

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4031909号
(P4031909)

(45) 発行日 平成20年1月9日(2008.1.9)

(24) 登録日 平成19年10月26日(2007.10.26)

(51) Int. Cl.		F I	
H03M 7/30	(2006.01)	H03M 7/30	A
G10L 11/00	(2006.01)	G10L 11/00	
G10L 19/00	(2006.01)	G10L 19/00	

請求項の数 39 (全 23 頁)

<p>(21) 出願番号 特願2000-601631 (P2000-601631)</p> <p>(86) (22) 出願日 平成12年2月18日 (2000.2.18)</p> <p>(65) 公表番号 特表2002-538644 (P2002-538644A)</p> <p>(43) 公表日 平成14年11月12日 (2002.11.12)</p> <p>(86) 国際出願番号 PCT/US2000/004248</p> <p>(87) 国際公開番号 W02000/051108</p> <p>(87) 国際公開日 平成12年8月31日 (2000.8.31)</p> <p>審査請求日 平成17年1月21日 (2005.1.21)</p> <p>(31) 優先権主張番号 09/259,512</p> <p>(32) 優先日 平成11年2月26日 (1999.2.26)</p> <p>(33) 優先権主張国 米国 (US)</p>	<p>(73) 特許権者 593181638 ソニー エレクトロニクス インク アメリカ合衆国 ニュージャージー州 O 7656 パークリッジ ソニー ドライ ブ 1</p> <p>(74) 代理人 100067736 弁理士 小池 晃</p> <p>(74) 代理人 100086335 弁理士 田村 榮一</p> <p>(74) 代理人 100096677 弁理士 伊賀 誠司</p> <p>(72) 発明者 ヒュアング、シェイジャン アメリカ合衆国 カリフォルニア州 95 035 ミルピタス ベイベリー ウエイ 416</p> <p style="text-align: right;">最終頁に続く</p>
---	--

(54) 【発明の名称】 時間領域エイリアシングを効率的に除去する装置及び方法

(57) 【特許請求の範囲】

【請求項1】

入力データシーケンスに対する前乗算処理を行い、第1の中間シーケンスを生成する前乗算手段(430)と、

上記第1の中間シーケンスに対する離散フーリエ変換処理を行い、第2の中間シーケンスを生成する離散フーリエ変換手段(440)と、

上記第2の中間シーケンスに対する後乗算処理を行い、出力データシーケンスを生成する後乗算手段(450)とを備え、

上記第1の中間シーケンスの算出は、nを上記入力データシーケンスの変数とし、pを上記第1の中間シーケンスの変数とし、jを虚数単位とし、Nを256として、以下の式に基づいて、上記入力データシーケンスの要素 $x[n]$ から上記第1の中間シーケンスの要素 $Z[p]$ を算出することを特徴とする変換回路(230)。

$$Z[p] = \frac{(x[2p] - x[2N-2p-1]) - (x[N+2p] + x[N-1-2p]) - j(x[2p] + x[2N-1-2p] + (x[N+2p] - x[N-1-2p]))}{(\cos(2\pi / (16N) * (8p+1)) - j \sin(2\pi / (16N) * (8p+1)))}$$

【請求項2】

上記前乗算手段(430)、上記離散フーリエ変換手段(440)、及び上記後乗算手段(450)における処理は、デジタル信号処理装置により実行されることを特徴とする請求項1記載の変換回路(230)。

【請求項3】

10

20

上記前乗算手段(430)、上記離散フーリエ変換手段(440)、及び上記後乗算手段(450)は、デジタルオーディオ信号用のエンコード/デコード装置内に配設されていることを特徴とする請求項1記載の変換回路(230)。

【請求項4】

上記エンコード/デコード装置は、AC-3規格に準拠することを特徴とする請求項3記載の変換回路(230)。

【請求項5】

上記出力データシーケンスは、上記入力データシーケンスを修正離散コサイン変換したものであることを特徴とする請求項1記載の変換回路(230)。

【請求項6】

上記入力データシーケンスは、入力データサンプルの長ブロックを含むことを特徴とする請求項5記載の変換回路(230)。

【請求項7】

上記長ブロックは、512個の入力データサンプルを含むことを特徴とする請求項6記載の変換回路(230)。

【請求項8】

上記第1の中間シーケンスは、128個の前乗算処理されたデータサンプルを含むことを特徴とする請求項7記載の変換回路(230)。

【請求項9】

上記前乗算手段(430)は、上記入力データシーケンスから上記第1の中間シーケンスを算出するコード手段を備えることを特徴とする請求項8記載の変換回路(230)。

【請求項10】

上記離散フーリエ変換は高速フーリエ変換であることを特徴とする請求項9記載の変換回路(230)。

【請求項11】

上記高速フーリエ変換は、基数4高速フーリエ変換に縦続接続された基数2高速フーリエ変換であることを特徴とする請求項10記載の変換回路(230)。

【請求項12】

上記入力データシーケンスは、入力データサンプルの短ブロックを含むことを特徴とする請求項5記載の変換回路(230)。

【請求項13】

上記短ブロックは、256個の入力データサンプルを含むことを特徴とする請求項12記載の変換回路(230)。

【請求項14】

上記第1の中間シーケンスは、64個の前乗算処理されたデータサンプルを含むことを特徴とする請求項7記載の変換回路(230)。

【請求項15】

上記前乗算手段(430)は、上記入力データシーケンスから上記第1の中間シーケンスを算出するコード手段を備えることを特徴とする請求項14記載の変換回路(230)。

【請求項16】

上記離散フーリエ変換は高速フーリエ変換であることを特徴とする請求項15記載の変換回路(230)。

【請求項17】

上記高速フーリエ変換は、基数4高速フーリエ変換であることを特徴とする請求項16記載の変換回路(230)。

【請求項18】

前乗算回路(430)により、入力データシーケンスに対する前乗算処理を行い、第1の中間シーケンスを生成するステップと、

離散フーリエ変換回路(440)により、上記第1の中間シーケンスに対する離散フー

10

20

30

40

50

リエ変換処理を行い、第 2 の中間シーケンスを生成するステップと、

後乗算回路 (4 5 0) により、上記第 2 の中間シーケンスに対する後乗算処理を行い、出力データシーケンスを生成するステップとを有し、

上記第 1 の中間シーケンスの算出は、n を上記入力データシーケンスの変数とし、p を上記第 1 の中間シーケンスの変数とし、j を虚数単位とし、N を 2 5 6 として、以下の式に基づいて、上記入力データシーケンスの要素 x [n] から上記第 1 の中間シーケンスの要素 Z [p] を算出することを特徴とする変換方法。

$$Z[p]=((x[2p]-x[2N-2p-1])-(x[N+2p]+x[N-1-2p]))-j(x[2p]+x[2N-1-2p]+(x[N+2p]-x[N-1-2p]))*(\cos(2\pi/(16N)*(8p+1))-j\sin(2\pi/(16N)*(8p+1)))$$

10

【請求項 1 9】

上記前乗算回路 (4 3 0)、上記離散フーリエ変換回路 (4 4 0)、及び上記後乗算回路 (4 5 0) における処理は、デジタル信号処理装置により実行されることを特徴とする請求項 1 8 記載の変換方法。

【請求項 2 0】

上記前乗算回路 (4 3 0)、上記離散フーリエ変換回路 (4 4 0)、及び上記後乗算回路 (4 5 0) は、デジタルオーディオ信号用のエンコード/デコード装置内に配設されていることを特徴とする請求項 1 8 記載の変換方法。

【請求項 2 1】

上記エンコード/デコード装置は、A C - 3 規格に準拠することを特徴とする請求項 2 0 記載の変換方法。

20

【請求項 2 2】

上記出力データシーケンスは、上記入力データシーケンスを修正離散コサイン変換したものであることを特徴とする請求項 1 8 記載の変換方法。

【請求項 2 3】

上記入力データシーケンスは、入力データサンプルの長ブロックを含むことを特徴とする請求項 2 2 記載の変換方法。

【請求項 2 4】

上記長ブロックは、5 1 2 個の入力データサンプルを含むことを特徴とする請求項 2 3 記載の変換方法。

30

【請求項 2 5】

上記第 1 の中間シーケンスは、1 2 8 個の前乗算処理されたデータサンプルを含むことを特徴とする請求項 2 4 記載の変換方法。

【請求項 2 6】

上記前乗算回路 (4 3 0) は、上記入力データシーケンスから上記第 1 の中間シーケンスを算出するコードを備えることを特徴とする請求項 2 5 記載の変換方法。

【請求項 2 7】

上記離散フーリエ変換は、q を上記第 2 の中間シーケンスの変数とし、p の範囲を 0 から N / 2 とし、下記の式により上記要素 Z [p] から上記第 2 の中間シーケンスの要素 z [q] を算出することを特徴とする請求項 1 8 記載の変換方法。

40

$$z[q]=Z[p]*(\cos(2\pi pq/(N/2))-j\sin(2\pi pq/(N/2)))$$

【請求項 2 8】

上記離散フーリエ変換は高速フーリエ変換であることを特徴とする請求項 1 8 記載の変換方法。

【請求項 2 9】

上記高速フーリエ変換は、基数 4 高速フーリエ変換に縦続接続された基数 2 高速フーリエ変換であることを特徴とする請求項 2 8 記載の変換方法。

【請求項 3 0】

上記入力データシーケンスは、入力データサンプルの短ブロックを含むことを特徴とする請求項 2 2 記載の変換方法。

50

【請求項 3 1】

上記短ブロックは、256個の入力データサンプルを含むことを特徴とする請求項 3 0 記載の変換方法。

【請求項 3 2】

上記第 1 の中間シーケンスは、64個の前乗算処理されたデータサンプルを含むことを特徴とする請求項 3 1 記載の変換方法。

【請求項 3 3】

上記前乗算回路(430)は、上記入力データシーケンスから上記第 1 の中間シーケンスを算出するコードを備えることを特徴とする請求項 3 2 記載の変換方法。

【請求項 3 4】

前乗算回路(430)により、入力データシーケンスに対する前乗算処理を行い、第 1 の中間シーケンスを生成するステップと、

離散フーリエ変換回路(440)により、上記第 1 の中間シーケンスに対する離散フーリエ変換処理を行い、第 2 の中間シーケンスを生成するステップと、

後乗算回路(450)により、上記第 2 の中間シーケンスに対する後乗算処理を行い、出力データシーケンスを生成するステップとを有し、

上記第 1 の中間シーケンスの算出は、n を上記入力データシーケンスの変数とし、p を上記第 1 の中間シーケンスの変数とし、j を虚数単位とし、N を 256 として、以下の式に基づいて、上記入力データシーケンスの要素 $x[n]$ から上記第 1 の中間シーケンスの要素 $Z_1[p]$ を算出し、

$$Z_1[p] = ((x[2p] - x[N-1-2p]) + j(x[N/2-1-2p] - x[N/2+2p] - x[N/2+2p])) \\ * (\cos(2\pi / (8N) * (8p+1)) - j \sin(2\pi / (8N) * (8p+1)))$$

以下の式に基づいて上記第 1 の中間シーケンスの要素 $Z_2[p]$ を算出することを特徴とする変換方法。

$$Z_2[p] = (0 - (x[N/2+2p+N] + x[N/2-1-2p+N]) - j(x[2p+N] + x[N-1-2p+N])) \\ * (\cos(2\pi / (8N) * (8p+1)) - j \sin(2\pi / (8N) * (8p+1)))$$

【請求項 3 5】

上記離散フーリエ変換は、q を上記第 2 の中間シーケンスの変数とし、p の範囲を 0 から $N/4$ とし、以下の式により上記要素 $Z[p]$ から上記第 2 の中間シーケンスの要素 $z_1[q]$ を算出し、

$$z_1[q] = Z_1[p] * (\cos(2\pi pq / (N/2)) - j \sin(2\pi pq / (N/2)))$$

以下の式により上記要素 $Z[p]$ から上記第 2 の中間シーケンスの要素 $z_2[q]$ を算出することを特徴とする請求項 3 4 記載の変換方法。

$$z_2[q] = Z_2[p] * (\cos(2\pi pq / (N/2)) - j \sin(2\pi pq / (N/2)))$$

【請求項 3 6】

上記離散フーリエ変換は高速フーリエ変換であることを特徴とする請求項 3 4 記載の変換方法。

【請求項 3 7】

上記高速フーリエ変換は、基数 4 高速フーリエ変換であることを特徴とする請求項 3 6 記載の変換方法。

【請求項 3 8】

請求項 1 8 又は請求項 3 4 の変換方法を実行するためのプログラム命令が記録されたコンピュータにより読取可能な記録媒体。

【請求項 3 9】

入力データシーケンスに対する前乗算処理を行い、第 1 の中間シーケンスを生成する前乗算手段と、

上記第 1 の中間シーケンスに対する離散フーリエ変換処理を行い、第 2 の中間シーケンスを生成する離散フーリエ変換手段と、

上記第 2 の中間シーケンスに対する後乗算処理を行い、出力データシーケンスを生成する後乗算手段とを備え、

10

20

30

40

50

上記第1の中間シーケンスの算出は、 n を上記入力データシーケンスの変数とし、 p を上記第1の中間シーケンスの変数とし、 j を虚数単位とし、 N を256として、以下の式に基づいて、上記入力データシーケンスの要素 $x[n]$ から上記第1の中間シーケンスの要素 $Z[p]$ を算出することを特徴とする修正離散コサイン変換を行う修正離散コサイン変換装置。

$$Z[p] = \frac{(x[2p] - x[2N-2p-1]) - (x[N+2p] + x[N-1-2p]) - j(x[2p] + x[2N-1-2p] + (x[N+2p] - x[N-1-2p])) * (\cos(2\pi / (16N) * (8p+1)) - j \sin(2\pi / (16N) * (8p+1)))}{}$$

【発明の詳細な説明】

【0001】

発明の背景

1. 発明の分野

本発明はデジタルオーディオ処理に関し、特にデジタルオーディオのエンコード処理における時間領域エイリアシングを効率的に除去する装置及び方法に関する。

【0002】

2. 背景技術

デジタルビデオディスク(DVD)、デジタル衛星システム(DSS)、デジタルテレビジョン(DTV)等において、デジタルオーディオ信号は広く使用されている。これら全てのシステムにおいて、記録媒体の容量又は帯域に制約があり、この2つの制約が各システムにおける共通の問題となっている。より多くのオーディオデータを容量が制限されている記録媒体に記録し、あるいは帯域幅が制限されているチャンネルを介して伝送するためには、デジタルオーディオ圧縮技術が必要となる。広く使用されている圧縮技術の1つに、知覚に基づくエンコード処理(perceptual encoding)がある。このエンコード処理における人間の聴覚に基づくモデルにより、人間が知覚できない音に対応する情報を削除することができる。

【0003】

アドバンスドテレビジョンシステム委員会(Advanced Television System Committee: ATSC)は、以前よりHDTVとして知られていたデジタルテレビジョン(DTV)方式において使用する知覚に基づくエンコード処理として、ドルビー研究所(Dolby(商標)Labs)の技術を採用した。この技術については、オーディオ圧縮バージョン3(Audio Compression version 3: AC-3)仕様書ATSC A/52(以下、AC-3仕様書という。)に記述されており、この記述は参照により本願に組み込まれるものとする。この後、AC-3仕様書は、第1地域(Region 1; 北アメリカ市場)のDVD及びDSS放送にも採用されている。

【0004】

AC-3仕様書に基づき、デジタルオーディオ信号用の標準的なデコーダを設計することができる。これにより、AC-3方式でエンコードされたデジタルオーディオ録音データを異なる製造業者の再生装置間で再生することができる。これに対して、AC-3エンコード処理における詳細については、AC-3仕様書において細かく定められてはいない。エンコーダ側は、AC-3規格のシンタックスに適合するビットストリーム、すなわち、デコード処理により、アプリケーションが意図する十分な品質のオーディオ信号が生成されるビットストリームを生成すればよい。このため、エンコーダの設計における詳細事項は、標準的に設計されたデコーダにより再生されるエンコードされたデジタルオーディオデータの品質に影響を与えない限り、各設計者の裁量により決定できるものが多い。オーディオデータは、通常、時間領域より周波数領域において圧縮した方が効率的である場合が多い。時間領域から周波数領域への変換の一手法として、修正離散コサイン変換(modified discrete cosine transform: 以下、MDCTという。)がある。MDCTは、離散変数の関数に作用する離散フーリエ変換の一種である。MDCTは、通常、時間領域データサンプルと呼ばれる離散変数の入力データシーケンスを周波数領域係数と呼ばれる離散変数の出力データシーケンスに変換する。時間領域データサンプルは、入力されてくるオ

10

20

30

40

50

オーディオデータを離散的な各時刻で測定して得られた値を示している。周波数領域係数は、離散的周波数値における対応する信号強度を表している。

【0005】

エンコードされたオーディオデータをデコードして再生する際、音が忠実に再現されるように、AC-3仕様書では、時間領域エイリアシング除去(time-domain aliasing cancellation: 以下、TDACという。)の手法を採用している。TDAC法により、エンコードされたオーディオデータをデコードして再生する際、オリジナルのオーディオデータを略完全に再構築することができる。TDAC法は、2つの処理を含んでいる。すなわち、窓係数による乗算を用いた、適切に選択された窓処理(window operation)及び、それに続くMDCTである。

10

【0006】

知覚に基づくエンコード処理の設計においては、MDCTのブロック長と呼ばれる、MDCT処理の位置時刻において変換されるデジタルサンプルの数の設定が重要である。遷移(transient)(時間領域サンプルのシーケンスにおける値の細かい変動)が観察されない場合、ブロックサイズ切替フラグblkswは、0に設定され、TDAC用に設計されたAC-3デコーダは、512サンプルの長ブロックMDCT演算(long-block MDCT calculation)に切り換えられる。一方、遷移が観察された場合、ブロックサイズ切替フラグblkswは、1に設定され、AC-3エンコーダは、256サンプルの短ブロックMDCT演算(short-block MDCT calculation)に切り換えられる。ブロック長が長い場合、周波数分解能は高まるが、時間分解能は低くなる。長ブロック変換は、通常、信号が比較的安定している場合に適用される。短ブロック変換は、信号が比較的不安定である場合、プレエコー効果(pre-echoing effect)を抑制するために適用される。すなわち、単一のMDCTブロック長を用いるのではなく、512サンプルからなるMDCTブロック長及び256サンプルからなるMDCTブロック長を切り換えて用いることにより、オーディオデータの状況に応じて、最大限に忠実な再生が可能となる。

20

【0007】

AC-3仕様書には、エンコーダにおけるMDCT演算のための基本式が示されている。しかしながら、基本式をMDCT演算に直接用いると、大量の処理電力が必要となってしまうため、実用的でコスト効率の良好な処理を行うエンコーダの実現が困難である。すなわち、効率的なAC-3エンコーダを設計するためには、異なるブロック長を用いるMDCTにおける演算を最適化する必要がある。

30

【0008】

発明の概要

本発明はデジタルオーディオエンコード処理における効率的な時間領域エイリアシング除去を行うための装置及び方法を提供する。本発明の具体例においては、本発明は、改良された修正離散コサイン変換(modified discrete cosine transform: MDCT)法により、ドルビー(Dolby: 商標)AC-3フォーマットのデジタルオーディオ信号に対する知覚に基づくエンコード圧縮処理を行う。また、この変形として、改良されたMDCT法を他の知覚に基づくエンコード処理に適用してもよい。

【0009】

本発明の一具体例では、複素前乗算処理(complex-valued premultiplication)及び複素後乗算処理(complex-valued postmultiplication)により、データサンプルを準備及び処理し、これにより長ブロック変換処理及び短ブロック変換処理の両方を効率的に実行することができる。前乗算処理及び後乗算処理は、離散フーリエ変換(discrete Fourier transformation: DFT)とともに作用するように周到に構築され、MDCTを直接計算した場合と同様の数値的な結果を得ることができる。さらに、複素前乗算処理、DFT、複素後乗算処理の組み合わせは、MDCTを直接計算した場合に比べて必要な演算量が著しく少ない。このため、本発明によれば、演算のための消費電力が少なく、したがって実際の製造コストが安いカスタム向けのデジタル信号プロセッサ(DSP)が実現される。

40

【0010】

50

好適な実施の形態の詳細な説明

本発明は、デジタル信号処理の改良に関する。以下の説明により、当該技術分野の専門家は、本発明を製造及び実施することができる。また、以下の説明は、特許出願のコンテキストに沿い、必要な技術を開示するものである。以下の説明において、本発明は、エンコーダ/デコーダ(CODEC)集積回路により実現されるオーディオ圧縮バージョン3(Audio Compression version 3:以下、AC-3という。)フォーマットによる知覚に基づくデジタルオーディオエンコード処理に適用される。しかしながら、本発明は、時間領域から周波数領域へのデータ変換において実行される時間領域エイリアシング除去(time-domain aliasing cancellation:以下、TDACという。)処理を用いるエンコード処理であれば、いかなる形式のものにも適用できる。当該技術分野の専門家は、以下に開示する実施の形態の様々な変形例を想到することができ、本発明の包括的な原理は、このような変形例にも適用される。すなわち、本発明は、以下の実施の形態に限定されるものではなく、以下の説明に対応する原理及び特徴の最も広い範囲に相当するものと解釈される。

10

【0011】

本発明の具体例において、本発明は、TDACによる知覚に基づくデジタルオーディオデータの圧縮処理を採用した修正離散コサイン変換(modified discrete cosine transform:以下、MDCTという。)を実現する効率的な装置及び方法を提供する。知覚に基づくエンコード処理は、経験的に判定された人間の聴覚の特性を利用して、人間の耳によっては知覚できない音に対応する情報を削除することによりオーディオデータを圧縮する。通常の処理においては、時間領域データサンプルのデジタルオーディオ入力データシーケンスは、離散フーリエ変換の手法を用いて、周波数領域係数の出力データデータシーケンスに変換される。一具体例においては、AC-3エンコーダは、MDCTによりこの変換を実現する。

20

【0012】

AC-3仕様書には、MDCT処理に必要な演算のための式が開示されている。しかしながら、この演算を直接実行するためには、相当大きな電力が必要となる。本発明の一具体例においては、MDCT変換回路(MDCT transformer)を用いる。このMDCT変換回路により、AC-3仕様書に開示される式を直接用いた演算処理により得られる結果と同様の結果を得ることができる。MDCT変換回路における演算は、3つのステップからなる。すなわち、複素前乗算ステップ(complex-valued premultiply step)、複素高速フーリエ変換(fast Fourier transform:以下、FFTという。)ステップ、及び複素後乗算ステップ(complex-valued postmultiply step)である。複素前乗算ステップにより、入力デジタルオーディオサンプルは、非常に効率的な複素FFT処理を行うために必要な形式に変換される。FFTによる変換処理に続いて、複素後処理ステップにより、FFT処理されたデータの実数部と虚数部が分離される。この実数部と虚数部は、AC-3仕様書に開示される式を直接演算した場合に得られる結果と等しい結果を示す。

30

【0013】

Fig. 1は、本発明に基づく読出/書込DVDプレイヤー100の構成を示す図である。この具体例において、読出/書込DVDプレイヤー100は、DVD102と、ヘッドアーム部材(head-arm assembly)104と、駆動制御回路(drive control electronics)106と、マルチプレクサ/デマルチプレクサ108と、モーションピクチャエキスパートグループ(Motion Picture Experts Group:以下、MPEGという。)ビデオコーデック110と、AC-3オーディオコーデック120と、制御プロセッサ130と、操作者制御及びディスプレイ132とを備える。この具体例において、DVD102は、オーディオ情報及びビデオ情報のデジタルデータが記録された光ディスクである。読出/書込DVDプレイヤー100において、DVD102は回転駆動され、駆動制御回路106は、ヘッドアーム部材を制御して、DVD102に記録されているデータを読み取る。駆動制御回路106は、ヘッドアーム部材を介して読み取った情報から、オーディオ情報とビデオ情報が結合されたデジタルビットストリームを抽出し、この結合されたマルチプレクサ/デ

40

50

マルチプレクサ108に供給する。

【0014】

マルチプレクサ/デマルチプレクサ108は、信号線114を介して供給されてくる結合されたデジタルビットストリームからオーディオビットストリーム及びビデオビットストリームを分離する。ビデオビットストリームは、例えば、MPEG-2フォーマットに準拠するものであり、MPEGビデオコーデック110に供給されて処理される。DVDに記録されたビデオデータがデコードされると、デコードされたデータはアナログフォーマットに変換され、ディスプレイ又は外部ビデオモニタ装置に出力される。また、外部データ源から供給されるビデオ入力データもMPEGビデオコーデック110によりエンコードされ、マルチプレクサ/デマルチプレクサ108を介してDVD102に記録される。

10

【0015】

この具体例においては、信号線114を介してマルチプレクサ/デマルチプレクサ108に供給される結合されたデジタルビットストリームにエンコードされているオーディオデータは、AC-3オーディオデータである。信号線114を介してDVD102に書き込まれ、及びDVD102から読み出されるAC-3オーディオデータは、6つの定義されたオーディオチャンネルを有する。すなわち、5つのフルバンドチャンネル(fbwチャンネル)及び1つの低周波効果チャンネル(lfeチャンネル)である。

【0016】

DVD102からのデータ再生処理において、AC-3コーデック120には、マルチプレクサ/デマルチプレクサ108からAC-3オーディオデータが供給され、AC-3コーデック120は、このAC-3オーディオデータをデコードして、線形パルスコード変調(linear pulse-code-modulation: 以下、LPCMという。)オーディオデータを生成する。LPCMオーディオデータは、さらにアナログ信号に変換され、増幅器及びラウドスピーカを備えるオーディオ装置により再生される。

20

【0017】

DVD102に対する記録処理においては、AC-3コーデック120には、入力LPCMデータが供給され、AC-3コーデック120は、この入力LPCMデータをAC-3フォーマットにエンコードする。このエンコード処理について、Fig. 2を用いて詳細に説明する。

【0018】

Fig. 2は、本発明に基づくFig. 1に示すAC-3コーデックの構成を示す図である。この具体例においては、AC-3コーデック120は、AC-3デコーダ200と、AC-3エンコーダ218とを備える。

30

【0019】

AC-3デコーダ200の設計に関する詳細については、AC-3仕様書に記載されており、AC-3仕様書は、参照により本発明に組み込まれるものとする。このFig. 2に示す具体例では、入力されてくるマルチチャンネルAC-3ビットストリームは、まずデマルチプレクサ202に供給され、デマルチプレクサ202は、ビットストリームデータの1フレームに相当するデータを一時的にバッファリングする。各フレームは、デジタルオーディオデータの最大6つの離散チャンネル毎の256個の周波数領域係数に関する圧縮された情報を含んでいる。デマルチプレクサ202は、圧縮制御データ(compression control data)から、圧縮されている周波数領域係数(オーディオデータ)を分離する。ビットアロケータ212は、圧縮制御データを用いて、圧縮された周波数領域係数をどのように解凍(decompress)すべきかを判定する。この解凍処理は、逆量子化回路204により実行される。逆量子化回路204は、解凍した周波数領域係数を逆MDCT変換回路206に供給する。逆MDCT変換回路206及び窓/オーバーラップ加算回路208における処理により、周波数領域係数は、時間領域サンプルに変換される。時間領域サンプルは、出力バッファ210を介して、出力に適した形式で出力される。

40

【0020】

AC-3エンコーダ218の構成については、AC-3仕様書には詳細な記述がない。A

50

C - 3仕様書には、汎用的な説明とアルゴリズム、及び標準的なAC - 3デコーダ200により出力AC - 3が確実にデコードされるために必要である場合に限り、詳細事項が記載されている。本発明の具体例では、AC - 3エンコーダ218の主回路ブロックは、入力バッファ220、3Hz高域通過フィルタ222、ブロックサイズ制御回路224、窓処理回路(windower)228、MDCT変換回路230、サブバンドブロック浮動小数点(floating point:以下、FPという。)変換回路236、量子化回路238、ビットアロケータ240、マルチプレクサ242を備える。

【0021】

入力バッファ220は、入力されてくるLPCMデジタルオーディオデータをバッファリングし、3Hz高域通過フィルタは、カットオフ周波数3Hz以下の成分をフィルタリングする。ブロックサイズ制御回路224は、遷移成分(transient content)(時間領域サンプルのシーケンスにおける値の細かい変動)を検出し、窓処理回路228及びMDCT変換回路230により実行される時間領域エイリアシング除去処理をサポートする。すなわち、ブロックサイズ制御回路224は、十分な遷移成分を検出すると、ブロック切換フラグblkswを1に設定し、これにより、MDCT変換回路230に、1つの長ブロックではなく、短ブロックの組に対して処理をするよう指示する。

10

【0022】

デジタルサンプルは、入力バッファ220から、3Hz高域通過フィルタ222及び窓処理回路228を介してMDCT変換回路230に供給される。窓処理回路230は、デジタルサンプルの入力ブロックにフィールド窓(Fielder's window)(AC - 3仕様書参照)を乗算し、変換境界効果(transform boundary effect)を低減させ、これにより周波数選択性を向上させる。窓処理回路228による窓処理の後、デジタルサンプルは、MDCT変換回路230において時間領域から周波数領域に変換される。

20

【0023】

AC - 3仕様書には、要求されるMDCT処理のための次のような式が開示されている。

【0024】

【数1】

式1A 長ブロック変換用

$$X_D[k] = \frac{-1}{N} \sum_{n=0}^{2N-1} x[n] \cos\left(\frac{\pi}{4N}(2n+1)(2k+1) + \frac{\pi}{4}(2k+1)\right)$$

30

ここで、 $0 \leq k < N$

【0025】

【数2】

式1B 短ブロック変換用

$$X_D[k] = \frac{-2}{N} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi}{4N}(2n+1)(2k+1) + \frac{\pi}{4}(2k+1)(1+\alpha)\right)$$

ここで、 $0 \leq k < N/2$

第1の短ブロック変換において、 $\alpha = -1$

第2の短ブロック変換において、 $\alpha = +1$

10

【0026】

式1A及び式1Bを用いた変換処理により、窓処理された時間領域サンプル $x[n]$ は、周波数領域係数 $X_D[k]$ に変換される。式1A及び式1Bにおいて、長ブロック変換の場合も短ブロック変換の場合も、 $N = 256$ である。なお、周波数領域係数の個数は、時間領域サンプルの個数の半分となる。

【0027】

式1A又は式1Bに示される処理の全てを実行して、シーケンス $X_D[k]$ を直接算出することは可能であるが、このような処理は極めて効率が悪い。式1A及び式1Bを直接計算した場合、演算処理は $O[N^2]$ 、すなわち N^2 桁の複雑性を有することとなる。本発明の具他例においては、中間シーケンス $Z[p]$ 及び $z[q]$ を算出する。これにより、シーケンス $X_D[k]$ を算出するための演算処理の複雑性は、 $O[N \log_2 N]$ に低減される。複素前乗算ステップは、 $x[n]$ を $Z[p]$ に変換する。高速フーリエ変換(FFT)として実現されるDFTは、 $Z[p]$ を $z[q]$ に変換する。そして、複素後乗算ステップは、 $z[q]$ を $X_D[k]$ に変換する。この3つのステップの詳細については、Fig. 4を用いて後に詳細に説明する。

20

【0028】

MDC T変換回路230により時間領域サンプルが周波数領域係数に変換された後、サブバンドブロックFP変換回路236は、周波数領域係数を浮動小数点データ(floating-point representation)に変換する。この浮動小数点データは、指数データ及び仮数データを含んでいる。サブバンドブロックFP変換回路236は、指数データをビットアロケータ240に供給し、仮数データ238を量子化回路238に供給する。量子化回路238は、ビットアロケータ240からの出力信号に基づいて、仮数データを量子化する。ビットアロケータ240及び量子化回路238は、マスキング閾数を超える音にのみビットを割り当て、有限数のビットにデータを量子化することにより、実際のデータ圧縮処理を行う。この処理により、人間の耳には感知されない音がデータビットに割り当てられるようなことはなくなる。量子化誤差が人間の耳に感知されない範囲における限界レベルで量子化を行うことにより、さらに圧縮効率を高めることができる。圧縮された周波数領域係数は、マルチプレクサ242に供給され、マルチプレクサ242は、この周波数領域係数をAC-3フレームにパッキングする。生成されたAC-3フレームは、マルチプレクサ242からエンコーダ218の外部に出力される。

30

40

【0029】

Fig. 3は、本発明に基づくサンプル変換処理及び時間領域エイリアシング除去処理のタイミングチャートである。この具体例においては、LPCMフォーマットの6つの独立したチャンネルのデジタルオーディオデータが供給される。Fig. 3では、説明を簡潔に行うため、チャンネル1に対応するデジタルデータシーケンスのみを示している。Fig. 3に示す番号付けされた各ブロックは、それぞれ512個のデジタルオーディオサンプルを含んでいる。6つの独立したチャンネルを有するこの具体例において、チャンネル1のブロックは、他のチャンネル(図示せず)のブロックとインターリーブされている。

50

【 0 0 3 0 】

F i g . 3 に示す具体例において、ブロックサイズ制御回路 2 2 4 は、遷移成分 (transient contents) が十分高いか否かを判定するにあたり、複数の判定基準を用いる。これら判定基準の 1 つにおいて、ブロックサイズ制御回路 2 2 4 は、ブロック後半 (second half) の遷移成分を検査する。様々な判定基準の結果、遷移成分が十分高いと判定された場合、ブロック切換フラグ blksw は、1 に設定される。F i g . 3 に示す具体例において、ブロック 1 (3 1 0) の遷移成分は十分高くないと判定され、したがってブロック切換フラグ blksw[1] は、0 (3 1 4) に設定される。これにより、M D C T 変換回路 2 3 0 は、現在のブロック 1 (3 1 0) に対して長ブロック変換処理 3 0 4 を実行する。ブロック 2 (3 2 4) に対しても同様の処理が行われる。

10

【 0 0 3 1 】

ブロック 3 (3 4 0) に対する処理において、ブロックサイズ制御回路 2 2 4 は、遷移成分が十分高いと判定し、したがってブロック切換フラグ blksw[1] は、1 (3 3 0) に設定される。ブロック切換フラグ blksw[1] が 1 に設定されると、M D C T 変換回路 2 3 0 はこれを認識し、現在のブロック 3 (3 4 0) に対して、2 個 1 組の短ブロック変換処理 3 3 2 , 3 2 4 を実行する。

【 0 0 3 2 】

後続する一連のブロックに対し、ブロックサイズ制御回路 2 2 4 は、バッファリングされたブロックにおける遷移成分を検査し、この検査の結果に応じて、ブロック切換フラグ blksw[1] を適切に設定する。このようにして、変換ブロックの長さは、略リアルタイムに連続的に調整される。これにより、ブロック長が不適切である場合に引き起こるブレイク効果の発生が抑制される。

20

【 0 0 3 3 】

F i g . 4 A は、本発明に基づく高速計算修正離散コサイン変換 (fast computational modified cosine transform) を行う F i g . 2 に示す M D C T 変換回路 2 3 0 の内部構成の具体例を示す図である。F i g . 4 A に示す M D C T 変換回路 2 3 0 は、デジタル信号処理 (digital signal processor: 以下、D S P という。) コア 4 0 0 と、読出 / 書込ランダムアクセスメモリ (以下、R A M という。) 4 1 0 と、プログラム可能読出専用メモリ (P R O M) 4 2 0 とを備える。この具体例において、D S P コア 4 0 0 は、3 つのソフトウェアモジュールを実行し、これにより T D A C のための M D C T 変換を効率的に実現するための 3 つの処理ステップを制御する。これら 3 つのソフトウェアモジュールとは、前乗算モジュール (premultiplier) 4 3 0、D F T モジュール 4 4 0、後乗算モジュール (postmultiplier) 4 5 0 である。F i g . 4 A に示す具体例において、前乗算モジュール 4 3 0 は、デジタルオーディオサンプルに対する乗算及び処理を行う。これにより、デジタルオーディオサンプルは、D F T 4 4 0 により処理できるデータとなる。前乗算モジュール 4 3 0 から供給され、D F T 4 4 0 により変換処理されたデータは、後乗算モジュール 4 5 0 によりさらに処理され、これにより、上述した式 1 A 及び式 1 B を直接演算して得られるデータと同等のデータが得られ、すなわち標準的な A C - 3 デコーダに互換性を有するデータが得られる。

30

【 0 0 3 4 】

前乗算モジュール 4 3 0、D F T モジュール 4 4 0、後乗算モジュール 4 5 0 における基本的な処理の概要を以下、擬似コードとして示す。ここに示す擬似コードは、説明のため、汎用のプログラミング言語を用いて書かれたソースコードとして示すが、これは例示的なものであり、特定のコンパイラによりコンパイルすることを限定するものではない。ここでは、例示的に、擬似コードを C プログラミング言語のフォーマット及び定義に基づいて記述する。下記のコード例 1 は、長ブロック変換のための前乗算モジュール 4 3 0 の擬似コードの具体例である。

40

【 0 0 3 5 】

【 数 3 】

コード例 1

```

for (p=0; p<N/2; p++)
{
    Z[p] = ((x[2p] - x[2N - 2p - 1]) - (x[N + 2p] + x[N - 1 - 2p]) -
            j(x[2p] + x[2N - 1 - 2p] + (x[N + 2p] - x[N - 1 - 2p])) *
            (cos(2 π/(16N) * (8p + 1)) - jsin(2 π/(16N) * (8p + 1))));
}

```

10

【 0 0 3 6 】

ここで、p は出力シーケンス Z [p] における変数を表し、j は虚数単位を表し、N = 256 であり、x [n] は窓処理された入力サンプルを表す。なお、出力シーケンス Z [p] は、N / 2 = 128 個の複素値要素 (complex-valued elements) を有する。

【 0 0 3 7 】

下記のコード例 2 は、短ブロック変換のための前乗算モジュール 430 の擬似コードの具体例である。この具体例において、前乗算モジュール 430 は、第 1 の短ブロックと第 2 の短ブロックの両方を同時に処理し、第 1 の短ブロックに対応する出力シーケンス Z 1 [p] と、第 2 の短ブロックに対応する出力シーケンス Z 2 [p] を生成する。

20

【 0 0 3 8 】

【 数 4 】

コード例 2

```

for (p=0; p<N/4; p++)
{
    Z1[p] = ((x[2p] - x[N-1-2p]) + j(x[N/2-1-2p] - x[N/2+2p] - x[N/2+2p])) *
            (cos(2 π/(8N) * (8p + 1)) - jsin(2 π/(8N) * (8p + 1)));
    Z2[p] = (0-(x[N/2+2p+N] + x[N/2-1-2p+N]) - j(x[2p+N] + x[N-1-2p+N])) *
            (cos(2 π/(8N) * (8p + 1)) - jsin(2 π/(8N) * (8p + 1)));
}

```

30

【 0 0 3 9 】

ここで、p は出力シーケンス Z 1 [p] 及び Z 2 [p] における変数を表し、j は虚数単位を表し、N = 256 であり、x [n] は窓処理された入力サンプルを表す。なお、サブシーケンス Z 1 [m] 及び Z 2 [m] は、それぞれ 64 (43) 個の要素を含み、したがって、各サブシーケンスは 4 を基数とする F F T (radix-4 FFT) による変換処理に適合している。

40

【 0 0 4 0 】

前処理モジュール 430 が入力シーケンス x [n] を Z [p] に変換した後、D F T 440 は、この Z [p] に対する変換処理を行う。長ブロック変換処理の場合、D F T 440 は、Z [p] の 128 個の要素を中間シーケンス z [q] の 128 個の要素に変換する。短ブロック変換処理の場合、D F T 440 は、Z 1 [p] の 64 個の要素を z 1 [q] の

50

64個の要素に変換し、 $Z[2p]$ の64個の要素を $z[2q]$ の64個の要素に変換する。

【0041】

下記のコード例3は、長ブロック変換のためのDFTモジュール440の擬似コードの具体例である。

【0042】

【数5】

コード例3

```

for(q=0; q<N/2; q++)
{
    z[q] = 0;
    for(p=0; p<N/2; p++)
    {
        z[q] += Z[p] * (cos(2 * pi * pq/(N/2)) - jsin(2 * pi * pq/(N/2)));
    }
}

```

10
20

【0043】

ここで、 p は複素入力シーケンス $Z[p]$ における変数を表し、 q は複素出力シーケンス $z[q]$ における変数を表し、 $N=256$ であり、 j は虚数単位を表す。 $z[q]$ の実数部と虚数部を $z[q] = z_r[q] + j z_i[q]$ のように表すと有効である。

【0044】

下記のコード例4は、短ブロック変換のためのDFTモジュール440の擬似コードの具体例である。この具体例において、DFTモジュール440は、第1の短ブロックと第2の短ブロックの両方を同時に処理し、第1の短ブロックに対応する出力シーケンス $z_1[q]$ と、第2の短ブロックに対応する出力シーケンス $z_2[q]$ を生成する。

【0045】

【数6】

コード例4

```

for(q=0; q<N/4; q++)
{
    z1[q]=z2[q]=0;
    for (p=0; p<N/4; p++)
    {
        z1[q] += Z1[p] * (cos(2 * pi * pq/(N/4)) - jsin(2 * pi * pq/(N/4)));
        z2[q] += Z2[p] * (cos(2 * pi * pq/(N/4)) - jsin(2 * pi * pq/(N/4)));
    }
}

```

40

【0046】

50

ここで、 p は複素入力シーケンス $Z[p]$ における変数を表し、 q は複素出力シーケンス $z[q]$ における変数を表し、 $N = 256$ であり、 j は虚数単位を表す。

【0047】

Fig. 4 に示す具体例では、DFTモジュール440がシーケンス $Z[p]$ を $z[q]$ に変換した後、後乗算モジュール450が $z[q]$ に対する処理を行う。長ブロック変換の場合、後乗算モジュール450は、 $z[q]$ の128個の要素を複素シーケンス $y[k]$ の128個の要素に変換する。 $y[k]$ の要素の実数部と虚数部は、分離及びシャッフルされ、これにより実数として表される最終的な出力シーケンス $XD[k]$ の256個の要素が生成される。短ブロック変換処理の場合、後乗算モジュール450は、 $z1[q]$ の64個の要素を複素シーケンス $y1[k]$ の64個の要素に変換し、 $z2[q]$ の64個の要素を複素シーケンス $y2[k]$ の64個の要素に変換する。 $y1[k]$ の要素の実数部と虚数部は、分離及びシャッフルされ、これにより実数の最終的な出力シーケンス $X1D[k]$ の128個の要素が生成される。また、 $y2[k]$ の要素の実数部と虚数部は、分離及びシャッフルされ、これにより実数として表される最終的な出力シーケンス $X2D[k]$ の128個の要素が生成される。

10

【0048】

下記のコード例5は、長ブロック変換のための後乗算モジュール450の擬似コードの具体例である。

【0049】

【数7】

20

コード例5

```
for(k=0; k<N/2; k++)
```

```
{
```

$$y[k] = (-1)^{\{k\}/(\sqrt{2})} * z[k] * (\cos(2\pi/(16N) * (8k + 1)) - j \sin(2\pi/(16N) * (8k + 1)));$$

```
}
```

30

【0050】

ここで、 k は出力シーケンス $y[k]$ における変数を表し、 $N = 256$ であり、 j は虚数単位を表す。

【0051】

実数として表される最終的な出力シーケンス $XD[k]$ は、複素シーケンス $y[k]$ の実数部及び虚数部を分離及びシャッフルして得られるものであり、ここで、 $y[k] = y_r[k] + j y_i[k]$ である。 k が偶数である場合、 $XD[k] = y_r[k/2]$ であり、 k が奇数である場合、 $XD[k] = y_i[N/2 - 1 - (k - 1)/2]$ である。

【0052】

40

下記のコード例6は、短ブロック変換のための後乗算モジュール450の擬似コードの具体例である。この具体例において、後乗算モジュール450は、第1の短ブロックと第2の短ブロックの両方を同時に処理し、第1の短ブロックに対応する出力シーケンス $X1D[k]$ と、第2の短ブロックに対応する出力シーケンス $X2D[k]$ を生成する。

【0053】

【数8】

コード例 6

```

for(k=0; k<N/2; k++)
{
    y1[k]= z1[k] * (cos(2 π/(8N) * (8k + 1)) -jsin(2 π/(8N) * (8k + 1)));
    y2[k]= z2[k] * (cos(2 π/(8N) * (8k + 1)) -jsin(2 π/(8N) * (8k + 1)));
}

```

10

【 0 0 5 4 】

ここで、 k は出力シーケンス $y_1[k]$ 及び $y_2[k]$ における変数を表し、 $N = 256$ であり、 j は虚数単位を表す。

【 0 0 5 5 】

実数として表される最終的な出力シーケンス $X_1D[k]$ は、複素シーケンス $y_1[k]$ の実数部及び虚数部を分離及びシャッフルして得られるものであり、ここで、 $y_1[k] = y_{1r}[k] + jy_{1i}[k]$ である。 k が偶数である場合、 $X_1D[k] = y_{1r}[k/2]$ であり、 k が奇数である場合、 $X_1D[k] = y_{1i}[N/4 - 1 - (k - 1)/2]$ である。同様に、実数として表される最終的な出力シーケンス $X_2D[k]$ は、複素シーケンス $y_2[k]$ の実数部及び虚数部を分離及びシャッフルして得られるものであり、ここで、 $y_2[k] = y_{2r}[k] + jy_{2i}[k]$ である。 k が偶数である場合、 $X_2D[k] = y_{2r}[k/2]$ であり、 k が奇数である場合、 $X_2D[k] = y_{2i}[N/4 - 1 - (k - 1)/2]$ である。

20

【 0 0 5 6 】

Fig. 4 に示す具体例により生成される実数として表される最終的な出力シーケンス $X_1D[k]$ は、上述した式 1 A 及び式 1 B (AC-3 仕様書に基づく) を用いた直接計算により得られる $X_1D[k]$ に一致する。

【 0 0 5 7 】

Fig. 4 B は、本発明に基づく、Fig. 2 に示す MDCT 変換回路 230 の変形例を示す図である。Fig. 4 B に示す本発明の変形例では、コード例 3 及びコード例 4 に示す離散フーリエ変換 (DFT) が高速フーリエ変換 (FFT) に置換されている。(高速フーリエ変換という用語は、ジェイ・ダブリュー・クーレイ (J.W. Cooley) 及びジェイ・ダブリュー・トゥキー (J.W. Tukey) により提唱された離散フーリエ変換のための一連の効率的なアルゴリズムを集合的に指すものとする。) DFT を FFT に置換することにより、MDCT 処理における計算の複雑性は、 $O(N^2)$ から $O(N \log^2 N)$ に低減される。

30

【 0 0 5 8 】

DFT 処理のための効率的な FFT アルゴリズムは、計算をより小さな DFT 計算に分割する。この計算の分割は、全ての FFT アルゴリズムの基本的な原理である。64 点 (26 又は 43 に相当する) の DFT 計算は、6 段からなる 2 点の DFT 計算又は 3 段からなる 4 点の DFT 計算に分割される。6 段からなる 2 点の DFT 計算は、基数 2 FFT アルゴリズム (radix-2 FFT algorithm) と呼ばれる。3 段からなる 4 点の DFT 計算は、基数 4 FFT アルゴリズム (radix-4 FFT algorithm) と呼ばれる。本発明においては、基数 4 FFT アルゴリズムは基数 2 アルゴリズムに比べて計算の複雑性が低いため、基数 4 アルゴリズムを用いることが好ましい。一般的に、基数が大きいほど、FFT における対称性の効果が発揮される。この対称性と、計算処理段の少なさのため、基数 4 FFT アルゴリズムは基数 2 FFT アルゴリズムより効率的である。

40

【 0 0 5 9 】

Fig. 4 B に示す具体例では、Fig. 4 A の DFT 440 が FFT 460 に置き換えられている。上述のとおり、TDAC 変換されるブロックの長さは、長ブロック変換の場

50

合 5 1 2 (2 9)、短ブロック変換の場合 (2 8 = 4 4) である。F i g . 4 B に示す具体例においては、前乗算モジュール 4 3 0 は、入力デジタルオーディオサンプル $x [n]$ を処理して、新たなシーケンス $Z [p]$ に変換する。長ブロック変換の場合、シーケンス $Z [p]$ は、1 2 8 個のサンプルを含んでいる。1 2 8 = 2 × 4 3 であるため、シーケンス $Z [p]$ は、2 個 1 組の基数 4 変換に縦続 (cascade) 接続された基数 2 変換により変換できる。短ブロック変換のためのサブシーケンス $Z 1 [p]$ 、 $Z 2 [p]$ は、それぞれ 6 4 個のサンプルを含んでいる。6 4 = 4 3 であるため、短ブロック変換のサブシーケンス $Z 1 [p]$ 、 $Z 2 [p]$ に対しては、より効率的な基数 4 F F T を行うことができる。

【 0 0 6 0 】

下記のコード例 7 は、長ブロック変換のための F F T モジュール 4 6 0 の擬似コードの具体例である。ここで、コード例 7 における関数 `FFT_radix4_128` は、関数 `FFT_radix4_64` を 2 回呼び出すことにより 2 個 1 組の基数 4 F F T に縦続接続された基数 2 F F T を用いる。関数 `FFT_radix4_64` の具体例は、コード例 8 として後に示す。

【 0 0 6 1 】

【 数 9 】

コード例 7

```

/** 128 点 FFT */
void FFT_radix4_128()
{
    long x[2],y[2];
    long X[2],Y[2];
    adr0 = 0;
    adr2 =64;
    /** 基数 2 変換 **/
    for(j = 0; j < 2; j++)
    {
        for(i = 0; i < 32; i++)
        {
            x[0] = R[adr0]; y[0] = I[adr0];
            x[1] = R[adr2]; y[1] = I[adr2];
            Wx=cos(2*pi*i*j/128);
            Wy=sin(2*pi*i*j/128);
            X[0] = (R[adr0] + R[adr2])/2;
            Y[0] = (I[adr0] + I[adr2])/2;
            X[1] = (R[adr0] - R[adr2])/2 * Wx - (Y[adr0] - Y[adr1])/2 * Wy;
            Y[1] = (R[adr0] - R[adr2])/2 * Wy + (Y[adr0] -Y[adr1])/2 * Wx;
            R[adr0] = X[0];
            I[adr0]=Y[0];
            R[adr2]=X[1];
            I[adr2]=Y[1];
            adr0++;
            adr2++;
        }
    }
    /** 一組の 64 点 FFT **/
    FFT_radix4_64(0, 16, 0);
    FFT_radix4_64(64, 16, 0);
}

```

【 0 0 6 2 】

ここで、 $R[i]$ は $Z[i]$ の実数部を表し、 $I[i]$ は $Z[i]$ の虚数部を表し、 $i = 0, 1, \dots, N/2 - 1$ である。

【 0 0 6 3 】

下記のコード例 8 は、短ブロック変換のための FFT モジュール 4 6 0 の擬似コードの具体例である。コード例 8 において、関数 FFT_radix4_64 は、入力データを含むアレイへの指示を行う関数である。

【 0 0 6 4 】

【 数 1 0 】

コード例 8

/** 64 点 FFT **/

void FFT_radix4_64 (short adr0_par, short off0_par, short adr3_par)

{

long x[4],y[4];

long X[4],Y[4],

10

/** インターフェース **/

adr0 = adr0_par; /* adr0 = 0 or 64 */

off0 = off0_par; /* off0 = 16 */

mod1 = 1;

for(k = 0; k < 3; k++) /* ステージループ */

{

off1 = off0 * 2;

for(j = 0; j < mod1; j++) /* グループループ: 1,4, 16 */

20

{

for(i = 0; i < off0; i++) /* バタフライループ */

{

x[0] = R[adr0+off0*0]; y[0] = I[adr0+off0*0];

x[1] = R[adr0+off0*1]; y[1] = I[adr0+off0*1];

x[2] = R[adr0+off0*2]; y[2] = I[adr0+off0*2];

x[3] = R[adr0+off0*3]; y[3] = I[adr0+off0*3];

30

fft4(&x, &y, &X, &Y); /* 基数 4 バタフライ **/

R[adr0+off0*0] = X[0]; I[adr0+off0*0] = Y[0];

R[adr0+off0*1] = X[1]; I[adr0+off0*1] = Y[1];

R[adr0+off0*2] = X[2]; I[adr0+off0*2] = Y[2];

R[adr0+off0*3] = X[3]; I[adr0+off0*3] = Y[3];

adr0 += 1;

40

}

adr0 += off1;

adr0 += off0;

}

【 0 0 6 5 】

【 数 1 1 】

```

if(k < 2)
{
    for(m=0; m < mod1; m++)    /* mod1: 1 4 */
    {
        for(n=0; n < off0; n++)    /* off0: 16 4 */
        {
            for(i=0; i<4; i++)
            {
                (R[adr1+i*off0+n] +j*I[adr1+i*off0+n]) =
                (R[adr1+i*off0+n] +j*I[adr1+i*off0+n]) * (cos(2*pi*(i*n)/64) +
                j sin(2*pi*(i*n)/64));
            }
        }
        adr1 += mod1*off0;
    }
}

mod1 = mod1 * 4;
off0 = off0 / 4;
}
}

/** 基数 4 バタフライ */
void fft4(long *x, long *y, long *X, long *Y)
{
    /** 実部 */
    *X    = (*x + *(x+1) + *(x+2) + *(x+3))/4;
    *(X+1) = (*x + *(y+1) - *(x+2) - *(y+3))/4;
    *(X+2) = (*x - *(x+1) + *(x+2) - *(x+3))/4;
    *(X+3) = (*x - *(y+1) - *(x+2) + *(y+3))/4;
    /** 虚部 */
    *Y    = (*y + *(y+1) + *(y+2) + *(y+3))/4;
    *(Y+1) = (*y - *(x+1) - *(y+2) + *(x+3))/4;
    *(Y+2) = (*y - *(y+1) + *(y+2) - *(y+3))/4;
    *(Y+3) = (*y + *(x+1) - *(y+2) - *(x+3))/4;
}

```

【 0 0 6 6 】

Fig. 5は、本発明に基づく修正離散コサイン変換の処理手順を示すフローチャートである。このFig. 5に示す手順において、窓処理回路(windower) 228窓処理したデジタルオーディオサンプルのブロックをオーディオチャンネルchからMDC T変換回路230に供給する。ブロックサイズ制御回路224は、オーディオチャンネルch用のブロッ

10

20

30

40

50

ク切換フラグblksw[ch]の値を判定する。チャンネルchのデジタルオーディオサンプルの初期ブロックが窓処理回路228からMDC T変換回路230に供給できる状態になると、処理はステップ500において開始される。

【0067】

MDC T変換回路230は、ステップ510において、窓処理回路228から512個のデジタルオーディオサンプルのブロックを受け取る。MDC T変換回路230は、判定ステップであるステップ514において、blksw[ch]フラグの値を直ちに確認する。blksw[ch]の値が0の場合、MDC T変換回路230は、長ブロック変換処理を行う。長ブロック変換処理は、ステップ518において開始され、このステップ518では、長ブロックに対する前乗算処理により入力シーケンス $x[n]$ が中間シーケンス $Z[p]$ に変換される。MDC T変換回路230は、ステップ520において、DFT処理を実行し、中間シーケンス $Z[p]$ を中間シーケンス $z[q]$ に変換する。そして、MDC T変換回路230は、ステップ524において、長ブロック後乗算処理を実行し、中間シーケンス $z[q]$ を出力シーケンス $XD[k]$ に変換する。

10

【0068】

MDC T変換回路230は、ステップ526において、上述のようにして得られた出力シーケンス $XD[k]$ をサブバンドブロックFP変換回路236に供給する。MDC T変換回路230は、ステップ544において、窓処理回路228内にさらなるデジタルオーディオサンプルのさらなるブロックが存在するか否かを判定する。この判定の結果がNOである場合、MDC T変換回路230は、ステップ550に進み、処理を終了する。一方、この判定の結果がYESである場合、MDC T変換回路230は、ステップ510に戻り、さらなるデジタルオーディオサンプルのブロックを入力し、Fig. 5に示す処理を繰り返す。

20

【0069】

上述の手順は、MDC T変換回路230がステップ514において、blksw[ch]フラグの値が0であると判定した場合の手順である。一方、blksw[ch]の値が1である場合、MDC T変換回路230は、2個1組の短ブロック変換処理を行う。MDC T変換回路230は、ステップ530において、短ブロック前乗算処理を実行し、これにより、入力シーケンス $x[n]$ は1組の中間シーケンス $Z1[p]$ 、 $Z2[p]$ に変換される。MDC T変換回路230は、ステップ534において、分岐DFT (bifurcated DFT)を行い、これにより、中間シーケンス $Z1[p]$ 、 $Z2[p]$ を中間シーケンス $z1[q]$ 、 $z2[q]$ に変換する。そして、MDC T変換回路230は、ステップ538において、短ブロック後乗算処理を実行し、中間シーケンス $z1[q]$ 、 $z2[q]$ を出力シーケンス $X1D[k]$ 、 $X2D[k]$ に変換する。

30

【0070】

MDC T変換回路230は、ステップ540において、上述のようにして得られた出力シーケンス $X1D[k]$ 、 $X2D[k]$ をサブバンドブロックFP変換回路236に供給する。MDC T変換回路230は、ステップ544において、窓処理回路228内にさらなるデジタルオーディオサンプルのさらなるブロックが存在するか否かを判定する。この判定の結果がNOである場合、MDC T変換回路230は、ステップ550に進み、処理を終了する。一方、この判定の結果がYESである場合、MDC T変換回路230は、ステップ510に戻り、さらなるデジタルオーディオサンプルのブロックを入力し、Fig. 5に示す処理を繰り返す。

40

【0071】

以上、本発明の好適な実施の形態を説明した。以上の説明から、当該技術分野の専門家は、この他の実施の形態を容易に想到することができる。例えば、本発明は、上述した実施の形態とは異なる異なる環境において、異なる技術を用いても実現できる。さらに、本発明は、ここで具体的に示したシステムとは異なるシステムにおいても効果的に実現できる。したがって、上述の実施の形態に対するこれらの及びその他の変形例は、本発明の範囲内にあり、本発明の範囲は、添付の請求の範囲によってのみ限定されるものである。

50

【図面の簡単な説明】

【図1】 Fig. 1は、本発明に基づく読出/書込DVD再生装置の具体例を示す図である。

【図2】 Fig. 2は、本発明に基づく、Fig. 1に示すエンコーダ/デコーダ(コーデック)の構成例を示す図である。

【図3】 Fig. 3は、本発明に基づくサンプル変換処理及び時間領域エイリアシング除去処理のタイミングチャートである。

【図4】 Fig. 4Aは、本発明に基づく、図2に示す高速計算を行う修正離散コサイン変換回路の構成例を示す図である。

【図5】 Fig. 4Bは、本発明に基づく、図2に示す高速計算を行う修正離散コサイン変換回路の変形構成例を示す図である。

10

【図6】 Fig. 5は、本発明に基づく修正離散コサイン変換処理の手順を示すフローチャートである。

【図1】

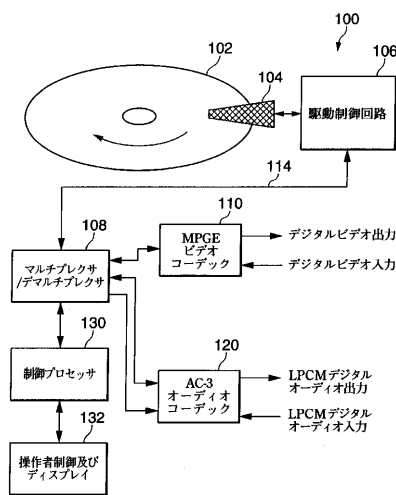


FIG.1

【図2】

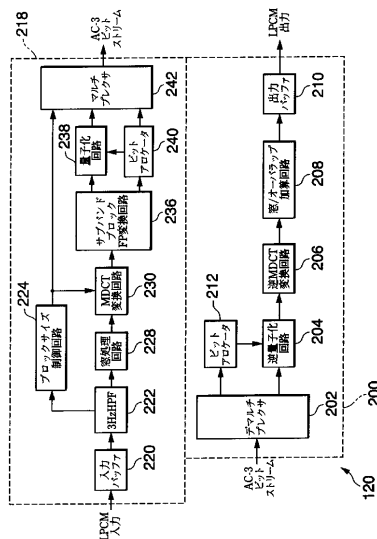


FIG.2

【 図 3 】

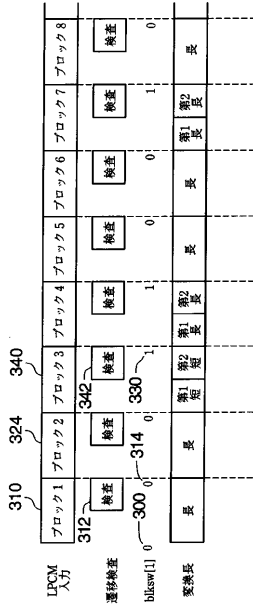


FIG.3

【 図 4 】

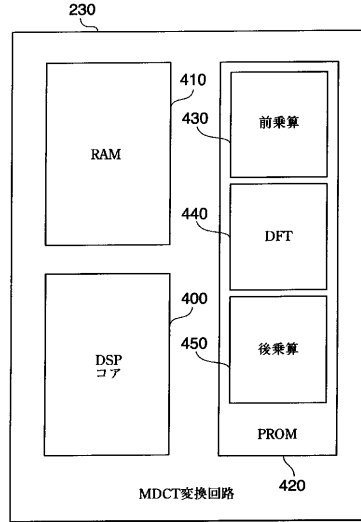


FIG.4A

【 図 5 】

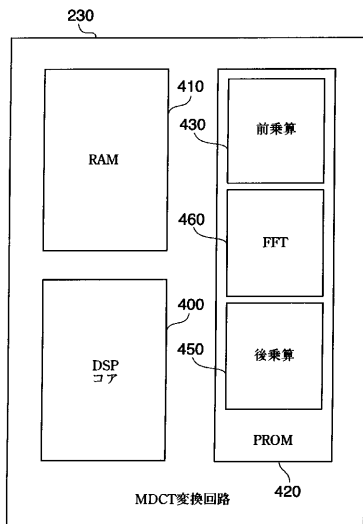


FIG.4B

【 図 6 】

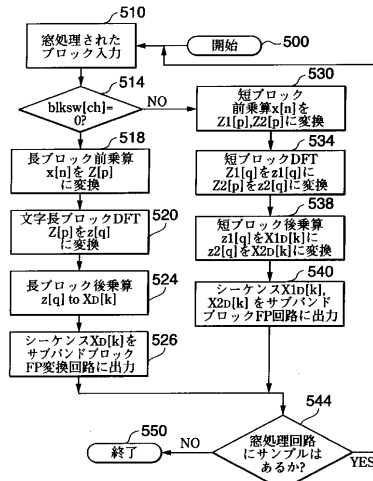


FIG.5

フロントページの続き

審査官 渡辺 未央子

- (56)参考文献 特開平04 - 044099 (JP, A)
特開平06 - 232824 (JP, A)
特開平05 - 183442 (JP, A)
米国特許第05230038 (US, A)
Digital Audio Compression Standard (AC-3), 1995年12月20日, URL, <http://www.sparta.lu.se/bjorn/whitney/references/ac3spec.pdf>