

(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2009-500682
(P2009-500682A)

(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)
G 1 0 L 19/00 (2006.01)	G 1 0 L 19/00 2 1 3	5 D 0 4 4
G 1 1 B 20/10 (2006.01)	G 1 0 L 19/00 2 2 0 F	
	G 1 1 B 20/10 3 1 1	
	G 1 0 L 19/00 2 5 0	

審査請求 有 予備審査請求 未請求 (全 38 頁)

(21) 出願番号	特願2008-521306 (P2008-521306)	(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 ドンポーク, ヨイドードン, 20
(86) (22) 出願日	平成18年7月10日 (2006.7.10)	(74) 代理人	100099759 弁理士 青木 篤
(85) 翻訳文提出日	平成20年3月5日 (2008.3.5)	(74) 代理人	100092624 弁理士 鶴田 準一
(86) 国際出願番号	PCT/KR2006/002678	(74) 代理人	100102819 弁理士 島田 哲郎
(87) 国際公開番号	W02007/008000	(74) 代理人	100151459 弁理士 中村 健一
(87) 国際公開日	平成19年1月18日 (2007.1.18)	(74) 代理人	100108383 弁理士 下道 晶久
(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)		

最終頁に続く

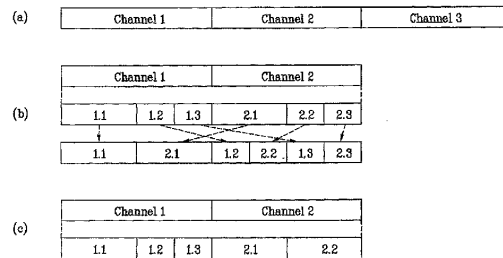
(54) 【発明の名称】 オーディオ信号のエンコーディング及びデコーディング装置及び方法

(57) 【要約】

【課題】 オーディオ信号のエンコーディング及びデコーディング装置及び方法を提供する。

【解決手段】 ある実施形態において、オーディオ信号のフレーム内の少なくとも第1及び第2のチャンネルは、これらが互いに相関しない場合にブロックに独立してサブ分割される。少なくとも2つのブロックは異なるブロックサイズを有する。さらに、第1及び第2のチャンネルは、これらが互いに相関する場合に第2のチャンネルがサブ分割されるブロックの長さが第1のチャンネルがサブ分割されるブロックの長さに対応するようにブロックに対応してサブ分割される。少なくとも2つのブロックは異なるブロック長を有する。

【選択図】 図6



【特許請求の範囲】**【請求項 1】**

オーディオ信号を処理する方法であって、

第 1 及び第 2 のチャンネルが互いに相関しない場合、前記オーディオ信号の 1 フレーム内の少なくとも第 1 及び第 2 のチャンネルをブロックに独立してサブ分割するが、前記ブロックのうち少なくとも 2 つが異なるブロック長を有するステップと、

前記第 1 及び第 2 のチャンネルが互いに相関する場合、前記第 2 のチャンネルがサブ分割されるブロックの長さが前記第 1 のチャンネルがサブ分割されるブロックの長さに対応するように前記第 1 及び第 2 のチャンネルをブロックに対応してサブ分割するが、前記ブロックのうち少なくとも 2 つが異なるブロック長を有するステップと、
を含む方法。

10

【請求項 2】

前記第 1 及び第 2 のチャンネルが対応してサブ分割される場合、前記第 1 及び第 2 のチャンネルのサブ分割されたブロックをインタリーブするステップをさらに含む請求項 1 に記載の方法。

【請求項 3】

前記独立してサブ分割するステップ及び前記対応してサブ分割するステップはサブ分割階層に応じて前記第 1 及び第 2 のチャンネルをサブ分割し、前記サブ分割階層は 1 以上のレベルを有し、各レベルは異なるブロック長と関連する請求項 2 に記載の方法。

【請求項 4】

前記サブ分割階層の上位レベルは、上位レベルと関連するブロック長の 2 倍となるブロック長と関連する請求項 3 に記載の方法。

20

【請求項 5】

前記独立してサブ分割するステップ及び前記対応してサブ分割するステップは、前記上位レベルのブロックをサブ分割して前記上位レベルにおいて 2 つのブロックを取得する請求項 3 に記載の方法。

【請求項 6】

前記第 1 及び第 2 のチャンネルが N の長さを有する場合、前記独立してサブ分割するステップ及び前記対応してサブ分割するステップは前記第 1 及び第 2 のチャンネルを前記複数のブロックにサブ分割して各ブロックが $N/2$ 、 $N/4$ 、 $N/8$ 、 $N/16$ 及び $N/32$ のうちいずれかの長さを有する請求項 3 に記載の方法。

30

【請求項 7】

前記第 1 及び第 2 のチャンネルが前記ブロックにサブ分割された旨を示す情報を生成するステップをさらに含む請求項 3 に記載の方法。

【請求項 8】

前記生成ステップは、前記情報の長さが前記サブ分割階層内のレベルの数に従属するように前記情報を生成する請求項 7 に記載の方法。

【請求項 9】

前記生成ステップは前記情報が情報ビットの数を含むように前記情報を生成し、前記情報ビットは前記第 1 及び第 2 のチャンネルが前記ブロックにサブ分割された旨を示す請求項 7 に記載の方法。

40

【請求項 10】

各情報ビットは前記サブ分割階層内のレベルと関連し、前記関連するレベルにおいてブロックと関連し、各情報ビットは前記関連するブロックがサブ分割されるかどうかを示す請求項 9 に記載の方法。

【請求項 11】

前記情報ビットが 1 の値を有する場合に前記関連するブロックはサブ分割され、前記情報ビットが 0 の値を有する場合には前記関連するブロックはサブ分割されない請求項 10 に記載の方法。

【請求項 12】

50

前記情報を送信するステップをさらに含む請求項 2 に記載の方法。

【請求項 1 3】

以前のデータサンプルに基づいて前記第 1 及び第 2 のチャンネルにおいて現在のデータサンプルを予測するが、前記予測ステップにおいて用いられた前記以前のデータサンプルの数が前記予測順序であるステップと、

前記予測されたデータサンプルに基づいて前記現在のデータサンプルの余りを取得するステップと、

をさらに含む請求項 2 に記載の方法。

【請求項 1 4】

前記予測ステップは、以前のデータサンプルが利用可能になるときに前記予測順序を所定の予測順序に漸増させる請求項 1 3 に記載の方法。

10

【請求項 1 5】

以前のデータサンプルを有さない初期のデータに対する前記予測順序は 0 であり、前記予測ステップは前記予測された初期のデータサンプルとして 0 を生成し、

前記取得ステップは、前記予測された初期のデータサンプルの結果が 0 のときに前記初期のデータサンプルを変更させない請求項 1 4 に記載の方法。

【請求項 1 6】

最大許容予測順序及び前記ブロックの長さに基づいて各ブロックに対する前記所定の予測順序を決めるステップをさらに含む請求項 1 4 に記載の方法。

【請求項 1 7】

最大許容予測順序及び前記ブロックの長さに基づいて各ブロックに対する最適な予測順序を決めるステップをさらに含む請求項 1 3 に記載の方法。

20

【請求項 1 8】

前記決定ステップは、

前記最大許容予測順序に基づいてグローバル予測順序を決めるステップと、

前記ブロックの長さに基づいてローカル予測順序を決めるステップと、

前記グローバル予測順序及び前記ローカル予測順序を前記最適な予測順序として選択するステップと、

を含む請求項 1 7 に記載の方法。

【請求項 1 9】

前記フレームが、以前のフレームが前記フレームをデコーディングするのに不要になるようにエンコーディングされるべきフレームであるランダムアクセスフレームの場合、

前記予測ステップは、前記ランダムアクセスフレームのみからの以前のデータサンプルが利用可能になるときに前記予測順序を所定の予測順序に漸増させる方法。

30

【請求項 2 0】

前記第 1 及び第 2 のチャンネルのサブ分割されたブロックは、前記第 1 及び第 2 のチャンネルが互いに相関しない場合にインタリーブされない請求項 1 に記載の方法。

【請求項 2 1】

前記独立してサブ分割するステップ及び前記対応してサブ分割するステップはサブ分割階層に応じて前記第 1 及び第 2 のチャンネルをサブ分割し、前記サブ分割階層は 1 以上のレベルを有し、各レベルは異なるブロック長と関連する請求項 2 0 に記載の方法。

40

【請求項 2 2】

前記サブ分割階層の上位レベルは、上位レベルと関連するブロック長の 2 倍となるブロック長と関連する請求項 2 1 に記載の方法。

【請求項 2 3】

前記独立してサブ分割するステップ及び前記対応してサブ分割するステップは、前記上位レベルのブロックをサブ分割して前記上位レベルにおいて 2 つのブロックを取得する請求項 2 1 に記載の方法。

【請求項 2 4】

前記第 1 及び第 2 のチャンネルが N の長さを有する場合、前記独立してサブ分割するス

50

テップ及び前記対応してサブ分割するステップは、前記第 1 及び第 2 のチャンネルを前記複数のブロックにサブ分割して、各ブロックが $N/2$ 、 $N/4$ 、 $N/8$ 、 $N/16$ 及び $N/32$ のうちいずれかの長さを有する請求項 2 1 に記載の方法。

【請求項 2 5】

前記第 1 及び第 2 のチャンネルが前記ブロックにサブ分割された旨を示す情報を生成するステップをさらに含む請求項 2 1 に記載の方法。

【請求項 2 6】

前記生成ステップは、前記情報の長さが前記サブ分割階層内のレベルの数に従属するように前記情報を生成する請求項 2 5 に記載の方法。

【請求項 2 7】

前記生成ステップは前記情報が情報ビットの数を含むように前記情報を生成し、前記情報ビットは前記第 1 及び第 2 のチャンネルが前記ブロックにサブ分割された旨を示す請求項 2 5 に記載の方法。

【請求項 2 8】

各情報ビットは前記サブ分割階層内のレベルと関連し、前記関連するレベルにおいてブロックと関連し、各情報ビットは前記関連するブロックがサブ分割されるかどうかを示す請求項 2 7 に記載の方法。

【請求項 2 9】

前記情報ビットが 1 の値を有する場合に前記関連するブロックはサブ分割され、前記情報ビットが 0 の値を有する場合には前記関連するブロックはサブ分割されない請求項 2 8 に記載の方法。

【請求項 3 0】

前記情報を送信するステップをさらに含む請求項 2 0 に記載の方法。

【請求項 3 1】

以前のデータサンプルに基づいて前記第 1 及び第 2 のチャンネルにおいて現在のデータサンプルを予測するが、前記予測ステップにおいて用いられた前記以前のデータサンプルの数が前記予測順序であるステップと、

前記予測されたデータサンプルに基づいて前記現在のデータサンプルの余りを取得するステップと、

をさらに含む請求項 2 0 に記載の方法。

【請求項 3 2】

前記予測ステップは、以前のデータサンプルが利用可能になるときに前記予測順序を所定の予測順序に漸増させる請求項 3 1 に記載の方法。

【請求項 3 3】

以前のデータサンプルを有さない初期のデータに対する前記予測順序は 0 であり、前記予測ステップは前記予測された初期のデータサンプルとして 0 を生成し、

前記取得ステップは、前記予測された初期のデータサンプルの結果が 0 のときに前記初期のデータサンプルを変更させない請求項 3 2 に記載の方法。

【請求項 3 4】

最大許容予測順序及び前記ブロックの長さに基づいて各ブロックに対する前記所定の予測順序を決めるステップをさらに含む請求項 3 2 に記載の方法。

【請求項 3 5】

最大許容予測順序及び前記ブロックの長さに基づいて各ブロックに対する最適な予測順序を決めるステップをさらに含む請求項 3 1 に記載の方法。

【請求項 3 6】

前記決定ステップは、

前記最大許容予測順序に基づいてグローバル予測順序を決めるステップと、

前記ブロックの長さに基づいてローカル予測順序を決めるステップと、

前記グローバル予測順序及び前記ローカル予測順序を前記最適な予測順序として選択するステップと、

10

20

30

40

50

を含む請求項 35 に記載の方法。

【請求項 37】

前記フレームが、以前のフレームが前記フレームをデコーディングするのに不要になるようにエンコーディングされるべきフレームであるランダムアクセスフレームの場合、

前記予測ステップは、前記ランダムアクセスフレームのみからの以前のデータサンプルが利用可能になるときに前記予測順序を所定の予測順序に漸増させる請求項 31 に記載の方法。

【請求項 38】

オーディオ信号を処理する方法であって、

複数のチャンネルが互いに相関しない場合、オーディオデータフレーム内の複数のチャンネルが分割されるブロックの長さを独立してスイッチングするステップと、

前記複数のチャンネルが互いに相関する場合、前記フレーム内の前記複数のチャンネルが分割されるブロックの長さをスイッチングして、各チャンネルのブロック長を同一長さに同時にスイッチングさせるステップと、

を含み、

前記フレームは複数のブロックに分割され、前記フレーム内の前記ブロックのうち少なくとも 2 つは異なるブロック長を有する方法。

【請求項 39】

オーディオ信号をエンコーディングする方法であって、

第 1 及び第 2 のチャンネルが互いに相関しない場合、前記オーディオ信号の 1 フレーム内の少なくとも第 1 及び第 2 のチャンネルをブロックに独立してサブ分割するが、前記ブロックのうち少なくとも 2 つが異なるブロック長を有するステップと、

前記第 1 及び第 2 のチャンネルが互いに相関する場合、前記第 2 のチャンネルがサブ分割されるブロックの長さが前記第 1 のチャンネルがサブ分割されるブロックの長さに対応するように前記第 1 及び第 2 のチャンネルをブロックに対応してサブ分割するが、前記ブロックのうち少なくとも 2 つが異なるブロック長を有するステップと、

前記ブロックをエンコーディングして圧縮ビットストリームを生成するステップと、を含む方法。

【請求項 40】

オーディオ信号をデコーディングする方法であって、

少なくとも第 1 及び第 2 のチャンネルを備えるオーディオフレームを受信するが、前記第 1 及び第 2 のチャンネルが互いに相関しない場合に前記第 1 及び第 2 のチャンネルをブロックに独立してサブ分割し、前記第 1 及び第 2 のチャンネルが互いに相関する場合には前記第 2 のチャンネルがサブ分割されるブロックの長さが前記第 1 のチャンネルがサブ分割されるブロックの長さに対応するように前記第 1 及び第 2 のチャンネルをブロックに対応してサブ分割し、前記ブロックのうち少なくとも 2 つが前記第 1 及び第 2 のチャンネルのそれぞれにおいて異なるブロック長を有するステップと、

前記第 1 及び第 2 のチャンネルをデコーディングするステップと、を含む方法。

【請求項 41】

オーディオ信号をデコーディングする装置であって、

第 1 及び第 2 のチャンネルが互いに相関しない場合に前記オーディオ信号の 1 フレーム内の少なくとも第 1 及び第 2 のチャンネルをブロックに独立してサブ分割し、前記ブロックのうち少なくとも 2 つが異なるブロック長を有するように構成されたエンコーダーを備え、

前記エンコーダーは、前記第 1 及び第 2 のチャンネルが互いに相関する場合、前記第 2 のチャンネルがサブ分割されるブロックの長さが前記第 1 のチャンネルがサブ分割されるブロックの長さに対応するように前記第 1 及び第 2 のチャンネルをブロックに対応してサブ分割し、前記ブロックのうち少なくとも 2 つが異なるブロック長を有するように構成され、

10

20

30

40

50

前記エンコーダーは、前記ブロックをエンコーディングして圧縮ストリームを生成する装置。

【請求項 4 2】

オーディオ信号をデコーディングする装置であって、

少なくとも第 1 及び第 2 のチャンネルを備えるオーディオフレームを受信するが、前記第 1 及び第 2 のチャンネルが互いに相関しない場合に前記第 1 及び第 2 のチャンネルをブロックに独立してサブ分割し、前記第 1 及び第 2 のチャンネルが互いに相関する場合には前記第 2 のチャンネルがサブ分割されるブロックの長さが前記第 1 のチャンネルがサブ分割されるブロックの長さに対応するように前記第 1 及び第 2 のチャンネルをブロックに対応してサブ分割し、前記ブロックのうち少なくとも 2 つが前記第 1 及び第 2 のチャンネルのそれぞれにおいて異なるブロック長を有するように構成されたデコーダーを備え、

前記デコーダーは、前記第 1 及び第 2 のチャンネルをデコーディングするように構成される装置。

【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、レコードプレーヤー）、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ー 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じて、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を処理することに関する。

【課題を解決するための手段】

【0005】

ある実施形態において、オーディオ信号のフレーム内の少なくとも第 1 及び第 2 のチャンネルは、これらが互いに相関しない場合にブロックに独立してサブ分割される。少なくとも 2 つのブロックは、異なるブロックサイズを有する。さらに、第 1 及び第 2 のチャンネルは、これらが互いに相関する場合に第 2 のチャンネルがサブ分割されるブロックの長さが第 1 のチャンネルがサブ分割されるブロックの長さに対応するようにブロックに対応してサブ分割される。少なくとも 2 つのブロックは異なるブロック長を有する。

【0006】

ある実施形態において、第 1 及び第 2 のチャンネルのサブ分割されたブロックは、第 1 及び第 2 のチャンネルが対応してサブ分割される場合にインターリーブされる。

【0007】

10

20

30

40

50

他の実施形態において、第1及び第2のチャンネルのサブ分割されたブロックは、第1及び第2のチャンネルが互いに相関しない場合にインタリーブされない。

【0008】

ある実施形態によれば、第1及び第2のチャンネルは、サブ分割された階層に応じて、独立して且つ対応してサブ分割される。サブ分割階層は1以上のレベルを有し、各レベルは異なるブロック長と関連する。例えば、サブ分割階層の上位レベルは上位レベルと関連するブロック長の2倍となるブロック長と関連する。

【0009】

ある実施形態において、チャンネルがNの長さを有する場合、チャンネルは複数のブロックにサブ分割されて、各ブロックがN/2、N/4、N/8、N/16及びN/32のうちいずれかの長さを有する。

10

【0010】

ある実施形態において、情報は情報の長さがサブ分割階層内のレベル数に従属するように生成される。例えば、情報は情報ビットの数を含むように生成され、情報ビットはチャンネルをブロックにサブ分割された旨を示す。特に、各情報ビットはサブ分割階層内のレベルと関連し、関連するレベルにおいてブロックと関連し、各情報ビットは関連するブロックがサブ分割されるかどうかを示す。

【0011】

ある実施形態において、方法は、以前のデータサンプルに基づいてチャンネル内の現在のデータサンプルを予測するステップを含む。予測ステップにおいて用いられる以前のデータサンプルの数は予測順序として言及される。現在のデータサンプルの余りは予測されたデータサンプルに基づいて取得される。

20

【0012】

ある実施形態において、予測は、以前のデータサンプルが利用可能になるときに、予測順序を所定の予測順序に漸増させながら行われる。例えば、このような漸進的な予測プロセスは、以前のフレームがフレームをデコーディングするのに不要になるようにエンコーディングされるべきフレームであるランダムアクセスフレームに対して行われる。

【0013】

他の実施形態において、各ブロックに対する所定の予測順序は、最大許容予測順序及びブロックの長さに基づいて決まる。

30

【0014】

オーディオ信号を処理する方法の他の実施形態において、オーディオデータフレーム内の複数のチャンネルそれぞれが分割されるブロックの長さは、複数のチャンネルが互いに相関しない場合に独立してスイッチングされる。さらに、複数のチャンネルが互いに相関する場合、各チャンネルのブロック長が同一長さに同時にスイッチングされるようにフレーム内の複数のチャンネルが分割されるブロックの長さがスイッチングされる。これにより、チャンネルは複数のブロックにそれぞれ分割され、各チャンネル内の少なくとも2つのブロックは異なるブロック長を有する。

【0015】

本発明はオーディオ信号をエンコーディングする方法及び装置、並びに、オーディオ信号をデコーディングする方法及び装置に関する。

40

【発明を実施するための最良の形態】

【0016】

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。

【0017】

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。

【0018】

50

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用語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。

【0019】

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、エンコーダー及びデコーダー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。

【0020】

[コーデック構造]

図1は、本発明によるエンコーダー1の例示図である。

【0021】

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。

【0022】

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。

【0023】

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。

【0024】

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。

【0025】

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。

【0026】

最後に、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、エンコーダー1は、デコーディングされ

10

20

30

40

50

たデータを検証するためにデコーダー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。エンコーダー側上において、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。

【0027】

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。エンコーダー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ーの説明後に行う。

10

【0028】

図2は、本発明によるデコーダー2の例示図である。特に、図2は、エンコーダー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ーを示し、その理由は、いかなる適用も行われなためである。

【0029】

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ー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。

20

30

【0030】

この後、デコーダー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、逆予測フィルターを適用して無損失再構成信号を演算する。デコーダー2の計算量は、エンコーダー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ー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「j s _ s t e r e o」フィールドは、ジョイント

40

50

ステレオ（チャンネル差分）がオンであるか、またはオフであることを示し、フレーム__データシンタックス（テーブル7）内の「j s _ s w i t c h」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。

【0031】

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。

【0032】

[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。

【0033】

[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。

【0034】

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。状況によっては、このようなスキームはいくらかの制限を有する。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、さらに有効である場合がある。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（チャンネルが関連しない場合には特にあてはまる）。

【0035】

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5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 $i = 1, 2, \dots, p$ 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。

【0036】

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。

【0037】

10

20

30

40

50

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。

【0038】

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。

10

【0039】

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。

20

【0040】

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たい旨を示す場合）、「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。

30

40

【0041】

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。

【0042】

テーブル1：ブロックスイッチングレベル

【表 1】

最大のレベル数	最小 N_B	「 <code>bs_info</code> 」に対するバイト数
0 (「 <code>block_switching=00</code> 」)	N	0
1 (「 <code>block_switching=01</code> 」)	N/2	1 (=8 bits)
2 (「 <code>block_switching=01</code> 」)	N/4	1 (=8 bits)
3 (「 <code>block_switching=01</code> 」)	N/8	1 (=8 bits)
4 (「 <code>block_switching=10</code> 」)	N/16	2 (=16 bits)
5 (「 <code>block_switching=11</code> 」)	N/32	4 (=32 bits)

10

【0043】

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。

20

【0044】

`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) 1 2 2 3 3 3 3 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5] 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 $N_B = N$ から始めるとすれば、`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、「`bs_info`」は (0) 0 0 0 0 0 0 0 0 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) 1 0 1...) を $N/4$ 長さの2つのブロックに分割して、「`bs_info`」は (0) 1 0 1 0 0 0 0 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) 1 1 0...) を $N/4$ 長さの2つのブロックに分割して、「`bs_info`」は (0) 1 1 0 0 0 0 0 0 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) 1 1 1...) 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) 1 1 1 0 1...) 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) 1 1 1 0 1 0 0 0 となる。

30

40

【0045】

上述したように、図5(e)及び図5(f)の例は、図5(e)の $N/2$ ブロックと図5(f)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。

【0046】

[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。

【0047】

50

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、エンコーダーにより自動的に行われうるか、または、ユーザー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)。

【0048】

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。

【0049】

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。

【0050】

以下、図6(a)から図6(c)について説明する。

【0051】

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。

【0052】

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、利得)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

10

20

30

40

50

)に從属し、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。

【0053】

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、「 $bs_info = (1)1010000$ 」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、「 $bs_info = (1)1000000$ 」の bs_info を有する。図6(c)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。

10

【0054】

[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。

20

【0055】

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從属して行われる。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ーにより並替え可能である。

【0056】

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。

【0057】

[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、エンコーダー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。

40

【0058】

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、「 $random_access = 00000000$ 」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、「 $random_access = 00000001$ 」の場合、これは、ランダムアクセ

50

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 = 1 1 1 1 1 1 1 1」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。

【0059】

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。

【0060】

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の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。

【0061】

[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ー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

10

20

30

40

50

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。

【0062】

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。

【0063】

テーブル2：チャンネル構成

【表2】

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

【0064】

[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総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。

【0065】

さらに、「frame_length」フィールドの16ビット値はエンコーダーにより用いられる値により決まり、ユーザー定義値と呼ばれる。すなわち、固定値の代わりに、ユーザー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。

【0066】

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。

【0067】

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

【0068】

特に、各チャンネルに対するフレームの総数は、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総数を、「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総数が、「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、その倍数の値がフレームの総数となる。しかしながら、「sample」フィールドにより決まったサンプルの総数が、「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、フレームの総数は倍数値以上に「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ことを示す。上述したように、エンコーダーとデコーダーとの間の標準化された規則を定義することにより、エンコーダーは、各チャンネルに対するサンプルの総数(「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、デコーダーは、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。

【0069】

[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以上のフィルター係数を含み、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18

10

20

30

40

50

0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。

【0070】

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。

【0071】

[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。

10

【0072】

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。予測は下記の式として与えられる。

【0073】

【数1】

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

20

【0074】

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。

【0075】

【数2】

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

30

【0076】

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。

【0077】

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、フォワード適応と呼ばれる。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。

40

【0078】

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。

【0079】

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、

50

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。これらのシステムのほとんどは、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、適応がエンコーダーとデコーダーの両方に行われる必要があるという欠点を有するため、デコーダーがフォワード適応型の場合よりも複雑になる。

【0080】

[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パラツキの項）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-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。

10

【0081】

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、予測エラーのパラツキは低減され、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。

20

【0082】

【数3】

$$R_{total}(K) = R_e(K) + R_c(K)$$

30

【0083】

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。

【0084】

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、対応する余りのパラツキ ϵ^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。

40

【0085】

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。

【0086】

[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

50

、ビットストリームに含まれてから送信される。

【0087】

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。

10

【0088】

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、「adapt_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。

【0089】

20

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。

【0090】

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。

30

40

【0091】

特に、opt_orderは、以下の式に基づいて決まる。

【0092】

$opt_order = \min(\text{グローバル予測順序}, \text{ローカル予測順序})$

【0093】

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 = $\text{ceil}(\log_2(\text{最大予測順序} + 1))$ 、及び

ローカル予測順序 = $\max(\text{ceil}(\log_2(nb \gg 3)) - 1), 1)$

50

として決まる。

【0094】

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。

【0095】

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。

10

【0096】

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。

20

【0097】

[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ー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。

【0098】

第1の2つのパーコール係数(a_1 及び a_2 に対応する)は、下記式を用いて量子化される。

30

【0099】

【数4】

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

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

40

【0100】

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。

【0101】

【数 5】

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

【0102】

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。

【0103】

[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。

【0104】

[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。

【0105】

【数 6】

$$\delta_k = a_k - \text{offset}_k$$

【0106】

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 $\text{offset}_k = a_k + 64$ を得るために、常に -64 である。逆に、「coef__table = 00」の場合にテーブル 3 が選択され、「coef__table = 01」の場合にテーブル 4 が選択され、「coef__table = 11」の場合にテーブル 5 が選択される。

【0107】

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わせられるプロセスを用いて予測係数を再構成する。

【0108】

10

20

30

40

【数 7】

$$a_k = \delta_k + \text{offset}_k$$

【0 1 0 9】

この後、第 1 の 2 つの係数 (a_1 及び a_2) の再構成は、

【0 1 1 0】

【数 8】

$$\text{par}_1 = \lfloor \hat{\gamma}_1 2^Q \rfloor = \Gamma(a_1);$$

$$\text{par}_2 = \lfloor \hat{\gamma}_2 2^Q \rfloor = -\Gamma(a_2);$$

を用いて行われる。

【0 1 1 1】

【数 9】

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、 $\Gamma(\cdot)$ は、経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。

【0 1 1 2】

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、48 kHz、96 kHz、192 kHz 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。

【0 1 1 3】

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。44.1 kHz の材料において、本発明の出願人は、48 kHz 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。

【0 1 1 4】

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 (48 kHz)

【表 3】

係数#	オフセット	ライスパラメーター
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

10

20

【0115】

30

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ー(96kHz)

【表 4】

係数#	オフセット	ライスパラメーター
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

10

20

【0116】

30

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ー(192kHz)

【表 5】

係数#	オフセット	ライスパラメーター
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

10

20

30

40

【0117】

[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。

【0118】

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ーが選択しなければならない。

【0119】

代案的に、エンコーダーは、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、

【数 10】

$$|e(n)| < e_{\max}$$

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。最後に、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。

10

【0120】

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。

20

【0121】

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。

【0122】

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。

30

【0123】

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。

40

【0124】

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。

【0125】

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。

50

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。

【0126】

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。

10

【0127】

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。

【0128】

20

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。

【0129】

[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。

30

【0130】

テーブル6：構成シンタックス

【表 6】

シンタックス	ビット	
ALSSpecificConfig()		
{		
samp_freq;	32	
samples;	32	
channels;	16	
file_type;	3	
resolution;	3	
floating;	1	10
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	20
joint_stereo;	1	
mc_coding;	1	
chan_config;	1	
chan_sort;	1	
crc_enabled;	1	
RLS_LMS	1	
(reserved)	6	
if (chan_config) {		
chan_config_info;	16	
}		
if (chan_sort) {		
for (c = 0; c < channels; c++)		
chan_pos[c];	8	30
}		
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	40
}		
}		
}		

【 0 1 3 1 】

テーブル 7 : フレーム_データシンタックス

【表 7】

シンタックス	ビット	
<pre> frame_data() { if ((ra_flag == 1) && (frame_id % random_access == 0)) { ra_unit_size } if (mc_coding && joint_stereo) { js_switch; byte_align; } if (!mc_coding js_switch) { for (c = 0; c < channels; c++) { if (block_switching) { bs_info; } 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++; } } } else{ if (block_switching) { bs_info; } for (b = 0; b < blocks; b++) { for (c = 0; c < channels; c++) { block_data(c); channel_data(c); } } } if (floating) { num_bytes_diff_float; diff_float_data(); } } </pre>	32	
	1	10
	8, 16, 32	
		20
	8, 16, 32	30
	32	40

【 0 1 3 2 】

テーブル 8 : ブロック_データシンタックス

【表 8】

シンタックス	ビット	
block_data() {		
block_type;	1	
if (block_type == 0) {		
const_block;	1	
js_block;	1	
(reserved)	5	
if (const_block == 1) {		10
{		
if (resolution == 8) {		
const_val;	8	
}		
else if (resolution == 16) {		
const_val;	16	
}		
else if (resolution == 24) {		
const_val;	24	
}		
else {		
const_val;	32	
}		
}		
}		
} else {		20
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	30
}		
}		
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 (!RLSMLS) {		
if (adapt_order == 1) {		
opt_order;	1...10	40
}		
for (p = 0; p < opt_order; p++) {		
quant_cof[p];	varies	
}		
}		
}		

【0133】

[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一般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

A u d i o (M A C 3 . 9 7) と比較される。以下、オープンソースコーデック F L A C はフォワード適応型予測を用い、M o n k e y ' s A u d i o (M A C 3 . 9 7) は圧縮において現在の最先端の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 m a c - c 4 0 0 ）を与えるオプションで行われる。エンコーダーに対する結果は（K - 6 0 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 - 1 0 2 3 ）と、5 0 0 m s 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、1 0 2 4 M B のメモリを有する 1 . 7 G H z 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、4 8、9 6 及び 1 9 2 k H z のサンプリングレートを有する 1 G のステレオ波形データと、1 6 及び 2 4 ビットの解像度を含む。

10

【 0 1 3 4 】

[圧縮レート]

次には、圧縮レートが

【 数 1 1 】

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

20

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（1 9 2 k h z 材料が F L A C コーデックにより支援される）。

【 0 1 3 5 】

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（k H z / ビット）

【 表 9 】

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

30

【 0 1 3 6 】

この結果は、最大レベルにおいて A L S が全てのフォーマット、特に、高品位材料（すなわち、9 6 k H z / 2 4 ビット以上）用の F L A C 及び M o n k e y ' s A u d i o (M A C 3 . 9 7) 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、A L S は最上の全体圧縮を伝える。

40

【 0 1 3 7 】

[複雑性]

異なるコーデックの複雑性は、実際の実行、特に、エンコーダーの実行に極めて強く依存する。上述したように、本発明のオーディオ信号エンコーダーは継続して開発中である。このため、出願人は、その分析をエンコーダーに制限して、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、最上のエンコーダー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 C P U 負荷は、テーブル 1 0 に示す。最大

50

複雑度においてさえ、デコーダーに対する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。

【0138】

テーブル10：オーディオフォーマット(kHz/ビット)及びALSエンコーダー複雑度に従属する平均CPU負荷(1.7GHzペンティアム(登録商標)-M上のパーセント)

【表10】

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

10

【0139】

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、デコーダー複雑度は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。

20

【0140】

エンコーダー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、デコーダーは、高い平均予測順序に影響されることがある。

30

【0141】

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。

【産業上の利用可能性】

【0142】

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。

40

【図面の簡単な説明】

【0143】

【図1】本発明の一実施形態によるエンコーダーの例示図である。

【図2】本発明の一実施形態によるデコーダーの例示図である。

【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。

50

【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】

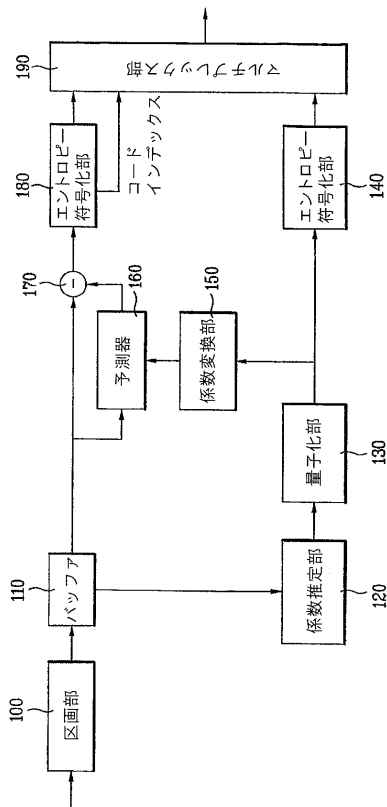


FIG. 1

【図2】

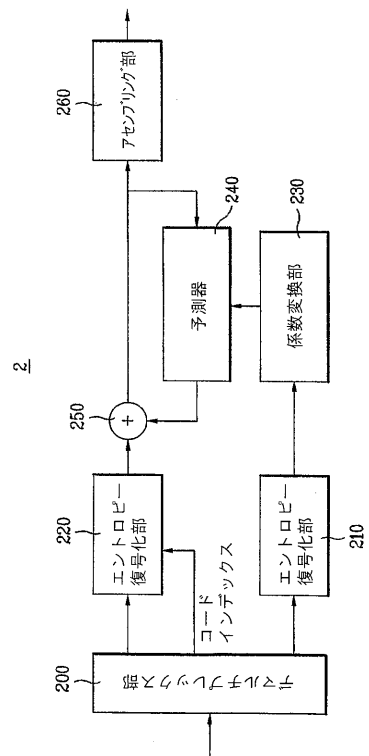
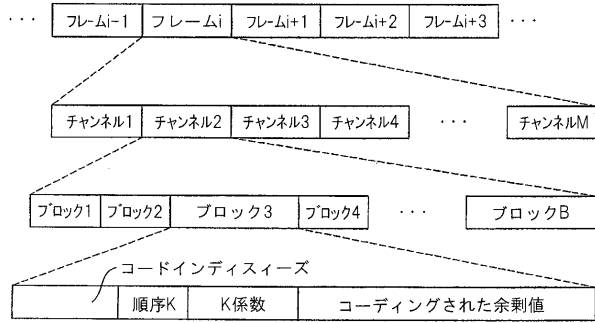


FIG. 2

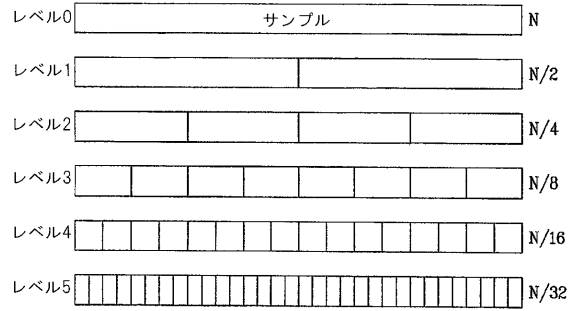
【 図 3 】

FIG. 3



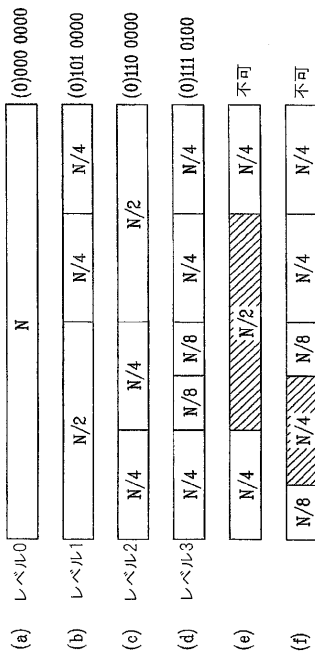
【 図 4 】

FIG. 4



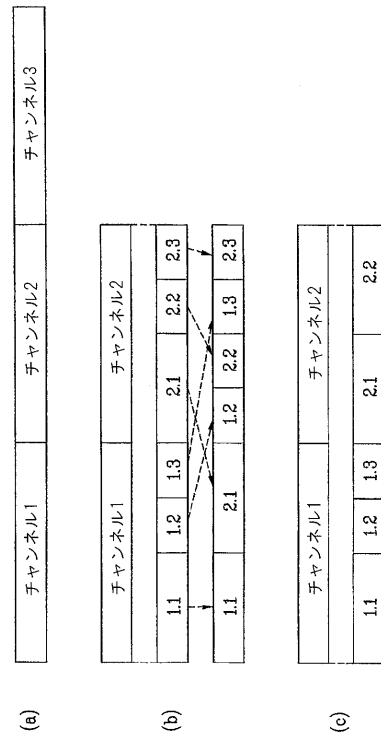
【 図 5 】

FIG. 5



【 図 6 】

FIG. 6



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR 2006/002678

A. CLASSIFICATION OF SUBJECT MATTER IPC⁸: 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) IPC⁸: G10L H04B H04N H04S 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) WPI EPODOC 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	EP 691751 A1 (Sony Corp.) 10 January 1996 (10.01.1996) <i>Page 9, lines 13 - 58, claims 1-13</i>	1-12, 20-30, 38-42
A	US 20040161116 A1 (Tsuji et al.) 19 August 2004 (19.08.2004) <i>paragraphs 0074 - 0087</i>	1-42
A	Kamamoto Y; Moriya T; Nishimoto T; Sagayama S: Lossless compression of multi-channel signals using inter-channel correlation Transactions of the Information Processing Society of Japan May 2005 <i>pages 1118 - 1128</i>	1-42
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 21 November 2006 (21.11.2006)		Date of mailing of the international search report 2 February 2007 (02.02.2007)
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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/KR 2006/002678

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
<u>A</u>		<u>none</u>	
EP A 691751		US A 5717670	1998-02-10
		WO A1 9515032	1995-06-01
		EP A1 0691751	1996-01-10
		JP A 7154265	1995-06-16

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 100114018

弁理士 南山 知広

(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 - 1 2 1 6 1 ベルリン, レフェフレシュトラッセ, 1 7

Fターム(参考) 5D044 AB05 BC01 BC02 CC01 CC04 DE12 GK08