

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第7部門第3区分

【発行日】平成17年10月20日(2005.10.20)

【公開番号】特開2003-143602(P2003-143602A)

【公開日】平成15年5月16日(2003.5.16)

【出願番号】特願2002-228573(P2002-228573)

【国際特許分類第7版】

H 0 4 N 7/24

H 0 3 M 7/30

【F I】

H 0 4 N 7/13 Z

H 0 3 M 7/30 Z

【手続補正書】

【提出日】平成17年7月1日(2005.7.1)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【書類名】明細書

【発明の名称】復号装置、復号方法、復号プログラム、および復号プログラム記録媒体

【特許請求の範囲】

【請求項1】複数の符号化データを入力し、復号して出力するデコード部と、前記デコード部の状態を保持するデコード状態保持部とを備え、

前記デコード部は、前記複数の符号化データから1つを選択し、前記デコード状態保持部に保持されたデコード部の状態のうち、前記選択した符号化データに対応したデコード部の状態に基づいて、前記選択した符号化データを復号し、デコード部の状態を前記デコード状態保持部に保存し、前記復号結果を前記選択した符号化データに対応した出力先に出力する、

ことを特徴とする復号装置。

【請求項2】請求項1記載の復号装置において、

前記デコード部は、符号化単位の境界で符号化データの入力を、復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化する、

ことを特徴とする復号装置。

【請求項3】請求項1記載の復号装置において、

前記デコード部は、各々の符号化データに記された情報に基づいて、復号結果を出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化する、

ことを特徴とする復号装置。

【請求項4】複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、

その復号結果を格納するメモリと、

いずれの符号化データを復号するかを指示するシステム制御部と、

前記メモリに格納した復号結果を出力する出力制御部とを備え、

前記システム制御部の指示に基づいて復号する符号化データを変更し、

前記出力制御部は、前記変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出力を行う、

ことを特徴とする復号装置。

【請求項5】複数の符号化データを入力して、それぞれの符号化データを復号するデ

コード部と、

その復号結果を格納するメモリと、

前記メモリに格納した復号結果を出力する出力制御部とを備え、

前記出力制御部は、各符号化データの復号結果を出力する際に、共通の時間指標に基づいて出力を制御する、

ことを特徴とする復号装置。

【請求項 6】 複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、

その復号結果を格納するメモリと、

前記メモリに格納した復号結果を出力する出力制御部とを備え、

前記出力制御部は、各符号化データの復号結果の出力を制御する際に、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御する、

ことを特徴とする復号装置。

【請求項 7】 請求項 1 記載の復号装置において、

前記デコード部は、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了する、

ことを特徴とする復号装置。

【請求項 8】 請求項 1 記載の復号装置において、

前記デコード部は、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定する、

ことを特徴とする復号装置。

【請求項 9】 請求項 1 記載の復号装置において、

前記デコード部は、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行う、

ことを特徴とする復号装置。

【請求項 10】 請求項 1 記載の復号装置において、

前記デコード部は、各符号化データの復号処理を、復号処理の種類を単位として切り替える、

ことを特徴とする復号装置。

【請求項 11】 請求項 1 記載の復号装置において、

前記デコード部は、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定する、

ことを特徴とする復号装置。

【請求項 12】 複数の符号化データを入力し、復号して出力する復号方法において、デコード部の状態を保持するステップと、

前記複数の符号化データから 1 つを選択するステップと、

保持された前記デコード部の状態のうち、選択された前記符号化データに対応したデコード部の状態に基づいて、選択された前記符号化データを復号するステップと、

復号された結果を選択された前記符号化データに対応した出力先に出力するステップと、

を含むことを特徴とする復号方法。

【請求項 13】 コンピュータに復号方法を実行させる復号プログラムにおいて、

前記復号方法は、複数の符号化データを入力し、復号して出力する復号方法であって、デコード部の状態を保持し、

前記複数の符号化データから 1 つを選択し、

保持された前記デコード部の状態のうち、選択された前記符号化データに対応したデコ

ド部の状態に基づいて、選択された前記符号化データを復号し、  
復号された結果を選択された前記符号化データに対応した出力先に出力する、  
ことを特徴とする復号プログラム。

【請求項 14】 コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、

前記復号方法は、複数の符号化データを入力し、復号して出力する復号方法であって、  
デコード部の状態を保持し、  
前記複数の符号化データから1つを選択し、  
保持された前記デコード部の状態のうち、選択された前記符号化データに対応したデコ  
ド部の状態に基づいて、選択された前記符号化データを復号し、  
復号された結果を選択された前記符号化データに対応した出力先に出力する、  
ことを特徴とする復号プログラム記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

本発明は、復号装置、復号方法、復号プログラム、および復号プログラム記録媒体に関するものであり、特に、デジタルの映像や音声などのデータを圧縮して生成されるビットストリームについて、1つの復号装置で複数の圧縮データを見かけ上同時に復号する処理を可能にした複数データ同時復号装置、およびこの復号装置に相当する機能を実現するための復号方法、復号プログラム、および復号プログラム記録媒体に関するものである。

##### 【0002】

##### 【従来の技術】

従来、デジタルの映像や音声などのデータを圧縮したビットストリームを復号する場合、例えば、図56に示す復号装置などによってその処理が行われている。ここでは図56の復号装置を例としてその動作を説明する。

図56に従来の復号装置のブロック図、図57に図56のブロック図を実現する、より詳細なブロック構成の一例を示す。

図56の復号装置は、ユーザ情報設定回路1110によりシステム制御回路1090に指示したチャネルIDあるいはシステム制御回路1090に保存したチャネルIDに基づいて、システム制御回路1090がビットストリーム出力回路108にいずれの放送を受信し、放送中のどのビットストリームを出力するかを指示する。

ビットストリーム出力回路108内では、図57に示すデジタル放送受信回路1081とデータ分離回路1082とによって、受信した放送からデコード回路101での復号対象となるオーディオビットストリームおよびビデオビットストリームを出力する。

##### 【0003】

まず図57のデジタル放送受信回路1081では、システム制御回路1090の指示に基づいて復号の対象となる放送を受信して通信路復号化を行いビットストリームBSTを出力する。データ分離回路1082ではシステム制御回路1090の指示に基づいてビットストリーム中のデコード対象となるオーディオビットストリームAST、ビデオビットストリームVSTを分離して出力する。

ビットストリーム出力回路108が出力したオーディオビットストリームとビデオビットストリームとを格納するオーディオビットストリームバッファ1042とビデオビットストリームバッファ1041とをフレームメモリ104内に確保し、それぞれを FIFOメモリとして機能させてデコード回路101に出力する。ここで、これらストリームバッファ1042および1041を FIFOメモリとして機能させるために、ビットストリーム出力回路108、フレームメモリ104、デコード回路101等によってバッファ残量の管理とストリーム入出力の制御を行う。

##### 【0004】

デコード回路101では、フレームメモリ104からオーディオビットストリーム、ビデオビットストリームの供給を受け、それぞれオーディオデコード回路1012、ビデオ

デコード回路 1011 で復号化を行い、復号結果をフレームメモリ 104 内に確保したオーディオデータバッファ 1044, ビデオデータバッファ 1043 に出力する。ここでデコード回路 101 において実時間の処理を保証するためには符号化されたデータの単位に対応した処理時間内に処理を完了する必要がある。このことをビデオの復号処理を例にとって図 58, 図 59 に図示する。

#### 【0005】

図 58, 図 59 のいずれについてもビデオはフレームを単位として符号化されているとする。ここでは 480i、即ち、走査線 480 本のインターレース映像、を符号化していると想定すると、このときビデオの復号処理 VDP はフレームの時間  $F$  内（フレームの時間の半分がフィールドの時間で、ビデオプランキングから次のビデオプランキングまでの間隔である  $V$  期間の 2 回分に相当する）に処理が完了すればよく、これを示すのが図 58 である。逆に図 59 はビデオの復号処理がフレームの時間内に完了しない場合を示し、結果として復号処理中にコマ落ちが発生したり、途中までしか復号されていない映像が出来たりといった不正な映像が出来ることになる。従って、不正な映像を出力しないようにデコード回路から正常にデコードを完了した映像を出力する制御を行う。

なお、この図 56 ではいずれの映像を出力するかの制御はデコード回路 101 で判断を行い、いずれの映像を出力するかをシステム制御回路 1090 を経て合成回路 1000 に通知しているが、デコード回路 101 から直接、合成回路 1000 に通知してもよい。

#### 【0006】

復号したオーディオデータはオーディオデータバッファ 1044 からオーディオ・ビデオ出力回路 1100 に出力され、オーディオ D/A 変換回路 1101a ではアナログオーディオ出力 a に変換し、オーディオデータ信号変換回路 11011a ではデジタルオーディオ出力 d a に変換して出力する。

また、システム制御回路 1090 では静止画やオンスクリーンディスプレイを生成し、フレームメモリ内の静止画用領域  $S_{t110}, S_{t111} (s0, s1)$  およびオンスクリーンディスプレイ用領域  $OSD0, OSD1 (00, 01)$  等に格納する。

#### 【0007】

合成回路 1000 ではビデオデータバッファに格納したビデオデータと静止画用領域  $S_{t110}, S_{t111} (s0, s1)$  等に格納した静止画およびオンスクリーンディスプレイ用領域  $OSD0, OSD1 (00, 01)$  等に格納したオンスクリーンディスプレイ画面のそれぞれをシステム制御回路 1090 から指示される合成パラメータ SP に基づいて合成し合成ビデオデータを出力する。

合成ビデオデータは合成回路 1000 からオーディオ・ビデオ出力回路 1100 に出力され、ビデオ D/A 変換回路 1101v ではアナログビデオ出力 v に変換し、ビデオデータ信号変換回路 11011v ではデジタルビデオ出力 d v に変換して出力する。

以上のように、図 56 の復号装置はビットストリームから映像と音声とを復号する。

#### 【0008】

##### 【発明が解決しようとする課題】

従来の技術による復号装置は、上記のように構成されており、1つのビットストリームからそれぞれ1つの映像と音声とを復号することを前提としている。従って、例えば BS (Broadcasting Satellite) デジタル放送、CS (Communication Satellite) デジタル放送、110° CS デジタル放送（東経 110° に打ち上げられた CS によるデジタル放送）などの複数のビットストリームから映像・音声を同時に復号するには、復号しようとするビットストリームと同数分の復号装置を必要とした。

本発明は、かかる事情に鑑みてなされたものであり、1つの復号処理で見かけ上同時に複数の映像・音声を復号することが可能な復号装置、復号方法、復号プログラム、および復号プログラム記録媒体を提供することを目的とする。

#### 【0009】

##### 【課題を解決するための手段】

上記の目的を達成するために、本発明の復号装置は、複数の符号化データを入力し、復

号して出力するデコード部と、前記デコード部の複数の状態を退避・復元するデコード状態保持部とを備え、前記デコード部は、復号すべき複数の符号化データの中の、復号対象となる符号化データを選択して復号するとともに、前記デコード状態保持部に対して、自らの状態の退避・復元を行い、復号した複数の復号結果をそれぞれに対応した出力先に出力するとともに、一連の処理を各々の符号化データに対応した所定の期間内に完了するようにしたものである。

#### 【 0 0 1 0 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、符号化単位の境界で符号化データの入力を、復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたものである。

#### 【 0 0 1 1 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、各々の符号化データに記された情報に基づいて、復号結果を出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたものである。

#### 【 0 0 1 2 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、その復号結果を格納するメモリと、いずれの符号化データを復号するかを指示するシステム制御部と、前記メモリに格納した復号結果を出力する出力制御部とを備え、前記システム制御部の指示に基づいて復号する符号化データを変更し、前記出力制御部は、前記変更に伴って前記フレームメモリ中のデコード結果を格納する配置が変更になる場合に、所定の出力を行うようにしたものである。

#### 【 0 0 1 3 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、その復号結果を格納するメモリと、前記メモリに格納した復号結果を出力する出力制御部とを備え、前記出力制御部は、各符号化データの復号結果を出力する際に、共通の時間指標に基づいて出力を制御するようにしたものである。

#### 【 0 0 1 4 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、その復号結果を格納するメモリと、前記メモリに格納した復号結果を出力する出力制御部とを備え、前記出力制御部は、各符号化データの復号結果の出力を制御する際に、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたものである。

#### 【 0 0 1 5 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するようにしたものである。

#### 【 0 0 1 6 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたものである。

#### 【 0 0 1 7 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、各符号化データの

復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたものである。

【 0 0 1 8 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたものである。

【 0 0 1 9 】

また、本発明の復号装置は、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、前記デコード部は、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたものである。

【 0 0 2 0 】

また、本発明の復号方法は、符号化データを入力し、復号して出力する復号方法において、復号対象である符号化データの復号を中断する際に、復号を行うデコード部の状態を待避し、復号対象である符号化データを他の符号化データに切り換えて復号を行い、復号を中断した符号化データの復号の再開は、待避したデコード部の状態を復元した後に行うようにしたものである。

【 0 0 2 1 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、符号化単位の境界で符号化データの入力を復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたものである。

【 0 0 2 2 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、各々の符号化データに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたものである。

【 0 0 2 3 】

また、本発明の復号方法は、複数の符号化データ中の復号すべき符号化データを指示に基づいて変更し、前記復号すべき符号化データを復号し、復号結果をメモリに格納し、前記指示による復号すべき符号化データの変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出力を行うようにしたものである。

【 0 0 2 4 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、共通の時間指標に基づいて出力を制御するようにしたものである。

【 0 0 2 5 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記から復号結果を出力する際、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたものである。

【 0 0 2 6 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するようにしたものである。

【 0 0 2 7 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データ

を復号する際、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたものである。

#### 【 0 0 2 8 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたものである。

#### 【 0 0 2 9 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたものである。

#### 【 0 0 3 0 】

また、本発明の復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたものである。

#### 【 0 0 3 1 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、符号化データを入力し、復号して出力する復号方法であって、復号対象である符号化データの復号を中断する際に、復号を行うデコード部の状態を待避し、復号対象である符号化データを他の符号化データに切り換えて復号を行い、復号を中断した符号化データの復号の再開は、待避したデコード部の状態を復元した後に行うようにしたものである。

#### 【 0 0 3 2 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、符号化単位の境界で符号化データの入力を復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたものである。

#### 【 0 0 3 3 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、各々の符号化データに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたものである。

#### 【 0 0 3 4 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データ中の復号すべき符号化データを指示に基づいて変更し、前記復号すべき符号化データを復号し、復号結果をメモリに格納し、前記指示による復号すべき符号化データの変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出力を行うようにしたものである。

#### 【 0 0 3 5 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、共通の時間指標に基づいて出力を制御するようにしたものである。

#### 【 0 0 3 6 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、所定

のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたものである。

#### 【 0 0 3 7 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するようにしたものである。

#### 【 0 0 3 8 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたものである。

#### 【 0 0 3 9 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたものである。

#### 【 0 0 4 0 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたものである。

#### 【 0 0 4 1 】

また、本発明の復号プログラムは、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたものである。

#### 【 0 0 4 2 】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、符号化データを入力し、復号して出力する復号方法であって、復号対象である符号化データの復号を中断する際に、復号を行うデコード部の状態を待避し、復号対象である符号化データを他の符号化データに切り換えて復号を行い、復号を中断した符号化データの復号の再開は、待避したデコード回路の状態を復元した後に行うようにしたものである。

#### 【 0 0 4 3 】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、符号化単位の境界で符号化データの入力を復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたものである。

#### 【 0 0 4 4 】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、各々の符号化デ

ータに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたものである。

【0045】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データ中の復号すべき符号化データを指示に基づいて変更し、前記復号すべき符号化データを復号し、復号結果をメモリに格納し、前記指示による復号すべき符号化データの変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出を行なうようにしたものである。

【0046】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、共通の時間指標に基づいて出力を制御するようにしたものである。

【0047】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたものである。

【0048】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するようにしたものである。

【0049】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたものである。

【0050】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたものである。

【0051】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたものである。

【0052】

また、本発明の復号プログラム記録媒体は、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力し

て、それぞれの符号化データを復号する際、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたものである。

#### 【 0 0 5 3 】

##### 【発明の実施の形態】

###### 実施の形態 1 ．

本発明の実施の形態 1 による復号装置は、デコード状態保持回路によってデコード回路の状態を退避・保持し、デコード回路の処理するビットストリームの変更に応じて状態を復元することで、1 つのデコード回路で見かけ上同時に複数のビットストリームをデコードできるようにしたものである。

図 1 は本実施の形態 1 による復号装置（請求項 1 に対応）の構成を示すブロック図である。図示するように、本実施の形態 1 による復号装置は、従来例に比べ主に以下の 2 つの部分が異なる。1 点目は新たにデコード状態保持回路を備えた点であり、2 点目は合成回路のかわりに出力制御回路を備えた点である。

#### 【 0 0 5 4 】

図 1 において、1 0 8 は複数のビットストリームを出力するビットストリーム出力回路、1 0 7 はビットストリーム出力回路 1 0 8 からの複数のビットストリームをフレーム単位で蓄積できるフレームメモリ（メモリ）、1 0 3 は複数のビットストリームのデコードを行う1 つのデコード回路（デコード部）、1 0 2 はデコード回路 1 0 3 のデコード状態を保持するデコード状態保持回路（デコード状態保持部）、1 1 0 は複数の各ビットストリームのデコードされた出力であるビデオ出力 v 1 , v 2 , … , オーディオ出力 a 1 , a 2 , … 、デジタルビデオ出力 v 1 1 , v 1 2 , … , デジタルオーディオ出力 a 1 1 , a 1 2 , … 等を出力するオーディオ・ビデオ出力回路、1 0 6 はオーディオ・ビデオ出力回路 1 1 0 に対する出力制御を行う出力制御回路（出力制御部）、1 0 9 はこの復号装置を制御するシステム制御回路（システム制御部）、1 1 1 はシステム制御回路 1 0 9 に対しユーザ情報を設定するユーザ情報を設定回路である。なお、デコード回路 1 0 3 は専用ハードウェアで構成されるものでも、DSP (Digital Signal Processor) 等のコンピュータのソフトウェアで実現されるものでもよい。

#### 【 0 0 5 5 】

この実施の形態 1 の復号装置の全体の動作は次のようになる。即ち、HDTV (High-Definition TV) 1 チャンネル分の帯域は SDTV (Standard-Definition TV) 複数チャンネル分の帯域に相当する。例えば SDTV 3 チャンネル分の情報を含んだ放送電波 B R はビットストリーム出力回路 1 0 8 に入力され、その内部のフロントエンド（図示しない）で復調される。ビットストリーム出力回路 1 0 8 は、その復調結果として SDTV のチャンネル 1 , 2 , 3 を取り出し、フレームメモリ 1 0 7 に出力する。フレームメモリ 1 0 7 に蓄積された、各チャンネルの SDTV 放送に対応するビットストリームは、システム制御回路 1 0 9 の制御により、例えば SDTV のチャンネル 1 がデコード回路 1 0 3 によりデコードされ、その 1 フレーム分等の所定のデータ量のデコードが完了した段階で、デコード回路 1 0 3 の状態がデコード状態保持回路 1 0 2 に退避され、次の SDTV のチャンネル 2 がデコード回路 1 0 3 によりデコードされ、同様に 1 フレーム分等の所定のデータ量のデコードが完了した段階で、デコード回路 1 0 3 の状態がデコード状態保持回路 1 0 2 に退避される…、という動作を巡回的に、例えばチャンネル 1 チャンネル 2 チャンネル 3 チャンネル 1 … などのように繰り返す。

#### 【 0 0 5 6 】

デコード回路 1 0 3 は 1 フレーム分の動作が完了した段階で、その状態がデコード状態保持回路 1 0 2 に保持されているため、以上のような動作が一巡して元のチャンネルに戻った段階で、前のフレームのデコードが終了した時点の状態をデコード状態保持回路 1 0 2 から呼び戻すため、途中でデコードしているチャンネルが変わっても、元のチャンネルに戻った時点で次のフレームのデコードに支障なく移行できる。

オーディオ・ビデオ出力回路 1 1 0 は、出力制御回路 1 0 6 により出力された各チャンネル毎のデコード結果に基づき、ビデオ出力 v 1 , v 2 , … デジタルビデオ出力 v 1 1 ,

v 1 2 , ... オーディオ出力 a 1 , a 2 , ... デジタルオーディオ出力 a 1 1 , a 1 2 , ... を外部に出力する。

そして、こうした動作を S D T V の各チャンネルに対するデコードを、 S D T V 1 チャンネルのみデコードする場合に要求される速度の 3 倍以上の速度で行うことにより、 1 つの復号装置を時分割で動作させることができ、 みかけ上 3 チャンネルの復調を同時に行うことができる。

#### 【 0 0 5 7 】

次に本復号装置を入力部、デコード部、出力部の 3 つに分けて各部の動作をそれぞれ説明する。

##### [ 入力部 ]

ビットストリーム出力回路 1 0 8 は複数のデジタル放送を含んだ放送電波 B R を受信し、システム制御回路 1 0 9 より指示を受けた複数のチャネルのオーディオビットストリームとビデオビットストリームとをフレームメモリ 1 0 7 に格納する。ビットストリーム出力回路 1 0 8 の構成例を図 2 に示す。

図 2 に示すように、ビットストリーム出力回路 1 0 8 は、デジタル放送受信回路 1 0 8 1 とデータ分離回路 1 0 8 2 とで回路を構成している。ここでデジタル放送受信回路 1 0 8 1 は複数のデジタル放送を含んだ放送電波 B R を受信して、図 3 に示すようにそれぞれの放送のビットストリーム B S T 0 , B S T 1 , B S T 2 からなるビットストリーム B S T を出力する。この複数のデジタル放送を含んだ放送電波 B R は、 1 つの H D T V の帯域を分割して同一時間帯に複数の S D T V 番組が放送されるものでもよい。また、 1 つの放送局から番組ごとに異なる放送方式で放送されるものであってもよい。これは、例えば番組 A は H D T V で放送され、番組 A に統一して放送される番組 B は S D T V で放送される場合などで、このような放送方式の混在は、番組の著作権者が放送方式の変換を許可しない場合などで生じる。また、これらビットストリーム B S T 0 , B S T 1 , B S T 2 のいくつかは、 D - V H S ( 登録商標 ) ビデオデッキ等によりタイムシフトした放送内容であってもよい。

#### 【 0 0 5 8 】

図 3 の例では、各放送ビットストリーム B S T はそれぞれ 3 つのデータから構成され、各データはパケットにより構成されている。データ分離回路 1 0 8 2 は図 4 に示すように、複数のビットストリーム B S T から、システム制御回路 1 0 9 からの指示を受けたオーディオビットストリーム A S T ( オーディオビットストリーム A S T 0 , A S T 1 , A S T 2 ) とビデオビットストリーム V S T ( ビデオビットストリーム V S T 0 , V S T 1 , V S T 2 ) とを出力し、互いに連動するセレクタ S L 2 , S L 1 により選択したフレームメモリ 1 0 7 のそれぞれ対応するオーディオビットストリームバッファ 1 0 7 0 a , 1 0 7 1 a , 1 0 7 2 a とビデオストリームバッファ 1 0 7 0 v , 1 0 7 1 v , 1 0 7 2 v とに格納する。これにより、図 4 のビットストリーム 0 の例では、放送 0 データ 0 0 に属するパケットが選択されてオーディオビットストリーム A S T 0 となり、放送 0 データ 0 1 に属するパケットが選択されてビデオビットストリーム V S T 0 となる。

#### 【 0 0 5 9 】

ビットストリーム出力回路の構成例としては図 5 、図 6 、図 1 0 、図 1 1 等がある。

図 5 はビットストリーム出力回路の構成例として、従来例と同様のデジタル放送受信回路 1 0 8 1 0 , 1 0 8 1 1 , 1 0 8 1 2 , ... とデータ分離回路 1 0 8 2 0 , 1 0 8 2 1 , 1 0 8 2 2 , ... とをそれぞれ直列接続した回路を、ビットストリームと同数分用意して互いに並列に配置し、これら複数の直列回路の出力をフレームメモリに出力するための複数のデータ分離バッファ 1 0 8 3 0 , 1 0 8 3 1 , 1 0 8 3 2 , ... を設けたものである。これらのバッファはデータ分離回路に対応して同数設けられ、その出力がセレクタ S 1 により切り替えられて 1 つのフレームメモリ 1 0 7 に出力される。

この図 5 のビットストリーム出力回路は、従来例と同様の回路により構成が可能であるので、構築が容易である。

#### 【 0 0 6 0 】

図6はビットストリーム出力回路の他の構成例として、従来例と同様のデジタル放送受信回路10810, 10811, 10812, …と、データ多重バッファ10840, 10841, 10842, …とをそれぞれ直列接続した回路を、ビットストリームと同数分用意して互いに並列に配置し、各データ多重バッファから出力したビットストリームを1つのデータ多重回路10850により多重化して図7に示すようなパケットがシリアルに配置された1つのビットストリームMとした後、データ分離回路1082により対象となるオーディオビットストリームとビデオビットストリームを抽出する。

#### 【0061】

各デジタル放送受信回路とデータ多重回路との間にそれぞれデータ多重バッファを設けたのは、多重化して1つのビットストリームを生成するために、データ多重回路10850において単純にパケットの多重化を行うのみでは各ビットストリームが使用している時間情報や各オーディオストリーム・ビデオストリームを示す指示情報にそれぞれ不整合が発生するが、こうした不整合が生じないように修正を行うためである。

#### 【0062】

例えば、時間情報に関しては1つのビットストリームを基準として、ある時点での基準のビットストリームと他のビットストリームの時間情報の差分を求め、基準以外のビットストリームの時間情報に対して、求めた差分の値だけ修正する。図8の例では、D01はビットストリームBST0とビットストリームBST1との間の時間情報の差分、D02はビットストリームBST0とビットストリームBST2との間の時間情報の差分である。この処理はビットストリームが示す時間指標だけでなく、デコードしたオーディオを出力する時間指標およびデコードした映像を出画する時間指標などビットストリーム中に含まれるすべての時間情報に対して行う必要がある。また、図9に示すように、各オーディオストリーム・ビデオストリームの指示情報については修正前の指示情報di0と修正後の指示情報di1とがお互いに重複しない値になるように変更する。

#### 【0063】

なお、図6のデジタル放送受信回路10810～10812、データ多重バッファ10840～10842、およびデータ多重回路10850からなるデジタル放送受信回路については、その出力のビットストリームをデジタル放送送信回路に接続することで、デジタル放送受信回路10810～10812により受信した放送をデジタルCATV等を用いて再送するのに利用することができる。

#### 【0064】

図10のビットストリーム出力回路は図6のビットストリーム出力回路をベースとしているが、これと異なる点は、それぞれのデジタル放送受信回路10810, 10811, 10812, …とデータ多重バッファ10840, 10841, 10842, …との間に、従来例と同様のデータ分離回路10820, 10821, 10822, …を挿入したことである。これにより不要なチャネルのデータを以降の回路に送ることがなくなり、このビットストリーム出力回路以降で多重化されたストリームに対して必要とされる帯域幅の条件が緩和される。

#### 【0065】

図11のビットストリーム出力回路は、デジタル放送受信回路10810, 10811, 10812, …を従来例と同様、複数のビットストリームと同数分、配置し、各デジタル放送受信回路10810, 10811, 10812, …と1つのデータ分離回路1082との間に、ビットストリームを供給するためのデータ分離バッファ10830, 10831, 10832, …をビットストリームと対応させて同数分設け、データ分離回路1082側にこれらデータ分離バッファ10830, 10831, 10832, …からの複数のビットストリームを分離する機能を持たせたものである。

#### 【0066】

以上、ビットストリーム出力回路について、いずれもデジタル放送受信回路のみをビットストリームのソースとする構成を示したが、図12に示すように、他のデジタル機器からのビットストリームを取り込むためのデジタルインターフェイス10860, 1086

1, ... や、デジタルデータを記憶しているデジタルデータ記憶回路 10870, ... をソースとして用いてもよく、これらデジタルインターフェイスやデジタルデータ記憶回路の出力をセレクタ S L 3, S L 4, S L 5 により選択し、デジタル放送受信回路 10810, 10811, 10812, ... の出力に代えてデータ分離バッファ 10830, 10831, 10832, ... に取り込むことができる。

#### 【0067】

また、1つのHDTV帯域を使用して3つのSDTVを放送する場合のように、同一放送のストリーム中に複数のビットストリームが含まれる放送を復号する場合は、図13のようにビットストリーム出力回路 108 を1つのデジタル放送受信回路 1081 とデータ分離回路 1082 とで構成したものにより実現が可能である。ここではデータ分離回路 1082 の前段にデジタル放送受信回路を設けることで実現した例を示したが、例えばDVDのような記録メディアからビットストリームを供給することも可能である。

#### 【0068】

##### [デコード部]

次に、フレームメモリ、デコード回路、デコード状態保持回路の内部構成の一例を図14に示す。基本的にはフレームメモリ 107 は各々のビデオビットストリームバッファ 1070v, 1071v, 1072v, ... をセレクタ S L 11 で、オーディオビットストリームバッファ 1070a, 1071a, 1072a, ... をセレクタ S L 21 で、それぞれ選択してビデオビットストリーム VST、オーディオビットストリーム AST を読み出し、ビデオデータおよびオーディオデータをそれぞれビデオデコード回路 1031 およびオーディオデコード回路 1032 で復号し、その結果をセレクタ S L 11, S L 21 とそれぞれ連動するセレクタ S L 12, S L 22 により対応するビデオデータバッファ 10710v, 10711v, 10712v, ... オーディオデータバッファ 10710a, 10711a, 10712a, ... を選択して格納する。このときいずれのバッファをデコード対象とするかはシステム制御回路 109 の設定に基づいて決定し、どのタイミングで処理対象のバッファを切り替えるかは各デコード回路 1031, 1032 にて決定し、その決定結果を示すビデオバッファ番号指示信号 VBD およびオーディオバッファ番号指示信号 ABD をセレクタ S L 11, S L 12 およびセレクタ S L 21, S L 22 b に伝送する。

#### 【0069】

ここで、処理の切り替えについて図16ないし図24の例を用いて説明する。なお、説明の簡略化のためビデオの処理についてのみ説明するが、オーディオの処理に関しても同様の説明が成り立つ。図15は図14のビデオの復号に関する部分のみを記したものである。各ビデオビットストリームバッファ 1070v, 1071v, 1072v からビデオデコード回路 1031 に供給するビデオビットストリーム VST0, VST1, VST2 の概要を図16に示す。図16の例では、ビデオストリーム VST0 はヘッダ 0-0, ビデオストリーム 0-0, ヘッダ 0-1, ビデオストリーム 0-1, ヘッダ 0-2, ビデオストリーム 0-2 がこの順で配置される。これらのビデオビットストリームをビデオデコード回路 1031 で処理するが、各々のビデオストリームに記述された情報に基づいて出力すべき順序を求め、求めた順序に従ってデータを選択してデコードし(請求項3, 14, 25, 36に対応)、MPEG2におけるスライスのヘッダ等、所定のタイミングで処理の対象となるビデオビットストリームをセレクタ S L 11 で切り替える。これにより、所定のタイミング以外のタイミングで切り替えを行った場合の複雑なデコード処理が不要となり、処理性能が低い場合であっても1つのデコード回路により複数の符号化データを同時に復号できる。

#### 【0070】

この処理の例を図17に示す。ここではビデオビットストリーム 0, 1, 2 の順でデコード処理を巡回して行うものとしている。まず、ビデオビットストリーム 0 のデコード処理を行い、フレームやスライスの区切り等の処理を中断すべき箇所 d p x g までのデコードが終了すれば、ビデオビットストリーム 0 の途中でデコード処理を中断し、次に処理対象を切り替えてビデオビットストリーム 1 のデコードを行う。デコード結果はセレクタ S

L 2 1 を切り替えて、ビデオデータバッファ 1 0 7 1 0 v , 1 0 7 1 1 v にそれぞれ出力する。この切り替えに先だってビデオビットストリーム 0 の以降の処理に用いる情報を保存し、再度ビデオビットストリーム 0 を中断した位置からデコード処理を再開する際にこの待避した情報を読み込む必要がある。

#### 【 0 0 7 1 】

このときの情報を保存するための回路がビデオデコード状態保持回路 1 0 2 0 v , 1 0 2 1 v , 1 0 2 2 v である。保存する情報の例としては、ヘッダ情報やビデオストリームをデコード中の状態、即ち、回路内の状態遷移やメモリ内容などがある。今ここで、デコード対象のビデオビットストリームが M P E G 2 に準拠したものであるとすると、回路の状態遷移等を除いて保存しなければならない情報の例としては、M P E G 2 フォーマットのシンタックス上に現れる情報 (Qmatrix情報や映像がプログレッシブであるか否かの情報、どの順序で D C T 係数が符号化されているかを示す情報等) やシンタックス上に現れない情報 (D C T の D C 成分の予測符号化のために保持した情報や動きベクトルの予測符号化のために保持した情報、フレーム間予測に用いる参照フレームがいずれであるかを示す情報等) がある。

#### 【 0 0 7 2 】

そして、例えば図 1 7 に示すデコード中断時点 d r 0 , デコード再開時点 d r 1 では、ビデオビットストリーム 0 のデコード状態をビデオデコード状態保持回路 1 0 2 0 v に保存した後、セレクタ S L 3 0 を切り替えビデオデコード状態保持回路 1 0 2 1 v から先のビデオビットストリーム 1 のデコード状態を復元して処理をビデオビットストリーム 1 に移行する。以上の処理を繰り返して複数のビデオビットストリームを処理単位時間内にデコードすることで、1 つのデコード回路で複数のビデオビットストリームの同時デコードを行うことが可能になる。

#### 【 0 0 7 3 】

図 1 8 はこれら一連の処理を示すフローチャートである (請求項 1 2 , 2 3 , 3 4 に対応)。まず、フレームメモリ 1 0 7 よりビットストリームデータを読み込む (ステップ S 1 0 1 参照)。次に、このビットストリームデータをデコード回路 1 0 3 によりデコードし (ステップ S 1 0 2 参照)、ビデオストリームの途中のスライスの終端等、デコードを中断すべき位置に到達するまで、処理を続ける (ステップ S 1 0 3 参照)。このデコードを中断すべき位置に到達すると、デコード状態保持回路 1 0 2 に状態を保持し (ステップ S 1 0 4 参照)、デコードすべきビットストリームを切り換える (ステップ S 1 0 5 参照)。この時、デコード対象であるビットストリームの各符号化単位の全てのデコードが完了しているか否かを判定し、完了していなければ (ステップ S 1 0 6 参照)、切り換えが一巡以上しているか否かを判定する (ステップ S 1 0 7 参照)。一巡以上していれば、デコード状態保持回路 1 0 2 に保持しているデコードの状態をデコード回路 1 0 3 に戻し (ステップ S 1 0 8 参照)、フレームメモリ 1 0 7 のアドレスをデコード処理を中断した位置に戻し (ステップ S 1 0 9 参照)、ステップ S 1 0 1 に戻る。これに対し、切り換えが一巡以上に達していなければ、ステップ S 1 0 8 , S 1 0 9 をスキップしてステップ S 1 0 1 に戻る。

#### 【 0 0 7 4 】

また、処理の切り替えをビデオストリームの途中では行わず、図 2 0 のように次のビデオストリームのヘッダとの境界部分でデコードの中止 (切り替え) を行えば (請求項 2 , 1 3 , 2 4 , 3 5 に対応)、この切り替え箇所 d p x g ではビデオストリーム自体については1つの処理単位が完了しているため、ビデオデコード状態保持回路にて保存すべき情報量を削減することができる。図 2 0 ではビデオストリーム終了後の時点 d r 0 と次のヘッダの開始前の時点 d r 1 との間でデコード処理を切り替えているが、ビデオストリームと次のヘッダを連続して処理し、次のヘッダと次のビデオストリームの間で処理を切り替えてよい。この場合、次のビデオストリームのヘッダ情報が予め得られるので、次のビデオストリームのデコード開始前にデコードのための前処理を行うことが可能になる。

#### 【 0 0 7 5 】

また、上記では回路実装的な観点から説明したが、例えば図19に示すようなDSP等によるソフトウェア実装を行った場合、ビデオデコード状態保持回路はDSP200内部のメモリ202で実現可能であり、その場合、状態の退避・復帰処理はポインタ（図示せず）が指示するメモリのアドレスを変更することで実現可能となる。但し、DSP内部にデコードのためのハードウェアアクセラレータを搭載している場合、その構成によってはハードウェアアクセラレータ部分の状態退避・復帰処理が別途必要になる。

#### 【0076】

なお、図19において、201はプログラムメモリ、203はDMA、外部メモリインターフェースユニット、204はタイマ、シリアルポート、2051は制御レジスタ、2052は制御ロジック、2053は割り込み回路、2054はプログラムフェッチ、2055は命令ディスパッч、2056は命令デコード、2057はレジスタファイルA、2058はレジスタファイルB、2059は命令実行ユニットA、2060は命令実行ユニットBである。

#### 【0077】

また更に、ビデオデコード状態保持回路をDSP内部のメモリのみで実現するのではなく、一部あるいはすべてをフレームメモリ内に確保した領域で実現することも可能である。但し、処理時間的にフレームメモリへのアクセスはDSP内部のメモリへのアクセスに比べ多くの時間を必要とするので、処理時間が増大しても支障をきたさないのであれば、DSP内部に確保するデコード状態保持用のメモリ容量を削減することが可能である。

#### 【0078】

また更に、ソフトウェア実装の場合、オーディオデコード処理についてもビデオデコード処理と同様に、オーディオの符号化単位の処理時間内に対象となる複数のチャネルのデコードを完了することで、デコード回路はビデオデコード回路とオーディオデコード回路とに分離することなく1つの回路で実装し、オーディオ・ビデオを複数同時に復号可能となる。但し、符号化単位の違いからオーディオとビデオのデコード処理の管理（オーディオの処理対象チャネル切り替えとビデオの処理対象チャネル切り替えに関する管理）は分離して行い、オーディオ・ビデオ間の処理の切り替えは例えばビデオデコード処理中でハードウェアアクセラレータの処理完了待ちの間にオーディオデコード処理を行うなどビデオデコード処理の待機中の期間に行わせる、などにより実現する。

#### 【0079】

##### [出力部]

次にデコード結果の出力にかかるフレームメモリ、出力制御回路、システム制御回路の構成例を図21に示す。前述のオーディオデコード回路およびビデオデコード回路によりデコードした結果がそれぞれオーディオデータバッファ10710a, 10711a, 10712a, . . . 、ビデオデータバッファ10710v, 10711v, 10712v, . . . に格納されており、これらを如何に出力するかを制御する。

#### 【0080】

オーディオ出力生成回路1061はシステム制御回路109の指示に基づいてオーディオデータバッファ10710a, 10711a, 10712a, . . . の出力を制御し、オーディオD/A変換回路1101a, 1102a, 1103a, . . . からオーディオ出力a1, a2, a3, . . . を、オーディオデータ信号変換回路11011a, 11012a, 11013a, . . . からデジタルオーディオ出力a11, a12, a13, . . . をそれぞれ出力する。

#### 【0081】

このとき図21ではシステム制御回路109の指示に基づいてオーディオD/A変換回路1101a, 1102a, 1103a, . . . 、オーディオデータ信号変換回路11011a, 11012a, 11013a, . . . が動作しているが、これに代えてオーディオ出力生成回路1061からの指示で動作してもよい。

#### 【0082】

図22はこれら一連の処理を示す。オーディオ出力生成回路1061は、システム制御

回路 109 からの指示がオーディオデータバッファ (1070a ないし 1072a のいずれか) に対するものであれば (ステップ S201 参照)、オーディオデータバッファ (1070a ないし 1072a のいずれか) からオーディオデータを取得し (ステップ S202 参照)、そうでなければシステム制御回路 109 またはデコード回路 103 により、例えば無音用のデータ等の所定のオーディオデータを生成する (ステップ S203 参照)。これを全てのチャネルについて繰り返し (ステップ S204 参照)、取得したオーディオデータまたは生成した所定のオーディオデータに基づきデジタル信号処理を行ってオーディオ出力を生成し、オーディオ D/A 変換回路 1101a ないし 1103a およびオーディオ信号変換回路 11011a ないし 11013a よりそれぞれオーディオ出力およびデジタルオーディオ出力を得る (ステップ S205 参照)。

#### 【0083】

ビデオ出力制御回路 1062 は、システム制御回路 109 の指示に基づいてビデオデータバッファ 10710v, 10711v, 10712v, ... およびオンスクリーンディスプレイ用領域 OSD0 00, OSD1 01, ...、静止画用領域 still0 s0, still1 s1, ... から出力映像を生成し、出画バッファ 1077 に出力する。出画バッファ 1077 に格納した映像をビデオ出力生成回路 1063 はシステム制御回路 109 に基づいてビデオ D/A 変換回路 1101v, 1102v, 1103v, ... およびビデオデータ信号変換回路 11011v, 11012v, 11013v, ... に出力し、ビデオ D/A 変換回路 1101v, 1102v, 1103v, ... およびビデオデータ信号変換回路 11011v, 11012v, 11013v, ... からはそれぞれビデオ出力 v1, v2, v3, ...、デジタルビデオ出力 v11, v12, v13, ...、を出力する。

#### 【0084】

図 23 はこれら一連の処理を示す。

ビデオ制御回路 1062 は、システム制御回路 109 からの指示がビデオデータバッファ (1070v ないし 1072v のいずれか) に対するものであれば (ステップ S301 参照)、ビデオデータバッファ (1070v ないし 1072v のいずれか) からビデオデータを取得し (ステップ S302 参照)、そうでなければシステム制御回路 109 またはデコード回路 103 により例えればブルースクリーン用等の、所定のビデオデータを生成する (ステップ S303 参照)。これを全てのチャネルについて繰り返す (ステップ S304 参照)。

#### 【0085】

また、出力制御回路 1062 は、システム制御回路 109 からの指示が OSD に対するものであれば (ステップ S305 参照)、OSD 領域 (00, 01 のいずれか) からビデオデータを取得し (ステップ S306 参照)、そうでなければシステム制御回路 109 またはデコード回路 103 により所定のビデオデータを生成する (ステップ S307 参照)。これを全てのチャネルについて繰り返す (ステップ S308 参照)。

#### 【0086】

さらに、出力制御回路 1062 は、システム制御回路 109 からの指示が still1 に対するものであれば (ステップ S309 参照)、still1 領域 (s0, s1 のいずれか) からビデオデータを取得し (ステップ S310 参照)、そうでなければシステム制御回路 109 またはデコード回路 103 により所定のビデオデータを生成する (ステップ S311 参照)。これを全てのチャネルについて繰り返す (ステップ S312 参照)。

#### 【0087】

そして、このようにして得たビデオデータを一旦出画バッファ 1077 に蓄積した後、デジタル信号処理を行ってビデオ出力を生成し、ビデオ D/A 変換回路 1101v ないし 1103v およびビデオ信号変換回路 11011v ないし 11013v よりそれぞれビデオ出力およびデジタルビデオ出力を得る (ステップ S313 参照)。

#### 【0088】

図 21 の説明では複数のオーディオビットストリーム、ビデオビットストリームをデコードした結果をそれぞれ独立して出力しているが、要求に応じて構成を変更することが可

能である。

図21の場合は、1台のデジタルチューナを複数の出力デバイス（テレビ等）で共用し、出力デバイスごとに異なる映像・音声を視聴する場合に有効であるが、例えば、1台の出力デバイスに複数の異なる映像を合成して1つの映像として出力する場合などは、出力制御回路にて映像の合成や重ね合わせの処理を行なう出力する音声の合成や選択等を行なって、出力を1系統にすることでオーディオ・ビデオ出力回路の簡略化が可能である。

#### 【0089】

また、デコードした映像・音声を選択して1台の出力デバイスに出力している場合、同時にデコードしている映像・音声間であれば高速に出力を切り替えることが可能となる。これは通常、出力を切り替える場合には、実際に出力が得られるまでにデコード処理（ビットストリームの選択、バッファへの蓄積、デコード等）の時間を要するためである。

ここまで説明では複数の映像・音声をデコードする場合について述べたが、映像のみのデコード、音声のみのデコードであっても、また映像と音声のデコードする数が異なってもよい。

#### 【0090】

以上、本装置で同時に複数の映像・音声をデコードする場合の動作について説明したが、1つの映像と音声をデコードする場合は図24、図25、図26のようになる。

このとき、複数の映像・音声をデコードする場合と異なるのはフレームメモリ内に確保する各バッファ（オーディオビットストリームバッファH、ビデオビットストリームバッファH、オーディオデータバッファH、ビデオデータバッファH）に関してである。このように、複数の映像・音声をデコードする場合と1つの映像・音声をデコードする場合でバッファの確保の仕方が異なるのは1つの映像・音声で必要とするバッファ容量が大きい場合に対応するためである。例えば複数の映像を復号する場合に扱える映像の最大サイズはNTSC程度の映像までで、1つの映像を復号する場合に扱える映像の最大サイズはハイビジョン程度の映像まで、といったことが考えられる。

#### 【0091】

このことから、複数の映像を復号する処理と、より大きなサイズの1つの映像を復号する処理が連続した場合、フレームメモリ内部の各バッファの配置が変更となる。このときの配置変更により出画に関するバッファへの影響が考えられるので、乱れた出画を抑制するために、配置変更後正常な出力が可能になるまで背景色を出力する、あるいは配置変更の影響を受けない出画用バッファを設け、出画用バッファの内容を出画することで不正な出力を抑制する等が考えられる（請求項4、15、26、37に対応）。

#### 【0092】

これは、複数の符号化データを入力した場合、即ち、複数の符号化データが入力される状況から单一の符号化データが入力される状況に推移した場合あるいは单一の符号化データが入力される状況から複数の符号化データが入力される状況に推移した場合について述べているが、メモリ内のデコード結果の配置が変更になる場合であれば、符号化データが单一の場合であっても、同様の処理を適用することが可能である。

#### 【0093】

例えば、入力の符号化データの内容がHDTVからSDTVを符号化したものに推移する場合やその逆の場合など、符号化映像のサイズに変更が伴う場合が挙げられる。

このとき、HDTV用のメモリ内の配置をそのままSDTVの復号に利用することも可能であるが、SDTV用の配置を別途設けることで、機能を向上することが可能となる。

具体的には、メモリ内の配置を変更することによりメモリの効率的な利用が可能となり、SDTV復号時にはHDTVの復号時に比べ多くの空き領域を、例えば追加のOSD（On Screen Display）や静止画の処理等、別の処理のために使用することができる。

#### 【0094】

例えば、図27(a)に示すように、HDTV放送用の領域107aの一部にNTSC用領域107bを設ける場合は、OSD（On Screen Display）用領域として1つの領域107cのみしか持つことができないが、図27(b)に示すように、HDTV放送用領

域を設けることなくNTSC用領域107dのみを配置すれば、OSD用領域として2つの領域107e, 107fを持つことができ、機能を向上することができる。

#### 【0095】

このような、メモリ内の配置の変更は、例えば、

i) 図1のビットストリーム出力回路108が、ビットストリームを解析し、その解析結果をシステム制御部109に通知してフレームメモリ107における記憶領域を切り替える。

ii) 図1のシステム制御部109がビットストリームを解析して、ビットストリームの種類を判断し、フレームメモリ107における記憶領域を切り替える。

iii) システム制御部109が、ビットストリームとは別に存在する、放送規格に定められた信号情報、すなわち、現在の放送がHDTVかSDTVであるか等を示す情報、に基づいて全体の動作を切り替える。

iv) システム制御部109がデコード回路103の復号結果、特にヘッダ情報に基づいて切り替える。

のいずれかにより実現できる。

#### 【0096】

このように、符号化データの特性に応じてメモリ内の配置を変更することが考えられるが、配置が変更になる場合には、出画の乱れを抑制するために、例えばブルースクリーンを出力する等の処理を行う。このブルースクリーンの出力は、例えば、出力制御回路106内で青を示す所定の値の信号を生成してもよいし、システム制御回路109によりフレームメモリ107の1画面の領域全面に青色を示す信号を書き込むことで実現できる。

なお、ここで、SDTV用の配置は、上述のようにSDTV専用線の領域を別途設けた配置、あるいは、複数の符号化データを復号する場合に設ける複数の領域の配置のうちの1つを使用する配置、のいずれであってもよい。

以上が本装置の基本的な動作である。以降、本装置の動作上で特に注目すべき点（出画制御・デコード順制御・デコード期間管理・デコード再開処理・デコード処理最適化）を述べる。

#### 【0097】

##### [出画制御]

まず、出力制御回路において行う出画映像決定に関して図28ないし図38の例を用いて説明する。図28は1つの映像に対して出画する映像を決定dし、その映像opを出画する場合の処理について時間軸に対応して記したものであり、図34は3つの映像を復号して出画する場合を示している。いずれも各ビデオプランキングbgの終了と開始の間で出画を行い、その出画に用いる映像をその出画開始までに決定する。この出画映像の決定は、本来出画すべき映像が正常に復号されているか、また復号した映像が出画するタイミングにあるか（後述）等に基づいて行う。図35は出画タイミングに基づいて出画する映像を決定する場合について1つの映像の例と3つの映像の例を示している。以下ではまず1つの映像の例について説明し、その後3つの映像の例について説明する。

#### 【0098】

映像が1つの場合には、図29に示すように、ビットストリーム中に記述された値を基に回路内で生成した時間指標を読み込み（ステップS401, S402参照）、その値から次の出画タイミングの時間指標を算出する（ステップS403参照）。この値と出画順にある映像に対応したビットストリーム中に記述があった出画の時間指標とを比較し（ステップS404参照）、出画すべき範囲内であれば（ステップS405参照）出画順にある映像を選択する（ステップS406参照）。出画順にある映像が出画に適していない場合には以下のスキップ処理あるいはフリーズ処理のうちのいずれかを行う（ステップS407参照）。

#### 【0099】

##### 1. スキップ処理

図30に示すように、出画順にある映像の出画の時間指標が出画タイミングの時間指標

と比較して、出画すべき範囲内でなく、小さい場合は（ステップS501参照）、この映像は出画せずに出画順でより以降の映像の出画を試みる処理を行う（ステップS502参照）。もし既に出画に適した映像がデコード済みであれば（ステップS503参照）その映像を出画し（ステップS504参照）、デコード済みの映像の中に出画に適した映像がない場合には、出画に適した映像をデコードするまでデコードを可能なだけ先行する処理を行う（ステップS505参照）。ただし、デコードが出画に間に合わない場合にはデコード済みの映像のうちいずれかを選択して出画するかあるいはデコードした映像を使用せず背景色を出力し、以降のデコード処理に関しては引き続き出画に適した映像をデコードするまで処理を可能なだけ先行する。このとき処理を先行させるために必要でない映像はデコードせずに読み飛ばしてもよい。

#### 【0100】

##### 2. フリーズ処理

図31に示すように、出画順にある映像の出画の時間指標が出画タイミングの時間指標と比較して、出画すべき範囲内でなく、大きい場合は（ステップS601参照）、出画に適したデコード済みの映像の有無を判定し（ステップS602参照）、出画に適したデコード済みの映像があればその映像を選択するか（ステップS603参照）、無ければデコード済みの映像のうち最も出画の時間指標が近い映像を選択し、適切な時間指標になるまで同映像の選択を続ける（ステップS604参照）。このときデコード処理も中断するかあるいは可能なだけデコードを先行した後デコード処理を中断する。ここで、デコードが先行可能な範囲はまだ出画していないデコード済みの映像を格納しているバッファと今後のデコード処理で参照の可能性がある映像を格納しているバッファを除いたビデオデータバッファに対してデコードした映像を格納可能な範囲である。

#### 【0101】

以上の時間指標の比較に関しては、各映像の出画の時間指標が信頼できる場合についてである。各映像の出画の時間指標に誤りが発生する可能性がある場合には、別途誤りに対する耐性を設ける必要がある。例えば、連続する出画の時間指標は基本的に大きく外れた値はとらないので、例えば、半日程度の大きく外れた値は、図32に示すように信用せず（ステップS701参照）、既に取得している値から予測する（ステップS703参照）、あるいは大きく外れた値が連続しつつそれらの相互の値が大きく外れていない場合には（ステップS702参照）、出画の時間指標の基準が変更されたと見なして、これらの時間指標を信頼できる値として採用する（ステップS704参照）。なお、時間情報の誤りは、例えば、出画対象となるか否かを判定した映像の時間情報と、現在出画しようとする映像の時間情報との差分を求め、差分が1時間等の一定値以上である場合には誤りであると判定する。

#### 【0102】

また、時間指標の読み込みについては図36に示したように特定のタイミングTで行うことが望ましい。これは読み込みのタイミングが一定でない場合、時間指標を読み込むタイミングと次の出画タイミングまでの時間を把握しなければならず、次の出画タイミングの時間指標を算出する手順が複雑になる、あるいは算出した時間指標が適切でないことが考えられるためである。従って、時間指標の読み込みは例えばビデオブランкиングの開始あるいは終了時点等の出画タイミングと一定の時間差にある部分で行うことにより適切な処理を行うことが可能である。またソフトウェアにより実装する場合、出画タイミングと一定の時間差にある部分で割り込みを発生させるなどして時間指標の読み込みルーチンを呼び出せばよい。

#### 【0103】

次に映像が3つの場合について説明する。映像が3つの場合でも、各映像について映像が1つの場合とほぼ同様の処理を行えばよい。但し、時間指標の読み込みについては注意が必要となる。図35は時間指標の読み込みについても、映像が1つの場合の映像pと3つの映像p1, p2, p3とで同様に行う処理を示している。この場合各々の映像の出画決定処理で時間指標が読み込まれるため、各映像で出画タイミングが異なることとなる。

ところが図35に示すように、3つの映像を1つの出画デバイスに出力する場合、実際の出画タイミングは各映像で共通となるため、各映像の出画映像決定処理で求めた出画タイミングは不適切なものとなる。従って、映像が3つの場合には、時間指標の読み込みについては各映像の出画映像決定処理で行うのではなく、図36、図37に示すように映像が1つの場合と同様に特定の時間Tに読み込み（図33のステップS801参照）（請求項5, 16, 27, 38に対応）、各映像の出画映像決定処理d0, d1, d2では予めメモリに読み込んだ時間指標を採用することにより（図33のステップS802参照）（請求項6, 17, 28, 39に対応）出力の制御を、適切な処理で行うことが可能である。このことは特に、同時に複数のアングルから撮影して生成した符号化データを復号して1つの映像に合成して出画する場合に有効である。

#### 【0104】

##### [デコード順制御]

次に、図38ないし図43を用いて3つの映像をデコードする場合の処理時間に関して説明する。

図38は3つの480iの映像をデコードした場合の処理時間を示す例である。各映像はフレーム単位で符号化されているとする。従来例と同様に各映像のデコード処理がフレームの時間内に完了すればよく、図38は正常にデコードが行われている状況をあらわす。

#### 【0105】

図39、図40は3つの映像のうち2つが480iの映像で、1つが480p、即ち、走査線480本のプログレッシブの映像、をデコードした場合を表わしている。いずれの映像もフレームを単位として符号化されているとすると、フレームの期間内にデコードを完了すればよい。但し、480iではフレームの期間は2Vの期間に相当し、480pでは1V期間に相当する。従って、480pの映像は480iの半分の時間内にデコードを完了しなければならない。図39は正常にデコードが行われている場合を表しているが、図40の状況では480pの映像のデコードを行っているチャネル2の復号処理がフレームの期間内に完了していない。このことから、複数の符号化されたデータを復号する場合、デコード処理を早期に完了する必要があるものがあるか否かを判定し（図41のステップS901参照）、早期に完了する必要があるものを優先的に処理することにより（図41のステップS902, S903参照）、必要とする処理能力を削減可能であるといえる。なお、デコード順序を早期に完了する必要があるか否かの判定には、出画順やフレームの期間がどれくらいであるかを示す情報、およびデコードを行う時間指標などから行う。また、デコード処理を早期に完了する必要がない場合、デコード処理は通常の所定の順序、即ち、復号処理の種類を単位として行う（図41のステップS904参照）。これにより、複数の符号化データを1つのデコード回路で復号することがより少ない資源で可能となる。

#### 【0106】

また、ここではデコードの順序を、映像が早期にデコードする必要があるか否かに基づいて決定する方法について述べたが、ユーザ情報設定回路からシステム制御回路に映像の優先順を設定し、この優先順に基づいて決定してもよい。これにより視聴者によりいずれの符号化データを優先して復号するかが選択可能になる。また放送の運用規定で放送時のチャネルによる優先順位づけ（ハイビジョンの放送と複数のNTSCの放送が連続的に放送された場合に、複数のNTSCチャネルのうちデフォルトで選択して復号を継続するものを優先順の先頭とする）に基づいてシステム制御回路に映像の優先順を設定することも可能である。また更に、ビットストリーム中にチャネルの優先順が記載されている場合についても、ビットストリーム出力回路により優先順を抽出し、システム制御回路により映像の優先順を設定することで優先順に基づいた複数の映像の復号が可能である。

#### 【0107】

##### [デコード期間管理]

図42は3つの480iの映像をデコードし、うち1つのデコードが正常でない場合をあらわしている。1つの映像のデコードがフレームの期間内に完了しなかった場合、他の映像のデコードにも影響が波及することがある。例えばデコード順で3つの映像のうち、チャ

ネル 0 の復号処理 d p 0 のように最初の映像で処理が所定の期間内に完了しなかった場合、チャネル 1, 2 の復号処理 d p 1, d p 2 のように残りの 2 つの映像が仮に通常にデコードできる状態にあったとしても、結果として所定の期間内にデコードを完了することができず、すべての映像がデコードできなかったことになる。これを改善するものが図 4 3 である。1 つの映像のデコード処理に通常よりも多くの時間を要する場合を記している。

【 0 1 0 8 】

図 4 4 はこの場合の一連の処理を示す。デコード処理（ステップ S 1 0 0 1 参照）が所定の期間内に終了しない場合（ステップ S 1 0 0 2 参照）、強制終了処理 f c によりデコードを強制終了し（ステップ S 1 0 0 3 参照）、デコード順で最後のチャネルの処理を完了していれば（ステップ S 1 0 0 4 参照）、デコード回路の状態を退避し（ステップ S 1 0 0 6 参照）他のチャネルに処理を移す（ステップ S 1 0 0 4 参照）（請求項 7, 18, 29, 40 に対応）。一方、デコード順で最後のチャネルの処理を完了していなければ（ステップ S 1 0 0 4 参照）、デコード回路の状態を退避・復元し（ステップ S 1 0 0 5 参照）、当初のデコード処理（ステップ S 1 0 0 1 ）に戻る。

【 0 1 0 9 】

この処理では、他の映像のデコードに影響を与えないために、デコードする各映像に対してそれぞれデコードに許容する最大の時間を設定し、この時間内にデコードが完了しない場合には強制的にデコード処理を終了することで他の映像のデコードへの影響を抑制する。この処理は、例えば、タイマなどを利用することにより実現が可能である。これにより、1 つの符号化データの復号処理で不具合が発生した場合でも他の符号化データの復号処理を正常に行なうことが可能となり、正常な映像のデコードは他の映像のデコードによる影響を受けることなくデコードが可能となる。

【 0 1 1 0 】

[ デコード再開処理 ]

一方、強制的にデコード処理を終了した映像に関してはデコードの復帰処理を行う。

図 4 5 に示すように、デコード中の映像に関して完全にデコード回路の状態を退避・復元可能であれば（ステップ S 1 1 0 1 参照）、デコード処理の強制終了を含む各チャネルのデコード処理後に、強制終了を行ったチャネルのデコードを再度試みるために、対象となるチャネルのデコード順序を決定し（ステップ S 1 1 0 2 参照）デコードを行うチャネルのデコード回路の状態を復元し（ステップ S 1 1 0 3 参照）、デコード処理を行う（ステップ S 1 1 0 4 参照）。その際、許容される最大限の時間（例えばフレーム期間の最大）までデコード処理に費やすことができ（ステップ S 1 1 0 5 参照）、正常な復号化が行えなかった符号化データに関しても復号化を再開することが可能になる。これに対し許容される最大限の時間に到達せずデコード処理を完了した場合には、デコード回路の状態を退避してデコード対象のチャネルを切り替える（ステップ S 1 1 0 6 参照）。これらステップ S 1 1 0 3 ないし S 1 1 0 6 の処理を繰り返すことにより、全てのチャネルについてデコードを行うことができる（ステップ S 1 1 0 7 参照）。またデコードの再開処理を行う際に、複数の映像で強制的なデコード処理の終了を実施していた場合、デコードの再開処理を行うチャネルの順番の決定はもとのデコード順を優先するあるいはバッファ残量等の他の要因を考慮してデコード可能性を判断して決定した優先順位に基づくなどにより行う。但し、強制的なデコード処理の終了を行った場合は、次に示すデコード処理の再開位置の変更と同じ処理を行う。

【 0 1 1 1 】

これに対し、デコード中の映像に関しての状態の退避・復元をサポートしていない場合（ステップ S 1 1 0 1 参照）、あるいは再度デコードを行ってもデコード処理が正常に終了しなかった場合、デコード対象の映像は破棄することになる。またこの場合、デコードの再開処理が必要になる（ステップ S 1 1 0 8 参照）。破棄した映像が後の映像を復号する際に参照されるか否かによって、デコード処理を再開する位置を変更する（請求項 8, 19, 30, 41 に対応）。これは復号化で参照する映像が存在しない場合、適切な復号化ができないために乱れた映像を出画する可能性があるためである。まず破棄した映像が

後のデコードで参照される可能性がない場合、デコード処理の再開は直後の映像から可能である。他方破棄した映像が後のデコードで参照される可能性がある場合、画面内符号化の映像の位置あるいは映像シーケンスの先頭をあらわす位置からデコード処理を再開する。ここでは破棄した映像の種類に基づいてデコードの再開する位置を変更する方法について述べたが、要望に応じて変更することが望ましい。例えば画面内符号化の映像や映像シーケンスの先頭が頻繁に挿入されている場合には、デコード処理の再開が直後の映像から可能でありながら、画面内符号化の映像や映像シーケンスの先頭の位置からデコードを再開しても、視聴者にとってこの時間差は大した問題では無いと考えられる。このような場合には処理を簡略化し、破棄した映像の種類によらず、デコード処理の再開は常に画面内符号化の映像の位置や映像シーケンスの先頭の位置とすることができます。

#### 【0112】

次にデコードを開始・再開する場合のデコード開始位置の検索について説明する。図46が典型的なデコード開始の例である。まず、図47に示すように、デコードを開始する映像を決定して、その映像に対応したヘッダ情報をヘッダサーチ<sub>hs</sub>処理により検索する(ステップS1201参照)。対象となる映像のヘッダ情報を発見した段階でデコードを開始する(ステップS1202参照)。これが基本的な考え方であるが、図48を用いてもう少し具体的な例で説明を行う。デコード開始の対象となるヘッダの種類としてはレベル0, 1, 2が存在するとし、現状ではデコーダが読み込む位置<sub>cs</sub>はヘッダ0で、ビデオビットストリームバッファにはビデオストリーム1の途中までが格納されているとする。このとき、図48に示すように、デコードを開始する対象をレベル0のヘッダとすると、レベル0のヘッダとして最初に出現するヘッダ3の先頭<sub>ds</sub>からデコードを開始することになる。このときの動作は図49に示す通り、まず、次のヘッダを検出し(ステップS1301参照)、このヘッダが検出単位に対するヘッダか否かを判断する(ステップS1302参照)。この判断が否であればステップS1301に戻る。一方、検出単位に対するヘッダであればデコード対象となるヘッダか否かを判断する(ステップS1303参照)。検出単位に対するヘッダであればデコード処理を行う(ステップS1304参照)が、否であれば連続的に検出処理を実施するか否かを判断する(ステップS1305参照)。ここで、連続的に検出処理を行うのであれば、ステップS1301に戻る。

#### 【0113】

ところで、図48に示すように、ビデオストリームバッファに格納されている末尾の位置<sub>1p</sub>がビデオストリーム1の途中であるとすると、ヘッダ3が見つかるまで検索を続けた場合、まだビデオビットストリームバッファに格納されていないデータに対しても連続的に検索処理を行うことになる。具体的には、通常ビットストリーム出力回路からのデータ供給よりも、デコード回路による検索処理のデータ消費の方が速いので、ビデオビットストリームバッファに格納されていないデータのほとんどに対して検索処理はデータの供給待ちとなる。即ち、この期間Aでは、ヘッダサーチでストリームを連続的に読み込んだ場合、アンダーフローの状態でサーチを続けることになる。この結果、検索処理を行い、対象となるヘッダを発見後に続けてデコード処理を行う場合、予定した期間内にデコード処理が完了することは保証できない。即ち、期間Bでは、必ずしも所定の期間内にビデオストリーム3のデコードが完了するとは限らない。これを改善する方法として、所定の期間内にデコード処理が完了しない場合でもデコードを継続する、また(不具合が無い限り)検出処理を連続的に行わず、図49に示すように、デコード処理が予定の期間内に完了可能であることを保証する(ステップS1306参照)、などの方法がある。また、デコード対象となるヘッダを検出し、1つの符号化単位を復号した後は通常のデコード動作となる。ただし、デコード処理が完了しない場合でもデコードを継続する方法では複数の映像をデコードする場合には前述したようにデコードに許容する最大の時間を設けるため機能の整合がとれない。デコード処理が予定の期間内に完了可能であることを保証する方法(ステップS1306参照)の例としてはヘッダサーチを連続的に行わずに映像単位で行うなどがある(図50参照)(請求項9, 20, 31, 42に対応)。フレームを単位とする映像の場合にはフレーム期間に1度ヘッダサーチを行う。1フレームに2つのヘッダ

が存在する場合には、1度のヘッダサーチで2つのヘッダを読み込む。デコードすべき時刻等に基づいてヘッダサーチを実施するタイミングを決定するが、1フレーム期間に1回に限定する等、特定の条件下では簡略化が可能である。このとき映像単位がフレームで符号化単位がフィールドである場合には、図51に示すように、検索対象でない映像に対しては1度のヘッダサーチで2つのヘッダを読み飛ばすことになる（ステップS1401参照）。また、ビデオビットストリーム中にデコード映像を繰り返し表示する指示があった場合は、出画順で表示を繰り返すことを考慮して、ヘッダサーチの実施間隔を制御する。また符号化した映像に出画の時間指標だけでなくデコードタイミングの時間指標が付随する場合には、ヘッダサーチのタイミングをこれに同期させてもよく（請求項11, 22, 33, 44に対応）、これにより、適切なタイミングで復号化を開始することが可能となる。またビデオビットストリーム中にバッファに関する情報が適切に付加されている場合には、バッファに関する情報に基づいて算出したバッファ残量分のビデオビットストリームがビデオビットストリームバッファに入力されるタイミングでデコードを再開してもよい（請求項11, 22, 33, 44に対応）。またこの場合はデコード開始位置の検索を従来の方法で行うことができる。

#### 【0114】

##### [デコード処理最適化]

次に、デコード回路の状態退避・復帰とソフトウェア実装時のプログラムロードの観点からデコードの処理順について説明する。図52はこれまでの複数映像のデコード処理に加え、各映像のデコードを制御する部分d\_cを追記したものである。これに更にソフトウェア実装時のプログラムのロードを記入したものが図53になる。この手順は図54に示すように、デコード回路の状態待避を復帰するタイミングか否かを判定し（ステップS1501参照）、復帰するタイミングTRであればロード動作cp1により制御プログラムをロードし（ステップS1502参照）、デコード回路の状態復帰を行う（ステップS1503参照）。次いで、デコード制御を実施し（ステップS1504参照）、ロード動作dp1によりデコードプログラムをロードして（ステップS1505参照）、実際のデコード処理を行った後（ステップS1506参照）、デコードするチャネルを切り換える（ステップS1507参照）。これより、デコード回路の状態退避・復帰の観点からは、デコード処理順としてデコードの制御と実際のデコード処理を連続して行うことで処理を削減することができる。ところが、ソフトウェア実装時のプログラムロードの観点からは、このデコード処理順は適しているとはいえない。次に、ソフトウェア実装時について述べる。

#### 【0115】

デコード回路をDSPなどにより実現する場合、デコード処理はプログラムにより実装されることになり、プログラムを格納する内部のメモリ容量を削減することが望まれる。映像のデコードをプログラムにより実装した場合、実際にデコードの処理を行う部分とデコードの処理を制御する部分等、復号処理の種類によりプログラムの分離が可能な場合が多い。これをを利用して、各分離したプログラムを必要なときにロードすることにより格納するメモリ量を削減することができる。このときプログラムロードの観点から改善したデコード処理順が図55である（請求項10, 21, 32, 43に対応）。この場合プログラムロードの観点からは処理が削減されているが、デコード回路の状態退避・復帰の観点では処理が増えている。

#### 【0116】

このように、状態退避・復帰に関わる処理量とプログラムロードの処理量はトレードオフの関係にあり、それぞれの処理に要する性能によっていずれを優先するかを選択することが望ましい。またデコード処理を行うDSPを設計する場合には、状態退避・復帰用のメモリおよびプログラム用のメモリをどの程度の容量まで内蔵するか等を性能の観点から最適化を行うことができる。

なお、ここではプログラムをデコード制御と実際のデコード処理の2つに分離する例をあげたが、起動時の初期化部分や、エラー時の処理等、プログラムの分離が可能であれば

どのように分離してもよい。例えば、通常のデコードでは発生頻度が少ないことが期待されるエラーやスキップ等の処理を行うプログラムを分離し、エラー等の発生時に別途コードすることで、さらにプログラムサイズをコンパクトにすることができる。

#### 【 0 1 1 7 】

##### 【発明の効果】

以上のように、本発明の復号装置によれば、符号化データを入力し、復号して出力するデコード部を有する復号装置において、符号化データを入力し、復号して出力するデコード部の複数の状態を退避・復元するデコード状態保持部と、複数の符号化データのうち、復号対象となる符号化データを選択して復号し、デコード状態保持部に対して、前記デコード部の状態の退避・復元を行い、復号した複数の復号結果をそれぞれに対応した出力先に出力し、一連の処理を各々の符号化データに対応した所定の期間内に完了するデコード部とを備えるようにしたので、デコードすべき複数の符号化データに対し時分割でデコードを行うことが可能となり、1つのデコード部により複数の符号化データをみかけ上同時に復号することが可能となる。

#### 【 0 1 1 8 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、符号化単位の境界で符号化データの入力を変更して複数の符号化データを復号化するデコード部を備えるようにしたので、符号化単位の境界以外でデコードすべき符号化データを切り替えるよりも、デコード処理の負荷が軽くなり、1つのデコード部により複数の符号化データをみかけ上同時に復号することが容易な処理で可能となる。

#### 【 0 1 1 9 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、各々の符号化データに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するデコード部を備えるようにしたので、M P E G 2 におけるスライスのヘッダ等、所定のタイミングで処理の対象となるビデオストリームを切り替えることができ、復号する順序を適切にすることで、処理性能が低い場合であっても1つのデコード回路で複数の符号化データを同時にデコードでき、1つのデコード部により複数の符号化データをみかけ上同時に復号することが、より低い処理性能で実現可能となる。

#### 【 0 1 2 0 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、復号結果を格納するメモリと、いずれの符号化データを復号するかを指示するシステム制御部と、メモリに格納した復号結果を出力する出力制御部とを備え、出力制御部は、システム制御部の指示に基づいて復号する符号化データを変更し、前記変更に伴ってメモリ中のデコード結果を格納する配置が変更になる場合に、所定の出を行なうようにしたので、例えば複数の映像を復号する処理と、より大きなサイズの1つの映像を復号する処理が連続した場合、フレームメモリ内部の各バッファの配置が変更になり、出画に関するバッファへの影響が考えられるので、配置変更後、正常な出力が可能になるまで背景色を出力することにより、復号対象を変更した場合にも不正な出力を抑制することが可能となる。

#### 【 0 1 2 1 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部と、復号結果を格納するメモリと、メモリに格納した復号結果を出力する出力制御部とを備え、出力制御部は各符号化データの復号結果を出力する際に、共通の時間指標に基づいて出力を制御するようにしたので、例えば3つの映像を1つの出画デバイスに出力する場合等で、時間指標の読み込みを映像が1つの場合と同様に特定の時間に読み込むことにより、出力の制御を適切に行なうことが可能となる。

#### 【 0 1 2 2 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化

データを復号するデコード部と、復号結果を格納するメモリと、メモリに格納した復号結果を出力する出力制御部とを備え、出力制御部は各符号化データの復号結果の出力を制御する際に、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたので、それぞれの映像の時間指標が異なる場合に、1つのチャンネルに対する時間指標を保持して他のチャンネルはその時間差分を記憶しておくことで時間指標の算出が可能となり、各チャンネルで時間指標が異なる符号化データに対しても、対応した時間指標を求めて出力の制御を適切に行なうことが、容易な処理で可能となる。

#### 【 0 1 2 3 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するデコード部を備え、デコード処理が所定の時間内に終了しない場合はデコードを強制終了するようにしたので、1つの符号化データの復号処理で不具合が発生した場合でも他の符号化データの復号処理を正常に行なうことが可能となる。

#### 【 0 1 2 4 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するデコード部を備えるようにしたので、復号化で参照する映像が存在しない場合に適切な復号化ができないために乱れた映像を出画する可能性があり、廃棄した映像が後の映像を復号する際に参照されるか否かによってデコード処理を再開する位置を変更するようにしたので、正常な復号化が行えなかった符号化データに関する、復号化を再開することが可能となる。

#### 【 0 1 2 5 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うデコード部を備えるようにしたので、ヘッダーサーチを連続的に行なわずに映像単位で行なうことにより、復号化の開始を適切に行なうことが可能となる。

#### 【 0 1 2 6 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、各符号化データの復号処理を復号処理の種類を単位として切り替えるデコード部を備えるようにしたので、デコード回路の行なう処理を複数に分け、それぞれの段階に適した処理を行なうように実装することで、複数の符号化データを1つのデコード部で復号することがより少ない資源で可能となる。

#### 【 0 1 2 7 】

また、本発明の復号装置によれば、複数の符号化データを入力して、それぞれの符号化データを復号するデコード部を有する復号装置において、復号を開始するタイミングは各符号化データに記された情報に基づいて決定するデコード部を備えるようにしたので、符号化した映像に出画の時間指標だけでなくデコードタイミングの時間指標が付随する場合は、ヘッダーサーチのタイミングをこれに同期させてもよく、適切なタイミングで復号化を開始することが可能となる。

#### 【 0 1 2 8 】

また、本発明の復号方法によれば、符号化データを入力し、復号して出力する復号方法において、復号対象である符号化データの復号を中断する際に、復号を行うデコード部の状態を待避し、復号対象である符号化データを他の符号化データに切り換えて復号を行な

、復号を中断した符号化データの復号の再開は、待避したデコード部の状態を復元した後に行なうようにしたので、デコードすべき複数の符号化データに対し時分割でデコードを行うことが可能となり、1つのデコード処理により複数の符号化データをみかけ上同時に復号することが可能となる。

#### 【0129】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、符号化単位の境界で符号化データの入力を復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたので、符号化単位の境界以外でデコードすべき符号化データを切り替えるよりも、デコード処理の負荷が軽くなり、1つのデコード処理により複数の符号化データをみかけ上同時に復号することが容易な処理で可能となる。

#### 【0130】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、各々の符号化データに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたので、MPEG2におけるスライスのヘッダ等、所定のタイミングで処理の対象となるビデオストリームを切り替えることができ、復号する順序を適切にすることで、処理性能が低い場合であっても1つのデコード回路で複数の符号化データを同時にデコードでき、1つのデコード処理により複数の符号化データをみかけ上同時に復号することが、より低い処理性能で実現可能となる。

#### 【0131】

また、本発明の復号方法によれば、複数の符号化データ中の復号すべき符号化データを指示に基づいて変更し、前記復号すべき符号化データを復号し、復号結果をメモリに格納し、前記指示による復号すべき符号化データの変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出力を行なうようにしたので、例えば複数の映像を復号する処理と、より大きなサイズの1つの映像を復号する処理が連続した場合、フレームメモリ内部の各バッファの配置が変更になり、出画に関するバッファへの影響が考えられるので、配置変更後、正常な出力が可能になるまで背景色を出力することにより、復号対象を変更した場合にも不正な出力を抑制することが可能となる。

#### 【0132】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、共通の時間指標に基づいて出力を制御するようにしたので、例えば3つの映像を1つの出画デバイスに出力する場合等で、時間指標の読み込みを映像が1つの場合と同様に特定の時間に読み込むことにより、出力の制御を適切に行なうことが可能となる。

#### 【0133】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたので、それぞれの映像の時間指標が異なる場合に、1つのチャンネルに対する時間指標を保持して他のチャンネルはその時間差分を記憶しておくことで時間指標の算出が可能となり、各チャンネルで時間指標が異なる符号化データに対しても、対応した時間指標を求めて出力の制御を適切に行なうことが、容易な処理で可能となる。

#### 【0134】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了する

ようにしたので、デコード処理が所定の時間内に終了しない場合はデコードが強制終了するため、1つの符号化データの復号処理で不具合が発生した場合でも他の符号化データの復号処理を正常に行うことが可能となる。

#### 【 0 1 3 5 】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する際、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたので、復号化で参照する映像が存在しない場合に適切な復号化ができないために乱れた映像を出画する可能性があり、廃棄した映像が後の映像を復号する際に参照されるか否かによってデコード処理を再開する位置を変更するようにしたので、1つの符号化データの復号処理で不具合が発生した場合でも他の符号化データの復号処理を正常に行うことが可能となる。

#### 【 0 1 3 6 】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたので、ヘッダーサーチを連続的に行わずに映像単位で行うことにより、復号化の開始を適切に行うことが可能となる。

#### 【 0 1 3 7 】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたので、デコード回路の行う処理を複数に分け、それぞれの段階に適した処理を行うように実装することで、複数の符号化データを1つのデコード部で復号することがより少ない資源で可能となる。

#### 【 0 1 3 8 】

また、本発明の復号方法によれば、複数の符号化データを入力して、それぞれの符号化データを復号する際、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたので、符号化した映像に出画の時間指標だけでなくデコードタイミングの時間指標が付随する場合は、ヘッダーサーチのタイミングをこれに同期させてよく、適切なタイミングで復号化を開始することが可能となる。

#### 【 0 1 3 9 】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、符号化データを入力し、復号して出力する復号方法であって、復号対象である符号化データの復号を中断する際に、復号を行うデコード部の状態を待避し、復号対象である符号化データを他の符号化データに切り換えて復号を行い、復号を中断した符号化データの復号の再開は、待避したデコード部の状態を復元した後に行うようにしたので、デコードすべき複数の符号化データに対し時分割でデコードを行うことが可能となり、1つのデコード処理により複数の符号化データをみかけ上同時に復号することが可能となる。

#### 【 0 1 4 0 】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、符号化単位の境界で符号化データの入力を復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたので、符号化単位の境界以外でデコードすべき符号化データを切り替えるよりも、デコード処理の負荷が軽くなり、1つのデコード処理により複数の符号化データをみかけ上同時に復号することが容易な処理で可能となる。

#### 【 0 1 4 1 】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化

データを復号する復号方法であって、前記復号は、各々の符号化データに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたので、MPEG2におけるスライスのヘッダ等、所定のタイミングで処理の対象となるビデオストリームを切り替えることができ、復号する順序を適切にすることで、処理性能が低い場合であっても1つのデコード回路で複数の符号化データを同時にデコードでき、1つのデコード処理により複数の符号化データをみかけ上同時に復号することが、より低い処理性能で実現可能となる。

#### 【0142】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データ中の復号すべき符号化データを指示に基づいて変更し、前記復号すべき符号化データを復号し、復号結果をメモリに格納し、前記指示による復号すべき符号化データの変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出を行なうようにしたので、例えば複数の映像を復号する処理と、より大きなサイズの1つの映像を復号する処理が連続した場合、フレームメモリ内部の各バッファの配置が変更になり、出画に関するバッファへの影響が考えられるので、配置変更後、正常な出力が可能になるまで背景色を出力することにより、復号対象を変更した場合にも不正な出力を抑制することが可能となる。

#### 【0143】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、共通の時間指標に基づいて出力を制御するようにしたので、例えば3つの映像を1つの出画デバイスに出力する場合等で、時間指標の読み込みを映像が1つの場合と同様に特定の時間に読み込むことにより、出力の制御を適切に行なうことが可能となる。

#### 【0144】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたので、それぞれの映像の時間指標が異なる場合に、1つのチャンネルに対する時間指標を保持して他のチャンネルはその時間差分を記憶しておくことで時間指標の算出が可能となり、各チャンネルで時間指標が異なる符号化データに対しても、対応した時間指標を求めて出力の制御を適切に行なうことが、容易な処理で可能となる。

#### 【0145】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するようにしたので、デコード処理が所定の時間内に終了しない場合はデコードが強制終了するため、1つの符号化データの復号処理で不具合が発生した場合でも他の符号化データの復号処理を正常に行なうことが可能となる。

#### 【0146】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、任意の符号化データの符号化単位について正常な復号化が行えなかつた場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたので、復号化で参照する映像が存在しない場合に適切な復号化ができないために乱

れた映像を出画する可能性があり、廃棄した映像が後の映像を復号する際に参照されるか否かによってデコード処理を再開する位置を変更するようにしたので、正常な復号化が行えなかった符号化データに関しては、復号化を再開することが可能となる。

#### 【 0 1 4 7 】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたので、ヘッダーサーチを連続的に行わずに映像単位で行うことにより、復号化の開始を適切に行うことが可能となる。

#### 【 0 1 4 8 】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたので、デコード回路の行う処理を複数に分け、それぞれの段階に適した処理を行うように実装することで、複数の符号化データを1つのデコード処理で復号することがより少ない資源で可能となる。

#### 【 0 1 4 9 】

また、本発明の復号プログラムによれば、コンピュータに復号方法を実行させる復号プログラムにおいて、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたので、符号化した映像に出画の時間指標だけでなくデコードタイミングの時間指標が付随する場合は、ヘッダーサーチのタイミングをこれに同期させてよく、適切なタイミングで復号化を開始することが可能となる。

#### 【 0 1 5 0 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、符号化データを入力し、復号して出力する復号方法であって、復号対象である符号化データの復号を中断する際に、復号を行うデコード部の状態を待避し、復号対象である符号化データを他の符号化データに切り替えて復号を行い、復号を中断した符号化データの復号の再開は、待避したデコード部の状態を復元した後に行うようにしたので、デコードすべき複数の符号化データに対し時分割でデコードを行うことが可能となり、1つのデコード部により複数の符号化データをみかけ上同時に復号することが可能となる。

#### 【 0 1 5 1 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、符号化単位の境界で符号化データの入力を復号中の符号化データから他の符号化データに変更して複数の符号化データを復号化するようにしたので、符号化単位の境界以外でデコードすべき符号化データを切り替えるよりも、デコード処理の負荷が軽くなり、1つのデコード部により複数の符号化データをみかけ上同時に復号することが容易な処理で可能となる。

#### 【 0 1 5 2 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する復号方法であって、前記復号は、各々の符号化データに記された情報に基づいて、出力すべき順序を求め、求めた順序に従って符号化データを選択して復号化するようにしたので、M P E G 2 におけるスライスのヘッダ等、所定のタイミングで処理の対象となるビデオストリームを切り替えることができ、復号する順序を適切にすることで、処理性能が低い場合であっても1つのデコード回路で複数の符号化データを同時にデコードでき、1つのデコード部により複数の符号化データをみ

かけ上同時に復号することが、より低い処理性能で実現可能となる。

#### 【 0 1 5 3 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データ中の復号すべき符号化データを指示に基づいて変更し、前記復号すべき符号化データを復号し、復号結果をメモリに格納し、前記指示による復号すべき符号化データの変更に伴って前記メモリ中のデコード結果を格納する配置が変更になる場合に、所定の出力をを行うようにしたので、例えば複数の映像を復号する処理と、より大きなサイズの1つの映像を復号する処理が連続した場合、フレームメモリ内部の各バッファの配置が変更になり、出画に関するバッファへの影響が考えられるので、配置変更後、正常な出力が可能になるまで背景色を出力することにより、復号対象を変更した場合にも不正な出力を抑制することが可能となる。

#### 【 0 1 5 4 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、共通の時間指標に基づいて出力を制御するようにしたので、例えば3つの映像を1つの出画デバイスに出力する場合等で、時間指標の読み込みを映像が1つの場合と同様に特定の時間に読み込むことにより、出力の制御を適切に行なうことが可能となる。

#### 【 0 1 5 5 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号し、その復号結果をメモリに格納し、前記メモリから復号結果を出力する際、所定のタイミングで時間指標を読み込み、前記時間指標から各符号化データに対応した時間指標を求め、前記各符号化データに対応した時間指標に基づいて出力を制御するようにしたので、それぞれの映像の時間指標が異なる場合に、1つのチャンネルに対する時間指標を保持して他のチャンネルはその時間差分を記憶しておくことで時間指標の算出が可能となり、各チャンネルで時間指標が異なる符号化データに対しても、対応した時間指標を求めて出力の制御を適切に行なうことが、容易な処理で可能となる。

#### 【 0 1 5 6 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データに記された情報から求めた符号化単位を復号化する際に許容される時間と復号化する符号化データの数に基づいて、各符号化データを復号化する際に使用する最大の復号化時間を求め、各符号化データを復号化する際に、前記最大の復号化時間に到達した場合には当該符号化データの符号化単位の復号化処理を終了するようにしたので、デコード処理が所定の時間内に終了しない場合はデコードが強制終了するため、1つの符号化データの復号処理で不具合が発生した場合でも他の符号化データの復号処理を正常に行なうことが可能となる。

#### 【 0 1 5 7 】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、任意の符号化データの符号化単位について正常な復号化が行えなかった場合、同符号化単位の符号化の種類に基づいて復号化を再開する位置を決定するようにしたので、復号化で参照する映像が存在しない場合に適切な復号化ができないために乱れた映像を出画する可能性があり、廃棄した映像が後の映像を復号する際に参照されるか否かによってデコード処理を再開する位置を変更するようにしたので、正常な復号化が行えなかった符号化データに関しても、復号化を再開すること

が可能となる。

【0158】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号化を開始する位置を検索する際に、符号化データに記された情報に基づいて求めたタイミングで符号化単位ごとに検索を行うようにしたので、ヘッダーサーチを連続的に行わずに映像単位で行うことにより、復号化の開始を適切に行うことが可能となる。

【0159】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、各符号化データの復号処理を、復号処理の種類を単位として切り替えるようにしたので、デコード回路の行う処理を複数に分け、それぞれの段階に適した処理を行うように実装することで、複数の符号化データを1つのデコード処理で復号することがより少ない資源で可能となる。

【0160】

また、本発明の復号プログラム記録媒体によれば、コンピュータに復号方法を実行させる復号プログラムを記録した記録媒体において、前記復号方法は、複数の符号化データを入力して、それぞれの符号化データを復号する際、復号を開始するタイミングを、各符号化データに記された情報に基づいて決定するようにしたので、符号化した映像に出画の時間指標だけでなくデコードタイミングの時間指標が付随する場合は、ヘッダーサーチのタイミングをこれに同期させてもよく、適切なタイミングで復号化を開始することが可能となる。

【図面の簡単な説明】

【図1】

本発明の同時データ復号装置の構成を示すブロック図である。

【図2】

本発明の同時データ復号装置の入力部分の具体的な構成を示すブロック図である。

【図3】

複数のビットストリームを説明するための図である。

【図4】

複数のビットストリームを分離した結果を説明するための図である。

【図5】

ビットストリーム出力回路の構成例1を示すブロック図である。

【図6】

ビットストリーム出力回路の構成例2を示すブロック図である。

【図7】

ビットストリーム出力回路の構成例2の中で生成されるビットストリームを示す図である。

【図8】

ビットストリームを生成する際の時間情報の修正を示す図である。

【図9】

ビットストリームを生成する際の指示情報の修正を示す図である。

【図10】

ビットストリーム出力回路の構成例3を示すブロック図である。

【図11】

ビットストリーム出力回路の構成例4を示すブロック図である。

【図12】

ビットストリーム出力回路の構成例5を示すブロック図である。

【図13】

ビットストリーム出力回路の構成例 6 を示すブロック図である。

【図 14】

本発明の同時データ復号装置のデコード部分の具体的な構成を示すブロック図である。

【図 15】

図 14 のデコード部分のうちビデオの復号に関する部分のみ抽出した図である。

【図 16】

ビデオビットストリームの例を示す図である。

【図 17】

複数のビデオビットストリームを 1 つのデコード回路で時分割して処理する例を示す図である。

【図 18】

複数のビデオビットストリームを 1 つのデコード回路で時分割して処理する場合の処理の流れを表すフローチャートを示す図である。

【図 19】

DSP の内部構造を示すブロック図である。

【図 20】

図 17 の時分割するタイミングを変更した例を示す図である。

【図 21】

本発明の同時データ復号装置の出力部分の具体的な構成を示すブロック図である。

【図 22】

オーディオ出力を生成する場合の処理の流れを表すフローチャートを示す図である。

【図 23】

ビデオ出力を生成する場合の処理の流れを表すフローチャートを示す図である。

【図 24】

本発明の同時データ復号装置により 1 つの映像と音声をデコードする場合の入力部分を示すブロック図である。

【図 25】

本発明の同時データ復号装置により 1 つの映像と音声をデコードする場合のデコード部分を示すブロック図である。

【図 26】

本発明の同時データ復号装置により 1 つの映像と音声をデコードする場合の出力部分を示すブロック図である。

【図 27】

入力の符号データが変化する場合にメモリ配置を変更する場合の一例を示す図である。

【図 28】

1 つの映像を出画する場合の処理を示す図である。

【図 29】

1 つの映像を出画する場合の処理の流れを表すフローチャートを示す図である。

【図 30】

スキップ処理の流れを表すフローチャートを示す図である。

【図 31】

フリーズ処理の流れを表すフローチャートを示す図である。

【図 32】

各映像の出画の時間指標に誤りが発生する可能性がある場合の処理の流れを表すフローチャートを示す図である。

【図 33】

映像が複数の場合の時間指標の読み込み動作を表すフローチャートを示す図である。

【図 34】

3 つの映像を出画する場合の処理を示す図である。

【図 35】

出画映像決定タイミングについて説明するための図である。

【図36】

出画映像決定タイミングに用いる時間指標について説明するための図である。

【図37】

図40の時間指標に基づいて出画映像を決定する例を示す図である。

【図38】

3つの480i映像を正常にデコードしている例を示す図である。

【図39】

2つの480i映像と1つの480p映像を正常にデコードしている例を示す図である

。

【図40】

2つの480i映像と1つの480p映像を正常にデコードできない例を示す図である

。

【図41】

デコードに要する時間が異なる映像をデコードする例を表すフローチャートを示す図である。

【図42】

3つの映像のデコードが正常に行えない例を示す図である。

【図43】

デコードの処理を強制終了させる例を示す図である。

【図44】

複数の映像の中に所定の期間内にデコードを完了することができないものが存在する場合の処理の流れを表すフローチャートを示す図である。

【図45】

デコードの復帰処理の流れを表すフローチャートを示す図である。

【図46】

ヘッダサーチに続けてデコードを行う処理を示す図である。

【図47】

デコード開始位置の検索を表すフローチャートを示す図である

【図48】

ヘッダサーチに続けてデコードを行う処理の詳細を示した図である。

【図49】

デコード開始位置の検索をより具体的に表すフローチャートを示す図である。

【図50】

ヘッダサーチを映像単位に行う例を示した図である。

【図51】

デコード処理が予定の期間内に完了可能であることを保証する手法の一例を表すフローチャートを示す図である。

【図52】

複数映像デコードにおいてデコード制御の処理を明記した図である。

【図53】

複数映像デコードをプログラムロードの観点でみた図である。

【図54】

デコード回路の状態待避を復帰するソフトウェア実装を表すフローチャートを示す図である。

【図55】

複数映像デコードの処理でプログラムロードの処理を削減した図である。

【図56】

従来の復号装置の構成を示すブロック図である。

【図57】

従来の復号装置の具体的な構成を示すブロック図である。

【図 5 8】

1 チャネルデコードを正常な処理時間内に完了することを説明するための図である。

【図 5 9】

1 チャネルデコードを正常な処理時間内に完了できない場合を説明するための図である

。

【符号の説明】

- 1 0 1 デコード回路
- 1 0 2 デコード状態保持回路
- 1 0 3 デコード回路
- 1 0 4 フレームメモリ
- 1 0 5 システム制御回路
- 1 0 6 出力制御回路
- 1 0 7 フレームメモリ
- 1 0 8 ビットストリーム出力回路
- 1 0 9 システム制御回路
- 1 1 0 オーディオ・ビデオ出力回路
- 1 1 1 ユーザ情報設定回路