(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2005 年12 月22 日 (22.12.2005)

PCT

(10) 国際公開番号
WO 2005/122569 A1

(51) 国際特許分類[7]: H04N 5/92, G11B 20/10, 20/12, 27/34

(21) 国際出願番号: PCT/JP2005/010901

(22) 国際出願日: 2005 年6 月8 日 (08.06.2005)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2004-174572 2004 年6 月11 日 (11.06.2004) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒1410001 東京都品川区北品川６丁目７番３５号 Tokyo (JP). 株式会社ソニー・コンピュータエンタテインメント (SONY COMPUTER ENTERTAINMENT INC.) [JP/JP]; 〒1070062 東京都港区南青山二丁目６番２１号 Tokyo (JP).

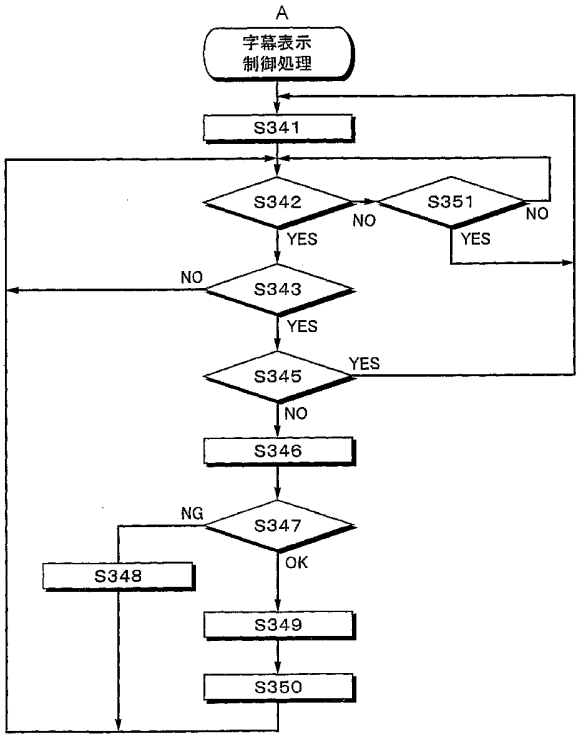
(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 藤波 靖 (FUJINAMI, Yasushi) [JP/JP]; 〒1410001 東京都品川区北品川６丁目７番３５号 ソニー株式会社内 Tokyo (JP). 高橋 邦明 (TAKAHASHI, Kuniaki) [JP/JP]; 〒1070062 東京都港区南青山二丁目６番２１号 株式会社ソニー・コンピュータエンタテインメント内 Tokyo (JP). 浜田 俊也 (HAMADA, Toshiya) [JP/JP]; 〒1410001 東京都品川区北品川６丁目７番３５号 ソニー株式会社内 Tokyo (JP).

[続葉有]

(54) Title: DATA PROCESSING DEVICE, DATA PROCESSING METHOD, PROGRAM, PROGRAM RECORDING MEDIUM, DATA RECORDING MEDIUM, AND DATA STRUCTURE

(54) 発明の名称: データ処理装置およびデータ処理方法、プログラムおよびプログラム記録媒体、データ記録媒体、ならびに、データ構造



A... SUBTITLE DISPLAY CONTROL

(57) Abstract: It is possible to control modification of the display method of subtitle data. Step S347 judges whether modification of the display of the subtitle data being reproduced is permitted according to configurable_flag indicating whether to permit modification of display of subtitle data contained in the subtitle data StaticInfo() from the default display method, among the StaticInfo() describing the information not changed during data reproduction for each data multiplexed in the program stream contained in the meta data concerning the program stream. When the modification of the display of the subtitle data being reproduced from the default display method is permitted, step S350 performs display of the subtitle data according to the instruction of the modification of the display method of the subtitle data. The present invention can be applied, for example, to a game device using a DVD.

(57) 要約: 字幕データの表示方式の変更を制御する。ステップＳ３４７において、プログラムストリームに関するメタデータに含まれる、プログラムストリームに多重化されているデータごとの、データの再生中に変化しない情報が記述されたStaticInfo()のうちの、字幕データのStaticInfo()に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表すconfigurable_flagに基づき、再生中の字幕データの表示を変更することが許可されているか否かが判定される。そして、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されている場合、ステップＳ３５０において、字幕データの表示方式の変更の指示にしたがって、字幕データの表示処理が行われる。本発明は、例えば、DVDを利用したゲーム装置などに適用できる。

WO 2005/122569 A1

(74) 代理人: 杉浦 正知, 外(SUGIURA, Masatomo et al.); 〒1710022 東京都豊島区南池袋 2丁目49番 7号 池袋パークビル７階 Tokyo (JP).

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): 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, HR, HU, ID, IL, IN, IS, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明 細 書

データ処理装置およびデータ処理方法、プログラムおよびプログラム記録媒体、データ記録媒体、ならびに、データ構造

技術分野

本発明は、データ処理装置およびデータ処理方法、プログラムおよびプログラム記録媒体、データ記録媒体、ならびに、に関し、特に、例えば、利便性等の高いデータ処理を可能とするデータ処理装置およびデータ処理方法、プログラムおよびプログラム記録媒体、データ記録媒体、ならびに、データ構造に関する。

背景技術

近年、大容量で、ランダムアクセスが可能な記録メディアとして、例えば、DVD(Digital Versatile Disc)が普及し、さらに、DVDを利用して各種の処理を行うDVD装置も広く普及している。

DVD装置としては、例えば、DVDに対して、テレビジョン放送番組のデータ等の記録再生を行うDVDレコーダや、DVDに地図情報等を記録し、その地図情報の表示を行うカーナビゲーションシステム、DVDにゲームのプログラム等を記録し、そのプログラムを実行するゲーム装置などがある。

なお、DVDについては、例えば、非特許文献1「DVD Specifications for Read-Only Disc Part 3; Version 1.1 Dcecmber 1997」に、その詳細が記載されている。

DVDのように、大量のデータを記録することができる記録メディアや、それを利用して各種の処理を行うDVD装置などには、そのような大量のデータについて、利便性等の高いデータ処理を行うことが要請

される。

発明の開示

本発明は、このような状況に鑑みてなされたものであり、利便性等の高いデータ処理を行うことができるようにするものである。

本発明のデータ処理装置は、少なくとも字幕データが多重化された多重化データに関するメタデータに含まれる、多重化データに多重化されているデータごとの、そのデータの再生中に変化しない不変情報のうちの、字幕データの不変情報を取得する取得手段と、取得手段において取得された不変情報に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報に基づき、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定手段と、判定手段において、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理手段とを備えることを特徴とする。

本発明のデータ処理方法は、少なくとも字幕データが多重化された多重化データに関するメタデータに含まれる、多重化データに多重化されているデータごとの、そのデータの再生中に変化しない不変情報のうちの、字幕データの不変情報を取得する取得ステップと、取得ステップにおいて取得された不変情報に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報に基づき、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップと、判定ステップにおいて、再生中の字幕データの表示をデフォルトの表

示方式から変更することが許可されていると判定された場合、字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップとを含むことを特徴とする。

本発明のプログラムは、少なくとも字幕データが多重化された多重化データに関するメタデータに含まれる、多重化データに多重化されているデータごとの、そのデータの再生中に変化しない不変情報のうちの、字幕データの不変情報を取得する取得ステップと、取得ステップにおいて取得された不変情報に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報に基づき、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップと、判定ステップにおいて、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップとを含むことを特徴とする。

本発明のプログラム記録媒体に記録されているプログラムは、少なくとも字幕データが多重化された多重化データに関するメタデータに含まれる、多重化データに多重化されているデータごとの、そのデータの再生中に変化しない不変情報のうちの、字幕データの不変情報を取得する取得ステップと、取得ステップにおいて取得された不変情報に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報に基づき、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップと、判定ステップにおいて、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、字幕データの表示方式の変更の指示にした

がって、その字幕データの表示処理を行う表示処理ステップとを含むことを特徴とする。

本発明のデータ記録媒体は、少なくとも字幕データが多重化された多重化データと、多重化データに関するメタデータとを含む記録データが記録されており、メタデータは、多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、字幕データの不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含むことを特徴とする。

本発明のデータ構造は、少なくとも字幕データが多重化された多重化データと、多重化データに関するメタデータとを含むデータのデータ構造において、メタデータは、多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、字幕データの不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含むことを特徴とする。

本発明のデータ処理装置およびデータ処理方法、並びにプログラムおよびプログラム記録媒体に記録されているプログラムにおいては、少なくとも字幕データが多重化された多重化データに関するメタデータに含まれる、多重化データに多重化されているデータごとの、そのデータの再生中に変化しない不変情報のうちの、字幕データの不変情報が取得され、その不変情報に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報に基づき、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かが判定される。そして、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可され

ていると判定された場合、字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理が行われる。

本発明のデータ記録媒体においては、少なくとも字幕データが多重化された多重化データと、多重化データに関するメタデータとを含む記録データが記録されており、メタデータは、多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、字幕データの不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含んでいる。

本発明のデータ構造においては、少なくとも字幕データが多重化された多重化データと、多重化データに関するメタデータとを含むデータのデータ構造において、メタデータは、多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、字幕データの不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含んでいる。

本発明によれば、利便性等の高いデータ処理が可能となる。特に、字幕データの表示方式の変更を制御することが可能となる。

図面の簡単な説明

第１図は、本発明を適用したディスク装置の一実施の形態のハードウェア構成例を示すブロック図、第２図Ａおよび第２図Ｂは、CPU１１２が実行するソフトウェアモジュール群の構成例を示すブロック図、第３図は、バッファ制御モジュール２１５の構成例を示すブロック図、第４図は、ディスク１０１におけるディレクトリ構成例を示す図、第５図は、"PLAYLIST.DAT"ファイルのシンタクスを示す図、第６図

は、PlayItem()のシンタクスを示す図、第７図は、PlayListMark()のシンタクスを示す図、第８図は、mark_typeの値と、Mark()のタイプとの関係を示す図、第９図は、PlayList(), PlayItem()、クリップ、およびクリップストリームファイルに格納されたプログラムストリームの関係を示す図、第１０図は、クリップ情報ファイルClip()のシンタクスを示す図、第１１図は、エレメンタリストリームを識別するstream_idおよびprivate_stream_idと、エレメンタリストリームとの関係を示す図、第１２図は、StaticInfo()のシンタクスを示す図、第１３図は、DynamicInfo()のシンタクスを示す図、第１４図は、EP_map()のシンタクスを示す図、第１５図Ａおよび第１５図Ｂは、MPEG-2 Systemのプログラムストリーム、プログラムストリームパック、およびプログラムストリームパックヘッダのシンタクスを示す図、第１６図Ａおよび第１６図Ｂは、MPEG-2 SystemのPESパケットのシンタクスを示す図、第１７図Ａ、第１７図Ｂおよび第１７図Ｃは、MPEG-2 SystemのPESパケットのシンタクスを示す図、第１８図Ａおよび第１８図Ｂは、MPEG-2 SystemのPESパケットのシンタクスを示す図、第１９図Ａおよび第１９図Ｂは、MPEG-2 SystemにおけるPES_packet()のstream_idに記述される値と、エレメンタリストリームの属性（種類）との関係を示す図、第２０図は、ディスク装置が採用するstream_idを示す図、第２１図は、private_stream1_PES_payload()のシンタクスを示す図、第２２図は、private_stream_idの値と、private_payload()に格納されるエレメンタリストリームの属性との関係を示す図、第２３図は、private_stream2_PES_payload()のシンタクスを示す図、第２４図は、au_information()のシンタクスを示す図、第２５図は、"PLAYLIST.DAT"ファイルの具体例を示す図、第２６図Ａおよび第２６図Ｂは、クリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"

の具体例を示す図、第２７図は、クリップ情報ファイル"00001.CLP"の中のEP_map()の具体例を示す図、第２８図は、PlayList #0とPlayList #1の中のPlayListMark()の具体例を示す図、第２９図は、再生前処理を説明するフローチャート、第３０図は、再生処理を説明するフローチャート、第３１図は、PlayItem乗り換え処理を説明するフローチャート、第３２図は、タイムコード表示処理を説明するフローチャート、第３３図は、ストリーム切り替え処理を説明するフローチャート、第３４図は、バッファ制御モジュール２１５の処理を説明するフローチャート、第３５図は、バッファ制御モジュール２１５の処理を説明するフローチャート、第３６図は、ビデオストリームの読み出しの処理を説明するフローチャート、第３７図は、オーディオストリームの読み出しの処理を説明するフローチャート、第３８図は、字幕ストリームの読み出しの処理を説明するフローチャート、第３９図は、再同期処理を説明するフローチャート、第４０図は、マーク処理を説明するフローチャート、第４１図は、出力属性の制御処理を説明するフローチャート、第４２図は、クリップ情報ファイル"00003.CLP"に記述されているpts_change_pointとDynamicInfo()とのセットの具体例を示す図、第４３図は、字幕表示制御処理を説明するフローチャート、第４４図は、キャプチャ制御処理とバックグラウンド／スクリーンセーバ処理を説明するフローチャート、第４５図は、private_stream2_PES_payload()の他のシンタクスを示す図、第４６図は、au_information()の他のシンタクスを示す図である。

発明を実施するための最良の形態

以下に本発明の実施の形態を説明するが、請求の範囲に記載の構成要件と、発明の実施の形態における具体例との対応関係を例示すると

、次のようになる。この記載は、請求の範囲に記載されている発明をサポートする具体例が、発明の実施の形態に記載されていることを確認するためのものである。従って、発明の実施の形態中には記載されているが、構成要件に対応するものとして、ここには記載されていない具体例があったとしても、そのことは、その具体例が、その構成要件に対応するものではないことを意味するものではない。逆に、具体例が構成要件に対応するものとしてここに記載されていたとしても、そのことは、その具体例が、その構成要件以外の構成要件には対応しないものであることを意味するものでもない。

さらに、この記載は、発明の実施の形態に記載されている具体例に対応する発明が、請求の範囲に全て記載されていることを意味するものではない。換言すれば、この記載は、発明の実施の形態に記載されている具体例に対応する発明であって、この出願の請求の範囲には記載されていない発明の存在、すなわち、将来、分割出願されたり、補正により追加される発明の存在を否定するものではない。

請求の範囲１に記載のデータ処理装置は、

データ記録媒体に記録されている記録データを処理するデータ処理装置（例えば、第１図のディスク装置）において、

前記記録データは、

少なくとも字幕データが多重化された多重化データ（例えば、第４図のファイル00001.PSに格納された、複数のエレメンタリストリームが多重化されたプログラムストリーム）と、

前記多重化データに関するメタデータ（例えば、第４図のファイル00001.CLPに格納された第１０図のClip()）と

を含み、

前記メタデータは、前記多重化データに多重化されているデータご

とに、そのデータの再生中に変化しない不変情報（例えば、第１０図や第１２図のStaticInfo()）を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報（例えば、第１２図のconfigurable_flag）を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得手段（例えば、第４３図のステップＳ３４６の処理を行う第２図Ａおよび第２図Ｂのプレイヤ制御モジュール２１２）と、

前記取得手段において取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定手段（例えば、第４３図のステップＳ３４７の処理を行う第２図Ａおよび第２図Ｂのプレイヤ制御モジュール２１２）と、

前記判定手段において、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理手段（例えば、第４３図のステップＳ３５０の処理を行う第２図Ａおよび第２図Ｂのグラフィクス処理モジュール２１９）と

を備えることを特徴とする。

請求の範囲３に記載のデータ処理方法は、

データ記録媒体に記録されている記録データを処理するデータ処理方法において、

前記記録データは、

少なくとも字幕データが多重化された多重化データ（例えば、第

４図のファイル00001.PSに格納された、複数のエレメンタリストリームが多重化されたプログラムストリーム）と、

前記多重化データに関するメタデータ（例えば、第４図のファイル00001.CLPに格納された第１０図のClip()）と

を含み、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報（例えば、第１０図や第１２図のStaticInfo()）を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報（例えば、第１２図のconfigurable_flag）を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得ステップ（例えば、第４３図のステップＳ３４６）と、

前記取得ステップにおいて取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップ（例えば、第４３図のステップＳ３４７）と、

前記判定ステップにおいて、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップ（例えば、第４３図のステップＳ３５０）と

を含むことを特徴とする。

請求の範囲４に記載のプログラム、および請求の範囲５に記載のプログラム記録媒体に記録されているプログラムは、

データ記録媒体に記録されている記録データを処理するデータ処理を、コンピュータに行わせるプログラムにおいて、

前記記録データは、

少なくとも字幕データが多重化された多重化データ（例えば、第４図のファイル00001.PSに格納された、複数のエレメンタリストリームが多重化されたプログラムストリーム）と、

前記多重化データに関するメタデータ（例えば、第４図のファイル00001.CLPに格納された第１０図のClip()）と

を含み、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報（例えば、第１０図や第１２図のStaticInfo()）を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報（例えば、第１２図のconfigurable_flag）を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得ステップ（例えば、第４３図のステップＳ３４６）と、

前記取得ステップにおいて取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップ（例えば、第４３図のステップＳ３４７）と、

前記判定ステップにおいて、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップ（例えば、第４３図のステ

ップＳ３５０）と　を含むことを特徴とする。

請求の範囲６に記載のデータ記録媒体は、

少なくとも字幕データが多重化された多重化データ（例えば、第４図のファイル00001.PSに格納された、複数のエレメンタリストリームが多重化されたプログラムストリーム）と、

前記多重化データに関するメタデータ（例えば、第４図のファイル00001.CLPに格納された第１０図のClip()）と

を含む記録データが記録されており、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報（例えば、第１０図や第１２図のStaticInfo()）を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報（例えば、第１２図のconfigurable_flag）を含む

ことを特徴とする。

以下、図面を参照して、本発明の実施の形態について説明する。

［ハードウェア構成］

第１図は、本発明を適用したディスク装置の一実施の形態のハードウェアの構成例を示すブロック図である。

第１図のディスク装置は、例えば、ディスクプレーヤや、ゲーム装置、カーナビゲーションシステムその他に適用することができる。

第１図のディスク装置において、ディスク１０１は、例えば、DVDなどの光ディスク、あるいは光磁気ディスク、磁気ディスクなどであり、ビデオデータや、オーディオデータ、字幕データなどのコンテンツデータ、さらには、コンテンツデータを再生するのに必要なデータが記録されている。

なお、ディスク１０１に記録されるデータ（記録データ）には、必要に応じて、コンピュータが実行可能なプログラムも含まれる。また、本実施の形態では、記録媒体として、ディスク状の記録媒体であるディスク１０１を採用するが、その他、記録媒体としては、例えば、半導体メモリや、テープ状の記録媒体であってもよい。さらに、第１図のディスク装置には、遠方にあるディスク１０１から読み出されて送信されてくるデータを入力することができる。即ち、ディスク１０１からのデータの読み出しは、ディスク装置に接続した別の装置で行い、その別の装置で読み出されたデータを、ディスク装置で受信して処理することができる。また、ディスク装置では、ディスク１０１に記録されたデータと同様のデータをストレージに記憶しているサーバ等から、インターネット等のネットワークを介して、データの配信を受けて処理することも可能である。さらに、ディスク装置では、サーバその他の装置からのデータを受信し、一旦、ディスク１０１に記録してから、そのディスク１０１に記録されたデータを処理することも可能である。

ディスクドライブ１０２には、ディスク１０１が着脱可能になっている。ディスクドライブ１０２は、図示せぬインターフェースを内蔵し、そのインターフェースを通じて、ドライブインターフェース１１４に接続されている。ディスクドライブ１０２は、そこに装着されたディスク１０１を駆動し、ドライブインターフェース１１４からの読み出し等の命令にしたがって、ディスク１０１からデータを読み出して、ドライブインターフェース１１４に供給する等の処理を行う。

バス１１１には、CPU(Central Processing Unit)１１２、メモリ１１３、ドライブインターフェース１１４、入力インターフェース１１５、ビデオデコーダ１１６、オーディオデコーダ１１７、ビデオ出力

インターフェース１１８、オーディオ出力インターフェース１１９が接続されている。

CPU１１２およびメモリ１１３は、コンピュータシステムを形成している。即ち、CPU１１２は、メモリ１１３に記憶されたプログラムである、後述するソフトウェアモジュール群を実行し、ディスク装置全体を制御するとともに、後述する各種の処理を行う。メモリ１１３は、CPU１１２が実行するソフトウェアモジュール群を記憶している。また、メモリ１１３は、CPU１１２の動作上必要なデータを一時記憶する。なお、メモリ１１３は、不揮発性メモリのみ、または揮発性メモリと不揮発性メモリとの組み合わせで構成することが可能である。また、第１図のディスク装置に、ハードディスクを設け、そのハードディスクに、CPU１１２が実行するソフトウェアモジュール群を記録（インストール）しておく場合には、メモリ１１３は、揮発性メモリのみで構成することが可能である。

ここで、CPU１１２が実行するプログラム（ソフトウェアモジュール群）は、ディスク装置に内蔵されている記録媒体としてのメモリ１１３に予め記録しておく（記憶させておく）ことができる。

あるいはまた、プログラムは、ディスク１０１、さらには、ディスク１０１以外のフレキシブルディスク、CD-ROM(Compact Disc Read Only Memory)，MO(Magneto Optical)ディスク、磁気ディスク、メモリカードなどのリムーバブル記録媒体に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバブル記録媒体は、いわゆるパッケージソフトウエアとして提供することができる。

なお、プログラムは、メモリ１１３にあらかじめ記憶させておくこと、あるいは、上述したようなリムーバブル記録媒体からディスク装置にインストールすることができる。また、プログラムは、ダウンロ

ードサイトから、ディジタル衛星放送用の人工衛星を介して、ディスク装置に無線で転送したり、LAN(Local Area Network)、インターネットといったネットワークを介して、ディスク装置に有線で転送し、ディスク装置では、そのようにして転送されてくるプログラムを、入力インターフェース１１５で受信し、内蔵するメモリ１１３にインストールすることができる。

さらに、プログラムは、１のCPUにより処理されるものであっても良いし、複数のCPUによって分散処理されるものであっても良い。

ドライブインターフェース１１４は、CPU１１２の制御の下、ディスクドライブ１０２を制御し、これにより、ディスクドライブ１０２がディスク１０１から読み出したデータを、バス１１１を介して、CPU１１２や、メモリ１１３、ビデオデコーダ１１６、オーディオデコーダ１１７などに供給する。

入力インターフェース１１５は、図示せぬキー（ボタン）や、リモコン（リモートコントローラ）がユーザに操作されることによって供給される信号を受信し、バス１１１を介して、CPU１１２に供給する。なお、入力インターフェース１１５は、その他、例えば、モデム（ADSL(Asymmetric Digital Subscriber Line)モデムを含む）や、NIC(Network Interface Card)などの通信インターフェースとしても機能する。

ビデオデコーダ１１６は、ディスクドライブ１０２によってディスク１０１から読み出され、ドライブインターフェース１１４およびバス１１１を介して供給される、ビデオデータの符号化データ（符号化オーディオデータ）をデコードし、その結果得られるビデオデータを、バス１１１を介して、CPU１１２やビデオ出力インターフェース１１８に供給する。

オーディオデコーダ１１７は、ディスクドライブ１０２によってディスク１０１から読み出され、ドライブインターフェース１１４およびバス１１１を介して供給される、オーディオデータの符号化データ（符号化オーディオデータ）をデコードし、その結果得られるオーディオデータを、バス１１１を介して、CPU１１２やオーディオ出力インターフェース１１９に供給する。

ビデオ出力インターフェース１１８は、バス１１１を介して供給されるビデオデータに必要な処理を施し、ビデオ出力端子１２０から出力する。オーディオ出力インターフェース１１９は、バス１１１を介して供給されるオーディオデータに必要な処理を施し、オーディオ出力端子１２１から出力する。

ビデオ出力端子１２０は、図示せぬCRT(Cathode Ray Tube)や、液晶パネル等のビデオ出力装置に接続されており、従って、ビデオ出力端子１２０から出力されるビデオデータは、ビデオ出力装置に供給されて表示される。オーディオ出力端子１２１は、図示せぬスピーカやアンプなどのオーディオ出力装置に接続されており、従って、オーディオ出力端子１２１から出力されるオーディオデータは、オーディオ出力装置に供給されて出力される。

なお、ディスク装置から、ビデオ出力装置とオーディオ出力装置へのビデオデータとオーディオデータの供給は、有線または無線のいずれによって行うことも可能である。

［ソフトウェアモジュール群の構成］

次に、第２図Ａおよび第２図Ｂは、第１図のCPU１１２が実行するソフトウェアモジュール群の構成例を示している。

CPU１１２が実行するソフトウェアモジュール群は、オペレーティングシステム（OS）２０１と、アプリケーションプログラムとしての

ビデオコンテンツ再生プログラム２１０に大別される。

「オペレーティングシステム２０１」

オペレーティングシステム２０１は、ディスク装置の電源が投入されると最初に起動し（CPU１１２がオペレーティングシステム２０１を実行し）、初期設定等の必要な処理を行い、アプリケーションプログラムであるビデオコンテンツ再生プログラム２１０を呼び出す。

オペレーティングシステム２０１は、ビデオコンテンツ再生プログラム２１０に対して、ファイルの読み出し等のインフラ（インフラストラクチャ(infrastructure)）的なサービスを提供する。即ち、オペレーティングシステム２０１は、例えば、ファイルの読み出しに関しては、ビデオコンテンツ再生プログラム２１０からのファイルの読み出しのリクエストに対して、ドライブインターフェース１１４を介してディスクドライブ１０２を操作して、ディスク１０１のデータを読み出し、ビデオコンテンツ再生プログラム２１０に渡すサービスを提供する。また、オペレーティングシステム２０１は、ファイルシステムの解釈等も行う。

なお、オペレーティングシステム２０１は、マルチタスク処理の機能を備えており、複数のソフトウェアモジュールを、時分割で（見かけ上）同時に動作させることができる。即ち、ビデオコンテンツ再生プログラム２１０は、幾つかのソフトウェアモジュールで構成されるが、各ソフトウェアモジュールは、並列で動作することができる。

「ビデオコンテンツ再生プログラム２１０」

ビデオコンテンツ再生プログラム２１０は、スクリプト制御モジュール２１１、プレイヤ制御モジュール２１２、コンテンツデータ供給モジュール２１３、デコード制御モジュール２１４、バッファ制御モジュール２１５、ビデオデコーダ制御モジュール２１６、オーディオ

デコーダ制御モジュール２１７、字幕デコーダ制御モジュール２１８、グラフィックス処理モジュール２１９、ビデオ出力モジュール２２０、およびオーディオ出力モジュール２２１で構成されている。

ビデオコンテンツ再生プログラム２１０は、ディスク１０１の再生にあたって中心的な役割を果たすソフトウェアであり、ディスク１０１がディスクドライブ１０２に装着（挿入）されると、そのディスク１０１が、コンテンツが記録された後述するフォーマットのディスクであるかを確認する。さらに、ビデオコンテンツ再生プログラム２１０は、ディスク１０１から、後述するスクリプトファイルを読み出して実行し、また、ディスク１０１から、そのディスク１０１に記録されたコンテンツを再生するのに必要なメタデータ（データベース情報）のファイルを読み出し、そのメタデータに基づいて、コンテンツの再生を制御する。

以下、第２図Ａおよび第２図Ｂのビデオコンテンツ再生プログラム２１０を構成するソフトウェアモジュールについて説明する。なお、第２図Ａおよび第２図Ｂにおいては、原則として、実線の矢印は、コンテンツのデータを表し、点線の矢印は、制御のデータを表す。

「スクリプト制御モジュール２１１」

スクリプト制御モジュール２１１は、ディスク１０１に記録されたスクリプトファイルに記述されているスクリプトプログラム（スクリプト）を解釈して実行する。スクリプトプログラムでは、例えば、「グラフィクス処理モジュール２１９を操作し、メニュー等の画像を作成して表示する」、「リモコン等のUI(User Interface)からの信号に従いメニューの表示を変更する（例えば、メニュー上のカーソルを移動する等）」、「プレイヤ制御モジュール２１２を制御する」等の動作を記述することができる。

「プレイヤ制御モジュール２１２」

プレイヤ制御モジュール２１２は、ディスク１０１に記録されているメタデータ（データベース情報）等を参照し、コンテンツの再生に関する制御を行う。即ち、プレイヤ制御モジュール２１２は、例えば、ディスク１０１に記録されている、後述するPlayList()やClip()を解析し、その解析結果にしたがって、コンテンツデータ供給モジュール２１３や、デコード制御モジュール２１４、バッファ制御モジュール２１５を制御する。また、プレイヤ制御モジュール２１２は、スクリプト制御モジュール２１１や入力インターフェース１１５からの指示にしたがい、再生対象のストリームを切り替える、後述するストリーム切り替え等の制御を行う。さらに、プレイヤ制御モジュール２１４は、デコード制御モジュール２１４から時刻を取得し、時刻表示や、後述するマーク（Mark()）の処理等を行う。

「コンテンツデータ供給モジュール２１３」

コンテンツデータ供給モジュール２１３は、プレイヤ制御モジュール２１２の制御にしたがい、あるいは、バッファ制御モジュール２１５に蓄積されたデータの量に基づき、ディスク１０１からのコンテンツのデータやメタデータ等の読み出しを、オペレーティングシステム２０１に要求する。

なお、オペレーティングシステム２０１が、コンテンツデータ供給モジュール２１３からの要求に応じてディスク１０１から読み出したメタデータ等は、必要なモジュールに供給される。また、オペレーティングシステム２０１が、コンテンツデータ供給モジュール２１３からの要求に応じてディスク１０１から読み出したコンテンツのデータは、バッファ制御モジュール２１５に供給される。

「デコード制御モジュール２１４」

デコード制御モジュール２１４は、プレイヤ制御モジュール２１２からの制御にしたがい、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、および字幕デコーダ制御モジュール２１８の動作を制御する。また、デコード制御モジュール２１４は、時刻を計時する計時部２１４Ａを内蔵し、ビデオデコーダ制御モジュール２１６の制御によって出力されるビデオデータの出力と、そのビデオデータと同期して出力されるべきデータ（出力データ）の出力、即ち、ここでは、オーディオデコーダ制御モジュール２１７の制御によって出力されるオーディオデータの出力との同期を管理する。

「バッファ制御モジュール２１５」

バッファ制御モジュール２１５は、第１図のメモリ１１３の記憶領域の一部であるバッファ２１５Ａを内蔵しており、そのバッファ２１５Ａに、コンテンツデータ供給モジュール２１３がオペレーティングシステム２０１に要求を行うことによってディスク１０１から読み出されたコンテンツのデータを一時記憶する。

また、バッファ制御モジュール２１５は、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、または字幕デコーダ制御モジュール２１８の要求にしたがって、バッファ２１５Ａに記憶されたデータを、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、または字幕デコーダ制御モジュール２１８に供給する。

即ち、バッファ制御モジュール２１５は、後述する第３図で説明するビデオ読み出し機能部２３３、オーディオ読み出し機能部２３４、および字幕読み出し機能部２３５を内蔵している。そして、バッファ制御モジュール２１５は、ビデオデコーダ制御モジュール２１６から

のデータの要求を、ビデオ読み出し機能部２３３で処理することにより、バッファ２１５Ａに記憶されたデータを、ビデオデコーダ制御モジュール２１６に供給する。同様に、バッファ制御モジュール２１５は、オーディオデコーダ制御モジュール２１７からのデータの要求を、オーディオ読み出し機能部２３４で処理することにより、バッファ２１５Ａに記憶されたデータを、オーディオデコーダ制御モジュール２１７に供給するとともに、字幕デコーダ制御モジュール２１８からのデータの要求を、字幕読み出し機能部２３５で処理することにより、バッファ２１５Ａに記憶されたデータを、字幕デコーダ制御モジュール２１８に供給する。

「ビデオデコーダ制御モジュール２１６」

ビデオデコーダ制御モジュール２１６は、バッファ制御モジュール２１５内のビデオ読み出し機能部２３３（第３図）を操作して、ビデオデータを符号化したデータ（ビデオ符号化データ）を、ビデオアクセスユニット単位で、バッファ制御モジュール２１５のバッファ２１５Ａから読み出し、第１図のビデオデコーダ１１６に供給する。また、ビデオデコーダ制御モジュール２１６は、ビデオデコーダ１１６を制御し、ビデオアクセスユニット単位のデータをデコードさせる。さらに、ビデオデコーダ制御モジュール２１６は、ビデオデコーダ１１６でのデコードの結果得られるビデオデータを、グラフィクス処理モジュール２１９に供給する。

ここで、ビデオアクセスユニットとは、例えば、ビデオデータの１ピクチャ（１フレームまたは１フィールド）分である。

「オーディオデコーダ制御モジュール２１７」

オーディオデコーダ制御モジュール２１７は、バッファ制御モジュール２１５内のオーディオ読み出し機能部２３４（第３図）を操作し

て、オーディオデータを符号化したデータ（オーディオ符号化データ）を、オーディオアクセスユニット単位で、バッファ制御モジュール２１５のバッファ２１５Ａから読み出し、第１図のオーディオデコーダ１１７に供給する。また、オーディオデコーダ制御モジュール２１７は、オーディオデコーダ１１７を制御し、オーディオアクセスユニット単位のデータをデコードさせる。さらに、オーディオデコーダ制御モジュール２１７は、オーディオデコーダ１１７でのデコードの結果得られるオーディオデータを、オーディオ出力モジュール２２１に供給する。

ここで、オーディオアクセスユニットとは、オーディオデータの所定のデータ量分（例えば、１ピクチャに同期して出力される分）である。本実施の形態では、オーディオアクセスユニットは、例えば、既知の固定長であるとする。

「字幕デコーダ制御モジュール２１８」

字幕デコーダ制御モジュール２１８は、バッファ制御モジュール２１５内の字幕読み出し機能部２３５（第３図）を操作して、字幕データを符号化したデータ（字幕符号化データ）を、字幕アクセスユニット単位で、バッファ制御モジュール２１５のバッファ２１５Ａから読み出す。また、字幕デコーダ制御モジュール２１８は、内部に、図示せぬ字幕デコードソフトウェアを備えており、バッファ２１５Ａから読み出したデータをデコードする。さらに、字幕デコーダ制御モジュール２１８は、そのデコードの結果得られる字幕データ（字幕の画像データ）を、グラフィクス処理モジュール２１９に供給する。

ここで、字幕アクセスユニットとは、字幕データの所定のデータ量分（例えば、１ピクチャに同期して出力される分）である。本実施の形態では、字幕アクセスユニットのサイズは、例えば、その字幕アク

セスユニットの先頭に記述されていることとする。

「グラフィクス処理モジュール２１９」

グラフィクス処理モジュール２１９は、プレイヤ制御モジュール２１２の制御（指示）にしたがい、字幕デコーダ制御モジュール２１８からの字幕データの拡大や縮小を行い、ビデオデコーダ制御モジュール２１６からのビデオデータと加算（オーバーレイ）する。さらに、グラフィクス処理モジュール２１９は、字幕データとの加算後のビデオデータのサイズ（画枠）を、第1図のビデオ出力端子１２０に接続されたビデオ出力装置の表示画面にあわせるための拡大または縮小等を行い、その結果得られるビデオデータを、ビデオ出力モジュール２２０に出力する。

また、グラフィクス処理モジュール２１９は、スクリプト制御モジュール２１１やプレイヤ制御モジュール２１２の指示（制御）に従い、メニューやメッセージ等を生成し、出力ビデオデータにオーバーレイする。

さらに、グラフィクス処理モジュール２１９は、第１図のビデオ出力端子１２０に接続されたビデオ出力装置のアスペクト比と、ディスク１０１に記録されたビデオデータのアスペクト比を指示する情報等とに基づいて、ビデオ出力モジュール２２０に出力するビデオデータのアスペクト比の変換を行う。

即ち、例えば、ビデオ出力装置のアスペクト比が16:9である場合において、ビデオデータのアスペクト比を指示する情報が4:3のアスペクト比を表しているときには、グラフィクス処理モジュール２１９は、ビデオ出力モジュール２２０に出力するビデオデータを、横方向（水平方向）にスクイーズ（縮小）処理し、左右に黒味を入れて出力する。また、例えば、ビデオ出力装置のアスペクト比が4:3である場合

において、ビデオデータのアスペクト比を指示する情報が16:9のアスペクト比を表しているときには、グラフィクス処理モジュール２１９は、ビデオ出力モジュール２２０に出力するビデオデータを、縦方向（垂直方向）にスクイーズ（縮小）処理し、上下に黒味を入れて出力する。

なお、ビデオ出力装置のアスペクト比と、ビデオデータのアスペクト比を指示する情報が表すアスペクト比とが、いずれも、4:3や16:9で、同一である場合、グラフィクス処理モジュール２１９は、ビデオ出力モジュール２２０に出力するビデオデータを、スクイーズ処理することなく、そのまま出力する。

その他、グラフィクス処理モジュール２１９は、例えば、プレイヤ制御モジュール２１２からの要求に応じて、現在処理中のビデオデータをキャプチャする。さらに、グラフィクス処理モジュール２１９は、そのキャプチャしたビデオデータを記憶し、あるいは、プレイヤ制御モジュール２１２に供給する。

「ビデオ出力モジュール２２０」

ビデオ出力モジュール２２０は、第１図のメモリ１１３の一部を排他的に占有してFIFO(First In First Out)２２０Ａ（バッファ）として使用し、グラフィクス処理モジュール２１９からのビデオデータを一時的に記憶し、また、そのFIFO２２０Ａに記憶されたビデオデータを適宜読み出して、ビデオ出力端子１２０（第１図）に出力する。

「オーディオ出力モジュール２２１」

オーディオ出力モジュール２２１は、第１図のメモリ１１３の一部を排他的に占有してFIFO２２１Ａ（バッファ）として使用し、オーディオデコーダ制御モジュール２１７（オーディオデコーダ１１７）からのオーディオデータを一時的に記憶し、また、そのFIFO２２１Ａに

記憶されたオーディオデータを適宜読み出して、オーディオ出力端子１２１（第１図）に出力する。

さらに、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータが、左チャネルが「主音声」のオーディオデータで、右チャネルの「副音声」のオーディオデータであるデュアル(Dual)（二ヶ国語）モードのオーディオデータである場合、あらかじめ指定された音声出力モードに従って、オーディオデコーダ制御モジュール２１７からのオーディオデータを、オーディオ出力端子１２１に出力する。

即ち、音声出力モードとして、例えば、「主音声」が指定されているときには、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータのうちの左チャネルのオーディオデータを、右チャネルのオーディオデータとしてコピーし、その左チャネルと右チャネルのオーディオデータ（「主音声」のオーディオデータ）を、オーディオ出力端子１２１に出力する。また、音声出力モードとして、「副音声」が指定されているときには、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータのうちの右チャネルのオーディオデータを、左チャネルのオーディオデータとしてコピーし、その左チャネルと右チャネルのオーディオデータ（「副音声」のオーディオデータ）を、オーディオ出力端子１２１に出力する。さらに、音声出力モードとして、「主・副」が指定されているときには、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータを、そのまま、オーディオ出力端子１２１に出力する。

なお、オーディオデコーダ制御モジュール２１７からのオーディオ

データが、ステレオ(Stereo)モードのオーディオデータである場合、オーディオ出力モジュール２２１は、音声出力モードの指定にかかわらず、オーディオデコーダ制御モジュール２１７からのオーディオデータを、そのまま、オーディオ出力端子１２１に出力する。

ここで、音声出力モードの指定は、例えば、ビデオコンテンツ再生プログラム２１０が生成するメニューが表示された画面等において、ユーザがリモコン等を操作することにより対話的に行うことができる。

［バッファ制御モジュール２１５の構成］

次に、第３図は、第２図Ａおよび第２図Ｂのバッファ制御モジュール２１５の構成例を示している。

バッファ制御モジュール２１５は、第１図のメモリ１１３の一部を、バッファ２１５Ａとして排他的に使用し、そのバッファ２１５Ａに、ディスク１０１から読み出されたデータを一時記憶させる。また、バッファ制御モジュール２１５は、バッファ２１５Ａに記憶されたデータを読み出して、第２図Ａおよび第２図Ｂのビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、または字幕デコーダ制御モジュール２１８に供給する。

即ち、バッファ制御モジュール２１５は、バッファ２１５Ａの他、メモリ１１３の一部であるデータ先頭ポインタ記憶部２３１、およびデータ書き込みポインタ記憶部２３２を有するとともに、内部モジュールとして、ビデオ読み出し機能部２３３、オーディオ読み出し機能部２３４、字幕読み出し機能部２３５を有する。

バッファ２１５Ａは、例えば、リングバッファであり、ディスク１０１から読み出されたデータを順次記憶し、その記憶容量分のデータを記憶した後は、最も古いデータに上書きする形で、最新のデータを

、いわば無限ループ状に記憶していく。

データ先頭ポインタ記憶部２３１は、バッファ２１５Ａに記憶されたデータのうち、まだ、バッファ２１５Ａから読み出されていない最も古いデータが記憶されている位置（アドレス）を指すデータ先頭ポインタを記憶する。

データ書き込みポインタ記憶部２３２は、ディスク１０１から読み出された最新のデータが書き込まれるバッファ２１５Ａの位置（アドレス）を指す書き込みポインタを記憶する。

ここで、データ書き込みポインタが指す位置は、バッファ２１５Ａに、ディスク１０１から読み出されたデータが記憶されるごとに、図中、右回り（時計回り）に更新されていき、データ先頭ポインタが指す位置は、バッファ２１５Ａからのデータの読み出しに応じて、図中、右回りに更新されていく。したがって、バッファ２１５Ａに記憶されたデータのうち、いわば有効なデータは、データ先頭ポインタが指す位置から、右回りに、データ書き込みポインタが指す位置までに記憶されているデータである。

ビデオ読み出し機能部２３３は、第２図Ａおよび第２図Ｂのビデオデコーダ制御モジュール２１６からの要求に応じて、バッファ２１５Ａからビデオストリーム（ビデオデータに関するエレメンタリストリーム）を読み出し、ビデオデコーダ制御モジュール２１６に供給する。オーディオ読み出し機能部２３４も、第２図Ａおよび第２図Ｂのオーディオデコーダ制御モジュール２１７からの要求に応じて、バッファ２１５Ａからオーディオストリーム（オーディオデータに関するエレメンタリストリーム）を読み出し、オーディオデコーダ制御モジュール２１７に供給する。字幕読み出し機能部２３５も、第２図Ａおよび第２図Ｂの字幕デコーダ制御モジュール２１８からの要求に応じて

、バッファ２１５Ａから字幕ストリーム（字幕データに関するエレメンタリストリーム）を読み出し、字幕デコーダ制御モジュール２１８に供給する。

即ち、光ディスク１０１には、例えば、MPEG(Moving Picture Experts Group)2の規格に準拠したプログラムストリーム（MPEG2-System Program Stream）が記録されており、バッファ２１５Ａには、光ディスク１０１から読み出されたプログラムストリームが記憶される。このプログラムストリームは、ビデオストリームや、オーディオストリーム、字幕ストリーム等の１以上のエレメンタリストリームが時分割多重されている。ビデオ読み出し機能部２３３は、プログラムストリームのデマルチプレクスの機能を有し、バッファ２１５Ａに記憶されたプログラムストリームから、ビデオストリームを分離して読み出す。

同様に、オーディオ読み出し機能部２３４も、プログラムストリームのデマルチプレクスの機能を有し、バッファ２１５Ａに記憶されたプログラムストリームから、オーディオストリームを分離して読み出す。字幕読み出し機能部２３５も、プログラムストリームのデマルチプレクスの機能を有し、バッファ２１５Ａに記憶されたプログラムストリームから、字幕ストリームを分離して読み出す。

ここで、ビデオ読み出し機能部２３３は、第１図のメモリ１１３の一部であるビデオ読み出しポインタ記憶部２４１、stream_idレジスタ２４２、およびau_information()レジスタ２４３を有している。

ビデオ読み出しポインタ記憶部２４１は、バッファ２１５Ａの、ビデオストリームが記憶された位置（アドレス）を指すビデオ読み出しポインタを記憶し、ビデオ読み出し機能部２３３は、バッファ２１５Ａの、ビデオ読み出しポインタが指す位置に記憶されているデータを

、ビデオストリームとして読み出す。stream_idレジスタ２４２は、バッファ２１５Ａに記憶されたプログラムストリームを解析し、そのプログラムストリームの中から読み出すビデオストリームを識別（特定）するための後述するstream_idを記憶する。au_information()レジスタ２４３は、バッファ２１５Ａからビデオストリームを読み出すために必要な（ビデオストリームの読み出しに利用される）データである後述するau_information()を記憶する。

オーディオ読み出し機能部２３４は、第１図のメモリ１１３の一部であるオーディオ読み出しポインタ記憶部２５１、stream_idレジスタ２５２、およびprivate_stream_idレジスタ２５３を有している。

オーディオ読み出しポインタ記憶部２５１は、バッファ２１５Ａの、オーディオストリームが記憶された位置（アドレス）を指すオーディオ読み出しポインタを記憶し、オーディオ読み出し機能部２３４は、バッファ２１５Ａの、オーディオ読み出しポインタが指す位置に記憶されているデータを、オーディオストリームとして読み出す。stream_idレジスタ２５２とprivate_stream_idレジスタ２５３は、バッファ２１５Ａに記憶されたプログラムストリームを解析し、そのプログラムストリームの中から読み出すオーディオストリームを識別するための後述するstream_idとprivate_stream_idを、それぞれ記憶する。

字幕読み出し機能部２３５は、第１図のメモリ１１３の一部である字幕読み出し機能フラグ記憶部２６１、字幕読み出しポインタ記憶部２６２、stream_idレジスタ２６３、およびprivate_stream_idレジスタ２６４を有している。

字幕読み出し機能フラグ記憶部２６１は、字幕読み出し機能フラグを記憶する。字幕読み出し機能フラグ記憶部２６１に記憶された字幕読み出し機能フラグが、例えば０である場合、字幕読み出し機能部２

３５は機能動作せず、字幕読み出し機能フラグ記憶部２６１に記憶された字幕読み出し機能フラグが、例えば１である場合、字幕読み出し機能部２３５は機能する。

字幕読み出しポインタ記憶部２６２は、バッファ２１５Ａの、字幕ストリームが記憶された位置（アドレス）を指す字幕読み出しポインタを記憶し、字幕読み出し機能部２３５は、バッファ２１５Ａの、字幕読み出しポインタが指す位置に記憶されているデータを、字幕ストリームとして読み出す。stream_idレジスタ２６３とprivate_stream_idレジスタ２６４は、バッファ２１５Ａに記憶されたプログラムストリームを解析し、そのプログラムストリームの中から読み出す字幕ストリームを識別するための後述するstream_idとprivate_stream_idを、それぞれ記憶する。

［ディスク１０１に記録されたデータのデータフォーマットの説明］

次に、ディスク１０１に記録されたデータのデータフォーマットについて説明する。

第４図は、ディスク１０１のディレクトリ構造を模式的に示している。

ディスク１０１のファイルシステムとしては、例えば、ISO(International Organization for Standardization)-9660や、UDF(Universal Disk Format:http://www.osta.org/specs/)などで規定されたファイルシステムが用いられており、ディスク１０１に記録されたデータのファイルはディレクトリ構造により階層的に管理されている。ここで、ファイルシステムは、上述したファイルシステムに限定されるものではない。

第４図では、ファイルシステムの基点を示すルート(root)ディレク

トリに、"VIDEO"ディレクトリが置かれ、"VIDEO"ディレクトリには、"CLIP"ディレクトリと、"STREAM"ディレクトリとの２つのディレクトリが置かれている。

"VIDEO"ディレクトリには、"CLIP"ディレクトリと"STREAM"ディレクトリとの２つのディレクトリの他に、"SCRIPT.DAT"ファイルと、"PLAYLIST.DAT"ファイルの２つのデータファイルが置かれている。

"SCRIPT.DAT"ファイルは、スクリプトプログラムが記述されたスクリプトファイルである。即ち、"SCRIPT.DAT"ファイルには、ディスク１０１の再生形態をインタラクティブなものとするために使用するスクリプトプログラムが記述されている。この"SCRIPT.DAT"ファイルに記述されたスクリプトプログラムは、第２図Ａおよび第２図Ｂのスクリプト制御モジュール２１１によって解釈、実行される。

"PLAYLIST.DAT"ファイルには、ディスク１０１に記録されたビデオデータ等のコンテンツの再生手順が記述されたプレイリスト（後述する第５図のPlayList()）が１以上格納されている。

"CLIP"ディレクトリには、１以上のクリップ情報ファイルが置かれ、"STREAM"ディレクトリには、１以上のクリップストリームファイルが置かれる。即ち、第４図では、"CLIP"ディレクトリには、３つのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"が置かれており、"STREAM"ディレクトリには、３つのクリップストリームファイル"00001.PS", "00002.PS", "00003.PS"が置かれている。

クリップストリームファイルには、ビデオデータ、オーディオデータ、字幕データなどの１以上のデータ（ストリーム）を圧縮、符号化して得られる１以上のエレメンタリストリームを時分割多重化したプログラムストリームが格納されている。

クリップ情報ファイルには、対応するクリップストリームファイル

の性質等の、クリップストリームに関する（ファイル）メタデータが記述されている。

即ち、クリップストリームファイルとクリップ情報ファイルとは、１対１に対応している。第４図では、クリップストリームファイルには、5文字の数字＋ピリオド＋"PS"という命名規則にしたがって、ファイル名が付されており、クリップ情報ファイルには、対応するクリップストリームファイルと同一の5文字の数字＋ピリオド＋"CLP"という命名規則にしたがって、ファイル名が付されている。

従って、ファイルが、クリップストリームファイルまたはクリップ情報ファイルのうちのいずれであるかは、ファイル名の拡張子（ピリオドより右側の部分）によって識別することができ、さらに、対応するクリップストリームファイルとクリップ情報ファイルとは、ファイル名の拡張子以外の部分（ピリオドより左側の部分）が一致するかどうかによって識別することができる。

以下、ディスク１０１に記録された各ファイルの詳細について説明する。

「PLAYLIST.DAT」

第５図は、第４図の"VIDEO"ディレクトリ下の"PLAYLIST.DAT"ファイルの内部構造（シンタクス(syntax)）を示している。

ここで、第５図において、"Syntax"の欄の記載がデータ構造を表し、"No. of bits"の欄の記載は、対応する行の"Syntax"の欄のデータのビット長を表す。さらに、"Mnemonic"の欄の記載のうちの"bslbf"(bit string left bit first)は、対応する行の"Syntax"の欄のデータが左のビットから送り出されることを意味し、"uimsbf"(unsigned integer most significant bit first)は、対応する行の"Syntax"の欄のデータが、符号なし整数値であり、最上位ビットから送り出される

ことを意味する。以下説明する、第５図と同様の図についても、同様である。

"PLAYLIST.DAT"ファイルにおいては、その先頭から、その名称（ファイル名）等の情報を記述するためのname_length（8ビット）とname_string（255バイト）が順次配置される。

即ち、name_lengthは、その後に配置されるname_stringのサイズを、バイト数で表す。name_stringは、"PLAYLIST.DAT"ファイルの名称（ファイル名）を表す。

なお、name_stringについては、その先頭から、name_lengthで表されるバイト数までが有効な名称として使用される。たとえばname_lengthが値10である場合には、name_stringの先頭から10バイト分が有効な名称として解釈される。

name_stringの後には、number_of_PlayLists（16ビット）が配置される。number_of_PlayListsは、続くPlayList()の個数を表す。number_of_PlayListsの後に、そのnumber_of_PlayListsの数だけのPlayList()が配置される。

PlayList()は、ディスク１０１に記録されたクリップストリームファイルの再生手順が記述されたプレイリストであり、以下のような内部構造を有する。

即ち、PlayList()の先頭には、PlayList_data_length（32ビット）が配置される。PlayList_data_lengthは、そのPlayList()のサイズを表す。

PlayList_data_lengthの後には、reserved_for_word_alignment（15ビット）とcapture_enable_flag_PlayList（1ビット）が順次配置される。15ビットのreserved_for_word_alignmentは、その後に配置される１ビットのcapture_enable_flag_PlayListの位置で、いわゆるワ

ードアライン(word alignment)をとるため（１６ビットの位置に揃えるため）に配置される。capture_enable_flag_PlayListは、PlayList()によって再生されるビデオストリームに対応するビデオデータ（PlayList()に属するビデオデータ）の、光ディスク１０１が再生される第１図のディスク装置内での２次利用を許可するか否かを表す１ビットのフラグである。capture_enable_flag_PlayListが、０または１のうちの、例えば１である場合、PlayList()に属するビデオデータの２次利用が許可されていることを表し、capture_enable_flag_PlayListが、０または１のうちの、例えば０である場合、PlayList()に属するビデオデータの２次利用が許可されていない（禁止されている）ことを表す。

なお、第５図では、capture_enable_flag_PlayListを１ビットとしたが、その他、capture_enable_flag_PlayListは、複数ビットで構成し、PlayList()に属するビデオデータの２次利用を、いわば段階的に許可するようにすることが可能である。即ち、capture_enable_flag_PlayListは、例えば、２ビットで構成することができる。そして、capture_enable_flag_PlayListの値が00B（Bは、その前の数字が２進数であることを表す）である場合には、ビデオデータの２次利用を禁止し、capture_enable_flag_PlayListの値が01Bである場合には、ビデオデータを、64×64ピクセル以下のサイズに縮小して利用する２次利用のみを許可することができる。また、capture_enable_flag_PlayListの値が10Bである場合には、サイズの制限なしで、ビデオデータの２次利用を許可することができる。

さらに、上述のように、ビデオデータの２次利用にあたって、サイズに制限を設けるのではなく、用途に制限を設けるようにすることも可能である。即ち、capture_enable_flag_PlayListの値が01Bである

場合には、ビデオコンテンツ再生アプリケーション２１０（第２図Ａおよび第２図Ｂ）のみでの２次利用を許可し、capture_enable_flag_PlayListの値が10Bである場合には、第１図のディスク装置内の、ビデオコンテンツ再生アプリケーション２１０を含む任意のアプリケーションによる２次利用を許可することができる。ここで、第１図のディスク装置内のビデオコンテンツ再生アプリケーション２１０以外のアプリケーションとしては、例えば、壁紙(バックグラウンド)やスクリーンセーバーの表示の処理を行うアプリケーションなどがある。

なお、capture_enable_flag_PlayListを、上述のように、２ビットとした場合、その前に配置されるreserved_for_word_alignmentは、ワードアラインをとるために、１４ビットとなる。

また、capture_enable_flag_PlayListにより、ビデオデータのディスク装置内での２次利用を許可する他、ディスク装置外での２次利用を許可するようにすることも可能である。ここで、ビデオデータの、ディスク装置外での２次利用を許可する場合には、ビデオデータは、例えば、ディスク装置に着脱可能な記録媒体やディスク装置に接続可能な他の装置に着脱可能な記録媒体に記録され、あるいはインターネット等のネットワークを介して、他の装置に送信（配信）される。この場合、ビデオデータには、そのビデオデータを記録媒体に記録する回数や配信する回数を制限する情報を付加するようにすることができる。

capture_enable_flag_PlayListに続いては、PlayList_name_length（8ビット）とPlayList_name_string（255バイト）とが順次配置される。PlayList_name_lengthは、その後に配置されるPlayList_name_stringのサイズを、バイト数で表し、PlayList_name_stringは、PlayList()の名称を表す。

PlayList_name_stringの後には、number_of_PlayItems（16ビット）が配置される。number_of_PlayItemsは、続くPlayItem()の個数を表す。

number_of_PlayItemsの後には、そのnumber_of_PlayItemsの数だけのPlayItem()の構造が記述される。

ここで、１つのPlayList()では、PlayItem()単位で、コンテンツの再生手順を記述することができる。

また、PlayList()の中の、number_of_PlayItemsの数だけのPlayItem()それぞれに対しては、そのPlayList()の中でユニークなID(Identification)が付される。即ち、PlayList()中の最初のPlayItem()には、IDとして0番が付され、以下、続くPlayItem()に対して、その出現順に、1番、2番、・・・と通し番号のIDが付される。

number_of_PlayItemsの数だけのPlayItem()の後には、１つのPlayListMark()が配置される。PlayListMark()は、PlayList()にしたがって行われる再生の時間軸上の印となる後述するMark()の集合で、その詳細については、第７図を参照して後述する。

「PlayItem()の説明」

次に、第６図は、第５図のPlayList()に含まれるPlayItem()の内部構造を示している。

PlayItem()の先頭には、length（16ビット）が配置され、lengthは、それを含むPlayItem()のサイズを表す。

lengthに続いては、Clip_Information_file_name_length（16ビット）とClip_Information_file_name（可変長）が順次配置される。Clip_Information_file_name_lengthは、その後に配置されるClip_Information_file_nameのサイズを、バイト数で表す。Clip_Information_file_nameは、PlayItem()によって再生するクリップストリームファ

イル（第４図の拡張子がPSのファイル）に対応するクリップ情報ファイル（第４図の拡張子がCLPのファイル）のファイル名を表す。なお、クリップストリームファイルおよびクリップ情報ファイルのファイル名の、上述した命名規則により、Clip_Information_file_nameから、PlayItem()によって再生するクリップ情報ファイルのファイル名を認識し、そのクリップストリームファイルを特定することができる。

Clip_Information_file_nameに続いては、IN_time（32ビット）とOUT_time（32ビット）が順次配置される。

IN_timeとOUT_timeは、それぞれ、Clip_Information_file_nameから特定されるクリップストリームファイルの再生開始位置と再生終了位置を指定する時刻情報である。

IN_timeによれば、クリップストリームファイルの（先頭を含む）途中の位置を再生開始位置として指定することができ、OUT_timeによれば、クリップストリームファイルの（最後を含む）途中の位置を再生終了位置として指定することができる。

ここで、PlayItem()によれば、Clip_Information_file_nameから特定されるクリップストリームファイルの、IN_timeからOUT_timeまでの間のコンテンツが再生される。このPlayItem()によって再生されるコンテンツを、以下、適宜、クリップという。

「PlayListMark()の説明」

次に、第７図は、第５図のPlayList()に含まれるPlayListMark()の内部構造を示している。

PlayListMark()は、上述したように、そのPlayListMark()を含むPlayList()（第５図）にしたがって行われる再生の時間軸上の印となる、０以上のMark()の集合である。１つのMark()は、PlayList()にしたがって行われる再生の時間軸上の１つの時刻（位置）を表す時刻情報

、Mark()のタイプを表すタイプ情報、およびタイプ情報がイベントを発生させるタイプを表しているときの、そのイベントの引数となる引数情報を、少なくとも有する。

即ち、PlayListMark()の先頭には、length（32ビット）が配置される。lengthは、それを含むPlayListMark()のサイズを表す。

lengthの後には、number_of_PlayList_marks（16ビット）が配置され、number_of_PlayList_marksは、それに続いて配置されるMark()の個数を表す。number_of_PlayList_marksの後には、そのnumber_of_PlayList_marksの数だけMark()の構造が記述される。

Mark()の先頭には、mark_type（8ビット）が配置される。mark_typeは、上述のタイプ情報であり、それを含むMark()のタイプを表す。

本実施の形態では、Mark()のタイプとして、例えば、チャプタ(Chapter)、インデクス(Index)、イベント(Event)の３種類が用意されている。

タイプがチャプタのMark()（以下、適宜、チャプタマークという）は、PlayList()を分割する頭出しの単位であるチャプタの先頭位置の印である。また、タイプがインデクスのMark()（以下、適宜、インデクスマークという）は、チャプタを細分化した単位であるインデクスの先頭位置の印である。タイプがイベントのMark()（以下、適宜、イベントマークという）は、PlayList()にしたがったコンテンツの再生中においてイベントを発生させる位置の印である。イベントマークによるイベントの発生は、後述するように、スクリプト制御モジュール２１１に通知される。

ここで、mark_typeの値と、Mark()のタイプとの関係を、第８図に示す。第８図によれば、チャプタマークのmark_typeには、１がセットされる。また、インデクスマークのmark_typeには、２がセットさ

れ、イベントマークのmark_typeには、３がセットされる。なお、第８図では、mark_typeで表される８ビットの値のうちの、０と、４乃至２５５は、将来の拡張のための予約(reserved)とされている。

第７図に戻り、mark_typeの後には、mark_name_length（8ビット）が配置される。また、Mark()の最後には、mark_name_string（24バイト）が配置される。mark_name_lengthとmark_name_stringは、Mark()の名称を記述するためのものであり、mark_name_lengthは、mark_name_stringの有効なサイズを、mark_name_stringは、Mark()の名称を、それぞれ表す。従って、mark_name_stringの先頭からmark_name_lengthが表すバイト数までが、Mark()の有効な名称を表す。

mark_name_lengthに続いては、PlayList()上で定義されるMark()をクリップストリームファイルと対応付ける４つの要素ref_to_PlayItem_id（16ビット）、mark_time_stamp（32ビット）、entry_ES_stream_id（8ビット）、entry_ES_private_stream_id（8ビット）が順次配置される。

ref_to_PlayItem_idには、Mark()が属するPlayItem()に対して通し番号で付されたIDが記述される。ref_to_PlayItem_idによって、Mark()が属するPlayItem()（第６図）が特定され、ひいては、第６図で説明したように、クリップ情報ファイルとクリップストリームファイルが特定される。

mark_time_stampは、ref_to_PlayItem_idによって特定されるクリップストリームファイル内でのMark()が表す位置（時刻）を表す。

ここで、第９図は、PlayList()，PlayItem()、クリップ、およびクリップストリームファイルに格納されたプログラムストリームの関係を示している。

第９図では、PlayList()は、３つのPlayItem()から構成されており

、その３つのPlayItem()それぞれには、通し番号で付されるID#0, #1, #2が付されている。ここで、以下、適宜、ID#iが付されたPlayItem()を、PlayItem#iと記述する。

また、第９図では、PlayItem#0, PlayItem#1, PlayItem#2によって再生されるコンテンツであるクリップが、それぞれ、クリップＡ、クリップＢ、クリップＣとして示されている。

クリップの実体は、第６図のPlayItem()におけるClip_Information_file_nameから特定される（クリップ情報ファイルから、さらに特定される）クリップストリームファイルに格納されたプログラムストリームのうちの、IN_timeからOUT_timeまでのプログラムストリームである。第９図では、クリップＡ、クリップＢ、クリップＣの実体としてのプログラムストリームが、プログラムストリームＡ、プログラムストリームＢ、プログラムストリームＣとして、それぞれ示されている。

例えば、第９図において、PlayList()にしたがって行われる再生の時間軸上の位置（時刻）t0の印となるMark()においては、そのref_to_PlayItem_idとmark_time_stampは、次のように記述される。

即ち、時刻t0は、PlayItem#1の再生が行われる時刻であるため、ref_to_PlayItem_idには、そのPlayItem#1のIDである１が記述される。さらに、時刻t0では、PlayItem#1によって、クリップＢの実体であるプログラムストリームＢが再生されるため、mark_time_stampには、プログラムストリームＢが格納されたクリップストリームファイルにおける時刻t0に相当する時刻が記述される。

再び、第７図に戻り、entry_ES_stream_idと、entry_ES_private_stream_idは、Mark()を、特定のエレメンタリストリームに関連付ける場合に、そのエレメンタリストリームを特定するために使用される。

即ち、entry_ES_stream_idには、Mark()を関連付けるエレメンタリストリーム（が配置される、後述する第１６図Ａおよび第１６図Ｂ乃至第１８図Ａおよび第１８図Ｂに示すPES_packet()）の後述するstream_idが記述される。また、entry_ES_private_stream_idには、必要に応じて、Mark()を関連付けるエレメンタリストリーム（が配置される、後述する第２１図に示すprivate_stream1_PES_payload()におけるprivate_header()）の後述するprivate_stream_idが記述される。

例えば、ビデオストリーム＃１とビデオストリーム＃２が多重化されているクリップにおいて、ビデオストリーム＃１を再生している場合と、ビデオストリーム＃２を再生している場合でチャプタの発生時刻を変更したいときには、ビデオストリーム＃１再生時のチャプタマーク発生時刻のMark()のentry_ES_stream_idとentry_ES_private_stream_idに、ビデオストリーム＃１のstream_idとprivate_stream_idが記述され、また、ビデオストリーム＃２再生時のチャプタマーク発生時刻のMark（）のentry_ES_stream_idとentry_ES_private_stream_idに、ビデオストリーム＃２のstream_idとprivate_stream_idが記述される。

なお、特定のエレメンタリストリームに関連付けないMark()のentry_ES_stream_idと、entry_ES_private_stream_idには、例えば、いずれも０が記述される。

entry_ES_private_stream_idの後には、mark_data（３２ビット）が配置される。mark_dataは、Mark()がイベントマークである場合に、そのイベントマークによって発生されるイベントの引数となる引数情報である。なお、mark_dataは、Mark()がチャプタマークやインデクスマークである場合に、そのチャプタマークやインデクスマークが表すチャプタやインデクスの番号として使用することも可能である。

「Clip()の説明」

次に、第４図の"CLIP"ディレクトリに置かれる、拡張子がCLPのクリップ情報ファイルの内部構造について説明する。

第４図では、"CLIP"ディレクトリに、３つのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"が置かれており、それぞれには、"STREAM"ディレクトリに置かれたクリップストリームファイル"00001.PS", "00002.PS", "00003.PS"の性質等を示すメタデータが格納されている。

第１０図は、そのようなクリップ情報ファイルClip()の内部構造を示している。

クリップ情報ファイルClip()の先頭には、presentation_start_timeとpresentation_end_time（いずれも３２ビット）が、順次配置される。presentation_start_timeとpresentation_end_timeは、クリップ情報ファイルClip()に対応するクリップストリームファイル（に格納されているプログラムストリーム）の先頭と最後の時刻を表す。なお、クリップストリームファイルの時刻は、MPEG2-Systemの時刻で使われている90kHzの倍数で記述される。

presentation_end_timeに続いては、reserved_for_word_alignment（7ビット）とcapture_enable_flag_Clip（1ビット）が順次配置される。7ビットのreserved_for_word_alignmentは、ワードアラインをとるためのもので、capture_enable_flag_Clipは、上述した第５図のcapture_enable_flag_PlayListと同様に、ビデオデータの２次利用を許可するか否かを表すフラグである。

但し、第５図のcapture_enable_flag_PlayListは、PlayList()によって再生されるビデオストリームに対応するビデオデータ（PlayList()に属するビデオデータ）の２次利用を許可するか否かを表すのに対

して、第１０図のcapture_enable_flag_Clipは、クリップ情報ファイルClip()に対応するクリップストリームファイルに格納されているビデオストリーム（ビデオのエレメンタリストリーム）に対応するビデオデータの２次利用を許可するか否かを表す。従って、第５図のcapture_enable_flag_PlayListと、第１０図のcapture_enable_flag_Clipとでは、２次利用を許可するビデオデータの単位（範囲）が異なる。

なお、第１０図のcapture_enable_flag_Clipも、第５図のcapture_enable_flag_PlayListで説明したように、１ビットではなく、複数ビットとすることが可能である。

capture_enable_flag_Clipの後には、number_of_streams（8ビット）が配置され、このnumber_of_streamsには、それに続くStreamInfo()構造の個数が記述される。従って、number_of_streamsに続いては、そのnumber_of_Streamsの数だけ、StreamInfo()の構造が記述される。

StreamInfo()の先頭には、length（16ビット）が配置され、このlengthは、それを含むStreamInfo()のサイズを表す。lengthに続いては、stream_id（8ビット）とprivate_stream_id（8ビット）が配置されており、このstream_idとprivate_stream_idによって、StreamInfo()に関連付けるエレメンタリストリームが特定（識別）される。

ここで、第１１図は、エレメンタリストリームを識別するstream_idおよびprivate_stream_idと、エレメンタリストリームとの関係を示している。

stream_idは、MPEG2-System規格において規定されているのと同一のものであり、その値は、MPEG2-System規格において、エレメンタリストリーム（データ）の属性（種類）ごとに、あらかじめ決められている。従って、MPEG2-System規格で定義されている属性のエレメンタ

リストリームは、stream_idだけで特定することができる。

本実施の形態では、MPEG2-System規格において規定されていない属性のエレメンタリストリームも扱うことが可能になっており、private_stream_idは、MPEG2-System規格において規定されていない属性のエレメンタリストリームを識別するための情報である。

第１１図では、MPEGで規定されている符号化（復号）方式でエンコードされたビデオのエレメンタリストリーム、ATRAC(Adaptive TRansform Acoustic Coding)方式でエンコードされたオーディオのエレメンタリストリーム（以下、適宜、ATRACオーディオストリームという）、LPCM(Linear Pulse Code Modulation)方式でエンコードされたオーディオのエレメンタリストリーム（以下、適宜、LPCMオーディオストリームという）、字幕のエレメンタリストリーム（以下、適宜、字幕ストリームという）の４つの属性のエレメンタリストリームについて、stream_idおよびprivate_stream_idとの関係を示している。

MPEG2-System規格では、MPEGで規定されている符号化方式でエンコードされたビデオのエレメンタリストリームは、0xE0乃至0xEFの範囲の値を（0xは、その後に続く文字列が１６進数であることを表す）、エレメンタリストリームを識別するstream_idとして用いて多重化することが規定されている。従って、MPEGで規定されている符号化方式でエンコードされたビデオのエレメンタリストリームについては、0xE0乃至0xEFの範囲の値のstream_idで識別することができる16本のビデオのエレメンタリストリームを、プログラムストリームに多重化することができる。

なお、MPEGで規定されている符号化方式でエンコードされたビデオのエレメンタリストリームの識別は、0xE0乃至0xEFの範囲の値のstream_idで行うことができるので、private_stream_idは不要である（無

視することができる）。

一方、MPEG2-Systemでは、ATRACオーディオストリーム、LPCMオーディオストリーム、字幕ストリームについては、stream_idは定義されていない。

そこで、本実施の形態では、MPEG2-Systemでstream_idが定義されていないエレメンタリストリームについては、そのstream_idに、MPEG2-Systemにおいてprivate_stream_1という属性を表す値である0xBDを採用し、さらに、第１１図に示すように、private_stream_idを用いて識別（特定）を行うこととしている。

即ち、ATRACオーディオストリームの識別には、0x00乃至0x0Fの範囲の値のprivate_stream_idが使用される。従って、プログラムストリームには、16本のATRACオーディオストリームを多重化することができる。また、LPCMオーディオストリームの識別には、0x10乃至0x1Fの範囲の値private_stream_idが使用される。従って、プログラムストリームには、16本のLPCMオーディオストリームを多重化することができる。さらに、字幕ストリームの識別には、0x80乃至0x9Fの範囲の値のprivate_stream_idが使用される。従って、プログラムストリームには、32本の字幕ストリームを多重化することができる。

なお、stream_idおよびprivate_stream_idについては、さらに後述する。

第１０図に戻り、private_stream_idの後には、StaticInfo(), reserved_for_word_alignment（8ビット）が順次配置される。StaticInfo()には、（そのStaticInfo()を含むStreamInfo()に記述された）stream_idおよびprivate_stream_idによって特定されるエレメンタリストリームの再生中に変化しない情報が記述される。StaticInfo()の詳細については、第１２図を参照して後述する。

reserved_for_word_alignmentは、ワードアラインをとるために使用される。

reserved_for_word_alignmentに続いては、number_of_DynamicInfo（8ビット）が配置され、number_of_DynamicInfoは、その後に続いて配置されるpts_change_point（32ビット）とDynamicInfo()のセットの数を表す。

従って、number_of_DynamicInfoに続いては、そのnumber_of_DynamicInfoの数だけのセット数のpts_change_pointとDynamicInfo()の構造が記述される。

pts_change_pointは、それとセットになっているDynamicInfo()の情報が有効になる時刻を表す。ここで、エレメンタリストリームの先頭の時刻を表すpts_change_pointは、そのエレメンタリストリームが格納されたクリップストリームファイルに対応するクリップ情報ファイルClip()の最初に記述されるpresentation_start_timeに等しい。

DynamicInfo()には、stream_idおよびprivate_stream_idによって特定されるエレメンタリストリームの再生中に変化する、いわば動的な情報が記述される。DynamicInfo()に記述された情報は、それとセットになっているpts_change_pointが表す再生時刻となったときに有効になる。なお、DynamicInfo()の詳細については、第１３図を参照して後述する。

number_of_DynamicInfoの数だけのセットのpts_change_pointとDynamicInfo()の後には、EP_map()が配置される。なお、EP_map()については、第１４図を参照して後述する。

「StaticInfo()の説明」

次に、第１２図を参照して、第１０図のStaticInfo()の詳細について説明する。

第１２図は、StaticInfo()のシンタクスを示している。

StaticInfo()は、対応するエレメンタリストリームの属性（種類）により内容が異なっている。StaticInfo()に対応するエレメンタリストリームの属性は、そのStaticInfo()を含む第１０図のStreamInfo()に含まれるstream_idとprivate_stream_idにより判断される。

StaticInfo()に対応するエレメンタリストリームがビデオストリームである場合(stream==VIDEO)、StaticInfo()は、picture_size（4ビット），frame_rate（4ビット），およびcc_flag（1ビット）と、ワードアラインをとるためのreserved_for_word_alingmentとで構成される。

picture_sizeは、ビデオストリームに対応するビデオデータ（によって表示される画像）の大きさを表す。frame_rateは、ビデオストリームに対応するビデオデータのフレーム周波数を表す。cc_flagは、ビデオストリームにクローズドキャプション(Closed Caption)データが含まれているか否かを表す。即ち、例えば、ビデオストリームにクローズドキャプションデータが含まれている場合には、cc_flagは１とされ、ビデオストリームにクローズドキャプションデータが含まれていない場合には、cc_flagは０とされる。

StaticInfo()に対応するエレメンタリストリームがオーディオストリームである場合(stream==AUDIO)、StaticInfo()は、audio_language_code（16ビット），channel_configuration（8ビット），lfe_existence（1ビット）、およびsampling_frequency（4ビット）と、ワードアラインをとるためのreserved_for_word_alingmentとで構成される。

audio_language_codeには、オーディオストリームに含まれているオーディオデータの言語を表すコードが記述される。channel_config

urationは、モノラル(mono)/ステレオ(stereo)/マルチチャネル等の、オーディオストリームに含まれているオーディオデータの属性を表す。lfe_existenceは、オーディオストリームに低域強調チャネルが含まれているかどうかを表し、含まれていれば1となり、含まれていなければ0となる。sampling_frequencyは、オーディオストリームに含まれているオーディオデータのサンプリング周波数を示す情報である。

StaticInfo()に対応するエレメンタリストリームが字幕ストリームである場合(stream==SUBTITLE)、StaticInfo()は、subtitle_language_code（16ビット）およびconfigurable_flag（1ビット）と、ワードアラインをとるためのreserved_for_word_alingmentとで構成される。

subtitle_language_codeには、字幕ストリームに含まれている字幕データの言語を表すコードが記述される。configurable_flagは、字幕ストリームに含まれている字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す情報で、例えば、表示方式の変更が許可されている場合には1が記述され、許可されていない場合には0が記述される。なお、字幕データの表示方式としては、字幕データの表示サイズや、表示位置、表示色、表示パターン（例えば、点滅表示など）、表示方向（例えば、垂直方向や水平方向）などがある。

「DynamicInfo()の説明」

次に、第１３図を参照して、第１０図のDynamicInfo()の詳細について説明する。

第１３図は、DynamicInfo()のシンタクスを示している。

DynamicInfo()の先頭には、ワードアラインのためのreserved_for_

word_alignment（8ビット）が配置されており、その後に続く要素は、DynamicInfo()に対応するエレメンタリストリームの属性により内容が異なっている。DynamicInfo()に対応するエレメンタリストリームの属性は、第１２図で説明したStaticInfo()における場合と同様に、DynamicInfo()を含む第１０図のStreamInfo()に含まれるstream_idとprivate_stream_idにより判断される。

DynamicInfo()には、第１０図で説明したように、エレメンタリストリームの再生中に変化する動的な情報が記述される。この動的な情報は、特に限定されるものではないが、第１３図の実施の形態では、DynamicInfo()に対応するエレメンタリストリームに対応するデータ、即ち、エレメンタリストリームが処理されることによって出力されるデータの出力属性（エレメンタリストリームから得られるデータの出力属性）が、DynamicInfo()に記述される。

具体的には、DynamicInfo()に対応するエレメンタリストリームがビデオストリームである場合(stream==VIDEO)、DynamicInfo()は、display_aspect_ratio（4ビット）と、ワードアラインのためのreserved_for_word_alingmentとで構成される。diplay_aspect_ratioには、ビデオストリームに対応するビデオデータの出力属性（表示方式）としての、例えば、そのビデオデータのアスペクト比が記述される。即ち、diplay_aspect_ratioには、例えば、アスペクト比が、16:9または4:3のうちのいずれであるかを表す情報が記述される。なお、ビデオストリームのDynamicInfo()には、アスペクト比の他、例えば、ビデオデータによって表示される画像のサイズ（Ｘ画素×Ｙ画素）などを記述することが可能である。

DynamicInfo()に対応するエレメンタリストリームがオーディオストリームである場合(stream==AUDIO)、DynamicInfo()は、channel_as

signment（4ビット）と、ワードアラインをとるためのreserved_for_word_alingmentとで構成される。channel_assignmentには、オーディオストリームに2チャネルのオーディオデータが含まれている場合に、その２チャネルの出力属性（出力方式）が記述される。即ち、channel_assignmentには、オーディオデータが、ステレオまたはデュアル（二ヶ国語）のうちのいずれのチャネル割り当てがされているものであるかを表す情報が記述される。

DynamicInfo()に対応するエレメンタリストリームが字幕ストリームである場合(stream==SUBTITLE)、DynamicInfo()は、ワードアラインをとるためのreserved_for_word_alingmentで構成される。即ち、第１３図の実施の形態では、字幕ストリームに関しては、動的な情報としての出力属性は定義されていない。

「EP_map()の説明」

次に、第１４図を参照して、第１０図のEP_map()の詳細について説明する。

第１４図は、EP_map()のシンタクスを示している。

EP_map()には、そのEP_map()を含む第１０図のクリップ情報ファイルClip()に対応するクリップストリームファイルに格納されたプログラムストリームに多重化されているエレメンタリストリーム毎に、各エレメンタリストリームの、デコードを開始することができるデコード開始可能点（エントリポイント）の情報が記述される。

ここで、固定レートのストリームについては、デコード開始可能点は、計算によって求めることができるが、可変レートのストリーム、あるいはMPEG規格にしたがって符号化されたビデオストリームのように、ビデオアクセスアクセスユニットごとにサイズが異なるストリームについては、デコード開始可能点は、計算によって求めることがで

きず、実際にストリームを解析しないと見つけることができない。デコード開始可能点を迅速に認識することは、ランダムアクセスを行うために重要であり、EP_map()によれば、デコード開始可能点を迅速に認識することができる。

なお、MPEG2-Videoでは、Sequence_header()（シーケンスヘッダ）等を含めたイントラピクチャの先頭が、デコード開始可能点である。

EP_map()の先頭には、ワードアラインのためのreserved_for_word_alignment（8ビット）が配置されており、続いてnumber_of_stream_id_entries（8ビット）が配置されている。number_of_stream_id_entriesは、EP_map()にデコード開始可能点の情報が記述されているエレメンタリストリームの本数を表す。

number_of_stream_id_entriesの後には、エレメンタリストリームを識別するための情報と、そのエレメンタリストリームのデコード開始可能点の情報とが、number_of_stream_id_entriesが表す数だけ繰り返し配置される。

即ち、number_of_stream_id_entriesの直後には、エレメンタリストリームを識別する情報としてのstream_id（8ビット）とprivate_stream_id（8ビット）が配置され、それに続けて、number_of_EP_entries（32ビット）が配置される。number_of_EP_entriesは、その直前のstream_idとprivate_stream_idで識別（特定）されるエレメンタリストリームのデコード開始可能点の数を表す。

number_of_EP_entriesの後には、その直前のstream_idとprivate_stream_idで特定されるエレメンタリストリームのデコード開始可能点の情報としてのPTS_EP_start（32ビット）とRPN_EP_start（32ビット）とのセットが、number_of_EP_entriesが表す数だけ繰り返し配置される。

デコード開始可能点の情報の１つであるPTS_EP_startは、上述のようにstream_idとprivate_stream_idで特定されるエレメンタリストリームが多重化されているプログラムストリームが格納されたクリップストリームファイル内での、デコード開始可能点の時刻（再生時刻）を表す。

デコード開始可能点の情報の他の１つであるRPN_EP_startには、上述のようにstream_idとprivate_stream_idで特定されるエレメンタリストリームが多重化されているプログラムストリームが格納されたクリップストリームファイル内での、デコード開始可能点の位置を、プログラムストリームのpack()単位で数えたときの値が記述される。なお、本実施の形態では、pack()のサイズは2048バイトで固定であるとする。また、本実施の形態では、ディスク１０１（第１図）の１セクタが、2048バイトであるとする。

ここで、ビデオストリームについては、そのデコード開始可能点（エントリポイント）の直前に、後述するprivate_stream_2パケット（private_stream_2の属性のPES_packet()）が配置されている。private_stream_2パケットは、そのprivate_stream_2パケットから、次のprivate_stream_2パケットまでの間に配置されているビデオストリームをデコードするのに利用される情報が格納されている。このため、ビデオストリームについては、デコード開始可能点の情報としてのRPN_EP_startには、実際のデコード開始可能点そのものではなく、実際のデコード開始可能点の直前に配置されているprivate_stream_2パケットの先頭の位置が記述される。

また、EP_map()において、デコード開始可能点の情報としてのPTS_EP_startとRPN_EP_startとのセットは、stream_idとprivate_stream_idで特定されるエレメンタリストリームごとに、あらかじめ、昇順に

ソートされている。これにより、デコード開始可能点の情報としてのPTS_EP_startとRPN_EP_startとのセットは、二分探索が可能となっている。

なお、可変レートのストリームや、ビデオアクセスアクセスユニットごとにサイズが異なるストリームを対象としたランダムアクセスの方法は、例えば、特開2000-341640号公報（特願平11-317738号）などに記載されている。

「クリップストリームファイルの説明」

次に、第４図の"STREAM"ディレクトリに置かれる、拡張子がPSのクリップストリームファイル（第４図では、"00001.PS", "00002.PS", "00003.PS"）の内部構造について説明する。

クリップストリームファイルは、MPEG-2 System（ISO/IEC 13818-1）に定義されたMPEG2_Program_Stream()をベースに構成されている。

即ち、第１５図Ａおよび第１５図Ｂは、MPEG-2 System(ISO/IEC 13818-1:2000)規格に記述されているTable2-31, Table2-32, Table2-33を示している。

クリップストリームファイルに格納されたプログラムストリームは、MPEG-2 System規格のTable2-31に定義されているMPEG2_Program_Stream()であり、１つ以上のpack()と、１つのMPEG_program_end_codeで構成される。なお、MPEG2_Program_Stream()の説明は、特許第2785220号などにも記載されている。

１つのpack()は、MPEG-2 System規格のTable2-32に定義されているように、１つのPack_header()と、任意の数のPES_packet()とで構成される。Pack_header()の詳細は、MPEG-2 System規格のTable2-33に定義されている。

ここで、MPEG-2 System規格では、pack()のサイズは、可変長とし

て定義されているが、ここでは、第１４図で説明したように、2048バイトで固定であるとする。さらに、ここでは、１つのpack()のPES_packet()の数は、１つ、２つ、または３つとする。Pack()が後述するprivate_stream_2パケットで始まる場合、その直後（同じPack()内）に対応するビデオストリームのPES_packet()が必ず存在する。またこれに加えて３つ目のPES_packet()としてpadding_packet（パディングパケット）を置くことができる。なおprivate_stream_2パケットは必ずPack()の先頭におかれる。

Pack()がprivate_stream_2パケットで始まらない場合には、Pack()の先頭にはビデオ、オーディオ、字幕などのコンテンツデータの格納されたPES_packet()が置かれる。これに加えて２つ目のPES_packet()としてpadding_packet（パディングパケット）を置くことができる。

第１６図Ａおよび第１６図Ｂ乃至第１８図Ａおよび第１８図Ｂは、MPEG-2 System規格のTable2-17で定義されているPES_packet()を示している。

PES_packet()は、packet_start_code_prefix, stream_id、およびPES_packet_length（第１６図Ａおよび第１６図Ｂ）と、stream_id等により構造の変化するヘッダ部分（stuffing_byteを含む）（第１６図Ａおよび第１６図Ｂ乃至第１８図Ａおよび第１８図Ｂ）と、PES_packet_data_byte（第１８図Ａおよび第１８図Ｂ）とに大別することができる。なお、PES_packet()が、padding_packetである場合(stream_id==padding_stream)、PES_packet_data_byteに代えて、padding_byte(0xFF)（第１８図Ａおよび第１８図Ｂ）が必要な数だけ繰り返し配置される。

ここで、PES_packet()のヘッダ部分には、第１６図Ａおよび第１６図Ｂ、ならびに、第１７図Ａ、第１７図Ｂおよび第１７図Ｃに示すよ

うに、PTS(Presentation Time Stamp)と呼ばれる表示タイミングを示す情報と、DTS(Decoding Time Stamp)と呼ばれるデコードタイミングを示す情報とを配置することができる。本実施の形態では、すべてのアクセスユニット（MPEG2-Systemで定義された、エレメンタリストリームを構成するデコード単位）に対してPTSが付加され、MPEG2-Systemに定める場合にDTSが付加されるとする。

プログラムストリームに多重化されるエレメンタリストリームは、PES_packet()のPES_packet_data_byte（第１８図Ａおよび第１８図Ｂ）に格納される。そして、PES_packet()のstream_idには、そのPES_packet_data_byteに格納されたエレメンタリストリームを識別するために、そのエレメンタリストリームの属性に応じた値が記述される。

PES_packet()のstream_idに記述される値と、エレメンタリストリームの属性（種類）との関係は、MPEG-2 System規格のTable 2-18に定義されている。ここで、第１９図Ａおよび第１９図Ｂに、MPEG-2 System規格のTable 2-18を示す。

本実施の形態では、第１９図Ａおよび第１９図Ｂに示したMPEG-2 System規格で定義されているstream_idのうちの、例えば、第２０図に示す値を採用する。

即ち、本実施の形態では、10111101B, 10111110B, 10111111B, 110xxxxxB, 1110xxxxBの５パターンを、stream_idの値として採用する。なお、"x"は、０または１のうちの任意の値を表す。

そして、private_stream_1と呼ばれる属性のエレメンタリストリームのPES_packet()のstream_idは、第２０図にしたがい、10111101Bとされる。また、padding_packetのPES_packet()のstream_idは、第２０図にしたがい、10111110Bとされる。さらに、private_stream_2と呼ばれる属性のエレメンタリストリームのPES_packet()のstream_id

は、第２０図にしたがい、10111111Bとされる。

また、MPEGで定義されたオーディオストリーム（オーディオのエレメンタリストリーム）のPES_packet()のstream_idは、110xxxxxBとされる。なお、110xxxxxBのうちの下位５ビットxxxxxは、オーディオストリームを区別するオーディオストリームナンバであり、プログラムストリーム（MPEG2_Program_Stream()）には、このオーディオストリームナンバで区別することのできる数である３２（＝2^5）本のオーディオストリーム（MPEGで定義されたオーディオストリーム）を多重化することができる。

さらに、MPEGで定義されたビデオストリーム（ビデオのエレメンタリストリーム）のPES_packet()のstream_idは、1110xxxxBとされる。なお、1110xxxxBのうちの下位４ビットxxxxは、ビデオストリームを区別するビデオストリームナンバであり、プログラムストリームには、このビデオストリームナンバで区別することのできる数である１６（＝2^4）本のビデオストリーム（MPEGで定義されたビデオストリーム）を多重化することができる。

ところで、stream_idが1110xxxxBのPES_packet()は、MPEGで定義されたビデオストリームを格納するために使用され、stream_idが110xxxxxBのPES_packet()は、MPEGで定義されたオーディオストリームを格納するために使用される。一方、MPEGで定義されていない符号化方式（たとえばATRAC方式）のエレメンタリストリームを格納するのに使用するPES_packet()のstream_idは、MPEGでは規定されておらず、従って、MPEGで定義されていない符号化方式のエレメンタリストリームは、MPEGで定義されたビデオストリームやオーディオストリームと同様に、単純に、stream_idを指定して、PES_packet()に格納することはできない。

そこで、本実施の形態では、private_stream_1のPES_packet()のPES_packet_data_byteを拡張し、MPEGで定義されていない符号化方式のエレメンタリストリームを格納する。

ここで、private_stream_1のPES_packet()の、拡張したPES_packet_data_byteを、private_stream1_PES_payload()と記述する。

「private_stream1_PES_payload()の説明」

第２１図は、private_stream1_PES_payload()のシンタクスを示している。

private_stream1_PES_payload()は、private_header()とprivate_payload()とで構成される。private_payload()には、ATRACオーディオストリームや、LPCMオーディオストリーム、字幕ストリームなどの、MPEGで定義されていない符号化方式のエレメンタリストリームが格納される。

private_header()の先頭には、private_stream_id（8ビット）が配置される。private_stream_idは、private_payload()に格納されるエレメンタリストリームを識別する識別情報で、その属性（種類）に応じて、例えば、以下のような値とされる。

即ち、第２２図は、private_stream_idの値と、private_payload()に格納されるエレメンタリストリームの属性との関係を示している。

第２２図では、0000xxxxB, 0001xxxxB, 100xxxxxBの３パターンが、private_stream_idの値として採用されている。なお、"x"は、第２０図における場合と同様に、０または１のうちの任意の値を表す。

第２２図によれば、ATRACオーディオストリームがprivate_payload()に格納されるprivate_stream1_PES_payload()のprivate_stream_idは、0000xxxxBとされる。なお、0000xxxxBのうちの下位４ビットxxxxは、ATRACオーディオストリームを区別するオーディオストリームナ

ンバであり、プログラムストリーム（MPEG2_Program_Stream()）には、このオーディオストリームナンバで区別することのできる数である１６（＝2^4）本のATRACオーディオストリームを多重化することができる。

さらに、第２２図によれば、LPCMオーディオストリームがprivate_payload()に格納されるprivate_stream1_PES_payload()のprivate_stream_idは、0001xxxxBとされる。なお、0001xxxxBのうちの下位４ビットxxxxは、LPCMオーディオストリームを区別するオーディオストリームナンバであり、プログラムストリームには、このオーディオストリームナンバで区別することのできる数である１６（＝2^4）本のLPCMオーディオストリームを多重化することができる。

また、第２２図によれば、字幕ストリームがprivate_payload()に格納されるprivate_stream1_PES_payload()のprivate_stream_idは、100xxxxxBとされる。なお、100xxxxxBのうちの下位５ビットxxxxxは、字幕ストリームを区別する字幕ストリームナンバであり、プログラムストリームには、この字幕ストリームナンバで区別することのできる数である３２（＝2^5）本の字幕ストリームを多重化することができる。

ここで、第２０図と第２２図の関係をまとめたものが、上述した第１１図である。

第２１図に戻り、private_stream1_PES_payload()において、private_stream_idに続く要素は、private_payload()に格納されるエレメンタリストリームの属性により内容が異なっている。private_payload()に格納されるエレメンタリストリームの属性は、private_header()の先頭のprivate_stream_idにより判断される。

private_payload()に格納されるエレメンタリストリームがATRACオ

ーディオストリームである場合(private_stream_id==ATRAC)、将来の拡張用のreserved_for_future_use（8ビット）が配置され、その後、AU_locator（16ビット）が配置される。AU_locatorは、そのAU_locatorの直後の位置を基準として、private_payload()に格納されたATRACオーディオストリームのオーディオアクセスユニット（ATRACオーディオアクセスユニット）（オーディオフレーム）の先頭位置を表す。private_payload()にオーディオアクセスユニットが存在しない場合、AU_locatorには、例えば0xFFFFが記述される。

private_payload()に格納されるエレメンタリストリームがLPCMオーディオストリームである場合(private_stream_id==LPCM)、fs_flag（1ビット），reserved_for_future_use（3ビット），ch_flag（4ビット）、およびAU_locator（16ビット）が順次配置される。

fs_flagは、private_payload()に格納されるLPCMオーディオストリームのサンプリング周波数を示す。即ち、例えば、サンプリング周波数が48KHzの場合、fs_flagは0とされ、サンプリング周波数が44.1KHzの場合、fs_flagは1とされる。

ch_flagは、private_payload()に格納されるLPCMオーディオストリームのチャネル数を示す。例えば、LPCMオーディオストリームがモノラルの場合、ch_flagは1とされ、LPCMオーディオストリームがステレオの場合、ch_flagは2とされる。

AU_locatorは、そのAU_locatorの直後の位置を基準として、private_payload()に格納されるLPCMオーディオストリームのオーディオアクセスユニット（LPCMオーディオアクセスユニット）（オーディオフレーム）の先頭位置を示す。private_payload()にオーディオアクセスユニットが存在しない場合、AU_locatorには、例えば0xFFFFが記述される。

private_payload()に格納されるエレメンタリストリームが字幕ストリームである場合(private_stream_id==SUBTITLE)、将来の拡張のためのreserved_for_future_use（８ビット）が配置され、その後に、AU_locator（16ビット）が配置される。AU_locatorは、そのAU_locatorの直後の位置を基準として、private_payload()に格納される字幕ストリームの字幕アクセスユニットの先頭位置を示す。private_payload()に字幕アクセスユニットが存在しない場合、AU_locatorには、例えば0xFFFFが記述される。

「private_stream2_PES_payload()の説明」

次に、第２３図は、private_stream2_PES_payload()のシンタクスを示している。

private_stream2_PES_payload()は、private_stream_2のPES_packet()のPES_packet_data_byte（第１８図Ａおよび第１８図Ｂ）を拡張したもの、即ち、private_stream_2のPES_packet()の、拡張したPES_packet_data_byteであり、ビデオストリームのデコードに利用される情報が記述される。

本実施の形態では、private_stream_2のPES_packet()は、ビデオストリームにおけるデコード開始可能点の直前に配置される。従って、本実施の形態では、プログラムストリームからprivate_stream_2のPES_packet()を見つければ、その直後のビデオストリームからデコードを開始することができる。

ここで、上述した第１４図のEP_map()のRPN_EP_startは、ビデオストリームについては、private_stream_2のPES_packet()の先頭の位置を示す。

private_stream2_PES_payload()の先頭には、将来の拡張用のreserved_for_future_use（8ビット）が配置され、続けて、video_stream_

id（8ビット），1stRef_picture（16ビット），2ndRef_picture（16ビット），3rdRef_picture（16ビット），4thRef_picture（16ビット），au_information()、およびVBI()が、順次配置される。

video_stream_idには、private_stream_2のPES_packet()の直後に配置されるビデオストリームのPES_packet()のstream_id（と同一の値）が記述される。このvideo_stream_idによって、private_stream_2のPES_packet()（のprivate_stream2_PES_payload()）に格納された情報を利用してデコードされるビデオストリーム（が格納されたPES_packet()）が特定される。

1stRef_picture, 2ndRef_picture, 3rdRef_picture, 4thRef_pictureは、video_stream_idによって特定されるビデオストリームの、private_stream_2のPES_packet()から次のprivate_stream_2のPES_packet()までの中の１，２，３，４番目の参照画像を含む最後のpack()の位置を相対値で、それぞれ表す。なお、1stRef_picture, 2ndRef_picture, 3rdRef_picture, 4thRef_pictureについては、例えば、特開平09-46712号公報（特願平07-211420号）に、bytes_to_first_P_pic bytes_to_second_P_picとして、その詳細が開示されている。

au_information()には、private_stream_2のPES_packet()から、次のprivate_stream_2のPES_packet()までのビデオストリームの中のビデオアクセスユニットに関する情報が記述される。au_information()の詳細については、第２４図を参照して後述する。

VBI()は、Closed Captionの情報を記述するために使用される。

以上のようなprivate_stream2_PES_payload()を有するprivate_stream_2のPES_packet()は、ビデオストリームごとの、デコード開始可能点ごとに配置される。

次に、第２４図は、第２３図のau_information()のシンタクスを示

している。

au_information()の先頭には、length（16ビット）が配置される。lengthは、それを含むau_information()のサイズを表す。lengthに続いては、reserved_for_word_alignment（8ビット）、およびnumber_of_access_unit（8ビット）が順次配置される。reserved_for_word_alignmentは、ワードアラインをとるために使用される。

number_of_access_unitは、それを含むprivate_stream_2のPES_packet()から、次のprivate_stream_2のPES_packet()までの間に含まれるビデオアクセスユニット（ピクチャ）の数を表す。

即ち、number_of_access_unitは、第２３図のprivate_stream2_PES_payload()が同一のvideo_stream_idを有するprivate_stream_2のPES_packet()の中で、このau_informatnio()から次のau_information()の直前までに（このau_infromation()がクリップストリームファイルで最後のau_information()であれば、クリップストリームファイルの最後までに）、video_stream_idで示されるビデオストリームに含まれるアクセスユニット（ピクチャ）の数を示す。

number_of_access_unitの後には、そのnumber_of_access_unitの数だけforループの内容が配置される。即ち、number_of_access_unitを含むprivate_stream_2のPES_packet()から、次のprivate_stream_2のPES_packet()までの間に含まれる１以上のビデオアクセスユニットそれぞれに関する情報が配置される。

forループ内に配置される情報（ビデオアクセスユニットに関する情報）は、以下のようになっている。

即ち、forループ内には、pic_struct_copy（4ビット），au_ref_flag（1ビット），AU_length（21ビット），reservedが配置される。

pic_struct_copyには、ビデオストリームがMPEG4-AVC（ISO/IEC 14

496-10）の場合に、対応するビデオアクセスユニットに対して設定されている、ISO/IEC 14496-10, D.2.2に定義されているpic_struct（）のコピーが記述される。なお、pic_struct（）は、例えば、ピクチャをフレームとして表示する、あるいは、ピクチャのトップフィールドを表示して、その後、ボトムフィールドを表示する、などといった情報である。

au_ref_flagは、対応するアクセスユニットが、他のアクセスユニット（のピクチャ）のデコードにあたって参照される参照画像であるか否かを表し、参照画像である場合には１とされ、参照画像でない場合には０とされる。

AU_lengthは、対応するアクセスユニットのサイズをバイト単位で表す。

［ディスク１０１に記録されたデータの具体例］

次に、第２５図乃至第２８図は、第１図のディスク１０１に記録された、上述したようなフォーマットのデータの具体例を示している。

ここで、第２５図乃至第２８図では、ビデオストリームとしては、MPEG2-Videoを採用し、オーディオストリームとしては、ATRACオーディオストリームを採用している。但し、ビデオストリームやオーディオストリームは、これに限定されるものではない。即ち、ビデオストリームとしては、例えば、MPEG4-VisualやMPEG4-AVCなどを採用することができる。さらに、オーディオストリームとしては、例えば、MPEG1/2/4オーディオやLPCMオーディオストリームなどを採用することができる。

なお、字幕ストリームは、ビデオストリームやオーディオストリームと異なり、同じ間隔で連続的なデコード・表示（出力）が行われるとは限らない。すなわち、字幕ストリームは、時折、第２図Ａおよび

第２図Ｂのバッファ制御モジュール２１５から字幕デコーダ制御モジュール２１８に供給されてデコードされる。

第２５図乃至第２８図は、ディスク１０１において、第４図に示したように、"CLIP"ディレクトリに、３つのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"が記録され、"STREAM"ディレクトリに、その３つのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"それぞれに対応する３つのクリップストリームファイル"00001.PS", "00002.PS", "00003.PS"が記録されている場合の、"PLAYLIST.DAT"ファイル、クリップ情報ファイル"00001.CLP", "00002.CLP"、および"00003.CLP"等の具体的な例を示している。但し、第２５図乃至第２８図では、"PLAYLIST.DAT"ファイル等のデータの一部を省略してある。

即ち、第２５図は、第５図で説明した"PLAYLIST.DAT"ファイルの具体例を示している。

第２５図では、number_of_PlayListsは2となっており、従って、"PLAYLIST.DAT"ファイルに含まれるPlayList()の数は2である。第２５図では、その２つのPlayList()のうちの１番目がPlayList #0と、２番目がPlayList #1と、それぞれ記載されている。

１番目のPlayList()であるPlayList #0については、capture_enable_flag_PlayListが１となっており、従って、PlayList #0にしたがって再生されるビデオデータの２次利用が許可されている。また、PlayList #0については、number_of_PlayItemsが２となっており、従って、PlayList #0に含まれるPlayItem()の数は2である。第２５図では、その２つのPlayItem()であるPlayItem#0とPlayItem#1の具体例が、「PlayList#0」の欄の下方に記載されている。

PlayList #0に含まれる１番目のPlayItem()であるPlayItem#0では

、第６図で説明したClip_Information_file_nameが"00001.CLP"に、In_timeが180,090に、OUT_timeが27,180,090に、それぞれなっている。従って、PlayList #0のPlayItem#0によって再生されるクリップは、クリップ情報ファイル"00001.CLP"に対応するクリップストリームファイル"00001.PS"の、時刻180,090から27,180,090までである。

PlayList #0に含まれる２番目のPlayItem()であるPlayItem#1では、第６図で説明したClip_Information_file_nameが"00002.CLP"に、In_timeが90,000に、OUT_timeが27,090,000に、それぞれなっている。従って、PlayList #0のPlayItem#1によって再生されるクリップは、クリップ情報ファイル"00002.CLP"に対応するクリップストリームファイル"00002.PS"の、時刻90,000から27,090,000までである。

一方、第２５図において、２番目のPlayList()であるPlayList #1については、capture_enable_flag_PlayListが0となっており、従って、PlayList #1にしたがって再生されるビデオデータの２次利用が許可されていない（禁止されている）。また、PlayList #1については、number_of_PlayItemsが１となっており、従って、PlayList #1に含まれるPlayItem()の数は１である。第２５図では、その１つのPlayItem()であるPlayItem#0の具体例が、「PlayList#1」の欄の下方に記載されている。

PlayList #1に含まれる１つのPlayItem()であるPlayItem#0では、第６図で説明したClip_Information_file_nameが"00003.CLP"に、In_timeが90,000に、OUT_timeが81,090,000に、それぞれなっている。従って、PlayList #1のPlayItem#0によって再生されるクリップは、クリップ情報ファイル"00003.CLP"に対応するクリップストリームファイル"00003.PS"の、時刻90,000から81,090,000までである。

次に、第２６図Ａおよび第２６図Ｂは、第１０図で説明したクリッ

プ情報ファイルClip()の具体例を示している。即ち、第２６図Ａおよび第２６図Ｂは、第４図のクリップ情報ファイル"00001.CLP"，"00002.CLP"，"00003.CLP"の具体例を示している。

クリップ情報ファイル"00001.CLP"においては、presentation_start_timeが90,000に、presentation_end_timeが27,990,000に、それぞれなっている。従って、クリップ情報ファイル"00001.CLP"に対応するクリップストリームファイル"00001.PS"に格納されたプログラムストリームによれば、310秒分（(27,990,000-90,000)/90kHz）のコンテンツが利用可能である。

また、クリップ情報ファイル"00001.CLP"においては、capture_enable_flag_Clipが１になっており、従って、クリップ情報ファイル"00001.CLP"に対応するクリップストリームファイル"00001.PS"に格納されたプログラムストリームに多重化されたビデオストリーム（に対応するビデオデータ）については、その２次利用が許可されている。

さらに、第２６図Ａおよび第２６図Ｂにおいて、クリップ情報ファイル"00001.CLP"のnumber_of_streamsは4となっており、従って、対応するクリップストリームファイル"00001.PS"に格納されたプログラムストリームには、４本のエレメンタリストリームが多重化されている。

いま、その４本のエレメンタリストリームを、stream#0，stream#1，stream#2，stream#3とすると、第２６図Ａおよび第２６図Ｂでは、その４本のエレメンタリストリームstream#0，stream#1，stream#2，stream#3それぞれのStreamInfo()（第１０図）の具体例が、「"00001.CLP"」の欄の下方に記載されている。

クリップストリームファイル"00001.PS"の１本目のエレメンタリストリームstream#0については、stream_idが0xE0となっており、従っ

て、このエレメンタリストリームstream#0は、第２０図および第２２図（あるいは第１１図）で説明したように、ビデオストリームである。なお、本実施の形態では、ビデオストリームについては、上述したように、private_stream_idは無関係であるが、第２６図Ａおよび第２６図Ｂでは、0x00になっている。

また、クリップストリームファイル"00001.PS"の１本目のエレメンタリストリームであるビデオストリームstream#0については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のpicture_sizeが'720×480'に、frame_rateが'29.97Hz'に、cc_flagが'Yes'に、それぞれなっている。従って、このビデオストリームstream#0は、720×480ピクセルの、フレーム周期が29.97Hzのビデオデータであり、さらに、クローズドキャプションデータを含んでいる。

さらに、クリップストリームファイル"00001.PS"の１本目のエレメンタリストリームであるビデオストリームstream#0については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが０になっており、pts_change_pointとDynamicInfo()とのセットは存在しない。

次に、クリップストリームファイル"00001.PS"の２本目のエレメンタリストリームstream#1については、stream_idが0xBDとなっており、private_stream_idが0x00となっている。従って、このエレメンタリストリームstream#1は、第２０図および第２２図で説明したように、ATRACオーディオストリームである。

また、クリップストリームファイル"00001.PS"の２本目のエレメンタリストリームであるATRACオーディオストリームstream#1については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のaudio_language_codeが'日本語'に、channel_configurationが'STEREO'に、lfe_existenceが'NO'に、sampling_frequencyが'48kHz'に、それぞれ

なっている。従って、このATRACオーディオストリームstream#1は、日本語で、かつステレオのオーディオデータである。また、低域強調チャネルは含まれておらず、サンプリング周波数は、48kHzである。

さらに、クリップストリームファイル"00001.PS"の２本目のエレメンタリストリームであるATRACオーディオストリームstream#1については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが０になっており、pts_change_pointとDynamicInfo()とのセットは存在しない。

次に、クリップストリームファイル"00001.PS"の３本目のエレメンタリストリームstream#2については、stream_idが0xBDとなっており、private_stream_idが0x80となっている。従って、このエレメンタリストリームstream#2は、第２０図および第２２図で説明したように、字幕ストリームである。

また、クリップストリームファイル"00001.PS"の３本目のエレメンタリストリームである字幕ストリームstream#2については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のsubtitle_language_codeが'日本語'に、configurable_flagが０に、それぞれなっている。従って、この字幕ストリームstream#2は、日本語の字幕データであり、また、その表示方式を変更することは許可されていない（禁止されている）。

さらに、クリップストリームファイル"00001.PS"の３本目のエレメンタリストリームである字幕ストリームstream#2については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが０になっており、pts_change_pointとDynamicInfo()とのセットは存在しない。

次に、クリップストリームファイル"00001.PS"の４本目のエレメンタリストリームstream#3については、stream_idが0xBDとなっており

、private_stream_idが0x81となっている。従って、このエレメンタリストリームstream#3は、第２０図および第２２図で説明したように、字幕ストリームである。

なお、クリップストリームファイル"00001.PS"の３本目のエレメンタリストリームである字幕ストリームstream#2と、４本目のエレメンタリストリームである字幕ストリームstream#3とを区別するために、それぞれのprivate_stream_idは、0x80と0x81とになっている。

また、クリップストリームファイル"00001.PS"の４本目のエレメンタリストリームである字幕ストリームstream#2については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のsubtitle_language_codeが'日本語'に、configurable_flagが１に、それぞれなっている。従って、この字幕ストリームstream#3は、日本語の字幕データであり、また、その表示方式を変更することが許可されている。

さらに、クリップストリームファイル"00001.PS"の４本目のエレメンタリストリームである字幕ストリームstream#3については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが０になっており、pts_change_pointとDynamicInfo()とのセットは存在しない。

次に、第２６図Ａおよび第２６図Ｂにおいて、クリップ情報ファイル"00002.CLP"については、presentation_start_timeが90,000に、presentation_end_timeが27,090,000に、それぞれなっている。従って、クリップ情報ファイル"00002.CLP"に対応するクリップストリームファイル"00002.PS"に格納されたプログラムストリームによれば、300秒分（(27,090,000-90,000)/90kHz）のコンテンツが利用可能である。

また、クリップ情報ファイル"00002.CLP"においては、capture_enable_flag_Clipが０になっており、従って、クリップ情報ファイル"00

002.CLP"に対応するクリップストリームファイル"00002.PS"に格納されたプログラムストリームに多重化されたビデオストリーム（に対応するビデオデータ）については、その２次利用が許可されていない（禁止されている）。

さらに、第２６図Ａおよび第２６図Ｂにおいて、クリップ情報ファイル"00002.CLP"のnumber_of_streamsは4となっており、従って、対応するクリップストリームファイル"00002.PS"に格納されたプログラムストリームには、上述したクリップストリームファイル"00001.PS"における場合と同様に、４本のエレメンタリストリームが多重化されている。

いま、その４本のエレメンタリストリームを、stream#0, stream#1, stream#2, stream#3とすると、第２６図Ａおよび第２６図Ｂでは、その４本のエレメンタリストリームstream#0, stream#1, stream#2, stream#3それぞれのStreamInfo()（第１０図）の具体例が、「"00002.CLP"」の欄の下方に記載されている。

ここで、第２６図Ａおよび第２６図Ｂでは、クリップストリームファイル"00002.PS"の１乃至４本目のエレメンタリストリームstream#0乃至#3それぞれのStreamInfo()の内容は、上述したクリップストリームファイル"00001.PS"の１乃至４本目のエレメンタリストリームstream#0乃至#3それぞれのStreamInfo()の内容と同一になっているので、その説明は省略する。

なお、上述のように、クリップストリームファイル"00002.PS"の１乃至４本目のエレメンタリストリームstream#0乃至#3それぞれのStreamInfo()の内容は、クリップストリームファイル"00001.PS"の１乃至４本目のエレメンタリストリームstream#0乃至#3それぞれのStreamInfo()の内容と同一であるので、クリップストリームファイル"00002.P

S"の１本目のエレメンタリストリームstream#0はビデオストリームであり、２本目のエレメンタリストリームstream#1はATRACオーディオストリームである。また、その３本目のエレメンタリストリームstream#2と、４本目のエレメンタリストリームstream#3は、いずれも字幕ストリームである。

次に、第２６図Ａおよび第２６図Ｂにおいて、クリップ情報ファイル"00003.CLP"については、presentation_start_timeが90,000に、presentation_end_timeが81,090,000に、それぞれなっている。従って、クリップ情報ファイル"00003.CLP"に対応するクリップストリームファイル"00003.PS"に格納されたプログラムストリームによれば、900秒分（(81,090,000-90,000)/90kHz）のコンテンツが利用可能である。

また、クリップ情報ファイル"00003.CLP"においては、capture_enable_flag_Clipが１になっており、従って、クリップ情報ファイル"00003.CLP"に対応するクリップストリームファイル"00003.PS"に格納されたプログラムストリームに多重化されたビデオストリームについては、その２次利用が許可されている。

さらに、第２６図Ａおよび第２６図Ｂにおいて、クリップ情報ファイル"00003.CLP"のnumber_of_streamsは３となっており、従って、対応するクリップストリームファイル"00003.PS"に格納されたプログラムストリームには、３本のエレメンタリストリームが多重化されている。

いま、その３本のエレメンタリストリームを、stream#0, stream#1, stream#2とすると、第２６図Ａおよび第２６図Ｂでは、その３本のエレメンタリストリームstream#0, stream#1, stream#2それぞれのStreamInfo()（第１０図）の具体例が、「"00003.CLP"」の欄の下方に

記載されている。

クリップストリームファイル"00003.PS"の１本目のエレメンタリストリームstream#0については、stream_idが0xE0となっており、従って、このエレメンタリストリームstream#0は、第２０図および第２２図（あるいは第１１図）で説明したように、ビデオストリームである。なお、クリップストリームファイル"00001.PS"の１本目のエレメンタリストリームstream#0と同様に、private_stream_idは、0x00になっている。

また、クリップストリームファイル"00003.PS"の１本目のエレメンタリストリームであるビデオストリームstream#0については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のpicture_sizeが'720×480'に、frame_rateが'29.97Hz'に、cc_flagが'No'に、それぞれなっている。従って、このビデオストリームstream#0は、720×480ピクセルの、フレーム周期が29.97Hzのビデオデータであり、さらに、クローズドキャプションデータを含んでいない。

さらに、クリップストリームファイル"00003.PS"の１本目のエレメンタリストリームであるビデオストリームstream#0については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが２になっており、従って、そのStreamInfo()には、pts_change_pointとDynamicInfo()とのセットが２セット記述されている。

次に、クリップストリームファイル"00003.PS"の２本目のエレメンタリストリームstream#1については、stream_idが0xE1となっており、従って、このエレメンタリストリームstream#1は、第２０図および第２２図（あるいは第１１図）で説明したように、ビデオストリームである。なお、クリップストリームファイル"00003.PS"の１本目のエレメンタリストリームであるビデオストリームstream#0と、２本目の

エレメンタリストリームであるビデオストリームstream#1とを区別するために、それぞれのstream_idは、0xE0と0xE1とになっている。また、クリップストリームファイル"00001.PS"の１本目のエレメンタリストリームstream#0と同様に、private_stream_idは、0x00になっている。

また、クリップストリームファイル"00003.PS"の２本目のエレメンタリストリームであるビデオストリームstream#1については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のpicture_size, frame_rate, cc_flagが、１本目のエレメンタリストリームであるビデオストリームstream#0についてのものと同一になっている。従って、クリップストリームファイル"00003.PS"の２本目のエレメンタリストリームであるビデオストリームstream#1は、720×480ピクセルの、フレーム周期が29.97Hzのビデオデータであり、さらに、クローズドキャプションデータを含んでいない。

さらに、クリップストリームファイル"00003.PS"の２本目のエレメンタリストリームであるビデオストリームstream#1については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが０になっており、pts_change_pointとDynamicInfo()とのセットは存在しない。

次に、クリップストリームファイル"00003.PS"の３本目のエレメンタリストリームstream#2については、stream_idが0xBDとなっており、private_stream_idが0x00となっている。従って、このエレメンタリストリームstream#2は、第２０図および第２２図で説明したように、ATRACオーディオストリームである。

また、クリップストリームファイル"00003.PS"の３本目のエレメンタリストリームであるATRACオーディオストリームstream#2については、そのStreamInfo()に含まれるStaticInfo()（第１２図）のaudio_

language_code, channel_configuration, lfe_existence, sampling_frequencyが、クリップストリームファイル"00001.PS"の２本目のエレメンタリストリームであるATRACオーディオストリームstream#1のものと同一になっている。従って、クリップストリームファイル"00003.PS"の３本目のエレメンタリストリームであるATRACオーディオストリームstream#2は、日本語で、かつステレオのオーディオデータである。また、低域強調チャネルは含まれておらず、サンプリング周波数は、48kHzである。

さらに、クリップストリームファイル"00003.PS"の３本目のエレメンタリストリームであるATRACオーディオストリームstream#2については、StreamInfo()（第１０図）のnumber_of_DynamicInfoが３になっており、従って、そのStreamInfo()には、pts_change_pointとDynamicInfo()とのセットが３セット記述されている。

次に、第２７図は、第１０図で説明したクリップ情報ファイルClip()のうちのEP_map()具体例を示している。即ち、第２７図は、第４図のクリップ情報ファイル"00001.CLP"の中の、第１４図のEP_map()の具体例を示している。

第２７図において、EP_map()のnumber_of_stream_id_entriesは1になっており、従って、このEP_map()には、１つのエレメンタリストリームについてのデコード開始可能点の情報が記述されている。

また、第２７図のEP_map()では、stream_idが0xE0になっている。従って、第２０図および第２２図で説明したことから、EP_map()には、0xE0となっているstream_idによって特定されるビデオストリームについてのデコード開始可能点の情報（PTS_EP_startとRPN_EP_start（第１４図））が記述されている。即ち、第２７図は、クリップ情報ファイル"00001.CLP"のEP_map()であり、クリップ情報ファイル"0000

1.CLP"に対応するクリップストリームファイル"00001.CLP"において、stream_idが0xE0のエレメンタリストリームは、第２６図Ａおよび第２６図Ｂで説明したように、クリップストリームファイル"00001.CLP"の１本目のビデオストリームstream#0であるから、第２７図のEP_map()に記述されている情報は、そのビデオストリームstream#0のデコード開始可能点のPTS_EP_startとRPN_EP_startである。

第２７図では、クリップストリームファイル"00001.CLP"の１本目のビデオストリームstream#0のデコード開始可能点のうちの、先頭から５点のPTS_EP_startとRPN_EP_startが記載されており、６点目以降のPTS_EP_startとRPN_EP_startの記載は省略してある。

なお、第２７図のEP_map()において、private_stream_idは0x00になっているが、stream_idがビデオストリームを表している場合は、上述したように、private_stream_idは無関係である（無視される）。

次に、第２８図は、第２５図で説明したPlayList #0とPlayList #1（第５図のPlayList()）の中のPlayListMark()の具体例を示している。

第２８図上側は、PlayList #0のPlayListMark()（第７図）を示している。

第２８図上側において、PlayList #0のPlayListMark()におけるnumber_of_PlayList_marksは7になっており、従って、PlayList #0（のPlayListMark()）に含まれるMark()の数は7である。

また、第２８図上側において、PlayList #0に含まれる７つのMark()のうちの１番目のMark()であるMark#0は、mark_type（第７図）が'Chapter'になっているので、チャプタマークである。さらに、Mark#0は、ref_to_PlayItem_id（第７図）が0になっているので、PlayList

#0に含まれる第２５図の２つのPlayItem#0と#1のうちの、PlayItem#0に属する。また、Mark#0は、mark_time_stampが180,090になっているので、PlayList #0に含まれるPlayItem#0によって再生されるクリップストリームファイルの時刻（再生時刻）180,090上の印である。さらに、Mark#0は、entry_ES_stream_idおよびentry_ES_private_stream_idがいずれも0になっているので、いずれのエレメンタリストリームにも関連付けられていない。また、Mark#0は、mark_dataが1になっているので、番号が１のチャプタを表す。

なお、ここでは、PlayList #0に含まれるPlayItem#0によって再生されるクリップストリームファイルは、第２５図で説明した、そのPlayItem#0のClip_Infomation_file_nameに記述されている"00001.CLP"から特定されるクリップストリームファイル"00001.PS"であり、従って、Mark#0のmark_time_stampが表す、上述した時刻180,090は、クリップストリームファイル"00001.PS"の時刻である。

第２８図上側において、PlayList #0に含まれる７つのMark()のうちの５番目のMark()であるMark#4も、１番目のMark#0と同様のチャプタマークである。

即ち、５番目のMark()であるMark#4は、mark_type（第７図）が'Chapter'になっているので、チャプタマークである。さらに、Mark#4は、ref_to_PlayItem_id（第７図）が1になっているので、PlayList #0に含まれる第２５図の２つのPlayItem#0と#1のうちの、PlayItem#1に属する。また、Mark#4は、mark_time_stampが90,000になっているので、PlayList #0に含まれるPlayItem#1によって再生されるクリップストリームファイルの時刻90,000上の印である。さらに、Mark#4は、entry_ES_stream_idおよびentry_ES_private_stream_idがいずれも0になっているので、いずれのエレメンタリストリームにも関連付けら

れていない。また、Mark#4は、mark_dataが2になっているので、番号が２のチャプタを表す。

なお、ここでは、PlayList #0に含まれるPlayItem#1によって再生されるクリップストリームファイルは、第２５図で説明した、そのPlayItem#1のClip_Infomation_file_nameに記述されている"00002.CLP"から特定されるクリップストリームファイル"00002.PS"であり、従って、Mark#4のmark_time_stampが表す、上述した時刻90,000は、クリップストリームファイル"00002.PS"の時刻である。

また、第２８図上側において、PlayList #0に含まれる７つのMark()のうちの２番目のMark()であるMark#1は、mark_type（第７図）が'Index'になっているので、インデクスマークである。さらに、Mark#1は、ref_to_PlayItem_id（第７図）が0になっているので、PlayList #0に含まれる第２５図の２つのPlayItem#0と#1のうちの、PlayItem#0に属する。また、Mark#1は、mark_time_stampが5,580,090になっているので、PlayList #0に含まれるPlayItem#0によって再生されるクリップストリームファイルの時刻5,580,090上の印である。さらに、Mark#1は、entry_ES_stream_idおよびentry_ES_private_stream_idがいずれも0になっているので、いずれのエレメンタリストリームにも関連付けられていない。また、Mark#1は、mark_dataが1になっているので、番号が１のインデクスを表す。

なお、ここでは、PlayList #0に含まれるPlayItem#0によって再生されるクリップストリームファイルは、上述したように、クリップストリームファイル"00001.PS"であり、従って、Mark#1のmark_time_stampが表す、上述した時刻5,580,090は、クリップストリームファイル"00001.PS"の時刻である。

第２８図上側において、PlayList #0に含まれる７つのMark()のう

ちの３番目、６番目、７番目のMark()であるMark#2, Mark#5, Mark#6も、２番目のMark#1と同様のインデクスマークである。

また、第２８図上側において、PlayList #0に含まれる７つのMark()のうちの４番目のMark()であるMark#3は、mark_type（第７図）が'Event'になっているので、イベントマークである。さらに、Mark#3は、ref_to_PlayItem_id（第７図）が0になっているので、PlayList #0に含まれる第２５図の２つのPlayItem#0と#1のうちの、PlayItem#0に属する。また、Mark#3は、mark_time_stampが16,380,090になっているので、PlayList #0に含まれるPlayItem#0によって再生されるクリップストリームファイルの時刻16,380,090上の印である。さらに、Mark#3は、entry_ES_stream_idおよびentry_ES_private_stream_idがいずれも0になっているので、いずれのエレメンタリストリームにも関連付けられていない。また、Mark#3は、mark_dataが0になっているので、引数として0を伴うイベントを発生させる。

なお、ここでは、PlayList #0に含まれるPlayItem#0によって再生されるクリップストリームファイルは、上述したように、クリップストリームファイル"00001.PS"であり、従って、Mark#3のmark_time_stampが表す、上述した時刻16,380,090は、クリップストリームファイル"00001.PS"の時刻である。

ここで、第２８図上側では、PlayList #0のPlayListMark()の一覧表の欄外の右側に、Mark()が属するPlayItem()の先頭からの時間を示してあり、さらにその右側に、PlayList #0の先頭からの時刻を示してある。

次に、第２８図下側は、PlayList#1のPlayListMark()（第７図）を示している。

第２８図下側において、PlayList#1のPlayListMark()におけるnumb

er_of_PlayList_marksは3になっており、従って、PlayList#1（のPlayListMark()）に含まれるMark()の数は3である。

また、第２８図下側において、PlayList#1に含まれる３つのMark()のうちの１番目のMark()であるMark#0は、mark_type（第７図）が'Chapter'になっているので、チャプタマークである。さらに、Mark#0は、ref_to_PlayItem_id（第７図）が0になっているので、PlayList#1に含まれる第２５図の１つのPlayItem#0に属する。また、Mark#0は、mark_time_stampが90,000になっているので、PlayList#1に含まれるPlayItem#0によって再生されるクリップストリームファイルの時刻90,000上の印である。さらに、Mark#0は、entry_ES_stream_idおよびentry_ES_private_stream_idがいずれも0になっているので、いずれのエレメンタリストリームにも関連付けられていない。また、Mark#0は、mark_dataが０になっているので、番号が０のチャプタを表す。

なお、ここでは、PlayList#1に含まれるPlayItem#0によって再生されるクリップストリームファイルは、第２５図で説明した、そのPlayItem#0のClip_Infomation_file_nameに記述されている"00003.CLP"から特定されるクリップストリームファイル"00003.PS"であり、従って、Mark#0のmark_time_stampが表す、上述した時刻90,000は、クリップストリームファイル"00003.PS"の時刻である。

第２８図下側において、PlayList#1に含まれる３つのMark()のうちの２番目のMark()であるMark#1は、mark_type（第７図）が'Event'になっているので、イベントマークである。さらに、Mark#1は、ref_to_PlayItem_id（第７図）が0になっているので、PlayList#1に含まれる第２５図の１つのPlayItem#0に属する。また、Mark#1は、mark_time_stampが27,090,000になっているので、PlayList#1に含まれるPlayItem#0によって再生されるクリップストリームファイルの時刻27,090,

000上の印である。さらに、Mark#1は、entry_ES_stream_idが0xE0で、entry_ES_private_stream_idが0になっているので、stream_idが0xE0で特定（識別）されるエレメンタリストリーム、即ち、第２０図および第２２図で説明したように、ビデオストリームに関連付けられている。また、Mark#1は、mark_dataが1になっているので、引数として1を伴うイベントを発生させる。

なお、ここでは、PlayList#1に含まれるPlayItem#0によって再生されるクリップストリームファイルは、上述したように、クリップストリームファイル"00003.PS"であり、従って、Mark#1のmark_time_stampが表す、上述した時刻27,090,000は、クリップストリームファイル"00003.PS"の時刻である。

また、Mark#1が関連付けられている、stream_idが0xE0のビデオストリームは、そのMark#1が属する、第２５図のPlayList#1に含まれるPlayItem#0のClip_Infomation_file_nameに記述されている"00003.CLP"に記述されているstream_idが0xE0のビデオストリーム、即ち、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"から特定される、クリップストリームファイル"00003.PS"に多重化されている３本のエレメンタリストリームstream#0乃至#2のうちの、１本目のエレメンタリストリーム（ビデオストリーム）stream#0である。

次に、第２８図下側において、PlayList#1に含まれる３つのMark()のうちの３番目のMark()であるMark#2は、mark_type（第７図）が'Event'になっているので、イベントマークである。さらに、Mark#2は、ref_to_PlayItem_id（第７図）が0になっているので、PlayList#1に含まれる第２５図の１つのPlayItem#0に属する。また、Mark#1は、mark_time_stampが27,540,000になっているので、PlayList#1に含まれるPlayItem#0によって再生されるクリップストリームファイルの時刻

27,540,000上の印である。さらに、Mark#2は、entry_ES_stream_idが0xE1で、entry_ES_private_stream_idが0になっているので、stream_idが0xE1で特定（識別）されるエレメンタリストリーム、即ち、第２０図および第２２図で説明したように、ビデオストリームに関連付けられている。また、Mark#2は、mark_dataが2になっているので、引数として2を伴うイベントを発生させる。

なお、ここでは、PlayList#1に含まれるPlayItem#0によって再生されるクリップストリームファイルは、上述したように、クリップストリームファイル"00003.PS"であり、従って、Mark#2が表す、上述した時刻27,540,000は、クリップストリームファイル"00003.PS"の時刻である。

また、Mark#2が関連付けられている、stream_idが0xE1のビデオストリームは、そのMark#2が属する、第２５図のPlayList#1に含まれるPlayItem#0のClip_Infomation_file_nameに記述されている"00003.CLP"に記述されているstream_idが0xE1のビデオストリーム、即ち、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"から認識される、クリップストリームファイル"00003.PS"に多重化されている３本のエレメンタリストリームstream#0乃至#2のうちの、２本目のエレメンタリストリーム（ビデオストリーム）stream#1である。

ここで、第２８図下側では、PlayList#1のPlayListMark()の一覧表の欄外の右側に、Mark()が属するPlayItem()の先頭からの時刻を示してある。

なお、第２８図においては、チャプタマークやインデクスマークが表すチャプタやインデクスの番号が、mark_dataに記述されているが、チャプタマークやインデクスマークが表すチャプタやインデクスの番号は、mark_dataに記述しなくても、例えば、PlayListMark()にお

けるチャプタマークやインデクスマークの数をカウントすることによって認識することができる。

［ディスク装置の動作説明］

次に、第１図のディスク１０１に、第２５図乃至第２８図で説明したようなデータ（ファイル）が記録されているとして、第１図のディスク装置の動作について説明する。

ディスク１０１がディスクドライブ１０２に挿入されると、その旨を示すメッセージがドライブインターフェース１１４、さらには、第２図Ａおよび第２図Ｂのオペレーティングシステム２０１を経由して、ビデオコンテンツ再生プログラム２１０に伝えられる。ビデオコンテンツ再生プログラム２１０は、ディスク１０１がディスクドライブ１０２に挿入された旨のメッセージをオペレーティングシステム２０１から受信すると、第２９図の再生前処理を開始する。

「再生前処理」

即ち、第２９図は、ビデオコンテンツ再生プログラム２１０が行う再生前処理を説明するフローチャートである。

ここで、以下、フローチャートによって説明するディスク装置の動作または処理は、必ずしもフローチャートとして記載された順序に沿って時系列に行われる必要はなく、並列的あるいは個別に行われることもある。但し、本明細書では、便宜上、ディスク装置の動作または処理を、フローチャートに沿って説明する。

再生前処理では、ビデオコンテンツ再生プログラム２１０は、ステップＳ１０１において、オペレーティングシステム２０１のファイルシステム機能を使用して、ディスク１０１をチェックし、ディスク１０１が、ビデオコンテンツ再生プログラム２１０用の正常なディスクであるか否かを判定する。

ここで、上述したように、ディスク１０１へのアクセス（ファイルの読み出し）は、オペレーティングシステム２０１のファイルシステム機能を使用して行われるが、以下では、その説明は、適宜省略する。

ステップＳ１０１において、ディスク１０１が正常なディスクでないと判定された場合、即ち、例えば、ディスク１０１に採用されているファイルシステムが、オペレーティングシステム２０１の対応していないタイプであった場合や、ルートディレクトリに"VIDEO"ディレクトリが置かれていない場合、ビデオコンテンツ再生プログラム２１０はディスク１０１に対応していないと判断して、ステップＳ１０２に進み、グラフィクス処理モジュール２１９が、エラー処理を行って、再生前処理を終了する。

即ち、グラフィックス処理モジュール２１９は、エラー処理として、ディスク１０１が正常でない旨のエラーメッセージ（のビデオデータ）を生成し、ビデオ出力モジュール２２０から出力させ、これにより、エラーメッセージを表示させる。なお、エラー処理としては、その他、例えば、オーディオ出力モジュール２２１からの警告音の出力や、ディスクドライブ１０２からのディスク１０１の排出等を行うようにすることが可能である。

一方、ステップＳ１０１において、ディスク１０１が正常なディスクであると判定された場合、ステップＳ１０３に進み、ビデオコンテンツ再生プログラム２１０は、コンテンツデータ供給モジュール２１３によって、オペレーティングシステム２０１に対し、ディスク１０１（第４図）の"VIDEO"ディレクトリに置かれている"SCRIPT.DAT"ファイルと、"PLAYLIST.DAT"ファイルとの２つのデータファイルを要求して読み込み、ステップＳ１０４に進む。ステップＳ１０４では、"S

CRIPT.DAT"ファイルが、スクリプト制御モジュール２１１に供給されるとともに、"PLAYLIST.DAT"ファイルが、プレイヤ制御モジュール２１２に供給される。

その後、ステップＳ１０４からＳ１０５乃至Ｓ１０７に進み、プレイヤ制御モジュール２１２は、初期化処理を行う。なお、スクリプト制御モジュール２１１は、プレイヤ制御モジュール２１２の初期化処理が終了するまで待つ。

「プレイヤ制御モジュール２１２の初期化処理」

初期化処理では、ステップＳ１０５において、プレイヤ制御モジュール２１２が、"PLAYLIST.DAT"ファイルの解析を行い、"PLAYLIST.DAT"ファイルの中で使われているクリップ情報ファイルの数とそのファイル名を調査する。

即ち、いまの場合、"PLAYLIST.DAT"ファイルは、第２５図に示したものとなっており、プレイヤ制御モジュール２１２は、第２５図の"PLAYLIST.DAT"ファイルにおいてnumber_of_PlayListsが2になっていることから、１番目のPlayList#0と２番目のPlayList#1との２つのPlayList()が存在することを認識する。さらに、プレイヤ制御モジュール２１２は、第２５図の"PLAYLIST.DAT"ファイルにおける１番目のPlayList#0について、number_of_PlayItemsが2となっていることから、そのPlayList#0には、１番目のPlayItem#0と２番目のPlayItem#1との２つのPlayItem()が存在することを認識する。そして、プレイヤ制御モジュール２１２は、第２５図の"PLAYLIST.DAT"ファイルにおける、PlayList#0に含まれる１番目のPlayItem#0と２番目のPlayItem#1それぞれのClip_Information_file_nameを参照することにより、PlayList#0に含まれる１番目のPlayItem#0のクリップ情報ファイル（のファイル名）が"00001.CLP"であり、２番目のPlayItem#1のクリップ情報ファ

イルが"00002.CLP"であることを認識する。

プレイヤ制御モジュール２１２は、２番目のPlayList#1についても同様にして、そのnumber_of_PlayItemsが1であることから、１つのPlayItem()（PlayItem#0）が存在し、さらに、そのPlayItem#0の中のClip_Information_file_nameから、そのPlayItem#0のクリップ情報ファイルが"00003.CLP"であることを認識する。

その後、ステップＳ１０５からＳ１０６に進み、プレイヤ制御モジュール２１２は、ディスク１０１から、ステップＳ１０５で認識したクリップ情報ファイル、即ち、ディスク１０１の"VIDEO"ディレクトリ内の"CLIP"ディレクトリから３つのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"を読み込む。

ここで、ステップＳ１０６でのクリップ情報ファイルの読み込みは、最初に再生されるPlayList()のPlayItemのクリップ情報ファイルだけで十分であるが、本実施の形態では、上述したように、すべてのPlayList()のPlayItem()のクリップ情報ファイルを先読みしておくこととする。

ステップＳ１０６の処理後は、ステップＳ１０７に進み、プレイヤ制御モジュール２１２は、ステップＳ１０５で認識したクリップ情報ファイルの読み込みに成功したかどうかを判定し、さらに、その読み込んだクリップ情報ファイルに対応するクリップストリームファイルが、ディスク１０１に存在するか否かを判定（チェック）する。即ち、ステップＳ１０７では、クリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"の読み込みに成功し、さらに、そのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"それぞれとファイル名の拡張子のみが異なるクリップストリームファイル"00001.PS"、"00002.PS"、"00003.PS"が、ディスク１０１の"VIDEO"ディレクト

リの下にある"STREAM"ディレクトリに存在するかどうかを判定する。

ステップＳ１０７において、ステップＳ１０５で認識したクリップ情報ファイルの読み込みに失敗したと判定されるか、または、クリップ情報ファイルに対応するクリップストリームファイルが、ディスク１０１に存在しないと判定された場合、即ち、例えば、"PLAYLIST.DAT"ファイルにしたがった再生に必要なクリップ情報ファイルやクリップストリームファイルが、ディスク１０１に記録されていない場合、ビデオコンテンツ再生プログラム２１０は、ディスク１０１に対応していないと判断して、ステップＳ１０２に進み、上述したエラー処理が行われ、再生前処理を終了する。

一方、ステップＳ１０７において、ステップＳ１０５で認識したクリップ情報ファイルの読み込みに成功し、かつ、クリップ情報ファイルに対応するクリップストリームファイルが、ディスク１０１に存在すると判定された場合、プレイヤ制御モジュール２１２は、初期化処理を終了し、ステップＳ１０８に進む。

ステップＳ１０８では、スクリプト制御モジュール２１１が、"SCRIPT.DAT"ファイルの解釈と実行を開始する。

例えば、いま、スクリプト制御モジュール２１１が、"SCRIPT.DAT"ファイルを実行することにより、１番目のPlayList()(PlayList#0)の再生が、プレイヤ制御モジュール２１２に指示されたとすると、第３０図の再生処理が行われる。

「再生処理」

即ち、第３０図は、ビデオコンテンツ再生プログラム２１０が行う再生処理を説明するフローチャートである。

「再生準備処理」

再生処理では、プレイヤ制御モジュール２１２は、まずステップＳ

１２１とＳ１２２において、スクリプト制御モジュール２１１から再生を指示されたPlayList()、即ち、１番目のPlayList()(PlayList#0)の再生準備処理を行う。

即ち、プレイヤ制御モジュール２１２は、ステップＳ１２１において、１番目のPlayList#0に含まれる１番目のPlayItem#0のIN_time（第６図）を確認して、ステップＳ１２２に進み、１番目のPlayList#0に含まれる１番目のPlayItem#0によって再生されるクリップストリームファイル"00001.PS"上の、そのPlayItem#0のIN_timeに相当する、再生を開始する再生開始位置を調査する。

ここで、PlayItem()のIN_time（第６図）が、クリップストリームファイルの先頭を指し示している場合には、クリップストリームファイルの先頭から、プログラムストリームを読み出せば良いが、IN_timeが、クリップストリームファイルの先頭以外を指し示している場合には、IN_timeに対応する位置を探し出し（調査し）、そこからプログラムストリームを読み出す必要がある。

具体的には、第２５図に示した場合、１番目のPlayList#0に含まれる１番目のPlayItem#0のIN_timeは、180,090である。プレイヤ制御モジュール２１２は、１番目のPlayList#0に含まれる１番目のPlayItem#0によって再生されるクリップストリームファイル"00001.CLP"の、第２７図に示したEP_map()から、PlayItem#0のIN_timeである180,090に適した再生開始位置を探し出す。

即ち、プレイヤ制御モジュール２１２は、例えば、EP_map()に記述されたデコード開始可能点を表すPTS_EP_startのうちの、式PTS_EP_start≦IN_timeを満たす最大のPTS_EP_startを、二分探索（バイナリサーチ）等を用いて検索する。ここで、IN_time以下のPTS_EP_startを検索するのは、IN_timeによって表される位置が、デコード開始可

能点であるとは限らないからである。

いまの場合、IN_timeは、上述したように、180,090である。また、１番目のPlayList#0に含まれる１番目のPlayItem#0によって再生されるクリップストリームファイル"00001.CLP"の、第２７図に示したEP_map()においては、式PTS_EP_start≦IN_timeを満たす最大のPTS_EP_startとして、180,090が記述されている。従って、プレイヤ制御モジュール２１２では、第２７図に示したEP_map()から、その180,090となっているPTS_EP_startが検索される。

さらに、プレイヤ制御モジュール２１２では、その検索されたPTS_EP_startに対応するRPN_EP_startである305（セクタ）が読み出され、その305であるRPN_EP_startによって表されるクリップストリームファイル"00001.PS"上の位置が、再生開始位置として決定される。

プレイヤ制御モジュール２１２は、以上のようにして再生開始位置を決定すると、ステップＳ１２２からＳ１２３に進み、タイムコードを表示するように、グラフィクス処理モジュール２１９を制御する。グラフィクス処理モジュール２１９は、プレイヤ制御モジュール２１２の制御にしたがい、タイムコード（のビデオデータ）を生成してビデオ出力モジュール２２０に出力する。これにより、タイムコードの表示が開始される。

ここで、ステップＳ１２３で表示が開始されるタイムコードは、例えば、PlayList()の先頭を00:00:00（時間:分:秒）に換算した値とする。なお、タイムコードとともに、またはタイムコードではなく、チャプタやインデクスの番号を表示するようにしても良い。

「PlaylistMark()の解析処理」

ステップＳ１２３でタイムコードの表示が開始された後は、ステップＳ１２４に進み、プレイヤ制御モジュール２１２は、スクリプト制

御モジュール２１１から再生を指示されたPlayList()、即ち、１番目のPlayList()(PlayList#0) に記述されているPlayListMark() （第７図）を解析する解析処理を行う。

具体的には、プレイヤ制御モジュール２１２は、既に読み込んである"PLAYLIST.DAT"ファイルにおける１番目のPlayList#0の、第２８図上側に示したPlayListMark()において、number_of_PlayList_marksが7になっていることから、そのPlayList#0に含まれるMark()の数が７であることを認識する。

さらに、プレイヤ制御モジュール２１２は、PlayList#0に含まれる、第２８図上側の７つのMark()を解析し、そのref_to_PlayItem_idから、７つのMark()のうちの１番目から４番目までの４つのMark()が、PlayList#0の１番目のPlayItem() （PlayItem#0）に属していることを認識する。

その後、プレイヤ制御モジュール２１２は、PlayList#0の１番目のPlayItem#0に属している４つのMark()のmark_time_stampを取り出し、要素数が4の配列として、デコード制御モジュール２１４に渡す。即ち、これにより、第２８図上側の７つのMark()のうちの１番目から４番目までの４つのMark()それぞれのmark_time_stampである{180,090}、{5,580,090}、 {10,980,090}、 {16,380,090}の４つの時刻が、プレイヤ制御モジュール２１２から、デコード制御モジュール２１４に渡される。このときこれら時刻の属性は「マーク処理」であることも、プレイヤ制御モジュール２１２から、デコード制御モジュール２１４に伝えられる。デコード制御モジュール２１４は、計時部２１４Ａで計時している時刻が、「マーク処理」の属性の時刻に一致したとき、その旨を示すメッセージ、「マーク処理」の属性の時刻に一致した時刻、および「マーク処理」の属性を、プレイヤ制御モジュール２

１２に伝える。

「再生するエレメンタリストリームの決定処理」

次に、ステップＳ１２４からＳ１２５に進み、プレイヤ制御モジュール２１２は、再生するエレメンタリストリームを決定する。

即ち、プレイヤ制御モジュール２１２は、スクリプト制御モジュール２１１から再生を指示されたPlayList()である１番目のPlayList#0における１番目のPlayItem#0（第２５図）のClip_Information_fime_nameにファイル名が記述されている。第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00001.CLP"において、number_of_streamsが4になっていることから、対応するクリップストリームファイル"00001.PS"に、４本のエレメンタリストリームが多重化されていることを認識する。さらに、プレイヤ制御モジュール２１２は、その４本のエレメンタリストリームに対する、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00001.CLP"のStaticInfo()のstream_idと必要なprivate_stream_idを順に調査し、その４本のエレメンタリストリームが、１本のビデオストリーム、１本のATRACオーディオストリーム、および２本の字幕ストリームであることを認識する。即ち、クリップストリームファイル"00001.PS"に多重化されている各属性のエレメンタリストリームの本数が認識される。

なお、クリップストリームファイルに多重化されている各属性のエレメンタリストリームの本数の情報は、再生中でのエレメンタリストリームの切り替え（オーディオ切り替え、字幕切り替え等）に使用される。また、字幕ストリームは、クリップストリームファイル中に存在しない（コンテンツに字幕が含まれない）場合があり、字幕ストリームが存在するかどうかの判断に、「字幕ストリーム」の属性のエレメンタリストリームの本数の情報が使用される。

プレイヤ制御モジュール２１２は、以上のようなStaticInfo()の調査の結果に基づいて、再生するエレメンタリストリームを選択、決定するが、いまの場合、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームの中に、「ビデオストリーム」と「オーディオストリーム」の属性のエレメンタリストリームは、それぞれ１本しかないので、「ビデオストリーム」と「オーディオストリーム」の属性のエレメンタリストリームについては、選択の余地がなく、その１本のビデオストリームとオーディオストリーム（ATRACオーディオストリーム）が、再生するエレメンタリストリームとして決定される。

また、「字幕ストリーム」の属性のエレメンタリストリームについては、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームの中に２本存在するので、その２本の字幕ストリームのうちのいずれか１本の字幕ストリームが、再生するエレメンタリストリームとして選択、決定される。ここでは、例えば、２本の字幕ストリームのうちの、クリップ情報ファイル"00001.CLP"での出現順で最初の字幕ストリームが選択されることとする。

ここで、上述のように、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームの属性と本数を認識するにあたっては、その４本のエレメンタリストリームそれぞれを特定する必要があるが、プレイヤ制御モジュール２１２は、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームの特定を、stream_idと必要なprivate_stream_idによって行う。

即ち、プレイヤ制御モジュール２１２は、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームの

うちの、「ビデオストリーム」の属性のエレメンタリストリームであるビデオストリームを、第２６図Ａおよび第２６図Ｂでクリップ情報ファイル"00001.CLP"について説明したように、0xE0となっているstream_idで特定する。

また、プレイヤ制御モジュール２１２は、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームのうちの、「オーディオストリーム」の属性のエレメンタリストリームであるATRACオーディオストリームを、第２６図Ａおよび第２６図Ｂでクリップ情報ファイル"00001.CLP"について説明したように、0xBDとなっているstream_id、および0x00となっているprivate_stream_idで特定する。

さらに、プレイヤ制御モジュール２１２は、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームにおける「字幕ストリーム」の属性のエレメンタリストリームである２本の字幕ストリームそれぞれを、第２６図Ａおよび第２６図Ｂでクリップ情報ファイル"00001.CLP"について説明したように、0xBDとなっているstream_id、および0x80となっているprivate_stream_idと、0xBDとなっているstream_id、および0x81となっているprivate_stream_idで、それぞれ特定する。

以上のように、クリップストリームファイルに対応するクリップ情報ファイルのメタデータとして記述されるstream_idとprivate_stream_idの組み合わせによって、そのクリップストリームファイルに多重化されているエレメンタリストリームを特定することができる。

ここで、stream_idとprivate_stream_idの組み合わせは、MPEG2-Systemの多重化を拡張するために設けたメカニズムである。このstream_idとprivate_stream_idの組み合わせを、メタデータ（データベース

）において、エレメンタリストリームを特定するために使用することにより、エレメンタリストリームを確実に特定することが可能になる。また、将来、private_stream_idの意味を拡張し、対応するエレメンタリストリームの本数や種類（属性）を増やした場合にも現在のメカニズムをそのまま使用可能であるため、拡張性において勝っている。

即ち、例えば、BD(Blue ray Disc)規格では、データの特定に、MPEG2規格のトランスポートストリーム(Transport Stream)のPID(Packet ID)が用いられるため、MPEG2規格に拘束される。また、例えば、DVD-Video規格では、private_stream_idに類似するsub_stream_idが定義されているが、sub_stream_idは、ストリームの特定のためにデータベース上に記述することができるようにはなっておらず、8本あるいは32本のストリーム情報を記述する固定的な領域に記述することができるにすぎないため（たとえばVI4-49、Table 4.2.1-2（VTS-AST_ATRT）やVI4-52、Table 4.2.1-3（VTS_SPST_ATRT）等参照）、拡張性に乏しい。

これに対して、stream_idとprivate_stream_idの組み合わせは、メタデータが記述される、例えば、第１０図のクリップ情報ファイルClip()において、number_of_streamsで表すことができる数だけ記述することができ、従って、クリップストリームファイルに多重化されているエレメンタリストリームを、その数によらず（但し、number_of_streamsで表すことができる数の範囲）、クリップ情報ファイルClip()に記述されたメタデータとしてのstream_idとprivate_stream_idの組み合わせから特定することが可能となる。

なお、本実施の形態では、stream_idとprivate_stream_idの組み合わせは、第１０図のクリップ情報ファイルにおいて、対応するクリッ

プストリームファイルに多重化されているエレメンタリストリームを特定するのに使用される他、例えば、第７図のPlayListMark()におけるentry_ES_stream_idとentry_ES_private_stream_idの組み合わせとして、Mark()を関連付けるエレメンタリストリームの特定にも使用される。さらに、stream_idとprivate_stream_idの組み合わせは、その他、例えば、第１４図のEP_map()において、デコード可能開始点の情報を記述するエレメンタリストリームの特定にも使用される。

「出力属性の制御処理」

その後、ステップＳ１２５からＳ１２６に進み、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリーム、即ち、ステップＳ１２５で再生すると決定したエレメンタリストリームの出力属性の制御処理を行う。

具体的には、プレイヤ制御モジュール２１２は、まず、再生対象のエレメンタリストリーム、即ち、ステップＳ１２５で再生すると決定したビデオストリーム、ATRACオーディオストリーム、字幕ストリームそれぞれについて、出力属性が記述されるDynamicInfo()（第１３図）の数を表すnumber_of_DynamicInfo（第１０図）を調査する。

ここで、いまの場合、再生対象のビデオストリーム、ATRACオーディオストリーム、字幕ストリームは、クリップストリームファイル"00001.PS"に多重化されているエレメンタリストリームであり、それらのnumber_of_DynamicInfoは、第２６図Ａおよび第２６図Ｂの"00001.CLP"で説明したように、いずれも0になっている。このように、再生対象のエレメンタリストリームのすべてについて、number_of_DynamicInfoが0である場合、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリームの出力属性の制御処理としては、特に処理を行わない。

なお、再生対象のエレメンタリストリームについてのnumber_of_DynamicInfoが0でない場合に、そのエレメンタリストリームの出力属性の制御として行われる処理については、後述する。

「再生開始の準備処理」

ステップＳ１２６の処理後は、ステップＳ１２７に進み、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリームの再生開始の準備処理を行う。

即ち、プレイヤ制御モジュール２１２は、コンテンツデータ供給モジュール２１３に対し、再生対象のエレメンタリストリームが多重化されているクリップストリームファイル"00001.PS"のファイル名と、ステップＳ１２２で決定した再生開始位置であるEP_map()に記述されたRPN_EP_start（=305）を与える。

さらに、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリームが多重化されているクリップストリームファイル"00001.PS"に格納されたプログラムストリームの、バッファ制御モジュール２１５への供給が開始される前に、バッファ制御モジュール２１５を初期化する。

具体的には、バッファ制御モジュール２１５（第３図）では、データ先頭ポインタ記憶部２３１に記憶されるデータ先頭ポインタ、データ書き込みポインタ記憶部２３２に記憶されるデータ書き込みポインタ、ビデオ読み出しポインタ記憶部２４１に記憶されるビデオ読み出しポインタ、オーディオ読み出しポインタ記憶部２５１に記憶されるオーディオ読み出しポインタ、字幕読み出しポインタ記憶部２６２に記憶される字幕読み出しポインタに、同じ値が代入される。

これにより、データ先頭ポインタ記憶部２３１に記憶されたデータ先頭ポインタと、データ書き込みポインタ２３２に記憶されたデータ

書き込みポインタとは、バッファ制御モジュール２１５のバッファ２１５Ａの同一の位置を指す。これは、バッファ２１５Ａに、有効なデータが蓄積されていない状態を表す。

さらに、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリームを識別（特定）するための識別情報としてのstream_id、さらには、必要に応じて、private_stream_idを、バッファ制御モジュール２１５に供給する。

即ち、上述したように、再生対象のエレメンタリストリームのうちの、「ビデオストリーム」の属性のビデオストリームは、0xE0となっているstream_idによって特定され、「オーディオストリーム」の属性のATRACオーディオストリームは、0xBDとなっているstream_id、および0x00となっているprivate_stream_idによって特定され、「字幕ストリーム」の属性の字幕ストリームは、0xBDとなっているstream_id、および0x80となっているprivate_stream_idによって特定される。プレイヤ制御モジュール２１２は、これらのstream_idとprivate_stream_idを、バッファ制御モジュール２１５に供給する。

バッファ制御モジュール２１５（第３図）では、ビデオ読み出し機能部２３３が、プレイヤ制御モジュール２１２からの、ビデオストリームについての0xE0となっているstream_idを、stream_idレジスタ２４２に記憶させる。また、オーディオ読み出し機能部２３４が、プレイヤ制御モジュール２１２からの、0xBDとなっているstream_idと、0x00となっているprivate_stream_idを、stream_idレジスタ２５２とprivate_stream_idレジスタ２５３に、それぞれ記憶させる。さらに、字幕読み出し機能部２３５が、プレイヤ制御モジュール２１２からの、0xBDとなっているstream_idと、0x80となっているprivate_stream_idを、stream_idレジスタ２６３とprivate_stream_idレジスタ２６４

に、それぞれ記憶させる。

なお、プレイヤ制御モジュール２１２は、バッファ制御モジュール２１５に供給した再生対象のエレメンタリストリームのstream_idとprivate_stream_idを、今後の処理のために記憶する。プレイヤ制御モジュール２１２は、これらのstream_idやprivate_stream_idを、後述するストリーム切り替えを要求するメッセージの発生時や、後述するマーク処理において現在再生中のストリームを特定するために使用する。

プレイヤ制御モジュール２１２は、バッファ制御モジュール２１５（第３図）の初期化として、さらに、再生対象のエレメンタリストリームが多重化されているクリップストリームファイルに応じた値の字幕読み出し機能フラグを、字幕読み出し機能フラグ記憶部２６１にセットする。

即ち、いまの場合、再生対象のエレメンタリストリームが多重化されているクリップストリームファイル"00001.PS"には、字幕ストリームが含まれるため、字幕読み出し機能部２３５を機能させるために、値が１の字幕読み出し機能フラグが、字幕読み出し機能フラグ記憶部２６１にセットされる。なお、再生対象のエレメンタリストリームが多重化されているクリップストリームファイルに字幕ストリームが含まれていない場合、字幕読み出し機能フラグ記憶部２６１には、値が０の字幕読み出し機能フラグがセットされる。この場合、字幕読み出し機能部２３５は機能しない（特に処理を行わない）。

また、プレイヤ制御モジュール２１２は、スクリプト制御モジュール２１１から再生を指示された１番目のPlayList#0に含まれる１番目のPlayItem#0（第２５図）のIN_timeである180,090と、OUT_timeである27,180,090とを、デコード制御モジュール２１４に対して与える。

デコード制御モジュール２１４では、IN_timeは、PlayItem()によって再生されるクリップのデコード開始の制御に、OUT_timeは、そのクリップのデコード終了、さらには、後述するPlayItem乗り換えの制御に、それぞれ使用される。

さらに、プレイヤ制御モジュール２１２は、グラフィクス処理モジュール２１９に対する字幕ストリームの表示方式の指示を初期化する。即ち、プレイヤ制御モジュール２１２は、字幕ストリームの表示方式を、例えば、デフォルトの表示方式とするように、グラフィクス処理モジュール２１９を制御する。

「データ読み込み開始」

その後、ステップＳ１２７からＳ１２８に進み、プレイヤ制御モジュール２１２は、コンテンツデータ供給モジュール２１３を制御し、これにより、コンテンツデータ供給モジュール２１３は、オペレーティングシステム２０１の機能を使用して、再生対象のエレメンタリストリームが多重化されたプログラムストリームが格納されたクリップストリームファイルを読み出す。すなわち、コンテンツデータ供給モジュール２１３は、ディスク１０１（第４図）の"VIDEO"ディレクトリの下にある"STREAM"ディレクトリのクリップストリームファイル"00001.PS"を指定し、さらに、ステップＳ１２２で決定された再生開始位置である305セクタを指定して、オペレーティングシステム２０１に対してファイル読み出しを要求する。また、コンテンツデータ供給モジュール２１３は、ディスク１０１から読み出したデータを、バッファ制御モジュール２１５に供給するように指定する。

これにより、ディスク１０１からの、クリップストリームファイル"00001.PS"に格納されたプログラムストリームの読み出しが開始され、そのプログラムストリームは、バッファ制御モジュール２１５に供

給される。

バッファ制御モジュール２１５（第３図）は、ディスク１０１から読み出されて供給されたプログラムストリームを、バッファ２１５Ａのデータ書き込みポインタ記憶部２３２のデータ書き込みポインタが指す位置に書き込み、書き込んだデータのサイズだけデータ書き込みポインタをインクリメントする。

ここで、以下、特に断らない限り、コンテンツデータ供給モジュール２１３は、バッファ制御モジュール２１５のバッファ２１５Ａに空きがあれば、ディスク１０１からデータを読み出し、バッファ制御モジュール２１５のバッファ２１５Ａに供給して記憶させることとする。従って、バッファ２１５Ａには、常時、十分なデータが蓄積されているとする。

「デコーダ制御開始」

以上のようにして、ディスク１０１からのデータの読み出しが開始され、そのデータがバッファ制御モジュール２１５のバッファ２１５Ａに蓄積され始めると、ステップＳ１２８からＳ１２９に進み、デコード制御モジュール２１４は、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、字幕デコーダ制御モジュール２１８を制御し、デコード動作の前段階として、バッファ２１５Ａからのデータの読み出しを開始させる。

即ち、これにより、ビデオデコーダ制御モジュール２１６は、バッファ制御モジュール２１５（第３図）のビデオ読み出し機能部２３３にデータを要求し、その要求に応じてバッファ制御モジュール２１５から渡される、バッファ２１５Ａに記憶された１つのビデオアクセスユニット、そのビデオアクセスユニットに付加されているPTSとDTS（以下、適宜、タイムスタンプという）、およびデコード開始可能点の

直前に配置されているprivate_stream_2のPES_packet()に記述された情報（以下、適宜、付加情報ともいう）であるpic_struct_copyや、au_ref_flag、AU_lengthなどを得る。なお、タイムスタンプは、ビデオデコーダ制御モジュール２１６がビデオアクセスユニットを得る毎に、ビデオデコーダ制御モジュール２１６からデコード制御モジュール２１４に渡される。

一方、オーディオデコーダ制御モジュール２１７も、バッファ制御モジュール２１５（第３図）のオーディオ読み出し機能部２３４にデータを要求し、その要求に応じてバッファ制御モジュール２１５から渡される、バッファ２１５Ａに記憶された１つの(ATRAC)オーディオアクセスユニットと、そのオーディオアクセスユニットに付加されているタイムスタンプ（PTS, DTS）を得る。なお、タイムスタンプは、オーディオデコーダ制御モジュール２１７がオーディオアクセスユニットを得る毎に、オーディオデコーダ制御モジュール２１７からデコード制御モジュール２１４に渡される。

さらに、字幕デコーダ制御モジュール２１８は、バッファ制御モジュール２１５（第３図）の字幕読み出し機能部２３５にデータを要求し、その要求に応じてバッファ制御モジュール２１５から渡される、バッファ２１５Ａに記憶された１つの字幕アクセスユニットと、その字幕アクセスユニットに付加されているタイムスタンプを得る。なお、タイムスタンプは、字幕デコーダ制御モジュール２１８が字幕アクセスユニットを得る毎に、字幕デコーダ制御モジュール２１８からデコード制御モジュール２１４に渡される。また、再生対象のエレメンタリストリームに、字幕ストリームが存在しない場合や、バッファ２１５Ａに、字幕アクセスユニットが記憶されていない場合は、バッファ制御モジュール２１５から字幕デコーダ制御モジュール２１８には

、データは渡されない。

ここで、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、および字幕デコーダ制御モジュール２１８は、バッファ制御モジュール２１５に対してデータを要求する毎に、そのデータの要求に対する結果を、デコード制御モジュール２１４に渡す。

また、バッファ制御モジュール２１５から、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、および字幕デコーダ制御モジュール２１８に対してデータが渡されるときの、そのデータのバッファ２１５Ａからの読み出しの詳細については、後述する。

「デコード開始」

以上のように、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、字幕デコーダ制御モジュール２１８が、バッファ制御モジュール２１５のバッファ２１５Ａからデータを読み出し始めると、ステップＳ１２９からＳ１３０に進み、そのデータのデコードが開始される。

即ち、デコード制御モジュール２１４は、ステップＳ１２７でプレイヤ制御モジュール２１２から与えられた、PlayList#0に含まれる１番目のPlayItem#0のIN_timeである180,090、さらには、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、字幕デコーダ制御モジュール２１８からステップＳ１２９で説明したように渡されるタイムスタンプに基づき、同期を確保するために必要であればタイミングをずらして、デコード開始を、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、および字幕デコーダ制御モジュール２１８に指令する。

ここで、同期を確保するためのタイミングをずらしたデコード開始の指令の方法は、例えば、特許第3496725号に記載されており、簡単には、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、字幕デコーダ制御モジュール２１８それぞれから渡されたタイムスタンプのうちの最小値を、計時部２１４Ａによって計時される時刻の初期値として設定して時刻の計時を開始し、計時部２１４Ａによって計時される時刻と、タイムスタンプとが一致した時点で、デコード開始を指令する方法がある。

ビデオデコーダ制御モジュール２１６は、デコード制御モジュール２１４からのデコード開始の指示を受け、その指示に応じて、バッファ制御モジュール２１５（第３図）のビデオ読み出し機能部２３３から得た１つのビデオアクセスユニットを、ビデオデコーダ１１６（第１図）に渡してデコードさせる。さらに、ビデオデコーダ制御モジュール２１６は、ビデオデコーダ１１６によるデコードの結果得られるビデオデータを、グラフィクス処理モジュール２１９に供給する。

以後、ビデオデコーダ制御モジュール２１６は、バッファ制御モジュール２１５のビデオ読み出し機能部２３３から得られる１ずつのビデオアクセスユニットを、ビデオデコーダ１１６で順次デコードし、そのデコードの結果得られるビデオデータを、グラフィクス処理モジュール２１９に供給していく。

一方、オーディオデコーダ制御モジュール２１７も、デコード制御モジュール２１４からのデコード開始の指示を受け、その指示に応じて、バッファ制御モジュール２１５（第３図）のオーディオ読み出し機能部２３４から得た１つのオーディオアクセスユニットを、オーディオデコーダ１１７（第１図）に渡してデコードさせる。さらに、オーディオデコーダ制御モジュール２１７は、オーディオデコーダ１１

７によるデコードの結果得られるオーディオデータを、オーディオ出力モジュール２２１に供給する。

以後、オーディオデコーダ制御モジュール２１７は、バッファ制御モジュール２１５のオーディオ読み出し機能部２３４から得られる１ずつのオーディオアクセスユニットを、オーディオデコーダ１１７で順次デコードし、そのデコードの結果得られるオーディオデータを、オーディオ出力モジュール２２１に供給していく。

また、字幕デコーダ制御モジュール２１８も、デコード制御モジュール２１４からのデコード開始の指示を受け、その指示に応じて、バッファ制御モジュール２１５（第３図）の字幕読み出し機能部２３５から得た１つの字幕アクセスユニットを、内部に持つ字幕デコードソフトウェアでデコードし、そのデコードの結果得られる字幕データ（字幕の画像データ）を、グラフィクス処理モジュール２１９に供給する。

以後、字幕デコーダ制御モジュール２１８は、バッファ制御モジュール２１５の字幕読み出し機能部２３５から得られる１ずつの字幕アクセスユニットを、内部に持つ字幕デコードソフトウェアで順次デコードし、そのデコードの結果得られる字幕データを、グラフィクス処理モジュール２１９に供給していく。

「グラフィクス処理」

その後、ステップＳ１３０からＳ１３１に進み、グラフィクス処理モジュール２１９は、上述したようにして、ビデオデコーダ制御モジュール２１６から供給されるビデオデータ、さらには、必要に応じて、字幕デコーダ制御モジュール２１８から供給される字幕データを対象に、グラフィクス処理を行う。

即ち、グラフィクス処理モジュール２１９は、まず字幕デコーダ制

御モジュール２１８からの字幕データを、プレイヤ制御モジュール２１２からの表示方式の指示に従って、拡大や縮小等する字幕処理を行う。プレイヤ制御モジュール２１２から、表示方式の指示がない場合、またはデフォルトの表示方式の指示があった場合、グラフィクス処理モジュール２１９は、字幕デコーダ制御モジュール２１８からの字幕データを、そのまま保存する。

さらに、グラフィクス処理モジュール２１９は、ビデオデコーダ制御モジュール２１６からのビデオデータと、字幕デコーダ制御モジュール２１８からの字幕データ、または字幕処理後の字幕データとを加算し、ビデオデコーダ制御モジュール２１６からのビデオデータに字幕データがオーバーレイされた出力ビデオデータを得て、ビデオ出力モジュール２２０に供給する。

なお、グラフィクス処理モジュール２１９は、スクリプト制御モジュール２１１やプレイヤ制御モジュール２１２から、例えば、メニューや、メッセージ、タイムコード、チャプタまたはインデクスの番号等の情報の表示の指示を受けた場合は、その情報を生成し、出力ビデオデータにオーバーレイして、ビデオ出力モジュール２２０に供給する。

「出力処理」

ステップＳ１３１の処理後は、ステップＳ１３２に進み、ビデオ出力モジュール２２０は、ステップＳ１３１で説明したようにしてグラフィクス処理モジュール２１９から供給される出力ビデオデータを、FIFO２２０Ａに順次記憶させ、そのFIFO２２０Ａに記憶された出力ビデオデータを、あらかじめ決められた出力レートで順次出力する。

ビデオ出力モジュール２２０は、FIFO２２０Ａの記憶容量（残量）に余裕がある限り、グラフィクス処理モジュール２１９からの出力ビ

デオデータを受け入れるが、余裕がない場合には、出力ビデオデータの受け入れの停止を、グラフィクス処理モジュール２１９に要求する。これにより、グラフィクス処理モジュール２１９は、処理を停止するとともに、処理の停止を、ビデオデコーダ制御モジュール２１６および字幕デコーダ制御モジュール２１８に要求する。これにより、ビデオデコーダ制御モジュール２１６および字幕デコーダ制御モジュール２１８が処理を停止する。

ビデオ出力モジュール２２０は、出力ビデオデータの受け入れの停止を、グラフィクス処理モジュール２１９に要求した後に、FIFO２２０Ａからの出力ビデオデータの出力が進み、FIFO２２０Ａに余裕ができた時点で、出力ビデオデータの受け入れを、グラフィクス処理モジュール２１９に要求する。この要求は、出力ビデオデータの受け入れの停止の要求と同様に、グラフィクス処理モジュール２１９から、ビデオデコーダ制御モジュール２１６および字幕デコーダ制御モジュール２１８に伝えられる。これにより、グラフィクス処理モジュール２１９、さらには、ビデオデコーダ制御モジュール２１６および字幕デコーダ制御モジュール２１８は、停止していた処理を再開する。

一方、オーディオ出力モジュール２２１も、ステップＳ１３０で説明したようにしてオーディオデコーダ制御モジュール２１７から供給されるオーディオデータを、FIFO２２１Ａに順次記憶させ、そのFIFO２２１Ａに記憶されたオーディオデータを、あらかじめ決められた出力レート（サンプリング周波数）で順次出力する。

オーディオ出力モジュール２２１は、FIFO２２１Ａの記憶容量（残量）に余裕がある限り、オーディオデコーダ制御モジュール２１７からのオーディオデータを受け入れるが、余裕がない場合には、オーディオデータの受け入れの停止を、オーディオデコーダ制御モジュール

２１７に要求する。これにより、オーディオデコーダ制御モジュール２１７は、処理を停止する。

オーディオ出力モジュール２２１は、オーディオデータの受け入れの停止を、オーディオデコーダ制御モジュール２１７に要求した後に、FIFO２２１Ａからのオーディオデータの出力が進み、FIFO２２１Ａに余裕ができた時点で、オーディオデータの受け入れを、オーディオデコーダ制御モジュール２１７に要求する。これにより、オーディオデコーダ制御モジュール２１７は、停止していた処理を再開する。

以上のようにして、ビデオ出力モジュール２２０およびオーディオ出力モジュール２２１からデータが出力されるにつれて、エレメンタリストリームのデコードが行われていく。

第１図のディスク装置がディスク１０１を再生するときの全体の処理または動作の流れは、第２９図および第３０図で説明したとおりであるが、以下、ディスク装置においてディスク１０１の再生が行われているときの、その他の処理または動作について説明する。

[PlayItem乗り換え]

第２９図および第３０図で説明したようにして、第２５図における１番目のPlayList#0の１番目のPlayItem#0の再生が始まるが、PlayList#0によれば、その１番目のPlayItem#0の再生が終了すると、２番目のPlayItem#1の再生が開始される。即ち、PlayItem#0からPlayItem#1にPlayItemを乗り換えるPlayItem乗り換えが行われる。

そこで、第３１図のフローチャートを参照して、このPlayItem乗り換えの処理について説明する。

第２９図および第３０図で説明したようにして、第２５図におけるPlayList#0の１番目のPlayItem#0（のクリップ）の再生が開始されると、デコード制御モジュール２１４（第２図Ａおよび第２図Ｂ）は、

その１番目のPlayItem#0の再生が行われている間、内蔵する計時部２１４Ａが計時している時刻を確認し続けている。

「PlayItem#0の再生終了」

そして、デコード制御モジュール２１４は、計時部２１４Ａが計時している時刻が、第３０図のステップ１２７でプレイヤ制御モジュール２１２から与えられた１番目のPlayItem#0のOUT_timeである27,180,090（第２５図）に等しくなると、ステップＳ１５１において、デコード中断制御を行い、PlayItem#0の再生を終了する。

即ち、デコード制御モジュール２１４は、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、字幕デコーダ制御モジュール２１８を操作して、デコード動作を停止させる。さらに、デコード制御モジュール２１４は、ビデオ出力モジュール２２０を制御し、現在出力中の出力ビデオデータを引き続き出力させる。

また、デコード制御モジュール２１４は、１番目のPlayItem#0の再生が終了した旨のメッセージを、プレイヤ制御モジュール２１２に伝える。

「PlayItem#1の再生開始」

プレイヤ制御モジュール２１２は、上述したように、第２９図のステップＳ１０５で、１番目のPlayList #0に、１番目のPlayItem#0と２番目のPlayItem#1とが存在することを認識しており、デコード制御モジュール２１４から、１番目のPlayItem#0の再生が終了した旨のメッセージが伝えられると、ステップＳ１５１からＳ１５２に進み、２番目のPlayItem#1の再生を、上述した１番目のPlayItem#0における場合と同様にして開始する。

即ち、２番目のPlayItem#1の再生手順を概説すれば、まず、プレイ

ヤ制御モジュール２１２は、第３０図のステップＳ１２２における場合と同様にして、２番目のPlayItem#1について、EP_map()に記述されたRPN_EP_startのうちのいずれかを、再生開始位置として決定する。

さらに、プレイヤ制御モジュール２１２では、第３０図のステップＳ１２４で説明したようにして、２番目のPlayItem#1に属するMark()の認識や、第３０図のステップＳ１２５で説明したようにして、PlayItem#1によって再生されるクリップストリームファイル"00002.PS"に多重化されている各属性のエレメンタリストリームの本数の認識、さらには、再生する（再生対象の）エレメンタリストリームの決定等が行われる。

そして、プレイヤ制御モジュール２１２は、第３０図のステップＳ１２７における場合と同様の処理を行う。

即ち、プレイヤ制御モジュール２１２は、再生開始位置として決定したEP_map()のRPN_EP_startと、再生対象のエレメンタリストリームが多重化されているクリップストリームファイルのファイル名、即ち、いまの場合、２番目のPlayItem#1（第２５図）のClip_Information_file_nameに記述された"00002.CLP"に対応するクリップストリームファイル"00002.PS"のファイル名を、コンテンツデータ供給モジュール２１３に対して与える。

さらに、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリームが多重化されているクリップストリームファイル"00002.PS"に格納されたプログラムストリームの、バッファ制御モジュール２１５への供給が開始される前に、バッファ制御モジュール２１５を初期化する。

即ち、これにより、バッファ制御モジュール２１５（第３図）において、データ先頭ポインタ記憶部２３１に記憶されるデータ先頭ポイ

ンタ、データ書き込みポインタ記憶部２３２に記憶されるデータ書き込みポインタ、ビデオ読み出しポインタ記憶部２４１に記憶されるビデオ読み出しポインタ、オーディオ読み出しポインタ記憶部２５１に記憶されるオーディオ読み出しポインタ、字幕読み出しポインタ記憶部２６２に記憶される字幕読み出しポインタに、同じ値が代入される。

さらに、プレイヤ制御モジュール２１２は、再生対象のエレメンタリストリームを識別するための識別情報としてのstream_id、さらには、必要に応じて、private_stream_idを、バッファ制御モジュール２１５に供給する。

バッファ制御モジュール２１５（第３図）では、ビデオ読み出し機能部２３３が、プレイヤ制御モジュール２１２からの、再生対象のエレメンタリストリームのうちのビデオストリームについてのstream_idを、stream_idレジスタ２４２に記憶させる。また、オーディオ読み出し機能部２３４が、プレイヤ制御モジュール２１２からの、再生対象のエレメンタリストリームのうちのオーディオストリームのstream_idとprivate_stream_idを、stream_idレジスタ２５２とprivate_stream_idレジスタ２５３に、それぞれ記憶させる。

さらに、いま再生対象となっているエレメンタリストリームが多重化されているクリップストリームファイル"00002.PS"には、字幕ストリームが含まれるため、プレイヤ制御モジュール２１２から字幕読み出し機能部２３５には、再生対象のエレメンタリストリームのうちの字幕ストリームのstream_idとprivate_stream_idが供給され、字幕読み出し機能部２３５は、そのstream_idとprivate_stream_idを、stream_idレジスタ２６３とprivate_stream_idレジスタ２６４に、それぞれ記憶させる。

そして、プレイヤ制御モジュール２１２は、バッファ制御モジュール２１５（第３図）の初期化として、さらに、再生対象のエレメンタリストリームが多重化されているクリップストリームファイルに応じた値の字幕読み出し機能フラグを、字幕読み出し機能フラグ記憶部２６１にセットする。

即ち、いまの場合、再生対象のエレメンタリストリームが多重化されているクリップストリームファイル"00002.PS"には、字幕ストリームが含まれるため、字幕読み出し機能部２３５を機能させるために、値が１の字幕読み出し機能フラグが、字幕読み出し機能フラグ記憶部２６１にセットされる。

また、プレイヤ制御モジュール２１２は、再生しようとしている２番目のPlayItem#1（第２５図）のIN_timeである90,000と、OUT_timeである27,090,000とを、デコード制御モジュール２１４に対して与える。

さらに、プレイヤ制御モジュール２１２は、グラフィクス処理モジュール２１９に対する字幕ストリームの表示方式の指示を初期化する。即ち、プレイヤ制御モジュール２１２は、字幕ストリームの表示方式をデフォルトの表示方式とするように、グラフィクス処理モジュール２１９を制御する。

なお、再生対象の字幕ストリームについて、configurable_flag（第１２図）が、表示方式の変更を許可する1になっている場合には、プレイヤ制御モジュール２１２からグラフィクス処理モジュール２１９に対する字幕ストリームの表示方式の指示は、現在の表示方式のままとするようにしても良い。

以下、２番目のPlayItem#1の再生は、１番目のPlayItem#0の再生と同様にして行われていく。そして、デコード制御モジュール２１４は

、その２番目のPlayItem#1の再生が行われている間、内蔵する計時部２１４Ａが計時している時刻を確認し続けており、計時部２１４Ａが計時している時刻が、ステップＳ１５２（第３１図）でプレイヤ制御モジュール２１２から与えられた２番目のPlayItem#1のOUT_timeである27,090,000（第２５図）に等しくなると、ステップＳ１５１における場合と同様のデコード中断制御を行い、PlayItem#1の再生を終了する。

［タイムコードの表示］

次に、上述したように、第３０図のステップＳ１２３において、タイムコードの表示が開始されるが、このタイムコードの表示は、順次更新されていく。

そこで、第３２図のフローチャートを参照して、タイムコードの表示の処理について説明する。

デコード制御モジュール２１４（第２図Ａおよび第２図Ｂ）は、その内蔵する計時部２１４Ａによって1秒が計時されると、ステップＳ１７１において、１秒が経過した旨のメッセージとともに、その計時部２１４Ａによって計時されている現在時刻を、プレイヤ制御モジュール２１２に供給し、ステップＳ１７２に進む。ステップＳ１７２では、プレイヤ制御モジュール２１２は、デコード制御モジュール２１４からのメッセージと現在時刻を受信し、その現在時刻を、タイムコードに換算して、ステップＳ１７３に進む。

ステップＳ１７３では、プレイヤ制御モジュール２１２は、ステップＳ１７２で得たタイムコードを表示するように、グラフィクス処理モジュール２１９を制御し、ステップＳ１７１に戻る。

これにより、タイムコードは、１秒ごとに更新される。なお、タイムコードの更新の間隔は、１秒に限定されるものではない。

［ストリーム切り替え］

次に、第２５図で説明した１番目のPlayList#0を構成する１番目のPlayItem#0によって再生されるクリップストリームファイル"00001.PS"や、２番目のPlayItem#1によって再生されるクリップストリームファイル"00002.PS"には、第２６図Ａおよび第２６図Ｂで説明したように、２本の字幕ストリームが多重化されている。

このように、クリップストリームファイルに、複数の、同一の属性のエレメンタリストリームが多重化されている場合においては、再生対象のエレメンタリストリームを、その複数の、同一の属性のエレメンタリストリームのうちの１つから、他の１つに切り替えるストリーム切り替えを行うことができる。

そこで、第３３図のフローチャートを参照して、ストリーム切り替えの処理について説明する。

ストリーム切り替えの要求は、例えば、"SCRIPT.DAT"ファイル（第４図）に、ストリーム切り替えの指示がスクリプトプログラムとして記述されている場合に、スクリプト制御モジュール２１１が、そのスクリプトプログラムを実行することによって、あるいは、ユーザがリモコンを操作することによって、プレイヤ制御モジュール２１２に与えられる。

即ち、スクリプト制御モジュール２１１は、ストリーム切り替えの指示が記述されているスクリプトプログラムを実行すると、ストリーム切り替えを要求するメッセージを、プレイヤ制御モジュール２１２に供給する。また、入力インターフェース１１５は、ユーザがリモコンを操作することによって、リモコンから、ストリーム切り替えを指示する信号を受信すると、ストリーム切り替えを要求するメッセージを、プレイヤ制御モジュール２１２に供給する。

例えば、いま、プレイヤ制御モジュール２１２に対して、字幕ストリームの切り替えを要求するメッセージである字幕ストリーム切り替えのメッセージが供給されたとすると、プレイヤ制御モジュール２１２は、ステップＳ１９１において、第３０図のステップＳ１２５で行われた再生対象のエレメンタリストリームの決定のときに認識した字幕ストリームの本数をチェックする。

プレイヤ制御モジュール２１２は、字幕ストリームの本数をチェックした結果、その本数が１本以下である場合、字幕ストリーム切り替えのメッセージを無視し、従って、以降のステップＳ１９２乃至Ｓ１９４の処理は行われない。

一方、字幕ストリームの本数が２本以上である場合、ステップＳ１９２乃至Ｓ１９４に順次進み、再生する字幕ストリームが、現在再生されている字幕ストリームから、他の字幕ストリームに切り替えられる。

即ち、ステップＳ１９２において、プレイヤ制御モジュール２１２は、現在再生中の字幕ストリームを、クリップ情報ファイル上で特定する。具体的には、例えば、いま、第２５図で説明した１番目のPlayList#0を構成する２番目のPlayItem#1によって、クリップストリームファイル"00002.PS"に多重化された、stream_idが0xBDで、private_stream_idが0x80の字幕ストリームが再生されていることとすると、ステップＳ１９２では、現在再生中の字幕ストリームが、クリップストリームファイル"00002.PS"に多重化された２本の字幕ストリームのうちの、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00002.CLP"上で３本目の字幕ストリームであるstream#2であることが特定される。

そして、ステップＳ１９３に進み、プレイヤ制御モジュール２１２

は、ステップＳ１９２で特定した字幕ストリームの、クリップ情報ファイル上で次の字幕ストリームを、次に再生する字幕ストリームとして認識（特定）する。第２６図Ａおよび第２６図Ｂでは、クリップ情報ファイル"00002.CLP"上で、３本目の字幕ストリームstream#2の次の字幕ストリームは、４本目の字幕ストリームstream#3であるから、ステップＳ１９３では、この４本目の字幕ストリームstream#3が、次に再生する字幕ストリームとして認識される。

なお、現在再生中の字幕ストリームが、クリップストリームファイル"00002.PS"に多重化された２本の字幕ストリームのうちの、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00002.CLP"上で４本目の字幕ストリームであるstream#3であることが特定された場合は、例えば、３本目の字幕ストリームstream#2が、次に再生する字幕ストリームとして認識される。

その後、ステップＳ１９４に進み、プレイヤ制御モジュール２１２は、ステップＳ１９３で認識した次に再生する字幕ストリームのstream_idとprivate_stream_idを、バッファ制御モジュール２１５（第３図）の字幕読み出し機能部２３５に対して与え、そのstream_idとprivate_stream_idを、次回からの、字幕アクセスユニットのバッファ２１５Ａからの読み出しから使用するように指示する。

バッファ制御モジュール２１５（第３図）の字幕読み出し機能部２３５では、ステップＳ１９４でプレイヤ制御モジュール２１２から与えられるstream_idとprivate_stream_idを、stream_idレジスタ２６３とprivate_stream_idレジスタ２６４に、それぞれ新たにセットし、次回以降のバッファ２１５Ａからの読み出しは、そのstream_idレジスタ２６３とprivate_stream_idレジスタ２６４にそれぞれ新たにセットされたstream_idとprivate_stream_idによって特定される字幕

アクセスユニットを対象として行われる。

以上のようにして、再生する字幕ストリームが、現在再生されている字幕ストリームから、他の字幕ストリームに切り替えられる。

［バッファ制御モジュール２１５の処理］

次に、第３４図乃至第３８図を参照して、バッファ制御モジュール２１５（第３図）の処理、即ち、バッファ２１５Ａへのデータの書き込みと、バッファ２１５Ａからのデータの読み出しについて説明する。

バッファ制御モジュール２１５は、第３図で説明したように、バッファ２１５Ａに対するデータの読み書きを行うための５つのポインタを有している。

即ち、第３４図および第３５図に示すように、バッファ制御モジュール２１５は、データ先頭ポインタ記憶部２３１に記憶されるデータ先頭ポインタ、データ書き込みポインタ記憶部２３２に記憶されるデータ書き込みポインタ、ビデオ読み出しポインタ記憶部２４１に記憶されるビデオ読み出しポインタ、オーディオ読み出しポインタ記憶部２５１に記憶されるオーディオ読み出しポインタ、および字幕読み出しポインタ記憶部２６２に記憶される字幕読み出しポインタを有している。

なお、第３４図および第３５図では、第３図におけるビデオ読み出し機能部２３３のstream_idレジスタ２４２およびau_information()レジスタ２４３、オーディオ読み出し機能部２３４のstream_idレジスタ２５２およびprivate_stream_idレジスタ２５３、並びに字幕読み出し機能部２３５の字幕読み出し機能フラグ記憶部２６１、stream_idレジスタ２６３、およびprivate_stream_idレジスタ２６４の図示は、省略してある。

データ先頭ポインタ記憶部２３１に記憶されたデータ先頭ポインタは、バッファ２１５Ａに残る最も古いデータ（読み出す必要があるデータであって、まだ読み出されていないデータのうちの最も古いデータ）の位置を表す。データ書き込みポインタ記憶部２３２に記憶されたデータ書き込みポインタは、バッファ２１５Ａへのデータの書き込みの位置を示し、この位置は、バッファ２１５Ａで最も新しいデータが書き込まれる位置である。

ビデオ読み出しポインタ記憶部２４１に記憶されたビデオ読み出しポインタは、バッファ２１５Ａから読み出すビデオストリームの位置を表す。また、オーディオ読み出しポインタ記憶部２５１に記憶されたオーディオ読み出しポインタは、バッファ２１５Ａから読み出すオーディオストリームの位置を表し、字幕読み出しポインタ記憶部２６２に記憶された字幕読み出しポインタは、バッファ２１５Ａから読み出す字幕ストリームの位置を表す。

なお、第３図で説明したように、データ先頭ポインタ、データ書き込みポインタ、ビデオ読み出しポインタ、オーディオ読み出しポインタ、および字幕読み出しポインタは、いずれも、バッファ２１５Ａを右回りに移動する。

さらに、本実施の形態では、データ先頭ポインタは、第３５図に示すように、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタのうちの、最も古いデータの位置を指しているものと同一の位置を指すように、常時更新されるものとする。ここで、第３５図では、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタのうちの、オーディオ読み出しポインタが、最も古いデータの位置を指しており、データ先頭ポインタは、そのオーディオ読み出しポインタと一致している。

以上のようなデータ先頭ポインタ、データ書き込みポインタ、ビデオ読み出しポインタ、オーディオ読み出しポインタ、および字幕読み出しポインタを有するバッファ制御モジュール２１５では、データ書き込みポインタは、ディスク１０１から新たなデータが読み出され、バッファ２１５Ａに書き込まれると、その書き込まれた新たなデータの直後の位置を指すように、右回りに更新される。

さらに、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタは、バッファ２１５Ａから、ビデオストリーム、オーディオストリーム、または字幕ストリームが読み出されると、その読み出し量に応じた分だけ、それぞれ、右回りに更新される。ここで読み出し量に応じた分とは、実際に読み出したビデオ、オーディオ、字幕のデータに対応する部分と、読み出したデータの間に含まれており、読み出しの際には読み飛ばしを行った、他のストリームのデータの部分をあわせたものとなる。

また、データ先頭ポインタは、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタが更新されると、そのビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタのうちの、最も古いデータの位置を指しているものと同一の位置を指すように更新される。

ここで、バッファ制御モジュール２１５は、バッファ２１５Ａへのデータの書き込みについては、データ書き込みポインタがデータ先頭ポインタを追い越さないように、バッファ２１５Ａへのデータの書き込みを制御する。

即ち、データ書き込みポインタによるデータ先頭ポインタの追い越しが発生しない限り、バッファ制御モジュール２１５では、ディスク１０１から読み出されたデータが、データ書き込みポインタが指すバ

ッファ２１５Ａの位置に書き込まれ、データ書き込みポインタが更新されていく。一方、データ書き込みポインタによるデータ先頭ポインタの追い越しが発生しそうになると、バッファ制御モジュール２１５では、コンテンツデータ供給モジュール２１３に対して、ディスク１０１からのデータの読み出しの停止（中断）が要求され、さらに、バッファ２１５Ａへのデータの書き込みが停止される。これにより、バッファ２１５Ａのオーバーフローを防止することができる。

以上のように、ディスク１０１から読み出されたデータの、バッファ２１５Ａへの書き込みは、データ先頭ポインタとデータ書き込みポインタとの２つのポインタの位置関係だけで制御される。

一方、バッファ制御モジュール２１５は、バッファ２１５Ａからのデータの読み出しについては、ビデオ読み出しポインタ、オーディオ読み出しポインタ、および字幕読み出しポインタ、ひいては、データ先頭ポインタが、データ書き込みポインタを追い越さないように、バッファ２１５Ａからのデータの読み出しを制御する。

即ち、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタによるデータ書き込みポインタの追い越しが発生しない限り、バッファ制御モジュール２１５では、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、または字幕デコーダ制御モジュール２１８からの要求に応じて、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタが指すバッファ２１５Ａの位置からデータが読み出され、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタが更新されるとともに、必要に応じて、データ先頭ポインタが更新される。一方、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタによるデータ

書き込みポインタの追い越しが発生しそうになると、バッファ制御モジュール２１５では、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、または字幕デコーダ制御モジュール２１８からの要求が、例えば凍結され、十分なデータが用意されるまで待たされる。これにより、バッファ２１５Ａのアンダーフローを防止することができる。

以上から、バッファ２１５Ａには、データ先頭ポインタが指す位置から、右回りに、データ書き込みポインタが指す位置までの範囲（第３４図および第３５図において影を付してある部分）に、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、および字幕デコーダ制御モジュール２１８に供給すべきデータが記憶されており、さらに、その範囲内に、ビデオ読み出しポインタ、オーディオ読み出しポインタ、および字幕読み出しポインタは存在する。

なお、上述の場合には、データ先頭ポインタは、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタが指している位置のうちの、最も古いデータの位置を指すように更新することとしたが、その他、データ先頭ポインタの更新は、例えば、その最も古いデータの位置から、所定の時間（例えば、１秒）分だけ過去のデータの位置を指すように行うことが可能である。

即ち、一般には、ビデオ読み出しポインタ、オーディオ読み出しポインタ、または字幕読み出しポインタのうちの、ビデオ読み出しポインタやオーディオ読み出しポインタが、最も古いデータの位置を指すことが多いと予想される。

従って、データ先頭ポインタを、ビデオ読み出しポインタまたはオーディオ読み出しポインタが指す最も古いデータの位置から、例えば

、１秒分だけ過去のデータの位置を指すように更新した場合、第３４図に示すように、ビデオ読み出しポインタまたはオーディオ読み出しポインタが指す最も古いデータの位置から過去１秒分のデータを、バッファ２１５Ａに残しておくことができる。ここで、第３４図では、オーディオ読み出しポインタが、最も古いデータの位置を指しており、データ先頭ポインタは、その位置から１秒分だけ過去のデータの位置を指している。

以上のように、１秒分だけ過去のデータの位置を指すように、データ先頭ポインタを更新することにより、ディスク装置の応答性を向上させることができる。

即ち、第３５図に示したように、オーディオ読み出しポインタが指している最も古いデータの位置を指すように、データ先頭ポインタを更新する場合には、例えば、リバース方向への特殊再生が指示されたときに、バッファ２１５Ａからの読み出しが終了したデータを、ディスク１０１から再度読み出す必要があるため、特殊再生が指示されてから、その特殊再生が可能となるまでに、ある程度の時間がかかる。

これに対して、第３４図に示したように、オーディオ読み出しポインタが指している最も古いデータの位置から１秒分だけ過去のデータの位置を指すように、データ先頭ポインタを更新する場合には、リバース方向への特殊再生が指示されたときに、その特殊再生を開始するのに必要なデータが、バッファ２１５Ａに記憶されている１秒分だけ過去のデータであれば、上述したようなディスク１０１からのデータの再読み出しを行わずに、即座に、特殊再生を開始することが可能となる。

なお、オーディオ読み出しポインタが指している最も古いデータの位置から１秒分だけ過去のデータの位置を指すように、データ先頭ポ

インタを更新する場合であっても、特殊再生を開始するのに必要なデータが、バッファ２１５Ａに記憶されていないことがあり得る。この場合には、特殊再生を開始するのに必要なデータが、ディスク１０１から再度読み出される。

次に、バッファ２１５Ａからのビデオストリーム、オーディオストリーム、字幕ストリームそれぞれの読み出しの詳細について説明する。

第３０図のステップＳ１２７で説明したように、クリップストリームファイルの再生が開始されるときに、バッファ制御モジュール２１５においては、データ先頭ポインタ、データ書き込みポインタ、ビデオ読み出しポインタ、オーディオ読み出しポインタ、字幕読み出しポインタが、すべて、バッファ２１５Ａ上の同じ位置を指すように初期化される。

そして、ディスク１０１からクリップストリームファイルに格納されたプログラムストリーム(MPEG2-System Program Stream)が読み出され、バッファ制御モジュール２１５に供給されると、バッファ制御モジュール２１５では、そのプログラムストリームが、バッファ２１５Ａのデータ書き込みポインタが指す位置に記憶されるとともに、データ書き込みポインタが、右回りに更新されていく。

さらに、バッファ制御モジュール２１５（第３図）では、ビデオ読み出し機能部２３３が、バッファ２１５Ａに記憶されたプログラムストリームの構文解析を行い、ビデオデコーダ制御モジュール２１６からの要求に応じて、バッファ２１５Ａに記憶されたプログラムストリームから、ビデオストリーム（ビデオアクセスユニット）を抽出（分離）して読み出し、ビデオデコーダ制御モジュール２１６に供給する。

同様に、オーディオ読み出し機能部２３４も、バッファ２１５Ａに記憶されたプログラムストリームの構文解析を行い、オーディオデコーダ制御モジュール２１７からの要求に応じて、バッファ２１５Ａに記憶されたプログラムストリームから、オーディオストリーム（オーディオアクセスユニット）を抽出して読み出し、オーディオデコーダ制御モジュール２１７に供給する。字幕読み出し機能部２３５も、バッファ２１５Ａに記憶されたプログラムストリームの構文解析を行い、字幕デコーダ制御モジュール２１８からの要求に応じて、バッファ２１５Ａに記憶されたプログラムストリームから、字幕ストリーム（字幕アクセスユニット）を抽出して読み出し、字幕デコーダ制御モジュール２１８に供給する。

「ビデオストリームの読み出し」

次に、第３６図のフローチャートを参照して、ビデオ読み出し機能部２３３（第３図）による、バッファ２１５Ａからのビデオストリームの読み出し処理の詳細について説明する。

ビデオ読み出し機能部２３３は、まず最初に、ステップＳ２１１において、バッファ２１５Ａに記憶されたプログラムストリーム中のprivate_stream_2のPES_packet()を探索して見つけ出す。すなわち、private_stream_2のPES_packet()のstream_idは、第２０図で説明したように、10111111B(=0xBF)であり、ビデオ読み出し機能部２３３は、stream_idが10111111Bとなっているを探索して見つけ出す。

ここで、例えば、いま、上述したように、クリップストリームファイル"00001.PS"に格納されたプログラムストリームに多重化されたエレメンタリストリームが、再生対象のエレメンタリストリームであるとすると、そのプログラムストリームをディスク１０１から読み出し

て、バッファ２１５Ａに記憶させるときに、第３０図のステップＳ１２２において、クリップストリームファイル"00001.PS"のEP_map()（第２７図）に記述されたデコード開始可能点の情報から、305セクタが再生開始位置として決定され、さらに、第３０図のステップＳ１２８において、再生開始位置である305セクタが指定され、オペレーティングシステム２０１に対して、クリップストリームファイル"00001.PS"に格納されたプログラムストリームの読み出しが要求される。

また、ビデオストリームについては、EP_map()に記述されたデコード開始可能点の情報は、実際のデコード開始可能点の直前に配置されたprivate_stream_2のPES_packet()の位置を表す。

従って、クリップストリームファイル"00001.PS"に格納されたプログラムストリームがディスク１０１から読み出され、バッファ２１５Ａに記憶された直後においては、データ先頭ポインタやビデオ読み出しポインタが指すバッファ２１５Ａの位置には、private_stream_2のPES_packet()が記憶されている。

ビデオ読み出し機能部２３３は、ステップＳ２１１において、private_stream_2のPES_packet()が見つかると、ステップＳ２１２に進み、そのprivate_stream_2のPES_packet()のPES_packet_data_byteであるprivate_stream2_PES_payload()（第２３図）に記述されているvideo_stream_idを抜き出し、そのvideo_stream_idが、第３０図のステップＳ１２７でstream_idレジスタ２４２（第３図）に記憶された、再生対象のビデオストリームのstream_idと一致するかどうかを判定する。

ステップＳ２１２において、private_stream2_PES_payload()に記述されているvideo_stream_idが、stream_idレジスタ２４２に記憶されたstream_idと一致しないと判定された場合、即ち、直前のステッ

プＳ２１１で見つけ出されたprivate_stream_2のPES_packet()が、再生対象のビデオストリームのデコード開始点に配置されたものでない場合、ステップＳ２１１に戻り、バッファ２１５Ａに記憶されたプログラムストリーム中の他のprivate_stream_2のPES_packet()の探索が行われ、以下、同様の処理が繰り返される。

一方、ステップＳ２１２において、private_stream2_PES_payload()に記述されているvideo_stream_idが、stream_idレジスタ２４２に記憶されたstream_idと一致すると判定された場合、即ち、直前のステップＳ２１１で見つけ出されたprivate_stream_2のPES_packet()が、再生対象のビデオストリームのデコード開始点に配置されたものである場合、ステップＳ２１３に進み、ビデオ読み出し機能部２３３は、そのprivate_stream_2のPES_packet()のprivate_stream2_PES_payload()に記述されているau_information()を、バッファ２１５Ａから読み出し、au_information()レジスタ２４３（第３図）に記憶させ、ステップＳ２１４に進む。

ステップＳ２１４では、ビデオ読み出し機能部２３３は、直前のステップＳ２１１で見つけ出したprivate_stream_2のPES_packet()（video_stream_id（第２３図）が、stream_idレジスタ２４２（第３図）に記憶されたstream_idと一致するprivate_stream_2のPES_packet()）のサイズだけ、ビデオ読み出しポインタ記憶部２３１に記憶されたビデオ読み出しポインタを更新する。

即ち、クリップストリームファイルでは、private_stream_2のPES_packet()の直後に、そのvideo_stream_idと一致するstream_idのビデオストリーム（PES_packet()）が配置されており、従って、ステップＳ２１４では、ビデオ読み出しポインタは、ビデオストリームの実際のデコード開始可能点の位置を指すように更新される。

その後、ステップＳ２１４からＳ２１５に進み、ビデオ読み出し機能部２３３は、ビデデコーダ制御モジュール２１６から、データの要求があったかどうかを判定し、ないと判定した場合、ステップＳ２１５に戻り、同様の処理を繰り返す。

また、ステップＳ２１５において、ビデオデコーダ制御モジュール２１６から、データの要求があったと判定された場合、ステップＳ２１６に進み、ビデオ読み出し機能部２３３は、ビデオ読み出しポインタが指しているバッファ２１５Ａの位置からのプログラムストリームの構文解析を行いつつ、au_information()レジスタ２４３に記憶されたau_information()のAU_lengthに記述されたバイト数のデータ、つまり１つのビデオアクセスユニットを、バッファ２１５Ａから読み出し、ビデオデコーダ制御モジュール２１６に供給するとともに、ビデオ読み出しポインタを、バッファ２１５Ａから読み出した１つのビデオアクセスユニットのサイズ分だけ更新する。

即ち、au_information()には、第２４図で説明したように、それを含むprivate_stream_2のPES_packet()から、次のprivate_stream_2のPES_packet()までの間に含まれるビデオアクセスユニット（ピクチャ）の数を表すnumber_of_access_unitが記述されている。

さらに、au_information()には、第２４図で説明したように、そのnumber_of_access_unitの数だけのビデオアクセスユニットそれぞれに関する情報としてのpic_struct_copy, au_ref_flag、およびAU_lengthが記述されている。

au_information()にnumber_of_access_unitの数だけ記述されているAU_lengthそれぞれは、第２４図で説明したように、それを含むprivate_stream_2のPES_packet()から、次のprivate_stream_2のPES_packet()までの間に含まれる、number_of_access_unitの数のビデオアク

セスユニットそれぞれのサイズを表すから、ビデオ読み出し機能部２３３は、そのAU_lengthを用いることで、ビデオストリームの構文解析を行うことなく、アクセスユニットの切り出しを行うことが出来る。

即ち、従来、MPEG2-VideoやMPEG4-AVCのアクセスユニットを切り出す場合には、ビデオストリームの構文を知った上で、ビデオストリームの構文解析を行う必要があったが、ディスク１０１に記録されたクリップストリームファイルに格納されたプログラムストリームは、ビデオアクセスユニット単位のビデオストリームにおける１以上の実際のデコード開始可能点それぞれの直前に、ビデオアクセスユニットのサイズを表すAU_lengthが記述されたprivate_stream_2のPES_packet()を含んでいるので、ビデオ読み出し機能２３３は、そのprivate_stream_2のPES_packet()に記述されたAU_lengthに基づき、ビデオストリームの構文解析を行うことなく、バッファ２１５Ａから、ビデオアクセスユニット（単位のビデオストリーム）を読み出し、ビデオデコーダ制御モジュール２１６に供給することができる。

なお、ビデオ読み出し機能部２３３は、ステップＳ２１６において、ビデオアクセスユニットを、ビデオデコーダ制御モジュール２１６に供給するときに、そのビデオアクセスユニットに関する情報としてau_information()に記述されているpic_struct_copy, au_ref_flag、およびAU_lengthと、ビデオアクセスユニット単位に付加されているタイムスタンプ（PTS, DTS）も、ビデオデコーダ制御モジュール２１６に供給する。

ステップＳ２１６において、バッファ２１５Ａから１つのビデオアクセスユニットが読み出され、ビデオデコーダ制御モジュール２１６に供給された後は、ステップＳ２１７に進み、ビデオ読み出し機能部

２３３は、au_information()レジスタ２４３に記憶されたau_information()（第２４図）のnumber_of_access_unitが表す数だけのアクセスユニットを処理したかどうかを判定する。

ステップＳ２１７において、number_of_access_unitが表す数だけのアクセスユニットを、まだ処理していないと判定された場合、即ち、number_of_access_unitが表す数だけのアクセスユニットを、まだ、バッファ２１５Ａから読み出してビデオデコーダ制御モジュール２１６に供給していない場合、ステップＳ２１５に戻り、以下、同様の処理が繰り返される。

また、ステップＳ２１７において、number_of_access_unitが表す数だけのアクセスユニットを処理したと判定された場合、即ち、number_of_access_unitが表す数だけのアクセスユニットを、バッファ２１５Ａから読み出してビデオデコーダ制御モジュール２１６に供給した場合、ステップＳ２１１に戻り、次のprivate_stream_2のPES_packet()の探索が行われ、以下、同様の処理が繰り返される。

「オーディオストリームの読み出し」

次に、第３７図のフローチャートを参照して、オーディオ読み出し機能部２３４（第３図）による、バッファ２１５Ａからのオーディオストリームの読み出し処理の詳細について説明する。

オーディオ読み出し機能部２３４は、まず最初に、ステップＳ２３０において、第３０図のステップＳ１２７でstream_idレジスタ２５２（第３図）に記憶された、再生対象のオーディオストリームのstream_idが、private_stream_1のPES_packet()を表しているかどうかを判定する。

ステップＳ２３０において、stream_idレジスタ２５２に記憶されたstream_idが、private_stream_1のPES_packet()を表していない判

定された場合、即ち、stream_idレジスタ２５２に記憶されたstream_idが、第２０図で説明したように、MPEG規格にしたがって符号化されたオーディオストリームに割り当てられる110xxxxxBである場合、ステップＳ２３１に進み、オーディオ読み出し機能部２３４は、バッファ２１５Ａに記憶されたプログラムストリームから、MPEG Audioで定められたオーディオフレーム先頭を表す同期コードを探す。同期コードの位置がオーディオフレーム先頭なので、オーディオ読み出し機能部２３４は、オーディオ読み出しポインタを、オーディオフレーム先頭の位置を示すように更新し、ステップＳ２３１からＳ２３２に進む。ステップＳ２３２では、オーディオ読み出し機能部２３４は、バッファ２１５Ａに記憶されたプログラムストリーム中の、stream_idレジスタ２５２に記憶されたstream_idに一致するPES_packet()を、オーディオ読み出しポインタが示す位置から探索して見つけ出し、ステップＳ２３３に進む。

ステップＳ２３３では、オーディオ読み出し機能部２３４は、オーディオ読み出しポインタ記憶部２５１に記憶されたオーディオ読み出しポインタを、直前のステップＳ２３２で見つけ出したPES_packet()のPES_packet_data_byte（第１６図Ａおよび第１６図Ｂ乃至第１８図Ａおよび第１８図Ｂ）の先頭を指すように更新し、ステップＳ２３７に進む。

ステップＳ２３７では、オーディオ読み出し機能部２３４は、オーディオデコーダ制御モジュール２１７から、データの要求があったかどうかを判定し、ないと判定した場合、ステップＳ２３７に戻り、同様の処理を繰り返す。

また、ステップＳ２３７において、オーディオデコーダ制御モジュール２１７から、データの要求があったと判定された場合、ステップ

Ｓ２３８に進み、オーディオ読み出し機能部２３４は、オーディオ読み出しポインタが指しているバッファ２１５Ａの位置からのプログラムストリームの構文解析を行いつつ、既知の固定長の１つのオーディオアクセスユニットを、バッファ２１５Ａから読み出し、そのオーディオアクセスユニットに付加されているタイムスタンプ（PTS, DTS）とともに、オーディオデコーダ制御モジュール２１７に供給する。

そして、オーディオ読み出し機能部２３４は、バッファ２１５Ａから読み出した１つのオーディオアクセスユニットのサイズ分だけ、オーディオ読み出しポインタを更新して、ステップＳ２３７に戻り、以下、同様の処理が繰り返される。

一方、ステップＳ２３０において、stream_idレジスタ２５２に記憶されたstream_idが、private_stream_1のPES_packet()を表していると判定された場合、即ち、stream_idレジスタ２５２に記憶されたstream_idが、10111101B(=0xBD)であり、第２０図で説明したように、private_stream_1のPES_packet()を表している場合、ステップＳ２３４に進み、オーディオ読み出し機能部２３４は、バッファ２１５Ａに記憶されたプログラムストリーム中のprivate_stream_1のPES_packet()を探索して見つけ出す。すなわち、オーディオ読み出し機能部２３４は、stream_idが10111101Bとなっているprivate_stream_1のPES_packet()を探索して見つけ出す。

オーディオ読み出し機能部２３４は、ステップＳ２３４において、private_stream_1のPES_packet()が見つかると、ステップＳ２３５に進み、そのprivate_stream_1のPES_packet()のPES_packet_data_byteであるprivate_stream1_PES_payload()（第２１図）に記述されているprivate_stream_idを抜き出し、そのprivate_stream_idが、第３０図のステップＳ１２７でprivate_stream_idレジスタ２５３（第３図

）に記憶された、再生対象のオーディオストリームのprivate_stream_idと一致するかどうかを判定する。

ステップＳ２３５において、private_stream1_PES_payload()に記述されているprivate_stream_idが、private_stream_idレジスタ２５３に記憶されたprivate_stream_idと一致しないと判定された場合、即ち、直前のステップＳ２３４で見つけ出されたprivate_stream_1のPES_packet()が、再生対象のオーディオストリームではない場合、ステップＳ２３４に戻り、バッファ２１５Ａに記憶されたプログラムストリーム中の他のprivate_stream_1のPES_packet()の探索が行われ、以下、同様の処理が繰り返される。

一方、ステップＳ２３５において、private_stream1_PES_payload()に記述されているprivate_stream_idが、private_stream_idレジスタ２５３に記憶されたprivate_stream_idと一致すると判定された場合、即ち、直前のステップＳ２３４で見つけ出されたprivate_stream_1のPES_packet()が、再生対象のオーディオストリームである場合、ステップＳ２３６に進み、オーディオ読み出し機能部２３４は、そのprivate_stream_1のPES_packet()のprivate_stream1_PES_payload()（第２１図）に記述されているAU_locatorを、バッファ２１５Ａから読み出し、そのAU_locatorの直後の位置と、そのAU_locatorが表す値とを加算することで、オーディオアクセスユニットの先頭位置を求める。

即ち、AU_locatorは、第２１図で説明したように、そのAU_locatorの直後の位置を基準として、private_stream1_PES_payload()のprivate_payload()に格納されるオーディオアクセスユニット（あるいは字幕アクセスユニット）の先頭位置を表すから、AU_locatorの直後の位置に、そのAU_locatorが表す値を加算することにより、オーディオア

クセスユニットの（絶対的な）先頭位置を求めることができる。

オーディオ読み出し機能部２３４は、さらに、ステップＳ２３６において、以上のようにして求めたオーディオアクセスユニットの先頭位置を指すように、オーディオ読み出しポインタ記憶部２５１に記憶されたオーディオ読み出しポインタを更新し、ステップＳ２３７に進む。

ステップＳ２３７では、オーディオ読み出し機能部２３４は、オーディオデコーダ制御モジュール２１７から、データの要求があったかどうかを判定し、ないと判定した場合、ステップＳ２３７に戻り、同様の処理を繰り返す。

また、ステップＳ２３７において、オーディオデコーダ制御モジュール２１７から、データの要求があったかと判定された場合、ステップＳ２３８に進み、オーディオ読み出し機能部２３４は、オーディオ読み出しポインタが指しているバッファ２１５Ａの位置からのプログラムストリームの構文解析を行いつつ、既知の固定長の１つのオーディオアクセスユニットを、バッファ２１５Ａから読み出し、そのオーディオアクセスユニットに付加されているタイムスタンプとともに、オーディオデコーダ制御モジュール２１７に供給する。

そして、オーディオ読み出し機能部２３４は、バッファ２１５Ａから読み出した１つのオーディオアクセスユニットのサイズ分だけ、オーディオ読み出しポインタを更新して、ステップＳ２３７に戻り、以下、同様の処理が繰り返される。

「字幕ストリームの読み出し」

次に、第３８図のフローチャートを参照して、字幕読み出し機能部２３５（第３図）による、バッファ２１５Ａからの字幕ストリームの読み出し処理の詳細について説明する。

字幕読み出し機能部２３５は、まず最初に、ステップＳ２５１において、第３０図のステップＳ１２７で字幕読み出し機能フラグ記憶部２６１に記憶された字幕読み出し機能フラグを判定する。ステップＳ２５１において、字幕読み出し機能フラグが0であると判定された場合、即ち、例えば、再生対象のエレメンタリストリームが多重化されているクリップストリームファイルに字幕ストリームが含まれておらず、第３０図のステップＳ１２７で字幕読み出し機能フラグ記憶部２６１に、0がセットされた場合、字幕読み出し機能部２３５は特に処理を行わない。

一方、ステップＳ２５１において、字幕読み出し機能フラグが1であると判定された場合、即ち、例えば、再生対象のエレメンタリストリームが多重化されているクリップストリームファイルに字幕ストリームが含まれており、第３０図のステップＳ１２７で字幕読み出し機能フラグ記憶部２６１に、1がセットされた場合、ステップＳ２５２に進み、字幕読み出し機能部２３５は、stream_idレジスタ２６３（第３図）に記憶された、再生対象の字幕ストリームのstream_idに一致するPES_packet()を、バッファ２１５Ａに記憶されたプログラムストリームから探索する。

ここで、第３０図のステップＳ１２７で説明したように、stream_idレジスタ２６３（第３図）には、再生対象の字幕ストリームのstream_idが記憶されるが、字幕ストリームのstream_idは、第２０図で説明したように、private_stream_1のPES_packet()を表す10111101B(=0xBD)である。

従って、ステップＳ２５２では、バッファ２１５Ａに記憶されたプログラムストリーム中のprivate_stream_1のPES_packet()が探索されることになる。

ステップＳ２５２において、private_stream_1のPES_packet()の探索が行われ、private_stream_1のPES_packet()が見つかると、ステップＳ２５３に進み、字幕読み出し機能部２３５は、そのprivate_stream_1のPES_packet()のPES_packet_data_byteであるprivate_stream1_PES_payload()（第２１図）に記述されているprivate_stream_idを抜き出し、そのprivate_stream_idが、第３０図のステップＳ１２７でprivate_stream_idレジスタ２６４（第３図）に記憶された、再生対象の字幕ストリームのprivate_stream_idと一致するかどうかを判定する。

ステップＳ２５３において、private_stream1_PES_payload()に記述されているprivate_stream_idが、private_stream_idレジスタ２６４に記憶されたprivate_stream_idと一致しないと判定された場合、即ち、直前のステップＳ２５２で見つかったprivate_stream_1のPES_packet()が、再生対象の字幕ストリームではない場合、ステップＳ２５２に戻り、バッファ２１５Ａに記憶されたプログラムストリーム中の他のprivate_stream_1のPES_packet()の探索が行われ、以下、同様の処理が繰り返される。

一方、ステップＳ２５３において、private_stream1_PES_payload()に記述されているprivate_stream_idが、private_stream_idレジスタ２６４に記憶されたprivate_stream_idと一致すると判定された場合、即ち、直前のステップＳ２５２で見つかったprivate_stream_1のPES_packet()が、再生対象の字幕ストリームである場合、ステップＳ２５４に進み、字幕読み出し機能部２３５は、そのprivate_stream_1のPES_packet()のprivate_stream1_PES_payload()（第２１図）に記述されているAU_locatorを、バッファ２１５Ａから読み出し、そのAU_locatorの直後の位置と、そのAU_locatorが表す値とを加算すること

で、字幕アクセスユニットの先頭位置を求める。

即ち、AU_locatorは、第２１図で説明したように、そのAU_locatorの直後の位置を基準として、private_stream1_PES_payload()のprivate_payload()に格納される字幕アクセスユニット（あるいはオーディオアクセスユニット）の先頭位置を表すから、AU_locatorの直後の位置に、そのAU_locatorが表す値を加算することにより、字幕アクセスユニットの（絶対的な）先頭位置を求めることができる。

字幕読み出し機能部２３５は、さらに、ステップＳ２５４において、以上のようにして求めた字幕アクセスユニットの先頭位置を指すように、字幕読み出しポインタ記憶部２６２に記憶された字幕読み出しポインタを更新し、ステップＳ２５５に進む。

ステップＳ２５５では、字幕読み出し機能部２３５は、字幕デコーダ制御モジュール２１８から、データの要求があったかどうかを判定し、ないと判定した場合、ステップＳ２５５に戻り、同様の処理を繰り返す。

また、ステップＳ２５５において、字幕デコーダ制御モジュール２１８から、データの要求があったかと判定された場合、ステップＳ２５６に進み、字幕読み出し機能部２３５は、字幕読み出しポインタが指しているバッファ２１５Ａの位置からのプログラムストリームの構文解析を行いつつ、字幕アクセスユニットの先頭に記述されているサイズ分の１つの字幕アクセスユニットを、バッファ２１５Ａから読み出し、その字幕アクセスユニットに付加されているタイムスタンプとともに、字幕デコーダ制御モジュール２１８に供給する。即ち、字幕アクセスユニットの先頭には、第２図Ａおよび第２図Ｂで説明したように、その字幕アクセスユニットのサイズが記述されており、字幕読み出し機能部２３５は、そのサイズ分のデータを、字幕読み出しポイ

ンタが指しているバッファ２１５Ａの位置から読み出し、その読み出したデータである字幕アクセスユニットを、その字幕アクセスユニットに付加されているタイムスタンプとともに、字幕デコーダ制御モジュール２１８に供給する。

そして、字幕読み出し機能部２３５は、バッファ２１５Ａから読み出した１つの字幕アクセスユニットのサイズ分だけ、字幕読み出しポインタを更新して、ステップＳ２５５に戻り、以下、同様の処理が繰り返される。

［再同期処理］

次に、第２図Ａおよび第２図Ｂのデコード制御モジュール２１４による、ビデオデータとオーディオデコーダとの同期制御について説明する。

第３０図のＳ１３０で説明したように、デコード制御モジュール２１４は、同期を確保するために必要であればタイミングをずらして、デコード開始を、ビデオデコーダ制御モジュール２１６、オーディオデコーダ制御モジュール２１７、および字幕デコーダ制御モジュール２１８に指令するが、例えば、その後のビデオデコーダ１１６とオーディオデコーダ１１７のデコード処理の進行程度によって、ビデオデータの出力と、そのビデオデータと同期して出力されるべき出力データとしてのオーディオデータの出力とがずれることがある。

そこで、デコード制御モジュール２１４では、ビデオデータの出力と、そのビデオデータと同期して出力されるべきオーディオデータの出力とに生じたずれを補正し、ビデオデータとオーディオデータとが同期して出力されるようにするための再同期処理が行われる。

第３９図のフローチャートを参照して、再同期処理について説明する。

再同期処理では、まず最初に、ステップＳ２７１において、デコード制御モジュール２１４は、ビデオデコーダ制御モジュール２１６からのビデオアクセスユニットのタイムスタンプと、オーディオ制御モジュール２１７からのオーディオアクセスユニットのタイムスタンプとのずれが大であるかどうかを判定する。

即ち、第３０図のステップＳ１２９で説明したように、ビデオデコーダ制御モジュール２１６は、バッファ制御モジュール２１５からビデオアクセスユニットを得るたびに、そのビデオアクセスユニットのタイムスタンプを、デコード制御モジュール２１４に供給する。同様に、オーディオ制御モジュール２１７も、バッファ制御モジュール２１５からオーディオアクセスユニットを得るたびに、そのオーディオアクセスユニットのタイムスタンプを、デコード制御モジュール２１４に供給する。

ステップＳ２７１では、デコード制御モジュール２１４は、ビデオデコーダ制御モジュール２１６とオーディオ制御モジュール２１７とのそれぞれから、同一タイミングで（同一タイミングとみなすことができる、ある時間内に）供給されるタイムスタンプどうしを比較し、それらのタイムスタンプのずれが大であるかどうかを判定する。

ステップＳ２７１において、ビデオデコーダ制御モジュール２１６からのビデオアクセスユニットのタイムスタンプと、オーディオ制御モジュール２１７からのオーディオアクセスユニットのタイムスタンプとのずれが大でないと判定された場合、即ち、ビデオアクセスユニットのタイムスタンプと、オーディオアクセスユニットのタイムスタンプとのずれが、あらかじめ定められた同期がとれているとみなすことができる範囲内である、例えば、２ビデオフレーム（約６６ミリ秒）である場合、ステップＳ２７１に戻り、タイムスタンプどうしのず

れの判定（監視）が続行される。

一方、ステップＳ２７１において、ビデオデコーダ制御モジュール２１６からのビデオアクセスユニットのタイムスタンプと、オーディオ制御モジュール２１７からのオーディオアクセスユニットのタイムスタンプとのずれが大であると判定された場合、即ち、ビデオアクセスユニットのタイムスタンプと、オーディオアクセスユニットのタイムスタンプとのずれが、あらかじめ定められた同期がとれているとみなすことができる範囲外である場合、ステップＳ２７２に進み、デコード制御モジュール２１４は、ビデオデコーダ制御モジュール２１６からのビデオアクセスユニットのタイムスタンプと、オーディオ制御モジュール２１７からのオーディオアクセスユニットのタイムスタンプとを比較することにより、ビデオデータの出力（デコード）と、オーディオデータの出力とのうちのいずれが遅れているかを判定する。

ステップＳ２７２において、ビデオデータの出力が、オーディオデータの出力よりも遅れていると判定された場合、ステップＳ２７３に進み、デコード制御モジュール２１４は、１ビデオアクセスユニットだけ、ビデオアクセスユニットの処理を進めるために、ビデオデコーダ制御モジュール２１６に対して、ビデオアクセスユニットのデコードと出力（表示）を行わない旨の指示、即ち、ビデオアクセスユニットの処理のスキップの指示を出力して、ステップＳ２７４に進む。

ステップＳ２７４では、ビデオデコーダ制御モジュール２１６は、デコード制御モジュール２１４からのスキップの指示を受信し、そのスキップの指示に応じて、バッファ制御モジュール２１５からのビデオアクセスユニットとともに供給されるau_ref_flag（第２４図）を検査する。

即ち、private_stream_2のPES_packet()のprivate_stream2_PES_pa

yload()（第２３図）に配置されたau_information()（第２４図）には、アクセスユニットに関する情報としてのau_ref_flagが含まれており、バッファ制御モジュール２１５は、第３０図のステップＳ１２９や、第３６図のステップＳ２１６で説明したように、ビデオアクセスユニットとともに、そのビデオアクセスユニットのau_ref_flagを、ビデオデコーダ制御モジュール２１６に供給する。

ステップＳ２７４では、このように、アクセスユニットとともに供給される、そのアクセスユニットのau_ref_flagが検査される。

そして、ステップＳ２７４からＳ２７５に進み、ビデオデコーダ制御モジュール２１６は、バッファ制御モジュール２１５から供給されたビデオアクセスユニットのau_ref_flagの検査の結果に基づき、そのビデオアクセスユニットが、他のピクチャのデコードにあたって参照されない非参照画像であるかどうかを判定する。

ここで、第２４図で説明したように、ビデオアクセスユニットのau_ref_flagは、そのアクセスユニットが参照画像であるか否かを表し、参照画像である場合には１とされ、参照画像でない場合、即ち、非参照画像である場合には０とされる。

ステップＳ２７５において、バッファ制御モジュール２１５から供給されたビデオアクセスユニットが非参照画像（のビデオアクセスユニット）でないと判定された場合、即ち、バッファ制御モジュール２１５から供給されたビデオアクセスユニットが参照画像である場合、ステップＳ２７６に進み、ビデオデコーダ制御モジュール２１６は、通常通り、そのビデオアクセスユニットを、ビデオデコーダ１１６に処理させ、次のビデオアクセスユニットが、バッファ制御モジュール２１５から供給されるのを待って、ステップＳ２７４に戻る。

また、ステップＳ２７５において、バッファ制御モジュール２１５

から供給されたビデオアクセスユニットが非参照画像であると判定された場合、ステップＳ２７７に進み、ビデオデコーダ制御モジュール２１６は、そのビデオアクセスユニットの、ビデオデコーダ１１６による処理をスキップさせ、次のビデオアクセスユニットが、バッファ制御モジュール２１５から供給されるのを待って、ステップＳ２７１に戻る。

このように、ビデオアクセスユニットの処理がスキップされることにより、ビデオアクセスユニットの処理が、ほぼ１ビデオアクセスユニット分だけ進められる（処理時間が短縮される）。その結果、オーディオデータの出力よりも遅れていたビデオデータの出力が早まることになる。

一方、ステップＳ２７２において、ビデオデータの出力が、オーディオデータの出力よりも遅れていないと判定された場合、即ち、オーディオデータの出力が、ビデオデータの出力よりも遅れている場合、ステップＳ２７８に進み、デコード制御モジュール２１４は、ビデオアクセスユニットの処理を待たせるために、ビデオデコーダ制御モジュール２１６に対して、いまデコードされているビデオアクセスユニットに対応するビデオデータを繰り返して出力する繰り返し出力の指示を出力して、ステップＳ２７９に進む。

ステップＳ２７９では、ビデオデコーダ制御モジュール２１６は、デコード制御モジュール２１４からの繰り返し出力の指示を受信し、その繰り返し出力の指示に応じて、いまビデオデコーダ１１６でデコードされているビデオアクセスユニットに対応するビデオデータを繰り返して、グラフィクス処理モジュール２１９に出力し、次のビデオアクセスユニットが、バッファ制御モジュール２１５から供給されるのを待って、ステップＳ２７１に戻る。

以上のように、デコード制御モジュール２１４では、ビデオデータの出力が、オーディオデータの出力よりも遅れているか否かを判定し、ビデオデータの出力が、オーディオデータの出力よりも遅れている場合には、１つのアクセスユニットの処理のスキップを、ビデオデコーダ制御モジュール２１６に指示する。そして、ビデオデコーダ制御モジュール２１６では、スキップが指示されたアクセスユニットのau_ref_flagに基づき、そのアクセスユニットが参照画像であるか、または非参照画像であるかを判定し、非参照画像である場合に、ビデオデコーダ１１６に、スキップが指示されたアクセスユニットの処理をスキップさせる。従って、ビデオデータの出力と、オーディオデータの出力との同期を、容易にとることができる。

即ち、処理をスキップするアクセスユニットが参照画像である場合、そのアクセスユニットに対応するビデオデータは、その後にデコードされる他のアクセスユニットのデコード時に参照するためにデコードする必要がある。従って、ビデオデータの出力と、オーディオデータの出力との同期をとるための同期制御において、参照画像のアクセスユニットの処理をスキップしてしまうと、その参照画像を参照する他のアクセスユニットをデコードすることができず、その結果、ビデオデータの表示において、同期制御がノイズとして現れてしまう。

このため、処理をスキップするのは、参照画像でないアクセスユニット、即ち、非参照画像のアクセスユニットとするのが望ましい。

一方、従来のエレメンタリストリームについて、非参照画像のアクセスユニットを探すためには、エレメンタリストリームの構文解析を行う必要があるが、例えば、MPEG4-AVCなどにしたがった符号化により得られるエレメンタリストリームは、構文が非常に複雑であるため、構文解析に、多大なコストがかかる。

これに対して、ディスク１０１に記録されたクリップストリームファイルに格納されたプログラムストリームには、ビデオアクセスユニットがPES_packet_data_byteに配置されるPES_packet()（第１６図Ａおよび第１６図Ｂ乃至第１８図Ａおよび第１８図Ｂ）とは別に、PES_packet_data_byteを拡張したprivate_stream2_PES_payload()（第２３図）が配置されたprivate_stream_2のPES_packet()が多重化されており、そのprivate_stream2_PES_payload()のau_information()（第２４図）には、ビデオアクセスユニットごとに、そのビデオアクセスユニットが参照画像であるか、または非参照画像であるかを表すau_ref_flagが記述されている。そして、そのau_ref_flagは、対応するビデオアクセスユニットとともに、バッファ制御モジュール２１５からビデオデコーダ制御モジュール２１６に供給される。従って、ビデオデコーダ制御モジュール２１６は、ビデオアクセスユニットとともに供給される、そのビデオアクセスユニットのau_ref_flagを検査することにより、コストをほとんどかけずに、ビデオアクセスユニットが参照画像であるか、または非参照画像であるかを認識することができる。

［マーク処理］

次に、第４０図のフローチャートを参照して、PlayListMark()（第７図）に記述されたMark()に基づいて行われるマーク処理について説明する。

デコード制御モジュール２１４は、内蔵する計時部２１４Ａによって計時されている現在時刻を、常時確認しており、ステップＳ３０１において、現在時刻が、PlayListMark()（第７図）に記述されたいずれかのMark()のmark_time_stampに一致したか否かを判定する。

即ち、第３０図のステップＳ１２４で説明したように、プレイヤ制

御モジュール２１２は、第２５図に示した１番目のPlayList#0の１番目のPlayItem#0を再生しようとするときに、第２８図上側に示したPlayListMark()に含まれる７つのMark()のうちの１番目から４番目までの４つのMark()が、PlayList#0の１番目のPlayItem#0に属していることを認識し、その４つのMark()のmark_time_stampである{180,090}、{5,580,090}、{10,980,090}、{16,380,090}を、そのmark_time_stampが表す時刻の属性が「マーク処理」である旨とともに、デコード制御モジュール２１４に渡している。

ステップＳ３０１では、デコード制御モジュール２１４において、現在時刻が、上述のようにしてプレイヤ制御モジュール２１２から供給された「マーク処理」の属性の時刻(mark_time_stamp)のうちのいずれかと一致するかどうかが判定される。

ステップＳ３０１において、現在時刻が、「マーク処理」の属性の時刻のうちのいずれとも一致しないと判定された場合、ステップＳ３０１に戻り、同様の処理が繰り返される。

また、ステップＳ３０１において、現在時刻が、「マーク処理」の属性の時刻のうちのいずれかと一致すると判定された場合、デコード制御モジュール２１４は、現在時刻が、「マーク処理」の属性の時刻となった旨のメッセージと、現在時刻と一致した「マーク処理」の属性の時刻とを、プレイヤ制御モジュール２１２に供給して、ステップＳ３０２に進む。

ステップＳ３０２では、プレイヤ制御モジュール２１２が、現在時刻が、「マーク処理」の属性の時刻となった旨のメッセージと、現在時刻と一致した「マーク処理」の属性の時刻(mark_time_stamp)とを、デコード制御モジュール２１４から受信し、mark_time_stampが現在時刻に一致したMark()を、マーク処理の処理対象とするMark()（以

下、適宜、処理対象markという）として認識する。

即ち、プレイヤ制御モジュール２１２は、現在再生されているPlayList()のPlayItem()を認識しており、そのPlayList()およびPlayItem()と、デコード制御モジュール２１４からの、現在時刻と一致した「マーク処理」の属性の時刻(mark_time_stamp)（以下、適宜、マーク時刻という）とから、"PLAYLIST.DAT"ファイル（第５図）のPlayListMark()（第７図）を参照することにより、処理対象markを認識する。

具体的には、例えば、いま、第２５図に示した１番目のPlayList#0の１番目のPlayItem#0が再生されているとすると、そのことにより、プレイヤ制御モジュール２１２は、マーク時刻が、第２８図上側に示したPlayListMark()に含まれる７つのMark()のうちの１番目から４番目までの４つのMark()のうちのいずれかのmark_time_stampであることを認識する。

そして、デコード制御モジュール２１４からプレイヤ制御モジュール２１２に供給されたマーク時刻が、例えば、16,380,090であったとすると、プレイヤ制御モジュール２１２は、第２８図上側に示したPlayListMark()に含まれる１番目から４番目までの４つのMark()のうちの、mark_time_stampが、マーク時刻である16,380,090に一致する４番目のMark()を、処理対象markとして認識する。

プレイヤ制御モジュール２１２は、以上のようにして、処理対象markを認識すると、ステップＳ３０２からＳ３０３に進み、処理対象markにおいて、エレメンタリストリームを特定するentry_ES_stream_idとentry_ES_private_stream_id（第７図）が記述されているかどうかを判定する。

ステップＳ３０３において、処理対象markに、エレメンタリストリームを特定するentry_ES_stream_idとentry_ES_private_stream_id（

第７図）が記述されていないと判定された場合、即ち、entry_ES_stream_idとentry_ES_private_stream_idが、いずれも0x00である場合、ステップＳ３０４をスキップして、ステップＳ３０５に進み、以下、処理対象markに応じた処理が行われる。

また、ステップＳ３０３において、処理対象markに、エレメンタリストリームを特定するentry_ES_stream_idとentry_ES_private_stream_id（第７図）が記述されていると判定された場合、ステップＳ３０４に進み、プレイヤ制御モジュール２１２は、再生中のエレメンタリストリームに、そのentry_ES_stream_id、さらには、必要に応じてentry_ES_private_stream_idによって特定されるエレメンタリストリームが含まれるかどうかを判定する。

ステップＳ３０４において、再生中のエレメンタリストリームに、処理対象markのentry_ES_stream_idとentry_ES_private_stream_idによって特定されるエレメンタリストリームが含まれないと判定された場合、ステップＳ３０１に戻る。即ち、処理対象markのentry_ES_stream_idとentry_ES_private_stream_idによって特定されるエレメンタリストリームが再生されていない場合、処理対象markは、無視される。

一方、ステップＳ３０４において、再生中のエレメンタリストリームに、処理対象markのentry_ES_stream_idとentry_ES_private_stream_idによって特定されるエレメンタリストリームが含まれると判定された場合、即ち、処理対象markのentry_ES_stream_idとentry_ES_private_stream_idによって特定されるエレメンタリストリームが再生されている場合、処理対象markは有効であるとして、ステップＳ３０５に進み、以下、その処理対象markに応じた処理が行われる。

即ち、ステップＳ３０５では、プレイヤ制御モジュール２１２は、

処理対象markのmark_type（第７図）を参照することにより、その処理対象markを判定する。

ステップＳ３０５において、処理対象markが、チャプタマークまたはインデクスマークであると判定された場合、即ち、処理対象markのmark_typeが、'Chapter'または'Index'である場合、ステップＳ３０６に進み、プレイヤ制御モジュール２１２は、グラフィクス処理モジュール２１９に命じて、チャプタまたはインデクスの番号の表示を、処理対象markであるチャプタマークまたはインデクスマークが表すチャプタまたはインデクスの番号に更新させて、ステップＳ３０１に戻る。

また、ステップＳ３０５において、処理対象markが、イベントマークであると判定された場合、即ち、処理対象markのmark_typeが、'Event'である場合、ステップＳ３０７に進み、プレイヤ制御モジュール２１２は、イベントの発生を表すイベントメッセージと、処理対象markのmark_dataを、スクリプト制御モジュール２１１に通知（供給）して、ステップＳ３０８に進む。

ステップＳ３０８では、スクリプト制御モジュール２１１が、プレイヤ制御モジュール２１２からのイベントメッセージとmark_dataとを受信し、イベントメッセージを割り込み要求として、あらかじめ"SCRIPT.DAT"ファイルに記述された一連の処理を、mark_dataを引数として行って、ステップＳ３０１に戻る。

即ち、スクリプト制御モジュール２１１では、mark_dataに対応した処理が行われる。

具体的には、例えば、第２８図下側に示したPlayList #1のPlayListMark()では、２番目のMark()（Mark#1）と３番目のMark()（Mark#2）とは、いずれも、mark_typeが'Event'であるが、mark_dataは、そ

れぞれ1（Mark#1）と2（Mark#2）で異なっている。

スクリプト制御モジュール２１１は、２番目のMark()に対応するイベントメッセージを受信した場合と、３番目のMark()に対応するイベントメッセージを受信した場合の、いずれも場合も、そのイベントメッセージに応じて、同一のイベントハンドラ（割り込み処理ルーチン）で処理を行うが、イベントハンドラ内において、イベントメッセージとともに供給されるmark_dataを検査することにより、イベントメッセージに対して、mark_dataごとに異なる処理を行う。

具体的には、例えば、mark_dataが1である場合には、スクリプト制御モジュール２１１は、グラフィクス処理モジュール２１９を制御して、第１の種類のアイコンの表示を行わせる。また、例えば、mark_dataが2である場合、スクリプト処理モジュール２１１は、グラフィクス処理モジュール２１９を制御して、第２の種類のアイコンの表示を行わせる。

なお、mark_dataは、１や２に限定されるものではなく、また、mark_dataに対応して行われる処理も、上述したような、単なるアイコンの表示限定されるものではない。

即ち、例えば、mark_dataが3乃至18の範囲の値である場合には、スクリプト制御モジュール２１１は、グラフィクス処理モジュール２１９を制御し、第１の種類のアイコンの表示を、mark_dataから2を減じた値（1～16の数値）に対応する明るさで行わせる。また、例えば、mark_dataが19乃至34の範囲の値である場合には、スクリプト制御モジュール２１１は、グラフィクス処理モジュール２１９を制御し、第２の種類のアイコンの表示を、mark_dataから18を減じた値（1～16の数値）に対応する明るさで行わせる。

その他、例えば、入力インターフェース１１５（第１図）に、ユー

ザが操作するコントローラが接続されており、そのコントローラが、DC(Direct Current)モータの軸に偏芯させたおもりを取り付けた、DCモータを動作させると振動が発生する振動モータを内蔵する場合には、mark_dataが35乃至42の範囲の値であるときに、その振動モータを、mark_dataから34を減じた値（1～8の数値）に応じた動作時間だけ動作させることができる。

mark_dataは数値であり、その使用法やアルゴリズムは、スクリプト制御モジュール２１１が実行するスクリプトプログラムにより記述することができる。従って、mark_dataは、事前に取り決められたルールで使用する他、ディスク１０１の製造者、あるいはディスク１０１に記録されるデータを提供するコンテンツプロバイダなどが独自に設定したルールで使用することが可能である。

以上のように、マーク処理では、現在時刻が、「マーク処理」の属性の時刻と一致すると、その「マーク処理」の属性の時刻であるマーク時刻から、処理対象markが認識される。さらに、処理対象markにおいて、エレメンタリストリームを特定するentry_ES_stream_idとentry_ES_private_stream_idが記述されていない場合には、処理対象markのmark_typeに応じた処理が行われる。また、処理対象markにおいて、エレメンタリストリームを特定するentry_ES_stream_idとentry_ES_private_stream_idが記述されている場合であっても、そのentry_ES_stream_idとentry_ES_private_stream_idによって特定されるエレメンタリストリームが再生中であれば、処理対象markのmark_typeに応じた処理が行われる。

従って、例えば、いま、第２５図に示した２番目のPlayList#1の再生が行われているとすると、以下のようなマーク処理が行われる。

即ち、２番目のPlayList#1のPlayListMark()においては、第２８図

下側に示したように、mark_time_stampがそれぞれ90,000, 27,090,000, 27,540,000に指定されている１番目のMark()(Mark#0)、２番目のMark() (Mark#1) 、３番目のMark() (Mark#2) が記述されている。

さらに、第２８図下側のPlayListMark()においては、２番目のMark()と３番目のMark()のentry_ES_stream_idには、それぞれ、0xE0と0xE1が記述されているから、２番目のMark()と３番目のMark()は、それぞれ、stream_idが0xE0と0xE1で特定されるエレメンタリストリームが関連付けられている。

ここで、第２５図で説明したように、２番目のPlayList#1には、１つのPlayItem()(PlayItem#0)だけが記述され、そのPlayItem#0によれば、クリップストリームファイル"00003.PS"が再生される。そして、クリップストリームファイル"00003.PS"には、そのクリップストリームファイル"00003.PS"に対応する第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"で説明したように、0xE0となっているstream_idで特定されるビデオストリームstream#0、0xE1となっているstream_idで特定されるビデオストリームstream#1、0xBDとなっているstream_idおよび0x00となっているprivate_stream_idで特定されるオーディオストリームstream#2の３つのエレメンタリストリームが多重化されている。

従って、第２８図下側のPlayListMark()の２番目のMark()には、クリップストリームファイル"00003.PS"に多重化されている、stream_idが0xE0となっているビデオストリームstream#0が関連付けられており、３番目のMark()には、クリップストリームファイル"00003.PS"に多重化されている、stream_idが0xE1となっているビデオストリームstream#1が関連付けられている。

第２５図の２番目のPlayList#1のPlayItem#0の再生が開始される場

合、第３０図のステップＳ１２４で説明したようにして、プレイヤ制御モジュール２１２は、第２８図下側に示したPlayListMark()に含まれる３つのMark()が、PlayList#1のPlayItem#0に属していることを認識し、その３つのMark()のmark_time_stampである{90,000}、{27,090,000}、{27,540,000}を、そのmark_time_stampが表す時刻の属性が「マーク処理」である旨とともに、デコード制御モジュール２１４に渡している。

マーク処理では、デコード制御モジュール２１４が、PlayList#1のPlayItem#0の再生中に、計時部２１４Ａによって計時される現在時刻が、属性が「マーク処理」の時刻{90,000}、{27,090,000}、{27,540,000}のうちのいずれかに一致するかを、常時確認しており（ステップＳ３０１）、現在時刻が、属性が「マーク処理」の時刻に一致すると、現在時刻と一致した「マーク処理」の属性の時刻であるマーク時刻と、現在時刻が、「マーク処理」の属性の時刻となった旨のメッセージとを、プレイヤ制御モジュール２１２に供給する。

即ち、例えば、いま、現在時刻が、「マーク処理」の属性の時刻{90,000}、{27,090,000}、{27,540,000}のうちの、27,090,000に一致したとすると、デコード制御モジュール２１４は、現在時刻と一致した「マーク処理」の属性の時刻であるマーク時刻27,090,000と、現在時刻が、「マーク処理」の属性の時刻となった旨のメッセージとを、プレイヤ制御モジュール２１２に供給する。

プレイヤ制御モジュール２１２は、PlayList#1のPlayItem#0が現在再生されていることを認識しており、そのPlayList#1の第２８図下側に示したPlayListMark()に記述されたMark()のうちの、PlayItem#0に属する３つのMark()のmark_time_stampである90,000, 27,090,000, 27,540,000それぞれと、デコード制御モジュール２１４からのマーク

時刻である27,090,000とを比較することにより、そのマーク時刻27,090,000に、mark_time_stampが一致するMark()、即ち、第２８図下側のPlayListMark()に記述された２番目のMark()(Mark#1)を、処理対象markとして認識する（ステップＳ３０２）。

処理対象markである、第２８図下側のPlayListMark()に記述された２番目のMark()においては、entry_ES_stream_idとして、0xE0が指定されている。この0xE0となっているentry_ES_stream_idは、上述したことから、クリップストリームファイル"00003.PS"に多重化されている、stream_idが0xE0となっているビデオストリームstream#0（第２６図Ａおよび第２６図Ｂ）を表すものであり、プレイヤ制御モジュール２１２は、再生中のエレメンタリストリームの中に、そのビデオストリームstream#0が含まれるかどうかを判定する（ステップＳ３０３，Ｓ３０４）。

そして、再生中のエレメンタリストリームの中に、ビデオストリームstream#0が含まれない場合には、処理対象markは無視される（ステップＳ３０４）。

一方、再生中のエレメンタリストリームの中に、ビデオストリームstream#0が含まれる場合には、処理対象markは有効であるとして、その処理対象markに応じた処理が行われる（ステップＳ３０５乃至Ｓ３０８）。

即ち、いまの場合、処理対象markである、第２８図下側のPlayListMark()に記述された２番目のMark()は、そのmark_typeが'Event'になっているからイベントマークであり、従って、プレイヤ制御モジュール２１２は、イベントの発生を表すイベントメッセージと、処理対象markのmark_dataを、スクリプト制御モジュール２１１に供給する（ステップＳ３０５，Ｓ３０７）。そして、スクリプト制御モジュール

２１１では、プレイヤ制御モジュール２１２からのイベントメッセージを割り込み要求として、あらかじめ"SCRIPT.DAT"ファイルに記述された一連の処理を、そのイベントメッセージとともに供給されたmark_dataを引数として行う（ステップＳ３０８）。

以上のように、マーク処理によれば、PlayList()の時間軸上の１つの再生時刻を表すmark_time_stampと、Mark()のタイプを表すmark_typeと、イベントマークの引数となるmark_dataとを含む０以上のMark()を有するPlayListMark()（第７図）を含むPlayList()（第５図）にしたがって再生されているクリップストリームファイルの再生時刻である現在時刻が、mark_time_stampに一致するか否かが判定され、現在時刻が、mark_time_stampに一致する場合に、その一致した現在時刻であるマーク時刻に等しいmark_time_stampを有するMark()が、処理対象markとして認識される。さらに、その処理対象markが有するmark_typeが、イベントを発生させるタイプを表している場合、即ち、処理対象markが、イベントマークである場合、処理対象markが有するmark_dataとイベントメッセージとが通知され、そのmark_dataに応じた処理が実行される。従って、クリップストリームファイルの再生時刻に応じ、mark_dataに応じた処理を実行することが可能となる。

［出力属性の制御処理］

次に、第４１図のフローチャートを参照して、第３０図のステップＳ１２６などで行われる出力属性の制御処理の詳細について説明する。

第３０図のステップＳ１２６で説明したように、プレイヤ制御モジュール２１２は、まず、再生対象の１以上のエレメンタリストリーム、即ち、第３０図のステップＳ１２５で再生すると決定した１以上のエレメンタリストリームそれぞれについて、出力属性が記述されるDy

namicInfo()（第１３図）の数を表すnumber_of_DynamicInfo（第１０図）を調査する。

そして、再生対象の１以上のエレメンタリストリームのすべてについて、number_of_DynamicInfoが0になっている場合、プレイヤ制御モジュール２１２は、特に処理を行わない。

一方、再生対象のエレメンタリストリームについてのnumber_of_DynamicInfoが0でない場合、プレイヤ制御モジュール２１２は、第４１図のフローチャートにしたがった出力属性の制御処理を行う。

従って、ディスク１０１に記録された３つのクリップ情報ファイル"00001.CLP", "00002.CLP", "00003.CLP"が、例えば、第２６図Ａおよび第２６図Ｂに示したようになっている場合に、クリップ情報ファイル"00001.CLP"に対応するクリップストリームファイル"00001.PS"（を再生する１番目のPlayList#0の１番目のPlayItem#0）が再生されるときには、クリップ情報ファイル"00001.CLP"（第２６図Ａおよび第２６図Ｂ）では、クリップストリームファイル"00001.PS"に多重化されている４つのエレメンタリストリームstream#0乃至stream#3のすべてについて、number_of_DynamicInfoが0になっているから、出力属性の制御処理は行われない。

同様に、クリップ情報ファイル"00002.CLP"に対応するクリップストリームファイル"00002.PS"（を再生する１番目のPlayList#0の２番目のPlayItem#1）が再生されるときも、クリップ情報ファイル"00002.CLP"（第２６図Ａおよび第２６図Ｂ）では、クリップストリームファイル"00002.PS"に多重化されている４つのエレメンタリストリームstream#0乃至stream#3のすべてについて、number_of_DynamicInfoが0になっているから、出力属性の制御処理は行われない。

一方、クリップ情報ファイル"00003.CLP"に対応するクリップスト

リームファイル"00003.PS"（を再生する２番目のPlayList#1のPlayItem#0）が再生されるときは、クリップ情報ファイル"00003.CLP"（第２６図Ａおよび第２６図Ｂ）において、クリップストリームファイル"00003.PS"に多重化されている３つのエレメンタリストリームstream#0乃至stream#2のうちの、１番目のエレメンタリストリームであるビデオストリームstream#0と、３番目のエレメンタリストリームであるオーディストリームstream#2について、number_of_DynamicInfoが0でない2と3に、それぞれなっているから、出力属性の制御処理が行われる。

即ち、出力属性の制御処理では、まず最初に、ステップＳ３２０において、プレイヤ制御モジュール２１２は、再生対象のクリップストリームファイルに対応するクリップ情報ファイルClip()（第１０図）に記述されたpts_change_pointを、「DynamicInfo()処理」の属性の時刻である旨とともに、デコード制御モジュール２１４に渡し、デコード制御モジュール２１４は、プレイヤ制御モジュール２１２からの「DynamicInfo()処理」の属性の時刻であるpts_change_pointを受信して、ステップＳ３２１に進む。

ステップＳ３２１では、デコード制御モジュール２１４が、計時部２１４Ａによって計時されている現在時刻が、「DynamicInfo()処理」の属性の時刻であるpts_change_point（のいずれか）に一致したかどうかを判定し、一致していないと判定した場合、ステップＳ３２１に戻る。

また、ステップＳ３２１において、現在時刻が、「DynamicInfo()処理」の属性の時刻（のいずれか）に一致したと判定された場合、デコード制御モジュール２１４は、現在時刻が、「DynamicInfo()処理」の属性の時刻となった旨のメッセージと、現在時刻と一致した「Dy

namicInfo()処理」の属性の時刻（以下、適宜、DynamicInfo時刻という）とを、プレイヤ制御モジュール２１２に供給して、ステップＳ３２２に進む。

ステップＳ３３２では、プレイヤ制御モジュール２１２が、現在時刻が、「DynamicInfo()処理」の属性の時刻となった旨のメッセージと、DynamicInfo時刻とを、デコード制御モジュール２１４から受信し、そのDynamicInfo時刻に一致するpts_change_point（第１０図）とセットになっているDynamicInfo()を、処理対象のDynamicInfo()である処理対象DynamicInfo()として認識して、ステップＳ３２３に進む。

ステップＳ３２３では、プレイヤ制御モジュール２１２は、処理対象DynamicInfo()となっているDynamicInfo()（第１３図）に記述された出力属性を、グラフィクス処理モジュール２１９またはオーディオ出力モジュール２２１に供給して、ステップＳ３２４に進む。

ステップＳ３２４では、グラフィクス処理モジュール２１９またはオーディオ出力モジュール２２１が、直前のステップＳ３２３でプレイヤ制御モジュール２１２から供給された出力属性にしたがって、ビデオデータまたはオーディオデータの出力の制御を、それぞれ開始し、ステップＳ３２１に戻る。

これにより、ビデオデータが、出力属性（表示方式）として記述された、例えばアスペクト比に応じて出力され、あるいは、オーディオデータが、出力属性（出力方式）として記述された、例えば、ステレオまたはデュアル（二ヶ国語）に応じて出力される。

次に、第４２図を参照して、出力属性の制御処理の詳細について、さらに説明する。

即ち、第４２図は、第２６図Ａおよび第２６図Ｂのクリップ情報フ

ァイル"00003.CLP"に記述されているpts_change_pointとDynamicInfo()とのセット（第１０図）を示している。

ここで、上述したように、クリップストリームファイル"00003.PS"に多重化されている３つのエレメンタリストリームstream#0乃至stream#2のうちの、１番目のエレメンタリストリームであるビデオストリームstream#0と、３番目のエレメンタリストリームであるオーディストリームstream#2については、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"において、number_of_DynamicInfoが、それぞれ2と3になっている。従って、クリップ情報ファイル"00003.CLP"において、クリップストリームファイル"00003.PS"の１番目のビデオストリームstream#0については、２セットのpts_change_pointおよびDynamicInfo()が記述されており、３番目のオーディオストリームstream#2については、３セットのpts_change_pointおよびDynamicInfo()が記述されている。

第４２図上側は、クリップストリームファイル"00003.PS"の１番目のビデオストリームstream#0について記述されている２セットのpts_change_pointおよびDynamicInfo()を示しており、第４２図下側は、クリップストリームファイル"00003.PS"の３番目のオーディオストリームstream#2について記述されている３セットのpts_change_pointおよびDynamicInfo()を示している。

なお、第４２図上側では、１番目のビデオストリームstream#0について記述されている２セットのpts_change_pointおよびDynamicInfo()の他に、そのビデオストリームstream#0について、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"に記述されているstream_id(=0xE0), private_stream_id(=0x00), number_of_DynamicInfo(=2)も、図示してある。同様に、第４２図下側でも、３番目のオー

ディオストリームstream#2について記述されている３セットのpts_change_pointおよびDynamicInfo()の他に、そのオーディオストリームstream#2について、第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"に記述されているstream_id(=0xBD), private_stream_id(=0x00), number_of_DynamicInfo(=3)も、図示してある。

第４２図上側において、ビデオストリームstream#0について記述されている２セットのpts_change_pointおよびDynamicInfo()のうちの１セット目では、pts_change_pointが90,000になっており、DynamicInfo()のdisplay_aspect_ratio（第１３図）が'4:3'になっている。さらに、その２セット目では、pts_change_pointが54,090,000になっており、DynamicInfo()のdisplay_aspect_ratioが'16:9'になっている。

一方、第４２図下側において、オーディオストリームstream#2について記述されている３セットのpts_change_pointおよびDynamicInfo()のうちの１セット目では、pts_change_pointが90,000になっており、DynamicInfo()のchannel_assignment（第１３図）が'Dual'になっている。さらに、その２セット目では、pts_change_pointが27,090,000になっており、DynamicInfo()のchannel_assignmentが'Stereo'になっている。また、その３セット目では、pts_change_pointが32,490,000になっており、DynamicInfo()のchannel_assignmentが'Dual'になっている。

例えば、いま、第３０図のステップＳ１２５において、クリップストリームファイル"00003.PS"の、0xE0となっているstream_idで特定される１番目のビデオストリームstream#0と、0xBDとなっているstream_idおよび0x00となっているprivate_stream_idで特定される３番目のオーディオストリームstream#2とが、再生対象のストリームとして

決定されたとする。

この場合、プレイヤ制御モジュール２１２は、0xE0となっているstream_idで特定されるビデオストリームstream#0について記述されている第４２図上側の２セットのpts_change_pointおよびDynamicInfo()と、0xBDとなっているstream_idおよび0x00となっているprivate_stream_idで特定されるオーディオストリームstream#2について記述されている第４２図下側の３セットのpts_change_pointおよびDynamicInfo()との中のpts_change_pointを調査し、初期値を認識する。

即ち、0xE0となっているstream_idで特定されるビデオストリームstream#0について記述されている第４２図上側の２セットのpts_change_pointおよびDynamicInfo()のうちの１セット目では、pts_change_pointが90,000になっている。そして、この90,000という時刻は、ビデオストリームstream#0が多重化されているクリップストリームファイル"00003.PS"に対応する第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"において、クリップストリームファイル"00003.PS"の先頭の時刻を表すpresentation_start_timeに記述されている時刻90,000に一致する。

同様に、0xBDとなっているstream_idおよび0x00となっているprivate_stream_idで特定されるオーディオストリームstream#2について記述されている第４２図下側の３セットのpts_change_pointおよびDynamicInfo()のうちの１セット目では、pts_change_pointが90,000になっている。そして、この90,000という時刻は、オーディオストリームstream#2が多重化されているクリップストリームファイル"00003.PS"に対応する第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"において、クリップストリームファイル"00003.PS"の先頭の時刻を表すpresentation_start_timeに記述されている時刻90,000に

一致する。

プレイヤ制御モジュール２１２は、クリップストリームファイル"00003.PS"の先頭の時刻を表すpresentation_start_timeに記述されている時刻90,000に一致するpts_change_pointを、初期値として認識する。従って、第４２図上側の２セットのpts_change_pointおよびDynamicInfo()のうちの１セット目のpts_change_pointと、第４２図下側の３セットのpts_change_pointおよびDynamicInfo()のうちの１セット目のpts_change_pointが、初期値として認識される。

そして、プレイヤ制御モジュール２１２は、クリップストリームファイル"00003.PS"の再生が開始される前に（第３０図のステップＳ１２６で）、初期値として認識したpts_change_pointとセットになっているDynamicInfo()にしたがって、対応するエレメンタリストリームの出力属性を指示する。

即ち、0xE0となっているstream_idで特定されるビデオストリームstream#0については、第４２図上側で、初期値である90,000になっているpts_change_pointとセットになっているDynamicInfo()において、display_aspect_ratioが'4:3'になっている。この場合、プレイヤ制御モジュール２１２は、display_aspect_ratioが'4:3'になっている旨、即ち、ビデオストリームstream#0が、４：３のアスペクト比のビデオデータである旨の出力属性の情報を、グラフィクス処理モジュール２１９を制御する。

また、0xBDとなっているstream_idおよび0x00となっているprivate_stream_idで特定されるオーディオストリームstream#2については、第４２図下側で、初期値である90,000になっているpts_change_pointとセットになっているDynamicInfo()において、channel_assignmentが'Dual'になっている。この場合、プレイヤ制御モジュール２１２は

、channel_assignmentが'Dual'になっている旨、即ち、オーディオストリームstream#2が、デュアルのオーディオデータである旨の出力属性の情報を、オーディオ出力モジュール２２１に供給する。

ここで、第３０図のステップＳ１２６では、以上のような初期値としてのpts_change_pointを対象とした出力属性の制御処理が行われる。

その後、プレイヤ制御モジュール２１２は、ビデオストリームstream#0についての第４２図上側の２つのpts_change_pointである90,000および54,090,000と、オーディオストリームstream#2についての第４２図下側の３つのpts_change_pointである90,000, 27,090,000、および32,490,000のうちの、初期値90,000以外の時刻である{27,090,000}, {32,490,000}, {54,090,000}を、「DynamicInfo()処理」の属性の時刻である旨とともに、デコード制御モジュール２１４に渡す（ステップＳ３２０）。

デコード制御モジュール２１４は、プレイヤ制御モジュール２１２からの、「DynamicInfo()処理」の属性の時刻{27,090,000}, {32,490,000}, {54,090,000}を受信し、さらに、ビデオストリームstream#0およびオーディオストリームstream#2の再生（クリップストリームファイル"00003.PS"を再生する２番目のPlayList#1のPlayItem#0の再生）の開始後、計時部２１４Ａによって計時されている現在時刻の監視を開始する。

そして、デコード制御モジュール２１４は、現在時刻が、「DynamicInfo()処理」の属性の時刻{27,090,000}, {32,490,000}, {54,090,000}のうちのいずれかに一致した場合、その現在時刻と一致した「DynamicInfo()処理」の属性の時刻であるDynamicInfo時刻を、プレイヤ制御モジュール２１２に供給する（ステップＳ３２１）。

即ち、例えば、現在時刻が、27,090,000になったとすると、デコード制御モジュール２１４は、「DynamicInfo()処理」の属性の時刻のうちの、現在時刻と一致する27,090,000を、DynamicInfo時刻として、プレイヤ制御モジュール２１２に供給する

プレイヤ制御モジュール２１２は、デコード制御モジュール２１４からのDynamicInfo時刻である27,090,000を受信し、ビデオストリームstream#0についての第４２図上側の２つのpts_change_pointと、オーディオストリームstream#2についての第４２図下側の３つのpts_change_pointとの中から、DynamicInfo時刻である27,090,000に一致するpts_change_pointを調査し、その27,090,000に一致するpts_change_pointとセットになっているDynamicInfo()、即ち、オーディオストリームstream#2についての第４２図下側の２番目のDynamicInfo()を、処理対象DynamicInfo()として認識する（ステップＳ３２２）。

処理対象DynamicInfo()が、ビデオストリームについてのDynamicInfo()である場合、プレイヤ制御モジュール２１２は、処理対象DynamicInfo()に記述されている出力属性を、グラフィクス処理モジュール２１９に供給する（ステップＳ３２３）。また、処理対象DynamicInfo()が、オーディオストリームについてのDynamicInfo()である場合、プレイヤ制御モジュール２１２は、処理対象DynamicInfo()に記述されている出力属性を、オーディオ出力モジュール２２１に供給する（ステップＳ３２３）。

グラフィクス処理モジュール２１９は、プレイヤ制御モジュール２１２から出力属性が供給されると、その出力属性にしたがって、ビデオデータの出力の制御を開始する（ステップＳ３２４）。

即ち、グラフィクス処理モジュール２１９は、例えば、プレイヤ制御モジュール２１２からの出力属性が表す、ビデオデータのアスペク

ト比の指示(display_aspect_ratio（第１３図）)と、第１図のビデオ出力端子１２０に接続されたビデオ出力装置のアスペクト比とに基づいて、ビデオ出力モジュール２２０に出力するビデオデータのアスペクト比の変換を行う。

具体的には、例えば、ビデオ出力装置のアスペクト比が16:9である場合において、出力属性としてのビデオデータのアスペクト比の指示が4:3のアスペクト比を表しているときには、グラフィクス処理モジュール２１９は、ビデオ出力モジュール２２０に出力するビデオデータを、横方向にスクイーズ処理し、左右に黒味を入れて出力する。また、例えば、ビデオ出力装置のアスペクト比が4:3である場合において、出力属性としてのビデオデータのアスペクト比の指示が16:9のアスペクト比を表しているときには、グラフィクス処理モジュール２１９は、ビデオ出力モジュール２２０に出力するビデオデータを、縦方向にスクイーズ処理し、上下に黒味を入れて出力する。さらに、例えば、ビデオ出力装置のアスペクト比と、出力属性としてのビデオデータのアスペクト比の指示が表すアスペクト比とが、いずれも、4:3や16:9で、同一である場合、グラフィクス処理モジュール２１９は、ビデオ出力モジュール２２０に出力するビデオデータを、スクイーズ処理することなく、そのまま出力する。

ここで、第４２図上側において、0xE0となっているstream_idで特定されるビデオストリームstream#0について記述されている２セットのpts_change_pointおよびDynamicInfo()によれば、ビデオストリームstream#0の再生開始時である時刻90,000から、時刻54,090,000の直前までは、ビデオストリームstream#0から、４：３のアスペクト比のビデオデータが得られる。そして、時刻54,090,000以後は、ビデオストリームstream#0から、１６：９のアスペクト比のビデオデータが得

られる。

従って、例えば、第１図のビデオ出力端子１２０に接続されたビデオ出力装置のアスペクト比が4:3であるとすると、グラフィクス処理モジュール２１９では、時刻90,000から時刻54,090,000の直前までは、ビデオストリームstream#0から得られる４：３のアスペクト比のビデオデータが、そのまま４：３のアスペクト比のビデオ出力装置に供給されて表示される。

そして、時刻54,090,000以後は、ビデオストリームstream#0から得られる１６：９のアスペクト比のビデオデータが、縦方向にスクイーズ処理され、さらに、上下に黒味が入った４：３のアスペクト比のビデオ信号に変換され、４：３のアスペクト比のビデオ出力装置に供給されて表示される。

一方、オーディオ出力モジュール２２１は、プレイヤ制御モジュール２１２から出力属性が供給されると、その出力属性にしたがって、オーディオデータの出力の制御を開始する（ステップＳ３２４）。

即ち、オーディオ出力モジュール２２１は、例えば、プレイヤ制御モジュール２１２からの出力属性が表す、オーディオデータのチャネル割り当ての指示(channel_assignment（第１３図）)と、ユーザがリモコンを操作することによって入力インターフェース１１５（第１図）を介してプレイヤ制御モジュール２１２から供給される音声出力モードとに基づいて、オーディオデコーダ制御モジュール２１７からのオーディオデータを処理し、オーディオ出力端子１２１（第１図）に出力する。

具体的には、例えば、出力属性が表すオーディオデータのチャネル割り当ての指示が、左チャネルが「主音声」のオーディオデータで、右チャネルの「副音声」のオーディオデータであるデュアル(Dual)（

二ヶ国語）モードを表している場合、オーディオ出力モジュール２２１は、プレイヤ制御モジュール２１２から供給される音声出力モードにしたがって、オーディオデコーダ制御モジュール２１７からのオーディオデータを処理して、オーディオ出力端子１２１に出力する。

即ち、音声出力モードとして、例えば、「主音声」が指定されているときには、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータのうちの左チャネルのオーディオデータを、右チャネルのオーディオデータとしてコピーし、その左チャネルと右チャネルのオーディオデータ（「主音声」のオーディオデータ）を、オーディオ出力端子１２１に出力する。また、音声出力モードとして、「副音声」が指定されているときには、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータのうちの右チャネルのオーディオデータを、左チャネルのオーディオデータとしてコピーし、その左チャネルと右チャネルのオーディオデータ（「副音声」のオーディオデータ）を、オーディオ出力端子１２１に出力する。さらに、音声出力モードとして、「主・副」が指定されているときには、オーディオ出力モジュール２２１は、オーディオデコーダ制御モジュール２１７からのオーディオデータを、そのまま、オーディオ出力端子１２１に出力する。

また、例えば、出力属性が表すオーディオデータのチャネル割り当ての指示が、ステレオ(Stereo)モードを表している場合、オーディオ出力モジュール２２１は、プレイヤ制御モジュール２１２から供給される音声出力モードにかかわらず、オーディオデコーダ制御モジュール２１７からのオーディオデータを、そのまま、オーディオ出力端子１２１に出力する。

ここで、第４２図下側において、0xBDとなっているstream_idおよび0x00となっているprivate_stream_idで特定されるオーディオストリームstream#2について記述されている３セットのpts_change_pointおよびDynamicInfo()によれば、オーディオストリームstream#2の再生開始時である時刻90,000から、時刻27,090,000の直前までは、オーディオストリームstream#2から、デュアルのオーディオデータが得られる。また、時刻27,090,000から、時刻32,490,000の直前までは、オーディオストリームstream#2から、ステレオのオーディオデータが得られ、時刻32,490,000以後は、オーディオストリームstream#2から、デュアルのオーディオデータが得られる。

従って、例えば、音声出力モードとして、「主音声」が指定されているとすると、オーディオ出力出力モジュール２２１では、時刻90,000から、時刻27,090,000の直前までは、オーディオストリームstream#2から得られるデュアルのオーディオデータのうちの左チャネルのオーディオデータが、右チャネルのオーディオデータとしてコピーされ、その左チャネルと右チャネルのオーディオデータが、オーディオ出力端子１２１に出力される。

また、時刻27,090,000から、時刻32,490,000の直前までは、オーディオストリームstream#2から得られるステレオのオーディオデータが、そのまま、オーディオ出力端子１２１に出力される。

そして、時刻32,490,000以後は、オーディオストリームstream#2から得られるデュアルのオーディオデータのうちの左チャネルのオーディオデータが、右チャネルのオーディオデータとしてコピーされ、その左チャネルと右チャネルのオーディオデータが、オーディオ出力端子１２１に出力される。

以上のように、出力属性の制御処理では、クリップストリームファ

イルに多重化されているエレメンタリストリームごとに、そのエレメンタリストリームの再生時刻を表すpts_change_pointと、そのエレメンタリストリームの出力属性を含むDynamicInfo()とのセットを０セット以上含むクリップ情報ファイルClip()（第１０図）の記述に基づき、再生中のエレメンタリストリームの再生時刻が、pts_change_pointに一致するか否かが判定される。そして、再生中のエレメンタリストリームの再生時刻が、pts_change_pointに一致する場合、そのpts_change_pointとセットになっているDynamicInfo()が認識され、その認識されたDynamicInfo()に含まれる出力属性にしたがって、再生中のエレメンタリストリームの出力が制御される。従って、エレメンタリストリームの再生時刻と出力属性に応じて、そのエレメンタリストリームの出力を制御することが可能となる。

［字幕表示制御処理］

次に、第４３図のフローチャートを参照して、字幕ストリームに対応する字幕データの表示を制御する字幕表示制御処理について説明する。

PlayList()（第５図）（のPlayItem()）の再生が開始されると、プレイヤ制御モジュール２１２は、ステップＳ３４１において、グラフィクス処理モジュール２１９に対する字幕データの表示方式の指示を初期化する。即ち、プレイヤ制御モジュール２１２は、字幕データの表示方式をデフォルトの表示方式とするように、グラフィクス処理モジュール２１９を制御する。なお、ステップＳ３４１で行われる表示方式の指示の初期化は、第３０図の１２７で説明した表示方式の指示の初期化に対応する。

ステップＳ３４１の処理後は、ステップＳ３４２に進み、プレイヤ制御モジュール２１２は、ユーザがリモコンを操作することにより入

カインターフェース１１５から、字幕データの表示について、新たな表示方式の指示があったかどうかを判定する。

ステップＳ３４２において、新たな表示方式の指示があったと判定された場合、ステップＳ３４３に進み、プレイヤ制御モジュール２１２は、字幕ストリーム（に対応する字幕データ）を、現在再生しているかどうかを判定する。

ステップＳ３４３において、字幕ストリームが再生されていないと判定された場合、ステップＳ３４２に戻る。

また、ステップＳ３４３において、字幕ストリームが再生されていると判定された場合、ステップＳ３４５に進み、プレイヤ制御モジュール２１２は、新たな表示方式の指示が、デフォルトの表示方式の指示であるかどうかを判定する。ステップＳ３４３において、新たな表示方式の指示が、デフォルトの表示方式の指示であると判定された場合、ステップＳ３４１に戻り、上述したように、プレイヤ制御モジュール２１２は、字幕データの表示方式をデフォルトの表示方式とするように、グラフィクス処理モジュール２１９を制御する。

一方、ステップＳ３４５において、新たな表示方式の指示が、デフォルトの表示方式の指示でないと判定された場合、即ち、新たな表示方式の指示が、例えば、字幕データを拡大や縮小して表示する、あるいは明るさを変えて見やすくする等、デフォルトでない表示方式の指示である場合、ステップＳ３４６に進み、プレイヤ制御モジュール２１２は、現在再生している字幕ストリームが多重化されたクリップストリームファイルに対応するクリップ情報ファイルClip()（第１０図）のStaticInfo()（第１２図）のうちの、現在再生している字幕ストリームについてのStaticInfo()を取得し、ステップＳ３４７に進む。

ステップＳ３４７では、プレイヤ制御モジュール２１２は、ステッ

プＳ３４６で取得したStaticInfo()のconfigurable_flagを判定する。

ステップＳ３４７において、configurable_flagが、字幕データの表示方式の変更を許可しない旨の0になっていると判定された場合、ステップＳ３４８に進み、プレイヤ制御モジュール２１２は、グラフィクス処理モジュール２１９を制御することにより、出力ビデオデータに、字幕データの表示方式を変更することができない旨のエラーメッセージをオーバーレイさせ、ステップＳ３４２に戻る。これにより、エラーメッセージが表示される。

一方、ステップＳ３４７において、configurable_flagが、字幕データの表示方式の変更を許可する旨の1になっていると判定された場合、ステップＳ３４９に進み、プレイヤ制御モジュール２１２は、ユーザがリモコンを操作することにより入力インターフェース１１５から供給された新たな表示方式の指示を、グラフィクス処理モジュール２１９に供給して、ステップＳ３５０に進む。

ステップＳ３５０では、グラフィクス処理モジュール２１９は、字幕デコーダ制御モジュール２１８から供給される字幕データを、直前のステップＳ３４９でプレイヤ制御モジュール２１２から供給された表示方式の指示にしたがって拡大または縮小等あるいは明るさを変える等の処理を開始し、ステップＳ３４２に戻る。これにより、字幕データは、ユーザがリモコンを操作することによって指示した表示方式にしたがった表示サイズや、表示位置、表示色等で表示される。

一方、ステップＳ３４２において、新たな表示方式の指示がなかったと判定された場合、ステップＳ３５１に進み、プレイヤ制御モジュール２１２は、第３１図で説明したPlayItem()の乗り換えが行われたかどうかを判定し、行われていないと判定した場合、ステップＳ３４

２に戻る。

また、ステップＳ３５１において、PlayItem()の乗り換えが行われたと判定された場合、ステップＳ３４１に戻り、上述したように、プレイヤ制御モジュール２１２は、字幕データの表示方式をデフォルトの表示方式とするように、グラフィクス処理モジュール２１９を制御する。即ち、この場合、PlayItem()の乗り換えが行われたときには、字幕データの表示方式は、デフォルトの表示方式に戻される。

以上のように、字幕表示制御処理においては、字幕ストリームのconfigurable_flagが、表示方式の変更を許可する旨の1になっている場合にのみ、その字幕ストリームに対応する字幕データの表示方式が、例えば、ユーザがリモコンを操作することにより入力される表示方式の指示に応じて変更される。

従って、例えば、第２６図Ａおよび第２６図Ｂに示したクリップ情報ファイル"00001.CLP"によれば、対応するクリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームのうちの３本目のエレメンタリストリームである字幕ストリームstream#2についてのconfigurable_flagは、表示方式の変更を許可しない旨の0になっているので、その字幕ストリームstream#2が表示されているときに、ユーザが字幕の表示を変更するようにリモコンを操作しても、その表示は変更されない。

一方、例えば、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームのうちの４本目のエレメンタリストリームである字幕ストリームstream#3についてのconfigurable_flagは、表示方式の変更を許可する旨の1になっているので、その字幕ストリームstream#3が表示されているときに、ユーザが字幕の表示を変更するようにリモコンを操作すると、その操作に応じて、字幕の

表示サイズ等が変更される。

即ち、例えば、いま、第２５図の１番目のPlayList#0の１番目のPlayItem#0にしたがい、クリップストリームファイル"00001.PS"が再生されているとする。また、第２６図Ａおよび第２６図Ｂでクリップ情報ファイル"00001.CLP"について説明したように、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームのうちの、３本目と４本目が字幕ストリームであるが、その３本目の字幕ストリームstream#2と、４本目の字幕ストリームstream#3のうちの、例えば、３本目の字幕ストリームstream#2が、現在再生されているとする。

ユーザが、リモコンを操作することにより、字幕の表示方式の指示を入力すると（ステップＳ３４２）、その表示方式の指示は、入力インターフェース１１５（第１図）からプレイヤ制御モジュール２１２に供給される。プレイヤ制御モジュール２１２は、表示方式の指示が供給されると、再生中の字幕ストリームに対応するStaticInfo()（第１０図）を、クリップ情報ファイルから探し出す（ステップＳ３４６）。

即ち、いまの場合、再生中の字幕ストリームは、クリップストリームファイル"00001.PS"に多重化されている３本目の字幕ストリームstream#2であり、プレイヤ制御モジュール２１２は、対応するクリップ情報ファイル"00001.CLP"から、３本目の字幕ストリームstream#2についてのStaticInfo()を探し出す。

さらに、プレイヤ制御モジュール２１２は、第２６図Ａおよび第２６図Ｂにおいて３本目の字幕ストリームstream#2についてのStaticInfo()に記述されている、0になっているconfigurable_flagを判定し（ステップＳ３４７）、これにより、３本目の字幕ストリームstream#2

については、表示方式の変更が許可されていないことを認識する。

この場合、プレイヤ制御モジュール２１２は、再生中の字幕ストリーム（に対応する字幕データ）が拡大縮小等に対応していないと判断し、グラフィクス処理モジュール２１９を制御することにより、その旨のエラーメッセージを生成させ（ステップＳ３４８）、ビデオデータにオーバーレイして出力させる。

一方、クリップストリームファイル"00001.PS"に多重化されている４本のエレメンタリストリームの３本目の字幕ストリームstream#2と、４本目の字幕ストリームstream#3のうちの、３本目の字幕ストリームstream#2ではなく、４本目の字幕ストリームstream#3が、現在再生されている場合には、ユーザがリモコンを操作することによって表示方式の指示の供給を受けたプレイヤ制御モジュール２１２は、対応するクリップ情報ファイル"00001.CLP"から、４本目の字幕ストリームstream#3についてのStaticInfo()を探し出す。

さらに、プレイヤ制御モジュール２１２は、第２６図Ａおよび第２６図Ｂにおいて４本目の字幕ストリームstream#3についてのStaticInfo()に記述されている、1になっているconfigurable_flagを判定し（ステップＳ３４７）、これにより、４本目の字幕ストリームstream#3については、表示方式の変更が許可されていることを認識する。

この場合、プレイヤ制御モジュール２１２は、再生中の字幕ストリーム（に対応する字幕データ）が拡大縮小等に対応していると判断し、ユーザがリモコンを操作することによって供給された表示方式の指示を、グラフィクス処理モジュール２１９に供給する（ステップＳ３４９）。

これにより、その後、グラフィックス処理制御モジュール２１９は、プレイヤ制御モジュール２１２からの表示方式の指示にしたがい、

字幕デコーダ制御モジュール２１８からの字幕データを拡大または縮小等し、ビデオデコーダ制御モジュール２１６からのビデオデータにオーバーレイして出力する。

なお、プレイヤ制御モジュール２１２は、PlayList()の最初のPlayItem()の再生開始時に、グラフィクス処理モジュール２１９に対する字幕データの表示方式の指示を初期化する（ステップＳ３４１）。即ち、プレイヤ制御モジュール２１２は、字幕データの表示方式をデフォルトの表示方式とするように、グラフィクス処理モジュール２１９を制御する。

さらに、プレイヤ制御モジュール２１２は、PlayItem()の乗り換え時にも、グラフィクス処理モジュール２１９に対する字幕データの表示方式の指示を初期化する（ステップＳ３４１，Ｓ３５１）。

但し、PlayItem()の乗り換え時においては、その後に新たに再生されるPlayItem()にしたがって再生される新たな字幕ストリームについてのconfigurable_flagを調査し、configurable_flagが0である場合には、グラフィクス処理モジュール２１９に対する字幕データの表示方式の指示を初期化し、configurable_flagが1である場合には、グラフィクス処理モジュール２１９に対する表示方式の指示を、PlayItem()の乗り換え前のまま維持するようにすることが可能である。

また、第４３図の字幕表示制御処理では、ユーザがリモコンを操作することにより、新たな表示方式の指示が入力された場合に、その新たな表示方式の指示を、グラフィクス処理モジュール２１９に供給するようにしたが（ステップＳ３４９）、表示方式の指示は、例えば、メモリ１１３（第１図）を構成する不揮発性メモリに記憶し、その不揮発性メモリに記憶された表示方式の指示を、グラフィクス処理モジュール２１９に供給するようにすることが可能である。

即ち、例えば、第１図のディスク装置の初期設定として、不揮発性メモリに、ユーザ設定の表示方式の指示を記憶させておき、ユーザがリモコンを操作することにより、新たな表示方式の指示が入力された場合には、不揮発性メモリに記憶された表示方式の指示を、新たな表示方式の指示に更新する一方、その不揮発性メモリに記憶された表示方式の指示を、グラフィクス処理モジュール２１９に供給するようにすることが可能である。この場合、不揮発性メモリには、前回の再生終了時における表示方式の指示が保持されるので、次回のPlayList()の再生時に、ユーザがリモコンを操作することにより、前回の再生終了時における表示方式の指示を入力しなくても、その表示方式で、字幕データの表示が開始される。

なお、この場合、不揮発性メモリに記憶させる表示方式の指示には、例えば、字幕データを拡大または縮小するときの拡大率または縮小率等が含まれるものとする。

以上のように、字幕表示制御処理によれば、クリップ情報ファイルClip()（第１０図）に含まれる、エレメンタリストリームごとの、そのエレメンタリストリームの再生中に変化しないStaticInfo()のうちの、字幕データのStaticInfo()が取得され、そのStaticInfo()に含まれる、字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表すconfigurable_flagに基づき、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かが判定される。そして、再生中の字幕データの表示をデフォルトの表示方式から変更することが許可されている場合には、字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理、即ち、例えば、字幕データを拡大または縮小、あるいは表示色を変更する等して表示する処理が行われる。従って、字幕データの表示

方式の変更を制御することができる。

［キャプチャ制御処理］

次に、第４４図のフローチャートを参照して、ビデオストリームに対応するビデオデータのキャプチャを制御するキャプチャ制御処理について説明する。なお、第４４図には、キャプチャ制御処理を説明するフローチャートとともに、そのキャプチャ制御処理によってキャプチャされたビデオデータを２次利用する処理の例であるバックグラウンド／スクリーンセーバ処理を説明するフローチャートも、図示してある。

キャプチャ制御処理は、例えば、ユーザがリモコンを操作することにより、ビデオデータのキャプチャを指示するキャプチャ指示が、入力インターフェース１１５（第１図）を介して、プレイヤ制御モジュール２１２に供給されると開始される。

即ち、キャプチャ制御処理では、まず最初に、ステップＳ３７１において、プレイヤ制御モジュール２１２が、ビデオストリームを再生中であるかどうかを判定し、再生中でないと判定した場合、キャプチャ制御処理は終了する。

一方、ステップＳ３７１において、ビデオストリームを再生中であると判定された場合、ステップＳ３７２に進み、プレイヤ制御モジュール２１２は、再生中のビデオストリームに対応するPlayList()（第５図）から、capture_enable_flag_PlayListを取得するとともに、再生中のビデオストリームに対応するクリップ情報ファイルClip()（第１０図）から、capture_enable_flag_Clipを取得する。

ここで、PlayList()におけるcapture_enable_flag_PlayListは、第５図で説明したように、そのPlayList()によって再生されるビデオストリームに対応するビデオデータ（PlayList()に属するビデオデータ

）の２次利用を許可するか否かを表す。また、クリップ情報ファイルClip()におけるcapture_enable_flag_Clipは、第１０図で説明したように、そのクリップ情報ファイルClip()に対応するクリップストリームファイルに格納されているビデオストリームに対応するビデオデータの２次利用を許可するか否かを表す。

ステップＳ３７２の処理後は、ステップＳ３７３に進み、プレイヤ制御モジュール２１２は、直前のステップＳ３７３で取得されたcapture_enable_flag_PlayListとcapture_enable_flag_Clipとに基づき、キャプチャ指示が入力インターフェース１１５（第１図）から入力されたときに再生されていたビデオデータのピクチャのキャプチャの可否を判定する。

ステップＳ３７３において、キャプチャ指示が入力インターフェース１１５から入力されたときに再生されていたビデオデータのピクチャのキャプチャが不可であると判定された場合、即ち、直前のステップＳ３７３で取得されたcapture_enable_flag_PlayListまたはcapture_enable_flag_Clipのうちの少なくとも一方が、ビデオデータの２次利用を許可しない旨の0になっている場合、ステップＳ３７４に進み、プレイヤ制御モジュール２１２は、グラフィクス処理モジュール２１９を制御することにより、ビデオデータのキャプチャが不可である旨のエラーメッセージをオーバーレイさせ、キャプチャ制御処理を終了する。これにより、エラーメッセージが表示される。

一方、ステップＳ３７３において、キャプチャ指示が入力インターフェース１１５から入力されたときに再生されていたビデオデータのピクチャのキャプチャが可能であると判定された場合、即ち、直前のステップＳ３７３で取得されたcapture_enable_flag_PlayListおよびcapture_enable_flag_Clipの両方が、ビデオデータの２次利用を許可

する旨の1になっている場合、ステップＳ３７５に進み、プレイヤ制御モジュール２１２は、キャプチャ指示が入力インターフェース１１５から入力されたときに再生されていたビデオデータのピクチャのキャプチャの指示を、グラフィクス処理モジュール２１９に供給し、ステップＳ３７６に進む。

ステップＳ３７６では、グラフィクス処理モジュール２１９は、プレイヤ制御モジュール２１２からのキャプチャの指示にしたがい、ビデオデコーダ制御モジュール２１６からのビデオデータのピクチャをキャプチャし、メモリ１１３（第１図）に記憶させて、キャプチャ制御処理を終了する。なお、capture_enable_flagが複数ビット構成になっており、使用条件の制約が行われている場合にはこの時点で対応が行われる。すなわちキャプチャした画像の大きさに制限がある場合には、この時点で縮小した画像がキャプチャされる。また使用するアプリケーションに制約がある場合にはその旨を知らせるフラグが同時に記録される。

以上のように、キャプチャ制御処理では、ユーザからのキャプチャ指示があったときに再生されているビデオストリームに対応するPlayList()（第５図）とクリップ情報ファイルClip()（第１０図）それぞれのcapture_enable_flag_PlayListとcapture_enable_flag_Clipとの論理積をとって、その論理積が1である場合、即ち、capture_enable_flag_PlayListとcapture_enable_flag_Clipが、いずれも、２次利用を許可する1になっている場合にのみ、ビデオデータの２次利用が可能であると判断され、キャプチャが行われる。

従って、例えば、第２５図における１番目のPlayList#0の１番目のPlayItem#0にしたがって、ビデオストリームの再生、即ち、クリップストリームファイル"00001.PS"に多重化されたビデオストリームの再

生が行われている場合に、ユーザからのキャプチャ指示があったときには、１番目のPlayList#0におけるcapture_enable_flag_PlayListは1であり、その１番目のPlayItem#0によって再生されるクリップストリームファイル"00001.PS"に対応する第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00001.CLP"におけるcapture_enable_flag_Clipは1であるから、再生中のビデオデータ（クリップストリームファイル"00001.PS"に多重化されたビデオストリームに対応するビデオデータ）の２次利用は可能であると判断され、キャプチャが行われる。

また、例えば、第２５図における１番目のPlayList#0の２番目のPlayItem#1にしたがって、ビデオストリームの再生、即ち、クリップストリームファイル"00002.PS"に多重化されたビデオストリームの再生が行われている場合に、ユーザからのキャプチャ指示があったときには、１番目のPlayList#0におけるcapture_enable_flag_PlayListは1であり、その２番目のPlayItem#1によって再生されるクリップストリームファイル"00002.PS"に対応する第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00002.CLP"におけるcapture_enable_flag_Clipは0であるから、再生中のビデオデータ（クリップストリームファイル"00002.PS"に多重化されたビデオストリームに対応するビデオデータ）の２次利用は不可であると判断され、キャプチャが行われない。

さらに、例えば、第２５図における２番目のPlayList#1のPlayItem#0にしたがって、ビデオストリームの再生、即ち、クリップストリームファイル"00003.PS"に多重化されたビデオストリームの再生が行われている場合に、ユーザからのキャプチャ指示があったときには、２番目のPlayList#1におけるcapture_enable_flag_PlayListは0であり、２番目のPlayList#1のPlayItem#0によって再生されるクリップストリームファイル"00003.PS"に対応する第２６図Ａおよび第２６図Ｂの

クリップ情報ファイル"00003.CLP"におけるcapture_enable_flag_Clipは1であるから、再生中のビデオデータ（クリップストリームファイル"00003.PS"に多重化されたビデオストリームに対応するビデオデータ）の２次利用は不可であると判断され、キャプチャは行われない。

なお、この場合、２番目のPlayList#1におけるcapture_enable_flag_PlayListが0であることが確認された時点で、ビデオデータの２次利用は不可であると判断することができるので、２番目のPlayList#1のPlayItem#0によって再生されるクリップストリームファイル"00003.PS"に対応する第２６図Ａおよび第２６図Ｂのクリップ情報ファイル"00003.CLP"におけるcapture_enable_flag_Clipの確認は省略することができる。

キャプチャ制御処理によってキャプチャされ、メモリ１１３に記憶されたピクチャは、バックグラウンド／スクリーンセーバ処理において２次利用することができる。

バックグラウンド／スクリーンセーバ処理は、例えば、プレイヤ制御モジュール２１２が動作しているが、エレメンタリストリームの再生が行われていない状態、即ち、ディスクドライブ１０２（第１図）にディスク１０１が挿入されていない状態、あるいはエレメンタリストリームの再生が終了した状態となったときなどに行われる。

即ち、バックグラウンド／スクリーンセーバ処理では、ステップＳ３８１において、プレイヤ制御モジュール２１２は、キャプチャ制御処理によってメモリ１１３に記憶されたピクチャを表示するように、グラフィクス処理モジュール２１９を制御する。グラフィクス処理モジュール２１９は、プレイヤ制御モジュール２１２からの制御にしたがい、キャプチャ制御処理によってメモリ１１３に記憶されたピクチャを表示させる。

ここで、グラフィクス処理モジュール２１９において、メモリ１１３に記憶されたピクチャを静止画で表示させれば、いわゆる壁紙(バックグラウンド)が実現され、一定周期で拡大や縮小、移動等しながら表示させれば、スクリーンセーバーが実現される。また、キャプチャ制御処理によってメモリ１１３に記憶されたピクチャの表示を行うバックグラウンド／スクリーンセーバ処理は、プレイヤ制御モジュール２１２ではなく、他の独立したアプリケーションによって行うことが可能である。

また、このときメモリ１１３に記憶されたピクチャに使用制限を表すフラグが付加されている場合にはその制限に従う。

以上のように、ビデオアクセスユニット単位より大きな単位の、例えば、PlayList()やPlayItem()に対応するビデオデータの２次利用を許可するか否かを表す、再生中のビデオデータに対するcapture_enable_flag_PlayListやcapture_enable_flag_Clipが取得され、そのcapture_enable_flag_PlayListやcapture_enable_flag_Clipに基づき、再生中のビデオデータの２次利用が許可されているか否かが判定される。そして、再生中のビデオデータの２次利用が許可されていると判定された場合、再生中のビデオデータがキャプチャされ、そのキャプチャされたビデオデータを利用したバックグラウンド／スクリーンセーバ処理が実行される。従って、ビデオデータの２次利用の制御が可能となる。

なお、第４４図のキャプチャ制御処理では、PlayList()（第５図）において、capture_enable_flag_PlayListを設けるとともに、PlayItem()によって再生されるクリップストリームファイルに対応するクリップ情報ファイルClip()（第１０図）において、capture_enable_flag_Clipを設け、そのcapture_enable_flag_PlayListとcapture_enable

_flag_Clipとの両方を用いて、２次利用の許可（可否）を判定するようにしたが、PlayList()（第５図）において、capture_enable_flag_PlayListを設けるだけか、または、PlayItem()によって再生されるクリップストリームファイルに対応するクリップ情報ファイルClip()（第１０図）において、capture_enable_flag_Clipを設けるだけにして、capture_enable_flag_PlayListまたはcapture_enable_flag_Clipの一方だけを用いて、２次利用の可否を判定するようにすることも可能である。

また、第４４図のキャプチャ制御処理では、ステップＳ３７６において、グラフィクス処理モジュール２１９が、プレイヤ制御モジュール２１２からのキャプチャの指示にしたがい、ビデオデコーダ制御モジュール２１６からのビデオデータのピクチャ、即ち、１つのピクチャだけをキャプチャするようにしたが、その他、複数のピクチャをキャプチャすることも可能である。つまり、ビデオデコーダ制御モジュール２１６が出力する時系列の複数のピクチャ（動画としての複数のピクチャのシーケンス）をキャプチャすることが可能である。この場合、一度にキャプチャされるピクチャの枚数は、例えば、あらかじめ決めておくことができる。あるいは、capture_enable_flag_PlayListやcapture_enable_flag_Clipのビットを拡張して、そのcapture_enable_flag_PlayListやcapture_enable_flag_Clipに、一度にキャプチャ可能なピクチャの枚数の情報を含めるようにしても良い。

さらに、上述の場合には、ビデオデータの２次利用を許可するか否かの利用許可情報(capture_enable_flag_PlayList, capture_enable_flag_Clip)を、PlayList()や、クリップ情報ファイルClip()に記述し、その利用許可情報によって、PlayList()によって再生されるビデオデータ全体や、クリップ情報ファイルClip()に対応するクリップスト

リームファイルに多重化されたビデオストリームに対応するビデオデータ全体についての２次利用の可否を判定するようにしたが、利用許可情報は、その他の任意の単位のビデオデータについて記述し、その利用許可情報によって、任意の単位のビデデータについての２次利用の可否を判定することが可能である。

即ち、第４５図は、利用許可情報が配置されたprivate_stream2_PES_payload()のシンタクスを示しており、第４６図は、利用許可情報が配置されたau_information()のシンタクスを示している。

なお、第４５図のprivate_stream2_PES_payload()は、video_stream_idの直前に、利用許可情報としてのcapture_enable_flag_ps2が配置されている他は、第２３図における場合と同様に構成されている。第４６図のau_information()も、pic_struct_copyの直前に、利用許可情報としてのcapture_enable_flag_AUが配置されている他は、第２４図における場合と同様に構成されている。

第４５図のprivate_stream2_PES_payload()に配置されたcapture_enable_flag_ps2は、そのprivate_stream2_PES_payload()を含むprivate_stream_2のPES_packet()から、次のprivate_stream_2のPES_packet()の直前までに配置されるビデオストリームに対応するビデオデータの２次利用を許可するか否かを表す。従って、第４５図のprivate_stream2_PES_payload()に配置されたcapture_enable_flag_ps2によれば、あるデコード開始可能点から次のデコード開始可能点までの間のビデオデータについて、その２次利用を許可するか否かを判定することができる。

また、第４６図のau_information()に配置されたcapture_enable_flag_AUは、そのcapture_enable_flag_AUに対応するビデオアクセスユニットのビデオデータの２次利用を許可するか否かを表す。従って、

第４６図のau_information()に配置されたcapture_enable_flag_AUによれば、ビデオアクセスユニット単位のビデオデータについて、即ち、ピクチャ単位で、その２次利用を許可するか否かを判定することができる。

ここで、PlayList()（第５図）における利用許可情報としてのcapture_enable_flag_PlayList、クリップ情報ファイルClip()（第１０図）における利用許可情報としてのcapture_enable_flag_Clip、private_stream2_PES_payload()（第４５図）における利用許可情報としてのcapture_enable_flag_ps2, au_information()（第４６図）における利用許可情報としてのcapture_enable_flag_AUは、そのうちの２以上を重複して採用することが可能であり、この場合、あるビデオデータのピクチャの２次利用の可否は、その重複して採用される２以上の利用許可情報の論理積等に基づいて判定することができる。

また、第４６図のau_information()が配置される第２３図または第４５図のprivate_stream2_PES_payload()を含むprivate_stream_2のPES_packet()は、第３６図のステップＳ２１１で説明したように、バッファ制御モジュール２１５（第３図）のビデオ読み出し機能部２３３が、バッファ２１５Ａに記憶されたプログラムストリーム中から探索する。従って、capture_enable_flag_ps2が配置された第４５図のprivate_stream2_PES_payload()や、capture_enable_flag_AUが配置された第４６図のau_information()を採用する場合には、プレイヤ制御モジュール２１２は、ビデオデータの２次利用の可否を判定するにあたって、capture_enable_flag_ps2やcapture_enable_flag_AUを、ビデオ読み出し機能部２３３に問い合わせる必要がある。

なお、本実施の形態では、上述した一連の処理を、ソフトウェアによって行うこととしたが、上述した一連の処理は、専用のハードウェ

アにより行うこともできる。

また、本実施の形態では、ビデオデコーダ１１６（第１図）として、ハードウェアデコーダを採用することとしたが、ビデオデコーダ１１６としては、ソフトウェアデコーダを採用することも可能である。オーディオデコーダ１１７（第１図）についても、同様である。

さらに、本実施の形態では、字幕デコーダとして、ソフトウェアデコーダを採用することとしたが、字幕デコーダとしては、ハードウェアデコーダを採用することも可能である。

請 求 の 範 囲

1．データ記録媒体に記録されている記録データを処理するデータ処理装置において、

前記記録データは、

少なくとも字幕データが多重化された多重化データと、

前記多重化データに関するメタデータと

を含み、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得手段と、

前記取得手段において取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定手段と、

前記判定手段において、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理手段と

を備えることを特徴とするデータ処理装置。

2．前記表示処理手段は、前記字幕データの拡大、縮小、または表示位置の変更の指示にしたがって、その字幕データの表示処理を行う

ことを特徴とする請求の範囲1に記載のデータ処理装置。

3．データ記録媒体に記録されている記録データを処理するデータ処

理方法において、

前記記録データは、

少なくとも字幕データが多重化された多重化データと、

前記多重化データに関するメタデータと

を含み、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得ステップと、

前記取得ステップにおいて取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップと、

前記判定ステップにおいて、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップと

を含むことを特徴とするデータ処理方法。

４．データ記録媒体に記録されている記録データを処理するデータ処理を、コンピュータに行わせるプログラムにおいて、

前記記録データは、

少なくとも字幕データが多重化された多重化データと、

前記多重化データに関するメタデータと

を含み、　前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得ステップと、

前記取得ステップにおいて取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップと、

前記判定ステップにおいて、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップと

を含むことを特徴とするプログラム。

5．データ記録媒体に記録されている記録データを処理するデータ処理を、コンピュータに行わせるプログラムが記録されているプログラム記録媒体において、

前記記録データは、

少なくとも字幕データが多重化された多重化データと、

前記多重化データに関するメタデータと

を含み、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォ

ルトの表示方式から変更することを許可するか否かを表す許可情報を含み、

前記字幕データが再生されているときに、その字幕データの前記不変情報を取得する取得ステップと、

前記取得ステップにおいて取得された前記不変情報に含まれる前記許可情報に基づき、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されているか否かを判定する判定ステップと、

前記判定ステップにおいて、再生中の前記字幕データの表示をデフォルトの表示方式から変更することが許可されていると判定された場合、前記字幕データの表示方式の変更の指示にしたがって、その字幕データの表示処理を行う表示処理ステップと

を含むことを特徴とするプログラムが記録されているプログラム記録媒体。

6．少なくとも字幕データが多重化された多重化データと、

前記多重化データに関するメタデータと

を含む記録データが記録されており、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、

前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含む

ことを特徴とするデータ記録媒体。

7．少なくとも字幕データが多重化された多重化データと、

前記多重化データに関するメタデータと

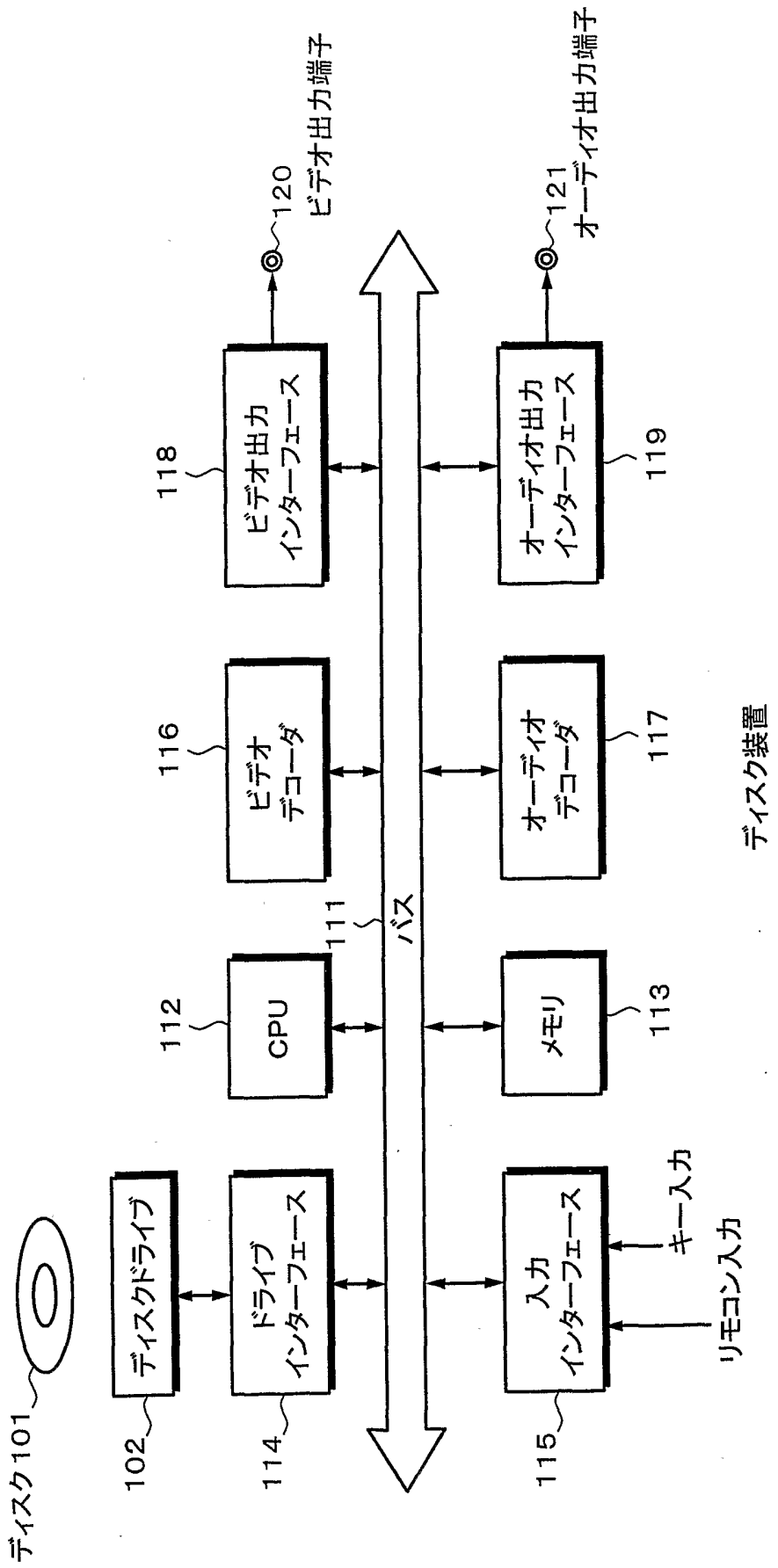
を含むデータのデータ構造において、

前記メタデータは、前記多重化データに多重化されているデータごとに、そのデータの再生中に変化しない不変情報を含み、

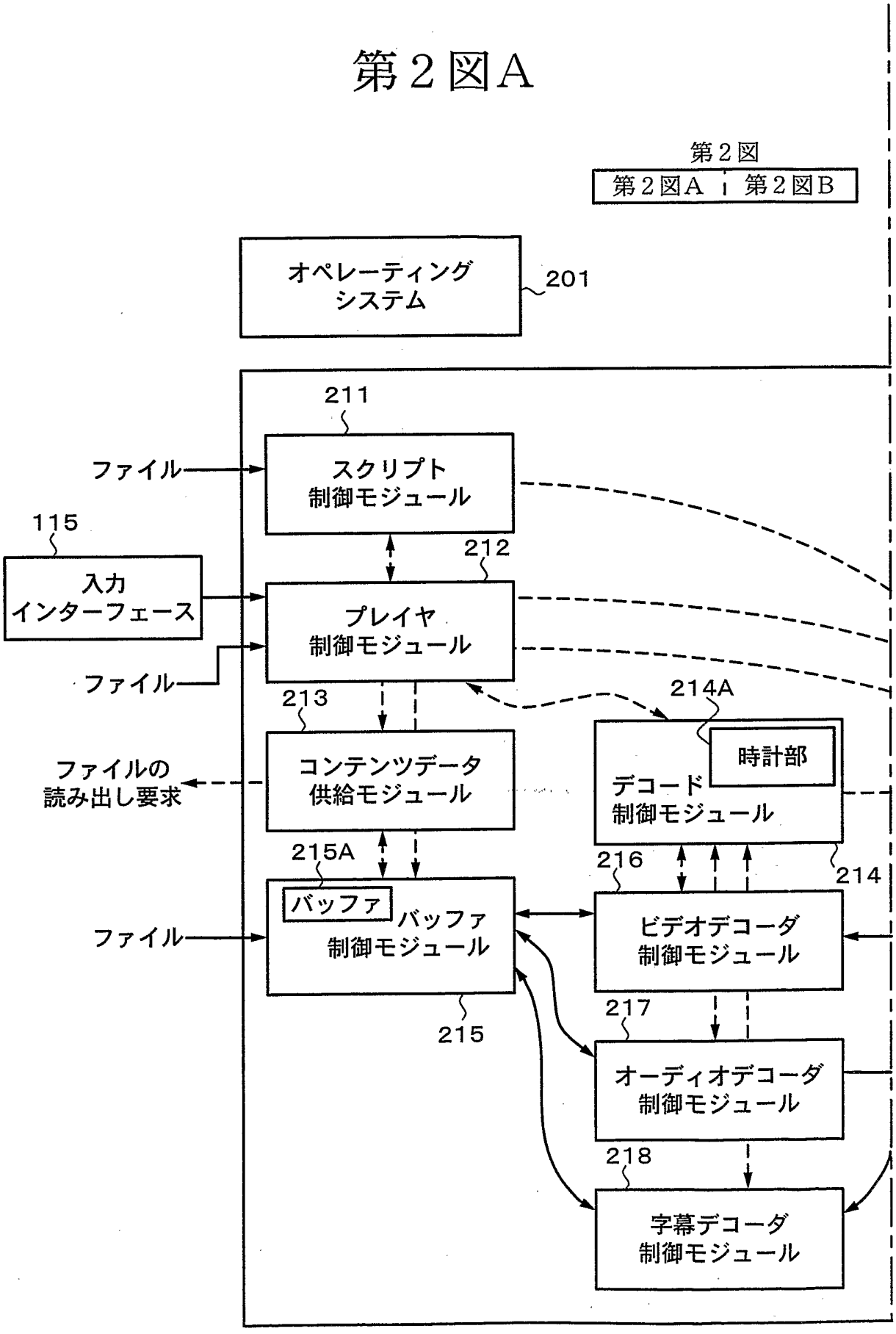
前記字幕データの前記不変情報は、その字幕データの表示をデフォルトの表示方式から変更することを許可するか否かを表す許可情報を含む

ことを特徴とするデータ構造。

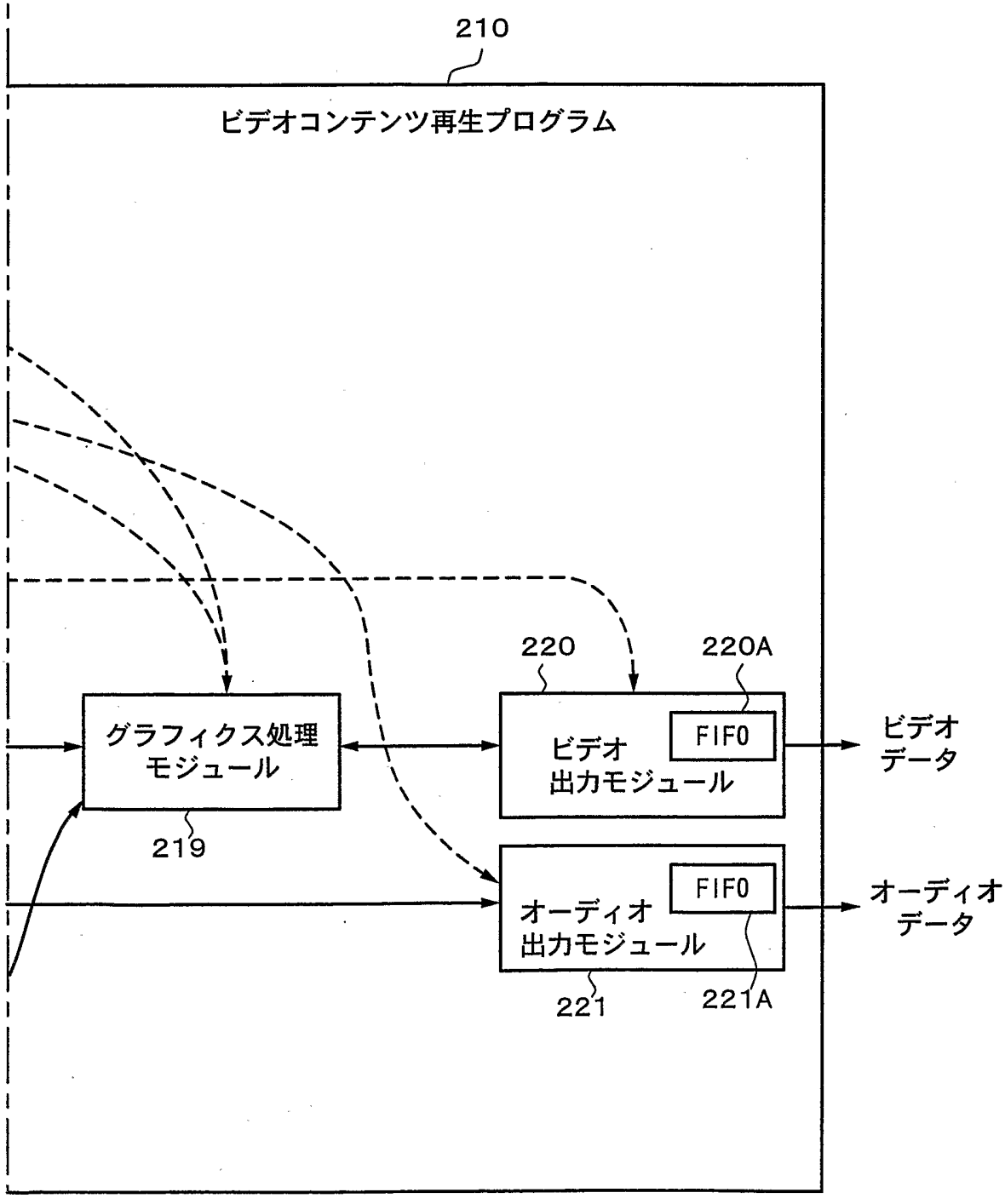
第１図


ディスク101
ディスクドライブ
102
ドライブインターフェース
114
CPU
112
ビデオデコーダ
116
ビデオ出力インターフェース
118
ビデオ出力端子
120
バス
111
メモリ
113
オーディオデコーダ
117
オーディオ出力インターフェース
119
オーディオ出力端子
121
入力インターフェース
115
キー入力
リモコン入力
ディスク装置

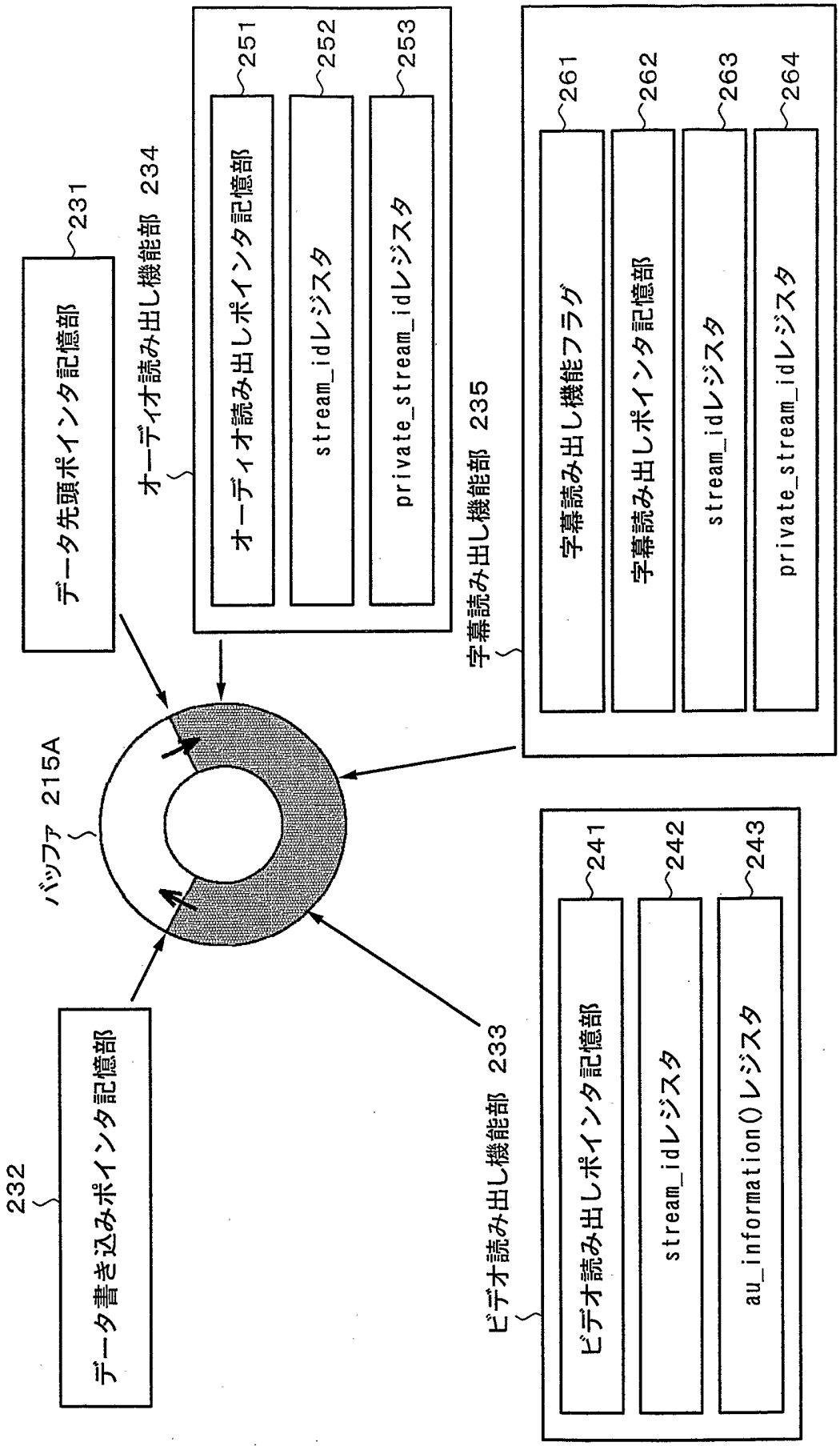
第２図Ａ


第２図
第２図Ａ
第２図Ｂ
オペレーティング
システム
201
211
スクリプト
制御モジュール
ファイル
115
入力
インターフェース
212
プレイヤ
制御モジュール
ファイル
213
214A
時計部
コンテンツデータ
供給モジュール
ファイルの
読み出し要求
デコード
制御モジュール
215A
216
214
バッファ
バッファ
制御モジュール
ファイル
ビデオデコーダ
制御モジュール
217
215
オーディオデコーダ
制御モジュール
218
字幕デコーダ
制御モジュール

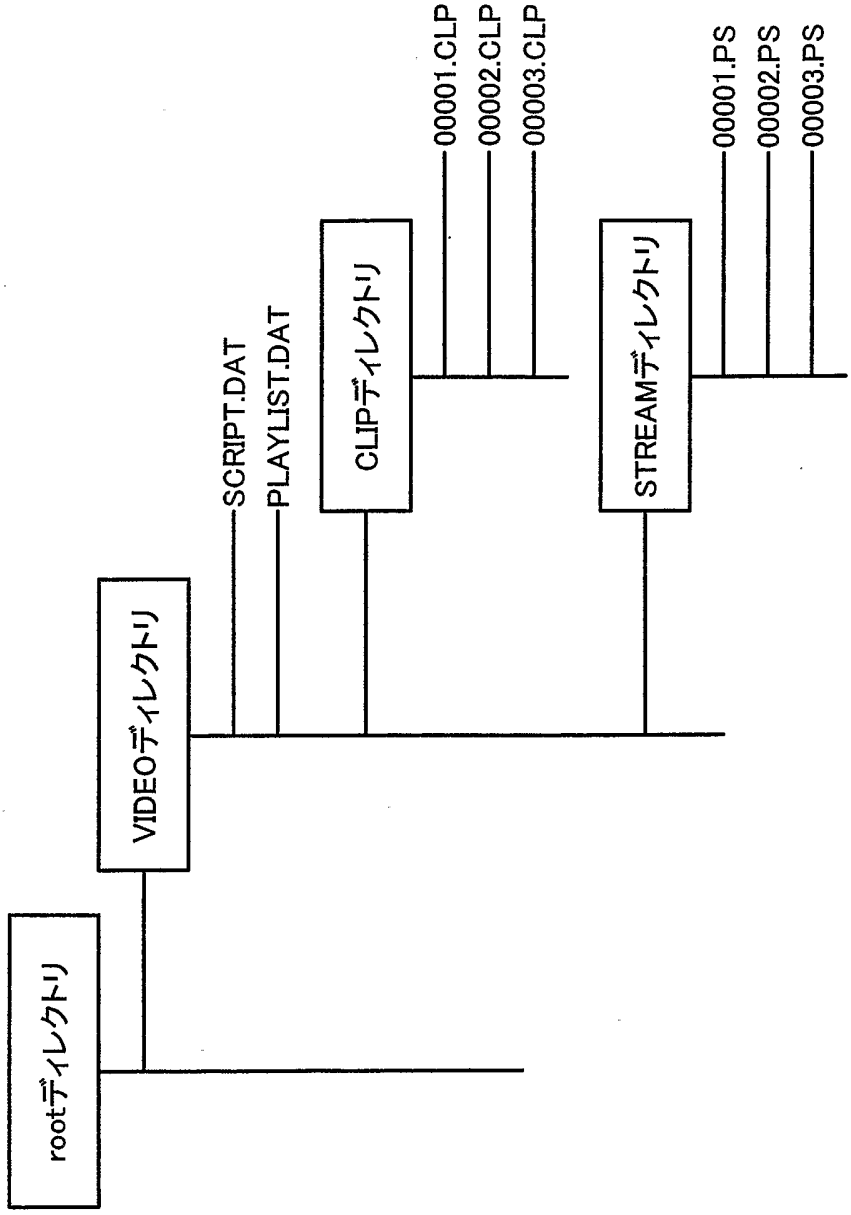
第２図Ｂ


210
ビデオコンテンツ再生プログラム
220
220A
グラフィクス処理
モジュール
FIFO
ビデオ
出力モジュール
ビデオ
データ
219
FIFO
オーディオ
出力モジュール
オーディオ
データ
221
221A

第３図


データ先頭ポインタ記憶部
231
オーディオ読み出し機能部 234
オーディオ読み出しポインタ記憶部
251
stream_idレジスタ
252
private_stream_idレジスタ
253
字幕読み出し機能部 235
字幕読み出し機能フラグ
261
字幕読み出しポインタ記憶部
262
stream_idレジスタ
263
private_stream_idレジスタ
264
バッファ 215A
データ書き込みポインタ記憶部
232
ビデオ読み出し機能部 233
ビデオ読み出しポインタ記憶部
241
stream_idレジスタ
242
au_information()レジスタ
243

第４図


rootディレクトリ
VIDEOディレクトリ
SCRIPT.DAT
PLAYLIST.DAT
CLIPディレクトリ
00001.CLP
00002.CLP
00003.CLP
STREAMディレクトリ
00001.PS
00002.PS
00003.PS

第５図

シンタクス	ビット数	ニーモニック
“PLAYLIST.DAT”{		
name_length	8	uimsbf
name_string	8x255	bslbf
number_of_PlayLists	16	uimsbf
for(i=0;i<number_of_PlayLists;i++){		
PlayList(){ // A PlayList()		
PlayList_data_length	32	uimsbf
// 属性情報		
reserved_for_word_alignment	15	bslbf
capture_enable_flag_PlayList	1	bslbf
PlayList_name_length	8	uimsbf
PlayList_name_string	8*255	bslbf
//		
number_of_PlayItems	16	uimsbf
for(i=0;i<number_of_PlayItems;i++){		
PlayItem()		
}		
PlayListMark()		
}		
}		
}		

第６図

シンタクス	ビット数	ニーモニック
PlayItem(){		
length	16	uimsbf
Clip_Information_file_name_length	16	uimsbf
Clip_Information_file_name	8*Clip_Information_file_name_length	bslbf uimsbf uimsbf
IN_time	32	
OUT_time	32	
}		

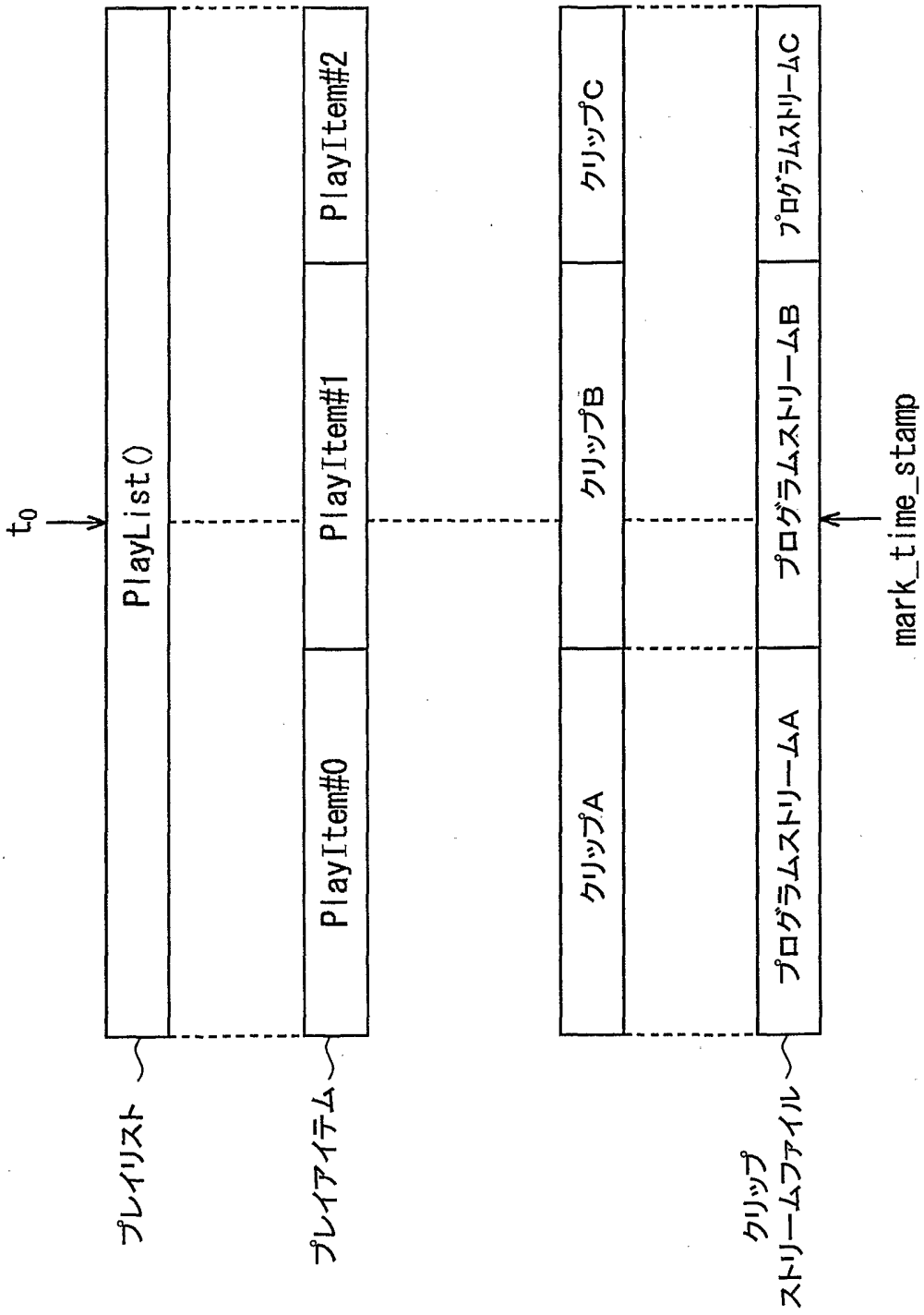
第７図

シンタクス	ビット数	ニーモニック
PlayListMark(){		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for(i=0;i < number_of_PlayList_marks;i++){		
Mark(){		
mark_type	8	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_stream_id	8	uimsbf
entry_ES_private_stream_id	8	uimsbf
mark_data	32	bslbf
mark_name_string	8*24	bslbf
}		
}		
}		

第８図

mark_type	ストリームコーディング
0	予約
1	"Chapter"マーク
2	"Index"マーク
3	"Event"マーク
4～255	予約

第９図


t0
PlayList()
PlayItem#0
PlayItem#1
PlayItem#2
クリップA
クリップB
クリップC
プログラムストリームA
プログラムストリームB
プログラムストリームC
mark_time_stamp
プレイリスト
プレイアイテム
クリップ
ストリームファイル

第１０図

シンタクス	ビット数	ニーモニック
XXXXX.CLP{		
presentation_start_time	32	uimsbf
presentation_end_time	32	uimsbf
reserved_for_word_alignment	7	bslbf
capture_enable_flag_Clip	1	bslbf
number_of_streams	8	uimsbf
for(i=0;i < number_of_streams;i++){		
StreamInfo(){		
length	16	uimsbf
stream_id	8	uimsbf
private_stream_id	8	uimsbf
StaticInfo()		
reserved_for_word_alignment	8	bslbf
number_of_DynamicInfo	8	uimsbf
for(j=0;j < number_of_DynamicInfo;j++){		
pts_change_point	32	uimsbf
DynamicInfo()		
}		
}		
}		
EP_map()		
}		

第１１図

エレメンタリ ストリーム種類	stream_id	private_stream_id
ビデオ	0xE0～0xEF	(なし)
ATRACオーディオ	0xBD	0x00～0x0F
LPCMオーディオ	0xBD	0x10～0x1F
字幕	0xBD	0x80～0x9F

第１２図

シンタクス	ビット数	ニーモニック
StaticInfo() {		
if(stream == VIDEO){		
reserved_for_word_alignment	16	bslbf
picture_size	4	uimsbf
frame_rate	4	uimsbf
reserved_for_word_alignment	7	bslbf
cc_flag	1	bslbf
}else if (stream == AUDIO) {		
audio_language_code	16	bslbf
channel_configuration	8	uimsbf
reserved_for_word_alignment	3	bslbf
lfe_existence	1	bslbf
sampling_frequency	4	uimsbf
}else if (stream == SUBTITLE) {		
subtitle_language_code	16	bslbf
reserved_for_word_alignment	15	bslbf
configurable_flag	1	uimsbf
}		
}		

第１３図

シンタクス	ビット数	ニーモニック
DynamicInfo(i,j){		
reserved_for_word_alignment	8	bslbf
if(stream == VIDEO){		
reserved_for_word_alignment	4	bslbf
display_aspect_ratio	4	uimsbf
}else if(stream == AUDIO){		
reserved_for_word_alignment	4	bslbf
channel_assignment	4	uimsbf
}else if(stream == SUBTITLE){		
reserved_for_word_alignment	8	bslbf
}		
}		

第１４図

シンタクス	ビット数	ニーモニック
EP_map(){		
reserved_for_word_alignment	8	bslbf
number_of_stream_id_entries	8	uimsbf
for(k=0;k<number_of_stream_id_entries;k++){		
stream_id	8	bslbf
private_stream_id	8	bslbf
number_of_EP_entries	32	uimsbf
for(i=0;i< number_of_EP_entries;i++){		
PTS_EP_start	32	uimsbf
RPN_EP_start	32	uimsbf
}		
}		
}		

第１５図Ａ

テーブル2-31, プログラムストリーム		
シンタクス	ビット数	ニーモニック

MPEG2_program_stream() {
　　do {
　　　　pack()
　　}while(nextbits()== pack_start_code)
　　MPEG_program_end_code
}
	32	bslbf

テーブル2-32, プログラムストリームパック		
シンタクス	ビット数	ニーモニック

pack() {
　　pack_header()
　　while(nextbits()==packet_start_code_prefix) {
　　PES_packet()
　　}
}		

第１５図Ｂ

テーブル2-33, プログラムストリームパックヘッダ

シンタクス	ビット数	ニーモニック
pack_header() {		
pack_start_code	32	bslbf
'01'	2	bslbf
system_clock_reference_base[32..30]	3	bslbf
marker_bit	1	bslbf
system_clock_reference_base[29..15]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_base[14..0]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_extension	9	uimsbf
marker_bit	1	bslbf
program_mux_rate	22	uimsbf
marker_bit	1	bslbf
marker_bit	1	bslbf
reserved	5	bslbf
pack_stuffing_length	3	uimsbf
for(i=0;i<pack_stuffing_length;i++) {		
stuffing_byte	8	bslbf
}		
if(nextbits()==system_header_start_code) {		
system_header()		
}		
}		

第１６図Ａ

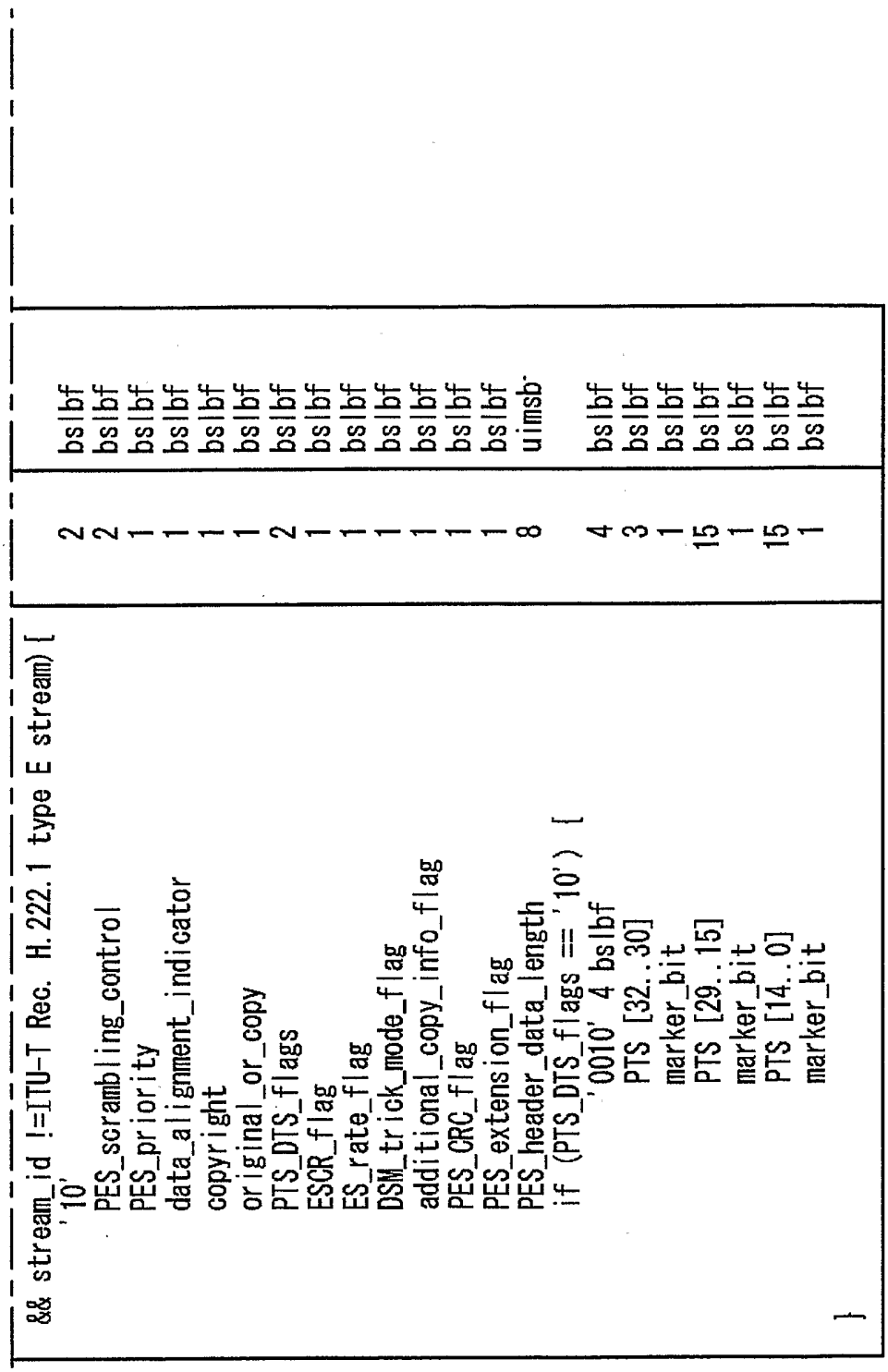
第１６図

第１６図Ａ
第１６図Ｂ

テーブル2-17, PES パケット

シンタクス	ビット数	ニーモニック
PES_packet() {		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if(stream_id !=program_stream_map		
&& stream_id !=padding_stream		
&& stream_id !=private_stream_2		
&& stream_id !=ECM		
&& stream_id !=EMM		
&& stream_id !=program_stream_directory		
&& stream_id !=DSMCC_stream		

第16図B



&& stream_id !=ITU-T Rec. H.222.1 type E stream){		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
copyright	1	bslbf
original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf
PES_header_data_length	8	uimsb
if (PTS_DTS_flags == '10') {		
'0010' 4 bslbf	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		

第１７図Ａ

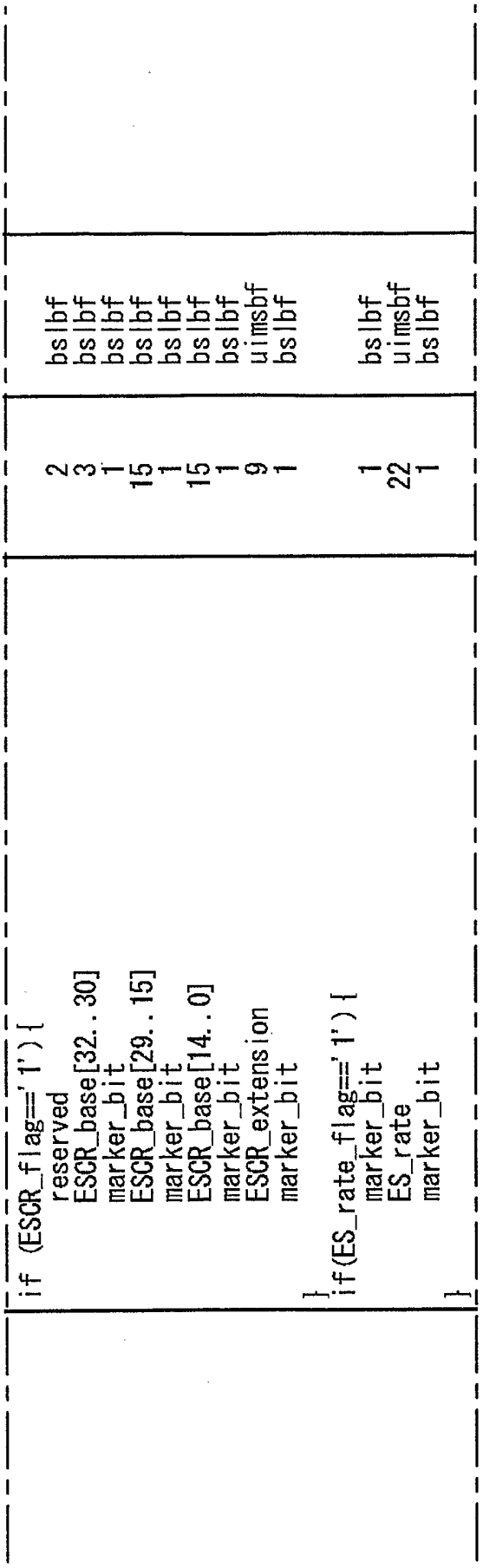
第１７図

第１７図Ａ
第１７図Ｂ
第１７図Ｃ

テーブル2-17, PES パケット （続き）

シンタクス	ビット数	ニーモニック
if(PTS_DTS_flags=='11'){		
'0011'	4	bslbf
PTS[32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS [32..30]	3	bslbf
marker_bit	1	bslbf
DTS [29..15]	15	bslbf
marker_bit	1	bslbf
DTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		

第17図B


if (ESCR_flag=='1'){
reserved 2 bslbf
ESCR_base[32..30] 3 bslbf
marker_bit 1 bslbf
ESCR_base[29..15] 15 bslbf
marker_bit 1 bslbf
ESCR_base[14..0] 15 bslbf
marker_bit 1 bslbf
ESCR_extension 9 uimsbf
marker_bit 1 bslbf
}
if(ES_rate_flag=='1'){
marker_bit 1 bslbf
ES_rate 22 uimsbf
marker_bit 1 bslbf
}

第１７図Ｃ

if(DSM_trick_mode_flag=='1'){		
trick_mode_control	3	uimsbf
if(trick_mode_control==fast_forward){		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if(trick_mode_control==slow_motion){		
rep_cntrl	5	uimsbf
}		
else if(trick_mode_control==freeze_frame){		
field_id	2	uimsbf
reserved	3	bslbf
}		
else if(trick_mode_control==fast_reverse){		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
else if(trick_mode_control==slow_reverse){		
rep_cntrl	5	uimsbf
}		
else		
reserved	5	bslbf
}		
if(additional_copy_info_flag=='1'){		
marker_bit	1	bslbf
additional_copy_info	7	bslbf
}		
if(PES_CRC_flag=='1'){		
previous_PES_packet_CRC	16	bslbf
}		

第１８図Ａ

第１８図

第１８図Ａ
第１８図Ｂ

テーブル2-17, PES パケット （終わり）

シンタクス	ビット数	ニーモニック
if(PES_extension_flag=='1'){		
PES_private_data_flag	1	bslbf
pack_header_field_flag	1	bslbf
program_packet_sequence_counter_flag	1	bslbf
P-STD_buffer_flag	1	bslbf
reserved	3	bslbf
PES_extension_flag_2	1	bslbf
if(PES_private_data_flag=='1'){		
PES_private_data	128	bslbf
}		
if (pack_header_field_flag == '1') {		
pack_field_length	8	uimsbf
pack_header ()		
}		
if (program_packet_sequence_counter_flag == '1') {		
marker_bit	1	bslbf
program_packet_sequence_counter	7	uimsbf
marker_bit	1	bslbf
MPEG1_MPEG2_identifier	1	bslbf
original_stuff_length	6	uimsbf
}		
if(P-STD_buffer_flag == '1') {		
'01'	2	bslbf
P-STD_buffer_scale	1	bslbf
P-STD_buffer_size	13	uimsbf
}		

第１８図Ｂ

if(PES_extension_flag_2 == '1'){		
marker_bit	1	bslbf
PES_extension_field_length	7	uimsbf
for(i=0;i<PES_extension_field_length;i++){		
reserved	8	bslbf
}		
}		
}		
for(i=0;i<N1;i++){		
stuffing_byte	8	bslbf
}		
for(i=0;i<N2;i++){		
PES_packet_data_byte	8	bslbf
}		
}		
else if(stream_id == program_stream_map		
||stream_id == private_stream_2		
||stream_id == ECM		
||stream_id == EMM		
||stream_id == program_stream_directory		
||stream_id == DSMCC_stream		
||stream_id == ITU-T Rec.H.222.1 type E stream){		
for(i=0;i<PES_packet_length;i++){		
PES_packet_data_byte	8	bslbf
}		
}		
else if(stream_id == padding_stream){		
for(i=0;i<PES_packet_length;i++){		
padding_byte	8	bslbf
}		
}		
}		

第１９図Ａ

第１９図

第１９図Ａ
第１９図Ｂ

テーブル2-18, stream_id アサインメント		
stream_id	注釈	ストリームコーディング
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100 … 1111 1110		reserved data stream
1111 1111	4	program_stream_directory

第１９図Ｂ

記号Xは値「0」または「1」のいずれも可能であり、ストリーム番号はXで与えられる。
注1: program__stream__mapのPESパケットは2. 5. 4. 1で規定の独自のシンタックスを使用する。
注2: private__stream__1およびISO／IEC__13552__streamのPESパケットはITU－T Rec. H.262であるISO／IEC13818－2ビデオ・ストリームおよびISO/IEC13818－3オーディオ・ストリームで規定されたものと同じPESパケットのシンタックスに従う。
注3: private__stream__2、ECM__STREAM、およびEMM__streamのPESパケットはPES__packet__lengthフィールドの後に何らシンタックスが指定されないことを除きprivate__stream__1に類似。
注4: program__stream__directoryのPESパケットは2. 5. 5で規定の独自のシンタックスを使用する。
注5: DSM—CC__streamはISO/IEC13818－6で規定された独自のシンタックスを使用する。
注6: このstream__idは表2－29のstream__type 0x09と組み合わせて使用する。
注7: このstream__idはプログラム・ストリームまたはISO/IEC11172－1システム・ストリームのデータ用のトランスポート・ストリームのPESパケットにのみ使用する(2. 4. 3. 7参照)。

第２０図

stream_id	ストリームコーディング (ストリームの種類)
1011 1101	private_stream_1
1011 1110	padding_stream
1011 1111	private_stream_2
110x xxxx	オーディオストリーム ナンバー x xxxx
1110 xxxx	ビデオストリーム ナンバー xxxx

第２１図

シンタクス	ビット数	ニーモニック
private_stream1_PES_payload(){		
private_header(){		
private_stream_id	8	uimsbf
if(stream == ATRAC){		
reserved_for_future_use	8	bslbf
AU_locator	16	uimsbf
if(stream == LPCM){		
fs_flag	1	uimsbf
reserved_for_future_use	3	bslbf
ch_flag	4	uimsbf
AU_locator	16	uimsbf
}else if (stream == SUBTITLE){		
reserved_for_future_use	8	bslbf
AU_locator	16	uimsbf
}		
}		
private_payload()		
}		

第２２図

private_stream_id	ストリームコーディング （ストリームの種類）
0000 xxxx	ATRAC オーディオストリーム ナンバー xxxx
0001 xxxx	LPCM ストリーム ナンバー xxxx
100x xxxx	サブタイトルストリーム ナンバー x xxxx

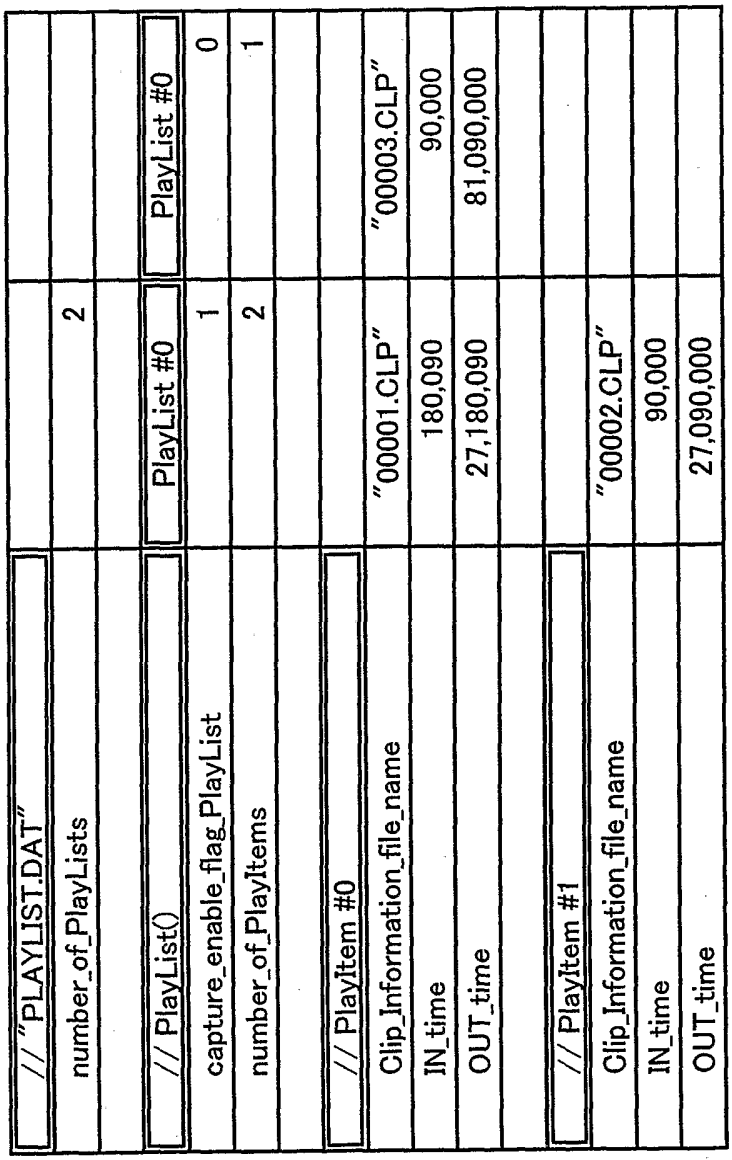
第２３図

シンタクス	ビット数	ニーモニック
private_stream2_PES_payload(){		
reserved_for_future_use	8	bslbf
video_stream_id	8	uimsbf
1stRef_picture	16	uimsbf
2ndRef_picture	16	uimsbf
3rdRef_picture	16	uimsbf
4thRef_picture	16	uimsbf
au_information()		
VBI()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
}		

第２４図

シンタクス	ビット数	ニーモニック
au_information(){		
length	16	uimsbf
reserved_for_word_alignment	8	bslbf
number_of_access_unit	8	uimsbf
for(i=0;i<number_of_access_unit;i++){		
reserved	4	bslbf
pic_struct_copy	4	uimsbf
au_ref_flag	1	uimsbf
reserved	2	bslbf
AU_length	21	uimsbf
}		uimsbf
}		

第２５図


// "PLAYLIST.DAT"
number_of_PlayLists
2
// PlayList()
PlayList #0
PlayList #0
capture_enable_flag_PlayList
1
0
number_of_PlayItems
2
1
// PlayItem #0
Clip_Information_file_name
"00001.CLP"
"00003.CLP"
IN_time
180,090
90,000
OUT_time
27,180,090
81,090,000
// PlayItem #1
Clip_Information_file_name
"00002.CLP"
IN_time
90,000
OUT_time
27,090,000

第２６図Ａ

第２６図

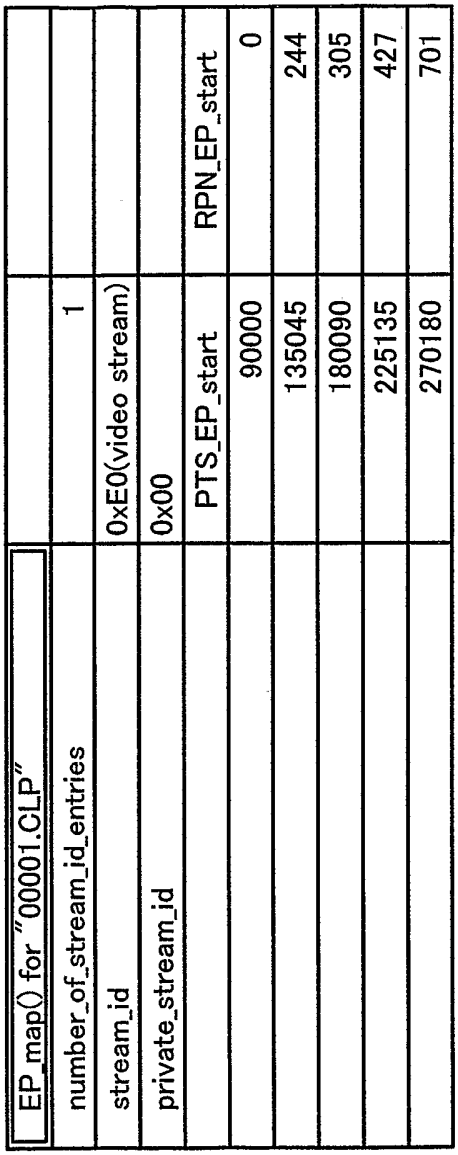
第２６図Ａ
第２６図Ｂ

// Clip()	"00001.CLP"	"00002.CLP"	"00003.CLP"
presentation_start_time	90,000	90,000	90,000
presentation_end_time	27,990,000	27,090,000	81,090,000
capture_enable_flag_Clip	1	0	1
number_of_streams	4	4	3
// stream #0	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xE0(video stream)	0xE0(video stream)	0xE0(video stream)
private_stream_id	0x00	0x00	0x00
	picture_size='720x480'	picture_size='720x480'	picture_size='720x480'
	frame_rate='29.97Hz'	frame_rate='29.97Hz'	frame_rate='29.97Hz'
	cc_flag='Yes'	cc_flag='Yes'	cc_flag='No'
number_of_DynamicInfo	0	0	2
// stream #1	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xBD(private_stream_1)	0xBD(private_stream_1)	0xE1(video stream)
private_stream_id	0x00(ATRAC)	0x00(ATRAC)	0x00
	audio_language_code='日本語'	audio_language_code='日本語'	picture_size='720x480'
	channel_configuration='STEREO'	channel_configuration='STEREO'	frame_rate='29.97Hz'
	lfe_existence='NO'	lfe_existence='NO'	cc_flag='No'
	sampling_frequency='48KHz'	sampling_frequency='48KHz'	
number_of_DynamicInfo	0	0	0

第２６図Ｂ

//stream #2	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xBD(private_stream_1)	0xBD(private_stream_1)	0xBD(private_stream_1)
private_stream_id	0x80(Subtitle)	0x80(Subtitle)	0x00(ATRAC)
	subtitle_language_code='日本語'	subtitle_language_code='日本語'	audio_language_code= '日本語'
	configurable_flag=0	configurable_flag=0	channel_configuration='STEREO'
			lfe_existence='NO'
			sampling_frequency='48KHz'
number_of_DynamicInfo	0	0	3
//stream #3	StreamInfo()	StreamInfo()	
stream_id	0xBD(private_stream_1)	0xBD(private_stream_1)	
private_stream_id	0x81(Subtitle)	0x81(Subtitle)	
	subtitle_language_code='日本語'	subtitle_language_code='日本語'	
	configurable_flag=1	configurable_flag=1	
number_of_DynamicInfo	0	0	

第２７図



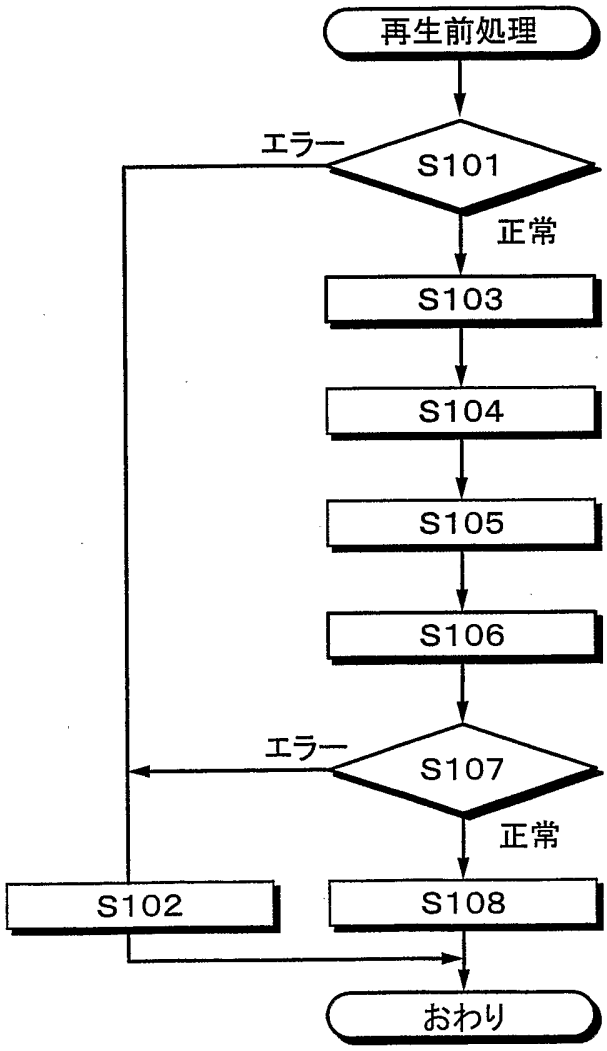
EP_map() for "00001.CLP"		
number_of_stream_id_entries	1	
stream_id	0xE0(video stream)	
private_stream_id	0x00	
	PTS_EP_start	RPN_EP_start
	90000	0
	135045	244
	180090	305
	225135	427
	270180	701

第２８図

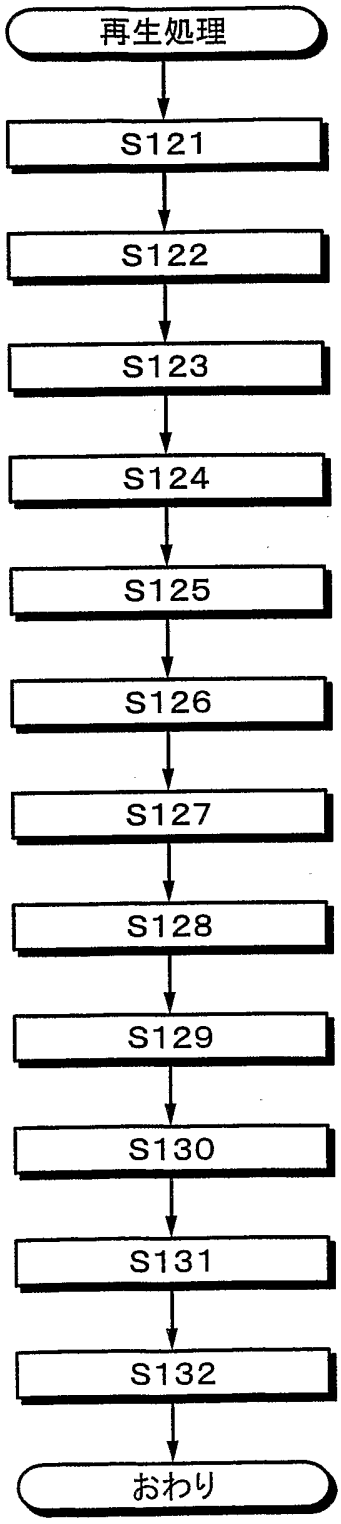
PlayList #0のPlayListMark() number_of_PlayList_marks==7								
Mark()	mark_type	ref_to_ PlayItem_id	mark_time_ stamp	entry_ES_ stream_id	entry_ES_ private_ stream_id	mark_data		
#0	Chapter	0	180,090	0	0	1	先頭	先頭
#1	Index	0	5,580,090	0	0	1	1分	1分
#2	Index	0	10,980,090	0	0	2	2分	2分
#3	Event	0	16,380,090	0	0	0	3分	3分
#4	Chpater	1	90,000	0	0	2	先頭	5分
#5	Index	1	5,490,000	0	0	1	1分	6分
#6	Index	1	10,890,000	0	0	2	2分	7分

PlayList #1のPlayListMark() number_of_PlayList_marks==3							
Mark()	mark_type	ref_to_ PlayItem_id	mark_time_ stamp	entry_ES_ stream_id	entry_ES_ private_ stream_id	mark_data	
#0	Chapter	0	90,000	0	0	0	先頭
#1	Event	0	27,090,000	0xE0	0	1	5分
#2	Event	0	27,540,000	0xE1	0	2	5分5秒(305秒)

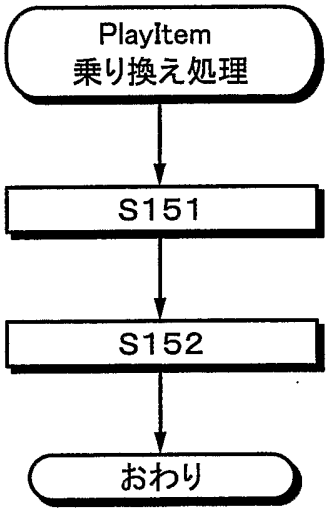
第２９図


再生前処理
エラー
S101
正常
S103
S104
S105
S106
エラー
S107
正常
S102
S108
おわり

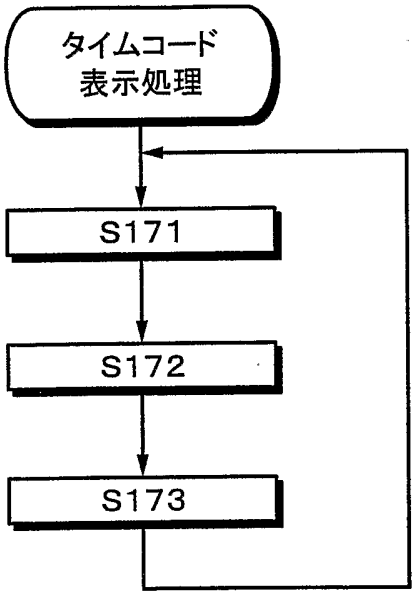
第３０図


再生処理
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130
S131
S132
おわり

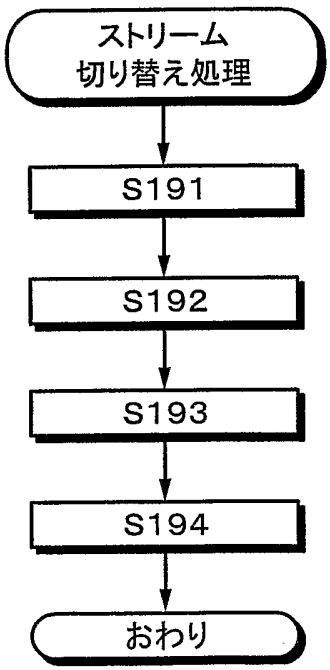
第３１図



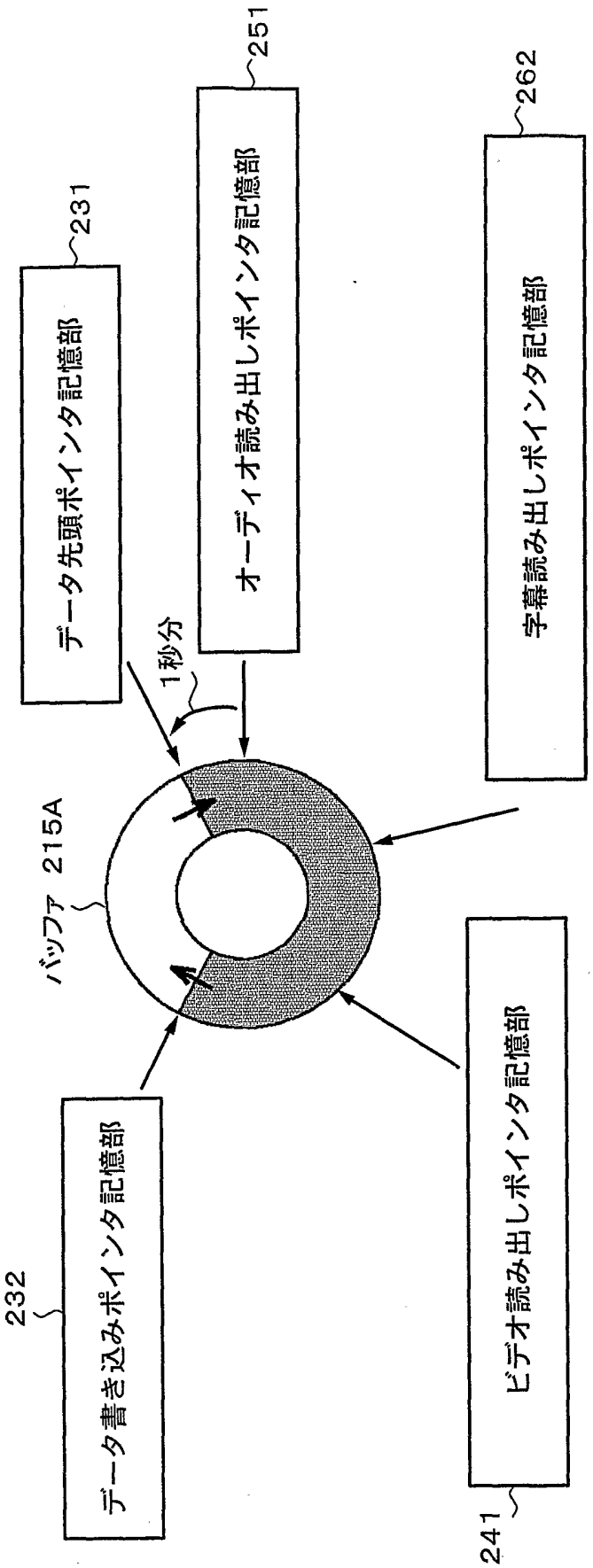
第３２図



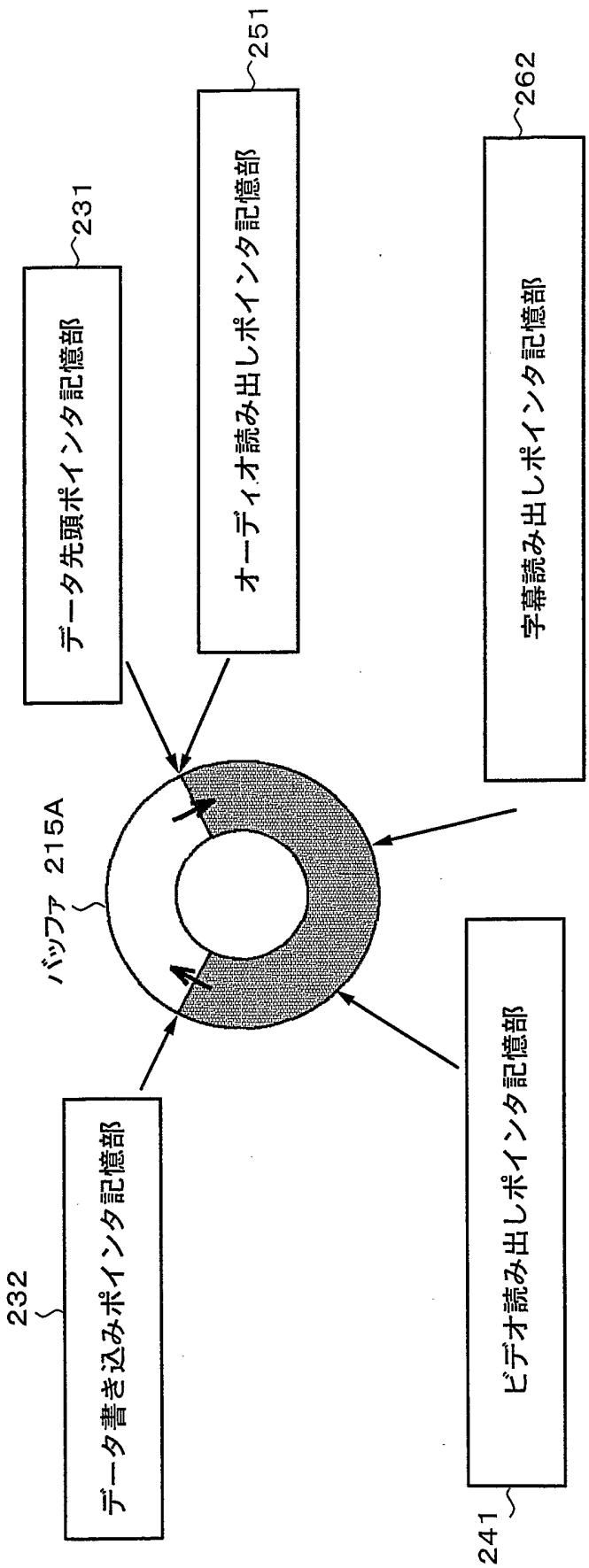
第３３図



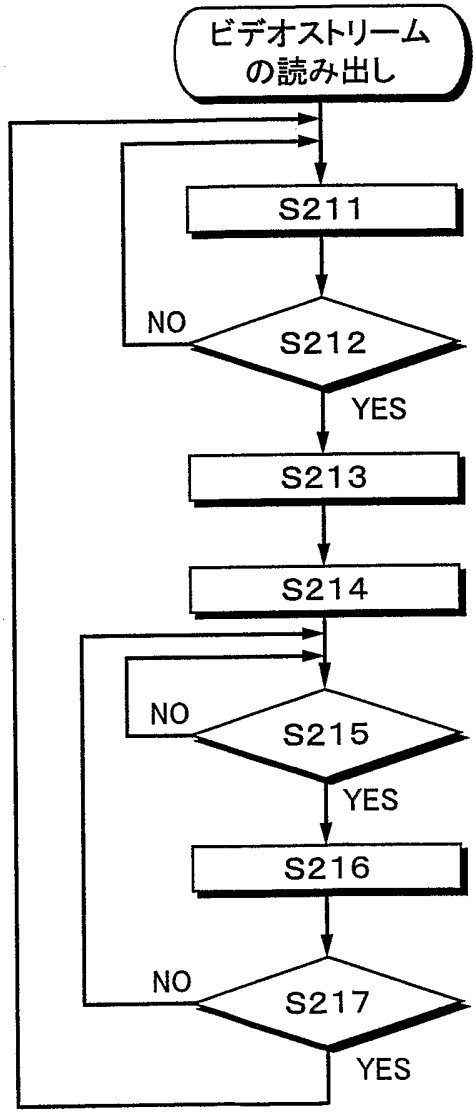
第３４図


データ先頭ポインタ記憶部
231
オーディオ読み出しポインタ記憶部
251
字幕読み出しポインタ記憶部
262
1秒分
バッファ 215A
データ書き込みポインタ記憶部
232
ビデオ読み出しポインタ記憶部
241

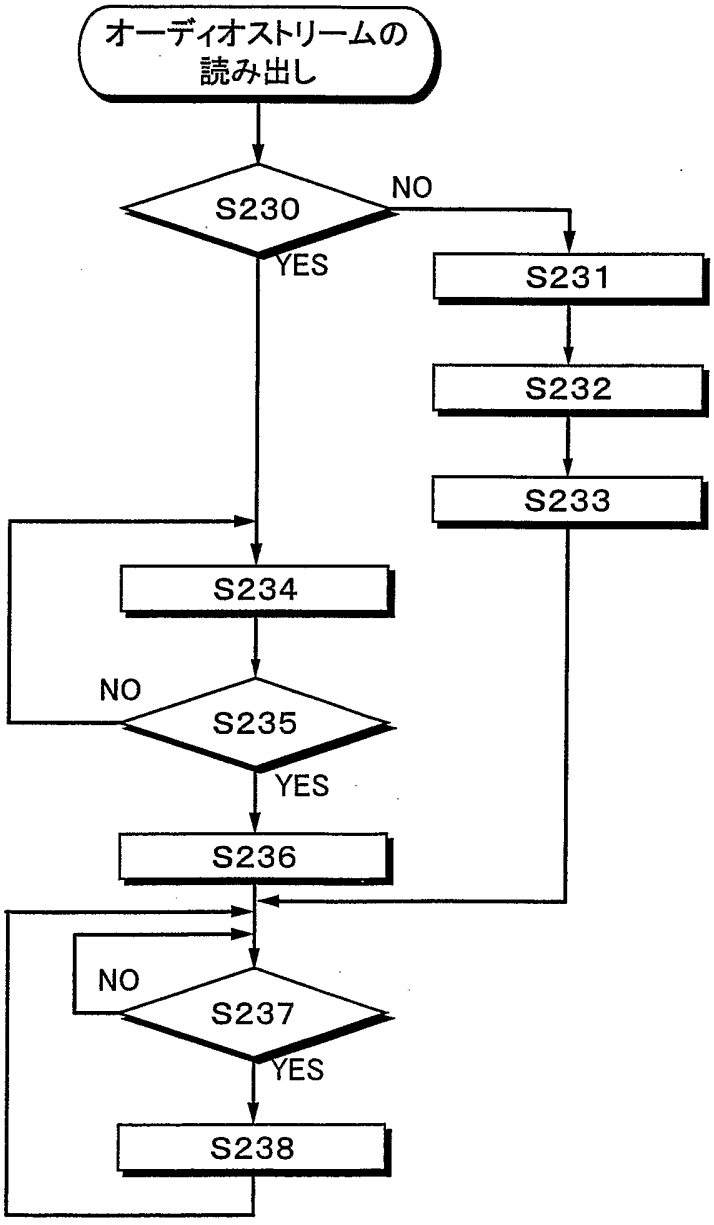
第３５図


データ先頭ポインタ記憶部
231
オーディオ読み出しポインタ記憶部
251
字幕読み出しポインタ記憶部
262
バッファ 215A
データ書き込みポインタ記憶部
232
ビデオ読み出しポインタ記憶部
241

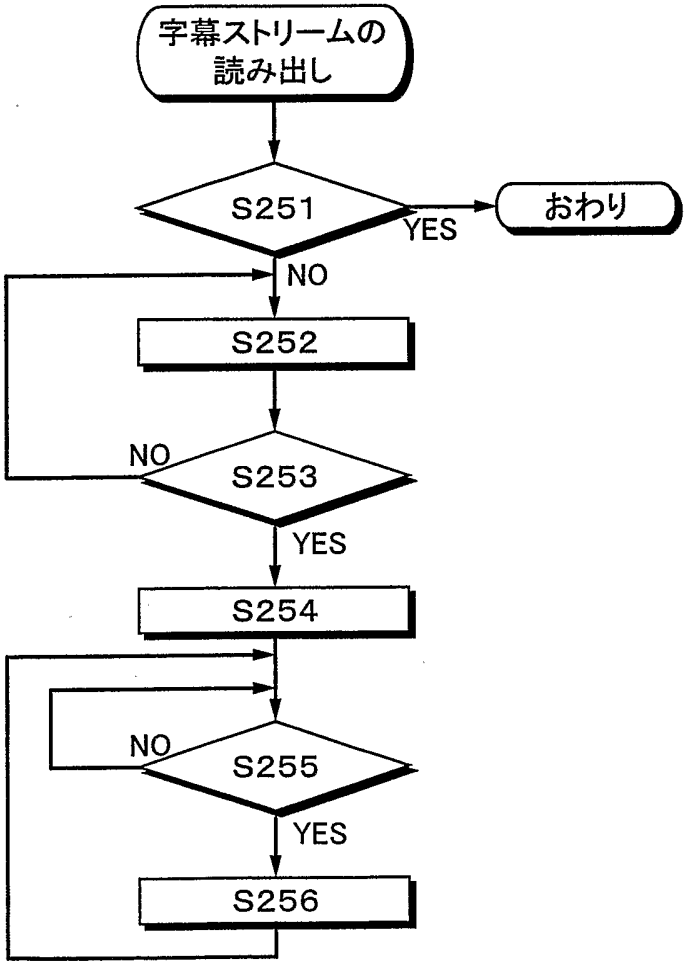
第３６図


ビデオストリーム
の読み出し
S211
NO
S212
YES
S213
S214
NO
S215
YES
S216
NO
S217
YES

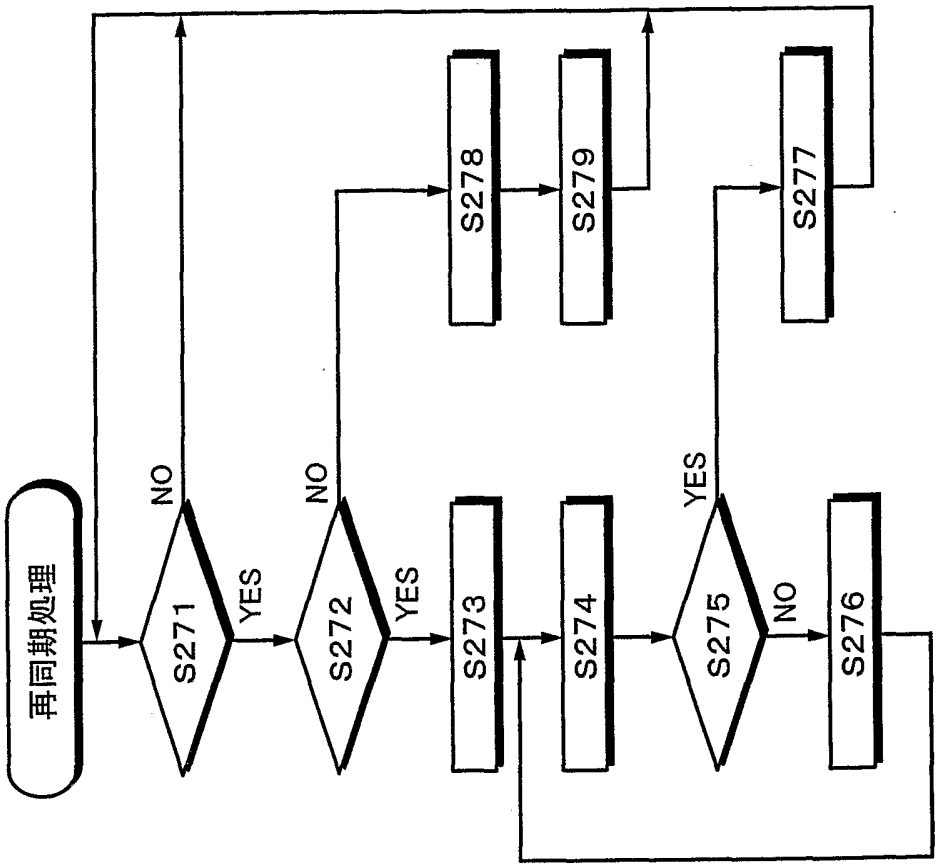
第３７図


オーディオストリームの
読み出し
S230
NO
YES
S231
S232
S233
S234
NO
S235
YES
S236
NO
S237
YES
S238

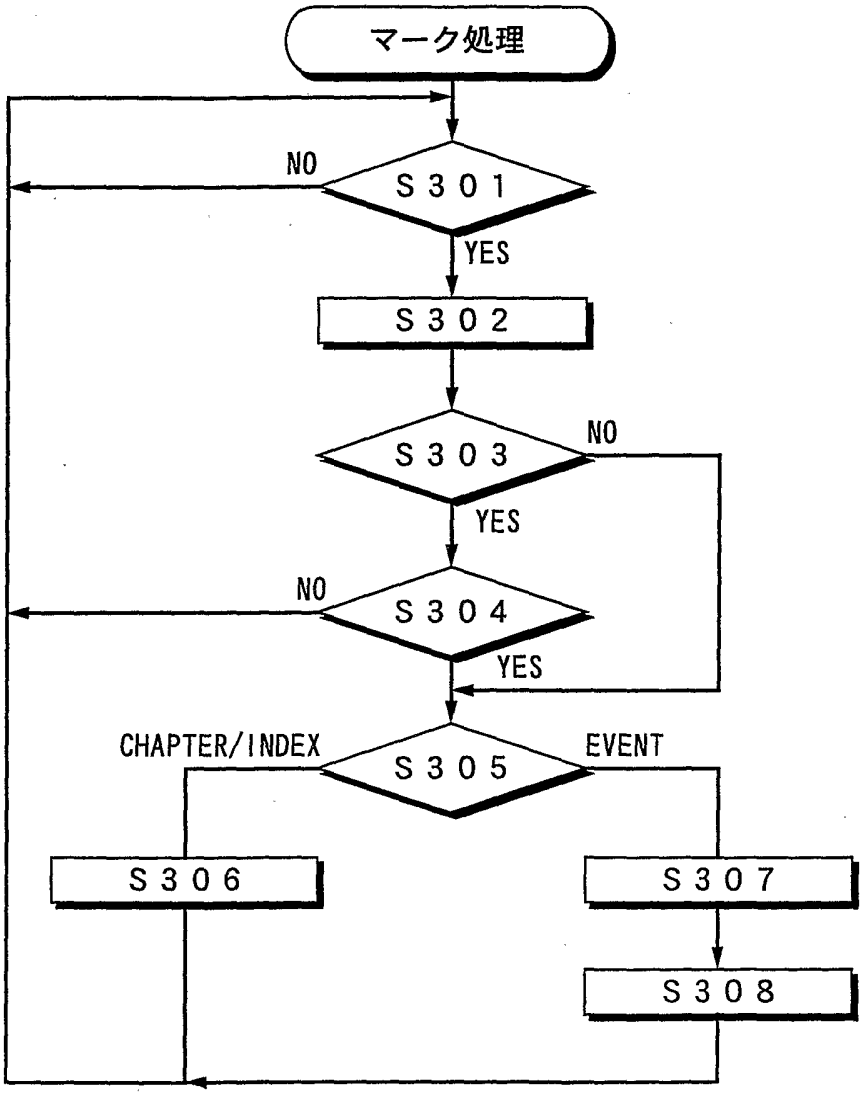
第３８図


字幕ストリームの
読み出し
S251
YES
おわり
NO
S252
NO
S253
YES
S254
NO
S255
YES
S256

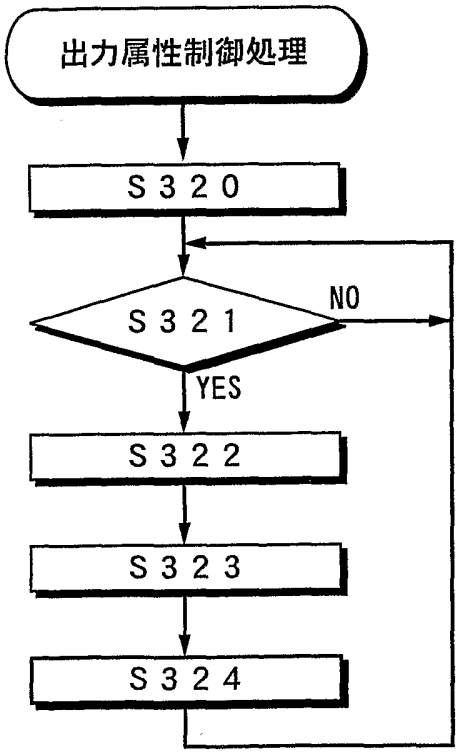
第３９図


再同期処理
S271
NO
YES
S272
NO
YES
S273
S274
S275
YES
NO
S276
S277
S278
S279

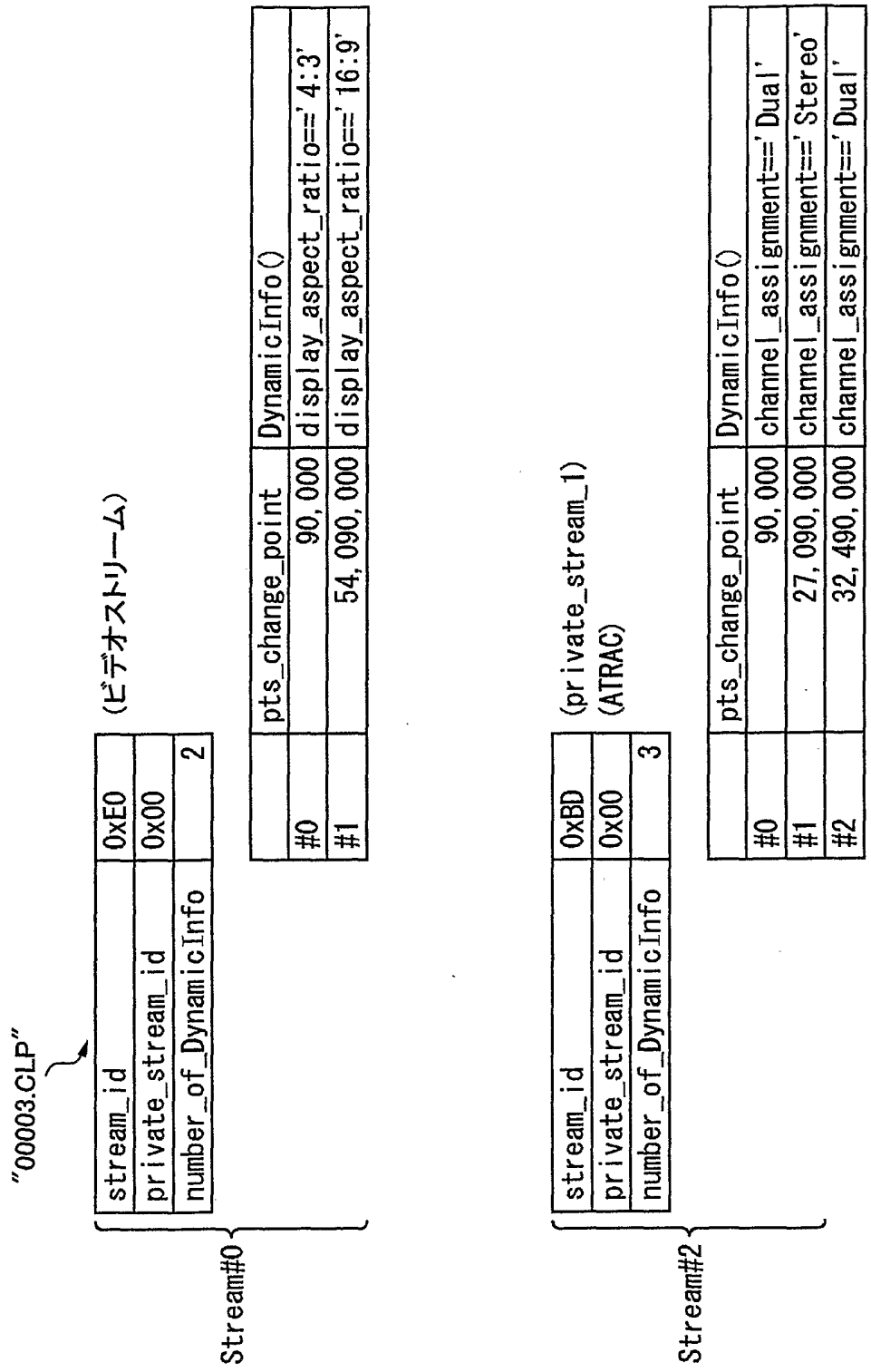
第４０図


マーク処理
S３０１
NO
YES
S３０２
S３０３
NO
YES
S３０４
NO
YES
S３０５
CHAPTER/INDEX
EVENT
S３０６
S３０７
S３０８

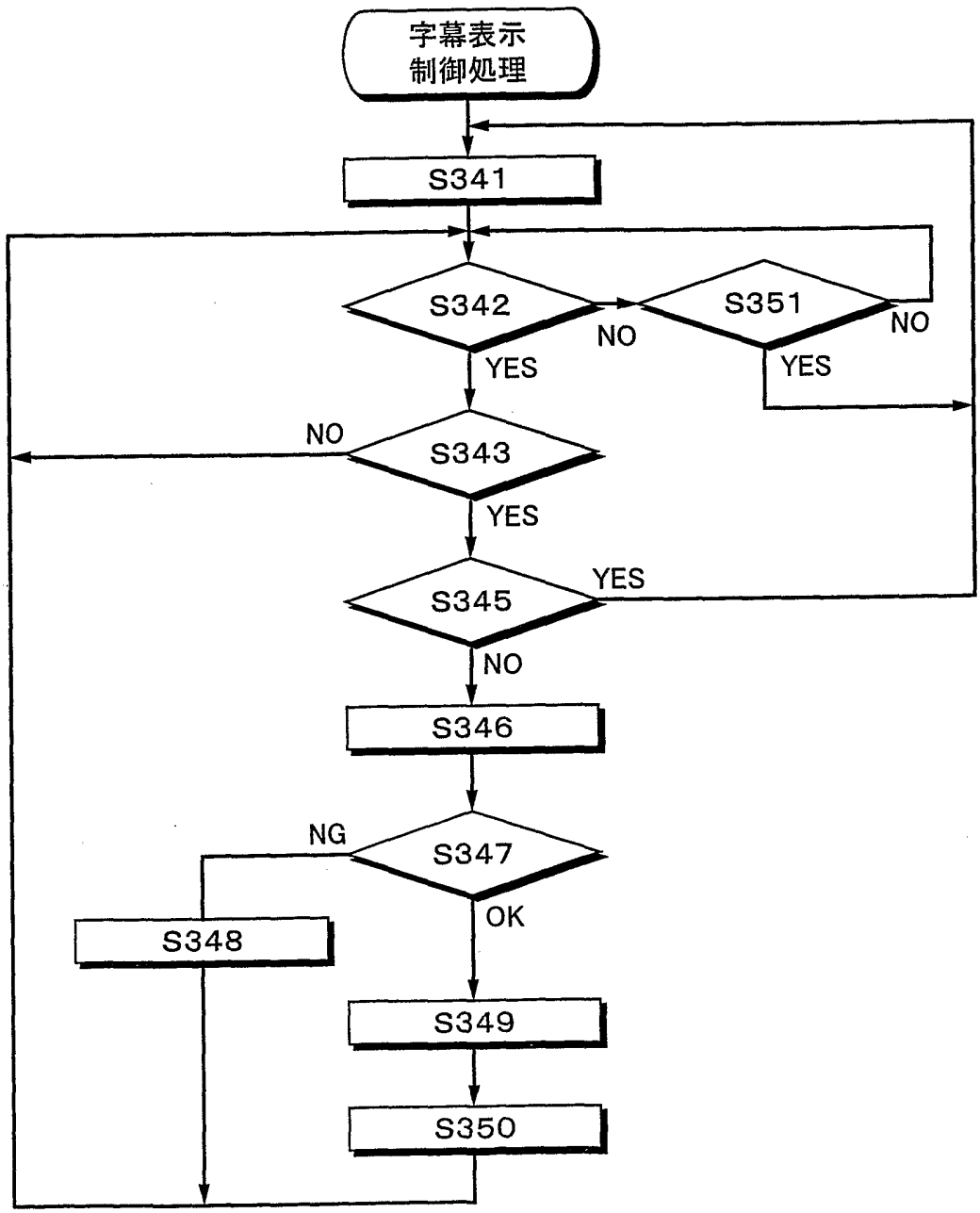
第４１図



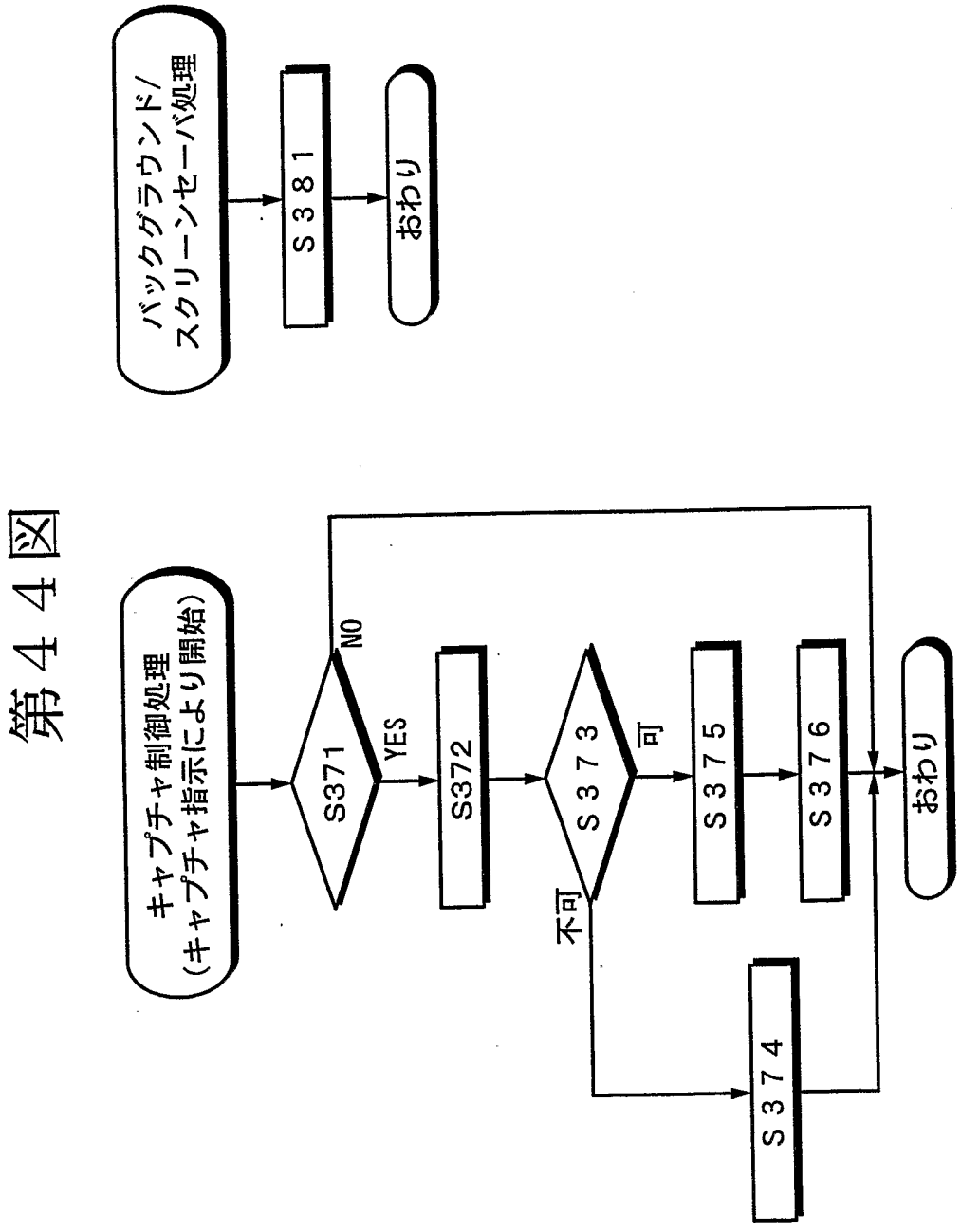
第４２図


"00003.CLP"
(ビデオストリーム)
stream_id
0xE0
private_stream_id
0x00
number_of_DynamicInfo
2
Stream#0
pts_change_point
DynamicInfo()
#0
90,000
display_aspect_ratio=='4:3'
#1
54,090,000
display_aspect_ratio=='16:9'
(private_stream_1)
(ATRAC)
stream_id
0xBD
private_stream_id
0x00
number_of_DynamicInfo
3
Stream#2
pts_change_point
DynamicInfo()
#0
90,000
channel_assignment=='Dual'
#1
27,090,000
channel_assignment=='Stereo'
#2
32,490,000
channel_assignment=='Dual'

第４３図


字幕表示
制御処理
S341
S342
NO
S351
NO
YES
YES
NO
S343
YES
YES
S345
NO
S346
NG
S347
OK
S348
S349
S350

第４４図


バックグラウンド/
スクリーンセーバ処理
S381
おわり
キャプチャ制御処理
(キャプチャ指示により開始)
S371
NO
YES
S372
S373
可
不可
S375
S376
S374
おわり

第４５図

シンタクス	ビット数	ニーモニック
private_stream2_PES_payload(){		
reserved_for_future_use	7	bslbf
capture_enable_ps2	1	uimsbf
video_stream_id	8	uimsbf
1stRef_picture	16	uimsbf
2ndRef_picture	16	uimsbf
3rdRef_picture	16	uimsbf
4thRef_picture	16	uimsbf
au_information()		
grain_variance_information()		
VBI()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
}		

第４６図

シンタクス	ビット数	ニーモニック
au_information(){		
length	16	uimsbf
reserved_for_word_align	8	bslbf
number_of_access_unit	8	uimsbf
for(i=0;i<number_of_access_unit;i++){		
reserved	3	bslbf
capture_enable_flag_AU	1	uimsbf
pic_struct_copy	4	uimsbf
au_ref_flag	1	uimsbf
reserved	2	bslbf
AU_length	21	uimsbf
}		uimsbf
}		

符号の説明

１０１	ディスク
１０２	ディスクドライブ
１１１	バス
１１２	CPU
１１３	メモリ
１１４	ドライブインターフェース
１１５	入力インターフェース
１１６	ビデオデコーダ
１１７	オーディオデコーダ
１１８	ビデオ出力インターフェース
１１９	オーディオ出力インターフェース
１２０	ビデオ出力端子
１２１	オーディオ出力端子
２０１	オペレーティングシステム
２１０	ビデオコンテンツ再生プログラム
２１１	スクリプト制御モジュール
２１２	プレイヤ制御モジュール
２１３	コンテンツデータ供給モジュール
２１４	デコード制御モジュール
２１４Ａ	計時部
２１５	バッファ制御モジュール
２１５Ａ	バッファ
２１６	ビデオデコーダ制御モジュール
２１７	オーディオデコーダ制御モジュール

２１８	字幕デコーダ制御モジュール
２１９	グラフィクス処理モジュール
２２０	ビデオ出力モジュール
２２０Ａ	FIFO
２２１	オーディオ出力モジュール
２２１Ａ	FIFO
２３１	データ先頭ポインタ記憶部
２３２	データ書き込みポインタ記憶部
２３３	ビデオ読み出し機能部
２３４	オーディオ読み出し機能部
２３５	字幕読み出し機能部
２４１	ビデオ読み出しポインタ記憶部
２４２	stream_idレジスタ
２４３	au_information()レジスタ２４３
２５１	オーディオ読み出しポインタ記憶部
２５２	stream_idレジスタ
２５３	private_stream_idレジスタ
２６１	字幕読み出し機能フラグ記憶部２６１
２６２	字幕読み出しポインタ記憶部
２６３	stream_idレジスタ
２６４	private_stream_idレジスタ
Ｓ１０１	ディスクチェック
Ｓ１０２	エラー処理
Ｓ１０３	SCRIPT.DATとPLAYLIST.DATの読み込み
Ｓ１０４	ファイル供給
Ｓ１０５	PLAYLIST.DATの解析

Ｓ１０６　クリップ情報ファイルの読み出し
Ｓ１０７　ファイルチェック
Ｓ１０８　SCRIPT.DATの解釈、実行
Ｓ１２１　IN_time認識
Ｓ１２２　再生開始位置調査
Ｓ１２３　タイムコード表示
Ｓ１２４　PlaylistMark()の解析
Ｓ１２５　再生するストリームの決定
Ｓ１２６　出力属性制御
Ｓ１２７　再生開始の準備
Ｓ１２８　データ読み込み開始
Ｓ１２９　デコーダ制御開始
Ｓ１３０　デコード開始
Ｓ１３１　グラフィクス処理開始
Ｓ１３２　出力処理開始
Ｓ１５１　デコード中断制御
Ｓ１５２　プレイアイテム再生開始
Ｓ１７１　メッセージ出力
Ｓ１７２　タイムコード換算
Ｓ１７３　タイムコード出力
Ｓ１９１　ストリーム数チェック
Ｓ１９２　再生しているストリームの特定
Ｓ１９３　次に再生するストリームの認識
Ｓ１９４　次に再生するストリームの読み出し指示
Ｓ２１１　private_stream_2 パケットの探索
Ｓ２１２　指定されたstream_idか？

Ｓ２１３	au_Information()を内部レジスタに読み込む
Ｓ２１４	ビデオ読み出しポインタに直後のビデオパケットの先頭を挿入する
Ｓ２１５	データ要求あり？
Ｓ２１６	プログラムストリームの構文解析を行いつつ、Au_length分のビデオデータを出力
Ｓ２１７	number-of-access-unitだけ処理したか？
Ｓ２３０	private_stream_1？
Ｓ２３１	同期コード探索
Ｓ２３２	MPEGオーディオパケットの探索
Ｓ２３３	オーディオ読み出しポインタ更新
Ｓ２３４	private_stream_1 パケットの探索
Ｓ２３５	指定されたprivate_stream_idか？
Ｓ２３６	オーディオ読み出しポインタ234←Au_locatorの直後の位置＋Au_locator
Ｓ２３７	データ要求あり？
Ｓ２３８	プログラムストリームの構文解析を行いつつ、固定長のオーディオアクセスユニットを出力
Ｓ２５１	字幕読み出し機能フラグ？
Ｓ２５２	private_stream_1 パケットの探索
Ｓ２５３	指定されたprivate_stream_idか？
Ｓ２５４	字幕読み出しポインタ←Au_locatorの直後の位置＋Au_locator
Ｓ２５５	データ要求あり？
Ｓ２５６	プログラムストリームの構文解析を行いつつ、

字幕アクセスユニットの先頭に記述をしている長さ分の字幕アクセスユニットを出力

S271 タイムスタンプずれが大？

S272 ビデオが遅れている？

S273 ビデオアクセスユニットのスキップを指示

S274 アクセスユニットのau_ref_flagを検査

S275 アクセスユニットは、非参照画像？

S276 ビデオアクセスユニットを処理

S277 ビデオアクセスユニットの処理をスキップ

S278 ビデオの繰り返し出力を指示

S279 ビデオの繰り返し出力

S301 現在時刻が、mark_time_stampに一致？

S302 処理対象とするmarkを認識

S303 処理対象markに、entry_ES_stream_id/entry_ES_private_stream_idが記述されている？

S304 再生中のストリームに、entry_ES_stream_id/entry_ES_private_stream_idのストリームが含まれる？

S305 処理対象markのmark_typeを判定

S306 チャプタ/インデックスの更新処理

S307 イベントメッセージとmark_dataを、スクリプト制御モジュール211に通知

S308 mark_dataに対応する処理

S320 pts_change_pointを渡す

S321 現在時刻が、pts_change_pointに一致？

S322 現在時刻に一致するpts_change_pointのDynamicInfo()を認識

S323 DynamicInfo()に記述された出力属性を供給

S324 出力属性にしたがって、ビデオ/オーディオの出力の制御を開始

S341 字幕の処理の初期化を指示

S342 字幕の表示について、新たな表示方式指示あり？

S343 字幕ストリームを再生中？

S345 デフォルト表示の指示？

S346 再生中の字幕ストリームのStaticInfo()を取得

S347 StaticInfo()のconfigurable_flagを判定

S348 エラー表示

S349 表示方式指示を供給

S350 表示方式指示にしたがって、字幕の処理を開始

S351 PlayItem乗り替え？

S371 ビデオを再生中？

S372 再生中のビデオのPlayList()とClip()から、capture_enable_flagを取得

S373 capture_enable_flagに基づき、キャプチャの可否を判定？

S374 エラー表示

S375 キャプチャ指示を供給

S376 キャプチャ、記憶

S381 ピクチャの表示

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/010901

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] H04N5/92, G11B20/10, 20/12, 27/34

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)
Int.Cl[7] H04N5/76-5/956, G11B20/10-20/12, 27/34

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

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2005
Kokai Jitsuyo Shinan Koho	1971-2005	Toroku Jitsuyo Shinan Koho	1994-2005

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2001-78149 A (Toshiba Corp.), 23 March, 2001 (23.03.01), Full text; all drawings (Family: none)	1-7
A	JP 2004-194131 A (Hitachi, Ltd.), 08 July, 2004 (08.07.04), Full text; all drawings (Family: none)	1-7
A	JP 2000-341640 A (Sony Corp.), 08 December, 2000 (08.12.00), Full text; all drawings & EP 1043892 A1	1-7

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 August, 2005 (12.08.05)	30 August, 2005 (30.08.05)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００５／０１０９０１

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.[7]　H04N5/92, G11B20/10, 20/12, 27/34

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.[7]　H04N5/76-5/956, G11B20/10-20/12, 27/34

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００５年
日本国実用新案登録公報	１９９６－２００５年
日本国登録実用新案公報	１９９４－２００５年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	JP 2001-78149 A（株式会社東芝）2001.03.23, 全文、全図（ファミリーなし）	１－７
A	JP 2004-194131 A（株式会社日立製作所）2004.07.08, 全文、全図（ファミリーなし）	１－７
A	JP 2000-341640 A（ソニー株式会社）2000.12.08, 全文、全図 & EP 1043892 A1	１－７

☐　Ｃ欄の続きにも文献が列挙されている。

☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日 １２．０８．２００５	国際調査報告の発送日 30.8.2005
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員）　5C　9185 鈴木　明 電話番号　０３－３５８１－１１０１　内線　３５４１