

(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
2001 年 11 月 1 日 (01.11.2001)

PCT

(10) 国際公開番号
WO 01/82608 A1

- (51) 国際特許分類: H04N 5/93, G11B 20/10 Motoki [JP/JP]. 浜田俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 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).
- (21) 国際出願番号: PCT/JP01/03414
- (22) 国際出願日: 2001 年 4 月 20 日 (20.04.2001) (74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語 (81) 指定国 (国内): CN, KR, US.
- (30) 優先権データ: (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
- 特願2000-183770 2000 年 4 月 21 日 (21.04.2000) JP
特願2000-268043 2000 年 9 月 5 日 (05.09.2000) 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]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).
- 添付公開書類:
— 国際調査報告書
— 補正書

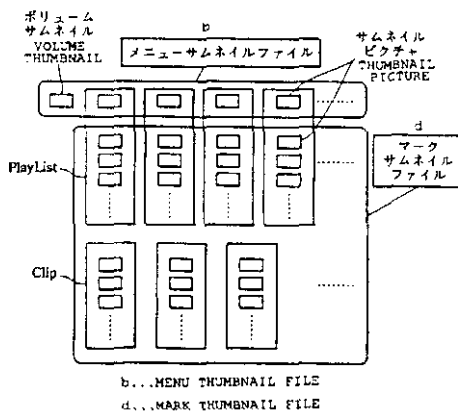
(72) 発明者; および

(75) 発明者/出願人 (米国についてののみ): 加藤元樹 (KATO,

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。

(54) Title: INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(54) 発明の名称: 情報処理装置及び方法、プログラム、並びに記録媒体



(57) Abstract: A ClipMark composed of a mark indicating a characteristic image extracted from an inputted AV stream is created as management information for managing the AV stream. A PlaylistMark composed of a mark indicating an image arbitrarily specified by the user is created from a reproduction section corresponding to a Playlist defining a combination of predetermined sections in the AV stream. The ClipMark and PlaylistMark are recorded as mutually independent tables on a recording medium. Therefore a quick and reliable access to a desired position of the AV stream can be made.

WO 01/82608 A1

[続葉有]



(57) 要約:

入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成し、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録するようにしたので、A Vストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

明細書

情報処理装置及び方法、プログラム、並びに記録媒体

技術分野

本発明は情報処理装置及び方法、プログラム、並びに記録媒体に関し、特に、A Vストリームの所望の位置に、迅速にアクセスすることができるようにした情報処理装置及び方法、プログラム、並びに記録媒体に関する。

背景技術

近年、記録可能で記録再生装置から取り外し可能なディスク型媒体として、各種の光ディスクが提案されている。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のA V (Audio Visual) 信号を記録するメディアとしての期待が高い。

この記録可能な光ディスクに記録するデジタルのA V信号のソース（供給源）としては、記録装置自身が、アナログ入力のオーディオビデオ信号を、MPEG-2方式で画像圧縮して作るビットストリームや、デジタルテレビジョン放送の電波から直接得られるMPEG-2方式のビットストリームなどがある。一般に、デジタルテレビジョン放送では、MPEG-2トランスポートストリームが使われる。トランスポートストリームは、トランスポートパケットが連続したストリームであり、トランスポートパケットは、例えば、MPEG-2ビデオストリームやMPEG-1オーディオストリームがパケット化されたものである。1つのトランスポートパケットのデータ長は188バイトである。デジタルテレビジョン放送で受信されるトランスポートストリームのA Vプログラムを記録装置で光ディスクにそのまま記録すれば、ビデオやオーディオの品質を全く劣化させることなく記録することが可能である。

ユーザが、光ディスクに記録されているトランスポートストリームの中から興

味のあるシーン、例えば番組の頭出し点などをサーチできるようにするために、再生装置はランダムアクセス再生ができることが求められる。

一般に、MPEG-2ビデオのストリームは、0.5秒程度の間隔でIピクチャを符号化し、それ以外のピクチャはPピクチャ又はBピクチャとして符号化される。したがって、MPEG-2ビデオのストリームが記録された光ディスクから、ランダムアクセスし、ビデオ再生する場合、はじめに、Iピクチャをサーチしなければならない。

しかしながら、従来は、光ディスクに記録されているトランスポートストリームに、ランダムアクセスし、ビデオ再生する場合に、Iピクチャの開始バイトを効率良くサーチすることが困難であった。すなわち、光ディスク上のトランスポートストリームのランダムなバイト位置から、読み出したビデオストリームのシンタクスを解析し、Iピクチャの開始バイトをサーチしなければならず、Iピクチャのサーチに時間がかかり、ユーザからの入力に対して応答の速いランダムアクセス再生を行うことが困難であった。

発明の開示

本発明の目的は、このような状況を鑑み、ユーザのランダムアクセス再生の指示に対して、記録媒体からのトランスポートストリームの読出位置の決定とストリームの復号開始を速やかに行えるようにすることにある。

本発明に係る情報処理装置は、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成すると共に、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成手段と、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有する。

ここで、生成手段は、ClipMarkをClipMarkInformationファイルとして生成すると共に、PlayListをPlayListファイルとして生成するようになっていることができる。

PlayListMarkは、PlayListを再生するときのResume点を示すマークを更に含むようにすることができる。

PlayListを再生するとき、PlayListの再生区間に対応するA VストリームのClipMarkを構成するマークを参照するようにすることができる。

PlayListMarkのマークは、プレゼンテーションタイムスタンプと、PlayListの再生経路を構成するA Vストリームデータ上の指定された1つの再生区間を示す識別情報を含むようにすることができる。

ClipMarkを構成するマーク、又は、PlayListMarkを構成するマークは、エントリーストリームのエントリポイントを特定する情報を含むようにすることができる。

PlayListMarkのマークは、ユーザが指定したお気に入りのシーンの開始点又はPlayListのResume点を少なくとも含むタイプの情報を含むようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

第1の記録手段による記録の際に検出された特徴的な画像のタイプを検出するタイプ検出手段を更に含み、第1の記録手段は、ClipMarkを構成するマークと、タイプ検出手段により検出されたタイプとを対応させて記録するようにすることができる。

ClipMarkのマークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含むようにすることができる。

本発明に係る情報処理方法は、入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合

わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを有する。

本発明に係る記録媒体のプログラムは、入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを含む。

本発明に係るプログラムは、入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させる。

本発明の情報処理装置は、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを読み出す読出手段と、読出手段により読み出された管理情報とPlayListMarkによる情報を提示する提示手段と、提示手段により提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照手段と、参照手段により参照されたClipMarkを含み、ClipMarkに対応する位置からA Vストリームを再生する再生手段とを含む。

ここで、提示手段は、PlayListMarkに対応するサムネイル画像によるリストをユーザに提示するようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

ClipMarkのマークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含むようにすることができる。

本発明に係る情報処理装置は、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのA Vストリームの再生を制御する再生制御ステップとを含む。

本発明に係る記録媒体のプログラムは、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのA Vストリームの再生を制御する再生制御ステップとを含む。

本発明に係るプログラムは、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのA Vストリームの再生を制御する再生制御ステップとをコンピュータに実行させる。

本発明に係る記録媒体には、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されている。

本発明に係る情報処理装置及び方法、並びにプログラムにおいては、入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが生成され、ClipMark、及びPlayListMarkが各々独立したテーブルとして記録媒体に記録される。

本発明に係る情報処理装置及び方法、並びにプログラムは、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが読み出され、その読み出された管理情報とPlayListMarkによる情報が提示され、提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkが参照され、参照されたClipMark

を含み、ClipMarkに対応する位置からAVストリームが再生される。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

図面の簡単な説明

図1は、本発明を適用した記録再生装置の一例を説明するブロック図である。

図2は、記録再生装置により記録媒体に記録されるデータのフォーマットについて説明する図である。

図3は、Real PlaylistとVirtual Playlistについて説明する図である。

図4A～図4Cは、Real Playlistの作成について説明する図である。

図5A～図5Cは、Real Playlistの削除について説明する図である。

図6A及び図6Bは、アセンブル編集について説明する図である。

図7は、Virtual Playlistにサブバスを設ける場合について説明する図である。

図8は、Playlistの再生順序の変更について説明する図である。

図9は、Playlist上のマークとClip上のマークについて説明する図である。

図10は、メニューサムネイルについて説明する図である。

図11は、Playlistに付加されるマークについて説明する図である。

図12は、クリップに付加されるマークについて説明する図である。

図13は、Playlist、Clip、サムネイルファイルの関係について説明する図である。

図14は、ディレクトリ構造について説明する図である。

図15は、info.dvrのシンタクスを示す図である。

図16は、DVR volumeのシンタクスを示す図である。

図17は、Resumevolumeのシンタクスを示す図である。

図18は、UIAppInfovolumeのシンタクスを示す図である。

図19は、Character set valueのテーブルを示す図である。

図20は、TableOfPlaylistのシンタクスを示す図である。

図21は、TableOfPlaylistの他のシンタクスを示す図である。

図 2 2 は、MakersPrivateDataのシンタクスを示す図である。

図 2 3 は、xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

図 2 4 A～図 2 4 C は、PlayListについて説明する図である。

図 2 5 は、PlayListのシンタクスを示す図である。

図 2 6 は、PlayList_typeのテーブルを示す図である。

図 2 7 は、UIAppinfoPlayListのシンタクスを示す図である。

図 2 8 A～図 2 8 C は、図 2 7 に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。

図 2 9 は、PlayItemについて説明する図である。

図 3 0 は、PlayItemについて説明する図である。

図 3 1 は、PlayItemについて説明する図である。

図 3 2 は、PlayItemのシンタクスを示す図である。

図 3 3 は、IN_timeについて説明する図である。

図 3 4 は、OUT_timeについて説明する図である。

図 3 5 は、Connection_Conditionのテーブルを示す図である。

図 3 6 A～図 3 6 D は、Connection_Conditionについて説明する図である。

図 3 7 は、BridgeSequenceInfoを説明する図である。

図 3 8 は、BridgeSequenceInfoのシンタクスを示す図である。

図 3 9 は、SubPlayItemについて説明する図である。

図 4 0 は、SubPlayItemのシンタクスを示す図である。

図 4 1 は、SubPath_typeのテーブルを示す図である。

図 4 2 は、PlayListMarkのシンタクスを示す図である。

図 4 3 は、Mark_typeのテーブルを示す図である。

図 4 4 は、Mark_time_stampを説明する図である。

図 4 5 は、zzzzz.Clipのシンタクスを示す図である。

図 4 6 は、ClipInfoのシンタクスを示す図である。

図 4 7 は、Clip_stream_typeのテーブルを示す図である。

図 4 8 は、offset_SPNについて説明する図である。

図 4 9 は、offset_SPNについて説明する図である。

図 5 0 A 及び図 5 0 B は、S T C 区間について説明する図である。

図 5 1 は、STC_Info について説明する図である。

図 5 2 は、STC_Info のシンタクスを示す図である。

図 5 3 は、ProgramInfo を説明する図である。

図 5 4 は、ProgramInfo のシンタクスを示す図である。

図 5 5 は、VideoCondngInfo のシンタクスを示す図である。

図 5 6 は、Video_format のテーブルを示す図である。

図 5 7 は、frame_rate のテーブルを示す図である。

図 5 8 は、display_aspect_ratio のテーブルを示す図である。

図 5 9 は、AudioCondngInfo のシンタクスを示す図である。

図 6 0 は、audio_coding のテーブルを示す図である。

図 6 1 は、audio_component_type のテーブルを示す図である。

図 6 2 は、sampling_frequency のテーブルを示す図である。

図 6 3 は、CPI について説明する図である。

図 6 4 は、CPI について説明する図である。

図 6 5 は、CPI のシンタクスを示す図である。

図 6 6 は、CPI_type のテーブルを示す図である。

図 6 7 は、ビデオ EP_map について説明する図である。

図 6 8 は、EP_map について説明する図である。

図 6 9 は、EP_map について説明する図である。

図 7 0 は、EP_map のシンタクスを示す図である。

図 7 1 は、EP_type values のテーブルを示す図である。

図 7 2 は、EP_map_for_one_stream_PID のシンタクスを示す図である。

図 7 3 は、TU_map について説明する図である。

図 7 4 は、TU_map のシンタクスを示す図である。

図 7 5 は、ClipMark のシンタクスを示す図である。

図 7 6 は、mark_type のテーブルを示す図である。

図 7 7 は、mark_type_stamp のテーブルを示す図である。

図 7 8 は、ClipMark のシンタクスの他の例を示す図である。

図 7 9 は、Mark_type のテーブルの他の例を示す図である。

図 8 0 は、mark_entry() と representative_picture_entry() の例を示す図である。

図 8 1 は、mark_entry() と representative_picture_entry() のシンタクスを示す図である。

図 8 2 は、mark_entry() と representative_picture_entry() のシンタクスの他の例を示す図である。

図 8 3 は、RSPN_ref_EP_start と offset_num_pictures の関係を説明する図である。

図 8 4 は、mark_entry() と representative_picture_entry() のシンタクスの他の例を示す図である。

図 8 5 は、ClipMark と EP_map の関係を説明する図である。

図 8 6 は、menu.thmb と mark.thmb のシンタクスを示す図である。

図 8 7 は、Thumbnail のシンタクスを示す図である。

図 8 8 は、thumbnail_picture_format のテーブルを示す図である。

図 8 9 A 及び図 8 9 B は、tn_block について説明する図である。

図 9 0 は、DVR MPEG 2 のトランスポートストリームの構造について説明する図である。

図 9 1 は、DVR MPEG 2 のトランスポートストリームのレコーダモデルを示す図である。

図 9 2 は、DVR MPEG 2 のトランスポートストリームのプレーヤモデルを示す図である。

図 9 3 は、source packet のシンタクスを示す図である。

図 9 4 は、TP_extra_header のシンタクスを示す図である。

図 9 5 は、copy permission indicator のテーブルを示す図である。

図 9 6 は、シームレス接続について説明する図である。

図 9 7 は、シームレス接続について説明する図である。

図 9 8 は、シームレス接続について説明する図である。

図 9 9 は、シームレス接続について説明する図である。

図 1 0 0 は、シームレス接続について説明する図である

図 1 0 1 は、オーディオのオーバーラップについて説明する図である。

図 1 0 2 は、BridgeSequenceを用いたシームレス接続について説明する図である。

図 1 0 3 は、BridgeSequenceを用いないシームレス接続について説明する図である。

図 1 0 4 は、DVR STDモデルを示す図である。

図 1 0 5 は、復号、表示のタイミングチャートを示す図である。

図 1 0 6 は、図 8 1 のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図 1 0 7 は、図 8 1 のシンタクスの場合における再生の動作を説明する図である。

図 1 0 8 は、EP_mapの例を示す図である。

図 1 0 9 は、ClipMarkの例を示す図である。

図 1 1 0 は、図 8 1 のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

図 1 1 1 は、図 8 1 のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

図 1 1 2 は、図 8 2 のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図 1 1 3 は、図 8 2 のシンタクスの場合における再生を説明する図である。

図 1 1 4 は、EP_mapの例を示す図である。

図 1 1 5 は、ClipMarkの例を示す図である。

図 1 1 6 は、図 8 2 のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図 1 1 7 は、図 8 2 のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図 1 1 8 は、図 8 4 のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図 1 1 9 は、図 8 4 のシンタクスの場合における再生を説明する図である。

図 1 2 0 は、EP_mapの例を示す図である。

図 1 2 1 は、ClipMarkの例を示す図である。

図 1 2 2 は、図 8 4 のシンタクスの場合における CMスキップ再生を説明するフローチャートである。

図 1 2 3 は、図 8 4 のシンタクスの場合における CMスキップ再生を説明するフローチャートである。

図 1 2 4 は、アプリケーションフォーマットを示す図である。

図 1 2 5 は、PlayList上のマークとClip上のマークを説明する図である。

図 1 2 6 は、ClipMarkのシンタクスの他の例を示す図である。

図 1 2 7 は、ClipMarkのシンタクスの更に他の例を示す図である。

図 1 2 8 は、アナログ A V 信号をエンコードして記録する場合のClipMarkの作成について説明するフローチャートである。

図 1 2 9 は、トランスポートストリームを記録する場合のClipMarkの作成について説明するフローチャートである。

図 1 3 0 は、RealPlayListの作成について説明するフローチャートである。

図 1 3 1 は、VirtualPlayListの作成について説明するフローチャートである。

図 1 3 2 は、PlayListの再生について説明するフローチャートである。

図 1 3 3 は、PlayListMarkの作成について説明するフローチャートである。

図 1 3 4 は、PlayListを再生する際の頭出し再生について説明するフローチャートである。

図 1 3 5 は、PlayListMarkのシンタクスを示す図である。

図 1 3 6 は、PlayListMarkのMark_typeを説明するための図である。

図 1 3 7 は、ClipMarkの他のシンタクスを示す図である。

図 1 3 8 は、ClipMarkのMark_typeを説明するための図である。

図 1 3 9 は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された実施例について、図面を参照して説明する。図 1 は、本発明を適用した記録再生装置 1 の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う記録部 2 の構成について説明する。記録再生装置 1 は、アナログデータ、又は、デジタルデータを入力し、記録することができる構成とされている。

端子 1 1 には、アナログのビデオ信号が、端子 1 2 には、アナログのオーディオ信号が、それぞれ入力される。端子 1 1 に入力されたビデオ信号は、解析部 1 4 と A V エンコーダ 1 5 に、それぞれ出力される。端子 1 2 に入力されたオーディオ信号は、解析部 1 4 と A V エンコーダ 1 5 に出力される。解析部 1 4 は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

A V エンコーダ 1 5 は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム (V)、符号化オーディオストリーム (A)、及び A V 同期等のシステム情報 (S) をマルチプレクサ 1 6 に出力する。

符号化ビデオストリームは、例えば、M P E G (Moving Picture Expert Group) 2 方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、M P E G - 1 方式により符号化されたオーディオストリームや、ドルビー A C 3 方式 (商標) により符号化されたオーディオストリーム等である。マルチプレクサ 1 6 は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ 1 7 を介して多重化ストリーム解析部 1 8 とソースパケッタイザ 1 9 に出力する。

多重化ストリームは、例えば、M P E G - 2 トランスポートストリームや M P E G - 2 プログラムストリームである。ソースパケッタイザ 1 9 は、入力された多重化ストリームを、そのストリームを記録させる記録媒体 1 0 0 のアプリケーションフォーマットに従って、ソースパケットから構成される A V ストリームに符号化する。A V ストリームは、E C C (誤り訂正) 符号化部 2 0 と変調部 2 1 で E C C 符号の付加と変調処理が施され、書込部 2 2 に出力される。書込部 2 2 は、制御部 2 3 から出力される制御信号に基づいて、記録媒体 1 0 0 に A V ストリームファイルを書き込む (記録する)。

ディジタルインタフェース又はディジタルテレビジョンチューナから入力されるディジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、これらは、トランスベアレントに記録する方式と、記録ビットレートを下げる等の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子24から制御部23へ入力される。

入力トランスポートストリームをトランスベアレントに記録する場合、端子13に入力されたトランスポートストリームは、スイッチ17を介して多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に入力される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、及びシステム情報(S)を抽出する。

デマルチプレクサ26により抽出されたストリーム(情報)の内、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、及び、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記

録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

以上のような記録再生装置 1 は、A V ストリームのファイルを記録媒体 100 に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部 23 により作成される。制御部 23 への入力情報は、解析部 14 からの動画像の特徴情報、多重化ストリーム解析部 18 からの A V ストリームの特徴情報、及び端子 24 から入力されるユーザからの指示情報である。

解析部 14 から供給される動画像の特徴情報は、A V エンコーダ 15 がビデオ信号を符号化する場合において、解析部 14 により生成されるものである。解析部 14 は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像（クリップマーク）に関係する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンチェンジ点や CM コマーシャルのスタート点・エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネイルも含まれる。更にオーディオ信号のステレオとモノラルの切り換え点や、無音区間などの情報も含まれる。

これらの画像の指示情報は、制御部 23 を介して、マルチプレクサ 16 へ入力される。マルチプレクサ 16 は、制御部 23 からクリップマークとして指定される符号化ピクチャを多重化する時に、その符号化ピクチャを A V ストリーム上で特定するための情報を制御部 23 に返す。具体的には、この情報は、ピクチャの P T S（プレゼンテーションタイムスタンプ）又はその符号化ピクチャの A V ストリーム上でのアドレス情報である。制御部 23 は、特徴的な画像の種類とその符号化ピクチャを A V ストリーム上で特定するための情報を関連付けて記憶する。

多重化ストリーム解析部 18 からの A V ストリームの特徴情報は、記録される A V ストリームの符号化情報に関する情報であり、解析部 18 により生成される。例えば、A V ストリーム内の I ピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、A V ストリームの符号化パラメータ、A V ストリームの中の符号化パラメータの変化点情報などが含まれる。また、端

子13から入力されるトランスポートストリームをトランスペアレントに記録する場合、多重化ストリーム解析部18は、入力トランスポートストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書込部22へ入力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのファイル)と、アプリケーションデータベース情報が再生部3により再生される場合、まず、制御部23は、読出部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読出部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29とECC復号部30の復調と誤り訂正処理を経て、制御部23へ入力される。

制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインタフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読出部28に指示する。読出部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、

所定の処理が施されることにより復調され、更にECC復号部30の処理を経て、ソースデバケッタイザ31出力される。

ソースデバケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26が処理可能なストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、及びAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

また、ユーザインタフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読出部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読出部28に指示する。

また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択した時(例えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネイル画像リストをユーザインタフェースに表示して、ユーザが、その中からある画像を選択することにより行われる)、制御部23は、Clip Informationの内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを読出部28へ指示する。すなわち、ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読出部28へ指示する。読出部28は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部29、ECC復号部30、ソースデバケッタイザ31の処理を経て、デマルチプレクサ26へ入力され、AVデコーダ27で復号されて、マーク点の

ピクチャのアドレスで示されるAVデータが再生される。

また、ユーザによって高速再生(Fast-forward playback)が指示されたとき、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のIピクチャデータを順次連続して読み出すように読出部28に指示する。

読出部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体100に記録されているAVストリームの編集をするときを説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

ユーザが、記録媒体100に記録されているAVストリームの一部を消去したいとき、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書込部22に指示する。

ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、且つ、それぞれの再生区間をシームレスに接続したい場合について説明する。このようなとき、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、更に、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

先ず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読出部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読出部28は、記録媒体100からデータを読み出し、そのデータ

は、復調部 29、ECC復号部 30、ソースデバケッタイザ 31を経て、デマルチプレクサ 26に出力される。

制御部 23は、デマルチプレクサ 26に入力されたデータを解析して、ビデオストリームの再エンコード方法（picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て）と、再多重化方式を決定し、その方式をAVエンコーダ 15とマルチプレクサ 16に供給する。

次に、デマルチプレクサ 26は、入力されたストリームをビデオストリーム（V）、オーディオストリーム（A）、及びシステム情報（S）に分離する。ビデオストリームは、AVデコーダ 27に入力されるデータとマルチプレクサ 16に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ 27で復号され、復号されたピクチャはAVエンコーダ 15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ 16に入力される。

マルチプレクサ 16は、制御部 23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部 20、変調部 21で処理されて、書込部 22に入力される。書込部 22は、制御部 23から供給される制御信号に基づいて、記録媒体 100にAVストリームを記録する。

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図 2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの 2つのレイヤを持つ。Volume Informationは、ディスク内の全てのClipとPlayListの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトとし、これをClipという。AVストリームファイルは、Clip AV stream fileといい、その付属情報は、Clip Information fileという。

1つのClip AV stream fileは、MPEG-2トランスポートストリームをアブ

リケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリポイント（Iピクチャ）は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemといい、これは、時間軸上のイン点（IN）とアウト点（OUT）の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図4Aは、Real PlayListのクリエイト(create：作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図4Bは、Real PlayListのディバイド(divide：分割)に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録（記録）し直したいといったようなときに行われる。この操作によ

り、Clipの内容が変更される（Clip自体が分割される）ことはない。

図4 Cは、Real PlayListのコンバイン(combine：結合)に関する図であり、2つのReal PlayListを結合して、1つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される（Clip自体が1つにされる）ことはない。

図5 Aは、Real PlayList全体のデリート(delete：削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

図5 Bは、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除されたとき、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図5 Cは、Real PlayListのミニマイズ(Minimize：最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayListにとって不必要なClipの、対応するストリーム部分は削除される。

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

このようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか？」といったメッセージなどを表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行、又は、キャンセルする。又は、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

次に、Virtual PlayListに対する操作について説明する。Virtual PlayListに

対して操作が行われたとしても、Clipの内容が変更されることはない。図6 A及び図6 Bは、アセンブル(Assemble)編集(IN-OUT編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

図6 Aに示したように、2つのReal PlayList 1, 2と、それぞれのReal Play Listに対応するClip 1, 2が存在している場合に、ユーザがReal PlayList 1内の所定の区間(In 1乃至Out 1までの区間: PlayItem 1)を再生区間として指示し、続けて再生する区間として、Real PlayList 2内の所定の区間(In 2乃至Out 2までの区間: PlayItem 2)を再生区間として指示したとき、図6 Bに示すように、PlayItem 1とPlayItem 2から構成される1つのVirtual PlayListが作成される。

次に、Virtual PlayListの再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加(append)、Virtual PlayListの中のPlayItemの削除などがある。また、Virtual PlayListそのものを削除することもできる。

図7は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブバスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインバスのAVストリームに、付加的なオーディオストリームが、サブバスとして付加される。

Real PlayListとVirtual PlayListで共通の操作として、図8に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList(図20などを参照して後述する)によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク(Mark)について説明する。マークは、図9に示されるように、Clip及びPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark(クリップマーク)と呼ばれる。Cl

ipMarkは、A Vストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。ClipMarkは、図1の例えば解析部14によって生成される。PlayListを再生する時、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、PlayListMark（プレイリストマーク）と呼ばれる。PlayListMarkは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、A Vストリームは何の変更もされない。

ClipMarkの別のフォーマットとして、ClipMarkが参照するピクチャをA Vストリームの中でのアドレススペースで指定するようにしてもよい。Clipにマークをセットすることは、マーク点のピクチャを示すアドレススペースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマーク点のピクチャを示すアドレススペースの情報を除去することである。したがって、マークの設定や削除により、A Vストリームは何の変更もされない。

次に、サムネイルについて説明する。サムネイルは、Volume、PlayList、及びClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（図示せず。）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクという。）を記録再生装置1の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを

想定している。

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、Playlistを表す代表画としてのサムネイルの2種類のサムネイルをメニューサムネイルという。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、全てのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlaylistには、複数のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnail）という。したがって、マークサムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図11は、Playlistに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネ

イル、PlayList、及びClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本例では、2種類のCPIを用いる。1つは、EP_mapであり、もう一つは、TU_mapである。

EP_mapは、エントリポイント (EP) データのリストであり、それはエレメンタリーストリーム及びトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ (PTS) と、このPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができる時、EP_mapが作成され、ディスクに記録される。

TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット (TU) データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができない時、TU_mapが作成され、ディスクに記録される。

STCInfoは、MPEG-2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。仮に、AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れる可能性がある。このため、AVストリーム上の所定の時刻をPTSベースで指すとき、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。

更に、このPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットでは、STC-sequenceといい、そのインデックスをSTC-sequence-idという。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。再生装置にとって、AVストリームのデコードに先だち、そのAVストリームの内容が分かることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートバケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど）などの情報である。

この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MPEG-2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートバケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化する等である。

ProgramInfoは、A Vストリームファイルの中でのプログラム内容の変化点の情報をストアする。A Vストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceという。Program-sequenceは、EP_mapを持つA Vストリームファイルで使用するものであり、TU_mapを持つA Vストリームファイルではオプションである。

本例では、セルフエンコードのストリームフォーマット (SESF) を定義する。SESFは、アナログ入力信号を符号化する目的、及びデジタル入力信号 (例えば D V) をデコードしてからMPEG-2トランスポートストリームに符号化する場合に用いられる。

SESFは、MPEG-2トランスポートストリーム及びA Vストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式の内のいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

或いは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

例えば、入力ストリームがISDB (日本のデジタルBS放送の規格名称) 準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、このSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にI

SDBフォーマットに準拠しなければならない。

ディジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP_mapが作成されてディスクに記録される。

又は、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にTU_mapが作成されてディスクに記録される。

次に、ディレクトリとファイルについて説明する。以下、記録再生装置1をDVR (Digital Video Recording) と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、及び“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしてもよいが、これらは、ここでのアプリケーションフォーマットでは、無視されとする。

“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。“M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。“DATA”ディレクトリは、ディジタルTV放送などのデータ放送のファイルがストアされる。

“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、“DVR”ディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。“DVR”ディレクトリの下には、ただ1つのinfo.dvrがなければならない。

ファイル名は、info.dvrに固定されとする。"menu.thmb"ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"mark.thmb"ファイルは、マークサムネイル画像に関連する情報をストアする。"DVR"ディレクトリの下には、0又は1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"PLAYLIST"ディレクトリは、2種類のPlayListファイルをストアするものであり、これらは、Real PlayListとVirtual PlayListである。"xxxxx.rpls"ファイルは、1つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、1つのファイルが作られる。ファイル名は、"xxxxx.rpls"である。ここで、"xxxxx"は、5個の0乃至9まで数字である。ファイル拡張子は、"rpls"でなければならないとする。

"yyyyy.vpls"ファイルは、1つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、5個の0乃至9まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。"zzzzz.clpi"ファイルは、1つのAVストリームファイル(Clip AV stream file 又は Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張

張子は、"m2ts"でなければならないとする。

"DATA" ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。先ず、"info.dvr" ファイルについて説明する。図15は、"info.dvr" ファイルのシンタクスを示す図である。"info.dvr" ファイルは、3個のオブジェクトから構成され、これらは、DVRVolume()、TableOfPlayLists()、及びMakersPrivateData()である。

図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakersPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word（パディングワード）は、info.dvrのシンタクスに従って挿入される。N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

DVRVolume()は、ボリューム（ディスク）の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVRVolume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

ResumeVolume()は、ボリュームの中で最後に再生したReal Playlist又はVirtual Playlistのファイル名を記憶している。但し、Real Playlist又はVirtual Playlistの再生をユーザが中断した時の再生位置は、PlaylistMark()において定義されるresume-markにストアされる（図42、図43）。

図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResu

meVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal Playlist又はVirtual Playlistのファイル名を示す。

図16に示したDVRVolume()のシンタクスの中の、UIAppInfoVolume は、ボリュームについてのユーザインタフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、これはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが0にセットされているとき、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、又は、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlaylistの一覧を表示させる。それぞれのPlaylistの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlaylist()の中に定義されるplayback_control_flagによって示される。

P I Nは、4個の0乃至9までの数字で構成され、それぞれの数字は、I S O / I E C 6 4 6に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

次に、図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、Playlist(Real PlaylistとVirtual Playlist)のファイル名をストアする。ボリュームに記録されている全てのPlaylistファイルは、TableOfPlayLists()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlaylistのデフォルトの再生順序を示す。

図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、I S O 6 4 6に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、Playlist_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlaylistの数に等しくなければならない。Playlist_file_nameの10バイトの数字は、Playlistのファイル名を示す。

図21は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlaylist(後述)を含ませた構成とされている。このように、UIAppinfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

図 15 に示した info.dvr のシンタクス内の MakersPrivateData について説明する。MakersPrivateData は、記録再生装置 1 のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData() の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義したメーカーを識別するために標準化された maker_ID を持つ。MakersPrivateData() は、1 つ以上の maker_ID を含んでもよい。

所定のメーカーが、プライベートデータを挿入したい時に、既に他のメーカーのプライベートデータが MakersPrivateData() に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータを MakersPrivateData() の中に追加するようにする。このように、ここでは、複数のメーカーのプライベートデータが、1 つの MakersPrivateData() に含まれることが可能であるようにする。

図 22 は、MakersPrivateData のシンタクスを示す図である。図 22 に示した MakersPrivateData のシンタクスについて説明するに、version_number は、この MakersPrivateData() のバージョンナンバを示す 4 個のキャラクター文字を示す。version_number は、ISO 646 に従って、“0045” と符号化されなければならない。length は、この length フィールドの直後から MakersPrivateData() の最後まで MakersPrivateData() のバイト数を示す 32 ビットの符号なし整数を示す。

mpd_blocks_start_address は、MakersPrivateData() の先頭のバイトからの相対バイト数を単位として、最初の mpd_block() の先頭バイトアドレスを示す。相対バイト数は 0 からカウントされる。number_of_maker_entries は、MakersPrivateData() の中に含まれているメーカープライベートデータのエン트리数を与える 16 ビットの符号なし整数である。MakersPrivateData() の中に、同じ maker_ID の値を持つメーカープライベートデータが 2 個以上存在してはならない。

mpd_block_size は、1024 バイトを単位として、1 つの mpd_block の大きさを与える 16 ビットの符号なし整数である。例えば、mpd_block_size=1 ならば、これは 1 つの mpd_block の大きさが 1024 バイトであることを示す。number_of_mpd_blocks は、MakersPrivateData() の中に含まれる mpd_block の数を与える 16 ビットの符号なし整数である。maker_ID は、そのメーカープライベートデータを作成

したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxx.rplsとyyyy.vplsについて説明する。図23は、xxxx.rpls (Real PlayList)、又は、yyyy.vpls (Virtual PlayList) のシンタクスを示す図である。xxxx.rplsとyyyy.vplsは、同一のシンタクス構成を持つ。xxxx.rplsとyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、これらは、PlayList()、PlayListMark()、及びMakersPrivateData()である。

PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakersPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

padding_word (パディングワード) は、PlayListファイルのシンタクスに従って挿入され、N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

ここで、既に、簡便に説明したが、PlayListについて更に説明する。ディスク内にある全てのReal PlayListによって、Bridge-Clip（後述）を除く全てのClipの中の再生区間が参照されていなければならない。且つ、2つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図24A～図24Cを参照して更に説明するに、図24Aに示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図24Bに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、どれかしらのReal PlayListを参照することにより、必ず視聴することが可能である。

図24Cに示したように、Virtual PlayListの再生区間は、Real PlayListの再生区間又はBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual PlayListにも参照されないBridge-Clipがディスクの中に存在してはならない。

Real PlayListは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual PlayListは、PlayItemのリストを含み、PlayList()の中に示されるCPI_typeがEP_map typeであり、且つPlayList_typeが0（ビデオとオーディオを含むPlayList）である場合、Virtual PlayListは、1つのSubPlayItemを含むことができる。本実施例におけるPlayList()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual PlayListが持つSubPlayItemの数は、0又は1でなければならない。

次に、PlayListについて説明する。図25は、PlayListのシンタクスを示す図である。図25に示したPlayListのシンタクスを説明するに、version_numberは、このPlayList()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までのPlayList()のバイト数を示す32ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

CPI_typeは、1ビットのフラグであり、PlayItem()及びSubPlayItem()によって

参照されるClipのCPI_typeの値を示す。1つのPlayListによって参照される全てのClipは、これらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0又は1である。付加的なオーディオストリームのパス（オーディオストリームパス）は、サブパスの一種である。

次に、図25に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインタフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、8ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

name_lengthは、8ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

record_time_and_dateは、PlayListが記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

durationは、PlayListの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、“0x014530”

と符号化される。

valid_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、"0x20010507"と符号化される。

maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置1）の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンスによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、このPlayListを視聴することができる。

write_protect_flagは、図28Aにテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去及び変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、又は上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

write_protect_flagが0にセットされているReal PlayListが存在し、且つ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、又は、このReal PlayListを"Minimize" する。

is_played_flagは、図28Bに示すように、フラグが1にセットされている場

合、そのPlayListは、記録されてから一度は再生されたことを示し、0にセットされている場合、このPlayListは、記録されてから一度も再生されたことがないことを示す。

archiveは、図28Cに示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_indexのフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF であるとき、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

次に、PlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、及び、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

PlayListが2つ以上のPlayItemから構成される時、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップ又はオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図29に示す。

図30は、PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time (図の中でIN_time1と示されているものは、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time (図の

中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN_time(図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのOUT_time(図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるS T C連続区間上の時間を指している。

図31に示すように、Playlist()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip A Vストリーム上の時間を指している。

PlayItemのシンタクスは、図32に示すようになる。図32に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip A V streamを示していなければならない。

STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するS T C連続区間のSTC_sequence_idを示す。Playlist()の中で指定されるCPI_typeがTU_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図33に示すように、Playlist()において定義されるCPI_typeによって異なる。

OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図34に示すように、Playlist()において定義されるCPI_typeによって異なる。

Connection_Conditionは、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36A～図36Dは、図35に示したConnection_Conditionの各状態について説明する図であ

る。

次に、BridgeSequenceInfoについて、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information file (図45) を指定するBridge_Clip_Information_file_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースバケットのアドレスであり、このソースバケットに続いてBridge-Clip AV streamファイルの最初のソースバケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。更に現在のPlayItemが参照するClip AV stream上のソースバケットのアドレスであり、このソースバケットの前にBridge-Clip AV streamファイルの最後のソースバケットが接続される。このアドレスは、RSPN_enter_to_current_Clipという。

図37において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースバケットのアドレスを示す。このアドレスは、ClipInfo() (図46) の中で定義される。

図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースバケットの相対アドレスであり、このソースバケットに続いてBridge-Clip AV streamファイルの最初のソースバケットが接続される。RSPN_exit_from_previous_Clipは、ソースバケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースバケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、Playlist()のCPI_typeがEP_map typeである場合だけに許される。ここでは、SubPlayItemはオーディオのアフレコの目的のためだけに使用されとする。SubPlayItem()は、次に示すデータを含む。まず、Playlistの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。更に、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、これはPlaylistの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItem

に対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

sync_start_PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

SubPath_OUT_timeの32ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値の上位32ビットを示す。

$$\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$$

ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem

_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、これはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される(後述)。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル)は、図45に示すように6個のオブジェクトから構成される。これらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、及びMakersPrivateData()である。AVストリーム(Clip AVストリーム又はBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の”zzzzz”が使用される。

図45に示したzzzzz.clpi (Clip information fileファイル)のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数は0から

カウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数は0からカウントされる。

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。MakersPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData ()の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word (パディングワード) は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。N 1, N 2, N 3, N 4、及びN 5は、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図46は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリーム又はBridge-Clip AVストリームファイル) の属性情報をストアする。

図46に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset_SPNの32ビットのフィールドは、AVストリーム (Clip AVストリーム又はBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

図48に示すように、AVストリームファイルのはじめの部分が編集によって

消去された時、offset_SPNは、0以外の値をとってもよい。本例では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_xxx（xxxは変形する。例、RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、A Vストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

A Vストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}$$

図48に、offset_SPNが4である場合の例を示す。

TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ（書込部22）へ又はDVRドライブ（読出部28）からのA Vストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するA Vストリームが記録された時の日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal（BCD）で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal（BCD）で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

time_controlled_flagのフラグは、A Vストリームファイルの記録モードを示す。このtime_controlled_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\begin{aligned} \text{TS\_average\_rate} * 192 / 188 * (t - \text{start\_time}) - \alpha &\leq \text{size\_Clip}(t) \\ &\leq \text{TS\_average\_rate} * 192 / 188 * (t - \text{start\_time}) + \alpha \end{aligned}$$

ここで、TS_average_rateは、A Vストリームファイルのトランスポートストリ

ームの平均ビットレートをbytes/second の単位で表したものである。

また、上式において、 t は、秒単位で表される時間を示し、start_timeは、A Vストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_Clip(t)は、時刻 t におけるA Vストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻 t までに10個のソースパケットが記録された場合、size_Clip(t)は10*192バイトである。 α は、TS_average_rateに依存する定数である。

time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とA Vストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランスポートパケットを符号化する。

RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip A V streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip A V streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip A V streamファイルの中での絶対アドレスは、上述した

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset\_SPN}$$

に基づいて算出される。

reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1である時、format_id

ntifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_servecce_ID_validのフラグが1である場合、servecce_IDのフィールドが有効であることを示す。

is_country_code_validのフラグが1である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration descriptor (ISO/IEC 13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

servecce_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservecce_IDの値を示す。country_codeの24ビットのフィールドは、ISO 3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO 8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x50 0x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

format_identifier、original_network_ID、transport_stream_ID、servecce_ID、country_code、及びstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50A及び図50Bは、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（但し、後述するように、Clipの最大時間長は制限されている）。したがって、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、 $N(N>0)$ 個のSTC不連続点を含む場合、Clipのシステムタイムベースは、 $(N+1)$ 個のSTC_sequenceに分割される。

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除く k 番目（ $k \geq 0$ ）のSTC_sequenceは、 k 番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、 $(k+1)$ 番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0をセットしてもよい。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-l

oopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしてもよい。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset\_SPN}$$

により算出される。

次に、図45に示したzzzz.Clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴を持つ時間区間をprogram_sequenceと呼ぶ。先ず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が増減しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。更に、オーディオエレメンタリーストリームの数が増減しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明するに、version_numberは、このProgramInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgr

amInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0にセットされてもよい。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは1以上の値でなければならない。

number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}$$

により算出される。シンタックスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオス

トリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図55は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明するに、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図45に示したzzzzz.Clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、これらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中

のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_map又は1つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

次に、図65に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、及び、RSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでもよい。

オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、及びRSPN_E

P_startというデータを持つ。stream_PIDは、オーディオストリームを送送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでもよい。

EP_mapとSTC_Infoの関係を説明するに、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較することにより、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる(図68を参照)。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明するに、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。又は、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数に持つfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム(ビデオ又はオーディオストリ

ーム)を伝送するトランスポートパケットのPIDを示す。EP_typeが0 ('video')に等しい場合、そのエレメンタリストリームはビデオストリームでなければならない。また、EP_typeが1 ('audio')に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k): この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない。XとYは、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取ってもよい。

図7.2は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図7.2に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。又は、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第1バイト目を含むソースパケットの相対アドレスを示す。

RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset

_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

次に、TU_mapについて、図73を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitという。

AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$\text{TU\_start\_time}(k) = \text{offset\_time} + k * \text{time\_unit\_size}$$

TU_start_time(k)は、45 kHzの精度を持つ。

図74は、TU_mapのシンタクスを示す図である。図74に示したTU_mapのシンタクスを説明するに、offset_timeの32ビット長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27 MHz精度のアライバルタイムクロックから導き出される45 kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeは0にセットされなければならない。

time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27 MHz精度のアライバルタイムクロックから導き出される45 kHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size<=45000) にすることがよい。number_of_time_unit_entriesの32ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリ数を示

す。

RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

図45に示したzzzzz.Clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number_of_clip_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number_of_clip_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

mark_time_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図77に示すように、Playlist()の中のCPI_typeにより異なる。

STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この8ビットのフィールドは、mark_time_stampが置かれているところのSTC連続区間の

STC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この8ビットのフィールドは何も意味を持たず、0にセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

図78は、図75に代わるClipMarkの他のシンタクスを示す図であり、図79は、その場合における、図76に代わるmark_typeのテーブルの例を示す。reserved_for_maker_IDは、mark_typeが、0xC0から0xFFの値を示す時に、そのmark_typeを定義しているメーカーのメーカーIDを示す16ビットのフィールドである。メーカーIDは、DVRフォーマットライセンサが指定する。mark_entry()は、マーク点に指定されたポイントを示す情報であり、そのシンタクスの詳細は後述する。representative_picture_entry()は、mark_entry()によって示されるマークを代表する画像のポイントを示す情報であり、そのシンタクスの詳細は後述する。

ClipMarkは、ユーザがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤは、GUI(グラフィカルユーザインタフェース)を使用して、ClipMarkの情報をユーザに提示する。ClipMarkの情報を視覚的に表示するためには、mark_entry()が示すピクチャよりもむしろrepresentative_picture_entry()が示すピクチャを示したほうがよい。

図80に、mark_entry()とrepresentative_picture_entry()の例を示す。例え

ば、あるプログラムが開始してから、しばらくした後（数秒後）、そのプログラムの番組名（タイトル）が表示されとする。ClipMarkを作るときは、mark_entry()は、そのプログラムの開始ポイントに置き、representative_picture_entry()は、そのプログラムの番組名（タイトル）が表示されるポイントに置くようにしてもよい。

D V R プレーヤは、representative_picture_entryの画像をG U Iに表示し、ユーザがその画像を指定すると、D V R プレーヤは、mark_entryの置かれたポイントから再生を開始する。

mark_entry() 及び representative_picture_entry()のシンタクスを、図 8 1 に示す。

mark_time_stampは、3 2 ビットフィールドであり、mark_entry()の場合はマークが指定されたポイントを示すタイムスタンプをストアし、またrepresentative_picture_entry()の場合、mark_entry()によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

次に、ClipMarkを指定するために、P T Sによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合のmark_entry() と representative_picture_entry()のシンタクスの例を図 8 2 に示す。

RSPN_ref_EP_startは、 mark_entry()の場合、A Vストリームの中でマーク点のピクチャをデコードするためのストリームのエントリポイントを示すソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表するピクチャをデコードするためのストリームのエントリポイントを示すソースパケットの相対アドレスを示す。RSPN_ref_EP_startの値は、EP_mapの中にRSPN_EP_startとしてストアされていなければならない、且つ、そのRSPN_EP_startに対応するPTS_EP_startの値は、EP_mapの中で、マーク点のピクチャのP T Sより過去で最も近い値でなければならない。

offset_num_picturesは、3 2 ビットのフィールドであり、RSPN_ref_EP_startにより参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、0 からカウントされる。図 8 3 の例の場合、offset_num_picturesは 6 となる。

次に、ClipMarkを指定するために、アドレスベースの情報を使用する場合のmark_entry() と representative_picture_entry()のシンタックスの別の例を図84に示す。

RSPN_mark_pointは、mark_entry()の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表する符号化ピクチャの第1バイト目を含むソースパケットの相対アドレスを示す。

RSPN_mark_pointは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義されるoffset_SPNの値を初期値としてカウントされる。

図85を用いて、ClipMarkとEP_mapの関係を説明する。この例の場合、EP_mapが、エントリポイントのアドレスとしてI0, I1, Inを指定しており、これらのアドレスからシーケンスヘッダに続くIピクチャが開始しているとする。ClipMarkが、あるマークのアドレスとして、M1を指定している時、そのソースパケットから開始しているピクチャをデコードできるためには、M1のアドレスより前で最も近いエントリポイントであるI1からデータを読み出し開始すればよい。

MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイル又はmark.thmbファイルにストアされる。これらのファイルは同じシンタックス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、及び、それぞれのPlayListを代表する画像をストアする。全てのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。全てのPlayList及びClipに対する全てのマークサムネイルは、ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除さ

れるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

図86は、menu.thmbとmark.thmbのシンタクスを示す図であり、図87は、図86に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図87に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数は0からカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデクス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、及びClipMark()の中のref_thumbnail_indexによって参照される。

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す

8ビットの符号なし整数で、図88に示すような値をとる。表中のDCFとPNGは”menu.thmb”内でのみ許される。マークサムネイルは、値”0x00”(MPEG-2 Video I-picture)をとらなければならない。

picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。tn_block番号は、0から始まり、tn_blockのfor-ループ中の変数kの値に関係する。

x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中の全てのtn_blockは、同じサイズ(固定長)であり、その大きさはtn_block_sizeによって定義される。

図89A及び図89Bは、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図89A及び図89Bのように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、”M2TS”ディレクトリ(図14)にストアされる。AVストリームファイルには、2つのタイプがあり、これらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

まず、DVR MPEG-2トランスポートストリームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図90に示すようになっている。AVストリームファイルは、DVR MPEG-2トランスポートスト

リームの構造を持つ。DVR MPEG-2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。1つのソースパケットは、TP_extra_headerとトランスポートパケットからなる。TP_extra_headerは、4バイト長であり、またトランスポートパケットは、188バイト長である。

1つのAligned unitは、32個のソースパケットからなる。DVR MPEG-2トランスポートストリームの中の最後のAligned unitも、また32個のソースパケットからなる。よって、DVR MPEG-2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が32の倍数でない時、ヌルパケット (PID=0x1FFFのトランスポートパケット) を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG-2トランスポートストリームに余分な情報を付加してはならない。

図91に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図91に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG-2トランスポートストリームは、フルトランスポートストリーム又はパーシャルトランスポートストリームである。入力されるMPEG-2トランスポートストリームは、ISO/IEC 13818-1又はISO/IEC 13818-9に従っていなければならない。MPEG-2トランスポートストリームの*i*番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder) 51とソースパケットタイザ (source packetizer) 54へ、時刻*t* (*i*) に同時に入力される。Rpkは、トランスポートパケットの入力レートの瞬時的な最大値である。

27MHz PLL 52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Prog

ram Clock Reference)の値にロックされる。アライバルタイムクロックカウンタ (arrival time clock counter) 53は、27MHzの周波数のパルスのカウントするバイナリカウンタである。Arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter 53のカウント値である。

source packetizer 54は、全てのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がTSD 51とソースパケットタイザ54の両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \% 230$$

2つの連続して入力されるトランスポートパケットの時間間隔が、230/27000000秒(約40秒)以上になる場合、その2つのトランスポートパケットのarrival_time_stampの差分は、230/270000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

スムージングバッファ (smoothing buffer) 55は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファ55は、オーバーフローしてはならない。Rmaxは、スムージングバッファ55が空でない時のスムージングバッファ55からのソースパケットの出力ビットレートである。スムージングバッファ55が空である時、スムージングバッファ55からの出力ビットレートは0である。

次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = \text{TS\_recording\_rate} * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording

_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照してもよい。

入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファ55の大きさ(smoothing buffer size)は0である。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファ55の大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照してもよい。

記録機(レコーダ)及び記録再生装置1(プレーヤ)は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536バイトである。

次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図92は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

27MHz Xtal(クリスタル発振器)61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 ± 30 ppm (27000000 $\pm$ 810 Hz)でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスのカウントするバイナリーカウンタである。arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter 62のカウント値である。

smoothing buffer 64において、Rmaxは、スムージングバッファ64がフルでない時のスムージングバッファ64へのソースパケットの入力ビットレートである。スムージングバッファ64がフルである時、スムージングバッファ64への入力ビットレートは0である。

MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スム

ージングバッファ 64 から引き抜かれる。Rpkは、トランスポートバケットレート
の瞬時的な最大値である。スムージングバッファ 64 は、アンダーフローしては
ならない。

DVR MPEG-2 トランスポートストリームのプレーヤモデルのパラメータ
については、上述したDVR MPEG-2 トランスポートストリームのレコーダ
モデルのパラメータと同一である。

図 9 3 は、Source packetのシンタクスを示す図である。transport_packet()
は、ISO/IEC 13818-1 で規定されるMPEG-2 トランスポートバ
ケットである。図 9 3 に示したSource packetのシンタクス内のTP_Extra_header
のシンタクスを図 9 4 に示す。図 9 4 に示したTP_Extra_headerのシンタクスにつ
いて説明するに、copy_permission_indicatorは、トランスポートバケットのペイ
ロードのコピー制限を表す整数である。コピー制限は、copy free、no more cop
y、copy once、又はcopy prohibitedとすることができる。図 9 5 は、copy_perm
ission_indicatorの値と、それらによって指定されるモードの関係を示す。

copy_permission_indicatorは、全てのトランスポートバケットに付加される。
IEEE 1394 デジタルインタフェースを使用して入力トランスポートスト
リームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochron
ous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けても
よい。IEEE 1394 デジタルインタフェースを使用しないで入力トランス
ポートストリームを記録する場合、copy_permission_indicatorの値は、トランス
ポートバケットの中に埋め込まれたCCIの値に関連付けてもよい。アナログ信
号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナロ
グ信号のCGMS-Aの値に関連付けてもよい。

arrival_time_stampは、次式

$$\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \% 230$$

において、arrival_time_stampによって指定される値を持つ整数値である。

Clip AVストリームの定義をするに、Clip AVストリームは、上述したよう
な定義がされるDVR MPEG-2 トランスポートストリームの構造を持たねば
ならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加

しなければならない。Clip AVストリームの中にシステムタイムベース（STCベース）の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG-2トランスポートストリームの中にシステムタイムベース（STCベース）の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG-2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間).と規定している。

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならない、且つ後述するDVR-STDに従わなければならない。

本例においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection

_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図9 6は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図9 6においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図9 6に示したTS 1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータからなる。

TS 1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図9 6においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS 1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

また、図9 6におけるTS 2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータからなる。TS 2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS 2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time (図9 6においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図9 7は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影を付けて示されている。図9 7におけるTS 1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータからなる。TS 1のClip1の影を付けられたスト

リームデータは、先行するPlayItemのIN_time（図97においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図97におけるTS2は、Clip2（Clip AVストリーム）の影を付けられたストリームデータからなる。

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time（図97においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図96と図97において、TS1とTS2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1及び／又はTS2の中に、上記以外のエレメンタリーストリーム又はライブストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図98は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1（Clip1のOUT_time）の後とIN_time2（Clip2のIN_time）の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

図98に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図99に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図98のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなる。そして、そのビデオストリームは先

行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

同様に、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなる。そして、そのビデオストリームは、正しくデコード開始することができて、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG-2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならない、それ以外のピクチャはオリジナルのClipからコピーすることができる。

図98に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図100に示す。Clip1のビデオストリームは、図98のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなり、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様に、Clip2のビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなり、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明するに、先ず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてIピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあってはならない。接続点において、トップーボトムのフィールドシーケンスは連続でなければならない。3-2ブルダウンを使用するエンコードの場合は、"top_field_first" 及び "repeat_first"

_field”フラグを書き換える必要があるかもしれない，又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明するに、TS 1とTS 2のオーディオのサンプリング周波数は、同じでなければならない。TS 1とTS 2のオーディオの符号化方法（例、MPEG 1レイヤ2，AC-3，SESF LPCM，AAC）は、同じでなければならない。

次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS 1のオーディオストリームの最後のオーディオフレームは、TS 1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS 2のオーディオストリームの最初のオーディオフレームは、TS 2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図101に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあってもよい。TS 2のエレメンタリーストリームを伝送する最初のバケットは、ビデオバケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

Clip及びBridge-Clipの制限について説明するに、TS 1とTS 2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS 1の最後のソースバケットとTS 2の最初のソースバケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS 2の最初のソースバケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースバケットは、Clip 1の中のどのソースバケットでもよい。これは、Aligned unitの境界である必要はない。BridgeSequenceInfo()にお

いて定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip 2の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。

PlayItemの制限について説明するに、先行するPlayItemのOUT_time (図9 6、図9 7において示されるOUT_time1) は、TS 1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time (図9 6、図9 7において示されるIN_time2) は、TS 2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clipを使用する場合のデータアロケーションの制限について、図1 0 2を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip A Vストリームファイル) とClip2 (Clip A Vストリームファイル) に接続されるBridge-Clip A Vストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

RSPN_exit_from_previous_Clip以前のClip 1 (Clip A Vストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip A Vストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip 2 (Clip A Vストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図1 0 3を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip A Vストリームファイル) の最後の部分とClip2 (Clip A Vストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

Clip1 (Clip A Vストリームファイル) の最後のストリーム部分が、ハーフフ

ラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip AV ストリームファイル) の最初のストリーム部分が、ハーフラグメント以上の連続領域に配置されていなければならない。

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG-2 トランスポートストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルでもある。

DVR-STDモデルを図104に示す。図104に示したモデルには、DVR MPEG-2 トランスポートストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, On及びPn(k)の表記方法は、ISO/IEC 13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデックス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

DVR-STDのデコーディングプロセスについて説明する。単一のDVR M

P E G - 2 トランスポートストリームを再生している間は、トランスポートパケットをTB1, TBn又はTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsys及びBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-S T Dと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-S T Dと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図9.6に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

図105は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図105においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図105においてATC1で示される）と同じでない。

また、TS2のシステムタイムベースの時間軸（図105においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図105においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあってもよい。

DVR-STD への入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn 又はTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

TS1の残りのパケットは、TS_recording_rate (TS1) のビットレートでD

VR-STDのTB_n又はTBsysのバッファへ入力されなければならない。
 ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T2である。したがって、時刻T1からT2までの区間では、ソースパケットのarrival_time_stampは無視される。

N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケットのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$\Delta T1 = T2 - T1 = N1 / TS\_recording\_rate(TS1)$$

時刻T1乃至T2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

T2の時刻において、arrival time clock counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1, TB_n 又はTBsysのバッファへの入力タイミングは、TS2のソースパケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

付加的なオーディオバッファリング及びシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻T1からT2までの区間の入力データを処理することができるよう、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図105ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、

T S 1 の最後のビデオプレゼンテーションユニットに対応する S T C 1 上の P T S であり、PTS2startは、T S 2 の最初のビデオプレゼンテーションユニットに対応する S T C 2 上の P T S であり、Tppは、T S 1 の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$STC\_delta = PTS1end + Tpp - PTS2start$$

オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあってもよく、それは0乃至2オーディオフレーム未満である（図105に図示されている"audio overlap"を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

D V D - S T D のシステムタイムクロックについて説明するに、時刻T5において、T S 1 の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T2からT5の間にオーバーラップしていてもよい。この区間では、D V R - S T D は、システムタイムクロックを古いタイムベースの値（S T C 1）と新しいタイムベースの値（S T C 2）の間で切り換える。S T C 2 の値は、次式により算出される。

$$STC2 = STC1 - STC\_delta$$

バッファリングの連続性について説明する。STC11video_endは、T S 1 の最後のビデオパケットの最後のバイトがD V R - S T D のT B 1 へ到着する時のシステムタイムベースS T C 1 上のS T C の値である。STC22video_startは、T S 2 の最初のビデオパケットの最初のバイトがD V R - S T D のT B 1 へ到着する時のシステムタイムベースS T C 2 上のS T C の値である。STC21video_endは、STC11video_end の値をシステムタイムベースS T C 2 上の値に換算した値である。STC21video_endは、次式により算出される。

$$STC21video\_end = STC11video\_end - STC\_delta$$

D V R - S T D に従うために、次の2つの条件を満たすことが要求される。先

ず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC22video\_start > STC21video\_end + \Delta T1$$

この不等式が満たされるように、Clip1及び／又は、Clip2の部分的なストリームを再エンコード及び／又は、再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフロー及びアンダーフローさせてはならない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

なお、本例は、多重化ストリームとしてMPEG-2トランスポートストリームを例にして説明しているが、これに限らず、MPEG-2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

次に、mark_entry()及びrepresentative_picture_entry()のシンタクスが、図81に示されるような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図106のフローチャートを参照して、説明する。

最初にステップS1において、記録再生装置1の制御部23は、記録媒体100から、DVRトランスポートストリームファイルのデータベースであるEP_Map（図70）、STC_Info（図52）、Program_Info（図54）、及びClipMark（図78）を読み出す。

ステップS2において、制御部23は、ClipMark（図78）のrepresentative_picture_entry（図81）、又はref_thumbnail_indexで参照されるピクチャから

サムネイルのリストを作成し、ユーザインタフェース入出力としての端子24から出力し、GUIのメニュー画面上に表示させる。この場合、ref_thumbnail_indexが有効な値を持つ場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップS3において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル画像を選択することで行われる。制御部23は、この選択操作に対応して、指定されたサムネイルに対応付けられているマーク点を取得する。

ステップS4において、制御部23は、ステップS3で指定されたmark_entry(図81)のmark_time_stampのPTSと、STC_sequence_idを取得する。

ステップS5において、制御部23は、STC_Info(図52)から、ステップS4で取得したSTC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

ステップS6において、制御部23は、ステップS5で取得したSTC時間軸が開始するパケット番号と、ステップS4で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、且つ、最も近いエントリポイント(Iピクチャ)のあるソースパケット番号を取得する。

ステップS7において、制御部23は、ステップS6で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

ステップS8において、制御部23は、AVデコーダ27を制御し、ステップS4で取得したマーク点のPTSのピクチャから表示を開始させる。

以上の動作を、図107乃至109を参照して更に説明する。

今、図107に示されているように、DVRトランスポートストリームファイルは、STC_sequence_id=id0のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Aのソースパケット番号より小さいものとする。そして、ソースパケット番号BからCまでの間に、CM(コマーシャル)が挿入されているものとする。

このとき、図70に示されるEP_Mapに対応するEP_Mapには、図108に示され

るように、RSPN_EP_startで示されるA, B, Cに対応して、それぞれのPTSが、PTS_EP_startとして、PTS(A), PTS(B), PTS(C)として登録される。

また、図109に示されるように、図78のClipMarkに対応するClipMarkには、図109に示されるように、シーンスタート、CMスタート、及びCMエンドを表すマークタイプ(図79) 0x92, 0x94, 0x95の値に対応して、mark_entryとrepresentative_picture_entryが記録される。

mark_entryのMark_Time_stampとしては、シーンスタート、CMスタート、及びCMエンドに対応して、それぞれPTS(a1), PTS(b0), PTS(c0)が登録されており、それぞれのSTC_sequence_idは、いずれもid0とされている。

同様に、Representative_picture_entryのMark_Time_stampとして、シーンスタート、CMスタート、及びCMエンドに対応して、それぞれPTS(a2), PTS(b0), PTS(c0)が登録されており、それらはいずれもSTC_sequence_idが、id0とされている。

PTS(A) < PTS(a1)の場合、ステップS6において、パケット番号Aが取得され、ステップS7において、パケット番号Aから始まるトランスポートストリームが、AVデコーダ27に供給され、ステップS8において、PTS(a1)のピクチャから表示が開始される。

次に、図110のフローチャートを参照して、mark_entryとrepresentative_picture_entryのシンタクスが、図81に示されるような構成である場合におけるCMスキップ再生の処理について、図110のフローチャートを参照して説明する。

ステップS21において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、及びClipMark(図78)を記録媒体100から読み出す。ステップS22において、ユーザは、ユーザインタフェース入出力としての端子24からCMスキップ再生を指定する。

ステップS23において、制御部23は、マークタイプ(図79)がCM開始点(0x94)であるマーク情報のPTSと、CM終了点(0x95)であるマーク情報のPTS、並びに対応するSTC_sequence_idを取得する(図81)。

ステップS 2 4において、制御部2 3は、STC_Info (図5 2) からCM開始点と終了点の、STC_sequence_idに対応するS T C時間軸が開始するソースパケット番号を取得する。

ステップS 2 5において、制御部2 3は、記録媒体1 0 0からトランスポートストリームを読み出させ、それをA Vデコーダ2 7に供給し、デコードを開始させる。

ステップS 2 6において、制御部2 3は、現在の表示画像がCM開始点のP T Sの画像か否かを調べる。現在の表示画像がCM開始点のP T Sの画像でない場合には、ステップS 2 7に進み、制御部2 3は、画像の表示が継続される。その後、処理はステップS 2 5に戻り、それ以降の処理が繰り返し実行される。

ステップS 2 6において、現在の表示画像がCM開始点のP T Sの画像であると判定された場合、ステップS 2 8に進み、制御部2 3は、A Vデコーダ2 7を制御し、デコード及び表示を停止させる。

次に、ステップS 2 9において、制御部2 3は、CM終了点のSTC_sequence_idに対応するS T C時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップS 2 3の処理で取得したCM終了点のP T Sとから、その点のP T Sより時間的に前で、且つ、最も近いエントリポイントのあるソースパケット番号を取得する。

ステップS 3 0において、制御部2 3は、ステップS 2 9の処理で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、A Vデコーダ2 7に供給させる。

ステップS 3 1において、制御部2 3は、A Vデコーダ2 7を制御し、CM終了点のP T Sのピクチャから表示を再開させる。

図1 0 7乃至図1 0 9を参照して、以上の動作を更に説明すると、CM開始点とCM終了点は、この例の場合、STC_sequence_id=id0という共通のS T C時間軸上に存在し、そのS T C時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Aより小さいものとされている。

トランスポートストリームがデコードされ、ステップS 2 6で、表示時刻がP T S (b 0) になったと判定された場合 (CM開始点であると判定された場合)、

A Vデコーダ27により表示が停止される。そして、PTS (C) < PTS (c0) の場合、ステップS30でパケット番号Cのデータから始まるストリームからデコードが再開され、ステップS31において、PTS (c0) のピクチャから表示が再開される。

なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される2点間のシーンをスキップして再生する場合にも、適用可能である。

次に、mark_entryとrepresentative_picture_entryが、図82に示すシンタクス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図112のフローチャートを参照して説明する。

ステップS41において、制御部23は、EP_map (図70)、STC_Info (図52)、Program_Info (図54)、及びClipMark (図78) の情報を取得する。

次にステップS42において、制御部23は、ステップS41で読み出したClipMark (図78) に含まれるrepresentative_picture_entry (図82) 又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面上に表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップS43において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップS42の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応付けられているマーク点を指定することで行われる。

ステップS44において、制御部23は、ステップS43の処理で指定されたマーク点のRSPN_ref_EP_startとoffset_num_pictures (図82) を取得する。

ステップS45において、制御部23は、ステップS44で取得したRSPN_ref_EP_startに対応するソースパケット番号からトランスポートストリームのデータを読み出し、A Vデコーダ27に供給させる。

ステップS46において、制御部23は、A Vデコーダ27を制御し、RSPN_ref_EP_startで参照されるピクチャから (表示はしないで)、表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_picturesになったとき、そのピクチャから表示を開始させる。

以上の処理を、図 1 1 3 乃至図 1 1 5 を参照して、更に説明する。この例においては、DVRトランスポートストリームファイルは、ソースパケット番号Aからシーンが開始しており、ソースパケット番号BからソースパケットCまでCMが挿入されている。このため、図 1 1 4 に示されるように、EP_mapには、RSPN_EP_startとしてのA, B, Cに対応して、PTS_EP_startとして、PTS(A), PTS(B), PTS(C)が登録されている。

また、図 1 1 5 に示されるように、シーンスタート、CMスタート、及びCMエンドのマークタイプに対応して、mark_entryとrepresentative_picture_entryが登録されている。mark_entryには、シーンスタート、CMスタート、及びCMエンドに対応して、RSPN_ref_EP_startとして、それぞれA, B, Cが登録され、offset_num_picturesとして、M1, N1, N2が登録されている。同様に、representative_picture_entryには、RSPN_ref_EP_startとして、シーンスタート、CMスタート、及びCMエンドに対応して、それぞれA, B, Cが登録され、offset_num_picturesとして、M2, N1, N2がそれぞれ登録されている。

シーンスタートにあたるピクチャから頭出して再生が指令された場合、パケット番号Aのデータから始まるストリームからデコードが開始され、PTS(A)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset_num_picturesが、M1の値になったとき、そのピクチャから表示が開始される。

更に、mark_entryとrepresentative_picture_entryのシンタクスが、図 8 2 に示される構成である場合におけるCMスキップ再生の処理について、図 1 1 6 のフローチャートを参照して説明する。

ステップS61において、制御部23は、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、及びClipMark（図78）の情報を取得する。

ステップS62において、ユーザがCMスキップ再生を指令すると、ステップS63において、制御部23は、マークタイプ（図79）がCM開始点とCM終了点である各点のマーク情報として、RSPN_ref_EP_STARTとoffset_num_pictures（図82）を取得する。そして、CM開始点のデータは、RSPN_ref_EP_start(1), offset_num_pictures(1)とされ、CM終了点のデータは、RSPN_ref_EP_start(2),

offset_num_pictures(2)とされる。

ステップS 6 4において、制御部2 3は、RSPN_ref_EP_start(1),RSPN_ref_EP_start(2)に対応するPTSをEP_map(図7 0)から取得する。

ステップS 6 5において、制御部2 3は、トランスポートストリームを記録媒体1 0 0から読み出させ、AVデコーダ2 7に供給させる。

ステップS 6 6において、制御部2 3は、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであるか否かを判定し、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャでない場合には、ステップS 6 7に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップS 6 5に戻り、それ以降の処理が繰り返し実行される。

ステップS 6 6において、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであると判定された場合、ステップS 6 8に進み、制御部2 3は、AVデコーダ2 7を制御し、RSPN_ref_EP_start(1)に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値がoffset_num_pictures(1)になったとき、表示を停止させる。

ステップS 6 9において、制御部2 3は、RSPN_ref_EP_start(2)のソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ2 7に供給させる。

ステップS 7 0において、制御部2 3は、AVデコーダ2 7を制御し、RSPN_ref_EP_start(2)に対応するPTSのピクチャから(表示をしないで)表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_pictures(2)になったとき、そのピクチャから表示を開始させる。

以上の動作を、図1 1 3乃至図1 1 5を参照して更に説明すると、先ず、EP_map(図1 1 4)をもとに、パケット番号B, Cに対応する時刻PTS(B), PTS(C)が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS(B)になったとき、PTS(B)のピクチャから表示ピクチャがカウントアップされ、その値がN 1(図1 1 5)になったとき、表示が停止される。

更に、パケット番号Cのデータから始まるストリームからデコードが再開され、PTS(C)のピクチャから(表示をしないで)表示すべきピクチャをカウント

アップしていき、その値がN2（図115）になったとき、そのピクチャから表示が再開される。

以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された2点間のシーンをスキップさせて再生する場合にも、適用可能である。

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図84に示すような構成である場合における、マーク点で示されるシーンの頭出し再生処理について、図118のフローチャートを参照して説明する。

ステップS81において、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、並びにClipMark（図78）の情報が取得される。

ステップS82において、制御部23は、ClipMark（図78）のrepresentative_picture_entry又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップS83において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応付けられているマーク点を指定することで行われる。

ステップS84において、制御部23は、ユーザから指定されたmark_entryのRSPN_mark_point（図84）を取得する。

ステップS85において、制御部23は、マーク点のRSPN_mark_pointより前にあり、且つ、最も近いエントリポイントのソースバケット番号を、EP_map（図70）から取得する。

ステップS86において、制御部23は、ステップS85で取得したエントリポイントに対応するソースバケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

ステップS87において、制御部23は、AVデコーダ27を制御し、RSPN_mark_pointで参照されるピクチャから表示を開始させる。

以上の処理を、図119乃至図121を参照して更に説明する。この例においては、DVRトランスポートストリームファイルが、ソースバケットAでシーン

スタートし、ソースパケット番号BからCまでCMが挿入されている。このため、図120のEP_mapには、RSPN_EP_startとしてのA、B、Cに対応して、PTS_EP_startがそれぞれPTS(A)、PTS(B)、PTS(C)として登録されている。また、図121に示されるClipMarkに、シーンスタート、CMスタート、及びCMエンドに対応して、markentryのRSPN_mark_pointとして、a1、b1、c1が、また、representative_picture_entryのRSPN_mark_pointとして、a2、b1、c1が、それぞれ登録されている。

シーンスタートにあたるピクチャから頭出して再生する場合、パケット番号A < a1 とすると、パケット番号Aのデータから始まるストリームからデコードが開始され、ソースパケット番号a1に対応するピクチャから表示が開始される。

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図84に示されるような構成である場合におけるCMスキップ再生の処理について、図122と図123のフローチャートを参照して説明する。

ステップS101において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、並びにClipMark(図70)の情報を取得する。

ステップS102において、ユーザは、CMスキップ再生を指定する。

ステップS103において、制御部23は、マークタイプ(図79)がCM開始点とCM終了点である各点のマーク情報のRSPN_mark_point(図84)を取得する。そして、制御部23は、CM開始点のデータをRSPN_mark_point(1)とし、CM終了点のデータをRSPN_mark_point(2)とする。

ステップS104において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、AVデコーダ27に出力し、デコードさせる。

ステップS105において、制御部23は、現在の表示画像がRSPN_mark_point(1)に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN_mark_point(1)に対応するピクチャでない場合には、ステップS106に進み、そのままピクチャを継続的に表示させる。その後、処理はステップS104に戻り、それ以降の処理が繰り返し実行される。

ステップS105において、現在の表示画像がRSPN_mark_point(1)に対応するピクチャであると判定された場合、ステップS107に進み、制御部23はA

Vデコーダ27を制御し、デコード及び表示を停止させる。

次に、ステップS108において、RSPN_mark_point(2)より前にあり、且つ、最も近いエントリポイントのあるソースパケット番号がEP_map(図70)から取得される。

ステップS109において、制御部23は、ステップS108で取得したエントリポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

ステップS110において、制御部23は、AVデコーダ27を制御し、RSPN_mark_point(2)で参照されるピクチャから表示を再開させる。

以上の処理を図119乃至図121の例で更に説明すると、Clip AV streamをデコードしていき、ソースパケット番号b1(図121)に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号C<ソースパケット番号c1とすると、パケット番号Cのデータから始まるストリームからデコードが再開され、ソースパケット番号c1に対応するピクチャになったとき、そのピクチャから表示が再開される。

以上のようにして、図124に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

より具体的には、図125に示されるように、PlayList上において、PlayList Markとしてブックマークやリジューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

なお、ClipMarkのシンタクスは、図78の例に替えて、図126に示すようにすることもできる。

この例においては、RSPN_markが、図78のreserved_for_MakerID, mark_entry()、及びrepresentative_picture_entry()に替えて挿入されている。このRSPN_markの32ビットのフィールドは、AVストリームファイル上で、そのマーク

が参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。RSPN_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset_SPNの値を初期値としてカウントされる。

その他の構成は、図7.8における場合と同様である。

ClipMarkのシンタクスは、更に図127に示すように構成することもできる。この例においては、図126におけるRSPN_markの代わりに、RSPN_ref_EP_startとoffset_num_picturesが挿入されている。これらは、図82に示した場合と同様のものである。

図128は、アナログAV信号をエンコードして記録する場合、図81に示したシンタクスのClipMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS200において、解析部14は端子11, 12からの入力AV信号を解析して、特徴点を検出する。特徴点は、AVストリームの内容に起因する特徴的なシーンを指定し、例えば、番組の頭だし点やシーンチェンジ点などである。

ステップS201において、制御部23は特徴点の画像のPTSを取得する。ステップS202において、制御部23は、特徴点の情報をClipMarkにストアする。具体的には、本例のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。ステップS203において、Clip Information fileとClip AV stream fileがディスクに記録される。

図129は、ディジタルインタフェースから入力されたトランスポートストリームを記録する場合、図81に示したシンタクスのClipMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS211において、デマルチプレクサ26、及び、制御部23は、記録するプログラムのエレメンタリストリームPIDを取得する。解析対象のエレメンタリストリームが複数ある場合、全てのエレメンタリストリームPIDが取得される。

ステップS212で、デマルチプレクサ26は、端子13から入力されるトランスポートストリームのプログラムからエレメンタリストリームを分離し、それ

をAVデコーダ27がAV信号にデコードする。ステップS213において、解析部14は、上記AV信号を解析して特徴点を検出する。

ステップS214において、制御部23は、特徴点の画像のPTSと、それが属するSTCのSTC-sequence-idを取得する。ステップS215で、制御部23は、特徴点の情報をClipMarkにストアする。具体的には、本例におけるClipMarkのシタクスとセマンティクスで説明した情報をストアする。

ステップS216において、Clip Information fileとClip AV stream fileがディスクに記録される。

図128に示したフローチャート、及び、図129に示したフローチャートのようにして、AVストリームファイル、すなわちClip AVストリームファイルの中の特徴的な画像を指し示すマークをストアするClipMarkが、前記AVストリームの管理情報データファイル、すなわちClip Informationファイルに記録される。

図130は、Real Playlistの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS221において、制御部23はClip AVストリームを記録する。ステップS222において、制御部23は、上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlaylist()を作成する。Clipの中にSTC不連続点があり、Playlist()が2つ以上のPlayItemからなる場合、PlayItem間のconnection_conditionもまた決定される。

ステップS223において、制御部23は、UIAppInfoPlaylist()を作成する。ステップS224において、制御部23は、PlaylistMarkを作成する。ステップS225において、制御部23は、MakersPrivateDataを作成する。ステップS226において、制御部23は、Real Playlistファイルを記録する。

このようにして、新規にClip AVストリームを記録する毎に、1つのReal Playlistファイルが作られる。

図131は、Virtual Playlistの作成について説明するフローチャートである。ステップS231において、ユーザインタフェースを通して、ディスクに記録されている1つのReal Playlistの再生が指定される。そして、そのReal Playlistの再生範囲の中から、ユーザインタフェースを通して、IN点とOUT点で示さ

れる再生区間が指定される。

ステップS 2 3 2において、制御部2 3は、ユーザによる再生範囲の指定操作が全て終了したか否かを判断する。ステップS 2 3 2において、ユーザによる再生範囲の指定操作はまだ終了していないと判断された場合、ステップS 2 3 1に戻り、それ以降の処理が繰り返され、終了したと判断された場合、ステップS 2 3 3に進む。

ステップS 2 3 3において、連続して再生される2つの再生区間の間の接続状態(connection_condition)が、ユーザがユーザインタフェースを通して決定されるか、又は制御部2 3により決定される。ステップS 2 3 4において、ユーザインタフェースを通して、ユーザがサブパス(アフレコ用オーディオ)情報を指定する。ユーザがサブパスを作成しない場合、ステップS 2 3 4における処理はスキップされる。

ステップS 2 3 5において、制御部2 3は、ユーザが指定した再生範囲情報、及びconnection_conditionに基づいて、PlayList()を作成する。ステップS 2 3 6において、制御部2 3はUIAppInfoPlayList()を作成する。ステップS 2 3 7において、制御部2 3は、PlayListMarkを作成する。ステップS 2 3 8において、制御部2 3は、MakersPrivateDataを作成する。ステップS 2 3 9において、制御部2 3は、Virtual PlayListファイルを、ディスクに記録させる。

このようにして、ディスクに記録されているReal PlayListの再生範囲の中から、ユーザが、見たい再生区間を選択し、その再生区間をグループ化したもの毎に、1つのVirtual PlayListファイルが作成される。

図1 3 2は、PlayListの再生について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS 2 4 1において、制御部2 3は、Info.dvr, Clip Information file, PlayList file及びサムネイルファイルの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 4 2において、ユーザインタフェースを通して、ユーザが1つのPlayListの再生を制御部2 3に指示する。ステップS 2 4 3において、制御部2 3は、現在のPlayItemのSTC-sequence-idとIN_timeのPTSから、IN_timeより時

間的に前で最も近いエントリポイントのあるソースパケット番号を取得する。ステップS 2 4 4において、制御部2 3は、上記エントリポイントのあるソースパケット番号からAVストリームのデータを読み出し、AVデコーダ2 7へ供給する。

上記PlayItemの時間的に前にPlayItemの再生があった場合、ステップS 2 4 5において、制御部2 3は、そのPlayItemとの表示の接続処理をconnection_conditionに従って行なわれるように制御を行う。ステップS 2 4 6において、AVデコーダ2 7は、IN_timeのPTSのピクチャから表示を開始する。

ステップS 2 4 7において、AVデコーダ2 7は、AVストリームのデコードを継続的に行う。ステップS 2 4 8において、制御部2 3は、現在表示の画像が、OUT_timeのPTSの画像か否かを判断する。ステップS 2 4 8において、現在表示の画像は、OUT_timeのPTSの画像であると判断された場合、ステップS 2 5 0に進み、PTSの画像ではないと判断された場合、ステップS 2 4 9に進む。

ステップS 2 4 9において、PTSの画像であると判断された画像を表示するための処理が実行され、その後ステップS 2 4 7に戻り、それ以降の処理が繰り返される。一方、ステップS 2 5 0においては、制御部2 3により、現在のPlayItemがPlayListの中で最後のPlayItemか否かが判断される。ステップS 2 5 0において、現在のPlayItemがPlayListの中で最後のPlayItemであると判断された場合、図1 3 2に示したフローチャートの処理は終了され、最後のPlayItemではないと判断された場合、ステップS 2 4 3に戻り、それ以降の処理が繰り返される。

図1 3 3は、PlayListMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS 2 6 1において、制御部2 3は、Info.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 6 2において、ユーザインタフェースを通して、ユーザにより1つのPlayListの再生が制御部2 3に指示される。ステップS 2 6 3において、再生部3は、指示されたPlayListの再生を開始する（図1 3 2のフローチャートを参照して説明したように行われる）。

ステップS 2 6 4において、ユーザインタフェースを通して、ユーザにより、お気に入りのシーンのところにマークのセットが制御部2 3に指示される。ステップS 2 6 5において、制御部2 3は、マークのPTSと、それが属するPlayItemのPlayItem_idを取得する。

ステップS 2 6 6において、制御部2 3は、マークの情報をPlayListMark()にストアする。ステップS 2 6 7において、PlayListファイルがディスクに記録される。

このようにして、PlayListの再生範囲の中からユーザが指定したマーク点、又は、そのPlayListを再生するときのResume点を示すマークをストアするPlayList Markを、PlayListファイルに記録される。

図1 3 4は、PlayListが再生される時、PlayListMark及びそのPlayListが参照するClipのClipMarkが使用された頭だし再生について説明するフローチャートである。ClipMark()のシンタクスは、図8 1に示すものとする。図1の記録再生装置1のブロック図を参照しながら説明する。

ステップS 2 7 1において、制御部2 3は、Info.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 7 2において、ユーザインタフェースを通して、ユーザにより1つのPlayListの再生が指示される。ステップS 2 7 3において、制御部2 3は、PlayListMark、及び、そのPlayListが参照するClipのClipMarkで参照されるピクチャから生成したサムネイルのリストを、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 7 4において、ユーザインタフェースを通して、制御部2 3に、ユーザにより再生開始点のマーク点が指定される。ステップS 2 7 5において、制御部2 3は、ステップS 2 7 4における処理で選択されたマークがPlayListMarkにストアされているマークか否かを判断する。ステップS 2 7 5において、選択されたマークがPlayListMarkにストアされているマークであると判断された場合、ステップS 2 7 6に進み、ストアされていないマークであると判断された場

合、ステップS 2 7 8に進む。

ステップS 2 7 6において、制御部2 3は、マークのPTSと、それが属するPlayItem_idを取得する。ステップS 2 7 7において、制御部2 3はPlayItem_idが指すPlayItemが参照するAVストリームのSTC-sequence-idを取得する。

ステップS 2 7 8において、制御部2 3は、STC-sequence-idとマークのPTSに基づいて、AVストリームをAVデコーダ2 7へ入力させる。具体的には、このSTC-sequence-idとマーク点のPTSを用いて、図1 3 2のフローチャートのステップS 2 4 3、S 2 4 4と同様の処理が行なわれる。ステップS 2 7 9において、再生部3は、マーク点のPTSのピクチャから表示を開始する。

図9を参照して説明したように、PlayListが再生される時、そのPlayListが参照するClipのClipMarkにストアされているマークを参照することができる。したがって、1つのClipを、Real PlayListや複数のVirtual PlayListによって参照している場合、それらのPlayListは、その1つのClipのClipMarkを共有することができるので、マークのデータを効率良く管理することができる。

仮に、ClipにClipMarkを定義しないで、PlayListだけにPlayListMarkとClipMarkを合わせたものを定義するようにした場合、上記の例のように1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それぞれのPlayListが同じ内容のClipのマーク情報を持つことになり、データの記録の効率が悪い。

図1 3 5は、PlayListMark()のシンタックスの別例を示す図である。lengthは、このlengthフィールドの直後のバイトからPlayListMark()の最後のバイトまでのバイト数を示す。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークのエントリ数を示す。

mark_invalid_flagは、1ビットのフラグであり、この値が0にセットされている時、このマークは有効な情報を持っていることを示し、また、この値が1にセットされている時、このマークは無効であることを示す。

ユーザがユーザインタフェース上で1つのマークのエントリを消去するオペレーションをした時、記録再生装置1は、PlayListMarkからそのマークのエントリを消去する代わりに、そのmark_invalid_flagの値を1に変更するようにしてもよ

い。

mark_typeは、マークのタイプを示し、図 1 3 6 に示す意味を持つ。mark_name_lengthは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。このフィールドの値は 3 2 以下である。ref_to_PlayItem_idは、マークが置かれているところのPlayItemを指定するところのPlayItem_idの値を示す。あるPlayItemに対応するPlayItem_idの値は、PlayList()において定義される。

mark_time_stampは、そのマークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampは、ref_to_PlayItem_idで示されるPlayItemの中で定義されているところのIN_timeとOUT_timeで特定される再生範囲の中の時間を指す。タイムスタンプの意味は、図 4 4 と同じである。

entry_ES_PIDが、0xFFFFにセットされている場合、そのマークはPlayListによって使用される全てのエレメンタリーストリームに共通の時間軸上へのポイントである。entry_ES_PIDが、0xFFFFでない値にセットされている場合、entry_ES_PIDは、そのマークによって指されるところのエレメンタリーストリームを含んでいるところのトランスポートパケットのPIDの値を示す。

ref_thumbnail_indexは、マークに付加されるサムネイル画像の情報を示す。その意味は、図 4 2 のref_thumbnail_indexと同じである。mark_nameは、マークの名前を示す。このフィールドの中の左からmark_name_lengthで示されるバイト数が、有効なキャラクター文字であり、名前を示す。このキャラクター文字は、UI AppInfoPlayListの中でcharacter_setによって示される方法で符号化されている。

mark_nameフィールドの中で、それら有効なキャラクター文字に続くバイトの値は、どんな値が入っていてもよい。このシンタックスの場合、マークが特定のエレメンタリーストリームを指すことができる。例えば、PlayListが、プログラムの中に複数のビデオストリームを持つマルチビュープログラムを参照している時、entry_ES_PIDは、そのプログラムの中の 1 つのビデオストリームを示すビデオPIDをセットするために使われる。

ユーザがマルチビュープログラムを参照するところのPlayListを再生しており、そのユーザは、マルチビュー中の 1 つのビューを見ているとする。今、ユーザが記録再生装置 1 に対して、次のマーク点に再生をスキップするようにコマンドを

送ったとする。この場合、記録再生装置 1 は、ユーザが現在見ているビューのビデオ P I D と同じ値であるところの entry_ES_PID のマークを使用すべきであり、記録再生装置 1 は、勝手にビューを変更すべきでない。記録再生装置 1 は、また、entry_ES_PID が 0xFFFF にセットされているマークを使用してもよい。この場合も記録再生装置 1 は、勝手にビューを変更しない。

図 1 3 7 は、図 8 1 に示すシンタクスの ClipMark() の別例を示す図である。length は、この length フィールドの直後のバイトから ClipMark() の最後のバイトまでのバイト数を示す。maker_ID は、mark_type が 0x60 から 0x7F の値を示す時に、その mark_type を定義しているメーカのメーカ I D を示す。

number_of_Clip_marks は、ClipMark の中にストアされているマークのエントリ数を示す。mark_invalid_flag は、1 ビットのフラグであり、これの値が 0 にセットされている時、このマークは有効な情報を持っていることを示し、また、これの値が 1 にセットされている時、このマークは無効であることを示す。

ユーザが、ユーザインタフェース上で 1 つのマークのエントリを消去するオペレーションをした時、記録機は ClipMark からそのマークのエントリを消去する代わりに、その mark_invalid_flag の値が 1 に変更されるようにしてもよい。mark_type は、マークのタイプを示し、図 1 3 8 に示す意味を持つ。

ref_to_STC_id は、mark_time_stamp と representative_picture_time_stamp の両方が置かれているところの STC-sequence を指定するところの STC-sequence-id を示す。STC-sequence-id の値は、STCInfo() の中で定義される。mark_time_stamp は、図 8 1 の mark_entry() の場合での mark_time_stamp と同じ意味である。

entry_ES_PID が、0xFFFF にセットされている場合、そのマークは Clip 中の全てのエレメンタリーストリームに共通の時間軸上へのポイントである。entry_ES_PID が、0xFFFF でない値にセットされている場合、entry_ES_PID は、そのマークによって指されるところのエレメンタリーストリームを含んでいるところのトランスポートバケットの P I D の値を示す。

ref_to_thumbnail_index は、マークに付加されるサムネイル画像の情報を示す。その意味は、図 7 8 の ref_thumbnail_index と同じである。representative_picture_time_stamp は、図 8 1 の representative_picture_entry() の場合での mark_t

ime_stampと同じ意味である。

図137に示したシンタクスの場合、マークが、特定のエレメンタリーストリームを指すことができる。例えば、Clipが、プログラムの中に複数のビデオストリームを持つマルチビュープログラムを含んでいるとき、entry_ES_PIDは、そのプログラムの中の1つのビデオストリームを示すビデオPIDをセットするために使われる。

ユーザが、マルチビュープログラムを参照するところのPlayListを再生しており、そのユーザは、マルチビュー中の1つのビューを見ているとする。今、ユーザが記録再生装置1に対して、次のマーク点に再生をスキップするようにコマンドを送ったとする。この場合、記録再生装置1は、ユーザが現在見ているビューのビデオPIDと同じ値であるところのentry_ES_PIDのマークを使用すべきであり、記録再生装置1は、勝手にビューを変更すべきでない。記録再生装置1は、また、entry_ES_PIDが0xFFFFにセットされているマークを使用してもよい。この場合も記録再生装置1は、勝手にビューを変更しない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体100に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが、再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡単に再生できるようにすることができる。

以上のようなデータベース構成によれば、PlayListファイルやClip Informationファイルを別々に分離して記録するので、編集などによって、所定のPlayListやClipの内容が変更されたとき、そのファイルに関係のない他のファイルを変更する必要がない。したがって、ファイルの内容の変更が容易に行え、またその変更及び記録にかかる時間を小さくできる。

また、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザインタフェースへ提示し、ユーザが再生指示したPlayListファイルと、それに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

仮に、全てのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、

そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明を適用することにより、このようなことを防ぐことが可能となる。

上述したように、A Vストリームファイル、すなわちClip A Vストリームファイルの中の特徴的な画像を指し示すマークをストアするClipMarkを、前記A Vストリームの管理情報データファイル、すなわちClip Informationファイルに記録し、また、A Vストリーム中の指定された区間の組み合わせにより定義される1つの再生手順の情報を持つオブジェクト、すなわちPlayListの再生範囲の中から、ユーザが指定したマーク点、又は、そのオブジェクトを再生するときのResume点を示すマークをストアするPlayListMarkを、オブジェクトに記録する。

このようにすることにより、PlayListが再生される時、そのPlayListが参照するClipのClipMarkにストアされているマークを参照することができる。したがって、1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それらのPlayListは、その1つのClipのClipMarkを共有することができるので、マークのデータを効率良く管理することができる。

仮に、ClipにClipMarkを定義しないで、PlayListだけにPlayListMarkとClipMarkを合わせたものを定義するようにした場合、上記の例のように1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それぞれのPlayListが同じ内容のClipのマーク情報を持つことになり、データの記録の効率が悪い。本発明を適用することにより、このようなことを防ぐことが可能となる。

以上のように、A Vストリームの付属情報として、エントリポイントのアドレスをストアするためのEP_mapと、マーク点のピクチャのタイプ（例えば番組の頭出し点）とそのピクチャのA Vストリームの中のアドレスをストアするためのClipMarkを、Clip Information Fileとしてファイル化して記録媒体100に記録することにより、A Vストリームの再生に必要なストリームの再生に必要なストリームの符号化情報を適切に管理することが可能である。

このClip Information file情報により、ユーザが、記録媒体100に記録されているA Vストリームの中から興味のあるシーン、例えば番組の頭出し点など、をサーチすることができ、ユーザのランダムアクセスや特殊再生の指示に対して、

記録媒体100からのAVストリームの読み出し位置の決定が容易になり、またストリームの復号開始を速やかに行うことができる。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、記録再生装置1は、図139に示されるようなパーソナルコンピュータにより構成される。

図139において、CPU (Central Processing Unit) 201は、ROM (Read Only Memory) 202に記憶されているプログラム、又は記憶部208からRAM (Random Access Memory) 203にロードされたプログラムに従って各種の処理を実行する。RAM 203には、CPC 201が各種の処理を実行する上において必要なデータなども適宜記憶される。

CPC 201、ROM 202、及びRAM 203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インタフェース205も接続されている。

入出力インタフェース205には、キーボード、マウスなどよりなる入力部206、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部207、ハードディスクなどより構成される記憶部208、モデム、ターミナルアダプタなどより構成される通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

入出力インタフェース205にはまた、必要に応じてドライブ210が接続され、磁気ディスク221、光ディスク222、光磁気ディスク223、或いは半導体メモリ224などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部208にインストールされる。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、又は、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

この記録媒体は、図139に示すように、コンピュータとは別に、ユーザにプ

プログラムを提供するために配布される、プログラムが記録されている磁気ディスク 2 2 1 (フロッピディスクを含む)、光ディスク 2 2 2 (CD-ROM (Compact Disk-Read Only Memory), DVD (Digital Versatile Disk) を含む)、光磁気ディスク 2 2 3 (MD (Mini-Disk) を含む)、若しくは半導体メモリ 2 2 4 などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されている ROM 2 0 2 や記憶部 2 0 8 が含まれるハードディスクなどで構成される。

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

以上の如く本発明に係る情報処理装置及び方法、並びにプログラムにおいては、入力された AV ストリームから抽出された特徴的な画像を指し示すマークで構成される ClipMark を、AV ストリームを管理するための管理情報として生成すると共に、AV ストリーム中の所定の区間の組み合わせを定義する Playlist に対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成される PlaylistMark を生成し、ClipMark、及び PlaylistMark を各々独立したテーブルとして記録媒体に記録するようにしたので、AV ストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

また、本発明に係る情報処理装置及び方法、並びにプログラムは、AV ストリームから抽出された特徴的な画像を指し示すマークで構成される ClipMark を含む AV ストリームを管理するための管理情報と、AV ストリーム中の所定の区間の組み合わせを定義する Playlist に対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成される PlaylistMark を読み出し、その読み

出された管理情報とPlayLisMarkによる情報を提示し、提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照し、参照されたClipMarkを含み、ClipMarkに対応する位置からA Vストリームを再生するようにしたので、A Vストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

請求の範囲

1. 入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記A Vストリームを管理するための管理情報として生成すると共に、

前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成手段と、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有する情報処理装置。

2. 前記生成手段は、前記ClipMarkをClipMarkInformationファイルとして生成すると共に、前記PlayListをPlayListファイルとして生成する請求の範囲第1項に記載の情報処理装置。

3. 前記PlayListMarkは、前記PlayListを再生するときのResume点を示すマークを更に含む特徴とする請求の範囲第1項に記載の情報処理装置。

4. 前記PlayListを再生するとき、前記PlayListの再生区間に対応する前記A VストリームのClipMarkを構成する前記マークを参照する請求の範囲第1項に記載の情報処理装置。

5. 前記PlayListMarkの前記マークは、プレゼンテーションタイムスタンプと、前記PlayListの再生経路を構成する前記A Vストリームデータ上の指定された1つの再生区間を示す識別情報を含む請求の範囲第1項に記載の情報処理装置。

6. 前記ClipMarkを構成する前記マーク、又は、前記PlayListMarkを構成する前記マークは、エレメンタリーストリームのエントリポイントを特定する情報を含む請求の範囲第1項に記載の情報処理装置。

7. 前記PlayListMarkの前記マークは、ユーザが指定したお気に入りのシーンの開始点又はPlayListのResume点を少なくとも含むタイプの情報を含む請求の範囲第1項に記載の情報処理装置。

8. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバ

ケットのアドレスで表される請求の範囲第1項に記載の情報処理装置。

9. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される請求の範囲第8項に記載の情報処理装置。

10. 前記第1の記録手段による記録の際に検出された前記特徴的な画像のタイプを検出するタイプ検出手段を更に含み、

前記第1の記録手段は、前記ClipMarkを構成する前記マークと、前記タイプ検出手段により検出された前記タイプとを対応させて記録する請求の範囲第1項に記載の情報処理装置。

11. 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含む請求の範囲第1項に記載の情報処理装置。

12. 入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを有する情報処理方法。

13. 入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

14. 入力されたAVストリームから抽出された特徴的な画像を指し示すマ

ークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させるプログラム。

15. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを読み出す読出手段と、

前記読出手段により読み出された前記管理情報と前記PlayListMarkによる情報を提示する提示手段と、

前記提示手段により提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照手段と、

前記参照手段により参照された前記ClipMarkを含み、前記ClipMarkに対応する位置から前記AVストリームを再生する再生手段とを含む情報処理装置。

16. 前記提示手段は、前記PlayListMarkに対応するサムネイル画像によるリストをユーザに提示する請求の範囲第15項に記載の情報処理装置。

17. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースバケットのアドレスで表される請求の範囲第15項に記載の情報処理装置。

18. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースバケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される請求の範囲第17項に記載の情報処理装置。

19. 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含む請求の範

図第 15 項に記載の情報処理装置。

20. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとを含む情報処理方法。

21. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

22. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlay

ListMarkの読み出しを制御する読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記A Vストリームの再生を制御する再生制御ステップとをコンピュータに実行させるプログラム。

23. A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記A Vストリームを管理するための管理情報と、前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されている記録媒体。

補正書の請求の範囲

[2001年8月3日(03.08.01)国際事務局受理:出願当初の請求の範囲3, 15, 16及び20-23は補正された;他の請求の範囲は変更なし。(4頁)]

1. 入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、

前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成手段と、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有する情報処理装置。

2. 前記生成手段は、前記ClipMarkをClipMarkInformationファイルとして生成すると共に、前記PlayListをPlayListファイルとして生成する請求の範囲第1項に記載の情報処理装置。

3. (補正後)前記PlayListMarkは、前記PlayListを再生するときのResume点を示すマークを更に含む請求の範囲第1項に記載の情報処理装置。

4. 前記PlayListを再生するとき、前記PlayListの再生区間に対応する前記AVストリームのClipMarkを構成する前記マークを参照する請求の範囲第1項に記載の情報処理装置。

5. 前記PlayListMarkの前記マークは、プレゼンテーションタイムスタンプと、前記PlayListの再生経路を構成する前記AVストリームデータ上の指定された1つの再生区間を示す識別情報を含む請求の範囲第1項に記載の情報処理装置。

6. 前記ClipMarkを構成する前記マーク、又は、前記PlayListMarkを構成する前記マークは、エレメンタリーストリームのエントリポイントを特定する情報を含む請求の範囲第1項に記載の情報処理装置。

7. 前記PlayListMarkの前記マークは、ユーザが指定したお気に入りのシーンの開始点又はPlayListのResume点を少なくとも含むタイプの情報を含む請求の範囲第1項に記載の情報処理装置。

8. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースバ

ークで構成されるClipMarkを、前記A Vストリームを管理するための管理情報として生成すると共に、前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させるプログラム。

15. (補正後) A Vストリーム、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生する情報処理装置であって、

前記記録媒体を再生する再生手段と、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて前記再生手段を制御する制御手段とを含む情報処理装置。

16. (補正後) 更に、前記PlayListMarkに対応するサムネイル画像によるリストをユーザに提示するよう制御する提示制御手段を有する請求の範囲第15項に記載の情報処理装置。

17. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバケットのアドレスで表される請求の範囲第15項に記載の情報処理装置。

18. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される請求の範囲第17項に記載の情報処理装置。

19. 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含む請求の範囲第15項に記載の情報処理装置。

20. (補正後) AVストリーム、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生する情報処理方法であって、

前記記録媒体を再生する再生ステップと、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて再生位置を制御する制御ステップとを含む情報処理方法。

21. (補正後) AVストリーム、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生するためのコンピュータが読み取り可能なプログラムが記録されている記録媒体であって、

前記記録媒体を再生する再生ステップと、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて再生位置を制御する制御ステップと

を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

22. (補正後) AVストリーム、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生するためのコンピュータが読み取り可能なプログラムであって、

前記記録媒体を再生する再生ステップと、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて再生位置を制御する制御ステップと

をコンピュータに実行させるプログラム。

23. (補正後) AVストリームが記録されると共に、当該AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されている記録媒体。

1/118

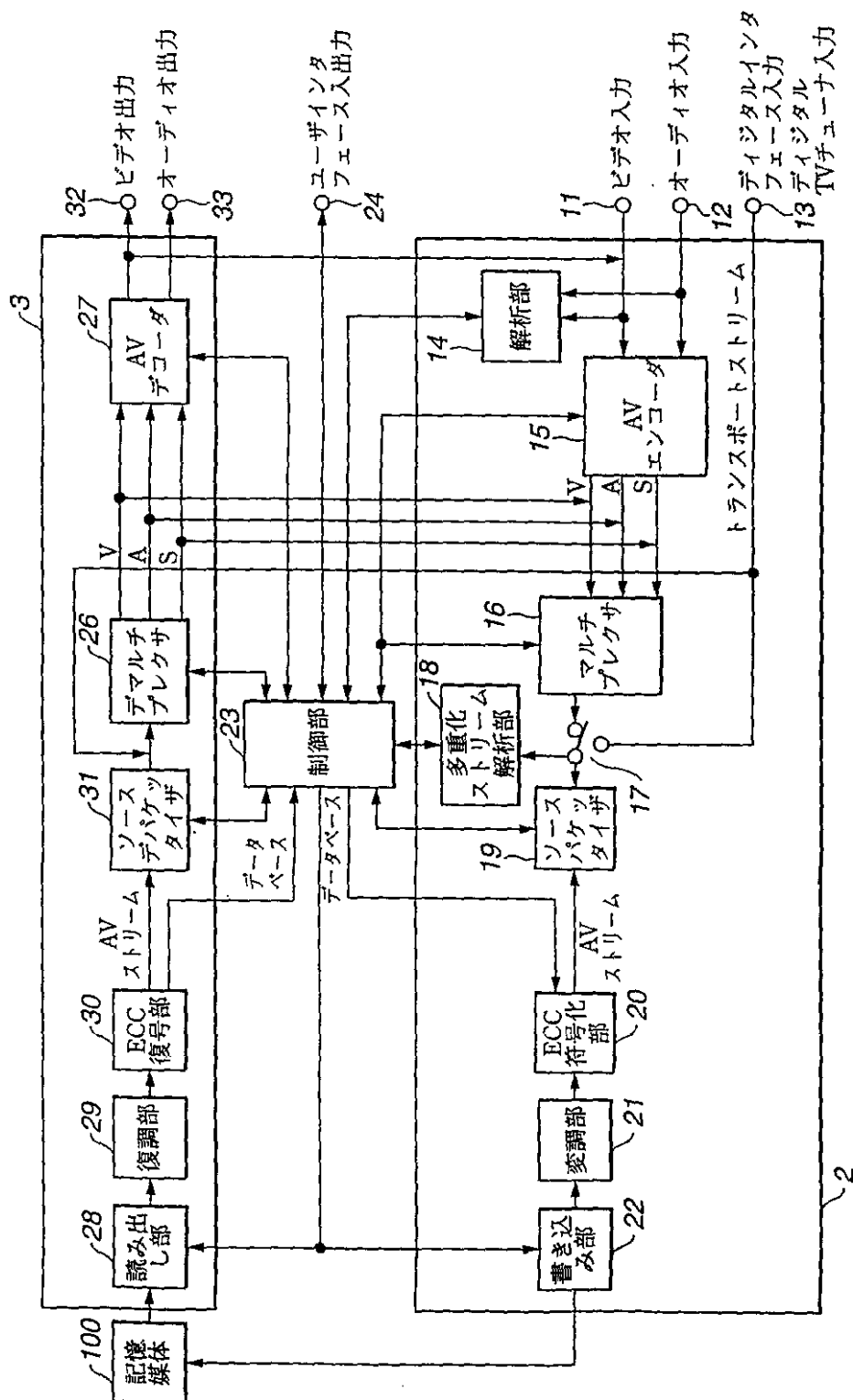


FIG.1

2/118

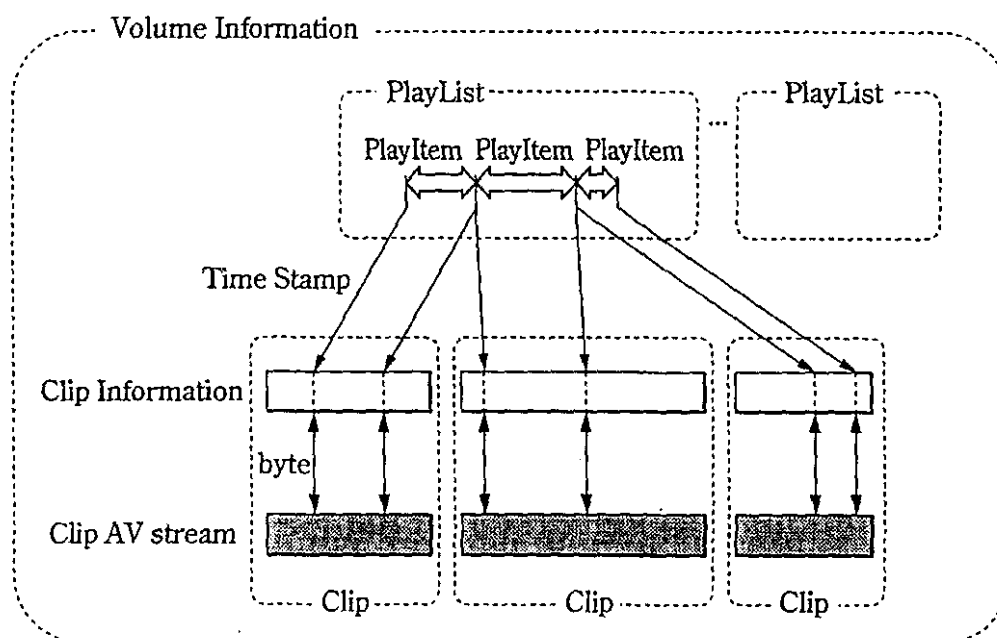


FIG.2

3/118

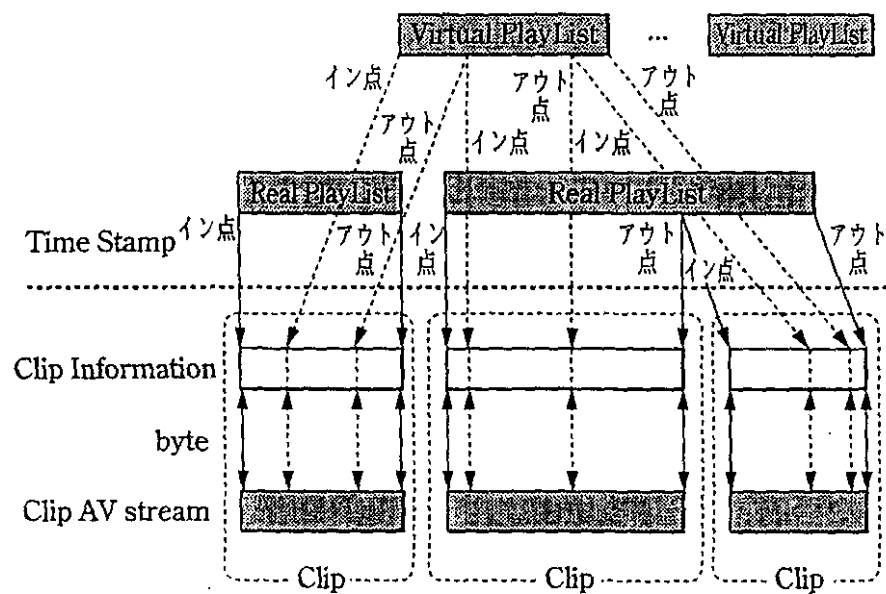


FIG.3

4/118

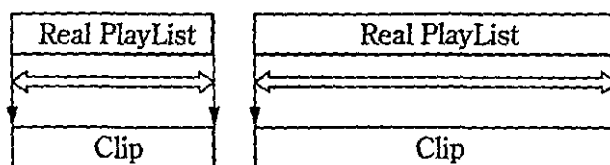


FIG.4A

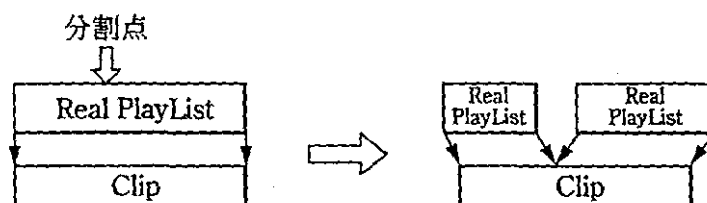


FIG.4B

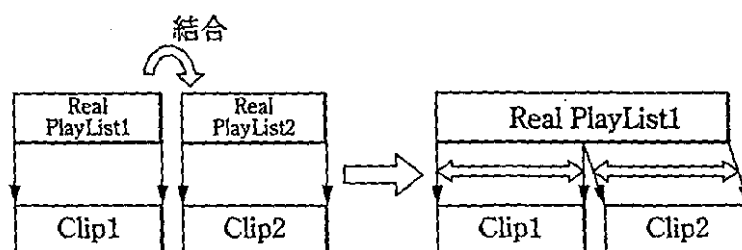


FIG.4C

5/118

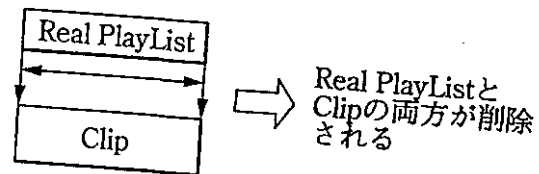


FIG.5A

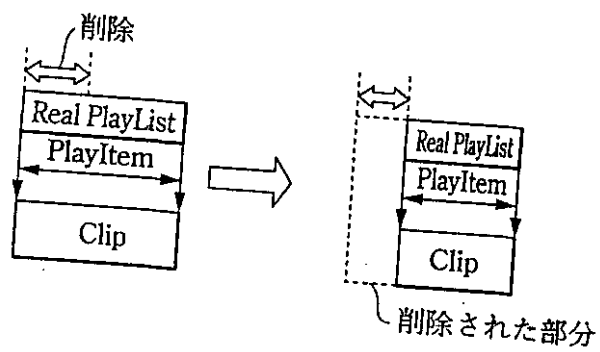


FIG.5B

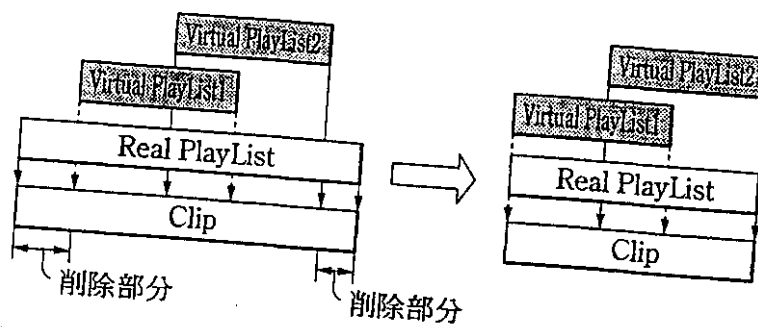
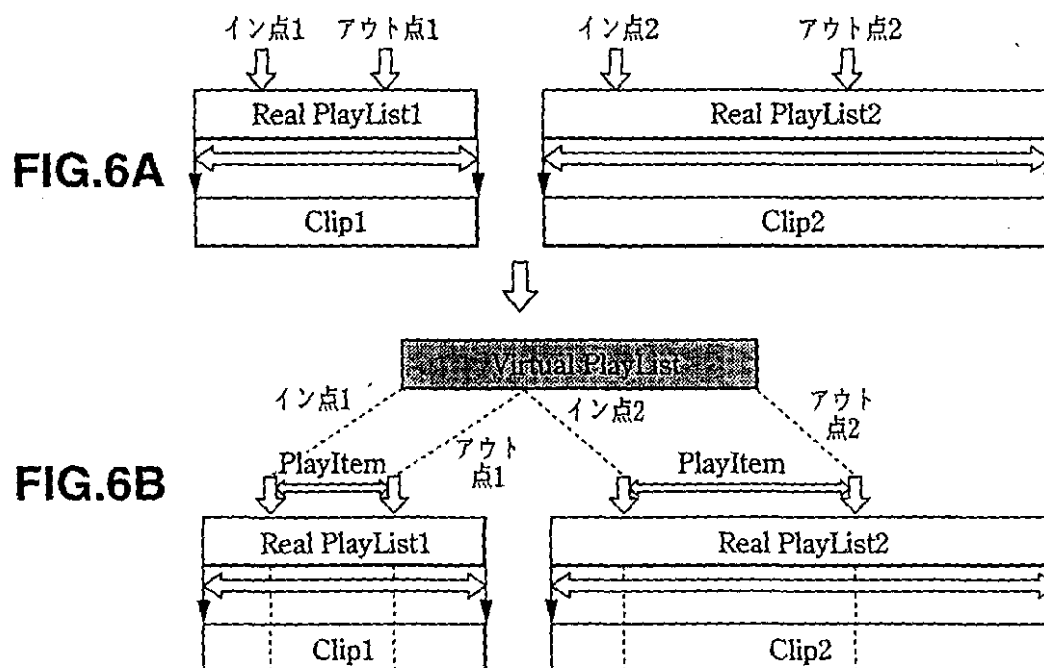


FIG.5C

6/118



7/118

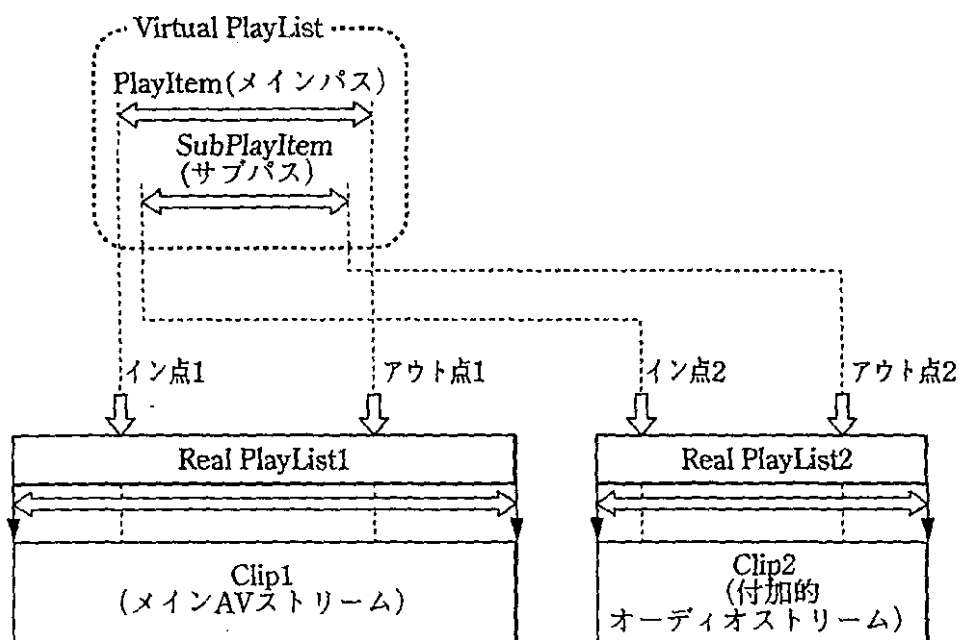


FIG.7

8/118

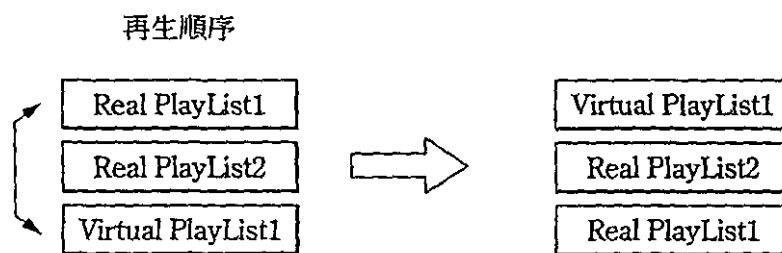


FIG.8

9/118

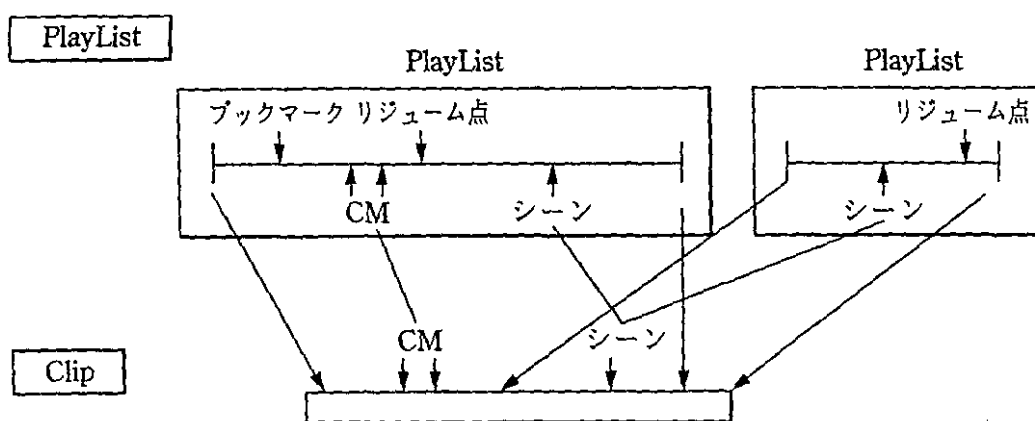


FIG.9

10/118

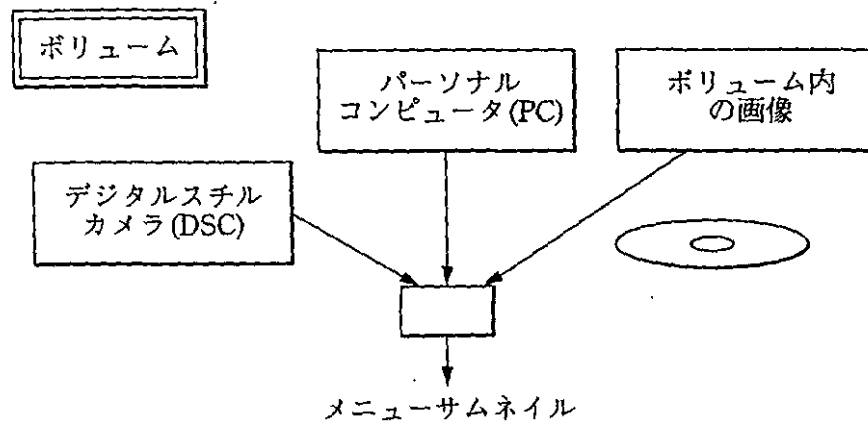


FIG.10

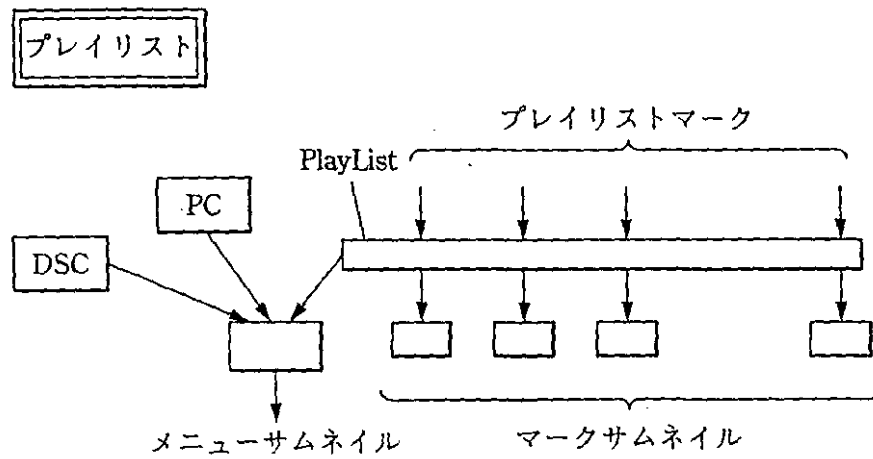


FIG.11

11/118

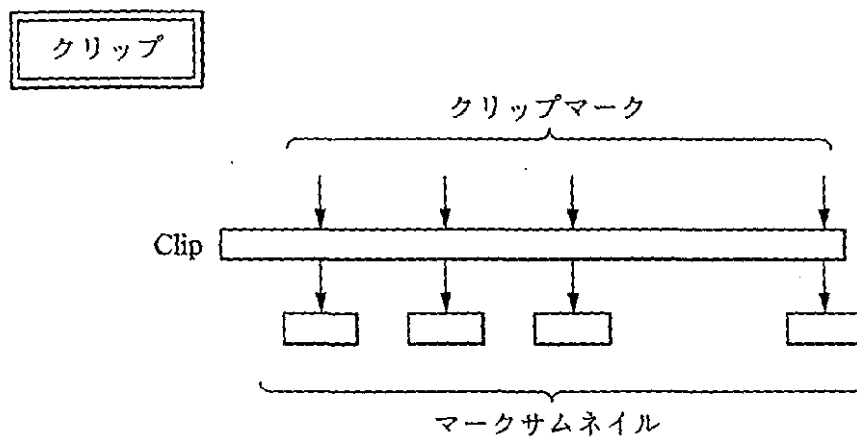


FIG.12

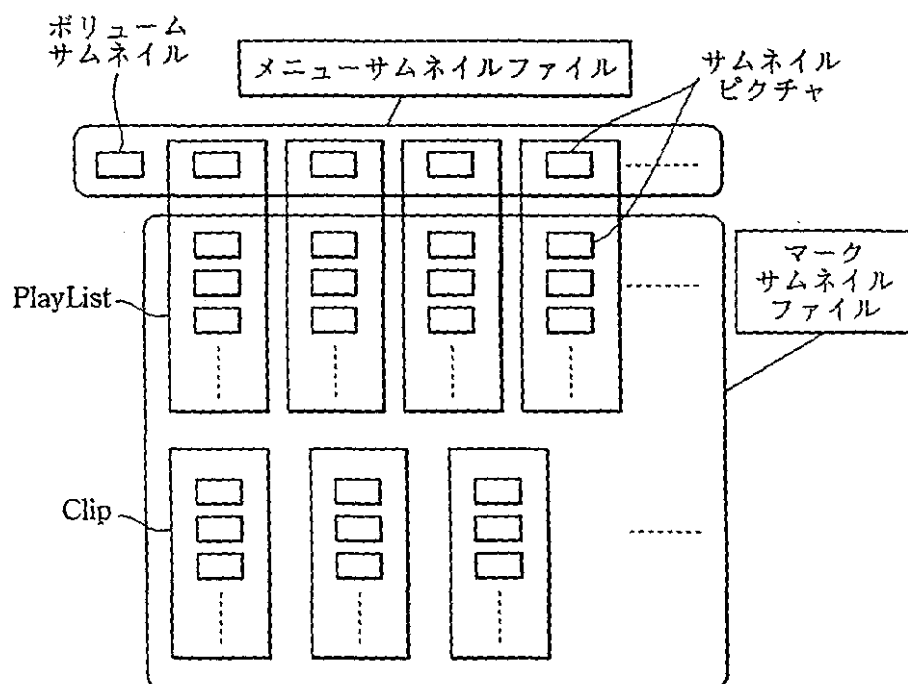


FIG.13

12/118

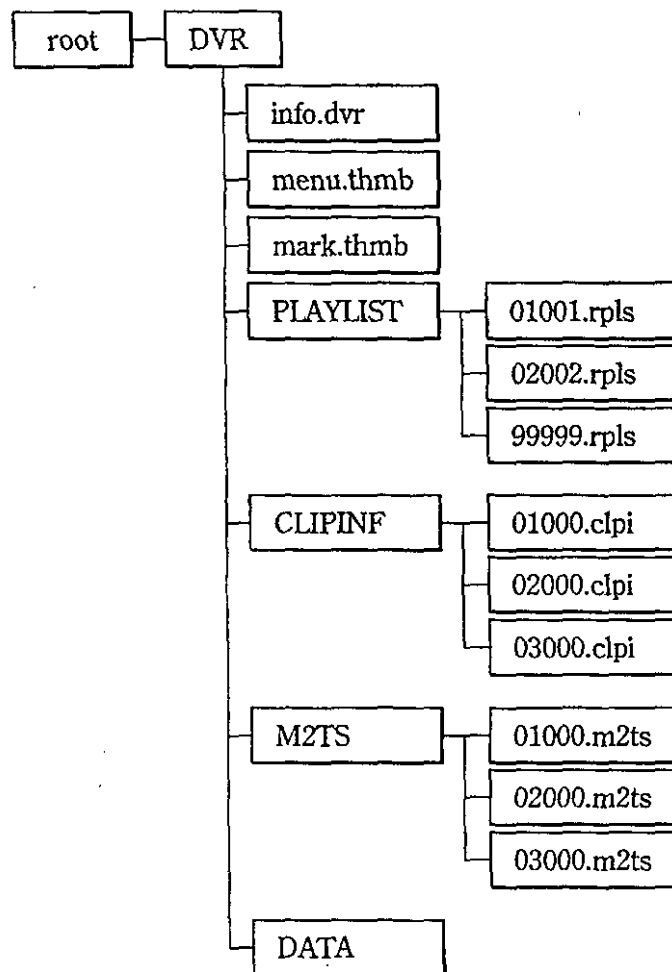


FIG.14

13/118

シンタクス	バイト数	略号
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.15

14/118

シンタクス	バイト数	略号
DVRVolume(){		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

FIG.16

15/118

シンタクス	バイト数	略号
ResumeVolume(){		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

16/118

シンタクス	バイト数	略号
UIAppInfoVolume(){		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.18

17/118

値	キャラクタ文字符号化
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

FIG.19

18/118

シンタクス	バイト数	略号
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
Playlist_file_name	8*10	bslbf
}		
}		

FIG.20

シンタクス	バイト数	略号
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
UAppInfoPlayList()		
}		
}		

FIG.21

20/118

シンタクス	バイト数	略号
MakersPrivateData(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

FIG.22

21/118

シンタクス	バイト数	略号
xxxxxx.rpls / yyyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.23

22/118

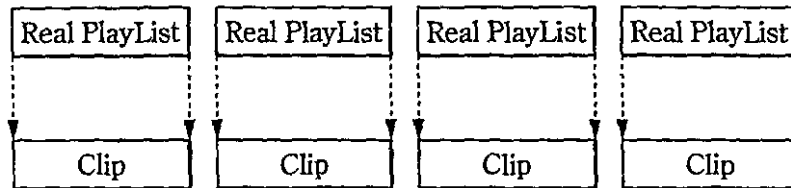


FIG.24A

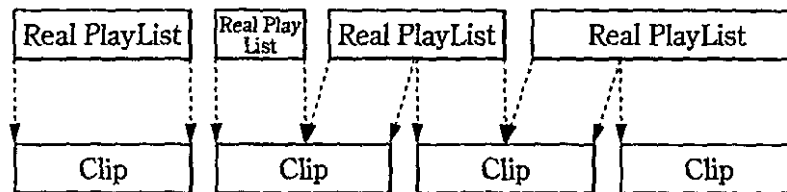


FIG.24B

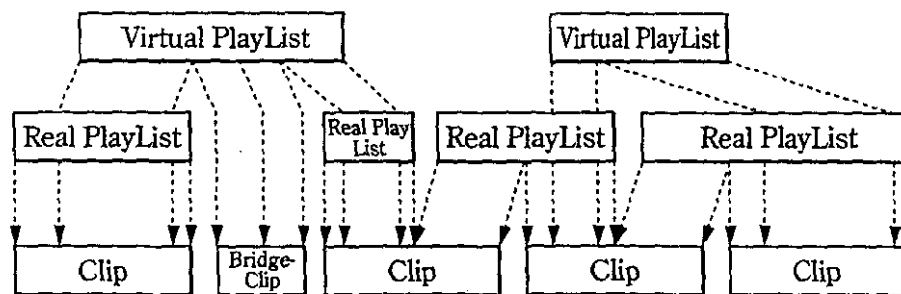


FIG.24C

23/118

シンタクス	バイト数	略号
PlayList(){		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() //sub path		
}		
}		
}		

FIG.25

24/118

PlayList_type	意味
0	AV記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2-255	reserved

FIG.26

25/118

シンタクス	バイト数	略号
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.27

26/118

write_protect_flag	意味
0b	そのPlayListを自由に消去しても良い。
1b	write_protect_flagを除いてそのPlayListの内容は、消去および変更されるべきではない。

FIG.28A

is_played_flag	意味
0b	そのPlayListは、記録されてから一度も再生されたことがない。
1b	PlayListは、記録されてから一度は再生された。

FIG.28B

archive	意味
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

FIG.28C

27/118

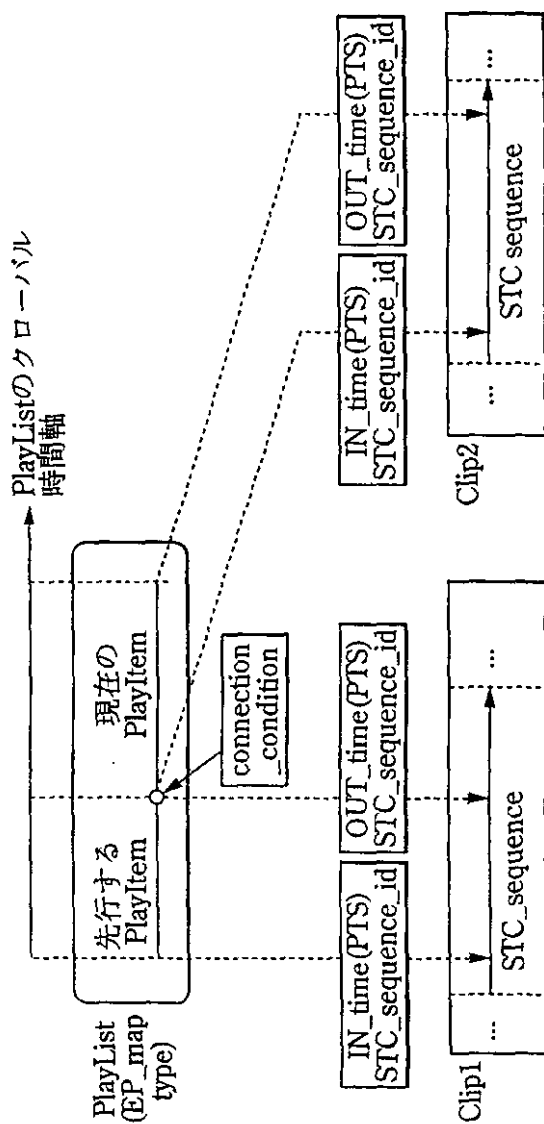


FIG.29

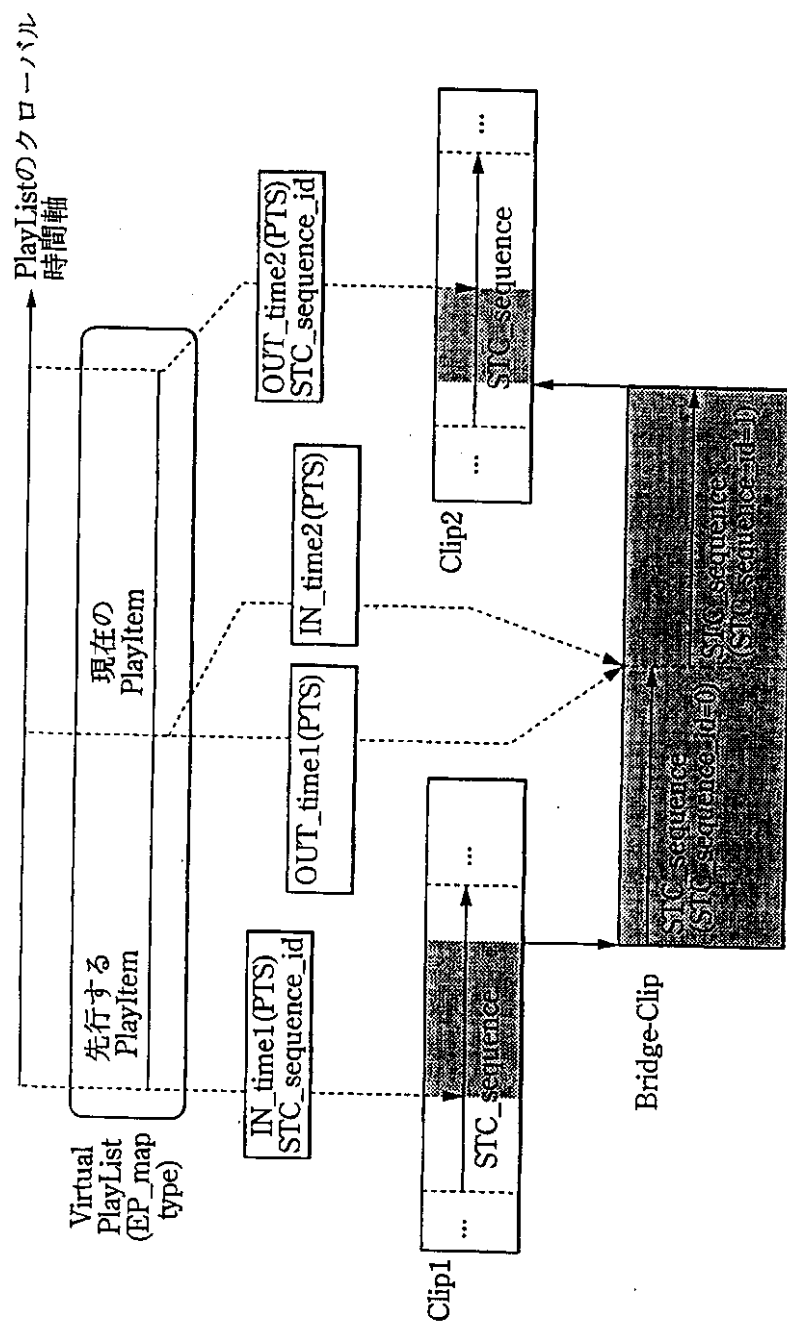


FIG.30

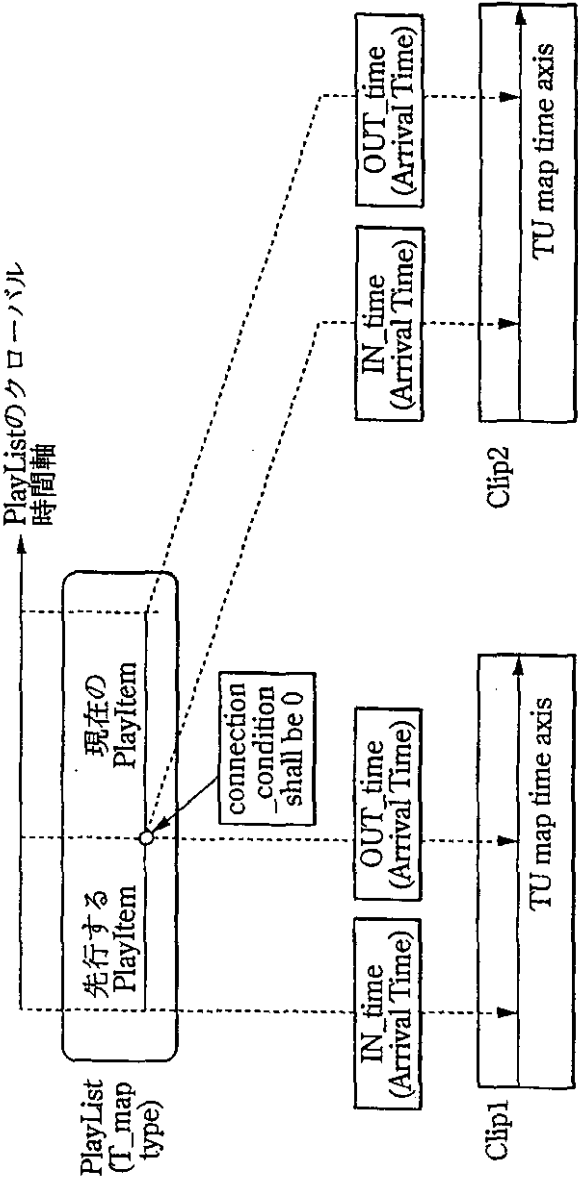


FIG.31

30/118

シンタクス	バイト数	略号
PlayItem(){		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual Playlist>){		
if (connection_condition=='10'){		
BridgeSequenceInfo()		
}		
}		
}		

FIG.32

31/118

CPI_type in the PlayList()	IN_timeのセマンティクス
EP_map type	IN_timeは、PlayItemの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	IN_timeは、TU_map_time_axis上の時刻でなければならない。かつ、IN_timeは、time_unitの精度に丸めて表さなければならない。IN_timeは、次に示す等式により計算される。 $IN\_time = TU\_start\_time \% 2^{32}$

FIG.33

32/118

CPI_type in the PlayList()	OUT_timeのセマンティクス
EP_map type	OUT_timeは、次に示す等式により計算される Presentation_end_TSの値の上位32ビットを示さなければならない。 $\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$ ここで、 PTS_outは、PlayItemの中で最後のプレゼンテーションユニットに対応する33ビット長のPTSである。 AU_durationは、最後のプレゼンテーションユニットの90kHz単位の表示期間である。
TU_map type	OUT_timeは、TU_map_time_axis上の時刻でなければならない。かつ、OUT_timeは、time_unitの精度に丸めて表さなければならない。OUT_timeは、次に示す等式により計算される。 $\text{OUT\_time} = \text{TU\_start\_time} \% 2^{32}$

FIG.34

33/118

connection _condition	意味
00	・ 先行するPlayItemと現在のPlayItemの接続は、シームレス再生の保証がなされていない。 ・ PlayListのCPI_typeがTU_map typeである場合、connection_conditionは、この値をセットされねばならない。
01	・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。 ・ 先行するPlayItemと現在のPlayItemは、システムタイムベース（STCベース）の不連続点があるために分割されていることを表す。
10	・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。 ・ この状態は、Virtual PlayListに対してだけ許される。 ・ 先行するPlayItemと現在のPlayItemとの接続は、シームレス再生の保証がなされている。 ・ 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用して接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならない。
11	・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。 ・ 先行するPlayItemと現在のPlayItemは、シームレス再生の保証がなされている。 ・ 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用しないで接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならない。

FIG.35

34/118

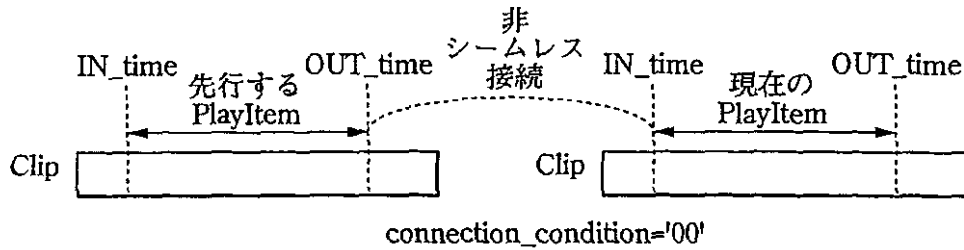


FIG.36A

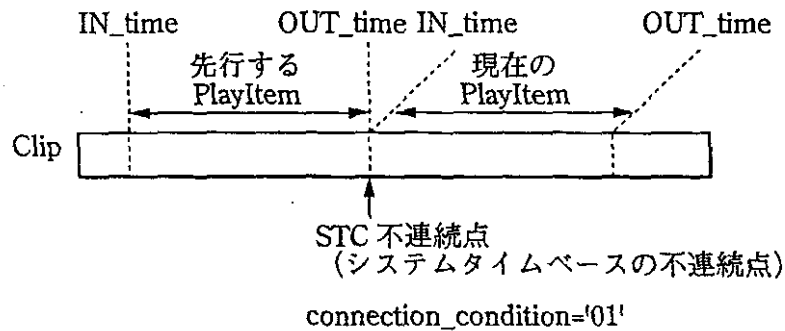


FIG.36B

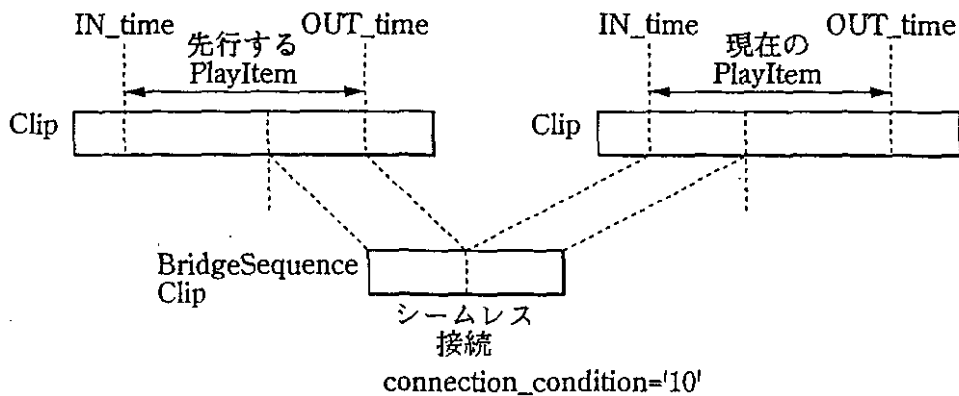


FIG.36C

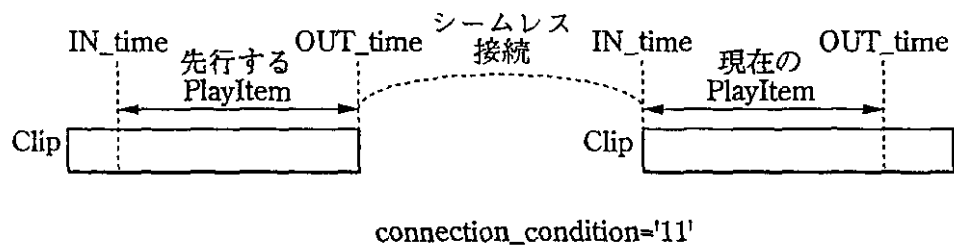


FIG.36D

35/118

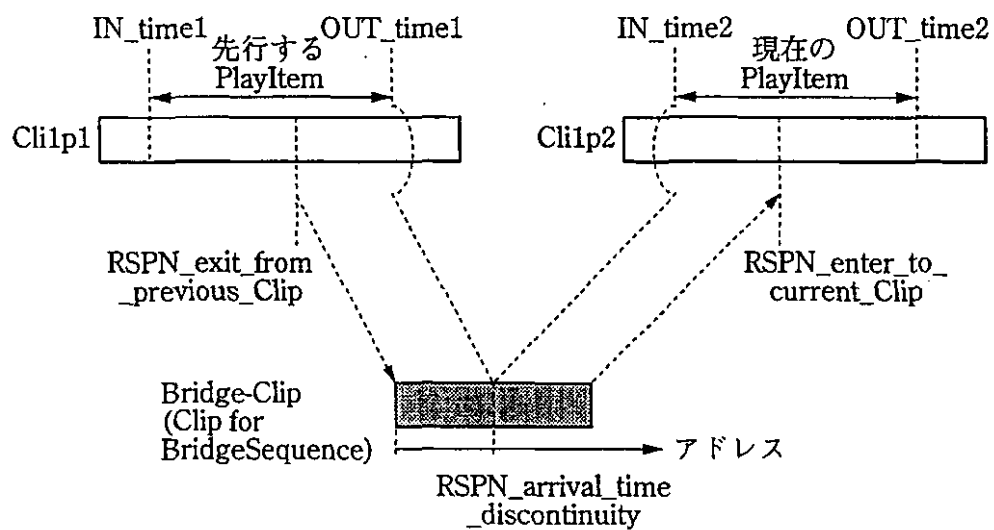


FIG.37

36/118

シンタクス	バイト数	略号
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

FIG.38

37/118

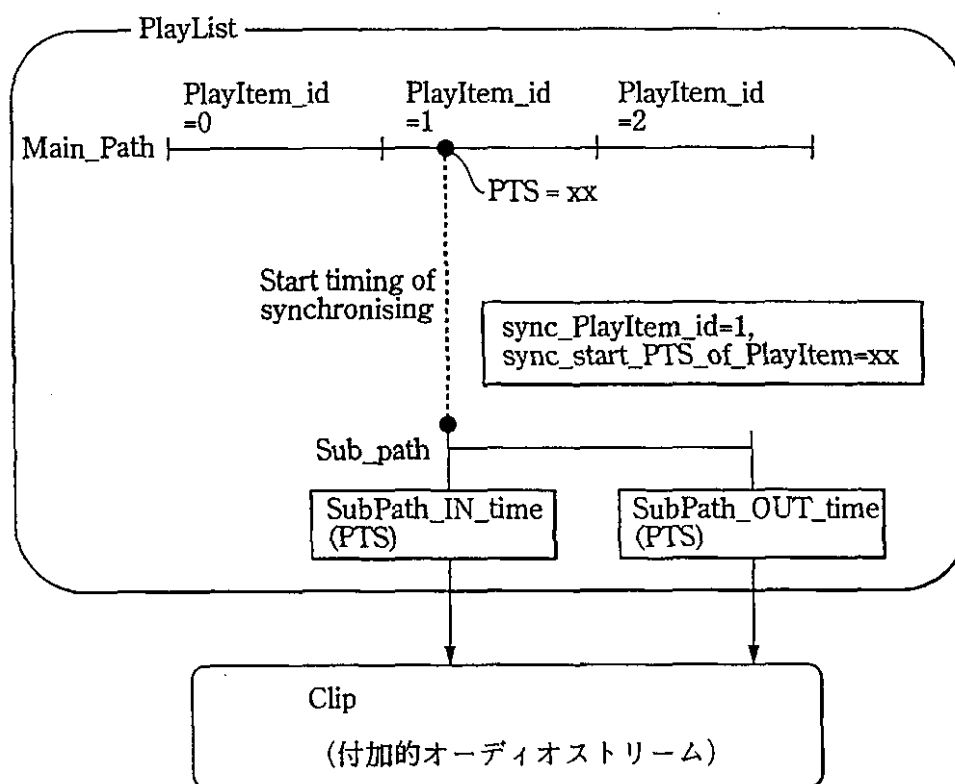


FIG.39

38/118

シンタクス	バイト数	略号
SubPlayItem0{		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

FIG.40

39/118

SubPath_type	意味
0x00	Auxiliary audio steam path
0x01-0xff	reserved

FIG.41

40/118

シンタクス	バイト数	略号
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.42

41/118

Mark_type	意味	コメント
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

FIG.43

42/118

CPI_type in the PlayList()	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次 に示す等式により計算される。 $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

FIG.44

43/118

シンタクス	バイト数	略号
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.45

44/118

シンタクス	バイト数	略号
ClipInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

FIG.46

45/118

Clip_stream_type	意味
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2-255	Reserved

FIG.47

46/118

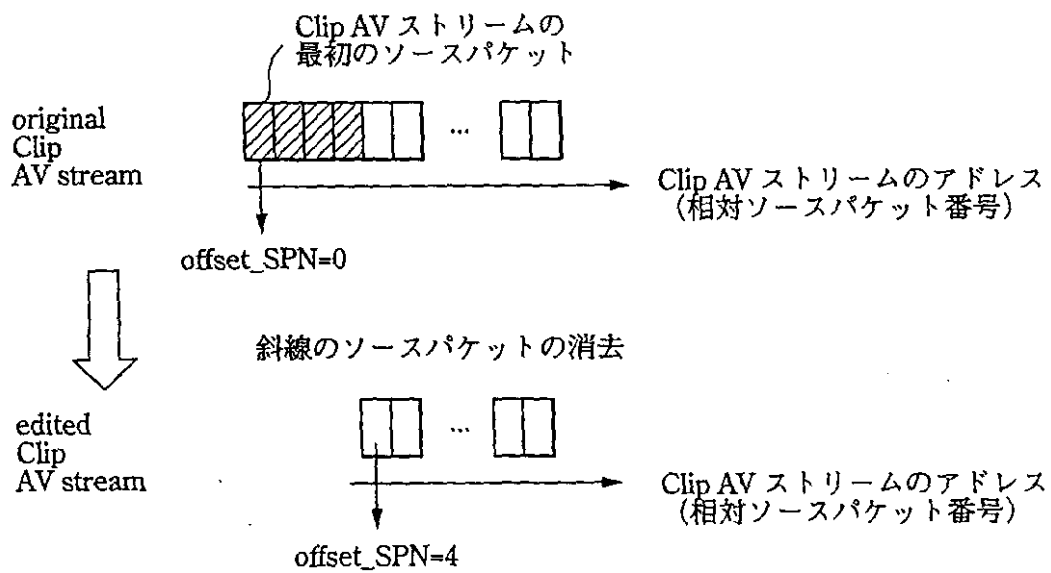


FIG.48

47/118

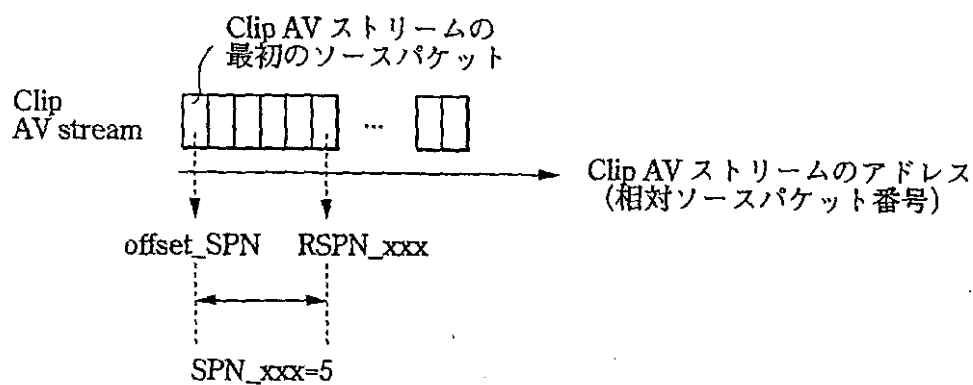


FIG.49

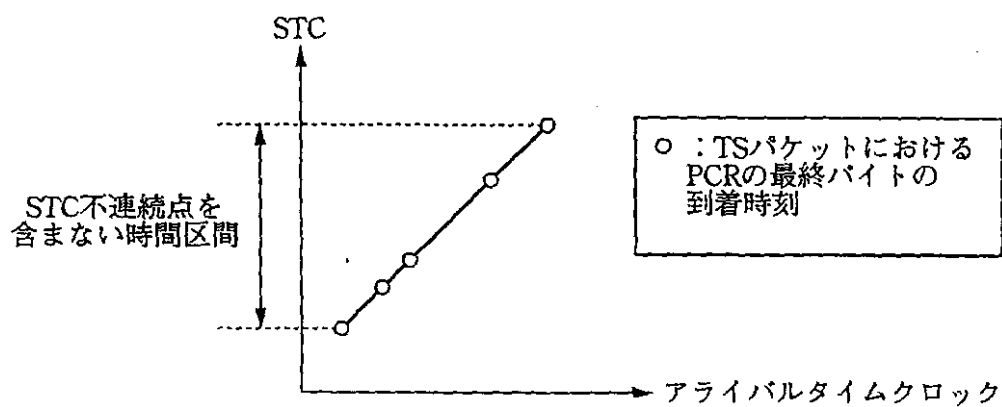


FIG.50A

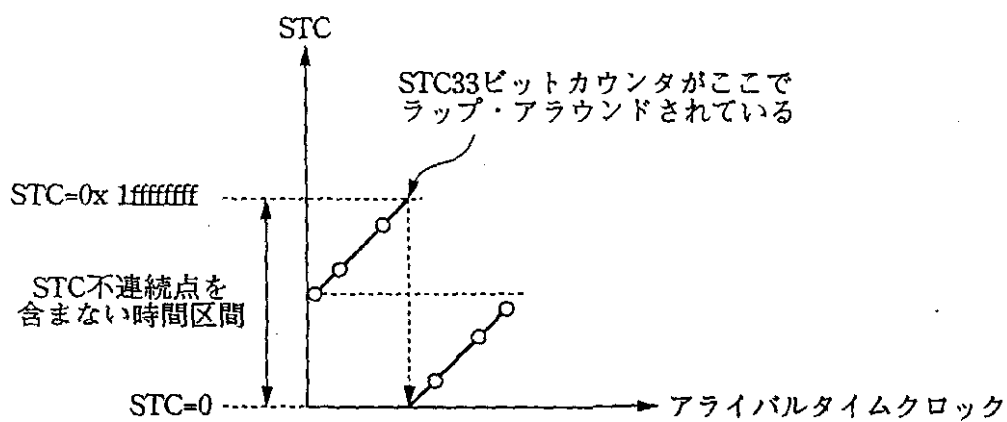


FIG.50B

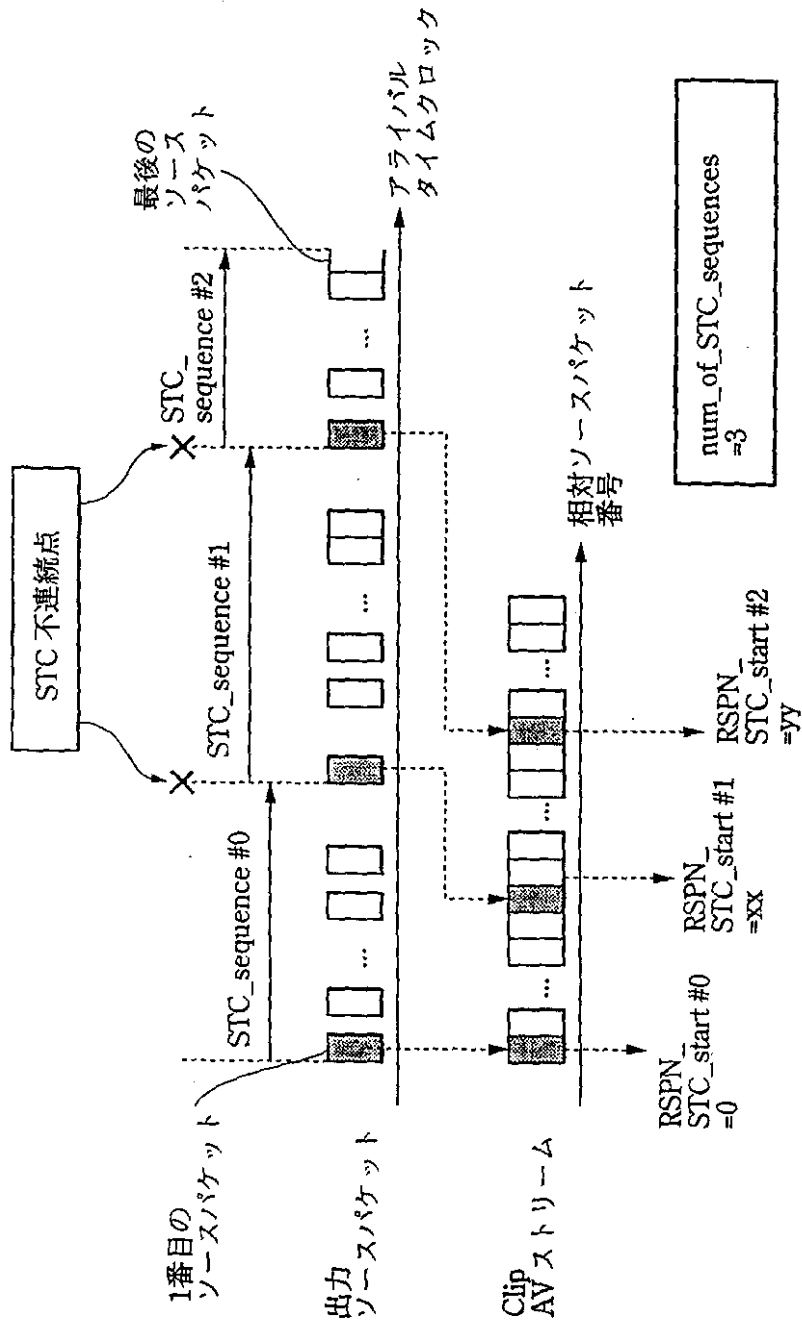


FIG.51

49/118

シンタクス	バイト数	略号
STC_Info(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for (STC_sequence_id=0; STC_sequence_id<num_of_STC_sequences; STC_sequence_id++){		
resereved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

FIG.52

50/118

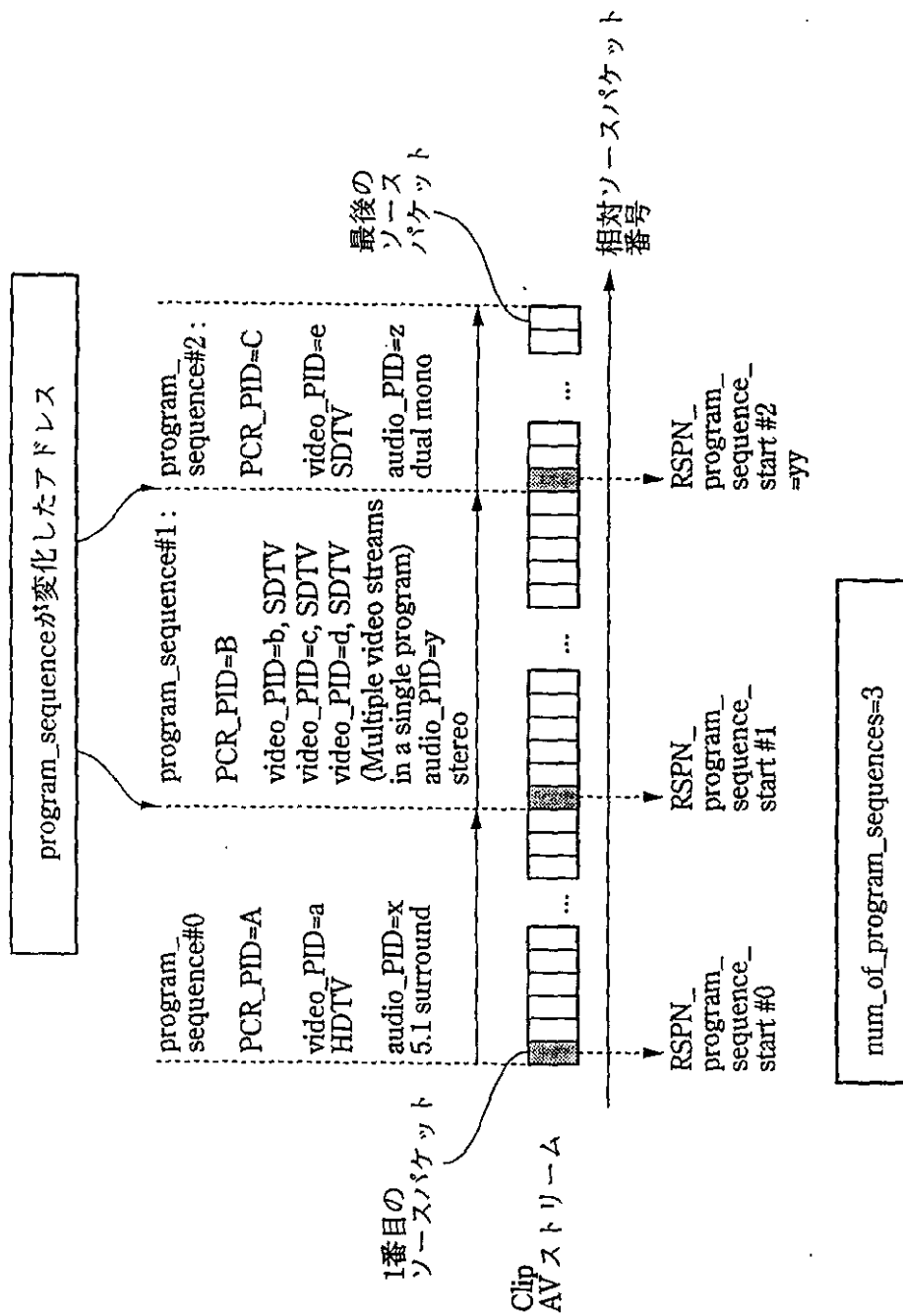


FIG.53

51/118

シンタクス	バイト数	略号
ProgramInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<number_of_program_sequences;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<number_of_videos;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<number_of_audios;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

FIG.54

52/118

シンタクス	バイト数	略号
VideoCodingInfo() (		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
)		

FIG.55

53/118

video_format	意味
0	480i
1	576i
2	480p (including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

FIG.56

frame_rate	意味
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

FIG.57

54/118

display_aspect_ratio	意味
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ration
4-254	reserved
255	No information

FIG.58

シンタクス	バイト数	略号
AudioCodingInfo() {		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

FIG.59

56/118

audio_coding	意味
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

FIG.60

57/118

audio_component_type	意味
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

FIG.61

sampling_frequency	意味
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

FIG.62

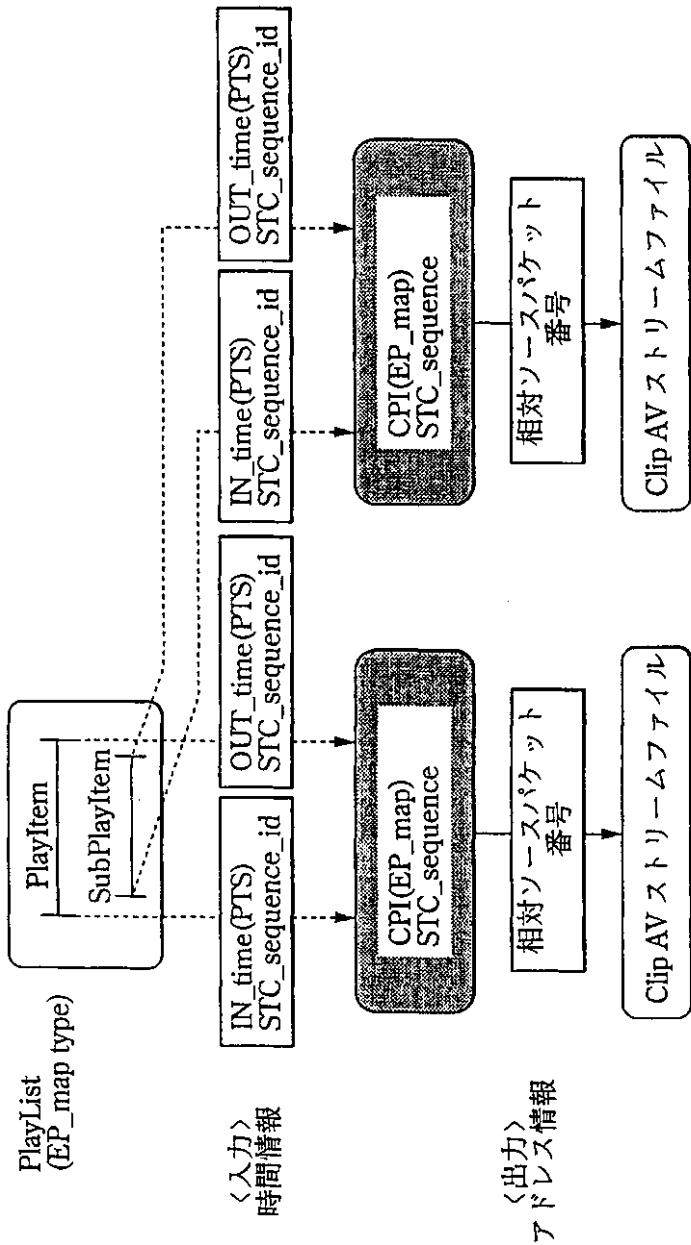


FIG.63

59/118

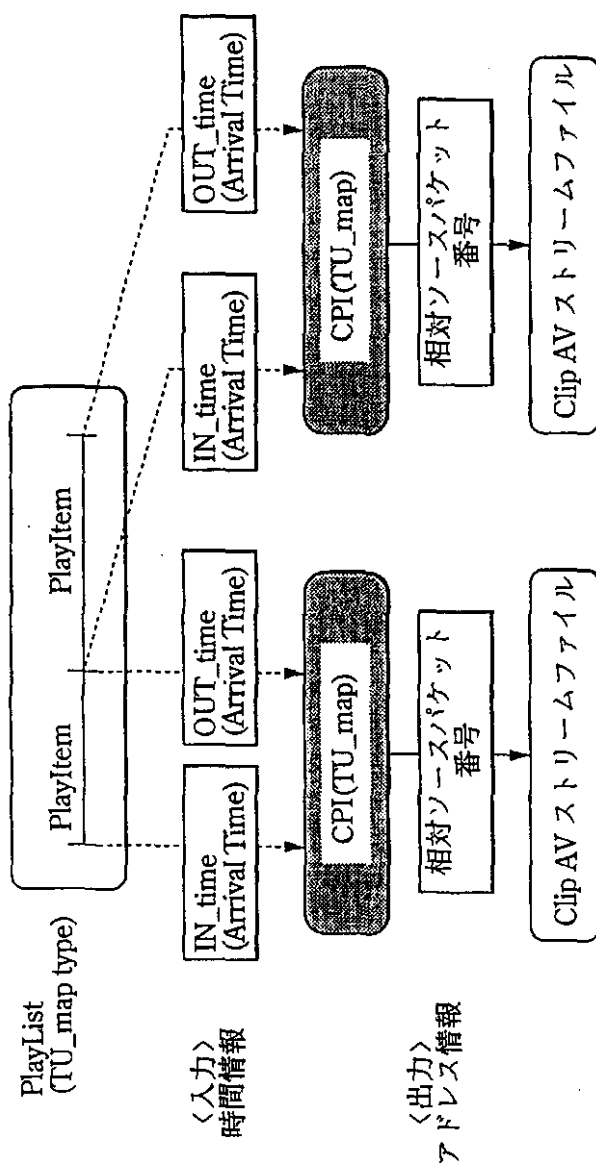


FIG. 64

60/118

シンタクス	バイト数	略号
CPI{		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

FIG.65

61/118

CPI_type	意味
0	EP map type
1	TU map type

FIG.66

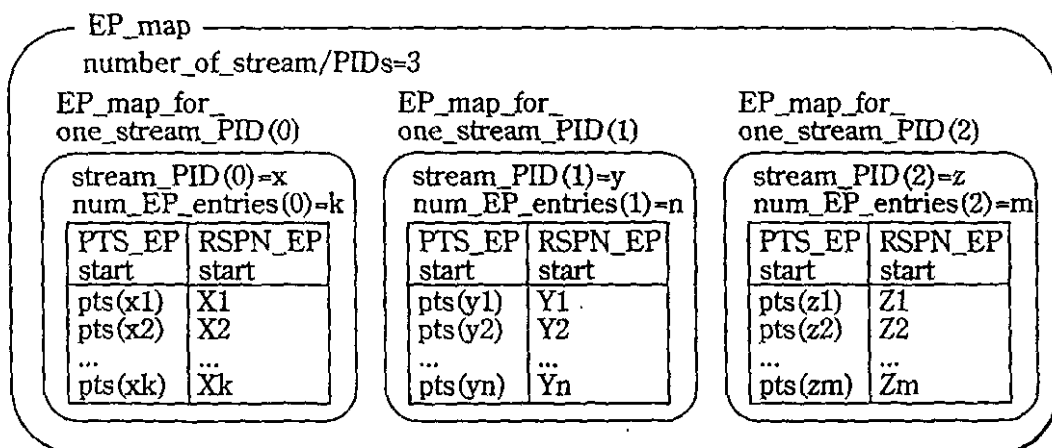
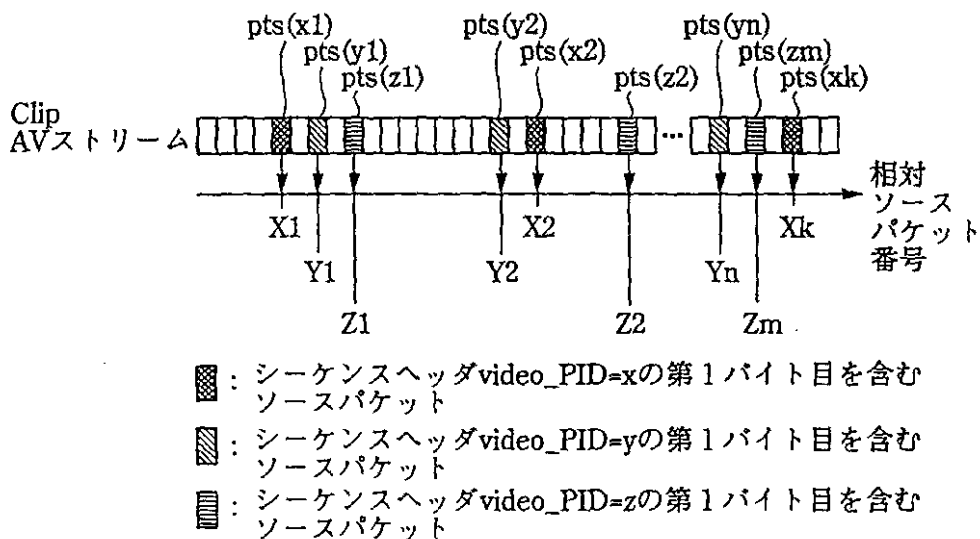
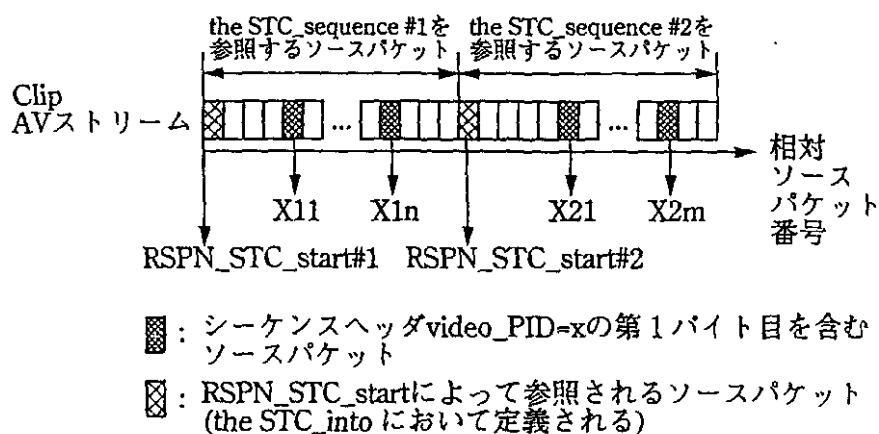


FIG.67

62/118



EP_map_for_one_stream_PID
video_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11	) STC_sequence #1に 属するデータ
pts(x1n)	X1n	
...	...	) STC_sequence #2に 属するデータ
pts(x21)	X21	
pts(x2m)	X2m	

→ boundary

RSPN_STC_start #2 < X21

FIG.68

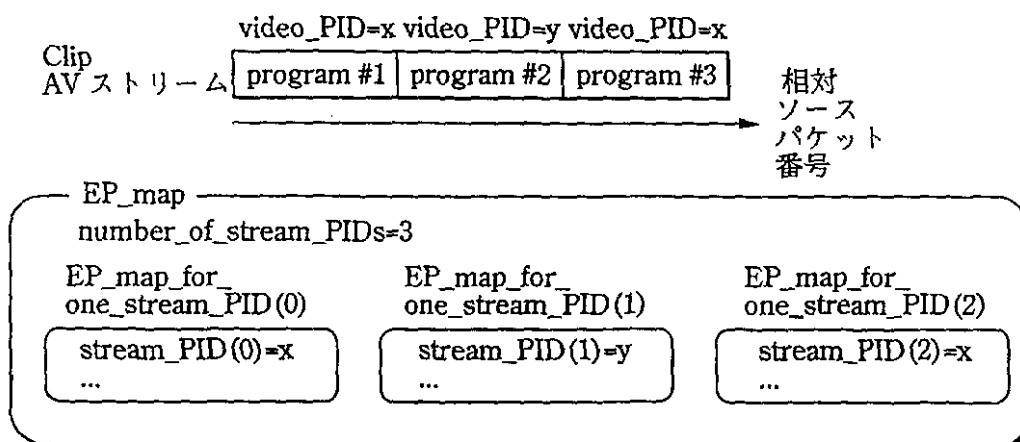


FIG.69

63/118

シンタクス	バイト数	略号
EP_map0{		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

FIG.70

64/118

EP_type	意味
0	video
1	audio
2-15	reserved

FIG.71

65/118

シンタクス	バイト数	略号
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

FIG.72

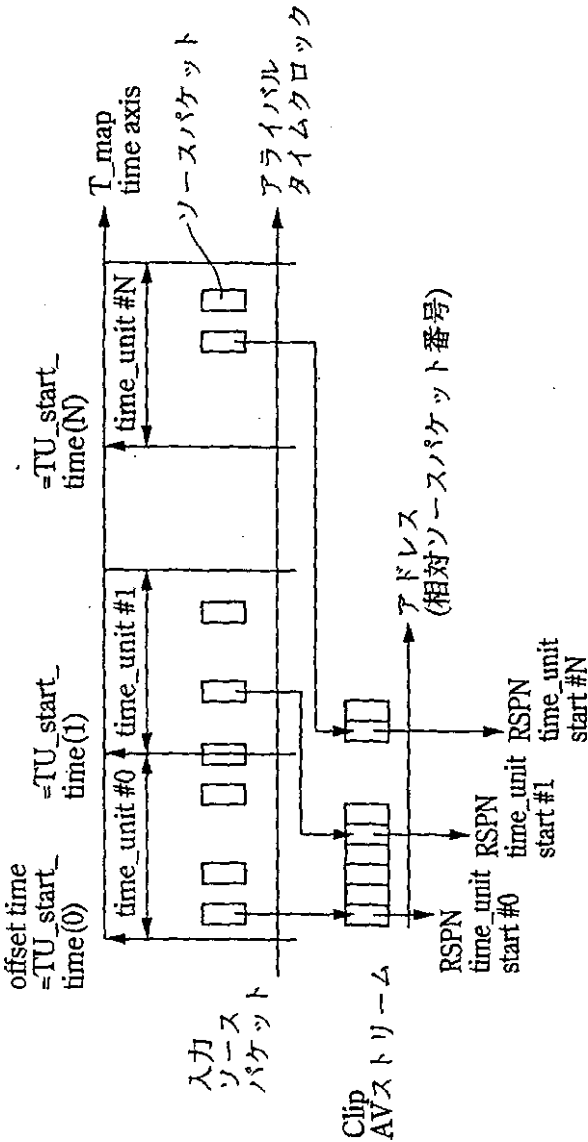


FIG.73

67/118

シンタクス	バイト数	略号
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

FIG.74

68/118

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.75

69/118

Mark_type	意味	コメント
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーンチェンジポイントを示すマーク点
0x93-0xFF	reserved	

FIG.76

70/118

CPI_type in the CPI0	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次に示す等式により計算される。 $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

FIG.77

71/118

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
reserved_for_MakerID	16	bslbf
mark_entry()		
representative_picture_entry()		
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.78

Mark_type	意味	コメント
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーン開始ポイントを示すマーク点
0x93	Scene-end mark	シーン終了ポイントを示すマーク点
0x94	CM-start mark	CM開始ポイントを示すマーク点
0x95	CM-end mark	CM終了ポイントを示すマーク点
0x96-0xBF	DVRフォーマットが、ClipMarkを将来、拡張する時のために予約されている	
0xC0-0xFF	メーカー独自のアプリケーションで利用するマークに割り当て可能	

FIG.79

72/118

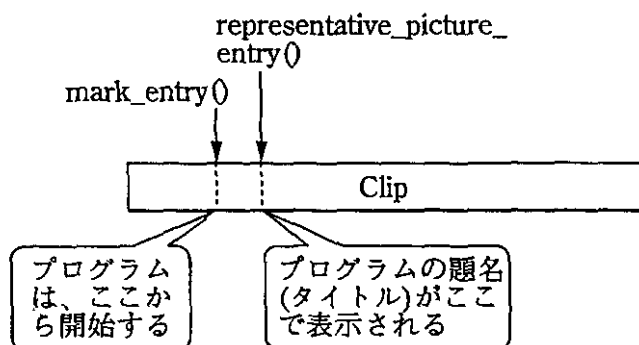


FIG.80

シンタクス	バイト数	略号
mark_entry()/representative_picture_entry(){		
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
}		

FIG.81

シンタクス	バイト数	略号
mark_entry()/representative_picture_entry(){		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

FIG.82

73/118

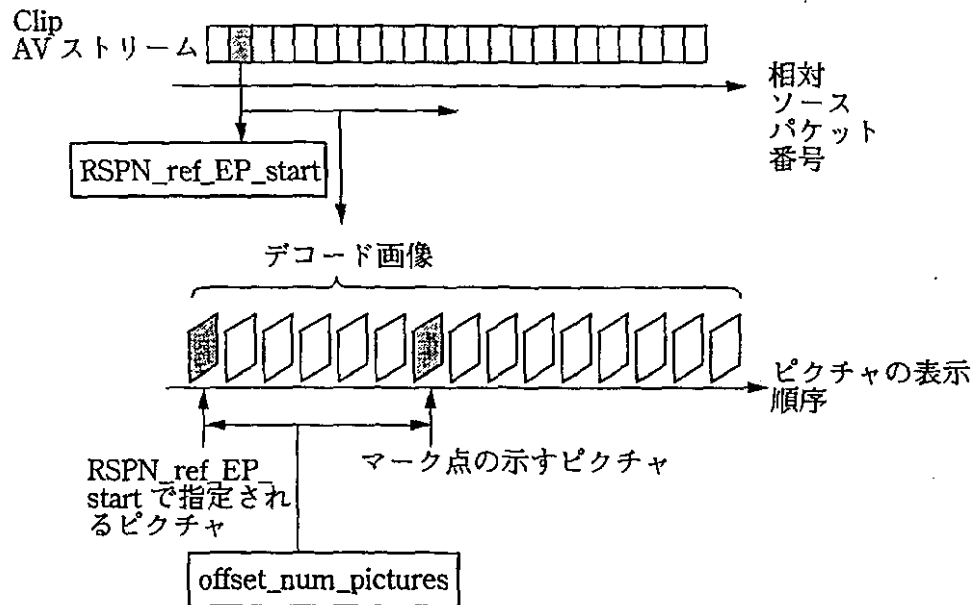


FIG.83

シンタクス	バイト数	略号
mark_entry()/representative_picture_entry0{		
RSPN_mark_point	32	uimsbf
}		

FIG.84

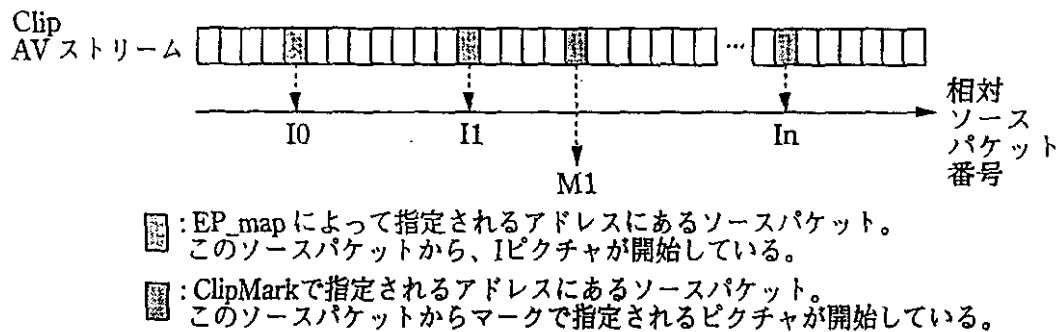


FIG.85

74/118

シンタクス	バイト数	略号
menu.thmb/mark.thmb() {		
reserved	256	bslbf
Thumbnail()		
for (i=0;i<N1;i++)		
padding_word	16	bslbf
}		

FIG.86

75/118

シンタクス	バイト数	略号
Thumbnail() {		
version_number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++) {		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (k=0; k<number_of_tn_blocks; k++) {		
tn_block	tn_block_size*1024*8	
}		
}		

FIG.87

76/118

Thumbnail_picture_format	意味
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

FIG.88

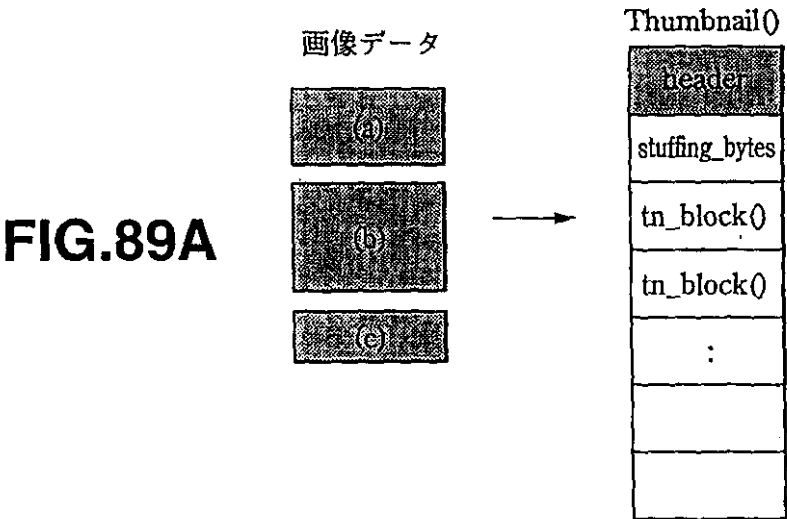
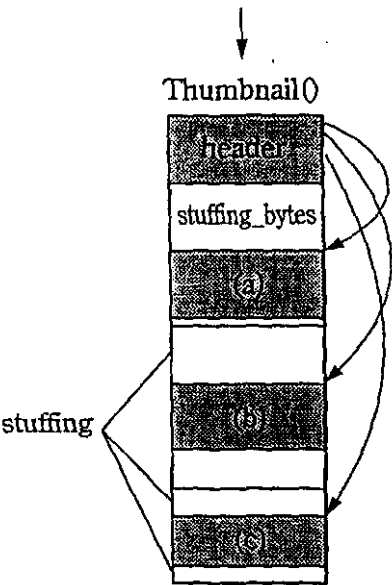


FIG.89B



77/118

DVR MPEG-2 トランスポートストリーム

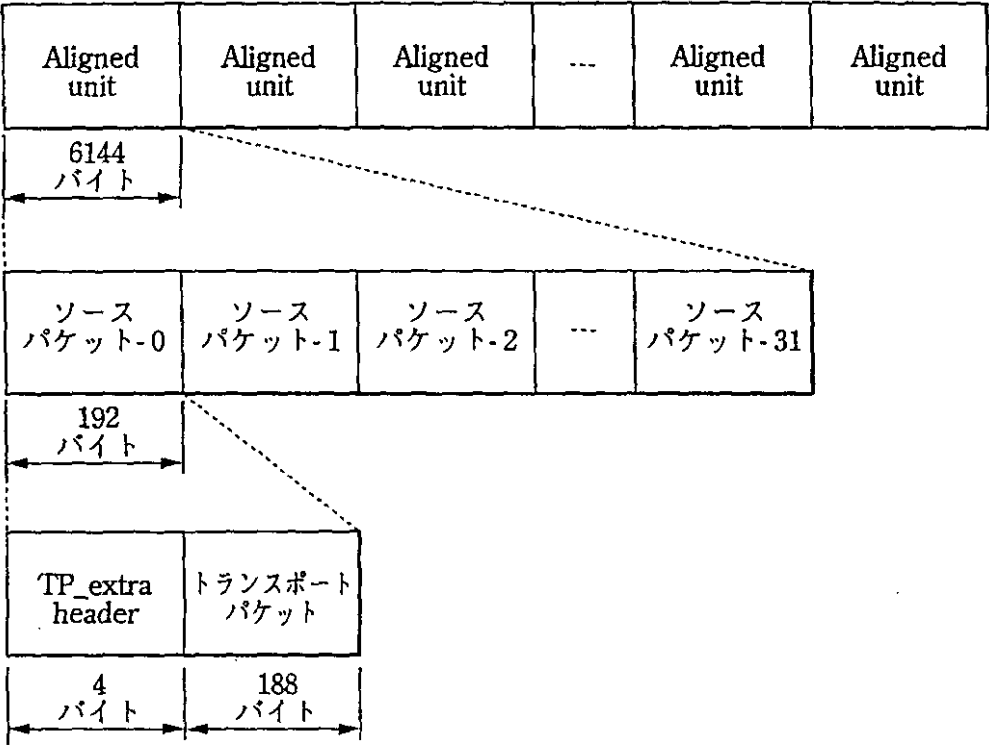


FIG.90

78/118

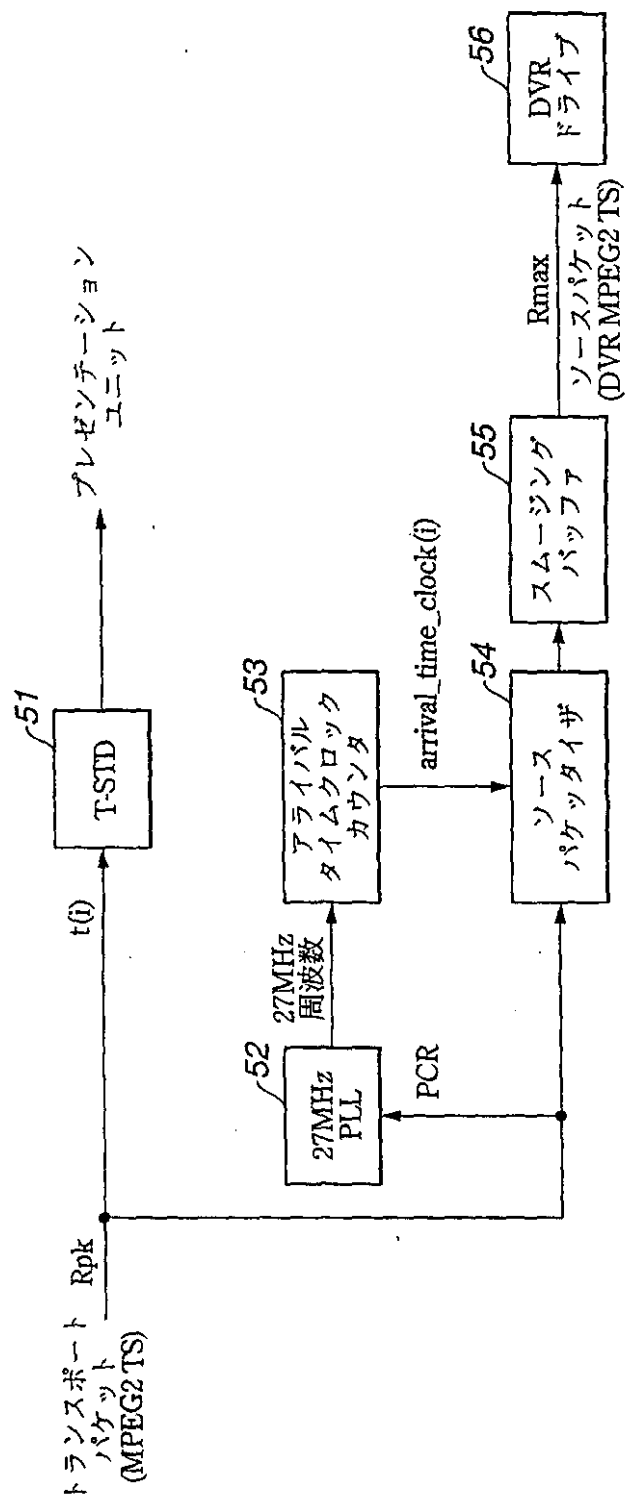


FIG.91

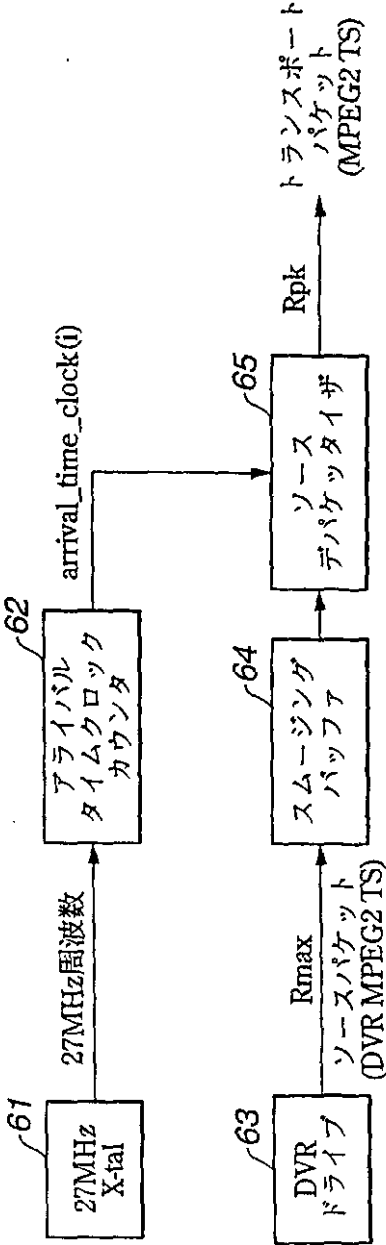


FIG.92

80/118

シンタクス	バイト数	略号
source_packet() {		
TP_extra_header()		
trasport_packet()		
}		

FIG.93

81/118

シンタクス	バイト数	略号
TP_extra_header() {		
copy_permission_indicator	2	uimbsf
arrival_time_stamp	30	uimbsf
}		

FIG.94

82/118

copy_permission _indicator	意味
00	copy free
01	no more copy
10	copy once
11	copy prohibited

FIG.95

83/118

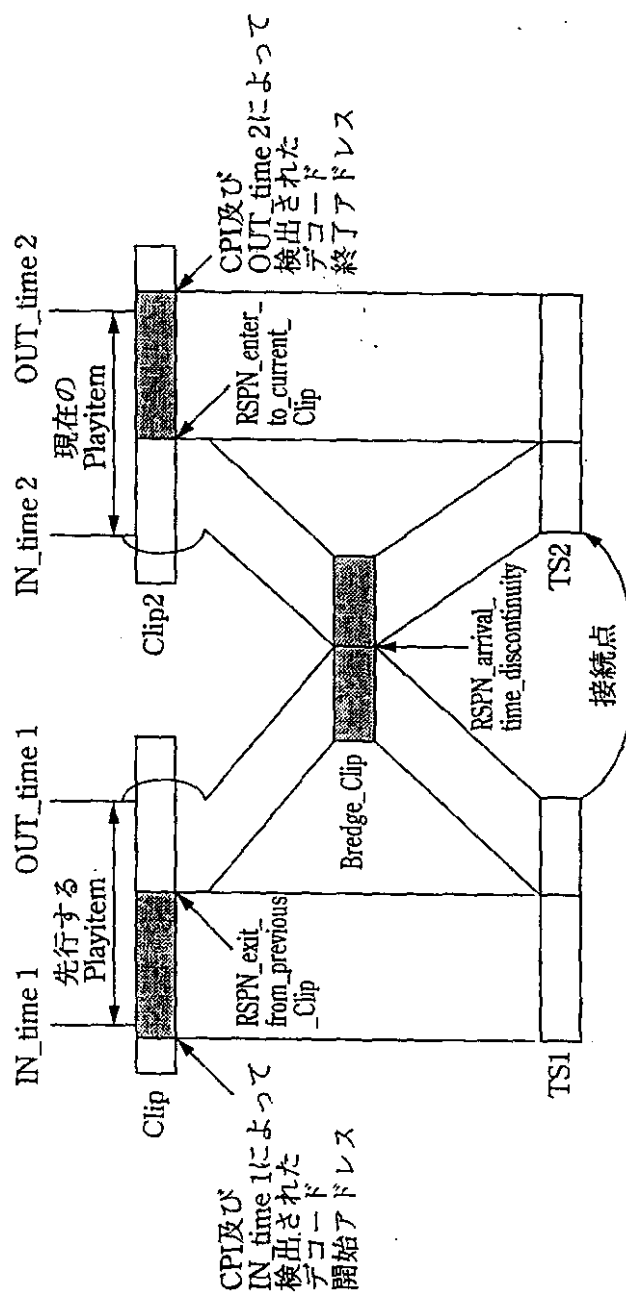


FIG.96

84/118

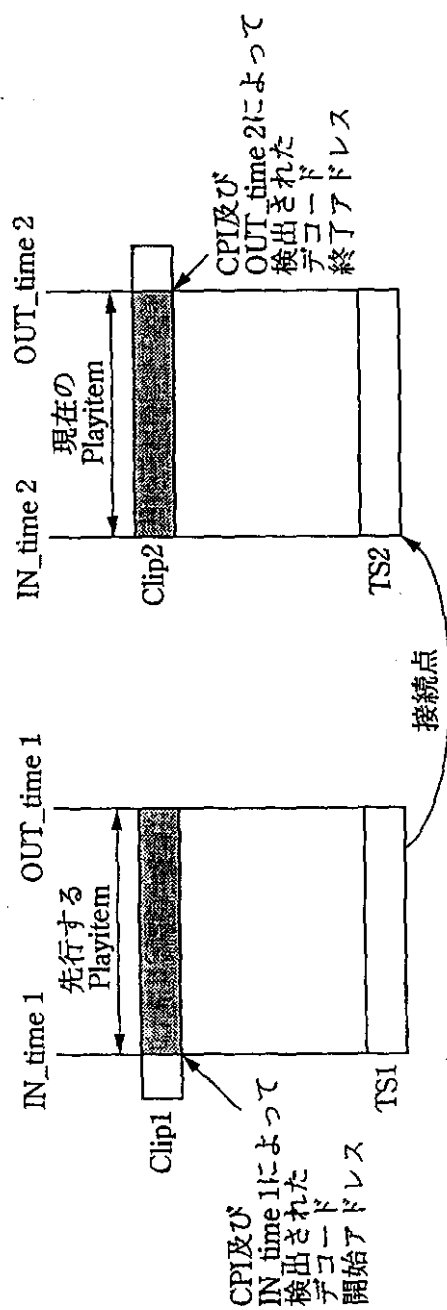


FIG.97

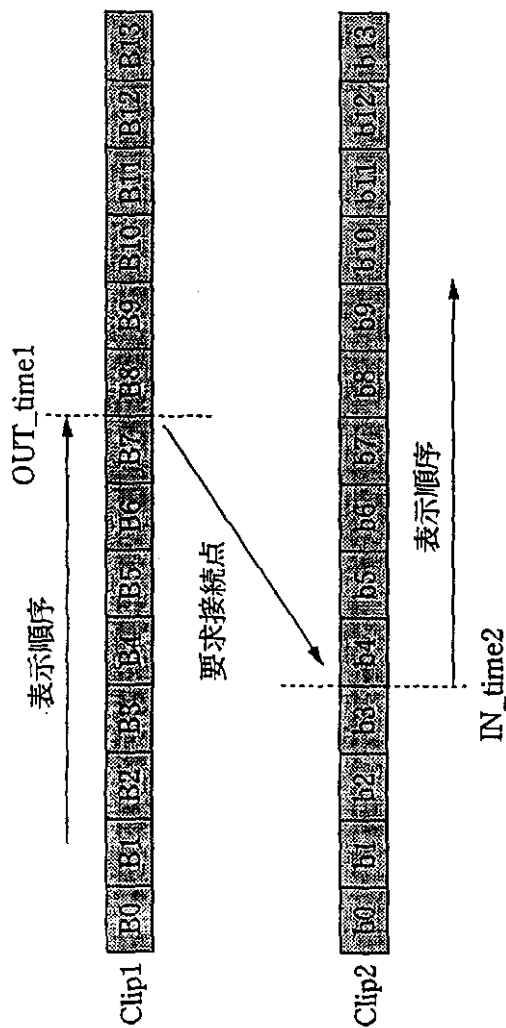


FIG.98

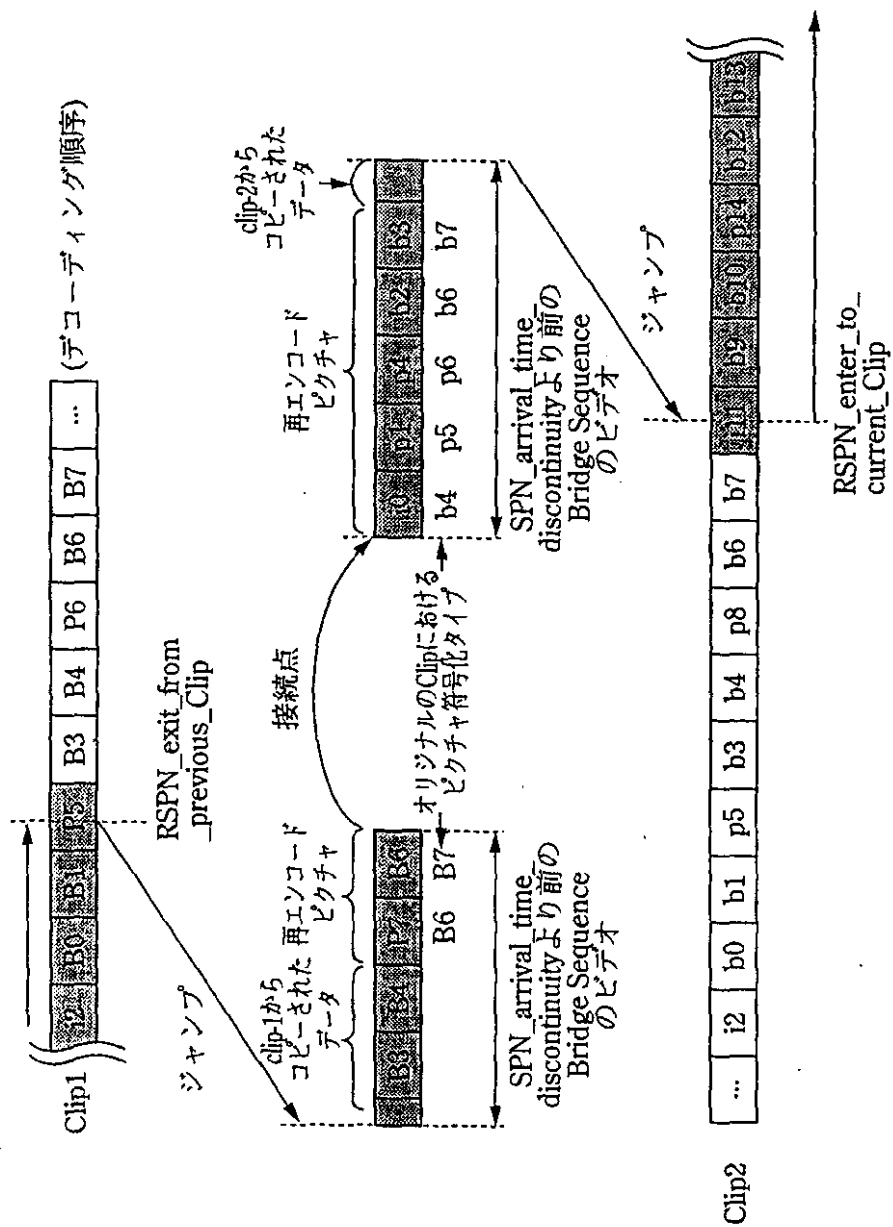


FIG.99

87/118

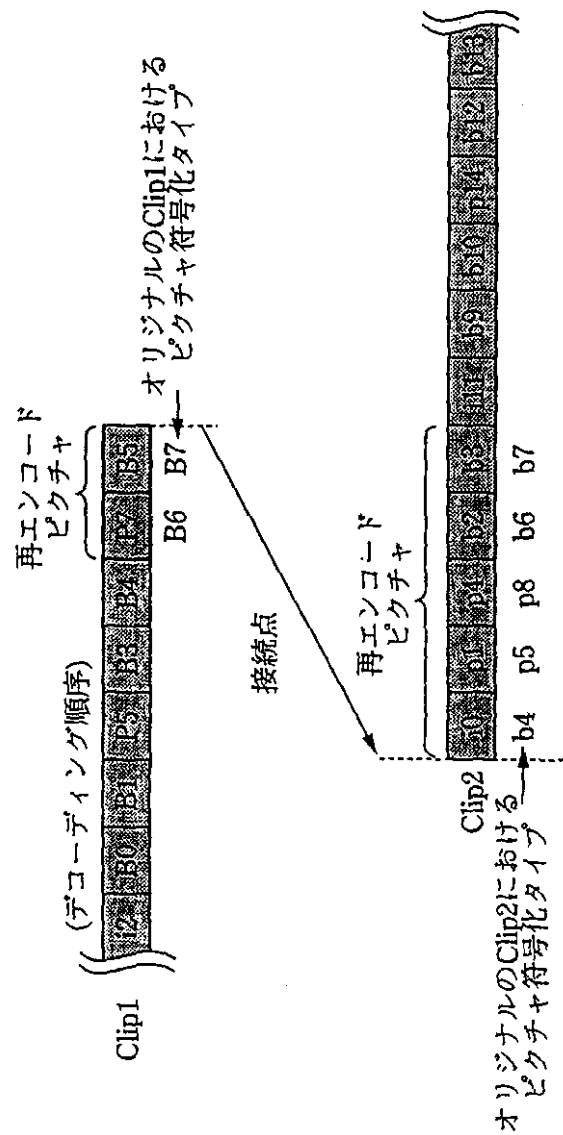


FIG.100

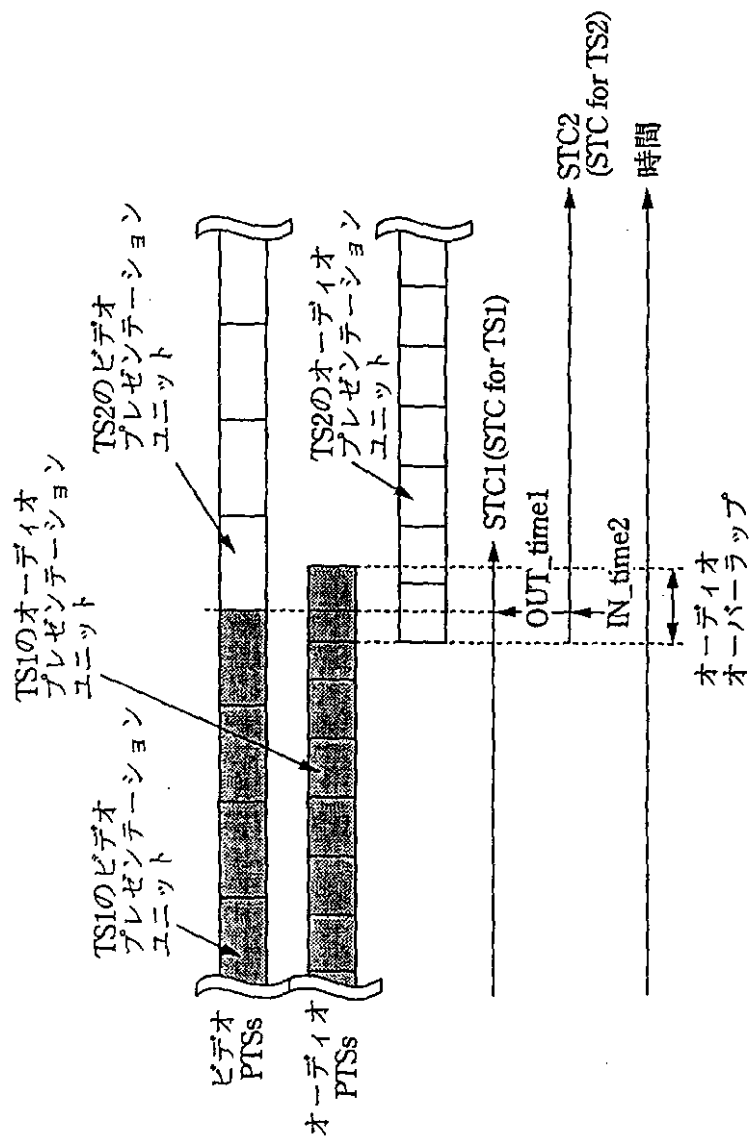


FIG. 101

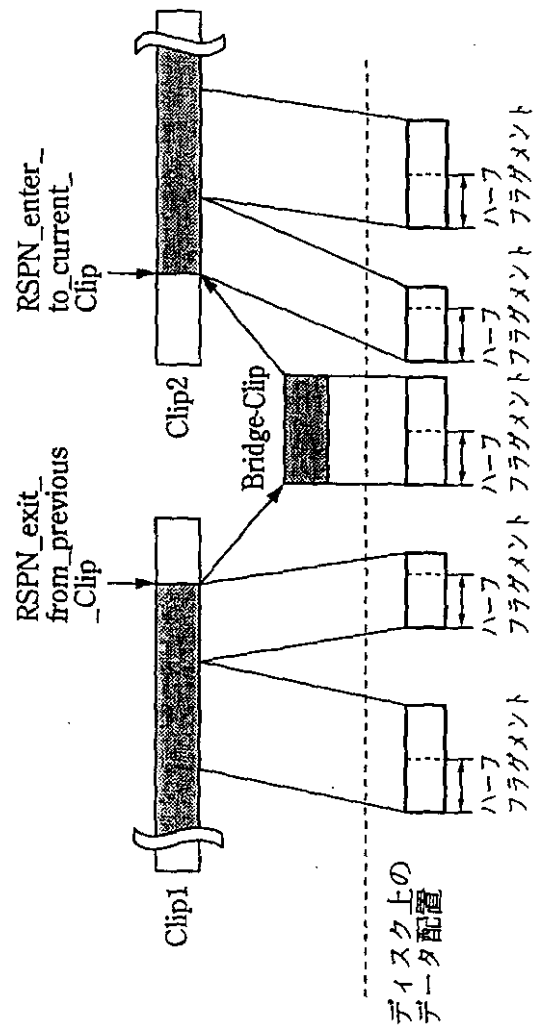


FIG.102

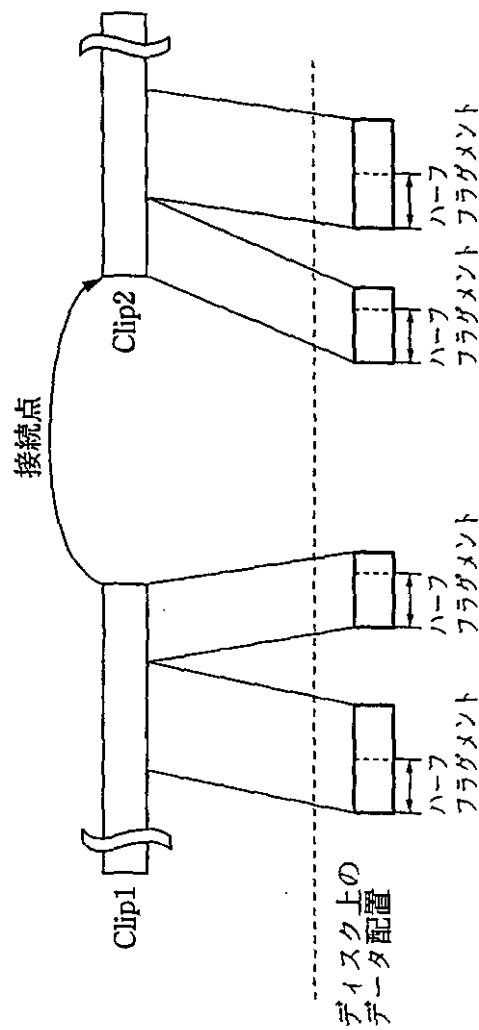


FIG.103

91/118

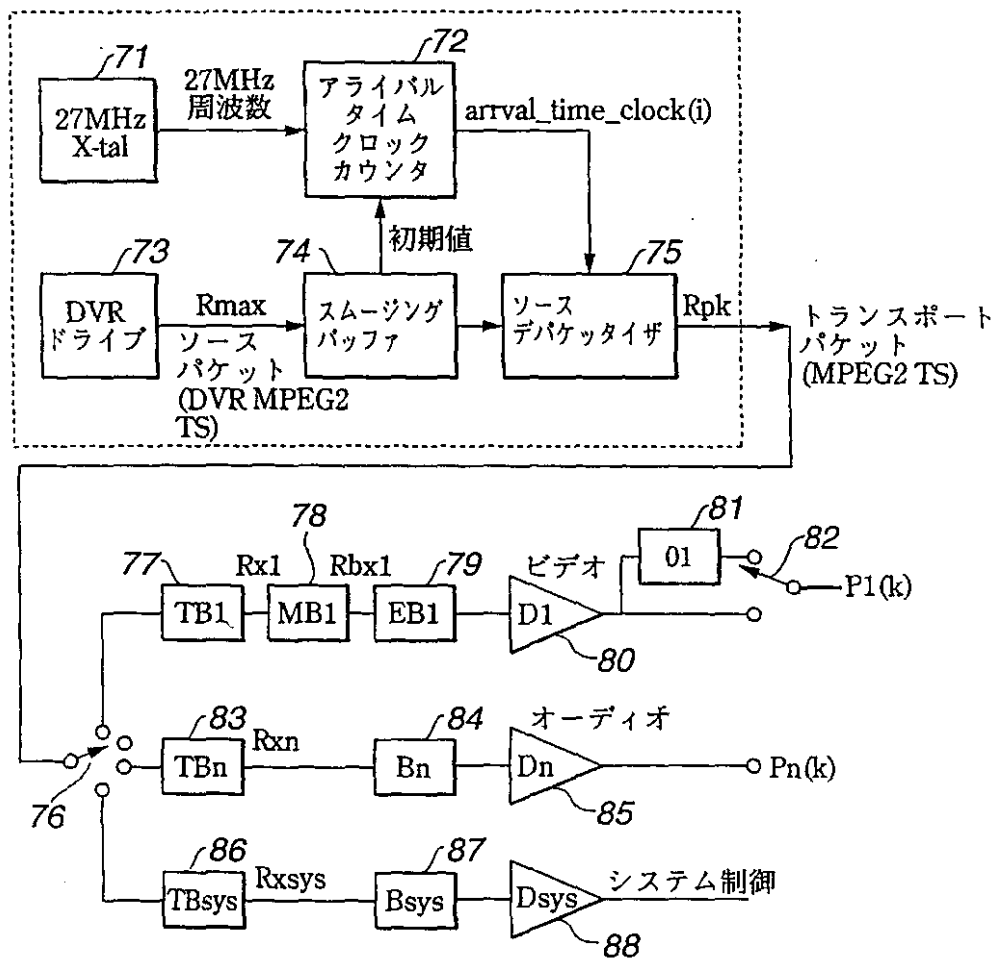


FIG.104

92/118

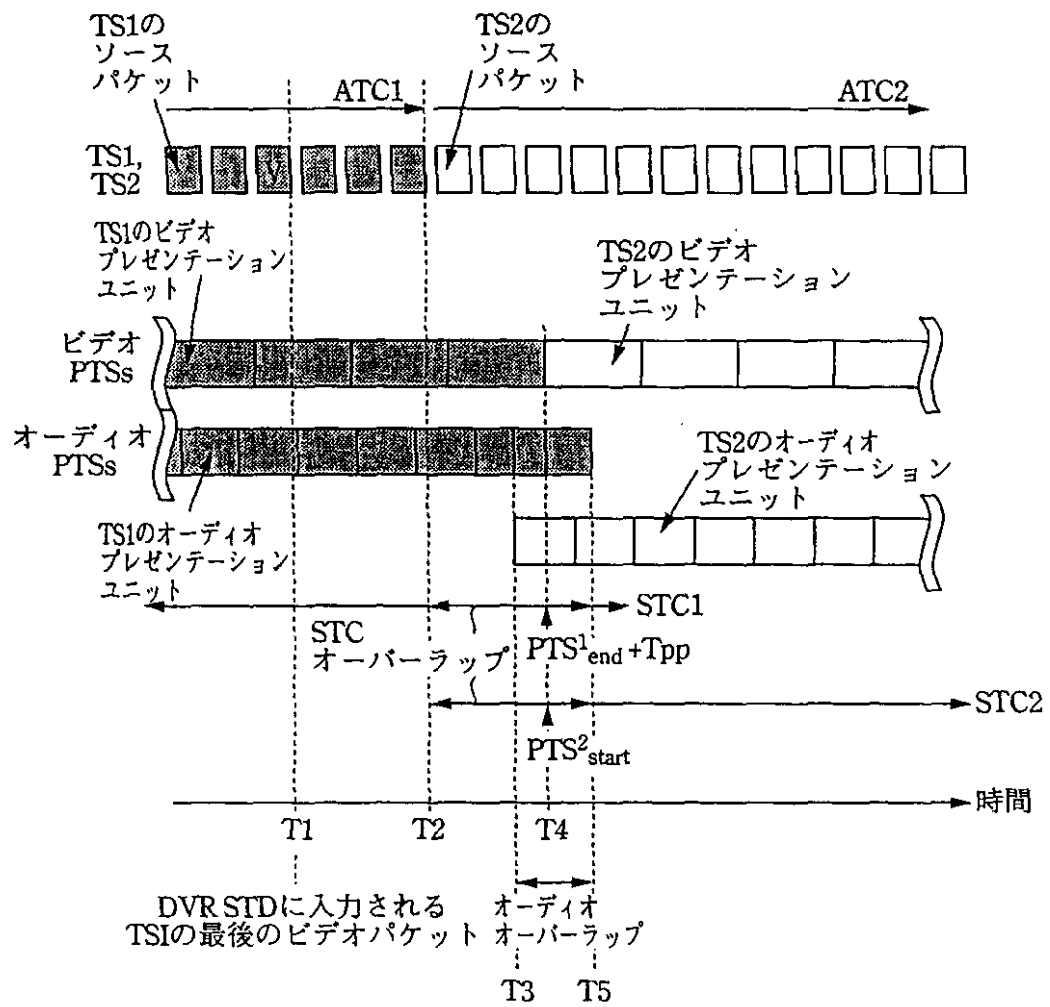


FIG.105

93/118

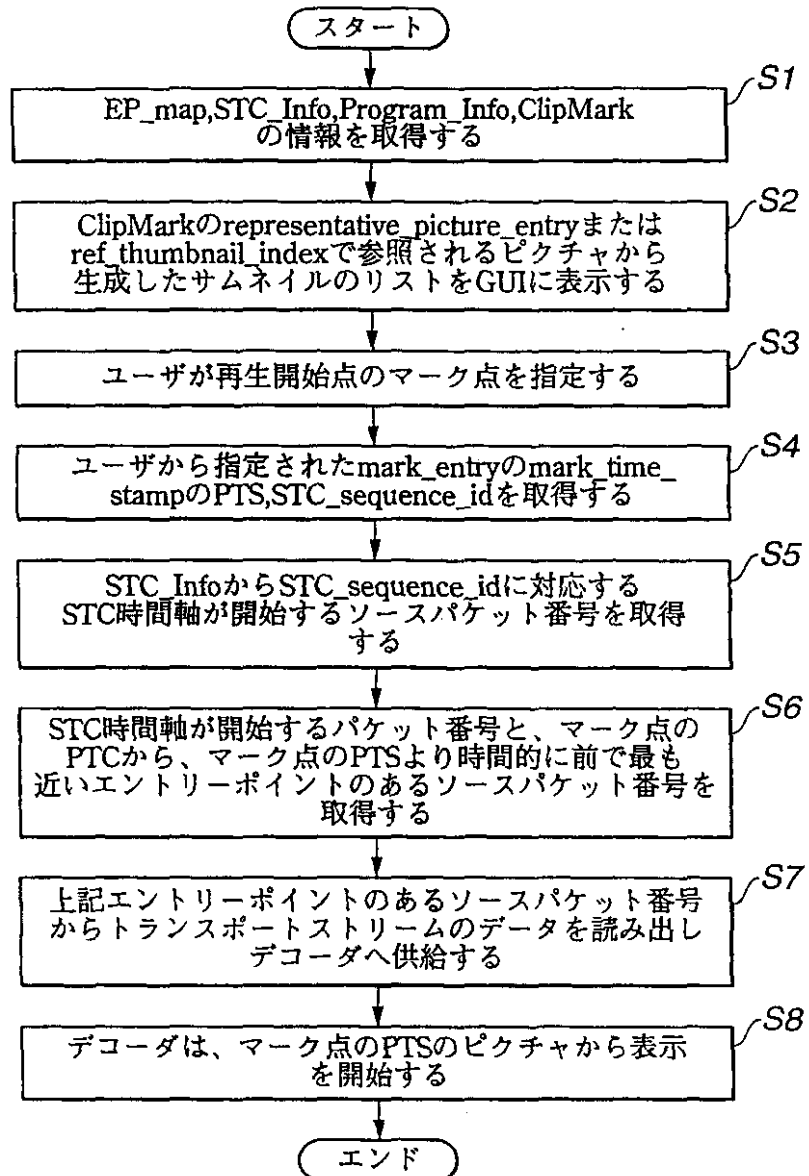


FIG.106

94/118

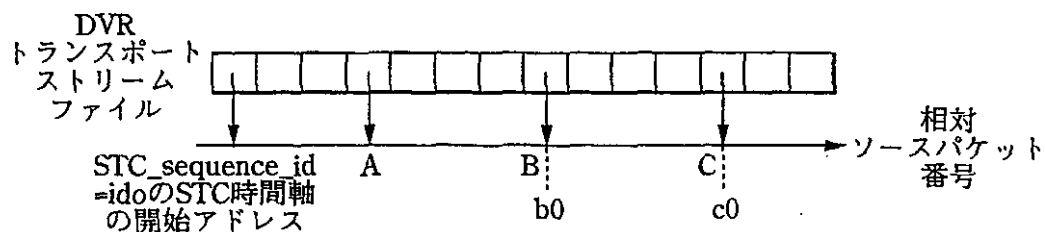


FIG.107

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.108

ClipMark

Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...	...	...	...	...
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(CM start)	PTS(b0)	id0	PTS(b0)	id0
0x95(CM end)	PTS(c0)	id0	PTS(c0)	id0
...	...	...	...	...

FIG.109

95/118

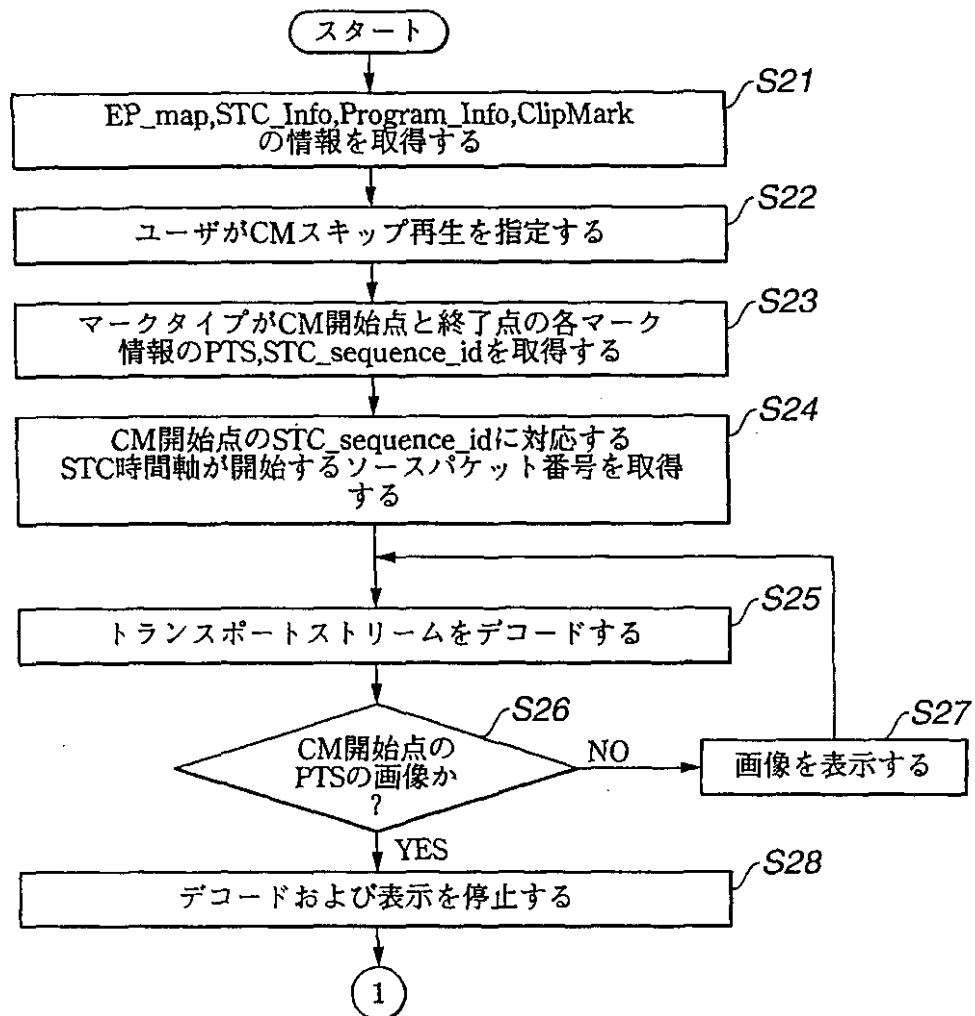


FIG.110

96/118

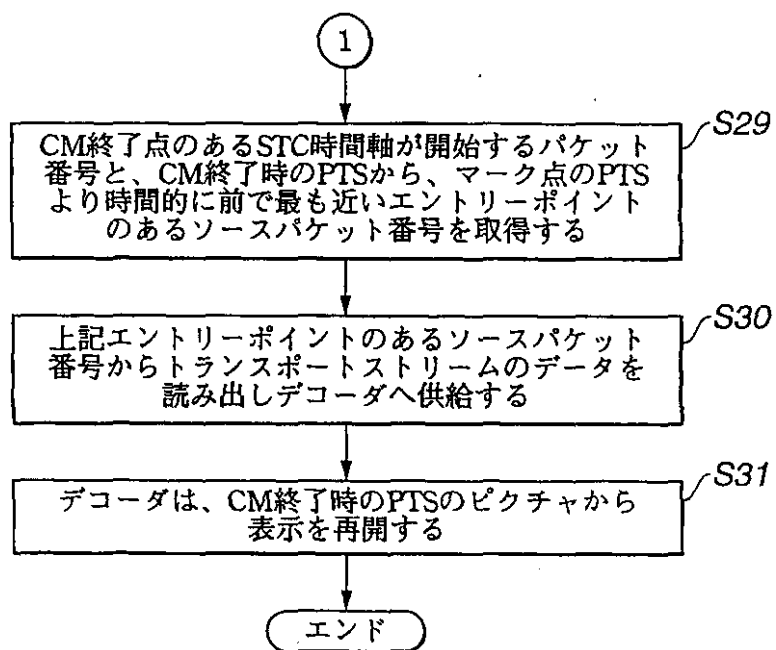


FIG.111

97/118

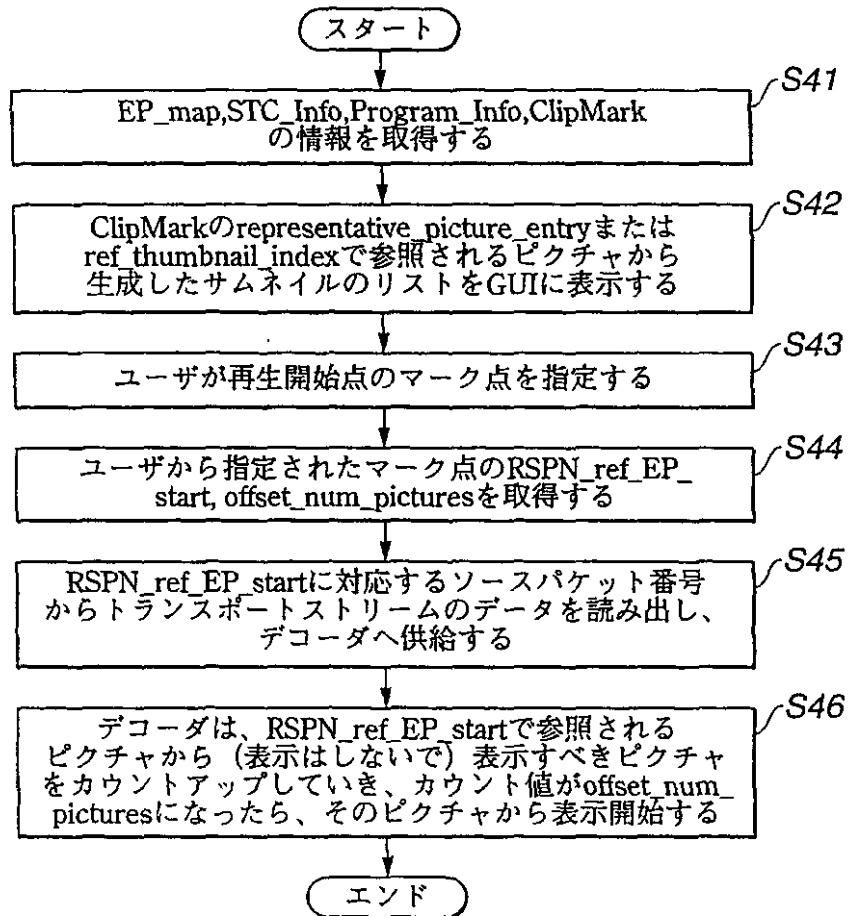


FIG.112

98/118

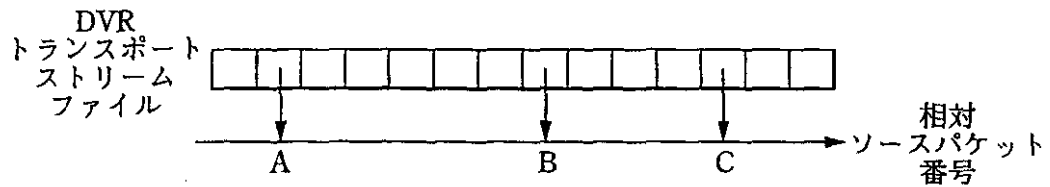


FIG.113

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.114

ClipMark

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...	...	...	...	...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...	...	...	...	...

FIG.115

99/118

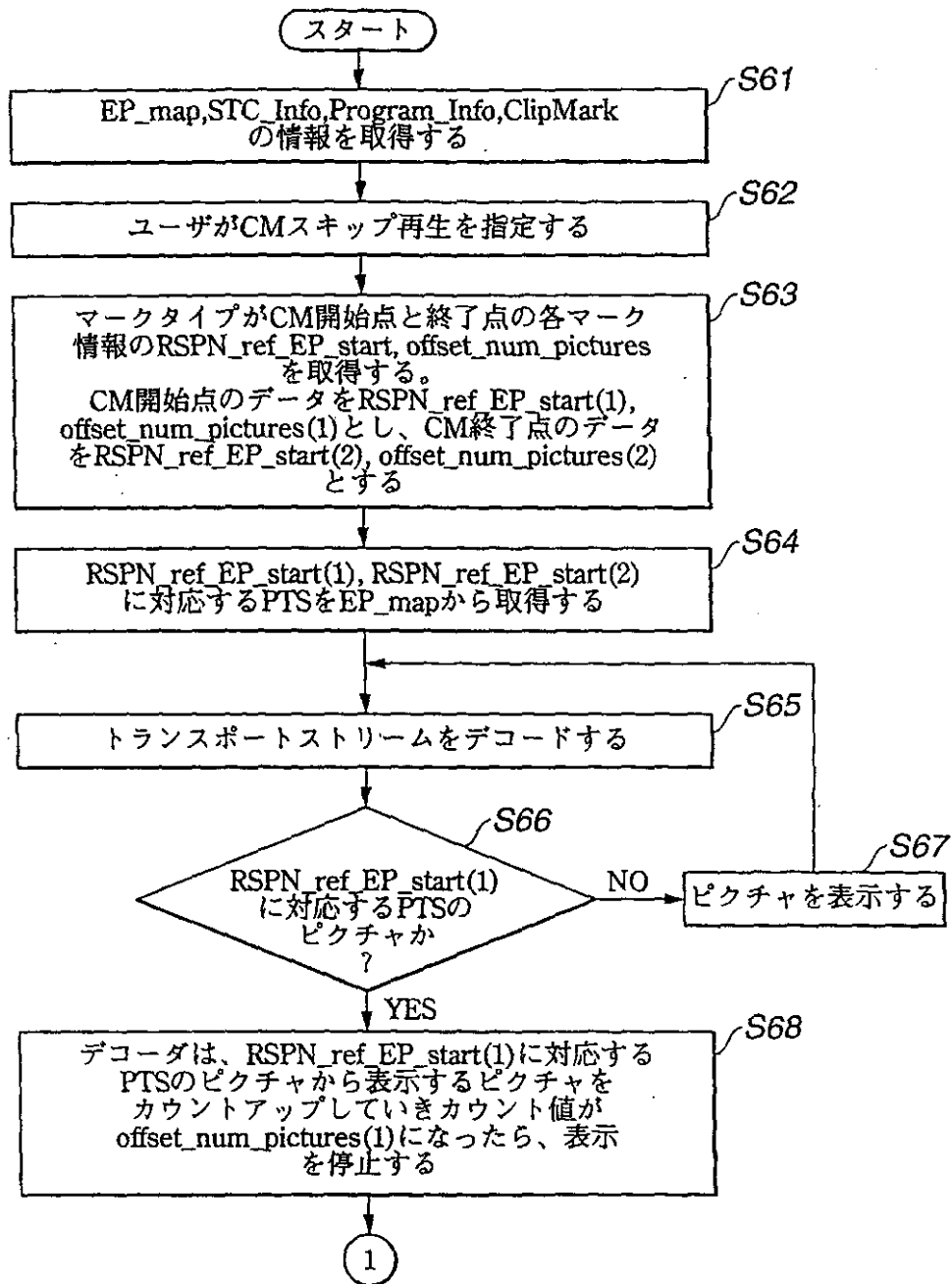


FIG.116

100/118

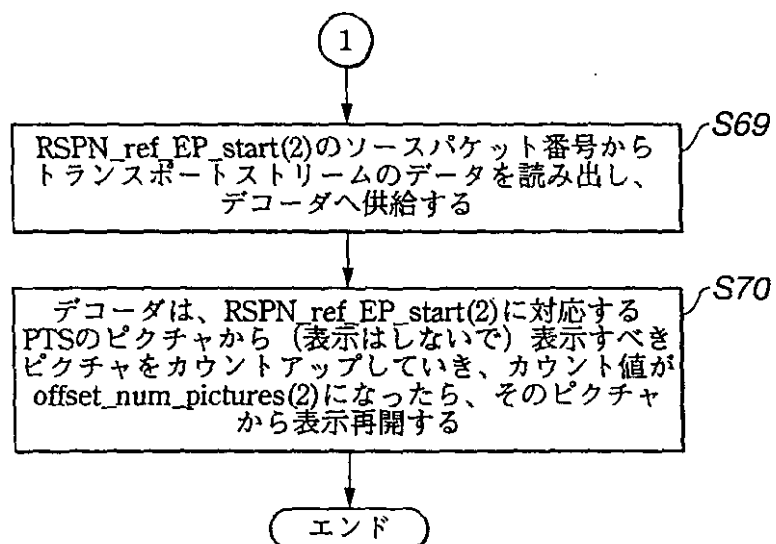


FIG.117

101/118

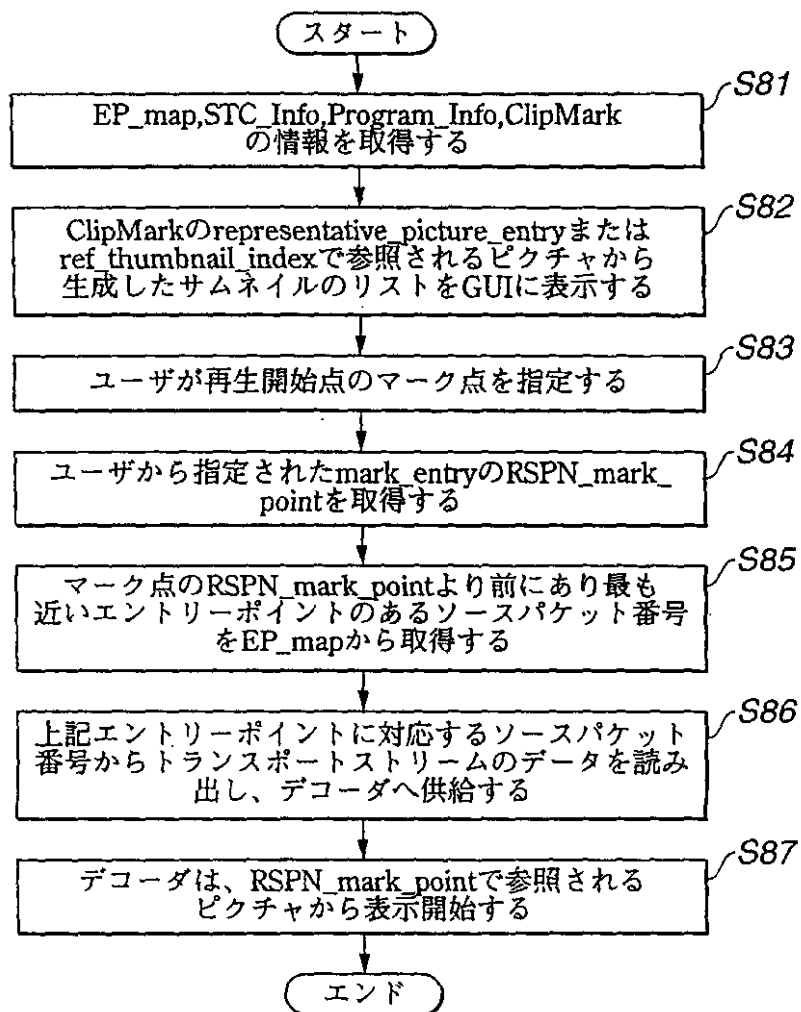


FIG.118

102/118

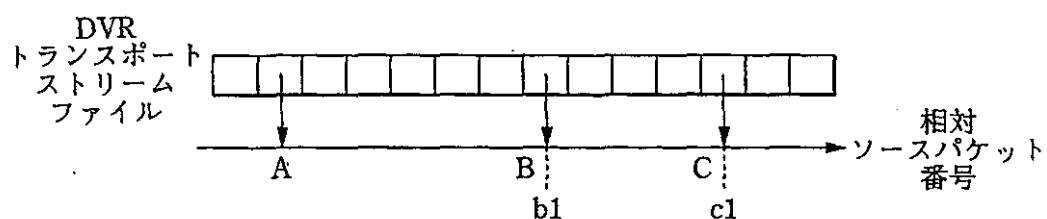


FIG.119

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.120

ClipMark

mark_type	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...	...	...
0x92(scene start)	a1	a2
0x94(CM start)	b1	b1
0x95(CM end)	c1	c1
...	...	...

FIG.121

103/118

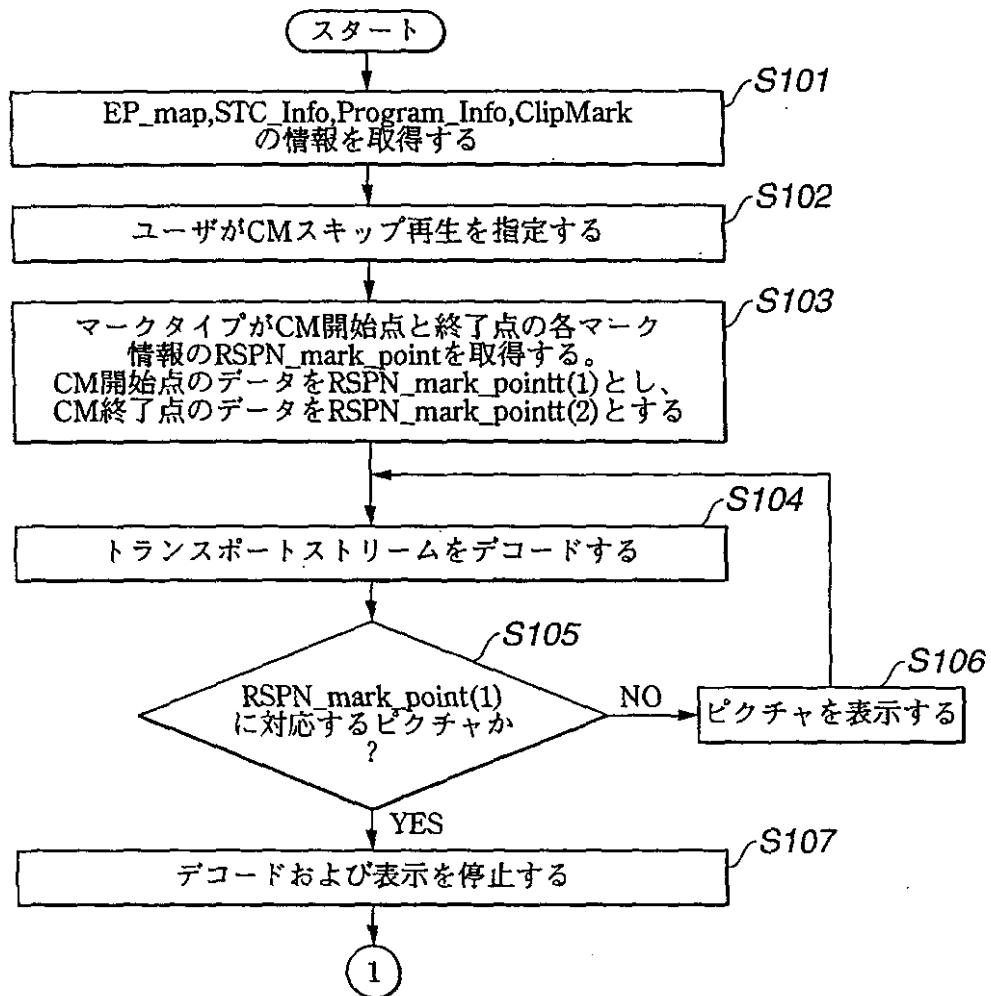


FIG.122

104/118

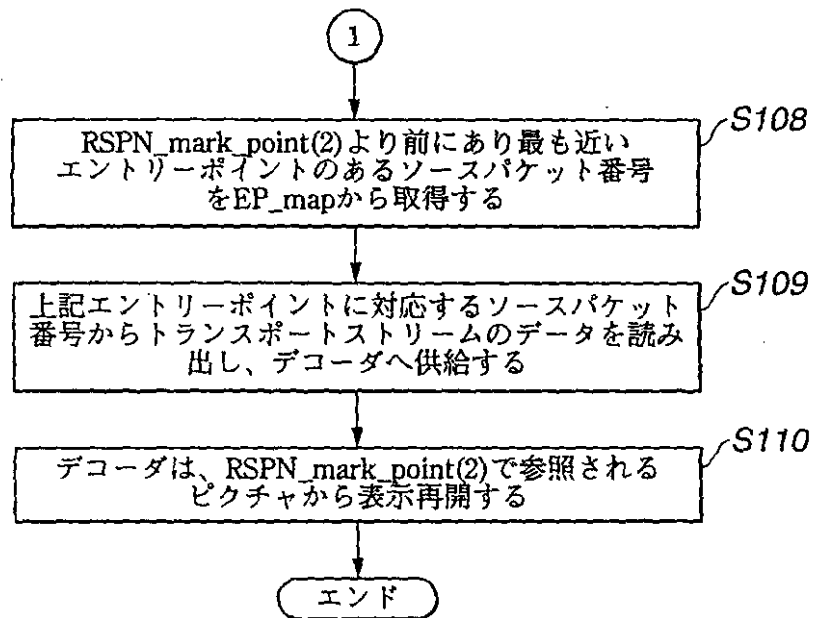


FIG.123

105/118

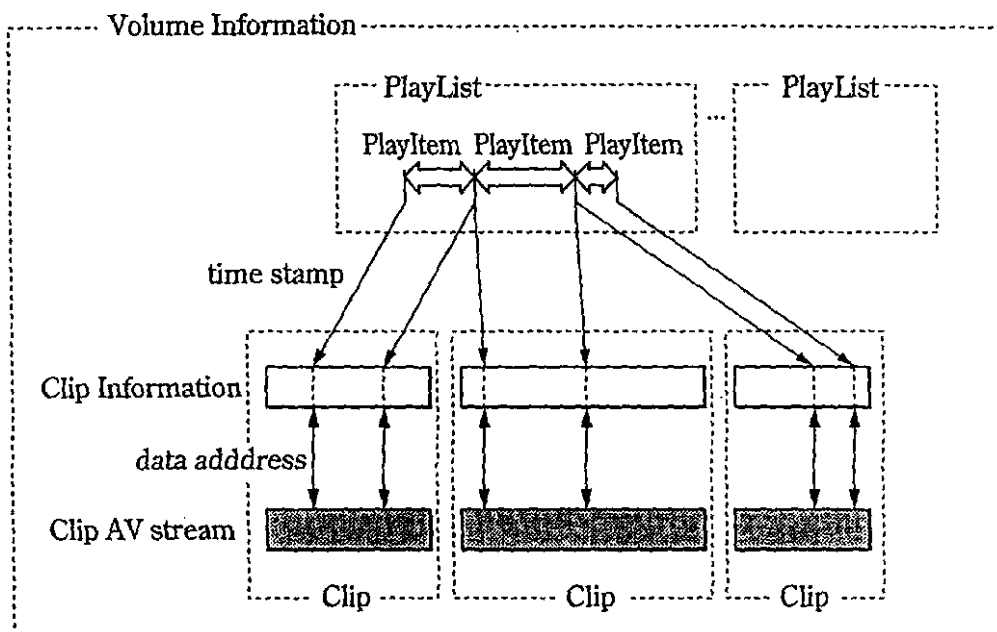


FIG.124

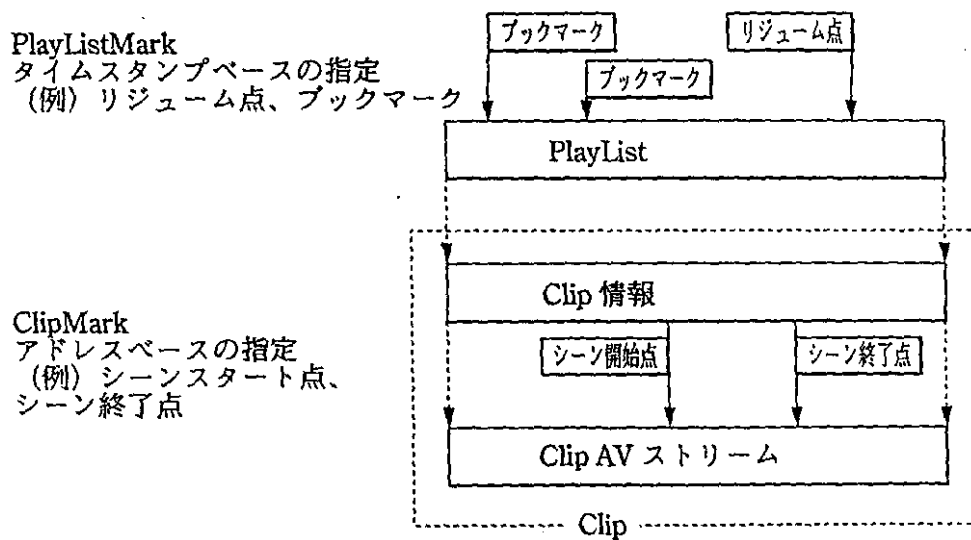


FIG.125

106/118

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_mark	32	uimsbf
reserved	32	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.126

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.127

107/118

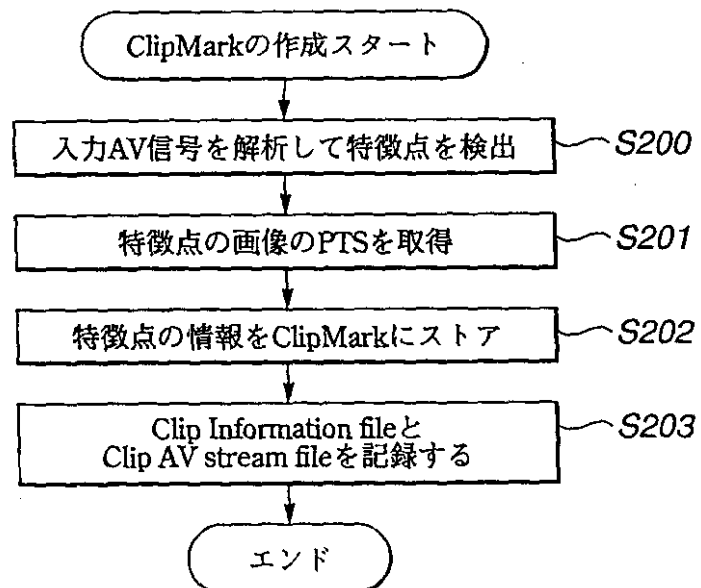


FIG.128

108/118

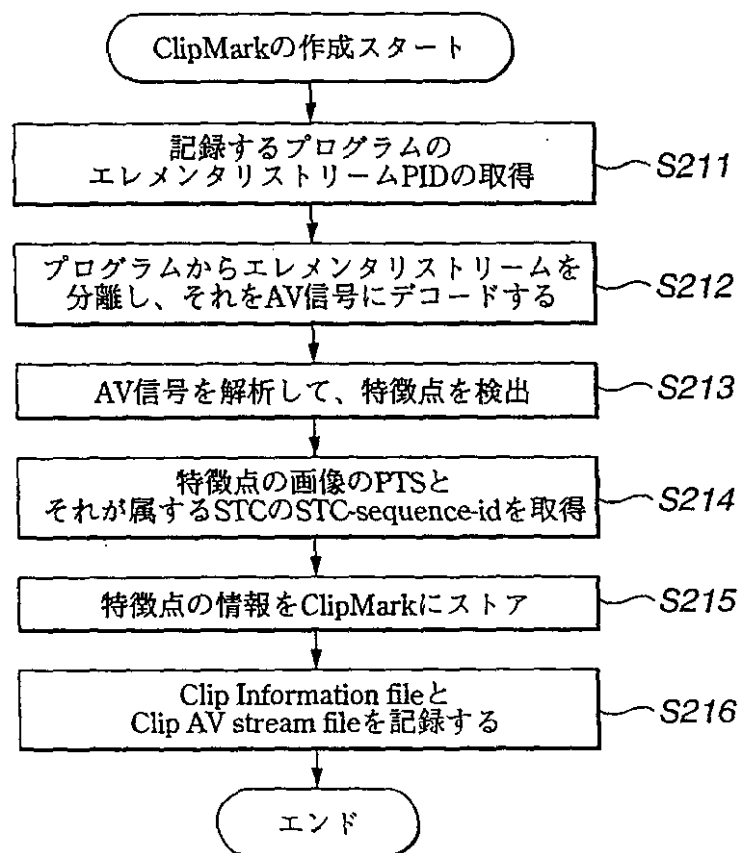


FIG.129

109/118

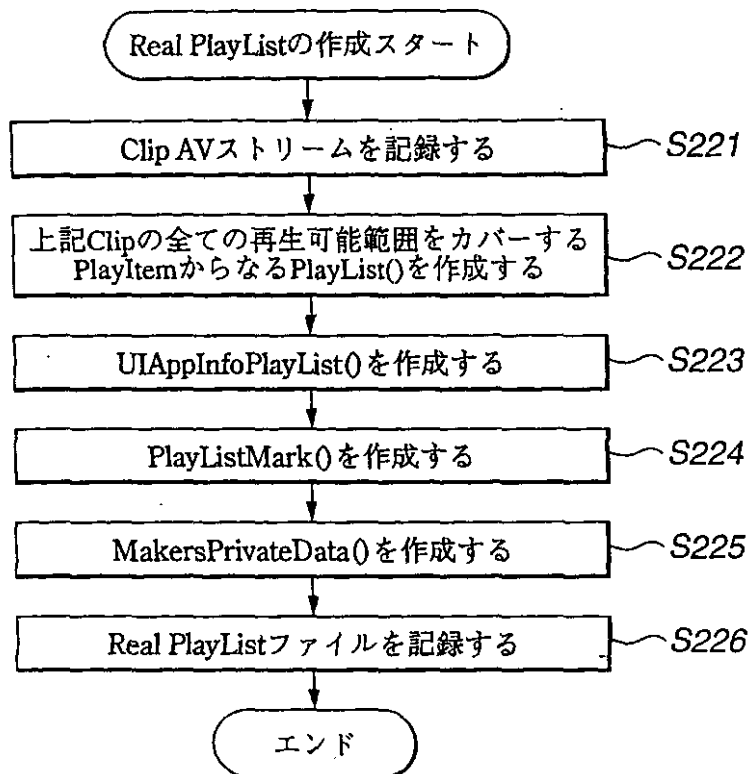


FIG.130

110/118

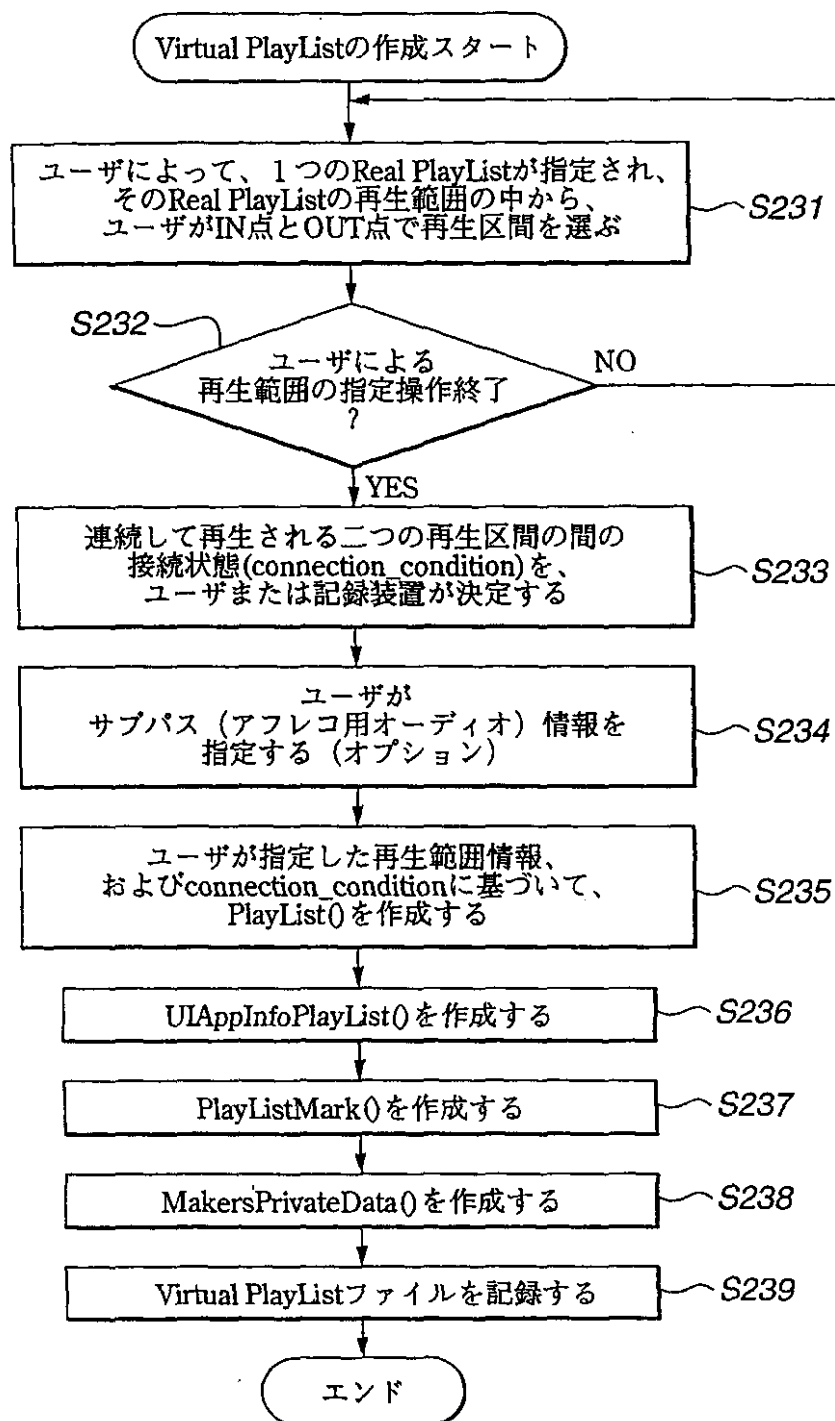


FIG.131

111/118

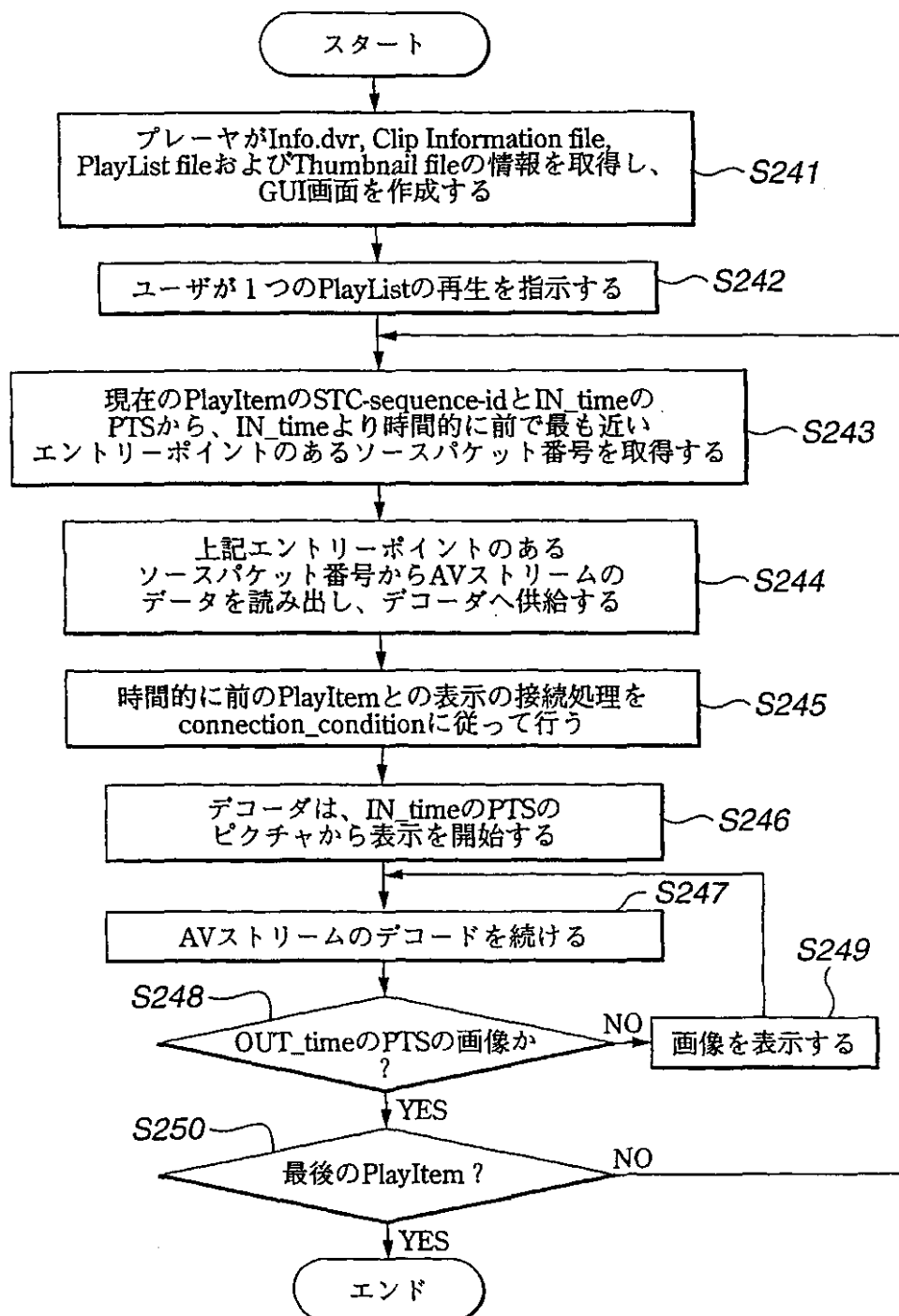


FIG.132

112/118

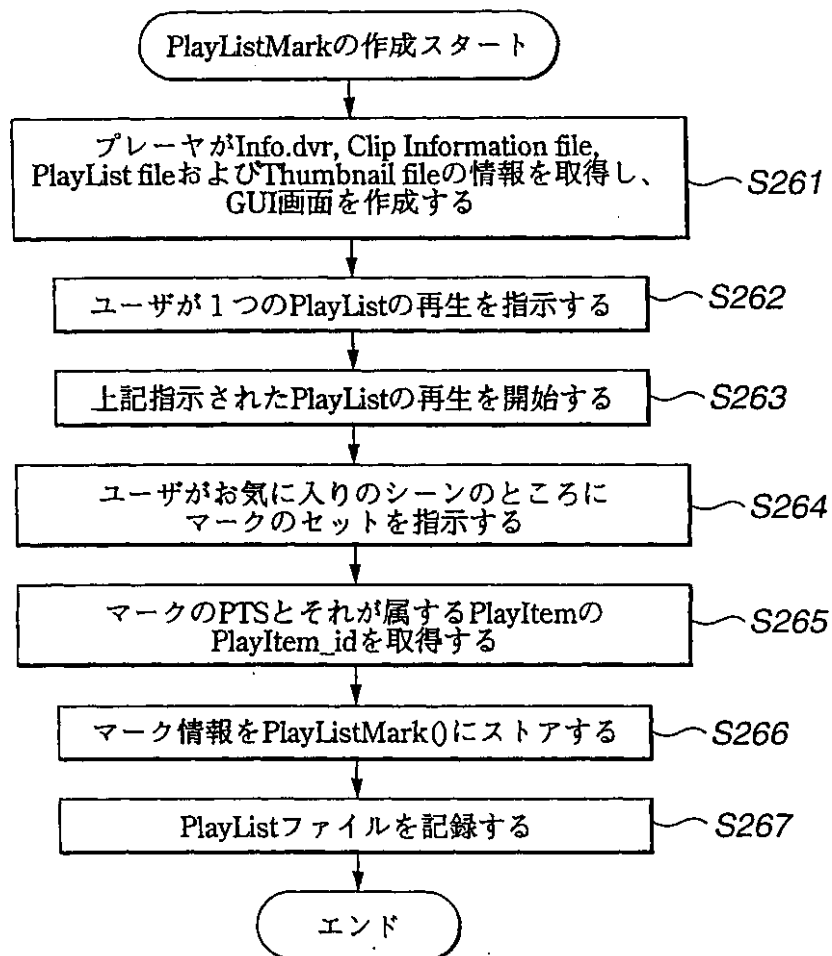


FIG.133

113/118

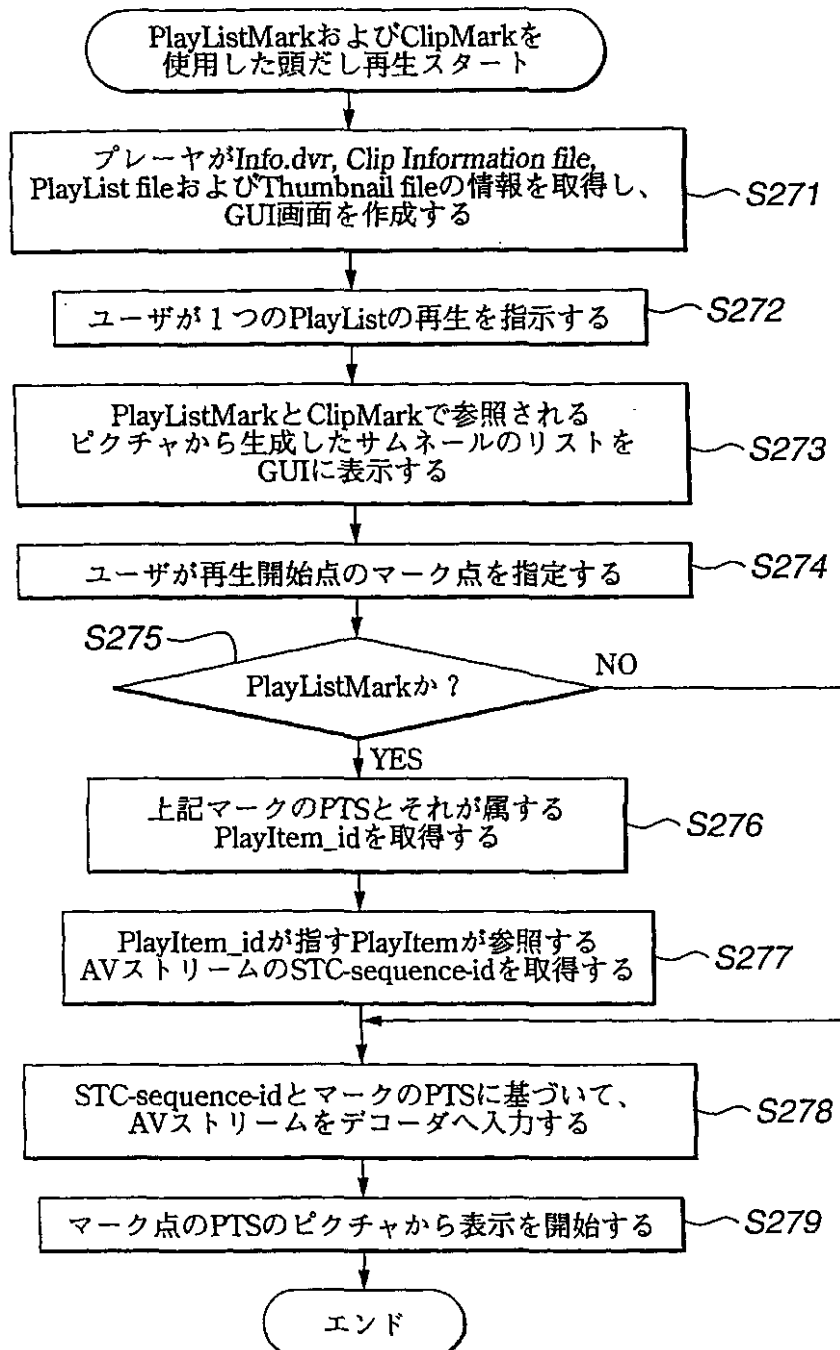


FIG.134

114/118

シンタクス	バイト数	略号
PlayListMark(){		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
mark_name	8*32	bslbf
}		
}		

FIG.135

115/118

値	意味	ノート
0x00	Resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	Book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。このマークは、PlayListMark()に複数あっても良い。
0x02	Chapter-mark	ユーザは、PlayListの中で1つのチャプターがこのマークから開始することを意図している。ユーザがセットすることができる。このマークは、PlayListMark()に複数あっても良い。
0x03	Skip-start-mark	PlayListMarkの中に1つのSkip-start-markがセットされる場合、そのSkip-start-markのエントリーの直後に1つのSkip-end-markがセットされなければならない。Skip-start-markのタイムスタンプからSkip-end-markのタイムスタンプまで、ユーザは、PlayListの再生をスキップすることを意図している。Skip-start-markとSkip-end-markは、同じref_to_PlayItem_idを持つ。また、Skip-start-markとSkip-end-markは、もしentry_ES_PIDが0xFFFFでないならば、同じentry_ES_PIDの値を持つ。ユーザがセットすることができるマークであり、このマークは、PlayListMark()に複数あっても良い。
0x04	Skip-end-mark	
0x05-0x3F	Reserved for future use	Reserved for PlayListMark
0x40-0x7F	Reserved for ClipMark	

FIG.136

116/118

シンタクス	バイト数	略号
ClipMark(){		
length	32	uimsbf
maker_ID	16	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
ref_to_STC_id	8	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
representative_picture_time_stamp	32	uimsbf
}		
}		

FIG.137

117/118

Mark_type	意味	ノート
0x00- 0x3F	reserved for future use	Reserved for PlayListMark
0x40	Scene-start-mark	シーンの開始ポイントを示すマーク点
0x41- 0x5F	Reserved for common ClipMark	
0x60- 0x7F	Maker defined ClipMark	maker_IDによって示されるメーカーが 自由に意味を定義できる

FIG.138

118/118

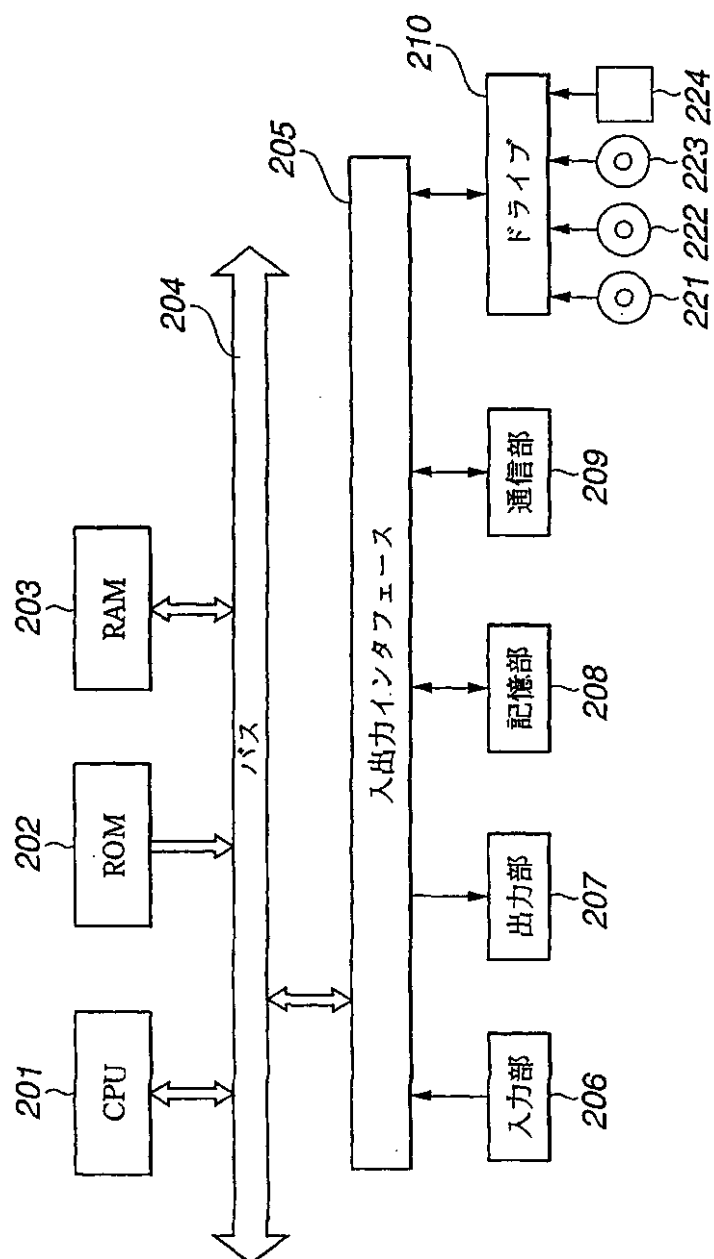


FIG.139

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/03414

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl.⁷ H04N 5/93, G11B 20/10

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 H04N 5/76-5/956, G11B 20/10-20/12

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

Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001
 Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001

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.
P, A	JP, 2000-341646, A (Sony Corporation), 08 December, 2000 (08.12.00), Full text; Figs. 1 to 10 (Family: none)	1-23
X	JP, 11-273227, A (NEC Software Kobe Ltd.), 08 October, 1999 (08.10.99), Full text; Figs. 1 to 3 (Family: none)	1-23
A	JP, 10-290432, A (Matsushita Electric Ind. Co., Ltd.), 27 October, 1998 (27.10.98), Full text; Figs. 1 to 10 (Family: none)	1-23

☐ 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 document 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
12 June, 2001 (12.06.01)Date of mailing of the international search report
19 June, 2001 (19.06.01)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

国際調査報告

国際出願番号 PCT/JPO1/03414

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int Cl⁷ H04N 5/93, G11B 20/10

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 Cl⁷ H04N 5/76-5/956, G11B 20/10-20/12

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	1971-2001年
日本国登録実用新案公報	1994-2001年
日本国実用新案登録公報	1996-2001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
P, A	JP, 2000-341646, A (ソニー株式会社) 08.12月.2000 (08.12.00) 全文、第1-10図 (ファミリーなし)	1-23
X	JP, 11-273227, A (神戸日本電気ソフトウェア株式会社) 08.10月.1999 (08.10.99) 全文、第1-3図 (ファミリーなし)	1-23
A	JP, 10-290432, A (松下電器産業株式会社) 27.10月.1998 (27.10.98) 全文、第1-10図 (ファミリーなし)	1-23

☐ 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

12.06.01

国際調査報告の発送日

19.06.01

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JPO)

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)

鈴木 明

5C 9185

電話番号 03-3581-1101 内線 3541

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04N 5/93

G11B 20/10

[12] 发明专利申请公开说明书

[21] 申请号 01801588.3

[43] 公开日 2002 年 12 月 4 日

[11] 公开号 CN 1383680A

[22] 申请日 2001.4.20 [21] 申请号 01801588.3

[30] 优先权

[32]2000.4.21 [33]JP [31]183770/00

[32]2000.9.5 [33]JP [31]268043/00

[86] 国际申请 PCT/JP01/03414 2001.4.20

[87] 国际公布 WO01/82608 日 2001.11.1

[85] 进入国家阶段日期 2002.2.4

[71] 申请人 索尼公司

地址 日本东京都

[72] 发明人 加藤元树 浜田俊也

[74] 专利代理机构 北京市柳沈律师事务所

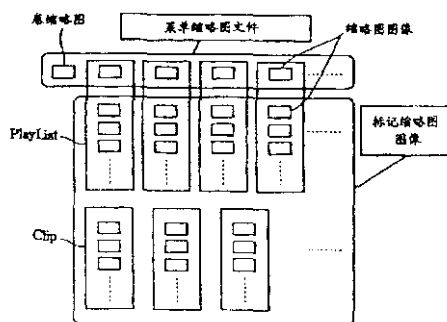
代理人 马莹 邵亚丽

权利要求书 4 页 说明书 67 页 附图 118 页

[54] 发明名称 信息处理设备和方法, 程序和记录介质

[57] 摘要

产生由指示从输入的 AV 流中提取的特征图象的标记组成的 ClipMark 作为管理 AV 流的管理信息。从对应于定义 AV 流中的预设域的组合的 PlayList 的再现部分中产生由指示用户任意指定的图象的标记组成的 PlayListMark。ClipMark 和 PlayListMark 被记录在记录介质上的相互独立的表中。因此能够快速和方便地对该 AV 流的所需部分进行访问。



ISSN 1008-4274

1. 一种信息处理设备,包括:

产生装置,用于产生通过指定从输入的 AV 流中提取的特征图象的标记
5 所形成的 ClipMark,作为监管 AV 流的管理信息,以及用于产生由指向通过
用户从对应于定义所述 AV 流中的预设域的组合的 PlayList 的再现域中可选指
定的图象的标记所形成的 PlayListMark; 和

记录装置,用于在记录介质上记录所述 ClipMark 和 PlayListMark 作为各
个独立的表。

10 2. 如权利要求 1 的信息处理设备,其中所述产生装置产生所述 ClipMark
和所述的 PlayListMark 分别作为 Clip 信息文件和 PlayList 文件。

3. 如权利要求 1 的信息处理设备,其特征在于:所述的 PlayListMark
还包括在再现所述 PlayList 中指定恢复点的标记。

4. 如权利要求 1 的信息处理设备,其中,在再现所述 PlayList 中,参考
15 形成对应于所述 PlayList 的再现域的所述 AV 流的 ClipMark 的所述标记。

5. 如权利要求 1 的信息处理设备,其中 PlayListMark 的所述标记包括呈
现时间戳和识别信息,用于指示在形成所述 PlayList 的再现路由的所述 AV 流
数据上指定的一个再现域。

6. 如权利要求 1 的信息处理设备,其中所述形成 ClipMark 的标记和所
20 述形成所述 PlayListMark 的标记包括指定基本流的入口点的信息。

7. 如权利要求 1 的信息处理设备,其中所述 PlayListMark 的标记包括至
少包含喜爱场景的开始点的恢复点或由用户指定的 PlayList 的恢复点的类型
的信息。

8. 如权利要求 1 的信息处理设备,其中所述 ClipMark 的标记和所述
25 PlayListMark 的标记是由对应于所述 AV 流的入口点的相关源数据包的地址表
现的。

9. 如权利要求 8 的信息处理设备,其中所述 ClipMark 的标记和所述
PlayListMark 的标记是由对应于所述 AV 流的入口点的相关源数据包的地址
和从所述第一地址偏移的第二地址分别表现的。

30 10. 如权利要求 1 的信息处理设备,还包括:

类型检测装置,用于在通过所述第一记录装置的记录时间上检测的所述

特征图象的类型;

所述第一记录装置记录形成所述 ClipMark 和由所述检测装置检测的所述类型的所述标记。

11. 如权利要求 1 的信息处理设备, 其中所述 ClipMark 的所述标记包括
5 场景改变点、商业广告开始点、商业广告结束点或标题显示场景。

12. 一种信息处理方法, 包括:

产生步骤, 用于产生通过指定从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark, 作为监管 AV 流的管理信息, 以及用于产生由指向通过用户从对应于定义所述 AV 流中的预设域的组合的 PlayList 的再现域中可选指
10 定的图象的标记所形成的 PlayListMark; 和

记录步骤, 用于在记录介质上记录所述 ClipMark 和 PlayListMark 作为各个独立的表。

13. 一种记录介质, 具有记录于其上的计算机可读程序, 所述计算机可读程序包括:

15 产生步骤, 用于产生通过指定从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark, 作为监管 AV 流的管理信息, 以及用于产生由指向通过用户从对应于定义所述 AV 流中的预设域的组合的 PlayList 的再现域中可选指定的图象的标记所形成的 PlayListMark; 和

记录步骤, 用于在记录介质上记录所述 ClipMark 和 PlayListMark 作为各
20 个独立的表。

14. 一种计算机可执行程序, 包括:

产生步骤, 用于产生通过指定从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark, 作为监管 AV 流的管理信息, 以及用于产生由指向通过用户从对应于定义所述 AV 流中的预设域的组合的 PlayList 的再现域中可选指
25 定的图象的标记所形成的 PlayListMark; 和

记录步骤, 用于在记录介质上记录所述 ClipMark 和 PlayListMark 作为各个独立的表。

15. 一种信息处理设备, 包括:

读出装置, 用于读取管理信息, 该管理信息用于监管包括通过指向从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark 的所述 AV 流, 并且还用于从定义所述 AV 流中的预设域的组合的 PlayList 所对应的再现域中读取
30

由指向用户可选指定的图象的标记所形成的 PlayListMark;

呈现装置, 用于呈现所述读出装置读取的所述管理信息和所述 PlayListMark 的信息;

参考装置, 用于从所述呈现装置呈现的所述信息中参考对应于用户命令

5 再现的 PlayList 的所述 ClipMark; 和

再现装置, 用于再现包括由所述参考装置从对应于所述 ClipMark 的位置中所参考的所述 ClipMark 的所述 AV 流。

16. 如权利要求 15 的信息处理设备, 其中所述呈现装置通过与所述 PlayListMark 相关的所述缩略图呈现一表。

10 17. 如权利要求 15 的信息处理设备, 其中通过对应于所述 AV 流的一个入口点的相关源数据包的地址表现对应于所述 ClipMark 的所述标记和对应于所述 PlayListMark 的所述标记。

18. 如权利要求 17 的信息处理设备, 其中通过对应于所述 AV 流的一个入口点的相关源数据包的地址和通过从所述第一地址偏移的第二地址分
15 别表现所述对应于所述 ClipMark 的所述标记和对应于所述 PlayListMark 的所述标记。

19. 如权利要求 15 的信息处理设备, 其中所述 ClipMark 的所述标记包括场景改变点、商业广告开始点、商业广告结束点或标题显示场景。

20. 一种信息处理方法, 包括:

20 读出步骤, 用于读取管理信息, 该管理信息用于监管包括通过指向从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark 的所述 AV 流, 并且还用于从定义所述 AV 流中的预设域的组合的 PlayList 所对应的再现域中读取由指向用户可选指定的图象的标记所形成的 PlayListMark;

呈现步骤, 用于呈现所述读出步骤读取的所述管理信息和所述
25 PlayListMark 的信息;

参考步骤, 用于从所述呈现步骤呈现的所述信息中参考对应于用户命令再现的 PlayList 的所述 ClipMark; 和

再现步骤, 用于从对应于所述 ClipMark 的位置再现包括由所述参考步骤中所参考的所述 ClipMark 的所述 AV 流。

30 21. 一种记录介质, 其上记录有计算机可读程序, 所述计算机可读程序包括:

读出步骤, 用于读取管理信息, 该管理信息用于监管包括通过指向从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark 的所述 AV 流, 并且还用于从定义所述 AV 流中的预设域的组合的 PlayList 所对应的再现域中读取由指向用户可选指定的图象的标记所形成的 PlayListMark;

- 5 呈现步骤, 用于呈现所述读出步骤读取的所述管理信息和所述 PlayListMark 的信息;

参考步骤, 用于从所述呈现步骤呈现的所述信息中参考对应于用户命令再现的 PlayList 的所述 ClipMark; 和

- 10 记录步骤, 用于将所述 ClipMark 和 PlayListMark 作为各个独立表记录在记录介质上。

22. 一种计算机可执行程序, 包括:

- 读出步骤, 用于读取管理信息, 该管理信息用于监管包括通过指向从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark 的所述 AV 流, 并且还用于从定义所述 AV 流中的预设域的组合的 PlayList 所对应的再现域中读取
15 由指向用户可选指定的图象的标记所形成的 PlayListMark;

呈现步骤, 用于呈现所述读出步骤读取的所述管理信息和所述 PlayListMark 的信息;

参考步骤, 用于从所述呈现步骤呈现的所述信息中参考对应于用户命令再现的 PlayList 的所述 ClipMark; 和

- 20 记录步骤, 用于将所述 ClipMark 和 PlayListMark 作为各个独立表记录在记录介质上。

23. 一种在其上已经记录管理信息作为独立表的程序, 该管理信息用于监管包括通过指向从输入的 AV 流中提取的特征图象的标记所形成的 ClipMark 的所述 AV 流, 并且还用于从定义所述 AV 流中的预设域的组合的
25 PlayList 所对应的再现域中读取由指向用户可选指定的图象的标记所形成的 PlayListMark。

信息处理设备和方法，程序和记录介质

5 技术领域

本发明涉及信息处理方法和设备，并涉及程序。尤其涉及信息处理方法和设备、记录介质和程序，其中，诸如 AV 流中的地址信息、编码参数、转换点信息和 I 图象标记的信息作为文件进行记录。

10 背景技术

近来已经提出了各种类型的光盘，作为能够从记录设备中移出的记录介质。这些可记录的光盘已经是作为几个 GB 的大容量介质被提出，并且认为有希望作为用于记录诸如视频信号的 AV(视听)信号的介质。在要记录在该可记录光盘上的数字 AV 信号源(供应源)中，有 CS 数字卫星广播和 BS 数字广播。另外，数字系统的地波电视广播也已经为今后使用而提出。

从这些源提供的数字视频信号在 MPEG2(移动图象专家组)系统下被例行地进行图象压缩。在记录设备中，设置了适合于该设备的记录速率。如果数字广播的数字视频信号记录在国内使用的常规图象存储介质中，数字视频信号被首先解码，并且随后进行带宽限制以用于记录。在当然包括 MPEG1 视频、MPEG2 视频和 DV 系统的数字记录系统的情况下，数字视频信号被首先解码，并且随后根据适合于随后记录的设备的记录速率的编码系统进行重新编码。

但是，其中所提供的位流在记录之前一次解码并随后进行带宽限制及重新编码的记录系统有变坏的图象质量。在记录图象压缩数字信号中，如果输入数字信号的传输速率小于记录和/或再现设备的记录速率，直接记录所提供的没有进行解码或者重新编码的位流的方法在图象质量上仅仅在很少程度上遭受破坏。但是，如果输入数字信号的传输速率超过记录和/或再现设备的记录速率，则重新编码位流和记录经过重新编码的位流确实是必须的，结果，在记录和/或再现设备中解码之后，传输速率将不高于盘记录速率的上限。

30 如果位流在其中输入数字信号的位速率随时间增加或者降低的可变速率系统中传输，则适于在缓冲器中暂时存储数据和突发方式记录该数据的盘记

录设备与具有由旋转头的固定 rpm(每分钟转速)强加的固定记录速率的带式记录系统相比, 其记录介质的容量浪费要少。

因此, 可以预计, 在不远的将来, 当数字广播变成主流时, 增加对记录和/或再现设备的需求, 其中广播信号记录为数字信号, DataStreamer 无需进行解码或重新编码, 并且其中盘用作记录介质。

如果具有其上记录了诸如由图象数据和声频数据构成的程序数据的复合数据的记录介质要通过上述设备再现的话, 则有如下问题: 在确定来自记录介质的 AV 流读出位置的处理或者需要响应来自用户的用于随机存取或者流解码的命令迅速完成解码该流的处理时, 随着记录介质上记录的数据量增加, 这种处理的迅速执行遇到了增加的困难。

发明公开

本发明的目的是克服现有技术中的不便性, 并且提供这种结构, 其中, 诸如地址信息、编码信息、转换点信息或者标记的 AV 流的 I 图象中的信息记录为文件, 以便能够迅速地确定和解码 AV 流中的读出位置和能够迅速地检索指定的标记。

根据本发明的信息处理设备包括: 用于产生其中 AV 流中编码信息是连续的域的开始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息的装置; 和用于将产生装置产生的信息记录为记录介质上的 Clip 信息(剪切信息)的装置。

其中 AV 流中编码信息是连续的域的开始地址信息是 STC 序列或者节目序列的开始地址, 有关时间信息和地址信息的信息是 EP_map 或者 TU_map, 特征图象的时间信息是 ClipMark。

记录装置还可以将与 AV 流记录速率的平均值有关的信息记录在记录介质上。

与平均值有关的信息可以是 TS_average_rate。

AV 流可以是传输流。

其中 AV 流中编码信息是连续的域的开始地址信息可以包括 STC 序列的开始地址, STC 序列是传输流的系统时钟为连续的域。

其中 AV 流中编码信息是连续的域的开始地址信息可以包括节目序列的开始地址, 节目序列是传输流的节目内容为不变的域。

其中 AV 流中编码信息是连续的域的起始地址信息可以包括域的起始地址, 在这里基于传输流传输数据包到达时间的到达时间是连续的。

有关时间信息和地址信息的信息可以包括传输流 I 图象的地址和其显示时间戳。

- 5 信息处理设备还可以包括压缩装置, 用于压缩有关 AV 流中时间信息和地址信息的信息的至少一部分。记录装置可以记录由压缩装置压缩的信息。

有关 AV 流中时间信息和地址信息的信息可以包括基于传输数据包的到达时间的到达时间和相关的传输数据包的 AV 流数据中的地址。

- 10 根据本发明的信息处理方法包括用于产生其中 AV 流中编码信息是连续的域的起始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息的步骤; 和用于将产生步骤产生的信息记录为记录介质上的 Clip 信息的步骤。

- 15 根据本发明记录介质的程序包括用于产生其中 AV 流中编码信息是连续的域的起始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息的步骤; 和用于将产生步骤产生的信息记录为记录介质上的 Clip 信息的步骤。

- 20 根据本发明的程序执行用于产生其中 AV 流中编码信息是连续的域的起始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息的步骤; 和用于将产生步骤产生的信息记录为记录介质上的 Clip 信息的步骤。

根据本发明的信息处理设备包括用于再现其中 AV 流中编码信息是连续的域的起始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息作为 Clip 信息的装置; 和用于基于再现装置再现的 Clip 信息来控制 AV 流的输出。

- 25 根据本发明的信息处理方法包括用于再现其中 AV 流中编码信息是连续的域的起始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息的步骤; 和用于基于再现装置再现的 Clip 信息来控制 AV 流的输出的步骤。

- 30 根据本发明的程序包括用于再现其中 AV 流中编码信息是连续的域的起始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息的步骤; 以及用于基于再现装置再现的 Clip 信息来控制 AV 流

的输出的步骤。

- 根据本发明的程序执行用于再现其中 AV 流中编码信息是连续的域的
开始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图
象的时间信息的步骤；以及用于基于再现装置再现的 Clip 信息来控制 AV 流
5 的输出的步骤。

根据本发明的记录介质在其上已经记录了 AV 流中编码信息是连续的域的
的开始地址信息、有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特
征图象的时间信息，将其作为 Clip 信息。

- 根据本发明，记录 AV 流中编码信息是连续的域的的开始地址信息、有关
10 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息。

而且，根据本发明，再现 AV 流中编码信息是连续的域的的开始地址信息、
有关 AV 流中时间信息和地址信息的信息、以及 AV 流中特征图象的时间信息。

附图简要说明

- 15 图 1 表示根据本发明记录和/或再现设备的实施例的结构。
图 2 表示在记录介质上通过记录和/或再现设备 1 记录的数据的数据格
式。
图 3 表示实 PlayList 和虚 PlayList。
图 4A, 4B 和 4C 表示实 PlayList 的创建。
20 图 5A, 5B 和 5C 表示实 PlayList 的删除。
图 6A 和 6B 表示汇编编辑。
图 7 表示虚 PlayList 中提供的子路径。
图 8 表示 PlayList 重放序列的改变。
图 9 表示 PlayList 上的标记和 Clip 上的标记(mark)。
25 图 10 表示菜单缩略图(menu thumbnail)。
图 11 表示加到 PlayList 的标记。
图 12 表示加到 Clip 的标记。
图 13 表示在 PlayList、Clip 和缩略图文件之间的关系。
图 14 表示目录结构。
30 图 15 表示 infr.dvr 的句法。
图 16 表示 DVRVolume 的句法。

- 图 17 表示 ResumeVolume 的句法。
- 图 18 表示 UIAppInfoVolume 的句法。
- 图 19 表示字符集值表。
- 图 20 表示 TableOfPlayList 的句法。
- 5 图 21 表示 TableOfPlayList 的另一个句法。
- 图 22 表示 MakersPrivateData 的句法。
- 图 23 表示 xxxx.rpls 和 yyyy.vpls 的句法。
- 图 24A 到 24C 表示 PlayList。
- 图 25 表示 PlayList 的句法。
- 10 图 26 表示 PlayList_type 表。
- 图 27 表示 UIAppInfoPlayList 的句法。
- 图 28A 到 28C 表示图 27 所示的 UIAppInfoPlayList 句法中的标志(flag)。
- 图 29 表示 PlayItem。
- 图 30 表示 PlayItem。
- 15 图 31 表示 PlayItem。
- 图 32 表示 PlayItem 的句法。
- 图 33 表示 IN_time。
- 图 34 表示 OUT_time。
- 图 35 表示 Connection_Condition 表。
- 20 图 36A 到 36D 表示 Connection_Condition。
- 图 37 表示 BridgeSequenceInfo。
- 图 38 表示 BridgeSequenceInfo 的句法。
- 图 39 表示 SubPlayItem。
- 图 40 表示 SubPlayItem 的句法。
- 25 图 41 表示 Mark_type 表。
- 图 42 表示 PlayListMark 的句法。
- 图 43 表示 Mark_type 表。
- 图 44 表示 Mark_time_stamp。
- 图 45 表示 zzzzz.clip 的句法。
- 30 图 46 表示 ClipInfo 的句法。
- 图 47 表示 Clip_stream_type 的表。

- 图 48 表示 offset_SPN。
图 49 表示 offset_SPN。
图 50A、50B 表示 STC 域。
图 51 表示 STC_Info。
5 图 52 表示 STC_Info 的句法。
图 53 表示 ProgramInfo。
图 54 表示 ProgramInfo 的句法。
图 55 表示 VideoCondInfo 的句法。
图 56 表示 Video_format 的表。
10 图 57 表示 frame_rate 的表。
图 58 表示 display_aspect_ratio 的表。
图 59 表示 AudioCondInfo 的句法。
图 60 表示 audio_coding 的表。
图 61 表示 audio_component_type 的表。
15 图 62 表示 sampling_frequency 的表。
图 63 表示 CPI。
图 64 表示 CPI。
图 65 表示 CPI 的句法。
图 66 表示 CPI_type 的表。
20 图 67 表示视频 EP_map。
图 68 表示 EP_map。
图 69 表示 EP_map。
图 70 表示 EP_map 的句法。
图 71 表示 EP_typevalues 的表。
25 图 72 表示 EP_map_for_one_stream_PID 的句法。
图 73 表示 TU_map。
图 74 表示 TU_map 的句法。
图 75 表示 ClipMark 的句法。
图 76 表示 Mark_type 的表。
30 图 77 表示 Mark_type_stamp 的表。
图 78 表示 ClipMark 的另一个句法。

- 图 79 表示 Mark_type 表的另一个例子。
- 图 80 表示 mark_entry()和 representative_picture_entry()的例子。
- 图 81 表示 mark_entry()和 representative_picture_entry()的句法。
- 图 82 表示 mark_entry()和 representative_picture_entry()的另一个例子。
- 5 图 83 表示 RSPN_ref_EP_start 和 offset_num_pictures 之间的关系。
- 图 84 表示 mark_entry()和 representative_picture_entry()的句法的另一个例子。
- 图 85 表示 ClipMark 和 EP_map 之间的关系。
- 图 86 表示 menu.thmb 和 mark.thmb 的句法。
- 10 图 87 表示 Thumbnail 的句法。
- 图 88 表示 thumbnail_picture_format 的表。
- 图 89A 和 89B 表示 tn_block。
- 图 90 表示 DVMPEG2 传输流的结构。
- 图 91 表示 DVMPEG2 传输流的记录器模式。
- 15 图 92 表示 DVMPEG2 传输流的播放器模式。
- 图 93 表示源数据包的句法。
- 图 94 表示 TP_extra_header 的句法。
- 图 95 表示拷贝允许指示符的表。
- 图 96 表示无缝连接。
- 20 图 97 表示无缝连接。
- 图 98 表示无缝连接。
- 图 99 表示无缝连接。
- 图 100 表示无缝连接。
- 图 101 表示音频重叠。
- 25 图 102 表示采用 BridgeSequence 的无缝连接。
- 图 103 表示不采用 BridgeSequence 的无缝连接。
- 图 104 表示 DVRSTD 模式。
- 图 105 表示用于解码和显示的时序图。
- 图 106 是用于说明在图 81 句法的情况下由标记点指示的场景索引再现的
- 30 流程图。
- 图 107 表示当图 75 或者图 78 中 ClipMark 的

mark_entry()/representative_picture_entry()是图 81 的句法时再现的操作。

图 108 表示 EP_mark 的例子。

图 109 表示 ClipMark 的例子。

图 110 是用于说明当图 75 或者图 78 中 ClipMark 的

- 5 mark_entry()/representative_picture_entry()是图 81 的句法时 CM 跳步再现处理的流程图。

图 111, 续图 110, 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 81 的句法时 CM 跳步(skip)再现处理的流程图。

- 10 图 112 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 82 的句法时由标记点指示的场景索引再现的流程图。

图 113, 续图 112, 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 82 的句法时再现的流程图。

- 15 图 114 表示 EP_map 的例子。

图 115 表示 ClipMark 的例子。

图 116 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 82 的句法时 CM 跳步再现处理的流程图。

- 20 图 117, 续图 116, 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 82 的句法时 CM 跳步再现处理的流程图。

图 118 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 84 的句法时由标记点指示的场景索引再现的流程图。

- 25 图 119, 续图 118, 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 84 的句法时再现的流程图。

图 120 表示 EP_map 的例子。

图 121 表示 ClipMark 的例子。

- 30 图 122 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 84 的句法时 CM 跳步再现处理

的流程图。

图 123, 续图 122, 是用于说明当图 75 或者图 78 中 ClipMark 的 mark_entry()/representative_picture_entry()是图 84 的句法时 CM 跳步再现处理的流程图。

5 图 124 表示应用格式。

图 125 表示 PLAYMARK 上的标记和 Clip 上的标记。

图 126 表示 ClipMark 句法的另一个例子。

图 127 表示 ClipMark 句法的又一个例子。

10 图 128 是用于说明当模拟 AV 信号被编码和记录的情况下 ClipMark 的准备的流程图。

图 129 是用于说明在传输流将被记录的情况下 ClipMark 的准备的流程图。

图 130 是用于说明实 PlayList 的准备的流程图。

图 131 是用于说明虚 PlayList 的准备的流程图。

15 图 132 是用于说明 PlayList 的再现的流程图。

图 133 是用于说明 PlayListMark 的准备的流程图。

图 134 是用于说明再现 PlayList 的定位再现的流程图。

图 135 表示 PlayListMark 的句法。

图 136 说明 PlayListMark 的 Mark_Type。

20 图 137 表示 ClipMark 的另一个句法。

图 138 说明 ClipMark 的 Mark_type。

图 139 说明一种介质。

实现本发明的最佳模式

25 引用附图, 现在详细地说明本发明的实施例。图 1 表示体现本发明的记录和/或再现设备 1 的典型内部结构。首先, 说明构造为记录从外部输入的信号的记录单元 2 的结构。记录和/或再现设备 1 构造为用模拟或者数字数据供给和记录模拟或数字数据。

30 模拟视频信号和模拟音频信号分别提供给端子 11, 12。输入到端子 11 的视频信号输出到分析单元 14 和 AV 编码器 15。输入到端子 12 的音频信号输出到分析单元 14 和 AV 编码器 15。分析单元 14 从输入的视频和音频信号

中提取诸如场景变化的特征点。

AV 编码器 15 编码输入的视频和音频信号以将诸如编码视频流(V)、编码音频流(A)和 AV 同步的系统信息输出给多路复用器 16。

5 编码视频流是例如用 MPEG(移动图象专家组)2 系统编码的视频流,而编码音频流是根据 MPEG1 系统编码的音频流,编码音频流是在例如 MPEG1 系统中编码的音频流或者根据杜比 AC3(商标)系统编码的音频流。多路复用器 16 基于输入系统信息多路复用输入视频流和音频流以通过开关 17 将多路复用流输出到多路复用的流分析单元 18 和源打包器(source packetizer)19。

10 多路复用流例如是 MPEG-2 传输流或者 MPEG-2 节目流。根据在其上记录该流的记录介质 100 的应用格式,源打包器 19 将输入的多路复用流编码成由源数据包构成的 AV 流。在输出到写单元 22 之前,AV 流在 ECC(错误校正和编码)单元 20 和调制单元 22 中用附带的 ECC 代码和调制进行处理,其然后基于由控制器 23 输出的控制信号写(记录)AV 流。

15 从数字接口或者数字电视调谐器输入的诸如数字电视广播的传输流输入到端子 13。有两个用于记录输入到端子 13 的传输流的记录系统:一个是透明记录系统,另一个是记录之前是重新编码的系统,重新编码的目的在于降低例如记录位速率。记录系统命令信息从作为用户接口的端子 24 输入到控制器 23。

20 在输入的传输流的透明记录中,输入到端子 13 的传输流通过开关 17 输出到多路复用流分析单元 18 和源打包器 19。如上述,在记录介质上记录 AV 流的随后处理与编码和记录模拟输入的音频和视频信号的随后处理相同,因此,为了简单起见在此不进行说明。

25 如果输入的传输流被重新编码和随后记录,则输入到端子 13 的传输流馈送到多路分用器 26,它多路分用输入的传输流以提取视频流(V)、音频流(A)和系统信息(S)。

在通过多路分用器 26 提取的流(信息)中,视频流输出到音频解码器 27,而音频流和系统信息输出到多路复用器 16。音频解码器 27 解码输入的传输流以将编码的视频流(V)输出到多路复用器 16。

30 从多路分用器 26 输出的且输入到多路复用器 16 的音频流和系统信息以及由 AV 编码器 15 输出的视频流基于输入系统信息被多路复用,并通过开关 17 作为多路复用流输出到多路复用流分析单元 18 和源打包器 19。如上述,

在记录介质上记录 AV 流的随后处理是与编码和记录模拟输入的音频和视频信号的随后处理相同, 因此, 为了简单起见在此不进行说明。

本实施例的记录和/或再现设备 1 在记录介质 100 上记录 AV 流文件, 同时也记录解释该文件的应用数据库信息。对控制器 23 的输入信息是来自分析单元 14 的移动图象的特征信息, 来自多路复用流分析单元 18 的 AV 流的特征信息和从端子 24 输入的用户命令信息。

在 AV 编码器 15 编码视频信号时, 从分析单元 14 提供的移动图象的特征信息是由分析单元 14 产生的。分析单元 14 分析输入视频和音频信号的内容以产生与输入的移动图象信号的图象特征(Clip 标记)有关的信息。该信息是表示诸如节目开始点、场景变化点、CM 商业广告开始和结束点、输入视频信号中的标题或者幻灯机(telop)的特征 Clip 标记点的图象的信息, 并且还包括与音频信号的立体音/非立体音转换点和消音部分有关的图象和信息的缩略图(thumbnail)。

上述图象表示信息通过控制 23 馈送到多路复用器 16。当多路复用由控制器 23 指定为 Clip 标记的编码图象时, 多路复用器 16 将用于指定 AV 流上编码图象的信息返回到控制器 23。具体地, 该信息是图象的 PTS(显示时间戳)或者是图象编码版本的 AV 流上的地址信息。控制器 23 存储特征图象的分类和用于指定 AV 流上相互关联的编码图象的信息。

来自多路复用流分析单元 18 的 AV 流的特征信息是与要记录的 AV 流编码信息相关的信息, 并且通过分析单元 18 记录。例如, 特征信息包括 AV 流中 I 图象的时间戳(time stamp)和地址信息、系统时钟的非连续点信息、AV 流的编码参数和 AV 流中编码参数的改变点信息。当透明地记录从端子 13 输入的传输流时, 多路复用流分析单元 18 从输入传输流中检测前述的 Clip 标记的图象, 并且产生用于指定由 Clip 标记和其类型指定的图象的信息。

来自端子 24 的用户指配信息是指定由用户指定的重放域的信息、用于解释重放域内容的字符字母、或者诸如书签或者由用户为他或她喜爱场景而设定的重新开始点的信息。

基于前述输入信息, 控制器 23 创建 AV 流数据库(Clip), AV 流重放域(PlayItem)的组(PlayList)的数据库, 记录介质 100 记录内容的管理信息(info.dvr)和关于缩略图图象的信息。类似于 AV 流, 从上述信息构成的应用数据库信息在 ECC 单元 20 和调制单元 21 中进行处理且输入到写单元 22, 其然后将数

数据库文件记录在记录介质 100 上。

随后将详细说明上述的应用数据库信息。

当在记录介质 100 上记录的 AV 流文件(图象数据和语音数据文件)和因此在记录介质 100 上记录的应用数据库信息通过再现单元 3 再现时, 控制器 23
5 首先命令读出单元 28 从记录介质 100 中读出应用数据库信息。读出单元 28 从记录介质 100 中读出应用数据库信息, 然后从记录介质 100 中读出应用数据库信息以通过由解调单元 29 和 ECC 解码器 30 进行的解调和错误校正处理来将该应用数据库信息发送到控制器 23。

基于应用数据库信息, 控制器 23 将在记录介质 100 上记录的 PlayList
10 表输出到端子 24 的用户接口。用户从 PlayList 表中选择希望再现的 PlayList。指定为要再现的与 PlayList 相关的信息输入到控制器 23。控制器 23 命令读出单元 28 读出在再现 PlayList 中必需的 AV 流文件。根据该命令, 读出单元 28 从记录介质 100 中读出对应的 AV 流以将所读出的 AV 流输出到解调单元 29。因此, 输入到解调单元 29 的 AV 流通过预置处理被解调和通过 ECC 解码器
15 30 的处理输出到源解数据包器(depacketizer)31。

源解数据包器 31 将从记录介质 100 读出的并以预置方式处理的应用格式的 AV 流变换成可由多路分用器 26 处理的流。多路分用器 26 将形成由控制器 23 指定的 AV 流的重放域(PlayItem)的诸如视频流(V)、音频流(A)或者 AV 同步的系统信息(S)输出到音频解码器 27, 该 AV 解码器 27 解码视频流和音频流以将重放视频信号和重放音频信号分别输出到相关的端子 32、33。
20

如果从作为用户接口的端子 24 提供指令进行随机存储重放或者指定重放的信息, 则控制器 23 基于 AV 流数据库(Clip)内容确定来自记录介质 100 的 AV 流的读出位置, 以命令读出单元 28 读出 AV 流。如果作为用户选择的 PlayList 将作为从预置时间点进行再现, 则控制器 23 命令读出单元 28 从具有
25 最靠近指定的时间点的时间戳的 I 图象读出数据。

当用户从存储在 Clip 信息的 ClipMark 中的节目的索引点或者场景变化点已经选择了某一 Clip 标记, 即正如作为用户接口所显示的, 当用户从存储在 ClipMark 中的索引点或者场景变化点的缩略图图象表中选择了某一图象时, 则控制器 23 确定来自记录介质 100 之 AV 流读出位置, 以命令读出单元
30 28 读出 AV 流。即, 控制器 23 命令读出单元 28 从具有最靠近已经存储用户选择的图象的 AV 流地址的地址的 I 图象中读出数据。读出单元 28 从指定地

址读出数据。读出的数据通过解调单元 29、ECC 解码器 30 和通过源打包器 19 处理以便提供到多路分用器 26, 并且通过音频解码器 27 解码以再现由标记点图象的地址表示的 AV 数据。

- 如果用户已经命令快进(fast forward)重放, 则控制器 23 命令读出单元 28
- 5 基于 AV 流数据库(Clip)依次连续地读出 AV 流中的 I 图象数据。

读出单元 28 从指定的随机存取点中读出 AV 流的数据。如此读出的数据通过由下游侧的各种元件的处理再现。

- 现在说明其中用户编辑记录在记录介质 100 上的 AV 流的情况。如果希望指定的记录在记录介质 100 上的 AV 流的重放域, 例如, 如果希望创建从
- 10 歌曲节目 A 中再现由歌唱者 A 演唱部分并且随后从另一个歌曲节目 B 中再现由同一歌唱者 A 演唱部分的重放例程, 则与重放域的 IN_point(开始点)和 OUT_point(结束点)相联系的信息从作为用户接口的端子输入到控制器 23。控制器 23 创建 AV 流重放域(PlayItem)的组的数据库(Playlist)。

- 当用户希望擦除记录在记录介质 100 上的一部分 AV 流时, 与擦除域的
- 15 IN_point 和 OUT_point 相关的信息输入到控制器 23, 其然后改进 Playlist 数据库以便仅仅参照需要的 AV 流。控制器 23 还命令写单元 22 擦除 AV 流的不需要的流部分。

- 现在说明这种情况, 其中用户希望指定记录在记录介质上的 AV 流的重放域以创建新的重放例程(route)和以无缝方式互连对应的重放域。在这种情况下, 控制器 23 创建 AV 流重放域(PlayItem)的组的数据库(Playlist)并进行部
- 20 分地重新编码与重新多路复用在重放域连接点附近的视频流。

- 在重放域 IN_point 的图象信息和 OUT_point 的图象信息从端子 24 输入到控制器 23。控制器 23 命令读出单元 28 读出所需要的在 IN_point 和在 OUT_point 上再现的图象数据。读出单元 28 从记录介质 100 上读出数据。如
- 25 此读出的数据通过解调单元 29、ECC 解码器 30 和源打包器 19 输出到多路分用器 26。

控制器 23 分析输入到多路分用器 26 的数据以确定对视频流的重新编码方法(picture_coding_type 的变化和重新编码之编码位数量的分配)和重新多路复用系统, 以将该系统送到 AV 编码器 15 和多路复用器 16。

- 30 多路分用器 26 然后将输入的流分离成视频流(V)、音频流(A)和系统信息(S)。视频流可以分类成输入到音频解码器 27 的数据和输入到多路复用器 16

的数据。前者是重新编码所需要的数据，并且通过音频解码器 27 解码，其中具有然后由 AV 编码器 15 重新编码和由此变成视频流的解码图象。后者数据是从没有重新编码的原始流中拷贝的数据。音频流和系统信息直接输入到多路复用器 16。

- 5 多路复用器 16 基于从控制器 23 输入的信息多路复用输入流以输出该多路复用流，其被 ECC 单元 20 和调制单元 21 处理以便发送到写单元 22。写单元 22 基于从控制器 23 提供的控制信号将 AV 流记录在记录介质 100 上。

- 下面解释应用数据库信息和基于该信息的诸如重放和编辑的操作。图 2 表示用于 AV 流管理的应用格式结构，该结构具有两层，即 PlayList 和 Clip。
10 卷(volume)信息管理盘中的所有 Clips 和 PlayList。这里，成对的一个 AV 流和其辅助信息认为是一个对象，并叫作 Clip。AV 流文件叫作 Clip AV 流文件，具有叫作 Clip 信息文件的辅助信息。

- 一个 Clip AV 流文件存储对应于配置成由应用格式指定的结构的 MPEG 2 传输流的数据。一般说来，文件作为字节串进行处理。Clip AV 流文件的内容在时间轴上扩展，其中具有主要以时间基指定的 Clip(I 图象)入口点。当给
15 定对预置 Clip 的存取点的时间戳时，Clip 信息文件在找到开始在 Clip AV 流文件中读出的数据的地址信息方面是有用的。

- 参照图 3，现在解释 PlayList，其是为用户从 Clip 中选择希望观看的重放域和方便地编辑重放域而提供的。一个 PlayList 是 Clip 中的一组重放域。预
20 置 Clip 中的一个重放域叫作 PlayItem 并且由时间轴上的一对 IN_point 和 OUT_point 表示。这样，PlayList 是由一组复合 PlayItem 形成的。

- PlayList 分类成两个类型，一种是实 PlayList，另一种是虚 PlayList。实 PlayList 共同拥有其引用的 Clip 流部分。即，实 PlayList 在盘上占据对应于其引用的 Clip 流部分之数据容量，并且当擦除实 PlayList 时，其引用的 Clip 流
25 部分的数据也被擦除。

 虚 PlayList 不是共同拥有 Clip 数据。因此，如果虚 PlayList 改变或者被删除，而 Clip 的内容不会改变。

 解释实 PlayList 的编辑。图 4A 表示实 PlayList 的创建，并且如果 AV 流作为新的 Clip 记录，则引用整个 Clip 的实 PlayList 是新的创建操作。

- 30 图 4B 表示实 PlayList 的划分，即在希望点上划分实 PlayList 的操作以将实 PlayList 分成两个实 PlayList。当两个节目在通过单个 PlayList 管理的一个

Clip 管理时, 并且当用户趋于重新寄存或者重新记录该节目作为单个独立节目时, 进行该划分操作。该操作不会导致 Clip 内容的改变, 即对 Clip 本身的划分。

- 图 4C 表示将两个实 PlayList 组合成一个新的实 PlayList 之操作的实 PlayList 的组合操作。该组合操作是当诸如用户希望重新寄存两个节目为单个程序时完成的。该操作不会导致 Clip 内容的改变, 即是将 Clip 本身组合为一个。

图 5A 表示整个实 PlayList 的删除。如果删除整个预置的实 PlayList 的操作, 则由所删除的实 PlayList 引用的 Clip 的相关流部分也被删除。

- 10 图 5B 表示实 PlayList 的部分删除。如果删除实 PlayList 的期望部分, 则相关 PlayItem 改变为仅仅引用所需要的 Clip 流部分。对应的 Clip 流部分被删除。

- 15 图 5C 表示实 PlayList 的最小化。其是使与实 PlayList 相关的 PlayItem 仅仅引用虚 PlayList 需要的 Clip 流部分的操作。对应的虚 PlayList 不需要的 Clip 流部分被删除。

如果实 PlayList 通过上述操作改变使得由实 PlayList 引用的 Clip 流部分被删除, 则有这种可能性, 即出现采用所删除的 Clip 的虚 PlayList, 使得在虚 PlayList 中可以产生因所删除的 Clip 导致的问题。

- 20 为了防止这种情况发生, 则向用户显示这种消息: “如果存在引用实 PlayList 正在引用的 Clip 流部分的虚 PlayList, 且实 PlayList 被删除, 则虚 PlayList 本身被删除——可以否?”, 通过证实或者警告响应用户的删除操作, 此后, 执行删除处理, 或者取消用户命令的对象。或者, 完成实 PlayList 的最小化操作以代替删除虚 PlayList。

- 25 现在解释虚 PlayList 的操作。如果操作是对虚 PlayList 进行的, 则 Clip 的内容不改变。图 6A 和 6B 表示汇编和编辑(IN-OUT 编辑)。这是创建用户希望观看的重放域的 PlayItem 以创建虚 PlayList 的操作。在 PlayItem 之间的无缝连接是由应用格式支持的, 如后述。

- 30 如果存在两个实 PlayList1、2 和与对应实 PlayList 相联系的 Clip1、2, 则用户指定实 PlayList1 中的预置域(从 IN1 到 OUT1 的域: PlayItem1)为重放域, 并且作为下一个要显示的域, 还指定实 PlayList2 中的预置域(从 IN2 到 OUT2 的域: PlayItem2)为重放域, 如图 6A 所示。准备了由 PlayItem1 和

PlayItem2 组成的单个虚 PlayList, 如图 6B 所示。

现在解释虚 PlayList 的重新编辑。该重新编辑可以通过交替虚 PlayList 中的 IN-或者 OUT 点将新 PlayItem 插入或者附加到虚 PlayList 以及删除虚 PlayList 中的 PlayItem 来列举。虚 PlayList 本身也可以被删除。

- 5 图 7 表示对虚 PlayList 的音频转录(后记录)。其是将音频后记录寄存在虚 PlayList 作为子路径的操作。该音频后记录由应用软件支持。附加音频流作为子路径被加到虚 PlayList 主路径的 AV 流。

- 实 PlayList 和虚 PlayList 的共同点是图 8 所示的改变(移动)PlayList 重放顺序的操作。该操作是盘(卷)中 PlayList 重放顺序的改变并且由在应用格式中定义的 TableOfPlayList 所支持, 正如下面引用例如图 20 要说明的。该操作不会导致 Clip 内容的改变。

- 现在解释标记(Mark)。标记是为指定 Clip 和 PlayList 中的加亮或者特征时间而提供的, 如图 9 所示。加到 Clip 的标记叫做 ClipMark。ClipMark 是例如节目索引点或者场景变化点, 用于指定起因于 AV 流中内容的特征场景。
- 15 ClipMark 是由例如图 1 分析单元 14 产生的。当 PlayList 再现时, 可以引用和使用由 PlayList 引用的 Clip 的标记。

- 附加到 PlayList 的标记叫做 PlayListMark(播放表标记)。PlayListMark 是例如由用户设置的书签点或者恢复点(resume point)。对 Clip 和对 PlayList 的标记的设置是将表示标记时间点的时间戳加到标记表。另一方面, 标记删除
- 20 就是从标记表中移去标记的时间戳。结果, AV 流不会由标记设置或者标记删除所改变。

- 作为 ClipMark 的另一个格式, 由 ClipMark 引用的图象可以以 AV 流中的地址基来指定。Clip 上的标记设置就是将表示标记点的图象的地址基信息加到标记表中。另一方面, 标记删除就是从标记表中移去表示标记点图象的地址基信息。结果, AV 流不会由标记设置或者标记删除所改变。
- 25

- 现在解释缩略图。缩略图是加到 Volume(卷)、PlayList 和 Clip 的静止图象。有两种类型的缩略图, 其中之一是作为表示内容的代表图象的缩略图。这主要是用在主图象中, 为的是让用户用光标(未示出)选择他或她希望观看的内容。另一种缩略图是表示由标记点指向的场景的图象。

- 30 Volume 和对应 PlayList 需要具有代表图象。当盘设置在记录和/或再现设备 1 中的位置时, Volume 的代表图象被预先提出用作初始地展示表示盘内容

的静止图象。注意，盘的意思是预先提出作为盘形状的记录介质 100。PlayList 的代表图象被预先提出用作表示 PlayList 内容的静止图象。

- 作为 PlayList 的代表图象，可以考虑使用 PlayList 的初始图象作为缩略图(代表图象)。但是，在 0 重放时间的引导图象不必要是表示内容的最佳图象。
- 5 因此，允许用户设置可选择性图象作为 PlayList 的缩略图。两种类型的缩略图，即作为表示 Volume 之代表图象的缩略图和作为表示 PlayList 之代表图象的缩略图，叫作菜单缩略图。由于经常显示菜单缩略图，这些缩略图需要从盘中以升高的速度读出。因此，以单个文件存储该全部菜单缩略图是有效率的。菜单缩略图不必是从卷的移动图象中提取出的图象，但是可以从个人
- 10 计算机或者是数字静止摄像机获取的图象，如图 10 所示。

另一方面，Clip 和 PlayList 需要用复合标记来做标记，同时标记点的图象需要被容易地观看，目的是获得标记位置的内容。表示这种标记点的图象叫作标记缩略图。因此，作为标记操作的正本的图象主要是所提取的标记点图象而不是从外部获取的图象。

- 15 图 11 表示附加到 PlayList 的标记和标记缩略图之间的关系，同时图 12 表示附加到 Clip 的标记和标记缩略图之间的关系。与菜单缩略图的区别，标记缩略图用在例如用于表示 PlayList 细节的子菜单中，而其不要求在短的存取时间中读出。所以，无论何时要求缩略图，记录和/或再现设备 1 打开文件和读出一部分该文件，同时即使由记录和/或再现设备 1 进行的文件打开和读
- 20 出一部分文件要占用一些时间，也不会出现任何问题。

为了减少在卷中出现的文件数，最好将整个标记缩略图存储在一个文件中。尽管 PlayList 可以具有一个菜单缩略图和多个标记缩略图，但不要求用户直接地选择 Clip(通常，Clip 是通过 PlayList 选择的)，因此没有必要提供菜单缩略图。

- 25 图 13 表示菜单缩略图，标记缩略图，PlayList 和 Clip 之间的关系。在菜单缩略图中，文件归档为从一个 PlayList 向另一个提供的菜单缩略图。在菜单缩略图中，文件包含有卷缩略图，其表示在盘上记录的数据内容。在菜单缩略图中，文件归档为从一个 PlayList 到另一个和从一个 Clip 到另一个创建的缩略图。

- 30 下面说明 CPI(特征点信息)。CPI 是包含在 Clip 信息文件中的数据，并且主要用于发现 Clip AV 流文件中的数据地址，在该地址，当提供 Clip 存取点

的时间戳时开始数据读出。在本实施例中使用两种类型的 CPI，其中之一是 EP_map，另一种是 TU_map。

EP_map 是从基本流和传输流中提取的入口点(entry point)(EP)数据的表。其具有用来发现在此开始解码的 AV 流中入口点地点的地址信息。一个 EP 数据由显示时间戳(PTS)和与 PTS 相联系的存取单元的 AV 流中的数据地址构成的，该数据地址与 PTS 配对。

EP_map 主要用于两个目的。第一，其用于发现在由 PlayList 的 PTS 引用的存取单元中的 AV 流的数据地址。第二，EP_map 用于快速前进重放或者快速后退重放。在通过记录和/或再现设备 1 记录输入 AV 流时，如果流的句法能够被分析，则在盘上创建和记录 EP_map。

TU_map 具有从通过数字接口输入的传输数据包的到达时间点得出的时间单元(TU)数据的表。其提供了基于到达时间的时间 arrival_time_based 和 AV 流中数据地址之间的关系。当记录和/或再现设备 1 记录输入 AV 流并且流的句法不能被分析时，则在盘上创建和记录 TU_map。

STCInfo 存储 AV 流文件中的非连续点信息，该 AV 流文件存储了 MPEG-2 传输流。

当 AV 流具有 STC 的非连续点时，相同的 PTS 值可以出现在 AV 流文件中。因此，如果 AV 流中的时间点是根据 PTS 基指定的，则存取点的 PTS 不足以指定该点。而且，还要求包含 PTS 的连续 STC 域的索引。在该格式中，连续 STC 域和其索引被分别叫做 STC 序列和 STC_sequence_id(STC 序列 id)。STC 序列信息是由 Clip 信息文件的 STCInfo 定义的。

STC_sequence_id 用在 AV 流文件中并且在具有 TU_map 的 AV 流文件中是可选择的。

节目是每个基本流的集合并且共同拥有对这些流进行同步再现的单个系统时间基。

再现设备(图 1 的记录和/或再现设备 1)在其解码之前知道 AV 流的内容是有用的。这些内容包括例如传输音频或者视频基本流的传输数据包的 PID 值，或者诸如 HDTV 视频或者 MPEG-2 AAC 音频流的视频或者音频元件的类型。该信息对创建用于给用户显示引用 AV 流的 PlayList 内容的屏幕菜单是有用的。其对于设置对应设备之 AV 解码器和多路分用器的初始状态是同样有用的。

由于这个原因, Clip 信息文件拥有用于说明节目内容的 ProgramInfo。

可以发生的是, 节目内容在其中存储了 MPEG-2 传输流的 AV 流文件中应当可以改变。例如, 可以改变传输视频基本流的传输数据包的 PID, 或者可以将视频流的元件类型从 SDTV 改变为 HDTV。

- 5 ProgramInfo 存储了关于 AV 流文件中节目内容的改变点的信息。其中节目内容保持不变的 AV 流文件的域叫作 program_sequence(节目序列)。

该节目序列用在具有 EP_map 的 AV 流文件中, 并且在具有 TU_map 的 AV 流文件中是可选择的。

- 10 本实施例定义了自身编码流格式(SESF)。该 SESF 用于编码模拟输入信号和用于解码数字输入信号, 其随后用于顺序地将解码的信号编码成 MPEG-2 传输流。

SESF 定义了与 MPEG-2 传输流和 AV 流有关的基本流。当记录和/或再现设备 1 编码和记录 SESF 流时, 在盘上创建和记录 EP_map。

- 15 数字广播流使用用于在记录介质 100 进行记录的下述系统之一: 第一, 数字广播流自动解码成 SESF 流。在这种情况下, 所记录的流一定要符合 SESF 并且在盘上一定要准备和记录 EP_map。

另外, 形成数字广播流的基本流自动解码成新的基本流, 并且重新多路复用成符合流格式的新的传输流, 该流格式由用于标准化数字广播流的组织指定的。在这种情况下, 在盘上一定要创建和记录 EP_map。

- 20 例如, 假设输入流是符合 ISDB(日本数字 BS 的标准名称)的 MPEG-2 传输流, 其中具有包含 HDTV 视频流和 MPEG AAC 音频流的传输流。HDTV 视频流自动解码成 SDTV 视频流, 该 SDTV 视频流和原始的 AAC 音频流重新多路复用成 TS。SDTV 流和传输流两者都需要符合 ISDB 格式。

- 25 在记录介质 100 上记录数字广播流的另一个系统是进行输入传输流的透明记录, 即记录不变的输入传输流, 在这种情况下, EP_map 被列出和记录在盘上。

或者, 输入传输流被透明地记录, 即输入传输流被不变地记录, 在这种情况下, TU_map 被创建和记录在盘上。

- 30 下面解释目录和文件。记录和/或再现设备 1 以下描述为 DVR(数字视频记录)。图 14 表示盘上的典型目录结构。DVR 盘的目录可以列举为: 包括 “DVR” 目录的根目录; 和包括 “PLAYLIST” 目录、“CLIPINF” 目录、“M2TS”

目录和“DATA(数据)”目录的“DVR”目录,如图14所示。尽管在根目录下可以创建除这些目录之外的其它目录,但这些在本实施例的应用格式中被忽略。

在“DATA”目录之下,存储有由DVR应用格式指定的所有文件和目录。

- 5 “DVR”目录包括四个目录。在“PLAYLIST”目录下放置了实PlayList和虚PlayList的数据库文件。后面的目录可以存在于没有PlayList的状态中。

在“CLIPINF”下放置了Clip数据库。该目录也可以存在于没有AV流文件的状态中。在“DATA”目录中,存储有诸如数字TV广播的数据广播文件。

- 10 “DVR”目录存储了下述文件。即在DVR目录下创建的“info.dvr”以存储应用层的综合信息。在DVR目录下,一定有单个info.dvr。假设该文件名称对info.dvr是固定的。“menu.thmb”存储了与菜单缩略图有关的信息。在DVR目录下,一定有0或1的标记缩略图。假设该文件名称对“menu.thmb”是固定的。如果没有菜单缩略图,该文件可以不存在。

- 15 “mark.thmb”文件存储了与标记缩略图图象有关的信息。在DVR目录下,一定有0或1的标记缩略图。假设该文件名称对“menu.thmb”是固定的。如果没有菜单缩略图,该文件可以不存在。

- “PLAYLIST”目录存储了两种类型的PlayList文件,它们是实PlayList和虚PlayList。“xxxxx.rpls”文件存储了与一个实PlayList有关的信息。对于
20 每个实PlayList创建一个文件。文件名称是“xxxxx.rpls”,这里“xxxxx”表示从0到9的五个数值数字。文件扩展名一定是“rpls”。

“yyyyy.vpls”存储了与一个虚PlayList有关的信息。具有文件名称“yyyyy.vpls”的一个文件是从一个虚PlayList到另一个被创建的,这里“yyyyy”表示从0到9的五个数值数字。文件扩展名一定是“vpls”。

- 25 “CLIPINF”目录存储了一个与每个AV流文件相联系的文件。

“zzzzz.clps”是对应于一个AV流文件(Clip AV流文件或者Bridge-Clip流文件)的Clip信息文件。文件名称是“zzzzz.clpi”,这里“zzzzz”表示从0到9的五个数值数字。文件扩展名一定是“clpi”。

- “M2TS”目录存储了AV流文件。“zzzzz.m2ts”文件是由DVR系统操
30 作的AV流文件。这是Clip AV流文件或者Bridge-Clip AV流文件。文件名称是“zzzzz.m2ts”,这里“zzzzz”表示从0到9的五个数值数字。文件扩展名

一定是“m2ts”。

“DATA”目录存储了从数据广播中传输的数据。例如该数据可以是 XML 或者 MPEG 文件。

现在解释每个目录(文件)的句法和语义。图 15 表示“info.dvr”文件的句法。
5 “info.dvr”文件是由三个对象构成,即 DVRVolume()、TableOfPlayLists() 和 MakersPrivateData()。

解释图 15 所示的 info.dvr 的句法。根据来自“info.dvr”文件的引导字节的相对字节数, TableOfPlayList_Start_address 表示 TableOfPlayLists()的引导地址。相对字节数是从 0 开始计算的。

10 根据来自“info.dvr”文件的引导字节的相对字节数, MakersPrivateData_Start_address 表示 MakersPrivateData()的引导地址。相对字节数是从 0 开始计算的。Padding_word 与“info.dvr”的句法相关地插入。N1 和 N2 是可选择正整数。每个填充字可以假设为可选择的值。

DVRVolume()存储了指示卷(盘)内容的信息。图 16 表示 DVRVolume 的句法。
15 现在解释图 16 所示的 DVRVolume()的句法。Version_number 表示指示 DVRVolume()的版本号的四个字符字母。Version_number 编码成与 ISO646 相联系的“0045”。

长度由 32 位无符号整数表示,其表明从长度字段之后直接到 DVRVolume()尾端的字节数目。

20 ResumeVolume()记忆了在 Volume 中最后产生的实 PlayList 或者虚 PlayList 的文件名称。但是,当用户已经中断了实 PlayList 或者虚 PlayList 的重放时的重放位置存储在 PlayListMark()定义的恢复标记中(见图 42 和 43)。

图 17 表示 ResumeVolume()的句法。解释图 17 所示的 ResumeVolume()的句法。Valid_flag 表示当该 1 位标志分别设置为 1 或者 0 时

25 resume_PlayList_name 字段是有效的或是无效的。

resume_PlayList_name 的 10 字节字段表示要恢复的实 PlayList 或者虚 PlayList 的文件名称。

图 16 所示的 DVRVolume()句法中的 UIAppInfoVolume 存储了与 Volume 相关的用户接口应用程序的参数。图 18 表示 UIAppInfoVolume 的句法,现在
30 解释其语义。Character_set 的 8 位字段表示在 Volume_name 字段中编码的字符字母的编码方法。该编码方法对应于图 19 所示的值。

Name_length 的 8 位字段表示在 Volume_name 字段中表示的 Volume 名称的字节长度。Volume_name 字段表示 Volume 的名称。从字段左边开始计数的 Name_length 数的字节数是有效字符数目，并且表示 Volume 的名称。在这些有效字符字母后面的值可以是任何值。

- 5 Volume_protect_flag 是表示卷中的内容是否能够无限制地展示给用户的标志。如果该标志设置为 1，则仅仅在用户已经成功地正确输入 PIN 号(通过口令)的情况下，卷中的内容允许被展示(再现)给用户。如果该标志设置为 0，即使在 PIN 号没有被用户输入的情况下，卷中的内容也允许展示给用户。

- 10 当用户已经将盘插入播放器时，如果该标志已经设置为 0，或者该标志设置为 1 但用户已经成功地正确输入 PIN 号的话，记录和/或再现设备 1 显示盘中的 PlayList 表。在对应 PlayList 之再现的限制是与 Volume_protect_flag 不相关的，并且是由 UIAppInfoVolume 中定义的 playback_control_flag 表示的。

- 15 PIN 是由从 0 到 9 的四个数值数字构成的，其中的每一个都是根据 ISO/IEC646 编码的。ref_thumbnail_index 字段表示加到卷中的缩略图图象的信息。如果 ref_thumbnail_index 字段是除 0xFFFF 之外的值，则缩略图图象被加到卷中。缩略图图象存储在 menu.thumb 文件中。该图象是使用 menu.thumb 文件中的 ref_thumbnail_index 的值引用的。如果 ref_thumbnail_index 字段是 0xFFFF，则其表示缩略图图象已经被加到卷中。

- 20 解释图 15 所示的 info.dvr 句法中的 TableOfPlayList()。TableOfPlayList() 存储了 PlayList(实 PlayList 和虚 PlayList)的文件名称。记录在卷中的所有 PlayList 文件都包含在 TableOfPlayList()中，该 TableOfPlayList()表示卷中 PlayList 的缺省的重放序列。

- 25 图 20 表示 TableOfPlayList()的句法，现在解释之。TableOfPlayList()的 version_number 表示四个字符字母，其表示 TableOfPlayList 的版本号。
version_number 一定要根据 ISO646 编码成“0045”。

- 30 长度是无符号的 32 位整数，其表明从长度字段之后直接到 TableOfPlayList()尾端的 TableOfPlayList()的字节数目。Number_of_PlayLists 的 16 位字段表示包括 PlayList_file_name 在内的循环(for-loop)的循环数。该数值数字一定要等于记录在卷中的 PlayList 的数目。PlayList_file_name 的 10 字节数值数字表示 PlayList 的文件名称。

图 21 表示 TableOfPlayList()句法的另一种结构。图 21 所示的句法是由图

20 所示的其中包含 UIAppInfoPlayList 的句法构成的。通过这种包括 UIAppInfoPlayList 的结构, 在读出 TableOfPlayList 时简单地创建菜单图象变成有可能。下面的解释是根据使用图 20 所示的句法进行的。

解释图 15 所示的 info.dvr 中的 MakersPrivateData。提供 MakersPrivateData
5 是为了允许记录和/或再现设备 1 的制造者将制造者的私有数据插入 MakersPrivateData()以用于不同公司的特殊应用。每个制造者的私有数据已经被标准化为 maker_ID, 以识别已经定义它的制造者。MakersPrivateData()可以包含一个或者多个 maker_ID。

如果预置制造者希望插入私有数据, 并且不同制造者的私有数据已经包
10 含在 MakersPrivateData()中, 则在不擦除预先存在的旧私有数据的情况下, 新私有数据加到 MakersPrivateData()。因此, 在本实施例中, 多个制造者的私有数据能够包含在一个 MakersPrivateData()中。

图 22 表示 MakersPrivateData 的句法。解释图 22 表示的 MakersPrivateData 的句法。TableOfPlayList()的 version_number 表示四个字符字母, 其表示
15 TableOfPlayList 的版本号。version_number 一定要根据 ISO646 编码成“0045”。长度是无符号 32 位整数, 其表明从长度字段之后直接到 MakersPrivateData()尾端的 TableOfPlayList()的字节数目。

Mpd_blocks_start_address 表示根据来自 MakersPrivateData()引导字节之
字节数的第一个 Mpd_block()的前端地址。Number_of_maker_entries 是 16 位
20 无代码整数, 其提供 MakersPrivateData()中包括的制造者私有数据的入口号。在 MakersPrivateData()中一定不会出现具有相同 maker_ID 值的两个或多个制造者私有数据。

mpd_blocks_size 是 16 位无符号整数, 其提供以 1024 字节为单位的一个
mpd_block 大小。例如, 如果 Mpd_blocks_size=1, 则其表示一个 Mpd_block
25 的大小是 1024 字节。Number_of_mpd_block 是 16 位无符号整数, 其提供 MakersPrivateData()中包含的 mpd_block 数。maker_ID 是 16 位无符号整数, 其表示已经创建制造者私有数据之 DVR 系统的模型数代码。编码成 maker_ID 的值是由发许可证者指定的。

maker_mode_code 是 16 位无符号整数, 其表示已经创建制造者私有数据
30 之 DVR 系统的模型数代码。编码成 maker_mode_code 的值是由制造者设置的, 该制造者已经接收了格式许可。start_mpd_block_number 是 16 位无符号

整数，其表示开始制造者私有数据的 mpd_block_number 号。制造者私有数据的前端一定要与 mpd_block 的前端对齐。start_mpd_block_number 对应于 mpd_block 循环中的变量 j。

mpd_length 是 32 位无符号整数，其表示制造者私有数据的大小。

- 5 mpd_block 是其中存储了制造者的私有数据的区。MakersPrivateData() 中的所有 mpd_block 一定要是相同大小。

解释实 Playlist 文件和虚 Playlist 文件，换言之，即 xxxxx.rpls 和 yyyyy.vpls。图 23 表示 xxxxx.rpls(实 Playlist)和 yyyyy.vpls(虚 Playlist)的句法，它们具有相同的句法结构。每一个 xxxxx.rpls 和 yyyyy.vpls 都是由三个对象构成，即 Playlist()、PlaylistMark()和 MakersPrivateData()。

根据来自 Playlist 文件前端的相对字节数为单位，PlaylistMark_start_address 表示 PlaylistMark()的引导地址。相对字节数是从 0 开始计算的。

- 根据来自 Playlist 文件前端的相对字节数为单位，
15 MakersPrivateData_start_address 表示 MakersPrivateData()的引导地址。相对字节数是从 0 开始计算的。

Padding_word(填充字)是根据 Playlist 文件的句法被插入的，其中 N1 和 N2 是可选的正整数。每个填充字可以假设为可选的值。

- 尽管其已经简要地解释了，下面仍然解释 Playlist。除 Bridge-Clip 之外的所有 Clips 中的重放域一定要由盘中的所有 Playlist 引用。而且，两个或者多个实 PlayLists 一定不重叠相同 Clip 中由它们的 PlayItem 表示的重放域。

参照图 24A、24B 和 24C。对于所有的 Clips，存在对应的实 Playlist，如图 24A 所示。即使在编辑操作已经关闭之后也可看到该规则，如图 24B 所示。因此，所有的 Clip 一定通过引用实 Playlist 之一来看到。

- 25 参照图 24C，虚 Playlist 的重放域一定包含在重放域和 Bridge-Clip 重放域中。盘中一定不出现不由任何虚 Playlist 引用的 Bridge-Clip。

- 包含 PlayItem 表的实 Playlist 一定不含有 SubPlayItem。虚 Playlist 包含 PlayItem 表，并且如果包含在 Playlist()中的 CPI_type 是 EP_map 类型以及 Playlist_type 是 0(含有视频和音频的 Playlist)，则虚 Playlist 可以包含一个
30 SubPlayItem。在本实施例的 Playlist()中，SubPlayItem 仅仅用于音频后记录。由一个虚 Playlist 拥有的 SubPlayItem 数一定是 0 或者 1。

下面解释 PlayList。图 25 表示现在要解释的 PlayList 句法。version_number 表示四个字符字母，其表示 PlayList() 的版本号。version_number 根据 ISO646 编码成“0045”。长度是无符号 32 位整数，其表明从长度字段之后直接到 PlayList() 尾端的 PlayList() 的总字节数目。PlayList_type 是 8 位字段，其表示 PlayList 类型，图 26 示出其一个例子。

CPI_type 是一位标志，其表示由 PlayItem() 和 SubPlayItem() 引用的 Clip 的 CPI_type 之值。在由一个 PlayList 引用的所有 Clips 的 CPI 中定义的 CPI_type 一定具有相同值。Number_of_PlayItems 是 16 位字段，其表示出现在 PlayList 中的 PlayItem 数。

10 对应于预置 PlayItem() 的 PlayItem_id 是由其中 PlayItem() 出现在包含 PlayItem() 之循环中的序列定义的。PlayItem_id 以 0 开始。

Number_of_SubPlayItems 是 16 位字段，其表示在 PlayList 中的 SubPlayItem 数。该值是 0 或者 1。附加音频流路径(音频流路径)是一种子路径类型。

15 解释图 25 所示的 PlayList 句法的 UIAppInfoPlayList。UIAppInfoPlayList 存储了涉及 PlayList 的用户接口应用程序的参数。图 27 表示现在要解释的 UIAppInfoPlayList 的句法。Character_set 是 8 位字段，其表示用于编码在 PlayList_name 字段中编码的字符字母的方法。该编码方法对应于与图 19 所示的表一致的值。

20 Name_length 是 8 位字段，其表示在 PlayList_name 字段中表示的 PlayList 名称的字节长度。PlayList_name 字段表示 PlayList 名称。从字段左边计数的 Name_length 数的字节数是有效字符数并且表示 PlayList 名称。在这些有效字符字母后面的值可以是任何值。

25 Record_time_and_date 是 56 位字段，其存储了记录 PlayList 的日期和时间。该字段是二进制编码的十进制(BCD)编码的年/月/日/小时/分钟/秒的 14 个数值数字。例如，2001/12/23: 01: 02: 03 编码成“0x20011223010203”。

持续时间(duration)是 24 位字段，其表示以小时/分钟/秒为单位的 PlayList 的总重放时间。该字段是二进制编码的十进制(BCD)编码的 6 个数值数字。例如，01: 45: 30 编码成“0x014530”。

30 Valid_period 是 32 位字段，其表示 PlayList 的有效时间周期。该字段是 4 位二进制编码的十进制(BCD)编码的 8 个数值数字。Valid_period 用在记录和/或再现设备 1 中，即，当有效周期已经消失的 PlayList 将自动擦除时，例如，

2001/05/07 编码成 “0x20010507”。

Maker_ID 是 16 位无符号整数，其表示是最近更新其 PlayList 的 DVR 播放器(记录和/或再现设备 1)的制造者。编码成 Maker_ID 的值分配给 DVD 格式的发放许可证者。Maker_code 是 16 位无符号整数，其表示是最近更新的 PlayList 的 DVR 播放器的模型数。编码成 Maker_code 的值是由制造者确定的，该制造者已经接收 DVR 格式的许可。

如果 playback_control_flag 的标志设置为 1，则其 PlayList 仅仅当用户成功地输入 PIN 号时再现。如果该标志设置为 0，则用户在不输入 PIN 号的情况下可以观看该 PlayList。

10 如果 write_protect_flag 设置为 1，则除 write_protect_flag 之外，该 PlayList 的内容既不能被擦除也不能改变。如果该标志设置为 0，用户可自由地擦除或者改变该 PlayList。如果该标志设置为 1，则在用户进行擦除、编辑或者重写 PlayList 之前记录和/或再现设备 1 显示请求用户进行重新确认的消息。

其中 write_protect_flag 设置为 0 的实 PlayList 可以存在，引用实 PlayList 之 Clip 的虚 PlayList 可以存在，并且虚 PlayList 的 write_protect_flag 可以设置为 1。如果用户希望擦除实 PlayList，记录和/或再现设备 1 发出报警给用户以便出现前述虚 PlayList 或者在擦除实 PlayList 之前“最小化”该实 PlayList。

如果 is_played_flag 设置为 1，如图 28B 所示，则其表示自从其记录以来该 PlayList 至少再现了一次，而如果其设置为 0，则其表示自从其记录以来该 PlayList 甚至没有再现过一次。

文档(Archive)是两位字段，其表示 PlayList 是原始的还是拷贝的，如图 28C 所示。ref_thumbnail_index 的字段表示代表 PlayList 之缩略图图象的信息。如果 ref_thumbnail_index 字段是除 0xFFFF 之外的值，则代表 PlayList 之缩略图图象被加在 PlayList 中，其中 PlayList 存储在 menu.thmb 文件中。该图象使用在 menu.thmb 文件中的 Ref_thumbnail_index 之值来引用。如果 Ref_thumbnail_index 是 0xFFFF，则没有代表 PlayList 之缩略图图象被加在 PlayList 中。

下面解释 PlayItem。一个 PlayItem()基本上包含下述数据：用于指定 Clip 文件名称的 Clip_Information_file_name，成对指定 Clip 重放域的 IN_time 和 OUT_time，在 PlayList()中定义的 CPI_type 是 EP_map 类型的情况下由 IN_time 和 OUT_time 引用的 STC_sequence_id，以及表示在先 PlayItem 和当前 PlayItem

之连接条件的 Connection_Condition。

如果 PlayList 是由两个或者多个 PlayItem 构成，则在 PlayList 的全球时间轴上这些 PlayItem 成行排列，没有时间间隙或者重叠。如果在 PlayList 中定义的 CPI_type 是 EP_map 类型和当前 PlayList 不具有 BridgeSequence()，则

- 5 IN_time 和 OUT_time 对在 STC 连续域上一定表示与 STC_sequence_id 指定时间相同的时间。这种例子示于图 29。

图 30 表示这种情况，其中由 PlayList()定义 CPI_type，如果当前 PlayItem 具有 BridgeSequence()，则应用现在解释的规则。表示成 IN_time1 的在当前 PlayItem 之前的 PlayItem 之 IN_time 表示在当前 PlayItem 之

- 10 BridgeSequenceInfo()中指定的 Bridge-Clip 的时间。该 OUT_time 一定遵守随后要解释的编码限制。

表示成 IN_time2 的当前 PlayItem 的 IN_time 表示在当前 PlayItem 的 BridgeSequenceInfo()中指定的 Bridge-Clip 的时间。该 IN_time 也一定遵守后面要解释的编码限制。表示成 OUT_time2 的当前 PlayItem 的 PlayItem 之

- 15 OUT_time 表示关于由当前 PlayItem 的 STC_sequence_id 指定的 STC 连续域的时间。

如果 PlayList()的 CPI_type 是 TU_map 类型，则成对的 PlayItem 的 IN_time 和 OUT_time 表示关于相同 Clip AV 流的时间，如图 31 所示。

PlayItem 句法示于图 32。对于图 32 所示的 PlayItem 的句法，

- 20 Clip_Information_file_name 的字段表示 Clip 信息的文件名称。由该 Clip 信息文件的 ClipInfo()定义的 Clip_stream_type 一定表示 Clip AV 流。

STC_sequence_id 是 8 位字段和表示由 PlayItem 引用的连续 STC 域的 STC_sequence_id。如果在 PlayList()指定的 CPI_type 是 TU_map 类型，则该 8 位字段没有意义和设置为 0。IN_time 是 32 位字段并且用于存储 PlayItem 的重放开始时间。IN_time 的语义不同于 PlayList()中定义的 CPI_type，如图 33 所示。

- 25 OUT_time 是 32 位字段并且用于存储 PlayItem 的重放结束时间。OUT_time 的语义不同于 PlayList()中定义的 CPI_type，如图 34 所示。

- Connection_condition 是 2 位字段，其表示在在先 PlayItem 和当前 PlayItem 之间的连接条件，如图 35 所示。图 36A 到 36D 表示图 35 所示的 Connection_condition 的各种状态。

参照图 37 解释 BridgeSequenceInfo。该 BridgeSequenceInfo 是当前 PlayItem 的辅助信息和包括下述信息。即，BridgeSequenceInfo 包括用于指定 Bridge_Clip AV 流文件的 Bridge_Clip_Information_file_name 和指定相应的 Clip 信息文件的 Bridge_Clip_Information_file_name(图 45)。

- 5 其也是关于由在先 PlayItem 引用的 Clip AV 流的源数据包的地址。该源数据包之后是连接 Bridge-Clip AV 流的第一源数据包。该地址叫作 RSPN_exit_from_previous_Clip。其也是关于由当前 PlayItem 引用的 Clip AV 流之源数据包的地址。该源数据包之前是连接 Bridge-Clip AV 流文件的最后的源数据包。该地址叫作 RSPN_enter_to_current_Clip。

- 10 图 37 中，RSPN_arrival_time_discontinuity 表示 Bridge-Clip AV 流的源数据包地址，其中在到达时间基上没有连续点。该地址被定义在 ClipInfo()中(图 46)。

- 图 38 表示 BridgeSequenceInfo 的句法。回到图 38 所示的 BridgeSequenceInfo 的句法，Bridge_Clip_Information_file_name 的字段表示对
15 应于 Bridge_Clip_Information_file 的 Clip 信息文件的文件名称。该 Clip 信息文件的 ClipInfo()中定义的 Clip_stream_type 一定表示 ‘Bridge_Clip AV 流’。

- RSPN_exit_from_previous_Clip 的 32 位字段是关于由在先 PlayItem 引用的 Clip AV 流的源数据包的相对地址。该源数据包之后是连接 Bridge-Clip AV 流文件的第一源数据包。RSPN_exit_from_previous_Clip 具有基于源数据包数为
20 单位的大小，并且以 ClipInfo()中定义的 offset_SPN 值从由在先 PlayItem 引用的 Clip AV 流的第一源数据包开始计数。

- RSPN_enter_to_current_Clip 的 32 位字段是关于由当前 PlayItem 引用的 Clip AV 流的源数据包的相对地址。该源数据包之前是连接 Bridge-Clip AV 流文件的最后源数据包。RSPN_enter_to_current_Clip 具有基于源数据包数为
25 位的大小。RSPN_enter_to_current_Clip 以 offset_SPN 值作为初始值开始计数，offset_SPN 值是由当前 PlayItem 引用的 Clip AV 流的第一源数据包的 ClipInfo()中定义的。

- 参照图 39 解释 SubPlayItem。仅仅在 PlayList()的 CPI_type 是 EP_map 类型时才允许使用 SubPlayItem()。在本实施例中，SubPlayItem 仅仅用于音频后
30 记录。SubPlayItem()包括下述数据。第一，其包括用于指定 PlayList 中子路径引用的 Clip 的 Clip_Information_file_name。

它还包括用于指定 Clip 中子路径重放域的 SubPath_IN_time 和 SubPath_OUT_time。另外, 其包括 sync_PlayItem_id 和用于指定在主路径时间轴上开始子路径再现的时间的 start_PTS_of_PlayItem。由子路径引用的 Clip AV 流一定不包含 STC 非连续点(系统时间基的非连续点)。用在子路径的 Clip 5 的音频采样的时钟锁定在主路径的音频采样的时钟。

图 40 表示 SubPlayItem 的句法。回到图 40 所示的 SubPlayItem 的句法, Clip_Information_file_name 的字段表示 Clip 信息文件的文件名称并且由 PlayList 中的子路径使用。该 ClipInfo() 中定义的 Clip_stream_type 一定表示 Clip AV 流。

10 sync_PlayItem_id 的 8 位字段表示子路径类型。这里, 仅仅设置了 '0x00', 如图 41 所示, 同时其它值被保留为今后使用。

sync_PlayItem_id 的 8 位字段表示 PlayItem 的 PlayItem_id, PlayItem 包含在主路径时间轴上的子路径的重放开始时间。对应于预置 PlayItem 的 PlayItem_id 的值在 PlayList() 中定义(图 25)。

15 sync_start_PTS_of_PlayItem 的 32 位字段表示在主路径时间轴上的子路径的重放开始时间, 并且表示由 sync_PlayItem_id 引用的 PlayItem 上的 PTS(显示时间戳)的高 32 位。SubPath_IN_time 的高 32 位字段存储了子路径的重放开始时间。SubPath_IN_time 表示对应于子路径中第一显示单元的 33 位 PTS 的高 32 位。

20 SubPath_OUT_time 的高 32 位字段存储了子路径的重放结束时间。SubPath_OUT_time 表示由下述方程计算的 Presentation_end_TS 之值的高 32 位。

$$\text{Presentation\_end\_TS} = \text{PTS\_OUT} + \text{AU\_duration}.$$

这里, PTS_OUT 是对应于 SubPath 最后显示单元的 33 位长度的 PTS, 25 AU_duration 是基于 SubPath 最后显示单元的显示周期 90kHz。

下面, 解释图 23 所示的 xxxxx.rpls 和 yyyyy.vpls 句法中的 PlayListMark()。与 PlayList 有关的标记信息被存储该 PlayListMark 中。图 42 表示 PlayListMark 的句法。回到图 42 所示的 PlayListMark 的句法, version_number 是四个字符字母, 其表示该 PlayListMark() 的版本号。version_number 一定要根据 ISO646 30 编码成 "0045"。

长度(Length)是无符号 32 位整数, 其表示从长度字段之后直接到

PlayListMark()尾端的 PlayListMark()字节数目。Number_of_PlayListMarks 是表示 16 位无符号整数,其表示存储在 PlayListMark 中的标记数目。

Number_of_PlayListMarks 可以是 0。Mark_type 是标记类型的 8 位字段并且被编码在图 43 表示的表中。

- 5 Mark_time_stamp 的 32 位文件存储表示由标记指定的点的时间戳。
Mark_time_stamp 的语义不同于在 PlayList()中定义的 CPI_type,如图 44 所示。
PlayItem_id 是指定 PlayItem 的 8 位字段,其中输入标记。对应于预置 PlayItem 的 PlayItem_id 的值被定义在 PlayList()中(见图 25)。

- 10 Character_set 的 8 位字段表示在 mark_name 字段中编码的字符字母的编码方法。编码方法对应于图 19 所示的值。Name_length 的 8 位字段表示在 mark_name 字段所示的标记名称的字节长度。mark_name 字段表示在 mark_name 字段中表明的标记名称。对应于 Name_length 数的从该字段左边开始的字节数是有效字符字母和表示该标记名称。在 mark_name 字段中,在这些有效字符字母后面的值可以是任意的。

- 15 ref_thumbnail_index 的字段表示加到标记的缩略图图象的信息。如果 ref_thumbnail_index 的字段不是 0xFFFF,则缩略图图象被加到其标记,同时缩略图图象存储在 mark.thmb 文件中。该图象在 mark.thmb 文件中被引用,其使用 ref_thumbnail_index 的值,如后面的解释。如果 ref_thumbnail_index 的字段是 0xFFFF,则其表示没有缩略图图象被加到标记。

- 20 现在解释 Clip 信息文件。zzzzz.clpi(Clip 信息文件)是由六个对象构成的,如图 45 所示。这些是 ClipInfo()、STC_Info()、Program()、CPI()、ClipMark()和 MarkersPrivateData()。对于 AV 流(Clip AV 流或者 Bridge-Clip AV 流)和对应的 Clip 信息文件,使用相同的“zzzzz”数字串。

解释回到图 45 所示的 zzzzz.clpi(Clip 信息文件)的句法。

- 25 ClipInfo_Start_address 表示 ClipInfo()的前端地址,其中具有从 zzzzz.clpi 文件的前端字节作为单位的相对字节数。该相对字节数是从 0 计数。

- STC_Info_Start_address 表示 STC_Info 的前端地址,其中具有从 zzzzz.clpi 文件的前端字节作为单位的相对字节数。ProgramInfo_Start_address 表示 ProgramInfo()的前端地址,其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。CPI_Start_address 表示 CPI()的前端地址,其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对
- 30

字节数是从 0 计数。

ClipMark_Start_address 表示 ClipMark() 的前端地址,其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。

- MakersPrivateData_Start_address 表示 MakersPrivateData() 的前端地址,其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。Padding_word 根据 zzzzz.clpi 文件的句法被插入。N1, N2, N3, N4 和 N5 一定是 0 或者是可选择的正整数。对应的填充字也可以假设为可选择值。

- 现在解释 ClipInfo。图 46 表示 ClipInfo 的句法。在 ClipInfo() 中存储了对应的 AV 流文件的属性信息(Clip AV 流或者 Bridge-Clip AV 流文件)。

- 回到图 46 所示的 ClipInfo 的句法, version_number 是四个字符字母, 其表示该 ClipInfo() 的版本号。version_number 一定要根据 ISO646 编码成“0045”。长度是无符号 32 位整数, 其表明从长度字段之后直接到 ClipInfo() 尾端的 ClipInfo() 的字节数目。Clip_stream_type 的 8 位字段表示对应于 Clip 信息文件的 AV 流的类型, 如图 47 所示。下面解释对应 AV 流的流类型。

Offset_SPN 的 32 位字段给出 AV 流(Clip AV 流或者 Bridge-Clip AV 流)第一源数据包的第一源数据包数的源数据包数的偏移值。当 AV 流文件首先记录在盘上时, 该 Offset_SPN 一定为 0。

- 参照图 48, 当 AV 流文件的开始部分是通过编辑被擦除时, 该 Offset_SPN 可以假定为除 0 之外的值。在本实施例中, 引用 Offset_SPN 的相对源数据包数(相对地址)经常以 RSPNxxx 形式说明, 这里 xxx 经更新使得 RSPN_xxx 是 RAPN_EP_start。相对源数据包数是以源数据包数作为单位定大小的并且是从 AV 流文件第一源数据包数开始计数, 其中具有 Offset_SPN 值作为初始值。

- 从 AV 流文件第一源数据包到引用相对源数据包数(SPN_xxx)的源数据包的源数据包数是通过下述方程计算的:

$$SPN\_xxx = RSPN\_xxx - Offset\_SPN$$

图 48 表示其中 Offset_SPN 为 4 的例子。

TS_recording_rate 是 24 位无符号整数, 其为 AV 流所要求的给 DVR 驱动(写单元 22)或者来自 DVR 驱动(读出单元 28)的输入/输出位速率。

- Record_time_and_date 是 56 位字段, 用于存储对应于 Clip 之 AV 流的记录日期和时间, 并且以 4 位二进制编码的十进制(BCD)进行编码表示成年/月/日/

小时/分钟的 14 个数值数字。例如, 2001/2/23: 01: 02: 03 编码成
“0x20011223010203”。

持续时间(duration)是 24 位字段, 其表示基于到达时钟的以小时/分钟/秒
的 Clip 总重放时间。该字段是 4 位二进制编码的十进制(BCD)编码的 6 个数值数字。例如, 01: 45: 30 编码成 “0x014530”。

标志 time_controlled_flag 表示 AV 流文件的记录模式。如果该
time_controlled_flag 是 1, 则表示记录模式是这种模式, 其中文件大小是正比
于自记录以来过去的时间, 结果由下述方程所示的条件:

$$\begin{aligned} & Ts\_average\_rate * 192 / 188 * (t - start\_time) - \alpha \leq size\_clip(t) \\ 10 \quad & \leq TS\_average\_rate * 192 / 188 * (t - start\_time) + \alpha \end{aligned}$$

这里 TS_average_rate 是用字节/秒表示的 AV 流文件传输流的平均位速率。

上述方程中, t 表示以秒为单位的时间, start_time 是当 AV 流文件的第一
源数据包被记录时的时间点。size_clip(t)是 10*192 字节, α 是依赖于
15 TS_average_rate 的常数。

如果 time_controlled_flag 设置为 0, 则其表示记录模式没有受到控制,
使得记录所过去的时间正比于 AV 流的文件大小。例如, 输入传输流以透明
方式进行记录。

如果 time_controlled_flag 设置为 1, TS_average_rate 的 24 位字段表示用
20 在上述方程中的 TS_average_rate 的值。如果 time_controlled_flag 设置为 0,
则该字段没有意义和一定要设置为 0。例如, 可变位速率传输流是通过下述
顺序编码的: 首先, 传输速率被设置成 TS_recording_rate。视频流用可变位
速率编码。传输数据包通过不采用空数据包被间断地编码。

RSPN_arrival_time_discontinuity 的 32 位字段是位置的相对地址, 在该位
25 置, 到达时间基不连续性是在 Bridge-Clip AV 流文件上再现的。

RSPN_arrival_time_discontinuity 是以源数据包数作为单位定大小的, 并且从
在 ClipInfo()中定义的作为来自 Bridge-Clip AV 流文件之第一源数据包的
offset_SPN 的值计数。Bridge-Clip AV 流文件中的绝对地址是基于上述方程计
算的:

$$30 \quad SPN\_xxx = RSPN\_xxx - offset\_SPN.$$

Reserver_for_system_use 的 144 位字段是保留给系统的。如果

is_format_identifier_valid 标志是 1, 则其表示 format_identifier 的字段是有效的。如果 is_format_identifier_valid 标志是 1, 则其表示 format_identifier 字段是有效的。如果 is_original_network_ID_valid 标志是 1, 则其表示 is_original_network_ID_valid 的字段是有效的。如果标志

- 5 is_transport_stream_ID_valid 是 1, 则其表示 transport_stream_ID 字节是有效的。如果 is_service_ID_valid 标志是 1, 则其表示 service_ID 字段是有效的。

如果 is_country_code_valid 标志是 1, 则其表示字段 country_code 是有效的。Format_identifier 的 32 位字段表示由传输流中的注册说明符

- 10 (ISO/IEC13818-1 中定义的)拥有的 format_identifier 的值。original_network_ID_ 的 16 位字段表示在传输流中定义的 original_network_ID 的值。

Service_ID 中的 16 位字段表示在传输流中定义的 Service_ID 的值。

country_code 的 24 位字段表示由 ISO3166 定义的国家代码。每个字符代码被 ISO8859-1 编码。例如, 日本表示成 “JPN” 和编码成 “0x4A0x500x4E”。

- 15 stream_format_name 是 ISO-646 的 15 个字符代码, 其表示提供传输流流定义之格式组织的名称。该字段中的无效字节具有 “0xFF” 值。

format_identifier、original_network_ID、transport_stream_ID、service_ID、country_code 和 stream_format_name 表示传输流的服务提供者。这允许识别对音频或者视频流的编码限制和除音视流或者 SI(服务信息)之外的私有数据的流定义。这些信息能够被用来检查解码器是否能够解码该流。如果这种解码是可能的, 则该信息可以在开始解码之前用来初始化解码器系统。

- 20 现在解释 STC_Info。不包含 STC 非连续点(系统时间基的非连续点)的 MPEG-2 传输流中的时间域叫作 STC_sequence。在 Clip 中, STC_sequence 由 STC_sequence_id 的值指定。图 50A 和 50B 表示连续 STC 域。相同的 STC 值决不出现在相同 STC_sequence 中, 尽管 Clip 的最大时间长度是有限的, 如后面的解释。因此, 相同 PTS 值也决不出现在相同 STC_sequence 中。如果 AV 流包含 N 个 STC 非连续点, 这里 $N > 0$, 则 Clip 系统时间基被分开成 $(N+1)$ 个 STC_sequence。

- 30 STC_Info 存储了地点的地址, 其中产生 STC 了不连续性(系统时间基不连续性)。正如参照图 51 的说明, RSPN_STC_start 表示地址, 且在由第 $k+1$ 个 RSPN_STC_start 引用的源数据包之到达时间点的开始和在最后源数据包之到达时间点的结束。

图 52 表示 STC_Info 的句法。回到图 52 所示的 STC_Info 的句法，
version_number 是四个字符字母，其表示该 STC_Info() 的版本号。

version_number 一定要根据 ISO646 编码成 “0045”。

长度是无符号 32 位整数，其表明从该长度字段正后面到 STC_Info 尾端
5 的 STC_Info() 字节数目。如果 CPI() 的 CPI_type 表示 TU_map 类型，则 0 可
以设置在该长度字段中。如果 CPI() 的 CPI_type 表示 EP_map 类型，则
num_of_STC_sequence 一定是不小于 1 的值。

num_of_STC_sequence 的 8 位无符号整数表示 Clip 中的序列数。该值表
示在该字段后面的循环数。对应于预置 STC_sequence 的 STC_sequence_id 是
10 由次序(order)定义的，该次序中出现对应于包含 RSPN_STC_start 之循环中的
STC_sequence 的 RSPN_STC_start。STC_sequence_id 由 0 开始。

RSPN_STC_start 的 32 位字段表示在此 STC_sequence 在 AV 流文件开始
的地址。RSPN_STC_start 表示在 AV 流文件中产生系统时间基不连续性的地
址。RSPN_STC_start 也可以是源数据包的相对地址，源数据包具有 AV 流中
15 新系统时间基的第一个 PCR。RSPN_STC_start 是基于源数据包数的大小，并
且是从具有在 ClipInfo() 中定义的作为初始值之 offset_SPN 的 AV 流文件第一
源数据包计数。在该 AV 流文件中，绝对地址是通过上述方程计算的，即：

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}.$$

现在参照图 53 解释图 45 所示的 zzzzz.clip 句法中的 ProgramInfo。具有
20 Clip 中下述特征的时间域叫作 program_sequence。这些特征是：PCR_PID 的
值不改变，音频基本流数也不改变，对应视频流中的 PID 值不改变，由其
VideoCodingInfo 定义的编码信息不改变，音频基本流数也不改变，对应音频
流中的 PID 值不改变，以及由其 AudioCodingInfo 定义的编码信息不改变。

Program_sequence 在相同时间点上仅仅具有一个系统时间基。

25 Program_sequence 在相同时间点上具有单个 PMT。ProgramInfo() 存储
Program_sequence 开始的地点地址。RSPN_program_sequence_start 表示地址。

图 54 表示 ProgramInfo 的句法。回到图 54 所示的 ProgramInfo，
version_number 是四个字符字母，其表示该 ProgramInfo() 的版本号。

version_number 一定要根据 ISO646 编码成 “0045”。

30 长度是无符号 32 位整数，其表明从该长度字段正后面到 ProgramInfo 尾
端的 ProgramInfo() 字节数目。如果 CPI() 的 CPI_type 表示 TU_map 类型，则

该长度字段可以设置为 0。如果 CPI() 的 CPI_type 表示 EP_map 类型，则 number_of_program 一定是不小于 1 的值。

- number_of_program_sequence 的 8 位无符号整数表示 Clip 中的 program_sequence 数。该值表示在该字段后面的循环数。如果在 Clip 中 program_sequence 不改变，则 1 一定被设置在 program_sequence 数中。
- 5 RSPN_program_sequence_start 的 32 位字段是相对地址，其中在 AV 流上开始该节目序列。

- RSPN_program_sequence_start 是以源数据包数作为单位定大小的，并且是从 AV 流文件的第一源数据包开始以 ClipInfo() 中定义的 offset_SPN 的值计
- 10 数。在该 AV 流文件中，绝对地址是通过下式计算的，即：

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}.$$

在循环句法中的 RSPN_program_sequence_start 的值一定要以升序出现。

- PCR_PID 的 16 位字段表示传输数据包的 PID，传输数据包包含对 program_sequence 有效的有效 PCR 字段。Number_of_audios 的 8 位字段表示
- 15 包含 audio_stream_PID 和 AudioCodingInfo() 之循环数。Video_stream_PID 的 16 位字段表示传输数据包的 PID，传输数据包包含对 program_sequence 有效的视频流。在该字段后面的 VideoCodingInfo() 一定解释由其 Video_stream_PID 引用的视频流的内容。

- Audio_stream_PID 的 16 位字段表示传输数据包的 PID，传输数据包包含
- 20 对其 program_sequence 有效的音频流。在该字段后面的 AudioCodingInfo() 一定解释由其 audio_stream_PID 引用的视频流的内容。

- 其中 Video_stream_PID 的值出现在循环句法中的次序一定等于对 program_sequence 有效的 PMT 中视频流 PID 编码的序列。另外，其中 audio_stream_PID 的值出现在循环句法中的次序一定等于对
- 25 program_sequence 有效的在 PMT 中对音频流 PID 编码的序列。

图 55 表示图 54 所示的 ProgramInfo 句法中的 VideoCodingInfo 的句法。回到图 55 所示的 VideoCodingInfo 的句法，video_format 的 8 位字段表示对应于 ProgramInfo() 的 Video_stream_PID 的视频格式，如图 56 所示。

- 参照图 57，frame_rate 的 8 位字段表示对应于 ProgramInfo() 中的
- 30 video_stream_PID 的视频帧速率。Display_aspect_ratio 的 8 位字段表示对应于 ProgramInfo() 的 video_stream_PID 的视频显示纵横比。

图 59 表示图 54 所示的 ProgramInfo 句法中的 AudioCodingInfo 的句法。回到图 59 所示的 AudioCodingInfo 的句法, audio_format 的 8 位字段表示对应于 ProgramInfo()的 audio_stream_PID 之音频编码方法, 如图 60 所示。

audio_component_type 的 8 位字段表示对应于 ProgramInfo()的

- 5 audio_stream_PID 之音频元件类型,如图 61 所示,同时 sampling_frequency 的 8 位字段表示对应于 ProgramInfo()的 audio_stream_PID 之音频采样频率,如图 62 所示。

- 解释图 45 所示 zzzzz.clip 句法中的 CPI(特征点信息)。CPI 用于将 AV 流中的时间信息与其文件中的地址相关联。CPI 有两种类型, 即 EP_map 和
- 10 TU_map。图 63 中, 如果 CPI()中的 CPI_type 是 EP_map, 则其 CPI()包含 EP_map。图 64 中, 如果 CPI()中的 CPI_type 是 TU_map, 则其 CPI()包含 TU_map。一个 AV 流具有一个 EP_map 或者一个 TU_map。如果 AV 流是 SESF 传输流, 则对应的 Clip 一定拥有 EP_map。

- 图 65 表示 CPI 的句法。回到图 65 所示的 CPI 的句法, version_number
- 15 是四个字符字母,其表示该 CPI()的版本号。version_number 一定要根据 ISO646 编码成“0045”。长度是无符号 32 位整数,其表明从该长度字段正后面到 CPI()尾端的字节数目。CPI_type 是 1 位标志并且表示 Clip 的 CPI 类型, 如图 66 所示。

- 解释图 65 所示 CPI 句法中的 EP_map。有两种类型 EP_map, 即用于视
- 20 频流的 EP_map 和用于音频流的 EP_map。EP_map 中的 EP_map_type 区分开这些 EP_map 类型。如果 Clip 包含一个或者多个视频流, 则一定使用用于视频流的 EP_map。如果 Clip 不包含视频流但包含一个或多个音频流, 则一定使用用于音频流的 EP_map。

- 参照图 67 解释用于视频流的 EP_map。用于视频流的 EP_map 具有数据
- 25 stream_PID, PTS_EP_start 和 RSPN_EP_start。stream_PID 表示传输视频流的传输数据包的 PID。PTS_EP_start 表示从视频流序列标头(header)开始的存取单元的 PTS。RSPN_EP_start 表示源数据包的地址, 该源数据包包括由 AV 流中的 PTS_EP_start 引用的存取单元的第一字节。

- 叫做 EP_map_for_one_stream_PID()的子表是从一个视频流创建的, 该视
- 30 频流由具有相互相同 PID 的传输数据包传输。如果多个视频流存在于 Clip 中, 则 EP_map 可以包含多个 EP_map_for_one_stream_PID()。

用于音频流的 EP_map 具有数据 stream_PID、PTS_EP_start 和 RSPN_EP_start。stream_PID 表示传输音频流的传输数据包的 PID。

PTS_EP_start 表示在音频流中的存取单元的 PTS。RSPN_EP_start 表示源数据包的地址，该源数据包包括由 AV 流的 PTS_EP_start 引用的存取单元的第一 5 字节。

叫做 EP_map_for_one_stream_PID() 的子表是从一个音频流创建的，该音频流由具有相互相同 PID 的传输数据包传输。如果多个音频流存在于 Clip 中，则 EP_map 可以包含多个 EP_map_for_one_stream_PID()。

回到在 EP_map 和 STC_Info 之间的关系，一个

10 EP_map_for_one_stream_PID() 是在与 STC 中的非连续点无关的一个表中创建。将 RSPN_EP_start 的值与 STC_Info() 中定义的 RSPN_STC_start 的值进行比较，则显示出属于对应 STC_sequence 的 EP_map 的数据边界(见图 68)。

EP_map 一定具有对由相同 PID 传输的连续流范围的一个

EP_map_for_one_stream_PID。在图 69 所示的情况下，节目#1 和节目#2 具有 15 相同的视频 PID，但是，数据范围不是连续的，使得必须对每个程序一定提供 EP_map_for_one_stream_PID。

图 70 表示 EP_map 句法。通过对图 70 所示 EP_map 句法的解释，EP_type 是 4 位字段并且显示 EP_map 入口点类型，如图 71 所示。EP_type 表示在该 20 字段后面的数据字段的语义。如果 Clip 包括一个或多个视频流，则 EP_type 一定被设置成 0(‘视频’)。另外，如果 Clip 不包括视频流但包含一个或多个音频流，则 EP_type 一定被设置成 1(‘音频’)。

Number_of_stream_PID 的 16 位字段表示具有在 EP_map 中 Number_of_stream_PID 作为变量的循环的循环的次数。Stream_PID(k) 的 16 25 位字节表示传输数据包的 PID，该传输数据包传输由

EP_map_for_one_stream_PID(num_EP_entries(k)) 引用的数 k 基本流(视频或者 音频流)。如果 EP_type 是 0(‘视频’)，其基本流一定是视频流。如果 EP_type 等于 1(‘音频’)，其基本流一定是音频流。

num_EP_entries(k) 的 16 位字段表示由 EP_map_entries(k) 引用的 num_EP_entries(k)。EP_map_for_one_stream_PID_Start_address(k): 该 32 30 位字节表示相对地址位置，在此，EP_map_for_one_stream_PID (num_EP_entries(k)) 开始于 EP_map()。该值是由从 EP_map() 第一字节开始的

大小表示的。

Padding_word 一定根据 EP_map() 句法插入。X 和 Y 一定是可选择的正整数。对应填充字可以假设任意可选值。

图 72 表示 EP_map_for_one_stream_PID 的句法。通过对图 72 所示

- 5 EP_map_for_one_stream_PID 句法的解释, PTS_EP_start 的 32 位字段的语义不同于 EP_map() 定义的 EP_type。如果 EP_type 等于 0(‘视频’), 则该字段具有开始于视频流序列标头的存取单元的 33 位精密 PTS 的高 32 位。如果 EP_type 等于 1(‘音频’), 该该字段具有音频流存取单元的 33 位精密 PTS 的高 32 位。
- 10 RSPN_EP_start 的 32 位字段的语义不同于 EP_map() 定义的 EP_type。如果 EP_type 等于 0(‘视频’), 则该字段表示源数据包的相对地址, 源数据包包括由 AV 流中 PTS_EP_start 引用的存取单元序列标头的第一字节。另外, 如果 EP_type 等于 1(‘音频’), 则该字段表示源数据包的相对地址, 源数据包包括由 AV 流中 PTS_EP_start 引用的存取单元音频流中的第一字节。
- 15 RSPN_EP_start 是基于源数据包数作为单位定大小的, 并且是从 AV 流文件的第一源数据包开始以 ClipInfo() 中定义的 offset_SPN 的值为初始值计数。在该 AV 流文件中, 绝对地址是通过下式计算的, 即:

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}.$$

注意, 句法中的 RSPN_EP_start 的值一定以升序出现。

- 20 现在参照图 73 解释 TU_map。TU_map 形成基于源数据包到达时钟(到达时间基的计时)的时间轴。该时间轴叫做 TU_map_time_axis。
TU_map_time_axis 的原点由 TU_map() 中的 offset_time 表示。
TU_map_time_axis 从 offset_time 以预置单位被分割, 该单位叫做 time_unit。
在 AV 流的每个 time_unit 中, 第一完全形式之源数据包 AV 流文件上的
25 地址存储在 TU_map 中。这些地址叫做 RSPN_time_unit_start。在
TU_map_time_axis 上开始第 k 个 ($k \geq 0$) time_unit 的时间叫做 TU_start_time(k)。
该值是基于下述方程计算的:

$$\text{TU\_start\_time}(k) = \text{offset\_time} + k * \text{time\_unit\_size}.$$

注意, TU_start_time(k) 具有 45kHz 的精确度。

- 30 图 74 表示 TU_map 的句法。通过解释图 74 所示的 TU_map 的句法, offset_time 的 32 位字段给出与 TU_map_time_axis 相关的偏移时间。该值表

示与 Clip 中第一 time_unit 相关的偏移时间。offset_time 的大小是以从 27MHz 精确到达时钟作为单位得出的 45kHz 为基础的。如果 AV 流将作为新 Clip 进行记录, 则 offset_time 一定设置为 0。

- time_unit_size 的 32 位字段提供 time_unit 的大小, 并且是以从 27MHz 精确到达时钟作为单位得出的 45kHz 为基础的。最好是, time_unit_size 不长于 1 秒($\text{time\_unit\_size} \leq 45000$)。number_of_time_unit_entries 的 32 位字段表示存储在 TU_map() 中的入口数。

- RSN_time_unit_start 的 32 位字段表示 AV 流中每个 time_unit 开始的位置的相对地址。RSN_time_unit_start 是基于源数据包数作为单位的大小, 并且用从 AV 流文件第一源数据包开始的在 ClipInfo() 中定义的 offset_SPN 的值作为初始值计数。AV 流文件中的绝对地址是通过下式计算的:

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}.$$

- 注意, 句法循环中的 RSN_time_unit_start 的值一定要以升序出现。如果在数(k+1)的 time_unit 中没有源数据包, 则数(k+1)的 RSN_time_unit_start 一定等于数 k 的 RSN_time_unit_start。

通过解释图 45 所示的 zzzzz.clip 句法中的 ClipMark, ClipMark 是与 Clip 有关的标记信息, 并且存储在 ClipMark 中。该标记不是通过用户设置的, 但是是通过记录器(记录和/或再现设备 1)设置的。

- 图 75 表示 ClipMark 的句法。通过解释图 75 所示的 ClipMark 的句法, version_number 是四个字符字母, 其表示该 ClipMark 的版本号。version_number 一定要根据 ISO646 编码成“0045”。

- 长度是无符号 32 位整数, 其表明从长度字段之后直接到 ClipMark() 尾端的 ClipMark() 的字节数目。Number_of_clip_marks 是 16 位无符号整数, 其表示存储在 ClipMark 中的标记数并且可以等于 0。Mark_type 是 8 位字段, 其表示标记类型和根据图 76 所示的表被编码。

Mark_time_stamp 是 32 位字段和存储了表示具有专用标记的指针的时间戳。Mark_time_stamp 的语义不同于 PlayList() 中的 CPI_type, 如图 77 所示。

- 如果 CPI() 中的 CPI_type 表示 EP_map 类型, 则该 8 位字段表示放置有 Mark_time_stamp 的连续 STC 域的 STC_sequence_id。如果 CPI() 中的 CPI_type 表示 TU_map 类型, 则该 8 位字段没有意义但设置为 0。Character_set 的 8 位字段表示在 mark_name 字段中编码的字符字母的表示方法。该编码方法对应

于图 19 所示值。

Name_length 的 8 位字段表示在 mark_name 字段中所示的标记名称的字节长度。该 mark_name 字段表示标记名称。对应于 Name_length 数的从该字段左边开始的字节数是有效字符数和表示标记名称。在 mark_name 字段中，
5 在这些有效字符字母之后的值可以是任意的。

ref_thumbnail_index 的字段表示附加到标记的缩略图图象的信息。如果 ref_thumbnail_index 字段是不同于 0xFFFF 的值，该缩略图图象被附加到其标记，缩略图图象存储在 mark.thumb 文件中。该图象是使用 mark.thumb 文件中的 ref_thumbnail_index 之值引用的。如果 ref_thumbnail_index 字段是等于
10 0xFFFF 的值，则缩略图图象不附加到其标记。

图 78 表示代替图 75 的 ClipMark 的另一个句法。图 79 表示在这种情况下代替图 76 的典型 mark_type 表。Reserved_for_maker_ID 是 16 位字段，其表示制造者的制造者 ID，如果 mark_type 表示从 0xC0 到 0xFF 的值，则制造者定义该 mark_type。制造者 ID 是由 DVR 格式许可证发放者指定的。

15 mark_entry() 是表示指定标记点的点的信息。随后将详细地解释该句法。Representative_picture_entry 是表示代表由 mark_entry() 表示的标记的信息点的信息。Representative_picture_entry 的句法随后将详细地解释。

使用 ClipMark 的目的是使再现 AV 流的用户能够可视地恢复内容。DVR 播放器使用 GUI(图形用户界面)以将 ClipMark 信息提供给用户。对于
20 ClipMark 信息的视频显示，由 Representative_picture_entry 表示的图象而非由 mark_entry() 表示的图象将被表示出。

图 80 表示 mark_entry() 和 Representative_picture_entry 的例子。假设某个节目的节目名称(标题)是在从这种节目开始的某个时间之后被显示的，例如几秒之后。如果要创建 ClipMark，则 mark_entry() 可以置于节目的开始点，
25 Representative_picture_entry() 然后置于节目的节目名称(标题)的显示点。

如果 Representative_picture_entry 的图象显示在 GUI 上并且用户已经指定该图象时，则 DVR 播放器从 mark_entry 被置入的点开始重新播放。

图 81 表示 Representative_picture_entry() 和 mark_entry() 的句法。

Mark_time_stamp 是 32 位字段并且存储了表示由标记指定的点的时间戳
30 和表示代表由 mark_entry() 指定的标记的图象点的时间戳。

图 82 表示在通过 PTS 采用基于地址信息而非采用时间戳基信息的情况

下 mark_entry()和 Representative_picture_entry()的典型句法。

在 mark_entry()情况下, RSPN_ref_EP_start 表示源数据包的相对地址, 其表示用于解码 AV 流中标记点图象的流的入口点。在

- Representative_picture_entry()情况下, RSPN_ref_EP_start 表示源数据包的相对地址, 其代表用于解码表示由 mark_entry()表明的标记的图象的流的入口点。RSPN_ref_EP_start 的值一定要作为 RSPN_ref_EP_start 存储在 EP_map 中。而且, 对应于 RSPN_EP_start 的 PTS_EP_start 的值一定要在从标记点图象的 PTS 开始的过去中最近的 EP_map。

- Offset_nm_pictures 是 32 位字段和表示在显示序列中从由 RSPN_ref_EP_start 表示的图象到由标记点表示的图象的偏移图象数。该数是从 0 计数的。在图 83 实施例的情况下, Offset_nm_pictures 是 6。

图 84 表示在采用用于指定 ClipMark 的基于地址的信息的情况下, mark_entry()和 Representative_picture_entry()的另一个典型句法。

- 在 mark_entry()情况下, RSPN_mark_point 表示源数据包的相对地址, 其包括由 AV 流中标记引用的存取单元的第一字节。在 Representative_picture_entry()情况下, RSPN_mark_point 表示源数据包的相对地址, 其包括代表由 mark_entry()表示的标记的编码图象的第一字节。

RSPN_mark_point 具有源数据包数作为单位的大小和使用在 Clip 信息中定义的 offset_SPN 值从 AV 流文件的第一源数据包计数。

- 参照图 85, 解释 ClipMark 和 EP_map 之间的关系。在本实施例中, 假设 EP_map 指定 I0, I1 和 In 作为入口点的地址, 并且假设开始从这些地址继续序列标头的 I 图象。如果 ClipMark 指定 M1 作为某个标记的地址, 并且如果从该源数据包开始的图象是可解码的, 则其满足是否读出的数据从作为在 M1 地址之前和最靠近 M1 地址的入口点的 I1 初始化。

- MakerPrivateData 已经参照图 22 进行了说明, 因此在此不进行具体说明。

下面, 解释 thumbnail_information。缩略图图象存储在 menu.thmb 文件或者 mark.thmb 文件中。这些文件具有相同的句法结构, 并且拥有单个 Thumbnail()。该 menu.thmb 文件存储表示对应 PlayList 的图象。全部菜单缩略图存储在专有 menu.thmb 文件中。

- mark.thmb 文件存储标记缩略图文件, 即表示标记点的图象。对应于全部 PlayList 和 Clip 的全部标记缩略图存储在单个 mark.thmb 文件中。由于经

常加上或删除缩略图，附加和部分删除的操作一定要可容易和快速地执行。由于这个原因，Thumbnail()具有块结构。图象数据分成多个部分，每个都存储在一个 tn_block 中。一个图象数据存储在连续的 tn_block 中。在 tn_block 串中，可以存在没有使用的 tn_block。专有缩略图图象的字节长度是可变的。

- 5 图 86 表示 menu.thmb 文件和 mark.thmb 文件的句法，同时图 87 表示在 menu.thmb 文件和 mark.thmb 文件的句法中缩略图句法的句法。通过解释图 87 所示的 Thumbnail 的句法，version_number 是四个字符字母，其表示该 Thumbnail()的版本号。version_number 一定要根据 ISO646 编码成“0045”。

- 长度是无符号 32 位整数，其表明从长度字段之后直接到 Thumbnail()尾
10 端的 makers_private_data 的字节数目。Tn_block_start_address 是 32 位无符号整数，其用从 Thumbnail()引导字节开始的相对字节数作为单位表示第一个 tn_block 的引导字节地址。相对字节数是从 0 计数的。Number_of_thumbnais 是 16 位无符号整数，其提供包含在 Thumbnail()中的缩略图图象的入口数。

- Tn_block_size 是 16 位无符号整数，其提供具有 1024 字节作为单位的一个
15 个 tn_block。例如，如果 Tn_block_size=1，其表示一个 tn_block 的大小是 1024 字节。Number_of_tn_block 是 16 位无符号整数，其表示 Thumbnail()中 tn_block 的入口数。Thumbnail_index 是 16 位无符号整数，其表示由缩略图信息表示的缩略图图象的索引数，该缩略图信息对应于从该 Thumbnail_index 字段开始的一个循环。值 0xFFFF 一定不用作为 Thumbnail_index。Thumbnail_index 是
20 由 UIAppInfVolume()、UIAppInfoPlayList()、PlayListMark()和 ClipMark()中的 ref_thumbnail_index 引用。

Thumbnail_picture_format 是表示缩略图图象之图象格式的 8 位无符号整数和假设为图 88 所示的值。在表中，仅仅允许 DCF 和 PNG 在“menu_thumb”中。标记缩略图一定假设值“0x00”(MPEG-2 视频 I 图象)。

- 25 Picture_data_size 是 32 位无符号整数，其表示根据字节作为单位的缩略图图象的字节长度。Start_tn_block_number 是 16 位无符号整数，其表示开始缩略图图象数据的 tn_block 的 tn_block 数。缩略图图象数据的前端一定与 tn_block 的前端一致。tn_block 数开始于 0 并且与 tn_block 循环中的变量 k 值相关。

- 30 X_picture_length 是 16 位无符号整数，其代表缩略图图象之帧图象水平方向上的象素数。Y_picture_length 是 16 位无符号整数，其代表缩略图图象

之帧图象垂直方向上的象素数。tn_block 是其中存储了缩略图图象的区域。Thumbnail() 中 tn_block 的总数具有相同的大小(固定长度)并由 tn_block_size 在大小上定义的。

图 89A 和 89B 示意地表示缩略图图象数据是如何存储在 tn_block 中。正如 5 如图 89A 和 89B 所示, 对应缩略图图象数据是从 tn_block 的前端开始的, 并且, 如果图象数据具有超过 tn_block 的大小, 则下一个连续的 tn_block 用于存储。通过这样做, 具有变化大小的图象数据可以组织成固定长度, 使得诸如删除的编辑能够用较简单的处理来进行。

现在解释 AV 流文件。AV 流文件存储在“M2TS”目录中(图 14)。有 10 两种类型的 AV 流文件, 一个是 CLIP AV 流, 另一种是 Bridge-Clip AV 流。两种流一定具有如后面定义的 DVR MPEG-2 传输流文件结构。

首先, 解释 DVR MPEG-2 传输流。DVR MPEG-2 传输流的结构示于图 90。AV 流文件具有由 Aligned(联合)单元的整数构造的 DVR MPEG-2 传输流结构。Aligned 单元具有 6144 字节的大小(= 2048x3 字节)。Aligned 单元开始 15 于具有 192 字节长度的源数据包的第一字节。一个源数据包是由 TP_extra_header 和传输数据包构成的。TP_extra_header 是 4 字节长, 其中具有 188 字节长度的传输数据包。

一个 Aligned 单元是由 32 个源数据包构成的。DVR MPEG-2 传输流中最后 Aligned 单元也是由 32 个源数据包构成的。因此, DVR MPEG-2 传输流 20 束于 Aligned 单元的边界。如果记录在盘上的输入传输流的传输数据包数不是 32 的整数倍, 则具有空数据包的源数据包(PID 传输数据包=0x1FFF)一定用作最后 Aligned 单元。文件系统不允许将冗余信息加到 DVR MPEG-2 传输流。

图 91 表示 DVR MPEG-2 传输流的记录器模型。图 91 所示的记录器是用 25 于指定记录模型的概念性模型。DVR MPEG-2 传输流遵从该模型。

解释 MPEG-2 传输流的输入时序。输入 MPEG-2 传输流是完全传输流或者部分传输流。输入 MPEG-2 传输流一定遵守 ISO/IEC13818-1 或者 ISO/IEC13818-9。MPEG-2 传输流的数 i 字节在时间 t(i)同时输入到在 ISO/IEC13818-1 中指定的 T-STD 传输流系统目标解码器 51 和源打包器 54 两 30 者。Rpk 是传输数据包输入速率的瞬时最大值。

27 MHz PLL 52 产生 27MHz 频率的时钟。27MHz 频率的时钟锁定在

MPEG-2 传输流的 PCR(节目始终基准)的值。到达时间时钟计数器 53 是二进制计数器, 其计数 27MHz 频率的脉冲。Arrival_time_clock(i)是到达时间时钟计数器 53 在时间 t(i)的计数值。

源打包器 54 将 TP_extra_header 附加到整个传输数据包以创建源数据包。

- 5 Arrival_time_stamp 表示传输数据包的第一字节到达 T-STD51 和源打包器 54 两者的时间。Arrival_time_stamp(k)是 Arrival_time_clock(k)的样本值, 正如下面的方程所示:

$$\text{Arrival\_time\_clock}(k) = \text{Arrival\_time\_clock}(k) \% 230$$

这里 k 表示传输数据包的第一字节。

- 10 如果两个连续的输入传输数据包的时间间隔达到不小于 230/27000000 秒 (大约 40 秒), 则两个传输数据包的 Arrival_time_clock(k)之间的差应当设置为 230/27000000 秒。记录器为此情况提供。

- 平滑缓冲器 55 平滑输入传输流的位速率。平滑缓冲器 55 不允许溢出。当平滑缓冲器 55 非空时 Rmax 是来自平滑缓冲器 55 的源数据包位速率。如果平滑缓冲器 55 是空的, 则平滑缓冲器 55 的输出位速率为 0。

下面解释 DVR MPEG-2 传输流的记录器模型的参数。Rmax 的值是由与 AV 流文件相联系的 ClipInfo()中定义的 TS_recording_rate 给出的。该值可以根据下述方程计算:

$$R_{\max} = \text{TS\_recording\_rate} * 192 / 188$$

- 20 这里 TS_recording_rate 的值是基于字节/秒为单位的大小。

- 如果输入传输流是 SESF 传输流, 则 Rpk 一定等于与 AV 流文件相联系的 ClipInfo()中定义的 TS_recording_rate。如果输入传输流不是 SESF 传输流, 则在诸如 smoothing_buffer_descriptor、short_smoothing_bugger_descriptor 或者 partial_transport_stream_descriptor 的 MPEG-2 传输流的说明符中定义的值可以
25 以被涉及为平滑缓冲器 55 的大小。

- 如果输入传输流是 SESF 传输流, 则平滑缓冲器 55 的大小(平滑缓冲器大小)为 0。如果输入传输流不是 SESF 传输流, 则在诸如 smoothing_buffer_descriptor、short_smoothing_bugger_descriptor 或者 partial_transport_stream_descriptor 的 MPEG-2 传输流的说明符中定义的值可以
30 以引用为平滑缓冲器 55 的大小。

对于记录器和再现单元, 需要使用足够大小的缓冲器。缺省缓冲器大小

为 1536 字节。

现在解释 DVR MPEG-2 传输流的播放器模型。图 92 表示 DVR MPEG-2 传输流的播放器模型。这是用于指定重新播放过程的概念性模型。DVR MPEG-2 传输流遵守该模型。

- 5 27MHz X-tal(晶体振荡器)61 产生 27MHz 频率。27MHz 频率的误差范围一定是 $\pm 30\text{ppm}(27000000 \pm 810\text{Hz})$ 。到达时间时钟计数器 62 是二进制计数器，其计数 27MHz 的脉冲。Arrival_time_clock(i)是到达时间时钟计数器 62 在时间 $t(i)$ 的计数值。

- 10 在平滑缓冲器 64 中，Rmax 是当平滑缓冲器 64 非空时源数据包对平滑缓冲器 64 的输入位速率。如果平滑缓冲器 64 是满的，则对平滑缓冲器 64 的输入位速率是 0。

- 通过解释 MPEG-2 传输流的输出时序，如果当前源数据包的 Arrival_time_stamp 等于 Arrival_time_clock(i)的 30LSB 位的值，则源数据包的传输数据包从平滑缓冲器 64 中提取出。Ppk 是传输数据包速率的瞬时最大值。
- 15 平滑缓冲器 64 不允许溢出。

DVR MPEG-2 传输流的播放器模型的参数与上述 DVR MPEG-2 传输流的记录器模型的参数相同。

- 图 93 表示源数据包的句法，同时 transport_packet()是在 ISO/IEC 13818-1 中指定的 MPEG-2 传输数据包。图 94 表示图 93 所示的源数据包句法中的 TP_extra_header。通过解释图 94 所示的 TP_extra_header 的句法，
- 20 copy_permission_indicator 是整数，表示传输数据包的有效负载的拷贝限制。拷贝限制可以设置为免费拷贝、不再拷贝、一次拷贝或者禁止拷贝。图 95 表示 copy_permission_indicator 的值和这些值代表的模式之间的关系。

- copy_permission_indicator 附加到整个传输数据包。如果输入传输流要使用 IEEE1394 数字接口进行记录的话，则 copy_permission_indicator 的值可以与 IEEE1394 等时数据包标头中的 EMI(加密模式指示符)的值相有关。如果记录输入传输流而不使用 IEEE1394 数字接口，则 copy_permission_indicator 的值可以与设置在传输数据包中的 CCI 的值相有关。在自身编码模拟信号输入中，copy_permission_indicator 的值可以与模拟信号的 CGMS-A 的值相有关。
- 25 注意，Arrival_time_stamp 是由下述方程中的 Arrival_time_stamp 指定的整数值。
- 30

$\text{Arrival\_time\_stamp}(k) = \text{Arrival\_time\_clock}(k) \% 230.$

在定义 Clip AV 流时, Clip AV 流一定具有上述定义的 DVR MPEG-2 传输流的结构。Arrival_time_clock(i)一定在 Clip AV 流中连续地增加。如果系统时间基(STC 基)的非连续点应当存在于 Clip AV 流中, Clip AV 流的

5 Arrival_time_clock(i)一定连续地增加。

在 Clip AV 流的开始和结束之间的 Arrival_time_clock(i)之差的最大值一定是 26 小时。该限制保证如果在 MPEG-2 传输流中没有系统时间基(STC 基)的非连续点, 在 Clip AV 流中会出现相同值的 PTS(显示时间戳)。MPEG-2 系统标准提供 PTS 的环绕时间应当是 233/90000 秒(大约 26.5 小时)。

10 在定义 Bridge-Clip AV 流中, Bridge-Clip AV 流一定具有上述定义的 DVR MPEG-2 传输流的结构。Bridge-Clip AV 流一定包括一个到达时间基非连续点。在到达时间基非连续点的开头和后面的传输流一定遵守如后面要解释的关于编码的限制, 并且也一定遵守后面要解释的 DVR-STD。

15 本实施例提供在编辑中的 PlayItem 之间视频和音频的无缝连接的无缝支持。在 PlayItem 之间的无缝连接保证给播放器/记录器和“无缝解码”的“连续数据提供”。“连续数据提供”意味着以确保禁止缓冲器下溢所必须位速率向解码器提供数据文件系统的能力。“连续数据提供”根据足够大小的块连续地保证数据存储, 以保证实时特性和从以盘为单位的数据读出。

20 “无缝解码”意味着在不产生解码器重新播放输出的暂停或者间隙的情况下, 播放器在显示记录于盘上的音频/视频数据的能力。

解释由无缝连接的 PlayItem 引用的 AV 流。不管在在先 PlayItem 和当前 PlayItem 之间的连接如何被保证, 使无缝显示能够从当前 PlayItem 中定义的连接_condition 字段中被验证。有在采用 Bridge-Clip 的 PlayItem 之间的无缝连接的方法和在不采用 Bridge-Clip 的 PlayItem 之间无缝连接的方法。

25 图 96 表示在先 PlayItem 和当前 PlayItem 之间的关系。图 96 表示由 Clip1(Clip AV 流)的阴影(shaded)流数据和在 Bridge_arrival_time_discontinuity 之前的阴影流数据构成的 TSI。

TSI 的 Clip1 的阴影流数据是从用于解码与在先 PlayItem 的 IN_time(由图 96 中的 IN_time 所示)相联系的显示单元所必须的流地址到由

30 RSPN_exit_from_previous_Clip 引用的源数据包的流数据。在包含于 TSI 的 RSPN_arrival_time_discontinuity 之前的阴影流数据是从 Bridge_Clip 的第一源

数据包到由 RSPN_arrival_time_discontinuity 引用的源数据包的流数据。

另一方面,图 96 的 TS2 是由 Clip2(Clip AV 流)的阴影流数据和在 Bridge_Clip 的 PN_arrival_time_discontinuity 之后的阴影流数据构成的。在包含于 TS2 的 Bridge_Clip 的 RSPN_arrival_time_discontinuity 之后的阴影流数据是从由 RSPN_enter_to_current-Clip 引用的源数据包到用于解码与当前 PlayItem 的 OUT_time(图 96 中 OUT_time2 所示)相联系的显示单元所要求的流的地址的流数据。

图 97 表示在不采用 Bridge_Clip 的情况下在在先 PlayItem 和当前 PlayItem 之间的关系。在这种情况下,从播放器中读出的流数据表示为阴影。图 97 中, TS1 的 TS1 阴影流数据是开始于用于解码与在先 PlayItem 的 IN_time(由图 97 中的 IN_time1 所示)相联系的显示时间所必须的流地址和延伸到 Clip1 的最后源数据包的数据。另一方面,图 97 中的 TS2 是由 Clip2(Clip AV 流)的阴影流数据构成的。

TS2 的 Clip2 的阴影流数据是开始于 Clip2 的第一源数据包和结束于用于解码与当前 PlayItem 的 OUT_time(由图 97 中的 OUT_time2 所示)相联系的显示单元所必须的流的地址的流数据。

图 96 和 97 中, TS1 和 TS2 是源数据包的连续流。接着,考虑 TS1 和 TS2 的流指定和其间的连接条件。作为传输流编码结构的限制,包含在 TS1 和 TS2 中的视频流的数一定是 1。包含在 TS1 和 TS2 中的音频流的数一定是 2 或小于 2。包含在 TS1 和 TS2 中的音频流的数一定彼此相等。注意, TS1 和/或 TS2 可以包含除上述那些之外的基本流或者私有流。

现在解释视频位流的限制。图 98 表示在图象显示序列中表示的无缝连接的例子。为了使视频流可以被无缝地表示,在 OUT_time(Clip1 的 OUT_time)之后和 IN_time2(Clip2 的 IN_time)之前表示的不需要图象一定要通过重新编码靠近结点的 Clip 的部分流的过程来去除。

图 99 表示使用图 98 所示的情况下的 BridgeSequence 来实现无缝连接的例子。在 RSPN_arrival_time_discontinuity 之前的 Bridge-Clip 的视频流是由到达与图 98 的 Clip1 的 OUT_time1 相联系的图象的编码视频流构成的。该视频流连接到在先 Clip1 的视频流并且重新编码,以便证明符合 MPEG-2 标准的一个连续基本流。

类似地方式,在 RSPN_arrival_time_discontinuity 之后的 Bridge-Clip 的视

频流是由从与图 98 的 Clip2 的 IN_time2 相对应的图象开始的编码视频流构成的。该视频流能够开始正确地解码，连接到下一个随后的 Clip2 的视频流和重新编码，以便变成符合 MPEG-2 标准的一个连续基本流。对于创建 Bridge-Clip，几个图象通常需要被重新编码，同时其它图象能够从初始 Clip 中拷贝。

图 100 表示在图 98 所示实施例情况下在不使用 BridgeSequence 时实现无缝连接的情况。Clip1 的视频流是由在与图 98 的 OUT1_time1 相联系图象内的编码视频流构成的，其被重新编码从而变成一个符合 MPEG-2 标准的连续基本流。类似地，Clip2 的视频流是由在与图 98 的 Clip2 的 IN_time2 相联系的图象之后的编码视频流构成的，其被重新编码从而变成一个符合 MPEG-2 标准的连续基本流。

通过解释视频流的编码限制，TS1 和 TS2 视频流的帧速率一定要彼此相等。TS1 视频流一定在 sequence_end_code 结束，而 TS2 视频流一定开始于 sequence_header，其是 GOP 标头和 I 图象中。TS2 视频流一定开始于封闭 GOP。位流中定义的视频表示单元(帧或者字段)一定是连续的，其间具有结点。在结点处，顶部/底部字段序列一定是连续的。在采用 3-2 下拉(3-2 pulldown)编码的情况下，重写“top_field_first”和“repeat_first_field”是必须的，同时局部重新编码也可以用于防止发生字段间隙。

通过解释音频位流的编码限制，TS1 的音频取样频率和 TS2 的音频取样频率一定要彼此相等。TS1 和 TS2 的诸如 MPEG1 层 2，AC-3，SESF，LPCM 和 AAC 的音频编码方法一定彼此相等。

通过解释 MPEG-2 传输流上的编码限制，TS1 音频流的最后音频帧一定包含音频样本，其具有等于在 TS1 显示图象的显示结束时间的显示时间。TS2 音频流的第一音频帧一定包含音频样本，其等于在 TS2 第一显示图象之显示开始时刻的样本。

在结点处，在音频表示单元的序列中一定没有间隙。如图 101 所示，可以有由音频表示单元的长度定义的重叠，其不大于两个音频帧域。传输 TS2 基本流的第一数据包一定是视频数据包。在结点处的传输流一定遵守随后要解释的 DVR-STD。

通过解释 Clip 和 Bridge-Clip 的限制，其中 TS1 和 TS2 一定不包含到达时间基的非连续点。

下述限制仅仅适用于使用 Bridge-Clip 的情况。Bridge-Clip AV 流在 TS1 的最后源数据包和 TS2 的第一源数据包之间的结点处仅仅具有一个到达时间基非连续点。在 ClipInfo()中定义的 RSPN_arrival_time_discontinuity 一定说明非连续点的地址，其又一定说明引用 TS2 第一源数据包的地址。

- 5 由在 BridgeSequenceInfo()中定义的 RSPN_exit_from_previous_Clip 引用的源数据包可以是 Clip 中的任何源数据包。该源数据包是 Aligned 单元的边界是不必要的。由在 BridgeSequenceInfo()中定义的 RSPN_exit_from_current_Clip 引用的源数据包可以是 Clip 中的任何源数据包。该源数据包不必是 Aligned 单元的边界。

- 10 通过解释 PlayItem 的限制，在先 PlayItem 的 OUT_time(图 96 和 97 所示的 OUT_time1)一定表示 TS1 的最后显示单元的显示结束时间。当前 PlayItem 的 IN_time(图 96 和 97 所示的 IN_time2)一定表示 TS2 第一视频表示单元的显示开始时间。

- 15 参照图 102，通过解释使用 Bridge-Clip 中的数据分配的限制，一定要创建无缝连接以便保证由文件系统提供的连续数据。这一定是通过排列连接到 Clip1(Clip AV 流文件)和 Clip2(Clip AV 流文件)的 Bridge-Clip AV 流完成的，目的是满足数据分配指定。

- 20 一定要选择 RSPN_exit_from_previous_Clip 以便在 RSPN_exit_from_previous_Clip(Clip AV 流文件)之前的 Clip1(Clip AV 流文件)的流部分将排列在不小于半段(half fragment)的连续区域中。一定要选择 Bridge-Clip AV 流的数据长度以便其被排列在不小于半段的连续区域中。一定要选择 RSPN_enter_to_current_Clip 使得在 RSPN_enter_to_current_Clip 之后的 Clip2(Clip AV 流文件)的流部分将被排列在不小于半段的连续区域中。

- 25 通过解释在不使用 Bridge-Clip 时无缝连接情况下数据分配的限制，一定要创建无缝连接以便连续数据提供将由文件系统保证。这一定是通过排列 Clip1(Clip AV 流文件)的最后部分的第一部分和 Clip2(Clip AV 流文件)的第一部分实现的，目的是满足数据分配指定。

- 30 Clip1(Clip AV 流文件)的第一个流部分一定要配置在不短于一个半段的连续区域中。Clip2(Clip AV 流文件)的第一个流部分一定要配置在不小于一个半段的连续区域中。

下面，解释 DVR-STD。DVR-STD 是概念性模型，用于模仿在产生和验

证 DVR MPEG-2 传输流中的解码处理。另一方面, DVR-STD 是概念性模型, 用于模仿在产生和验证由上述无缝连接的两个 PlayItem 引用的 AV 流中的解码处理。

图 104 表示 DVR-STD 模型。图 104 所示的模型包含作为构成单元的 DVR MPEG-2 传输流播放器模型。诸如 n , T_{bn} , M_{bn} , E_{bn} , T_{bsys} , B_{sys} , R_{xn} , R_{bxn} , R_{xsys} , D_n , D_{sys} , O_n 和 $P_n(k)$ 的符号是与在 ISO/IEC13818-1 的 T-STD 中定义的符号相同。即, n 是基本流的索引数, T_{bn} 是基本流 n 的传输缓冲器。

M_{bn} 是基本流 n 的多路复用缓冲器, 并且只出现在视频流中。 E_{bn} 是基本流 n 中的基本流缓冲器。其只对视频流出现。 T_{bsys} 是用于正在解码的节目中的系统信息的输入缓冲器。 B_{sys} 是用于正在解码的节目的系统信息的系统目标解码器中的主缓冲器。 R_{bxn} 是数据从 M_{bn} 中去除的传输速率并且只对视频流出现。

R_{xsys} 是数据从 T_{bsys} 中去除的传输速率。 D_n 是基本流 n 中的解码器。 D_{sys} 是与正在解码的节目的系统信息有关的解码器。 O_n 是视频流中的重新排序缓冲器。 $P_n(k)$ 是基本流 n 中的数 k 显示单元。

解释用于 DVR-STD 的解码过程。在专有 DVR MPEG-2 传输流再现的时间期间, 传输数据包输入到 T_{bsys} 缓冲器的时序是由源数据包的 `arrival_time_stamp` 确定的。对解码操作和显示操作的指定是与 ISO/IEC 13818-1 中的指定相同。

解释在再现无缝连接 PlayItem 同时的解码过程。这里, 解释由无缝连接 PlayItem 引用的两个 AV 流的再现。下述说明中, 解释例如图 96 所示的前述 TS1 和 TS2 的再现。同时, TS1 和 TS2 分别是在先流和当前流。

图 105 表示当从所给 AV 流(TS1)传送到无缝连接的下一个 AV 流(TS2)时用于输入, 解码和显示传输数据包的时序图。在从预置 AV 流(TS1)传送到无缝连接的下一个 AV 流(TS2)时间期间, (图 105 中 STC2 表示的)TS2 到达时间基的时间轴是与(图 105 中 STC1 表示的)TS1 到达时间基的时间轴不相同。

而且, (图 105 中 STC2 表示的)TS2 系统时间基的时间轴与(图 105 中 STC1 表示的)TS1 系统时间基的时间轴不同。要求视频显示是无缝连接的, 同时在音频表示单元的显示时间中可以有重叠。

解释对 DVR-STD 的输入时序。直到时间 T_1 , 即直到 TS1 的最后视频数

据包完全输入到 DVR-STD 的 TB1 为止的时间, 对 DVR-STD 的 TB1, Tbn 或者 Tbsys 的输入时序是由 TS1 源数据包的 arrival_time_stamp 确定的。

- TS1 的剩余数据包一定以 TS_recording_rate(TS1)的位速率输入到 Tbn 或者 Tbsys 的缓冲器, 这里 TS_recording_rate(TS1)是在与 Clip1 相联系的
- 5 ClipInfo()中定义的 TS_recording_rate 的值。TS1 的最后字节输入到缓冲器的时间是时间 T2。因此, 在从时间 T1 到时间 T2 的域中, 源数据包的 arrival_time_stamp 减少(discount)。

- 如果 N1 是在 TS1 最后视频数据包之后的传输数据包字节数, 从时间 T1 到时间 DT1 的时间 DT1 是 N1 字节被以 TS_recording_rate(TS1)的位速率完全
- 10 输入所要求的时间, 并且是根据下述方程计算的:

$$DT1=T2-T1=N1/TS\_recording\_rate(TS1).$$

在从时间 T1 到 T2 的时间期间, Rxn 和 Rxsys 的值两者都变成了 TS_recording_rate(TS1)的值。除该规则之外的缓冲器操作是与 T-STD 的相同。

- 在时间 T2, 到达时间时钟计数器被复位到 TS2 第一源数据包的
- 15 arrival_time_stamp 的值。TB1, Tbn 或者 Tbsys 对 DVR-STD 的输入时序是由 TS2 源数据包的 arrival_time_stamp 确定的。Rxn 和 Rxsys 两者都变成在 T-STD 中定义的值。

- 通过解释附加音频缓冲器和系统数据缓冲器, 除了在 T-STD 中定义的缓冲器量之外, 要求音频解码器和系统解码器具有剩余缓冲器量(近似一秒等效
- 20 数据量)。

通过解释视频显示时序, 视频表示单元的显示一定要通过结点以没有间隙的方式连接。注意, STC1 是 TS1 系统时间基的时间轴(图 105 中的 STC1 所示), 而 STC2 是 TS2 系统时间基的时间轴(图 97 中的 STC2 所示)。具体地说, STC2 开始于第一 PCR 输入到 T-STD 的时间点。

- 25 STC1 和 STC2 之间的偏移按如下确定: 如果 PTS1end 是与 TS1 最后视频表示单元相联系的 STC1 上的 PTS, 而 PTS2start 是与 TS2 第一视频表示单元相联系的 STC2 上的 PTS, 同时 Tpp 是 TS1 最后视频表示单元的显示周期, 则在两系统时间基之间的偏移 STC_delta 可以根据下述方程计算:

$$STC\_delta=PTS1end+Tpp-PTS2start.$$

- 30 通过解释音频显示, 可以有音频表示单元之显示时序的重叠, 该重叠可以是 0 到小于 2 个音频帧(见图 105 所示的“音频重叠”)。应当选择的音频样

本和音频表示单元的显示对结点后面的校正时间基的重新同步都设置在播放器上。

通过解释 DVR-STD 的系统时间时钟, TS1 最后音频表示单元在时间 T5 上被显示。系统时间时钟可以在时间 T2 和时间 T5 之间重叠。在该域期间,

- 5 DVR-STD 在旧时间基值(STC1)和 t 新时间基值(STC2)之间转换。STC2 的值可以根据下述方程计算:

$$\text{STC2} = \text{STC1} - \text{STC\_delta}$$

解释缓冲器连续性。STC1video_end 是当 TS1 的最后视频数据包的最后字节到达 DVR-STD 的 TB1 时系统时间基 STC1 上的 STC 值, 而

- 10 STC2video_end 是计算到关于系统时间基 STC2 的值的值。STC2video_end 是通过下述方程计算的:

$$\text{STC2video\_end} = \text{STC1video\_end} - \text{STC\_delta}$$

为了遵守 DCR-STD, 要求下面两个条件。第一, DVR-STD 的 TB1 上的第一视频数据包的第一字节的到达时序一定要满足下述不等式:

- 15 $\text{STC22video\_end} > \text{STC21video\_end} + \Delta T1$.

如果为了满足该不等式, Clip1 和/或 Clip2 的部分流将重新编码和/或重新多路复用, 则其根据需要进行。

- 第二, 在其中 STC1 和 STC2 转换到相同时间轴上的系统时间基的时间轴上, 来自 TS1 的视频数据包输入和来自 TS2 的下一个视频数据包输入一定
20 既不上溢出又不下溢出视频缓冲器。

基于该句法、数据结构和规则, 记录在记录介质上的内容或者重新播放的信息能够被适当地管理, 使得用户能够验证适当地记录在记录介质上的数据的内容或者容易地再现所希望的数据。

- 尽管本实施例是将 MPEG-2 传输流作为多路复用流来作为例子解释的,
25 但这仅仅是示例性的, 本实施例可以应用到 MPEG2 节目流或者用在美国 DirecTV 服务(商标)的 DSS 传输流。

参照图 106 的流程图解释在 mark_entry()和 representative_picture_entry()的句法构造成图 81 所示的情况下定位由标记点表示的场景再生的过程。

- 首先, 在步骤 S1, 记录和/或再现设备 1 的控制器 23 从记录介质 100 中
30 读出 EP_map(图 70), STC_Info(图 52)和 ClipMark(图 78)作为 DVR 传输流的数据库。

在步骤 S2, 控制器 23 列出来自由 `representative_picture_entry()`(图 81)或者 `ref_thumbnail_index` 所引用的图象的缩略图表以将如此形成的表从作为用户接口输入/输出之端子 24 中输出, 以用于在 GUI 菜单上显示。如果在这种情况下, `ref_thumbnail_index` 具有有效值, 则 `ref_thumbnail_index` 相对于
5 `representative_picture_entry` 是优先排序的。

在步骤 S3, 用户指定再现开始点的标记点。这是通过用户从作为 GUI 显示的菜单屏幕上选择缩略图图象实现的。控制器 23 响应这个选择操作以获得与该指定的缩略图相联系的标记点。

在步骤 S4, 控制器 23 获得 `mark_Time_stamp` 的 PTS 和在步骤 S3 指定
10 的 `mark_entry` 的 `STC_sequence_id`(图 81)。

在步骤 S5, 控制器 23 从 `STC_Info`(图 52)中获得源数据包数, 在此开始与在步骤 S4 获得的 `STC_sequence_id` 相联系的 STC 时间轴。

在步骤 S6, 控制器 23 从在此开始在步骤 S5 要求的 STC 时间轴的数据包数和从在步骤 S4 要求的标记点的 PTS 中获得源数据包数, 在这里有时间
15 上在标记点 PTS 之前和最靠近标记点 PTS 的入口点(I 图象)。

在步骤 S7, 控制器 23 从包含步骤 S6 要求的入口点之源数据包数中读出传输流的数据以将所读出数据发送到 AV 记录器 27。

在步骤 S8, 控制器 23 控制 AV 记录器 27 以启动从步骤 S4 获得的标记点 PTS 的图象开始的显示。

20 上述操作参照图 107 到 109 进行进一步地解释。

假设将 CM(商业广告)插入在 DVR 传输流文件中的 DV 传输流具有 `STC_sequence_id = id = id0` 的 `STC_EP_start` 的 STC 时间轴, 如图 107 所示, 并且假设时间轴开始于此的源数据包数小于源开始点 A 的源数据包数。还假设 CM(商品广告)被加在源数据包数 B 和 C 之间。

25 在这种情况下, 寄存有与表示成 `RSPN_EP_start` 的作为 PTS(A)、PTS(B)和 PTS(C)的对应 PTS 之 A、B 和 C 相联系的 `EP_map`, 其对应于图 70 所示 `EP_map`, 如图 108 所示。

参照图 109, 在图 78 的 `ClipMark` 中记录有与表示场景开始、CM 开始和 CM 结束的 0x92、0x94 和 0x95 之标记类型值(图 79)相联系的 `mark_entry` 和
30 `representative_picture_entry`, 如图 109 所示。

作为 `mark_entry` 的 `mark_time_stamp`、PTS(a1)、PTS(b0)和 PTS(c0)与场

景开始、CM 开始和 CM 结束相联系而被寄存，其中对应 STC_sequence_id 为 id0。

类似地，作为 representative_picture_entry 的 mark_time_stamp、PTS(a2)、PTS(b0)和 PTS(c)与场景开始、CM 开始和 CM 结束相联系而被寄存，其中对应 STC_sequence_id 为 id0。

如果 $PTS(A) < PTS(a1)$ ，在步骤 S6 获得数据包数 A。在步骤 S7，开始于数据包数 A 的传输流提供给 AV 解码器 27，并且在步骤 S8，显示从 PTS(a1)开始初始化。

参照图 110 的流程图，在 mark_entry 和 representative_picture_entry 的句法构造为图 81 所示的情况下，CM 的处理跳步再现。

在步骤 S21，控制器 23 从记录介质 100 中读出 EP_map(图 70)，STC_Info(图 52)，Program_Info(图 54)和 ClipMark(图 78)。在步骤 S22，用户从作为用户接口输入/输出的端子 24 中指定 CM 跳步再现。

在步骤 S23，控制器 23 获得作为 CM 开始点(0x94)和相关的 STC_sequence_id 之标记信息的 PTS(图 81)。

在步骤 S24，控制器 23 从 STC_Info(图 52)中获得 CM 开始和结束点的源数据包数，在此开始对应于 STC_sequence_id 的 STC 时间轴。

在步骤 S25，控制器 23 使传输流从记录介质中读出，以将其发送到 AV 解码器 27 开始解码。

在步骤 S26，控制器 23 验证当前显示图象是否是 CM 开始点之 PTS 的图象。如果当前显示图象不是 CM 开始时间点的图象，则控制器处理到步骤 S27 继续显示该图象。然后处理返回步骤 S25 以重复随后的步骤。

如果在步骤 S26，当前显示图象被验证是 CM 开始点之 PTS 的图象，则处理进行到步骤 S28，在此控制器 23 控制 AV 解码器 27 以暂停解码和显示。

在步骤 S29，控制器 23 获得在此开始与 CM 结束点的 STC_sequence_id 相联系之时间轴的数据包数，同时从该数据包数和由在步骤 S23 处理所获得的 CM 结束点的 PTS 中获得其中在时间上在该点的 PTS 之前和最靠近该点 PTS 之入口点的源数据包数。

在步骤 S30，控制器 23 控制 AV 解码器 27 以重新初始化来自 CM 结束点的 PTS 图象的显示。

在步骤 S31，控制器 23 控制 AV 解码器 27 以重新初始化来自在 CM 结

束点的 PTS 图象的显示。

参照图 107 到 109, 进一步解释上述操作。CM 开始时间点和 CM 结束点在实施例中出现于 STC_sequence_id=id0 的公共时间轴上。选择在此开始 STC 时间轴的源数据包数为小于场景开始点的源数据包数 A。

- 5 如果传输流被解码, 并且在步骤 S26, 发现显示时间是 PTS(b0), 即如果显示时间点发现是 CM 开始时间点, 则显示被 AV 解码器 27 暂停。如果 $PTS(C) < PTS(c0)$, 在步骤 S30, 从该流开始解码, 解码开始于数据包数 C 的数据。在步骤 S31, 显示是从 PTS(c0) 的图象重新开始。

- 10 该方法能够不仅应用于 CM 跳步再现而且能够应用于在通常由 ClipMark 指定的两点之间的场景的跳步再现。

参照图 112 的流程图, 解释在 mark_entry 和 representative_picture_entry 为图 82 所示句法的情况下由标记点指明的 CM 定位(locating)再现处理。

在步骤 S41, 控制器 23 获得 EP_map(图 70), STC_Info(图 52), Program_Info(图 54)和 ClipMark(图 78)的信息。

- 15 在步骤 S42, 控制器 23 产生来自由 representative_picture_entry()(图 82)引用的图象或者由包含在 ClipMark(图 78)中的 ref_thumbnail_index 所引用的图象的缩略图表, 在步骤 S41 读出, 以将如此产生的表显示在菜单屏幕上。如果 ref_thumbnail_index 具有有效值, 则 ref_thumbnail_index 相对于 representative_picture_entry 是优先排序的。

- 20 在步骤 S43, 用户指定重新播放开始点的标记点。该种指定是例如通过用户从由步骤 S42 处理所显示的屏幕上选择缩略图图象以指定与该缩略图相联系的标记点来实现的。

在步骤 S44, 控制器 23 获得由步骤 S43 处理所指定的标记点之 RSPN_ref_EP_start 和 offset_num_pictures(图 82)。

- 25 在步骤 S45, 控制器 23 使传输流数据从对应于在步骤 S44 获得的 RSPN_ref_EP_start 的源数据包数中读出, 以将其送到 AV 解码器 27。

在步骤 S46, 控制器 23 控制 AV 解码器 27 以从由 RSPN_ref_EP_start 引用的(没有显示)图象开始计数要显示的图象并启动从对应于 offset_num_pictures 计数值的图象开始的显示。

- 30 参照图 113 到 115, 进一步解释上述处理。在本实施例中, 场景是在 DVR 传输流中启动的, 从而 CM 从源数据包数 B 到源数据包 C 插入。这样, PTS(A)、

PTS(B)和 PTS(C)作为 RSPN_EP_start 寄存在 EP_map 中,与 PTS_EP_start 的 A、B 和 C 相关,如图 114 所示。

而且, mark_entry 和 representative_picture_entry 与场景开始、CM 开始和 CM 结束相联系地寄存,如图 115 所示。在 mark_entry 中, A、B 和 C 寄存为 RSPN_ref_EP_start,与场景开始、CM 开始和 CM 结束相联系,同时 M1、N1 和 N2 寄存为 offset_num_pictures。类似地,在 representative_picture_entry 中, M2、N1 和 N2 寄存为 offset_num_pictures,与场景开始、CM 开始和 CM 结束相联系,同时 M1、N1 和 N2 寄存为 offset_num_pictures。

如果命令定位再现为从对应于场景开始的图象,解码是从数据包数 A 数据开始的流进行,以计数要显示的从没有显示的图象 PTS(A)开始的图象和启动从其 offset_num_pictures 已经假设为 M1 值之图象开始的显示。

现在参照图 116 的流程图解释在 mark_entry 和 representative_picture_entry 的句法是图 82 所示结构的情况下 CM 跳步再现的处理。

在步骤 S61,控制器 23 获得关于 EP_map(图 70), STC_Info(图 52), Program_Info(图 54)和 ClipMark(图 78)的信息。

如果在步骤 S62 用户命令 CM 跳步再现,控制器 23 在步骤 S63 获得与 CM 开始点和 CM 结束点一样的对应点标记信息的 RSPN_ref_EP_start 和 offset_num_pictures(图 82)。CM 开始点数据是 RSPN_ref_EP_start(1)和 offset_num_pictures(1),而 CM 结束点数据是 RSPN_ref_EP_start(2)和 offset_num_pictures(2)。

在步骤 S64,控制器 23 获得对应于来自 EP_map 的 RSPN_ref_EP_start(1)和 RSPN_ref_EP_start(2)的 PTS(图 70)。

在步骤 S65,控制器 23 使传输流从记录介质 100 中读出以将其发送到 AV 解码器 27。

在步骤 S66,控制器 23 验证当前显示图象是否是 RSPN_ref_EP_start(1)和 RSPN_ref_EP_start(2)相联系的 PTS 的图象,并且,如果当前显示图象不是与 RSPN_ref_EP_start(1)相联系的 PTS 的图象,则其处理到步骤 S67 以使图象被连续地显示。然后控制器 23 返回步骤 S65 以重复随后的处理。

如果在步骤 S66,当前显示图象是与 RSPN_ref_EP_start(1)相联系的 PTS 的图象,则控制器 23 处理到步骤 S68 以控制 AV 解码器 27 计数从与

RSPN_ref_EP_start(1)相联系的 PTS 的图象开始显示的图象, 当计数值是 offset_num_pictures(1)时暂停该显示。

在步骤 S69, 控制器 23 使传输流读出, 其开始于 RSPN_ref_EP_start(2)的源数据包数, 以将所读出数据发送到 AV 解码器 27。

- 5 在步骤 S70, 控制器 23 计数要显示的从与 RSPN_ref_EP_start(2)相关的 PTS 的(没有显示)图象开始的图象, 并且当计数值是 offset_num_pictures(2)时, 控制器使显示在对应图象处开始。

通过参照图 113 到图 115 进一步解释上述操作, 分别对应于数据包数 B 和 C 的时间 PTS(B)和 PTS(C)首先基于 EP_map 获得(图 114)。Clip AV 流被解
10 码, 并且当该值是 N1 时(图 15), 暂停显示。

解码是从开始于数据包数 C 之数据的流重新启动的, 以计数要显示的从 PTS(C)图象开始的图象, 并且当该值是 N2 时(图 115), 显示从该图象开始被重新启动。

- 上述处理不仅可以适用于 CM 跳步再现的情况, 也可以适用于具有在由
15 ClipMark 指定的两点之间跳步的场景的再现。

参照图 118 的流程图解释在 mark_entry 和 representative_picture_entry 基本流的句法是如图 84 所示情况下定位由标记点指定的场景和再现如此定位的场景的处理。

- 在步骤 S81, 获得关于 EP_map(图 70), STC_Info(图 52), Program_Info(图
20 54)和 ClipMark(图 78)的信息。

在步骤 S82, 控制器 23 产生由 representative_picture_entry 或者由 ClipMark(图 78)的 ref_thumbnail_index 所引用的缩略图表, 以将如此产生的表显示为 GUI 菜单图象。如果 ref_thumbnail_index 具有有效值, 则 ref_thumbnail_index 相对于 representative_picture_entry 是优先排序的。

- 25 在步骤 S83, 用户指定再现开始点的标记点。该指定是通过用户从菜单图象选择缩略图图象并且指定与该缩略图相联系的标记点完成的。

在步骤 S84, 控制器 23 获得由用户指定的 RSPN_mark_point(图 84)。

在步骤 S85, 控制器 23 从 EP_map 获得在 mark_point 的 RSPN_mark_point 之前和最靠近的入口点的源数据包数。

- 30 在步骤 S86, 控制器 23 从与在步骤 S85 获得的入口点相联系的源数据包数中读出传输流的数据, 以将如此读出的数据送到 AV 解码器 27。

在步骤 S87, 控制器 23 控制 AV 解码器 27 启动从由 RSPN_mark_point 引用的图象开始的显示。

参照图 119 到 121 更详细地解释上述处理。在本实施例中, DVR 传输流文件是这种文件, 其中场景开始于源数据包并且 CM 被插入在从源数据包数
5 B 到源数据包数 C。这样, PTS_EP_start 被寄存作为与图 20 的 RSPN_EP_start 一样的与 A,B 和 C 相关的 PTS(A)、PTS(B)和 PTS(C)。在图 21 所示的 ClipMark 中, a1、b1 和 c1 寄存为制造者入口的 RSPN_mark_point, 同时 a2、b1 和 c1 寄存为 representative_picture_entry 的 RSPN_mark_point, 其分别与场景开始、CM 开始和 CM 结束相联系。

10 在用于再演的场景开始处定位图象时, 如果数据包数 A 是使得 $A < a1$, 则解码是从开始于数据包数 A 之数据的流开始的, 使得显示是从对应于源数据包数 a1 的图象开始。

现在参照图 122 和 123 的流程图解释在 mark_entry 和 representative_picture_entry 的句法是如图 84 所示情况下的 CM 跳步再现处理。

15 在步骤 S101, 控制器 23 获得关于 EP_map(图 70)、STC_Info(图 52)、Program_Info(图 54)和 ClipMark(图 78)的信息。

在步骤 S102, 用户指定 CM 跳步再现。

在步骤 103, 控制器 23 获得其标记类型(图 79)是 CM 开始点或者 CM 结束点的每个点标记信息的 RSPN_mark_point(图 84)。控制器 23 将 CM 开始点
20 的数据和 CM 结束点的数据分别设置为 RSPN_mark_point(1)和 RSPN_mark_point(2)。

在步骤 S104, 控制器 23 使传输流从记录介质 100 中读出以输出该用于解码的读出的流。

在步骤 S105, 控制器 23 验证当前图象是否是对应于 RSPN_mark_point(1)的图象。如果当前图象不是对应于 RSPN_mark_point(1)的图象, 则控制器 23
25 处理到步骤 S106 以继续显示该图象。然后, 处理返回到步骤 S11 以重复随后的处理。

如果在步骤 S105 当前显示图象已经被证实是与 RSPN_mark_point(1)相联系的图象, 则处理转移到步骤 S107, 在此控制器 23 控制 AV 解码器 27 暂
30 停解码和显示。

然后, 在步骤 S108, 在 RSPN_mark_point(2)之前和最靠近的源数据包

数是从 EP_map 获得的(图 70)。

在步骤 S109, 控制器 23 从与在步骤 S108 获得的入口点相联系的源数据包数中读出传输流数据以将其送到 AV 解码器 27。

- 在步骤 S110, 控制器 23 控制 AV 解码器 27 重新启动从由 RSPN_
- 5 mark_point(2)引用的图象开始的显示。

通过参照图 119 到 121 的实施例更详细地解释上述处理, ClipAV 流被解码和显示在对应于源数据包数 b1 的显示图象处暂停(图 21)。如果源数据包数 C<源数据包数 c1, 解码在开始于数据包数 C 之数据的流中重新启动, 并且显示被在开始于数据包数 c1 之数据的流中重新启动。

- 10 正如上述, 预置定位在 PlayList 上是通过时间戳指定的, 该时间戳在每个 Clip 的每个 Clip 信息中转换为对 ClipAV 流的预置位置的访问的数据地址。

具体地说, 如果书标记或者恢复点由用户指定为与时间轴上的时间戳一样的 PlayList 标记, 则再现的 PlayList 可以使用由 PlayList 引用的 ClipMark 的 ClipMark 以访问 ClipAV 流的场景开始或者结束点。

- 15 同时, ClipMark 句法可以是图 126 所示, 而不是图 78 所示。

在本实施例中, RSPN_mark 插入在图 78 的 reserved_for_MakerID 、 mark_entry() 和 representative_picture_entry()的位置。该 RSPN_mark 的 32 位字段表示包含标记正在引用的存取单元之第一字节的源数据包的相对地址。RSPN_mark 是根据源数据包数作为单位来定大小的。其是在从 AV 流文件第一源数据包开始的 Clip 信息文件中定义的并且以 offset_SPN 的值作为初始值计数的。

20

否则该结构与图 78 的结构相同。

- ClipMark 句法还可以构造成图 127 所示。在本实施例中, RSPN_ref_EP_start 和 offset_num_pictures 插入在图 126 的 RSPN_mark 的位置。
- 25 这些类似于图 82 中所示的。

图 128 是用于说明在编码和记录模拟 AV 信号过程中, 图 81 中所示句法的 ClipMark 的准备。参考图 1 所示的记录和/或再现设备 1 的方框图, 在步骤 S200, 分析单元 14 分析来自端子 11, 12 的输入 AV 信号以检测特征点, 该特征点规定起因于 AV 流内容的特征场景, 例如节目定位点或者场景变化点

30 在步骤 S201, 控制器 23 获得特征点图象的 PTS。

在步骤 S202, 控制器 23 在 ClipMark 中存储关于特征点的信息。具体地

说, 存储在本实施例 ClipMark 的语义和句法方面解释的信息。在步骤 S203, Clip 信息文件和 Clip AV 流文件被记录在盘上。

图 129 用于说明在记录从数字接口输入的传输流的过程中, 图 81 中所示句法的 ClipMark 的准备。参考图 1 所示的记录和/或再现设备 1 的方框图, 在
5 步骤 S211, 多路分用器 26 和控制器 23 获得要被记录的节目的基本流 PID。如果有多个要被分析的基本流, 则获得全部的基本流 PID。

在步骤 S212, 多路分用器 26 分离出来自从端子 13 输入的传输流节目中的基本流, AV 解码器 27 将该基本流解码成 AV 信号。在步骤 S213, 分析单元 14 分析 AV 信号以检测特征点。

10 在步骤 S214, 控制器 23 获得特征点图象的 PTS 和其所属的 STC 的 STC_sequence_id。具体地, 控制器 23 存储在本实施例中的 ClipMark 的句法和语义中解释的信息。

在步骤 S216, Clip 信息文件和 Clip AV 流文件被记录在盘上。

如图 128 和 129 的流程图所示, AV 流文件, 即用于存储 Clip AV 流文件中
15 中指示特征图象的标记的 ClipMark, 被记录在 AV 流文件中的管理信息数据文件中, 即被记录在 Clip 信息文件中。

图 130 是用于说明实 PlayList 的准备的流程图。参考图 1 的记录和/或再现设备 1 的方框图进行说明, 在步骤 S221, 控制器 23 记录 Clip AV 流。在步骤 S222, 控制器 23 明确表示包括转换 Clip 的全部可再现范围的 PlayItem 的
20 PlayList()。如果在 Clip 中存在 STC 非连续点, 并且 PlayList()是由两个或多个 PlayItem 组成时, 则还确定 PlayItem 之间的 connection_condition。

在步骤 S223, 控制器 23 创建 UIAppInfoPlayList()。在步骤 S224, 控制器 23 形成 PlayListMark。在步骤 S225, 控制器 23 形成 MakersPrivateData。在步骤 S226, 控制器 23 记录实 PlayList 文件。

25 所以, 每次都创建一个实 PlayList 文件, Clip AV 流被重新记录。

图 131 是用于说明虚 PlayList 的形成的流程图。在步骤 S231, 通过用户接口确定一个记录在该盘上的实 PlayList 的再现。从该实 PlayList 的再现范围中, 确定由 IN 和 OUT 指示的再现域。

在步骤 S232, 控制器检验由用户指定的再现范围的处理是否已经全部完
30 成。如果在步骤 S232 检验出由用户指定的再现范围的处理还没有全部完成, 处理返回到步骤 S231 重复后续的处理。如果在步骤 S232 检验出该操作已经

全部完成, 则处理转到步骤 S233。

在步骤 S233, 由用户通过用户接口或由处理器 23 确定两个连续再现的域之间的连接条件(connection_condition)。如果用户无法产生 sub-path, 则跳过步骤 S234 的处理。

5 在步骤 S235, 控制器 23 根据由用户指定的再现范围信息和 connection_condition 形成 PlayList()。在步骤 S236, 控制器 23 形成 UIAppInfoPlayList()。在步骤 S237, 控制器 23 形成 PlayListMark。在步骤 S238, 控制器 23 形成 MakerPrivateData。在步骤 S239, 控制器 23 将使虚 PlayList 文件记录在盘上。

10 从记录在盘上的实 PlayList 的重放范围中, 用户选择他或她希望观看的重放范围, 并且从一组到另一组的集中分组的重放范围中形成一个虚 PlayList。

图 132 是用于说明 PlayList 的再现的流程图。还要参考图 1 的表示记录和/或再现设备 1 的方框图。在步骤 S241, 控制器 23 获得 Info.dvr、Clip 信息
15 文件、PlayList 文件和关于缩略图(thumbnail)文件的信息以形成一个表示在盘上记录的 PlayList 的表从而在 GUI 上显示。

在步骤 S242, 用户命令控制器 23 通过用户接口再现一个 PlayList。在步骤 S243, 控制器 23 从 IN_time 的 PTS 和当前 PlayItem 的 STC-sequence-id 中获得一个具有暂时在先的、并且最接近 IN_time 的进入点的源数据包号。在
20 步骤 S244, 控制器 23 从源数据包号中读出 AV 流的数据以将这样读出的数据传送到 AV 解码器 27。

如果在上述的 PlayItem 之前该 PlayItem 被暂时再现, 则在步骤 S245 控制器 23 管理控制从而根据 connection_condition 执行对 PlayItem 的显示的连接处理。在步骤 S246, AV 解码器 27 从 IN_time 的 PTS 的图象开始显示。

25 在步骤 S247, AV 解码器 27 连续解码 AV 流。在步骤 S248, 控制器 23 检查当前显示的图象是否是 OUT_time 的 PTS 的图象。如果在步骤 S248 确定当前显示的图象是 OUT_time 的 PTS 的图象, 则处理转到步骤 S250, 否则, 转到步骤 S249。

在步骤 S249, 执行对检验将为 PTS 图象的图象进行显示的处理。接着处理返回到步骤 S247 以重复后续的处理。另一方面, 在步骤 S250, 控制器 23
30 检查 PlayItem 是否是当前 PlayList 中的最后一个, 并且, 如果该 PlayItem 被

验证将是当前 PlayList 中的最后一个, 则终止图 132 的流程图的处理。如果该 PlayItem 被验证将不是当前 PlayList 中的最后一个, 则处理返回到步骤 S243 以重复后续的处理。

图 133 是用于说明 PlayListMark 的准备的流程图。参考图 1 的表示记录
5 和/或再现设备 1 的方框图。

在步骤 S262, 用户命令控制器 23 再现一个 PlayList。在步骤 S263, 再现单元 3 开始再现指定的 PlayList, 如参考图 132 的流程图所做的说明。

在步骤 S264, 用户通过用户接口命令控制器 23 设置喜爱的场景的标记。在步骤 S265, 控制器 23 获得标记的 PTS 和其所属的 PlayItem 的 PlayList_id。

10 在步骤 S266, 控制器 23 将标记信息存储在 PlayListMark() 中。在步骤 S267, 在盘上记录 PlayList 文件。

在这种方式中, 为存储指定再现 Clip 的 ClipMark 中的恢复点的标记, 将用户从 PlayList 的再现范围中指定的标记点, 或该 PlayListMark 记录在 PlayList 文件上。

15 图 134 是用于说明使用由 PlayList 参考得到的 Clip 的 ClipMark 和在再现该 PlayList 过程中的 PlayListMark 进行的定位再现。ClipMark() 的句法在图 81 中示出。参考图 1 的表示记录和/或再现设备 1 的方框图进行说明。

在步骤 S271 中, 控制器 23 通过用户接口获得 Info.dvr、Clip 信息文件、PlayList 文件和关于缩略图文件的信息以创建一个表示在盘上记录的 PlayList
20 的表从而在 GUI 61 上显示。

在步骤 S272, 用户通过用户接口命令一个 PlayList 的再现。在步骤 S273, 控制器 23 通过用户接口在 GUI 上显示从由 PlayList 参考的 Clip 的 ClipMark 所参考的图象中产生的缩略图的表。

在步骤 S274, 通过用户将再现开始点的标记点指定到控制器 23。在步骤
25 S275, 控制器 23 检验在步骤 S274 通过处理选择的标记是否是在 PlayListMark 中存储的标记。如果在步骤 S275 检验的所选择的标记是将存储在 PlayListMark 中的标记, 则处理转到步骤 S276。否则处理转到步骤 S278。

在步骤 S276, 控制器 23 获得该标记的 PTS 以及其所属的 PlayItem_id。在步骤 S277, 控制器 23 获得由 PlayItem_id 指示的 PlayItem 所参考的 AV 流
30 的 STC-Sequence-id。

在步骤 S278, 控制器 23 根据 STC-Sequence-id 和标记的 PTS 使 AV 流输

入到 AV 解码器 27。具体地, 使用 STC-Sequence-id 和标记的 PTS, 执行与图 132 的流程图的步骤 S243 和 S244 相似的处理。在步骤 S279, 再现单元 3 从标记的 PTS 的图象开始显示。

当再现 PlayList 时, 如参考图 9 中的说明, 能够参考由 PlayList 参考的
5 存储在 Clip 的 ClipMark 中的标记。因此, 如果一个 Clip 被实 PlayList 或多个虚 PlayList 所参考, 则这些 PlayList 可以共有该 Clip 的 ClipMark, 从而能够有效地管理标记数据。

如果在 PlayList 中定义将 PlayListMark 和 ClipMark 组合在一起, 而不需要定义该 Clip 中的 ClipMark, 并且如果一个 Clip 被实 PlayList 或多个虚
10 PlayList 所参考, 则各个 PlayList 具有相同内容的 ClipMark 信息, 因此恶化了数据记录效率。

图 135 表示 PlayListMark() 的另一个典型句法。Length 表示从紧接该长度字段之后的字节至 PlayListMark() 的最后字节之间的字节数。
Number_of_PlayList_mark 表示在 PlayListMark 中存储的标记的入口数。

15 mark_invalid_flag 是一个位的标志, 并且如果这个值被设置成 0, 即指示这个标记具有有效信息。如果这个值被设置成 1, 则指示这个标记是有效的。

如果用户在用户接口上已经进行了擦除标记入口的操作, 则记录和/或再现设备 1 可以将 mark_invalid_flag 的值改变成 1 而不从 PlayListMark 中擦除标记的入口。

20 mark_type 表示标记类型并具有图 136 中所示的意义。mark_name_length 表示在 mark_name 字段中指示的标记名字的字节长度。该字段的值不大于 32。ref_to_PlayItem_id 表示当输入标记时由 PlayItem 指定的 PlayItem_id 的值。在对应于给定的 PlayItem 的 PlayList() 中定义的值是在 PlayList() 中被定义的。

mark_time_stamp 存储指示由该标记指定的点的时间戳。
25 mark_time_stamp 指定在由 ref_to_PlayItem_id 指定的 PlayItem 中定义的 IN_time 和 OUT_time 中指定的再现范围中的时间。时间戳的意义与图 44 中的相同。

如果 entry_ES_PID 被设置成 0xFFFF, 则该标记是一个在为 PlayList 使用的基本流总体所共有的公共时间轴上的指针。如果 entry_ES_PID 被设置成
30 其他的不是 0xFFFF 的值, 则 entry_ES_PID 表示包含由该标记指定的基本流的传输包的 PID 的值。

ref_thumbnail_index 表示附加在该标记上的缩略图的信息。该信息的意义与 ref_thumbnail_index 相同。mark_name 表示标记名字。由来自该字段的左边的 mark_name_length 定义的字节数是有效的字符字母，并表示该名字。由 UIAppInfoPlayList 中的 character_set 指定的方法编码该字符字母。

- 5 在 mark_name 字段中的该有效字符字母的下一个字节的值可以是任何值。在这种句法的情况下，标记可以表示指定的基本流。例如，如果 PlayList 参考一个其中具有多个视频流的多画面(muti-view)节目，则 entry_ES_PID 用于设置表示该节目中一个视频流的视频 PID。

- 10 假设用户正在再现该参考多画面节目的 PlayList，并且用户正在观看该多画面的一个画面。还假设用户已经发送了一个跳过该再现至下一个标记点的命令。在这种情况下，记录和/或再现设备 1 应该使用与该用户当前看到的画面的视频 PID 的值相同的 entry_ES_PID 的标记。该记录和/或再现设备不应该任意改变画面。该记录和/或再现设备 1 可以使用在 entry_ES_PID 被设置成 0xFFFF 的标记。在这种情况下，记录和/或再现设备 1 不任意改变该画面。
- 15 面。

图 137 表示了图 81 中所示的句法的 ClipMark()的改进。Length 表示从紧接该长度字段之后的字节至 ClipMark()的最后字节之间的字节数。如果 mark_type 指定从 0x60 至 0x7F 中的一个值，则 maker_ID 表示已经定义了 mark_type 的制造商的 ID。

- 20 numer_of_Clip_mark 表示在 ClipMark 中存储的标记的入口数。mark_invalid_flag 是一个一位的标志，如果该值被设置成 0 或 1，则它分别表示该标记具有有效信息或该标记是无效的。

- 如果用户在用户接口上已经进行了擦除标记入口的操作，则记录和/或再现设备 1 可以将 mark_invalid_flag 的值改变成 1 而不从 PlayListMark 中擦除
- 25 标记的入口。mark_type 指示标记类型并具有图 138 中所示的意义。

ref_to_STC_id 表 示 指 定 mark_time_stamp 和
representative_picture_time_stamp 所位于的 STC-sequence 的 STC-sequence-id。
STC-sequence-id 的值在 STCInfo()中被定义。mark_time_stamp 具有在图 81 中
mark_entry()的情况下的 mark_time_stamp 的相同的意义。

- 30 如果 entry_ES_PID 被设置成 0xFFFF，则该标记是一个在 Clip 中的基本流总体所共有的时间轴上的指针。如果 entry_ES_PID 被设置成其他的不是

0xFFFF 的值, 则 entry_ES_PID 表示包含由该标记指定的基本流的传输包的 PID 的值。

ref_thumbnail_index 表示附加在该标记上的缩略图的信息。该信息的意义与图 78 的 ref_thumbnail_index 相同。representative_time_stamp 具有在图 81 中 representative_time_stamp 的情况下的 mark_time_stamp 的相同的意义。

在图 137 中所示的句法的情况中, 该标记可以指示一个特定的基本流。例如, 如果 Clip 包含具有一个节目中的多个视频流的多画面节目, 则 entry_ES_PID 用于设置指定该节目中一个视频流的视频 PID。

假设用户正在再现该参考多画面节目的 PlayList, 并且用户正在观看该多画面的一个画面。还假设用户已经向记录和/或再现设备 1 发送了一个跳过该再现至下一个标记点的命令。在这种情况下, 记录和/或再现设备 1 应该使用与该用户当前看到的画面的视频 PID 的值相同的 entry_ES_PID 的标记。而该记录和/或再现设备不允许任意改变画面。该记录和/或再现设备 1 可以使用用于将 entry_ES_PID 设置成 0xFFFF 的标记。在这种情况下, 记录和/或再现设备 1 不允许任意改变该画面。

根据上述句法、数据结构和规则, 能够适当管理记录在记录介质 100 上的数据的内容或重放信息以便相当方便地再现所需数据。

使用上述的数据结构, 例如 PlayList 文件和 Clip 信息文件被彼此分离, 并以这种形式被记录。因此, 如果预设的 PlayList 或 Clip 的内容通过编辑被改变, 就不需要改变与该文件无关的其他文件。结果, 文件内容能够被方便地改变, 并且对于这样的改动或记录是能够在较短的时间内完成的。

如果读取了初始的 Info.dvr, 则盘记录内容被呈现在用户接口, 并且仅从盘中读出用户命令再现的 PlayList 文件和有关于此的 Clip 信息文件, 可以减少用户的排队时间。

如果全部 PlayList 文件或 Clip 信息文件被形成一个文件, 并被记录在一个文件中, 则文件大小增加很多, 并且因此在用于记录的改变文件内容中的时间比在分开记录文件的情况下要多得多。通过使用本发明可以防止这样低效率是发生。

如上所述, AV 流文件, 即用于存储指定 AV 流文件中特征图象的标记的 ClipMark, 被记录在 AV 流的管理信息数据文件中, 而由用户从 PlayList 的再现范围中指定的标记点, 即具有关于由 AV 流中指定的域的组合所定义的再

现序列的信息的对象，或用于存储指示在再现该对象中的恢复点的标记的 PlayListsMark，被存储在该对象上。

通过这样做，在再现该 PlayList 中能够参考存储在由 PlayList 参考的 Clip 的 ClipMark 中的标记。因此，当一个 Clip 被实 PlayList 或多个虚 PlayList
5 参考时，这些 PlayList 可以共有一个 ClipMark，从而能够有效地管理标记数据。

如果在 Clip 中没有定义 ClipMark，但仅在 PlayList 中定义将 PlayListMark 和 ClipMark 组合在一起，一个 Clip 被实 PlayList 或多个虚 PlayList 参考的情况下，则各个 PlayList 具有相同内容的 ClipMark 信息，那么数据效率是低的。
10 可以通过使用本发明防止这种情况的发生。

如果用于将入口点的地址存储为 AV 流辅助信息的 EP_mark 和用于存储标记点的图象类型的 ClipMark，诸如节目定位点和图象的 AV 流中的地址，被放在一个文件中，并以这种形式记录在记录介质 100 上，则能够适当管理再现 AV 文件的所必须的流的编码信息。

15 通过这个 Clip 信息文件信息，用户能够从记录介质 100 中记录的 AV 流中搜索感兴趣的场景，诸如节目定位点，因此方便了响应于用户的随机访问的命令或特定再现的命令，从记录介质 100 中确定 AV 流的读出位置，并使流的解码立即开始。

上述操作顺序不仅能够在硬件上而且也能在软件上执行。在这种情况下，
20 记录和/或再现设备 1 能够由诸如图 139 中所示的个人计算机来形成。

参照图 139，CPU(中央处理单元)201 根据存储在 ROM(只读存储器)202 中存储的程序或从存储单元 208 载入到 RAM(随机存取存储器)203 中的程序执行各种处理操作，其中存储了在又 CPU 201 将要执行的各种操作中所必须的 RAM 数据。

25 CPU 201、ROM 202 和 RAM 203 通过总线 204 互联，该总线还连接到输入/输出接口 205。

与输入/输出接口 205 连接的有：包括键盘和鼠标的输入单元 206、由 CRT 和 LCD 组成的显示器、诸如扬声器的输出单元 207、诸如硬盘的存储器装置 208 和例如由调制解调器或终端适配器组成的通信单元 209。该通信单元 209
30 通过网络执行通信处理。

与输入/输出接口 205 连接的有：驱动器 210、磁盘 221、光盘 222、磁光

盘 223 或如果必须, 半导体存储器, 从所必须的存储器装置 208 中读出的计算机程序被安装。

上述操作顺序不仅能够在硬件上而且也能在软件上执行。如果操作顺序将在软件上执行, 则形成软件的程序或多个程序被安装在计算机的专用硬盘上, 使得该程序被从记录介质中安装到从诸如通用个人计算机中。

参照图 139, 除了计算机之外, 该记录介质可以由所分配的用于将程序提供给用户的包介质构成。包介质例如可以是磁盘 221(包括软盘)、光盘 222(包括 CD-ROM(致密只读存储器))和 DVD(数字视盘)、磁光盘 223(包括 MD(小型盘))或半导体存储器 224。此外, 该记录介质是通过配置给用户的硬盘所构成, 它被预先装载在计算机上, 该计算机包括具有其中所存储的程序的 ROM 202 或存储器 208。

在本说明书中, 说明由介质提供的程序的顺序的各步骤不仅包括根据所说明的顺序依次执行的处理, 还包括并行或批处理中所执行的处理。

本说明书中的系统表示多个设备组成的整个设备。

15

工业的可应用性

在根据本发明的信息处理设备和方法以及程序中, 产生作为监管 AV 流的管理信息的由执行从输入 AV 流中提取的特征图象的标记所形成的 ClipMark, 以及产生指向由用户从对应于定义了 AV 流中预设域的组合的 PlayList 的再现域中可选指定的图象的标记所形成的 PlayListMark, 同时 ClipMark 和 PlayListMark 分别记录在记录介质上的单独表中。因此, 能够方便快捷地存取 AV 流的所需位置。

而且, 在根据本发明的信息处理设备和方法以及程序中, 读出用于监管包括由指向从输入 AV 流中提取的特征图象的标记所形成的 ClipMark 的 AV 流的管理信息, 并且从对应于定义了 AV 流中预设域的组合的 PlayList 的再现域中读出指向由用户可选指定的图象的标记所形成的 PlayListMark。而且, 呈现在读取步骤中读出的管理信息和使用 PlayListMark 的信息, 从呈现装置所呈现的信息中参考对应于用户命令将被再现的 PlayList 的 ClipMark, 并且在参考步骤被参考的包括 ClipMark 的 AV 流被从对应于 ClipMark 的位置中再现。因此能够方便快捷地存取 AV 流的所需位置。

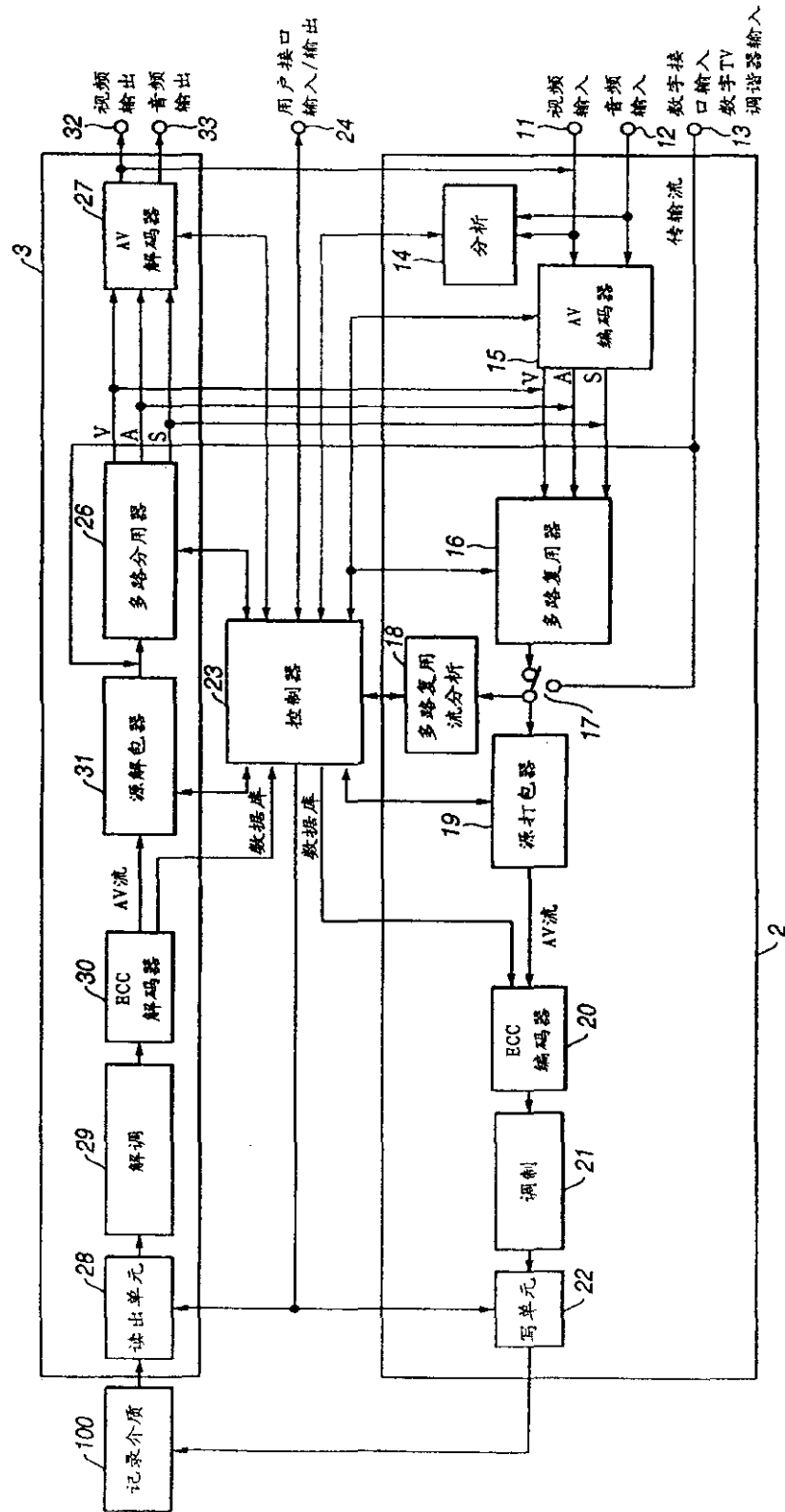


图 1

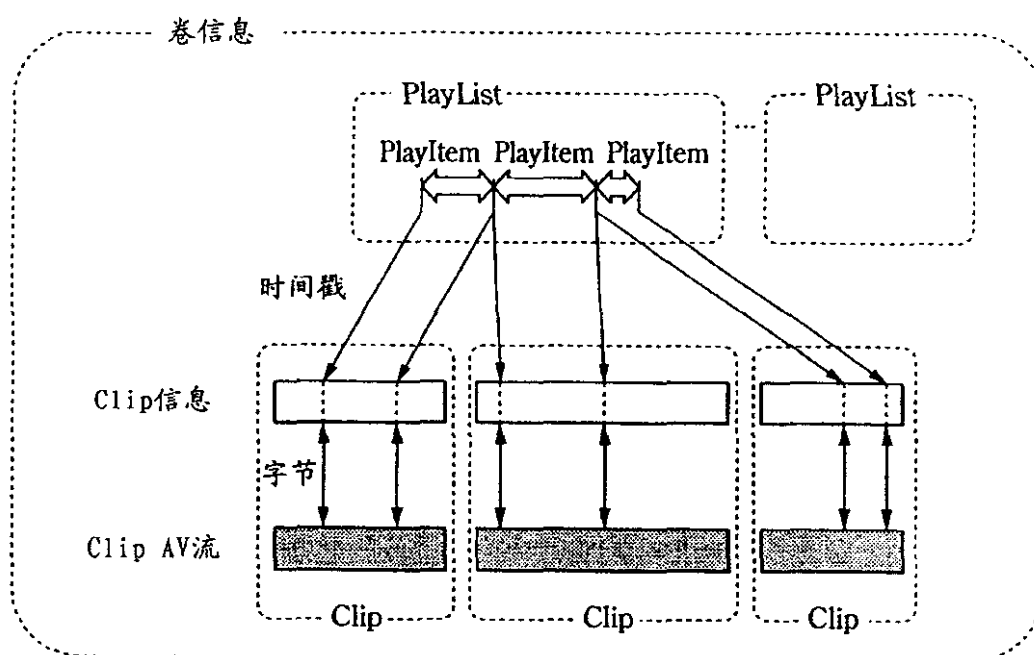


图 2

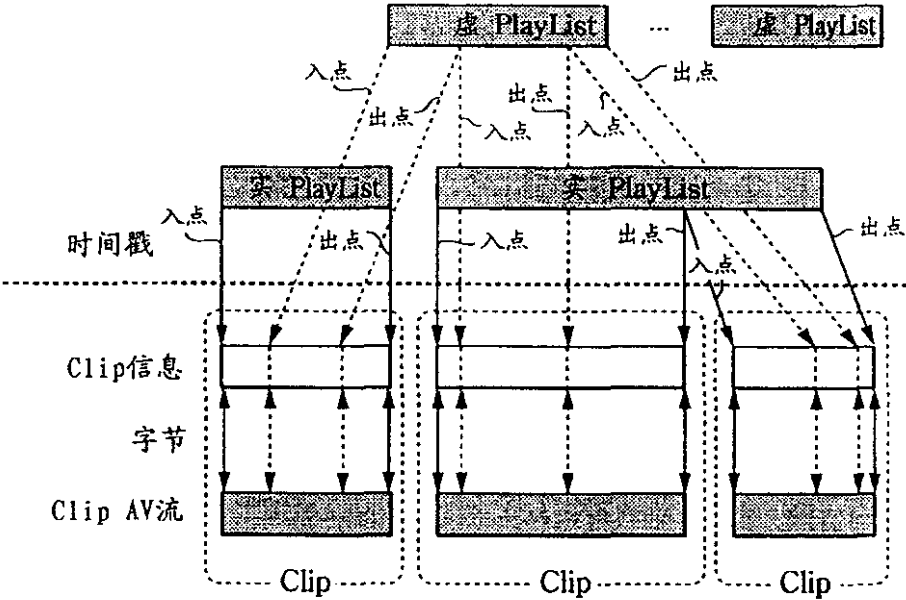


图 3

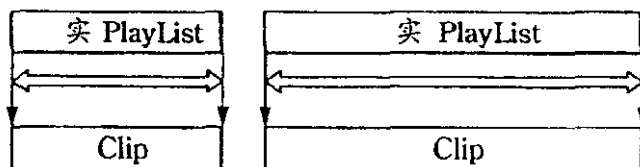


图 4A

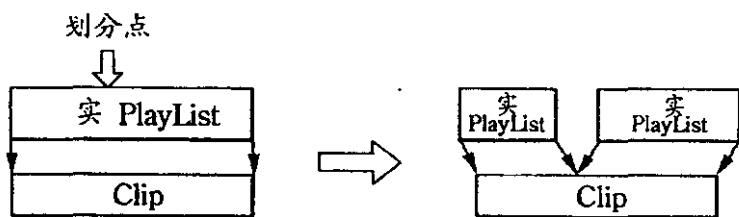


图 4B

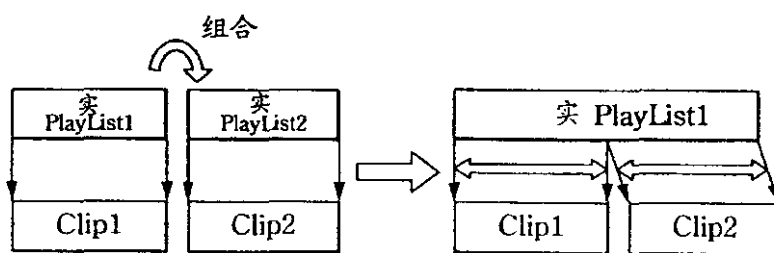


图 4C

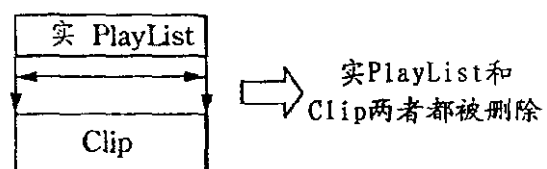


图 5A

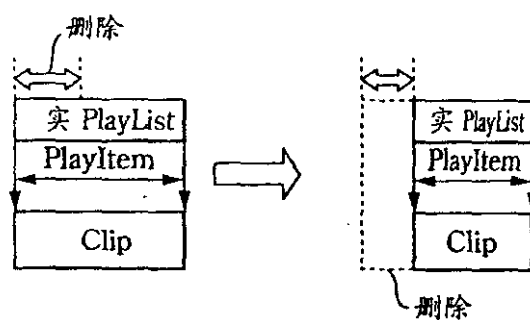


图 5B

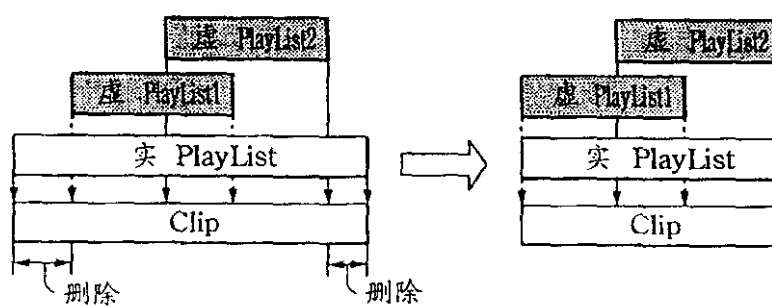
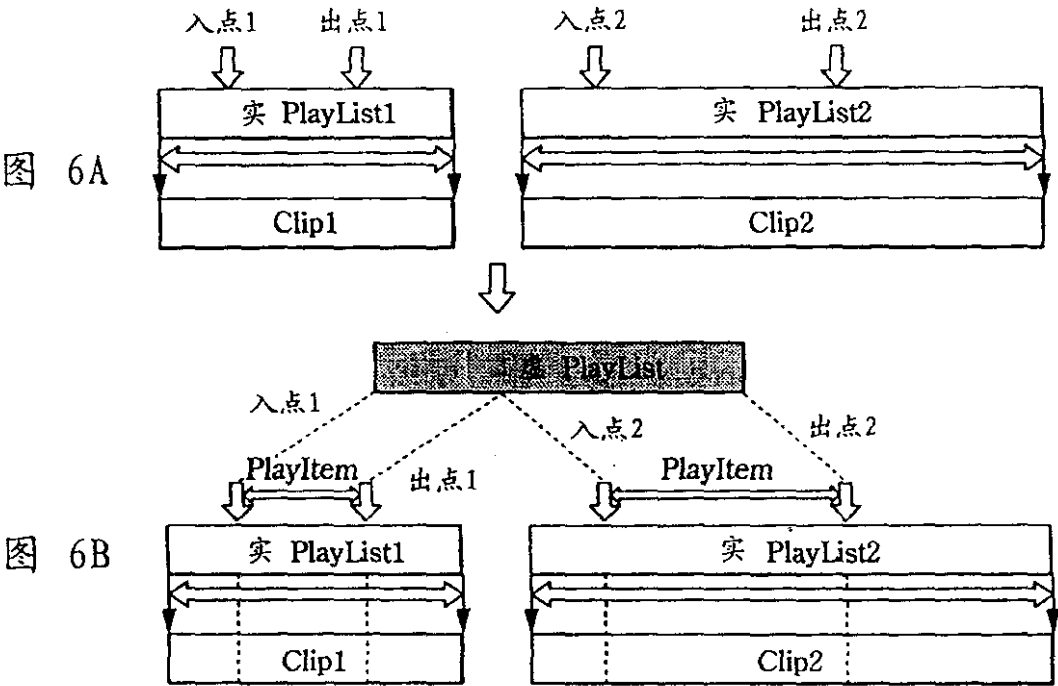


图 5C



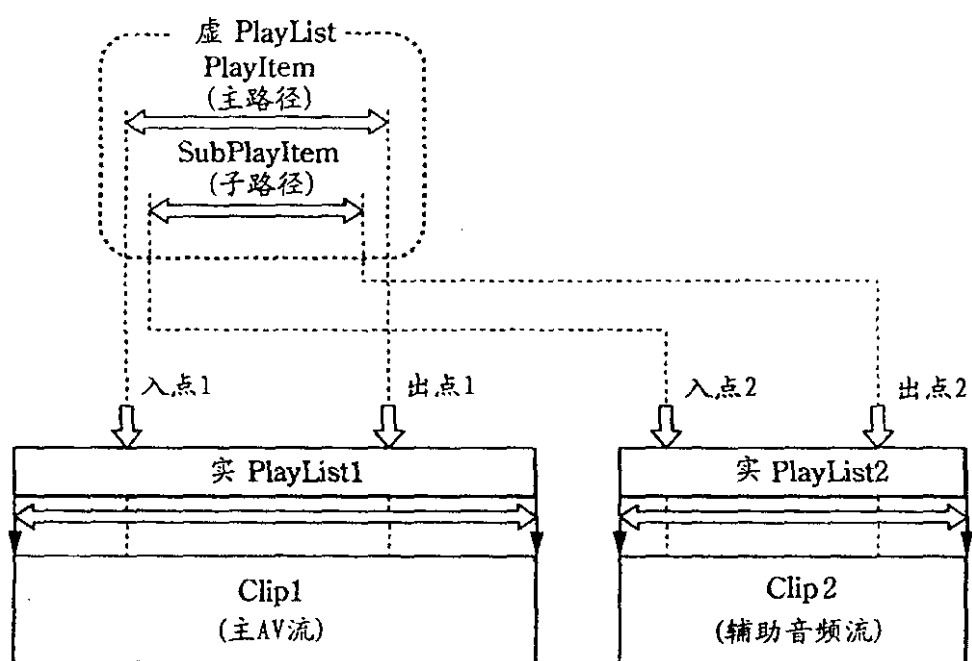


图 7

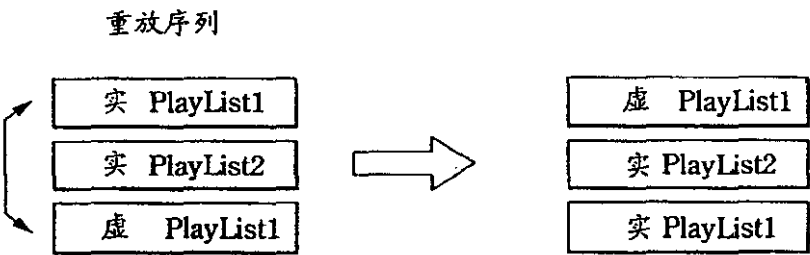


图 8

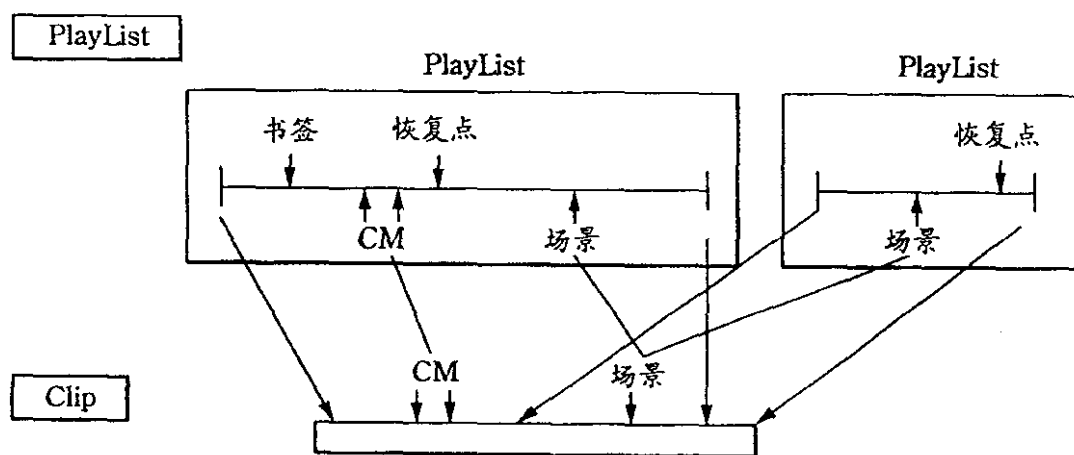


图 9

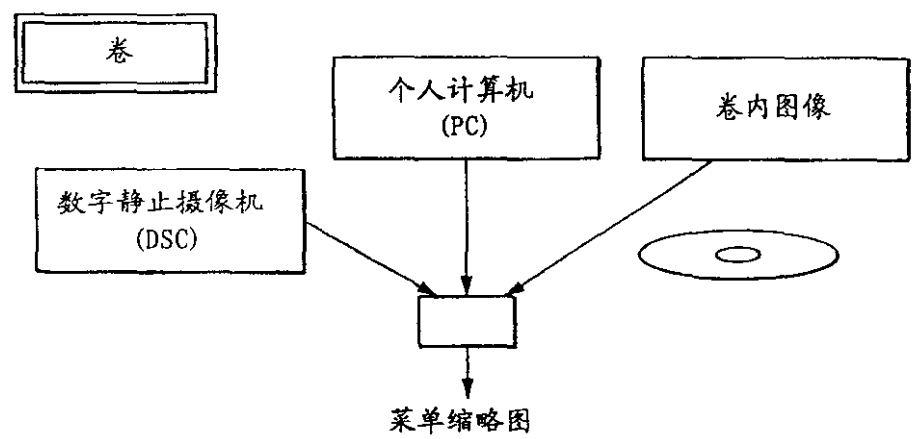


图 10

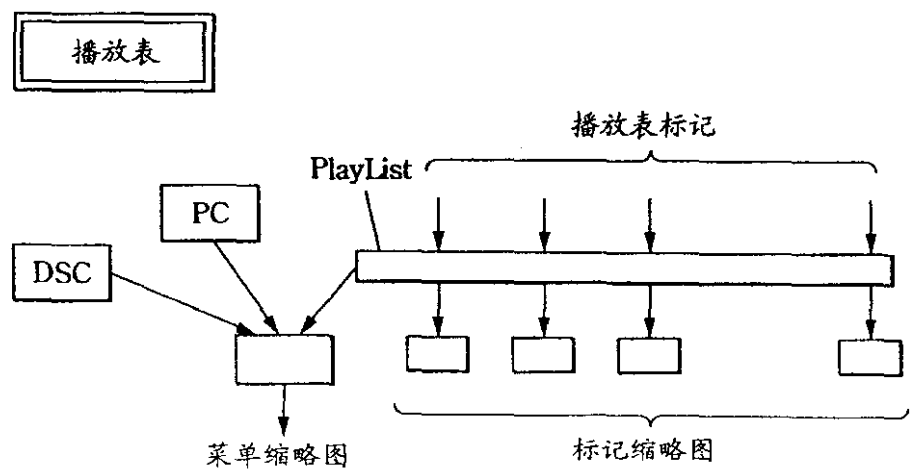


图 11

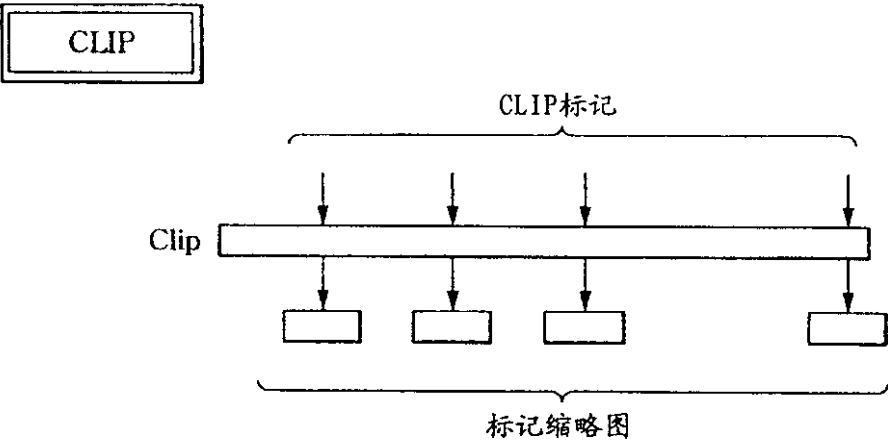


图 12

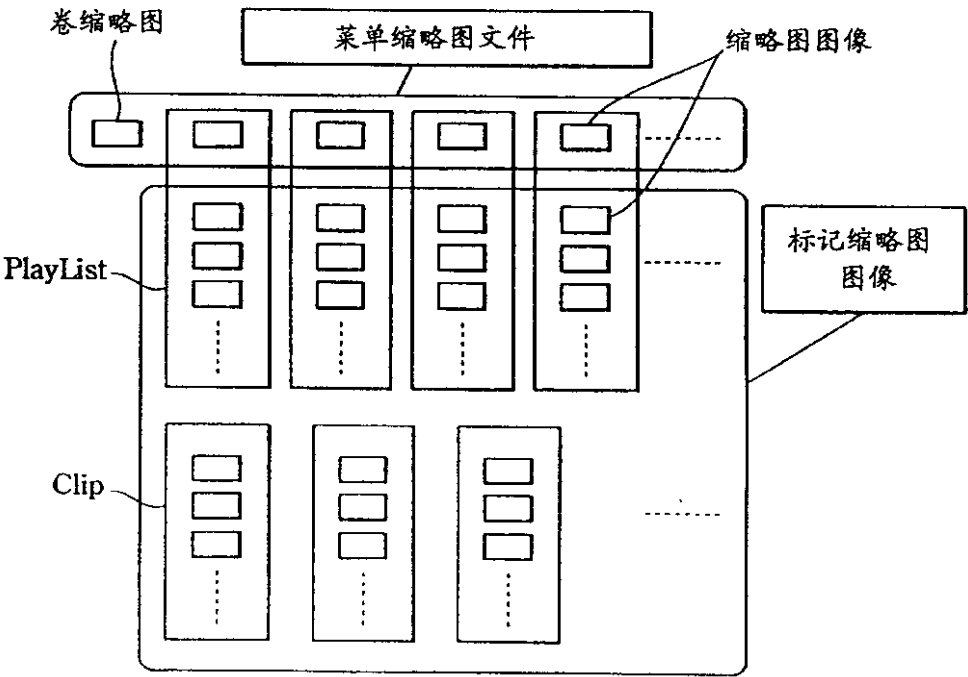


图 13

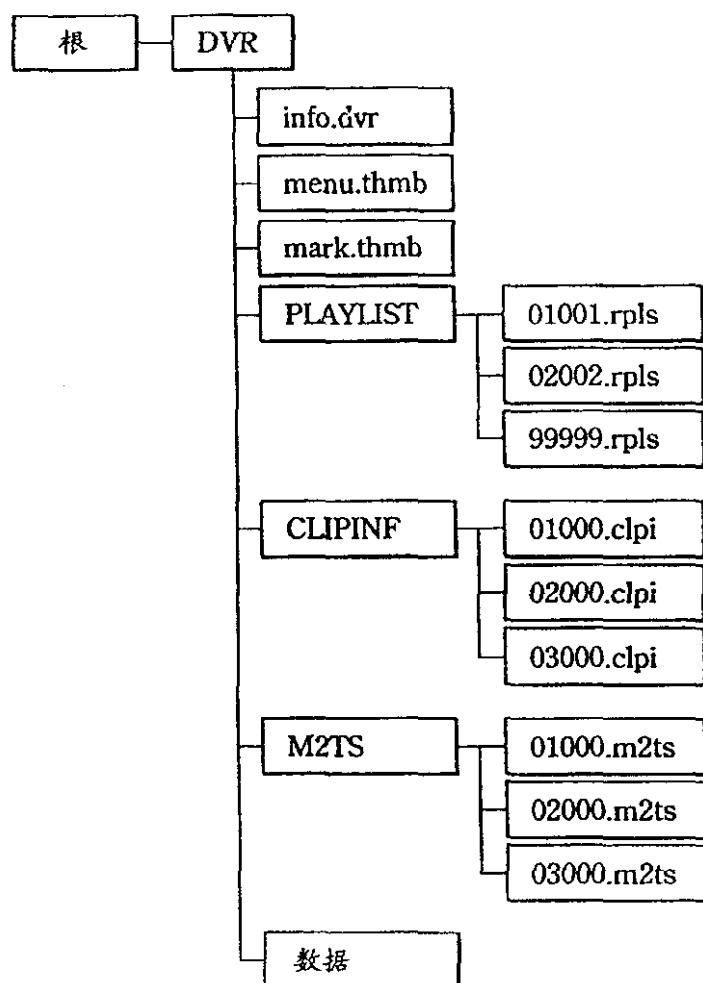


图 14

句法	字节数	缩写
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
保留	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

图 15

句法	字节数	缩写
DVRVolume(){		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

图 16

句法	字节数	缩写
ResumeVolume(){		
保留	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

图 17

句法	字节数	缩写
UIAppInfoVolume(){		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
保留	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

图 18

值	字符字母编码
0x00	保留
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (单一码)
0x03-0xff	保留

图 19

句法	字节数	缩写
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i< <i>number_of_PlayLists</i> ; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

图 20

句法	字节数	缩写
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

图 21

句法	字节数	缩写
MakersPrivateData(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
保留	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
保留	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

图 22

句法	字节数	缩写
xxxxx.rpls / yyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
保留	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

图 23

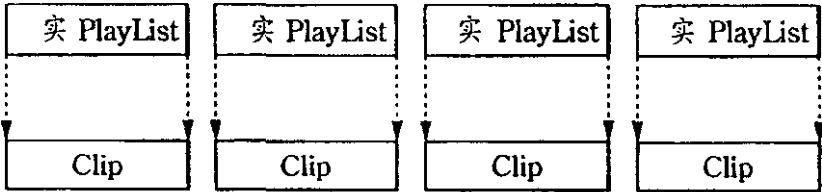


图 24A

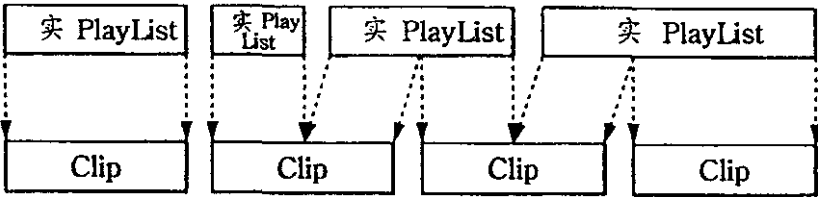


图 24B

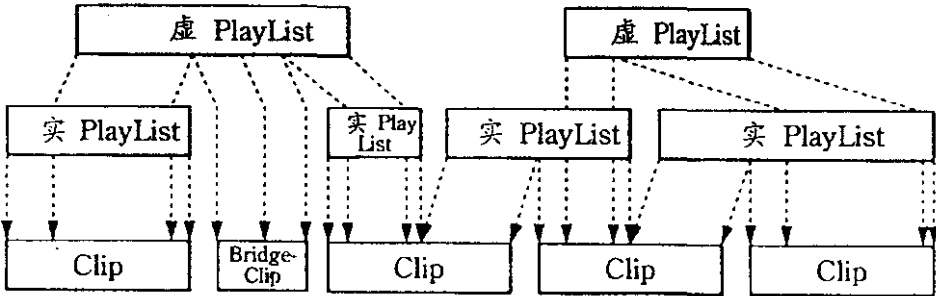


图 24C

句法	字节数	缩写
PlayList() {		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
保留	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // 主路径	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // 子路径	16	uimsbf
}else{		
保留	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() // 主路径		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() // 子路径		
}		
}		
}		

图 25

PlayList_type	含义
0	AV记录的播放表; 在该播放表中引用的所有CLIPS一定包含一个或多个视频流
1	音频记录的播放表; 该播放表中引用的所有CLIPS一定包含一个或多个音频流并且一定不包含视频流
2-255	保留

图 26

句法	字节数	缩写
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
保留	8	bslbf
record_time_and_date	4*14	bslbf
保留	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
保留	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

图 27

write_protect_flag	含义
0b	PlayList能被自由地删除
1b	除write_protect_flag之外PlayList内容应不能被擦除和改变

图 28A

is_played_flag	含义
0b	自从其记录开始, PlayList没有被再现
1b	自从其记录开始, PlayList被再现一次

图 28B

文档	含义
00b	没有定义意义
01b	原始的
10b	拷贝
11b	保留

图 28C

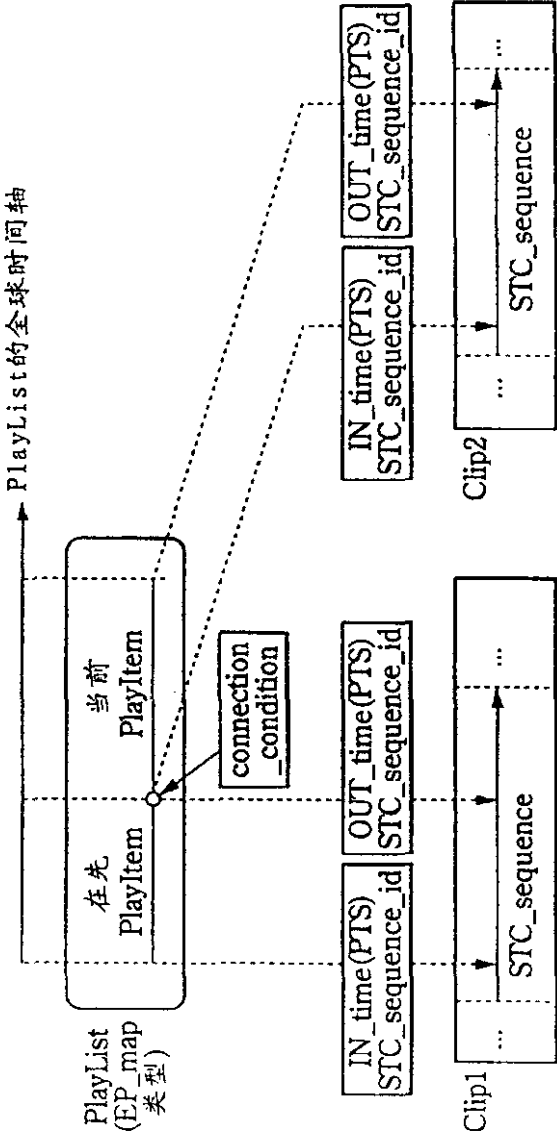


图 29

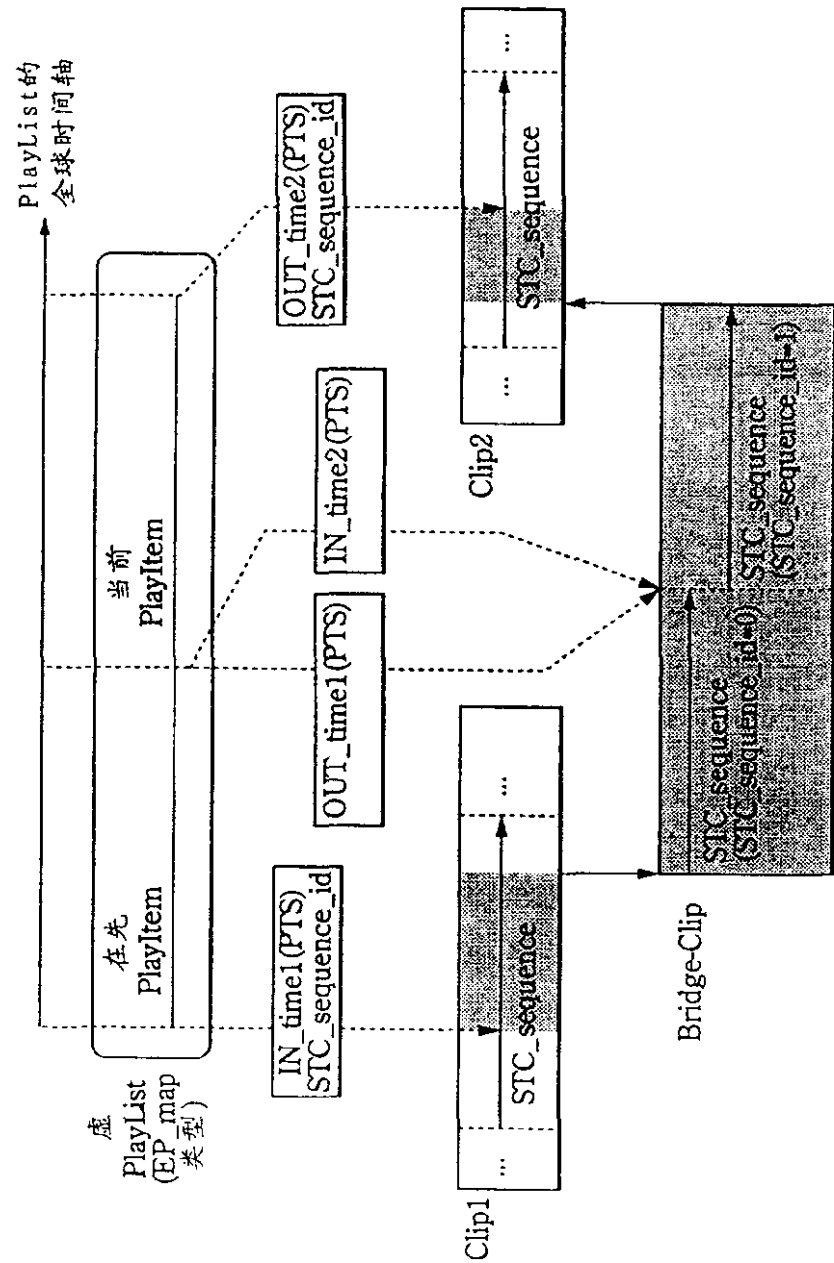


图 30

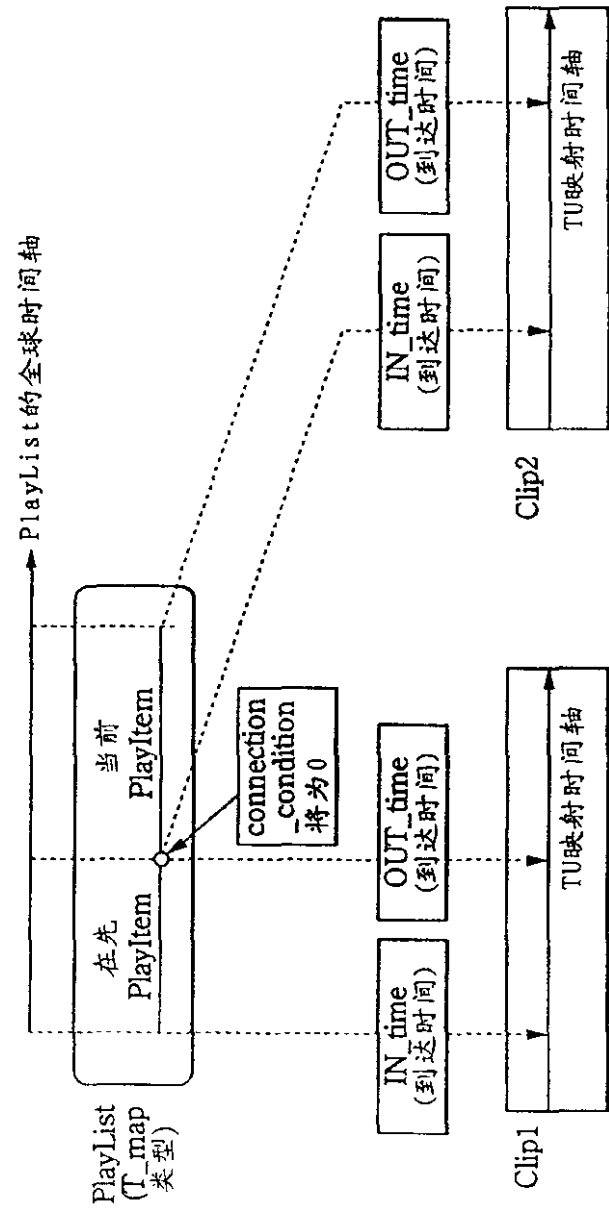


图 31

句法	字节数	缩写
PlayItem() {		
Clip_information_file_name	8*10	bslbf
保留	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
保留	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (connection_condition=='10'){		
BridgeSequenceInfo()		
}		
}		
}		

图 32

PlayList() 中的CPI_type	IN_time的语义
EP_map类型	IN_time一定表示对应于PlayItem中的第一显示单元的33位长度的高32位
TU_map类型	IN_time一定是TU_map-time-axis上的时间，并一定四舍五入到time_unit精度。IN_time是由下述方程计算： $IN\_time = TU\_start\_time \% 2^{32}$

图 33

PlayList() 中的CPI.type	OUT.time的语义
EP_map类型	OUT.time一定表示由下述方程计算的 Presentation_end_TS值的高32位: $\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$ 这里PTS-out是对应于PlayItem中的最后显示单元的33位长的PTS。AU-duration是最后显示单元的90kHz显示时间。
TU_map类型	OUT.time一定是TU_map.time-axis上的时间, 并四舍五入到time.unit精度。OUT.time由下式计算: $\text{OUT\_time} = \text{TU\_start\_time} \% 2^{32}$

图 34

connection _condition	含义
00	. 在先PlayItem对当前PlayItem的连接不保证为无缝重新播放。 . 如果PlayList的CPI_type是TU_map类型, 该值一定设置在connection_condition中。
01	. 该状态仅仅当PlayList的CPI_type为EP_map类型时是允许的。 . 因为系统时基的非连续点(STC基), 当前PlayItem和在先PlayItem表示分开。
10	. 该状态仅仅当PlayList的CPI_type是EP_map类型时是允许的。 . 该状态仅对虚PlayList是允许的。 . 在先PlayItem对当前PlayItem的连接被保证为无缝重新播放。 . 在先PlayItem使用BridgeSequence连接到当前PlayItem, DVR MPEG-2传输流一定遵守后面说明的DVR-STD。
11	. 该状态仅当PlayList的CPI_type是EP_map类型时是允许的。 . 在先PlayItem到当前PlayItem的连接保证为无缝重新播放。 . 在先PlayItem在不使用BridgeSequence的情况下连接到当前PlayItem, DVR MPEG-2传输流一定遵守后面说明的DVR-STD。

图 35

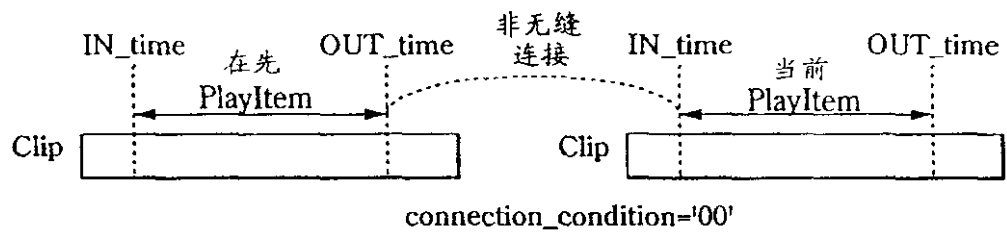


图 36A

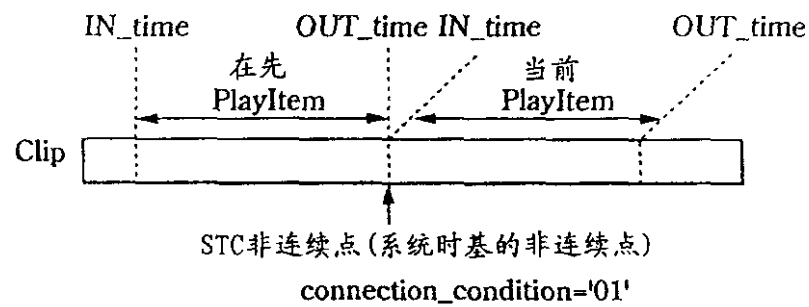


图 36B

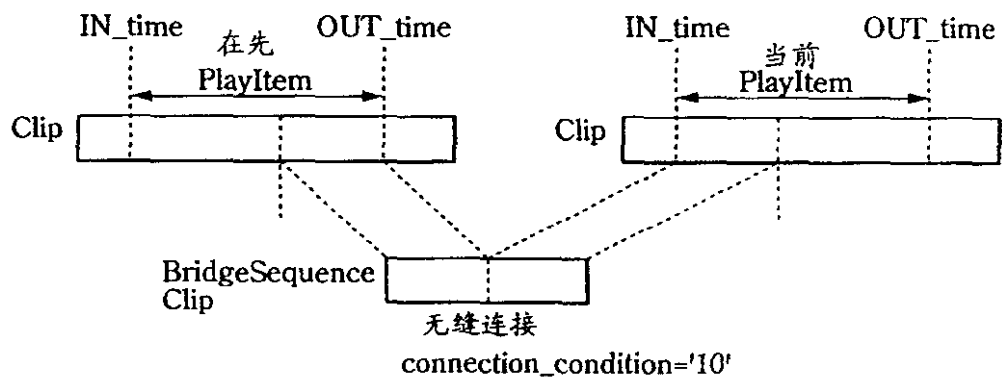


图 36C

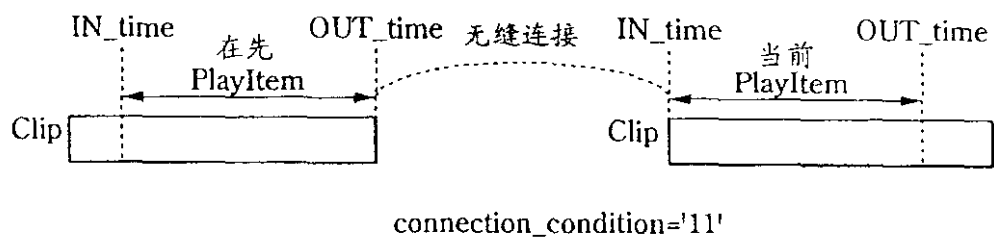


图 36D

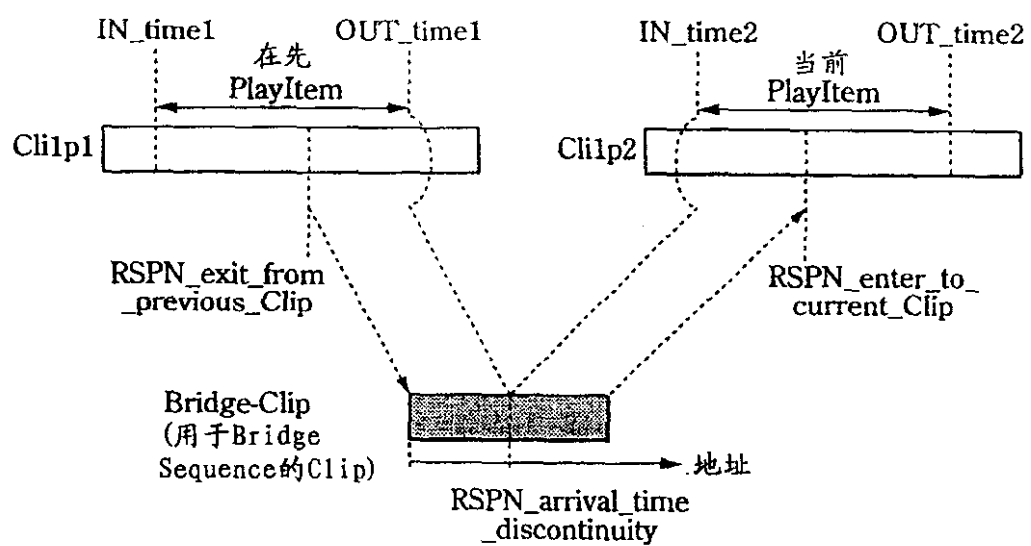


图 37

句法	字节数	缩写
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

图 38

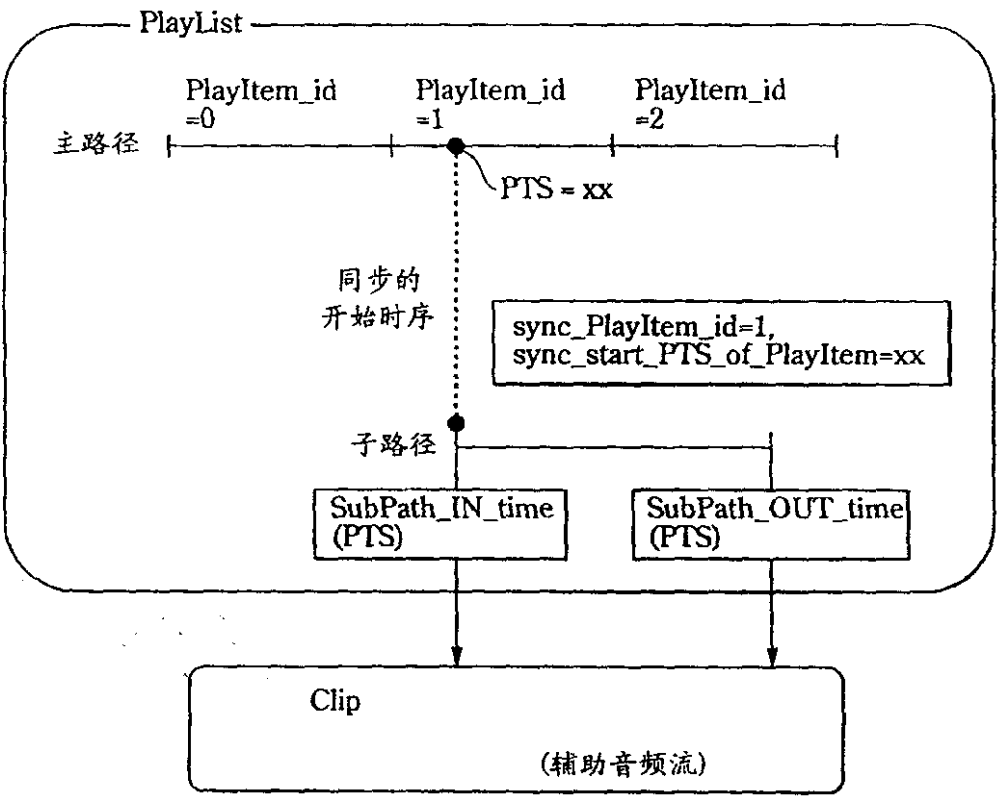


图 39

句法	字节数	缩写
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

图 40

SubPath_type	含义
0x00	辅助音频流路径
0x01-0xff	保留

图 41

句法	字节数	缩写
PlaylistMark() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0; i<number_of_PlayList_marks; i++) {		
保留	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
保留	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

图 42

Mark_type	含义	说明
0x00	恢复-标记	重新播放恢复点。在PlayListMark()中定义的重新播放恢复点数一定为0或1。
0x01	书-标记	重新播放PlayList的入口点。该标记能够由用户设置和用作为指定喜爱的场景开始点的标记。
0x02	跳步-标记	跳步标记点。播放器从该点到程序结束跳过程序。在PlayListMark()中定义的跳步标记点数一定是0或1。
0x03-0x8F	保留	
0x90-0xFF	保留	保留用于ClipMark()

图 43

PlayList() 中的CPI_type	mark_time_stamp的语义
EP_map类型	mark_time_stamp一定表示对应于由标记引用的表示单元的33位长度PTS的32位
TU_map类型	mark_time_stamp一定是关于TU_map_time_axis的时间并且一定四舍五入到time_unit精度。 mark_time_stamp通过下述方程计算: $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

图 44

句法	字节数	缩写
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
保留	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

图 45

句法	字节数	缩写
ClipInfo() {		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
保留	8	bslbf
record_time_and_date	4*14	bslbf
保留	8	bslbf
duration	4*6	bslbf
保留	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
<i>if (Clip_stream_type==1) // Bridge-Clip AV stream</i>		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
保留	32	bslbf
reserved_for_system_use	144	bslbf
保留	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

图 46

Clip_stream_type	含义
0	Clip AV流
1	Bridge-Clip AV流
2-255	保留

图 47

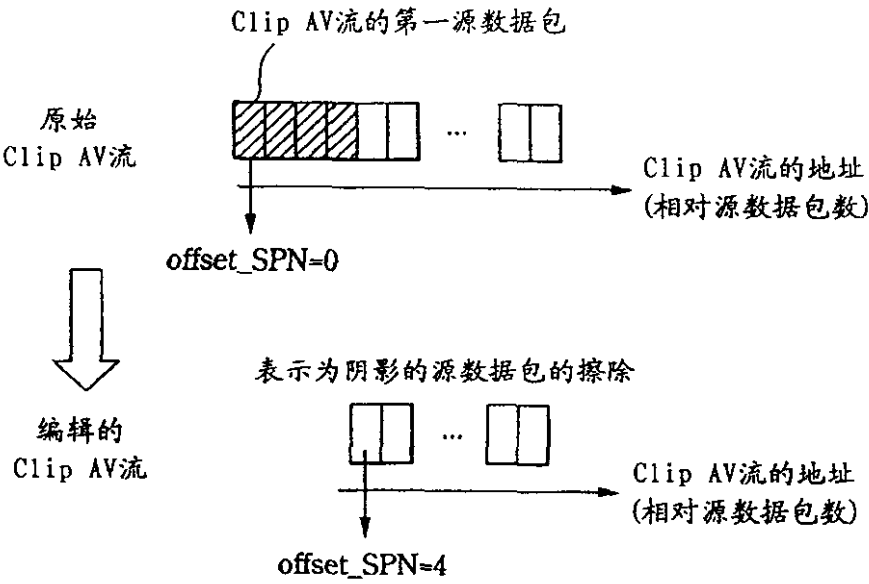


图 48

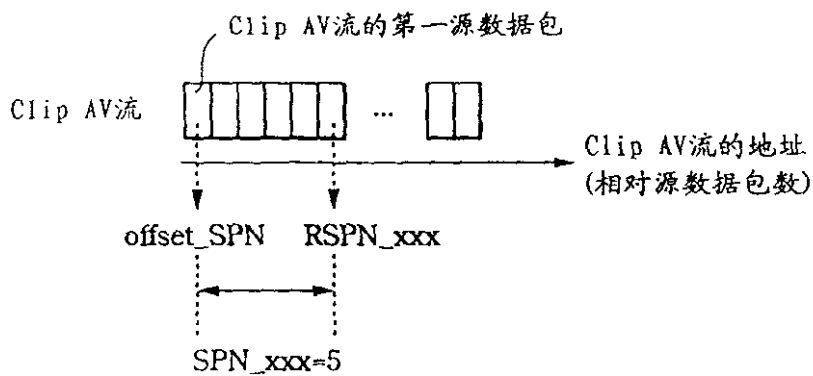


图 49

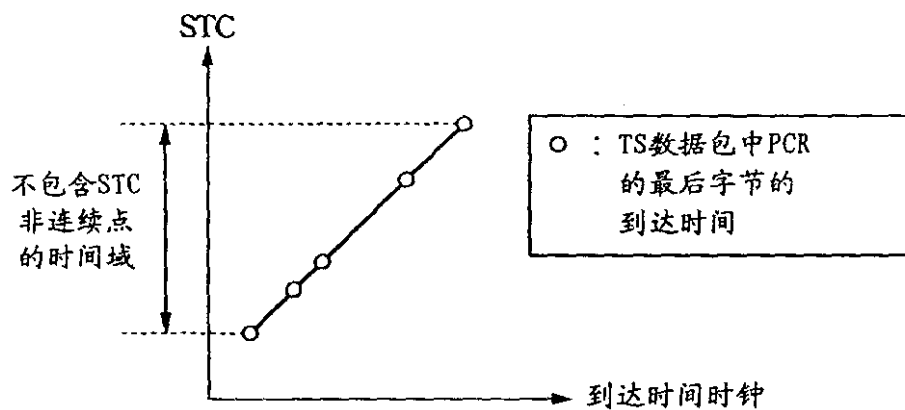


图 50A

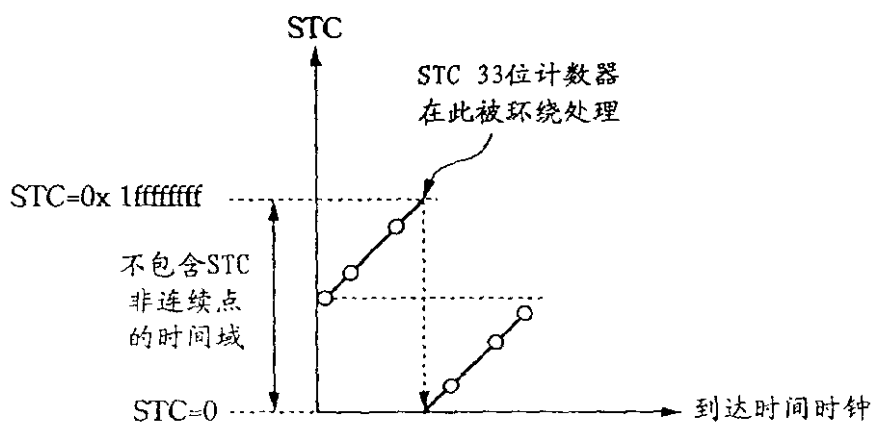


图 50B

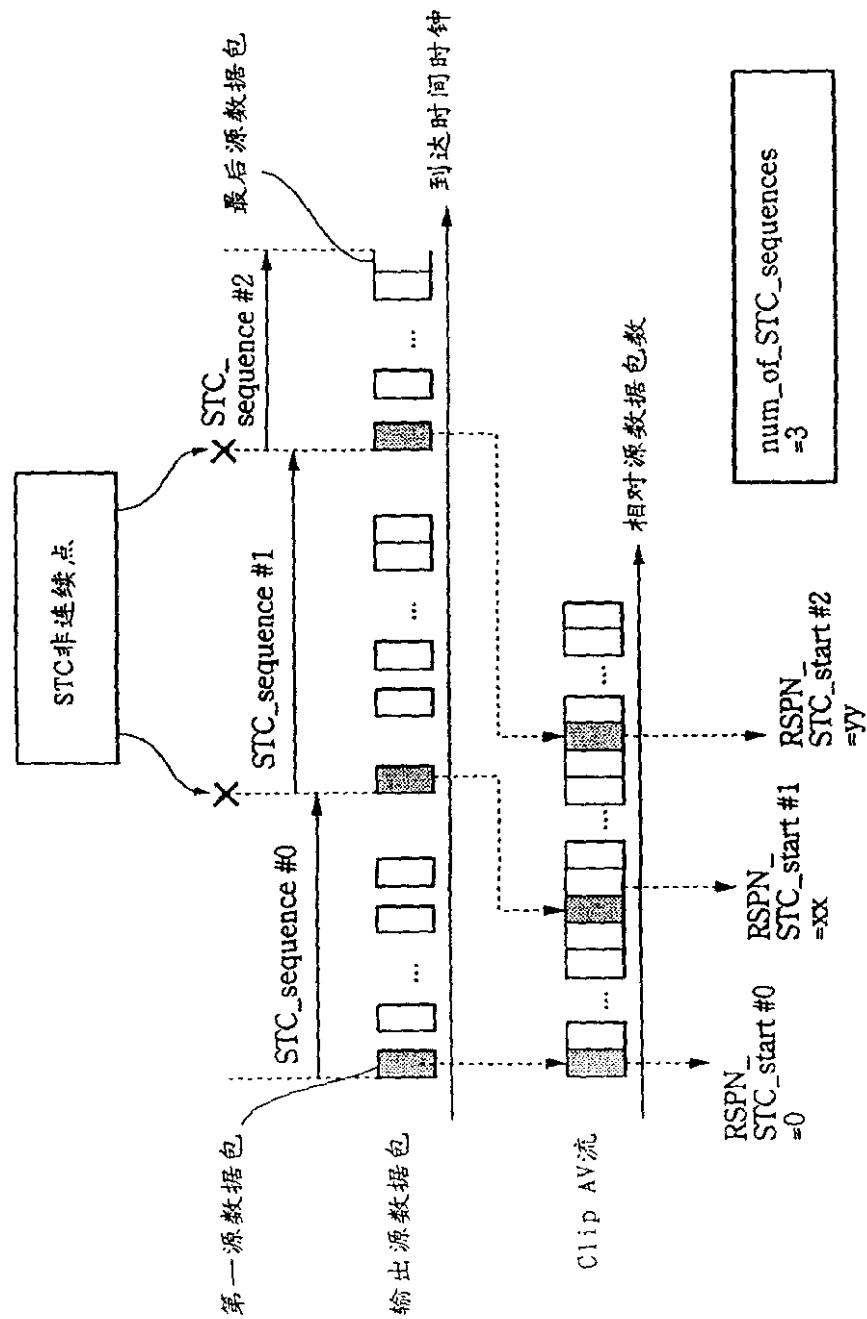


图 51

句法	字节数	缩写
STC_Info0{		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
保留	8	bslbf
num_of_STC_sequences	8	uimsbf
for (STC_sequence_id=0; STC_sequence_id<num_of_STC_sequences; STC_sequence_id++){		
保留	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

图 52

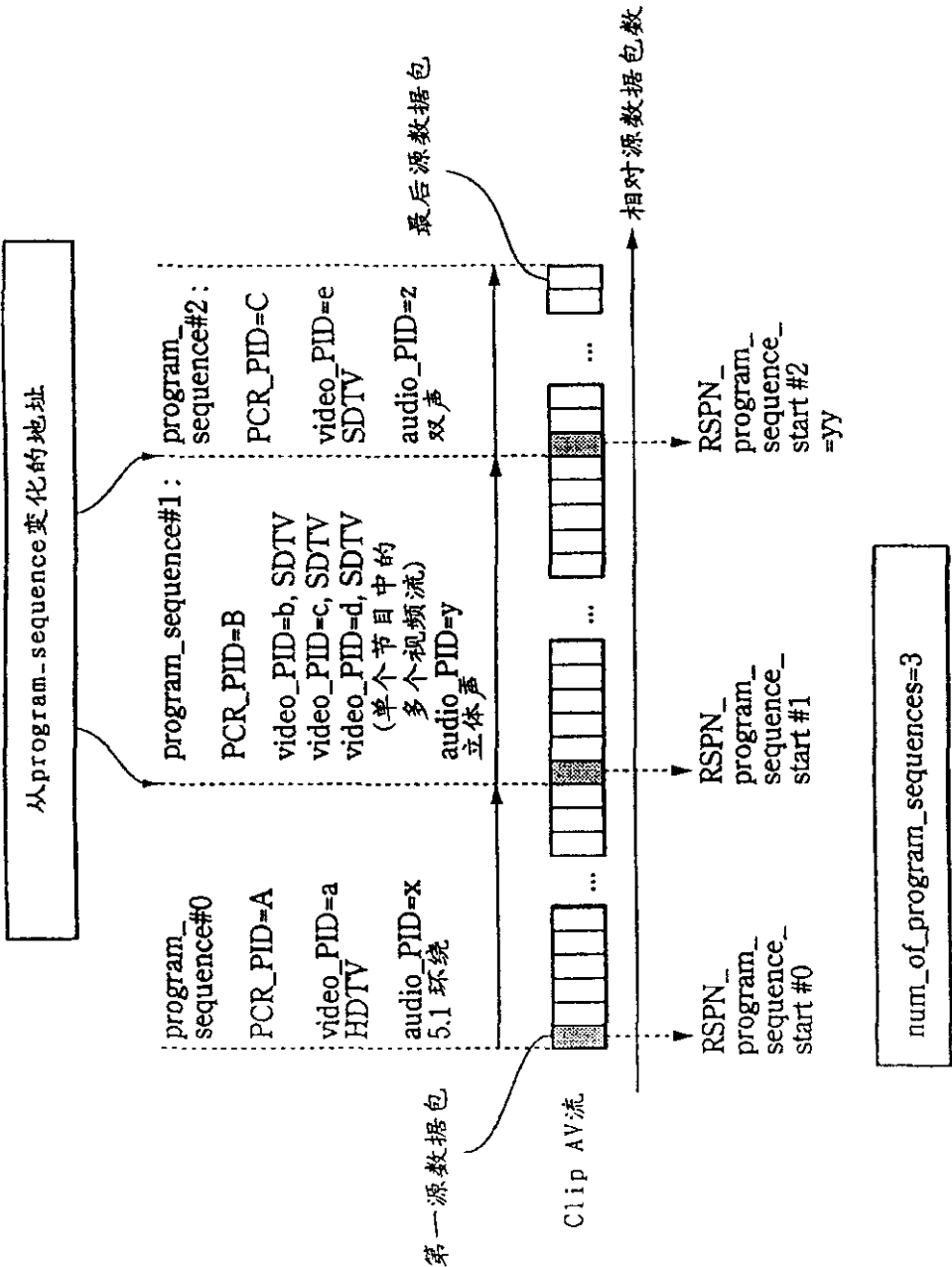


图 53

句法	字节数	缩写
ProgramInfo()		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
保留	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<number_of_program_sequences;i++){		
RSPN_program_sequence_start	32	uimsbf
保留	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<number_of_videos;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<number_of_audios;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

图 54

句法	字节数	缩写
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
保留	8	bslbf
}		

图 55

video_format	含义
0	480i
1	576i
2	480p (包括640×480p格式)
3	1080i
4	720p
5	1080p
6-254	保留
255	无信息

图 56

frame_rate	含义
0	禁止
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97...)
5	30
6	50
7	60 000/1001 (59.94...)
8	60
9-254	保留
255	无信息

图 57

display_aspect_ratio	含义
0	禁止
1	保留
2	4:3显示纵横比
3	16:9显示纵横比
4-254	保留
255	无信息

图 58

句法	字节数	缩写
AudioCodingInfo() {		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
保留	8	bslbf
}		

图 59

audio_coding	含义
0	MPEG-1 音频层 I 或 II
1	杜比 AC-3 音频
2	MPEG-2 AAC
3	MPEG-2 多声道音频, 与 MPEG-1 兼容的反向
4	SESF LPCM 音频
5-254	保留
255	无信息

图 60

audio_component_type	含义
0	单声道
1	双声道
2	立体声 (2声道)
3	多舌音, 多声道
4	环绕声
5	视觉损坏的音频说明
6	难听的音频
7-254	保留
255	无信息

图 61

sampling_frequency	含义
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	保留
255	无信息

图 62

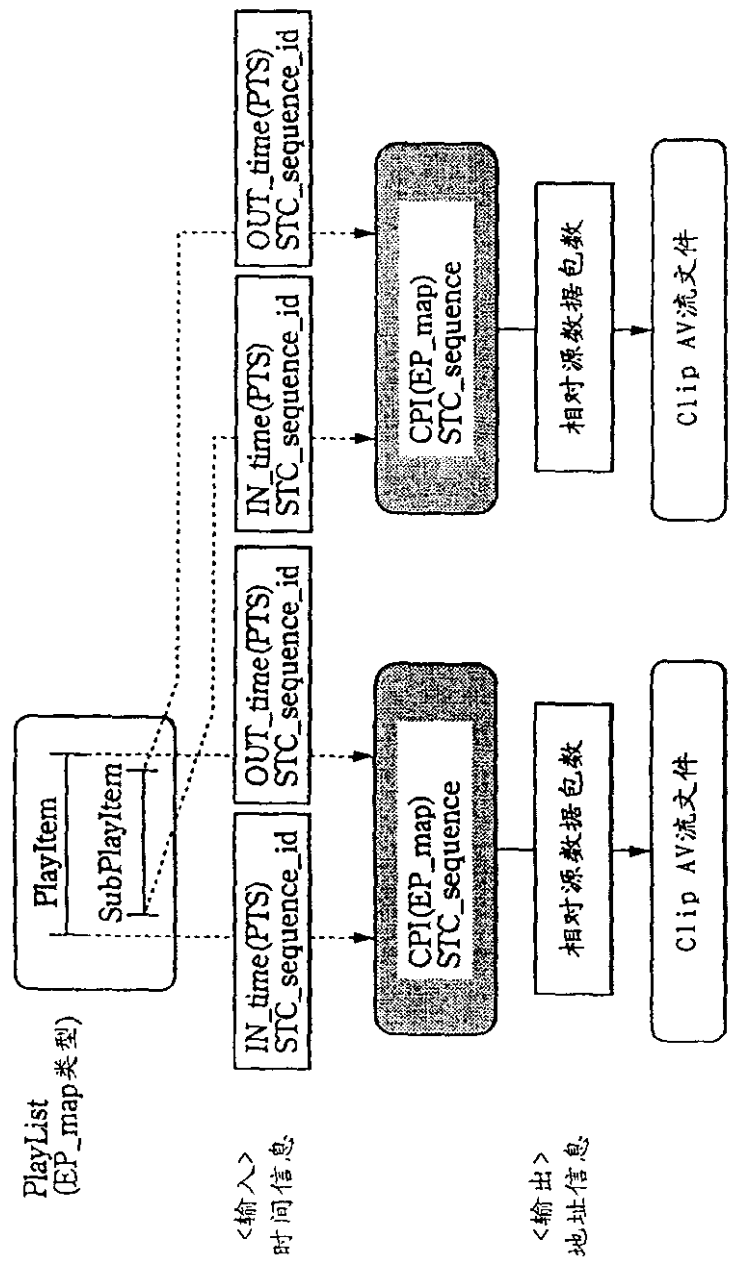


图 63

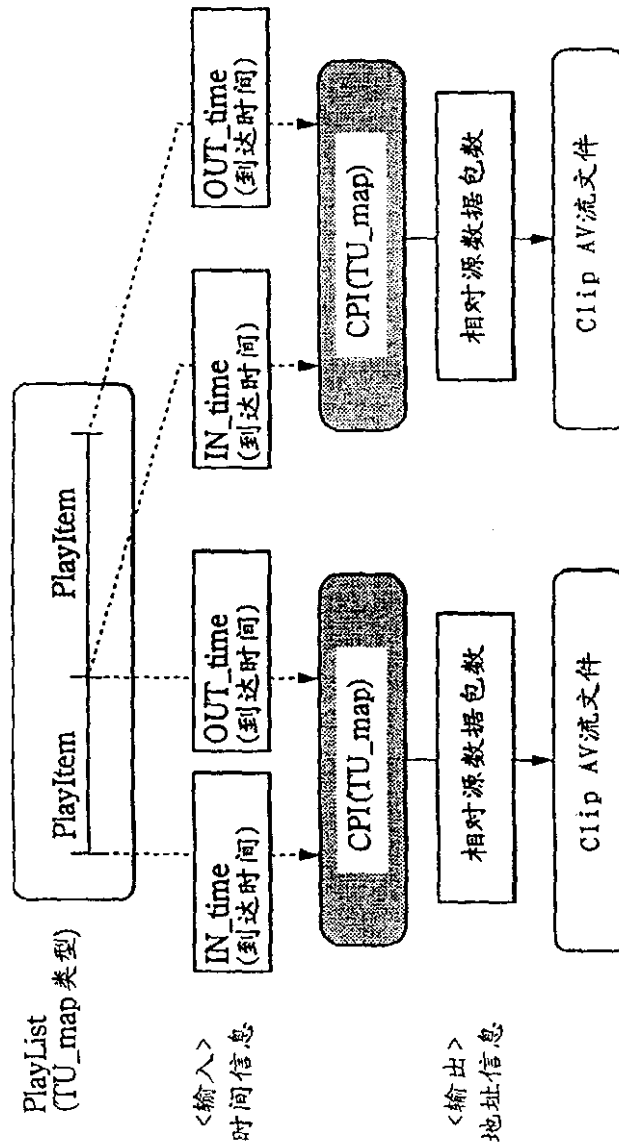


图 64

句法	字节数	缩写
CPI(){		
version_number	8*4	bslbf
length	32	uimsbf
保留	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

图 65

CPI_type	含义
0	EP映射类型
1	TU映射类型

图 66

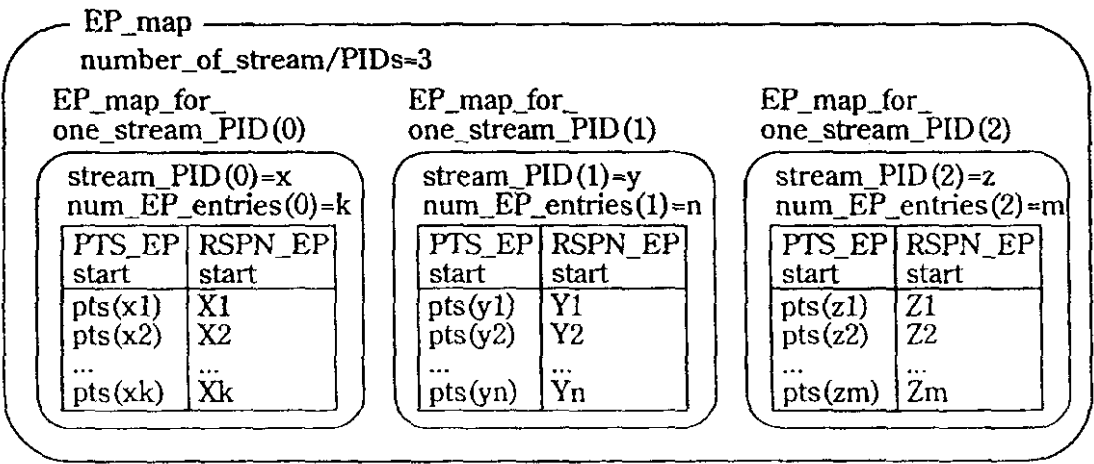
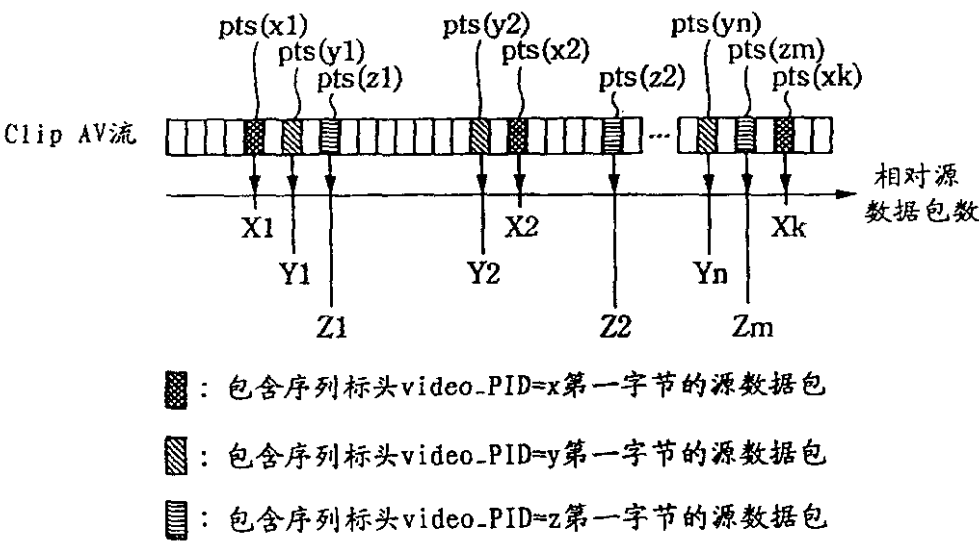
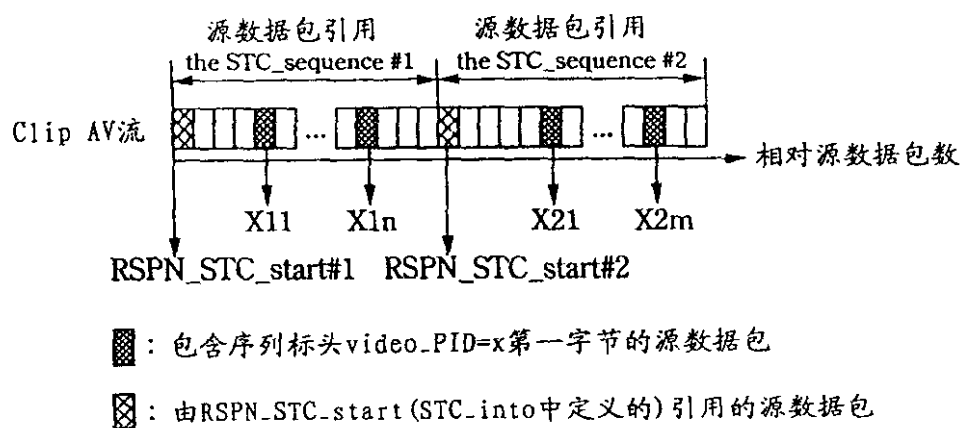


图 67



EP_map_for_one_stream_PID
video_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11	属于STC-sequence#1的数据
...	...	
pts(x1n)	X1n	边界
pts(x21)	X21	
...	...	属于STC-sequence#2的数据
pts(x2m)	X2m	

RSPN_STC_start #2 < X21

图 68

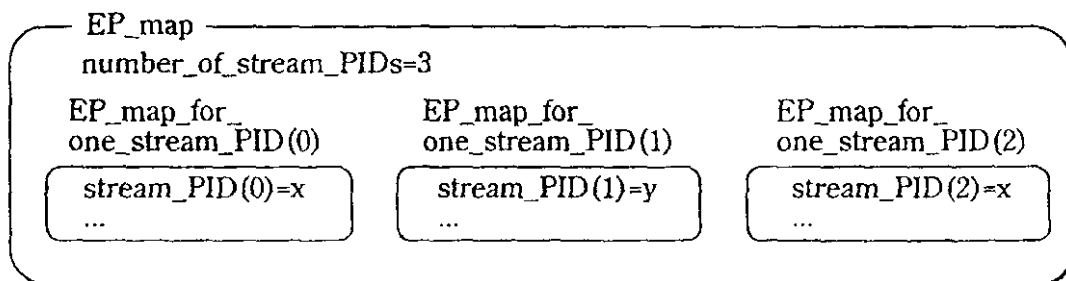
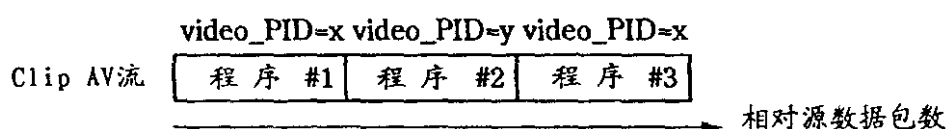


图 69

句法	字节数	缩写
EP_map(){		
保留	12	bslbf
EP_type	4	uimbsbf
number_of_stream_PIDs	16	uimbsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimbsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimbsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

图 70

EP_type	含义
0	视频
1	音频
2-15	保留

图 71

句法	字节数	缩写
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

图 72

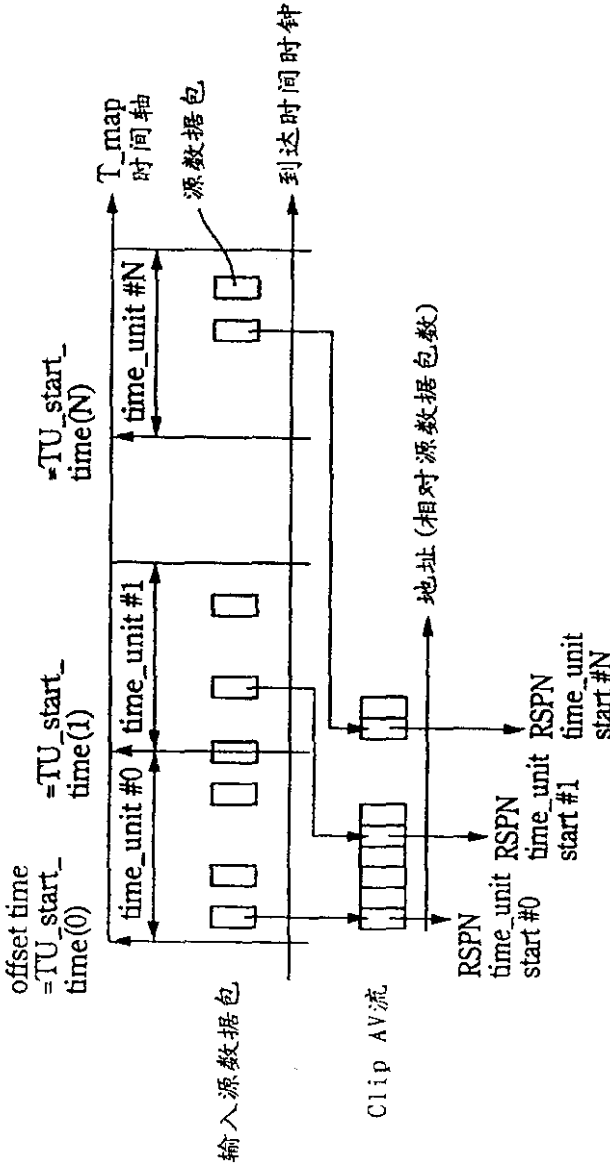


图 73

句法	字节数	缩写
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

图 74

句法	字节数	缩写
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i< <i>number_of_clip_marks</i> ; i++){		
保留	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
保留	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

图 75

Mark_type	含义	注释
0x00-0x8F	保留	保留用于PlaylistMark()
0x90	事件开始标记	说明节目开始点的标记点
0x91	本地事件开始标记	说明节目中本地场景的标记点
0x92	场景开始标记	表示场景变化点的标记点
0x93-0xFF	保留	

图 76

Playlist () 中的CPI-type	mark_time_stamp的语义
EP_map类型	mark_time_stamp一定表示对应于由标记 引用的表示单元的33位长度PTS的高32位
TU_map类型	mark_time_stamp一定是TU_map-time-axis 上的时间并一定四舍五入到time-unit精度。 mark_time_stamp由下述方程计算: $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

图 77

句法	字节数	缩写
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
保留	8	bslbf
mark_type	8	bslbf
reserved_for_MakerID	16	bslbf
<i>mark_entry()</i>		
<i>representative_picture_entry()</i>		
ref_thumbnail_index	16	uimsbf
}		
}		

图 78

Mark_type	含义	注释
0x00-0x8F	保留	保留用于PlayListMark()
0x90	事件开始标记	说明节目开始点的标记点
0x91	本地事件开始标记	说明节目中本地场景的标记点
0x92	场景开始标记	说明场景开始点的标记点
0x93	场景结束标记	说明场景结束点的标记点
0x94	CM开始标记	说明CM开始点的标记点
0x95	CM结束标记	说明CM结束点的标记点
0x96-0xBF	DVR格式保留用于ClipMark的今后扩展	
0xC0-0xFF	可分配给用于制造者专用之应用的标记	

图 79

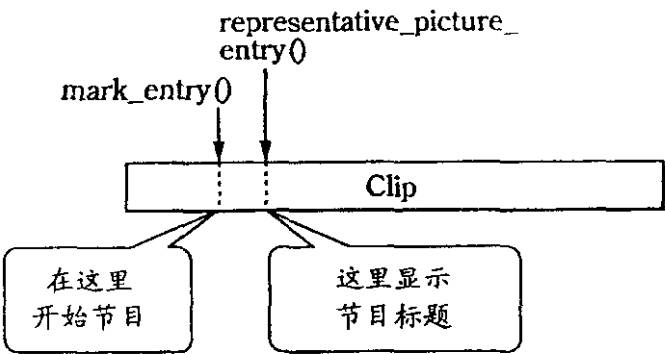


图 80

句法	字节数	缩写
mark_entry()/representative_picture_entry(){		
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
保留	24	bslbf
}		

图 81

句法	字节数	缩写
mark_entry()/representative_picture_entry(){		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

图 82

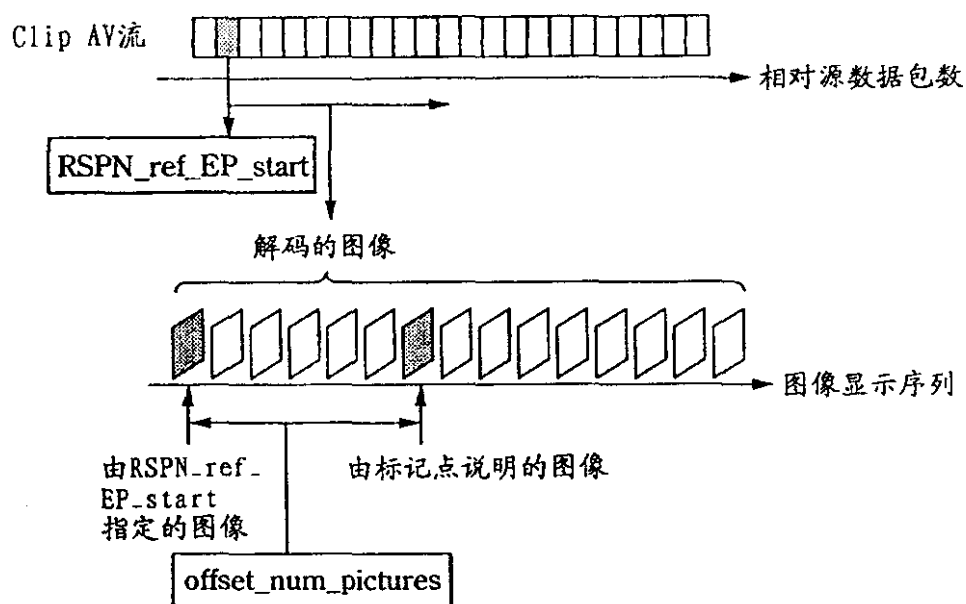


图 83

句法	字节数	缩写
mark_entry0/representative_picture_entry0{		
RSPN_mark_point	32	uimsbf
}		

图 84

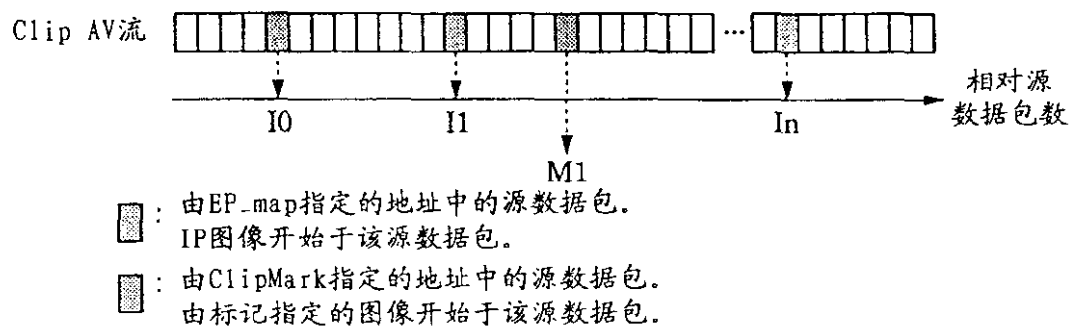


图 85

句法	字节数	缩写
menu.thmb/mark.thmb() {		
保留	256	bslbf
Thumbnail()		
for (i=0;i<N1;i++)		
padding_word	16	bslbf
}		

图 86

句法	字节数	缩写
Thumbnail(){		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
保留	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
保留	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
保留	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
tn_block	tn_block_size*1024*8	
}		
}		

图 87

Thumbnail_picture_format	含义
0x00	MPEG-2视频I-图像
0x01	DCF (限制JPEG)
0x02	PNG
0x03-0xff	保留

图 88

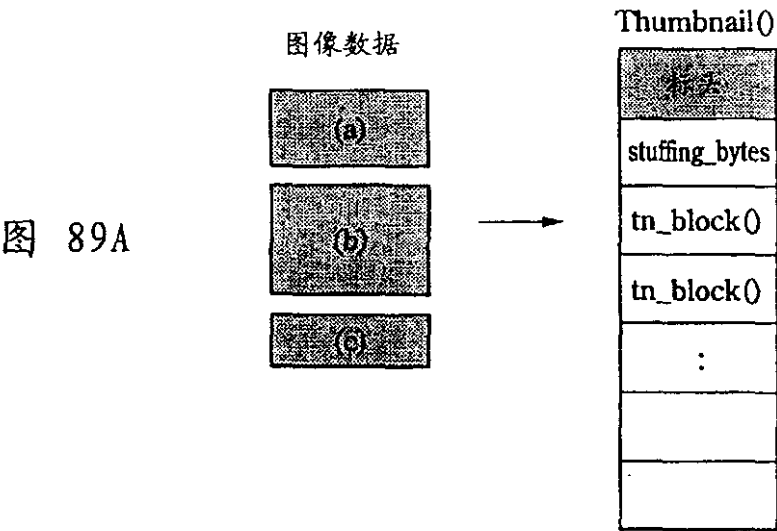
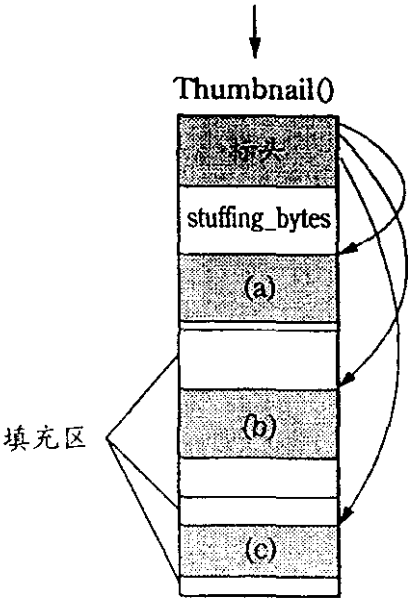


图 89B



DVR MPEG-2传输流

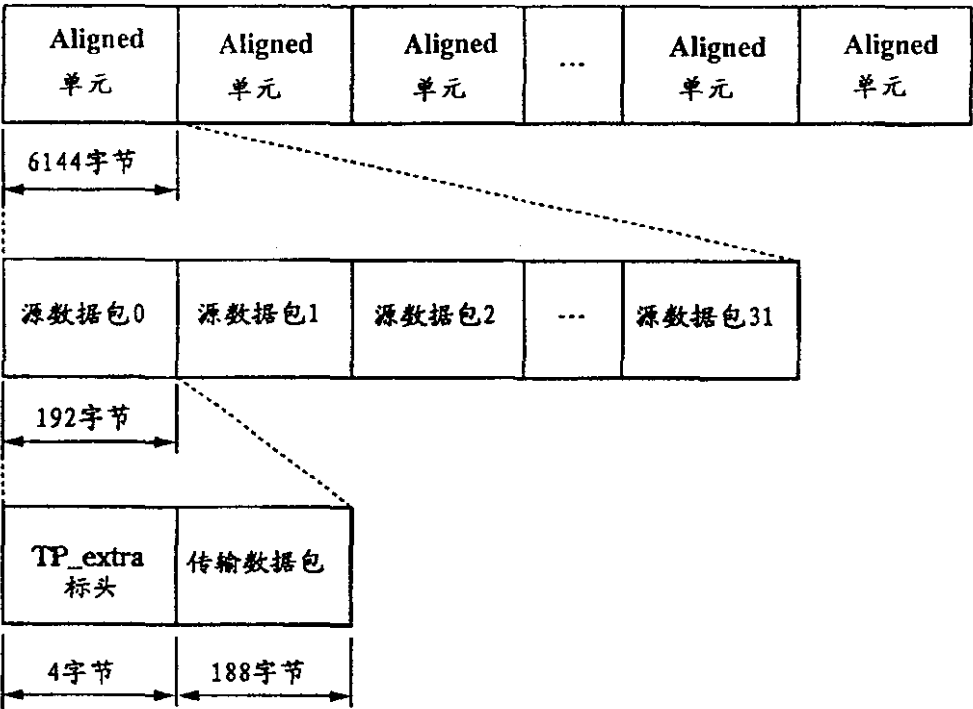


图 90

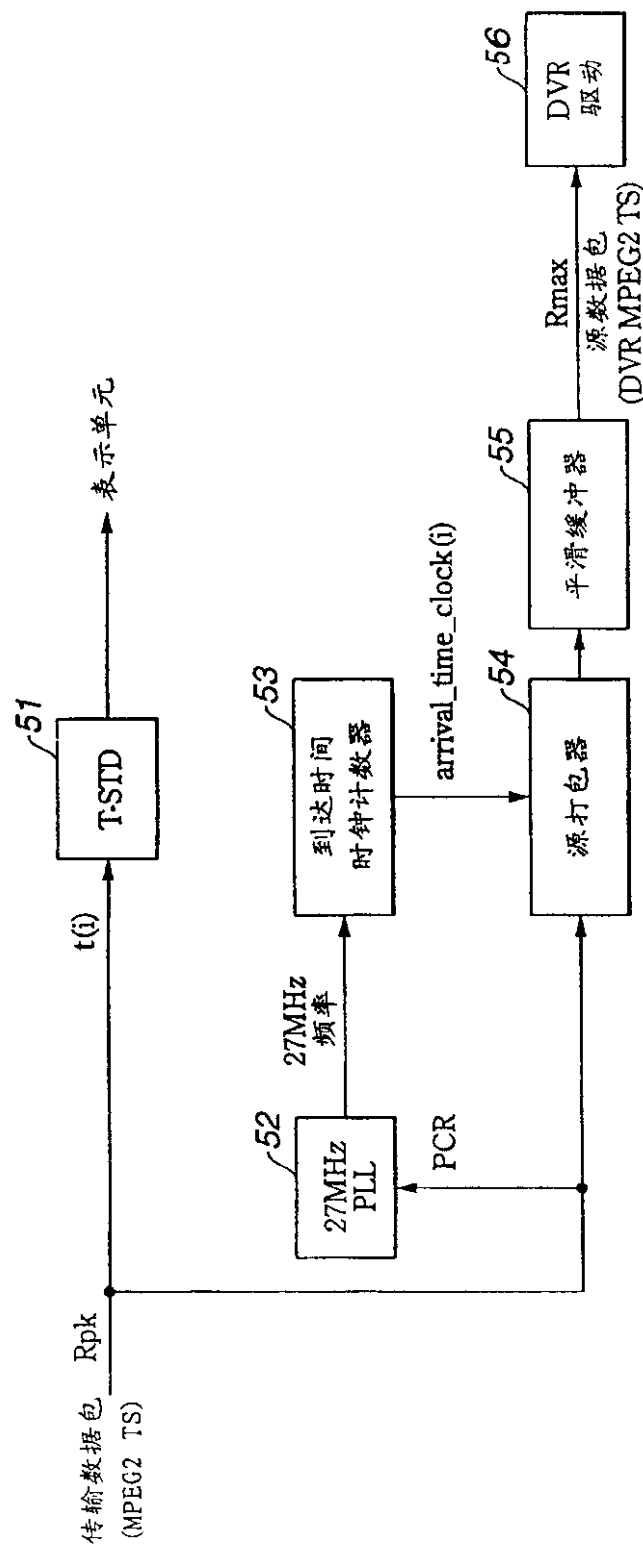


图 91

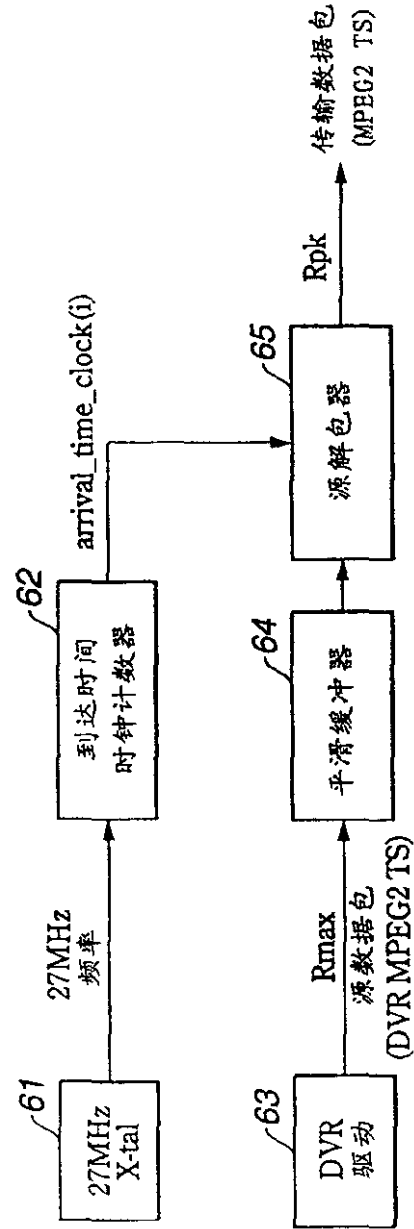


图 92

句法	字节数	缩写
source_packet() {		
TP_extra_header()		
trasport_packet()		
}		

图 93

句法	字节数	缩写
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

图 94

copy_permission indicator	含义
00	自由拷贝
01	不再拷贝
10	一次拷贝
11	禁止拷贝

图 95

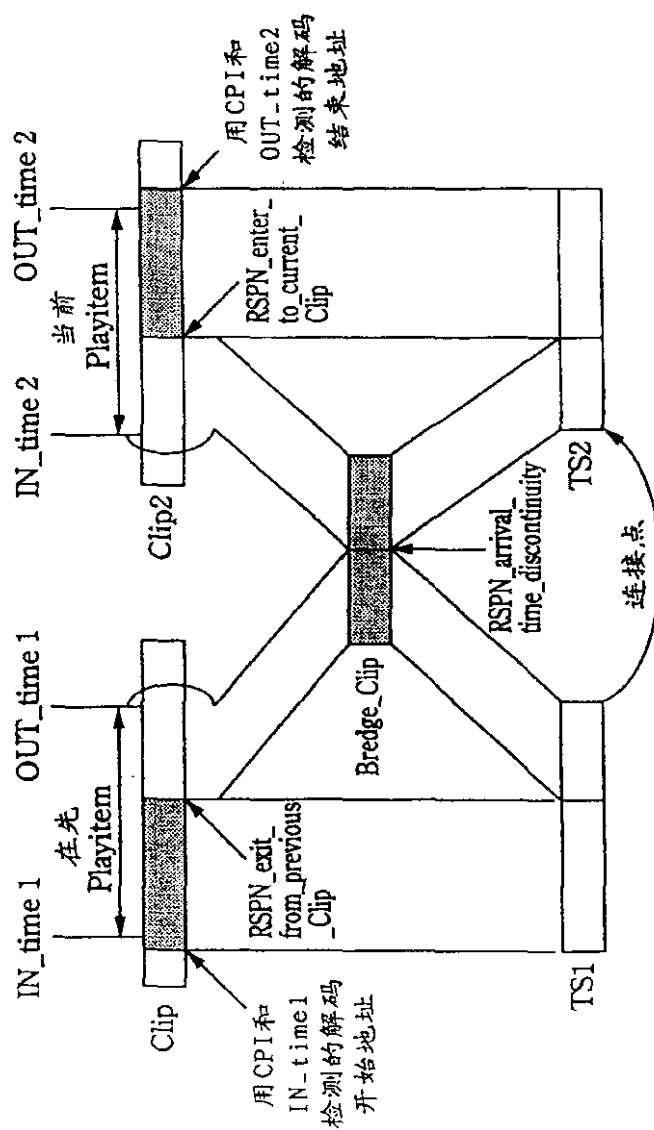


图 96

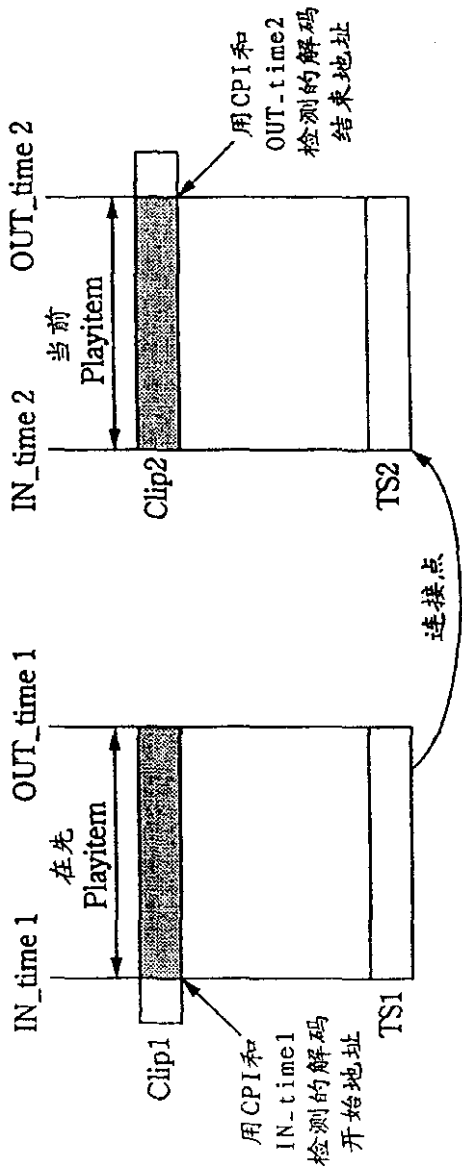


图 97

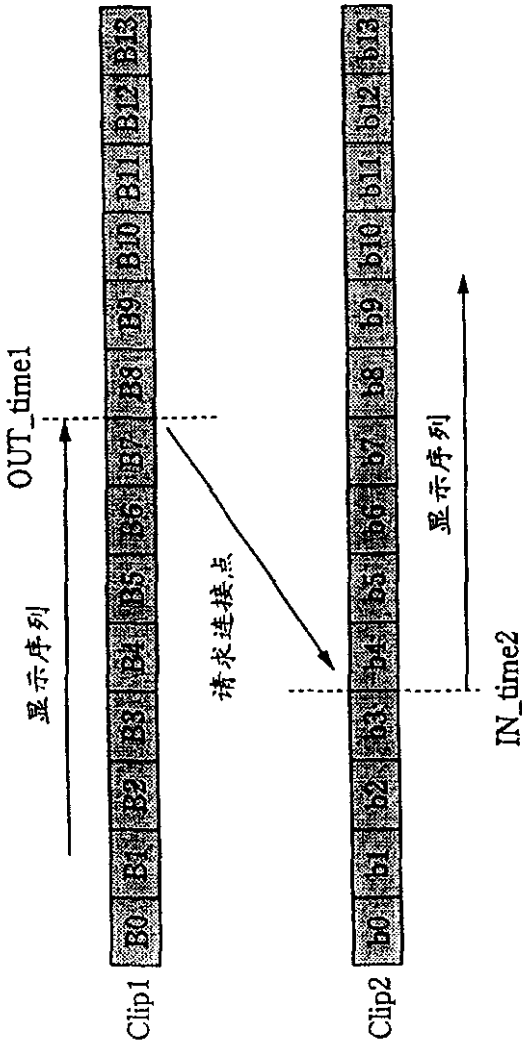


图 98

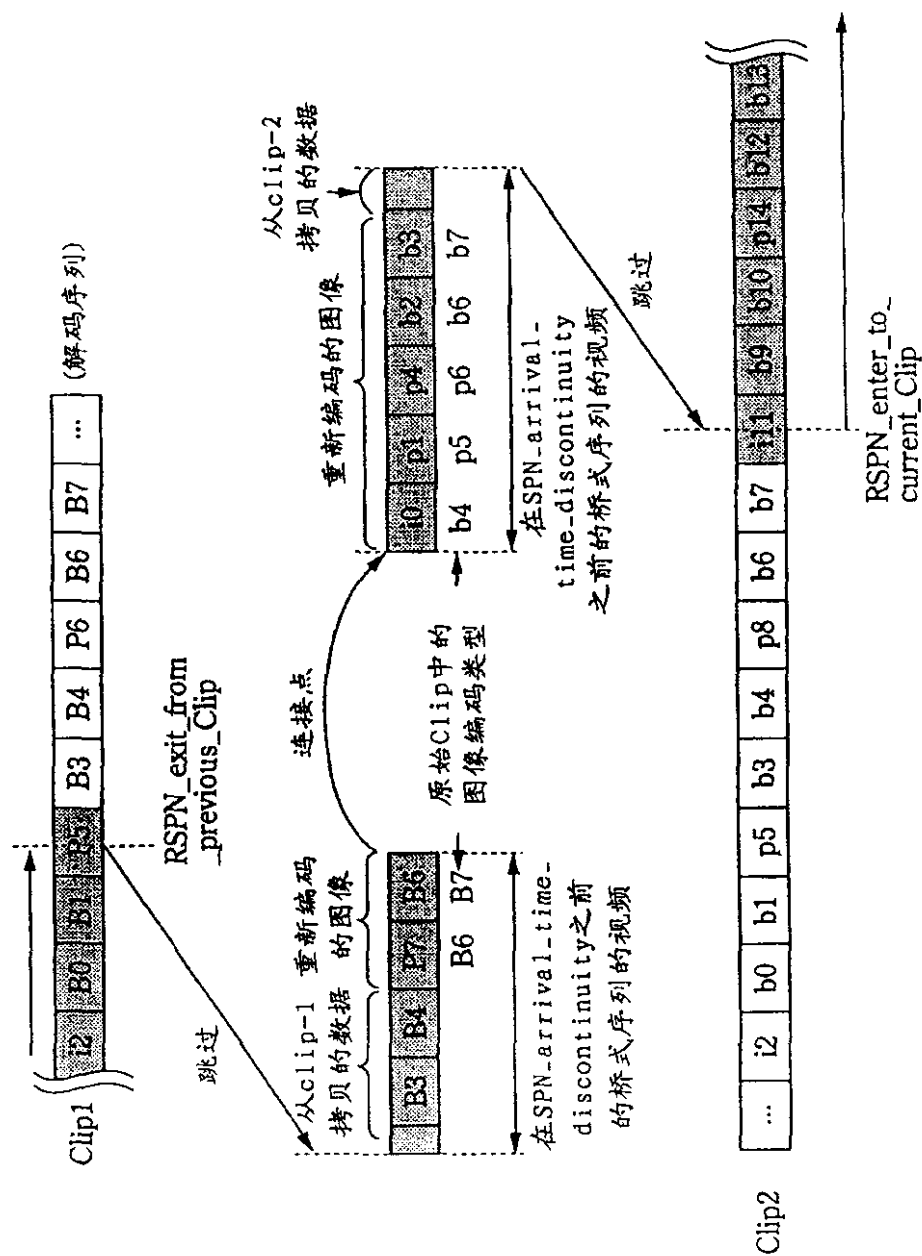


图 99

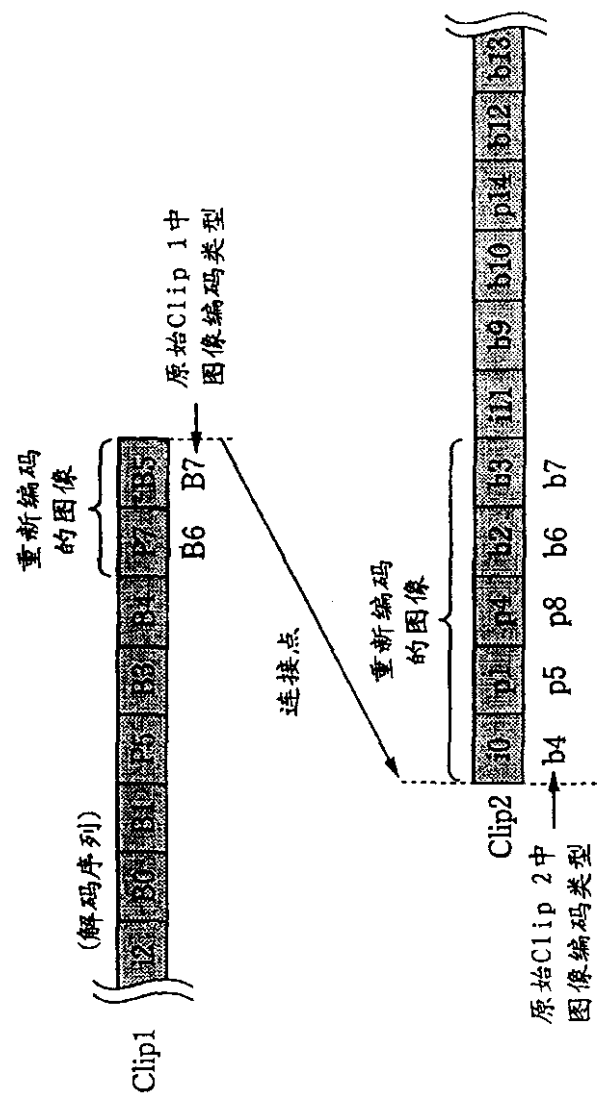


图 100

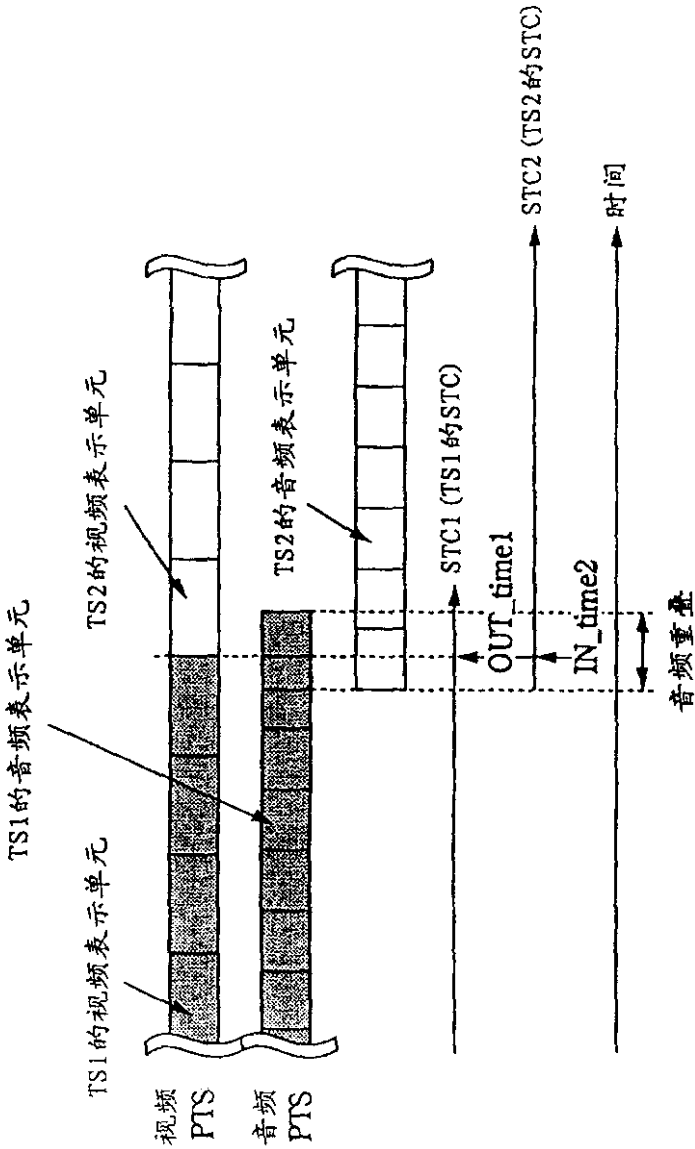


图 101

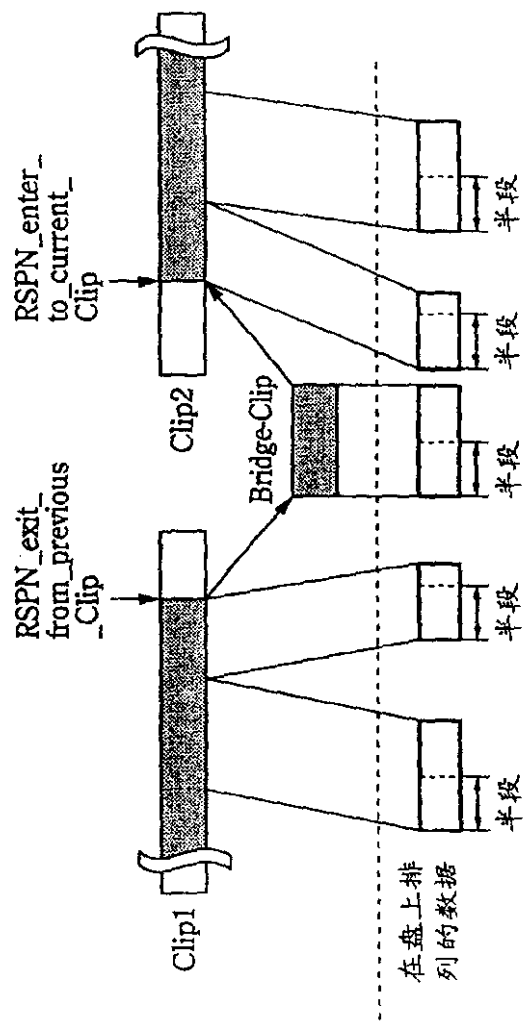


图 102

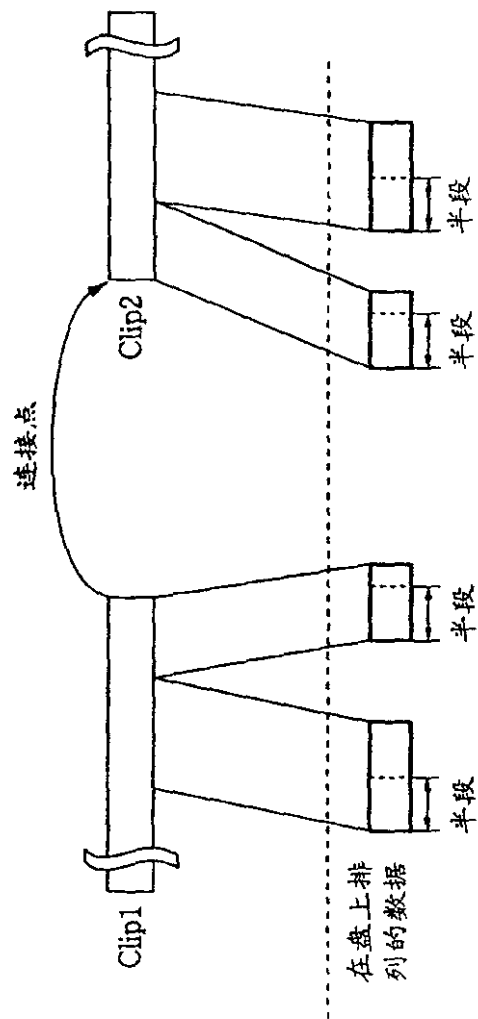


图 103

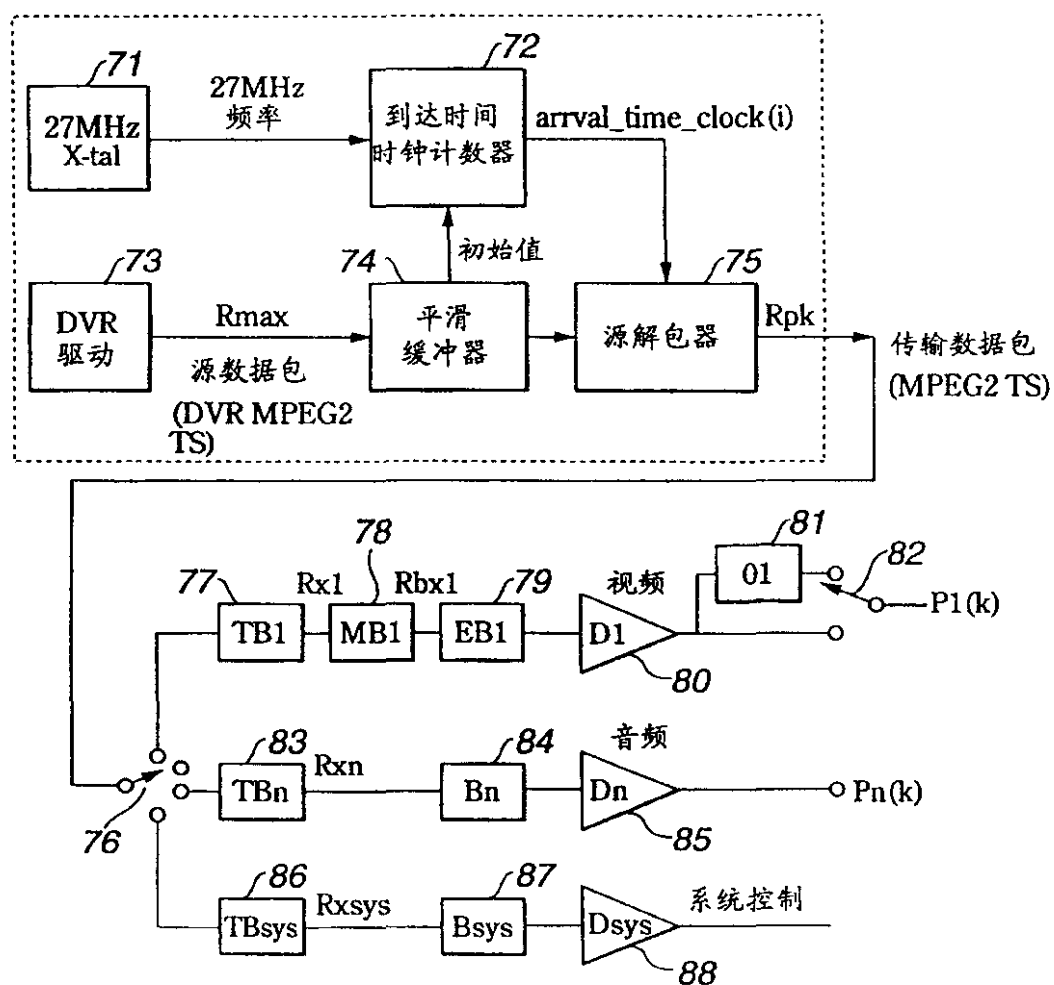


图 104

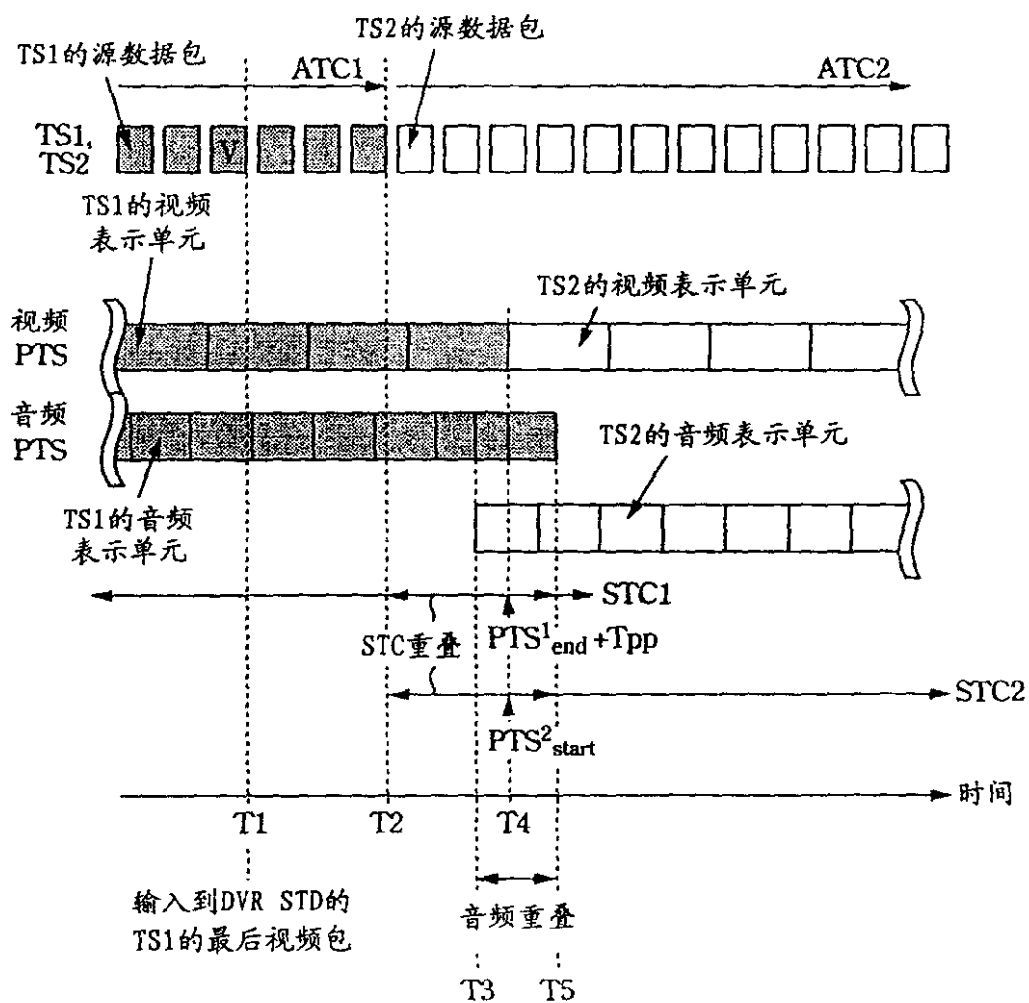


图 105

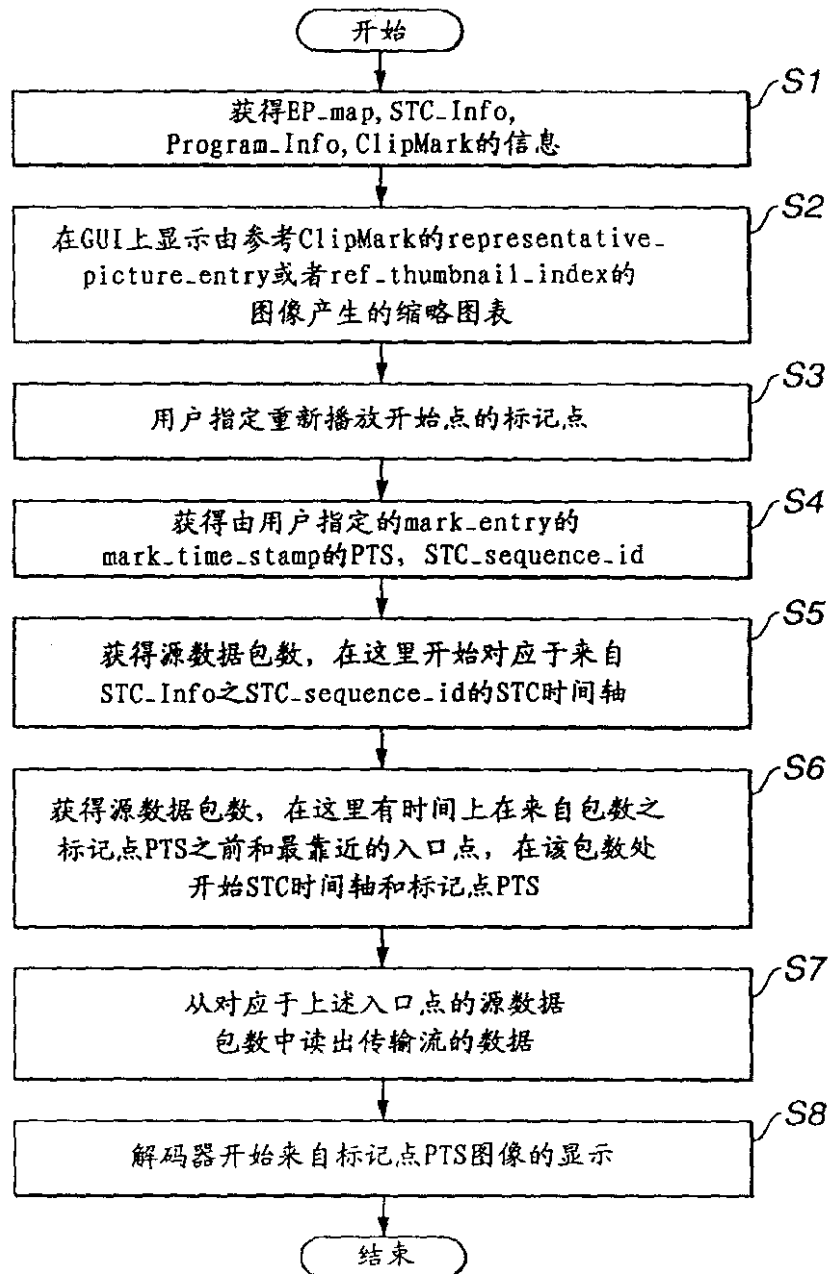


图 106

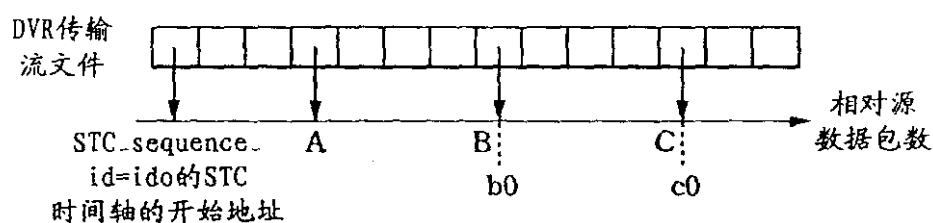


图 107

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

图 108

ClipMark

Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...	...	...	...	...
0x92 (scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94 (CM start)	PTS(b0)	id0	PTS(b0)	id0
0x95 (CM end)	PTS(c0)	id0	PTS(c0)	id0
...	...	...	...	...

图 109

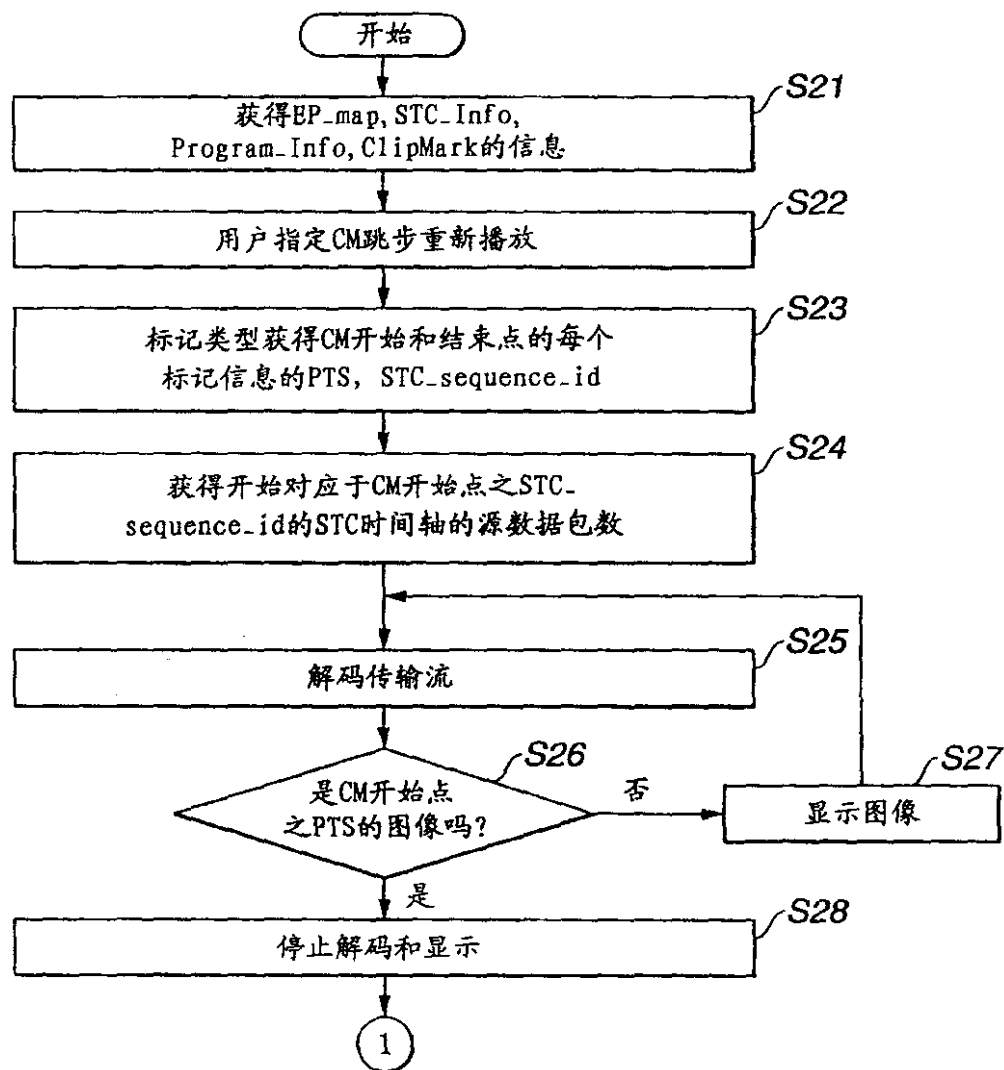


图 110

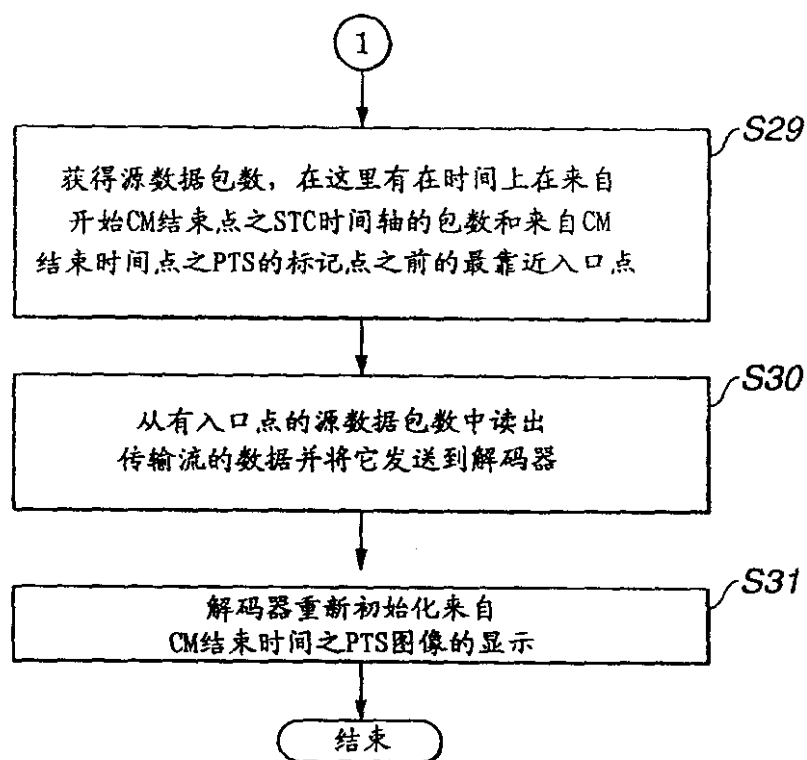


图 111

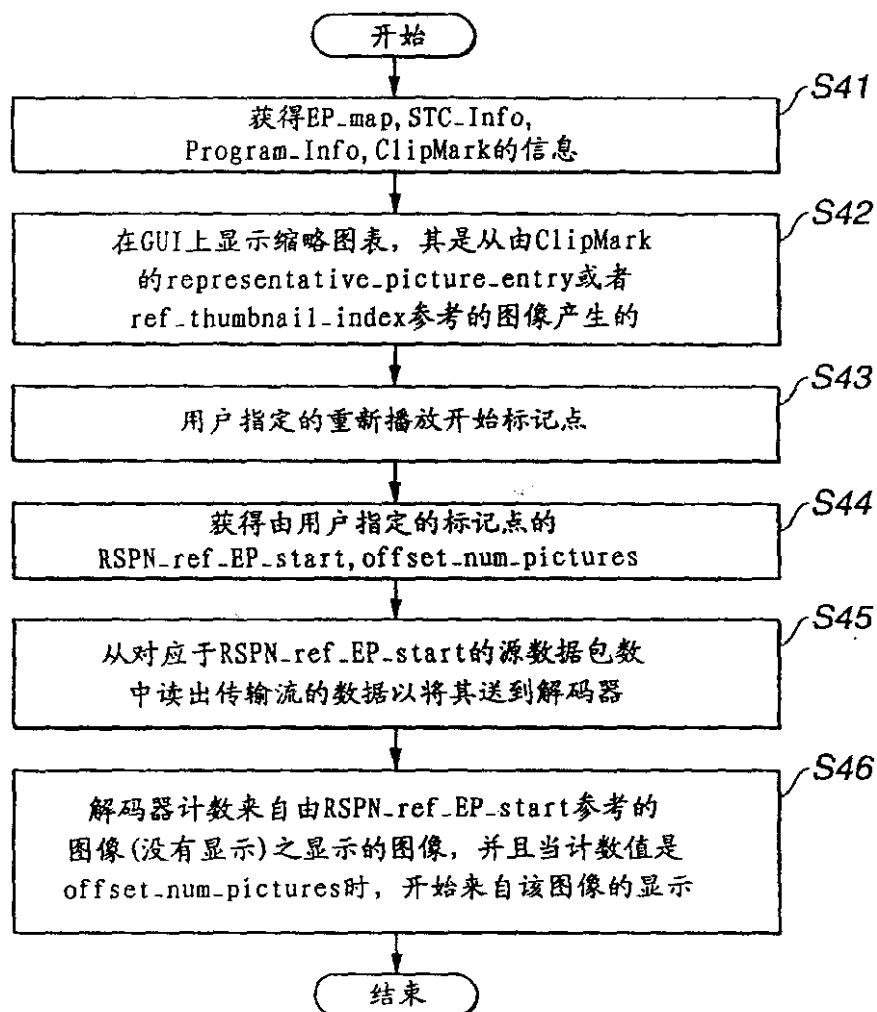


图 112

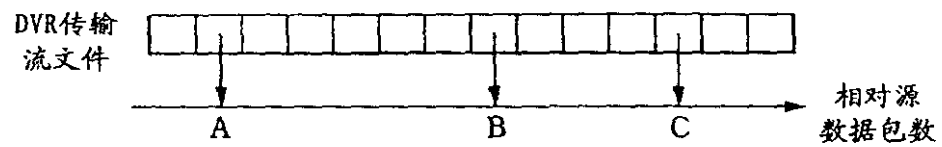


图 113

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

图 114

ClipMark

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...	...	...	...	...
0x92(场景开始)	A	M1	A	M2
0x94(CM开始)	B	N1	B	N1
0x95(CM结束)	C	N2	C	N2
...	...	...	...	...

图 115

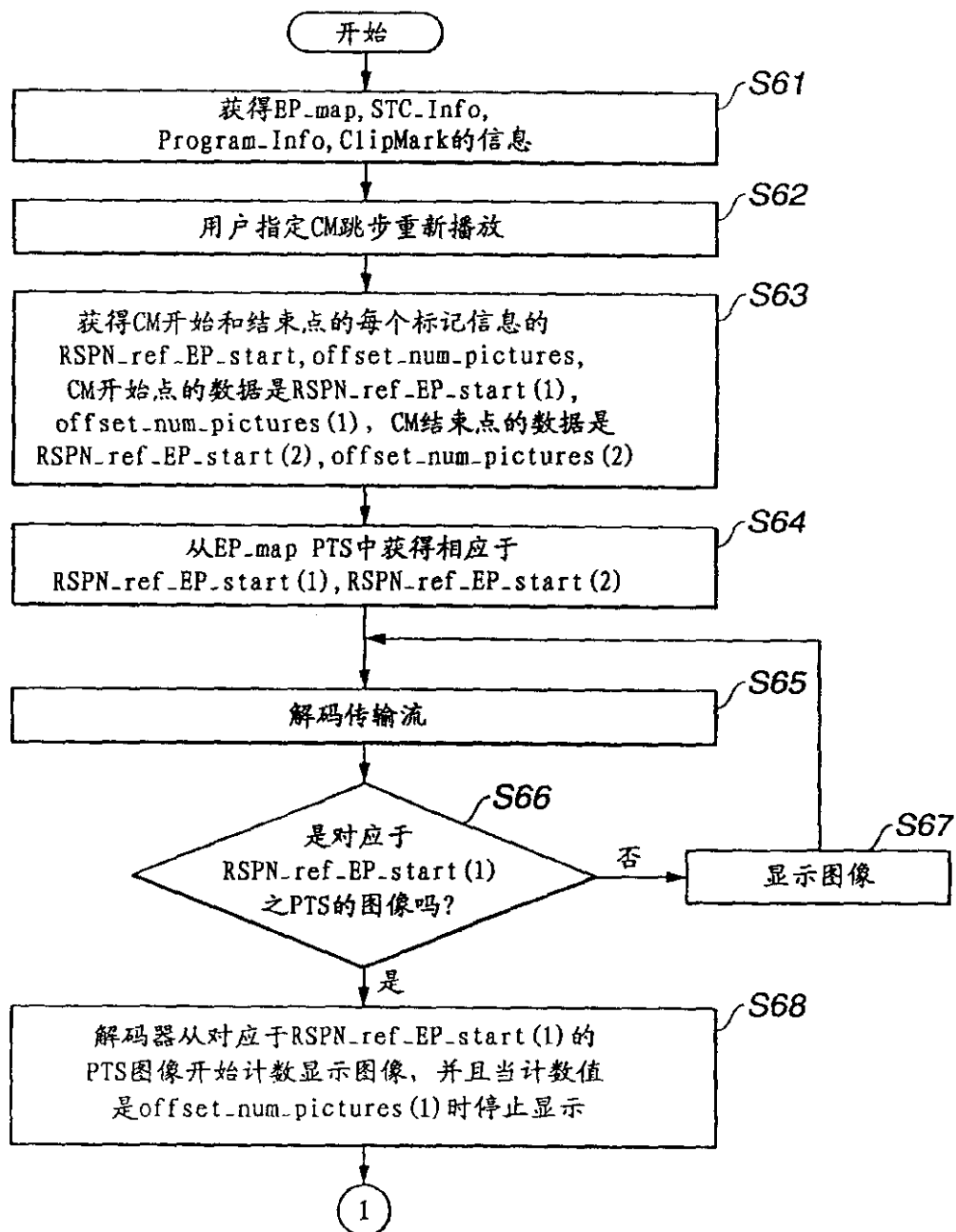


图 116

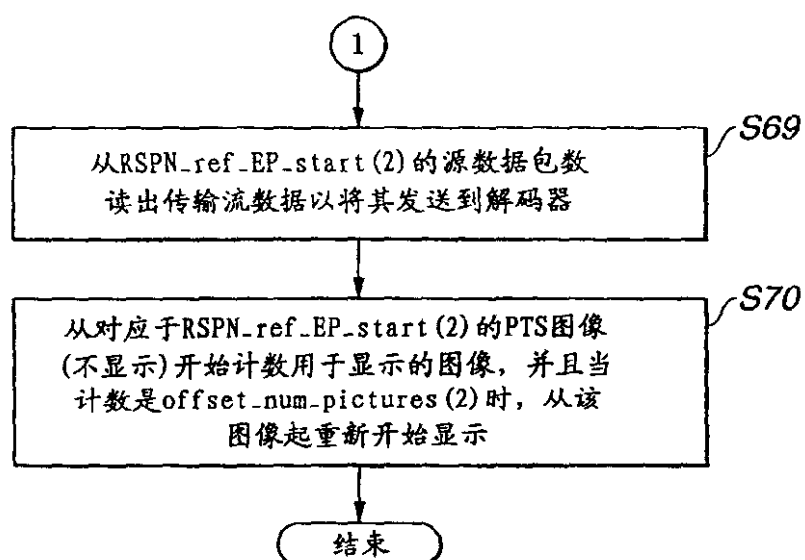


图 117

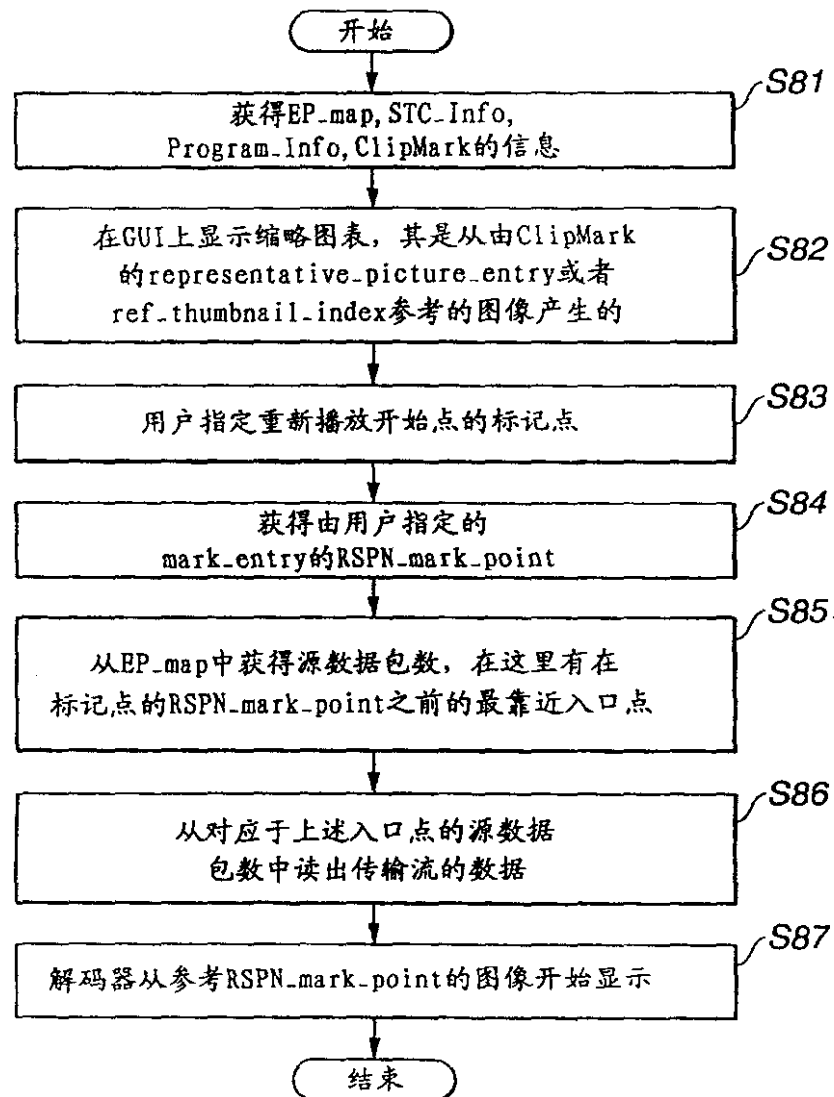


图 118

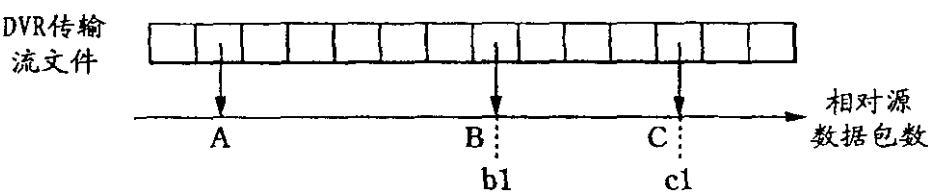


图 119

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

图 120

ClipMark

mark_type	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...	...	...
0x92 (场景开始)	a1	a2
0x94 (CM开始)	b1	b1
0x95 (CM结束)	c1	c1
...	...	...

图 121

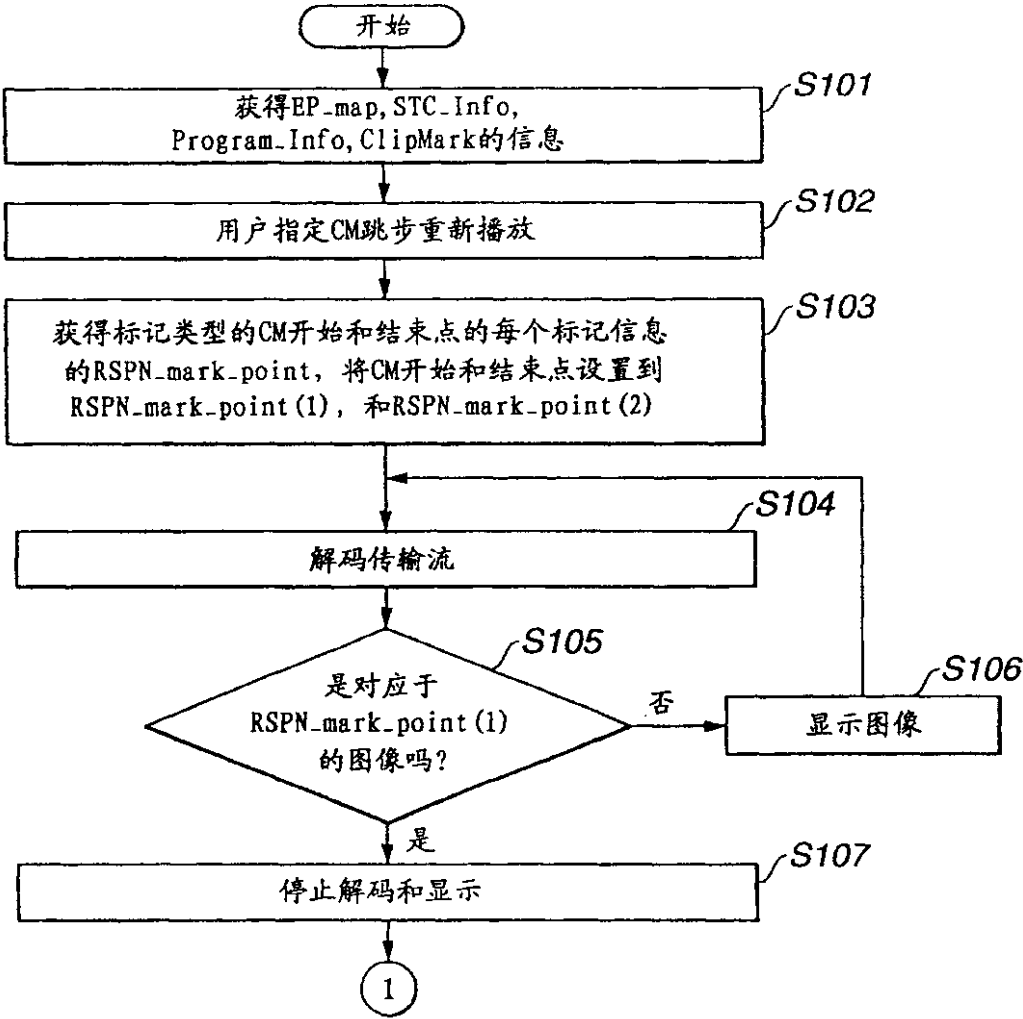


图 122

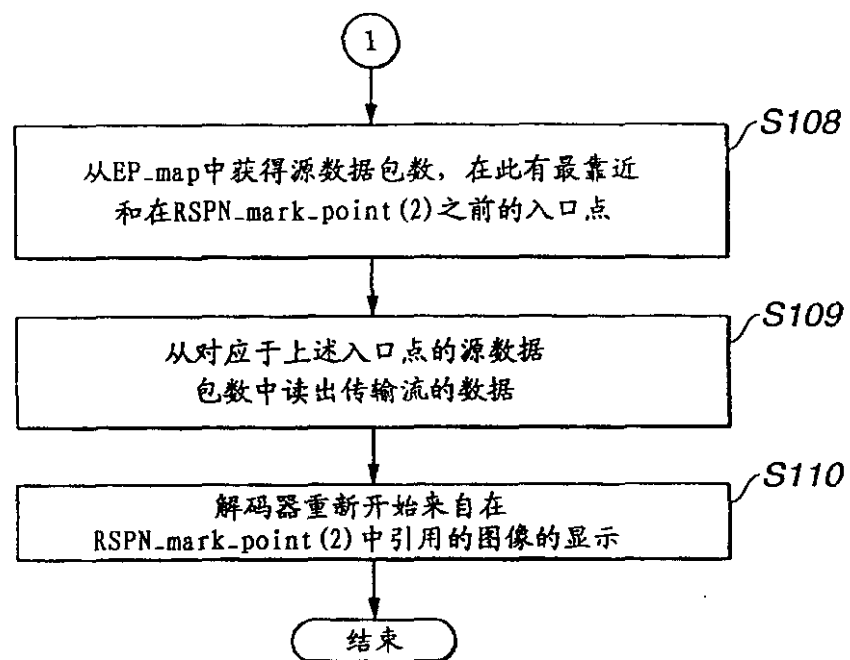


图 123

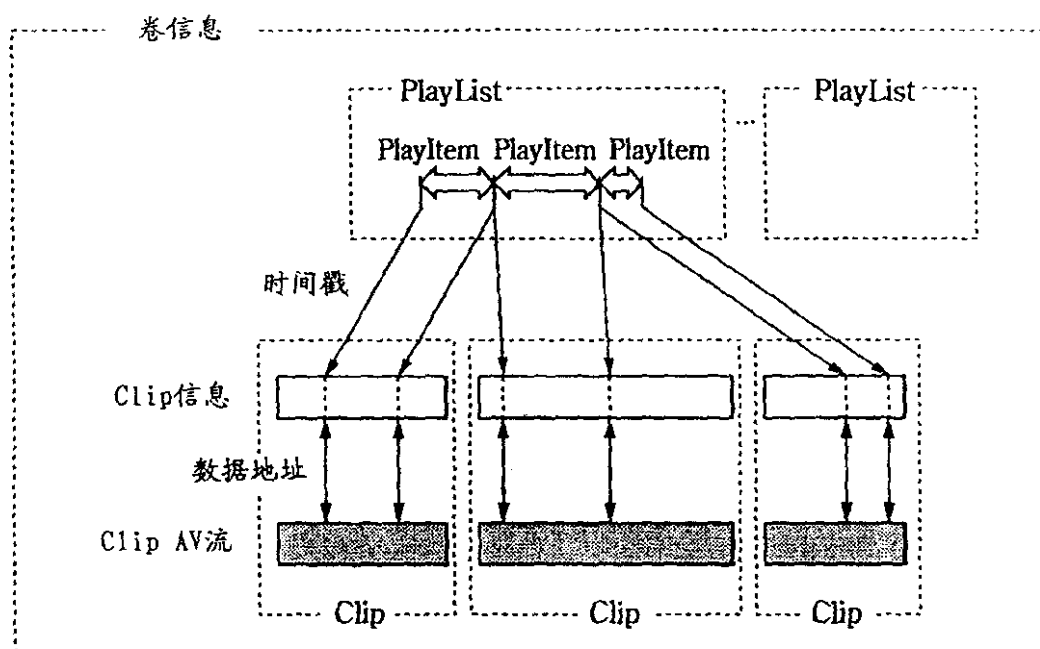


图 124

PlaylistMark
时间戳基标志
(例如: 恢复点, 书签)

ClipMark
地址基指配
(例如: 场景开始点, 场景结束点)

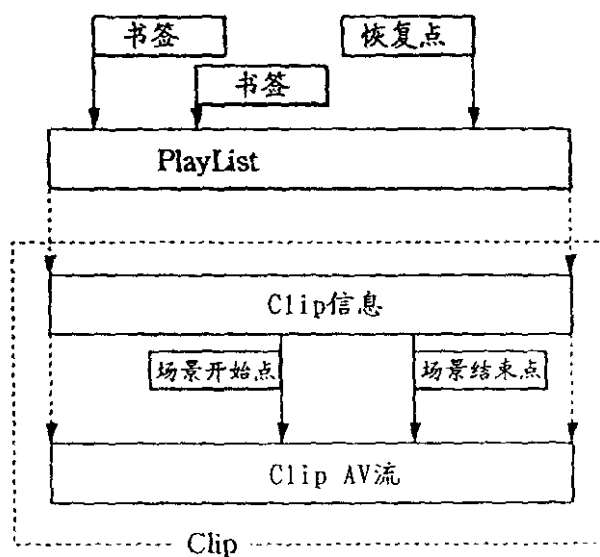


图 125

句法	字节数	缩写
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
保留	8	bslbf
mark_type	8	bslbf
RSPN_mark	32	uimsbf
保留	32	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

图 126

句法	字节数	缩写
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
保留	8	bslbf
mark_type	8	bslbf
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
ref_thumbnail_index	16	uimsbf
}		
}		

图 127

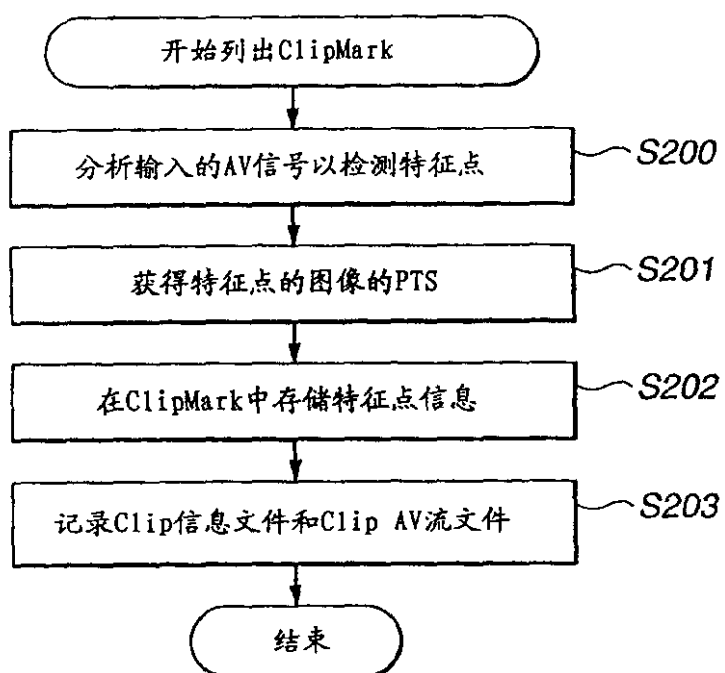


图 128

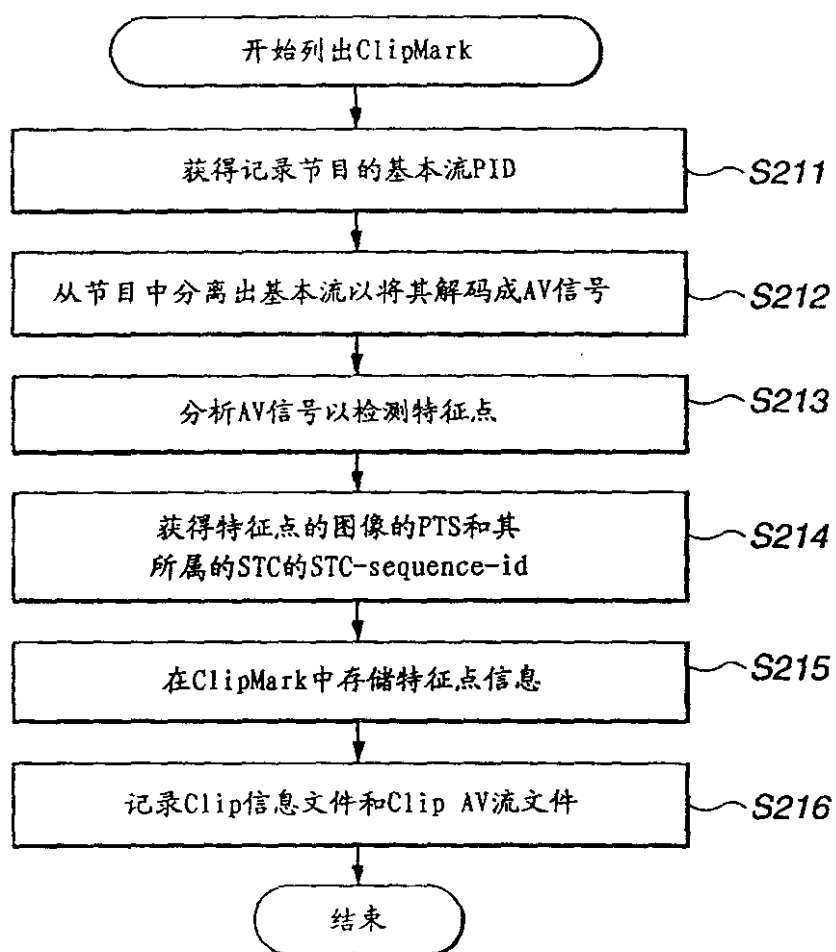


图 129

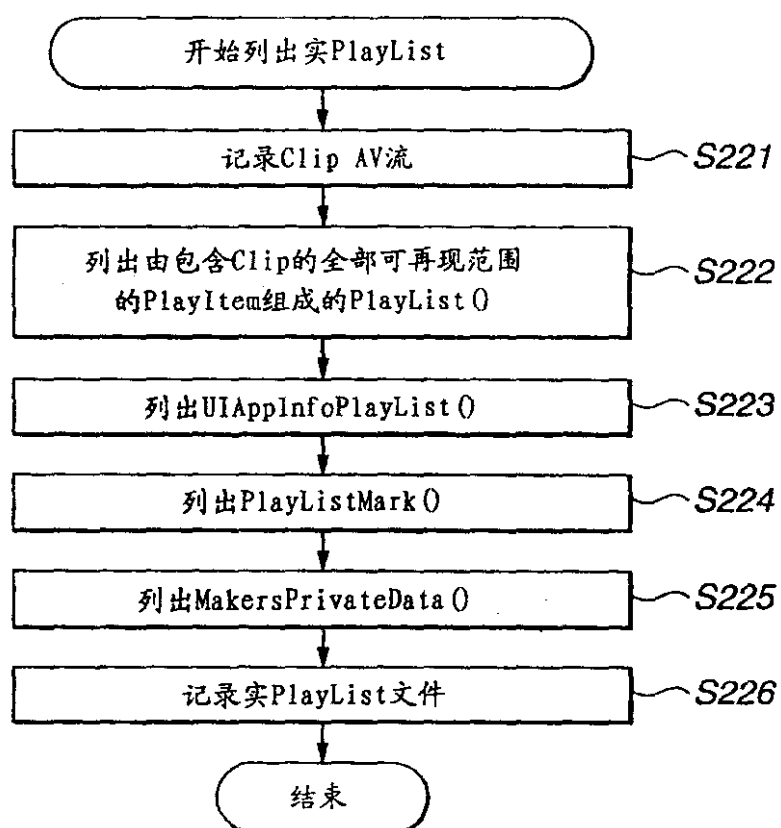


图 130

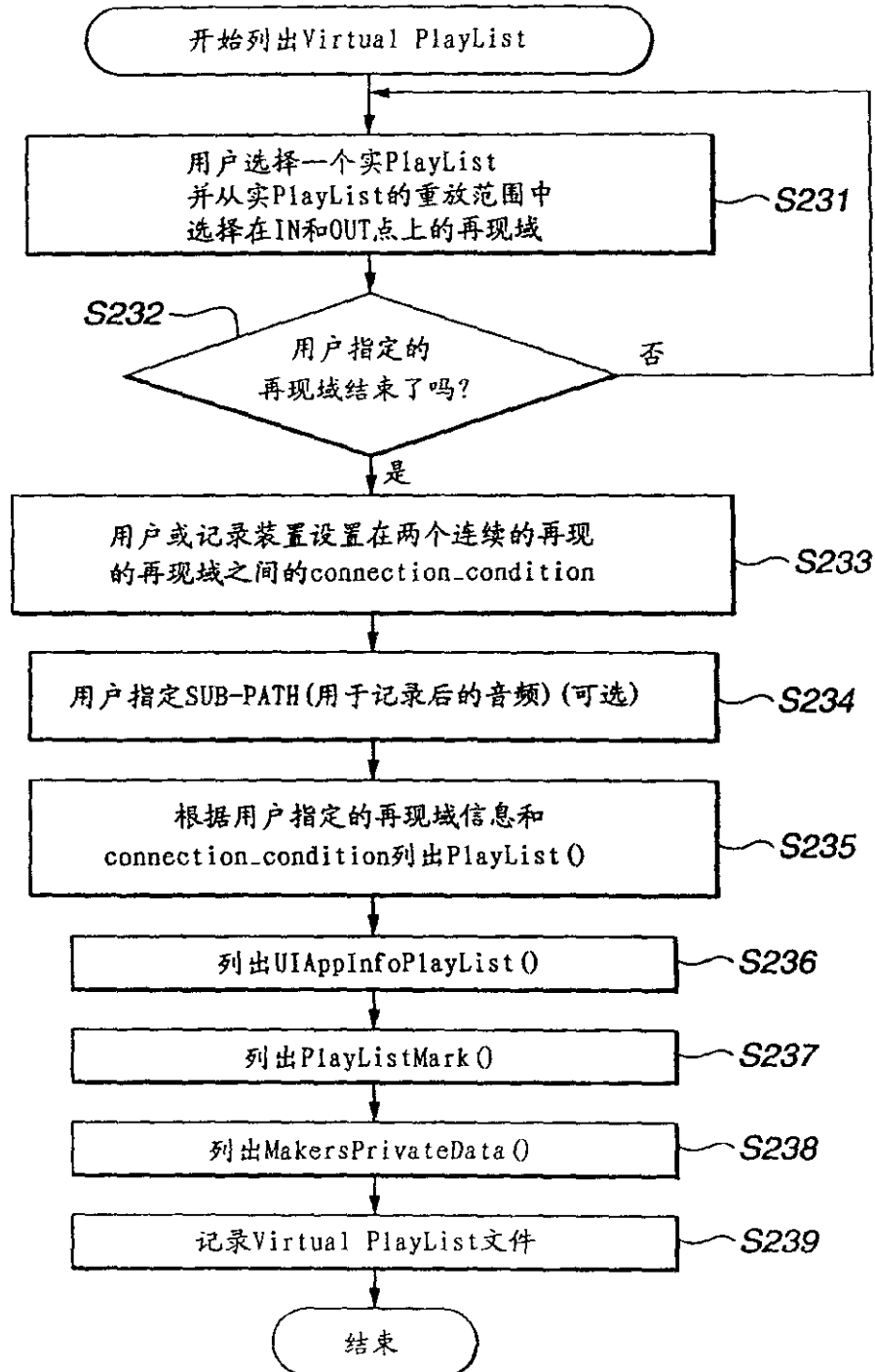


图 131

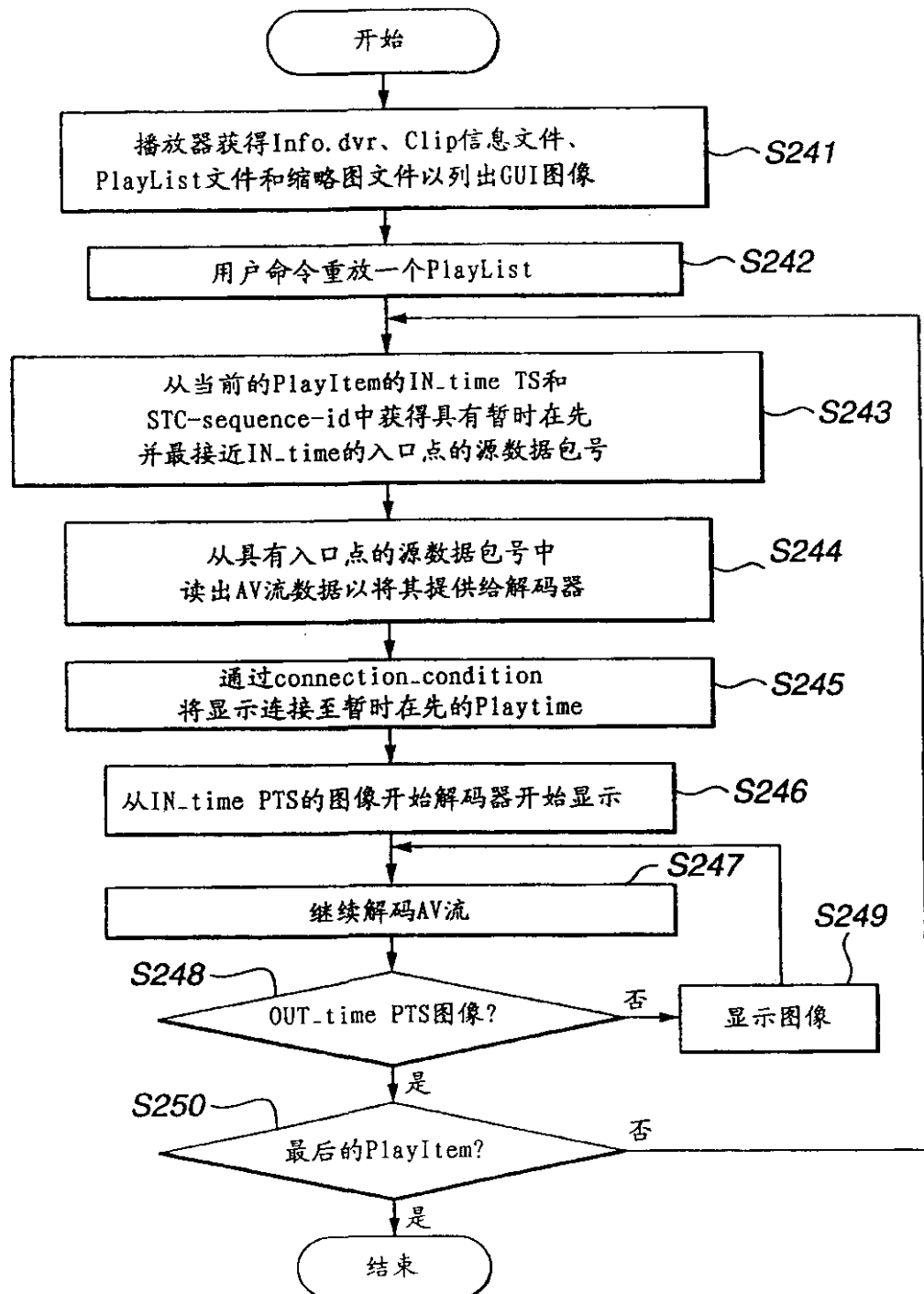


图 132

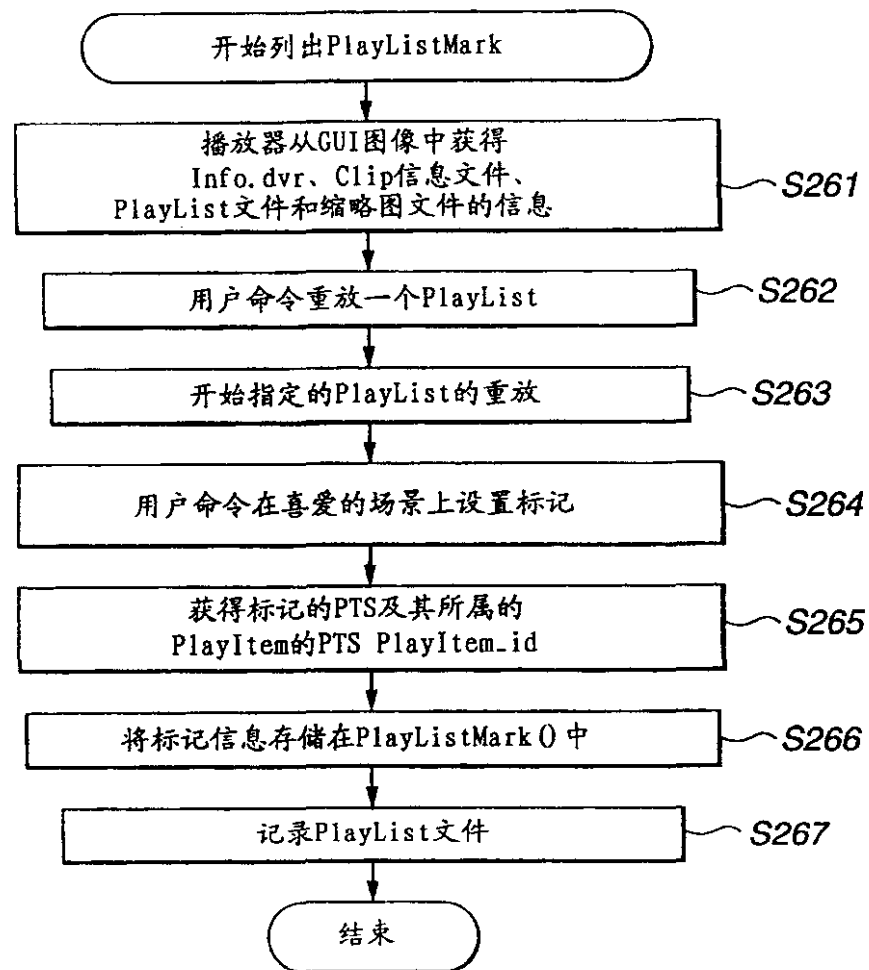


图 133

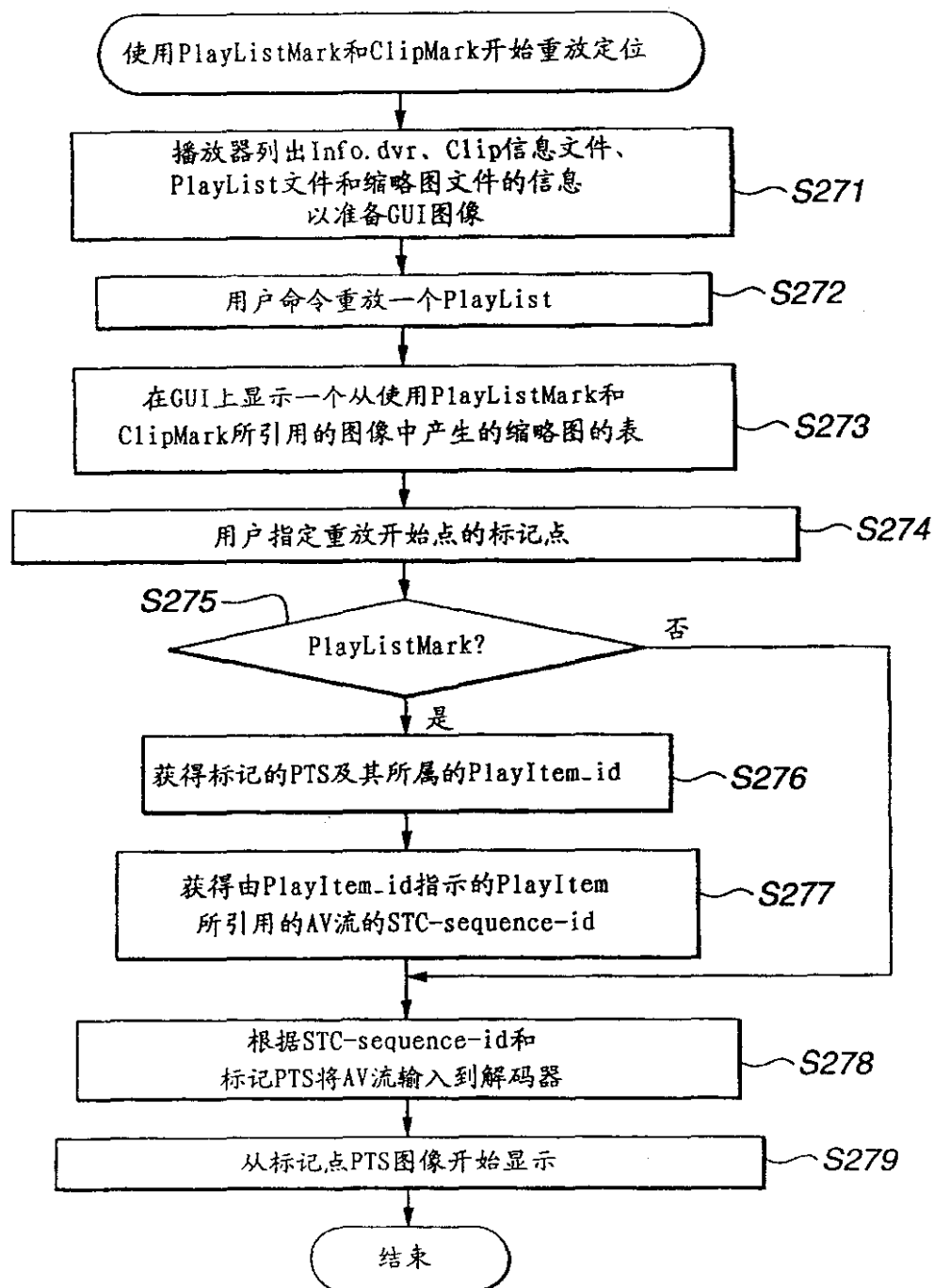


图 134

句法	字节数	缩写
PlayListMark(){		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
mark_name	8*32	bslbf
}		
}		

图 135

值	含义	注释
0x00	恢复-标记	重放恢复点, 在PlayListMark()中定义的重放恢复点数必需是0或1
0x01	书-标记	PlayList的重放入口点。 该标记可由用户设置, 并用做指定喜爱场景的开始点的标记。 该标记的数量可以在PlayListMark()中提供。
0x02	章节-标记	用户使用该标记提示PlayList中的章节开始。 该标记的数量可以在PlayListMARK()中提供。
0x03	跳步-开始-标记	如果在PlayListMark中设置一个跳步-开始-标记, 则在该跳步-开始-标记入口后必须直接设置一个跳步-结束-标记。 用户提示以跳过从跳步-开始-标记的时间戳到跳步-结束-标记的时间戳的PlayList重放。 跳步-开始-标记具有和跳步-结束-标记相同的意义。如果entry-ES-PID不是0xFFFF, 则跳步-开始-标记具有和跳步-结束-标记相同的entry-ES-PID值。 在PlayListMark()中可能存在多个这样的标记。
0x04	跳步-结束-标记	
0x05- 0x3F	为将来 使用保留	为PlayListMark保留
0x40- 0x7F	为ClipMark 保留	

图 136

句法	字节数	缩写
ClipMark(){		
length	32	uimsbf
maker_ID	16	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
ref_to_STC_id	8	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
representative_picture_time_stamp	32	uimsbf
}		
}		

图 137

Mark_type	含义	注释
0x00-0x3F	为将来使用保留	为PlaylistMark保留
0x40	场景-开始-标记	指示场景开始点的标记点
0x41-0x5F	为公共ClipMark保留	
0x60-0x7F	制造商定义的ClipMark	maker_ID指定的 制造商能用自定义含义

图 138

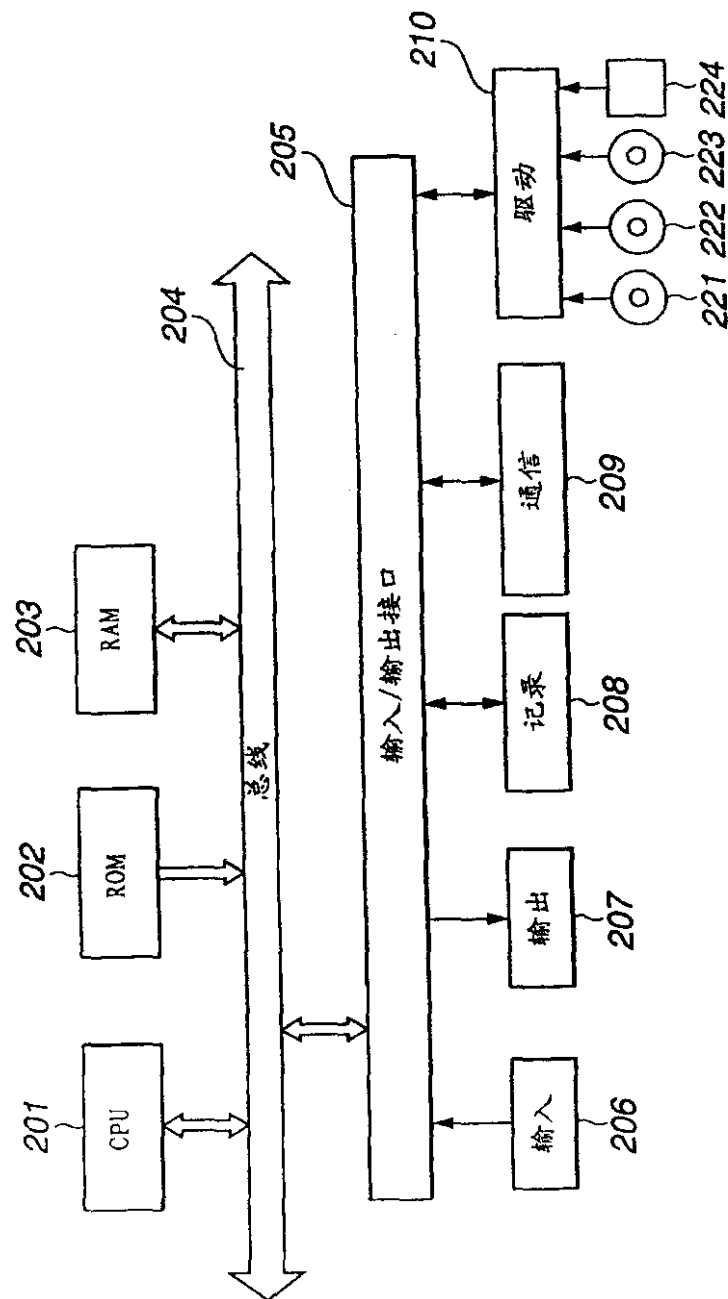


图 139