｃ）は、本発明の実施形態によるブロックスイッチングの例示図である。

【００３７】

特に、図６（ａ）は、チャンネル１、２及び３に対してブロックスイッチングが行われない例を示す。図６（ｂ）は、２つのチャンネル（チャンネル１及び２）が一つのチャンネル対を構成する例を示し、ブロックスイッチングは、チャンネル１及び２において同期的に行われる。インタリーブもまたこの例に適用される。図６（ｃ）は、２つのチャンネル（チャンネル１及び２）が一つのチャンネル対を構成する例を示し、チャンネル１及び２のブロックスイッチングは独立して行われる。以下、チャンネル対は、２つの任意のオーディオチャンネルを指し示す。どのチャンネルがチャンネル対にグループ分けされるかの判断は、エンコーダにより自動的に行われうるか、または、ユーザによりマニュアルにて行われうる（例えば、Ｌ、Ｒチャンネル、Ｌｓ、Ｒｓチャンネル）。

【００３８】

独立ブロックスイッチングに際し、たとえ、各チャンネルの長さが全てのチャンネルに対して同じであるとしても、ブロックスイッチングは各チャンネルに対して独立して行われることが可能である。すなわち、図６（ｃ）に示すように、チャンネルはブロックに異なって分割される。チャンネル対の２つのチャンネルが互いに相関し、且つ、差分コーディングが用いられる場合、チャンネル対の両チャンネルは同期化されてブロックスイッチングされる。同期化ブロックスイッチングの際に、チャンネルは同じ方式によりブロックスイッチング（すなわち、ブロックに分割）される。図６（ｂ）はこのような例を示し、ブロックがインタリーブされることをさらに示す。チャンネル対のうち２つのチャンネルが互いに相関していない場合に差分コーディングは利点がなく、これにより、チャンネルを同期的にブロックスイッチングする必要がない。むしろ、チャンネルを独立してスイッチングした方がさらに好適である。さらに、本発明の他の実施形態によれば、独立または同期化ブロックスイッチングの上述した方法は、３以上のチャンネル数を有するマルチチャンネル群に適用される。例えば、マルチチャンネル群の全てのチャンネルが互いに相関している場合、マルチチャンネル群の全てのチャンネルは同期的にスイッチングされる。一方、マルチチャンネル群の全てのチャンネルが互いに相関していない場合にマルチチャンネル群の各チャンネルは独立してスイッチングされる。

【００３９】

さらに、「ｂｓ＿ｉｎｆｏ」フィールドはブロックスイッチング結果を示す情報として用いられる。また、「ｂｓ＿ｉｎｆｏ」フィールドは、ブロックスイッチングがチャンネル対を構成する各チャンネルに対して独立してまたは同期化して行われるかどうかを示す情報として用いられる。この場合、上述したように、「ｂｓ＿ｉｎｆｏ」フィールド内の特定のビット（例えば、第１のビット）が用いられる。例えば、チャンネル対の２つのチャンネルが互いに独立している場合、「ｂｓ＿ｉｎｆｏ」フィールドの第１のビットは「１」に設定される。一方、チャンネル対の２つのチャンネルが互いに同期化される場合に

「ｂｓ＿ｉｎｆｏ」フィールドの第１のビットは「０」に設定される。

【００４０】

以下、図６（ａ）から図６（ｃ）について説明する。

【００４１】

図６（ａ）を参照すれば、いかなるチャンネルにおいてもブロックスイッチングが行われないため、関連する「ｂｓ＿ｉｎｆｏ」は生成されない。

【００４２】

図６（ｂ）を参照すれば、チャンネル１及び２はチャンネル対を構成し、２つのチャンネルは互いに同期化され、ブロックスイッチングが同期的に行われる。例えば、図６（ｂ）において、チャンネル１及び２の両方は長さＮ／４のブロックに分割され、同じｂｓ＿ｉｎｆｏ「ｂｓ＿ｉｎｆｏ＝（０）１０１　００００」を有する。このため、１つの「ｂｓ＿ｉｎｆｏ」は各チャンネル対に対して送信されて、ビットレートを減らすという結果を招く。さらに、チャンネル対が同期化されれば、チャンネル対内の各ブロックは互いにインタリーブされることが求められる。インタリーブは利点（または、効果）がある。例えば、チャンネル対内の１チャンネルのブロック（例えば、図６（ｂ）のブロック１．２）は両チャンネル内の以前のブロック（例えば、図６（ｂ）のブロック１．１及び２．１）に従属し、このため、これらの以前のブロックは現在ブロックに先行して利用可能なものである必要がある。

【００４３】

図６（ｃ）を参照すれば、チャンネル１及び２はチャンネル対を構成する。しかしながら、この例において、ブロックスイッチングは独立して行われる。特に、チャンネル１はＮ／４までのサイズ（または、長さ）のブロックに分割され、「ｂｓ＿ｉｎｆｏ＝（１）１０１　００００」のｂｓ＿ｉｎｆｏを有する。チャンネル２はＮ／２までのサイズのブロックに分割され、「ｂｓ＿ｉｎｆｏ＝（１）１００　００００」のｂｓ＿ｉｎｆｏを有する。図６Ｃに示す例において、ブロックスイッチングは各チャンネル間に独立して行われ、このため、ブロック間のインタリーブプロセスは行われない。すなわち、ブロックが独立して行われたチャンネルにおいて、チャンネルデータは独立して配列される。

【００４４】

ジョイントチャンネルコーディング

また、ジョイントチャンネルコーディングはジョイントステレオとも呼ばれ、ステレオ信号の２チャンネル間またはマルチチャンネル信号の任意の２チャンネル間の従属性を活用するのに使用可能である。たとえ、２チャンネルx_1（ｎ）及びx_2（ｎ）を独立して処理することが容易であるとしても、チャンネル間の従属性を活用する単純な方法は、差分信号、x_1（ｎ）またはx_2（ｎ）の代わりにｄ（ｎ）＝x_2（ｎ）－x_1（ｎ）をエンコーディングすることである。

【００４５】

各ブロックにおけるx_1（ｎ）、x_2（ｎ）及びｄ（ｎ）間のスイッチングは。２つの信号が最も効率よくコーディング可能な個別信号の比較に従属して行われる。スイッチングされた差分コーディングによるこのような予測は、２つのチャンネルが互いに極めて類似している場合に利点がある。マルチチャンネルの場合、チャンネルは適宜なチャンネル対を割り当てるためにエンコーダにより並替え可能である。

【００４６】

また、単純な差分コーディングに加えて、無損失オーディオコーデックは、マルチチャンネル信号の任意のチャンネル間のチャンネル間冗長を活用するさらに複雑なスキームを支援する。

【００４７】

ランダムアクセス

本発明はオーディオ無損失コーディングに関するものであり、ランダムアクセスに対応可能である。ランダムアクセスは、以前の部分のデコーディングに高いコストをかけることなくエンコーディングされたオーディオ信号の所定の部分を高速にてアクセスすること

を示す。圧縮されたデータの検索、編集またはストリーミングを採用するアプリケーションに重要な特徴がある。ランダムアクセスを可能にするためには、ランダムアクセス部内において、エンコーダが以前のフレームをデコーディングすることなくデコーディング可能なフレームを挿入する必要がある。挿入されたフレームは「ランダムアクセスフレーム」として言及される。このようなランダムアクセスフレームにおいて、以前のフレームからいかなるサンプルも予測に用いられない。

【００４８】

以下、本発明によるランダムアクセス用の情報について説明する。構成シンタックスを参照すれば（テーブル６参照）、ランダムアクセスと関連する情報は構成情報として送信される。例えば、「ｒａｎｄｏｍ＿ａｃｃｅｓｓ」フィールドは、ランダムアクセスが許容されるかどうかを示す情報として用いられ、８ビットを用いて表現される。さらに、ランダムアクセスが許容される場合、８ビットの「ｒａｎｄｏｍ＿ａｃｃｅｓｓ」フィールドは、ランダムアクセス部を構成するフレーム数を指定する。例えば、「ｒａｎｄｏｍ＿ａｃｃｅｓｓ＝０００００　００００」の場合、ランダムアクセスには対応されない。すなわち、「ｒａｎｄｏｍ＿ａｃｃｅｓｓ>0」の場合、ランダムアクセスに対応する。特に、「ｒａｎｄｏｍ＿ａｃｃｅｓｓ＝０００　０００１」の場合、これは、ランダムアクセス部を構成するフレーム数が１であることを示す。これは、ランダムアクセスが全てのフレーム部において許容されることを意味する。さらに、「ｒａｎｄｏｍ＿ａｃｃｅｓｓ＝１１１１　１１１１」の場合、これは、ランダムアクセス部を構成するフレーム数が２５５であることを示す。これにより、「ｒａｎｄｏｍ＿ａｃｃｅｓｓ」情報は、現在のランダムアクセス部内のランダムアクセスフレームと次のランダムアクセス部内のランダムアクセスフレームとの距離に相当する。以下、この距離はフレーム数により表現される。

【００４９】

３２ビットの「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」フィールドは、ビットストリームに含まれて送信される。以下、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」フィールドは、現在のランダムアクセスフレームから次のランダムアクセスフレームまでのサイズをバイト単位で表わす。これにより、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」フィールドは、構成シンタックス（テーブル６）に含まれるか、あるいは、フレーム－データシンタックス（テーブル７）に含まれる。構成シンタックス（テーブル６）は、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報のビットストリーム内における保存先を示す情報をさらに含む。この情報は、２ビットの「ｒａ＿ｆｌａｇ」フィールドとして表現される。特に、例えば、「ｒａ＿ｆｌａｇ＝００」の場合、これは、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報がビットストリームに保存されない旨を示す。「ｒａ＿ｆｌａｇ＝０１」の場合、これは、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報がビットストリーム内のフレームデータシンタックス（テーブル７）に保存されない旨を示す。さらに、「ｒａ＿ｆｌａｇ＝１０の場合、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報はビットストリーム内の構成シンタックス（テーブル６）に保存される。「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報が構成シンタックスに含まれる場合、これは、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報が１回のみビットストリーム上において送信され、且つ、全てのランダムアクセス部に同様に適用されることを示す。「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報がフレーム－データシンタックスに含まれる場合、これは、現在のランダムアクセス部内のランダムアクセスフレームと次のランダムアクセス部内のランダムアクセスフレームとの距離を示す。このため、「ｒａ＿ｕｎｉｔ＿ｓｉｚｅ」情報は、ビットストリーム内の各ランダムアクセス部に対して送信される。これにより、構成シンタックス（テーブル６）内の「ｒａｎｄｏｍ＿ａｃｃｅｓｓ」フィールドは、第１の一般情報として言及される。また、「ｒａ＿ｆｌａｇ」フィールドは、第２の一般情報として言及される。本発明のこのような態様において、オーディオ信号は構成情報及び複数のランダムアクセス部を含み、各ランダムアクセス部は１以上のオーディオデータフレームを含み、これらのうちいずれか一つはランダムアクセスフレームであり、構成情報は２つの隣り合うランダムアクセスフレーム間の距離をフレームにより表わす第１の一般情報を含み、第２の一般情報は各ランダムアクセス部に対するランダムアクセス部のサイズ情報が保存されることを示す。ランダムアクセス部の

サイズ情報は、２つの隣り合うランダムアクセスフレーム間の距離をバイトで表わす。

【００５０】

選択的に、本発明のこのような態様において、オーディオ信号をデコーディングする方法は、構成情報及び複数のランダムアクセス部を備えるオーディオ信号を受信するが、各ランダムアクセス部は１以上のオーディオデータフレームを含み、これらのうちいずれかはランダムアクセスフレームであるステップと、２つの隣り合うランダムアクセスフレーム間を距離をフレームで表わす第１の一般情報を構成情報から読み取るステップと、各ランダムアクセス部に対するランダムアクセスサイズ情報が保存される旨を示す第２の一般情報を読み取るステップと、を含み、このようなランダムアクセス部のサイズ情報は、２つの隣り合うランダムアクセスフレーム間の距離をバイトで表わす。

【００５１】

チャンネル構成

図３に示すように、オーディオ信号は、本発明によるマルチチャンネル情報を含む。例えば、各チャンネルは、オーディオスピーカの位置と一対一の対応関係でマッピングされる。構成シンタックス（テーブル６）は、１６ビットの「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールド及び１６ビットの「ｃｈａｎｎｅｌ」フィールドとして表現されるチャンネル構成情報を含む。「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールドは、チャンネルをスピーカ位置にマッピングするための情報を含み、１６ビットの「ｃｈａｎｎｅｌ」フィールドは、チャンネルのトータル数を示す情報を含む。例えば、「ｃｈａｎｎｅｌ」フィールドが「０」と同じ場合、これは、チャンネルがモノチャンネルに相当する旨を示す。「ｃｈａｎｎｅｌ」フィールドが「１」と同じ場合、これは、チャンネルがステレオチャンネルのうちいずれかに相当する旨を示す。「ｃｈａｎｎｅｌ」フィールドが「２」以上と同じ場合、これは、チャンネルがマルチチャンネルのうちいずれかに相当する旨を示す。

【００５２】

テーブル２は「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールドを構成する各ビットとこれに対応する各チャンネルの例を示す。特に、対応するチャンネルが送信されたビットストリーム内に存在する場合、「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールド内の対応するビットは「１」に設定される。代案的に、対応するチャンネルが送信されたビットストリーム内に存在しない場合、「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールド内の対応するビットは「０」に設定される。また、本発明は、「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールドが構成シンタックス（テーブル６）内に存在するかどうかを示す情報を含む。この情報は、１ビットの「ｃｈａｎ＿ｃｏｎｆｉｇ」フラグとして表現される。特に、「ｃｈａｎ＿ｃｏｎｆｉｇ＝０」は、「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールドが存在しない旨を示す。「ｃｈａｎ＿ｃｏｎｆｉｇ＝１」は、「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールドが存在する旨を示す。このため、「ｃｈａｎ＿ｃｏｎｆｉｇ＝０」の場合、これは、「ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ」フィールドが構成シンタックス（テーブル６）内において新たに定義される旨を示す。

【００５３】

【表２】

テーブル２：チャンネル構成

スピーカ位置	略字	ｃｈａｎ＿ｃｏｎｆｉｇ＿ｉｎｆｏ内におけるビットの位置
左	Ｌ	１
右	Ｒ	２
左後方	Ｌｒ	３
右後方	Ｒｒ	４
左側	Ｌｓ	５
右側	Ｒｓ	６
中心	Ｃ	７
中心後方／サラウンド	Ｓ	８
低周波数効果	ＬＦＥ	９
左ダウンミックス	Ｌ０	１０
右ダウンミックス	Ｒ０	１１
モノダウンミックス	Ｍ	１２
（予備）		１３－１６

【００５４】

フレーム長

図３に示すように、オーディオ信号は、本発明により多数または多重のチャンネルを含む。このため、エンコーディングを行うとき、１フレームを構成するマルチチャンネルの数に関する情報及び各チャンネルに対するサンプルの数に関する情報はビットストリーム内に挿入されて送信される。構成シンタックス（テーブル６）を参照すれば、３２ビットの「ｓａｍｐｌｅ」フィールドは、各チャンネルを構成するオーディオデータサンプルのトータル数を示す情報として用いられる。さらに、１６ビットの「ｆｒａｍｅ＿ｌｅｎｇｔｈ」フィールドは、対応するフレーム内の各チャンネルに対するサンプルの数を示す情報として用いられる。

【００５５】

さらに、「ｆｒａｍｅ＿ｌｅｎｇｔｈ」フィールドの１６ビット値はエンコーダにより用いられる値により決まり、ユーザ定義値と呼ばれる。すなわち、固定値の代わりに、ユーザ定義値はエンコーディングのプロセス時に任意に決まる。

【００５６】

このため、デコーディングプロセス中にビットストリームが図２に示すデマルチプレックス部２００を介して受信されるとき、各チャンネルのフレーム数が先行して取得されることが必要である。この値は、後述するアルゴリズムにより取得される。

【００５７】

```
frames = samples / frame_length;
 rest = samples % frame_length;
 if (rest)
  {
        frames++;
        frlen_last = rest;
 }
 else
        frlen_last = frame_length;
```

【００５８】

特に、各チャンネルに対するフレームのトータル数は、ビットストリームを介して送信された「ｓａｍｐｌｅ」フィールドにより決まる各チャンネルに対するサンプルのトータル数を、「ｆｒａｍｅ＿ｌｅｎｇｔｈ」フィールドにより決まる各チャンネルに対するフレーム内のサンプル数で割ることで演算される。例えば、「ｓａｍｐｌｅ」フィールドにより決まったサンプルのトータル数が、「ｆｒａｍｅ＿ｌｅｎｇｔｈ」フィールドにより決まる各フレーム内のサンプル数のちょうど倍数である場合、多数の値がフレームのトータル数となる。しかしながら、「ｓａｍｐｌｅ」フィールドにより決まったサンプルのトータル数が、「ｆｒａｍｅ＿ｌｅｎｇｔｈ」フィールドにより決まったサンプル数のちょうど倍数ではない場合、且つ、余りが存在する場合、フレームのトータル数は多数値以上に「１」ずつインクリメントする。さらに、最終フレームｆｒｌｅｎ＿ｌａｓｔのサンプル数は余りとして決まる。これは、最終フレームのサンプル数だけがそれ以前のフレームとは異なるという旨を示す。上述したように、エンコーダとデコーダとの間の標準化された規則を定義することにより、エンコーダは、各チャンネルに対するサンプルのトータル数（「ｓａｍｐｌｅ」フィールド）及び各チャンネルのフレーム内のサンプル数（「ｆｒａｍｅ＿ｌｅｎｇｔｈ」フィールド）を自由に決めて送信することができる。さらに、デコーダは、送信された情報に対する上述したアルゴリズムを用いて、デコーディングに用いられる各チャンネルに対するフレーム数を正確に決めることができる。

【００５９】

線形予測

本発明において、線形予測は無損失オーディオコーディングに適用される。図１に示す予測器１６０は少なくとも１以上のフィルタ係数を含み、以前のサンプル値から現在のサンプル値を予測する。この後、第２のエントロピー符号化部１８０は、予測値とオリジナル値との間の差分に対応する余り値に対するエントロピー符号化を行う。また、予測器１６０に適用される各ブロックに対する予測係数値は、係数推定部１２０から最適値として選択される。さらに、予測係数値は、第１のエントロピー符号化部１４０によりエントロピー符号化される。第１のエントロピー符号化部及び第２のエントロピー符号化部１８０によりコーディングされたデータは、マルチプレックス部１９０によりビットストリームの一部として挿入されてから送信される。

【００６０】

以下、本発明により線形予測を行う方法について説明する。

【００６１】

ＦＩＲフィルタによる予測

線形予測は、音声及びオーディオ信号の処理のための多数のアプリケーションに用いられる。以下、予測器１６０の例示的な動作を有限インパルス応答（ＦＩＲ：Ｆｉｎｉｔｅ Ｉｍｐｕｌｓｅ Ｒｅｓｐｏｎｓｅ）フィルタを元に説明する。しかしながら、この例が本発明の範囲を制限しないことは言うまでもない。

【００６２】

時間離散信号ｘ（ｎ）は、以前のサンプルｘ（ｎ－ｋ）から大まかに予測可能である。予測は。下記式として与えられる。

【００６３】

【数１】

$$\hat{x}(n) = \sum_{k=1}^{K} h_k * x(n-k),$$

【００６４】

Ｋは予測の順序である。予測サンプルが原サンプルに近づく場合、余りは以下の通りである：

【００６５】

【数２】

$$e(n) = x(n) - \hat{x}(n)$$

【００６６】
これは、ｘ（ｎ）そのものよりも小さなバラツキを有し、その結果、ｅ（ｎ）はさらに効率よくエンコーディングされる。
【００６７】
入力サンプルのセグメントをフィルタリングする前にこのようなセグメントから予測係数を推定する手続きは、フォワード適応として言及される。この場合、係数は送信される必要がある。一方、係数が以前に処理されたセグメントまたはサンプル、例えば、余りから推定される場合、バックワード適応が参考として行われる。バックワード適応の手続きは、係数を推定するのに必要なデータがデコーダにも利用可能であるため、いかなる係数送信も不要であるという利点を有する。
【００６８】
約１０の順序を有するフォワード適応型の予測方法は音声コーディングに汎用され、無損失オーディオコーディングにも採用可能である。ほとんどのフォワード適応型の無損失予測スキームの最大順序は、例えば、Ｋ＝３２のように遥かに小さい。例外として、スーパーオーディオＣＤに対する１ビット無損失コーデックがあり、これは、１２８までの予測順序を用いる。
【００６９】
一方、数百個の係数を有するバックワード適応型のＦＩＲフィルタは、多くの領域、例えば、チャンネル等化及びエコー削除に一般的に用いられる。ほとんどのこれらのシステムは、また、無損失オーディオコーディングに提案されるＬＭＳアルゴリズムまたはその変形に基づく。高次順を有するこのようなＬＭＳ基盤のコーディングスキームは、予測係数がサイド情報として送信される必要がないために作用可能であり、その数はデータレートに寄与しない。しかしながら、バックワード適応型のコーデックは、適応がエンコーダとデコーダの両方に行われる必要があるという欠点を有するため、デコーダがフォワード適応型の場合よりも複雑になる。
【００７０】
フォワード適応型予測
以下、本発明の例示的な実施形態として、フォワード適応型予測について説明する。フォワード適応型線形予測において、最適な予測係数h_k（余りの最小化されたバラツキの項）は、通常、自動相関方法または共分散を用いて係数推定部１２０により各ブロックに対して推定される。従来のレビンソンーダービンアルゴリズムを用いる自動相関方法は、予測器順序を繰り返し適応するための簡単な手段を提供することができるというさらなる利点を有する。さらに、このアルゴリズムは、対応するパーコール係数も固有に計算する。
【００７１】
フォワード適応型予測の他の態様は、適切な予測順序を決めることである。順序が増大するに伴い、予測エラーのバラツキは低減され、余りに対する小さなビットレートR_eをもたらす。一方、予測係数に対するビットレートR_cは、送信される係数の数を増やす。このため、全体のビットレートを最小化する最適な順序を見つけることが課題となる。これは、予測係数Ｋに関する下記式を最小化して表現することができる：
【００７２】
$R_{total}(K) = R_e(K) + R_c(K)$
【００７３】
予測が高次順になるにつれて単調増加することにより、R_eはＫに減少される。一方、

R_cは、係数の増加数が送信される必要があるため、Ｋで単調増加する。
【００７４】
【数３】

最適な順序の探索は、増加する順序で全ての予測器を繰り返し決める係数推定部１２０により効率よく行われうる。それぞれの順序において、予測係数の完全な組が演算される。さらに、対応する余りのバラツキσ_e^2が導出可能であり、余りに対する予測されたビットレートの推定を招く。係数に対するビットレートと共に、全体のビットレートは、それぞれの繰り返し、すなわち、各予測順序に対して決まりうる。最適な順序は、全体のビットレートがそれ以上減少されない個所において見つけられる。

【００７５】
たとえ、係数ビットレートが全体のビットレートに直接的な効果を有することが上述した数式から明らかであるとしても、また、R_cの遅い増加は、R_{total}の最小が高次順にシフトされるように許容し（ここで、R_eは、もちろん小さい）、良好な圧縮をもたらす。その結果、予測係数の効率的で且つ正確な等化は、最大圧縮を達成する上で重要な役割を果たす。
【００７６】
予測順序
本発明において、線形予測に対する予測係数の数を決める予測順序Ｋが決まる。また、予測順序Ｋは、係数推定部１２０により決まる。以下、決まった予測順序に関する情報は、ビットストリームに含まれてから送信される。
【００７７】
係数シンタックス（テーブル６）は、予測順序Ｋに関する情報を含む。例えば、１ビットから１０ビットの「ｍａｘ＿ｏｒｄｅｒ」フィールドは、最大の順序値を示す情報に対応する。１ビットから１０ビットの「ｍａｘ＿ｏｒｄｅｒ」フィールドの最高次値は、Ｋ＝１０２３（例えば、１０ビット）である。予測係数Ｋに関する他の情報として、構成シンタックス（テーブル６）は、各ブロックに対する最適な順序が存在するかどうかを示す１ビットの「ａｄａｐｔ＿ｏｒｄｅｒ」フィールドを含む。例えば、「ａｄａｐｔ＿ｏｒｄｅｒ＝１」の場合、最適な順序は、各ブロックに対して与えられる必要がある。ブロック＿データシンタックス（テーブル８）において、最適な順序は１ビットから１０ビットの「ｏｐｔ＿ｏｒｄｅｒ」フィールドとして与えられる。さらに、「ａｄａｐｔ＿ｏｒｄｅｒ＝０」の場合、個別の最適な順序は各ブロックに対して与えられない。この場合、「ｍａｘ＿ｏｒｄｅｒ」フィールドは、全てのブロックに適用された最終順序となる。
【００７８】
最適な順序ｏｐｔ＿ｏｒｄｅｒは、ｍａｘ＿ｏｒｄｅｒフィールドの値と対応するブロックのサイズN_Bに基づいて決まる。特に、例えば、ｍａｘ＿ｏｒｄｅｒがK_{max}＝１０として決まり、且つ、「ａｐａｐｔ＿ｏｒｄｅｒ＝１」の場合、各ブロックに対するｏｐｔ＿ｏｒｄｅｒが対応するブロックのサイズを考慮して決まる。一部の場合に、ｍａｘ＿ｏｒｄｅｒ（K_{max}＝１０）よりも大きなｏｐｔ＿ｏｒｄｅｒ値が可能である。
【００７９】
特に、本発明は、高次予測順序に関するものである。階層ブロックスイッチングが存在しないとき、この実施形態によれば、長いブロック長と短いブロック長（例えば、４０９６及び１０２４、または８１９２及び２０４８）との間には、４の要因が存在する。一方、階層ブロックスイッチングが実現される実施形態においては、このような要因は、（例

えば、３２まで）増加可能であり、より大きな範囲（例えば、高いサンプリングレートに対して１６３８４から５１２へと減少、また、３２７６８から１０２４へと減少）を可能にする。

【００８０】

階層ブロックスイッチングが実現される実施形態において、極めて長いブロックを上手に用いるためには、高い最大予測順序が採用可能である。最大順序は、K_{max}＝１０２３でありうる。この実施形態において、K_{max}は、ブロック長N_Bに限られる。例えば、$K_{max}<N_B/8$である（例えば、N_B＝２０４８に対してK_{max}＝２５５である）。このため、K_{max}＝１０２３を用いる場合、少なくともN_B＝８１９２のブロック長を必要とする。この実施形態において、構成シンタックス（テーブル６）内の「ｍａｘ＿ｏｒｄｅｒ」フィールドは１０ビットまでであり得、また、ブロックデータシンタックス（テーブル８）内の「ｏｐｔ＿ｏｒｄｅｒ」フィールドは１０ビットまででありうる。特定のブロックのビットの実際数は、１ブロックに対して許容された最大順序に従属する。ブロックが短ければ、ローカル予測順序はグローバル予測順序よりも小さくなる。以下、ローカル予測順序は対応するブロック長N_Bを考慮して決まり、グローバル予測順序は構成シンタックスにおいて「ｍａｘ＿ｏｒｄｅｒ」K_{max}から決まる。例えば、K_{max}＝１０２３、または、N_B＝２０４８の場合、「ｏｐｔ＿ｏｒｄｅｒ」フィールドは２５５のローカル予測順序に起因して（１０ビットの代わりに）８ビットとして決まる。

【００８１】

特に、ｏｐｔ＿ｏｒｄｅｒは、以下の式に基づいて決まる。

【００８２】

ｏｐｔ＿ｏｒｄｅｒ＝ｍｉｎ（グローバル予測順序、ローカル予測順序）

【００８３】

グローバル及びローカル予測順序は、

グローバル予測順序＝ｃｅｉｌ（ｌｏｇ２（最大予測順序＋１））、及び

ローカル予測順序＝ｍａｘ（ｃｅｉｌ（ｌｏｇ２（ｎｂ>>３）－１））、１）

として決まる。

【００８４】

実施形態において、チャンネルからサブ分割されたブロックのデータサンプルは予測される。現在のブロックの第１のサンプルは、以前のブロックの最終的なＫサンプルを用いて予測される。Ｋ値は、上述した式から導き出されたｏｐｔ＿ｏｒｄｅｒから決まる。

【００８５】

現在のブロックがチャンネルの第１のブロックである場合、以前のブロックからいかなるサンプルも用いられない。この場合、漸進的な順序を有する予測が採用される。例えば、ｏｐｔ＿ｏｒｄｅｒ値が対応するブロックに対してＫ＝５であるとすれば、ブロック内の第１のサンプルは予測を行うことができない。ブロックの第２のサンプルは、（Ｋ＝１と同様に）予測を行うためにブロックの第１のサンプルを用い、ブロックの第３のサンプルは、（Ｋ＝２と同様に）予測を行うためにブロックの第１及び第２のサンプルを用いる。このため、６番目のサンプルから始まってそれ以降のサンプルに対して、予測がＫ＝５のｏｐｔ＿ｏｒｄｅｒに応じて行われる。上述したように、予測順序は、Ｋ＝１からＫ＝５に漸増する。

【００８６】

上述した予測の漸進的な順序形態は、ランダムアクセスフレームにおいて用いられるときに極めて有利になる。ランダムアクセスフレームがランダムアクセス部の基準フレームに対応するため、ランダムアクセスフレームは、以前のフレームサンプルを用いて予測を行うことができない。すなわち、このような漸進的な予測技術は、ランダムアクセスフレームの開始時に適用可能である。

【００８７】

予測係数の量子化

上述した予測係数は、図１の量子化部１３０において量子化される。予測係数h_kの直

接的な量子化は、小さな量子化エラーが最適な予測フィルタの所定のスペクトル特徴から大きなバラツキをもたらすため、送信に極めて非効率的である。この理由から、予測係数の量子化は、係数推定部１２０により演算可能なパーコール（反射）係数r_kに基づく。上述したように、例えば、係数推定部１２０は、従来のレビンソン－ダービンアルゴリズムを用いて処理される。

【００８８】

第１の２つのパーコール係数（γ_1及びγ_2に対応する）は、下記式を用いて量子化される：

【００８９】

【数４】

$$a_1 = \left\lfloor 64\left(-1+\sqrt{2}\sqrt{\gamma_1+1}\right)\right\rfloor;$$

$$a_2 = \left\lfloor 64\left(-1+\sqrt{2}\sqrt{-\gamma_2+1}\right)\right\rfloor;$$

【００９０】

これに対し、余り係数は単純な７ビットの均一量子化器を用いて量子化される：

【００９１】

【数５】

$$a_k = \lfloor 64\,\gamma_k \rfloor; \qquad (k>2).$$

【００９２】

あらゆる場合に、最終的に量子化された値a_kは、範囲［－６４、６３］に限られる。

【００９３】

エントロピー符号化

図１に示すように、２通りの方式のエントロピー符号化が本発明に適用される。特に、第１のエントロピー符号化部１４０は、上述した予測係数をコーディングするのに用いられる。また、第２のエントロピー符号化部１８０は、上述したオーディオ原サンプル及びオーディオ余りサンプルをコーディングするのに用いられる。以下、２通りの方式のエントロピー符号化について説明する。

【００９４】

予測係数の第１のエントロピー符号化

従来のライスコードは、本発明による第１のエントロピー符号化方法として用いられる。例えば、量子化された係数a_kの送信は、余り値を生成して行われる：

【００９５】

$\delta_k = a_k - offset_k$

【００９６】

これは、例えば、ライスコード方法と同じ第１のエントロピー符号化部１４０を用いてエンコーディングされる。このようなプロセスにおいて用いられるライスコードの対応するオフセット及びパラメータは、テーブル３、４、５に示す組のうちいずれかからグローバル的に選択される。テーブルシンタックス（すなわち、２ビットの「ｃｏｅｆ＿ｔａｂｌｅ」）は、構成シンタックス（テーブル６）に現れる。「ｃｏｅｆ＿ｔａｂｌｅ＝１１」の場合、これは、いかなるエントロピー符号化も適用されず、量子化された係数は７ビットのそれぞれに送信されることを示す。この場合、オフセットは［０、２７］に限られる非署名値$\delta_k = a_k + 64$を得るために、常に－６４である。逆に、「ｃｏｅｆｆ＿ｔａ

ｂｌｅ＝００」の場合にテーブル３が選択され、「ｃｏｅｆｆ＿ｔａｂｌｅ＝０１」の場合にテーブル４が選択され、「ｃｏｅｆｆ＿ｔａｂｌｅ＝１１」の場合にテーブル５が選択される。

【００９７】

図２のデコーダにおいて量子化された係数を受信するとき、第１のエントロピー復号化部２２０は、余り値δ_kがパーコール係数a_kの量子化されたインデックスを生成するためにオフセットと組み合せられるプロセスを用いて予測係数を再構成する：

【００９８】

$a_k = \delta_k + offset_k$

【００９９】

この後、第１の２つの係数（γ_1及びγ_2）の再構成は、

【０１００】

【数６】

$$\mathrm{par}_1 = \left\lfloor \hat{\gamma}_1 2^Q \right\rfloor = \Gamma(a_1);$$
$$\mathrm{par}_2 = \left\lfloor \hat{\gamma}_2 2^Q \right\rfloor = -\Gamma(a_2);$$

を用いて行われる。

【０１０１】

【数７】

ここで、2^Qは、再構成された係数の整数表現に求められる一定の（$Q = 20$）スケール因子を示し、$\Gamma(.)$は、先験的に決まったマッピングテーブル（マッピングテーブルは実現により変わるために図示しない）である。

【０１０２】

このため、第１のエントロピー符号化に用いられるこれらのタイプの係数テーブルは、サンプリング周波数に応じて与えられる。例えば、サンプリング周波数は、４８ｋＨｚ、９６ｋＨｚ、１９２ｋＨｚに分割可能である。３つのテーブル３、４、５のそれぞれは、各サンプリング周波数に対して与えられる。

【０１０３】

単一テーブルを用いる代わりに、３つの異なるテーブルのうちいずれかが全体ファイルのために選択可能である。テーブルは、サンプリングレートに応じて典型的に選択されなければならない。４４．１ｋＨｚの材料において、本発明の出願人は、４８ｋＨｚテーブルを用いることを推奨している。しかしながら、一般的に、テーブルは、他の基準により選択可能である。

【０１０４】

【表３】

テーブル３：量子化された係数のエンコーディングに用いられるライスコードパラメータ（４８ｋＨｚ）

係数＃	オフセット	ライスパラメータ
１	－５２	４
２	－２９	５
３	－３１	４
４	１９	４
５	－１６	４
６	１２	３
７	－７	３
８	９	３
９	－５	３
１０	６	３
１１	－４	３
１２	３	３
１３	－３	２
１４	３	２
１５	－２	２
１６	３	２
１７	－１	２
１８	２	２
１９	－１	２
２０	２	２
２ｋ－１、ｋ＞１０	０	２
２ｋ、ｋ＞１０	１	２

【０１０５】

【表４】

テーブル４：量子化された係数のエンコーディング時に用いられるライスコードパラメータ（９６ｋＨｚ）

係数＃	オフセット	ライスパラメータ
１	－５８	３
２	－４２	４
３	－４６	４
４	３７	５
５	－３６	４
６	２９	４
７	－２９	４
８	２５	４
９	－２３	４
１０	２０	４
１１	－１７	４
１２	１６	４
１３	－１２	４
１４	１２	３
１５	－１０	４
１６	７	３
１７	－４	４
１８	３	３
１９	－１	３
２０	１	３
２ｋ－１、ｋ＞１０	０	２
２ｋ、ｋ＞１０	１	２

【０１０６】

【表５】

テーブル５：量子化された係数のエンコーディングに用いられるライスコードパラメータ１９２ｋＨｚ）

係数＃	オフセット	ライスパラメータ
１	－５９	３
２	－４５	５
３	－５０	４
４	３８	４
５	－３９	４
６	３２	４
７	－３０	４
８	２５	３
９	－２３	３
１０	２０	３
１１	－２０	３
１２	１６	３
１３	－１３	３
１４	１０	３
１５	－７	３
１６	３	３
１７	０	３
１８	－１	３
１９	２	３
２０	－１	２
２ｋ－１、ｋ＞１０	０	２
２ｋ、ｋ＞１０	１	２

【０１０７】

余りの第２のエントロピー符号化

本発明は、後述するように、図１の第２のエントロピー符号化部１８０に適用されるコーディング方法の２つの異なるモードを含む。

【０１０８】

単純なモードにおいて、余り値ｅ（ｎ）は、ライスコードを用いてエントロピー符号化される。各ブロックにおいて、全ての値が同じライスコードを用いてコーディング可能であるか、あるいは、ブロックが４つの部分にさらに分割可能であり、これらのそれぞれは異なるライスコードによりエンコーディングされる。適用されたコーディングのインデックスは、図１に示すように送信される。所定のデータ組に対する最適なライスコードを決めるための異なる方式が存在するため、余りの統計により適切なコードをエンコーダが選択しなければならない。

【０１０９】

代案的に、エンコーダは、ＢＧＭＣモードを用いてより複雑で且つ効率よいコーディングスキームを用いることができる。ＢＧＭＣモードにおいて、余りのエンコーディングは、２つのカテゴリに分配を分割して達成される。２通りの方式は、分布の中心領域、｜ｅ（ｎ）｜＜e_{max}に属する余りとその尻尾に属する余りを含む。尻尾内の余りは簡単に再中心合わせが行われ（すなわち、ｅ（ｎ）＞e_{max}に対して、e_t（ｎ）＝ｅ（ｎ）－e_{max}が与えられる）、上述したようにライスコードを用いてエンコーディングされる。しか

しながら、分布の中心において余りをエンコーディングするためには、ＢＧＭＣは、先ず、ＬＳＢ及びＭＳＢ成分に余りを分割した後、ＢＧＭＣは、ブロックギルバート－ムーア（算術）コードを用いてＭＳＢをエンコーディングする。結局として、ＢＧＭＣは、直接的な固定長コードを用いてＬＳＢを送信する。パラメータe_{max}及び直接的に送信されたＬＳＢの数の全ては、やや複雑ではないコーディングを許容しながら、このようなスキームのコーディング効率性にのみ影響するように選択される。

【０１１０】

本発明による構成シンタックス（テーブル６）及びブロック＿データシンタックス（テーブル８）は、ライスコード及びＢＧＭＣコードのコーディングと関連する情報を含む。情報は新たに詳述される。

【０１１１】

構成シンタックス（テーブル６）は、先ず、１ビットの「ｂｇｍｃ＿ｍｏｄｅ」フィールドを含む。例えば、「ｂｇｍｃ＿ｍｏｄｅ＝０」はライスコードを示し、「ｂｇｍｃ＿ｍｏｄｅ＝１」はＢＧＭＣコードを示す。構成シンタックス（テーブル６）は、１ビットの「ｓｂ＿ｐａｒｔ」フィールドを含む。「ｓｂ＿ｐａｒｔ」フィールドは、１ブロックをサブブロックに区画し、区画されたサブブロックをコーディングする方法と関連する情報に相当する。以下、「ｓｂ＿ｐａｒｔ」フィールドの意味は、「ｂｇｍｃ＿ｍｏｄｅ」フィールドの値に応じて変わる。

【０１１２】

例えば、「ｂｇｍｃ＿ｍｏｄｅ＝０」の場合、すなわち、ライスコードが適用される場合、「ｓｂ＿ｐａｒｔ＝０」は、ブロックがサブブロックに区画されない旨を示す。代案的に、「ｓｂ＿ｐａｒｔ＝１」は、ブロックが１：４サブブロック区画比にて区画される旨を示す。また、「ｂｇｍｃ＿ｍｏｄｅ＝１」の場合、すなわち、ＢＧＭＣコードが適用される場合、「ｓｂ＿ｐａｒｔ＝０」は、１：４サブブロック区画比にてブロックが区画される旨を示す。代案的に、「ｓｂ＿ｐａｒｔ＝１」は、１：２：４：８サブブロック区画比にてブロックが区画される旨を示す。

【０１１３】

構成シンタックス（テーブル６）に含まれている情報に対応する各ブロックに対するブロック＿データシンタックス（テーブル８）は、０ビットから２ビットの可変「ｅｃ＿ｓｕｂ」フィールドを含む。特に、「ｅｃ＿ｓｕｂ」フィールドは、実際に対応するブロックに存在するサブブロックの数を示す。以下、「ｅｃ＿ｓｕｂ」フィールドの意味は、構成シンタックス（テーブル６）内の「ｂｇｍｃ＿ｍｏｄｅ」＋「ｓｂ＿ｐａｒｔ」フィールドの値に応じて変わる。

【０１１４】

例えば、「ｂｇｍｃ＿ｍｏｄｅ＋ｓｂ＿ｐａｒｔ＝０」は、ライスコードがサブブロックを構成しない旨を示す。以下、「ｅｃ＿ｓｕｂ」フィールドは、いかなる情報も含まれていない旨を示す０ビットフィールドである。

【０１１５】

また、「ｂｇｍｃ＿ｍｏｄｅ＋ｓｂ＿ｐａｒｔ＝１」は、ライスコードまたはＢＧＭＣコードがブロックを１：４レートにてサブブロックに区画するのに用いられる旨を示す。以下、１ビットだけが「ｅｃ＿ｓｕｂ」フィールドに割り当てられる。例えば、「ｅｃ＿ｓｕｂ＝０」は１サブブロック（すなわち、ブロックはサブブロックに区画されない）、「ｅｃ＿ｓｕｂ＝１」は４つのサブブロックが構成される旨を示す。

【０１１６】

さらに、「ｂｇｍｃ＿ｍｏｄｅ＋ｓｂ＿ｐａｒｔ＝２」は、ＢＧＭＣコードがブロックを１：２：４：８レートにてサブブロックに区画するのに用いられる旨を示す。以下、２ビットは「ｅｃ＿ｓｕｂ」フィールドに割り当てられる。例えば、「ｅｃ＿ｓｕｂ＝００」は１サブブロック（すなわち、ブロックはサブブロックに区画されない）を示し、「ｅｃ＿ｓｕｂ＝０１」は２サブブロックを示す。また、「ｅｃ＿ｓｕｂ＝１０」は４個のサブブロックを、「ｅｃ＿ｓｕｂ＝１１」は８個のサブブロックを示す。

【０１１７】
上述したように、各ブロック内に定義されたサブブロックは、他のコーディング方法を用いて第２のエントロピー符号化部１８０によりコーディングされる。以下、ライスコードを用いる例について説明する。余り値の各ブロックにおいては、全ての値が同じライスコードを用いてエンコーディングされるか、または、構成シンタックス内の「ｓｂ＿ｐａｒｔ」フィールドが設定される場合にブロックは４個のサブブロックに区画可能であり、これらのそれぞれのエンコーディングされたサブブロックは、異なるのライスコードを有する。後者の場合、ブロック＿データシンタックス（テーブル８）内の「ｅｃ＿ｓｕｂ」フィールドは、１または４個のブロックが用いられるかどうかを示す。
【０１１８】
第１のサブブロックのパラメータｓ［ｉ＝１］が４ビット（解像度≦１６ビット）または５ビット（解像度＞１６ビット）で直接的に送信されるとしても、次のパラメータｓ［ｉ＞０］の差分（ｓ［ｉ］－ｓ［ｉ－１］）が送信される。これらの差分は適切に選択されたライスコードを再度用いてさらにエンコーディングされる。この場合、差分に用いられるライスコードパラメータは「０」の値を有する。
【０１１９】
シンタックス
本発明の実施形態によれば、オーディオビットストリームに含まれている各種の情報のシンタックスは、下記のテーブルに示す。テーブル６は、オーディオ無損失コーディング用の構成シンタックスを示す。構成シンタックスはビットストリームに周期的に配置されたヘッダーを形成し、各フレームに対するヘッダーを形成する。テーブル７はフレーム－データシンタックスを示し、テーブル８はブロック－データシンタックスを示す。
【０１２０】

【表６－１】

テーブル６：構成シンタックス

シンタックス	ビット
ALSSpecificConfig()	
{	
samp_freq;	32
samples;	32
channels;	16
file_type;	3
resolution;	3
floating;	1
msb_first;	1
frame_length;	16
random_access;	8
ra_flag;	2
adapt_order;	1
coef_table;	2
long_term_prediction;	1
max_order;	10
block_switching;	2
bgmc_mode;	1
sb_part;	1
joint_stereo;	1
mc_coding;	1
chan_config;	1
chan_sort;	1
crc_enabled;	1

【表６－２】

RLSLMS	1
(reserved)	6
if (chan_config) {	
chan_config_info;	16
}	
if (chan_sort) {	
for (c = 0; c < channels; c++)	
chan_pos[c];	8
}	
header_size;	16
trailer_size;	16
orig_header[];	header_size * 8
orig_trailer[];	trailer_size * 8
if (crc_enabled) {	
crc;	32
}	
if ((ra_flag == 2) && (random_access > 0)) {	
for (f = 0; f < (samples - 1 / frame_length) + 1; f++) {	
ra_unit_size	32
}	
}	
}	

【０１２１】

【表７－１】

テーブル７：フレーム__データシンタックス

シンタックス	ビット

frame_data()
{
if ((ra_flag == 1) && (frame_id % random_access == 0)
) {
ra_unit_size
	32

}
if (mc_coding && joint_stereo) {
js_switch;
	1

byte_align;
}
if (!mc_coding || js_switch) {
for (c = 0; c < channels; c++) {
if (block_switching) {
bs_info;
	8, 16, 32

}
if (independent_bs) {
for (b = 0; b < blocks; b++) {
block_data(c);
}	

}
else{
for (b = 0; b < blocks; b++) {
block_data(c);
block_data(c+1);
}	

c++;
}
}	

【表７－２】

else{	
if (block_switching) {	
bs_info;	8, 16, 32
}	
for (b = 0; b < blocks; b++) {	
for (c = 0; c < channels; c++) {	
block_data(c);	
channel_data(c);	
}	
}	
}	
if (floating)	
{	
num_bytes_diff_float;	32
diff_float_data();	
}	
}	

【０１２２】

【表８－１】

テーブル８：ブロック_データシンタックス

シンタックス	ビット

block_data()
{
　　block_type;
　　if (block_type == 0) {
　　　　const_block;
　　　　js_block;
　　　　(reserved)
　　　　if (const_block == 1) {
　　　　{
　　　　　　if (resolution == 8) {
　　　　　　　　const_val;
　　　　　　}
　　　　　　else if (resolution == 16) {
　　　　　　　　const_val;
　　　　　　}
　　　　　　else if (resolution == 24) {
　　　　　　　　const_val;
　　　　　　}
　　　　　　else {
　　　　　　　　const_val;
　　　　　　}
　　　　}
　　}
	
	1
	
	1
	1
	5
	
	8
	
	16
	
	24
	
	32

【表８－２】

else {	
js_block;	1
if ((bgmc_mode == 0) && (sb_part == 0) {	
sub_blocks = 1;	
}	
else if ((bgmc_mode == 1) && (sb_part ==1) {	
ec_sub;	2
sub_blocks = 1 << ec_sub;	
}	
else {	
ec_sub;	1
sub_blocks = (ec_sub == 1) ? 4 : 1;	
}	
if (bgmc_mode == 0) {	
for (k = 0; k < sub_blocks; k++) {	
s[k];	varies
}	
}	
else {	
for (k = 0; k < sub_blocks; k++) {	
s[k], sx[k];	varies
}	
}	
sb_length = block_length / sub_blocks;	
shift_lsbs;	1
if (shift_lsbs == 1) {	
shift_pos;	4
}	
if (!RLSLMS) {	
if (adapt_order == 1) {	
opt_order;	1…10
}	
for (p = 0; p < opt_order; p++) {	
quant_cof[p];	varies
}	
}	

【０１２３】

圧縮結果

次には、無損失オーディオコーデックが無損失オーディオ圧縮用の最も大衆的なプログラムのうち２つ、すなわち、オープンソースコーデックＦＬＡＣ及びＭｏｎｋｅｙ’ｓ Ａｕｄｉｏ（ＭＡＣ３．９７）と比較される。以下、オープンソースコーデックＦＬＡＣはフォワード適応型予測を用い、Ｍｏｎｋｅｙ’ｓ Ａｕｄｉｏ（ＭＡＣ３．９７）は圧縮において現在のステート・オフ・ザー・アート・アルゴリズムとして用いられるバックワード適応型コーデックである。これらのコーデックは、最大圧縮（すなわち、フラック－８及びｍａｃ－ｃ４００）を与えるオプションで行われる。エンコーダに対する結果は（Ｋ－６０に限られる予測順序を有する）中間圧縮レベルと、最大圧縮レベル（Ｋ－１０

２３）と、５００ｍｓのランダムアクセスを有するこれらの両レベルに対して決まる。テストは、１０２４ＭＢのメモリを有する１．７ＧＨｚペンティアム（登録商標）－Ｍシステム上において行われる。テストは、４８、９６及び１９２ｋＨｚのサンプリングレートを有する１Ｇのステレオ波形データと、１６及び２４ビットの解像度を含む。

【０１２４】

圧縮レート

次には、圧縮レートが

【数８】

$$C = \frac{CompressedFileSize}{OriginalFileSize} * 100\%$$

、

により定義される。ここで、小さな値の方が良好な圧縮を示す。検査されたオーディオフォーマットに対する結果はテーブル９に示す（１９２ｋＨｚ材料がＦＬＡＣコーデックにより支援される）。

【０１２５】

【表９】

テーブル９：異なるオーディオフォーマットに対する平均圧縮比の比較（ｋＨｚ／ビット）

フォーマット	ＦＬＡＣ	ＭＡＣ	ＡＬＳ媒体	ＡＬＳ媒体
４８／１６	４８．６	４５．３	４５．５	４４．７
４８／２４	６８．４	６３．２	６３．３	６２．７
９６／２４	５６．７	４８．１	４６．５	４６．２
１９２／２４	－	３９．１	３７．７	３７．６
Ｔｏｔａｌ	－	４８．９	４８．３	４７．８

【０１２６】

この結果は、最大レベルにおいてＡＬＳが全てのフォーマット、特に、高品位材料（すなわち、９６ｋＨｚ／２４ビット以上）用のＦＬＡＣ及びＭｏｎｋｅｙ’ｓ Ａｕｄｉｏ（ＭＡＣ３．９７）の両方よりも低くなることを示す。中間レベルにおいてさえ、ＡＬＳは最上の全体圧縮を伝える。

【０１２７】

複雑性

異なるコーデックの複雑性は、実際の実現、特に、エンコーダの実現に極めて強く従属する。上述したように、本発明のオーディオ信号エンコーダはますます開発中である。このため、出願人は、その分析をエンコーダに制限して、さらなる最適化無しに単純なＣコードを実現する。圧縮されたデータは、現在、最上のエンコーダの実現により生成される。異なる複雑度レベルでエンコーディングされる各種のオーディオフォーマットのリアルタイムなデコーディングに対する平均ＣＰＵ負荷は、テーブル１０に示す。最大複雑度においてさえ、デコーダに対するＣＰＵ負荷は約２０～２５％だけであり、ファイル基盤のデコーディングの方が少なくともリアルタイムよりも４－５倍速いことを意味することになる。

【０１２８】

【表１０】

テーブル１０：オーディオフォーマット（ｋＨｚ／ビット）及びＡＬＳエンコーダ複雑度に従属する平均ＣＰＵ負荷（１．７ＧＨｚペンティアム－Ｍ上のパーセント）

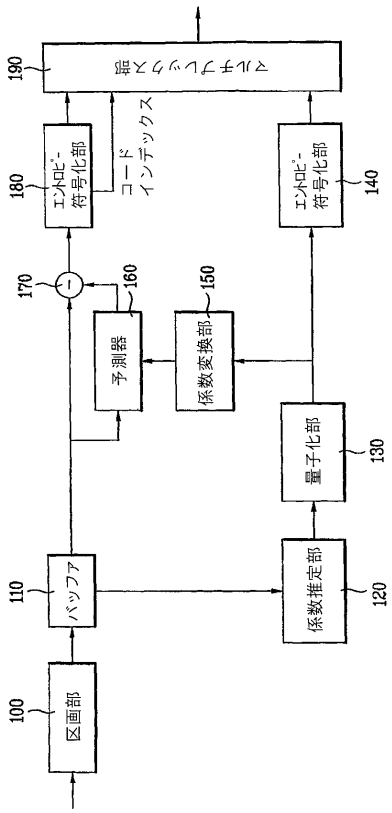
フォーマット	低ＡＬＳ	中間ＡＬＳ	最大ＡＬＳ
４８／１６	１．６	４．９	１８．７
４８／２４	１．８	５．８	１９．６
９６／２４	３．６	１２．０	２３．８
１９２／２４	６．７	２２．８	２６．７

【０１２９】
コーデックは、複雑度レベルの大きな範囲を与えるように設計される。最大レベルが最も遅いエンコーディング及びデコーディング速度を犠牲にして最も高い圧縮を達成するとしても、速い中間レベルだけが圧縮を劣化させるが、デコーディングは、最大レベル（すなわち、４８ｋＨｚ材料に対する約５％のＣＰＵ負荷）においてより複雑度が下がる。低い複雑度レベルを用いれば（すなわち、Ｋ－１５、ライスコーディング）、中間レベルよりも１から１．５％だけ圧縮が劣化するが、デコーダ複雑度は３の因子に見合う分だけさらに低減する（すなわち、４８ｋＨｚ材料に対する２％未満の負荷）。このため、オーディオデータは極めて低い演算電力を有するハードウェア上においてさえデコーディング可能である。
【０１３０】
エンコーダの複雑度が高い最大順序及びさらに精巧なブロックスイッチングアルゴリズム（この実施形態による）により増大するとしても、デコーダは、高い平均予測順序に影響されることがある。
【０１３１】
上述した実施形態（例えば、階層ブロックスイッチング）及び利点は単なる例示的なものに過ぎず、特許請求の範囲を制限するものとして解釈されてはならない。上述した教示は当業者にとって自明なものであり、他の装置及び方法に作用可能である。多くの代案、修正及び変形が当業者にとって自明である。
【０１３２】
例えば、ブロック図が本発明の構造的な様相を例示するのに用いられるとしても、これら様相はこれと関連する適宜なメモリー構成要素を備えたプログラミング可能なコンピュータプロセッサーとして具現可能であることを理解すべきである。
【産業上の利用可能性】
【０１３３】
当業者にとっては、本発明に対する種々の修正及び変更が本発明の範囲及び精神から逸脱することなく行われることは自明に理解できるであろう。例えば、本発明の態様及び実施形態は損失あるオーディオ信号コーデックと同じ他のオーディオ信号コーデックにも容易に適用可能である。よって、本発明は特許請求の範囲及びその均等物内に提供される本発明の修正及び変動をカバーする。
【図面の簡単な説明】
【０１３４】
本発明のさらなる理解を提供するために添付された図面は本発明の一部として一体化及び構成されて本発明の実施形態を例示し、説明とともに本発明の原理を説明するのに寄与する。
【０１３５】
【図１】本発明の一実施形態によるエンコーダの例示図である。
【図２】本発明の一実施形態によるデコーダの例示図である。

【図３】本発明の一実施形態による、圧縮Ｍチャンネルのビットストリーム構造の例示図である。
【図４】本発明の一実施形態による階層ブロックスイッチングの概念の例示図である。
【図５】ブロックスイッチング例と対応するブロックスイッチング情報コードの例示図である。
【図６】本発明の一実施形態による、複数のチャンネルのためのブロックスイッチング方法の例示図である。

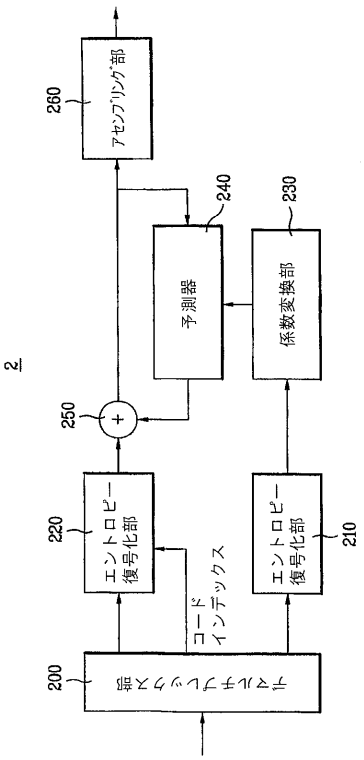
【図１】

FIG. 1



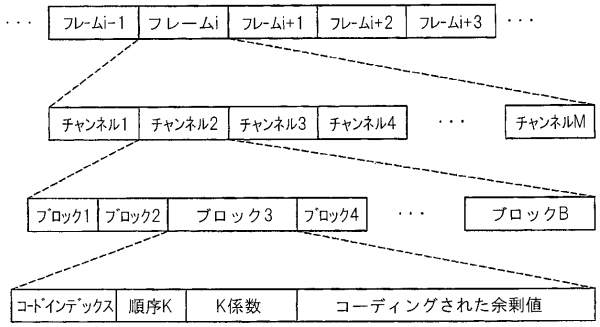
【図２】

FIG. 2



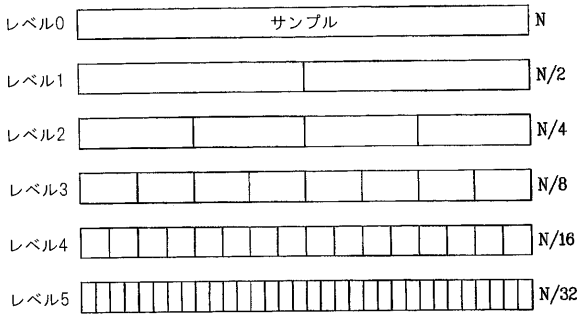
【図３】

FIG. 3



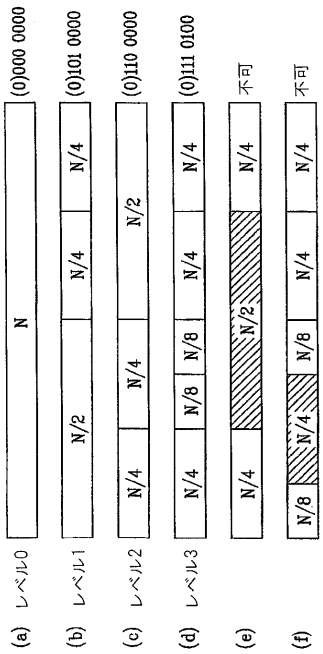
【図４】

FIG. 4



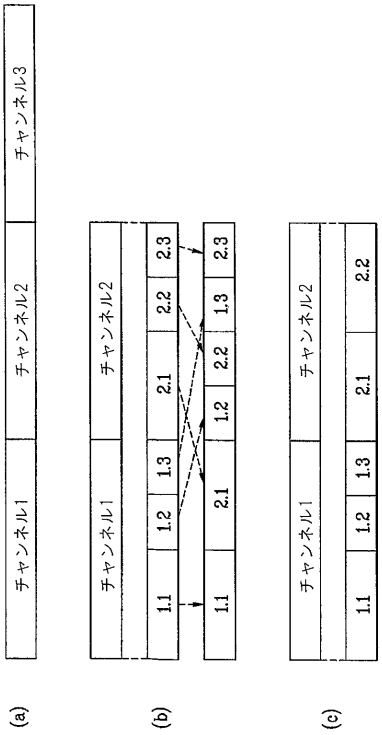
【図５】

FIG. 5



【図６】

FIG. 6



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR 2006/002685

A. CLASSIFICATION OF SUBJECT MATTER
IPC[8]: ***G10L 19/00*** (2006.01)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC[8]: G10L H04B H04H

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPODOC WPI INSPEC IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 20040247035 A1 (Schroder et al.) 9 December 2004 (09.12.2004) *paragraphs 0025 - 0040*	1-9
X	EP 599825 A2 (KONINKL PHILIPS ELECTRONICS NV) 1 June 1994 (01.06.1994) *Fig 12 and description*	1-9

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 October 2005 (25.10.2005)	24 November 2006 (24.11.2006)
Name and mailing address of the ISA/ AT **Austrian Patent Office** Dresdner Straße 87, A-1200 Vienna Facsimile No. +43 / 1 / 534 24 / 535	Authorized officer SCHLECHTER B. Telephone No. +43 / 1 / 534 24 / 448

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/KR 2006/002685

Patent document cited in search report			Publication date	Patent family member(s)			Publication date
EP	A	599825		AT	T	343873T	2006-11-15
				KR	B1	100149862B	1999-05-15
				US	A1	2006206341	2006-09-14
				UA	C2	73532	2004-03-15
				US	A1	2004138895	2004-07-15
				US	A1	2001044713	2001-11-22

Form PCT/ISA/210 (patent family annex) (July 1998; reprint January 2004)

フロントページの続き

(31)優先権主張番号 PCT/KR2005/002292
(32)優先日 平成17年7月16日(2005.7.16)
(33)優先権主張国 韓国(KR)
(31)優先権主張番号 PCT/KR2005/002306
(32)優先日 平成17年7月18日(2005.7.18)
(33)優先権主張国 韓国(KR)
(31)優先権主張番号 PCT/KR2005/002307
(32)優先日 平成17年7月18日(2005.7.18)
(33)優先権主張国 韓国(KR)
(31)優先権主張番号 PCT/KR2005/002308
(32)優先日 平成17年7月18日(2005.7.18)
(33)優先権主張国 韓国(KR)
(31)優先権主張番号 60/700,570
(32)優先日 平成17年7月19日(2005.7.19)
(33)優先権主張国 米国(US)

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IS,IT,LT,LU,LV,MC,NL,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LV,LY,MA,MD,MG,MK,MN,MW,MX,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM,ZW

(74)代理人 100122965
弁理士 水谷 好男
(72)発明者 リーブヒェン，ティルマン
ドイツ連邦共和国，デー－１２１６１ ベルリン，レフェフレシュトラーセ，１７