

3. 姓 名：(中) 各務辰哉
(英) KAKUMU, TATSUYA
國 籍：(中) 日本
(英) JAPAN

4. 姓 名：(中) 上田曉彥
(英) UEDA, AKIHIKO
國 籍：(中) 日本
(英) JAPAN

5. 姓 名：(中) 井原宏二
(英) IHARA, KOJI
國 籍：(中) 日本
(英) JAPAN

● 姓 名：(中) 內海秀介
(英) UTSUMI, SHUSUKE
國 籍：(中) 日本
(英) JAPAN

四、聲明事項：

◎本案申請前已向下列國家（地區）申請專利 ☐ 主張國際優先權：

【格式請依：受理國家（地區）；申請日；申請案號數 順序註記】

1. 日本 ; 2004/12/02 ; 2004-350487 ☒ 有主張優先權

3.姓 名：(中) 各務辰哉
(英) KAKUMU, TATSUYA
國 籍：(中) 日本
(英) JAPAN

4.姓 名：(中) 上田曉彥
(英) UEDA, AKIHIKO
國 籍：(中) 日本
(英) JAPAN

5.姓 名：(中) 井原宏二
(英) IHARA, KOJI
國 籍：(中) 日本
(英) JAPAN

● 姓 名：(中) 內海秀介
(英) UTSUMI, SHUSUKE
國 籍：(中) 日本
(英) JAPAN

四、聲明事項：

◎本案申請前已向下列國家（地區）申請專利 ☐ 主張國際優先權：

【格式請依：受理國家（地區）；申請日；申請案號數 順序註記】

1. 日本 ; 2004/12/02 ; 2004-350487 ☒ 有主張優先權

九、發明說明

【發明所屬之技術領域】

本發明是有關資料記錄裝置及資料記錄方法、資料處理裝置及資料處理方法、程式及程式記錄媒體、資料記錄媒體及資料構造，特別是有關例如可進行高便利性等的資料處理之資料記錄裝置及資料記錄方法、資料處理裝置及資料處理方法、程式及程式記錄媒體、資料記錄媒體及資料構造。

【先前技術】

近年來，大容量，可隨機存取的記錄媒體，例如 DVD(Digital Versatile Disc)已普及，且利用 DVD 來進行各種處理的 DVD 裝置也已廣泛普及。

就 DVD 裝置而言，例如有對 DVD 進行電視廣播節目的資料等的記錄再生之 DVD 記錄器，或在 DVD 中記錄地圖資訊等，而進行該地圖資訊的顯示之汽車衛星導航系統，或在 DVD 中記錄遊戲的程式等，而執行該程式之遊戲裝置等。

另外，有關 DVD 方面，例如在非專利文獻 1「DVD Specifications for Read-Only Disc Part 3；Version 1.1 December 1997」中有其詳細記載。

如 DVD 那樣可處理大量資料的裝置中，會被要求進行高便利性的資料處理。

【發明內容】

本發明是有鑑於如此的狀況而研發者，而使能夠進行高便利性等的資料處理。

本發明之資料記錄裝置的特徵係具備：

檢測部，其係構成選擇視頻資料中所含之成為隨機存取點(random access point)的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定部，其係構成指定藉由檢測部所檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄部，其係構成和視頻資料一起將藉由檢測部所檢測出的複數個圖像的位置資訊、及藉由指定部所指定的圖像的資訊記錄於資料記錄媒體。

可更使具備抽出部，其係抽出藉由上述檢測部所檢測出的複數個圖像的位置資訊中，對應於所定的區段數之圖像的位置資訊；

指定部可根據藉由抽出部所抽出的複數個圖像的位置資訊中，對應於所定的區段數之圖像的位置資訊，來指定藉由檢測部所檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像的位置資訊的圖像；

記錄部更將藉由抽出部所抽出的複數個圖像的位置資訊中，對應於所定的區段數之圖像位置的資訊記錄於資料記錄媒體。

上述所定的區段數可按照再生記錄於資料記錄媒體的

視頻資料的裝置所緩衝視頻資料的容量來設定。

本發明的資料記錄方法的特徵係具備：

檢測步驟，其係選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定步驟，其係指定在檢測步驟的處理所被檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄步驟，其係和視頻資料一起將在檢測步驟的處理所被檢測出的複數個圖像的位置資訊、及在指定步驟的處理所被指定的圖像的資訊記錄於資料記錄媒體。

本發明之第1程式的特徵係使包含下列步驟的處理執行於電腦：

檢測步驟，其係選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定步驟，其係指定在檢測步驟的處理所被檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄步驟，其係和視頻資料一起將在檢測步驟的處理所被檢測出的複數個圖像的位置資訊、及在指定步驟的處理所被指定的圖像的資訊記錄於資料記錄媒體。

本發明之第1程式記錄媒體的特徵係記錄有包含下列步驟的程式：

件以外的構成要件。

又，該記載是對應於發明的實施形態中所被記載的具體例之發明，但並非意味全部記載於申請專利範圍。換言之，該記載是對應於發明的實施形態中所被記載的具體例之發明，而非否定未記載於該申請案的申請專利範圍之發明的存在，亦即將來被分割申請，或藉由補正而追加之發明的存在。

亦即，本發明之資料記錄裝置的特徵係具有：

檢測部(例如，第61圖的多重化器421)，其係構成選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定部(例如，第61圖的控制器425)，其係構成指定藉由檢測部所檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄部(例如，第60圖的碟片驅動器409)，其係構成和上述視頻資料一起將藉由上述檢測部所檢測出的複數個圖像的位置資訊、及藉由上述指定部所指定的圖像的資訊記錄於資料記錄媒體。

本發明之資料記錄方法的特徵係具備：

檢測步驟(例如，第62圖的流程圖的步驟S383的處理)，其係選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定步驟(例如，第62圖的流程圖的步驟S386的處理)，其係指定在檢測步驟的處理所被檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄步驟(例如，第62圖的流程圖的步驟S386的處理)，其係和視頻資料一起將在檢測步驟的處理所被檢測出的複數個圖像的位置資訊、及在指定步驟的處理所被指定的圖像的資訊記錄於資料記錄媒體。

本發明之資料處理裝置的特徵係具備：

讀出部(例如，第1圖的碟片驅動器102)，其係構成和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、及分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，各個僅離幀內圖像所定的距離之圖像的指定資訊；

選擇部(例如，第2圖A，B的播放器控制模組212)，其係構成根據指定藉由讀出部所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離上述幀內圖像所定的距離的圖像之指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像；及

再生部(例如，第2圖A，B的解碼器控制模組214)，其係構成接續於藉由選擇部所選擇的幀內圖像的複數個圖像中，依照幀內圖像及僅離幀內圖像所定的距離的圖像為

態的硬體構成例的方塊圖。

第1圖的碟片再生裝置，例如可適用於碟片播放器，遊戲裝置，汽車衛星導航系統及其他。

在第1圖的碟片再生裝置中，碟片101例如為DVD等的光碟，或光磁碟，磁碟等，記錄有視頻資料，音頻資料，或字幕資料等的內容資料，還有用以再生內容資料所必要的資料。

另外，在記錄於碟片101的資料中，因應所需，亦含有電腦可執行的程式。又，本實施形態中，雖記錄媒體為採用碟片狀的記錄媒體之碟片101，但亦可為其他的記錄媒體，例如半導體記憶體或帶狀的記錄媒體。又，第1圖的碟片再生裝置中，可輸入自遠方的碟片101讀出而傳送來的資料。亦即，來自碟片101的資料之讀出，可使用連接於碟片再生裝置的其他裝置來進行，而以碟片再生裝置來接收該其他裝置所讀出的資料。又，碟片再生裝置，亦可從伺服器經由網際網路等的網路來接受資料的配信，而進行處理。該伺服器是將與記錄於碟片101的資料同樣的資料記憶於儲存器(storage)。又，碟片再生裝置，亦可接收來自伺服器以外的其他裝置的資料，一旦記錄於碟片101之後，處理記錄於該碟片101的資料。

碟片101可裝卸於碟片驅動器102。碟片驅動器102內藏未圖示的介面，經由該介面來連接至驅動器介面114。碟片驅動器102會進行驅動裝著於該處的碟片101，按照來自驅動器介面114的讀出等的命令，從碟片101讀出資料，

而供應給驅動器介面 114 等的處理。

在匯流排 111 連接 CPU (Central Processing Unit) 112，記憶體 113，驅動器介面 114，輸入介面 115，視頻解碼器 116，音頻解碼器 117，視頻輸出介面 118，及音頻輸出介面 119。

CPU 112 及記憶體 113 是形成電腦系統。亦即，CPU 112 會執行記憶於記憶體 113 的程式，即後述的軟體模組群，控制碟片再生裝置全體，且進行後述的各種處理。記憶體 113 會記憶 CPU 112 所執行的軟體模組群。並且，記憶體 113 會一時記憶 CPU 112 的動作上必要的資料。另外，記憶體 113 可只以非揮發性記憶體，或揮發性記憶體與非揮發性記憶體的組合來構成。此外，在第 1 圖的碟片再生裝置中設置硬碟，事先對該硬碟記錄(安裝)CPU 112 所執行的軟體模組群時，記憶體 113 可只以揮發性記憶體來構成。

在此，CPU 112 所執行的程式(軟體模組群)可預先記錄(記憶)於碟片再生裝置中所內藏的記錄媒體，亦即記憶體 113。

或者，程式亦可一時或永續儲存於碟片 101，或碟片 101 以外的軟碟，CD-ROM (Compact Disc Read Only Memory)，MO (Magneto Optical) 碟片，磁碟，記憶卡等的可移動 (removable) 記錄媒體。如此的可移動記錄媒體可提供所謂的套裝軟體。

又，程式可事先使記憶於記憶體 113，或從上述的可移動記錄媒體來安裝於碟片再生裝置。又，程式可從下載

位置經由數位衛星廣播用的人造衛星，以無線來傳送至碟片再生裝置，或者經由 LAN(Local Area Network)，網際網路 (internet) 等的網路，以有線來傳送至碟片再生裝置，在碟片再生裝置中，以輸入介面 115 來接收如此傳送來的程式，且安裝於內藏的記憶體 113。

又，程式可藉由 1 個 CPU 來處理者，或藉由複數個 CPU 來分散處理者。

驅動器介面 114 是在 CPU112 的控制之下，控制碟片驅動器 102，藉此，碟片驅動器 102 會經由匯流排 111 來將從碟片 101 讀出的資料予以供應給 CPU112，記憶體 113，視頻解碼器 116，或音頻解碼器 117 等。

輸入介面 115 會接收藉由使用者操作未圖示的鍵 (按鈕) 或遙控器而被供給的信號，經由匯流排 111 來供給至 CPU112。另外，輸入介面 115 除此以外，例如亦具有作為數據機 (包含 ADSL (Asymmetric Digital Subscriber Line) 數據機) 或 NIC (Network Interface Card) 等的通信介面之功能。

視頻解碼器 116 是將藉由碟片驅動器 102 來從碟片 101 讀出，且經由驅動器介面 114 及匯流排 111 來供給之編碼視頻資料予以解碼，並經由匯流排 111 來將其結果取得的視頻資料予以供應給 CPU112 或視頻輸出介面 118。

音頻解碼器 117 是將藉由碟片驅動器 102 來從碟片 101 讀出，且經由驅動器介面 114 及匯流排 111 來供給之編碼音頻資料) 予以解碼，並經由匯流排 111 來將其結果取得的音

頻資料予以供應給 CPU112或音頻輸出介面 119。

視頻輸出介面 118會對經由匯流排 111而被供給的視頻資料施以必要的處理，從視頻輸出端子 120輸出。音頻輸出介面 119會對經由匯流排 111而被供給的音頻資料施以必要的處理，從音頻輸出端子 121輸出。

視頻輸出端子 120會被連接至未圖示的 CRT(Cathode Ray Tube)或液晶面板等的視頻輸出裝置，因此，從視頻輸出端子 120所被輸出的視頻資料會被供給至視頻輸出裝置而顯示。音頻輸出端子 121會被連接至未圖示的喇叭或擴大器等音頻輸出裝置，因此，從音頻輸出端子 121所被輸出的音頻資料會被供給至音頻輸出裝置而輸出。

另外，從碟片再生裝置往視頻輸出裝置及音頻輸出裝置的視頻資料及音頻資料之供給可藉由有線或無線來進行。

[軟體模組群的構成]

其次，第 2圖 A 及第 2圖 B 是表示第 1圖的 CPU112所執行之軟體模組群的構成例。

CPU112所執行的軟體模組群大致分成操作系統(OS)201，及應用程式的視頻內容再生程式 210。

「操作系統 201」

一旦碟片再生裝置的電源被開啓，則最初操作系統 201會起動，進行初期設定等的必要處理，叫出應用程式

的視頻內容再生程式 210。

操作系統 201 會對視頻內容再生程式 210 提供檔案的讀出等之基礎 ((infrastructure)) 性的服務。亦即，操作系統 201，例如有關檔案的讀出，是針對來自視頻內容再生程式 210 之檔案的讀出要求，經由驅動器介面 114 來操作碟片驅動器 102，而讀出碟片 101 的資料，提供給視頻內容再生程式 210 的服務。又，操作系統 201 也會進行檔案系統的解釋等。

另外，操作系統 201 具備多路處理的功能，可以時間分割來使複數個軟體模組動作。亦即，視頻內容再生程式 210 雖是以幾個軟體模組來構成，但各軟體模組可並列動作。

「視頻內容再生程式 210」

視頻內容再生程式 210 是由腳本控制模組 211，播放器控制模組 212，內容資料供給模組 213，解碼控制模組 214，緩衝控制模組 215，視頻解碼器控制模組 216，音頻解碼器控制模組 217，字幕解碼器控制模組 218，繪圖 (graphics) 處理模組 219，視頻輸出模組 220，及音頻輸出模組 221 所構成。

視頻內容再生程式 210 是在碟片 101 的再生時發揮中心任務的軟體，一旦碟片 101 被裝著 (插入) 碟片驅動器 102，則會確認該碟片 101 是否為記錄有內容之後述格式的碟片。又，視頻內容再生程式 210 會從碟片 101 讀出後述的腳本

檔案而執行，且從碟片 101 讀出爲了再生記錄於該碟片 101 的內容所必要的元資料 (metadata) 的檔案，根據該元資料來控制內容的再生。

以下，說明有關構成第 2 圖 A 及第 2 圖 B 的視頻內容再生程式 210 之軟體模組。並且，在第 2 圖 A 及第 2 圖 B 中，原則上，實線的箭形符號是表示內容的資料，點線的箭形符號是表示控制的資料。

「腳本控制模組 211」

腳本控制模組 211 是解釋記錄於碟片 101 的腳本檔案中所記述的腳本程式 (腳本) 而執行。在腳本程式中，例如可記述「操作繪圖處理模組 219，作成選單等的畫像，而進行顯示」，「按照來自遙控器等的 UI (User Interface) 之信號，變更選單的顯示 (例如，移動選單上的游標等)」，「控制播放器控制模組 212」等之動作。

「播放器控制模組 212」

播放器控制模組 212 是參照記錄於碟片 101 的元資料等來進行有關內容的再生之控制。亦即，播放器控制模組 212，例如會解析記錄於碟片 101 之後述的 PlayList() 或 Clip()，按照該解析結果，控制內容資料供給模組 213，解碼控制模組 214，或緩衝控制模組 215。又，播放器控制模組 212 會按照來自腳本控制模組 211 或輸入介面 115 的指示，進行切換再生對象的串流之後述的串流切換等的控制。

又，播放器控制模組 214 會從解碼控制模組 214 取得時刻，進行時刻顯示或後述的標示 (Mark()) 之處理等。

「內容資料供給模組 213」

內容資料供給模組 213 是按照播放器控制模組 212 的控制，或根據積蓄於緩衝控制模組 215 的資料量，對操作系統 201 要求來自碟片 101 之內容資料或元資料等的讀出。

另外，操作系統 201 所按照來自內容資料供給模組 213 的要求而從碟片 101 讀出的元資料等會被供應給必要的模組。又，操作系統 201 按照來自內容資料供給模組 213 的要求而從碟片 101 讀出的內容資料會被供應給緩衝控制模組 215。

「解碼控制模組 214」

解碼控制模組 214 是按照來自播放器控制模組 212 的控制，而控制視頻解碼器控制模組 216，音頻解碼器控制模組 217，及字幕解碼器控制模組 218 的動作。又，解碼控制模組 214 內藏計時時刻的計時部 214A，管理藉由視頻解碼器控制模組 216 的控制而輸出之視頻資料的輸出，及應與該視頻資料同步輸出之資料 (輸出資料) 的輸出，亦即，在此是與藉由音頻解碼器控制模組 217 的控制而輸出之音頻資料的輸出同步。

計時部 214A 可藉由計數自外部供給的基準時脈或同步於解碼器等的內部時脈，來自律性的計時。

但，藉由控制第2圖 A 及第2圖 B 所示的各種模組來實現之軟體基礎的解碼器中，有別於該等的處理，另外利用軟體來使獨立的計時處理執行，將會造成 CPU112的處理負荷增大。因應於此，要求計時部 214A 按照來自解碼器的視頻資料的輸出而更新時刻。

第3圖是表示計時部 214A 為獨立的時鐘時的時刻與實際的時間經過之關係。在第3圖中，計時部 214A 是以 90kHz 的時脈來計數完畢，因此從時刻 0 起以右上升的直線增加，在時刻 33.3 毫秒的瞬間，時鐘所示的時刻係形成 3003。

第4圖是表示計時部 214A 按照視頻解碼器的視頻資料的輸出來更新時刻的時鐘時的時刻之一例。第4圖的情況，時刻是例如若經過 33.3 毫秒，則時鐘的輸出時刻會被更新成 3003，若經過 83.3 毫秒，則輸出時刻會被更新成 7507，且若經過 116 毫秒，則輸出時刻會被更新成 10510。在此，1 幀的輸出時間為 16.66 毫秒。

另外，在第3圖中，實際上時刻是以 1/90kHz 的分解能呈現的階段狀推移，但為了與第4圖作對比，而以直線來表現。並且，在往後中，計時部 214A 是以參照第4圖所說明之按照視頻資料的輸出來更新時刻之方式進行說明。

「緩衝控制模組 215」

緩衝控制模組 215 內藏第1圖的記憶體 113 的記憶領域的一部份之緩衝器 215A，在該緩衝器 215A 中，內容資料

供給模組 213 會對操作系統 201 進行要求，一時記憶從碟片 101 讀出之內容的資料。

又，緩衝控制模組 215 會按照視頻解碼器控制模組 216，音頻解碼器控制模組 217，或字幕解碼器控制模組 218 的要求，將記憶於緩衝器 215A 的資料予以供應給視頻解碼器控制模組 216，音頻解碼器控制模組 217，或字幕解碼器控制模組 218。

亦即，緩衝控制模組 215 內藏後述之第 5 圖所說明的視頻讀出功能部 233，音頻讀出功能部 234，及字幕讀出功能部 235。然後，緩衝控制模組 215 會在視頻讀出功能部 233 處理來自視頻解碼器控制模組 216 之資料的要求，藉此來將記憶於緩衝器 215A 的資料供應給視頻解碼器控制模組 216。同樣的，緩衝控制模組 215 會在音頻讀出功能部 234 處理來自音頻解碼器控制模組 217 之資料的要求，藉此來將記憶於緩衝器 215A 的資料供應給音頻解碼器控制模組 217，且在字幕讀出功能部 235 處理來自字幕解碼器控制模組 218 之資料的要求，藉此來將記憶於緩衝器 215A 的資料供應給字幕解碼器控制模組 218。

「視頻解碼器控制模組 216」

視頻解碼器控制模組 216 會操作緩衝控制模組 215 內的視頻讀出功能部 233 (第 5 圖)，而以視頻存取單元單位來從緩衝控制模組 215 的緩衝器 215A 讀出編碼視頻資料後的資料 (視頻編碼資料)，且供應給第 1 圖的視頻解碼器 116。又

，視頻解碼器控制模組 216 會控制視頻解碼器 116，使視頻存取單元單位的資料解碼。又，視頻解碼器控制模組 216 會將視頻解碼器 116 的解碼結果所取得的視頻資料供應給繪圖處理模組 219。

在此，所謂視頻存取單元是例如為視頻資料的 1 圖像 (picture)(1 幀 (frame) 或 1 場 (field)) 份。

「音頻解碼器控制模組 217」

音頻解碼器控制模組 217 會操作緩衝控制模組 215 內的音頻讀出功能部 234 (第 5 圖)，而以音頻存取單元單位來從緩衝控制模組 215 的緩衝器 215A 讀出編碼音頻資料後的資料 (音頻編碼資料)，且供應給第 1 圖的音頻解碼器 117。又，音頻解碼器控制模組 217 會控制音頻解碼器 117，使音頻存取單元單位的資料解碼。又，音頻解碼器控制模組 217 會將音頻解碼器 117 的解碼結果所取得的音頻資料供應給音頻輸出模組 221。

在此，所謂音頻存取單元是音頻資料的所定資料量份 (例如，與 1 圖像同步輸出的份量)。在本實施形態中，音頻存取單元是例如為既知的固定長。

「字幕解碼器控制模組 218」

字幕解碼器控制模組 218 會操作緩衝控制模組 215 內的字幕讀出功能部 235 (第 5 圖)，而以字幕存取單元單位來從緩衝控制模組 215 的緩衝器 215A 讀出編碼字幕資料後的資

料(字幕編碼資料)。又，字幕解碼器控制模組218在內部具備未圖示的字幕解碼軟體，解碼由緩衝器215A讀出的資料。又，字幕解碼器控制模組218會將該解碼結果所取得的字幕資料(字幕的畫像資料)供應給繪圖處理模組219。

在此，所謂字幕存取單元是字幕資料的所定資料量份(例如，與1圖像同步輸出的份量)。在本實施形態中，字幕存取單元的大小，例如會被記述於該字幕存取單元的前頭。

「繪圖處理模組219」

繪圖處理模組219是按照播放器控制模組212的指示，進行來自字幕解碼器控制模組218之字幕資料的擴大或縮小，使字幕資料重疊於來自視頻解碼器控制模組216的視頻資料。又，繪圖處理模組219會進行供以使與字幕資料重疊後的視頻資料的圖框配合連接於第1圖的視頻輸出端子120的視頻輸出裝置的顯示畫面之擴大或縮小等，且將該結果所取得的視頻資料輸出至視頻輸出模組220。

又，繪圖處理模組219會按照腳本控制模組211或播放器控制模組212的指示來產生選單或訊息等，重疊於輸出視頻資料。

又，繪圖處理模組219會根據連接於第1圖的視頻輸出端子120之視頻輸出裝置的寬高比(Aspect Ratio)及指示記錄於碟片101之視頻資料的寬高比的資訊等來進行輸出至

視頻輸出模組 220 之視頻資料的寬高比的變換。

亦即，例如，在視頻輸出裝置的寬高比為 16：9 時，當指示視頻資料的寬高比之資訊為顯示 4：3 的寬高比時，繪圖處理模組 219 會使輸出至視頻輸出模組 220 的視頻資料在橫方向(水平方向)進行壓縮(Squeeze)處理，加入使黑色顯示於左右的資料而輸出。又，例如，在視頻輸出裝置的寬高比為 4：3 時，當指示視頻資料的寬高比之資訊為顯示 16：9 的寬高比時，繪圖處理模組 219 會使輸出至視頻輸出模組 220 的視頻資料在縱方向(垂直方向)進行壓縮處理，加入使黑色顯示於上下的資料而輸出。

另外，視頻輸出裝置的寬高比，及指示視頻資料的寬高比之資訊所顯示的寬高比，皆為 4：3 或 16：9，亦即同樣時，繪圖處理模組 219 不會對輸出至視頻輸出模組 220 的視頻資料進行壓縮處理，而原封不動輸出。

其他，繪圖處理模組 219，例如會按照來自播放器控制模組 212 的要求，擷取現在處理中的視頻資料。又，繪圖處理模組 219 會記憶該擷取後的視頻資料，或供應給播放器控制模組 212。

「視頻輸出模組 220」

視頻輸出模組 220 會排他性的佔有第 1 圖的記憶體 113 的一部份，作為 FIFO(First In First Out) 的緩衝器 220A 使用，一時的記憶來自繪圖處理模組 219 的視頻資料，且適宜讀出記憶於該 FIFO 220A 的視頻資料，而輸出至視頻

輸出端子 120(第 1 圖)。

「音頻輸出模組 221」

音頻輸出模組 221 會排他性的佔有第 1 圖的記憶體 113 的一部份，作為 FIFO 的緩衝器 221A 使用，一時的記憶來自音頻解碼器控制模組 217(音頻解碼器 117)的音頻資料，且適宜讀出記憶於該緩衝器 221A 的音頻資料，而輸出至音頻輸出端子 121(第 1 圖)。

又，來自音頻解碼器控制模組 217 的音頻資料為左頻道屬「主聲音」的音頻資料，右頻道屬「副聲音」的音頻資料之雙(Dual)(雙語)模式的音頻資料時，音頻輸出模組 221 會按照預先被指定的聲音輸出模式，將來自音頻解碼器控制模組 217 的音頻資料輸出至音頻輸出端子 121。

亦即，聲音輸出模式，例如被指定為「主聲音」時，音頻輸出模組 221 會拷貝來自音頻解碼器控制模組 217 的音頻資料中的左頻道的音頻資料，而作為右頻道的音頻資料，將該左頻道及右頻道的音頻資料(「主聲音」的音頻資料)輸出至音頻輸出端子 121。又，聲音輸出模式被指定為「副聲音」時，音頻輸出模組 221 會拷貝來自音頻解碼器控制模組 217 的音頻資料中的右頻道的音頻資料，而作為左頻道的音頻資料，將該左頻道及右頻道的音頻資料(「副聲音」的音頻資料)輸出至音頻輸出端子 121。又，聲音輸出模式被指定為「主·副」時，音頻輸出模組 221 會將來自音頻解碼器控制模組 217 的音頻資料原封不動地輸出至

音頻輸出端子 121。

另外，來自音頻解碼器控制模組 217 的音頻資料為立體 (Stereo) 模式的音頻資料時，無論聲音輸出模式的指定如何，音頻輸出模組 221 會將來自音頻解碼器控制模組 217 的音頻資料原封不動地輸出至音頻輸出端子 121。

在此，聲音輸出模式的指定，例如可在視頻內容再生程式 210 所生成的選單所顯示的畫面等中，藉由使用者操作遙控器等來對話性地進行。

[緩衝控制模組 215 的構成]

其次，第 5 圖是表示第 2 圖 A 及第 2 圖 B 的緩衝控制模組 215 的構成例。

緩衝控制模組 215 是以第 1 圖的記憶體 113 的一部份作為緩衝器 215A 來排他性的使用，在該緩衝器 215A 中，使從碟片 101 讀出的資料一時記憶。並且，緩衝控制模組 215 會讀出記憶於緩衝器 215A 的資料，而供應給第 2 圖 A 及第 2 圖 B 的視頻解碼器控制模組 216，音頻解碼器控制模組 217，或字幕解碼器控制模組 218。

亦即，緩衝控制模組 215，除了緩衝器 215A 以外，還具有記憶體 113 的一部份之資料前頭指標 (pointer) 記憶部 231，及資料寫入指標記憶部 232，且內部模組具有視頻讀出功能部 233，音頻讀出功能部 234，字幕讀出功能部 235。

緩衝器 215A 是例如為環形緩衝器 (Ring Buffer)，依次

記憶從碟片 101 讀出的資料，在記憶其記憶容量份的資料之後，以覆蓋最舊資料的形式，將最新的資料記憶成所謂的無限迴路狀。

資料前頭指標記憶部 231 是記憶被記憶於緩衝器 215A 之資料中，指出記憶有還未自緩衝器 215A 讀出之最舊的資料之位置(位址)的資料前頭指標。

資料寫入指標記憶部 232 是記憶用以指出自碟片 101 所讀出之最新資料被寫入之緩衝 215A 位置(位址)的指標。

在此，資料寫入指標所指之位置是每在緩衝器 215A 上述憶自碟片 101 被讀出之資料，如圖中右轉(順時鐘)更新時，資料前頭指標所指之位置，是因應來自緩衝器 215A 之資料讀出，而向圖中順時鐘地更新。因此，被記憶於緩衝器 215A 之資料中，有效之資料可以說是自資料前頭指標所指示之位置起，順時鐘被記憶至資料寫入指標所指示之位置為止的資料。

視頻讀出功能部 233 是按照來自第 2 圖 A 及第 2 圖 B 的視頻解碼器控制模組 216 之要求，從緩衝器 215A 讀出視頻串流(有關視頻資料的基本串流)，供應給視頻解碼器控制模組 216。音頻讀出功能部 234 也是按照來自第 2 圖 A 及第 2 圖 B 的音頻解碼器控制模組 217 之要求，從緩衝器 215A 讀出音頻串流(有關音頻資料的基本串流)，供應給音頻解碼器控制模組 217。字幕讀出功能部 235 也是按照來自第 2 圖 A 及第 2 圖 B 的字幕解碼器控制模組 218 之要求，從緩衝器 215A 讀出字幕串流(有關字幕資料的基本串流)，供

應給字幕解碼器控制模組 218。

亦即，碟片 101 是記錄有以例如 MPEG(Moving Picture Experts Group)2 之規格為根據之程式串流(MPEG2-System Program Stream)，在緩衝器 215A 中記錄有自光碟 101 所讀出之程式串流。該程式串流是多重時間分割視頻串流、音頻串流、字幕串流等之 1 個以上之基本串流。視頻讀出功能部 233 是具有程式串流之多路分解(demultiplexing)功能，自被記憶於緩衝器 215A 之程式串流，分離視頻串流而予以讀出。

同樣地，音頻讀出功能部 234 也具有程式串流之多路分解功能，自被記憶於緩衝器 215A 之程式串流起分離音頻串流而予以讀出。字幕讀出功能部 235 也具有程式串流之多路分解功能，自被記憶於緩衝器 215A 之程式串流分離字幕串流而予以讀出。

在此，視頻讀出功能部 233 具有第 1 圖的記憶體 113 的一部份之視頻讀出指標記憶部 241，stream_id 暫存器 242，及 au_information()暫存器 243。

視頻讀出指標記憶部 241 是記憶緩衝器 215A 之指示記憶有視頻串流之位置(位址)之視頻讀出指標，並且視頻讀出功能部 233 會將緩衝器 215A 之被記憶於視頻讀出指標所指示之位置上的資料，當作視頻串流予以讀出。stream_id 暫存器 242 是解析被記憶於緩衝器 215A 之程式串流，並記憶用以特定自該程式串流中所讀出之視頻串流的後述 stream_id。au_information()暫存器 243 是記憶為了自緩衝

器 215A 讀出視頻串流所需(利用於視頻串流之讀出)之資料的後述 `au_information()`。

音頻讀出功能部 234 具有第 1 圖的記憶體 113 的一部份之音頻讀出指標記憶部 251，`stream_id` 暫存器 252，及 `private_stream_id` 暫存器 253。

音頻讀出指標記憶部 251 是記憶緩衝器 215A 之指出記憶有音頻串流之位置(位址)的音頻讀出指標，音頻讀出功能部 234 會將緩衝器 215A 之被記憶於音頻讀出指標所指示之位置上的資料，當作音頻串流予以讀出。`stream_id` 暫存器 252 和 `private_stream_id` 暫存器 253 是解析被記憶於緩衝器 215A 之程式串流，各記憶用以識別自該程式串流中所讀出之音頻串流的後述 `stream_id` 和 `private_stream_id`。

字幕讀出功能部 235 具有第 1 圖的記憶體 113 的一部份之字幕讀出功能旗標記憶部 261，字幕讀出指標記憶部 262，`stream_id` 暫存器 263，及 `private_stream_id` 暫存器 264。

字幕讀出功能旗標記憶部 261 會記憶字幕讀出功能旗標。當記憶於字幕讀出功能旗標記憶部 261 的字幕讀出功能旗標例如為 0 時，字幕讀出功能部 235 不會使功能動作，當記憶於字幕讀出功能旗標記憶部 261 的字幕讀出功能旗標例如為 1 時，字幕讀出功能部 235 會使功能動作。

字幕讀出指標記憶部 262 是記憶緩衝器 215A 之指示記憶有字幕串流之位置(位址)的字幕讀出指標，字幕讀出功



能功能部 235 會將緩衝器 215A 之被記憶於字幕讀出指標之位置上之資料，當作字幕串流予以讀出。Stream_id 暫存器 263 和 private_stream_id 暫存器 264 是解析被記憶於緩衝器 215A 中之程式串流，各記憶用以識別自該程式串流中讀出之字幕串流的後述 stream_id 和 private_stream_id。

[記錄於碟片 101 之資料的資料格式(format)的說明]

其次，說明有關記錄於碟片 101 的資料的資料格式。

第 6 圖是表示碟片 101 的目錄構造模式。

碟片 101 的檔案系統，例如為使用以

ISO(International Organization for Standardization)-9660

，或 UDF(Universal Disk Format：[http :](http://www.osta.org/specs/)

[//www.osta.org/specs/](http://www.osta.org/specs/))等所規定的檔案系統，記錄於碟片

101 的資料檔案是藉由目錄構造來階層性的管理。在此，

檔案系統並非限於上述檔案系統者。

第 6 圖中，在顯示檔案系統的基點的根(root)目錄中放置 "VIDEO" 目錄，在 "VIDEO" 目錄中放置 "CLIP" 目錄及 "STREAM" 目錄的 2 個目錄。

在 "VIDEO" 目錄中，除了 "CLIP" 目錄及 "STREAM" 目錄的 2 個目錄以外，還放置 "SCRIPT.DAT" 檔案及 "PLAYLIST.DAT" 檔案的 2 個資料檔案。

"SCRIPT.DAT" 檔案為記述有腳本程式的腳本檔案。亦即，在 "SCRIPT.DAT" 檔案中記錄有為了使碟片 101 的再生形態形成交互型(interactive)而使用的腳本程式。記述



於該 "SCRIPT.DAT" 檔案的腳本程式是藉由第 2 圖 A 及第 2 圖 B 的腳本控制模組 211 來解釋，執行。

在 "PLAYLIST.DAT" 檔案中，記述有碟片 101 中所被記錄的視頻資料等的內容的再生程序之播放清單(後述的第 7 圖的 PlayList()) 會被儲存 1 個以上。

在 "CLIP" 目錄中放置 1 個以上的剪輯資訊檔案，在 "STREAM" 目錄中放置 1 個以上的剪輯串流檔案。亦即，第 6 圖中，在 "CLIP" 目錄中放置 3 個剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP"，在 "STREAM" 目錄中放置 3 個剪輯串流檔案 "00001.PS"，"00002.PS"，"00003.PS"。

在剪輯串流檔案中，儲存有將對視頻資料，音頻資料，字幕資料等 1 個以上的資料進行壓縮、編碼而取得的 1 個以上的基本串流予以時間分割多重化後的程式串流。

在剪輯資訊檔案中，記述有所對應的剪輯串流檔案的性質等之關於剪輯串流的元資料。

亦即，剪輯串流檔案與剪輯資訊檔案是 1 對 1 對應。第 6 圖中，在剪輯串流檔案中，按照 5 個文字的數字 + 句點 (period) + "PS" 之命名規則，取檔案名，在剪輯資訊檔案中，按照與所對應的剪輯串流檔案同一 5 個文字的數字 + 句點 + "CLP" 之命名規則，取檔案名。

因此，檔案為剪輯串流檔案或剪輯資訊檔案的其中哪一個，可藉由檔案名的副檔名(句點右側的部份)來識別，又，所對應的剪輯串流檔案與剪輯資訊檔案，可藉由檔案

名的副檔名以外的部份(句點左側的部份)是否一致來識別。

以下，詳細說明有關記錄於碟片101的各檔案。

「PLAYLIST.DAT」

第7圖是表示第6圖的"VIDEO"目錄下的
"PLAYLIST.DAT"檔案的內部構造(語法(syntax))。

在此，於第5圖中，"語法"欄的記載為表示資料構造，
"No.of bits"欄的記載為表示所對應的行的"語法"欄的資料的位元長。又，
"Mnemonic"欄的記載中的"bslbf"(bit string left bit first)是意指所對應的行的"語法"欄的資料會由左邊的位元送出，
且"uimbsf"(unsigned integer most significant bit first)是意指所對應的行的"語法"欄的資料為無編碼整數值，
由最上位位元送出。有關以下說明之與第7圖同樣的圖亦相同。

在"PLAYLIST.DAT"檔案中，從其前頭開始，依次配置有用以記述其名稱(檔案名)等的資訊之 `name_length`(8位元)及 `name_string`(255位元組)。

亦即，`name_length` 是以位元組數來表示之後所被配置的 `name_string` 的大小。`name_string` 是表示
"PLAYLIST.DAT"檔案的名稱(檔案名)。

又，有關 `name_string` 是從其前頭到以 `name_length` 所示的位元組數會當作有效的名稱使用。例如當 `name_length` 為值10時，從 `name_string` 的前頭到10位元

組分會當作有效的名稱解釋。

在 `name_string` 之後，配置有 `number_of_PlayLists`(16位元)。 `number_of_PlayLists` 是表示接續之 `PlayList()` 的個數。在 `number_of_PlayLists` 之後，配置有該 `number_of_PlayLists` 的數量之 `PlayList()`。

`PlayList()` 是記述有碟片 101 中所被記錄的剪輯串流檔案的再生程序之播放清單，具有以下那樣的內部構造。

亦即，在 `PlayList()` 的前頭，配置有 `PlayList_data_length`(32位元)。 `PlayList_data_length` 是表示該 `PlayList()` 的大小。

在 `PlayList_data_length` 之後，依次配置有 `reserved_for_word_alignment`(15位元) 及 `capture_enable_flag_PlayList`(1位元)。15位元的 `reserved_for_word_alignment` 是爲了在其後配置的1位元的 `capture_enable_flag_PlayList` 的位置一致於16位元的位置而配置。 `capture_enable_flag_PlayList` 是表示對應於依照 `PlayList()` 所再生的視頻串流之視頻資料(屬於 `PlayList()` 的視頻資料)在光碟 101 所被再生的第1圖的碟片再生裝置內是否許可2次利用之1位元的旗標。當 `capture_enable_flag_PlayList` 爲0或1的其中之例如1時，表示屬於 `PlayList()` 的視頻資料的2次利用被許可，當 `capture_enable_flag_PlayList` 爲0或1的其中之例如0時，表示屬於 `PlayList()` 的視頻資料的2次利用未被許可。

另外，在第7圖中，雖 `capture_enable_flag_PlayList`



爲1位元，但除此以外，`capture_enable_flag_PlayList`可以複數位元構成，使屬於`PlayList()`的視頻資料的2次利用能夠階段的許可。亦即，`capture_enable_flag_PlayList`，例如可以2位元構成。而且，當`capture_enable_flag_PlayList`的值爲00B(B是表示其前的數字爲2進位)時，禁止視頻資料的2次利用，當`capture_enable_flag_PlayList`的值爲01B時，可以許可視頻資料的2次利用，但只限縮小至64×64畫素(pixel)以下的大小來利用。又，當`capture_enable_flag_PlayList`的值爲10B時，則可在大小無限制下，許可視頻資料的2次利用。

又，如上述，視頻資料的2次利用，亦可不對大小設限，而對用途設限。亦即，當`capture_enable_flag_PlayList`的值爲01B時，只許可在視頻內容再生程式210(第2圖A及第2圖B)的2次利用，當`capture_enable_flag_PlayList`的值爲10B時，可以許可第1圖的碟片再生裝置內之包含視頻內容再生程式210的任意應用之2次利用。在此，第1圖的碟片再生裝置內的視頻內容再生程式210以外的應用，例如有進行壁紙(背景)或螢幕保護的顯示處理之應用等。

另外，當`capture_enable_flag_PlayList`爲上述2位元時，配置於其前的`reserved_for_word_alignment`爲了取辭彙對齊，而形成14位元。

此外，除了依據`capture_enable_flag_PlayList`來許可

視頻資料在碟片再生裝置內的2次利用以外，亦可許可在碟片再生裝置外的2次利用。在此，當許可視頻資料在碟片再生裝置外的2次利用時，視頻資料是例如被記錄於可裝卸於碟片再生裝置的記錄媒體或可連接於碟片再生裝置的其他裝置的記錄媒體，或者經由網際網路等的網路來傳送至其他的裝置。此情況，可在視頻資料中附加限制將該視頻資料記錄於記錄媒體的次數或配送的次數之資訊。

接續於 `capture_enable_flag_PlayList`，依次配置有 `PlayList_name_length`(8位元)及 `PlayList_name_string`(255位元組)。`PlayList_name_length` 是以位元組數來表示其後配置的 `PlayList_name_string` 的大小，`PlayList_name_string` 是表示 `PlayList()` 的名稱。

在 `PlayList_name_string` 之後，配置有 `number_of_PlayItems`(16位元)。`number_of_PlayItems` 是表示接續之 `PlayItem()` 的個數。

在 `number_of_PlayItems` 之後，記述有僅該 `number_of_PlayItems` 的數量之 `PlayItem()` 的構造。

在此，在1個 `PlayList()` 中，可以 `PlayItem()` 單位來記述內容的再生程序。

再者，對於 `PlayList()` 中之僅有 `number_of_PlayItems` 之數量的 `PlayItem()` 之各個，在該 `PlayList()` 中，賦予獨特之 ID(Identification)。即是，對 `PlayList()` 上賦予0號以作為 ID，以下對接續的 `PlayItem()`，依照出現之順序，賦予1號、2號、...之連續號碼之 ID。

在僅 `number_of_PlayItems` 的數量之 `PlayItem()` 之後，配置有 1 個 `PlayListMark()`。`PlayListMark()` 是按照 `PlayList()` 來進行之再生的時間軸上的記號之後述的 `Mark()` 的集合，其詳細是參照第 9 圖來敘述。

「`PlayItem()` 的說明」

其次，第 8 圖是表示第 7 圖的 `PlayList()` 中所含的 `PlayItem()` 的內部構造。

在 `PlayItem()` 的前頭配置有 `length`(16 位元)，`length` 是表示含彼之 `PlayItem()` 的大小。

接續於 `length`，依次配置有 `Clip_Information_file_name_length`(16 位元) 及 `Clip_Information_file_name`(可變長)。
`Clip_Information_file_name_length` 是以位元組數來表示其後配置的 `Clip_Information_file_name` 的大小。

`Clip_Information_file_name` 是表示對應於依據 `PlayItem()` 而再生的剪輯串流檔案(第 6 圖的副檔名為 PS 的檔案)之剪輯資訊檔案(第 6 圖的副檔名為 CLP 的檔案)的檔案名。另外，可根據剪輯串流檔案及剪輯資訊檔案的檔案名之上述命名規則，由 `Clip_Information_file_name` 來認定依據 `PlayItem()` 而再生之剪輯資訊檔案的檔案名，特定該剪輯串流檔案。

接續於 `Clip_Information_file_name`，依次配置有 `IN_time`(32 位元) 及 `OUT_time`(32 位元)。

IN_time 及 OUT_time 是分別指定由 Clip_Information_file_name 所特定之剪輯串流檔案的再生開始位置及再生終了位置之時刻資訊。

若利用 IN_time，則可指定剪輯串流檔案的(包含前頭)途中之位置作為再生開始位置，若利用 OUT_time，則可指定剪輯串流檔案的(包含最後)途中之位置作為再生終了位置。

在此，若依據 PlayItem()，則由 Clip_Information_file_name 所特定之剪輯串流檔案從 IN_time 到 OUT_time 之間的内容會被再生。以下，將依據該 PlayItem()而再生的内容稱為剪輯。

「 PlayListMark()的說明 」

其次，第9圖是表示第7圖的 PlayList()中所含的 PlayListMark()的内部構造。

如上述，PlayListMark()是按照包含該 PlayListMark()的 PlayList()(第7圖)來進行之再生的時間軸上的記號之0以上的 Mark()的集合。1個 Mark()至少具有顯示按照 PlayList()來進行之再生的時間軸上的1個時刻之時刻資訊，及顯示 Mark()的類型之類型資訊，以及顯示類型資訊為使發生事件的類型時之形成該事件的引數之引數資訊。

亦即，在 PlayListMark()的前頭配置有 length(32位元)。Length 是表示含彼之 PlayListMark()的大小。

在 length 之後配置有 number_of_PlayList_marks(16

位元)，`number_of_PlayList_marks` 是表示接續配置之 `Mark()` 的個數。在 `number_of_PlayList_marks` 之後，記述有僅該 `number_of_PlayList_marks` 的數量之 `Mark()` 的構造。

在 `Mark()` 的前頭配置有 `mark_type`(8 位元)。
`mark_type` 為上述類型資訊，顯示含彼之 `Mark()` 的類型。

在本實施形態中，`Mark()` 的類型，例如有章節 (Chapter)，索引 (Index)，事件 (Event) 的 3 種類。

類型為章節的 `Mark()`(以下稱為章節標示)是分割 `PlayList()` 的出頭的單位之章節的前頭位置的記號。又，類型為索引的 `Mark()`(以下稱為索引標示)是細分章節的單位之索引的前頭位置的記號。類型為事件的 `Mark()`(以下稱為事件標示)是在按照 `PlayList()` 的內容再生中使事件發生的位置的記號。根據事件標示之事件的發生，如後述，會通知腳本控制模組 211。

在此，將 `mark_type` 的值與 `Mark()` 的類型之關係顯示於第 10 圖。根據第 10 圖，在章節標示的 `mark_type` 會被設定為 1。此外，在索引標示的 `mark_type` 會被設定為 2，在事件標示的 `mark_type` 會被設定為 3。另外，在第 10 圖中，`mark_type` 所示的 8 位元的值中，0 及 4～255 為將來擴張用的預約 (reserved)。

回到第 9 圖，在 `mark_type` 之後配置有 `mark_name_length`(8 位元)。並且，在 `Mark()` 的最後配置有 `mark_name_string`(24 位元組)。`mark_name_length` 及

`mark_name_string` 是用以記述 `Mark()` 的名稱者，
`mark_name_length` 是表示 `mark_name_string` 的有效大小，
`mark_name_string` 是表示 `Mark()` 的名稱。因此，從
`mark_name_string` 的前頭到 `mark_name_length` 所示的位
元組數為止是表示 `Mark()` 的有效名稱。

接續於 `mark_name_length`，依次配置有使被定義於
`PlayList()` 上的 `Mark()` 與剪輯串流檔案有所對應的 4 個要素
`ref_to_PlayItem_id` (16 位元)，`mark_time_stamp` (32 位元)，
`entry_ES_stream_id` (8 位元)，
`entry_ES_private_stream_id` (8 位元)。

`ref_to_PlayItem_id` 是被記述著對 `Mark()` 所屬之
`PlayItem()` 賦予連續號碼之 ID。依據 `ref_to_PlayItem_id`
，特定 `Mark()` 所屬之 `PlayItem()` (第 8 圖)，進一步如第 8 圖
說明般，特定剪輯資訊檔案和剪輯流檔案。

`mark_time_stamp` 是表示依據 `ref_to_PlayItem_id` 所
特定的剪輯串流檔案內之 `Mark()` 所示的時刻。

在此，第 11 圖是表示 `PlayList()`，`PlayItem()`，剪輯，
及儲存於剪輯串流檔案的程式串流的關係。

在第 11 圖中，`PlayList()` 是由 3 個 `PlayItem()` 所構成，
在該 3 個 `PlayItem()` 各個上附有以連續號碼表示之 ID#0、
#1、#2。在此，以下適當將附有 ID#i 之 `PlayItem()` 記述
為 `PlayItem#i`。

又，第 11 圖中，依據 `PlayItem#0`，`PlayItem#1`，
`PlayItem#2` 而再生的內容之剪輯會分別顯示為剪輯 A，剪

輯 B， 剪輯 C。

剪輯之實體是被儲存於由第 8 圖之 `PlayItem()` 中之 `Clip_information_file_name` 所特定之剪輯串流檔案中之程式串流之中的從 `IN_time` 至 `OUT_time` 的程式串流。第 11 圖是表示剪輯 A、剪輯 B、剪輯 C 之實體的程式串流各當作程式串流 A、程式串流 B、程式串流 C。

例如，在第 11 圖中，按照 `PlayList()` 來進行之再生的時間軸上的時刻 `t0` 的記號之 `Mark()` 中，該 `ref_to_PlayItem_id` 及 `mark_time_stamp` 是如其次那樣記述。

亦即，時刻 `t0` 為進行 `PlayItem#1` 的再生之時刻，因此在 `ref_to_PlayItem_id` 中記述有該 `PlayItem#1` 的 ID 之 1。又，因為在時刻 `t0` 會依據 `PlayItem#1` 來再生剪輯 B 的實體之程式串流 B，所以在 `mark_time_stamp` 中記述有相當於儲存有程式串流 B 的剪輯串流檔案的時刻 `t0` 之時刻。

再次，回到第 9 圖，`entry_ES_stream_id`，和 `entry_ES_private_stream_id` 是於將 `Mark()` 賦予與特定基本串流有所關聯時，為了特定該基本串流而被使用。即是，在 `entry_ES_stream_id` 記述著將 `Mark()` 賦予關聯之基本串流之後述 `stream_id`。再者，`entry_ES_private_stream_id` 是因應所需，記述著將 `Mark()` 賦予關聯之基本串流之後述的 `private_stream_id`。

例如，視頻串流 #1 和視頻串流 #2 被多重化之剪輯中，當在再生視頻串流 #1 之時，和再生視頻串流 #2 之時，欲變

更章節之發生時刻時，則在視頻串流#1再生時之章節標示發生時刻之 `Mark()`之 `entry_ES_stream_id` 和 `entry_ES_private_stream_id`，記述視頻串流#2之 `stream_id` 和 `private_stream_id`。

並且，在不賦予與特定基本串流關聯之 `Mark()`之 `entry_ES_stream_id` 和 `entry_ES_stream_id` 和 `entry_ES_private_stream_id` 中之任一者也記述著 0。

於 `entry_ES_private_stream_id` 之後，配置 `mark_data`(32位元)。`mark_data` 是 `Mark()`為事件標示之時，將成為藉由該事件標示所發生之事件之引數的引數資訊。並且，`mark_data` 是 `Mark()`為章節標示或索引標示時，亦可以當作該章節標示或索引標示所表示之章節或索引之號碼使用。

「Clip()的說明」

其次，說明有關被放置於第6圖的"CLIP"目錄之副檔名為 CLP 的剪輯資訊檔案的內部構造。

第6圖中，在"CLIP"目錄中放置有3個剪輯資訊檔案"00001.CLP"，"00002.CLP"，"00003.CLP"，且各個儲存有顯示被放置於"STREAM"目錄的剪輯串流檔案"00001.PS"，"00002.PS"，"00003.PS"的性質等之元資料。

第12圖是表示如此的剪輯資訊檔案 Clip()的內部構造。

在剪輯資訊檔案 Clip()的前頭依次配置有 presentation_start_time 及 presentation_end_time(皆為 32 位元)。presentation_start_time 及 presentation_end_time 是表示對應於剪輯資訊檔案 Clip()之剪輯串流檔案的前頭及最後的時刻。另外，剪輯串流檔案的時刻是以使用於 MPEG2-System 的時刻之 90kHz 的倍數來記述。

接續於 presentation_end_time，依次配置有 reserved_for_word_alignment(7位元)及 capture_enable_flag_Clip(1位元)。7位元的 reserved_for_word_alignment 是用以取辭彙對齊者，capture_enable_flag_Clip 是與上述第 7 圖的 capture_enable_flag_PlayList 同樣，表示是否許可視頻資料的 2 次利用之旗標。

但是，第 7 圖之 capture_enable_flag_PlayList 是表示是否許可對應於依據 PlayList()而再生的視頻串流之屬於 PlayList()的視頻資料之 2 次利用，相對的，第 12 圖之 capture_enable_flag_Clip 是表示是否許可對應於被儲存於對應於剪輯資訊檔案 Clip()的剪輯串流檔案之視頻的基本串流的視頻資料的 2 次利用。因此，第 7 圖之 capture_enable_flag_PlayList 和第 12 圖之 capture_enable_flag_Clip 是許可 2 次利用之視頻單位(範圍)為不同。

另外，第 12 圖的 capture_enable_flag_Clip 亦如第 7 圖的 capture_enable_flag_PlayList 所述，並非限於 1 位元，

可為複數位元。

在 `capture_enable_flag_Clip` 之後配置有 `number_of_streams`(8位元)，在此 `number_of_streams` 中記述有接續之 `StreamInfo()` 構造的個數。因此，接續於 `number_of_streams` 記述有僅該 `number_of_streams` 的數量之 `StreamInfo()` 的構造。

在 `StreamInfo()` 的前頭配置有 `length`(16位元)，此 `length` 是表示含彼之 `StreamInfo()` 的大小。接續於 `length`，配置有 `stream_id`(8位元)及 `private_stream_id`(8位元)，藉此 `stream_id` 及 `private_stream_id` 來特定(識別)與 `StreamInfo()` 有所關聯的基本串流。

在此，第13圖是表示識別基本串流的 `stream_id` 及 `private_stream_id` 與基本串流的關係。

`stream_id` 是與 MPEG2-System 規格中所規定者同一者，其值是在 MPEG2-System 規格中，依基本串流的屬性來事先決定。因此，以 MPEG2-System 規格所定義之屬性的基本串流可僅以 `stream_id` 來特定。

在本實施形態中，未被規定於 MPEG2-System 規格中之屬性的基本串流亦可處理，`private_stream_id` 是用以識別未被規定於 MPEG2-System 規格中之屬性的基本串流的資訊。

在第13圖中顯示有關以 MPEG 所規定的編碼方式來編碼之視頻的基本串流，以 ATRAC(Adaptive TRansform Acoustic Coding)方式來編碼之音頻的基本串流(以下稱為

ATRAC 音頻串流)，以 LPCM(Linear Pulse Code Modulation)方式來編碼之音頻的基本串流(以下稱為 LPCM 音頻串流)，字幕的基本串流(以下稱為字幕串流)的 4 個屬性的基本串流與 `stream_id` 及 `private_stream_id` 的關係。

在 MPEG2-System 規格中，以 MPEG 所規定之編碼方式編碼之視頻得基本串流，是被規定著將 0xE0 至 0xEF 之範圍的值(0x 是表示之後接續的文字列為 16 進位)，當作用以識別基本串流之 `stream_id` 使用而予以多重化。因此，針對以在 MPEG 所規定之編碼化方式被編碼之視頻之基本串流，可將能以 0xE0 至 0xEF 之範圍的值之 `stream_id` 來識別的 16 個視頻基本串流，予以多重化成程式串流。

另外，以 MPEG 所規定的編碼方式來編碼之視頻的基本串流的識別，可使用 0xE0 ~ 0xEF 的範圍的值的 `stream_id` 來進行，因此不需要 `private_stream_id`。

另一方面，在 MPEG2-System 中，有關 ATRAC 音頻串流，LPCM 音頻串流，字幕串流方面，`stream_id` 未被定義。

於是，在本實施形態中，於 MPEG2-System 中有關 `stream_id` 未被定義的基本串流方面，其 `stream_id` 採用 MPEG2-System 中顯示 `private_stream_1` 的屬性的值之 0xBD，且如第 11 圖所示，使用 `private_stream_id` 來進行識別(特定)。

亦即，在 ATRAC 音頻串流的識別中，0x00 ~ 0x0F 的

範圍的值的 `private_stream_id` 會被使用。因此，在程式串流中，可使 16 個 ATRAC 音頻串流多重化。並且，在 LPCM 音頻串流的識別中，`0x10 ~ 0x1F` 的範圍的值 `private_stream_id` 會被使用。因此，在程式串流中，可使 16 個 LPCM 音頻串流多重化。而且，在字幕串流的識別中，`0x80 ~ 0x9F` 的範圍的值的 `private_stream_id` 會被使用。因此，在程式串流中，可使 32 個字幕串流多重化。

另外，有關 `stream_id` 及 `private_stream_id` 會在往後敘述。

回到第 12 圖，在 `private_stream_id` 之後依次配置有 `StaticInfo()`，`reserved_for_word_alignment`(8 位元)。在 `StaticInfo()` 中記述有依據記述於包含該 `StaticInfo()` 的 `StreamInfo()` 之 `stream_id` 及 `private_stream_id` 所特定之基本串流的再生中不會變化的資訊。有關 `StaticInfo()` 的詳細會在往後參照第 14 圖來敘述。

`reserved_for_word_alignment` 是爲了取辭彙對齊而使用。

接續於 `reserved_for_word_alignment` 而配置有 `number_of_DynamicInfo`(8 位元)，`number_of_DynamicInfo` 是表示在之後接著配置的 `pts_change_point`(32 位元)與 `DynamicInfo()` 的組合數。

因此，接續於 `number_of_DynamicInfo` 記述有僅該 `number_of_DynamicInfo` 的數量之組合數的 `pts_change_point` 與 `DynamicInfo()` 的構造。

`pts_change_point` 是表示與彼成組的 `DynamicInfo()` 的資訊為形成有效的時刻。在此，表示基本串流的前頭的時刻之 `pts_change_point` 是與對應於儲存有該基本串流的剪輯串流檔案之剪輯資訊檔案 `Clip()` 的最初所被記述的 `presentation_start_time` 相等。

在 `DynamicInfo()` 中，記述有在依據 `stream_id` 及 `private_stream_id` 而特定之基本串流的再生中變化之所謂動態的資訊。記述於 `DynamicInfo()` 的資訊是在形成與彼成組的 `pts_change_point` 所示的再生時刻時形成有效。另外，有關 `DynamicInfo()` 的詳細會在往後參照第 15 圖來敘述。

在僅 `number_of_DynamicInfo` 的數量之組合的 `pts_change_point` 與 `DynamicInfo()` 之後配置有 `EP_map()`。另外，有關 `EP_map()` 會在往後參照第 16 圖來敘述。

「StaticInfo()的說明」

其次，參照第 14 圖來詳細說明有關第 12 圖的 `StaticInfo()`。

第 14 圖是表示 `StaticInfo()` 的語法。

`StaticInfo()` 是依照所對應之基本串流的屬性(種類)而內容會有所不同。對應於 `StaticInfo()` 之基本串流的屬性是依據包含該 `StaticInfo()` 的第 12 圖的 `StreamInfo()` 中所含的 `stream_id` 及 `private_stream_id` 來判斷。

當對應於 `StaticInfo()` 的基本串流為視頻串流時

(stream==VIDEO)，StaticInfo()是以 picture_size(4位元)，frame_rate(4位元)，cc_flag(1位元)，及用以取辭彙對齊的 reserved_for_word_alingment 所構成。

picture_size 是表示依照對應於視頻串流的視頻資料而顯示的畫像的大小。frame_rate 是表示對應於視頻串流的視頻資料的幀頻率。cc_flag 是表示在視頻串流中是否含有隱藏式字幕 (Closed Caption)資料。亦即，例如在視頻串流中含有隱藏式字幕資料時，cc_flag 為形成1，在視頻串流中未含隱藏式字幕資料時，cc_flag 為形成0。

當對應於 StaticInfo()的基本串流為音頻串流時 (stream==AUDIO)，StaticInfo()是以 audio_language_code(16位元)，channel_configuration(8位元)，lfe_existence(1位元)，sampling_frequency(4位元)，及用以取辭彙對齊的 reserved_for_word_alingment 所構成。

在 audio_language_code 中記述有表示音頻串流中所含的音頻資料的語言的碼。channel_configuration 是表示單音(mono)/立體(stereo)/多(multi)頻道等之音頻串流中所含的音頻資料的屬性。lfe_existence 是表示在音頻串流中是否含有低域強調頻道，若含有則會形成1，若未含有則會形成0。sampling_frequency 是表示在音頻串流中所含之音頻資料的取樣頻率的資訊。

當對應於 StaticInfo()的基本串流為字幕串流時 (stream==SUBTITLE)，StaticInfo()是以

`subtitle_language_code`(16位元)，`configurable_flag`(1位元)，及用以取辭彙對齊的 `reserved_for_word_alignment` 所構成。

在 `subtitle_language_code` 中記述有表示字幕串流中所含的字幕資料的語言的碼。`configurable_flag` 是表示是否許可由內定(default)的顯示方式來變更字幕串流中所含的字幕資料的顯示之資訊，例如當顯示方式的變更被許可時，記述有1，當未被許可時，記述有0。另外，字幕資料的顯示方式，有字幕資料的顯示大小，顯示位置，顯示色，顯示式樣(例如點滅顯示等)，顯示方向(例如垂直方向或水平方向)等。

「DynamicInfo()的說明」

其次，參照第15圖來詳細說明有關第12圖的 `DynamicInfo()`。

第15圖是表示 `DynamicInfo()`的語法。

在 `DynamicInfo()`的前頭配置有辭彙對齊用的 `reserved_for_word_alignment`(8位元)，其後接續的要素會依對應於 `DynamicInfo()`的基本串流的屬性而內容有所不同。對應於 `DynamicInfo()`的基本串流的屬性是與第14圖所說明的 `StaticInfo()`時同樣，依據包含 `DynamicInfo()`的第12圖的 `StreamInfo()`中所含的 `stream_id` 及 `private_stream_id` 來判斷。

在 `DynamicInfo()`中，如第12圖所說明那樣，記述有

在基本串流的再生中變化的動態資訊。此動態資訊並無特別加以限定，在第15圖的實施形態中，對應於基本串流（對應於 `DynamicInfo()`）的資料，亦即，藉由基本串流的處理而輸出的資料的輸出屬性（由基本串流所取得的資料的輸出屬性）會被記述於 `DynamicInfo()`。

具體而言，當對應於 `DynamicInfo()` 的基本串流為視頻串流時（`stream==VIDEO`），`DynamicInfo()` 是以 `display_aspect_ratio`（4位元）及辭彙對齊用的 `reserved_for_word_alingment` 來構成。在 `diplay_aspect_ratio` 中，記述有作為對應於視頻串流的視頻資料的輸出方式之例如該視頻資料的寬高比。亦即，在 `diplay_aspect_ratio` 中，例如記述有顯示寬高比為 16：9 或 4：3 的資訊。並且，在視頻串流的 `DynamicInfo()` 中，除了寬高比以外，例如可記述根據視頻資料而顯示之畫像的大小（X 畫素×Y 畫素）等。

當對應於 `DynamicInfo()` 的基本串流為音頻串流時（`stream==AUDIO`），`DynamicInfo()` 是以 `channel_assignment`（4位元）及用以取辭彙對齊的 `reserved_for_word_alingment` 來構成。當音頻串流中含有 2 頻道的音頻資料時，在 `channel_assignment` 中記述有該 2 頻道的輸出方式。亦即，在 `channel_assignment` 中記述有顯示音頻資料被分配為立體或雙語的其中任一的資訊。

當對應於 `DynamicInfo()` 的基本串流為字幕串流時（`stream==SUBTITLE`），`DynamicInfo()` 是以用以取辭彙對

齊的 `reserved_for_word_alingment` 所構成。亦即，在第 13 圖的實施形態中，有關字幕串流方面，動態的資訊之輸出屬性未被定義。

「`EP_map()`的說明」

其次，參照第 16 圖來詳細說明有關第 12 圖的 `EP_map()`。

第 16 圖是表示 `EP_map()` 的語法。

`EP_map()` 是被記述著針對被多重化成儲存於對應包含該 `EP_map()` 之第 12 圖之剪輯資訊檔案 `Clip()` 之剪輯串流檔案中之程式串流的每基本串流，各基本串流之可以開始解碼之解碼開始可能點(登錄點)之資訊。

在此，針對固定比率之串流，解碼開始可能點是可以依據計算而求出，但是如可變率之串流或是依據 MPEG 規格而被編碼號之視頻串流般，針對每視頻存取單元大小不同之串流，解碼開始可能點是無法藉由計算求出，若不實際解析串流就無法找到。要迅速認定解碼開始可能點用以執行隨機存取極為重要，若依據 `EP_map()`，則可迅速認定解碼開始可能點。

並且，MPEG2-Video 是包含有 `Sequence_header()`(順序標頭)等之幀內畫面(intrapicture)之前頭為解碼開始可能點。

`EP_map()` 之前頭是被配置有辭彙對齊用之 `reserved_for_word_alignment`(8 位元)，接著配置有

`number_of_stream_id_entries`(8位元)。

`number_of_stream_id_entries` 是表示在 `EP_map()` 有解碼開始可能點之資訊的基本串流之條數。

`number_of_stream_id_entries` 之後，用以識別基本串流之資訊和該基本串流之解碼開始可能點之資訊是以僅表示 `number_of_stream_id_entries` 之數量重複配置。

亦即，於 `number_of_stream_id_entries` 之後，配置當作用以識別基本串流之資訊的 `stream_id`(8位元)和 `private_stream_id`(8位元)，接續此配置 `number_of_EP_entries`(32位元)。`number_of_EP_entries` 是表示以之前的 `stream_id` 和 `private_stream_id` 所特定之基本串流之解碼開始可能點之數量。

於 `number_of_EP_entries` 之後，以之前的 `stream_id` 和 `private_stream_id` 所特定之基本串流之解碼開始可能點之資訊是以僅表示 `number_of_EP_entries` 之數量重複被配置。

亦即，該當基本串流為視頻串流時，首先 `index_N_minus1` 及 `N-th_Ref_picture_copy` 會被放置，其後作為解碼開始可能點的資訊之 `PTS_EP_start`(32位元)及 `RPN_EP_start`(32位元)會被放置，該等會被重複配置。有關 `index_N_minus1` 及 `N-th_Ref_picture_copy` 會在往後敘述。

又，該當基本串流為視頻串流以外時，在16位元的 `reserved_for_future_use` 之後作為解碼開始可能點的資訊

之 PTS_EP_start(32位元)及 RPN_EP_start(32位元)會被放置，該等會被重複配置。

在 N-th_Ref_picture_copy 中拷貝有記錄於以後述的 RPN_EP_start 所示的 private_stream_2之 1stRef_picture、2ndRef_picture、3rdRef_picture、4thRef_picture 的其中任一個。並且，在 index_N_minus1中到底拷貝了哪個場(field)之資訊會利用第17圖所示的值來記錄。亦即，如第17圖所示，在 index_N_minus1中，當 1stRef_picture 被拷貝時，記錄 0，當 2ndRef_picture 被拷貝時，記錄 1，當 3rdRef_picture 被拷貝時，記錄 2，當 4thRef_picture 被拷貝時，記錄 4。

解碼開始可能點之資訊之1的 PTS_EP_start 是表示儲存有如上述般以 stream_id 和 private_stream_id 所特定之基本串流被多重化之程式串流的剪輯串流檔案內之解碼開始點之時刻(再生時刻)。

在解碼開始可能點之資訊之其他之1的 RPN_EP_start 上，記述有以程式串流之 pack()單位，計數儲存有如上述般以 stream_id 和 private_stream_id 所特定之基本串流被多重化之程式串流之剪輯串流檔案內的解碼開始可能點之位置時之值。並且，於本實施形態中，pack()之大小是以2048位元組固定。再者，本實施形態中，光碟101(第1圖)之1區段為2048位元組。

在此，針對視頻串流，是於該解碼開始可能點(登錄點)之前，配置有後述之 private_stream_2封包

(private_stream_2之屬性之 PES_packet()。

private_stream_2封包是由該 private_stream_2封包所構成，儲存有爲了解碼被配置於至下一個 private_stream_2封包之間的視頻串流而所利用之資訊。因此，針對視頻串流，在當作解碼開始可能點之資訊的 RPN_EP_start 上並非記述著實際之解碼開始可能點，而是記述著被配置在實際解碼開始可能點之前的 private_stream_2封包之前頭位置。

再者，在 EP_map()中，當作解碼開始可能點之資訊的 PTS_EP_start 和 RPN_EP_start 之組是於每以 stream_id 和 private_stream_id 所特定之基本串流，是事先依序被分類。依此，當作解碼開始可能點之資訊的 PTS_EP_start 和 RPN_EP_start 之組是成爲可二元搜尋。

並且，將每可變率之串流或視頻存取單元大小爲不同之串流當作對象之隨機存取方法是記載於日本特開 2000-341640 號公報 (日本特願平 11-317738 號) 等中。

「剪輯串流檔案的說明」

其次，說明有關被放置於第 6 圖的 "STREAM" 目錄之副檔名爲 PS 的剪輯串流檔案 (在第 6 圖中爲 "00001.PS"，"00002.PS"，"00003.PS") 的内部構造。

剪輯串流檔案是以定義於 MPEG-2 System (ISO/IEC 13818-1) 的 MPEG2_Program_Stream() 爲基礎構成。

亦即，第 18 圖 A 及第 18 圖 B 是表示記述於 MPEG-2 System (ISO/IEC 13818-1 : 2000) 規格的表 2-31，表 2-32，

表 2-33。

儲存於剪輯串流檔案的程式串流是 MPEG-2 System 規格的表 2-31中所被定義的 MPEG2_Program_Stream()，由1個以上的 pack()及1個 MPEG_program_end_code 所構成。另外，MPEG2_Program_Stream()的說明在專利第 2785220號等中亦有記載。

1個 pack()是如定義於 MPEG-2 System 規格的表 2-32 所示，由1個 Pack_header()及任意數的 PES_packet()所構成。Pack_header()的詳細是被定義於 MPEG-2 System 規格的表 2-33。

在此，於 MPEG-2 System 規格中，pack()的大小是定義為可變長，但在此，如第16圖所說明，被固定為2048位元組。又，在此，1個 pack()的 PES_packet()的數量為1個，2個，或3個。當 Pack()開始於後述的 private_stream_2 封包時，必存在其後所對應的視頻串流的 PES_packet()。並且還可以追加放置 padding_packet(填充封包)來作為第3個 PES_packet()。另外，private_stream_2封包是一定放置於 Pack()的前頭。

當 Pack()不開始於 private_stream_2 封包時，在 Pack()的前頭中放置有儲存視頻，音頻，字幕等的內容資料之 PES_packet()。並且可追加放置 padding_packet(填充封包)來作為第2個 PES_packet()。

第19圖 A 及第19圖 B～第21圖 A 及第21圖 B 是表示在 MPEG-2 System 規格的表 2-17中所被定義的

PES_packet()。

PES_packet()大致可分為：packet_start_code_prefix，stream_id，及 PES_packet_length(第19圖 A 及第19圖 B)，及依據 stream_id 等而構造變化的標頭部份(包含 stuffing_byte)(第19圖 A 及第19圖 B～第21圖 A 及第21圖 B)，及 PES_packet_data_byte(第21圖 A 及第21圖 B)。另外，當 PES_packet()為 padding_packet 時 (stream_id==padding_stream)，取代 PES_packet_data_byte，重複配置必要數量的 padding_byte(0xFF)(第21圖 A 及第21圖 B)。

在此，於 PES_packet()的標頭部份，如第19圖 A 及第19圖 B，第20圖 A，第20圖 B 及第20圖 C 所示，可配置顯示被稱為 PTS(Presentation Time Stamp)的顯示時序之資訊，及顯示被稱為 DTS(Decoding Time Stamp)的解碼時序之資訊。本實施形態中是針對所有的存取單元(定義於 MPEG2-System 之構成基本串流的解碼單位)附加 PTS，且在決定於 MPEG2-System 時附加 DTS。

在程式串流中被多重化的基本串流會被儲存於 PES_packet()的 PES_packet_data_byte(第21圖 A 及第21圖 B)。而且，在 PES_packet()的 stream_id 中，為了識別被儲存於該 PES_packet_data_byte 的基本串流，而記述有對應於該基本串流的屬性之值。

被記述於 PES_packet()的 stream_id 之值與基本串流的屬性(種類)之關係是被定義於 MPEG-2 System 規格的表

2-18。在此，於第 22 圖 A 及第 22 圖 B 中顯示 MPEG-2 System 規格的表 2-18。

本實施形態是採用以第 22 圖 A 及第 22 圖 B 所示的 MPEG-2 System 規格來定義之 stream_id 中例如第 23 圖所示的值。

亦即，本實施形態是採用 10111101B，10111110B，10111111B，110xxxxxB，1110xxxxB 的 5 式樣來作為 stream_id 的值。另外，"x"是表示 0 或 1 的其中任意的值。

而且，被稱為 private_stream_1 的屬性的基本串流的 PES_packet() 的 stream_id 按照第 23 圖成為 10111101B。又，padding_packet 的 PES_packet() 的 stream_id 按照第 20 圖成為 10111110B。又，被稱為 private_stream_2 的屬性的基本串流的 PES_packet() 的 stream_id 按照第 23 圖成為 10111111B。

又，以 MPEG 所定義的音頻串流(音頻的基本串流)的 PES_packet() 的 stream_id 為 110xxxxxB。另外，110xxxxxB 的其中下位 5 位元 xxxxx 為區別音頻串流的音頻串流號碼(number)，在程式串流(MPEG2_Program_Stream())中，可使得能為該音頻串流號碼所區別的數量，亦即 $32(=2^5)$ 個音頻串流(以 MPEG 所定義的音頻串流)多重化。

又，以 MPEG 所定義的視頻串流(視頻的基本串流)的 PES_packet() 的 stream_id 為 1110xxxxB。另外，1110xxxxB 的其中下位 4 位元 xxxx 為區別視頻串流的視頻

串流號碼，在程式串流中，可使得能為該視頻串流號碼所區別的數量，亦即 $16(=2^4)$ 個視頻串流(以 MPEG 所定義的視頻串流)多重化。

`stream_id` 為 `1110xxxxB` 的 `PES_packet()` 是用以儲存 MPEG 定義的視頻串流，`stream_id` 為 `110xxxxxB` 的 `PES_packet()` 是用以儲存 MPEG 定義的音頻串流。另一方面，用以儲存非 MPEG 定義的編碼方式(例如 ATRAC 方式)的基本串流之 `PES_packet()` 的 `stream_id`，因為不為 MPEG 所規定，所以非 MPEG 定義的編碼方式的基本串流無法與 MPEG 定義的視頻串流或音頻串流同樣，單純地指定 `stream_id` 來儲存於 `PES_packet()`。

於是，本實施形態會擴張 `private_stream_1` 的 `PES_packet()` 的 `PES_packet_data_byte`，儲存非 MPEG 定義的編碼方式的基本串流。

在此，將 `private_stream_1` 的 `PES_packet()` 之擴張後的 `PES_packet_data_byte` 記述為 `private_stream1_PES_payload()`。

「`private_stream1_PES_payload()` 的說明」

第 24 圖是表示 `private_stream1_PES_payload()` 的語法。

`private_stream1_PES_payload()` 是以 `private_header()` 及 `private_payload()` 所構成。在 `private_payload()` 中儲存有 ATRAC 音頻串流或 LPCM 音頻串流，字幕串流等非

MPEG 定義的編碼方式的基本串流。

在 `private_header()` 的前頭配置有 `private_stream_id` (8 位元)。`private_stream_id` 為識別儲存於 `private_payload()` 的基本串流之識別資訊，按照其屬性(種類)，例如形成以下那樣的值。

亦即，第 25 圖是表示 `private_stream_id` 的值與儲存於 `private_payload()` 的基本串流的屬性之關係。

在第 25 圖中是採用 `0000xxxxB`，`0001xxxxB`，`100xxxxxB` 的 3 式樣來作為 `private_stream_id` 的值。另外，"x" 是與第 20 圖時同樣，表示 0 或 1 的其中任意的值。

若根據第 25 圖，則 ATRAC 音頻串流所被儲存於 `private_payload()` 的 `private_stream1_PES_payload()` 的 `private_stream_id` 為形成 `0000xxxxB`。另外，`0000xxxxB` 中的下位 4 位元 `xxxx` 為區別 ATRAC 音頻串流的音頻串流號碼，在程式串流 (`MPEG2_Program_Stream()`) 中，可使得能為該音頻串流號碼所區別的數量，亦即 $16(=2^4)$ 個 ATRAC 音頻串流多重化。

又，若根據第 25 圖，則 LPCM 音頻串流所被儲存於 `private_payload()` 的 `private_stream1_PES_payload()` 的 `private_stream_id` 為形成 `0001xxxxB`。另外，`0001xxxxB` 中的下位 4 位元 `xxxx` 為區別 LPCM 音頻串流的音頻串流號碼，在程式串流中，可使得能為該音頻串流號碼所區別的數量，亦即 $16(=2^4)$ 個 LPCM 音頻串流多重化。

又，若根據第 25 圖，則字幕串流所被儲存於

`private_payload()`的 `private_stream1_PES_payload()`的 `private_stream_id` 為形成 100xxxxxB。另外，100xxxxxB 中的下位 5 位元 xxxxx 為區別字幕串流的字幕串流號碼，在程式串流中，可使得為該字幕串流號碼所區別的數量，亦即 $32(=2^5)$ 個字幕串流多重化。

在此，彙整第 23 圖與第 25 圖的關係者為上述第 11 圖。

回到第 24 圖，在 `private_stream1_PES_payload()` 中，接續於 `private_stream_id` 的要素是依照儲存於 `private_payload()` 的基本串流的屬性而內容有所不同。儲存於 `private_payload()` 的基本串流的屬性是依據 `private_header()` 的前頭的 `private_stream_id` 來判斷。

當儲存於 `private_payload()` 的基本串流為 ATRAC 音頻串流時 (`private_stream_id==ATRAC`)，配置有將來擴張用的 `reserved_for_future_use`(8 位元)，其後配置有 `AU_locator`(16 位元)。`AU_locator` 是表示以緊接著該 `AU_locator` 之後的位置為基準，儲存於 `private_payload()` 的 ATRAC 音頻串流的音頻存取單元的前頭位置。當音頻存取單元不存在於 `private_payload()` 時，在 `AU_locator` 中例如記述有 0xFFFF。

當儲存於 `private_payload()` 的基本串流為 LPCM 音頻串流時 (`private_stream_id==LPCM`)，會依次配置 `fs_flag`(1 位元)，`reserved_for_future_use`(3 位元)，`ch_flag`(4 位元)，及 `AU_locator`(16 位元)。

`fs_flag` 是表示儲存於 `private_payload()` 的 LPCM 音頻

串流的取樣頻率。亦即，例如當取樣頻率為48KHz時，`fs_flag`為形成0，當取樣頻率為44.1KHz時，`fs_flag`為形成1。

`ch_flag`是表示儲存於`private_payload()`的LPCM音頻串流的頻道數。例如，當LPCM音頻串流為非立體聲(monaural)時，`ch_flag`為形成1，當LPCM音頻串流為立體時，`ch_flag`為形成2。

`AU_locator`是表示以緊接該`AU_locator`之後的位置為基準，儲存於`private_payload()`的LPCM音頻串流的音頻存取單元的前頭位置。當儲存於`private_payload()`的音頻存取單元不存在時，在`AU_locator`中例如記述有0xFFFF。

當儲存於`private_payload()`的基本串流為字幕串流時(`private_stream_id==SUBTITLE`)，配置有將來擴張用的`reserved_for_future_use`(8位元)，其後配置有

`AU_locator`(16位元)。`AU_locator`是表示以緊接著該`AU_locator`之後的位置為基準，儲存於`private_payload()`的字幕串流的字幕存取單元的前頭位置。當`private_payload()`中不存在字幕存取單元時，在`AU_locator`中例如記述有0xFFFF。

「`private_stream2_PES_payload()`的說明」

其次，第26圖是表示`private_stream2_PES_payload()`的語法。

`private_stream2_PES_payload()`是擴張

private_stream_2的 PES_packet()的 PES_packet_data_byte(第 21圖 A 及第 21圖 B)者，亦即， private_stream_2的 PES_packet()之擴張後的 PES_packet_data_byte，記述有利用於視頻串流的解碼之資訊。

在本實施形態中，private_stream_2的 PES_packet()是被配置於視頻串流的解碼開始可能點之前。因此，在本實施形態中，若從程式串流發現 private_stream_2的 PES_packet()，則可從緊接著之後的視頻串流來開始解碼。

在此，上述第 16圖的 EP_map()的 RPN_EP_start 是表示有關視頻串流方面，private_stream_2的 PES_packet()的前頭的位置。

在 private_stream2_PES_payload()的前頭配置有將來擴張用的 reserved_for_future_use(8位元)，接著依次配置有 video_stream_id(8位元)，1stRef_picture(16位元)，2ndRef_picture(16位元)，3rdRef_picture(16位元)，4thRef_picture(16位元)，au_information()，及 VBI()。

在 video_stream_id 中記述有與配置於緊接著 private_stream_2的 PES_packet()之後的視頻串流的 PES_packet()的 stream_id 同一值。依據該 video_stream_id 來特定有利用儲存於 private_stream_2的 PES_packet()的 private_stream2_PES_payload()之資訊來解碼的視頻串流所被儲存的 PES_packet()。

以相對值各表示依據 1stRef_picture、2ndRef_picture、3rdRef_picture、4thRef_picture、video_stream_id 所特定之視頻串流之包含有從 private_stream_2之 PES_packet()到下一個 private_stream_2之 PES_packet()為止中之 1、2、3、4 號之參照畫像的最後之 pack()之位置。並且，針對 1stRef_picture、2ndRef_picture、3rdRef_picture、4thRef_picture 例如於日本專利特開平 09-46712 號公報 (特願平 07-211420 號) 中詳細揭示 bytes_to_frist_P_pic bytes_to_second_P_pic。

在 au_information() 中記述有關於從 private_stream_2 的 PES_packet() 到其次的 private_stream_2 的 PES_packet() 之視頻串流中的視頻存取單元的資訊。有關 au_information() 的詳細會在往後參照第 27 圖來敘述。

VBI() 是爲了記述 Closed Caption 的資訊而被使用。

具有以上所述 private_stream2_PES_payload() 的 private_stream_2 的 PES_packet() 是被配置於各視頻串流的各解碼開始可能點。

其次，第 27 圖是表示第 26 圖的 au_information() 的語法。

在 au_information() 的前頭配置有 length(16 位元)。Length 是表示含彼之 au_information() 的大小。接續於 length，依次配置有 reserved_for_word_alignment(8 位元)，及 number_of_access_unit(8 位元)。reserved_for_word_alignment 是爲了取辭彙對齊而被使用

。

`number_of_access_unit` 是表示從含彼之 `private_stream_2` 的 `PES_packet()` 到其次的 `private_stream_2` 的 `PES_packet()` 之間所含的視頻存取單元的數量。

亦即，`number_of_access_unit` 是在具有第 26 圖之 `private_stream2_PES_payload()` 為相同之 `video_stream_id` 之 `private_stream_2` 之 `PES_packet()` 之中，從該 `au_information()` 到下一個 `au_information()` 之前為止，亦即，若該 `au_information()` 為剪輯串流檔案中最後之 `au_information()` 時，則至剪輯串流檔案之最後，表示 `video_stream_id` 所表示之視頻串流所包含之存取單元(圖像)之數量。

在 `number_of_access_unit` 之後配置有僅該 `number_of_access_unit` 的數量之 `for` 迴路(`loop`)的內容。亦即，配置有從包含 `number_of_access_unit` 的 `private_stream_2` 的 `PES_packet()` 到其次的 `private_stream_2` 的 `PES_packet()` 之間所含的 1 以上的視頻存取單元的各個相關資訊。

配置於 `for` 迴路內的資訊(有關視頻存取單元的資訊)是形成以下所述。

亦即，在 `for` 迴路內配置有 `pic_struct_copy`(4 位元)，`au_ref_flag`(1 位元)，`AU_length`(21 位元)，`reserved`。

在 `pic_struct_copy` 於視頻流為 MPEG4-AVC(ISO/IEC

14496-10)時，記述有對所對應之視頻存取單元所設定之被 ISO/IEC 14496-10，D.2.2所定義之 `pic_struct()`之複製。並且，`pic_struct()`是例如將圖像當作幀予以顯示，或是顯示圖像之頂部場，之後顯示底部場等之資訊。

在此，第28圖是表示 `pic_struct` 的表。

`pic_struct` 是作為指示如何顯示圖像的顯示方式之顯示方式指示資訊使用。

若根據第28圖之 `pic_struct` 的表，則在依照圖像來指示顯示1幀的情況時，該圖像的 `pic_struct` 中，如第28圖的最左的 `Value` 的欄所示，會被設定0。同樣的，依照圖像來指示顯示頂部場或底部場的情況時，該圖像的 `pic_struct` 中，分別會被設定1或2。又，依照圖像來指示依次顯示頂部場及底部場的情況時，該圖像的 `pic_struct` 中，會被設定3，在指示依次顯示底部場及頂部場的情況時，在 `pic_struct` 中，會被設定4。又，依照圖像來指示依次顯示頂部場、底部場、重複頂部場的情況時，該圖像的 `pic_struct` 中，會被設定5，在指示依次顯示底部場、頂部場、重複底部場的情況時，在 `pic_struct` 中，會被設定6。又，依照圖像來重複2次或3次顯示1幀的情況時，該圖像的 `pic_struct` 中，會分別被設定7或8。

`au_ref_flag` 是表示所對應的存取單元是否為其他的存取單元(的圖像)的解碼時所被參照的參照畫像，為參照畫像時形成1，非參照畫像時形成0。

`AU_length` 是以位元組單位來表示所對應的存取單元

的大小。

[記錄於碟片 101 的資料的具體例]

其次，第 29 圖～第 32 圖是表示記錄於第 1 圖的碟片 101 之上述格式的資料的具體例。

在此，第 29 圖～第 32 圖中，視頻串流為採用 MPEG2-Video，音頻串流為採用 ATRAC 音頻串流。但，視頻串流或音頻串流並非限於此。亦即，視頻串流，例如可採用 MPEG4-Visual 或 MPEG4-AVC 等。又，音頻串流，例如可採用 MPEG1/2/4 音頻或 LPCM 音頻串流等。

另外，字幕串流與視頻串流或音頻串流相異，並非限於以相同間隔來進行連續性的解碼或顯示。亦即，字幕串流有時會從第 2 圖 A 及第 2 圖 B 的緩衝控制模組 215 來供給至字幕解碼器控制模組 218 而解碼。

第 29 圖～第 32 圖是表示在碟片 101 中，如第 6 圖所示，在 "CLIP" 目錄中記錄有 3 個剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP"，在 "STREAM" 目錄中記錄有分別對應於該 3 個剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP" 之 3 個剪輯串流檔案 "00001.PS"，"00002.PS"，"00003.PS" 時之 "PLAYLIST.DAT" 檔案，剪輯資訊檔案 "00001.CLP"，"00002.CLP"，及 "00003.CLP" 等的具體例。但，在第 29 圖～第 32 圖中，省略 "PLAYLIST.DAT" 檔案等的資料的一部份。

亦即，第 29 圖是表示第 7 圖所說明之 "PLAYLIST.DAT"

檔案的具體例。

在第 29 圖中，`number_of_PlayLists` 是形成 2，因此 "PLAYLIST.DAT" 檔案中所含的 `PlayList()` 的數目為 2。在第 29 圖中，該 2 個 `PlayList()` 中的第 1 個為 `PlayList #0`，第 2 個為 `PlayList #1`。

有關第 1 個 `PlayList()` 之 `PlayList #0` 方面，
`capture_enable_flag_PlayList` 會形成 1，因此按照
`PlayList #0` 而再生之視頻資料的 2 次利用會被許可。又，
有關 `PlayList #0` 方面，`number_of_PlayItems` 會形成 2，因此，
`PlayList #0` 中所含的 `PlayItem()` 的數目為 2。在第 29
圖中，該 2 個 `PlayItem()` 之 `PlayItem#0` 及 `PlayItem#1` 的具體例
會被記載於「`PlayList#0`」欄的下方。

在 `PlayList #0` 中所含的第 1 個 `PlayItem()` 之
`PlayItem#0` 中，在第 8 圖所說明的
`Clip_Information_file_name` 會形成 "00001.CLP"，`In_time`
會形成 180,090，`OUT_time` 會形成 27,180,090。因此，依據
`PlayList #0` 的 `PlayItem#0` 而再生的剪輯是對應於剪輯資訊
檔案 "00001.CLP" 之剪輯串流檔案 "00001.PS" 的時刻
180,090 ~ 27,180,090。

在 `PlayList #0` 中所含的第 2 個 `PlayItem()` 之
`PlayItem#1` 中，在第 8 圖所說明的
`Clip_Information_file_name` 會形成 "00002.CLP"，`In_time`
會形成 90,000，`OUT_time` 會形成 27,090,000。因此，依據
`PlayList #0` 的 `PlayItem#1` 而再生的剪輯是對應於剪輯資訊

檔案 "00002.CLP"之剪輯串流檔案 "00002.PS"的時刻 90,000 ~ 27,090,000。

另一方面，在第29圖中，有關第2個 `PlayList()`之 `PlayList #1`方面，`capture_enable_flag_PlayList`會形成0，因此，按照 `PlayList #1`而再生之視頻資料的2次利用未被許可。又，有關 `PlayList #1`方面，`number_of_PlayItems`會形成1，因此，`PlayList #1`中所含的 `PlayItem()`的數目為1。在第29圖中，該1個 `PlayItem()`之 `PlayItem#0`的具體例會被記載於「`PlayList#1`」欄的下方。

在 `PlayList #1`中所含的1個 `PlayItem()`之 `PlayItem#0`中，在第8圖所說明的 `Clip_Information_file_name`會形成 "00003.CLP"，`In_time`會形成 90,000，`OUT_time`會形成 81,090,000。因此，依據 `PlayList #1`的 `PlayItem#0`而再生的剪輯是對應於剪輯資訊檔案 "00003.CLP"之剪輯串流檔案 "00003.PS"的時刻 90,000 ~ 81,090,000。

其次，第30圖 A 及第30圖 B 是表示在第12圖所說明之剪輯資訊檔案 `Clip()`的具體例。亦即，第30圖 A 及第30圖 B 是表示第6圖的剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP"的具體例。

在剪輯資訊檔案 "00001.CLP"中，`presentation_start_time`會形成 90,000，`presentation_end_time`會形成 27,990,000。因此，若利用對應於剪輯資訊檔案 "00001.CLP"的剪輯串流檔案

"00001.PS"中所儲存的程式串流，則可利用310秒量
((27,990,000-90,000)/90kHz)的內容。

並且，在剪輯資訊檔案"00001.CLP"中，
capture_enable_flag_Clip會形成1，因此，有關對應於剪
輯資訊檔案"00001.CLP"的剪輯串流檔案"00001.PS"中所
被儲存的程式串流中被多重化的視頻串流，其2次利用會
被許可。

而且，在第30圖A及第30圖B中，剪輯資訊檔案
"00001.CLP"的 number_of_streams 會形成4，因此，對應
之剪輯串流檔案"00001.PS"中所被儲存的程式串流中，4
個基本串流會被多重化。

若該4個基本串流為 stream#0，stream#1，stream#2，
stream#3，則於第30圖A及第30圖B中，該4個基本串流
stream#0，stream#1，stream#2，stream#3的各個
StreamInfo()(第12圖)的具體例會被記載於「"00001.CLP"
」欄的下方。

有關剪輯串流檔案"00001.PS"的第1個基本串流
stream#0方面，stream_id會形成0xE0，因此，該基本串
流 stream#0，如第23圖及第25圖(或第13圖)所說明那樣，
為視頻串流。另外，本實施形態中，有關視頻串流方面，
如上述，無關 private_stream_id，在第30圖A及第30圖B
中是形成0x00。

又，有關剪輯串流檔案"00001.PS"的第1個基本串流
之視頻串流 stream#0方面，其 StreamInfo()中所含的

StaticInfo()(第 14 圖)的 picture_size 會形成 '720×480'，frame_rate 會形成 '29.97Hz'，cc_flag 會形成 'Yes'。因此，此視頻串流 stream#0 為 720×480 畫素之幀週期為 29.97Hz 的視頻資料，且含隱藏式字幕資料。

又，有關剪輯串流檔案 "00001.PS" 的第 1 個基本串流之視頻串流 stream#0 方面，StreamInfo()(第 12 圖)的 number_of_DynamicInfo 會形成 0，pts_change_point 與 DynamicInfo()的組合不存在。

其次，有關剪輯串流檔案 "00001.PS" 的第 2 個基本串流 stream#1 方面，stream_id 會形成 0xBD，private_stream_id 會形成 0x00。因此，該基本串流 stream#1，如第 23 圖及第 25 圖所說明那樣，為 ATRAC 音頻串流。

又，有關剪輯串流檔案 "00001.PS" 的第 2 個基本串流之 ATRAC 音頻串流 stream#1 方面，該 StreamInfo()中所含的 StaticInfo()(第 14 圖)的 audio_language_code 會形成 '日本語'，channel_configuration 會形成 'STEREO'，lfe_existence 會形成 'NO'，sampling_frequency 會形成 '48kHz'。因此，該 ATRAC 音頻串流 stream#1 為日本語，且立體的音頻資料。並且，不含低域強調頻道，取樣頻率為 48kHz。

又，有關剪輯串流檔案 "00001.PS" 的第 2 個基本串流之 ATRAC 音頻串流 stream#1 方面，StreamInfo()(第 12 圖)的 number_of_DynamicInfo 會形成 0，pts_change_point 與

DynamicInfo()的組合不存在。

其次，有關剪輯串流檔案"00001.PS"的第3個基本串流 stream#2方面，stream_id 會形成 0xBD，private_stream_id 會形成 0x80。因此，該基本串流 stream#2，如第 23圖及第 25圖所說明那樣，為字幕串流。

又，有關剪輯串流檔案"00001.PS"的第3個基本串流之字幕串流 stream#2方面，該 StreamInfo()中所含的 StaticInfo()(第 14圖)的 subtitle_language_code 會形成'日本語'，configurable_flag 會形成 0。因此，該字幕串流 stream#2為日本語的字幕資料，且不被許可變更其顯示方式。

又，有關剪輯串流檔案"00001.PS"的第3個基本串流之字幕串流 stream#2方面，StreamInfo()(第 12圖)的 number_of_DynamicInfo 會形成 0，pts_change_point 與 DynamicInfo()的組合不存在。

其次，有關剪輯串流檔案"00001.PS"的第4個基本串流 stream#3方面，stream_id 會形成 0xBD，private_stream_id 會形成 0x81。因此，該基本串流 stream#3，如第 23圖及第 25圖所說明那樣，為字幕串流。

又，為了區別剪輯串流檔案"00001.PS"的第3個基本串流之字幕串流 stream#2與第4個基本串流之字幕串流 stream#3，各個 private_stream_id 是形成 0x80與 0x81。

又，有關剪輯串流檔案"00001.PS"的第4個基本串流之字幕串流 stream#2方面，該 StreamInfo()中所含的

StaticInfo()(第14圖)的 subtitle_language_code 會形成 '日本語'，configurable_flag 會形成 1。因此，該字幕串流 stream#3 為日本語的字幕資料，且許可變更其顯示方式。

又，有關剪輯串流檔案 "00001.PS" 的第 4 個基本串流之字幕串流 stream#3 方面，StreamInfo()(第12圖)的 number_of_DynamicInfo 會形成 0，pts_change_point 與 DynamicInfo() 的組合不存在。

其次，在第30圖 A 及第30圖 B 中，有關剪輯資訊檔案 "00002.CLP" 方面，presentation_start_time 會形成 90,000，presentation_end_time 會形成 27,090,000。因此，若利用對應於剪輯資訊檔案 "00002.CLP" 的剪輯串流檔案 "00002.PS" 中所儲存的程式串流，則可利用 300 秒量 $((27,090,000 - 90,000) / 90\text{kHz})$ 的內容。

並且，在剪輯資訊檔案 "00002.CLP" 中，capture_enable_flag_Clip 會形成 0，因此有關對應於剪輯資訊檔案 "00002.CLP" 的剪輯串流檔案 "00002.PS" 中所被儲存的程式串流中被多重化的視頻串流，其 2 次利用未被許可。

而且，在第30圖 A 及第30圖 B 中，剪輯資訊檔案 "00002.CLP" 的 number_of_streams 會形成 4，因此，在儲存於所對應之剪輯串流檔案 "00002.PS" 的程式串流中，與上述剪輯串流檔案 "00001.PS" 的情況時同樣，4 個基本串流會被多重化。

若該 4 個基本串流為 stream#0，stream#1，stream#2，

stream#3，則於第30圖 A 及第30圖 B 中，該4個基本串流 stream#0，stream#1，stream#2，stream#3的各個 StreamInfo()(第12圖)的具體例會被記載於「"00002.CLP"」欄的下方。

在此，於第30圖 A 及第30圖 B 中，剪輯串流檔案 "00002.PS"的第1個～第4個基本串流 stream#0～#3的各個 StreamInfo()的內容是與上述剪輯串流檔案 "00001.PS"的第1～第4個基本串流 stream#0～#3的各個 StreamInfo()的內容相同，因此省略該說明。

另外，如上述，由於剪輯串流檔案 "00002.PS"的第1～第4個基本串流 stream#0～#3的各個 StreamInfo()的內容與剪輯串流檔案 "00001.PS"的第1～第4個基本串流 stream#0～#3的各個 StreamInfo()的內容相同，因此剪輯串流檔案 "00002.PS"的第1個基本串流 stream#0為視頻串流，第2個基本串流 stream#1為 ATRAC 音頻串流。並且，該第3個基本串流 stream#2與第4個基本串流 stream#3皆為字幕串流。

其次，在第30圖 A 及第30圖 B 中，有關剪輯資訊檔案 "00003.CLP" 方面，presentation_start_time 會形成 90,000，presentation_end_time 會形成 81,090,000。因此，若利用對應於剪輯資訊檔案 "00003.CLP"的剪輯串流檔案 "00003.PS"中所被儲存的程式串流，則可利用 900秒量 $((81,090,000-90,000)/90\text{kHz})$ 的內容。

又，有關剪輯資訊檔案 "00003.CLP"方面，

`capture_enable_flag_Clip` 會形成 1，因此，有關對應於剪輯資訊檔案 "00003.CLP" 的剪輯串流檔案 "00003.PS" 中所被儲存的程式串流中被多重化的視頻串流，其 2 次利用會被許可。

此外，在第 30 圖 A 及第 30 圖 B 中，剪輯資訊檔案 "00003.CLP" 的 `number_of_streams` 是形成 3，因此，在儲存於所對應之剪輯串流檔案 "00003.PS" 的程式串流中，3 個基本串流會被多重化。

若該 3 個基本串流為 `stream#0`，`stream#1`，`stream#2`，則於第 30 圖 A 及第 30 圖 B 中，該 3 個基本串流 `stream#0`，`stream#1`，`stream#2` 的各個 `StreamInfo()` (第 12 圖) 的具體例會被記載於「"00003.CLP"」欄的下方。

有關剪輯串流檔案 "00003.PS" 的第 1 個基本串流 `stream#0` 方面，`stream_id` 會形成 0xE0，因此，該基本串流 `stream#0`，如第 23 圖及第 25 圖 (或第 13 圖) 所說明那樣，為視頻串流。另外，與剪輯串流檔案 "00001.PS" 的第 1 個基本串流 `stream#0` 同樣的，`private_stream_id` 是形成 0x00。

又，有關剪輯串流檔案 "00003.PS" 的第 1 個基本串流之視頻串流 `stream#0` 方面，該 `StreamInfo()` 中所含的 `StaticInfo()` (第 14 圖) 的 `picture_size` 會形成 '720×480'，`frame_rate` 會形成 '29.97Hz'，`cc_flag` 會形成 'No'。因此，該視頻串流 `stream#0` 為 720×480 畫素之幀週期為 29.97Hz 的視頻資料，且未含隱藏式字幕資料。

又，有關剪輯串流檔案 "00003.PS" 的第 1 個基本串流之視頻串流 `stream#0` 方面，`StreamInfo()` (第 12 圖) 的 `number_of_DynamicInfo` 會形成 2，因此，在該 `StreamInfo()` 中，`pts_change_point` 與 `DynamicInfo()` 的組合會有 2 組被記述。

其次，有關剪輯串流檔案 "00003.PS" 的第 2 個基本串流 `stream#1` 方面，`stream_id` 會形成 0xE1，因此，該基本串流 `stream#1`，如第 23 圖及第 25 圖 (或第 13 圖) 所說明那樣，為視頻串流。另外，為了區別剪輯串流檔案 "00003.PS" 的第 1 個基本串流之視頻串流 `stream#0` 與第 2 個基本串流之視頻串流 `stream#1`，各個 `stream_id` 是形成 0xE0 與 0xE1。並且，與剪輯串流檔案 "00001.PS" 的第 1 個基本串流 `stream#0` 同樣的，`private_stream_id` 是形成 0x00。

又，有關剪輯串流檔案 "00003.PS" 的第 2 個基本串流之視頻串流 `stream#1` 方面，該 `StreamInfo()` 中所含的 `StaticInfo()` (第 14 圖) 的 `picture_size`，`frame_rate`，`cc_flag` 會與有關第 1 個基本串流之視頻串流 `stream#0` 者形成相同。因此，剪輯串流檔案 "00003.PS" 的第 2 個基本串流之視頻串流 `stream#1` 為 720×480 畫素之幀週期為 29.97Hz 的視頻資料，且未含隱藏式字幕資料。

又，有關剪輯串流檔案 "00003.PS" 的第 2 個基本串流之視頻串流 `stream#1` 方面，`StreamInfo()` (第 12 圖) 的 `number_of_DynamicInfo` 會形成 0，`pts_change_point` 與 `DynamicInfo()` 的組合不存在。

其次，有關剪輯串流檔案 "00003.PS" 的第3個基本串流 stream#2 方面，stream_id 會形成 0xBD，private_stream_id 會形成 0x00。因此，該基本串流 stream#2，如第23圖及第25圖所說明那樣，為 ATRAC 音頻串流。

又，有關剪輯串流檔案 "00003.PS" 的第3個基本串流之 ATRAC 音頻串流 stream#2 方面，該 StreamInfo() 中所含的 StaticInfo()(第14圖)的 audio_language_code，channel_configuration，lfe_existence，sampling_frequency 會與剪輯串流檔案 "00001.PS" 的第2個基本串流之 ATRAC 音頻串流 stream#1 者形成相同。因此，剪輯串流檔案 "00003.PS" 的第3個基本串流之 ATRAC 音頻串流 stream#2 為日本語且立體的音頻資料。並且，未含低域強調頻道，取樣頻率為 48kHz。

又，有關剪輯串流檔案 "00003.PS" 的第3個基本串流之 ATRAC 音頻串流 stream#2 方面，StreamInfo()(第12圖)的 number_of_DynamicInfo 會形成 3，因此在該 StreamInfo() 中，pts_change_point 與 DynamicInfo() 的組合會有 3 組被記述。

其次，第31圖是表示在第12圖所說明之剪輯資訊檔案 Clip() 中的 EP_map() 具體例。亦即，第31圖是表示第6圖的剪輯資訊檔案 "00001.CLP" 之中的第16圖的 EP_map() 的具體例。

在第31圖中，EP_map() 的

`number_of_stream_id_entries` 是形成 1，因此，在該 `EP_map()` 中記述有關於 1 個基本串流的解碼開始可能點的資訊。

並且，在第 31 圖的 `EP_map()` 中，`stream_id` 會形成 `0xE0`。因此，由第 23 圖及第 25 圖所說明的內容，在 `EP_map()` 中記述有關於依據形成 `0xE0` 的 `stream_id` 而特定的視頻串流之解碼開始可能點的 `RAPI`(隨機存取點資訊) 的資訊，`PTS_EP_start` 及 `RPN_EP_start`(第 16 圖)。亦即，第 31 圖是剪輯資訊檔案 "00001.CLP" 的 `EP_map()`，在對應於剪輯資訊檔案 "00001.CLP" 的剪輯串流檔案 "00001.CLP" 中，`stream_id` 為 `0xE0` 的基本串流，如第 30 圖 A 及第 30 圖 B 所說明那樣，為剪輯串流檔案 "00001.CLP" 的第 1 個視頻串流 `stream#0`，因此第 31 圖的 `EP_map()` 中所被記述的資訊為該視頻串流 `stream#0` 的解碼開始可能點的 `PTS_EP_start` 及 `RPN_EP_start`。

在第 31 圖中，記載有剪輯串流檔案 "00001.CLP" 的第 1 個視頻串流 `stream#0` 的解碼開始可能點之中從前頭開始 5 點的 `PTS_EP_start` 及 `RPN_EP_start`，第 6 點以後的 `PTS_EP_start` 及 `RPN_EP_start` 的記載省略。

`RPN_EP_start`、`PTS_EP_start`、`1stRef_Picture`、`2ndRef_Picture`、`3rdRef_Picture`、`4thRef_Picture` 是除了多重化串流中所被多重化的所有 `RAPI` 的開始位置以外，還有緊接於 `RAPI` 之後所被放置的幀內畫面及接續的第 2 張、第 3 張、第 4 張的參照畫像的終了位置。

前頭的 RAPI 的位置為 0(區段)，緊接著之後的幀內畫面的 PTS 為 90,000。並且，該當幀內畫面、及第 2 張、第 3 張、第 4 張的參照畫像的終了位置係來自 RAPI 的前頭的相對區段數分別為 28，37，48，58。

又，第 2 個 RAPI 的位置為 244(區段)，緊接著之後的幀內畫面的 PTS 為 135,045。並且，該當幀內畫面、及第 2 張、第 3 張、第 4 張的參照畫像的終了位置係來自 RAPI 的前頭的相對區段數分別為 10，18，25，31。

又，第 3 個 RAPI 的位置為 305(區段)，緊接著之後的幀內畫面的 PTS 為 180,090。並且，該當幀內畫面、及第 2 張、第 3 張、第 4 張的參照畫像的終了位置係來自 RAPI 的前頭的相對區段數分別為 25，44，50，54。

又，第 4 個 RAPI 的位置為 427(區段)，緊接著之後的幀內畫面的 PTS 為 225,135。並且，該當幀內畫面、及第 2 張、第 3 張、第 4 張的參照畫像的終了位置係來自 RAPI 的前頭的相對區段數分別為 8，15，22，29。

又，第 5 個 RAPI 的位置為 701(區段)，緊接著之後的幀內畫面的 PTS 為 270,180。並且，該當幀內畫面、及第 2 張、第 3 張、第 4 張的參照畫像的終了位置係來自 RAPI 的前頭的相對區段數分別為 26，32，41，48。

N-th_Ref_picture_copy 係儲存有 4 點的參照畫像終了位置 (1stRef_Picture、2ndRef_Picture、3rdRef_Picture、4thRef_Picture) 中，接近所定的區段數(在編碼處理中，集中讀出可能的區段數)的值。在第 31 圖中，區段數最接

近 "30" 者 會 被 選 擇 。

例如，對前頭的記入項(entry)而言為
PTS_EP_start=90,000、RPN_EP_start=0。N-th_Ref_picture_copy 係由 28，37，48，58之中選擇最接近
"30"的 28，其結果在 index_N_minus1中儲存有顯示
1stRef_picture 的值 "0"。

其次，對第2個的記入項而言為 PTS_EP_start=135，
045、RPN_EP_start=244。N-th_Ref_picture_copy 係由 10
， 18， 25， 31之中選擇最接近 "30"的 31，其結果在
index_N_minus1中儲存有顯示 4thRef_picture 的值 "3"。

藉此，針對第31圖之例的5點的記入項之
index_N_minus1及 N-th_Ref_picture_copy 係儲存有 (0，
28)，(3，31)，(0，25)，(3，29)，(1，32)。

此選擇算法是考量再生機器的再生品質等來綜合性地
決定。因次，在本實施例中，雖是選擇最接近較小的區段
數 "30" 者，但亦可為除此以外的區段數。又，當
index_N_minus1的值小時，從 RAPI 的位置來讀入僅 N-
th_Ref_picture_copy 大小的資料時，於此所含的參照畫像
數少，相反的，當 index_N_minus1的值大，所含的參照
畫像數多。

又，此例中所有5點的資料中皆記述有4張的參照畫像
的終了位置，但依照視頻編碼的方法、或幀內畫面的間隔
，參照畫像可能為未滿4張。此情況，可為記述對各進入
點(entry point)所存在之4張以下的最大的參照畫像的終了

位置之方法。

另外，在第31圖的 `EP_map()` 中，`private_stream_id` 是形成 0x00，當 `stream_id` 為表示視頻串流時，如上述，無關 `private_stream_id`。

其次，第32圖是表示在第29圖所說明之 `PlayList #0` 及 `PlayList #1` (第7圖的 `PlayList()`) 之中的 `PlayListMark()` 的具體例。

第32圖上側是表示 `PlayList #0` 的 `PlayListMark()` (第9圖)。

在第32圖上側，`PlayList #0` 的 `PlayListMark()` 之 `number_of_PlayList_marks` 是形成 7，因此，`PlayList #0` 的 `PlayListMark()` 中所含的 `Mark()` 的數目為 7。

並且，在第32圖上側，因為 `mark_type` (第9圖) 是形成 'Chapter'，所以 `PlayList #0` 中所含的 7 個 `Mark()` 之中的第 1 個 `Mark()` 之 `Mark#0` 為章節標示。又，因為

`ref_to_PlayItem_id` (第9圖) 形成 0，所以 `Mark#0` 是屬於 `PlayList #0` 中所含的第29圖的 2 個 `PlayItem#0` 及 `#1` 之中的 `PlayItem#0`。又，因為 `mark_time_stamp` 形成 180,090，所以 `Mark#0` 為依據 `PlayList #0` 中所含的 `PlayItem#0` 而再生之剪輯串流檔案的時刻 (再生時刻) 180,090 上的記號。又，因為 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 皆形成 0，所以 `Mark#0` 無關聯於任何基本串流。又，因為 `mark_data` 形成 1，所以 `Mark#0` 是表示號碼為 1 的章節。

另外，在此，依據 `PlayList #0` 中所含的 `PlayItem#0` 而

再生的剪輯串流檔案是第29圖所說明之由記述於該 PlayItem#0 的 Clip_Infomation_file_name 的 "00001.CLP" 來特定的剪輯串流檔案 "00001.PS"，因此，Mark#0 的 mark_time_stamp 所示之上述的時刻 180,090 為剪輯串流檔案 "00001.PS" 的時刻。

在第32圖上側，PlayList #0中所含的7個 Mark() 之中的第5個 Mark() 之 Mark#4 也是與第1個 Mark#0 同樣的章節標示。

亦即，因為 mark_type(第9圖)是形成 'Chapter'，所以第5個 Mark() 之 Mark#4 為章節標示。又，因為 ref_to_PlayItem_id(第9圖)形成 1，所以 Mark#4 是屬於 PlayList #0 中所含的第29圖的2個 PlayItem#0 及 #1 之中的 PlayItem#1。又，因為 mark_time_stamp 形成 90,000，所以 Mark#4 為依據 PlayList #0 中所含的 PlayItem#1 而再生之剪輯串流檔案的時刻 90,000 上的記號。又，因為 entry_ES_stream_id 及 entry_ES_private_stream_id 皆形成 0，所以 Mark#4 無關聯於任何的基本串流。又，因為 mark_data 形成 2，所以 Mark#4 是表示號碼為 2 的章節。

另外，在此，依據 PlayList #0 中所含的 PlayItem#1 而再生的剪輯串流檔案是第29圖所說明之由記述於該 PlayItem#1 的 Clip_Infomation_file_name 的 "00002.CLP" 來特定的剪輯串流檔案 "00002.PS"，因此，Mark#4 的 mark_time_stamp 所示之上述的時刻 90,000 為剪輯串流檔案 "00002.PS" 的時刻。

並且，在第32圖上側，因為 `mark_type`(第9圖)是形成 'Index'，所以 `PlayList #0`中所含的7個 `Mark()`之中的第2個 `Mark()`之 `Mark#1`為索引標示。又，因為 `ref_to_PlayItem_id`(第9圖)形成0，所以 `Mark#1`是屬於 `PlayList #0`中所含的第29圖的2個 `PlayItem#0`及 `#1`之中的 `PlayItem#0`。又，因為 `mark_time_stamp` 形成5,580,090，所以 `Mark#1`為依據 `PlayList #0`中所含的 `PlayItem#0`而再生之剪輯串流檔案的時刻5,580,090上的記號。又，因為 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 皆形成0，所以 `Mark#1`是無關聯於任何的基本串流。又，因為 `mark_data` 形成1，所以 `Mark#1`是表示號碼為1的索引。

另外，在此，依據 `PlayList #0`中所含的 `PlayItem#0`而再生的剪輯串流檔案，如上述，為剪輯串流檔案 "00001.PS"，因此，`Mark#1`的 `mark_time_stamp` 所示之上述的時刻5,580,090為剪輯串流檔案 "00001.PS"的時刻。

在第32圖上側，`PlayList #0`中所含的7個 `Mark()`之中的第3個，第6個，第7個 `Mark()`之 `Mark#2`，`Mark#5`，`Mark#6`也是與第2個 `Mark#1`同樣的索引標示。

並且，在第32圖上側，`mark_type`(第9圖)形成 'Event'，所以 `PlayList #0`中所含的7個 `Mark()`之中的第4個 `Mark()`之 `Mark#3`為事件標示。又，因為 `ref_to_PlayItem_id`(第9圖)形成0，所以 `Mark#3`是屬於 `PlayList #0`中所含的第29圖的2個 `PlayItem#0`及 `#1`之中的 `PlayItem#0`。又，因為 `mark_time_stamp` 形成16,380,090，

所以 Mark#3 為依據 PlayList #0 中所含的 PlayItem#0 而再生之剪輯串流檔案的時刻 16,380,090 上的記號。又，因為 entry_ES_stream_id 及 entry_ES_private_stream_id 皆形成 0，所以 Mark#3 無關聯於任何的基本串流。又，因為 mark_data 形成 0，所以 Mark#3 是作為引數來使隨同 0 的事件發生。

另外，在此，依據 PlayList #0 中所含的 PlayItem#0 而再生的剪輯串流檔案，如上述，為剪輯串流檔案 "00001.PS"，因此，Mark#3 的 mark_time_stamp 所示之上述的時刻 16,380,090 為剪輯串流檔案 "00001.PS" 的時刻。

在此，第 32 圖上側，於 PlayList #0 的 PlayListMark() 的一覽表的欄外右側顯示有來自 Mark() 所屬的 PlayItem() 的前頭之時間，更右側顯示有來自 PlayList #0 的前頭之時刻。

其次，第 32 圖下側是表示 PlayList#1 的 PlayListMark() (第 9 圖)。

在第 32 圖下側，PlayList#1 的 PlayListMark() 之 number_of_PlayList_marks 是形成 3，因此，PlayList#1 (的 PlayListMark()) 中所含的 Mark() 的數目為 3。

此外，在第 32 圖下側，因為 mark_type (第 9 圖) 是形成 'Chapter'，所以 PlayList#1 中所含的 3 個 Mark() 之中的第 1 個的 Mark() 之 Mark#0 為章節標示。又，因為 ref_to_PlayItem_id (第 9 圖) 形成 0，所以 Mark#0 是屬於 PlayList#1 中所含的第 29 圖的 1 個 PlayItem#0。又，因為

mark_time_stamp 形成 90,000，所以 Mark#0 為依據 PlayList#1 中所含的 PlayItem#0 而再生之剪輯串流檔案的時刻 90,000 上的記號。又，因為 entry_ES_stream_id 及 entry_ES_private_stream_id 皆形成 0，所以 Mark#0 無關聯於任何的基本串流。又，因為 mark_data 形成 0，所以 Mark#0 是表示號碼為 0 的章節。

另外，在此，依據 PlayList#1 中所含的 PlayItem#0 而再生的剪輯串流檔案是第 29 圖所說明之由記述於該 PlayItem#0 的 Clip_Infomation_file_name 的 "00003.CLP" 來特定的剪輯串流檔案 "00003.PS"，因此，Mark#0 的 mark_time_stamp 所示之上述的時刻 90,000 為剪輯串流檔案 "00003.PS" 的時刻。

在第 32 圖下側，因為 mark_type(第 9 圖)是形成 'Event'，所以 PlayList#1 中所含的 3 個 Mark() 之中的第 2 個 Mark() 之 Mark#1 為事件標示。又，因為 ref_to_PlayItem_id(第 9 圖)形成 0，所以 Mark#1 是屬於 PlayList#1 中所含的第 29 圖的 1 個 PlayItem#0。又，因為 mark_time_stamp 形成 27,090,000，所以 Mark#1 為依據 PlayList#1 中所含的 PlayItem#0 而再生之剪輯串流檔案的時刻 27,090,000 上的記號。又，因為 entry_ES_stream_id 為 0xE0，entry_ES_private_stream_id 形成 0，所以 Mark#1 是在 stream_id 為 0xE0 下所被特定的基本串流，亦即，如第 23 圖及第 25 圖所說明那項，與視頻串流有所關聯。又，因為 mark_data 形成 1，所以 Mark#1 是作為引數來使隨同 1 的事

件發生。

另外，在此，依據 PlayList#1中所含的 PlayItem#0而再生的剪輯串流檔案，如上述，為剪輯串流檔案 "00003.PS"，因此，Mark#1的 mark_time_stamp 所示之上述的時刻 27,090,000為剪輯串流檔案 "00003.PS"的時刻。

再者，Mark#1被賦予關聯之 stream_id 為 0xE0之視頻串流，是該 Mark#1所屬之被記述在第 29圖之 PlayList#1所包含之 PlayItem#0之 Clip_Information_file_name 的 "0003.CLP"中所記述之 stream_id 為 0xE0之視頻串流，亦即，由第 30A 及 B 圖之剪輯資訊檔案 "0003.CLP"所特定之剪輯串流檔案 "00003.PS"中被多重化之 3個基本串流 stream#0至 #2中之第 1個基本串流(視頻串流)stream#0。

其次，在第 32圖下側，因為 mark_type(第 9圖)是形成 'Event'，所以 PlayList#1中所含的 3個 Mark()之中的第 3個 Mark()之 Mark#2為事件標示。又，因為 ref_to_PlayItem_id(第 9圖)形成 0，所以 Mark#2是屬於 PlayList#1中所含的第 29圖的 1個 PlayItem#0。又，因為 mark_time_stamp 形成 27,540,000，所以 Mark#1為依據 PlayList#1中所含的 PlayItem#0而再生之剪輯串流檔案的時刻 27,540,000上的記號。又，因為 entry_ES_stream_id 為 0xE1，entry_ES_private_stream_id 形成 0，所以 Mark#2是在 stream_id 為 0xE1下所被特定的基本串流，亦即，如第 23圖及第 25圖所說明那樣，與視頻串流有所關聯。又，因為 mark_data 形成 2，所以 Mark#2是作為引數來

使隨同2的事件發生。

另外，在此，依據 `PlayList#1`中所含的 `PlayItem#0`而再生的剪輯串流檔案，如上述，為剪輯串流檔案 "`00003.PS`"，因此，`Mark#2`所示之上述的時刻 `27,540,000`為剪輯串流檔案 "`00003.PS`"的時刻。

再者，`Mark#2`被賦予關聯之 `stream_id` 為 `0xE1`之視頻串流，是該 `Mark#2`所屬之被記述在第29圖之 `PlayList#1`所包含之 `PlayItem#0`之 `Clip_Information_file_name` 的 "`0003.CLP`"上所記述之 `stream_id` 為 `0xE1`之視頻串流，亦即，由第30A及B圖之剪輯資訊檔案 "`0003.CLP`"所特定之剪輯串流檔案 "`00003.PS`"中被多重化之3個基本串流 `stream#0`至`#2`中之第2個基本串流(視頻串流)`stream#1`。

在此，第32圖下側，於 `PlayList#1`的 `PlayListMark()`的一覽表的欄外右側顯示有來自 `Mark()`所屬的 `PlayItem()`的前頭之時刻。

另外，在第32圖中，雖然章節標示或索引標示所示的章節或索引的號碼會被記述於 `mark_data`，但即使章節標示或索引標示所示的章節或索引的號碼不被記述於 `mark_data`，照樣可以例如藉由計數 `PlayListMark()`之章節標示或索引標示的數目來認定。

[碟片再生裝置的動作說明]

其次，說明有關在第1圖的碟片101記錄有第29圖～第32圖所說明的資料之第1圖的碟片再生裝置的動作。

在此，若依照多重方式之 `MPEG2-system` 的規定，則

顯示時間標籤不必附加於所有的存取單元，其間隔可為0.7秒以下。亦即，存在具有顯示時間標籤的存取單元不具有顯示時間標籤的存取單元。

另外，就此所說明的例子而言，是假設在解碼開始位置的視頻串流的存取單元中必定附加有顯示時間標籤。亦即「再生準備處理」如後述，解碼開始位置是使用EP_map()，利用二分探索等來檢索符合式 $PTS_EP_start \leq IN_time$ 的最大PTS_EP_start。在登錄於EP_map()之緊接於視頻的再生開始位置之後的存取單元必定附加有顯示時間標籤。

此外，孤立場(non-paired field)並不存在。亦即，在pic_struct=1的存取單元之後必定放置pic_struct=2的存取單元。並且，在pic_struct=2的存取單元之後必定放置pic_struct=1的存取單元。

再者，此例中，pic_struct=7及8不會發生。

一旦碟片101被插入碟片驅動器102，則該訊息會經由驅動器介面114，以及第2圖A及第2圖B的操作系統201來交付給視頻內容再生程式210。一旦視頻內容再生程式210從操作系統201接到碟片101被插入碟片驅動器102的訊息，則開始第33圖的再生前處理。

「再生前處理」

亦即，第33圖是用以說明視頻內容再生程式210所進行的再生前處理的流程圖。

在此，以下根據流程圖來說明的碟片再生裝置的動作

或處理，並非一定要按照流程圖記載的順序來進行，亦可並列或個別進行。但，本說明書中，爲了便於說明，是按照流程圖來說明碟片再生裝置的動作或處理。

在再生前處理中，視頻內容再生程式210是在步驟S101中，使用操作系統201的檔案系統功能，檢查碟片101，判定碟片101是否爲視頻內容再生程式210用的正常碟片。

在此，如上述，對碟片101的存取(檔案的讀出)是使用操作系統201的檔案系統功能來進行，但在以下省略其說明。

在步驟S101中，當碟片101被判定爲不是正常的碟片時，亦即，例如，採用於碟片101的檔案系統不是對應於操作系統201的類型時，或者在根目錄中未被放置"VIDEO"目錄時，判斷成視頻內容再生程式210非對應於碟片101，而前進至步驟S102，繪圖處理模組219會進行錯誤(error)處理，終了再生前處理。

亦即，繪圖處理模組219所進行的錯誤處理是產生碟片101非正常的錯誤訊息，使由視頻輸出模組220輸出，藉此來顯示錯誤訊息。另外，其他錯誤處理，例如可由音頻輸出模組221來輸出警告聲音，或從碟片驅動器102來排出碟片101等。

另一方面，在步驟S101中，當碟片101被判定爲正常的碟片時，前進至步驟S103，視頻內容再生程式210會藉由內容資料供給模組213來對操作系統201要求讀入放置於

碟片 101(第 6 圖)的 "VIDEO"目錄之 "SCRIPT.DAT"檔案及 "PLAYLIST.DAT"檔案的 2 個資料檔案，前進至步驟 S104。在步驟 S104 中，"SCRIPT.DAT"檔案會被供應給腳本控制模組 211，且 "PLAYLIST.DAT"檔案會被供應給播放器控制模組 212。

然後，從步驟 S104 前進至 S105～S107，播放器控制模組 212 會進行初期化處理。並且，腳本控制模組 211 會等待至播放器控制模組 212 的初期化處理終了。

「播放器控制模組 212 的初期化處理」

在初期化處理中，於步驟 S105 中，播放器控制模組 212 會進行 "PLAYLIST.DAT"檔案的解析，調查 "PLAYLIST.DAT"檔案中所被使用的剪輯資訊檔案的數目及其檔案名。

亦即，現在的情況，"PLAYLIST.DAT"檔案是形成第 29 圖所示者，播放器控制模組 212 是由第 29 圖的 "PLAYLIST.DAT"檔案中 `number_of_PlayLists` 形成 2，來認定存在第 1 個 `PlayList#0` 及第 2 個 `PlayList#1` 的 2 個 `PlayList()`。又，播放器控制模組 212 是由有關第 29 圖的 "PLAYLIST.DAT"檔案之第 1 個 `PlayList#0` 方面，`number_of_PlayItems` 形成 2，來認定該 `PlayList#0` 中存在第 1 個 `PlayItem#0` 及第 2 個 `PlayItem#1` 的 2 個 `PlayItem()`。而且，播放器控制模組 212 是藉由參照第 29 圖的 "PLAYLIST.DAT"檔案之 `PlayList#0` 中所含的第 1 個

PlayItem#0及第2個 PlayItem#1的各個 Clip_Information_file_name 來認定 PlayList#0中所含的第1個 PlayItem#0的剪輯資訊檔案為 "00001.CLP"，第2個 PlayItem#1的剪輯資訊檔案為 "00002.CLP"。

同樣有關第2個 PlayList#1方面，播放器控制模組 212 是由該 number_of_PlayItems 為 1來認定存在1個 PlayItem()(PlayItem#0)，且由該 PlayItem#0之中的 Clip_Information_file_name 來認定該 PlayItem#0的剪輯資訊檔案為 "00003.CLP"。

然後，從步驟 S105前進至 S106，播放器控制模組 212 會從碟片 101讀入在步驟 S105所認定的剪輯資訊檔案，亦即，從碟片 101的 "VIDEO"目錄內的 "CLIP"目錄來讀入3個剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP"。

在此，步驟 S106之剪輯資訊檔案的讀入，只要最初再生之 PlayList()的 PlayItem 的剪輯資訊檔案即夠充分，但在本實施形態中，如上述，先讀出所有 PlayList()的 PlayItem()的剪輯資訊檔案。

步驟 S106的處理後，前進至步驟 S107，播放器控制模組 212會判定在步驟 S105所認定之剪輯資訊檔案的讀入是否成功，且會判定對應於該讀入的剪輯資訊檔案之剪輯串流檔案是否存在於碟片 101。亦即，在步驟 S107中判定剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP" 的讀入成功，且分別與該剪輯資訊檔案 "00001.CLP"，

"00002.CLP"，"00003.CLP"僅檔案名的副檔名相異的剪輯串流檔案"00001.PS"，"00002.PS"，"00003.PS"是否存在於碟片101的"VIDEO"目錄之下的"STREAM"目錄。

在步驟 S107中，若判定在步驟 S105所認定之剪輯資訊檔案的讀入失敗，或判定對應於剪輯資訊檔案的剪輯串流檔案不存在於碟片101時，亦即，例如，按照"PLAYLIST.DAT"檔案而再生時所必要的剪輯資訊檔案或剪輯串流檔案未被記錄於碟片101時，視頻內容再生程式210會判斷成未對應於碟片101，前進至步驟 S102，進行上述的錯誤處理，終了再生前處理。

另一方面，在步驟 S107中，判定在步驟 S105所認定之剪輯資訊檔案的讀入成功，且對應於剪輯資訊檔案的剪輯串流檔案存在於碟片101時，播放器控制模組212會終了初期化處理，前進至步驟 S108。

在步驟 S108中，腳本控制模組211會開始"SCRIPT.DAT"檔案的解釋及執行。

例如，若現在腳本控制模組211執行"SCRIPT.DAT"檔案，而對播放器控制模組212指示第1個PlayList()(PlayList#0)的再生，則會進行第30圖的再生處理。

「再生處理」

亦即，第34圖是用以說明視頻內容再生程式210所進行的再生處理的流程圖。

「再生準備處理」

在再生處理中，播放器控制模組 212 會首先在步驟 S121 及 S122 中，進行由腳本控制模組 211 指示再生的 `PlayList()`，亦即第 1 個 `PlayList()(PlayList#0)` 的再生準備處理。

亦即，播放器控制模組 212 會在步驟 S121 中，確認第 1 個 `PlayList#0` 中所含的第 1 個 `PlayItem#0` 的 `IN_time` (第 8 圖)，然後前進至步驟 S122，調查依據第 1 個 `PlayList#0` 中所含的第 1 個 `PlayItem#0` 而再生的剪輯串流檔案 "00001.PS" 上之相當於該 `PlayItem#0` 的 `IN_time` 之開始再生的再生開始位置。

在此，當 `PlayItem()` 的 `IN_time` (第 8 圖) 為指示剪輯串流檔案的前頭時，只要從剪輯串流檔案的前頭讀出程式串流即可，但當 `IN_time` 為指示剪輯串流檔案的前頭以外時，必須檢測出對應於 `IN_time` 的位置，由此讀出程式串流。

具體而言，第 29 圖所示情況時，第 1 個 `PlayList#0` 中所含的第 1 個 `PlayItem#0` 的 `IN_time` 為 180,090。播放器控制模組 212 會從依據第 1 個 `PlayList#0` 中所含的第 1 個 `PlayItem#0` 而再生的剪輯串流檔案 "00001.CLP" 之第 31 圖所示的 `EP_map()` 來探出適於 `PlayItem#0` 的 `IN_time` 的 180,090 之再生開始位置。

亦即，播放器控制模組 212 是使用二元搜尋 (Binary

Search))等檢索例如表示被記述在 EP_map()之解碼開始可能點之 PTS_EP_start 中之滿足式 $PTS_EP_start \leq IN_time$ 之最大 PTS_EP_start。在此，之所以檢索 IN_time 以下之 PTS_EP_start，那是因為依據 IN_time 所表示之位置，不限於解碼開始可能點之故。

此情況，IN_time 是如上述般，為180,090。並且，依據第1個 PlayList#0中所含之第1個 PlayItem#0而再生之剪輯串流檔案“00001.CLP”之第31圖所示之 EP_map 中，記述有180,090來作為滿足式 $PTS_EP_start \leq IN_time$ 之最大 PTS_EP_start。因此，在播放器控制模組212中會從第31圖所示之 EP_map()來檢索形成該180,090的 PTS_EP_start。

並且，在播放器控制模組212中會讀出對應於該被檢索之 PTS_EP_start 之 RPN_EP_start 之305(區段)，依據該305之 RPN_EP_start 而被表示之剪輯流檔案“00001.PS”上的位置是當作再生開始位置而被決定。

一旦播放器控制模組212如以上所述決定再生開始位置，則會從步驟S122前進至S123，控制繪圖處理模組219，而使能夠顯示時間代碼。繪圖處理模組219會按照播放器控制模組212的控制，產生時間代碼，而輸出至視頻輸出模組220。藉此，開始時間代碼的顯示。

在此，於步驟S123中顯示開始的時間代碼是例如將 PlayList()的前頭換算成00:00:00(時間:分:秒)的值。另外，亦可與時間代碼或非時間代碼一起顯示章節或索引

的號碼。

「PlaylistMark()的解析處理」

在步驟 S123中時間代碼的顯示開始之後，前進至步驟 S124，播放器控制模組 212會進行解析處理，其係解析由腳本控制模組 211指示再生的 PlayList()，亦即記述於第 1個 PlayList()(PlayList#0)的 PlayListMark()(第 9圖)。

具體而言，播放器控制模組 212是在已經讀取之“PLAYLIST.DAT”檔案中之第 1個 PlayList#0之第 32圖上側所示之 PlayListMark()中，可由 number_of_PlayList_marks 成為 7之事，認定該 PlayList#0中所含之 Mark()的數量為 7。

並且，播放器控制模組 212會解析 PlayList#0中所含之第 32圖上側的 7個 Mark()，由該 ref_to_PlayItem_id 來認定從 7個 Mark()之中的第 1個到第 4個的 4個 Mark()為屬於 PlayList#0之第 1個 PlayItem()(PlayItem#0)。

然後，播放器控制模組 212會取出屬於 PlayList#0的第 1個 PlayItem#0的 4個 Mark()的 mark_time_stamp，當作要素數為 4的配列，交付給解碼控制模組 214。亦即，藉此，在第 32圖上側的 7個 Mark()中從第 1個到第 4個的 4個 Mark()的各個 mark_time_stamp 之 {180,090}，{5,580,090}，{10,980,090}，{16,380,090}的 4個時刻會從播放器控制模組 212交付給解碼控制模組 214。此刻，該等時刻的屬性為「標示處理」也從播放器控制模組 212來交付給解碼控

制模組 214。當計時部 214A 所計時的時刻與「標示處理」的屬性的時刻一致時，解碼控制模組 214 會將此訊息，及與「標示處理」的屬性的時刻一致的時刻，以及「標示處理」的屬性交付給播放器控制模組 212。

「再生之基本串流的決定處理」

其次，由步驟 S124 前進至 S125，播放器控制模組 212 會決定再生的基本串流。

亦即，播放器控制模組 212 是在自腳本控制模組 211 指示再生之 `PlayList()` 之第 1 號之 `PlayList#0` 之第 1 號之 `PlayList#0` (第 29 圖) 之 `Clip_Information_fime_name` 中記述有檔案名的第 30A 及 B 圖之剪輯資訊檔案 “00001.CLP” 中，由 `number_of_stream` 形成 4 之事，認定所對應之剪輯串流檔案 “00001.PS” 中 4 個基本串流會多重化。並且，播放器控制模組 212 是對該 4 個基本串流，依次調查第 30A 及 B 圖之剪輯資訊檔案 “00001.CLP” 之 `StaticInfo()` 之 `stream_id` 和所需之 `private_stream_id`，該 4 個基本串流會認定 1 個視頻串流，1 個 ATRAC 音頻串流及 2 個字幕串流。亦即，剪輯串流檔案 “00001.PS” 中被多重化之各屬性的基本串流的個數會被認定。

並且，被多重化於剪輯串流檔案之各屬性的基本串流之個數資訊，是被使用於再生中之基本串流之切換 (音頻切換、字幕切換等)。再者，字幕串流是有不存在於剪輯串流檔案中 (內容不包含有字幕) 之情形，字幕串流是否存

在之判斷，是使用「字幕串流」的屬性之基本串流的個數之資訊。

播放控制模組 212 是根據以上般之 `StaticInfo()` 之調查結果，選擇決定再生之基本串流，但此時被多重化於剪輯串流檔案“00001.PS”之 4 個基本串流中，因「視頻串流」和「音頻串流」的屬性之基本串流各僅有 1 個，故針對「視頻串流」和「音頻串流」的屬性之基本串流，無選擇餘地，將該 1 個視頻串流和音頻串流 (ATRAC 音頻串流) 當作再生之基本串流而被決定。

再者，針對「字幕串流」的屬性之基本串流，因在被多重化於剪輯串流檔案“00001.PS”的 4 個基本串流中存在有 2 個，故該 2 個字幕串流中之任一個之字幕串流，是被當作再生之基本串流而被選擇決定。在此，例如以 2 個字幕串流中之剪輯資訊檔案“00001.CLP”的出現順序來選擇最初的字幕串流。

在此，如上述般，為了認定被多重化於剪輯串流檔案“00001.PS”的 4 個基本串流的屬性和個數，必須各特定該 4 個基本串流，播放器控制模組 212 會依據 `stream_id` 和必要的 `private_stream_id` 來進行被多重化於剪輯串流檔案“00001.PS”的 4 個基本串流之特定。

亦即，播放器控制模組 212 是如第 30A 及 B 圖針對剪輯資訊檔案“00001.CLP”所說明般，以形成 0xE05 的 `stream_id` 來特定被多重化於剪輯串流檔案“00001.PS”的 4 個基本串流之中的「視頻串流」的屬性之基本串流的視

頻串流。

再者，播放器控制模組 212 是如第 30A 及 B 圖針對剪輯資訊檔案 “00001.CLP” 所說明般，以形成 0xBD 的 `stream_id` 及形成 0x00 的 `private_stream_id` 來特定被多重化於剪輯串流檔案 “00001.PS” 的 4 個基本串流中之「音頻串流」的屬性的基本串流之 ATRAC 音頻串流。

又，播放器控制模組 212 是如第 30 圖針對剪輯資訊檔案 “00001.CLP” 所說明般，各以成為 0xBD 的 `stream_id`、成為 0x80 的 `private_stream_id` 和成為 0xBD 的 `stream_id`、及成為 0x81 的 `private_stream_id` 來分別特定被多重化於剪輯串流檔案 “00001.PS” 的 4 個基本串流之「字幕串流」的屬性的基本串流之 2 個字幕串流。

如上述般，作為對應於剪輯串流檔案之剪輯資訊檔案的元資料而被記述之 `stream_id` 和 `private_stream_id` 的組合，可以特定被多重化於該剪輯串流檔案的基本串流。

在此，`stream_id` 和 `private_stream_id` 的組合是為了解擴張 MPEG2-System 的多重化而所設置之機構。為了特定基本串流而於元資料中使用該 `stream_id` 和 `private_stream_id` 的組合，可確實特定基本串流。再者，將來，擴張 `private_stream_id` 的意思，於增所對應之基本串流的個數或屬性時，因亦可原樣地使用現在的機構，故在擴張性上為優。

亦即，例如，BD(Blue ray Disc)規格中，因為在資料的規定中使用 MPEG 規格的傳送串流(Transport Stream)之

PID(Packet ID)，所以會被 MPEG2規格所拘束。再者，例如 DVD-Video 規格是定義類似於 `private_stream_id` 的 `sub_stream_id`，但是 `sub_stream_id` 爲了特定串流是不成爲可以記述在資料庫上，因只記述於記載8個或32個的串流資訊的固定區域(例如參照 VI4-49、Table4.2.1-2(VTS_AST_ATRT)或 VI4-52、Table4.2.1-3(VTS_SPST_STRT)等)，故缺乏擴張性。

相對的，`stream_id` 和 `private_stream_id` 的組合是可以記述元資料，例如在第12圖之剪輯資訊檔案 `Clip()`中僅記述可以以 `number_of_stream` 表示之數，因此，可不依賴該數(但是，可以以 `number_of_stream` 表示之數的範圍)，由作爲被記述於剪輯資訊檔案 `Clip()`的元資料之 `stream_id` 和 `private_stream_id` 的組合來特定被多重化於剪輯串流檔案的基本串流。

並且，本實施形態中，`stream_id` 和 `private_stream_id` 的組合是在第12圖之剪輯資訊檔案中，除了使用於特定被多重化於所對應的剪輯串流檔案之基本串流以外，亦可使用於例如作爲第9圖的 `PlayListMark()`中之 `entry_ES_stream_id` 和 `entry_ES_private_stream_id` 的組合，特定對 `Mark()`賦予關聯之基本串流。並且，`stream_id` 和 `private_stream_id` 的組合，其他亦在如第16圖之 `EP_map()`中，使用於特定記述可解碼開始點之基本串流。

「輸出屬性之控制處理」

之後，由步驟 S125前進至 S126，播放器控制模組 212 會執行再生對象之基本串流，即是決定以步驟 S125再生的基本串流之輸出屬性之控制處理。

具體而言，播放器控制模組 212首先各針對再生對象之基本串流，即是決定在步驟 S125再生之視頻串流、ATRAC 音頻串流、字幕串流，調查表示記述輸出屬性之 DynamicInfo()(第 15圖)之數量之 number_of_DynamicInfo(第 12圖)。

在此，此時，再生之視頻串流、ATRAC 音頻串流、字幕串流為被多重化於剪輯串流檔案“00001.PS”之基本串流，該些 number_of_DynamicInfo 是如第 30圖 A 及 B 之“00001.CLP”說明般，任一者皆為 0。如此一來，針對所有再生對象之基本串流，number_of_DynamicInfo 為 0之時，播放器控制模組 212，作為再生對象的基本串流的輸出屬性的控制處理不會特別進行處理。

另外，有關再生對象的基本串流的 number_of_DynamicInfo 不為 0時，有關作為該基本串流的輸出屬性的控制而進行的處理會在往後敘述。

「再生開始之準備處理」

步驟 S126處理後，前進至步驟 S127，播放器控制模組 212會進行再生對象之基本串流之再生開始之準備處理。

亦即，播放器控制模組 212 會對內容資料供給模組 213，給予再生對象之基本串流被多重化之剪輯串流檔案“00001.PS”之檔案名，和被記述在步驟 S122 決定之再生開始位置的 `EP_map()` 之 `RPN_EP_start(=305)`。

並且，播放器控制模組 212 於被儲存於再生對象之基本串流被多重化之剪輯串流檔案“00001.PS”中之程式串流，開始被供給至緩衝器控制模組 215 之前，使緩衝器控制模組 215 予以初期化。

具體而言，緩衝器控制模組 215 (第 5 圖) 中，被記憶於資料前頭記憶部 231 之資料前頭指標、被記憶於資料寫入指標記憶部 232 之資料寫入指標、被記憶於視頻讀出指標記憶部 241 之視頻讀出指標、被記憶於音頻讀出指標記憶部 251 之音頻讀出指標、被記憶於字幕讀出指標記憶部 262 之字幕讀出指標是被代入相同值。

依此，被記憶於資料前頭指標記憶部 231 之資料前頭指標，和被記憶於資料寫入指標記憶部 232 之資料寫入指標，是指緩衝器控制模組 215 之緩衝器 A 之相同位置。該是表示緩衝器 215A 無儲存有效資料之狀態。

並且，播放器控制模組 212 是將當作用以特定再生對象之基本串流之識別資訊的 `stream_id`，還有因其所需並將 `private_stream_id` 供給至緩衝器控制模組 215。

亦即，如上述般，再生對象之基本串流中之「視頻串流」之屬性之視頻串流是藉由成為 0xE0 之 `stream_id` 而被特定，「音頻串流」之屬性之 ATRAC 音頻串流是藉由成

為 0xBD 之 stream_id 及成為 0x00 之 private_stream_id 而被特定，「字幕串流」之屬性之字幕串流是藉由成為 0xBD 之 stream_id 及成為 0x80 之 private_stream_id 而被特定。播放器控制模組 212 是將該些 stream_id 和 private_stream_id 供給至緩衝器控制模組 215。

緩衝器控制模組 215(第 5 圖)中，視頻讀出功能部 233 會使來自播放器控制模組 212 之有關視頻串流之成為 0xE0 的 stream_id 記憶於 stream_id 暫存器 242。再者，音頻讀出功能部 234 是各使來自播放器控制模組 212 之成為 0xBD 的 stream_id，和成為 0x00 的 private_stream_id 記憶於 stream_id 暫存器 252 和 private_stream_id 暫存器 253。並且，字幕讀出功能部 235 是各使來自播放器控制模組 212 之成為 0xBD 的 stream_id，和成為 0x80 的 private_stream_id 記憶於 stream_id 暫存器 263 和 private_stream_id 暫存器 264。

並且，播放器模組 212 是為了日後處理記憶供給至緩衝器控制模組 215 之再生對象之基本串流之 stream_id 和 private_stream_id。播放器控制模組 212 是於後述要求串流切換之訊息發生時，或是在後述標記處理中為了特定現在再生中之串流，使用該些 stream_id 或 private_id。

播放器控制模組 212 會將作為緩衝器控制模組 215(第 5 圖)的初期化，且對應於再生對象的基本串流被多重化的剪輯串流之值的字幕讀出功能旗標設定於字幕讀出功能旗標記憶部 261。

亦即，此時，因為在再生對象之串流被多重化之剪輯串流檔案“00001.PS”中含有字幕串流，所以為了使字幕讀出功能部235發揮功能，值為1之字幕讀出功能旗標會被設定於字幕讀出功能旗標記憶部261。並且，在再生對象之基本串流被多重化之剪輯串流檔案中不含字幕串流時，在字幕讀出功能旗標記憶部261中設定有值為0之字幕讀出功能旗標。此時，字幕讀出功能部235尤其不執行處理。

再者，播放器控制模組212會對解碼控制模組214賦予從腳本控制模組211指示再生之第1個 PlayItem#0(第29圖)的 IN_time 之180,090，和 OUT_time 之27,180,090。在解碼控制模組214中，IN_time 是使用於依據 PlayItem()而再生之剪輯的解碼開始的控制，OUT_time 是使用於該剪輯的解碼終了及後述 PlayItem 掉換的控制。

並且，播放器控制模組212會將對繪圖處理模組219之字幕串流的顯示方式之指示予以初期化。亦即，播放器控制模組212會控制繪圖處理模組219，而使字幕串流的顯示方式成為例如預設的顯示方式。

「資料讀入開始」

然後，從步驟 S127前進至 S128，播放器控制模組212會控制內容資料供給模組213，藉此，內容資料供給模組213會使用操作系統201的功能來讀出儲存有再生對象的基本串流會被多重化的程式串流之剪輯串流檔案。亦即，內容資料供給模組213會指定位於碟片101(第6圖)的"VIDEO"

目錄之下的 "STREAM" 目錄的剪輯串流檔案 "00001.PS"，且指定在步驟 S122 所決定的再生開始位置之 305 區段，而對操作系統 201 要求檔案讀出。並且，內容資料供給模組 213 會以能夠將從碟片 101 讀出後的資料予以供給至緩衝控制模組 215 之方式來指定。

藉此，來自碟片 101 之儲存於剪輯串流檔案 "00001.PS" 的程式串流的讀出會開始，該程式串流會被供給至緩衝控制模組 215。

緩衝控制模組 215 (第 5 圖) 會將從碟片 101 讀出而供給的程式串流予以寫入緩衝器 215A 的資料寫入指標記憶部 232 的資料寫入指標所指示的位置，僅以寫入後的資料大小來增加資料寫入指標。

在此，並不特別限制，內容資料供給模組 213 若在緩衝器控制模組 215 之緩衝器 215A 中有空白時，則會從碟片 101 讀出資料，且供給至緩衝器控制模組 215 的緩衝器 215A 而使記憶。因此，在緩衝器 215A 中經常被儲存著充分的資料。

「解碼器控制開始」

如此一來，來自碟片 101 之資料的讀出會開始進行，一旦該資料被開始儲存於緩衝控制模組 215 的緩衝器 215A，則會從步驟 S128 前進至 S129，解碼控制模組 214 會控制視頻解碼器控制模組 216，音頻解碼器控制模組 217 及字幕解碼器控制模組 218，使來自緩衝器 215A 之資料的讀出開

始進行，亦即為解碼動作的前階段。

亦即，藉此，視頻解碼器控制模組 216 會對緩衝控制模組 215 (第 5 圖) 的視頻讀出功能部 233 要求資料，取得按照該要求來從緩衝控制模組 215 交付之記憶於緩衝器 215A 的 1 個視頻存取單元，及附加於該視頻存取單元的 PTS 及 DTS (以下稱為時間戳記 (time stamp))，及配置於解碼開始可能點之前的 `private_stream_2` 的 `PES_packet()` 中所記述的資訊 (以下亦稱為附加資訊) 之 `pic_struct_copy`，或 `au_ref_flag`，`AU_length` 等。另外，時間戳記是在每次視頻解碼器控制模組 216 取得視頻存取單元時，從視頻解碼器控制模組 216 交付給解碼控制模組 214。

在此為了以下的時刻更新而使用的 `pic_struct_copy` 是由視頻讀出功能部 233 所授與者，但亦可使用構文解析的結果所取得的位元串流中所含的 `pic_struct`。

另一方面，音頻解碼器控制模組 217 也會對緩衝控制模組 215 (第 5 圖) 的音頻讀出功能部 234 要求資料，取得按照該要求來從緩衝控制模組 215 交付之記憶於緩衝器 215A 的 1 個 (ATRAC) 音頻存取單元，及附加於該音頻存取單元的時間戳記 (PTS, DTS)。另外，時間戳記是在每次音頻解碼器控制模組 217 取得音頻存取單元時，從音頻解碼器控制模組 217 交付給解碼控制模組 214。

又，字幕解碼器控制模組 218 會對緩衝控制模組 215 (第 5 圖) 的字幕讀出功能部 235 要求資料，取得按照該要求來從緩衝控制模組 215 交付之記憶於緩衝器 215A 的 1 個字

幕存取單元，及附加於該字幕存取單元的時間戳記。另外，時間戳記是在每次字幕解碼器控制模組 218 取得字幕存取單元時，從字幕解碼器控制模組 218 交付給解碼控制模組 214。並且，在再生對象的基本串流中未存在字幕串流時，或在緩衝器 215A 中未記憶有字幕存取單元時，資料不會從緩衝控制模組 215 交付給字幕解碼器控制模組 218。

在此，視頻解碼器控制模組 216，音頻解碼器控制模組 217，及字幕解碼器控制模組 218 是在每次對緩衝控制模組 215 要求資料時，將對該資料的要求之結果交付給解碼控制模組 214。

又，有關從緩衝控制模組 215 來對視頻解碼器控制模組 216，音頻解碼器控制模組 217，及字幕解碼器控制模組 218 交付資料時，該資料從緩衝器 215A 讀出的詳細會在往後敘述。

「解碼開始」

如此一來，一旦視頻解碼器控制模組 216，音頻解碼器控制模組 217 及字幕解碼器控制模組 218 從緩衝控制模組 215 的緩衝器 215A 開始讀出資料，則會從步驟 S129 前進至 S130，開始進行該資料的解碼。

亦即，解碼控制模組 214 是根據在步驟 S127 從播放器控制模組 212 所給予之 `PlayList#0` 中所含的第 1 個 `PlayItem#0` 之 `IN_time` 的 180,090，還有如在步驟 S129 所說明般從視頻解碼器控制模組 216、音頻解碼器控制模組

217、字幕解碼器控制模組 218 被交給的時間戳記，爲了確保同步若有必要偏移時間，而對視頻解碼器控制模組 216、音頻解碼器控制模組 217、字幕解碼器控制模組 218 指示開始解碼。

在此，偏移用以確保同步的時間之解碼開始指令的方法，是例如被記載於日本專利第 3496725 號，簡單而言，有將分別從視頻解碼器控制模組 216、音頻解碼器控制模組 217、字幕解碼器控制模組 218 所交付的時間戳記中之最小值，當作藉由計時部 214A 所計時之時刻的初期值而予以設定，開始時刻之計時，且在藉由計時部 214A 來計時的時刻與時間戳記一致之時間點，命令解碼開始的方法。

視頻解碼器控制模組 216 會接受來自解碼控制模組 214 之解碼開始的指示，按照該指示，將從緩衝控制模組 215 (第 5 圖) 的視頻讀出功能部 233 所取得的 1 個視頻存取單元交付給視頻解碼器 116 (第 1 圖) 而使解碼。又，視頻解碼器控制模組 216 會將藉由視頻解碼器 116 的解碼結果所取得的視頻資料供應給繪圖處理模組 219。

以後，視頻解碼器控制模組 216 會以視頻解碼器 116 來依次解碼從緩衝控制模組 215 的視頻讀出功能部 233 所取得的各個視頻存取單元，且將該解碼的結果所取得的視頻資料供應給繪圖處理模組 219。

此刻，在視頻解碼器 116 內部，在解碼及輸出會發生視頻資料的順序替換(亂序(reordering))。例如第 35 圖所示，解碼的順序雖爲 I1，B0，P3，B2，P5，B4，但顯示的

順序卻形成 $B_0, I_1, B_2, P_3, B_4, P_5$ 。因應於此，在視頻解碼器 116 內設有用以事先將解碼後的畫像儲存的 decoded picture buffer。在第 35 圖中， I_n 是表示第 n 個的 I 圖像， B_n 是表示第 n 個的 B 圖像， P_n 是表示第 n 個的 P 圖像。

另一方面，音頻解碼器控制模組 217 也會接受來自解碼控制模組 214 之解碼開始的指示，按照該指示，將從緩衝控制模組 215 (第 5 圖) 的音頻讀出功能部 234 所取得的 1 個音頻存取單元交付給音頻解碼器 117 (第 1 圖) 而使解碼。又，音頻解碼器控制模組 217 會將藉由音頻解碼器 117 的解碼結果所取得的音頻資料供應給音頻輸出模組 221。

以後，音頻解碼器控制模組 217 會以音頻解碼器 117 來依次解碼從緩衝控制模組 215 的音頻讀出功能部 234 所取得的各個音頻存取單元，且將該解碼結果所取得的音頻資料供應給音頻輸出模組 221。

又，字幕解碼器控制模組 218 也會接受來自解碼控制模組 214 之解碼開始的指示，按照該指示，以內部所持有的字幕解碼軟體來解碼從緩衝控制模組 215 (第 5 圖) 的字幕讀出功能部 235 所取得的 1 個字幕存取單元，且將該解碼結果所取得的字幕資料 (字幕的畫像資料) 供應給繪圖處理模組 219。

以後，字幕解碼器控制模組 218 會以內部所持有的字幕解碼軟體來依次解碼從緩衝控制模組 215 的字幕讀出功能部 235 所取得的各個字幕存取單元，且將該解碼結果所

取得的字幕資料供應給繪圖處理模組 219。

「繪圖處理」

然後，從步驟 S130前進至 S131，繪圖處理模組 219會以從視頻解碼器控制模組 216所供給的視頻資料，甚至因應所需，以從字幕解碼器控制模組 218所供給的字幕資料為對象，進行繪圖處理。

亦即，繪圖處理模組 219會按照來自播放器控制模組 212的顯示方式的指示，首先對來自字幕解碼器控制模組 218的字幕資料進行擴大或縮小等的字幕處理。當自播放器控制模組 212無顯示方式的指示時，或有內定的顯示方式的指示時，繪圖處理模組 219會原封不動保存來自字幕解碼器控制模組 218的字幕資料。

又，繪圖處理模組 219會加算來自視頻解碼器控制模組 216的視頻資料，及來自字幕解碼器控制模組 218的字幕資料，或字幕處理後的字幕資料，取得字幕資料會被重疊於來自視頻解碼器控制模組 216的視頻資料之輸出視頻資料，然後供應給視頻輸出模組 220。

另外，從腳本控制模組 211或播放器控制模組 212接到例如選單，訊息，時間代碼，章節或索引的號碼等之資訊的顯示的指示時，繪圖處理模組 219會產生該資訊，且重疊於輸出視頻資料來供應給視頻輸出模組 220。

「輸出處理」

步驟 S131的處理後，前進至步驟 S132，視頻輸出模組 220會如在步驟 S131所說明那樣，使由繪圖處理模組 219所供給的輸出視頻資料依次記憶於 FIFO220A，且以事先決定的輸出速率來依次輸出記憶於該 FIFO220A 的輸出視頻資料。

只要 FIFO220A 的記憶容量(餘量)夠充裕，視頻輸出模組 220便會接受來自繪圖處理模組 219的輸出視頻資料，但不夠充裕時，會要求繪圖處理模組 219停止接受輸出視頻資料。藉此，繪圖處理模組 219會停止處理，且要求視頻解碼器控制模組 216及字幕解碼器控制模組 218停止處理。藉此，視頻解碼器控制模組 216及字幕解碼器控制模組 218會停止處理。

視頻輸出模組 220在要求繪圖處理模組 219停止接受輸出視頻資料之後，來自 FIFO220A 的輸出視頻資料的輸出會繼續進行，在 FIFO220A 夠充裕時，要求繪圖處理模組 219接受輸出視頻資料。此要求是與停止接受輸出視頻資料的要求同樣，從繪圖處理模組 219來交付給視頻解碼器控制模組 216及字幕解碼器控制模組 218。藉此，繪圖處理模組 219，以及視頻解碼器控制模組 216及字幕解碼器控制模組 218會使停止的處理再度開始。

另一方面，音頻輸出模組 221亦如步驟 S130所說明那樣，使自音頻解碼器控制模組 217供給的音頻資料依次記憶於 FIFO221A，且以事先決定的輸出速率(取樣頻率)來依次輸出記憶於該 FIFO221A 的音頻資料。

只要 FIFO221A 的記憶容量(餘量)夠充裕，音頻輸出模組 221 便會接受來自音頻解碼器控制模組 217 的音頻資料，但若不夠充裕時，會要求音頻解碼器控制模組 217 停止接受音頻資料。藉此，音頻解碼器控制模組 217 會停止處理。

音頻輸出模組 221 在要求音頻解碼器控制模組 217 停止接受音頻資料之後，來自 FIFO221A 的音頻資料的輸出會繼續進行，在 FIFO221A 夠充裕時，要求音頻解碼器控制模組 217 接受音頻資料。藉此，音頻解碼器控制模組 217 會使停止的處理再度開始。

如此一來，基本串流的解碼會隨著從視頻輸出模組 220 及音頻輸出模組 221 輸出資料而進行。

[視頻解碼器 116 的内部構造的說明]

第 36 圖是表示視頻解碼器 116 的内部構造。就此例而言，視頻解碼器 116 的内部是以視頻解碼引擎 116A 及 DPB(解碼圖像緩衝器)116B 來構成。DPB116B 的内部更被細分化，以 DPB116B-1 ~ DPB116B-n(往後無須特別區別時，僅稱為 DBP116B)來構成。又，如第 37 圖所示，DPB116B 的内部是被分成視頻緩衝器 301 及附加資訊緩衝器 302。

視頻解碼引擎 116A 是在視頻資料的解碼處理時，為了一時保存解碼中的視頻資料，或為了作為將來的參照畫像使用而保持等的用途，而使用 DPB116B 的視頻緩衝器

301。此刻，在附加資訊緩衝器302中記錄有應該對應於視頻緩衝器301中所保存的視頻資料，由視頻讀出功能部233所取得的附加資訊，或對存取單元進行構文解析所取得的參數(例如 `pic_struct` 等)。

其次，第1圖的碟片再生裝置在再生碟片101時的全體處理或動作流程是如第33圖及第34圖所說明那樣，但以下則是針對碟片再生裝置中進行碟片101的再生時之其他的處理或動作來進行說明。

[將時刻資訊交付解碼控制模組214]

以下，說明有關時鐘(計時部214A)的更新。視頻解碼器控制模組216是指示視頻解碼器116來解碼輸入後的視頻存取單元。在利用視頻解碼器116的解碼及亂序(reordering)處理之後，1幀(2場)分的視頻資料會被輸出至繪圖處理模組219，同時該當視頻資料的顯示時間標籤(PTS/DTS)及 `pic_struct` 資訊會從視頻解碼器控制模組216來交付解碼控制模組214。

當存取單元的 `pic_struct` 為1或2時，該等為1場分的存取單元，因此為了成為1幀分，而於進行2個分的存取單元的輸出之階段，先行場的 `pic_struct`、及先行場的存取單元中具有顯示時間標籤時，顯示時間標籤會從視頻解碼器控制模組216來交付解碼控制模組214。此先行場未具有顯示時間標籤時，交付無顯示時間標籤的資訊。如上述，由於不允許孤立場，因此在 `pic_struct` 為1或2的場之後，

放置 `pic_struct` 為 2 或 1 的場。在將此 2 個場集中於 1 個來進行處理時，以先行的場的顯示時間標籤作為代表值使用。

又，當存取單元的 `pic_struct` 為 0，3，4，5，6 時，在進行 1 存取單元的輸出的階段，在 `1pic_struct`、及該當存取單元中具有顯示時間標籤時，顯示時間標籤會從視頻解碼器控制模組 216 來交付解碼控制模組 214。在未具有顯示時間標籤的存取單元會被交付無顯示時間標籤的資訊。

解碼控制模組 214 是利用所受取的顯示時間標籤及 `pic_struct` 的資訊來更新計時部 214A。

以下，參照第 38 圖的流程圖來說明更新的方法。

解碼控制模組 214 係判定所受取的存取單元是否附加有顯示時間標籤(步驟 S141)。例如，在存取單元中附加有顯示時間標籤時，解碼控制模組 214 係將顯示時間標籤(PTS)的值設定於計時部 214A(步驟 S142)。如上述，由於解碼開始後必存在顯示時間標籤，因此即使是初次也不會發生問題。在存取單元中未被附加顯示時間標籤時，於現在時刻加算前次的 `pic_struct` 所決定的值(步驟 S144)。然後，為了下次的處理，保存這次的 `pic_struct`，而終了(步驟 S143)。

在 `pic_struct` 所決定的值，如第 39 圖所示，當所保存的 `pic_struct` 為 0，3 或 4 時，計時部 214A 會加算 2 場時間分的時間。又，當 `pic_struct` 為 5 或 6 時，計時部 214A 會加算 3 場時間分的時間。又，當所保存的 `pic_struct` 為 1 或 2 時，計時部 214A 會加算 2 場時間分的時間。

藉由如此的時刻變更處理，計時部 214A 所示的時刻值是表示由視頻解碼器控制模組 216 至繪圖處理模組 219 的輸出終了 (1 幀分) 的存取單元的顯示開始時刻。亦即，該當視頻資料為具有顯示時間標籤時，PTS 會被代入。又，當未具有顯示時間標籤時，以顯示順序來加算之前的視頻資料的顯示間隔。

此例中，視頻編碼方式為使用 AVC，但例如在 MPEG2-Video 中亦可藉由使用 repeat_first_field 來得知存取單元的顯示 duration。

另外，如上述，此情況同樣是在 FIFO220A 的記憶容量不夠充裕時，來自視頻解碼器控制模組 216 的視頻資料的輸出會停止。此時，計時部 214A 的更新也會自動停止。並且，一旦對 FIFO220A 之視頻資料的更新再開始，則計時部 214A 的更新也會自動再開始。

亦即，根據使用者的指示，再生模式遷移至停止狀態時，視頻輸出模組 220 的更新會停止，藉此連帶著視頻解碼器控制模組 216 會停止，且連結的時鐘 (計時部 214A) 也連帶停止。又，自停止狀態恢復至通常再生狀態時，視頻輸出模組的更新會被許可，藉此連帶著視頻解碼器控制模組 216 的動作及視頻資料的輸出會再開始，且連結的時鐘 (計時部 214A) 也連帶著更新再開始。

如此的動作對於慢再生而言亦可期待。亦即，慢再生是以短週期來交替重複進行停止及通常再生的狀態，此刻時鐘 (計時部 214A) 是與視頻輸出同步更新。

另外，此例雖是將計時部 214A 的更新與來自視頻解碼器控制模組 216 的視頻資料輸出同步進行，但當視頻解碼器控制模組 216 以後的延遲，在此為繪圖處理模組 219 及視頻輸出模組 220 所產生的延遲大時，實際上對使用者提供的視頻資料與時鐘 (計時部 214A) 的關係會有偏移的可能性。如此的情況，可藉由採用將時鐘 (計時部 214A) 的更新與來自視頻輸出模組 220 的視頻資料輸出同步進行之方式來防止。

具體而言，在繪圖處理模組 219 與視頻輸出模組 220 及 FIFO 220A 的視頻資料的處理部分，追加參照第 37 圖來說明的附加資訊緩衝器 302，至輸出為止，經常將視頻資料及附加資訊當作一組來進行處理。又，來自視頻輸出模組 220 之視頻資料的輸出時，所對應的附加資訊會被交付解碼控制模組 214。解碼控制模組是藉由上述的算法來更新時鐘 (計時部 214A)。

藉由取如此的方法，不論視頻解碼器以後的延遲多少，可使所被顯示的視頻資料與時鐘 (計時部 214A) 同步。

結果，再生串流資料的裝置，即使為未具備單獨地計數的時鐘之狀態，還是能夠正確地再生串流資料，可使 CPU 112 等的處理負荷降低。

[PlayItem 掉換]

如第 33 圖及第 34 圖所說明那樣，第 29 圖之第 1 個 PlayList#0 的第 1 個 PlayItem#0 的再生會開始，但若該第 1

個 PlayItem#0 的再生終了，則第 2 個 PlayItem#1 的再生會開始。亦即，進行從 PlayItem#0 掉換成 PlayItem#1 的 PlayItem 掉換。

其次，參照第 40 圖的流程圖來說明有關此 PlayItem 掉換的處理。

如第 33 圖及第 34 圖所說明那樣，若第 29 圖之 PlayList#0 的第 1 個 PlayItem#0 (的剪輯) 的再生開始，則解碼控制模組 214 (第 2 圖 A 及第 2 圖 B) 會在該第 1 個 PlayItem#0 的再生所進行的期間，持續確認內藏的計時部 214A 所計時的時刻。

「PlayItem#0 的再生終了」

若計時部 214A 所計時的時刻與第 34 圖的步驟 127 中從播放器控制模組 212 所被賦予的第 1 個 PlayItem#0 的 OUT_time 之 27,180,090 (第 29 圖) 相等，則解碼控制模組 214 會在步驟 S151 中，進行解碼中斷控制，終了 PlayItem#0 的再生。

當計時部 214A 不是在 90kHz 變化者時，亦即，按照視頻資料的輸出來更新時刻的方式時，計時部 214A 所計時的時刻嚴格來講有時不會與 PlayItem#0 的 OUT_time 相等。此刻，以 PlayItem#0 的 OUT_time 的時刻與計時部 214A 所計時的時刻形成附近的值之時序來進行解碼中斷控制，終了 PlayItem#0 的再生。另外，有關該處理是參照第 51 圖，第 52 圖來詳細後述。

亦即，解碼控制模組 214 會操作視頻解碼器控制模組 216，音頻解碼器控制模組 217，字幕解碼器控制模組 218，而使解碼動作停止。又，解碼控制模組 214 會控制視頻輸出模組 220，使現在輸出中的輸出視頻資料繼續輸出。

又，解碼控制模組 214 會將第 1 個 `PlayItem#0` 的再生終了之訊息交付給播放器控制模組 212。

「`PlayItem#1` 的再生開始」

播放器控制模組 212，如上述，在第 33 圖的步驟 S105 中，認定在第 1 個 `PlayList #0` 中存在第 1 個 `PlayItem#0` 及第 2 個 `PlayItem#1`，一旦從解碼控制模組 214 交付第 1 個 `PlayItem#0` 的再生終了之訊息，則會從步驟 S151 前進至 S152，與上述第 1 個 `PlayItem#0` 時同樣的開始第 2 個 `PlayItem#1` 的再生。

亦即，若概說第 2 個 `PlayItem#1` 的再生程序，則首先，播放器控制模組 212 會與第 34 圖的步驟 S122 時同樣的，針對第 2 個 `PlayItem#1` 來決定記述於 `EP_map()` 的 `RPN_EP_start` 的其中之一作為再生開始位置。

並且，在播放器控制模組 212 中進行如第 34 圖的步驟 S124 所說明那樣，屬於第 2 個 `PlayItem#1` 之 `Mark()` 的認定，及如第 34 圖的步驟 S125 所說明那樣，依據 `PlayItem#1` 而再生的剪輯串流檔案 "00002.PS" 中被多重化的各屬性的基本串流的個數的認定，以及再生的基本串流的決定等。

而且，播放器控制模組 212 是進行與第 34 圖的步驟

S127時同樣的處理。

亦即，播放器控制模組212會對內容資料供給模組213賦予作為再生開始位置而決定的EP_map()的RPN_EP_start，及再生對象的基本串流所被多重化的剪輯串流檔案的檔案名，亦即目前的情況，對應於第2個PlayItem#1(第29圖)的Clip_Information_file_name中所記述的"00002.CLP"之剪輯串流檔案"00002.PS"的檔案名。

又，播放器控制模組212會在再生對象的基本串流被多重化的剪輯串流檔案"00002.PS"中所儲存的程式串流開始供應給緩衝控制模組215之前使緩衝控制模組215初期化。

亦即，藉此，在緩衝控制模組215(第5圖)中，記憶於資料前頭指標記憶部231的資料前頭指標，記憶於資料寫入指標記憶部232的資料寫入指標，記憶於視頻讀出指標記憶部241的視頻讀出指標，記憶於音頻讀出指標記憶部251的音頻讀出指標，及記憶於字幕讀出指標記憶部262的字幕讀出指標中代入相同的值。

又，播放器控制模組212會將用以識別再生對象的基本串流之識別資訊，亦即stream_id供應給緩衝控制模組215，甚至因應所需，將private_stream_id緩衝控制模組215。

在緩衝控制模組215(第5圖)中，視頻讀出功能部233會使來自播放器控制模組212之有關再生對象的基本串流中的視頻串流的stream_id記憶於stream_id暫存器242。

又，音頻讀出功能部 234 會使來自播放器控制模組 212 之再生對象的基本串流中的音頻串流的 `stream_id` 及 `private_stream_id` 分別記憶於 `stream_id` 暫存器 252 及 `private_stream_id` 暫存器 253。

又，由於在目前形成再生對象的基本串流被多重化的剪輯串流檔案 "00002.PS" 中含有字幕串流，因此會從播放器控制模組 212 對字幕讀出功能部 235 供給再生對象的基本串流中的字幕串流的 `stream_id` 及 `private_stream_id`，且字幕讀出功能部 235 會使該 `stream_id` 及 `private_stream_id` 分別記憶於 `stream_id` 暫存器 263 及 `private_stream_id` 暫存器 264。

又，播放器控制模組 212 會將緩衝控制模組 215 (第 5 圖) 的初期化，且對應於再生對象的基本串流被多重化的剪輯串流檔案之值的字幕讀出功能旗標設定於字幕讀出功能旗標記憶部 261。

亦即，目前的情況，因為在再生對象的基本串流被多重化的剪輯串流檔案 "00002.PS" 中含有字幕串流，所以為了使字幕讀出功能部 235 作用，而值為 1 的字幕讀出功能旗標會被設定於字幕讀出功能旗標記憶部 261。

又，播放器控制模組 212 會對解碼控制模組 214 賦予所欲再生的第 2 個 `PlayItem#1` (第 29 圖) 的 `IN_time` 之 90,000，及 `OUT_time` 之 27,090,000。

又，播放器控制模組 212 會使對繪圖處理模組 219 之字幕串流的顯示方式的指示初期化。亦即，播放器控制模組

212會以字幕串流的顯示方式能夠成為內定的顯示方式之方式來控制繪圖處理模組219。

另外，有關再生對象的字幕串流方面，當 `configurable_flag`(第14圖)為形成許可變更顯示方式的1時，由播放器控制模組212對繪圖處理模組219之字幕串流的顯示方式的指示亦可為保持現在的顯示方式不動。

以下，第2個 `PlayItem#1`的再生是與第1個 `PlayItem#0`的再生同樣進行。而且，解碼控制模組214會在第2個 `PlayItem#1`的再生進行之間，持續確認內藏的計時部214A所計時的時刻，若計時部214A所計時的時刻與步驟S152(第40圖)中從播放器控制模組212所被賦予的第2個 `PlayItem#1`的 `OUT_time` 之27,090,000(第29圖)相等，則會進行與步驟S151的情況同樣的解碼中斷控制，終了 `PlayItem#1`的再生。另外，如上述，計時部214A所計時的時刻嚴格來講有時不會與 `PlayItem#0`的 `OUT_time` 相等。此刻，以 `PlayItem#0`的 `OUT_time` 的時刻與計時部214A所計時的時刻形成附近的值之時序來進行解碼中斷控制，終了 `PlayItem#0`的再生。另外，有關該處理是參照第51圖，第52圖來詳細後述。

[時間代碼的顯示]

其次，如上述，在第34圖的步驟S123中，時間代碼的顯示會開始，但此時間代碼的顯示會依次被更新。

在此，參照第41圖的流程圖來說明有關時間代碼的顯

示處理。

解碼控制模組 214(第 2 圖 A 及第 2 圖 B)，若藉由其內藏的計時部 214A 而計時 1 秒，則會在步驟 S171 中，將經過 1 秒的訊息，以及藉由該計時部 214A 而計時的現在時刻供應給播放器控制模組 212，前進至步驟 S172。在步驟 S172 中，播放器控制模組 212 會接受來自解碼控制模組 214 的訊息及現在時刻，且將該現在時刻換算成時間代碼，然後前進至步驟 S173。

在步驟 S173 中，播放器控制模組 212 會以能夠顯示在步驟 S172 中所取得的時間代碼之方式來控制繪圖處理模組 219，回到步驟 S171。

藉此，時間代碼會被每秒更新。另外，時間代碼的更新間隔並非限於 1 秒。

[串流切換]

其次，在依據第 1 個 `PlayItem#0`(構成第 29 圖所說明的第 1 個 `PlayList#0`)所再生的剪輯串流檔案 "00001.PS"，或依據第 2 個 `PlayItem#1`所再生的剪輯串流檔案 "00002.PS" 中，如第 30 圖 A 及第 30 圖 B 所說明那樣，2 個字幕串流會被多重化。

如此，在剪輯串流檔案中複數個同一屬性的基本串流被多重化時，可將再生對象的基本串流從該複數個同一屬性的基本串流的其中 1 個切換成其他的 1 個，亦即進行串流切換。

在此，參照第42圖的流程圖來說明有關串流切換的處理。

串流切換的要求是例如在"SCRIPT.DAT"檔案(第6圖)中記述有串流切換的指示作為腳本程式時，腳本控制模組211會藉由執行該腳本程式，或藉由使用者操作遙控器來賦予播放器控制模組212。

亦即，一旦執行記述有串流切換的指示之腳本程式，則腳本控制模組211會將要求串流切換的訊息供應給播放器控制模組212。又，若輸入介面115藉由使用者操作遙控器來從遙控器接受指示串流切換的信號，則會將要求串流切換的訊息供應給播放器控制模組212。

例如，若現在對播放器控制模組212供給要求切換字幕串流的訊息，亦即字幕串流切換的訊息，則播放器控制模組212會在步驟S191中，檢查在第34圖的步驟S125中所被進行之再生對象的基本串流的決定時認定後的字幕串流的個數。

當檢查字幕串流的個數之結果，其個數為1以下時，播放器控制模組212會無視字幕串流切換的訊息，因此以後的步驟S192～S194之處理不會被進行。

另一方面，當字幕串流的個數為2以上時，依次前進至步驟S192～S194，所再生的字幕串流會從現在再生的字幕串流切換成其他字幕串流。

亦即，在步驟S192中，播放器控制模組212會在剪輯資訊檔案上特定現在再生中的字幕串流。具體而言，例如

若現在依據第2個 PlayItem#1(構成第29圖所說明的第1個 PlayList#0)來再生剪輯串流檔案"00002.PS"中被多重化之 stream_id 為 0xBD， private_stream_id 為 0x80的字幕串流，則在步驟 S192中，現在再生中的字幕串流會被特定為剪輯串流檔案"00002.PS"中被多重化的2個字幕串流之中在第30圖 A 及第30圖 B 的剪輯資訊檔案"00002.CLP"上為第3個字幕串流的 stream#2。

然後，前進至步驟 S193，播放器控制模組 212會將步驟 S192所特定的字幕串流之剪輯資訊上下一個字幕串流，當作下一個再生的字幕串流而予以特定。在第30A 及 B 圖中，於剪輯資訊檔案“00002.CLP”上，因為第3個字幕串流 stream#2之下一個字幕串流為第4個字幕串流 stream#3，所以在步驟 S193中，該第4個字幕串流 stream#3會接著當作再生字幕串流而被認定。

並且，現在再生中之字幕串流被認定為剪輯串流檔案“00002.PS”中被多重化的2個字幕串流之中，第30A 及 B 圖的剪輯資訊檔案“00002.CLP”上第4個字幕串流的 stream#3之時，例如第3個字幕串流 stream#2會被當作下一個再生之字幕串流而被認定。

之後，前進至步驟 S194，播放器控制模組 212會對緩衝器控制模組 215(第5圖)的字幕讀出功能部 235賦予在步驟 S193認定的下一個再生之字幕串流之 stream_id 和 private_stream_id，且指示該 stream_id 和 private_stream_id 使用在下次的字幕存取單元之緩衝器

215A 的讀出。

緩衝器控制模組 215(第 5 圖)的字幕讀出功能部 35 是在步驟 S194 將從播放器控制模組 212 所給予之 `stream_id` 和 `private_stream_id` 各重新設定於 `stream_id` 暫存器 263 和 `private_stream_id` 暫存器 264，且來自下次以後的緩衝器 215A 之讀出是以藉由分別被重新設定於該 `stream_id` 暫存器 263 和 `private_stream_id` 暫存器 264 之 `stream_id` 和 `private_stream_id` 所特定的字幕存取單元作為對象而被執行。

如上述般，再生之字幕串流會從現在再生之字幕串流來切換成其他字幕串流。

[緩衝控制模組 215 的處理]

其次，參照第 43 圖～第 47 圖來說明有關緩衝控制模組 215(第 5 圖)的處理，亦即往緩衝器 215A 的資料寫入，及來自緩衝器 215A 的資料讀出。

緩衝控制模組 215，如第 5 圖所說明，具有用以對緩衝器 215A 進行資料的讀寫之 5 個的指標。

亦即，如第 43 圖及第 44 圖所示，緩衝控制模組 215 具有記憶於資料前頭指標記憶部 231 的資料前頭指標，記憶於資料寫入指標記憶部 232 的資料寫入指標，記憶於視頻讀出指標記憶部 241 的視頻讀出指標，記憶於音頻讀出指標記憶部 251 的音頻讀出指標，及記憶於字幕讀出指標記憶部 262 的字幕讀出指標。

另外，在第43圖及第44圖中，第5圖之視頻讀出功能部233的 `stream_id` 暫存器242及 `au_information()`暫存器243，音頻讀出功能部234的 `stream_id` 暫存器252及 `private_stream_id` 暫存器253，以及字幕讀出功能部235的字幕讀出功能旗標記憶部261，`stream_id` 暫存器263，及 `private_stream_id` 暫存器264的圖示會省略。

記憶於資料前頭指標記憶部231的資料前頭指標是表示留在緩衝器215A中最舊的資料(為必須讀出的資料，未被讀出的資料中最舊資料)的位置。記憶於資料寫入指標記憶部232的資料寫入指標是表示對緩衝器215A之資料的寫入位置，此位置是在緩衝器215A寫入最新資料的位置。

記憶於視頻讀出指標記憶部241的視頻讀出指標是表示從緩衝器215A讀出的視頻串流的位置。又，記憶於音頻讀出指標記憶部251的音頻讀出指標是表示從緩衝器215A讀出的音頻串流的位置，記憶於字幕讀出指標記憶部262的字幕讀出指標是表示從緩衝器215A讀出的字幕串流的位置。

另外，如第5圖所說明那樣，資料前頭指標，資料寫入指標，視頻讀出指標，音頻讀出指標，及字幕讀出指標皆順時鐘移動於緩衝器215A。

又，本實施形態中，如第44圖所示，資料前頭指標是以能夠指出與視頻讀出指標，音頻讀出指標，或字幕讀出指標之中指出最舊資料的位置者同一位置之方式來經常被

更新者。在此，第44圖中，視頻讀出指標，音頻讀出指標，或字幕讀出指標之中的音頻讀出指標為指出最舊資料的位置，資料前頭指標是與該音頻讀出指標一致。

在以上那樣的資料前頭指標，資料寫入指標，視頻讀出指標，音頻讀出指標，及字幕讀出指標之緩衝控制模組215中，若從碟片101讀出新的資料，且寫入緩衝器215A，則資料寫入指標會以能夠指出緊接著該被寫入的新資料之後的位置之方式來順時鐘被更新。

又，視頻讀出指標、音頻讀出指標或是字幕讀出指標是一旦自緩衝器215A讀出視頻串流、音頻串流或字幕串流，則會僅對應於該讀出量之份量，分別順時鐘地被更新。在此所謂對應於讀出量之份量是指組合實際對應於讀出後的視頻、音頻、字幕的資料之部分，及含於讀出後的資料之間，在讀出時進行跳讀的其他串流的資料之部分。

又，若視頻讀出指標，音頻讀出指標，或字幕讀出指標被更新，則資料前頭指標會以能夠指出與該視頻讀出指標，音頻讀出指標，或字幕讀出指標之中指出最舊資料的位置者同一位置之方式來更新。

在此，有關對緩衝器215A之資料的寫入方面，緩衝控制模組215是以資料寫入指標不會追過資料前頭指標之方式來控制對緩衝器215A之資料的寫入。

亦即，只要不發生資料寫入指標追過資料前頭指標，在緩衝控制模組215中，從碟片101讀出的資料便會被寫入資料寫入指標所指的緩衝器215A的位置，資料寫入指標

會被更新。另一方面，若發生資料寫入指標越過資料前頭指標，則緩衝控制模組 215 會對內容資料供給模組 213 要求停止從碟片 101 讀出資料，且停止對緩衝器 215A 寫入資料。藉此，可防止緩衝器 215A 的溢出(overflow)。

如以上所述，從碟片 101 讀出的資料對緩衝器 215A 的寫入是僅以資料前頭指標與資料寫入指標的 2 個指標的位置關係來控制。

另一方面，有關來自緩衝器 215A 的資料之讀出，緩衝控制模組 215 是以視頻讀出指標，音頻讀出指標，及字幕讀出指標，進而資料前頭指標不會追過資料寫入指標之方式，控制來自緩衝器 215A 的資料之讀出。

亦即，只要不發生視頻讀出指標，音頻讀出指標，或字幕讀出指標追過資料寫入指標，便可在緩衝控制模組 215 中，按照來自視頻解碼器控制模組 216，音頻解碼器控制模組 217，或字幕解碼器控制模組 218 的要求，由視頻讀出指標，音頻讀出指標，或字幕讀出指標所指的緩衝器 215A 的位置來讀出資料，更新視頻讀出指標，音頻讀出指標，或字幕讀出指標，且因應所需，更新資料前頭指標。另一方，若發生視頻讀出指標，音頻讀出指標，或字幕讀出指標追過資料寫入指標，則在緩衝控制模組 215 中，來自視頻解碼器控制模組 216，音頻解碼器控制模組 217，或字幕解碼器控制模組 218 的要求會例如被凍結，等待至準備有充分的資料為止。藉此，可防止緩衝器 215A 的下溢(underflow)。

由以上所述，在緩衝器 215A 中記憶有從資料前頭指標所指的位置到順時鐘資料寫入指標所指的位置的範圍(在第 43 圖及第 44 圖中陰影的部份)中應供給至視頻解碼器控制模組 216，音頻解碼器控制模組 217，及字幕解碼器控制模組 218 的資料，且在該範圍內存在視頻讀出指標，音頻讀出指標，及字幕讀出指標。

另外，上述情況，雖資料前頭指標是以能夠指出視頻讀出指標，音頻讀出指標，或字幕讀出指標所指的位置中最舊資料的位置之方式來更新，但其他，資料前頭指標的更新，例如亦可以能夠從該最舊資料的位置指出所定時間(例如，1 秒)量的過去資料的位置之方式來進行。

亦即，一般可預測大多視頻讀出指標，音頻讀出指標，或字幕讀出指標之中的視頻讀出指標或音頻讀出指標會指出最舊資料的位置。

因此，以能夠從視頻讀出指標或音頻讀出指標所指的最舊資料的位置來指出例如 1 秒的過去資料的位置之方式更新資料前頭指標時，如第 34 圖所示，可使視頻讀出指標或音頻讀出指標所指的最舊資料的位置開始過去 1 秒量的資料留在緩衝器 215A 中。在此，於第 43 圖中，音頻讀出指標會指出最舊資料的位置，資料前頭指標會指出該位置開始過去 1 秒量的資料的位置。

如以上所述，以能夠指出過去 1 秒量的資料的位置之方式來更新資料前頭指標，藉此可使碟片再生裝置的應答性提升。

亦即，如第44圖所示，以能夠指出音頻讀出指標所指的最舊資料的位置之方式來更新資料前頭指標時，例如有往逆(reverse)方向的特殊再生被指示時，必須從碟片101再度讀出自緩衝器215A完成讀出後的資料，因此從指示特殊再生到該特殊再生可能為止需要花費某程度的時間。

相對的，如第43圖所示，以能夠指出音頻讀出指標所指的最舊資料的位置開始過去1秒量的資料的位置之方式來更新資料前頭指標時，當有往逆方向的特殊再生被指示時，只要開始進行該特殊再生時所必要的資料為記憶於緩衝器215A的過去1秒量的資料，便可不必進行像上述那樣從碟片101再讀出資料，馬上可開始進行特殊再生。

另外，以能夠指出音頻讀出指標所指的最舊資料的位置開始過去1秒量的資料的位置之方式來更新資料前頭指標時，亦有可能開始進行特殊再生時所必要的資料未被記憶於緩衝器215A。此情況，開始進行特殊再生時所必要的資料會從碟片101再度讀出。

其次，說明有關來自緩衝器215A之視頻串流，音頻串流，字幕串流的各個讀出的詳細。

如第34圖的步驟S127所說明那樣，當剪輯串流檔案的再生被開始進行時，在緩衝控制模組215中，會以資料前頭指標，資料寫入指標，視頻讀出指標，音頻讀出指標，及字幕讀出指標能夠全體指出緩衝器215A上的相同位置之方式被初期化。

然後，一旦從碟片101讀出被儲存於剪輯串流檔案的

程式串流 (MPEG2-System Program Stream)，且供應給緩衝控制模組 215，則於緩衝控制模組 215 中，該程式串流會被記憶於緩衝器 215A 的資料寫入指標所指的位置，且資料寫入指標會被順時鐘更新。

並且，在緩衝控制模組 215(第 5 圖)中，視頻讀出功能部 233 會進行記憶於緩衝器 215A 之程式串流的構文解析，按照來自視頻解碼器控制模組 216 的要求，從記憶於緩衝器 215A 的程式串流來抽出視頻存取單元而讀出，且供應給視頻解碼器控制模組 216。

同樣的，音頻讀出功能部 234 也會進行記憶於緩衝器 215A 之程式串流的構文解析，按照來自音頻解碼器控制模組 217 的要求，從記憶於緩衝器 215A 的程式串流來抽出音頻存取單元而讀出，且供應給音頻解碼器控制模組 217。字幕讀出功能部 235 也會進行記憶於緩衝器 215A 之程式串流的構文解析，按照來自字幕解碼器控制模組 218 的要求，從記憶於緩衝器 215A 的程式串流來抽出字幕存取單元而讀出，且供應給字幕解碼器控制模組 218。

「視頻串流的讀出」

其次，參照第 45 圖的流程圖來說明有關利用視頻讀出功能部 233(第 5 圖)之來自緩衝器 215A 的視頻串流的讀出處理的詳細。

視頻讀出功能部 233，首先最初會在步驟 S211 中，搜尋找出記憶於緩衝器 215A 的程式串流中的

`private_stream_2`的 `PES_packet()`。亦即，

`private_stream_2`的 `PES_packet()`的 `stream_id` 如第 23 圖所說明那樣為 `10111111B(=0xBF)`，視頻讀出功能部 233 會搜尋找出 `stream_id` 為形成 `10111111B` 的 `PES_packet()`。

在此，例如現在像上述那樣儲存於剪輯串流檔案 "00001.PS" 的程式串流中被多重化的基本串流為再生對象的基本串流，則從碟片 101 讀出該程式串流，而使記憶於緩衝器 215A 時，在第 34 圖的步驟 S122 中，由記述於剪輯串流檔案 "00001.PS" 的 `EP_map()` (第 31 圖) 之解碼開始可能點的資訊決定 305 區段為再生開始位置，且在第 34 圖的步驟 S128 中，指定再生開始位置的 305 區段，對操作系統 201 要求讀出儲存於剪輯串流檔案 "00001.PS" 的程式串流。

又，有關視頻串流方面，記述於 `EP_map()` 之解碼開始可能點的資訊是顯示配置於實際的解碼開始可能點之前的 `private_stream_2` 的 `PES_packet()` 的位置。

因此，儲存於剪輯串流檔案 "00001.PS" 的程式串流會從碟片 101 讀出，在剛記憶於緩衝器 215A 之後，`private_stream_2` 的 `PES_packet()` 會被記憶於資料前頭指標或視頻讀出指標所指的緩衝器 215A 的位置。

視頻讀出功能部 233 在步驟 S211 中，若發現 `private_stream_2` 的 `PES_packet()`，則會前進至步驟 S212，抽出該 `private_stream_2` 的 `PES_packet()` 的 `PES_packet_data_byte` 之記述於

`private_stream2_PES_payload()`(第 26 圖)的 `video_stream_id`，判定該 `video_stream_id` 是否與在第 34 圖的步驟 S127 中記憶於 `stream_id` 暫存器 242(第 5 圖)的再生對象的視頻串流的 `stream_id` 一致。

在步驟 S212 中，當判定記述於 `private_stream2_PES_payload()` 的 `video_stream_id` 與記憶於 `stream_id` 暫存器 242 的 `stream_id` 不一致時，亦即，在之前的步驟 S211 所被找出的 `private_stream_2` 的 `PES_packet()` 不是被配置於再生對象的視頻串流的解碼開始點者時，回到步驟 S211，進行記憶於緩衝器 215A 的程式串流中的其他 `private_stream_2` 的 `PES_packet()` 之搜尋，以下，重複同樣的處理。

另一方面，在步驟 S212 中，當判定記述於 `private_stream2_PES_payload()` 的 `video_stream_id` 與記憶於 `stream_id` 暫存器 242 的 `stream_id` 一致時，亦即在之前的步驟 S211 所被找出的 `private_stream_2` 的 `PES_packet()` 為配置於再生對象的視頻串流的解碼開始點者時，前進至步驟 S213，視頻讀出功能部 233 會從緩衝器 215A 來讀出記述於該 `private_stream_2` 的 `PES_packet()` 的 `private_stream2_PES_payload()` 之 `au_information()`，使記憶於 `au_information()` 暫存器 243(第 5 圖)，前進至步驟 S214。

在步驟 S214 中，視頻讀出功能部 233 會只針對在之前的步驟 S211 所找出的 `private_stream_2` 的

PES_packet()(video_stream_id(第 26 圖)與記憶於 stream_id 暫存器 242(第 5 圖)的 stream_id 一致之 private_stream_2 的 PES_packet())的大小來更新記憶於視頻讀出指標記憶部 231 的視頻讀出指標。

亦即，在剪輯串流檔案中，在 private_stream_2 的 PES_packet()之後，配置有與該 video_stream_id 一致的 stream_id 的視頻串流(PES_packet())，因此在步驟 S214 中，視頻讀出指標是以能夠指出視頻串流的實際解碼開始可能點的位置之方式來更新。

然後，從步驟 S214 前進至 S215，視頻讀出功能部 233 會判定是否有從視頻解碼器控制模組 216 要求資料，若判定沒有時，回到步驟 S215，重複同樣的處理。

又，在步驟 S215 中，當判定有從視頻解碼器控制模組 216 要求資料時，前進至步驟 S216，視頻讀出功能部 233 會一方面進行來自視頻讀出指標所指的緩衝器 215A 的位置之程式串流的構文解析，一方面從緩衝器 215A 讀出記憶於 au_information()暫存器 243 的 au_information()的 AU_length 中所被記述的位元組數的資料，亦即 1 個視頻存取單元，供應給視頻解碼器控制模組 216，且只針對自緩衝器 215A 讀出的 1 個視頻存取單元的大小份量來更新視頻讀出指標。

亦即，在 au_information()中，如第 27 圖所說明那樣，記述有顯示從含彼之 private_stream_2 的 PES_packet()到其次的 private_stream_2 的 PES_packet()之間所含的視

頻存取單元(圖像)的數量之 `number_of_access_unit`。

又，在 `au_information()` 中，如第 27 圖所說明那樣，記述有分別關於僅該 `number_of_access_unit` 的數量之視頻存取單元的資訊，亦即 `pic_struct_copy`，`au_ref_flag`，及 `AU_length`。

在 `au_information()` 中僅記述有 `number_of_access_unit` 之數量的各 `AU_length`，是如第 27 圖所說明般，表示從包含此之 `private_stream_2` 的 `PES_packet()` 到下一個 `private_stream_2` 的 `PES_packet()` 為止之間所包含的 `number_of_access_unit` 之數量的各視頻存取單元的大小，因此視頻讀出功能部 233 可藉由使用該 `AU_length`，在不進行視頻串流的構文解析之下，進行存取單元之刪除。

亦即，以往刪除 MPEG2-Video 或 MPEG4-AVC 之存取單元時，必須知道視頻串流之構文，且必須執行視頻串流之構文解析，但是被儲存於記錄在碟片 101 之剪輯串流檔案之程式串流，因在視頻存取部件單位之視頻串流之 1 個以上之各實際解碼開始可能點之前，包含有記述著表示視頻存取單元之大小的 `AU_length` 的 `private_stream_2` 之 `PES_packet()`，故視頻讀出功能 233 是根據被記述於該 `private_stream_2` 之 `PES_packet()` 之 `AU_length`，可以不執行視頻串流之構文解析，自緩衝 215A 讀出視頻存取單元(單位之視頻串流)，並供給視頻解碼器控制模組 216。

並且，視頻讀出功能部 233 是在步驟 S216 中，當將視

頻存取單元供給至視頻解碼器控制模組 216 之時，將作為與該視頻存取單元有關之資訊而被記述於 `au_informayion` 之 `pic_struct_copy`、`au_ref_flag` 及 `AU_length` 和被附加於視頻存取部件單位之時間戳記 (PTS、DTS) 也供給至視頻解碼器控制模組 216。

在步驟 S216 中，自緩衝器 215A 讀入 1 個視頻存取單元，被供給至視頻解碼液控制模組 216 之後，則前進入步驟 S217，視頻讀出功能部 233 會判定是否處理僅被記憶於 `au_information()` 暫存器的 `au_information()` (第 27 圖) 的 `number_of_access_unit` 所示的數量之處理。

在步驟 S217 中，當判定僅 `number_of_access_unit` 所示之數量的存取單元尚未處理時，亦即，僅 `number_of_access_unit` 所示之數量的存取單元尚未自緩衝器 215A 讀出，而供給至視頻控制模組 216 時，則返回步驟 S215，以下重複相同之處理。

並且，在步驟 S217 中，當判定已處理 `number_of_access_unit` 所示之數量的存取單元時，亦即，從緩衝器 215A 讀出 `number_of_access_unit` 所示之數量的存取單元，而供給至視頻控制模組 216 時，則返回步驟 S211，進行下一個 `private_stream_2` 的 `PES_packet()` 之探索，以下重複相同之處理。

「音頻串流的讀出」

其次，參照第 46 圖的流程圖來說明有關利用音頻讀出

功能部 234(第 5 圖)之來自緩衝器 215A 的音頻串流的讀出處理的詳細。

音頻讀出功能部 234會首先最初在步驟 S230中，判定在第 34 圖的步驟 S127中記憶於 stream_id 暫存器 252(第 5 圖)之再生對象的音頻串流的 stream_id 是否顯示 private_stream_1 的 PES_packet()。

在步驟 S230中，當判定記憶於 stream_id 暫存器 252 的 stream_id 未顯示 private_stream_1 的 PES_packet()時，亦即，記憶於 stream_id 暫存器 252 的 stream_id，如第 23 圖所說明那樣，為分配於按照 MPEG 規格而編碼的音頻串流之 110xxxxxB 時，前進至步驟 S231，音頻讀出功能部 234會從記憶於緩衝器 215A 的程式串流來尋找表示音頻幀前頭(為 MPEG Audio 所定者)的同步碼。同步碼的位置為音頻幀前頭，因此音頻讀出功能部 234會以能夠顯示音頻幀前頭的位置之方式來更新音頻讀出指標，從步驟 S231 前進至 S232。在步驟 S232中，音頻讀出功能部 234會從音頻讀出指標所示的位置來搜尋找出記憶於緩衝器 215A 的程式串流中之與記憶於 stream_id 暫存器 252 的 stream_id 一致的 PES_packet()，前進至步驟 S233。

在步驟 S233中，音頻讀出功能部 234會以能夠指出在之前的步驟 S232中所找出的 PES_packet()的 PES_packet_data_byte(第 19 圖 A 及第 19 圖 B～第 21 圖 A 及第 21 圖 B)的前頭之方式來更新記憶於音頻讀出指標記憶部 251 的音頻讀出指標，前進至步驟 S237。

在步驟 S237中，音頻讀出功能部 234會判定是否有從音頻解碼器控制模組 217要求資料，當判定沒有時，回到步驟 S237，重複同樣的處理。

又，在步驟 S237中，當判定有從音頻解碼器控制模組 217要求資料時，前進至步驟 S238，音頻讀出功能部 234會一方面進行來自音頻讀出指標所指的緩衝器 215A 的位置之程式串流的構文解析，一方面從緩衝器 215A 讀出既定固定長的 1個音頻存取單元，與附加於該音頻存取單元的時間戳記 (PTS, DTS)一起供應給音頻解碼器控制模組 217。

而且，音頻讀出功能部 234會只針對從緩衝器 215A 讀出的 1個音頻存取單元的大小份量來更新音頻讀出指標，然後回到步驟 S237，以下，重複同樣的處理。

另一方面，在步驟 S230中，判定記憶於 stream_id 暫存器 252的 stream_id 為顯示 private_stream_1的 PES_packet()時，亦即，記憶於 stream_id 暫存器 252的 stream_id 為 10111101B(=0xBD)，如第 23圖所說明那樣，顯示 private_stream_1的 PES_packet()時，前進至步驟 S234，音頻讀出功能部 234會搜尋找出記憶於緩衝器 215A 的程式串流中的 private_stream_1的 PES_packet()。亦即，音頻讀出功能部 234會搜尋找出 stream_id 為形成 10111101B 的 PES_packet()。

若音頻讀出功能部 234在步驟 S234中發現 private_stream_1的 PES_packet()，則會前進至步驟 S235

，抽出該 `private_stream_1` 的 `PES_packet()` 的 `PES_packet_data_byte` 之記述於 `private_stream1_PES_payload()` (第 24 圖) 的 `private_stream_id`，判定該 `private_stream_id` 是否與在第 34 圖的步驟 S127 中記憶於 `private_stream_id` 暫存器 253 (第 5 圖) 的再生對象的音頻串流的 `private_stream_id` 一致。

在步驟 S235 中，當判定記述於 `private_stream1_PES_payload()` 的 `private_stream_id` 與記憶於 `private_stream_id` 暫存器 253 的 `private_stream_id` 不一致時，亦即，在之前的步驟 S234 中所被找出的 `private_stream_1` 的 `PES_packet()` 不是再生對象的音頻串流時，回到步驟 S234，進行記憶於緩衝器 215A 的程式串流中的其他 `private_stream_1` 的 `PES_packet()` 之搜尋，以下，重複同樣的處理。

另一方面，在步驟 S235 中，當判定記述於 `private_stream1_PES_payload()` 的 `private_stream_id` 與記憶於 `private_stream_id` 暫存器 253 的 `private_stream_id` 一致時，亦即在之前的步驟 S234 中所被找出的 `private_stream_1` 的 `PES_packet()` 為再生對象的音頻串流時，前進至步驟 S236，音頻讀出功能部 234 會從緩衝器 215A 讀出記述於該 `private_stream_1` 的 `PES_packet()` 的 `private_stream1_PES_payload()` (第 21 圖) 之 `AU_locator`，藉由加算緊接於該 `AU_locator` 之後的位置與表示該 `AU_locator` 的值來求取音頻存取單元的前頭位置。

亦即，如第24圖所說明那樣，AU_locator是以緊接為該AU_locator之後的位置為基準來表示儲存於private_stream1_PES_payload()的private_payload()之音頻存取單元或字幕存取單元的前頭位置，因此可藉由在緊接於AU_locator之後的位置加算表示該AU_locator的值來求取音頻存取單元的(絕對性的)前頭位置。

音頻讀出功能部234更於步驟S236中，以能夠指出以上那樣求取的音頻存取單元的前頭位置之方式來更新記憶於音頻讀出指標記憶部251的音頻讀出指標，前進至步驟S237。

在步驟S237中，音頻讀出功能部234會判定是否有從音頻解碼器控制模組217要求資料，當判定沒有時，回到步驟S237，重複同樣的處理。

並且，在步驟S237中，判定有從音頻解碼器控制模組217要求資料時，前進至步驟S238，音頻讀出功能部234會一方面進行來自音頻讀出指標所指的緩衝器215A的位置之程式串流的構文解析，一方面從緩衝器215A讀出既知固定長的1個音頻存取單元，與附加於該音頻存取單元的時間戳記一起供應給音頻解碼器控制模組217。

而且，音頻讀出功能部234會只針對從緩衝器215A讀出的1個音頻存取單元的大小份量來更新音頻讀出指標，回到步驟S237以下，重複同樣的處理。

「字幕串流的讀出」

其次，參照第47圖的流程圖來說明有關利用字幕讀出功能部235(第5圖)之來自緩衝器215A的字幕串流的讀出處理的詳細。

字幕讀出功能部235會首先最初在步驟S251中，判定在第34圖的步驟S127中記憶於字幕讀出功能旗標記憶部261的字幕讀出功能旗標。在步驟S251中，當判定字幕讀出功能旗標為0時，亦即，例如在再生對象的基本串流被多重化的剪輯串流檔案中不含字幕串流，在第34圖的步驟S127中，在字幕讀出功能旗標記憶部261中設定0時，字幕讀出功能部235不會特別進行處理。

另一方面，在步驟S251中，當判定字幕讀出功能旗標為1時，亦即，例如在再生對象的基本串流被多重化的剪輯串流檔案中含有字幕串流，在第34圖的步驟S127中，在字幕讀出功能旗標記憶部261中設定1時，前進至步驟S252，字幕讀出功能部235會從記憶於緩衝器215A的程式串流來搜尋記憶於stream_id暫存器263(第5圖)之與再生對象的字幕串流的stream_id一致的PES_packet()。

在此，如第34圖的步驟S127所說明那樣，在stream_id暫存器263(第5圖)中記憶有再生對象的字幕串流的stream_id，且字幕串流的stream_id，如第23圖所說明那樣，為表示private_stream_1的PES_packet()之10111101B(=0xBD)。

因此，在步驟S252中，記憶於緩衝器215A之程式串流中的private_stream_1的PES_packet()會被搜尋。

在步驟 S252中，若進行 `private_stream_1` 的 `PES_packet()` 的搜尋，而發現 `private_stream_1` 的 `PES_packet()`，則會前進至步驟 S253，字幕讀出功能部 235 會抽出該 `private_stream_1` 的 `PES_packet()` 的 `PES_packet_data_byte` 之 `private_stream1_PES_payload()` (第 24 圖) 中所記述的 `private_stream_id`，判定該 `private_stream_id` 是否與在第 34 圖的步驟 S127 中記憶於 `private_stream_id` 暫存器 264 (第 5 圖) 之再生對象的字幕串流的 `private_stream_id` 一致。

在步驟 S253 中，當判定記述於 `private_stream1_PES_payload()` 的 `private_stream_id` 與記憶於 `private_stream_id` 暫存器 264 的 `private_stream_id` 不一致時，亦即，在之前的步驟 S252 所發現的 `private_stream_1` 的 `PES_packet()` 不是再生對象的字幕串流時，回到步驟 S252，進行記憶於緩衝器 215A 的程式串流中的其他 `private_stream_1` 的 `PES_packet()` 之搜尋，以下，重複同樣的處理。

另一方面，在步驟 S253 中，當判定記述於 `private_stream1_PES_payload()` 的 `private_stream_id` 與記憶於 `private_stream_id` 暫存器 264 的 `private_stream_id` 一致時，亦即，在之前的步驟 S252 所發現的 `private_stream_1` 的 `PES_packet()` 為再生對象的字幕串流時，前進至步驟 S254，字幕讀出功能部 235 會從緩衝器 215A 來讀出該 `private_stream_1` 的 `PES_packet()` 的

`private_stream1_PES_payload()`(第 24 圖)中所記述的 `AU_locator`，藉由加算緊接於該 `AU_locator` 之後的位置與表示該 `AU_locator` 的值來求取字幕存取單元的前頭位置。

亦即，如第 24 圖所說明那樣，`AU_locator` 是以緊接於該 `AU_locator` 之後的位置為基準來表示儲存於 `private_stream1_PES_payload()` 的 `private_payload()` 之字幕存取單元(或音頻存取單元)的前頭位置，因此可藉由在緊接於 `AU_locator` 之後的位置加算該 `AU_locator` 所示的值來求取字幕存取單元的(絕對性的)前頭位置。

字幕讀出功能部 235 更於步驟 S254 中，以能夠指出以上那樣求取的字幕存取單元的前頭位置之方式來更新記憶於字幕讀出指標記憶部 262 的字幕讀出指標，前進至步驟 S255。

在步驟 S255 中，字幕讀出功能部 235 會判定是否有從字幕解碼器控制模組 218 要求資料，當判定沒有時，回到步驟 S255，重複同樣的處理。

並且，在步驟 S255 中，當判定有從字幕解碼器控制模組 218 要求資料時，前進至步驟 S256，字幕讀出功能部 235 會一方面進行來自字幕讀出指標所指的緩衝器 215A 的位置之程式串流的構文解析，一方面從緩衝器 215A 讀出記述於字幕存取單元的前頭之大小份量的 1 個字幕存取單元，且與附加於該字幕存取單元的時間戳記一起供應給字幕解碼器控制模組 218。亦即，在字幕存取單元的前頭中

，如第2圖 A 及第2圖 B 所說明那樣，記述有該字幕存取單元的大小，字幕讀出功能部 235 會從字幕讀出指標所指的緩衝器 215A 的位置來讀出其大小分量的資料，且將該讀出後的資料之字幕存取單元一起與附加於該字幕存取單元的時間戳記供應給字幕解碼器控制模組 218。

而且，字幕讀出功能部 235 會只針對從緩衝器 215A 讀出的 1 個字幕存取單元的大小份量來更新字幕讀出指標，然後回到步驟 S255，以下，重複同樣的處理。

[再同步處理]

其次，說明有關利用第2圖 A 及第2圖 B 的解碼控制模組 214 之視頻資料與音頻解碼器的同步控制。

如第34圖的步驟 S130 所說明，解碼控制模組 214 爲了確保同步，有時會因應所需挪動時序，對視頻解碼器控制模組 216，音頻解碼器控制模組 217，及字幕解碼器控制模組 218 下令解碼開始，例如根據之後的視頻解碼器 116 與音頻解碼器 117 的解碼處理的進行程度，有時視頻資料的輸出與音頻資料(應與該視頻資料同步輸出的輸出資料)的輸出會移動。

於是，在解碼控制模組 214 中會針對視頻資料的輸出與音頻資料(應與該視頻資料同步輸出者)的輸出所產生偏差進行補正，進行供以使視頻資料與音頻資料能夠同步輸出之再同步處理。

參照第48圖的流程圖來說明有關再同步處理。

再同步處理是首先最初在步驟 S271中，解碼控制模組 214會判定來自視頻解碼器控制模組 216之視頻存取單元的時間戳記與來自音頻解碼器控制模組 217之音頻存取單元的時間戳記的偏差是否大。

亦即，如第 34 圖的步驟 S129所說明那樣，視頻解碼器控制模組 216會在從緩衝控制模組 215取得視頻存取單元時，將該視頻存取單元的時間戳記供應給解碼控制模組 214。同樣的，音頻解碼器控制模組 217也會在從緩衝控制模組 215取得音頻存取單元時，將該音頻存取單元的時間戳記供應給解碼控制模組 214。

在步驟 S271中，解碼控制模組 214會比較以同一時序(可視為同一時序的某時間內)來分別從視頻解碼器控制模組 216與音頻解碼器控制模組 217供給的時間戳記彼此之間，判定該等時間戳記的偏差是否大。

在步驟 S271中，若判定來自視頻解碼器控制模組 216之視頻存取單元的時間戳記與來自音頻解碼器控制模組 217之音頻存取單元的時間戳記的偏差不大時，亦即，視頻存取單元的時間戳記與音頻存取單元的時間戳記的偏差為事先認定可視為取得同步的範圍內，例如，為 2 視頻幀(約 66 毫秒)時，回到步驟 S271，持續進行時間戳記彼此之間的偏差的判定(監視)。

另一方面，在步驟 S271中，當判別來自視頻解碼器控制模組 216之視頻存取單元的時間戳記與來自音頻解碼器控制模組 217之音頻存取單元的時間戳記的偏差大時，

亦即，視頻存取單元的時間戳記與音頻存取單元的時間戳記的偏差為事先認定可視為取得同步的範圍時，前進至步驟 S272，解碼控制模組 214 會藉由比較來自視頻解碼器控制模組 216 之視頻存取單元的時間戳記與來自音頻解碼器控制模組 217 之音頻存取單元的時間戳記，判別視頻資料的輸出(解碼)與音頻資料的輸出的其中哪個較慢。

在步驟 S272 中，當判別視頻資料的輸出比音頻資料的輸出更慢時，前進至步驟 S273，解碼控制模組 214 為了加快視頻存取單元的處理(1 視頻存取單元)，而對視頻解碼器控制模組 216 輸出不進行視頻存取單元的解碼及顯示的指示，亦即，視頻存取單元的處理跳過(skip)的指示，然後前進至步驟 S274。

在步驟 S274 中，視頻解碼器控制模組 216 會接受來自解碼控制模組 214 之跳過的指示，按照該跳過的指示，檢查與來自緩衝控制模組 215 的視頻存取單元一起被供給的 `au_ref_flag`(第 27 圖)。

亦即，在配置於 `private_stream_2` 的 `PES_packet()` 的 `private_stream2_PES_payload()`(第 26 圖)之 `au_information()`(第 27 圖)中含有關於存取單元的資訊之 `au_ref_flag`，如第 34 圖的步驟 S129 或第 45 圖的步驟 S216 所說明那樣，緩衝控制模組 215 會與視頻存取單元一起將該視頻存取單元的 `au_ref_flag` 供應給視頻解碼器控制模組 216。

在步驟 S274 中，檢查如此與存取單元一起被供給之

該存取單元的 `au_ref_flag`。

然後，從步驟 S274前進至 S275，視頻解碼器控制模組 216會根據從緩衝控制模組 215所供給之視頻存取單元的 `au_ref_flag` 的檢查結果，判定該視頻存取單元是否為其他圖像的解碼時不被參照的非參照畫像。

在此，如第 27 圖所說明那樣，視頻存取單元的 `au_ref_flag` 是表示該存取單元是否為參照畫像，是參照畫像時，會形成 1，不是參照畫像時，亦即，非參照畫像時，會形成 0。

在步驟 S275中，當判定從緩衝控制模組 215所供給的視頻存取單元不是非參照畫像的視頻存取單元時，亦即，從緩衝控制模組 215所供給的視頻存取單元為參照畫像時，前進至步驟 S276，視頻解碼器控制模組 216會如往常，使該視頻存取單元處理於視頻解碼器 116，其次的視頻存取單元會等待從緩衝控制模組 215供給，而回到步驟 S274。

並且，在步驟 S275中，當判定從緩衝控制模組 215所供給的視頻存取單元為非參照畫像時，前進至步驟 S277，視頻解碼器控制模組 216會使該視頻存取單元之利用視頻解碼器 116的處理跳過，其次的視頻存取單元會等待從緩衝控制模組 215供給，而回到步驟 S271。

如此藉由視頻存取單元的處理跳過，視頻存取單元的處理會大致加快 1 視頻存取單元量。其結果，比音頻資料的輸出更慢之視頻資料的輸出會變快。

另一方面，在步驟 S272中，當判定視頻資料的輸出不會比音頻資料的輸出更慢時，亦即，音頻資料的輸出比視頻資料的輸出更慢時，前進至步驟 S278，解碼控制模組 214爲了使視頻存取單元的處理等待，而對視頻解碼器控制模組 216輸出重複輸出的指示，亦即重複輸出對應於目前被解碼的視頻存取單元之視頻資料，而前進至步驟 S279。

在步驟 S279中，視頻解碼器控制模組 216會接受來自解碼控制模組 214之重複輸出的指示，按照該重複輸出的指示，重複對應於在視頻解碼器 116所被解碼的視頻存取單元之視頻資料，然後輸出至繪圖處理模組 219，其次的視頻存取單元會等待從緩衝控制模組 215供給，而回到步驟 S271。

如以上那樣，在解碼控制模組 214中，判定視頻資料的輸出是否比音頻資料的輸出更慢，當視頻資料的輸出比音頻資料的輸出更慢時，會指示視頻解碼器控制模組 216跳過 1個存取單元的處理。然後，在視頻解碼器控制模組 216中，根據被指示跳過的存取單元的 `au_ref_flag` 來判定該存取單元爲參照畫像或非參照畫像，當爲非參照畫像時，使被指示跳過的存取單元的處理在視頻解碼器 116中跳過。因此，可容易取得視頻資料的輸出與音頻資料的輸出同步。

亦即，當跳過處理的存取單元爲參照畫像時，對應於該存取單元的視頻資料爲了在之後被解碼的其他存取單元

的解碼時參照，而必須解碼。因此，在用以取視頻資料的輸出與音頻資料的輸出同步之同步控制中，一旦跳過參照畫像的存取單元的處理，則會無法解碼參照該參照畫像的其他存取單元，其結果，在視頻資料的顯示中，同步控制會成雜訊出現。

因此，不是參照畫像的存取單元，亦即非參照畫像的存取單元不期望跳過處理。

另一方面，針對以往的基本串流，爲了搜尋非參照畫像的存取單元，而必須進行基本串流的構文解析，例如藉由按照 MPEG4-AVC 等的編碼而取得的基本串流，因爲構文非常複雜，所以構文解析需要花費更大的成本。

相對的，被儲存於記錄在碟片 101 的剪輯流檔案中的程式串流，是視頻存取單元與被配置在

PES_packet_data_byte 之 PES_packet()(第 19 圖 A 及第 19 圖 B ~ 第 21 圖 A 及第 21 圖 B)，另外多重化配置有擴張

PES_packet_data_byte 之 private_stream2_PES_payload()(第 26 圖)之 private_stream_2 之 PES_packet()，在該

private_stream2_PES_payload()之 au_information()(第 27

圖)中，是記述於每視頻存取單元，表示該視頻存取單元

爲參照畫像，或是非參照畫像之 au_ref_flag。然後，該

au_ref_flag 是與所對應之視頻存取單元，同時自緩衝器

控制模組 215 被供給至視頻解碼器控制模組 216。因此，視

頻解碼器控制模組 216 藉由檢查與視頻存取單元同時被供

給之該視頻存取單元的 au_ref_flag，則可以幾乎不需花

費成本，認定視頻存取單元為參照畫像或是非參照畫像。

[標示處理]

其次，參照第49圖的流程圖來說明有關根據 `PlayListMark()`(第9圖)中所記述的 `Mark()`來進行的標示處理。

解碼控制模組214會經常確認藉由內藏的計時部214A而計時的現在時刻，在步驟S301中，判定現在時刻是否與記述於 `PlayListMark()`(第9圖)的 `Mark()`的 `mark_time_stamp`一致。

亦即，如第34圖的步驟S124所說明那樣，播放器控制模組212會在所欲再生第29圖所示的第1個 `PlayList#0`的第1個 `PlayItem#0`時，認定從第32圖上側所示的 `PlayListMark()`中所含的7個 `Mark()`中的第1個到第4個的4個 `Mark()`屬於 `PlayList#0`的第1個 `PlayItem#0`者，且將該4個 `Mark()`的 `mark_time_stamp`之{180,090}，{5,580,090}，{10,980,090}，{16,380,090}與該 `mark_time_stamp`所示時刻的屬性為「標示處理」的主旨一起交付給解碼控制模組214。

步驟S301是在解碼控制模組214中，判定現在時刻是否與上述從播放器控制模組212所供給的「標示處理」的屬性的時刻(`mark_time_stamp`)的其中之一個一致。

在步驟S301中，當判定現在時刻與「標示處理」的屬性的時刻的其中任何一個不一致時，回到步驟S301，

重複同樣的處理。

[標示處理的一致判定]

在此，有關標示處理，在步驟 S301中，解碼控制模組 214係進行現在時刻與 `mark_time_stamp` 的任一個是否一致的判定。但，在本實施例中，如第4圖所示，由於計時部 214A 所示的時刻為顯示離散的值，因此單純的一致判定會有問題發生。

利用第50圖說明稍微單純化的例子。在第50圖的上部，I0，P1，P2，P3為視頻存取單元。該等的 `pic_struct` 皆為3，亦即顯示 `duration` 為1幀(以90kHz換算，3003)。並且，在此例中，解碼順序與顯示順序為相同，不會發生亂序。又，如在「再生準備處理」所說明那樣，I0為登錄於 `EP_map()` 的存取單元，具有顯示時間標籤，PTS=180,090。P1，P2，P3的存取單元不具有顯示時間標籤。

在處理如此的視頻資料時，計時部 214A 所示的時鐘是如第50圖的下部所示那樣被更新。亦即，I0的輸出時，I0的 PTS 及 I0的 `pic_struct` 會被交付，PTS 會存在，因此 PTS 會被代入計時部 214A，而形成180,090。在 P1的輸出時，PTS 不會存在，P1的 `pic_struct` 會被交付，I0的 `pic_struct` 為3，因此1幀分的時間(90kHz 3003)會被加算於計時部 214A，形成183,093。同樣的，在 P2的輸出時，P1的 `pic_struct` 為3，因此3003會被加算，而於186,096、P3的輸出時，3003也會被加算而形成189,099。

在此，可想成被登錄於 `PlayListMark()`(第9圖及第32圖)的其中一個標示的 `mark_time_stamp` 為186,000時的處理。如上述，時鐘(計時部214A)的輸出值為180,090、183,093、186,096、189,099，所以與標示的時刻一致的186,000之時刻不會輸出。因此，`mark_time_stamp` 與時刻的單純比較，亦即在確認差分是否為0時會發生問題。

於是，在此對於時刻的一致判定，適用一定的規則。亦即，在此當所定的事件的 `mark_time_stamp` 為含於所定的畫像的顯示 `duration` 中時，該所定的事件是定義成發生於該當的畫像的顯示開始時刻。例如就上述的例子而言，`mark_time_stamp=186,000`是含於畫像 P1的顯示 `duration` 中。因此，該事件是發生於 P1的顯示開始時刻亦即183,093。

在如此的定義之下說明有關用以進行一致判定的解碼控制模組214的判定算法。

另外，此例的情況，時刻(計時部214A)的更新是僅發生於視頻資料的更新時。亦即，第49圖的步驟 S301是只在時刻的更新時進行。在以軟體構成的再生機器中，可有利於大幅度地減少處理。

又，按照時刻的更新來叫出第49圖的處理。解碼控制模組214係於步驟 S301中，確認現在時刻被判定一致的事件是否存在。亦即，從現在時刻及顯示中的畫像的顯示 `duration` 來確認是否有含於現在顯示中的畫像的顯示 `duration` 中的事件存在，當被判定成一致的事件不存在時

，處理會回到步驟 S301。當被判定成一致的事件存在時，處理會前進至步驟 S302。另外，當事件不存在時，可終了處理。

具體而言，例如 I0為顯示中時，時刻為180,090，I0的 `pic_struct` 為3，所以可知 I0的顯示 `duration` 為3003。因此，尋找符合 $180,090 \leq \text{mark_time_stamp} < 180,090 + 3003$ 的 `mark\_time\_stamp`。此刻所例示的事件時刻186,000是未符合該式，因此不會被判定成一致。

其次，例如 I1為顯示中時，時刻為183,093，I1的 `pic_struct` 為3，所以可知 I0的顯示 `duration` 為3003。因此，尋找符合 $183,093 \leq \text{mark_time_stamp} < 183,093 + 3003$ 的 `mark\_time\_stamp`。由於此刻所例示的事件時刻186,000符合該式，因此時刻會被判定成一致，步驟 S302以後的處理會被進行。

雖此例為時刻的一致判定之一例，但亦可適用其他的定義。例如，某事件的 `mark\_time\_stamp` 為含"某畫像的顯示開始- α "的時刻以上，"其次的顯示畫像的顯示開始時刻- α "的時刻未滿時，亦可定義成該事件發生於該當的畫像的顯示開始時刻。又，亦可維持判定基準，將事件的發生時刻定義成發生於例如"該當畫像的顯示開始時刻- α "。

另外，藉由導入如此的定義，標示的時刻亦即 `mark\_time\_stamp` 的設定時不必知道視頻串流的時刻。因此，編寫(authoring)的視頻編碼及資料基礎作成的獨立性高，可分離作業。

並且，在步驟 S301中，當判定現在時刻與「標示處理」的屬性的時刻的其中一個一致時，解碼控制模組 214會將現在時刻為形成「標示處理」的屬性的時刻之訊息，及與現在時刻一致的「標示處理」的屬性的時刻供應給播放器控制模組 212，然後前進至步驟 S302。

在步驟 S302中，播放器控制模組 212會從解碼控制模組 214來接受現在時刻為形成「標示處理」的屬性的時刻之訊息，及與現在時刻一致的「標示處理」的屬性的時刻 (mark_time_stamp)，且將 mark_time_stamp 與現在時刻一致的 Mark()當作標示處理的處理對象的 Mark()(以下稱為處理對象 mark)來認定。

亦即，播放器控制模組 212會認定現在所被再生的 PlayList()的 PlayItem()，從該 PlayList()及 PlayItem()，以及來自解碼控制模組 214之與現在時刻一致的「標示處理」的屬性的時刻 (mark_time_stamp)(以下稱為標示時刻)，參照 "PLAYLIST.DAT"檔案(第7圖)的 PlayListMark()(第9圖)來認定處理對象 mark。

具體而言，例如若現在第29圖所示的第1個 PlayList#0的第1個 PlayItem#0被再生，則播放器控制模組 212會認定標示時刻為第32圖上側所示的 PlayListMark()中所含的7個 Mark()中的第1個～第4個的4個 Mark()的其中一個的 mark_time_stamp。

若例如從解碼控制模組 214供應給播放器控制模組 212的標示時刻為 16,380,090，則播放器控制模組 212會將第32

圖上側所示的 `PlayListMark()`中所含的第1個～第4個的4個 `Mark()`之中 `mark_time_stamp` 與標示時刻的16,380,090一致的 第4個 `Mark()`作為處理對象 `mark` 來認定。

若播放器控制模組 212如以上那樣來認定處理對象 `mark`，則會從步驟 S302前進至 S303，判定在處理對象 `mark` 中是否記述有特定基本串流的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id`(第9圖)。

在步驟 S303中，當判定在處理對象 `mark` 中未記述有特定基本串流的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id`(第9圖)時，亦即，`entry_ES_stream_id` 及 `entry_ES_private_stream_id` 皆為 0x00時，跳過步驟 S304，前進至步驟 S305，以下，進行對應於處理對象 `mark` 的處理。

並且，在步驟 S303中，當判定在處理對象 `mark` 中記述有特定基本串流的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id`(第9圖)時，前進至步驟 S304，播放器控制模組 212會判定在再生中的基本串流中是否含有依據該 `entry_ES_stream_id`，甚至因應所需依據 `entry_ES_private_stream_id` 來特定的基本串流。

在步驟 S304中，當判定在再生中的基本串流中未含有依據處理對象 `mark` 的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 來特定的基本串流時，回到步驟 S301。亦即，當依據處理對象 `mark` 的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 來特定

的基本串流未被再生時，無視處理對象 `mark`。

另一方面，在步驟 S304中，當判定在再生中的基本串流中含有依據處理對象 `mark` 的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 來特定的基本串流時，亦即，依據處理對象 `mark` 的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 來特定的基本串流會被再生時，處理對象 `mark` 為有效，前進至步驟 S305，以下，進行對應於該處理對象 `mark` 的處理。

亦即，在步驟 S305中，播放器控制模組 212會藉由參照處理對象 `mark` 的 `mark_type`(第9圖)來判定該處理對象 `mark`。

在步驟 S305中，當判定處理對象 `mark` 為章節標示或索引標示時，亦即，處理對象 `mark` 的 `mark_type` 為 'Chapter'或 'Index'時，前進至步驟 S306，播放器控制模組 212會命令繪圖處理模組 219，使章節或索引的號碼的顯示更新成處理對象 `mark` 之章節標示或索引標示所示的章節或索引的號碼，而回到步驟 S301。

並且，在步驟 S305中，當判定處理對象 `mark` 為事件標示時，亦即，處理對象 `mark` 的 `mark_type` 為 'Event'時，前進至步驟 S307，播放器控制模組 212會將顯示事件的發生之事件訊息，及處理對象 `mark` 的 `mark_data` 通知(供應)給腳本控制模組 211，而前進至步驟 S308。

在步驟 S308中，腳本控制模組 211會接受來自播放器控制模組 212的事件訊息及 `mark_data`，將事件訊息當作中

斷要求，以 `mark_data` 為引數來進行事先記述於

"SCRIPT.DAT"檔案的一連串處理，而回到步驟 S301。

亦即，在腳本控制模組 211中，進行對應於 `mark_data` 的處理。

具體而言，例如，在第 32圖下側所示的 `PlayList #1` 的 `PlayListMark()` 中，第 2個 `Mark()(Mark#1)` 與第 3個 `Mark()(Mark#2)` 皆是 `mark_type` 為 'Event'，但 `mark_data` 有所不同，分別為 1(Mark#1)及 2(Mark#2)。

腳本控制模組 211在接受對應於第 2個 `Mark()` 的事件訊息時，及接受對應於第 3個 `Mark()` 的事件訊息時，皆會按照該事件訊息以同一事件處理器(中斷(interrupt)處理程序)來進行處理，但在事件處理器(handler)內，會藉由檢查與事件訊息一起被供給的 `mark_data` 來對事件訊息進行每個 `mark_data` 相異的處理。

具體而言，例如當 `mark_data` 為 1時，腳本控制模組 211會控制繪圖處理模組 219來使進行第 1種類的圖像(icon)顯示。又，例如當 `mark_data` 為 2時，腳本處理模組 211會控制繪圖處理模組 219來使進行第 2種類的圖像顯示。

另外，`mark_data` 並非限於 1或 2，且對應於 `mark_data` 來進行的處理也非限於上述一般的圖像顯示。

亦即，例如當 `mark_data` 為 3～18的範圍的值時，腳本控制模組 211會控制繪圖處理模組 219，使第 1種類的圖像顯示以對應於由 `mark_data` 減去 2的值(1～16的數值)之亮度來進行。又，例如當 `mark_data` 為 19～34的範圍的值

時，腳本控制模組 211 會控制繪圖處理模組 219，使第 2 種類的圖像顯示以對應於由 `mark_data` 減去 18 的值 (1 ~ 16 的數值) 之亮度來進行。

其他，例如在輸入介面 115 (第 1 圖) 中連接有使用者所操作之控制器，該控制器是於內藏有安裝使偏心於 DC (Direct Current) 馬達之軸的鉛墜，且當動作 DC 馬達時發生震動的震動馬達時，於 `mark_data` 為 35 至 42 之範圍的值時，可僅以使該震動馬達僅在因應自 `mark_data` 減 34 之值 (1 至 8 之數值) 之動作時間予以動作。

`mark_data` 為數值，其使用法或演算法 (algorithm) 可藉由腳本控制模組 211 所執行的腳本程式來記述。因此，`mark_data` 除了以事前決定的規則來使用之外，碟片 101 的製造者，或提供記錄於碟片 101 的資料的內容提供者等還可以獨自設定的規則來使用。

如以上所述，在標示處理中，若現在時刻與「標示處理」的屬性的時刻一致，則可由該「標示處理」的屬性的時刻之標示時刻來認定處理對象 `mark`。此外，在處理對象 `mark` 中，未記述有特定基本串流的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 時，會進行對應於處理對象 `mark` 的 `mark_type` 的處理。另外，在處理對象 `mark` 中，記述有特定基本串流的 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 時，只要依據該 `entry_ES_stream_id` 及 `entry_ES_private_stream_id` 來特定的基本串流為再生中

，便可進行對應於處理對象 mark 的 mark_type 之處理。

因此，例如現在若進行第29圖所示的第2個 PlayList#1的再生，則會進行以下所述的標示處理。

亦即，在第2個 PlayList#1的 PlayListMark()中，如第32圖下側所示，記述有 mark_time_stamp 分別被指定為 90,000，27,090,000，27,540,000的第1個 Mark()(Mark#0)，第2個 Mark()(Mark#1)，第3個 Mark()(Mark#2)。

又，在第32圖下側的 PlayListMark()中，因為在第2個 Mark()與第3個 Mark()的 entry_ES_stream_id 中分別記述有 0xE0及 0xE1，所以第2個 Mark()與第3個 Mark()會分別和以 stream_id 為 0xE0及 0xE1所特定的基本串流有關。

在此，如第29圖所說明那樣，在第2個 PlayList#1中僅記述1個 PlayItem()(PlayItem#0)，剪輯串流檔案 "00003.PS"會依據該 PlayItem#0而再生。而且，在剪輯串流檔案 "00003.PS"中，如對應於該剪輯串流檔案 "00003.PS"之第30圖 A 及第30圖 B 的剪輯資訊檔案 "00003.CLP"所說明那樣，以 0xE0的 stream_id 所特定的視頻串流 stream#0，以 0xE1的 stream_id 所特定的視頻串流 stream#1，以 0xBD 的 stream_id 及 0x00的 private_stream_id 所特定的音頻串流 stream#2等3個基本串流會被多重化。

因此，在第32圖下側的 PlayListMark()的第2個 Mark()中是有關剪輯串流檔案 "00003.PS"中被多重化之 stream_id 為形成 0xE0的視頻串流 stream#0，在第3個

Mark()中是有關剪輯串流檔案 "00003.PS"中被多重化之 stream_id 為形成 0xE1的視頻串流 stream#1。

當第29圖的第2個 PlayList#1的 PlayItem#0的再生開始時，如第34圖的步驟 S124所說明那樣，播放器控制模組212會認定第32圖下側所示的 PlayListMark()中所含的3個 Mark()屬於 PlayList#1的 PlayItem#0，且將該3個 Mark()的 mark_time_stamp 之 {90,000}，{27,090,000}，{27,540,000}與該 mark_time_stamp 所示的時刻的屬性為「標示處理」的主旨一起交付給解碼控制模組214。

在標示處理中，解碼控制模組214會在 PlayList#1的 PlayItem#0的再生中，經常確認藉由計時部214A而計時的現在時刻是否會與屬性為「標示處理」的時刻{90,000}，{27,090,000}，{27,540,000}的其中之一個一致(步驟S301)，若現在時刻與屬性為「標示處理」的時刻一致，則會將與現在時刻一致的「標示處理」的屬性的時刻之標示時刻及現在時刻為「標示處理」的屬性的時刻之訊息供應給播放器控制模組212。

亦即，例如目前若現在時刻與「標示處理」的屬性的時刻{90,000}，{27,090,000}，{27,540,000}之中的27,090,000一致，則解碼控制模組214會將與現在時刻一致的「標示處理」的屬性的時刻之標示時刻27,090,000及現在時刻為「標示處理」的屬性的時刻之訊息供應給播放器控制模組212。

播放器控制模組212會認定 PlayList#1的 PlayItem#0

現在被再生，比較記述於該 PlayList#1 的第 32 圖下側所示的 PlayListMark() 之 Mark() 中屬於 PlayItem#0 的 3 個 Mark() 的 mark_time_stamp 之各個 90,000，27,090,000，27,540,000 與來自解碼控制模組 214 的標示時刻之 27,090,000，藉此以該標示時刻 27,090,000 中，mark_time_stamp 一致的 Mark()，亦即記述於第 32 圖下側的 PlayListMark() 的第 2 個 Mark() (Mark#1) 為處理對象 mark 來認定 (步驟 S302)。

在處理對象 mark 之記述於第 32 圖下側的 PlayListMark() 的第 2 號 Mark() 中，entry_ES_stream_id 會被指定為 0xE0。此 0xE0 的 entry_ES_stream_id 是表示剪輯串流檔案 "00003.PS" 中被多重化之 stream_id 為形成 0xE0 的視頻串流 stream#0 (第 30 圖 A 及第 30 圖 B) 者，播放器控制模組 212 會判定再生中的基本串流之中是否含有該視頻串流 stream#0 (步驟 S303，S304)。

然後，當再生中的基本串流之中未含有視頻串流 stream#0 時，無視於處理對象 mark (步驟 S304)。

另一方面，當在再生中的基本串流之中含有視頻串流 stream#0 時，處理對象 mark 為有效，進行對應於該處理對象 mark 的處理 (步驟 S305～S308)。

亦即，現在的情況，處理對象 mark 之記述於第 32 圖下側的 PlayListMark() 的第 2 個 Mark()，因為其 mark_type 為形成 'Event'，所以為事件標示，因此，播放器控制模組 212 會將顯示發生事件的事件訊息及處理對象 mark 的

`mark_data` 供應給腳本控制模組 211(步驟 S305, S307)。
然後，在腳本控制模組 211中，將來自播放器控制模組 212
的事件訊息當作中斷要求，以和該事件訊息一起被供給的
`mark_data` 作為引數來進行事先記述於 "SCRIPT.DAT" 檔案
的一連串處理(步驟 S308)。

如以上所述，利用標示處理來判定按照包含具有 0 以
上的 `Mark()`(包含表示 `PlayList()` 的時間軸上的 1 個再生時
刻之 `mark_time_stamp`，及表示 `Mark()` 的類型之
`mark_type`，及形成事件標示的引數之 `mark_data`) 的
`PlayListMark()`(第 9 圖)之 `PlayList()`(第 7 圖)來再生的剪輯
串流檔案的再生時刻之現在時刻是否與 `mark_time_stamp`
一致，當現在時刻與 `mark_time_stamp` 一致時，具有相等
於該一致的現在時刻的標示時刻的 `mark_time_stamp` 之
`Mark()` 會作為處理對象 `mark` 來認定。另外，當該處理對
象 `mark` 所持有的 `mark_type` 為表示使事件發生的類型時
，亦即，處理對象 `mark` 為事件標示時，處理對象 `mark` 所
持有的 `mark_data` 及事件訊息會被通知，執行對應於該
`mark_data` 的處理。因此，可按照剪輯串流檔案的再生時
刻，執行對應於 `mark_data` 的處理。

[out_time 處理的一致判定]

但，如上述，若計時部 214A 所計時的時刻相等於從
播放器控制模組 212 所被賦予的 `PlayItem` 的 `OUT_time`，
則解碼控制模組 214 會進行解碼中斷控制，進行所謂終了

PlayItem 的再生之控制。在本實施例中，有關 PlayItem#0 的終了方面，是以第 40 圖的流程圖之步驟 S151 來說明。

此情況，同樣在時刻與 OUT_time 的單純比較之一致判定中會有可能發生問題。因此，在時刻與 OUT_time 的比較中亦使用上述一致判定的定義。

亦即，如第 51 圖所示那樣，當對應於 playListEnd 之 PlayItem 的 OUT_time 比在 playList 最後顯示的 FoCFP(再生中的視頻串流內的 frame or complementary field pair) 的 PET 更小時，在使相當於 PlayListEnd 的時刻之 OUT_time 含於顯示 duration 中的 FoCFP 的顯示開始時刻 (PTS) 發生 PlayListEnd 事件時，亦即

$PTS_{FOCFP[3]} \leq OUT_time < PET_{FOCFP[3]}$ 時，PlayListEnd 事件會在 FoCFP[3] 的顯示開始時刻 $PTS_{FOCFP[3]}$ 發生。在此 $PET_{FOCFP[k]}$ 是表示「在 $PTS_{FOCFP[k]}$ 追加決定於 pic_struct 的顯示 duration 之時刻」。

藉此，時刻的一致判定會被限定於視頻輸出時，因此處理會便輕。又，如上述，視頻串流的準備及資料基礎的準備之獨立性高。

此外，PlayItem 的再生終了會從解碼控制模組 214 來傳遞給播放器控制模組 212，若播放器控制模組 212 判定此為再生中的最後 PlayList 的 PlayItem，則會對腳本控制模組 211 發生 playListEnd 的事件。

腳本控制模組 211 係藉由接受 playListEnd 的事件，得知指示再生的 PlayList 的再生終了，然後繼續進行被程式

化的動作。亦即，執行再生別的 `playList` 之顯示選單的終了動作等。

另外，在第52圖所示那樣的情況時，亦即，
`OUT_time` 為等於 `PlayItem` 中最後顯示的畫像的顯示終了時刻時，就上述的一致判定而言，有時無法處理。在第52圖中，例如顯示 `FoCFP[2]` 而暫停(`pause`)時，若 `playStop()` 被 `call`，則會顯示 `FoCFP[3]` 而暫停。然後，若再度 `playStop()` 被 `call`，則在所被顯示的畫像中會發生無變更的 `playListEnd`。

亦即，形成決定於最後的畫像的顯示開始時刻
`+pic_struct` 之 `duration=OUT_time`，因此不會符合決定於最後的畫像的顯示開始時刻 `+pic_struct` 之
`duration<OUT_time`。

如此的情況，是視頻解碼器控制模組216在輸出最後的畫像之後，將該當畫像的顯示 `duration` 經過後表示顯示終了的信息交付解碼控制模組214，藉此使時鐘前進至"決定於最後的畫像的顯示開始時刻 `+pic_struct` 的 `duration`"，而使能夠符合一致條件。

[字幕解碼]

字幕解碼器控制模組218是在由緩衝器控制模組215(第5圖)的字幕讀出功能部235來取得記憶於緩衝器215A的1個字幕存取單元、及附加於該字幕存取單元的顯示時間標籤時，以內部所具有的字幕解碼軟體來開始解碼，同時

將顯示時間標籤及 duration 交付解碼控制模組 214。

解碼控制模組 214 是在根據來自視頻解碼器控制模組 216 的資訊變更時鐘 (計時部 214A) 的時刻時，斟酌自字幕解碼器控制模組 218 所交付的字幕存取單元的 PTS。亦即，若字幕存取單元的 PTS 與時刻按照上述的一致判定基準來判斷成一致，則解碼控制模組 214 會對繪圖處理模組 219 進行字幕輸入的指示，同時對字幕解碼器控制模組 218 進行輸出的指示。

自解碼控制模組 214 接受輸出的指示之字幕解碼器控制模組 218 會將所被解碼的字幕畫像資料供應給繪圖處理模組 219。繪圖處理模組 219 會儲存所被輸入的字幕資料，且對以後輸入的視頻資料進行合成。

解碼控制模組 214 更進行字幕的顯示 duration 的檢查。亦即、若 "該當字幕的顯示開始時刻 + 顯示 duration" 的值與現在時刻按照上述的一致判定基準來判斷成一致，則解碼控制模組 214 會對繪圖處理模組 219 命令字幕的消去。繪圖處理模組 219 會消去所儲存之輸入的字幕資料，不對以後輸入的視頻資料進行合成。

[標示間隔的必然性]

在以上所說明的一致判定基準中，是將一定範圍的時刻予以單一時刻量子化。亦即，符合某視頻資料的顯示開始時刻 $\leq t <$ 顯示終了時刻之所有的 t 時刻是在該當視頻資料的顯示開始時刻進行量子化 (rounding)。

因此，依據相鄰的二個事件的位置關係，有時會被搓合於同一時刻。例如就第50圖的例子而言，位於186,000的事件之前的事件的 `mark_time_stamp` 為184,000時，該二個事件皆被定義成發生於 P1的顯示開始時刻。

爲了避免如此的狀況，必須保證可對單一的視頻只設定單一的事件。因此，在相隣的事件間隔爲3場以上(被設定於 `pic_struct` 的最大顯示時間以上)的情況下，保證上述條件。

第53圖是表示上述條件之一例。亦即，在第53圖中，事例 A 是幀速度(`frame rate`)爲5005/240000(循序式(`progressive`)的23.976Hz)，該情況，以90kHz來表現的最小事件間隔是形成7507，事例 B 是幀速度(`frame rate`)爲4004/240000(交錯式(`interlace`)的59.94Hz)，該情況，以90kHz來表現的最小事件間隔是形成6006。

就 AVC 或 MPEG2 Video 等的視頻編碼方式而言，爲了有效率地編碼 2-3PULL-DOWN 畫像，而具有使1幀信號僅顯示3場的時間之功能。因此，1幀的信號的最大 `duration` 爲3場。亦即，使相隣的事件的間隔只分離3場以上的時間，藉此防止判斷成該二個相隣的事件發生於單一的視頻資料的顯示開始時刻。

又，亦可以越過3場的間隔來定義。例如亦可將相隣的事件的間隔定義成2幀以上。

另外，亦可藉由調查全體事件所對應的視頻資料，確認該等是否未重複之方法。

[輸出屬性的控制處理]

其次，參照第54圖的流程圖來說明有關在第34圖的步驟 S126等中所被進行之輸出屬性的控制處理的詳細。

如第34圖的步驟 S126所說明那樣，播放器控制模組 212會首先針對再生對象的1個以上的基本串流，亦即針對在第34圖的步驟 S125中再生後決定的1個以上的各個基本串流來調查表示記述有輸出屬性的 `DynamicInfo()`(第15圖)的數量之 `number_of_DynamicInfo`(第12圖)。

有關再生對象的1個以上的基本串流全體，當 `number_of_DynamicInfo` 為形成0時，播放器控制模組 212不會特別進行處理。

另一方面，有關再生對象的基本串流之 `number_of_DynamicInfo` 不是0時，播放器控制模組 212會進行按照第54圖的流程圖之輸出屬性的控制處理。

因此，記錄於碟片 101的3個剪輯資訊檔案 "00001.CLP"，"00002.CLP"，"00003.CLP"，例如為形成第30圖 A 及第30圖 B 所示那樣時，當再生對應於剪輯資訊檔案 "00001.CLP"的剪輯串流檔案 "00001.PS"之第1個 `Playlist#0`的第1個 `PlayItem#0`被再生時，在剪輯資訊檔案 "00001.CLP"(第30圖 A 及第30圖 B)中，因為有關剪輯串流檔案 "00001.PS"中被多重化的4個基本串流 `stream#0`～`stream#3`全體，`number_of_DynamicInfo` 會形成0，因此輸出屬性的控制處理不會被進行。

同樣的，當對應於剪輯資訊檔案 "00002.CLP" 的剪輯串流檔案 "00002.PS" (第 1 個 PlayList#0 的第 2 個 PlayItem#1) 被再生時，在剪輯資訊檔案 "00002.CLP" (第 30 圖 A 及第 30 圖 B) 中，因為有關剪輯串流檔案 "00002.PS" 中被多重化的 4 個基本串流 stream#0 ~ stream#3 全體，number_of_DynamicInfo 會形成 0，所以輸出屬性的控制處理不會被進行。

另一方面，當對應於剪輯資訊檔案 "00003.CLP" 的剪輯串流檔案 "00003.PS" (第 2 個 PlayList#1 的 PlayItem#0) 被再生時，在剪輯資訊檔案 "00003.CLP" (第 30 圖 A 及第 30 圖 B) 中，因為有關剪輯串流檔案 "00003.PS" 中被多重化的 3 個基本串流 stream#0 ~ stream#2 之中的第 1 個基本串流之視頻串流 stream#0，及第 3 個基本串流之音頻串流 stream#2 方面，number_of_DynamicInfo 會分別形成非 0 的 2 及 3，所以輸出屬性的控制處理會被進行。

亦即，在輸出屬性的控制處理中，首先最初在步驟 S320 中，播放器控制模組 212 會將對應於再生對象的剪輯串流檔案之剪輯資訊檔案 Clip() (第 12 圖) 中所記述的 pts_change_point 與「DynamicInfo() 處理」的屬性的時刻一起交付給解碼控制模組 214，解碼控制模組 214 會接受來自播放器控制模組 212 的「DynamicInfo() 處理」的屬性的時刻之 pts_change_point，前進至步驟 S321。

在步驟 S321 中，解碼控制模組 214 會判定藉由計時部 214A 所計時的現在時刻是否與「DynamicInfo() 處理」的

屬性的時刻之 `pts_change_point` 一致，當判定為非一致時，回到步驟 S321。

並且，在步驟 S321中，當判定現在時刻與「`DynamicInfo()`處理」的屬性的時刻一致時，解碼控制模組 214會將現在時刻為形成「`DynamicInfo()`處理」的屬性的時刻之訊息，及與現在時刻一致的「`DynamicInfo()`處理」的屬性的時刻(以下稱為 `DynamicInfo` 時刻)供應給播放器控制模組 212，而前進至步驟 S322。

在步驟 S332中，播放器控制模組 212會從解碼控制模組 214來接受現在時刻為形成「`DynamicInfo()`處理」的屬性的時刻之訊息及 `DynamicInfo` 時刻，以和一致於該 `DynamicInfo` 時刻的 `pts_change_point`(第 12 圖)成組的 `DynamicInfo()`作為處理對象的 `DynamicInfo()`之處理對象 `DynamicInfo()`來認定，而前進至步驟 S323。

在步驟 S323中，播放器控制模組 212會將形成處理對象 `DynamicInfo()`之 `DynamicInfo()`(第 15 圖)中所記述的輸出屬性供應給繪圖處理模組 219或音頻輸出模組 221，而前進至步驟 S324。

在步驟 S324中，繪圖處理模組 219或音頻輸出模組 221會按照之前的步驟 S323中從播放器控制模組 212所供給的輸出屬性來分別開始進行視頻資料或音頻資料的輸出控制，回到步驟 S321。

藉此，視頻資料會按照作為輸出方式而被記述之例如寬高比來輸出，或，音頻資料會按照作為輸出方式而被記

述之例如立體或雙語來輸出。

其次，參照第55圖來更加說明有關輸出屬性的控制處理的詳細。

亦即，第55圖是表示第30圖 A 及第30圖 B 的剪輯資訊檔案 "00003.CLP" 中所記述的 pts_change_point 與 DynamicInfo() 的組合(第12圖)。

在此，如上述，有關剪輯串流檔案 "00003.PS" 中被多重化的3個基本串流 stream#0～stream#2 之中的第1個基本串流的視頻串流 stream#0，及第3個基本串流的音頻串流 stream#2，在第30圖 A 及第30圖 B 的剪輯資訊檔案 "00003.CLP" 中，number_of_DynamicInfo 會分別形成2及3。因此，在剪輯資訊檔案 "00003.CLP" 中，有關剪輯串流檔案 "00003.PS" 的第1個視頻串流 stream#0，記述有2組的 pts_change_point 及 DynamicInfo()，有關第3個音頻串流 stream#2，記述有3組的 pts_change_point 及 DynamicInfo()。

第55圖上側是表示針對剪輯串流檔案 "00003.PS" 的第1個視頻串流 stream#0 來記述的2組 pts_change_point 及 DynamicInfo()，第55圖下側是表示針對剪輯串流檔案 "00003.PS" 的第3個音頻串流 stream#2 來記述的3組 pts_change_point 及 DynamicInfo()。

另外，在第55圖上側，除了針對第1個視頻串流 stream#0 來記述的2組 pts_change_point 及 DynamicInfo() 以外，亦圖示有針對該視頻串流 stream#0 來記述於第30圖

A 及第 30 圖 B 的剪輯資訊檔案 "00003.CLP" 之
stream_id(=0xE0)，private_stream_id(=0x00)，
number_of_DynamicInfo(=2)。同樣的，在第 55 圖下側，
除了針對第 3 個音頻串流 stream#2 來記述的 3 組
pts_change_point 及 DynamicInfo() 以外，亦圖示有針對該
音頻串流 stream#2 來記述於第 30 圖 A 及第 30 圖 B 的剪輯
資訊檔案 "00003.CLP" 之 stream_id(=0xBD)，
private_stream_id(=0x00)，number_of_DynamicInfo(=3)。

在第 55 圖上側，針對視頻串流 stream#0 而記述的 2 組
pts_change_point 及 DynamicInfo() 之中的第 1 組中，
pts_change_point 會形成 90,000，DynamicInfo() 的
display_aspect_ratio(第 15 圖)會形成 '4 : 3'。又，該第 2 組
中，pts_change_point 會形成 54,090,000，DynamicInfo()
的 display_aspect_ratio 會形成 '16 : 9'。

另一方面，在第 55 圖下側，針對音頻串流 stream#2 而
記述的 3 組 pts_change_point 及 DynamicInfo() 之中的第 1
組中，pts_change_point 會形成 90,000，DynamicInfo() 的
channel_assignment(第 15 圖)會形成 'Dual'。又，該第 2 組
中，pts_change_point 會形成 27,090,000，DynamicInfo()
的 channel_assignment 會形成 'Stereo'。又，該第 3 組中，
pts_change_point 會形成 32,490,000，DynamicInfo() 的
channel_assignment 會形成 'Dual'。

例如，目前在第 34 圖的步驟 S125 中，剪輯串流檔案
"00003.PS" 之以 0xE0 的 stream_id 所特定的第 1 個視頻串流

stream#0，及以 0xBD 的 stream_id 及 0x00 的 private_stream_id 所特定的第 3 個音頻串流 stream#2 會被決定作為再生對象的串流。

此情況，播放器控制模組 212 會調查針對以 0xE0 的 stream_id 所特定的視頻串流 stream#0 來記述的第 55 圖上側的 2 組 pts_change_point 及 DynamicInfo()，與針對以 0xBD 的 stream_id 及 0x00 的 private_stream_id 所特定的音頻串流 stream#2 來記述的第 55 圖下側的 3 組 pts_change_point 及 DynamicInfo() 之中的 pts_change_point，認定初期值。

亦即，在針對以 0xE0 的 stream_id 所特定的視頻串流 stream#0 來記述的第 55 圖上側的 2 組 pts_change_point 及 DynamicInfo() 之中的第 1 組中，pts_change_point 會形成 90,000。而且，此 90,000 的時刻是在對應於視頻串流 stream#0 所被多重化的剪輯串流檔案 "00003.PS" 之第 30 圖 A 及第 30 圖 B 的剪輯資訊檔案 "00003.CLP" 中，與表示剪輯串流檔案 "00003.PS" 的前頭的時刻之 presentation_start_time 中所記述的時刻 90,000 一致。

同樣的，在針對以 0xBD 的 stream_id 及 0x00 的 private_stream_id 所特定的音頻串流 stream#2 來記述的第 55 圖下側的 3 組 pts_change_point 及 DynamicInfo() 之中的第 1 組中，pts_change_point 會形成 90,000。而且，此 90,000 的時刻是在對應於音頻串流 stream#2 所被多重化的剪輯串流檔案 "00003.PS" 之第 30 圖 A 及第 30 圖 B 的剪輯資

訊檔案 "00003.CLP" 中，與表示剪輯串流檔案 "00003.PS" 的前頭的時刻之 `presentation_start_time` 中所記述的時刻 90,000 一致。

播放器控制模組 212 會將與表示剪輯串流檔案 "00003.PS" 的前頭的時刻之 `presentation_start_time` 中所記述的時刻 90,000 一致的 `pts_change_point` 作為初期值來認定。因此，第 55 圖上側的 2 組 `pts_change_point` 及 `DynamicInfo()` 之中的第 1 組 `pts_change_point`，及第 55 圖下側的 3 組 `pts_change_point` 及 `DynamicInfo()` 之中的第 1 組 `pts_change_point` 會作為初期值來認定。

而且，播放器控制模組 212 會在剪輯串流檔案 "00003.PS" 的再生開始之前(在第 34 圖的步驟 S126)，按照與作為初期值而認定後的 `pts_change_point` 成組的 `DynamicInfo()` 來指示所對應之基本串流的輸出屬性。

亦即，有關以 0xE0 的 `stream_id` 所特定的視頻串流 `stream#0` 方面，在第 55 圖上側，在與初期值 90,000 的 `pts_change_point` 成組的 `DynamicInfo()` 中，`display_aspect_ratio` 會形成 '4 : 3'。此情況，播放器控制模組 212 會按照 `display_aspect_ratio` 為形成 '4 : 3'，亦即，視頻串流 `stream#0` 為 4 : 3 的寬高比的視頻資料之輸出屬性的資訊來控制繪圖處理模組 219。

又，有關以 0xBD 的 `stream_id` 及 0x00 的 `private_stream_id` 來特定的音頻串流 `stream#2` 方面，在第 55 圖下側，在與初期值 90,000 的 `pts_change_point` 成組的

DynamicInfo()中，channel_assignment 會形成 'Dual'。此情況，播放器控制模組 212 會將 channel_assignment 為形成 'Dual'，亦即音頻串流 stream#2 為雙音頻資料之輸出屬性的資訊供應給音頻輸出模組 221。

在此，於第 34 圖的步驟 S126 中，以上述初期值的 pts_change_point 作為對象之輸出屬性的控制處理會被進行。

之後，播放器控制模組 212 是將針對視頻流 stream# 之第 55 圖上側之兩個 pts_change_point 之 90,000 及 54,090,000，和針對音頻流 stream#2 之第 55 圖下側之 3 個 pts_change_point 之 90,000、27,090,000 及 32,490,000 中，初期值 90,000 以外時刻的 {27,090,000}、{32,490,000}、{54,090,000}，與「DynamicInfo()處理」之屬性時刻的主旨，同時交付給解碼控制模組 214 (步驟 320)。

解碼控制模組 214 會接受來自播放器控制模組 212 之「DynamicInfo()處理」的屬性的時刻 {27,090,000}，{32,490,000}，{54,090,000}，且於視頻串流 stream#0 及音頻串流 stream#2 的再生開始後，開始監視藉由計時部 214A 而計時的現在時刻。

然後，當現在時刻一致於「DynamicInfo()處理」的屬性的時刻 {27,090,000}，{32,490,000}，{54,090,000} 的其中之一時，解碼控制模組 214 會將與該現在時刻一致的「DynamicInfo()處理」的屬性的時刻之 DynamicInfo 時刻供應給播放器控制模組 212 (步驟 S321)。

亦即，例如，若現在時刻為形成 27,090,000，則解碼控制模組 214 會將「DynamicInfo()處理」的屬性的時刻之中與現在時刻一致的 27,090,000 作為 DynamicInfo 時刻來供應給播放器控制模組 212。

播放器控制模組 212 會接受來自解碼控制模組 214 的 DynamicInfo 時刻之 27,090,000，從有關視頻串流 stream#0 之第 55 圖上側的 2 個 pts_change_point，及有關音頻串流 stream#2 之第 55 圖下側的 3 個 pts_change_point 之中來調查一致於 DynamicInfo 時刻的 27,090,000 之 pts_change_point，且將與一致於該 27,090,000 的 pts_change_point 成組的 DynamicInfo()，亦即將有關音頻串流 stream#2 之第 55 圖下側的第 2 個 DynamicInfo() 當作處理對象 DynamicInfo() 來認定(步驟 S322)。

當處理對象 DynamicInfo() 為有關視頻串流的 DynamicInfo() 時，播放器控制模組 212 會將記述於處理對象 DynamicInfo() 的輸出屬性供應給繪圖處理模組 219 (步驟 S323)。又，當處理對象 DynamicInfo() 為有關音頻串流的 DynamicInfo() 時，播放器控制模組 212 會將記述於處理對象 DynamicInfo() 的輸出屬性供應給音頻輸出模組 221 (步驟 S323)。

一旦繪圖處理模組 219 從播放器控制模組 212 被供給輸出屬性，則會按照該輸出屬性來開始控制視頻資料的輸出(步驟 S324)。

亦即，繪圖處理模組 219，例如會根據來自播放器控

制模組 212 的輸出屬性所示之視頻資料的寬高比的指示 (`display_aspect_ratio`(第 15 圖))，及連接於第 1 圖的視頻輸出端子 120 之視頻輸出裝置的寬高比，來進行輸出至視頻輸出模組 220 之視頻資料的寬高比的變換。

具體而言，例如視頻輸出裝置的寬高比為 16：9 時，當輸出屬性之視頻資料的寬高比的指示為顯示 4：3 的寬高比時，繪圖處理模組 219 會在橫方向壓縮處理輸出至視頻輸出模組 220 的視頻資料，且於左右加入顯示黑色的資料而輸出。又，例如，視頻輸出裝置的寬高比為 4：3 時，當輸出屬性之視頻資料的寬高比的指示為顯示 16：9 的寬高比時，繪圖處理模組 219 會在縱方向壓縮處理輸出至視頻輸出模組 220 的視頻資料，且於上下加入顯示黑色的資料而輸出。又，例如，視頻輸出裝置的寬高比與輸出屬性之視頻資料的寬高比的指示所顯示的寬高比皆為同樣 4：3 或 16：9 時，繪圖處理模組 219 不會壓縮處理輸出至視頻輸出模組 220 的視頻資料，而原封不動地輸出。

在此，於第 55 圖上側，若根據針對以 0xE0 的 `stream_id` 所特定的視頻串流 `stream#0` 來記述的 2 組 `pts_change_point` 及 `DynamicInfo()`，則從視頻串流 `stream#0` 的再生開始時的時刻 90,000 到時刻 54,090,000 之前，可由視頻串流 `stream#0` 來取得 4：3 的寬高比的視頻資料。而且，在時刻 54,090,000 以後，可由視頻串流 `stream#0` 來取得 16：9 的寬高比的視頻資料。

因此，例如若連接於第 1 圖的視頻輸出端子 120 之視頻

輸出裝置的寬高比為 4 : 3，則在繪圖處理模組 219 中，從時刻 90,000 到時刻 54,090,000 之前，從視頻串流 `stream#0` 所取得的 4 : 3 的寬高比的視頻資料會原封不動供應給 4 : 3 的寬高比的視頻輸出裝置而顯示。

而且，時刻 54,090,000 以後，從視頻串流 `stream#0` 所取得的 16 : 9 的寬高比的視頻資料會在縱方向被壓縮處理，且變換成上下加入黑色成份的 4 : 3 的寬高比的視頻信號，供應給 4 : 3 的寬高比的視頻輸出裝置而顯示。

另一方面，若音頻輸出模組 221 從播放器控制模組 212 被供給輸出屬性，則會按照該輸出屬性來開始控制音頻資料的輸出(步驟 S324)。

亦即，音頻輸出模組 221 會例如根據來自播放器控制模組 212 的輸出屬性所示之音頻資料的頻道分配的指示(`channel_assignment`(第 15 圖))，及藉由使用者操作遙控器來經由輸入介面 115(第 1 圖)而從播放器控制模組 212 供給的聲音輸出模式，來處理來自音頻解碼器控制模組 217 的音頻資料，而輸出至音頻輸出端子 121(第 1 圖)。

具體而言，例如輸出屬性所示之音頻資料的頻道分配的指示，顯示左頻道為「主聲音」的音頻資料，右頻道為「副聲音」的音頻資料之雙(Dual)(雙語)模式時，音頻輸出模組 221 會按照從播放器控制模組 212 所供給的聲音輸出模式來處理來自音頻解碼器控制模組 217 的音頻資料，而輸出至音頻輸出端子 121。

亦即，聲音輸出模式，例如被指定為「主聲音」時，

音頻輸出模組 221 會將來自音頻解碼器控制模組 217 的音頻資料之中的左頻道的音頻資料當作右頻道的音頻資料來拷貝，將該左頻道及右頻道的音頻資料（「主聲音」的音頻資料）輸出至音頻輸出端子 121。又，當聲音輸出模式被指定為「副聲音」時，音頻輸出模組 221 會將來自音頻解碼器控制模組 217 的音頻資料之中的右頻道的音頻資料當作左頻道的音頻資料來拷貝，將該左頻道及右頻道的音頻資料（「副聲音」的音頻資料）輸出至音頻輸出端子 121。又，當聲音輸出模式被指定為「主·副」時，音頻輸出模組 221 會將來自音頻解碼器控制模組 217 的音頻資料原封不動地輸出至音頻輸出端子 121。

又，例如輸出屬性所示之音頻資料的頻道分配的指示為顯示立體 (Stereo) 模式時，無論從播放器控制模組 212 所供給的聲音輸出模式如何，音頻輸出模組 221 會將來自音頻解碼器控制模組 217 的音頻資料原封不動地輸出至音頻輸出端子 121。

在此，於第 55 圖下側中，若根據針對以 0xBD 的 `stream_id` 及 0x00 的 `private_stream_id` 所特定的音頻串流 `stream#2` 來記述的 3 組 `pts_change_point` 及 `DynamicInfo()`，則從音頻串流 `stream#2` 的再生開始時的時刻 90,000 到時刻 27,090,000 之前，可從音頻串流 `stream#2` 來取得雙音頻資料。又，從時刻 27,090,000 到時刻 32,490,000 之前，可由音頻串流 `stream#2` 來取得立體的音頻資料，在時刻 32,490,000 以後，可由音頻串流 `stream#2` 來取得雙音頻資

料。

因此，例如若聲音輸出模式被指定為「主聲音」，則在音頻輸出輸出模組 221 中，從時刻 90,000 到時刻 27,090,000 之前，由音頻串流 `stream#2` 所取得的雙音頻資料之中的左頻道的音頻資料會當作右頻道的音頻資料來拷貝，該左頻道及右頻道的音頻資料會被輸出至音頻輸出端子 121。

又，從時刻 27,090,000 到時刻 32,490,000 之前，由音頻串流 `stream#2` 所取得的立體音頻資料會原封不動地輸出至音頻輸出端子 121。

又，時刻 32,490,000 以後，由音頻串流 `stream#2` 所取得的雙音頻資料之中的左頻道的音頻資料會當作右頻道的音頻資料來拷貝，該左頻道及右頻道的音頻資料會輸出至音頻輸出端子 121。

如以上所述，輸出屬性的控制處理是在各剪輯串流檔案中被多重化的基本串流中，根據剪輯資訊檔案 `Clip()` (包含 0 組以上顯示該基本串流的再生時刻的 `pts_change_point` 與包含該基本串流的輸出屬性的 `DynamicInfo()` 的組合)(第 12 圖)的記述來判定再生中的基本串流的再生時刻是否一致於 `pts_change_point`。當再生中的基本串流的再生時刻一致於 `pts_change_point` 時，與該 `pts_change_point` 成組的 `DynamicInfo()` 會被認定，按照該被認定的 `DynamicInfo()` 中所含的輸出屬性來控制再生中的基本串流的輸出。因此，可按照基本串流的再生時

刻與輸出屬性來控制該基本串流的輸出。

[字幕顯示控制處理]

其次，參照第56圖的流程圖來說明有關控制對應於字幕串流的字幕資料的顯示之字幕顯示控制處理。

一旦 `PlayList()` 的再生開始，則播放器控制模組 212 會在步驟 S341 中，使對繪圖處理模組 219 之字幕資料的顯示方式之指示初期化。亦即，播放器控制模組 212 是以字幕資料的顯示方式能夠成為內定的顯示方式之方式控制繪圖處理模組 219。另外，在步驟 S341 中所進行之顯示方式的指示的初期化是對應於第30圖的 S127 所說明之顯示方式的指示的初期化。

在步驟 S341 的處理後，前進至步驟 S342，播放器控制模組 212 會判定是否有使用者操作遙控器來從輸入介面 115 針對字幕資料的顯示指示新的顯示方式。

在步驟 S342 中，當判定有新的顯示方式之指示時，前進至步驟 S343，播放器控制模組 212 會判定是否使字幕串流現在再生。

在步驟 S343 中，當判定字幕串流未再生時，回到步驟 S342。

又，在步驟 S343 中，當判定字幕串流再生時，前進至步驟 S345，播放器控制模組 212 會判定新的顯示方式之指示是否為內定的顯示方式之指示。在步驟 S343 中，當判定新的顯示方式之指示為內定的顯示方式之指示時，回

到步驟 S341，如上述，播放器控制模組 212 會以字幕資料的顯示方式能夠成為內定的顯示方式之方式來控制繪圖處理模組 219。

另一方面，在步驟 S345 中，當判定新的顯示方式之指示不為內定的顯示方式之指示時，亦即，新的顯示方式之指示，例如為擴大或縮小顯示字幕資料，或改變亮度使容易看等，非內定的顯示方式之指示時，前進至步驟 S346，播放器控制模組 212 會取得對應於現在再生的字幕串流所被多重化的剪輯串流檔案之剪輯資訊檔案 Clip()(第 12 圖)的 StaticInfo()(第 14 圖)中有關現在再生的字幕串流之 StaticInfo()，且前進至步驟 S347。

在步驟 S347 中，播放器控制模組 212 會判定在步驟 S346 所取得的 StaticInfo() 的 configurable_flag。

在步驟 S347 中，判定 configurable_flag 為形成不許可變更字幕資料的顯示方式之主旨的 0 時，前進至步驟 S348，播放器控制模組 212 會控制繪圖處理模組 219，藉此使無法變更字幕資料的顯示方式之主旨的錯誤訊息重疊於輸出視頻資料，回到步驟 S342。藉此，顯示錯誤訊息。

另一方面，在步驟 S347 中，當判定 configurable_flag 為形成許可變更字幕資料的顯示方式之主旨的 1 時，前進至步驟 S349，播放器控制模組 212 會將藉由使用者操作遙控器而從輸入介面 115 供給的新的顯示方式之指示供應給繪圖處理模組 219，前進至步驟 S350。

在步驟 S350 中，繪圖處理模組 219 會針對從字幕解碼

器控制模組 218 所供給的字幕資料，按照在之前的步驟 S349 從播放器控制模組 212 所供給的顯示方式之指示來開始進行擴大或縮小或改變亮度等的處理，回到步驟 S342。藉此，字幕資料會以按照藉由使用者操作遙控器而指示的顯示方式之顯示大小，顯示位置及顯示色等來顯示。

另一方面，在步驟 S342 中，當判定無新的顯示方式之指示時，前進至步驟 S351，播放器控制模組 212 會判定是否有進行第 40 圖所說明之 `PlayItem()` 的掉換，當判定未進行時，回到步驟 S342。

此外，在步驟 S351 中，當判定有進行 `PlayItem()` 的掉換時，回到步驟 S341，如上述，播放器控制模組 212 會以字幕資料的顯示方式能夠成為內定的顯示方式之方式來控制繪圖處理模組 219。亦即，此情況，當 `PlayItem()` 的掉換被進行時，字幕資料的顯示方式會回到內定的顯示方式。

如以上所述，在字幕顯示控制處理中，只有字幕串流的 `configurable_flag` 為形成許可顯示方式的變更之主旨的 1 時，對應於該字幕串流的字幕資料的顯示方式，例如會按照藉由使用者操作遙控器而輸入的顯示方式之指示來變更。

因此，例如若利用第 30 圖 A 及第 30 圖 B 所示的剪輯資訊檔案 "00001.CLP"，則有關所對應之剪輯串流檔案 "00001.PS" 中被多重化的 4 個基本串流中的第 3 個基本串流之字幕串流 `stream#2` 的 `configurable_flag` 是形成不許可顯

示方式的變更之主旨的 0，因此當該字幕串流 `stream#2` 被顯示時，即使使用者以能夠變更字幕的顯示之方式來操作遙控器，其顯示不會被變更。

另一方面，例如有關剪輯串流檔案 "00001.PS" 中被多重化的 4 個基本串流中的第 4 個基本串流之字幕串流 `stream#3` 的 `configurable_flag` 是形成許可顯示方式的變更之主旨的 1，因此當該字幕串流 `stream#3` 被顯示時，一旦使用者以能夠變更字幕的顯示之方式來操作遙控器，則字幕的顯示大小等會按照該操作來變更。

亦即，例如目前剪輯串流檔案 "00001.PS" 會按照第 29 圖的第 1 個 `PlayList#0` 的第 1 個 `PlayItem#0` 再生。又，如第 30 圖 A 及第 30 圖 B 中針對剪輯資訊檔案 "00001.CLP" 所說明那樣，剪輯串流檔案 "00001.PS" 中被多重化的 4 個基本串流之中的第 3 個及第 4 個為字幕串流，該第 3 個字幕串流 `stream#2` 及第 4 個字幕串流 `stream#3` 之中，例如第 3 個字幕串流 `stream#2` 會現在再生。

一旦使用者操作遙控器來輸入字幕的顯示方式之指示 (步驟 S342)，則該顯示方式的指示會從輸入介面 115 (第 1 圖) 來供應給播放器控制模組 212。一旦播放器控制模組 212 被供給顯示方式的指示，則會從剪輯資訊檔案來探出對應於再生中的字幕串流之 `StaticInfo()` (第 12 圖) (步驟 S346)。

亦即，目前的情況，再生中的字幕串流為剪輯串流檔案 "00001.PS" 中被多重化的第 3 個字幕串流 `stream#2`，播

播放器控制模組 212 會從所對應的剪輯資訊檔案 "00001.CLP" 來探出有關第 3 個字幕串流 stream#2 的 StaticInfo()。

又，播放器控制模組 212 會判定記述於第 30 圖 A 及第 30 圖 B 中有關第 3 個字幕串流 stream#2 的 StaticInfo() 之形成 0 的 configurable_flag (步驟 S347)，藉此來認定有關第 3 個字幕串流 stream#2 為未被許可變更顯示方式。

此情況，播放器控制模組 212 會判斷成再生中的字幕串流不會對應於擴大縮小等，控制繪圖處理模組 219 來使該主旨的錯誤訊息產生 (步驟 S348)，重疊於視頻資料而輸出。

另一方面，被多重化於剪輯流檔案 "00001.PS" 之 4 個基本串流的第 3 個字幕串流 stream#2，和第 4 個字幕串流 stream#3 之中，不是第 3 個字幕串流 stream#2，而是第 4 個字幕串流 stream#3 現在被再生時，藉由使用者操作遙控而接收到顯示方式之指示供給的播放器控制模組 212 會從所對應之剪輯資訊檔案 "00001.CLP" 找出有關第 4 個字幕串流 stream#3 之 StaticInfo()。

又，播放器控制模組 212 會判定記述於第 30 圖 A 及第 30 圖 B 中有關第 4 個字幕串流 stream#3 的 StaticInfo() 之形成 1 的 configurable_flag (步驟 S347)，藉此來認定有關第 4 個字幕串流 stream#3 為被許可變更顯示方式。

此情況，播放器控制模組 212 會判斷再生中的字幕串流可對應於擴大縮小等，將藉由使用者操作遙控器而供給的顯示方式之指示供應給繪圖處理模組 219 (步驟 S349)。

藉此，之後，繪圖處理模組 219 會按照來自播放器控制模組 212 之顯示方式的指示，擴大或縮小來自字幕解碼器控制模組 218 的字幕資料，重疊於來自視頻解碼器控制模組 216 的視頻資料而輸出。

另外，播放器控制模組 212 會在 `PlayList()` 的最初之 `PlayItem()` 的再生開始時，使對繪圖處理模組 219 之字幕資料的顯示方式之指示初期化 (步驟 S341)。亦即，播放器控制模組 212 會以字幕資料的顯示方式能夠成為內定的顯示方式之方式來控制繪圖處理模組 219。

又，播放器控制模組 212 在 `PlayItem()` 的掉換時也會使對繪圖處理模組 219 之字幕資料的顯示方式之指示初期化 (步驟 S341，S351)。

但是，在 `PlayItem()` 之掉換時，調查針對依照之後重新被再生之 `PlayItem()` 而被再生之新字幕流的 `configurable_flag`，`configurable_flag` 為 0 之時，初期化對繪圖處理模組 219 指示字幕資料之顯示方式，`configurable_flag` 為 1 之時，可將對繪圖處理模組 219 指示的顯示方式，維持 `PlayItem()` 掉換前的原樣。

並且，在第 56 圖之字幕顯示控制處理中，藉由使用者操作遙控，於輸入新的顯示方式之指示時，雖然使該新的顯示方式之指示能夠供給至繪圖處理模組 219 (步驟 S349)，但是顯示方式之指示是可例如記憶於構成記憶體 113 (第 1 圖) 之非揮發性記憶體中，使被記憶於該非揮發性記憶體中之顯示方式之指示能夠供給至繪圖處理模組 219。

亦即，例如，當作第1圖之碟片再生裝置之初期設定，是先在非揮發性記憶體中，記憶使用者設定之顯示方式之指示，藉由使用者操作遙控，輸入新的顯示方式之指示時，將被記憶於非揮發性記憶體中之顯示方式之指示，更新至新的顯示方式之指示，另外，可使被記憶於非揮發性記憶體之顯示方式之指示能夠供給至繪圖處理模組219。此時，非揮發性記憶體因保持上次再生完成時之顯示方式之指示，故於下次之 `PlayList()` 之再生時，即使藉由使用者操作遙控，而不輸入上次再生完成時之顯示方式之指示，亦可以該顯示方式開始字幕資料之顯示。

另外，此情況使記憶於非揮發性記憶體的顯示方式之指示中，例如含有擴大或縮小字幕資料時的擴大率或縮小率等。

如上述般，若依據字幕顯示控制處理時，則取得剪輯資訊檔案 `Clip()` (第12圖) 所包含之每基本流之該基本流再生中無變化之 `StaticInfo()` 中之字幕資料之 `StaticInfo()`，根據該 `StaticInfo()` 所包含之表示是否是否許可自預設顯示方式變更字幕資料之顯示的 `configurable_flag`，判定是否許可自預設之顯示方式變更再生中之字幕資料之顯示。然後，許可自預設之顯示方式變更再生中之字幕資料之顯示時，依照指示字幕資料之顯示方式之變更，執行該字幕資料之顯示處理，即是例如執行擴大或縮小字幕資料，或是變更顯示顏色等而予以顯示之處理。因此，可以控制字幕資料之顯示方式之變更。

[擷取控制處理]

其次，參照第57圖的流程圖來說明有關控制對應於視頻串流的視頻資料的擷取之擷取控制處理。並且，在第57圖中，亦揭示有用以說明擷取控制處理的流程圖，及用以說明2次利用藉由該擷取控制處理而擷取的視頻資料之處理的例子，亦即背景/螢幕保護處理的流程圖。

例如藉由使用者操作遙控器，指示視頻資料的擷取之擷取指示經由輸入介面115(第1圖)來供應給播放器控制模組212，則會開始進行擷取控制處理。

亦即，在擷取控制處理中，首先最初在步驟S371中，播放器控制模組212會判定視頻串流是否為再生中，當判定為非再生中時，擷取控制處理會終了。

另一方面，在步驟S371中，當判定視頻串流為再生中時，前進至步驟S372，播放器控制模組212會從對應於再生中的視頻串流之PlayList()(第7圖)來取得capture_enable_flag_PlayList，且從對應於再生中的視頻串流之剪輯資訊檔案Clip()(第12圖)來取得capture_enable_flag_Clip。

在此，PlayList()之capture_enable_flag_PlayList，如第7圖所說明那樣，顯示是否許可對應於依據該PlayList()而再生的視頻串流之視頻資料(屬於PlayList()的視頻資料)的2次利用。又，剪輯資訊檔案Clip()之capture_enable_flag_Clip，如第10圖所說明那樣，顯示是

否許可2次利用對應於該剪輯資訊檔案 Clip()的剪輯串流檔案中所儲存之對應於視頻串流的視頻資料。

步驟 S372的處理後，前進至步驟 S373，播放器控制模組 212會根據在之前的步驟 S373中所取得的 `capture_enable_flag_PlayList` 及 `capture_enable_flag_Clip` 來判定擷取指示由輸入介面 115(第1圖)輸入時所被再生的視頻資料的圖像可否擷取。

在步驟 S373中，當判定擷取指示從輸入介面 115輸入時所被再生的視頻資料的圖像不可擷取時，亦即，在之前的步驟 S373中所取得的 `capture_enable_flag_PlayList` 或 `capture_enable_flag_Clip` 的其中至少一方為形成不許可視頻資料的2次利用之主旨的0時，前進至步驟 S374，播放器控制模組 212會藉由控制繪圖處理模組 219，使視頻資料的擷取不可之主旨的錯誤訊息重疊，終了擷取控制處理。藉此，顯示錯誤訊息。

另一方面，在步驟 S373中，當判定擷取指示從輸入介面 115輸入時所被再生的視頻資料的圖像可擷取時，亦即在之前的步驟 S373中所取得的 `capture_enable_flag_PlayList` 及 `capture_enable_flag_Clip` 雙方為形成許可視頻資料的2次利用之主旨的1時，前進至步驟 S375，播放器控制模組 212會將擷取指示從輸入介面 115輸入時所被再生的視頻資料的圖像的擷取指示供應給繪圖處理模組 219，前進至步驟 S376。

在步驟 S376中，繪圖處理模組 219是依照來自播放器

控制模組 212 之擷取的指示，自視頻解碼器控制模組 216 擷取視頻資料之畫素，並使記憶於記憶體 113 (第 1 圖)，完成擷取控制處理。並且，`capture_enable_flag` 成為多數位元構成，於執行使用條件之制約時，擷取到在該時間點縮小的畫像。再者，於所使用之應用程式有制約時，同時記錄通知該主旨之旗標。

如上述般，擷取控制處理是取用對應於當使用者有指示擷取之時再生之視頻串流的 `PistList()` (第 7 圖) 和剪輯資訊檔案 `Clip()` (第 12 圖)，各個的 `capture_enable_flag_PlayList` 和 `capture_enable_flag_Clip` 的邏輯積，該邏輯積為 1 之時，即是，`capture_enable_flag_PlayList` 和 `capture_enable_flag_Clip` 中之任一者僅在成為許可 2 次利用之 1 時，判斷可 2 次利用視頻資料，執行擷取。

因此，例如，按照第 29 圖中之第 1 個 `PlayList#0` 的第 1 個 `PlayItem#0` 來進行視頻串流的再生，亦即進行被多重化於剪輯串流檔案 “00001.PS” 之視頻串流的再生時，有來自使用者之擷取指示時，第 1 個 `PlayList#0` 中之 `capture_enable_flag_PlayList` 為 1，對應於藉由該第 1 個 `PlayItem#0` 而被再生之剪輯串流檔案 “00001.PS” 之第 30 圖 A 及第 30 圖 B 之剪輯資訊檔案 “00001.CLP” 中之 `capture_enable_flag_Clip` 因是 1，故被判斷可 2 次利用再生中之視頻資料 (對應於剪輯串流檔案 “00001.PS” 中所被多重化的視頻串流之視頻資料)，進行擷取。

再者，例如，按照第29圖中之第1個 PlayList#0之第2個 PlayItem#1來進行視頻串流之再生，亦即被多重化於剪輯串流檔案“00002.PS”的視頻串流之再生時，有來自使用者之擷取指示時，因第1個 PlayList#0中之 capture_enable_flag_PlayList 為1，該對應於藉由第2個 PlayItem#1而被再生之剪輯串流檔案“00002.PS”之第30圖A及第30圖B之剪輯資訊檔案“00002.CLP”中之 capture_enable_flag_Clip 為0，故判斷不可2次利用再生中之視頻資料(對應於被多重化於剪輯串流檔案“00002.PS”的視頻串流之視頻資料)，不進行擷取。

並且，例如，按照第29圖中之第2個 PlayList#1之 PlayItem#0來進行視頻串流之再生，亦即被多重化於剪輯串流檔案“00003.PS”的視頻串流之再生時，有來自使用者之擷取指示時，第2個 PlayList#1中之 capture_enable_flag_PlayList 為0，該對應於藉由第2個 PlayItem#1之 PlayItem#0而被再生之剪輯串流檔案“00003.PS”之第30圖A及第30圖B之剪輯資訊檔案“00003.CLP”中之 capture_enable_flag_Clip 為1，故判斷不可2次利用再生中之視頻資料(對應於被多重化於剪輯串流檔案“00003.PS”的視頻串流之視頻資料)，不進行擷取。

並且，此時，在確認第2個 PlayList#1中之 capture_enable_flag_PlayList 為0之時間點，因可以判斷不可2次利用視頻資料，故可以省略對應於藉由第2個 PlayList#1之 PlayItem#0而被再生之剪輯串流檔案

”00003.PS”之第30圖 A 及第30圖 B 之剪輯資訊檔案

”00003.CLP”中之 `capture_enable_flag_Clip` 之確認。

藉由擷取控制處理來擷取，記憶於記憶體 113 的圖像，可於背景/螢幕保護處理中 2 次利用。

背景/螢幕保護處理是例如在播放器控制模組 212 動作，但基本串流的再生不進行的狀態，亦即，碟片 101 未被插入碟片驅動器 102 (第 1 圖) 的狀態，或基本串流的再生終了後的狀態時進行。

亦即，背景/螢幕保護處理，在步驟 S381 中，播放器控制模組 212 會控制繪圖處理模組 219，而使能夠藉由擷取控制處理來顯示記憶於記憶體 113 的圖像。繪圖處理模組 219 會按照來自播放器控制模組 212 的控制，藉由擷取控制處理來使記憶於記憶體 113 的圖像顯示。

在此，於繪圖處理模組 219 中，只要以靜止畫面來使記憶於記憶體 113 的圖像顯示，便可實現所謂的壁紙，只要以一定週期一邊擴大，縮小或移動，一邊顯示，便可實現螢幕保護。又，藉由擷取控制處理來進行記憶於記憶體 113 的圖像顯示之背景/螢幕保護處理並非限於播放器控制模組 212，亦可藉由其他獨立的應用來進行。

又，此刻記憶於記憶體 113 的圖像中附加有顯示使用限制的旗標時，則會按照該限制。

如上述般，取得比視頻存取單元大之單位，例如表示是否許可 2 次利用對應於 `PlayList()` 或 `PlayItem()` 之視頻資料的相對於再生中之視頻資料之

`capture_enable_flag_PlayList` 或 `capture_enable_flag_Clip`，根據該 `capture_enable_flag_PlayList` 或 `capture_enable_flag_Clip`，判定是否許可2次利用再生中之視頻資料時，擷取再生中之視頻資料，執行利用該被擷取之視頻資料之背景/螢幕保護處理。因此，可控制視頻資料之2次利用。

另外，第57圖的擷取控制處理是在 `PlayList()`(第7圖)中設置 `capture_enable_flag_PlayList`，且在對應於依據 `PlayItem()`而再生的剪輯串流檔案之剪輯資訊檔案 `Clip()`(第12圖)中設置 `capture_enable_flag_Clip`，利用該 `capture_enable_flag_PlayList` 與 `capture_enable_flag_Clip` 的雙方，判定2次利用的許可，但亦可在 `PlayList()`(第7圖)中只設置 `capture_enable_flag_PlayList`，或在對應於依據 `PlayItem()`而再生的剪輯串流檔案之剪輯資訊檔案 `Clip()`(第12圖)中只設置 `capture_enable_flag_Clip`，而只利用 `capture_enable_flag_PlayList` 或 `capture_enable_flag_Clip` 的一方來判定2次利用的可否。

又，第57圖的擷取控制處理是在步驟 S376中，繪圖處理模組219會按照來自播放器控制模組212的擷取指示，而使能夠擷取來自視頻解碼器控制模組216的視頻資料的圖像，亦即只擷取1個圖像，但除此以外亦可擷取複數個圖像。亦即，可擷取視頻解碼器控制模組216所輸出之時序的複數個圖像。此情況，一次所被擷取的圖像張數，例如可事先決定。或者亦可擴張

`capture_enable_flag_PlayList` 或 `capture_enable_flag_Clip` 的位元，而使能夠在該 `capture_enable_flag_PlayList` 或 `capture_enable_flag_Clip` 中包含一次可擷取的圖像張數的資訊。

另外，上述情況會將是否許可視頻資料的2次利用之利用許可資訊(`capture_enable_flag_PlayList`，`capture_enable_flag_Clip`)記述於 `PlayList()`或剪輯資訊檔案 `Clip()`，根據該利用許可資訊來判定有關依據 `PlayList()`而再生的視頻資料全體，或對應於剪輯串流檔案(該剪輯串流檔案對應於剪輯資訊檔案 `Clip()`)中被多重化的視頻串流的視頻資料全體之2次利用的可否，但利用許可資訊可針對其他任意單位的視頻資料記述，根據該利用許可資訊來判定有關任意單位的視頻資料之2次利用的可否。

亦即，第58圖是表示配置有利用許可資訊之 `private_stream2_PES_payload()`的語法，第59圖是表示配置有利用許可資訊之 `au_information()`的語法。

另外，第58圖的 `private_stream2_PES_payload()`除了在 `video_stream_id` 之前配置有利用許可資訊的 `capture_enable_flag_ps2`以外，其餘則與第26圖的情況時同樣構成。第59圖的 `au_information()`也是除了在 `pic_struct_copy` 之前配置有利用許可資訊的 `capture_enable_flag_AU` 以外，其餘則與第24圖的情況時同樣構成。

第 58 圖的 `private_stream2_PES_payload()` 中所配置的 `capture_enable_flag_ps2` 是表示是否許可對應於從包含該該 `private_stream2_PES_payload()` 的 `private_stream_2` 的 `PES_packet()` 到其次的 `private_stream_2` 的 `PES_packet()` 之前所被配置的視頻串流之視頻資料的 2 次利用。因此，若利用第 58 圖的 `private_stream2_PES_payload()` 中所配置的 `capture_enable_flag_ps2`，則可針對從某解碼開始可能點到其次的解碼開始可能點之間的視頻資料來判定是否許可其 2 次利用。

又，第 59 圖的 `au_information()` 中所配置的 `capture_enable_flag_AU` 是表示是否許可對應於該 `capture_enable_flag_AU` 的視頻存取單元之視頻資料的 2 次利用。因此，若利用第 59 圖的 `au_information()` 中所配置的 `capture_enable_flag_AU`，則可針對視頻存取單元單位的視頻資料亦即圖像單位來判定是否許可其 2 次利用。

在此，`PlayList()`(第 7 圖)之利用許可資訊的 `capture_enable_flag_PlayList`，剪輯資訊檔案 `Clip()`(第 12 圖)之利用許可資訊的 `capture_enable_flag_Clip`，`private_stream2_PES_payload()`(第 58 圖)之利用許可資訊的 `capture_enable_flag_ps2`，及 `au_information()`(第 59 圖)之利用許可資訊的 `capture_enable_flag_AU`，可重複採用其中的 2 個以上，此情況，某視頻資料的圖像之 2 次利用的可否，可根據其重複採用的 2 個以上的利用許可資訊的邏輯積等來判定。

又，包含配置有第59圖的 `au_information()` 的第26圖或第58圖的 `private_stream2_PES_payload()` 之 `private_stream_2` 的 `PES_packet()`，如第45圖的步驟 S211 所說明那樣，緩衝控制模組 215 (第5圖) 的視頻讀出功能部 233 會從記憶於緩衝器 215A 的程式串流中搜尋。因此，採用配置有 `capture_enable_flag_ps2` 的第58圖的 `private_stream2_PES_payload()`，或配置有 `capture_enable_flag_AU` 的第59圖的 `au_information()` 時，播放器控制模組 212 在判定視頻資料的2次利用的可否時，必須對視頻讀出功能部 233 詢問 `capture_enable_flag_ps2` 或 `capture_enable_flag_AU`。

其次、參照第60圖來說明有關碟片記錄裝置的硬體構成。

第60圖的碟片記錄裝置是例如可適用於碟片播放器或遊戲裝置、汽車衛星導航系統及其他。

在第60圖的碟片記錄裝置中，碟片 410 例如為 DVD 等的光碟片、或光磁碟片、磁碟片等，可記錄視頻資料、音頻資料、字幕資料等的內容資料。又，碟片 410 可藉由記錄各種的資料來作為第1圖的碟片 101 使用。

視頻輸入端子 400A 是被連接於未圖示之攝影裝置等的視頻輸入裝置，將藉由視頻輸入裝置所供給的視頻資料供應給視頻輸入介面 401。音頻輸入端子 400B 是被連接於未圖示之麥克風或擴大器等音頻輸入裝置，將所被輸入的音頻資料供應給音頻輸入介面 402。

視頻輸入介面 401 會對所被輸入的視頻資料施以必要的處理，經由匯流排 411 來供應給視頻編碼器 403。音頻輸入介面 402 會對所被輸入的音頻資料施以所定的處理，經由匯流排 411 來供應給音頻編碼器 404。

視頻編碼器 403 是在於編碼藉由 CPU405 或視頻輸入介面 401 所供給的視頻資料，將其結果所取得之壓縮編碼資料（編碼視頻資料：例如，MPEG2 視頻串流）經由匯流排 411 利用碟片驅動器 409 來使記錄於碟片 410。

音頻編碼器 404 是在於編碼藉由 CPU405 或音頻輸入介面 402 所供給的音頻資料，將其結果所取得之壓縮編碼資料（編碼音頻資料：例如，MPEG2 音頻串流）經由匯流排 411 利用碟片驅動器 409 來使記錄於碟片 410。

CPU405 及記憶體 406 係形成電腦系統。亦即，CPU405 係執行記憶於記憶體 406 的程式，控制碟片記錄裝置全體，且進行後述的各種處理。記憶體 406 係記憶 CPU405 所執行的程式。並且，記憶體 406 係一時記憶 CPU405 的動作上必要的資料。又，記憶體 406 可僅以非揮發性記憶體來構成，或揮發性記憶體及非揮發性記憶體的組合。當在第 60 圖的碟片記錄裝置中設置硬碟，事先在該硬碟中記錄（安裝）CPU405 所執行的程式時，記憶體 406 可僅以揮發性記憶體來構成。

在此，CPU405 所執行的程式可預先儲存於碟片記錄裝置中所內藏的記錄媒體，即記憶體 406。

或，程式可暫時性或永久性事先儲存於碟片驅動器

409、碟片驅動器 409 以外的軟碟、CD-ROM(Compact Disc Read Only Memory)，MO(Magneto Optical)碟片、磁碟片、記憶卡等的可攜式記錄媒體。如此的可攜式記錄媒體可作為所謂的封裝軟體來提供。

又，程式可預先使記憶於記憶體 406，或可由上述的可攜式記錄媒體來安裝於碟片記錄裝置。又，程式可從下載網站經由數位衛星播放用的人工衛星來以無線傳送至碟片再生裝置，或經由 LAN(Local Area Network)，網際網路，以有線來傳送至碟片再生裝置，在碟片再生裝置中，使用輸入介面 408 來接受如此傳送來的程式，安裝於所內藏的記憶體 406。

又，程式可藉由 1 個 CPU 來處理，或藉由複數個 CPU 來分散處理。

驅動器介面 407 是在 CPU405 的控制下，控制碟片驅動器 409，藉此，經由匯流排 411 來對碟片驅動器 409 供給由 CPU405 或記憶體 406、視頻編碼器 403、音頻編碼器 404 所供給的資料，使記憶於碟片 410，或讀出記錄於碟片 410 的資料，經由匯流排 411 來供應給 CPU405 或記憶體 406。

輸入介面 408 會接受藉由使用者操作未圖示的鍵(按鈕)或遙控器而供給的信號，經由匯流排 411 來供應給 CPU405。另外，輸入介面 408 除此以外，例如亦可具有作為數據機(包含 ADSL(Asymmetric Digital Subscriber Line)數據機)或 NIC(Network Interface Card)等的通信介面功能。

又，來自視頻輸入裝置及音頻輸入裝置之往碟片記錄裝置的視頻資料及音頻資料的供給，可藉由有線或無線來進行。

碟片 410 可裝卸於碟片驅動器 409。碟片驅動器 409 係內藏未圖示的介面，經由該介面來連接至驅動器介面 407。碟片驅動器 409 會驅動該處所裝著的碟片 410，按照來自驅動器介面 407 的記錄等的命令，進行在碟片 410 中記錄資料等的處理。

另外，在記錄於碟片 410 的資料(記錄資料)中，可因應所需，含電腦所能執行的程式。在本實施形態中，記錄媒體為採用碟片狀的記錄媒體之碟片 410，但除此以外，例如亦可採用半導體記憶體或帶狀的記錄媒體。

在匯流排 411 連接有 CPU(Central Processing Unit)405、記憶體 406、驅動器介面 407、輸入介面 408、視頻編碼器 403、音頻編碼器 404、視頻輸入介面 401、音頻輸入介面 402。

其次，參照第 61 圖來說明有關藉由本發明的資料編碼方法之碟片記錄裝置所實現的功能。在藉由該圖所示的碟片記錄裝置來實現的功能中，音頻編碼器 404 係將經由音頻輸入端子 400B 及音頻輸入介面 402 所輸入的音頻信號予以壓縮編碼，而輸出至多重化器 421。

又，視頻編碼器 403 係將經由視頻輸入端子 400A 及視頻輸入介面 401 所輸入的視頻信號予以壓縮編碼，而輸出至多重化器 421。

多重化器 421 係將所被輸入的 MPEG2 視頻串流及 MPEG2 音頻串流予以封包化，如利用上述第 18 圖 A 及第 18 圖 B～第 27 圖來說明那樣予以時分割多重化。多重化器 421 係選擇串流中的幀內畫面，以約 1 秒 2 次的頻率來插入第 26 圖所示的 `private_stream_2` 的 `PES_packet()`。

多重化器 421 係將多重化串流經由 FIFO 422 來輸出至 RAPI 重寫器 424，且 RAPI 資訊抽出器 423 亦同時輸出。在 RAPI 資訊抽出器 423 檢測出多重化串流中所被多重化的視頻串流之中的 `private_stream_2` 的 `PES_packet()` 的開始位置、及緊接於 `private_stream_2` 的 `PES_packet()` 之後所被放置的幀內畫面所持有的顯示時間標籤 (PTS) 值、及該幀內畫面、以及接續的第 2 張、第 3 張、第 4 張的參照畫像的終了位置，且予以記憶。

在此所謂的 RAPI 為 `private_stream_2` 的 `PES_packet()`。

RAPI 資訊抽出器 423 係將檢測出的該當幀內畫面、及接續的第 2 張、第 3 張、第 4 張的參照畫像的終了位置輸出至 RAPI 重寫器 424。在 RAPI 重寫器 424 中，RAPI 的資訊，即第 26 圖的 `1stRef_picture`、`2ndRef_picture`、`3rdRef_picture`、`4thRef_picture` 的場 (field) 會被覆寫，各前頭、第 2 張、第 3 張、第 4 張的參照畫像的終了位置會以區段單位的數值來記錄，儲存於輸出伺服器 426。

一旦多重化串流全體的處理終了，則控制器 425 係取得在 RAPI 資訊抽出器 423 所被抽出、儲存之多重化串流

中所被多重化的所有 RAPI 的開始位置、及緊接於 RAPI 之後所被放置的幀內畫面及接續的第2張、第3張、第4張的參照畫像的終了位置。

控制器425係使用所被輸入的資訊，作成利用第16圖來說明的 EP_map()。

控制器425係由自 RAPI 資訊抽出器423所輸入的資料來對 RAPI 選出其位址緊接著之後的幀內畫面所持有的 PTS，以及幀內畫面及接續的第2張、第3張、第4張的參照畫像的終了位置的其中之一，而作成剪輯資訊檔案內的 EP_map()，儲存於輸出伺服器426。

在此，參照第62圖的流程圖來說明有關 EP_map()生成處理。

在步驟 S381中，視頻編碼器403係將經由視頻輸入端子400A 及視頻輸入介面401而輸入的視頻信號予以壓縮編碼，然後輸出至多重化器421。又，音頻編碼器404係將經由音頻輸入端子400B 及音頻輸入介面402而輸入的音頻信號予以壓縮編碼，然後輸出至多重化器421。此情況，自音頻編碼器404所輸出的串流係成為 MPEG2音頻串流，自視頻編碼器403所輸出的串流係此成為 MPEG2視頻串流。

在步驟 S382中，多重化器421係使所被輸入的 MPEG2 視頻串流及 MPEG2音頻串流封包化，而如第18圖 A 及第18圖 B～第27圖所述一般時分割多重化，選擇串流中的幀內畫面，以約1秒2次的頻率來插入第26圖所示之 private_stream_2的 PES_packet()。在此 private_stream_2

的 PES_packet()是表示即使不參照其他的圖像也能夠解碼的視頻的幀內畫面會緊接於之後放置。此刻該當幀內畫面中必須賦予顯示時間標籤(PTS/DTS)。

另外，在此階段，第26圖所述之 1stRef_picture、2ndRef_picture、3rdRef_picture、4thRef_picture 的場中未記錄有資料。又，雖未圖示，但亦可將副標題串流輸入至多重化器421，而與視頻串流、音頻串流一同多重化。

在步驟 S383中，多重化器421係將多重化串流經由FIFO422來輸出至 RAPI 重寫器424，且 RAPI 資訊抽出器423亦同時輸出。在 RAPI 資訊抽出器423係檢測出多重化串流中所被多重化的視頻串流之中的 private_stream_2的 PES_packet()的開始位置、及緊接於 private_stream_2的 PES_packet()之後所被放置的幀內畫面所持有顯示時間標籤(PTS)值、及該當幀內畫面、以及接續的第2張、第3張、第4張的參照畫像的終了位置，且予以記憶。

又，RAPI 資訊抽出器423係將所檢測出的該當幀內畫面、及接續的第2張、第3張、第4張的參照畫像的終了位置輸出至 RAPI 重寫器424。RAPI 重寫器424係覆寫 RAPI 的資訊中第26圖的 1stRef_picture、2ndRef_picture、3rdRef_picture、4thRef_picture 的場，且以區段單位的數值來分別記錄前頭、第2張、第3張、第4張的參照畫像的終了位置，使儲存於輸出伺服器426。

在步驟 S385中，在 RAPI 資訊抽出器423所被抽出、儲存之多重化串流中所被多重化的所有 RAPI 的開始位置

、及緊接於 RAPI 之後所被放置的幀內畫面及接續的第 2 張、第 3 張、第 4 枚張的參照畫像的終了位置會被輸入至控制器 425。

控制器 425 係使用所被輸入的資訊，作成利用第 16 圖來說明的 EP_map()。在此所作成的 EP_map() 為僅含視頻串流的資訊。有關視頻的 EP_map() 是被放置於串流中的所有 RAPI、亦即 private_stream_2 的 PES_packet() 的位置，此資訊是由 RAPI 資訊抽出器 423 輸出至控制器 425 的資訊所作成。

更詳而言之，控制器 425 係由自 RAPI 資訊抽出器 423 所輸入的資料來對 RAPI 選出其位址緊接著之後的幀內畫面所持有的 PTS、以及幀內畫面及接續的第 2 張、第 3 張、第 4 張的參照畫像的終了位置的其中之一，作成剪輯資訊檔案內的 EP_map()，儲存於輸出伺服器 426。亦即，控制器 425 係將 4 點的參照畫像終了位置 (1stRef_Picture、2ndRef_Picture、3rdRef_Picture、4thRef_Picture) 的其中接近所定的區段數 (在編碼處理中可集中讀出的的區段數) 的值予以拷貝成 N-th_Ref_picture_copy。

在步驟 S386 中，控制器 425 係根據 N-th_Ref_picture_copy 來決定、記錄 index_minus1，使記錄於碟片 410。在此儲存於輸出伺服器 426 的串流資料及資料基礎系的檔案是由驅動器介面 407 來供給至碟片驅動器 409，記錄於碟片 410。

藉由以上的處理，如第 31 圖所示，EP_map 會被生成

。

。在此，快進處理是根據藉由參照第64圖的流程圖所說明的進入點選定處理(步驟S395)而選定的進入點來僅解碼參照畫像而顯示，但在解碼控制模組214中會預先被傳送 `index_N_minus1`，解碼控制模組214係僅解碼所被指定數量的參照畫像，依次傳送至後段而使顯示(步驟S396)。

一旦顯示終了，則播放器控制模組212會選擇其次顯示的 `EP_map()` 進入點，重複以上說明，而來輸出快進畫像(步驟S397)。

說明此刻在選擇其次顯示的 `EP_map()` 內的跳躍端的進入點時，使用 `index_N_minus1` 的方法。如既已說明那樣，`index_N_minus1` 係至 `N-th_Ref_Picture_copy` 為止讀入時，顯示其資料中所含的參照畫像的張數。就第31圖的資料例而言，在第1個及第3個的進入點，由於 `index_N_minus1` 為0，因此幀內畫面1張，在第2個及第4個的進入點，由於 `index_N_minus1` 為3，因此含有4張的參照畫像。

快進畫像係藉由輸出2張以上的參照畫像，而具有其主觀品質會提升的性質。但，爲了輸出較多的參照畫像，而比須要有更多的資料讀出量，更新的頻率會變慢。因此權衡關係，播放器控制模組212會在選擇其次顯示的 `EP_map()` 內的進入點時，評價 `index_N_minus1` 的值。

亦即，當快進的速度快時，雖 `EP_map()` 內的進入點的選拔間隔會變大，但此刻會優先選擇 `index_N_minus1` 較大者(亦即，主觀畫質高者)。並且，當快進的速度慢時

，`index_N_minus1`的較小者亦選擇。

在上述所說明的 `index_N_minus1` 決定的算法中是選擇 `N-th_Ref_picture_copy` 的值接近 "30" 者。亦即，在如此的算法下所作成的 `EP_map()` 是在按照 `N-th_Ref_picture_copy` 讀出時，資料讀出量大致形成 "30" 區段。當資料讀出速度為支配性時，因為一定的讀出時間重要，所以如此選擇進入點的方法亦有效。

其次，參照第 64 圖的流程圖來說明有關快進時的進入點的選擇處理。

播放器控制模組 212 係於步驟 S401 中，判定快進的模式是否為高速模式或低速模式。當判定快進為高速模式時，處理會前進至步驟 S402，當判定為低速模式時，處理會前進至步驟 S411。

<低速模式快進時的選擇說明>

在步驟 S402 中，因為快進為低速模式，所以播放器控制模組 212 係使所選擇的進入點的號碼（該當記入項 (N)）從前次的該當記入項僅增加 2（該當記入項號碼 += 2）。然後，在步驟 S403 中，播放器控制模組 212 係由該當記入項 N、記入項 (N-1)、記入項 (N+1) 來讀出 `index_N_minus1`。

在步驟 S404 中，播放器控制模組 212 係判斷 `index_N_minus(N)`，亦即第 N 個的進入點所持有之 `index_N_minus1` 的值是否為 0 或 1。在步驟 S404 中，例如 `index_N_minus1` 的值為 0 或 1 時，處理會前進至步驟 S405

，播放器控制模組 212 係選定第 N 個的進入點，而終了處理。另一方面，在步驟 S404 中，當 $index_N_minus1$ 的值不是 0 也不是 1 時，處理會前進至步驟 S406。

在步驟 S406 中，播放器控制模組 212 係判定 $inde_N_minus(N+1)$ ，亦即第 $(N+1)$ 個的進入點所持有的 $index_N_minus1$ 的值是否為 0 或 1。在步驟 S406 中，當 $inde_N_minus(N+1)$ 為 0 或 1 時，處理會前進至步驟 S407，播放器控制模組 212 係選定第 $(N+1)$ 個的進入點，而終了處理。另一方面，在步驟 S406 中，當 $inde_N_minus(N+1)$ 不是 0 也不是 1 時，處理會前進至步驟 S408。

在步驟 S408 中，播放器控制模組 212 係判定 $inde_N_minus(N-1)$ ，亦即第 $(N-1)$ 個的進入點所持有的 $index_N_minus1$ 的值是否為 0 或 1。在步驟 S408 中，當 $index_N_minus1$ 的值為 0 或 1 時，處理會前進至步驟 S409，播放器控制模組 212 係選定第 $(N-1)$ 個的進入點，而終了處理。當不是 0 也不是 1 時，則前進至步驟 S410。

在步驟 S410 中，由於判定在記入項 N 、 $(N+1)$ 、 $(N-1)$ 的所有進入點中 $index_N_minus1$ 不是 0 也不是 1，因此播放器控制模組 212 係選定第 N 號的進入點，而終了處理。

<高速模式快進時的選擇說明>

在步驟 S411 中，由於快進為高速模式，因此播放器控制模組 212 係使所選擇的進入點的號碼(該當記入項)(N)從前次的該當進入點僅增加 5。(該當記入項號碼 $+=5$)。然

後，在步驟 S412中，播放器控制模組 212係由該當記入項 N、記入項 (N-1)、記入項 (N+1)來讀取 `index_N_minus1`。

在步驟 S413中，播放器控制模組 212係判定 `inde_N_minus(N)`，亦即第 N 個的進入點所持有的 `index_N_minus1` 的值是否為 3 或 2。在步驟 S413中，當第 N 號的進入點所持有的 `index_N_minus1` 的值為 3 或 2 時，處理會前進至步驟 S414，播放器控制模組 212係選定第 N 個的進入點，而終了處理。在步驟 S413中，當第 N 個的記入項所持有的 `index_N_minus1` 的值不是 3 也不是 2 時，處理會前進至步驟 S415。

在步驟 S415 中，播放器控制模組 212 係判定 `inde_N_minus(N+1)`，亦即第 (N+1) 個的進入點所持有的 `index_N_minus1` 的值是否為 3 或 2。在步驟 S415 中，當第 (N+1) 個的進入點所持有的 `index_N_minus1` 的值為 3 或 2 時，處理會前進至步驟 S416，播放器控制模組 212 係選定第 (N+1) 個的進入點，而終了處理。另一方面，在步驟 S415 中，當第 (N+1) 個的進入點所持有的 `index_N_minus1` 的值不是 3 也不是 2 時，處理會前進至步驟 S417。

在步驟 S417 中，播放器控制模組 212 係判定 `inde_N_minus(N-1)`，亦即第 (N-1) 個的進入點所持有 `index_N_minus1` 的值是否為 3 或 2。在步驟 S417 中，當第 (N-1) 個的進入點所持有的 `index_N_minus1` 的值為 3 或 2 時，處理會前進至步驟 S418，播放器控制模組 212 係選定第 (N-1) 個的進入點，而終了處理。當不是 3 也不是 2 時，則

裝置的功能之功能方塊圖。另外，在第65圖的碟片記錄裝置中，針對與第61圖的碟片記錄裝置的功能相同的部份賦予同樣的符號，且適當地省略其說明。

Subtitle 解碼器443是藉由字幕素材伺服器442來讀取字幕的素材，予以壓縮編碼，而寫入字幕資料伺服器444。字幕資料是與視頻資料或音頻資料相異，在時間軸上間歇性地存在。因此，記錄於字幕素材伺服器442的字幕素材及記錄於字幕資料伺服器444的字幕資料的各個顯示開始時刻及顯示 duration 是被準備於字幕時序資訊445中。

多重化器441是具有與第61圖的多重化器421基本同樣的功能，但更使字幕資料及字幕時序資訊多重化。亦即，多重化器441會讀入所被輸入的 MPEG2視頻串流與 MPEG2音頻串流、及自字幕資料伺服器444所供給的字幕資料與字幕時序資訊445，如利用上述第18圖 A 及第18圖 B～第27圖來說明那樣進行時分割多重化。

多重化器441係選擇串流中的幀內畫面，以約1秒2次的頻率來插入第23圖所示之 private_stream_2的 PES_packet()。在此 private_stream_2的 PES_packet()是表示即使不參照其他的圖像也能夠解碼的視頻的幀內畫面會緊接於之後放置。此刻該當幀內畫面中必須賦予顯示時間標籤(PTS/DTS)。

並且，在字幕資料的所有存取單元中必須賦予顯示時間標籤。

另外，在此階段，第26圖所述之 1stRef_picture、

2ndRef_picture、3rdRef_picture、4thRef_picture 的場中未記錄有資料。又，雖未圖示，但亦可將副標題串流輸入至多重化器 441，而與視頻串流、音頻串流一同多重化。

多重化器 441 是經由 FIFO 422 來將多重化串流供應給 RAPI 重寫器 424，且亦同時供應給 RAPI 資訊抽出器 423。RAPI 資訊抽出器 423 係抽出多重化串流中所被多重化的視頻串流的資訊及字幕串流的資訊，予以記憶。亦即，有關視頻串流方面，RAPI 資訊抽出器 423 係檢測出 private_stream_2 的 PES_packet() 的開始位置、及緊接於 private_stream_2 的 PES_packet() 之後所被放置的幀內畫面所持有顯示時間標籤 (PTS) 值、及該當幀內畫面、以及接續的第 2 張、第 3 張、第 4 張的參照畫像的終了位置，且予以記憶。並且，有關字幕串流方面，檢測出所有字幕存取單元的開始位置及顯示時間標籤，且予以記憶。

控制器 446 係使用所被輸入的資訊，例如作成第 66 圖所示的 EP_map()。在此作成的 EP_map() 為含有關於視頻串流及字幕串流的資訊。關於視頻的 EP_map() 的主要資訊是串流中所被放置的所有 RAPI，亦即 private_stream_2 的 PES_packet() 的位置及緊接著之後的幀內畫面的顯示時間標籤，該等的資訊可以自 RAPI 資訊抽出器 423 來輸入至控制器 446 的資訊所作成。又，有關字幕的 EP_map() 的主要資訊是字幕存取單元的位置及顯示時間標籤，該等的資訊亦可以自 RAPI 資訊抽出器 423 來輸入至控制器 446 的資訊所作成。

控制器 446 係由自 RAPI 資訊抽出器 423 所輸入的資料來作成 EP_map() 的資訊中尚未確定之後述的 priority_flag。亦即，針對所有視頻的進入點 (RAPI 的進入點) 及字幕存取單元來評價 (後述) 其顯示時間標籤，設定 priority_flag。因應於此，在控制器 446 中會被輸入章節・場景變化資訊 447。

[EP_map 的說明]

在此，參照第 66 圖來說明有關對檔案設定優先次序時的 EP_map。如第 66 圖所示，在 number_of_EP_entries 之後，以之前的 stream_id 及 private_stream_id 所特定的基本串流的解碼開始可能點的資訊之 priority_flag (2 位元) 及 reserved_for_future_use (14 位元)、以及 PTS_EP_start (32 位元) 及 RPN_EP_start (32 位元) 的組合會僅以 number_of_EP_entries 所示的數量重複配置。

priority_flag 係具有第 67 圖所示的意義。亦即，在視頻串流的記入項，當 priority_flag 的值為 3 時，表示該記入項會對應於章節前頭。又，當 priority_flag 的值為 2 時，表示上述以外相當於每隔 1 分的重要場景變化之記入項。又，當 priority_flag 的值為 1 時，表示上述以外相當於場景變化之每隔 3 秒的記入項。又，除此以外的記入項是所有 priority_flag 的值為 0。

對應於 Subtitle 串流的記入項也是同樣地，當 priority_flag 的值為 3 時，表示該記入項會對應於章節前

頭。又，當 `priority_flag` 的值為 2 時，表示上述以外相當於重要的場景變化之記入項。又，當 `priority_flag` 的值為 1 時，表示上述以外相當於場景變化之記入項。又，除此以外的記入項是所有 `priority_flag` 的值為 0。

例如，若此剪輯為 2 小時的電影，存在 1 秒 2 次的隨機存取點，則記入項數是全體形成 $14400 (= 2 \text{ 小時} \times 3600 \text{ 秒} \times 2 \text{ 次})$ 。又，當章節數約為數十個時，`priority_flag=3` 的記入項是形成數十個的同數。由於重要的場景變化 (`priority_flag=2`) 及其他的場景變化 (`priority_flag=1`) 數量是依內容而定，因此不能以一概全，但 `priority_flag=3` 或 2 的記入項約為 200，`priority_flag=3` 或 2 或 1 的記入項約為 2400，若包含其他所有，則全部形成 14400，在此例如 `priority_flag=2` 及 `=1` 者總計為 1000。此情況，若僅讀入 `priority_flag=3`，2，1 的記入項，則與讀入全部時相較之下，約形成 $1000/14,400 = 14 \text{ 分之 } 1$ 的記憶體量。又，由於此例一個記入項為 10 位元組，因此對一個視頻串流而言，可節約 $10 \text{ 位元組} \times (14400 - 1000) = 120134 \text{ 千位元組}$ 的記憶體。

又，有關字幕方面，對 2 小時的電影而言，存在 1000 ~ 2000 的字幕，針對於此，由於章節為數十個，因此若僅讀入 `priority_flag=3` 的記入項，則可節約數十分之一的記憶體 (1000 或 2000 除以數十個)。由於字幕串流的數量比視頻串流更多，因此削減效果亦充分。

另外，在此例中，雖是以 3，2，1，0 的值來表示旗標

，但亦可以位元來分別表現，將所對應的位元設定為1之方法。亦即，將此場(field)當作3位元，最上位位元為1時，表示章節前頭，其次的位元為1時，表示每隔1分的記入項，最下位位元為1時，表示每隔5秒的記入項，若所有的位元為0，則定義為不含於該等三個的範疇。

在對應於 Subtitle 串流的記入項中，當 priority_flag 的值為1時，表示該記入項為對應於章節前頭。在除此以外的記入項所有 priority_flag 的值為0。

其次，參照第68圖的流程圖來說明有關 priority_flag 設定處理。

在步驟 S441中，控制器446係針對視頻的記入項來判定是否為視頻的章節的前頭位置，亦即評價中的記入項是否為相當於章節・場景變化資訊447中所具有的章節的時刻。在此定義差分為0。例如，相當於章節的時刻時，在步驟 S442中，控制器446係設定為 priority_flag=3，而使記憶於輸出伺服器426。

另一方面，在步驟 S441中，當判定為不是視頻的章節的前頭位置時，在步驟 S443中，控制器446會判定是否為視頻的重要的場景變化的位置，亦即其次評價中的記入項是否為相當於從章節・場景變化資訊447中所具有的「重要的場景變化」前頭起每隔1分的位置。在步驟 S443中，例如相當於從「重要的場景變化」前頭起每隔1分的位置時，在步驟 S444中，控制器446係設定成 priority_flag=2。

在步驟 S443中，當判定為不是重要的場景變化時，在步驟 S445中，控制器 446係判定是否為視頻的通常場景變化，亦即其次評價中的記入項是否為相當於從章節・場景變化資訊 447中所具有的「場景變化」的前頭起每隔3秒的位置。在步驟 S445中，例如相當於從「場景變化」的前頭起每隔3秒的位置時，在步驟 S446中，控制器 446係設定成 `priority_flag=1`。

在步驟 S445中，當判定為不是通常的場景變化時，亦即判定為皆非相當於任何的場景變化時，在步驟 S447中，控制器 446係設定成 `priority_flag=0`。

在步驟 S448中，控制器 446係判定有關所有視頻的全部記入項是否處理終了，當判定為未終了時，處理會回到步驟 S441，重複以後的處理。亦即，至有關視頻的記入項所有的 `priority_flag` 被設定為止，重複步驟 S441～S448的處理。

然後，在步驟 S448中，當判定所有的處理終了時，亦即判定有關視頻的所有記入項的處理終了時，該處理會前進至步驟 S449。

在步驟 S449中，控制器 446係判定有關字幕方面，記入項是否為相當於章節前頭位置，亦即評價中的記入項是否為相當於章節・場景變化資訊 447中所具有的章節的時刻。在步驟 S449中，記入項為相當於章節的時刻時，在步驟 S450中，控制器 446係設定成 `priority_flag=3`。

另一方面，在步驟 S449中，當判定不是字幕的章節

的前頭位置時，在步驟 S451中，控制器 446係判定是否為字幕的重要場景變化的位置，亦即其次評價中的記入項是否為相當於從章節・場景變化資訊 447中所具有的「重要的場景變化」前頭起每隔 1分的位置。在步驟 S451中，例如相當於從「重要的場景變化」前頭起每隔 1分的位置時，在步驟 S452中，控制器 446係設定成 `priority_flag=2`。

在步驟 S451中，當判定為不是重要的場景變化時，在步驟 S453中，控制器 446或判定是否為字幕的通常場景變化，亦即其次評價中的記入項是否為相當於從章節・場景變化資訊 447中所具有的「場景變化」的前頭起每隔 3秒的位置。在步驟 S453中，例如從「場景變化」的前頭起每隔 3秒的位置時，在步驟 S454中，控制器 446係設定成 `priority_flag=1`。

在步驟 S453中，當判定為不是通常的場景變化時，亦即皆非相當於任何的場景變化時，在步驟 S455中，控制器 446係設定成 `priority_flag=0`。

在步驟 S456中，控制器 446會判定有關字幕的所有記入項是否處理終了，當判定未終了時，處理會回到步驟 S449，重複以後的處理。亦即，至有關字幕的記入項所有的 `priority_flag` 被設定為止，重複步驟 S449～S456的處理，在步驟 S456中，若判定為有關字幕的所有記入項處理終了，則控制器 446會對輸出伺服器 426輸出按照第 66圖的構文之 `EP_map()`的資料。

[再生側的動作：EP_map()的選拔]

根據以上那樣設定的 `priority_flag`，碟片再生裝置係依據記憶體（例如、第1圖的記憶體113）的大小來選拔 `EP_map()`。亦即，在爲了降低成本而減少功能的碟片再生裝置中，僅使具有高值的 `priority_flag` 之記入項留在記憶體。當然，在記憶體的容量爲確保能夠儘可能保持 `EP_map()` 全體的機器中，並非一定要進行如此的作業。

例如，在第33圖的流程圖之步驟 S106的處理中，有關視頻方面是將 `priority_flag=1` 以上的記入項保持於記憶體，有關字幕方面是將 `priority_flag=1` 以上的記入項保持於記憶體。此情況，在 `EP_map()` 的讀出時，依據 `stream_id` 及 `private_stream_id` 的值，播放器控制模組 212 係串流的種類爲視頻時，`priority_flag` 的值爲 3，2 或 1 的記入項會讀入記憶體，`priority_flag` 的值爲 0 的記入項不會被讀入記憶體。又，當串流的種類爲字幕時，`prioirty_flag=3` 及 21 的記入項會讀入，`priority_flag` 的值爲 1，0 的記入項會捨讀。

藉由以上的處理，針對一個視頻串流之 `EP_map` 所必要的記憶體容量可爲 $1/6 \sim 1/10$ 程度。又，針對一個字幕串流之 `EP_map` 所必要的記憶體量可約爲數十分之一程度。其結果，即使低成本的碟片再生裝置，照樣可以按照記憶體的容量來使記入項記憶，可實現有效率的再生處理。

另外，在此雖 `priority_flag=3` 爲意指章節前頭，但可定義成章節前頭及重要的場景變化點等的任意意思。

又，本實施形態中，雖是藉由軟體來進行上述一連串
的處理，但上述一連串的處理亦可藉由專用的硬體來進行
。

又，本實施形態中，視頻解碼器 116(第 1 圖)及視頻編
碼器 403(第 60 圖)為採用硬體解碼器，但視頻解碼器 116 及
視頻編碼器 403(第 60 圖)亦可採用軟體解碼器。有關音頻
解碼器 117(第 1 圖)及音頻編碼器 404(第 60 圖)亦相同樣。

又，本實施形態中，字幕解碼器為採用軟體解碼器，
但字幕解碼器亦可採用硬體解碼器。

【圖式簡單說明】

第 1 圖是表示適用本發明的碟片裝置之一實施形態的
硬體構成例的方塊圖。

第 2 圖 A 及第 2 圖 B 是表示 CPU112 所執行的軟體模組
群的構成例的方塊圖。

第 3 圖是表示實際的時間經過與 90kHz 的時鐘的計時
之關係。

第 4 圖是表示實際的時間經過與按照視頻解碼器的視
頻資料的輸出來更新時刻的時鐘的計時之關係。

第 5 圖是表示緩衝控制模組 215 的構成例的方塊圖。

第 6 圖是表示碟片 101 的目錄(directory)構成例。

第 7 圖是表示 "PLAYLIST.DAT" 檔案的語法(syntax)。

第 8 圖是表示 PlayItem()的語法。

第 9 圖是表示 PlayListMark()的語法。

第 10 圖是表示 `mark_type` 的值與 `Mark()` 的類型之關係。

第 11 圖是表示 `PlayList()`，`PlayItem()`，剪輯 (`clip`)，及儲存於剪輯串流 (`stream`) 檔案的程式串流之關係。

第 12 圖是表示剪輯資訊檔案 `Clip()` 的語法。

第 13 圖是識別基本 (`elementary`) 串流的 `stream_id` 及 `private_stream_id` 與基本串流之關係。

第 14 圖是表示 `StaticInfo()` 的語法。

第 15 圖是表示 `DynamicInfo()` 的語法。

第 16 圖是表示 `EP_map()` 的語法。

第 17 圖是表示第 16 圖的 `index_N_minus1` 的值與 `1stRef_Picture` ~ `4thRef_Picture` 之關係。

第 18 圖 A 及第 18 圖 B 是表示 MPEG-2 System 的程式串流，程式串流包 (`pack`)，及程式串流包標頭的語法。

第 19 圖 A 及第 19 圖 B 是表示 MPEG-2 System 的 PES 封包的語法。

第 20 圖 A，第 20 圖 B 及第 20 圖 C 是表示 MPEG-2 System 的 PES 封包的語法。

第 21 圖 A 及第 21 圖 B 是表示 MPEG-2 System 的 PES 封包的語法。

第 22 圖 A 及第 22 圖 B 是表示在 MPEG-2 System 的 `PES_packet()` 的 `stream_id` 中所記述的值與基本串流的屬性之關係。

第 23 圖是表示碟片裝置所採用的 `stream_id`。

第 24 圖 是 表 示 `private_stream1_PES_payload()` 的 語 法

。

第 25 圖 是 表 示 `private_stream_id` 的 值 與
`private_payload()` 中 所 儲 存 的 基 本 串 流 的 屬 性 之 關 係 。

第 26 圖 是 表 示 `private_stream2_PES_payload()` 的 語 法

。

第 27 圖 是 表 示 `au_information()` 的 語 法 。

第 28 圖 是 用 以 說 明 `pic_struct` 。

第 29 圖 是 表 示 "PLAYLIST.DAT" 檔 案 的 具 體 例 。

第 30 圖 A 及 第 30 圖 B 是 表 示 剪 輯 資 訊 檔 案
"00001.CLP"， "00002.CLP"， "00003.CLP" 的 具 體 例 。

第 31 圖 是 表 示 剪 輯 資 訊 檔 案 "00001.CLP" 之 中 的
`EP_map()` 的 具 體 例 。

第 32 圖 是 表 示 `PlayList #0` 及 `PlayList #1` 之 中 的
`PlayListMark()` 的 具 體 例 。

第 33 圖 是 用 以 說 明 再 生 前 處 理 的 流 程 圖 。

第 34 圖 是 用 以 說 明 再 生 處 理 的 流 程 圖 。

第 35 圖 是 用 以 說 明 解 碼 順 序 與 輸 出 順 序 之 關 係 的 流 程
圖 。

第 36 圖 是 用 以 說 明 視 頻 解 碼 器 的 構 成 。

第 37 圖 是 用 以 說 明 第 36 圖 的 `DPB` 的 構 成 。

第 38 圖 是 用 以 說 明 時 刻 更 新 處 理 的 流 程 圖 。

第 39 圖 是 用 以 說 明 對 應 於 `pic_struct` 的 值 之 時 刻 更 新
處 理 。

第 40 圖 是用以說明 `PlayItem` 掉換處理的流程圖。

第 41 圖 是用以說明時間代碼 (`time code`)顯示處理的流程圖

第 42 圖 是用以說明串流切換處理的流程圖

第 43 圖 是用以說明緩衝控制模組 215 的處理流程圖。

第 44 圖 是用以說明緩衝控制模組 215 的處理流程圖。

第 45 圖 是用以說明視頻串流的讀出處理的流程圖。

第 46 圖 是用以說明音頻串流的讀出處理的流程圖。

第 47 圖 是用以說明字幕串流的讀出處理的流程圖。

第 48 圖 是用以說明再同步處理的流程圖。

第 49 圖 是用以說明標示處理的流程圖。

第 50 圖 是用以說明標示處理的一致判定。

第 51 圖 是用以說明 `playListEnd` 時序。

第 52 圖 是用以說明 `playListEnd` 時序。

第 53 圖 是用以說明事件的間隔。

第 54 圖 是用以說明輸出屬性的控制處理的流程圖。

第 55 圖 是表示剪輯資訊檔案 "00003.CLP" 中所記述的 `pts_change_point` 與 `DynamicInfo()` 的組合之具體例。

第 56 圖 是用以說明字幕顯示控制處理的流程圖。

第 57 圖 是用以說明擷取 (`capture`) 控制處理與背景 (`background`)/螢幕保護 (`screen saver`) 處理的流程圖。

第 58 圖 是表示 `private_stream2_PES_payload()` 的其他語法。

第 59 圖 是表示 `au_information()` 的其他語法。

- 117：音頻解碼器
- 118：視頻輸出介面
- 119：音頻輸出介面
- 120：視頻輸出端子
- 121：音頻輸出端子
- 201：操作系統
- 210：視頻內容再生程式
- 211：腳本控制模組
- 212：撥放器控制模組
- 213：內容資料供給模組
- 214：解碼控制模組
- 214A：計時部
- 215：緩衝控制模組
- 215A：緩衝器
- 216：視頻解碼器控制模組
- 217：音頻解碼器控制模組
- 218：字幕解碼器控制模組
- 219：繪圖處理模組
- 220：視頻輸出模組
- 220A：FIFO
- 221：音頻輸出模組
- 221A：FIFO
- 231：資料前頭指標記憶部
- 232：資料寫入指標記憶部

- 233：視頻讀出功能部
- 234：音頻讀出功能部
- 235：字幕讀出功能部
- 241：視頻讀出指標記憶部
- 242：stream_id 暫存器
- 243：au_information()暫存器
- 251：音頻讀出指標記憶部
- 252：stream_id 暫存器
- 253：private_Stream_id 暫存器
- 261：字幕讀出功能旗標記憶部
- 262：字幕讀出指標記憶部
- 263：stream_id 暫存器
- 264：private_Stream_id 暫存器
- 301：視頻緩衝器
- 302：附加資訊緩衝器
- 400A：視頻輸入端子
- S101：碟片檢查
- S102：錯誤處理
- S103：SCRIPT.DAT 及 PLAYLIST.DAT 的讀入
- S104：檔案供給
- S105：PLAYLIST.DAT 的解析
- S106：剪輯資訊檔案的讀出
- S107：檔案檢查
- S108：SCRIPT.DAT 的解釋、執行

- S121 : IN_time 認定
- S122 : 再生開始位置調查
- S123 : 時間代碼顯示
- S124 : PlaylistMark()的解析
- S125 : 再生之串流的決定
- S126 : 輸出屬性控制
- S127 : 再生開始的準備
- S128 : 資料讀入開始
- S129 : 解碼器控制開始
- S130 : 解碼開始
- S131 : 繪圖處理開始
- S132 : 輸出處理開始
- S141 : 是否持有顯示時間標籤?
- S142 : 代入顯示時間標籤
- S143 : 保存此次的 pic_struct
- S144 : 加算在前次的 pic_struct 所決定的間隔
- S151 : 解碼中斷控制
- S152 : 播放項目再生開始
- S171 : 訊息輸出
- S172 : 時間代碼換算
- S173 : 時間代碼輸出
- S191 : 串流數檢查
- S192 : 再生之串流的特定
- S193 : 其次再生之串流的認定

S194：其次再生之串流的讀出指示

S211：private_Stream_2 packet 的探索

S212：是否為所指定的 stream_id?

S213：將 au_Information()讀入內部暫存器 --

S214：在視頻讀出指標中插入之後的視頻封包的前頭

S215：是否有要求資料?

S216：一方面進行程式串流的構文解析，一方面輸出

Au_length 份的視頻資料

S217：是否僅處理 number-of-access-unit?

S230：private_Stream_1?

S231：同步碼探索

S232：MPEG 音頻封包的探索

S233：音頻讀出指標更新

S234：private_Stream_1 packet 的探索

S235：是否為所被指定的 private_Stream_id?

S236：音頻讀出指標←Au_locator 之後的位置+Au_locator

S237：是否有要求資料?

S238：一方面進行程式串流的構文解析，一方面輸出

固定長的音頻存取單元

S251：字幕讀出功能旗標?

S252：private_Stream_1 packet 的探索

S253：是否為所被指定的 private_Stream_id?

S254：字幕讀出指標←Au_locator 之後的位置+Au_locator

S255：是否有要求資料?

S256：一邊進行程式串流的構文解析，一邊輸出記述於字幕存取單元的前頭之長度份的字幕存取單元

S271：顯示時間標籤的偏差是否大？

S272：視頻是否較慢？

S273：指示視頻存取單元的跳過

S274：檢查存取單元的 `au_ref_flag`

S275：存取單元是否為非參照畫像？

S276：處理視頻存取單元

S277：跳過視頻存取單元的處理

S278：指示視頻的重複輸出

S279：視頻的重複輸出

S301：現在時刻是否一致於 `mark_time_Stamp`？

S302：認定作為處理對象的 `mark`

S303：在處理對象 `mark` 中是否記述有

`entry_ES_Stream_id/entry_ES_Private_Stream_id`？

S304：在再生中的串流中是否含有

`entry_ES_Stream_id/entry_ES_Private_Stream_id` 的串流？

S305：判定處理對象 `mark` 的 `mark_type`

S306：章節/索引的更新處理

S307：將事件訊息及 `mark_data` 通知給腳本控制模組

S308：對應於 `mark_data` 的處理

S320：交付 `pts_Change_Point`

S321：現在時刻是否一致於 `Pts_Change_Point`？

S322：確定一致於現在時刻的 `pts_Change_Point` 的

DynamicInfo()

- S323：供給記述於 **DynamicInfo()** 的輸出屬性
- S324：按照輸出屬性，開始視頻/音頻的輸出的控制
- S341：指示字幕的處理的初期化
- S342：有關字幕的顯示，是否有新的顯示方式指示？
- S343：字幕串流是否再生中？
- S345：是否為內定顯示的指示？
- S346：取得再生中的字幕串流的 **StaticInfo()**
- S347：判定 **staticInfo()** 的 **Configurable_flag**
- S348：錯誤顯示
- S349：供給顯示方式指示
- S350：按照顯示方式指示，開始字幕的處理
- S351：**PlayItem** 是否掉換？
- S371：視頻是否再生中？
- S372：由再生中的視頻的 **PlayList()** 及 **Clip()** 來取得

capture_enable_flag

- S373：根據 **capture_enable_flag**，判定擷取的可否？
- S374：錯誤顯示
- S375：供給擷取指示
- S376：擷取、記憶
- S380：圖像的顯示
- S381：進行編碼
- S382：進行多重化，選擇幀內畫面
- S383：檢測出幀內畫面及接續的圖像的位置

S384：重寫 RAPI 資訊

S385：根據 RAPI 資訊來將接近30區段的圖像位置拷貝於 N-th_Ref_picture_copy

S386：根據 N-th_Ref_picture_copy 來決定 index_minus1，予以記錄

S391：選擇再生可能位置

S392：根據 RPN_EP_start 及 N-th_Ref_Picture_copy 來決定讀出資料

S393：讀出儲存有再生對象的基本串流所被多重化的程式串流之剪輯串流

S394：多重化分離視頻串流

S395：進入點選定處理

S396：解碼顯示參照畫像

S397：是否指示快進終了？

S401：是否高速？

S402：記入項號碼(N)+=2

S403：讀出第 N，N+1，N-1個的 index_N_minus1

S404：index_N_minus(N)==0or1?

S405：使用 N

S406：index_N_minus(N+1)==0or1?

S407：使用 N+1

S408：index_N_minus(N-1)==0or1?

S409：使用 N-1

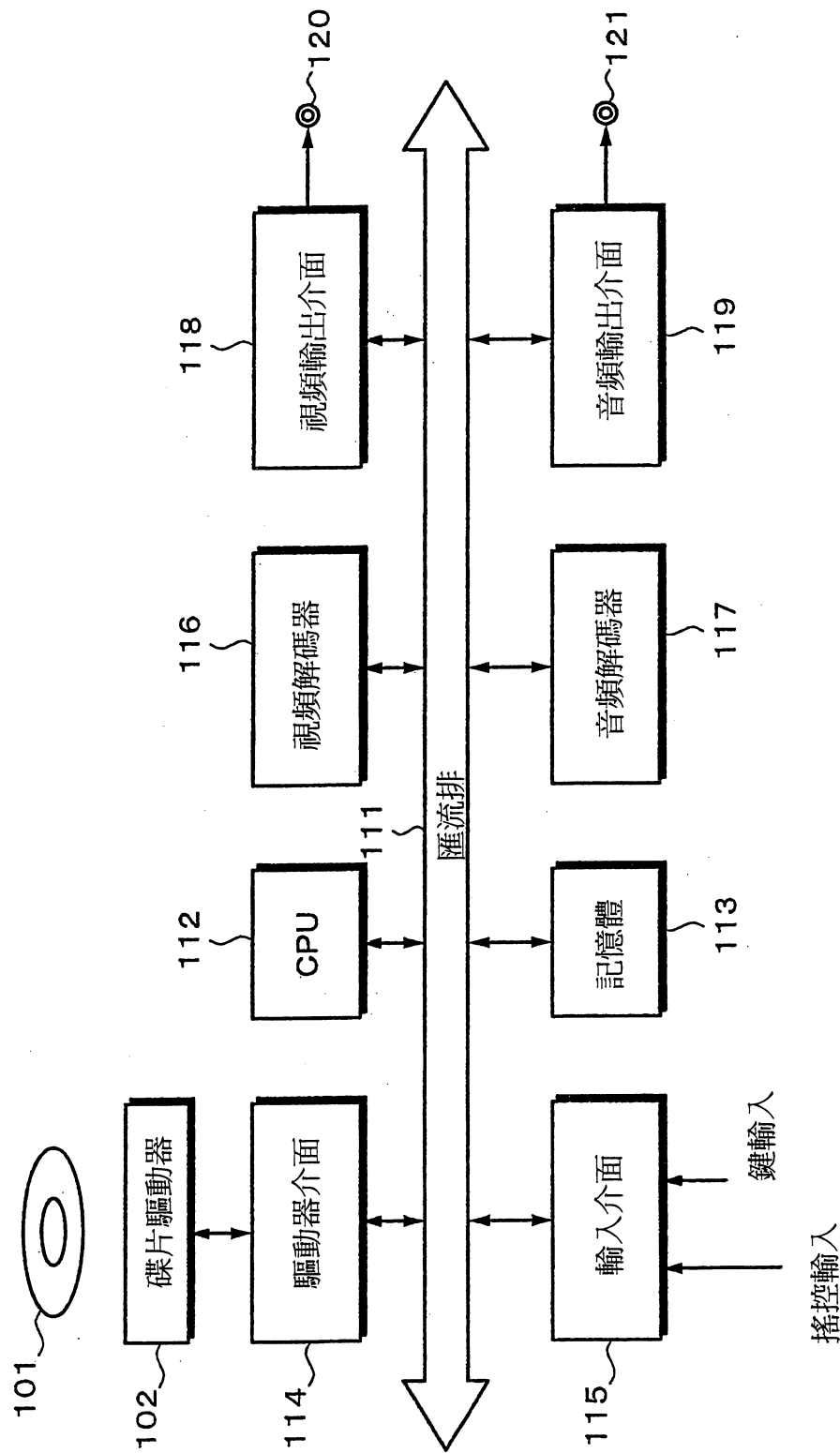
S410：使用 N

- S411 : 記入項號碼 (N)+=5
- S412 : 讀出第 N , N+1 , N-1個的 index_N_minus1
- S413 : index_N_minus(N)==3 or 2?
- S414 : 使用 N
- S415 : index_N_minus(N+1)==3 or 2?
- S416 : 使用 N+1
- S417 : index_N_minus(N-1)==3 or 2?
- S418 : 使用 N-1
- S419 : 使用 N
- S441 : 是否相當於視頻的章節前頭?
- S442 : priority_flag=3
- S443 : 是否為視頻的重要場景變化?
- S444 : priority_flag=2
- S445 : 是否為視頻的通常場景變化?
- S446 : priority_flag=1
- S447 : priority_flag=0
- S448 : 是否全部終了?
- S449 : 是否相當於字幕的章節前頭?
- S450 : priority_flag=3
- S451 : 是否為字幕的重要場景變化?
- S452 : priority_flag=2
- S453 : 是否為字幕的通常場景變化?
- S454 : priority_flag=1
- S455 : priority_flag=0

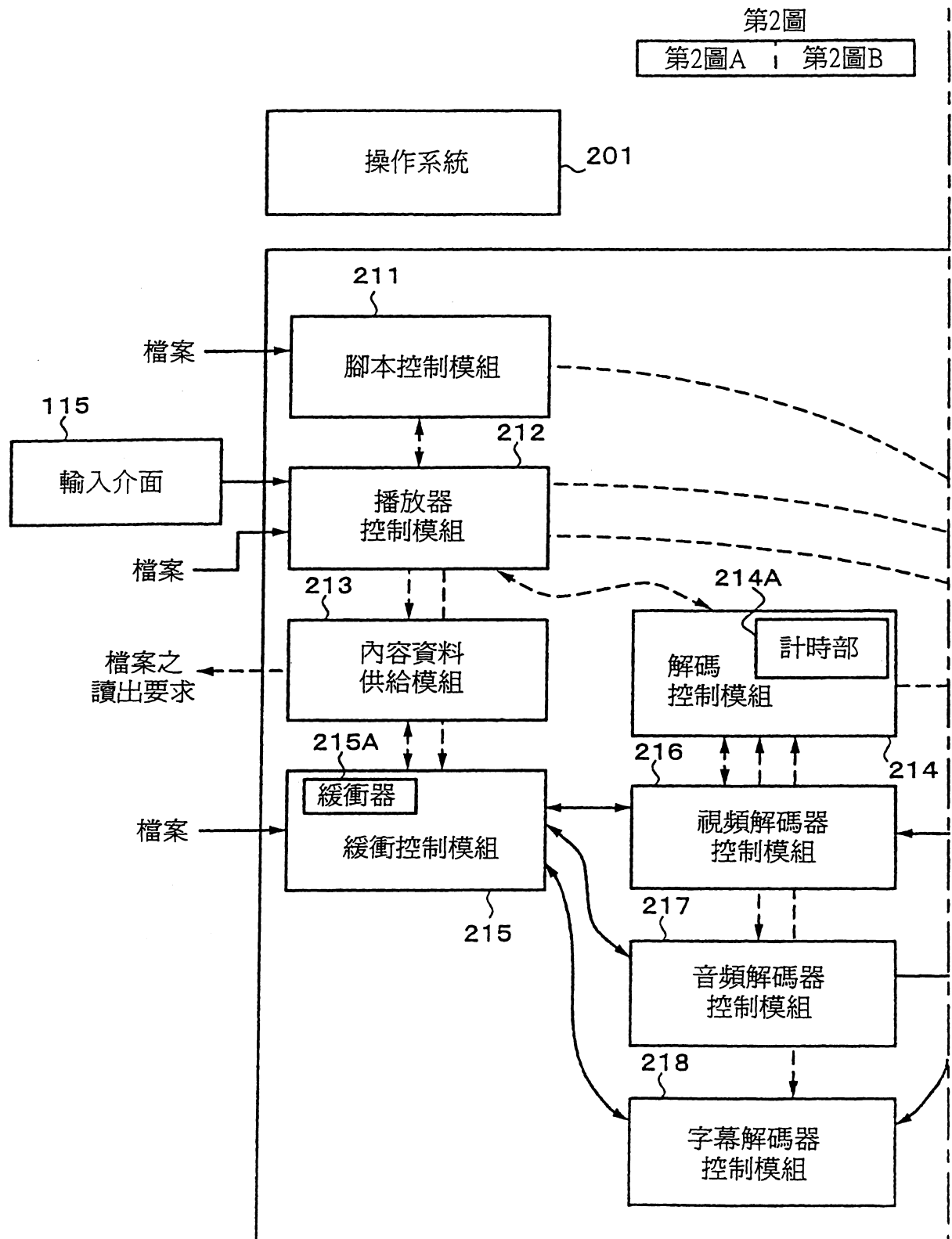
I299958 (226)

S456：是否全部終了？

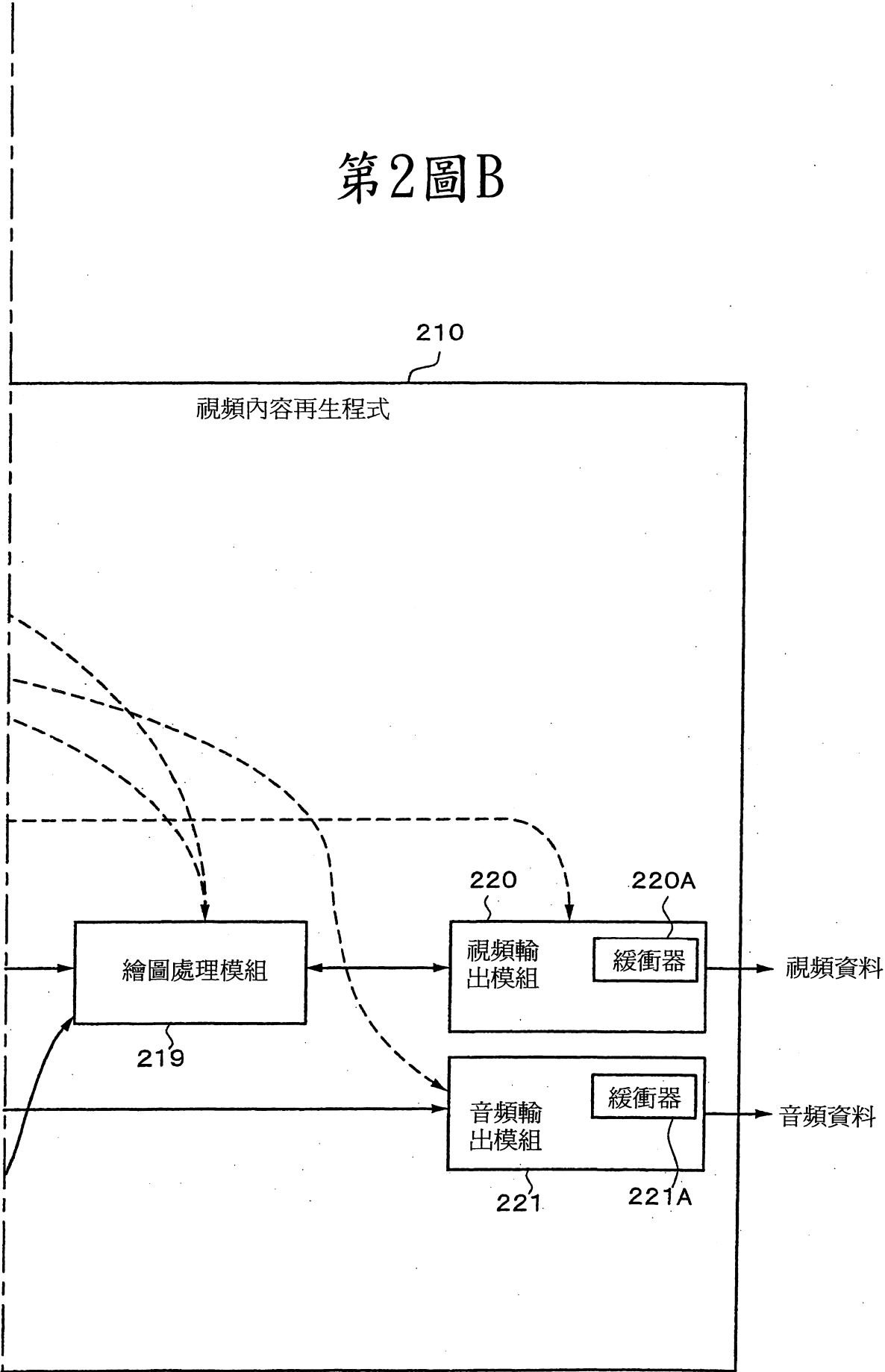
第1圖



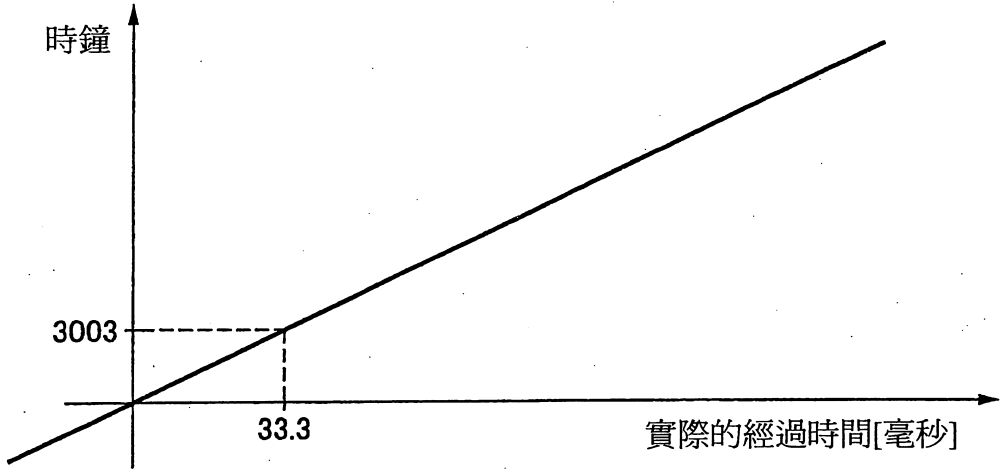
第2圖A



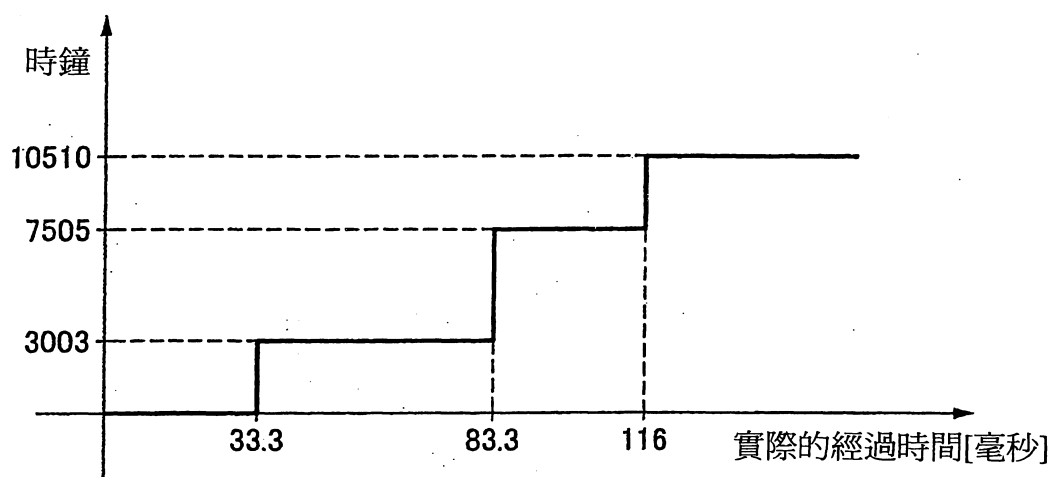
第2圖B



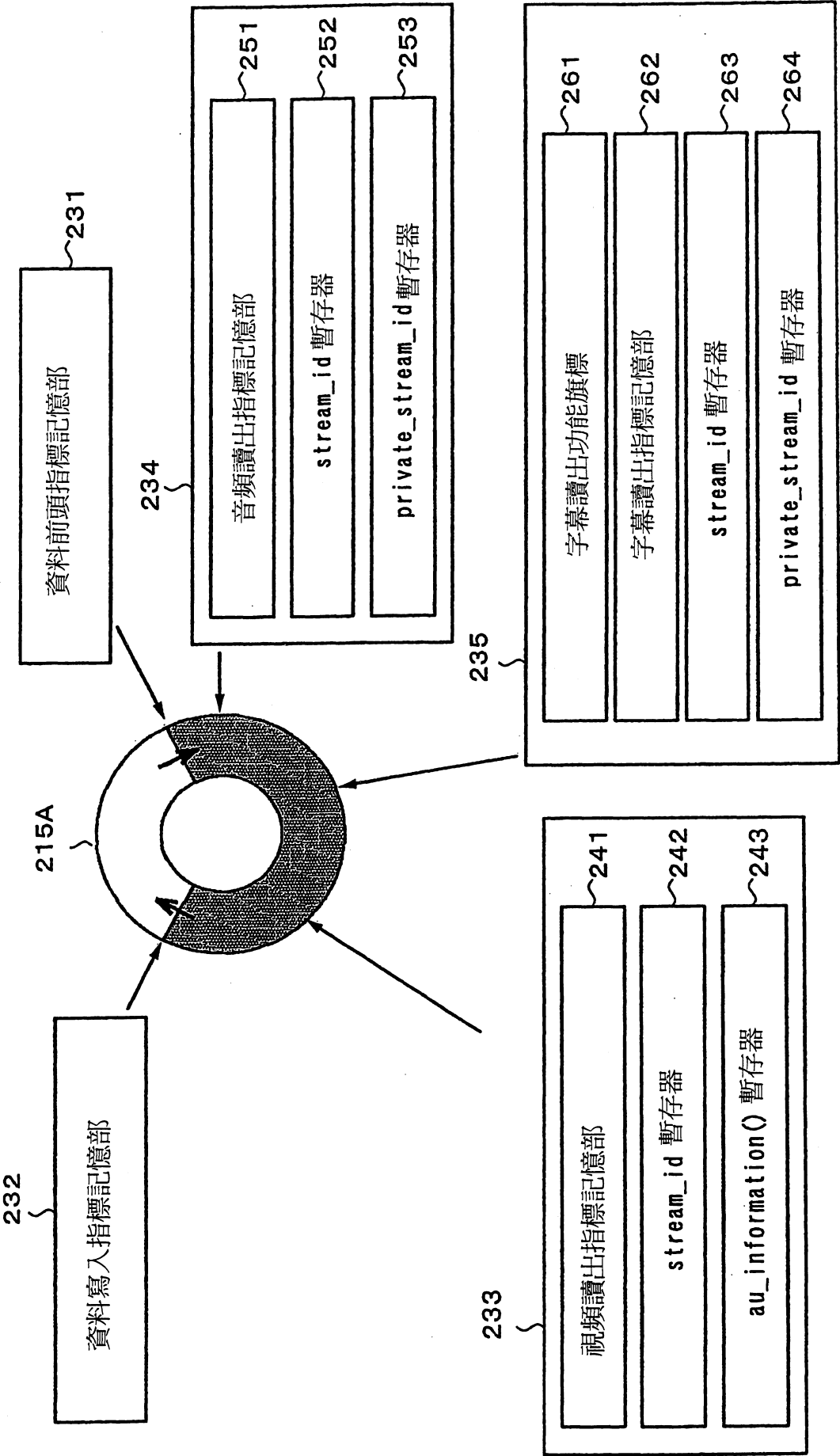
第3圖



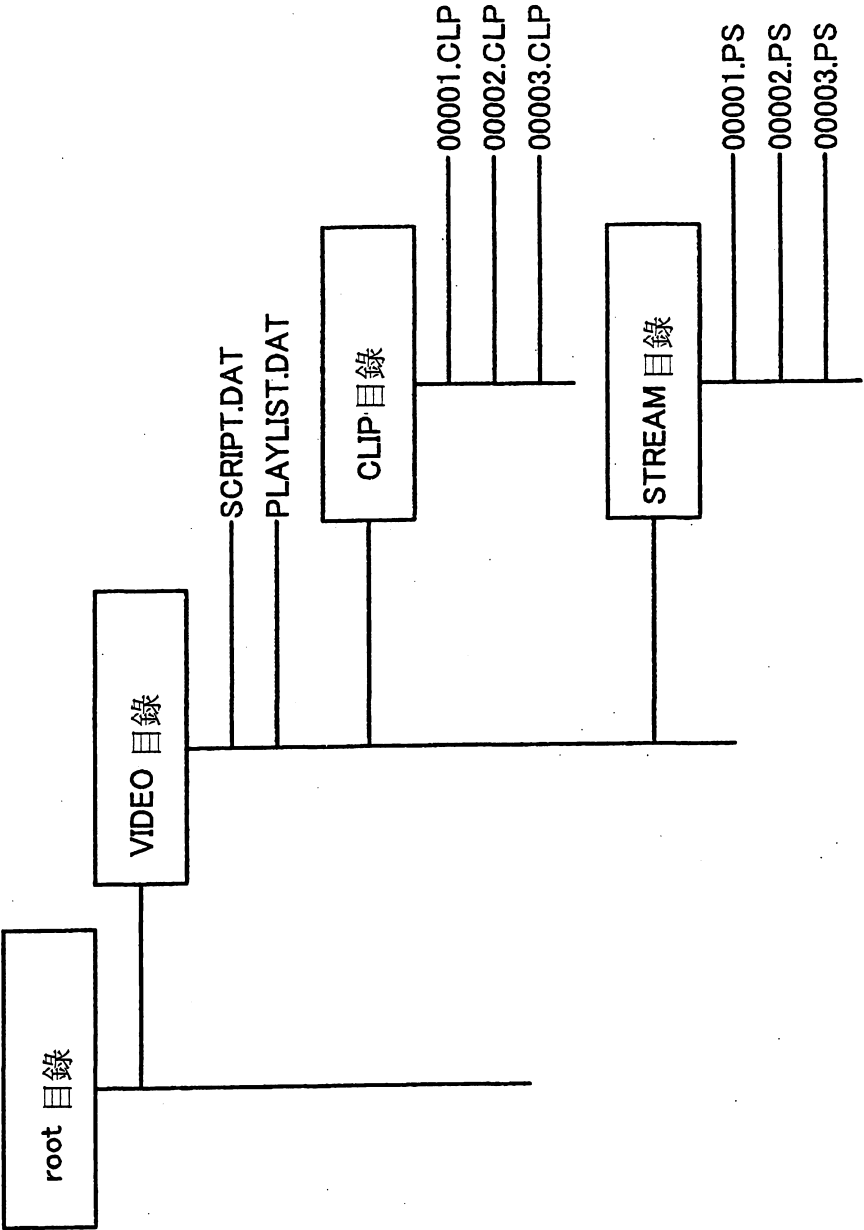
第4圖



第5圖



第6圖



第7圖

PlayList.PAT 檔案			
語法	位元數	助憶碼	
"PLAYLIST.DAT"{			
name_length	8	uimbsbf	
name_string	8x255	bslbf	
number_of_PlayLists	16	uimbsbf	
for (i=0; i<number_of_PlayLists; i++) {			
PlayList() { // A PlayList()			
PlayList_data_length	32	uimbsbf	
// 屬性資訊			
reserved_for_word_alignment	15	bslbf	
capture_enable_flag_PlayList	1	bslbf	
PlayList_name_length	8	uimbsbf	
PlayList_name_string	8*255	bslbf	
//			
number_of_PlayItems	16	uimbsbf	
for (i=0; i<number_of_PlayItems; i++) {			
PlayItem()			
}			
PlayListMark()			
}			
}			

第8圖

PlayItem()		
語法	位元數	助憶碼
PlayItem() {		
length	16	uimsbf
Clip_Information_file_name_length	16	uimsbf
Clip_Information_file_name	8*Clip_Information_file_name_length	bslbf uimsbf uimsbf
IN_time	32	
OUT_time	32	
}		

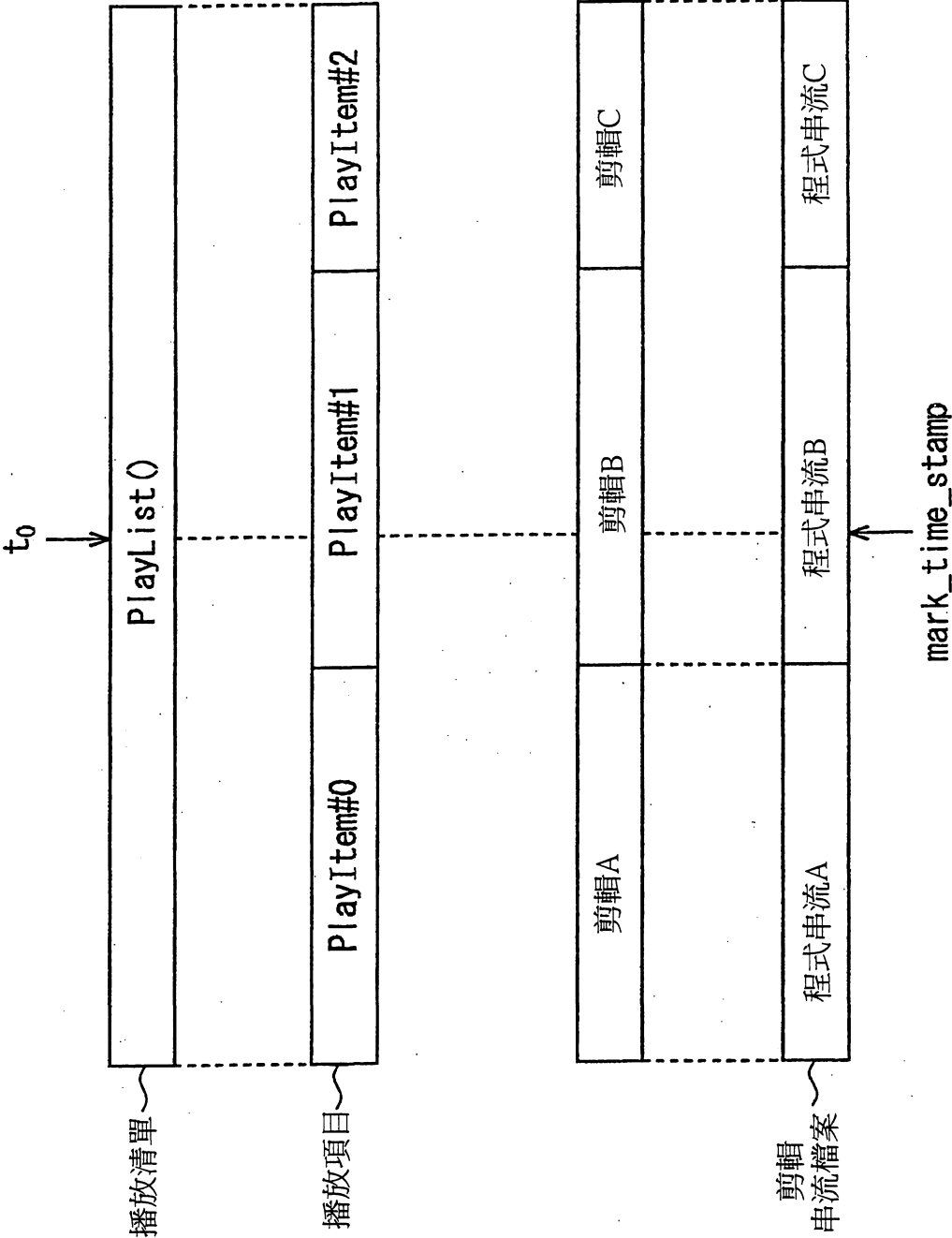
第9圖

PlayListMark()			
語法	位元數	助憶碼	
PlayListMark() {			
length	32	uimsbf	
number_of_PlayList_marks	16	uimsbf	
for (i=0; i < number_of_PlayList_marks; i++) {			
Mark() {			
mark_type	8	uimsbf	
mark_name_length	8	uimsbf	
ref_to_PlayItem_id	16	uimsbf	
mark_time_stamp	32	uimsbf	
entry_ES_stream_id	8	uimsbf	
entry_ES_private_stream_id	8	uimsbf	
mark_data	32	bslbf	
mark_name_string	8*24	bslbf	
}			
}			
}			

第10圖

mark_type	串流編碼
0	預約
1	"Chapter" 標示
2	"Index" 標示
3	"Event" 標示
4~255	預約

第11圖



第12圖

clip()「剪輯資訊檔案」(.CLP)			
語法	位元數	助憶碼	
XXXXX.CLIP {			
presentation_start_time	32	uimsbf	
presentation_end_time	32	uimsbf	
reserved_for_word_alignment	7	bslbf	
capture_enable_flag_Clip	1	bslbf	
number_of_streams	8	uimsbf	
for (i=0; i < number_of_streams; i++) {			
StreamInfo() {			
length	16	uimsbf	
stream_id	8	uimsbf	
private_stream_id	8	uimsbf	
StaticInfo() {			
reserved_for_word_alignment	8	bslbf	
number_of_DynamicInfo	8	uimsbf	
for (j=0; j < number_of_DynamicInfo; j++) {			
pts_change_point	32	uimsbf	
DynamicInfo() {			
}			
}			
}			
EP_map() {			
}			

第13圖

基本串流 種類	stream_id	private_stream_id
視頻 ATRAC 音頻 LPCM 音頻 字幕	0xE0~0xEF 0xBD 0xBD 0xBD	(無) 0x00~0x0F 0x10~0x1F 0x80~0x9F

第14圖

StaticInfo()			
語法	位元數	助憶碼	
StaticInfo() {			
if(stream == VIDEO) {			
reserved_for_word_alignment	16	bslbf	
picture_size	4	uimsbf	
frame_rate	4	uimsbf	
reserved_for_word_alignment	7	bslbf	
cc_flag	1	bslbf	
}else if (stream == AUDIO) {			
audio_language_code	16	bslbf	
channel_configuration	8	uimsbf	
reserved_for_word_alignment	3	bslbf	
lfe_existence	1	bslbf	
sampling_frequency	4	uimsbf	
}else if (stream == SUBTITLE) {			
subtitle_language_code	16	bslbf	
reserved_for_word_alignment	15	bslbf	
configurable_flag	1	uimsbf	
}			
}			

第15圖

DynamicInfo()		
語法	位元數	助憶碼
DynamicInfo(i, j) {		
reserved_for_word_alignment	8	bslbf
if(stream == VIDEO) {		
reserved_for_word_alignment	4	bslbf
display_aspect_ratio	4	uimsbf
} else if(stream == AUDIO) {		
reserved_for_word_alignment	4	bslbf
channel_assignment	4	uimsbf
} else if(stream == SUBTITLE) {		
reserved_for_word_alignment	8	bslbf
}		
}		

第16圖

EP_map()			
語法	位元數	助憶碼	
EP_map() {			
reserved_for_word_alignment	8	bslbf	
number_of_stream_id_entries	8	uimsbf	
for (k=0; k<number_of_stream_id_entries;k++) {			
stream_id	8	bslbf	
private_stream_id	8	bslbf	
number_of_EP_entries	32	uimsbf	
for (i=0; i<number_of_EP_entries;i++) {			
if (isVideoStream) {			
index_N_minus1	2	uimsbf	
N-th_Ref_picture_copy	14	uimsbf	
} else			
reserved_for_future_use	16	bslbf	
}			
PTS_EP_start	32	uimsbf	
RPN_EP_start	32	uimsbf	
}			
}			

第17圖

	index_N_minus1
1stRef_Picture	0
2ndRef_Picture	1
3rdRef_Picture	2
4thRef_Picture	3

第18圖A

表2-31,程式串流		
語法	位元數	助憶碼
<pre>MPEG2_program_stream() { do { pack() } while(nextbits() == pack_start_code) MPEG_program_end_code }</pre>	32	bslbf

表2-32,程式串流包		
語法	位元數	助憶碼
<pre>pack() { pack_header() while(nextbits() == packet_start_code_prefix) { PES_packet() } }</pre>		

第18圖B

表2-33,程式串流包標頭		
語法	位元數	助憶碼
<pre>pack_header() { pack_start_code '01' system_clock_reference_base[32..30] marker_bit system_clock_reference_base[29..15] marker_bit system_clock_reference_base[14..0] marker_bit system_clock_reference_extension marker_bit program_mux_rate marker_bit marker_bit reserved pack_stuffing_length for (i=0; i<pack_stuffing_length; i++) { stuffing_byte } if (nextbits() == system_header_start_code) { system_header() } }</pre>	32 2 3 1 15 1 15 1 9 1 22 1 1 5 3 8	bslbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf uimbsbf bslbf uimbsbf bslbf bslbf bslbf uimbsbf bslbf

第19圖A

第19圖

第19圖A
第19圖B

表2-17,PES封包		
語法	位元數	助憶碼
PES_packet() { packet_start_code_prefix stream_id PES_packet_length if (stream_id !=program_stream_map && stream_id !=padding_stream && stream_id !=private_stream_2 && stream_id !=ECM && stream_id !=ENMM && stream_id !=program_stream_directory && stream_id !=DSMCC_stream	24 8 16	bslbf uimbsf uimbsf

第19圖B

<pre>&& stream_id !=ITU-T Rec. H. 222.1 type E stream) [,10' PES_scrambling_control PES_priority data_alignment_indicator copyright original_or_copy PTS_DTS_flags ESCR_flag ES_rate_flag DSM_trick_mode_flag additional_copy_info_flag PES_CRC_flag PES_extension_flag PES_header_data_length if (PTS_DTS_flags == '10') { ,0010' 4 bsbf PTS [32..30] marker_bit PTS [29..15] marker_bit PTS [14..0] marker_bit }]</pre>			2 2 1 1 1 1 2 1 1 1 1 1 1 1 8	bsbf bsbf bsbf bsbf bsbf bsbf bsbf bsbf bsbf bsbf bsbf bsbf bsbf uimsb
			4 3 1 15 1 15 1	bsbf bsbf bsbf bsbf bsbf bsbf bsbf

第20圖C

<pre>if(DSM_trick_mode_flag=='1'){ trick_mode_control if(trick_mode_control==fast_forward){ field_id intra_slice_refresh frequency_truncation } else if(trick_mode_control==slow_motion){ rep_cntrl } else if(trick_mode_control==freeze_frame){ field_id reserved } else if(trick_mode_control==fast_reverse){ field_id intra_slice_refresh frequency_truncation } else if(trick_mode_control==slow_reverse){ rep_cntrl } else reserved } if(additional_copy_info_flag=='1'){ marker_bit additional_copy_info } if(PES_CRC_flag=='1'){ previous_PES_packet_CRC }</pre>	3 2 1 2 5 2 3 2 1 2 5 5 1 7 16	uimbsf bslbf bslbf bslbf uimbsf uimbsf bslbf bslbf bslbf bslbf uimbsf bslbf bslbf bslbf bslbf
---	--	---

第21圖A

第21圖

第21圖A
第21圖B

表2-17.PES封包(終了)		
語法	位元數	助憶碼
if (PES_extension_flag=='1') { PES_private_data_flag pack_header_field_flag program_packet_sequence_counter_flag P-STD_buffer_flag reserved PES_extension_flag_2 if (PES_private_data_flag=='1') { PES_private_data } if (pack_header_field_flag == '1') { pack_field_length pack_header() } if (program_packet_sequence_counter_flag == '1') { marker_bit program_packet_sequence_counter marker_bit MPEG1_MPEG2_identifier original_stuff_length } if (P-STD_buffer_flag == '1') { '01' P-STD_buffer_scale P-STD_buffer_size } }	1	bslbf
	1	bslbf
	1	bslbf
	1	bslbf
	3	bslbf
	1	bslbf
	128	bslbf
	8	uimbsf
	1	bslbf
	7	uimbsf
	1	bslbf
	1	bslbf
	6	uimbsf
	2	bslbf
	1	bslbf
	13	uimbsf

第21圖B

<pre>if(PES_extension_flag_2 == '1') { marker_bit PES_extension_field_length for (i=0; i<PES_extension_field_length; i++) { reserved } for (i=0; i<N1; i++) { stuffing_byte } for (i=0; i<N2; i++) { PES_packet_data_byte } } else if (stream_id == program_stream_map stream_id == private_stream_2 stream_id == ECM stream_id == EMM stream_id == program_stream_directory stream_id == DSMCC_stream stream_id == ITU-T Rec. H. 222.1 type E stream) { for (i=0; i<PES_packet_length; i++) { PES_packet_data_byte } } else if (stream_id == padding_stream) { for (i=0; i<PES_packet_length; i++) { padding_byte } }</pre>		1 7 8 8 8 8 8	bslbf uimsbf bslbf bslbf bslbf bslbf bslbf
--	--	---	--

第22圖A

第22圖

第22圖A

第22圖B

表 2-18, stream_id 分配

stream_id	注釋	串流編碼
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3
		audio stream number x xxxx
1110 xxxx		ITU-T Rec. H. 262 ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2
		video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H. 222.0 ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6 DSMCC_stream
1111 0011	2	ISO/IEC 13522_stream
1111 0100	6	ITU-T Rec. H. 222.1 type A
1111 0101	6	ITU-T Rec. H. 222.1 type B
1111 0110	6	ITU-T Rec. H. 222.1 type C
1111 0111	6	ITU-T Rec. H. 222.1 type D
1111 1000	6	ITU-T Rec. H. 222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100 ... 1111 1110		reserved data stream
1111 1111	4	program_stream_directory

第22圖B

記號X可為值「0」或「1」的其中一個，串流的形態相同。串流號碼是以X來賦予。

注 1: `program_stream_map` 的 PES 封包是使用以 2.5.4.1 所規定的獨白的語法。

注 2: `private_stream_1` 及 `ISO/IEC_13552_stream` 的 PES 封包是按照與以 ITU-T Rec.H.262 之 `ISO/IEC13818-2` 視頻串流及 `ISO/IEC13813-3` 音頻串流所規定者相同 PES 封包的語法。

注 3: `private_stream_2`、`ECM_STREAM`、及 `EMM_stream` 的 PES 封包是除了 `PES_packet_length` 場之後未被指定任何語法以外類似於 `private_stream_1`。

注 4: `program_stream_directory` 的 PES 封包是使用以 2.5.5 所規定的獨白的語法。

注 5: `DSM-CC_stream` 是使用以 `ISO/IEC13818-6` 所規定的獨白的語法。

注 6: 該 `stream_id` 是與表 2-29 的 `stream_type 0x09` 組合使用。

注 7: 該 `stream_id` 是僅使用於程式串流或 `ISO/IEC11172-1` 系統串流的資料用的傳輸串流的 PES 封包 (2.4.3.7 參照)。

第23圖

stream_id	串流編碼 (串流的種類)
1011 1101	private_stream_1
1011 1110	padding_stream
1011 1111	private_stream_2
110x xxxx	音頻串流號碼 x xxxx
1110 xxxx	視頻串流號碼 xxxx

第24圖

語法	位元數	助憶碼
private_stream1_PES_payload() {		
private_header() {		
private_stream_id	8	uimsbf
if(stream == ATRAC) {		
reserved_for_future_use	8	bslbf
AU_locator	16	uimsbf
if(stream == LPCM) {		
fs_flag	1	uimsbf
reserved_for_future_use	3	bslbf
ch_flag	4	uimsbf
AU_locator	16	uimsbf
}else if (stream == SUBTITLE) {		
reserved_for_future_use	8	bslbf
AU_locator	16	uimsbf
}		
}		
private_payload()		
}		

第25圖

private_stream_id	串流編碼 (串流的種類)
0000 xxxx	ATRAC 音頻串流號碼 xxxx
0001 xxxx	LPCM 串流號碼 xxxx
100x xxxx	副標題串流號碼 x xxxx

第26圖

語法	位元數	助憶碼
private_stream2_PES_payload() {		
reserved_for_future_use	8	bslbf
video_stream_id	8	uimsbf
1stRef_picture	16	uimsbf
2ndRef_picture	16	uimsbf
3rdRef_picture	16	uimsbf
4thRef_picture	16	uimsbf
au_information()		
VBI()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
}		

第27圖

語法	位元數	助憶碼
au_information() {		
length	16	uimsbf
reserved_for_word_alignment	8	bslbf
number_of_access_unit	8	uimsbf
for (i=0; i<number_of_access_unit; i++) {		
reserved	4	bslbf
pic_struct_copy	4	uimsbf
au_ref_flag	1	uimsbf
reserved	2	bslbf
AU_length	21	uimsbf
}		uimsbf
}		

第28圖

D-1, pic_struct 的說明			
值	圖像的顯示	約制	NumClockTS
0	幀	field_pic_flag = 0	1
1	頂部場	field_pic_flag = 1 bottom_field_flag = 0	1
2	底部場	field_pic_flag = 1 bottom_field_flag = 1	1
3	頂部場, 底部場的順序	field_pic_flag = 0	2
4	底部場, 頂部場的順序	field_pic_flag = 0	2
5	頂部場, 底部場, 頂部場(重複顯示)的順序	field_pic_flag = 0	3
6	底部場, 頂部場, (底部場(重複顯示)的順序	field_pic_flag = 0	3
7	幀的2次重複	field_pic_flag = 0 fixed_frame_rate_flag = 1	2
8	幀的3次重複	field_pic_flag = 0 fixed_frame_rate_flag = 1	3
9..15	預約		

第29圖

// "PLAYLIST.DAT"		
number_of_PlayLists	2	
// Playlist()	Playlist #0	Playlist #1
capture_enable_flag_PlayList	1	0
number_of_PlayItems	2	1
// PlayItem #0		
Clip_Information_file_name	"00001.CLP"	"00003.CLP"
IN_time	180,090	90,000
OUT_time	27,180,090	81,090,000
// PlayItem #1		
Clip_Information_file_name	"00002.CLP"	
IN_time	90,000	
OUT_time	27,090,000	

第30圖

第30圖A

第30圖B

第30圖A

// Clip()	"00001.CLP"	"00002.CLP"	"00003.CLP"
presentation_start_time	90,000	90,000	90,000
presentation_end_time	27,990,000	27,090,000	81,090,000
capture_enable_flag_Clip	1	0	1
number_of_streams	4	4	3
// stream #0	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xE0 (video stream)	0xE0 (video stream)	0xE0 (video stream)
private_stream_id	0x00	0x00	0x00
	picture_size='720x480'	picture_size='720x480'	picture_size='720x480'
	frame_rate='29.97Hz'	frame_rate='29.97Hz'	frame_rate='29.97Hz'
	cc_flag='Yes'	cc_flag='Yes'	cc_flag='No'
number_of_DynamicInfo	0	0	2
// stream #1	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xBD (private_stream_1)	0xBD (private_stream_1)	0xE1 (video stream)
private_stream_id	0x00 (ATRAC)	0x00 (ATRAC)	0x00
	audio_language_code='日本語'	audio_language_code='日本語'	picture_size='720x480'
	channel_configuration='STEREO'	channel_configuration='STEREO'	frame_rate='29.97Hz'
	lfe_existence='NO'	lfe_existence='NO'	cc_flag='No'
	sampling_frequency='48KHz'	sampling_frequency='48KHz'	
number_of_DynamicInfo	0	0	0

第30圖B

// stream #2	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xBD (private_stream_1)	0xBD (private_stream_1)	0xBD (private_stream_1)
private_stream_id	0x80 (Subtitle)	0x80 (Subtitle)	0x00 (ATRAC)
	subtitle_language_code='日本語'	subtitle_language_code='日本語'	audio_language_code= '日本語'
	configurable_flag=0	configurable_flag=0	channel_configuration='STEREO'
			life_existence='NO'
			sampling_frequency='48KHz'
number_of_DynamicInfo	0	0	3
// stream #3	StreamInfo()	StreamInfo()	
stream_id	0xBD (private_stream_1)	0xBD (private_stream_1)	
private_stream_id	0x81 (Subtitle)	0x81 (Subtitle)	
	subtitle_language_code='日本語'	subtitle_language_code='日本語'	
	configurable_flag=1	configurable_flag=1	
number_of_DynamicInfo	0	0	

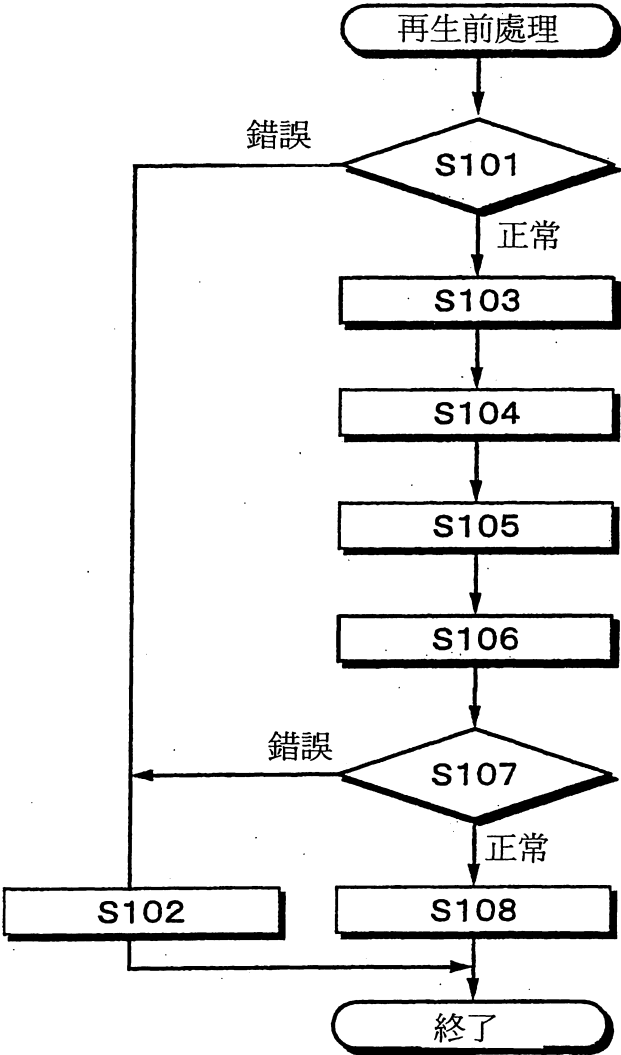
第32圖

PlayList #00のPlayListMark() number_of_PlayList_marks==7						
Mark()	mark_type	ref_to_PlayItem_id	mark_time_stamp	entry_ES_stream_id	entry_ES_private_stream_id	mark_data
#0	Chapter	0	180,090	0	0	1 前頭
#1	Index	0	5,580,090	0	0	1 1分
#2	Index	0	10,980,090	0	0	2 2分
#3	Event	0	16,380,090	0	0	0 3分
#4	Chpater	1	90,000	0	0	2 5分
#5	Index	1	5,490,000	0	0	1 1分
#6	Index	1	10,890,000	0	0	2 2分

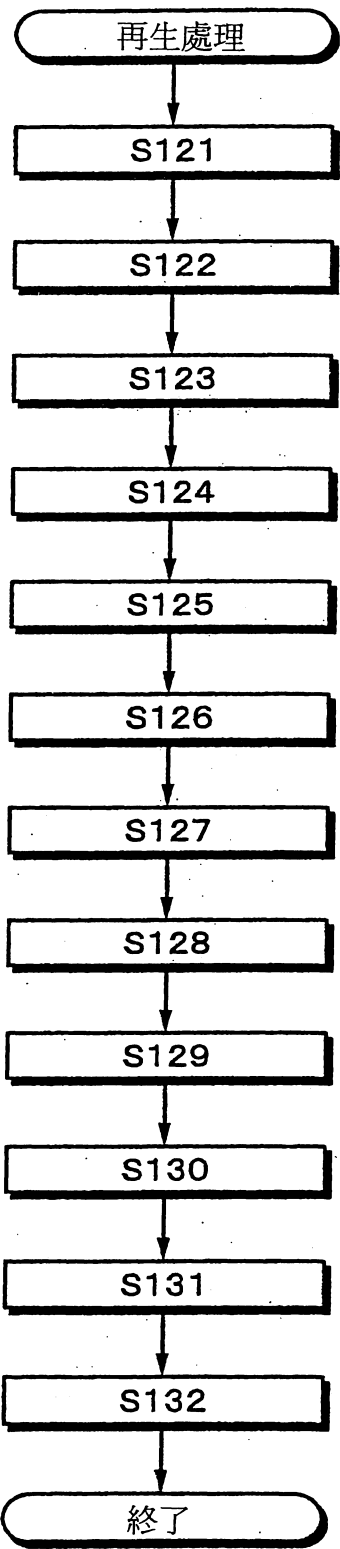
PlayList #10のPlayListMark() number_of_PlayList_marks==3					
Mark()	mark_type	ref_to_PlayItem_id	mark_time_stamp	entry_ES_stream_id	entry_ES_private_stream_id
#0	Chapter	0	90,000	0	0
#1	Event	0	27,090,000	0xE0	0
#2	Event	0	27,540,000	0xE1	0

0	前頭
1	5分
2	5分5秒(305秒)

第33圖



第34圖

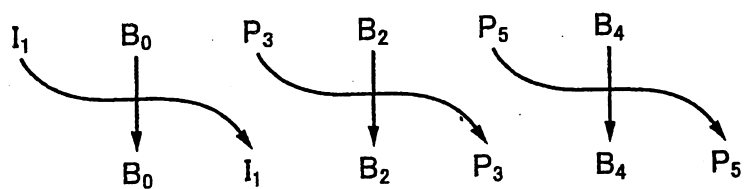


第35圖

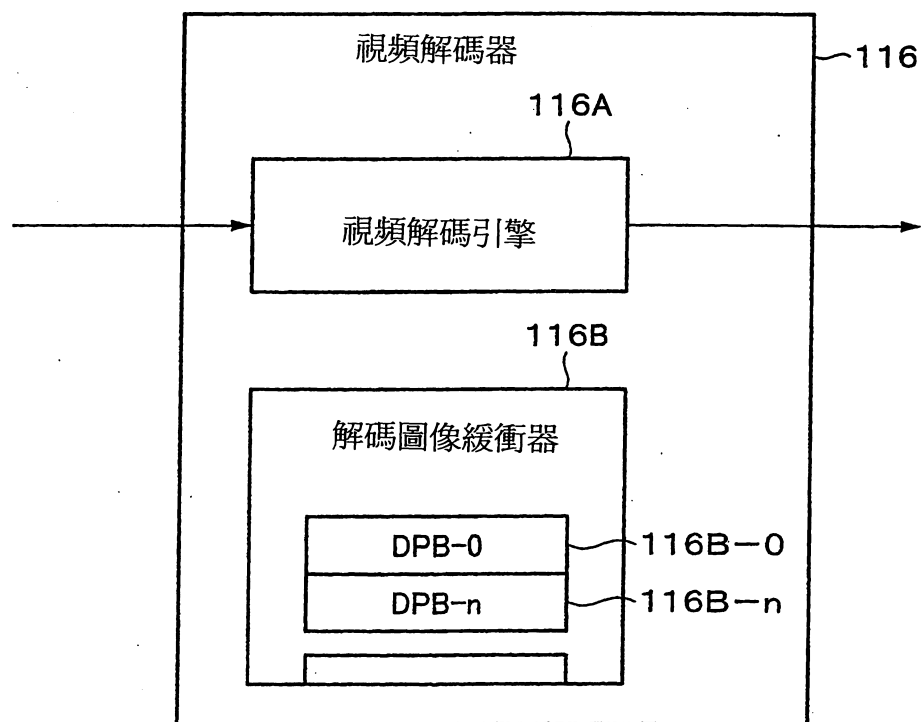
解碼順序



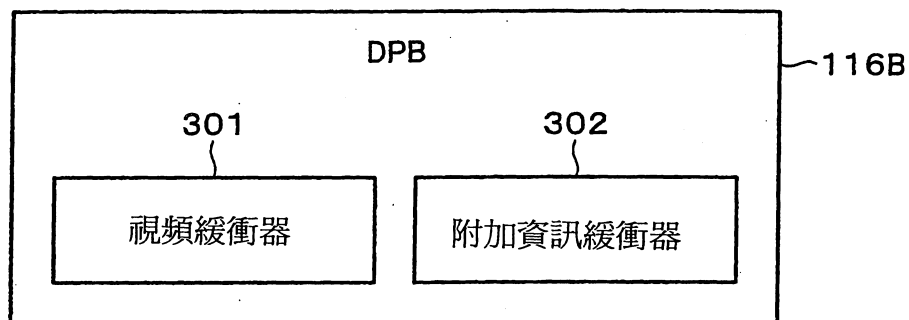
輸出順序



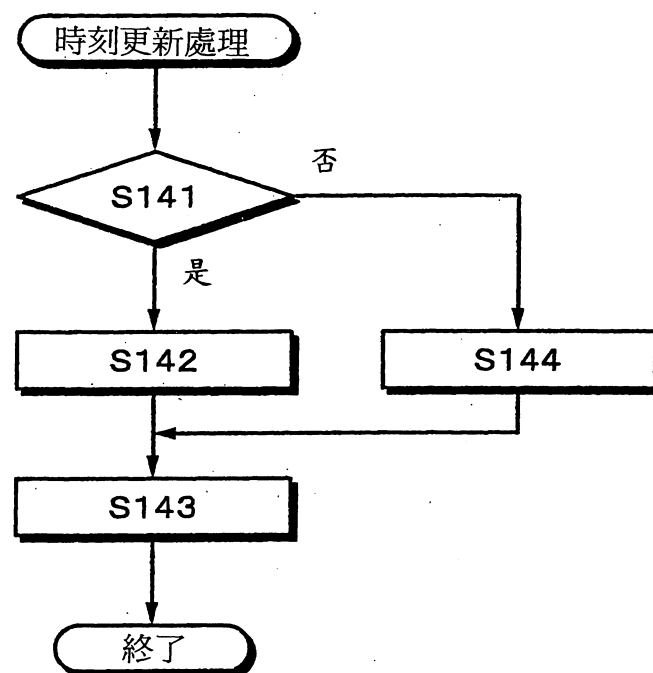
第36圖



第37圖



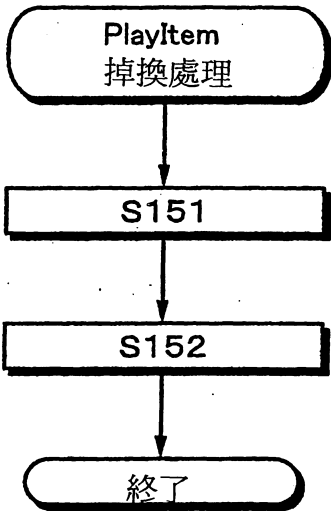
第38圖



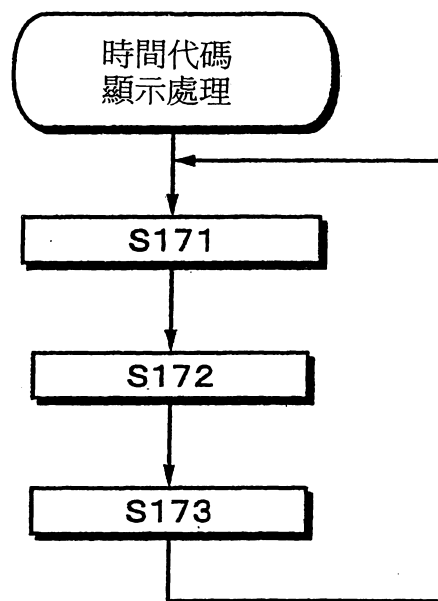
第39圖

PTS	保存有的 pic_struct 之值	動作
有	—	代入 PTS
無	0,1,2,3,4	時刻+2場 (3003)
	5,6	時刻+3場 (4505)

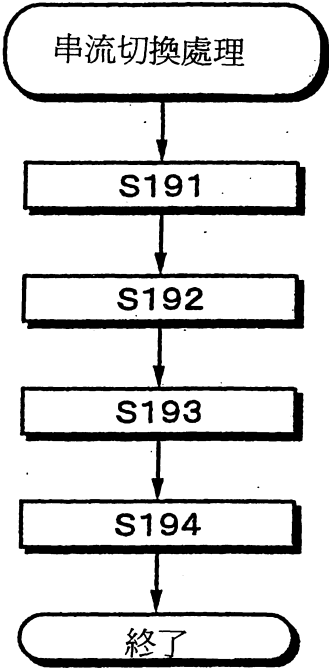
第40圖



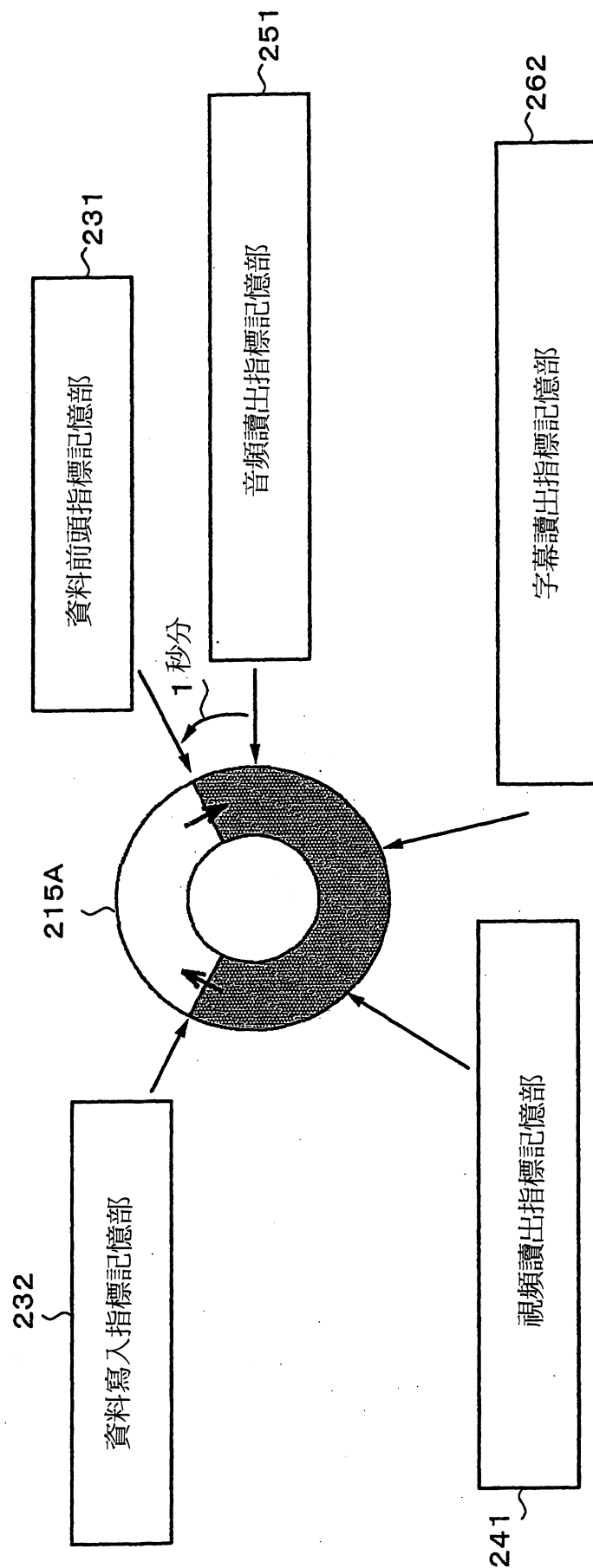
第41圖



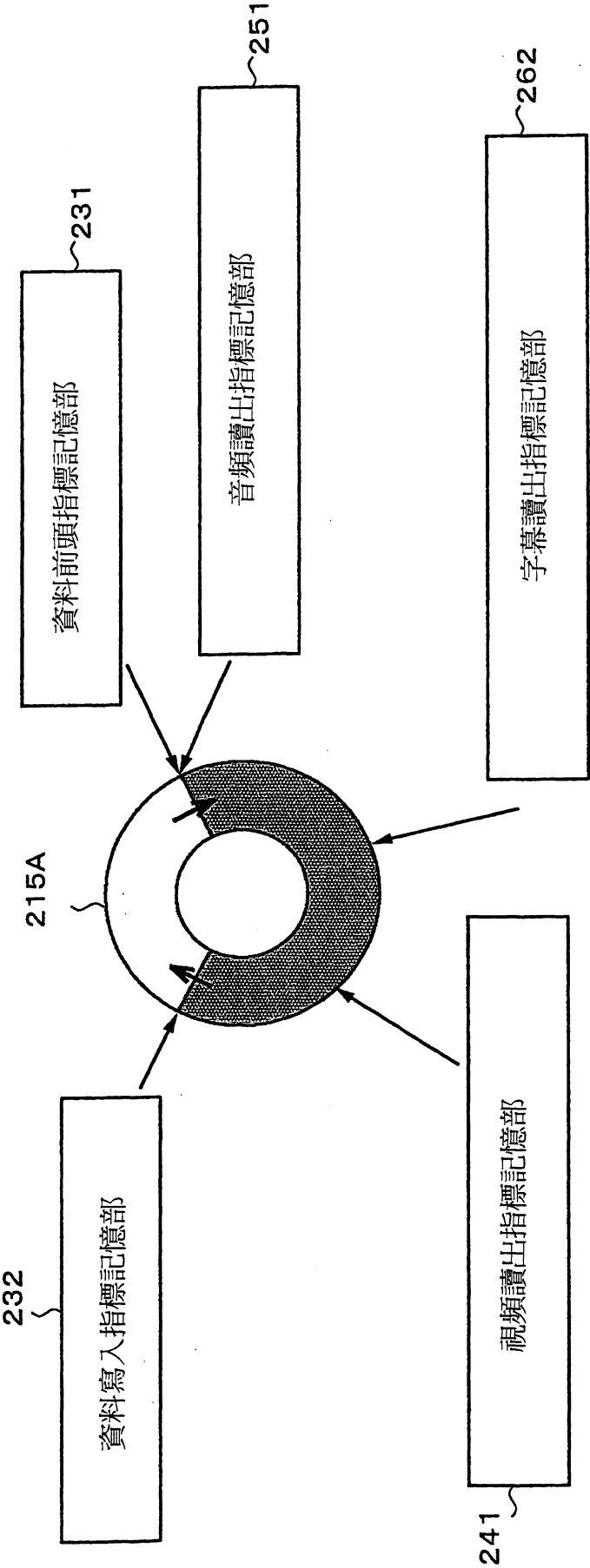
第42圖



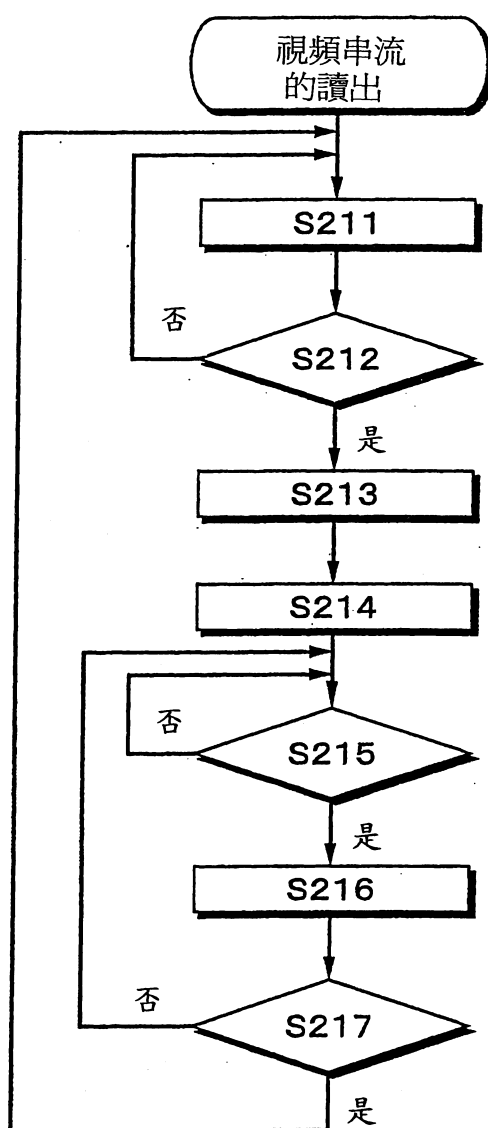
第43圖



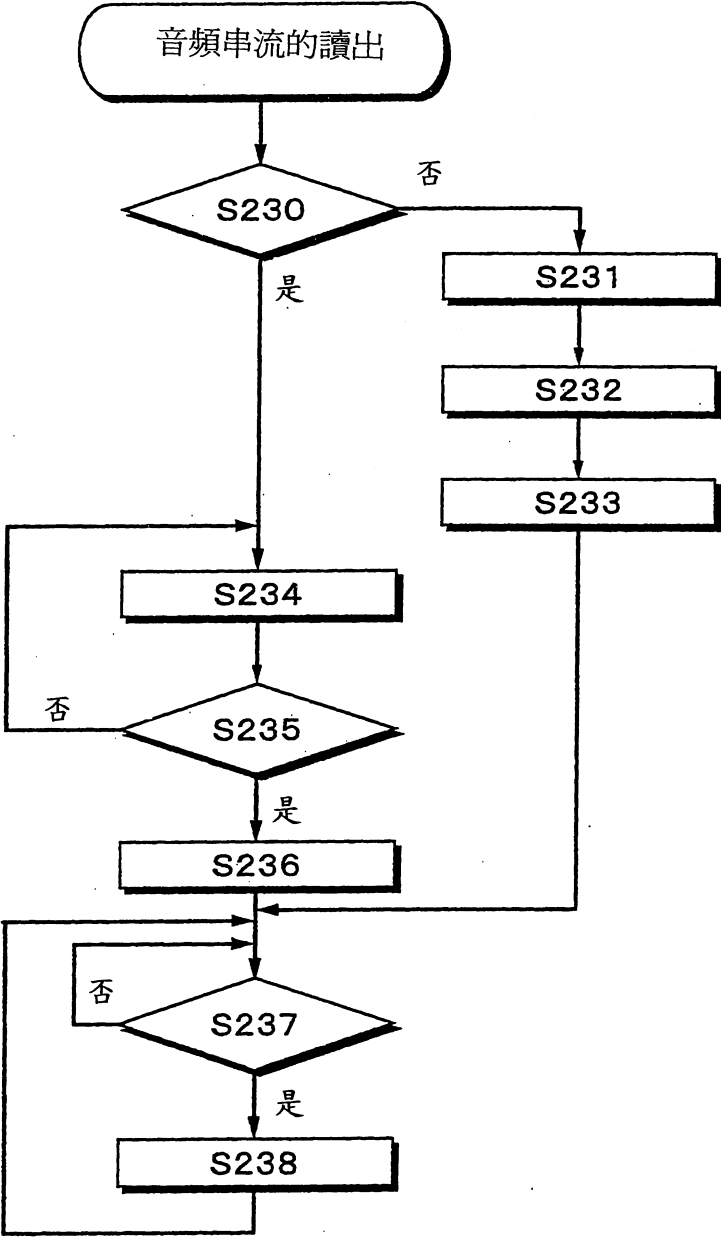
第44圖



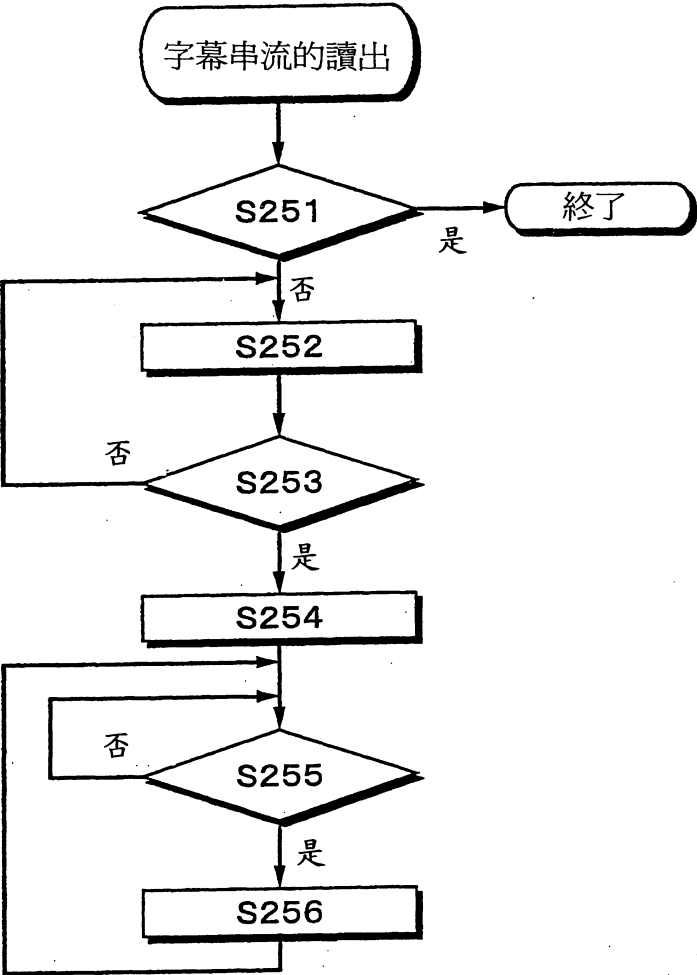
第45圖



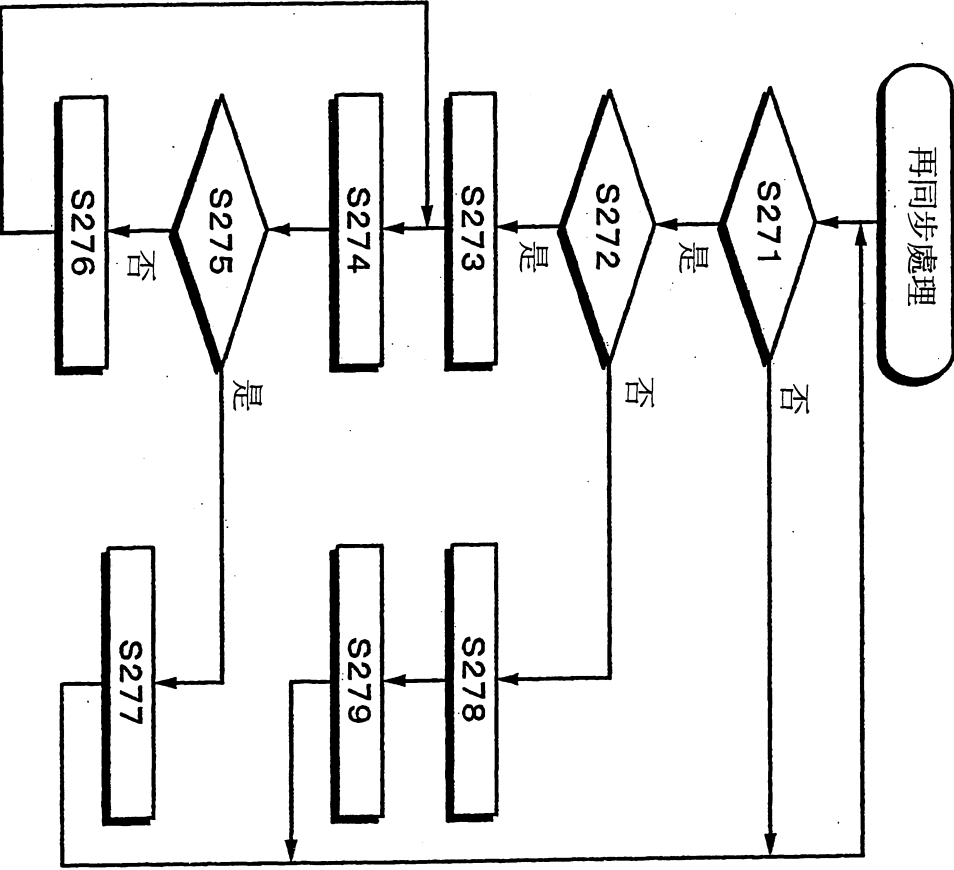
第46圖



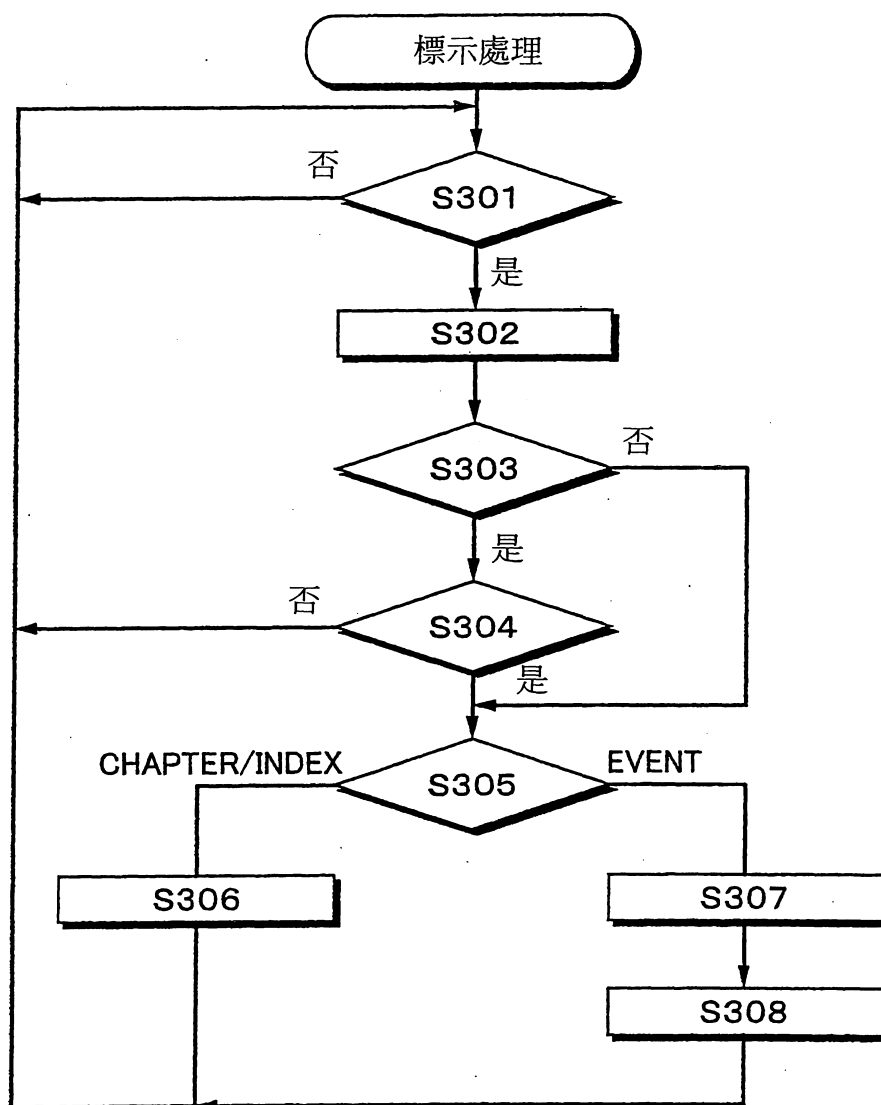
第47圖



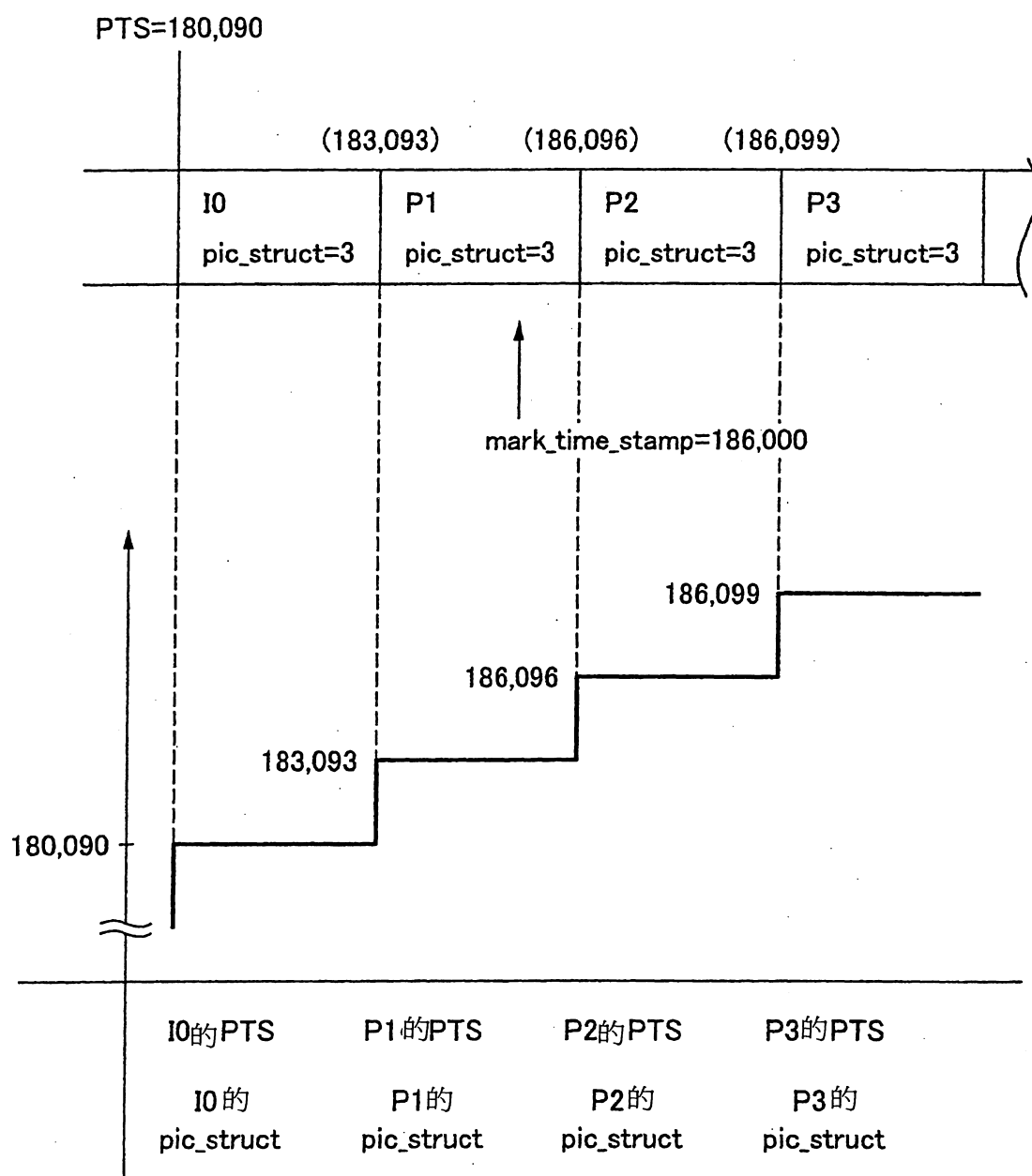
第48圖



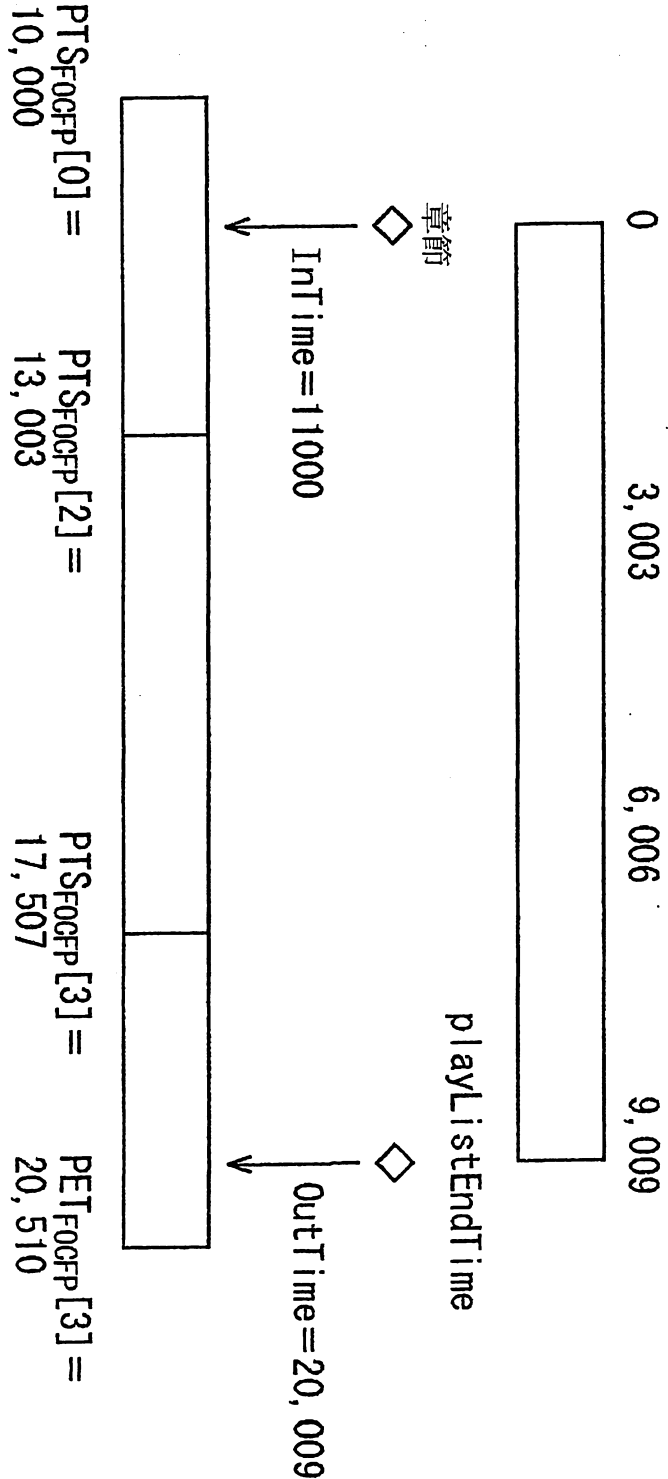
第49圖



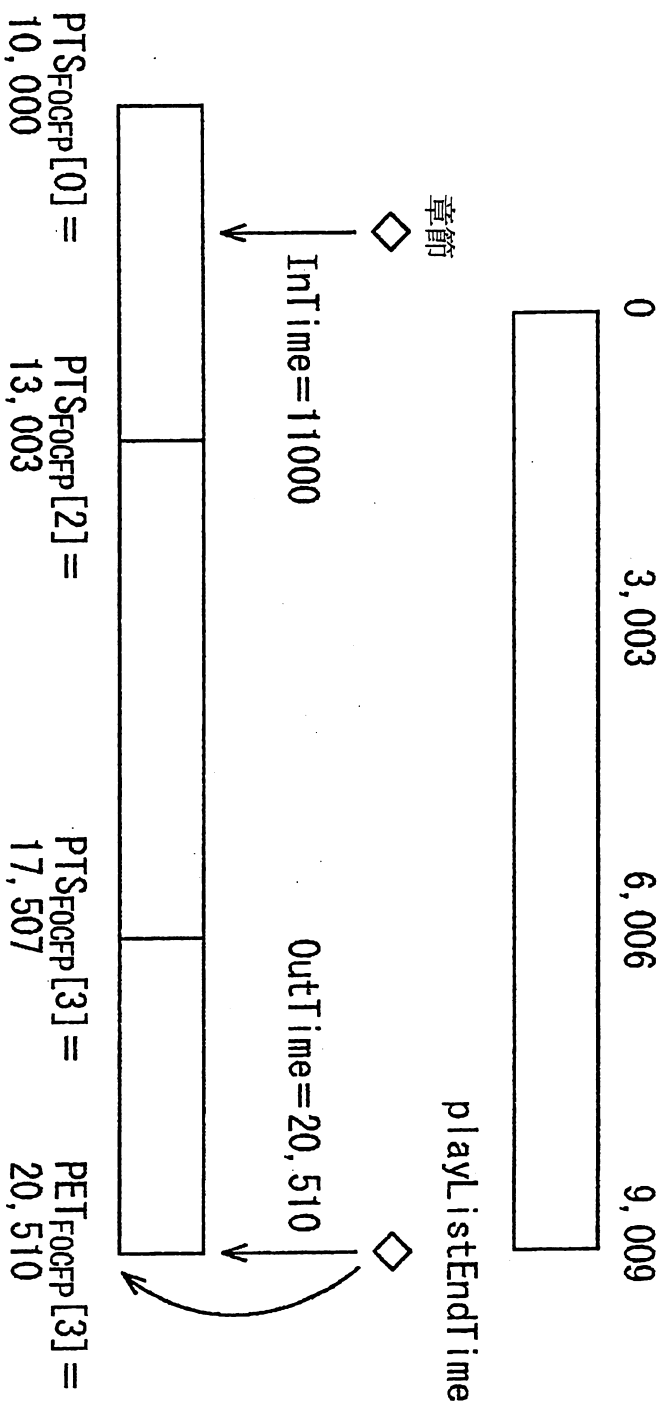
第50圖



第51圖



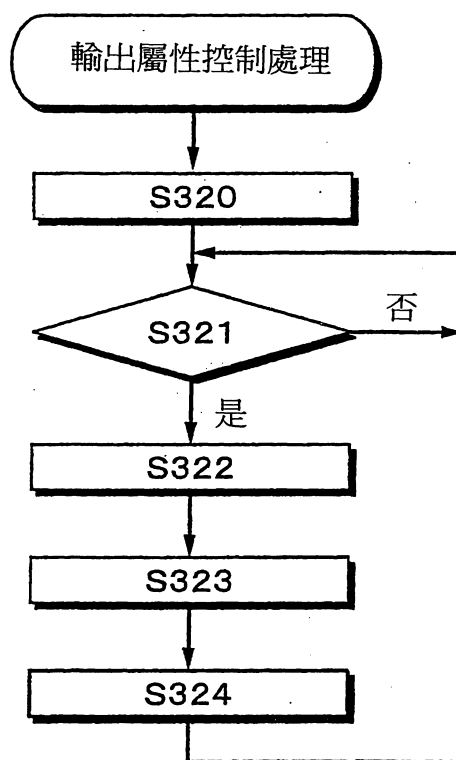
第52圖



第53圖

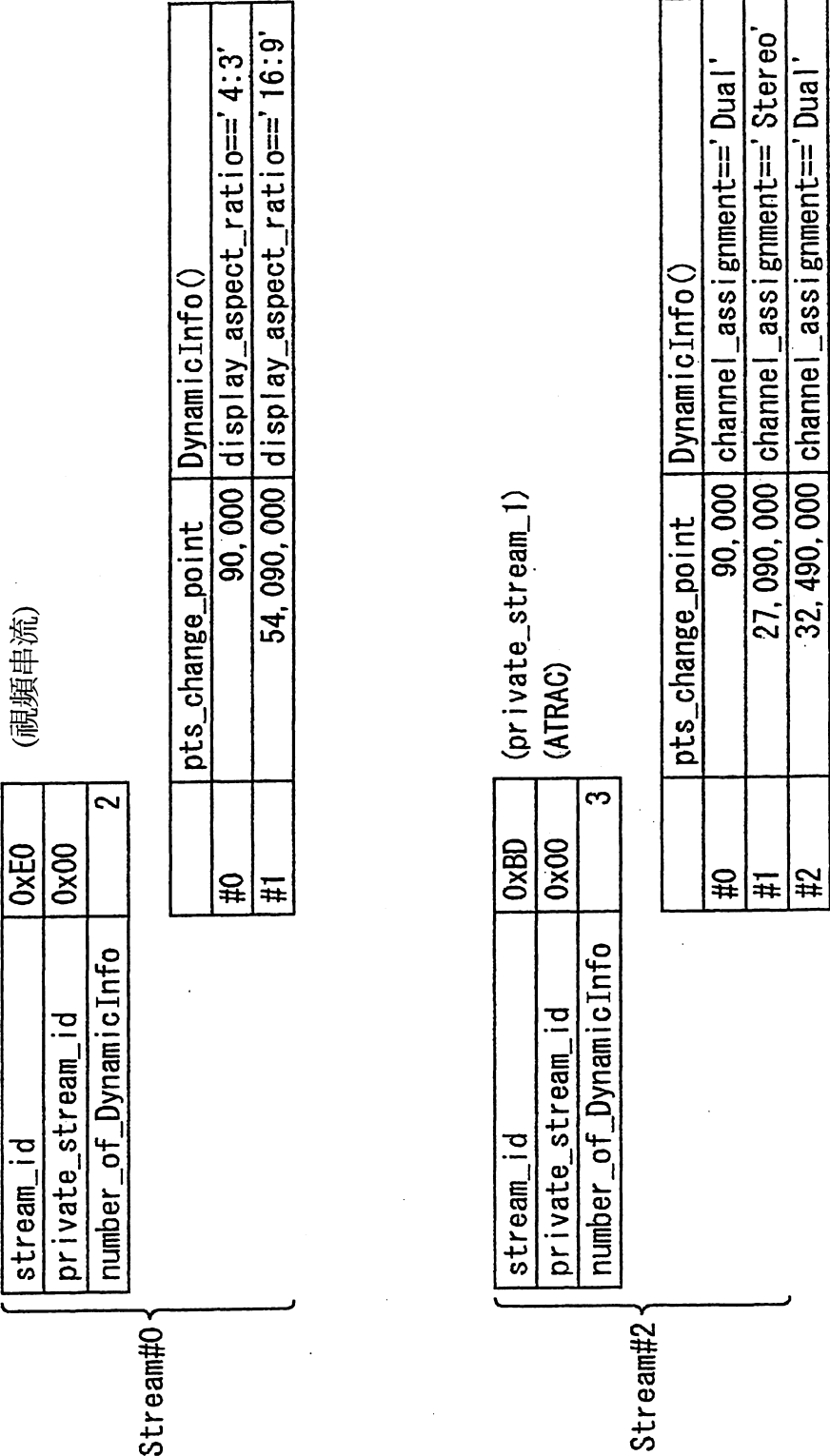
	事例 (23.976p)	事例 (59.94i)
幀速度	5005/240000	4004/240000
以90kHz來表現的 最小事件間隔	7507	6006

第54圖

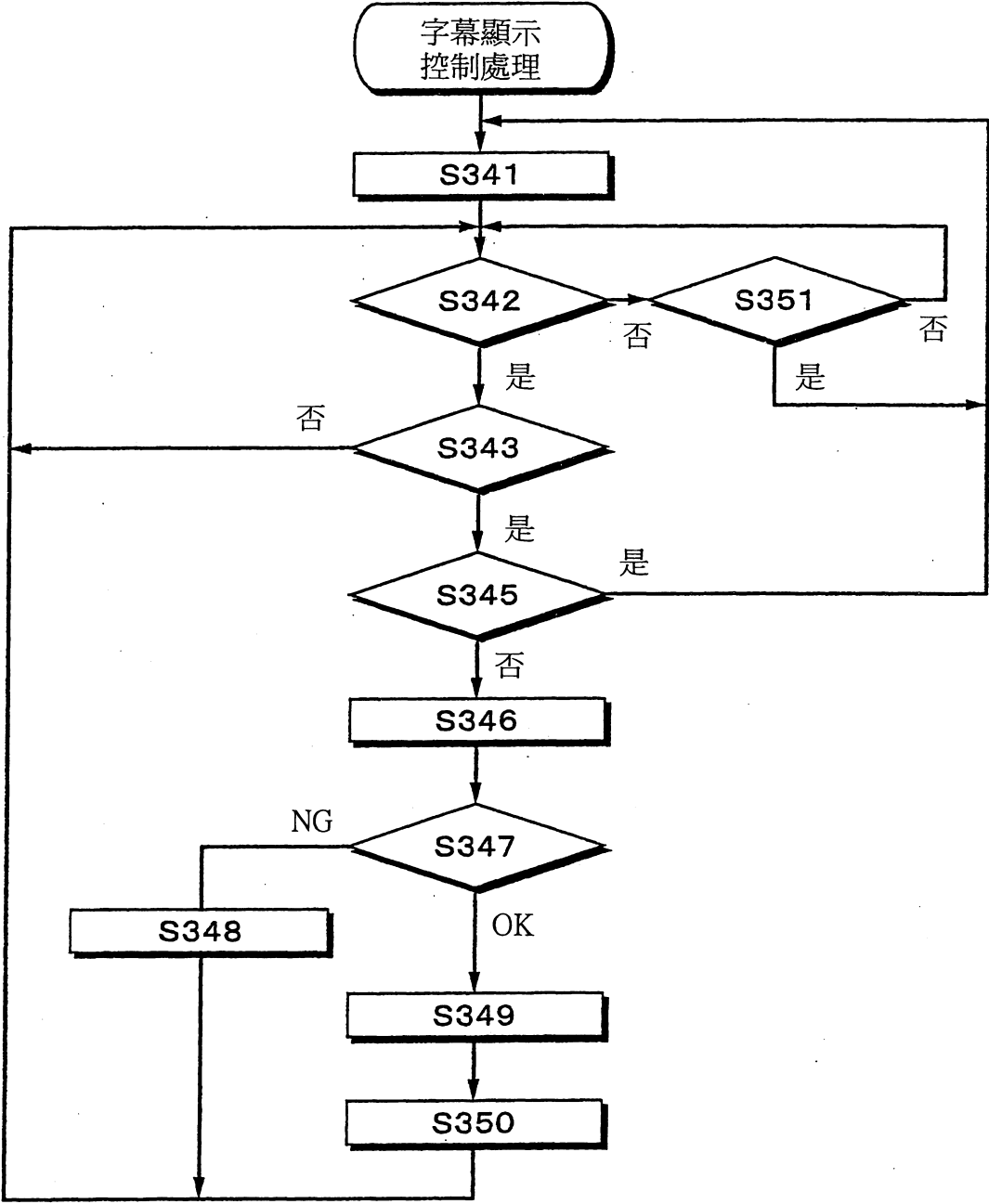


第55圖

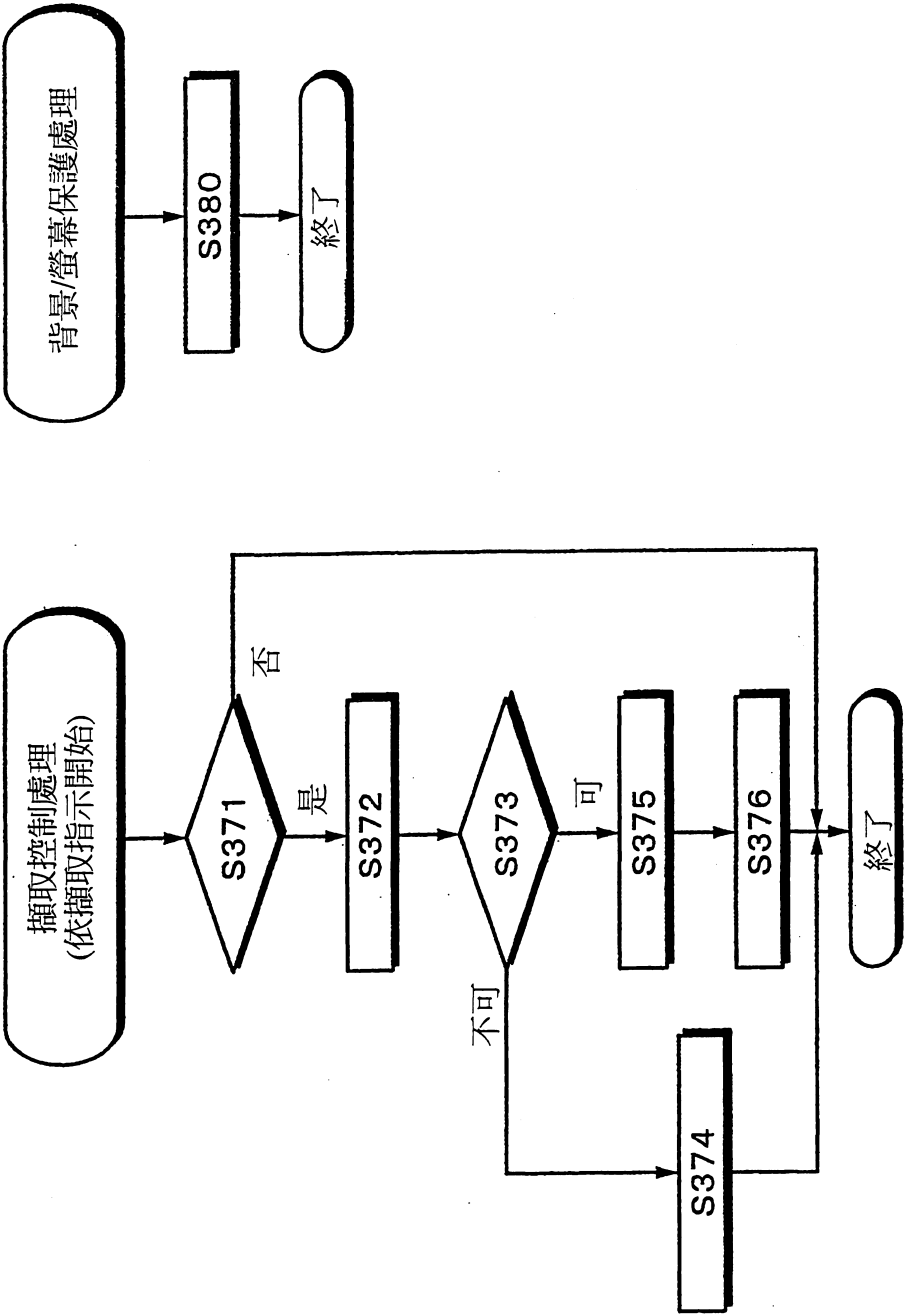
"00003.CLP"



第56圖



第57圖



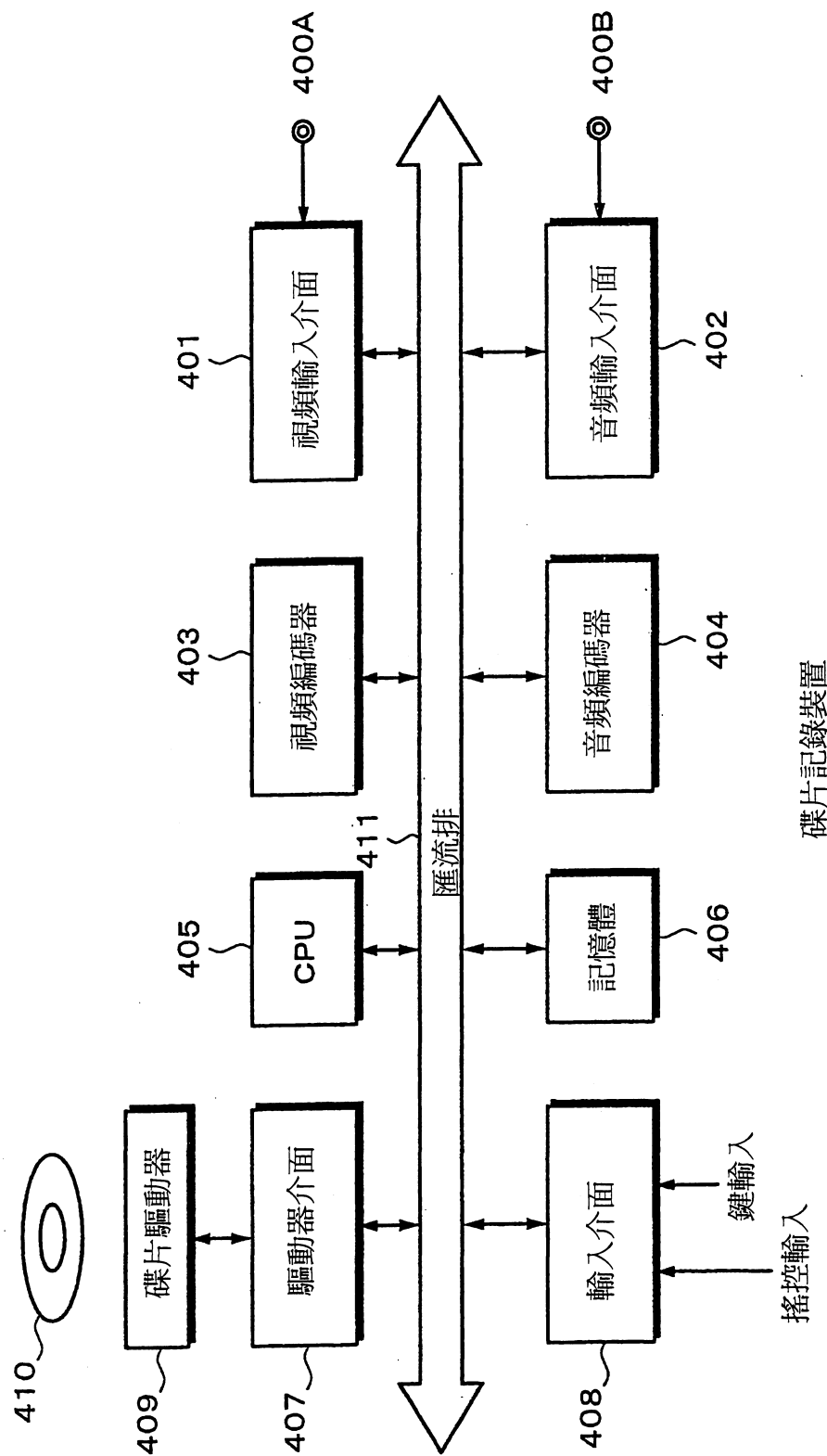
第58圖

語法	位元數	助憶碼
private_stream2_PES_payload() {		
reserved_for_future_use	7	bslbf
capture_enable_ps2	1	uimsbf
video_stream_id	8	uimsbf
1stRef_picture	16	uimsbf
2ndRef_picture	16	uimsbf
3rdRef_picture	16	uimsbf
4thRef_picture	16	uimsbf
au_information()		
grain_variance_information()		
VBI()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
}		

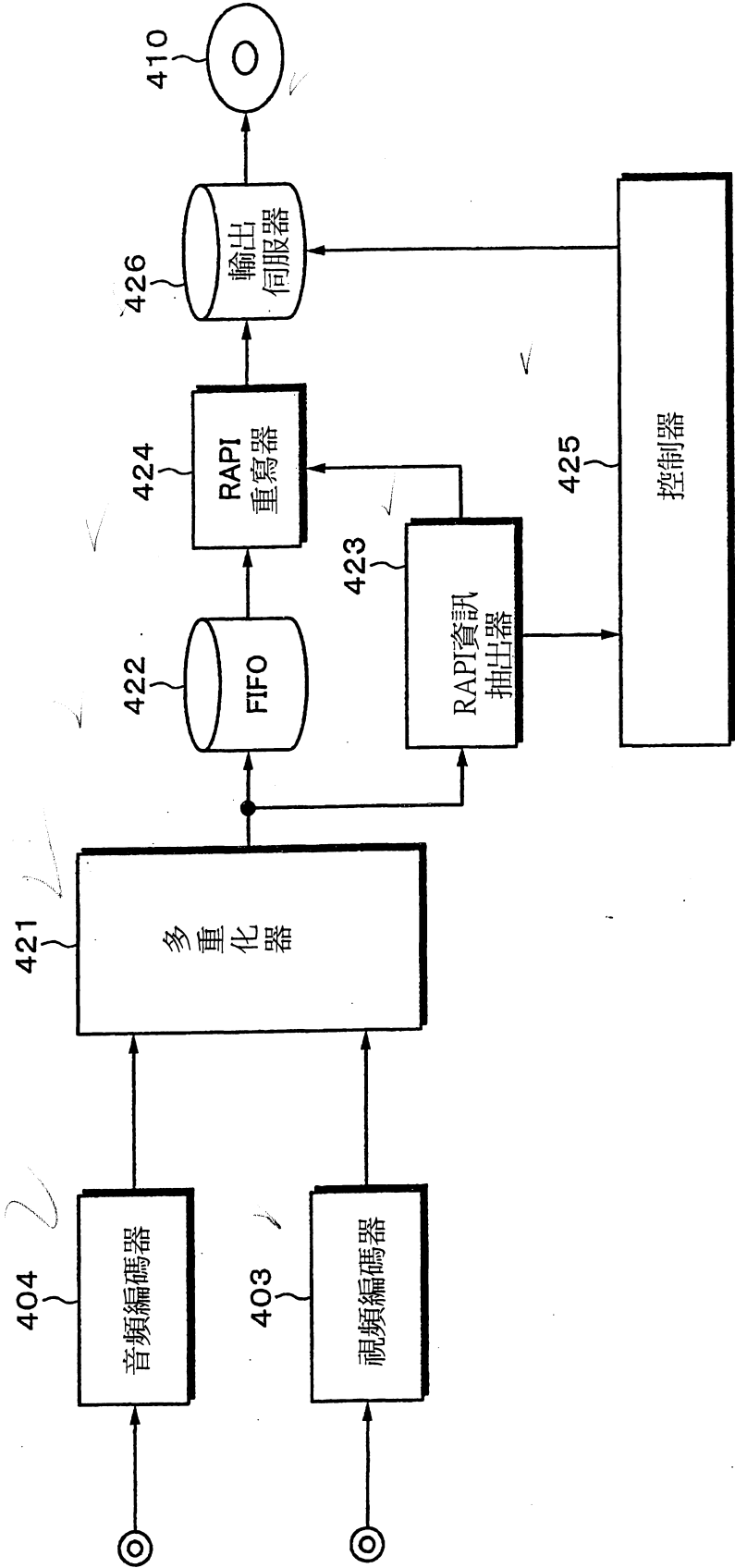
第59圖

語法	位元數	助憶碼
au_information() {		
length	16	uimsbf
reserved_for_word_align	8	bslbf
number_of_access_unit	8	uimsbf
for (i=0; i<number_of_access_unit; i++) {		
reserved	3	bslbf
capture_enable_flag_AU	1	uimsbf
pic_struct_copy	4	uimsbf
au_ref_flag	1	uimsbf
reserved	2	bslbf
AU_length	21	uimsbf
}		uimsbf
}		

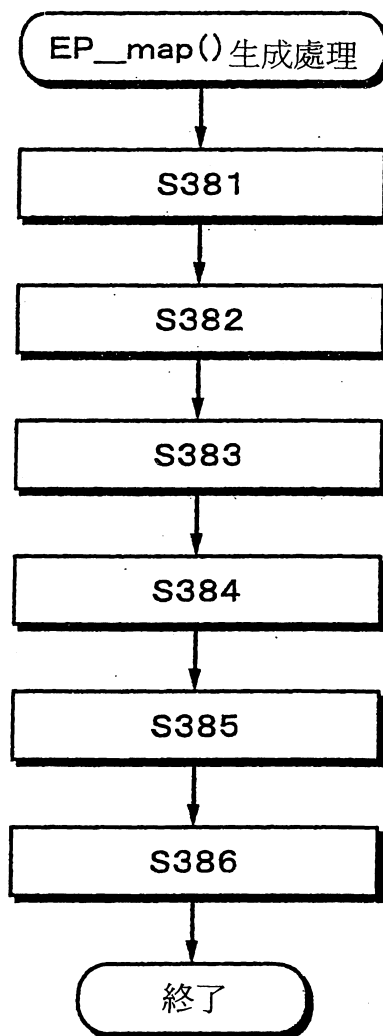
第60圖



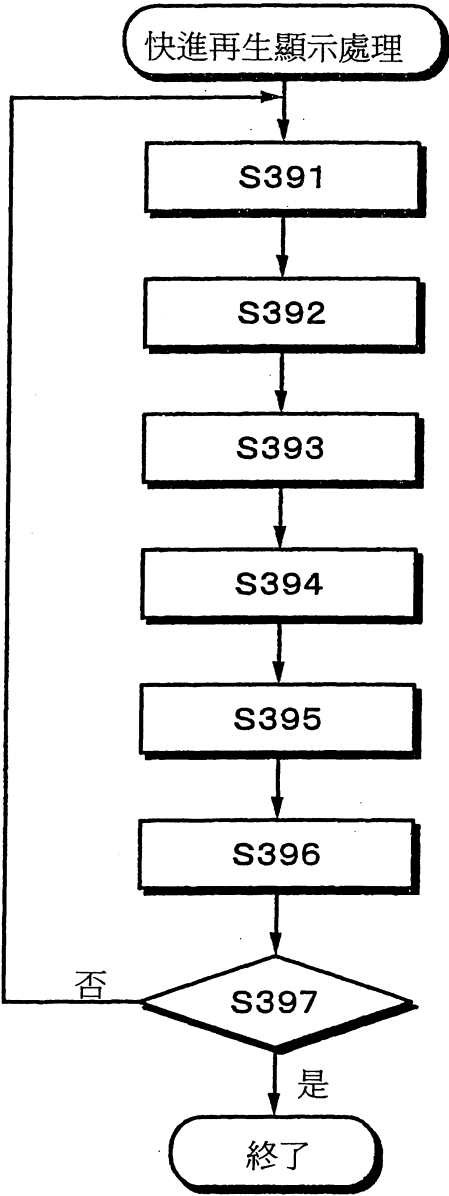
第61圖



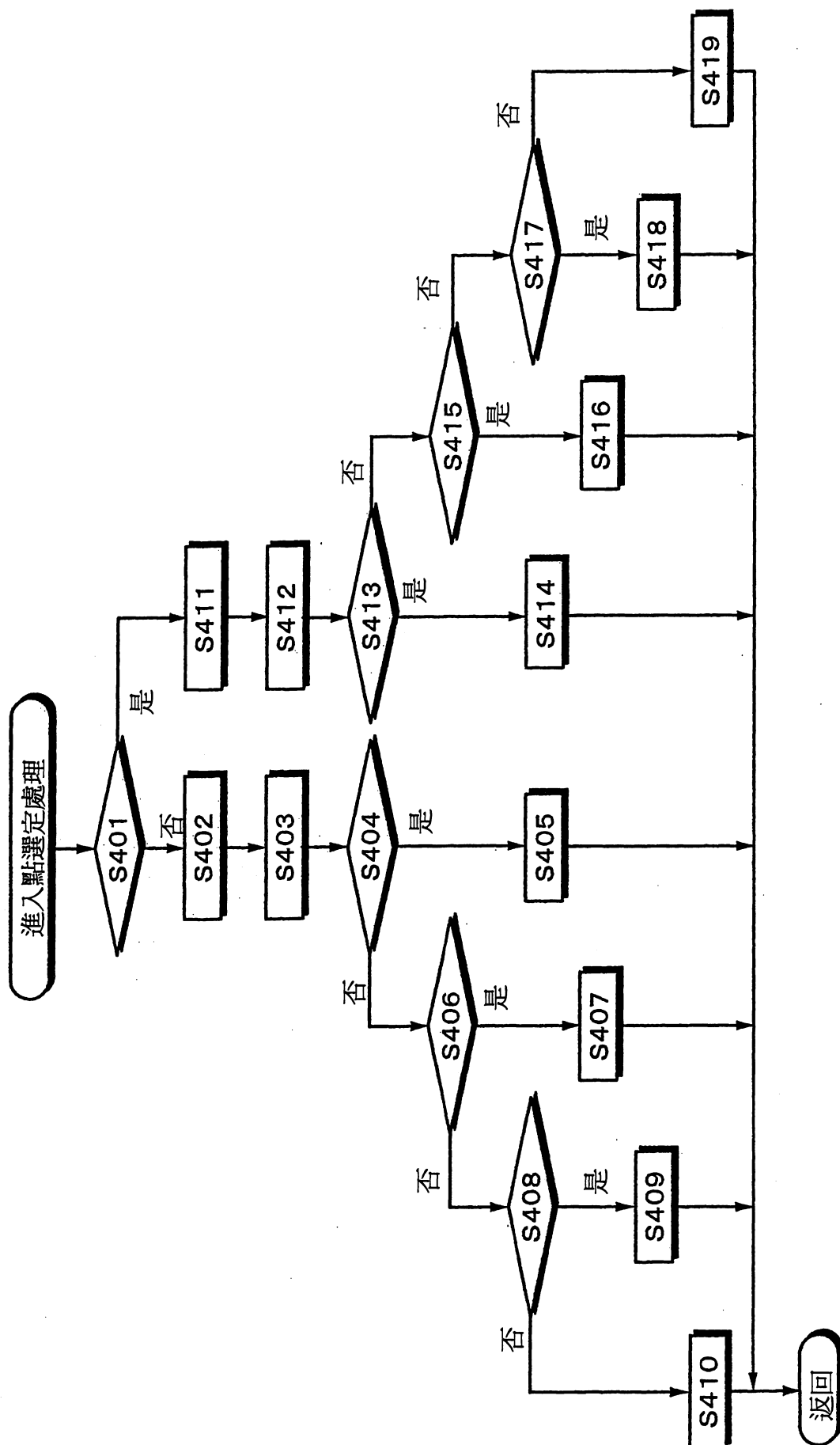
第62圖



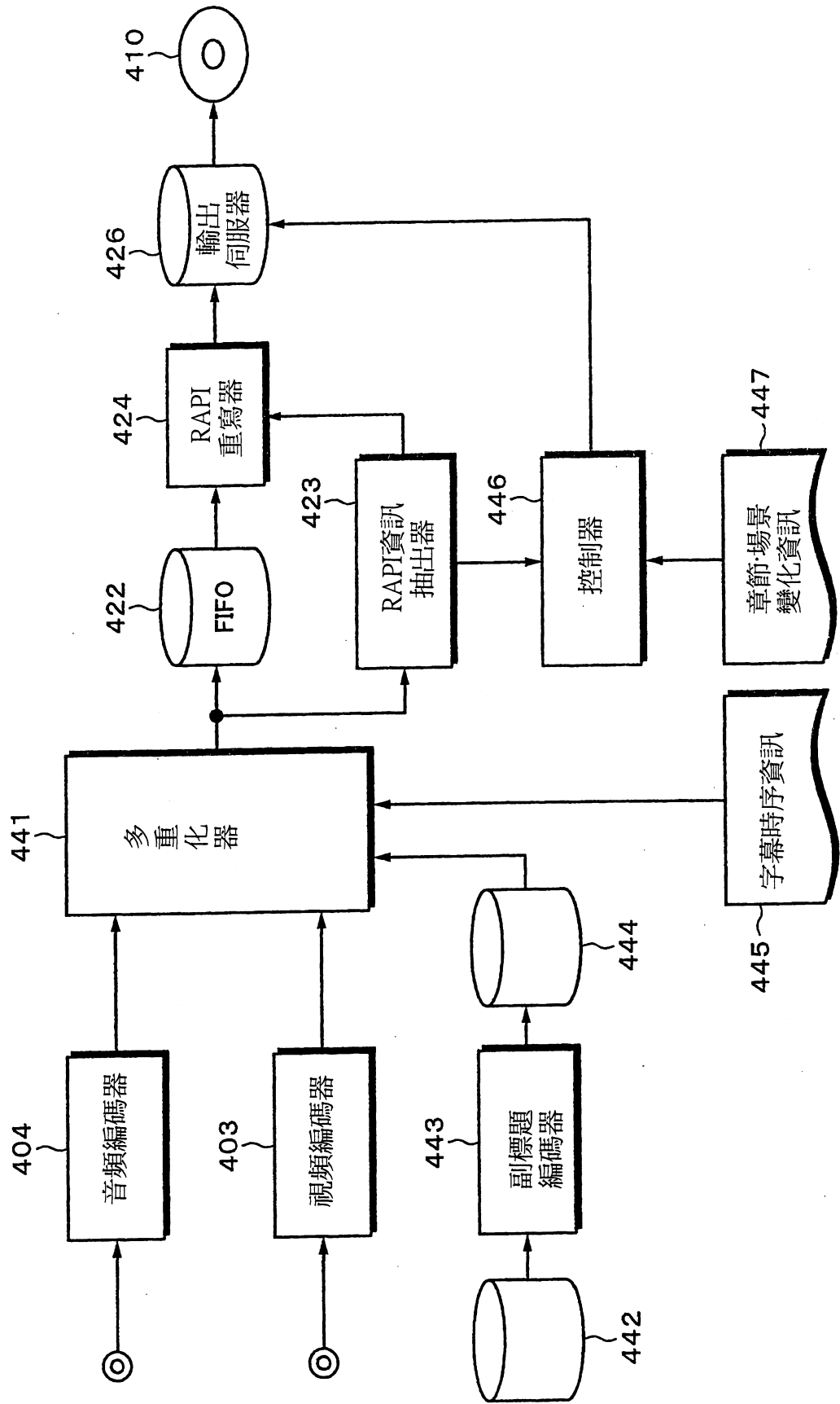
第63圖



第64圖



第65圖



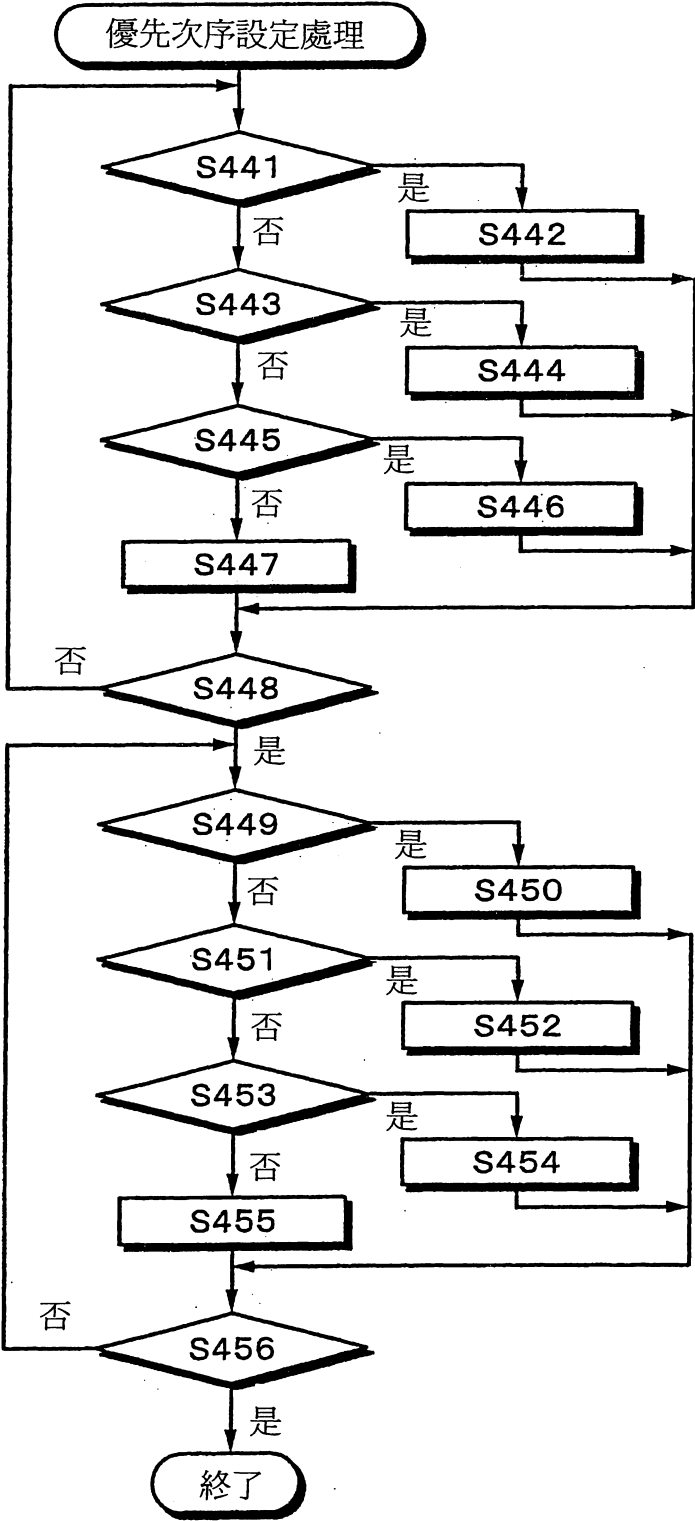
第66圖

EP_map()			
語法	位元數	助憶碼	
EP_map() {			
reserved_for_word_alignment	8	bslbf	
number_of_stream_id_entries	8	uimsbf	
for (k=0;k<number_of_stream_id_entries;k++) {			
stream_id	8	bslbf	
private_stream_id	8	bslbf	
number_of_EP_entries	32	uimsbf	
for (i=0; i< number_of_EP_entries;i++) {			
priority_flag	2	uimsbf	
reserved_for_future_use	14	bslbf	
PTS_EP_start	32	uimsbf	
RPN_EP_start	32	uimsbf	
}			
}			
}			

第67圖

priority_flag	視頻串流時	副標頭串流時
3	章節前頭	章節前頭
2	重要的場景變化	重要的場景變化
1	通常的場景變化	通常的場景變化
0	其他所有的記入項	其他所有的記入項

第68圖



七、指定代表圖：

(一) 本案指定代表圖為：第 (61) 圖

(二) 本代表圖之元件符號簡單說明：

404：音頻編碼器
403：視頻編碼器
421：多重化器
422：FIFO
424：RAPI重寫器
426：輸出伺服器
410：碟片
423：RAPI資訊抽出器
425：控制器

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：無

1299958

(此處由本局於收文時黏貼條碼)

第 94142106 號專利申請案
中文說明書修正頁 月 日修(更)正替換頁

民國 96 年 11 月 12 日

公告本
759626

發明專利說明書

(本申請書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※申請案號：94142106

※申請日期：94 年 11 月 30 日

※IPC 分類：

H04N5/85(2006.01)

G11B20/10(2006.01)

G11B27/00(2006.01)

G11B27/10(2006.01)

H04N5/92(2006.01)

一、發明名稱：

(中) 資料記錄裝置及資料記錄方法，資料處理裝置及資料處理方法，電腦可讀取之記錄媒體，以及資料記錄媒體

(英)

二、申請人：(共 2 人)

1. 姓 名：(中) 新力股份有限公司

(英) SONY CORPORATION

代表人：(中) 1. 中鉢良治

(英) 1. CHUBACHI, RYOJI

地 址：(中) 日本國東京都品川區北品川六丁目七番三五號

(英) 7-35, Kitashinagawa 6-Chome, Shinagawa-Ku, Tokyo, Japan

國籍：(中英) 日本 JAPAN

2. 姓 名：(中) 新力電腦娛樂股份有限公司

(英) SONY COMPUTER ENTERTAINMENT INC.

代表人：(中) 1. 久多良木健

(英) 1. KUTARAGI, KEN

地 址：(中) 日本國東京都港區南青山二丁目六番二一號

(英) 2-6-21, Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan

國籍：(中英) 日本 JAPAN

三、發明人：(共 6 人)

1. 姓 名：(中) 藤波靖

(英) FUJINAMI, YASUSHI

國 籍：(中) 日本

(英) JAPAN

2. 姓 名：(中) 濱田俊也

(英) HAMADA, TOSHIYA

國 籍：(中) 日本

(英) JAPAN

檢測步驟，其係選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定步驟，其係指定在檢測步驟的處理所被檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄步驟，其係和視頻資料一起將在檢測步驟的處理所被檢測出的複數個圖像的位置資訊、及在指定步驟的處理所被指定的圖像的資訊記錄於資料記錄媒體。

本發明之資料處理裝置的特徵係具備：

讀出部，其係構成和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，各個僅離幀內圖像所定的距離之圖像的指定資訊；

選擇部，其係構成根據指定藉由讀出部所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離幀內圖像所定的距離之圖像的指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像；及

再生部，其係構成接續於藉由選擇部所選擇的幀內圖像的複數個圖像中，依照幀內圖像及僅離幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生視頻資料。

僅離上述幀內圖像所定的距離之圖像，可形成從幀內



圖像到離所定的區段數近旁的距離之圖像。

上述所定的區段數可形成按照記錄於資料記錄媒體的視頻資料所再生時被緩衝的容量來設定。

本發明之資料處理方法的特徵係具備：

讀出步驟，其係和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，各個僅離幀內圖像所定的距離之圖像的指定資訊；

選擇步驟，其係根據指定在讀出步驟的處理所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離幀內圖像所定的距離的圖像之指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像；及

再生步驟，其係依照接續於在選擇步驟的處理所選擇的幀內圖像的複數個圖像中，幀內圖像及僅離幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生視頻資料。

本發明之第2程式的特徵係使包含下列步驟的處理執行於電腦：

讀出步驟，其係和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，各個

僅離幀內圖像所定的距離之圖像的指定資訊；

選擇步驟，其係根據指定在讀出步驟的處理所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離幀內圖像所定的距離的圖像之指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像；及

再生步驟，其係依照接續於在選擇步驟的處理所選擇的幀內圖像的複數個圖像中，幀內圖像及僅離幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生視頻資料。

本發明之第2程式記錄媒體的特徵係記錄有包含下列步驟的程式：

讀出步驟，其係和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，各個僅離幀內圖像所定的距離之圖像的指定資訊；

選擇步驟，其係根據指定在讀出步驟的處理所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離幀內圖像所定的距離的圖像之指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像；及

再生步驟，其係依照接續於在選擇步驟的處理所選擇的幀內圖像的複數個圖像中，幀內圖像及僅離幀內圖像所

定的距離的圖像為止的全體圖像來快速前進再生視頻資料。

本發明之資料記錄媒體的特徵係記錄有包含：

接續於從視頻資料中所含之成為隨機存取點的幀內圖像的資訊選擇的幀內圖像之複數個圖像的位置資訊；及

指定複數個圖像的位置資訊中，離幀內圖像所定的距離之圖像的資訊。

本發明之資料構造的特徵係包含：

接續於從視頻資料中所含之成為隨機存取點的幀內圖像的資訊選擇的幀內圖像之複數個圖像的位置資訊；及

指定複數個圖像的位置資訊中，離幀內圖像所定的距離之圖像的資訊。

在本發明的資料記錄裝置及方法、以及第1程式中，視頻資料中所含之成為隨機存取點的幀內圖像的資訊會被選擇，接續於所被選擇的幀內圖像的複數個圖像的位置資訊會被檢測出，所被檢測出的複數個圖像的位置資訊中，僅離幀內圖像所定的距離之圖像位置資訊的圖像會被指定，所被檢測出的複數個圖像的位置資訊、及所被指定的圖像的資訊會與視頻資料一起記錄於資料記錄媒體。

在本發明的資料處理裝置及方法，以及第2程式中，視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，僅離各個幀內圖像所定的距離的圖像的指定資訊會與視頻資料一

起從資料記錄媒體讀出，根據指定所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離幀內圖像所定的距離的圖像之指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像，依照接續於所選擇的幀內圖像的複數個圖像中，幀內圖像及從幀內圖像到離所定的距離的圖像為止的全體圖像來快速前進再生視頻資料。

在本發明的資料記錄媒體及資料構造中，包含：

接續於從視頻資料中所含之成為隨機存取點的幀內圖像的資訊選擇的幀內圖像之複數個圖像的位置資訊；及

指定複數個圖像的位置資訊中，離幀內圖像所定的距離之圖像的資訊。

本發明的資料記錄裝置及資料處理裝置可為獨立的裝置，或進行資料處理的區塊。

若利用本發明，則可進行便利性等高的資料處理。特別是在一定的資料讀出時間下，可實現效率佳的再生處理。

【實施方式】

以下說明本發明的實施形態，其申請專利範圍中所記載的構成要件與發明的實施形態之具體例的對應關係如以下所述。該記載是用以確認支持申請專利範圍中所被記載的發明之具體例為記載於發明的實施形態中者。因此，雖於發明的實施形態中有記載，但作為對應於構成要件者，亦即具體例有未記載於此處時，並非意味該具體例不是對應於該構成要件者。相反的，具體例作為對應於構成要件者而記憶於此時，也非意味該具體例不會對應於該構成要

止的全體圖像來快速前進再生視頻資料。

本發明之資料處理方法的特徵係具備：

讀出步驟(例如，第62圖的流程圖的步驟 S391的處理)，其係和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於複數個幀內圖像的複數個圖像中，各個僅離幀內圖像所定的距離之圖像的指定資訊；

選擇步驟(例如，第62圖的流程圖的步驟 S395的處理)，其係根據指定在讀出步驟的處理所讀出之記錄於資料記錄媒體的視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離幀內圖像所定的距離的圖像之指定資訊，來選擇幀內圖像及接續於幀內圖像的複數個圖像；及

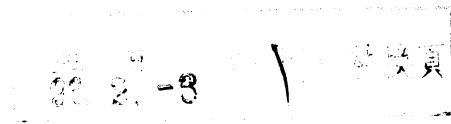
再生步驟(例如，第62圖的流程圖的步驟 S396的處理)，其係依照接續於在選擇步驟的處理所選擇的幀內圖像的複數個圖像中，幀內圖像及僅離幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生上述視頻資料。

另外，有關程式記錄媒體及程式方面，因為與資料記錄方法同樣，因此其說明省略。

以下，參照圖面來說明有關本發明的實施形態。

[硬體構成]

第1圖是表示適用本發明的碟片再生裝置之一實施形



[1stRef_Picture、2ndRef_Picture、3rdRef_Picture、4thRef_Picture
的使用方法]

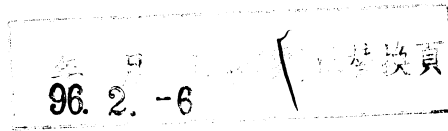
其次，參照第63圖的流程圖來說明有關使用第31圖的
EP_map()之快進(fast-forward)再生處理。

未圖示的使用者會對視頻內容再生程式210指示快進。
播放器控制模組212係由再生中的串流的剪輯資訊檔案
來選擇登錄於EP_map()的再生可能開始位置之一(步驟
S391)，由記述於此的RPN_EP_start來決定僅N-
th_Ref_Picture_copy的大小之讀出資料(步驟S392)。播
放器控制模組212係將此資訊傳給內容資料供給模組213，
且對解碼控制模組214指示快速前進再生。

內容資料供給模組213係使用操作系統201的功能來讀
出儲存有再生對象的基本串流所被多重化的程式串流之剪
輯串流檔案，而使供應給緩衝器控制模組215(步驟S393)
。由於檔案名等既已指示，所以在此不重複。與再生開始
時相異，交付讀入開始的位址及傳送的資料大小，而指示
讀入。

視頻讀出功能部233係將輸入緩衝器控制模組215的多
重化資料予以多重化分離(步驟S394)，只將視頻串流供應
給視頻解碼器控制模組216。此情況，由於為快進的處理
，因此音頻解碼器控制模組217、字幕解碼器控制模組218
、音頻讀出功能部234、字幕讀出功能部233不動作。

在所被輸入的資料中含有1張以上4張以下的參照畫像



前進至步驟 S419。

在步驟 S419中，由於判定在 N 、 $(N+1)$ 、 $(N-1)$ 的所有記入項中 `index_N_minus1` 不是 3 也不是 2，因此播放器控制模組 212 係選定第 N 個的進入點，而終了處理。

亦即，當快進的速度快時，雖 `EP_map()` 內的進入點的選拔間隔會變大，但

亦即，當快進的速度快時，雖 `EP_map()` 內的進入點的選拔間隔會變大，但此刻會優先選擇 `index_N_minus1` 較大者（亦即，主觀畫質高者）。並且，當快進的速度慢時，`index_N_minus1` 的較小者亦選擇。

藉由以上的處理，可在不使主觀畫質降低下，以高速來實現快速前進再生。並且，在低速的快速前進再生中，由於可利用更多的參照畫像來使快速前進再生，因此可抑止再生畫像的畫質降低。

以上的例子，在再生處理中，所被讀出的進入點的資訊是以經常保持一定為前提，但此情況，就處理能力高的碟片再生裝置而言，在多數的進入點被讀出的情況下可提高再生畫像，另一方面，在處理能力低的碟片再生裝置，若使讀入的進入點的資訊大量形成，則處理速度恐會有變慢之虞。於是，對讀入的進入點的資訊設定優先次序，在持有高處理能力的碟片再生裝置中，使用所有進入點的資訊，相反的，在處理能力低的碟片再生裝置中，僅讀出優先次序高的進入點來進行處理。

第 65 圖是用以說明對進入點設定優先次序之碟片記錄

第 60 圖是表示碟片記錄裝置的硬體構成例的方塊圖

第 61 圖是用以說明藉由第 60 圖的碟片記錄裝置來實現的功能的方塊圖。

第 62 圖是用以說明 Ep_map 生成處理的流程圖。

第 63 圖是用以說明快速前進再生處理的流程圖。

第 64 圖是用以說明進入點選定處理的流程圖。

第 65 圖是用以說明藉由第 60 圖的碟片記錄裝置來實現的其他功能的方塊圖。

第 66 圖是表示 EP_map() 的其他語法。

第 67 圖是用以說明第 65 圖的 priority_flag。

第 68 圖是用以說明優先次序設定處理的流程圖。

【主要元件符號說明】

101：碟片

102：碟片驅動器

111：匯流排

112：CPU

113：記憶體

114：驅動器介面

115：輸入介面

116：視頻解碼器

116A：視頻解碼引擎

116B：DPB(Decoded Picture Buffer)

116B：-0～116B-n DPB-0～DPB-n

五、中文發明摘要

發明之名稱：資料記錄裝置及資料記錄方法，資料處理裝置及資料處理方法，電腦可讀取之記錄媒體，以及資料記錄媒體

實現不使畫質低落的變速再生。控制器425係針對來自 RAPI 資訊抽出器423的 RAPI，選出其位址緊接著之後的幀內圖像所持的 PTS，幀內圖像(Intra Picture)及接續的第2張、第3張、第4張參照畫像的終了位置的其中之一，而作成剪輯資訊檔案內的 EP_map()，儲存於輸出伺服器426。亦即，控制器425係將4點的參照畫像終了位置(1stRef_Picture、2ndRef_Picture、3rdRef_Picture、4thRef_Picture)中，接近所定的區段數(在編碼處理中，可彙整讀出的區段數)的值拷貝成 N-th_Ref_picture_copy，根據 N-th_Ref_picture_copy 來決定 index_minus1，使記錄於碟片。

六、英文發明摘要

發明之名稱：

(1)

十、申請專利範圍

第 94142106 號專利申請案

中文申請專利範圍修正本

民國 96 年 11 月 **12** 日修正

1. 一種資料記錄裝置，其特徵係具備：

檢測部，其係構成選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定部，其係構成指定藉由上述檢測部所檢測出的複數個圖像的位置資訊中，僅離上述幀內圖像所定的距離之圖像位置資訊的圖像；

抽出部，其係抽出藉由上述檢測部所檢測出的複數個圖像的位置資訊中，對應於所定的區段數之圖像的位置資訊；及

記錄部，其係構成和上述視頻資料一起將藉由上述檢測部所檢測出的複數個圖像的位置資訊、及藉由上述指定部所指定的圖像的資訊記錄於資料記錄媒體。

上述指定部係根據藉由上述抽出部所抽出的複數個圖像的位置資訊中，對應於所定的區段數之圖像的位置資訊，來指定藉由上述檢測部所檢測出的複數個圖像的位置資訊中，僅離上述幀內圖像所定的距離之圖像的位置資訊的圖像；

上述記錄部更將藉由上述抽出部所抽出的複數個圖像的位置資訊中，對應於上述所定的區段數之圖像位置的資

訊記錄於資料記錄媒體。

2.如申請專利範圍第1項之資料記錄裝置，其中上述所定的區段數係按照再生記錄於上述資料記錄媒體的上述視頻資料的裝置所緩衝視頻資料的容量來設定。

3.一種資料記錄方法，其特徵係具備：

檢測步驟，其係選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定步驟，其係指定在上述檢測步驟的處理所被檢測出的複數個圖像的位置資訊中，僅離上述幀內圖像所定的距離之圖像位置資訊的圖像；及

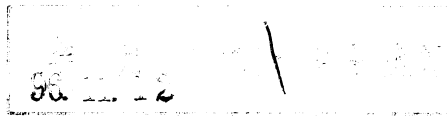
記錄步驟，其係和上述視頻資料一起將在上述檢測步驟的處理所被檢測出的複數個圖像的位置資訊、及在上述指定步驟的處理所被指定的圖像的資訊記錄於資料記錄媒體。

4.一種電腦可讀取之記錄媒體，其特徵係記錄有包含下列步驟的程式：

檢測步驟，其係選擇視頻資料中所含之成為隨機存取點的幀內圖像的資訊，檢測出接續於所被選擇的幀內圖像之複數個圖像的位置資訊；

指定步驟，其係指定在上述檢測步驟的處理所被檢測出的複數個圖像的位置資訊中，僅離上述幀內圖像所定的距離之圖像位置資訊的圖像；及

記錄步驟，其係和上述視頻資料一起將在上述檢測步



(3)

驟的處理所被檢測出的複數個圖像的位置資訊、及在上述指定步驟的處理所被指定的圖像的資訊記錄於資料記錄媒體。

5.一種資料處理裝置，其特徵係具備：

讀出部，其係構成和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於上述複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於上述複數個幀內圖像的複數個圖像中，各個僅離上述幀內圖像所定的距離之圖像的指定資訊；

選擇部，其係構成根據指定藉由上述讀出部所讀出之記錄於上述資料記錄媒體的成為視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離上述幀內圖像所定的距離的圖像之指定資訊，來選擇上述幀內圖像及接續於上述幀內圖像的複數個圖像；及

再生部，其係構成接續於藉由上述選擇部所選擇的上述幀內圖像的複數個圖像中，依照上述幀內圖像及僅離上述幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生上述視頻資料。

6.如申請專利範圍第5項之資料處理裝置，其中僅離上述幀內圖像所定的距離之圖像，係從上述幀內圖像到離所定的區段數近旁的距離之圖像。

7.如申請專利範圍第6項之資料處理裝置，其中上述所定的區段數係按照記錄於上述資料記錄媒體的上述視頻



(4)

資料所再生時被緩衝的容量來設定。

8.一種資料處理方法，其特徵係具備：

讀出步驟，其係和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於上述複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於上述複數個幀內圖像的複數個圖像中，各個僅離上述幀內圖像所定的距離之圖像的指定資訊；

選擇步驟，其係根據指定在上述讀出步驟的處理所讀出之記錄於上述資料記錄媒體的成為視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離上述幀內圖像所定的距離的圖像之指定資訊，來選擇上述幀內圖像及接續於上述幀內圖像的複數個圖像；及

再生步驟，其係依照接續於在上述選擇步驟的處理所選擇的上述幀內圖像的複數個圖像中，上述幀內圖像及僅離上述幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生上述視頻資料。

9.一種電腦可讀取之記錄媒體，其特徵係記錄有包含下列步驟的程式：

讀出步驟，其係和視頻資料一起由資料記錄媒體來讀出視頻資料中所含之成為隨機存取點的複數個幀內圖像、與分別接續於上述複數個幀內圖像的複數個圖像的位置資訊、及指定分別接續於上述複數個幀內圖像的複數個圖像

(5)

中，各個僅離上述幀內圖像所定的距離之圖像的指定資訊；

選擇步驟，其係根據指定在上述讀出步驟的處理所讀出之記錄於上述資料記錄媒體的成為視頻資料的隨機存取點的複數個幀內圖像、與接續於該幀內圖像的複數個圖像中，僅離上述幀內圖像所定的距離的圖像之指定資訊，來選擇上述幀內圖像及接續於上述幀內圖像的複數個圖像；及

再生步驟，其係依照接續於在上述選擇步驟的處理所選擇的上述幀內圖像的複數個圖像中，上述幀內圖像及僅離上述幀內圖像所定的距離的圖像為止的全體圖像來快速前進再生上述視頻資料。

10.一種資料記錄媒體，其特徵係記錄有包含：

接續於從視頻資料中所含之成為隨機存取點的幀內圖像的資訊選擇的幀內圖像之複數個圖像的位置資訊；及

指定上述複數個圖像的位置資訊中，僅離上述幀內圖像所定的距離之圖像的資訊之資料。