

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3993626号  
(P3993626)

(45) 発行日 平成19年10月17日(2007.10.17)

(24) 登録日 平成19年8月3日(2007.8.3)

(51) Int. Cl.

F I

HO4N 5/92 (2006.01)  
G11B 20/12 (2006.01)  
G11B 20/10 (2006.01)  
G11B 27/00 (2006.01)  
HO4N 5/91 (2006.01)

HO4N 5/92 C  
G11B 20/12  
G11B 20/10 321Z  
G11B 20/10 E  
G11B 27/00 D

請求項の数 6 (全 77 頁) 最終頁に続く

(21) 出願番号 特願2006-516863 (P2006-516863)  
(86) (22) 出願日 平成16年7月5日(2004.7.5)  
(65) 公表番号 特表2007-521737 (P2007-521737A)  
(43) 公表日 平成19年8月2日(2007.8.2)  
(86) 国際出願番号 PCT/JP2004/009873  
(87) 国際公開番号 W02005/004478  
(87) 国際公開日 平成17年1月13日(2005.1.13)  
審査請求日 平成18年10月17日(2006.10.17)  
(31) 優先権主張番号 60/485,207  
(32) 優先日 平成15年7月3日(2003.7.3)  
(33) 優先権主張国 米国(US)

早期審査対象出願

前置審査

(73) 特許権者 000005821  
松下電器産業株式会社  
大阪府門真市大字門真1006番地  
(74) 代理人 100090446  
弁理士 中島 司朗  
(72) 発明者 マクロッサン ジョセフ  
アメリカ合衆国、カリフォルニア州 91  
501、#シーバーバンク、サンホセ  
アベニュー 472  
(72) 発明者 岡田 智之  
奈良県奈良市富雄元町1-8-19-30  
3  
(72) 発明者 持永 和寛  
兵庫県伊丹市千僧3-40

最終頁に続く

(54) 【発明の名称】 再生装置、集積回路、プログラム、再生方法

(57) 【特許請求の範囲】

【請求項1】

動画ストリーム、グラフィクスストリームが多重化されたデジタルストリームについての再生装置であって、

動画ストリームをデコードして動画像を得るビデオデコーダと、

グラフィクスを、動画像に合成して表示させるグラフィクスデコーダとを備え、

グラフィクスストリームは、複数パケットからなるパケット列であり、

パケットには、グラフィクスデータを格納したデータパケットと、制御情報を格納した制御パケットとがあり、

前記グラフィクスデコーダは、

動画ストリームの現在の再生時点が、グラフィクスデータを格納したパケットのデコードタイムスタンプに示される時点に達すれば、グラフィクスデータのデコードを開始し、同パケットのプレゼンテーションタイムスタンプの値に示される時点に達すれば、グラフィクスデータのデコードを終了するプロセッサと、

制御パケットのプレゼンテーションタイムスタンプに従い、デコードされたグラフィクスデータを、動画ストリームに合成して表示させるコントローラと、

圧縮状態のグラフィクスデータを格納するコーデッドデータバッファと、

プロセッサのデコードにより得られた非圧縮状態のグラフィクスデータを格納するオブジェクトバッファとを備え、

再生装置は、オブジェクトバッファに格納された非圧縮状態のグラフィクスデータのう

10

20

ち、表示に供されるべきものを格納するグラフィックスプレーンを備え、

前記コントローラは、メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、

メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、含まれている場合、動画ストリームの現在の再生時点が、制御パケットのデコードタイムスタンプの値に到達した段階で、コーデッドデータバッファ、オブジェクトバッファ、グラフィックスプレーンをクリアすることでメモリ管理を開始して、制御パケットから制御情報を読み出す

ことを特徴とする再生装置。

【請求項 2】

10

制御パケットのプレゼンテーションタイムスタンプの値は、制御パケットのデコードタイムスタンプの値に所定の値を足し合わせた値であり、

所定の値は、画面クリアに要する時間、及び、グラフィクスデータのデコードに要する時間のうち長い方と、画面へのグラフィクスデータへの書き込みに要する時間とに基づいており、

前記コントローラは、現在の再生時点が、制御パケットのプレゼンテーションタイムスタンプの値になれば、グラフィックスプレーンの格納内容の表示を行う。

ことを特徴とする請求項 1 記載の再生装置。

【請求項 3】

動画ストリーム、グラフィクスストリームが多重化されたデジタルストリームを再生する制御を、再生装置内で実行する集積回路であって、

20

動画ストリームをデコードして動画像を得るビデオデコーダと、

グラフィクスを、動画像に合成して表示させるグラフィクスデコーダとを備え、

グラフィクスストリームは、複数パケットからなるパケット列であり、

パケットには、グラフィクスデータを格納したデータパケットと、制御情報を格納した制御パケットとがあり、

前記グラフィクスデコーダは、

動画ストリームの現在の再生時点が、グラフィクスデータを格納したパケットのデコードタイムスタンプに示される時点に達すれば、グラフィクスデータのデコードを開始し、同パケットのプレゼンテーションタイムスタンプの値に示される時点に達すれば、グラフィクスデータのデコードを終了するプロセッサと、

30

制御パケットのプレゼンテーションタイムスタンプに従い、デコードされたグラフィクスデータを、動画ストリームに合成して表示させるコントローラと、

圧縮状態のグラフィクスデータを格納するコーデッドデータバッファと、

プロセッサのデコードにより得られた非圧縮状態のグラフィクスデータを格納するオブジェクトバッファとを備え、

再生装置は、オブジェクトバッファに格納された非圧縮状態のグラフィクスデータのうち、表示に供されるべきものを格納するグラフィックスプレーンを備え、

前記コントローラは、メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、

40

メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、含まれている場合、動画ストリームの現在の再生時点が、制御パケットのデコードタイムスタンプの値に到達した段階で、コーデッドデータバッファ、オブジェクトバッファ、グラフィックスプレーンをクリアすることでメモリ管理を開始して、制御パケットから制御情報を読み出す

ことを特徴とする集積回路。

【請求項 4】

動画ストリーム、グラフィクスストリームが多重化されたデジタルストリームについての再生をコンピュータに実行させるプログラムであって、

動画ストリームをデコードして動画像を得る処理、

50

グラフィクスを描画して、動画像に合成し表示させる処理をコンピュータに行わせるものであり、

グラフィクスストリームは、複数パケットからなるパケット列であり、

パケットには、グラフィクスデータを格納したデータパケットと、制御情報を格納した制御パケットがあり、

前記コンピュータは、

圧縮状態のグラフィクスデータを格納するコーデッドデータバッファと、

プロセッサのデコードにより得られた非圧縮状態のグラフィクスデータを格納するオブジェクトバッファと、

オブジェクトバッファに格納された非圧縮状態のグラフィクスデータのうち、表示に供されるべきものを格納するグラフィックスプレーンとを備え、 10

前記コンピュータによりなされるグラフィクス描画は、

メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、含まれている場合、動画ストリームの現在の再生時点が、制御パケットのデコードタイムスタンプの値に到達した段階で、コーデッドデータバッファ、オブジェクトバッファ、グラフィックスプレーンをクリアすることでメモリ管理を開始して、制御パケットから制御情報を読み出す処理と、

動画ストリームの現在の再生時点が、グラフィクスデータを格納したパケットのデコードタイムスタンプに示される時点に達すれば、グラフィクスデータのデコードを開始し、同パケットのプレゼンテーションタイムスタンプの値に示される時点に達すれば、グラフィクスデータのデコードを終了する処理と、 20

制御パケットのプレゼンテーションタイムスタンプに従い、デコードされたグラフィクスデータを、動画ストリームに合成して表示させる処理とを含む

ことを特徴とするプログラム。

#### 【請求項5】

動画ストリーム、グラフィクスストリームが多重化されたデジタルストリームの再生をコンピュータ上で実行する再生方法であって、

動画ストリームをデコードして動画像を得る処理、

グラフィクスを描画して、動画像に合成して表示させる処理を備え、

グラフィクスストリームは、複数パケットからなるパケット列であり、 30

パケットには、グラフィクスデータを格納したデータパケットと、制御情報を格納した制御パケットがあり、

前記コンピュータは、

圧縮状態のグラフィクスデータを格納するコーデッドデータバッファと、

プロセッサのデコードにより得られた非圧縮状態のグラフィクスデータを格納するオブジェクトバッファと、

オブジェクトバッファに格納された非圧縮状態のグラフィクスデータのうち、表示に供されるべきものを格納するグラフィックスプレーンとを備え、

前記グラフィクス描画は、

メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、含まれている場合、動画ストリームの現在の再生時点が、制御パケットのデコードタイムスタンプの値に到達した段階で、コーデッドデータバッファ、オブジェクトバッファ、グラフィックスプレーンをクリアすることでメモリ管理を開始して、制御パケットから制御情報を読み出す処理と、 40

動画ストリームの現在の再生時点が、グラフィクスデータを格納したパケットのデコードタイムスタンプに示される時点に達すれば、グラフィクスデータのデコードを開始し、同パケットのプレゼンテーションタイムスタンプの値に示される時点に達すれば、グラフィクスデータのデコードを終了する処理と、

制御パケットのプレゼンテーションタイムスタンプに従い、デコードされたグラフィクスデータを、動画ストリームに合成して表示させる処理とを含む 50

ことを特徴とする再生方法。

【請求項 6】

デジタルストリームが記録されている記録媒体と前記記録媒体を再生する再生装置とを備えた記録媒体再生システムであって、

前記記録媒体は、

デジタルストリームが記録されており、

デジタルストリームは、グラフィクスストリームと動画ストリームとを多重したものであり、グラフィクスストリームは、複数パケットからなるパケット列であり、

グラフィクスストリームを構成するパケットには、グラフィクスデータを格納したデータパケットと、制御情報を格納した制御パケットとがあり、

データパケットのタイムスタンプの値は、グラフィクスデータのデコードタイミングを示し、

前記制御情報は、メモリ管理の始まりである旨を示すタイプ情報を有しており、

制御パケットは、プレゼンテーションタイムスタンプと、デコードタイムスタンプとを有しており、

プレゼンテーションタイムスタンプの値は、デコードされたグラフィクスデータを、動画ストリームに合成して表示させるタイミングを示し、

前記デコードタイムスタンプの値は、デジタルストリームの再生時間軸におけるメモリ管理の開始時点、及び、当該制御情報をメモリに読み出すタイミングを示し、

前記再生装置は、

前記記録媒体に記録されている動画ストリームをデコードして動画像を得るビデオデコーダと、

前記記録媒体に記録されているグラフィクスストリームを、動画像に合成して表示させるグラフィクスデコーダとを備え、

前記グラフィクスデコーダは、

動画ストリームの現在の再生時点が、グラフィクスデータを格納したパケットのデコードタイムスタンプに示される時点に達すれば、グラフィクスデータのデコードを開始し、同パケットのプレゼンテーションタイムスタンプの値に示される時点に達すれば、グラフィクスデータのデコードを終了するプロセッサと、

制御パケットのプレゼンテーションタイムスタンプに従い、デコードされたグラフィクスデータを、動画ストリームに合成して表示させるコントローラと、

圧縮状態のグラフィクスデータを格納するコーデッドデータバッファと、

プロセッサのデコードにより得られた非圧縮状態のグラフィクスデータを格納するオブジェクトバッファとを備え、

前記オブジェクトバッファに格納された非圧縮状態のグラフィクスデータのうち、表示に供されるべきものを格納するグラフィックスプレーンを備え、

前記コントローラは、メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、

メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、含まれている場合、動画ストリームの現在の再生時点が、制御パケットのデコードタイムスタンプの値に到達した段階で、コーデッドデータバッファ、オブジェクトバッファ、グラフィックスプレーンをクリアすることでメモリ管理を開始して、制御パケットから制御情報を読み出す、

ことを特徴とする記録媒体再生システム。

【発明の詳細な説明】

【技術分野】

【0001】

BD-ROM等の記録媒体、再生装置に関し、グラフィクス表示により字幕表示や対話表示を実現する技術に関する。

【背景技術】

10

20

30

40

50

## 【 0 0 0 2 】

グラフィクスによる字幕表示は、登場人物の台詞を世界中のあらゆる地域に居住する人々に伝えるという重要な使命をもつ。字幕表示を実現する先行技術には、ETSI EN 300 743標準規格(ETSI:European Telecommunication Standards Institute)における字幕付けアプリケーションがある。字幕付けアプリケーションとは、グラフィクスによる字幕表示を伴いながら、再生されるビデオストリームである。ここで字幕にあたるグラフィクスは、MPEG2規格のデータストリームとして実現される。このデータストリームは、PESパケットの配列であり、個々のPESパケットはPTS(Presentation Time Stamp)をもっている。ETSI EN 300 743標準規格は、このPTSを用いて、字幕付けアプリケーションにおける字幕表示のタイミングを規定している。これによりビデオストリームのあるピクチャが表示されるタイミングで、グラフィクスを表示するという動画 - グラフィクス間の同期が確立されることになる。これらの技術を記載した文献には、以下の非特許文献に掲げたものがある。

10

## 【 非特許文献 1 】

“ Digital Video Broadcasting(DVB) Subtitling systems;Final draft ETSI EN 300 743 ” ETSI STANDARDS,EUROPEAN TELECOMMUNICATION STANDARDS INSTITUTE,SOPHIA-ANTIPOLIS,FR

## 【 発明の開示 】

## 【 発明が解決しようとする課題 】

## 【 0 0 0 3 】

ところで、BD-ROMによる字幕付けアプリケーションの供給にあたっては、グラフィクスの解像度を更に高めたいとのニーズがある。具体的な数値を挙げれば、1920×1080というレベルにまで、解像度を向上したいと要請されている。しかしこのような高画質化を実現しようとすると、再生時のデコード負荷が膨大なものになる。

20

一方、ETSI EN 300 743標準規格に規定されているのは、PTSで示される時点にデコードを行い、即表示するという再生制御である。これをBD-ROMに応用しようとすると、再生装置においては、表示時点の直前において膨大なデコード負荷が集中することになる。かかる集中があると、再生装置のハードウェア・ソフトウェアの能力を高いレベルに設定することがグラフィクス表示の必須の条件になってしまう。かかる条件が必須になることは、再生装置の製造コストの高騰化を招き、再生装置の普及を阻害する。

## 【 0 0 0 4 】

本発明の目的は、製造コストの高騰化を避けつつも、グラフィクス表示の高画質化を図ることができる再生装置を提供することである。

30

## 【 課題を解決するための手段 】

## 【 0 0 0 5 】

上記目的を達成するため、本発明に係る再生装置は、動画ストリーム、グラフィクスストリームが多重化されたデジタルストリームについての再生装置であって、動画ストリームをデコードして動画像を得るビデオデコーダと、グラフィクスを、動画像に合成して表示させるグラフィクスデコーダとを備え、

グラフィクスストリームは、複数パケットからなるパケット列であり、パケットには、グラフィクスデータを格納したデータパケットと、制御情報を格納した制御パケットとがあり、前記グラフィクスデコーダは、動画ストリームの現在の再生時点が、グラフィクスデータを格納したパケットのデコードタイムスタンプに示される時点に達すれば、グラフィクスデータのデコードを開始し、同パケットのプレゼンテーションタイムスタンプの値に示される時点に達すれば、グラフィクスデータのデコードを終了するプロセッサと、制御パケットのプレゼンテーションタイムスタンプに従い、デコードされたグラフィクスデータを、動画ストリームに合成して表示させるコントローラと、圧縮状態のグラフィクスデータを格納するコーデッドデータバッファと、プロセッサのデコードにより得られた非圧縮状態のグラフィクスデータを格納するオブジェクトバッファとを備え、再生装置は、オブジェクトバッファに格納された非圧縮状態のグラフィクスデータのうち、表示に供されるべきものを格納するグラフィックスプレーンを備え、

40

前記コントローラは、メモリ管理の始まりである旨を示すタイプ情報が制御情報に含ま

50

れているか否かを判定し、メモリ管理の始まりである旨を示すタイプ情報が制御情報に含まれているか否かを判定し、含まれている場合、動画ストリームの現在の再生時点が、制御パケットのデコードタイムスタンプの値に到達した段階で、コーデッドデータバッファ、オブジェクトバッファ、グラフィックスプレーンをクリアすることでメモリ管理を開始して、制御パケットから制御情報を読み出すことを特徴としている。

【0006】

再生時間軸においてデコードが行われる期間が、データパケットのタイムスタンプに示される。グラフィックスの表示は、制御情報に付加されたタイムスタンプの値により規定されるので、本発明では、「デコードはされたが表示はされていない状態」、つまり、非圧縮のグラフィックスがバッファリングされている状態を、再生時間軸上に定義することができる。

10

【0007】

かかるバッファリング時期の定義により、膨大なデコード負荷の一点集中を回避することができる。またデコードを行うためのハードウェア資源の利用が、他の処理と競合したとしても、バッファリング時期を設けて、グラフィックスのデコード時期を変化させることができるので、かかる競合を回避することができる。

ここで、上述した目的達成にあたって、バッファリングの考えを導入すると、再生装置の開発に携わる技術者は、どの程度の規模で、メモリを実装すれば、正常動作を保証できるのかの判断に迷う。また字幕付けアプリケーションを製作する技術者も、自らが製作する字幕付けアプリケーションが、再生装置により確実に再生されるのかどうかという不安を抱く。何故なら、このバッファリングのためのメモリ占有量は、再生時間軸における再生進行に伴い、時間的な変遷を遂げるためである。メモリ占有量の時間的遷移が未知数のままになり、これを調整する術はないので、再生装置開発に携わる技術者や、字幕製作に携わる技術者の不安を払拭することはできない。

20

【0008】

この問題を解決するべく、上述した記録媒体では、前記制御パケットのデコードタイムスタンプの値は、デジタルストリームの再生時間軸におけるメモリ管理の開始時点、及び、当該制御情報をメモリに読み出すタイミングを示している。

制御パケットのデコードタイムスタンプは、メモリ管理の始まりを示すので、これを参照すれば、再生時間軸上のどの時点で、デコードモデルにおける各種バッファをフラッシュすればよいかを知得することができる。このフラッシュ時点をメモリ管理の起算点として考えれば、制御情報を格納するバッファや、デコード前のグラフィックスを格納するバッファ、デコード後のグラフィックスを格納するバッファの占有量の時間的変移が把握し易くなる。そしてこのデコードタイムスタンプの値を変化させることで、バッファ状態の時間的遷移を調整することができる。かかる調整により、再生装置側バッファのオーバーフローを回避することができるので、再生装置の開発にあたってのハードウェア、ソフトウェアの実装が簡易になる。

30

【0009】

また時間的変移の把握や調整が簡易になるので、オーサリングで得られたグラフィックスストリームが、BD-ROM規格が想定するデコードモデルの制約を満たすかどうかの検証も容易になる。オーサリング担当者は、自身が作成したグラフィックスの正常動作の確信を得た上で、オーサリング作業を進めることができる。

40

BD-ROMのデコードモデルを想定する場合、本発明の実施には更なる構成要件が必要となる。ここでBD-ROMのデコードモデルにおいてグラフィックスのデコード主体(プロセッサ)は、グラフィックスのアップデートの制御主体(コントローラ)と別々の構成要素になっている。デコード主体と、アップデート制御の主体とを別々に設けておくのは、グラフィックスが字幕である場合、グラフィックスを徐々に表示したり、消したりするような高度なアップデートを実行するためである。アップデートの制御主体がデコード主体とは別であると、プロセッサ-コントローラ間の密なる連携が重要になる。何故なら、プロセッサがグラフィックスデータのデコードを完了した後、コントローラが遅滞なくアップデートを実行せねば

50

ならないからである。

#### 【 0 0 1 0 】

プロセッサによるデコード完了をどのようにコントローラに通知するかは、装置におけるプロセッサ、コントローラの実装による。プロセッサ、コントローラがプログラムとして実装されている場合は、プロセス間通信でかかる通知はなされる。別々のハードウェア要素として実装されている場合は割込信号でかかる通知はなされる。かかる通知にタイムラグがどれだけ存在するかも、装置における実装による。そうすると、上述した通知のタイムラグが大きい実装であれば、グラフィックスのアップデートを動画の表示レートと同期させることができないケースが出現する。

#### 【 0 0 1 1 】

これを避けるには、前記制御パケットのプレゼンテーションタイムスタンプの値は、制御パケットのデコードタイムスタンプの値に所定の値を足し合わせた値であり、所定の値は、画面クリアに要する時間、及び、グラフィックスデータのデコードに要する時間のうち長い方と、画面へのグラフィックスデータへの書き込みに要する時間とに基づき設定しておくことが望ましい。

#### 【 0 0 1 2 】

デコード終了時刻は、データパケットのプレゼンテーションタイムスタンプに示されており、制御パケットのプレゼンテーションタイムスタンプは、デコード終了時刻に所定の期間を足し合わせた時刻を示しているので、コントローラはプレゼンテーションタイムスタンプを参照しさえすれば、たとえプロセッサからグラフィックスデータのデコード完了の通知を受けなくても、適切なタイミングで、アップデートを実行することができる。かかるアップデートの実行により、再生装置における実装がどのようなものであろうと、動画の表示レートと同期したアップデートを保障することができる。

#### 【 0 0 1 3 】

装置におけるプロセッサ - コントローラの実装を画一化することなく、プロセッサ - コントローラ間の密なる連携を実現することができるので、装置設計の自由度を守ることができ、装置の低コスト化を促進することができる。

#### 【 発明を実施するための最良の形態 】

#### 【 0 0 1 4 】

##### (第 1 実施形態)

以降、本発明に係る記録媒体の実施形態について説明する。先ず始めに、本発明に係る記録媒体の実施行為のうち、使用行為についての形態を説明する。図 1 は、本発明に係る記録媒体の、使用行為についての形態を示す図である。図 1 において、本発明に係る記録媒体は、BD-ROM 1 0 0 である。このBD-ROM 1 0 0 は、再生装置 2 0 0、テレビ 3 0 0、リモコン 4 0 0 により形成されるホームシアターシステムに、映画作品を供給するという用途に供される。

#### 【 0 0 1 5 】

以上が本発明に係る記録媒体の使用形態についての説明である。

続いて本発明に係る記録媒体の実施行為のうち、生産行為についての形態について説明する。本発明に係る記録媒体は、BD-ROMの応用層に対する改良により実施することができる。図 2 は、BD-ROMの構成を示す図である。

本図の第 4 段目にBD-ROMを示し、第 3 段目にBD-ROM上のトラックを示す。本図のトラックは、BD-ROMの内周から外周にかけて螺旋状に形成されているトラックを、横方向に引き伸ばして描画している。このトラックは、リードイン領域と、ボリウム領域と、リードアウト領域とからなる。本図のボリウム領域は、物理層、ファイルシステム層、応用層というレイヤモデルをもつ。ディレクトリ構造を用いてBD-ROMの応用層フォーマット(アプリケーションフォーマット)を表現すると、図中の第 1 段目のようになる。本図に示すようにBD-ROMには、ROOTディレクトリの下にBDMVディレクトリがあり、BDMVディレクトリの配下には、AVClipを格納したファイル(XXX.M2TS)、AVClipの管理情報を格納したファイル(XXX.CLPI)、AVClipにおける論理的な再生経路(PL)を定義したファイル(YYY.MPLS)が存

10

20

30

40

50

在する。本図に示すようなアプリケーションフォーマットを作成することにより、本発明に係る記録媒体は生産される。尚、XXX.M2TS、XXX.CLPI、YYY.MPLSといったファイルが、それぞれ複数存在する場合は、BDMVディレクトリの配下に、STREAMディレクトリ、CLIPINFディレクトリ、PLAYLISTディレクトリという3つのディレクトリを設け、STREAMディレクトリにXXX.M2TSと同じ種別のファイルを、CLIPINFディレクトリにXXX.CLPIと同じ種別のファイルを、PLAYLISTディレクトリにYYY.MPLSと同じ種別のファイルを格納することが望ましい。

#### 【0016】

このアプリケーションフォーマットにおけるAVClip(XXX.M2TS)について説明する。

AVClip(XXX.M2TS)は、MPEG-TS(Transport Stream)形式のデジタルストリームであり、ビデオストリーム、1つ以上のオーディオストリーム、プレゼンテーショングラフィクスストリームを多重化することで得られる。ビデオストリームは映画の動画部分を、オーディオストリームは映画の音声部分を、プレゼンテーショングラフィクスストリームは、映画の字幕をそれぞれ示している。図3は、AVClipがどのように構成されているかを模式的に示す図である。

10

#### 【0017】

AVClipは(中段)、複数のビデオフレーム(ピクチャp1,2,3)からなるビデオストリーム、複数のオーディオフレームからなるオーディオストリームを(上1段目)、PESパケット列に変換し(上2段目)、更にTSパケットに変換し(上3段目)、同じくプレゼンテーショングラフィクスストリーム(下1段目)を、PESパケット列に変換し(下2段目)、更にTSパ

20

#### 【0018】

本図において多重されるプレゼンテーショングラフィクスストリームは1つであるが、BD-ROMが多言語対応型である場合、その言語毎のプレゼンテーショングラフィクスストリームがAVClipに多重される。かかる過程を経て生成されたAVClipは、通常のコンピュータファイル同様、複数のエクステンツに分割され、BD-ROM上の領域に記録される。

続いてプレゼンテーショングラフィクスストリームについて説明する。図4(a)は、プレゼンテーショングラフィクスストリームの構成を示す図である。第1段目は、AVClipを構成するTSパケット列を示す。第2段目は、グラフィクスストリームを構成するPESパケット列を示す。第2段目におけるPESパケット列は、第1段目におけるTSパケットのうち、所定のPIDをもつTSパケットからペイロードを取り出して、連結することにより構成される。

30

#### 【0019】

第3段目は、グラフィクスストリームの構成を示す。グラフィクスストリームは、PCS(Presentation Composition Segment)、WDS(Window Defeine Segment)、PDS(Palette Definition Segment)、ODS(Object\_Definition\_Segment)、END(END of Display Set Segment)と呼ばれる機能セグメントからなる。これらの機能セグメントのうち、PCSは、画面構成セグメントと呼ばれ、WDS,PDS,ODS,ENDは定義セグメントと呼ばれる。PESパケットと機能セグメントとの対応関係は、1対1の関係、1対多の関係である。つまり機能セグメントは、1つのPESパケットに変換されてBD-ROMに記録されるか、又は、フラグメント化され、複数PESパケットに変換されてBD-ROMに記録される。

40

#### 【0020】

図4(b)は、機能セグメントを変換することで得られるPESパケットを示す図である。図4(b)に示すようにPESパケットは、パケットヘッダと、ペイロードとからなり、このペイロードが機能セグメント実体にあたる。またパケットヘッダには、この機能セグメントに対応するDTS、PTSが存在する。尚以降の説明では、機能セグメントが格納されるPESパケットのヘッダ内に存在するDTS及びPTSを、機能セグメントのDTS及びPTSとして扱う。

#### 【0021】

これら様々な種別の機能セグメントは、図5のような論理構造を構築する。図5は、様

50



々な種別の機能セグメントにて構成される論理構造を示す図である。本図は第3段目に機能セグメントを、第2段目にDisplay Setを、第1段目にEpochをそれぞれ示す。

第2段目のDisplay Set(DSと略す)とは、グラフィクスストリームを構成する複数機能セグメントのうち、一画面分のグラフィクスを構成するものの集合をいう。図中の破線は、第3段目の機能セグメントが、どのDSに帰属しているかという帰属関係を示す。PCS-WDS-PDS-ODS-ENDという一連の機能セグメントが、1つのDSを構成していることがわかる。再生装置は、このDSを構成する複数機能セグメントをBD-ROMから読み出せば、一画面分のグラフィクスを構成することができる。

#### 【0022】

第1段目のEpochとは、AVClipの再生時間軸上においてメモリ管理の連続性をもっている一つの期間、及び、この期間に割り当てられたデータ群をいう。ここで想定しているメモリとは、一画面分のグラフィクスを格納しておくためのグラフィクスプレーン、伸長された状態のグラフィクスデータを格納しておくためのオブジェクトバッファである。これらについてのメモリ管理に、連続性があるというのは、このEpochにあたる期間を通じてこれらグラフィクスプレーン及びオブジェクトバッファのフラッシュは発生せず、グラフィクスプレーン内のある決められた矩形領域内でのみ、グラフィクスの消去及び再描画が行われることをいう(ここでフラッシュとは、プレーン及びバッファの格納内容を全部クリアしてしまうことである)。この矩形領域の縦横の大きさ及び位置は、Epochにあたる期間において、終始固定されている。グラフィクスプレーンにおいて、この固定化された領域内で、グラフィクスの消去及び再描画を行っている限り、映像とグラフィクスとの同期が保障される。つまりEpochは、映像 - グラフィクスの同期の保障が可能な再生時間軸上の一単位ということができる。グラフィクスプレーンにおいて、グラフィクスの消去・再描画を行うべき領域を変更したい場合は、再生時間軸上においてその変更時点を定義し、その変更時点以降を、新たなEpochにせねばならない。この場合、2つのEpochの境界では、映像 - グラフィクスの同期は保証されない。

#### 【0023】

Epochにおける字幕の位置関係にたとえば、再生時間軸上において、画面上のある決まった矩形領域内に字幕が出現している期間が、Epochということができる。図6は、字幕の表示位置と、Epochとの関係を示す図である。本図では、動画の各ピクチャの絵柄に応じて字幕の位置を変更するという配慮がなされている。つまり5つの字幕「本当は」「ウソだった」「あなたが」「ずっと」「好きだった」のうち、3つの字幕「本当は」「ウソだった」「あなたが」は画面の下側に、「ずっと」「好きだった」は画面の上側に配置されている。これは画面の見易さを考え、画面中の余白にあたる位置に字幕を配置することを意図している。かかる時間的な変動がある場合、AVClipの再生時間軸において、下側の余白に字幕が出現している期間が1つのEpoch1、上側の余白に字幕が出現している期間が別のEpoch2になる。これら2つのEpochは、それぞれ独自の字幕の描画領域をもつことになる。Epoch1では、画面の下側の余白が字幕の描画領域(window1)になる。一方Epoch2では、画面の上側の余白が字幕の描画領域(window2)になる。これらのEpoch1,2において、バッファ・プレーンにおけるメモリ管理の連続性は保証されているので、上述した余白での字幕表示はシームレスに行われる。以上がEpochについての説明である。続いてDisplay Setについて説明する。

#### 【0024】

図5における破線hk1,2は、第2段目の機能セグメントが、どのEpochに帰属しているかという帰属関係を示す。Epoch Start, Acquisition Point, Normal Caseという一連のDSは、第1段目のEpochを構成していることがわかる。『Epoch Start』、『Acquisition Point』、『Normal Case』、『Epoch Continue』は、DSの類型である。本図におけるAcquisition Point, Normal Caseの順序は、一例にすぎず、どちらが先であってもよい。

#### 【0025】

『Epoch Start』は、「新表示」という表示効果をもたらすDSであり、新たなEpochの開始を示す。そのためEpoch Startは、次の画面合成に必要な全ての機能セグメントを含ん

10

20

30

40

50

でいる。Epoch Startは、映画作品におけるチャプター等、頭出しがなされることが判明している位置に配置される。

『Acquisition Point』は、“表示リフレッシュ”という表示効果をもたらすDisplay Setであり、先行するEpoch Startと全く同じDisplay Setをいう。Acquisition PointたるDSは、Epochの開始時点ではないが、次の画面合成に必要な全ての機能セグメントを含んでいるので、Acquisition PointたるDSから頭出しを行えば、グラフィックス表示を確実に実現することができる。つまりAcquisition PointたるDSは、Epochの途中からの画面構成を可能するという役割をもつ。

#### 【0026】

Acquisition PointたるDisplay Setは、頭出し先になり得る位置に組み込まれる。そのような位置には、タイムサーチにより指定され得る位置がある。タイムサーチとは、何分何秒という時間入力をユーザから受け付けて、その時間入力に相当する再生時点から頭出しを行う操作である。かかる時間入力は、10分単位、10秒単位というように、大まかな単位でなされるので、10分間隔の再生位置、10秒間隔の再生位置がタイムサーチにより指定され得る位置になる。このようにタイムサーチにより指定され得る位置にAcquisition Pointを設けておくことにより、タイムサーチ時のグラフィックスストリーム再生を好適に行うことができる。

#### 【0027】

『Normal Case』は、“表示アップデート”という表示効果をもたらすDSであり、前の画面合成からの差分のみを含む。例えば、あるDSvの字幕は、先行するDSuと同じ内容であるが、画面構成が、この先行するDSuとは異なる場合、PCSのみのDSv、又は、PCSのみのDSvを設けてこのDSvをNormal CaseのDSにする。こうすれば、重複するODSを設ける必要はなくなるので、BD-ROMにおける容量削減に寄与することができる。一方、Normal CaseのDSは、差分にすぎないので、Normal Case単独では画面構成は行えない。

#### 【0028】

『Epoch Continue』とは、AVClipの境界において、Epochが連続していることを示す。1つのDSnのComposition StateがEpoch Continueに設定されていれば、たとえDSnが、その直前に位置するDSn-1と異なるAVClip上に存在していたとしても、同じEpochに属することになる。これらDSn,DSn-1は、同じEpochに帰属するので、たとえこれら2つのDS間で、AVClipの分岐が発生したとしても、グラフィックスプレーン、オブジェクトバッファのフラッシュは発生しない。

続いてDefinition Segment(ODS,WDS,PDS)について説明する。

#### 【0029】

『Object\_Definition\_Segment』は、グラフィックスオブジェクトを定義する機能セグメントである。このグラフィックスオブジェクトについて以下説明する。BD-ROMに記録されているAVClipは、ハイビジョン並みの高画質をセールスポイントにしているため、グラフィックスオブジェクトの解像度も、1920×1080画素という高精細な大きさに設定されている。1920×1080という解像度があるので、BD-ROMでは、劇場上映用の字幕の字体、つまり、手書きの味わい深い字体の字幕表示を鮮やかに再現できる。画素の色にあたっては、一画素当たりのインデックス値(赤色差成分(Cr値),青色差成分(Cb値),輝度成分Y値,透明度(T値))のビット長が8ビットになっており、これによりフルカラーの16,777,216色から任意の256色を選んで画素の色として設定することができる。グラフィックスオブジェクトによる字幕は、透明色の背景に、文字列を配置することで描画される。

#### 【0030】

ODSによるグラフィックスオブジェクトの定義は、図7(a)に示すようなデータ構造をもってなされる。ODSは、図7(a)に示すように自身がODSであることを示す『segment\_type』と、ODSのデータ長を示す『segment\_length』と、EpochにおいてこのODSに対応するグラフィックスオブジェクトを一意に識別する『object\_id』と、EpochにおけるODSのバージョンを示す『object\_version\_number』と、『last\_in\_sequence\_flag』と、グラフィックスオブジェクトの一部又は全部である連続バイト長データ『object\_data\_fragment』と

10

20

30

40

50

からなる。

#### 【0031】

『object\_id』は、EpochにおいてこのODSに対応するグラフィクスオブジェクトを一意に識別する。グラフィクスストリームのEpochには、同じIDをもつODSが複数含まれる。同じIDをもつ複数のODSは縦横のサイズが同じであり、オブジェクトバッファの共通の領域が割り当てられる。同じIDをもつ複数ODSのうち1つが、この領域に読み出されれば、オブジェクトバッファにおけるODSは、同じIDをもつ後続のODSにより上書きされる。このように、ビデオストリームの再生進行に伴い、オブジェクトバッファ上のODSが、同じ識別子をもつ後続のODSにて上書きされることにより、ODSにおける絵柄は、時間的な変遷を遂げる。同じIDをもつグラフィクスオブジェクトの縦横サイズを同じにすると制限は、1つのEpoch内の制限に過ぎない。あるEpochに属するグラフィクスオブジェクトと、次のEpochに属するグラフィクスオブジェクトとでは縦横のサイズが変わっても良い。

10

#### 【0032】

『last\_in\_sequence\_flag』、『object\_data\_fragment』について説明する。PESパケットのペイロードの制限から、字幕を構成する非圧縮グラフィクスが1つのODSでは格納できない場合がある。そのような場合、グラフィクスを分割することにより得られた1部分(フラグメント)がobject\_data\_fragmentに設定される。1つのグラフィクスオブジェクトを複数ODSで格納する場合、最後のフラグメントを除く全てのフラグメントは同じサイズになる。つまり最後のフラグメントは、それ以前のフラグメントサイズ以下となる。これらフラグメントを格納したODSは、DSにおいて同じ順序で出現する。グラフィクスオブジェクトの最後は、last\_in\_sequence\_flagをもつODSにより指示される。上述したODSのデータ構造は、前のPESパケットからフラグメントを詰めてゆく格納法を前提にしているが、各PESパケットに空きが生じるように、詰めてゆくという格納法であっても良い。以上がODSの説明である。

20

#### 【0033】

『Palette Definition Segment(PDS)』は、色変換用のパレットを定義する情報である。PDSのデータ構造を図7(b)に示す。図7(b)に示すようにPDSは、自身がPDSであることを示す『segment\_type』、PDSのデータ長を示す『segment\_length』、このPDSに含まれるパレットを一意に識別する『pallet\_id』、EpochにおけるEpochのPDSのバージョンを示す『pallet\_version\_number』、各エントリーについての情報『pallet\_entry』からなる。『pallet\_entry』は、各エントリーにおける赤色差成分(Cr値)、青色差成分(Cb値)、輝度成分Y値、透明度(T値)を示す。

30

#### 【0034】

続いてWDSについて説明する。

『window\_definition\_segment』は、グラフィックスプレーンの矩形領域を定義するための機能セグメントである。Epochでは、クリア及び再描画が、グラフィックスプレーンにおけるある矩形領域内で行われている場合のみ、メモリ管理に連続性が生ずることは既に述べている。このグラフィックスプレーンにおける矩形領域は“window”と呼ばれ、このWDSで定義される。図8(a)は、WDSのデータ構造を示す図である。本図に示すようにWDSは、グラフィックスプレーンにおいてウィンドウを一意に識別する『window\_id』と、グラフィックスプレーンにおける左上画素の水平位置を示す『window\_horizontal\_position』と、グラフィックスプレーンにおける左上画素の垂直位置を示す『window\_vertical\_position』と、グラフィックスプレーンにおけるウィンドウの横幅を示す『window\_width』と、グラフィックスプレーンにおける縦幅を示す『window\_height』とを用いて表現される。

40

#### 【0035】

window\_horizontal\_position、window\_vertical\_position、window\_width、window\_heightがとりうる値について説明する。これらが想定している座標系は、グラフィックスプレーンの内部領域であり、このグラフィックスプレーンの内部領域においてウィンドウは、縦:video\_height、横:video\_widthという二次元状の大きさをもつ。

50

window\_horizontal\_positionは、グラフィックスプレーンにおける左上画素の水平アドレスであるので、0～video\_width-1の値をとり、window\_vertical\_positionは、グラフィックスプレーンにおける左上画素の垂直アドレスであるので0～video\_height-1の値をとる。

#### 【 0 0 3 6 】

window\_widthは、グラフィックスプレーンにおけるウィンドウの横幅であるので、0～video\_width-window\_horizontal\_position-1の値をとり、window\_heightは、グラフィックスプレーンにおける縦幅であるので0～video\_height-window\_vertical\_position-1の値をとる。

WDSのwindow\_horizontal\_position、window\_vertical\_position、window\_width、window\_heightにより、グラフィックスプレーンの何処にウィンドウを配置するか、ウィンドウの大きさをどれだけにするかをEpoch毎に規定することができる。そのため、あるEpochに属するピクチャが表示されている期間において、ピクチャ内の絵柄の邪魔にならないように、ピクチャ上の余白にあたる位置に、ウィンドウが現れるようオーサリング時に調整しておくことができる。これによりグラフィックスによる字幕表示を見易くすることができる。WDSはEpoch毎に定義可能なので、ピクチャの絵柄に時間的な変動があっても、その変動に応じて、グラフィックスを見易く表示することができる。そのため、結果として、字幕を映像本体に組み込むのと同じレベルにまで映画作品の品質を高めることができる。

#### 【 0 0 3 7 】

続いて『END of Display Set Segment』について説明する。END of Display Set Segmentは、Display Setの伝送の終わりを示す指標であり、Display Setにおける機能セグメントのうち、最後のODSの直後に配置される。このEND of Display Set Segmentの内部構成は、自身がEND of Display Set Segmentであることを示す『segment\_type』と、当該機能セグメントのデータ長を示す『segment\_length』とからなり、これといって説明が必要な構成要素はない。故に図示は省略する。

#### 【 0 0 3 8 】

以上がODS、PDS、WDS、ENDについての説明である。続いてPCSについて説明する。

PCSは、対話的な画面を構成する機能セグメントである。PCSは、図8(b)に示すデータ構造で構成される。本図に示すようにPCSは、『segment\_type』と、『segment\_length』と、『composition\_number』と、『composition\_state』と、『pallet\_update\_flag』と、『pallet\_id』と、『composition\_object(1)～(m)』とから構成される。

#### 【 0 0 3 9 】

『composition\_number』は、0から15までの数値を用いてDisplay Setにおけるグラフィックスアップデートを識別する。どのように識別するかというと、Epochの先頭から本PCSまでにグラフィックスアップデートが存在すれば、それらグラフィックスアップデートを経由する度に、インクリメントされるというルールでcomposition\_numberは設定される。

『composition\_state』は、本PCSから始まるDisplay Setが、Normal Caseであるか、Acquisition Pointであるか、Epoch Startであるかを示す。

#### 【 0 0 4 0 】

『pallet\_update\_flag』は、本PCSにおいてPalletOnly Display Updateがなされているかどうかを示す。PalletOnly Display Updateとは、直前のパレットのみを新たなものに切り換えることでなされるアップデートをいう。本PCSでかかるアップデートがなされれば、本フィールドは"1"に設定される。

『pallet\_id』は、PalletOnly Display Updateに用いられるべきパレットを示す。

#### 【 0 0 4 1 】

『composition\_object(1)・・・(n)』は、このPCSが属するDisplay Set内の、個々のウィンドウをどのように制御するかを示す情報である。図8(b)の破線wd1は、任意のcomposition\_object(i)の内部構成をクローズアップしている。この破線wd1に示すように、composition\_object(i)は、『object\_id』、『window\_id』、『object\_cropped\_flag』、『object\_horizontal\_position』、『object\_vertical\_position』、『cropping\_rectang

le情報(1)(2)・・・(n)』からなる。

【0042】

『object\_id』は、composition\_object(i)に対応するウィンドウ内に存在するODSの識別子を示す。

『window\_id』は、本PCSにおいて、グラフィクスオブジェクトに割り当てられるべきウィンドウを示す。1つのウィンドウには最大2つのグラフィクスオブジェクトが割り当てられる。

【0043】

『object\_cropped\_flag』は、オブジェクトバッファにおいてクロップされたグラフィクスオブジェクトを表示するか、グラフィクスオブジェクトを非表示とするかを切り換えるフラグである。“1”と設定された場合、オブジェクトバッファにおいてクロップされたグラフィクスオブジェクトが表示され、“0”と設定された場合、グラフィクスオブジェクトは非表示となる。

【0044】

『object\_horizontal\_position』は、グラフィックスプレーンにおけるグラフィクスオブジェクトの左上画素の水平位置を示す。

『object\_vertical\_position』は、グラフィックスプレーンにおける左上画素の垂直位置を示す。

『cropping\_rectangle情報(1)(2)・・・(n)』は、『object\_cropped\_flag』が1に設定されている場合に有効となる情報要素である。破線wd2は、任意のcropping\_rectangle情報(i)の内部構成をクローズアップしている。この破線に示すようにcropping\_rectangle情報(i)は、『object\_cropping\_horizontal\_position』、『object\_cropping\_vertical\_address』、『object\_cropping\_width』、『object\_cropping\_height』からなる。

【0045】

『object\_cropping\_horizontal\_position』は、グラフィックスプレーンにおけるクロップ矩形の左上画素の水平位置を示す。クロップ矩形は、グラフィクスオブジェクトの一部を切り出すための枠であり、ETSI EN 300 743標準規格における“Region”に相当する。

『object\_cropping\_vertical\_address』は、グラフィックスプレーンにおけるクロップ矩形の左上画素の垂直位置を示す。

【0046】

『object\_cropping\_width』は、グラフィックスプレーンにおけるクロップ矩形の横幅を示す。

『object\_cropping\_height』は、グラフィックスプレーンにおけるクロップ矩形の縦幅を示す。

以上がPCSのデータ構造である。続いてPCSの具体的な記述について説明する。この具体例は、図6に示した字幕の表示、つまり動画の再生進行に伴い、三回のグラフィックスプレーンへの書き込みで『ほんとは』『ウソだった』『あなたが』というように徐々に表示させるというものである。図9は、かかる字幕表示を実現するための記述例である。本図におけるEpochは、DS1(EPOCH Start)、DS2(Normal Case)、DS3(Normal Case)を有する。DS1は、字幕の表示枠となるwindowを定義するWDS、台詞『ほんとは ウソだった あなたが』を表すODS、1つ目のPCSを備える。DS2(Normal Case)は、2つ目のPCSを有する。DS3(Normal Case)は3つ目のPCSを有する。

【0047】

次に個々のPCSをどのように記述するかについて説明する。Display Setに属するWDS、PCSの記述例を図10～図12に示す。図10は、DS1におけるPCSの記述例を示す図である。

図10において、WDSのwindow\_horizontal\_position、window\_vertical\_positionは、グラフィックスプレーンにおけるウィンドウの左上座標LP1を、window\_width、window\_heightは、ウィンドウの表示枠の横幅、縦幅を示す。

10

20

30

40

50

## 【 0 0 4 8 】

図 1 0 におけるクロップ情報のobject\_cropping\_horizontal\_position,object\_cropping\_vertical\_positionは、オブジェクトバッファにおけるグラフィクスオブジェクトの左上座標を原点とした座標系においてクロップ範囲の基準ST1を示している。そして基準点からobject\_cropping\_width、object\_cropping\_heightに示される範囲(図中の太枠部分)がクロップ範囲になる。クロップされたグラフィクスオブジェクトは、グラフィックスプレーンの座標系においてobject\_horizontal\_position,object\_vertical\_positionを基準点(左上)とした破線の範囲cp1に配置される。こうすることにより、『本当は』がグラフィックスプレーンにおけるウィンドウ内に書き込まれる。これにより字幕『本当は』は動画像と合成され表示される。

10

## 【 0 0 4 9 】

図 1 1 は、DS2におけるPCSの記述例を示す図である。本図におけるWDSの記述は、図 1 0 と同じなので説明を省略する。クロップ情報の記述は、図 1 0 と異なる。図 1 1 におけるクロップ情報のobject\_cropping\_horizontal\_position,object\_cropping\_vertical\_position,は、オブジェクトバッファ上の字幕『本当はウソだった。あなたが』のうち、『ウソだった』の左上座標を示し、object\_cropping\_height,object\_cropping\_widthは、『ウソだった』の横幅、縦幅を示す。こうすることにより、『ウソだった』がグラフィックスプレーンにおけるウィンドウ内に書き込まれる。これにより字幕『ウソだった』は動画像と合成され表示される。

## 【 0 0 5 0 】

20

図 1 2 は、DS3におけるPCSの記述例を示す図である。本図におけるWDSの記述は、図 1 0 と同じなので説明を省略する。クロップ情報の記述は、図 1 0 と異なる。図 1 2 におけるクロップ情報のobject\_cropping\_horizontal\_position,object\_cropping\_vertical\_position,は、オブジェクトバッファ上の字幕『本当はウソだった。あなたが』のうち、『あなたが』の左上座標を示し、object\_cropping\_height,object\_cropping\_widthは、『あなたが』の横幅、縦幅を示す。こうすることにより、『あなたが』がグラフィックスプレーンにおけるウィンドウ内に書き込まれる。これにより字幕『あなたが』は動画像と合成され表示される。

## 【 0 0 5 1 】

DS1,2,3のPCSを以上のように記述することで、字幕の表示効果を実現することができる。その他の表示効果を実現するにあたっての、PCSの記述作法について説明する。

30

先ず初めにCut-In/Outの記述作法について説明する。図 1 3 は、時間軸に沿った連続写真的な表記で、Cut-In/Outを実行する際のDisplay Setの記述例を示す図である。

本図におけるwindow(x,y,u,v)におけるx,yは、window\_vertical\_position、window\_horizontal\_positionの値であり、u,vは、window\_width、window\_heightの値を示す。また本図のCropping Rectangle(a,b,c,d)におけるa,bは、object\_cropping\_vertical\_position、object\_cropping\_horizontal\_positionの値であり、c,dはobject\_cropping\_width、object\_cropping\_heightの値である。本図の再生時間軸の時点t11,t12,t13には、DS11,DS12,DS13が存在している。

## 【 0 0 5 2 】

40

時点t11に存在するDS11は、Composition\_StateがEpoch Startに設定され、Object\_Cropped\_Flagが0(No Cropping Rectangle Visible)に設定されたPCS#0、グラフィックスプレーンの(100,100)の位置に、横700×縦500のwindowを宣言するWDS#0、PDS#0、字幕「Credit:」を表すODS#0、ENDを有する。

時点t12に存在しているDS12は、Composition\_StateがNormal Caseであり、オブジェクトバッファの(0,0)から横600×縦400の範囲でクロップを行い(Cropping Rectangle#0(0,0,600,400))、クロップされたグラフィクスオブジェクトをグラフィックスプレーンの座標(0,0)に配置する(on window(0,0))という手順を示すPCS#1を含む。

## 【 0 0 5 3 】

時点t13に存在しているDS13は、Composition\_StateがNormal Caseであり、クロップさ

50

れたグラフィクスオブジェクトを消すよう、object\_cropped\_flagが0に設定されたPCS#2を含む(No Cropping Rectangle Visible)。

以上のDisplay Setにより、t11では字幕「Credit:」は非表示になっているが、t12で表示され、t13で再び非表示になる。これによりCut-In/Cut-Outという表示効果が実現される。

#### 【 0 0 5 4 】

続いてFade-In/Outという表示効果を実現する際の、Display Setの記述作法について説明する。図 1 4 は、時間軸に沿った連続写真的な表記で、Fade-In/Outを実行する際のDisplay Setの記述例を示す図である。本図の時点t21,t22,t23,t24には、DS21,DS22,DS23,DS24が存在している。

10

時点t21に存在するDS21は、Composition\_StateがEpoch Startに設定され、オブジェクトバッファの(0,0)から横600×縦400の範囲でクロップを行い(Cropping Rectangle#0(0,0,600,400))、クロップされたグラフィクスオブジェクトをグラフィックスプレーンの座標(0,0)に配置する(on window(0,0))という手順を示すPCS#0、グラフィックスプレーンの(100,100)の位置に、横700×縦500のwindowを宣言するWDS#0、PDS#0、字幕「Fin」を表すODS#0、ENDを有する。

#### 【 0 0 5 5 】

時点t22に存在しているDS22は、Composition\_StateがNormal CaseであるPCS#1と、PDS#1とを含む。このPDS#1は、PDS#0と同じ色彩の赤色差、青色差を示しておりながら、高い輝度値をもっているものとする。

20

時点t23に存在しているDS23は、Composition\_StateがNormal CaseであるPCS#3と、PDS#2と、ENDとを含む。このPDS#2は、PDS#0と同じ色彩の赤色差、青色差を示しておりながら、低い輝度値をもっているものとする。

#### 【 0 0 5 6 】

時点t24に存在するDS24は、Composition\_StateがNormal Caseに設定され、Object\_Cropped\_Flagが0(No Cropping Rectangle Visible)に設定されたPCS、ENDを有する。

これら一連のDisplay Setは、1つ前のDisplay Setと異なるPDSを指定しているため、Epoch内の複数PCSにおいて表示されるグラフィクスオブジェクトは、徐々に輝度値が高められたり、弱められることになる。これによりFade-In/Outという表示効果の実現が可能になる。

30

#### 【 0 0 5 7 】

続いてScrollingの記述作法について説明する。図 1 5 は、時間軸に沿った連続写真的な表記で、Scrollingがどのように行われるかを記述した図である。本図における表記は、図 1 3 の一例に準ずる。

AVClip再生時間軸の時点t31に存在するDS31は、Composition\_StateがEpoch Startに設定され、Object\_Cropped\_Flagが0(No Cropping Rectangle Visible)に設定されたPCS#0、グラフィックスプレーンの(100,100)の位置に、横700×縦500のwindowを宣言するWDS#0、PDS#0、字幕「Credit: Company」を表すODS#0、ENDを有する。

#### 【 0 0 5 8 】

時点t32に存在するDS32は、Composition\_StateがNormal Caseであり、オブジェクトバッファにおける(0,0)から横600×縦400の範囲でクロップを行い(Cropping Rectangle#0(0,0,600,400))、クロップされたグラフィクスオブジェクトをグラフィックスプレーンの座標(0,0)に配置する(on window#0(0,0))という手順を示すPCS#1を含む。オブジェクトバッファにおいて(0,0)から横600×縦400の範囲には、「Credit:」「Company」という2段表記の字幕のうち「Credit:」が存在するので、この「Credit:」がグラフィックスプレーンに出現することになる。

40

#### 【 0 0 5 9 】

時点t33に存在するDS33は、Composition\_StateがNormal Caseであり、オブジェクトバッファ上の座標(0,100)から横600×縦400の範囲を切り出し(Cropping Rectangle#0(0,100,600,400))、ウィンドウにおける座標(0,0)に配置する(on window#0(0,0))という手順を

50

示すPCS#2を含む。オブジェクトバッファ上の座標(0,100)から横600×縦400の範囲には、「Credit:」「Company」という2段表記の字幕のうち、「Credit:」の下半分と、「Company」の上半分とが存在するので、これらの部分がグラフィックスプレーンに出現することになる。

【0060】

時点t34に存在するDS34は、Composition\_StateがNormal Caseであり、オブジェクトバッファの座標(0,200)から横600×縦400の範囲を切り出し(Cropping Rectangle#0(0,200,600,400))、ウィンドウにおける座標(0,0)に配置する(on window#0(0,0))という手順を示すPCS#3を含む。オブジェクトバッファ上の座標(0,200)から横600×縦400の範囲には、「Credit:」「Company」という2段表記の字幕のうち、「Company」が存在するので、2段表記の字幕のうち「Company」の部分が画面上に現れることになる。以上のPCSの記述により、「Credit:」「Company」という2段表記の字幕の下スクロールがなされることになる。

10

【0061】

続いてWipe-In/Outの記述作法について説明する。図16は、時間軸に沿った連続写真的な表記で、Wipe-In/Outがどのように行われるかを記述した図である。本図における表記は、図13の一例に準ずる。

本図の時点t51に存在するDS51は、Composition\_StateがEpoch Startに設定され、Object\_Cropped\_Flagが0(No Cropping Rectangle Visible)に設定されたPCS#0、グラフィックスプレーンの(100,100)の位置に、横700×縦500のwindowを宣言するWDS#0、PDS#0、字幕「Fin」を表すODS#0、ENDを有する。

20

【0062】

時点t52に存在するDS52は、Composition\_StateがNormal Caseであり、オブジェクトバッファの(0,0)から横600×縦400の範囲でクロップを行い(Cropping Rectangle#0(0,0,600,400))、クロップされたグラフィクスオブジェクトをグラフィックスプレーンの座標(0,0)に配置する(on window#0(0,0))という手順を示すPCS#1を含む。(0,0)から横600、縦400の範囲には、「Fin」という字幕が存在するので、この「Fin」がグラフィックスプレーンに出現することになる。

【0063】

時点t53に存在するDS53は、Composition\_StateがNormal Caseであり、オブジェクトバッファにおける(200,0)から縦400×横400の範囲でクロップを行い(Cropping Rectangle#0(200,0,400,400))、座標(200,0)を基準にして、表示を行わせるPCS#2を含む(on window#0(200,0))。そうすると、(200,0)～(400,400)が表示範囲内になり、(0,0)～(199,400)は、非表示範囲になる。

30

【0064】

時点t54に存在するDS54は、Composition\_StateがNormal Caseであり、オブジェクトバッファにおけるグラフィクスオブジェクトの(400,0)から横200×縦400の範囲でクロップを行い(Cropping Rectangle#0(400,0,200,400))、座標(400,0)を基準にして、ウィンドウの表示を行わせる(on window#0(400,0))PCS#3を含む。そうすると、(0,0)～(399,400)は、非表示範囲になる。

40

【0065】

グラフィックスプレーンの非表示範囲が広がるに伴い、表示範囲が狭まるのでWipe-In/Outが実現されることになる。

このようにPCSの記述次第でCut-In/Out,Fade-In/Out,Wipe-In/Out,Scrollというような多様な表示効果を実現することができるので、字幕描画に様々な工夫を凝らすことができる。

【0066】

かかる表示効果実現にあたっての制約について説明する。図13～図16の表示効果のうち、Scrollingの実現には、ウィンドウクリアと、ウィンドウ再描画とが必要になる。図15のt32 - t33間で考えれば、Scrollingの実現のためには、t32における「Credit:」

50



なるグラフィクスオブジェクトをグラフィックスプレーンから消し去るというウィンドウクリアと、「Credit:」の下半分、「Company」の上半分をグラフィックスプレーンに書き込むというウィンドウ再描画とを  $t_{32} - t_{33}$  の時間間隔で実現せねばならない。この時間間隔がビデオフレームの時間間隔である場合、オブジェクトバッファと、グラフィックスプレーンとの間の転送レートはどれだけ必要であろうか。

#### 【0067】

ここでウィンドウをどれだけの大きさとするかの制限について検討する。オブジェクトバッファ - グラフィックスプレーン間の転送レートを  $R_c$  とすると、ワーストケースでは、この転送レート  $R_c$  でウィンドウクリアと、ウィンドウ再描画とを行わねばならない。そうするとウィンドウクリア、ウィンドウ再描画のそれぞれを  $R_c$  の半分の転送レート ( $R_c/2$ ) で

10

#### 【0068】

これらウィンドウクリア - ウィンドウ再描画をビデオフレームと同期させるには、  
ウィンドウサイズ  $\times$  フレームレート  $R_c/2$   
を満たす必要がある。このフレームレートが 29.97 であるなら、  
 $R_c$  は、ウィンドウサイズ  $\times 2 \times 29.97$  になる。

字幕の表示にあたっては、グラフィックスプレーン全体に対し、最低でも 25% ~ 33% 程度の大きさが必要となる。ここでグラフィックスプレーンの総画素数は  $1920 \times 1080$  であり、一画素当たりのインデックスのビット長を 8 ビットとすると、グラフィックスプレーンの総容量は 2M バイト ( $1920 \times 1080 \times 8$ ) になる。

20

#### 【0069】

この総容量の 1/4 をウィンドウサイズとすると、500k バイト ( $= 2\text{M バイト}/4$ ) になる。これを上述した式に代入して  $R_c$  を導けば、 $R_c$  は 256Mbps ( $500\text{K バイト} \times 2 \times 29.97$ ) と算出することができる。

この 25% ~ 33% という大きさであれば、256Mbps という転送レートで字幕の表示を行っている限り、如何なる表示効果を実現する場合であっても、動画との同期を維持することができる。

#### 【0070】

仮に、ウィンドウクリア及び再描画のレートがビデオフレームのフレームレートの 1/2, 1/4 でよいなら、 $R_c$  がたとえ同じであってもウィンドウサイズを 2 倍, 4 倍にすることができる。以上が window の大きさについての説明である。続いて、window の位置、範囲について説明する。上述したように、Epoch においてウィンドウの位置、範囲は一貫している。

30

Epoch においてウィンドウの位置、範囲を一貫させておくのは以下の理由による。ウィンドウの位置・範囲を変えれば、グラフィックスプレーンに対する書込先アドレスを変えねばならず、オーバーヘッドが発生するので、かかるオーバーヘッドによりオブジェクトバッファからグラフィックスプレーンへの転送レートが低下するからである。

#### 【0071】

ウィンドウには、グラフィクスオブジェクトの個数が制限されている。この個数制限は、デコードされたグラフィクスオブジェクトの転送にあたってのオーバーヘッドを低減する目的で設けられている。ここでのオーバーヘッドは、グラフィクスオブジェクトのエッジ部分のアドレスを設定する際に発生する。そうすれば、エッジの部分が多く存在する程、オーバーヘッドの発生回数が多くなる。

40

#### 【0072】

図 17 は、ウィンドウ内にグラフィクスオブジェクトが 4 つ存在する場合と、2 つの存在する場合とを対比して示す図である。グラフィクスオブジェクトが 4 つ存在する場合、エッジの数は 8 つになるのでグラフィクスオブジェクトが 2 つ存在する場合と比べると、エッジの数は 2 倍になっている。

ウィンドウにおけるグラフィクスオブジェクトの数に制限がないと、グラフィクス転送にあたって発生するオーバーヘッド数が未知数になり、転送負荷の増減が激しくなる。一方、ウィンドウにおけるグラフィクスの個数が 2 つまでであると、最悪 4 つのオーバーヘッ

50

ドが発生すると見込んで転送レートを設定すればよいので、ミニマムスタンダードたる転送レートを数値化し易くなる。

#### 【 0 0 7 3 】

以上がウィンドウについての説明である。続いてこれらPCS、ODSを有したDisplay Setが、AVClipの再生時間軸上にどのように割り当てられるかについて説明する。Epochは、再生時間軸上においてメモリ管理が連続する期間であり、Epochは1つ以上のDisplay Setから構成されるので、Display SetをどうやってAVClipの再生時間軸に割り当ててくるかが問題になる。ここでAVClipの再生時間軸とは、AVClipに多重されたビデオストリームを構成する個々のピクチャデータのデコードタイミング、再生タイミングを規定するための想定される時間軸をいう。この再生時間軸においてデコードタイミング、再生タイミングは、90KHzの時間精度で表現される。Display Set内のPCS、ODSに付加されたDTS、PTSは、この再生時間軸において同期制御を実現すべきタイミングを示す。このPCS、ODSに付加されたDTS、PTSを用いて同期制御を行うことが、再生時間軸へのDisplay Setの割り当てである。

10

#### 【 0 0 7 4 】

まず、ODSに付加されたDTS、PTSにより、どのような同期制御がなされるかについて説明する。

DTSは、ODSのデコードを開始すべき時間を90KHzの時間精度で示しており、PTSはデコード終了時刻を示す。

ODSのデコードは、瞬時には完了せず、時間的な長さをもっている。このデコード期間の開始点・終了点を明らかにしたいとの要望から、ODSについてのDTS、PTSはデコード開始時刻、デコード終了時刻を示している。

20

#### 【 0 0 7 5 】

PTSの値はデッドラインであるので、PTSに示される時刻までにODSのデコードがなされて、非圧縮状態のグラフィックオブジェクトが、再生装置上のオブジェクトバッファに得られなければならない。

DSnに属する任意のODSjのデコード開始時刻は、90KHzの時間精度でDTS(DSn[ODSj])に示されるので、これにデコードを要する最長時間を加えた時刻が、Display SetのODSjのデコード終了時刻になる。

#### 【 0 0 7 6 】

ODSjのサイズを"SIZE(DSn[ODSj])"、ODSのデコードレートを"Rd"とすると、デコードに要する最長時間(秒)は、"SIZE(DSn[ODSj])/Rd"になる。ここで"//"は、小数点以下切り上げの割り算を示す演算子である。

30

この最長時間を90KHzの時間精度に変換し、ODSjのDTSに加算することにより、PTSで示されるべきデコード終了時刻(90KHz)は算出される。

DSnに属するODSjのPTSを、数式で表すと、以下の式ようになる。

$$PTS(DSn[ODSj]) = DTS(DSn[ODSj]) + 90,000 \times (SIZE(DSn[ODSj]) / Rd)$$

そして互いに隣接する2つのODS(ODSj, ODSj+1)の間では、以下の関係を満たす必要がある。

$$PTS(DSn[ODSj]) \leq DTS(DSn[ODSj+1])$$

40

続いてPCSのDTS、PTSの設定について説明する。

#### 【 0 0 7 7 】

PCSは、DSnにおける最初のODS(ODS1)のデコード開始時点(DTS(DSn[ODS1]))以前、及び、DSnにおける最初のPDS(PDS1)が有効になる時点(PTS(DSn[PDS1]))以前に、再生装置上のバッファにロードされねばならない。よって以下の式の関係を満たす値に、設定されねばならない。

$$DTS(DSn[PCS]) \leq DTS(DSn[ODS1])$$

$$DTS(DSn[PCS]) \leq PTS(DSn[PDS1])$$

そしてDSnにおけるPCSのPTSは、以下の数式から算出される。

$$PTS(DSn[PCS]) = DTS(DSn[PCS]) + \text{decodeduration}(DSn)$$

50

ここで`decodeduration(DSn)`は、PCSのアップデートに用いられる全グラフィクスオブジェクトのデコード時間である。このデコード時間は、固定値ではない。しかし各再生装置の状態や再生装置の実装により変動するものでもない。本`DSn.PCSn`の画面合成に用いられるオブジェクトを`DSn.PCSn.OBJ[j]`とした場合、`decodeduration(DSn)`は、ウィンドウクリアに要する時間(i)、`DSn.PCSn.OBJ`のデコード期間(ii)、`DSn.PCSn.OBJ`の書き込みに要する時間(iii)により変動を受ける値になる。`Rd,Rc`さえ予め定められていれば、どのような実装の再生装置であっても、同じ値になる。そのためオーサリング時にあたっては、これらの期間の長さを算出して、この値からPTSを計算している。

#### 【 0 0 7 8 】

`decode_duration`の計算は、図 1 8 のプログラムに基づき行われる。また図 1 9、図 2 0 ( a ) ( b ) は、このプログラムのアルゴリズムを図式化したフローチャートである。以降、これらの図を参照しながら`decode_duration`の算出について説明する。図 1 9 のフローチャートにおいて、先ず初めに、`PLANEINITIALIZATINTIME`関数を呼び出し、戻り値を`decode_duration`に加算する(図 1 9 のステップ S 1)。`PLANEINITIALIZATINTIME`関数(図 2 0 ( a ))は、Display Setを表示するにあたってグラフィックスプレーンの初期化に要する時間を求める関数を呼出すための関数であり、図 1 9 のステップ S 1 では、`DSn,DSnPCS.OBJ[0],decode_durtation`を引数に設定して、この関数を呼び出す。

#### 【 0 0 7 9 】

続いて図 2 0 ( a ) を参照しながら`PLANEINITIALIZATINTIME`関数について説明する。図 2 0 ( a ) において`initialize_duration`とは、`PLANEINITIALIZATINTIME`関数の戻り値を示す変数である。

図 2 0 のステップ S 2 は、`DSn`のPCSにおける`composition_state`がEpoch Startかどうかにより、処理を切り換えるif文である。もし`composition_state`がEpoch Startであるなら(図 1 8 の`DSn.PCS.composition_state==EPOCH_START`、ステップ S 2 = Yes)、グラフィックスプレーンのクリアに要する時間を`initialize_duration`に設定する(ステップ S 3)。

#### 【 0 0 8 0 】

オブジェクトバッファ - グラフィックスプレーン間の転送レート`Rc`を上述したように256,000,000とし、グラフィックスプレーンの総サイズを`video_width*video_height`とすると、クリアに要する時間(秒)は、「`video_width*video_height//256,000,000`」になる。これに、90.000Hzを乗じPTSの時間精度で表現すれば、グラフィックスプレーンのクリアに要する時間は「`90000 × video_width*video_height//256,000,000`」になる。この時間を、`initialize_duration`に加えて、`initialize_duration`を戻り値としてリターンする。

#### 【 0 0 8 1 】

もし`composition_state`がEpoch Startでないなら(ステップ S 2 = No)、WDSにて定義される`window[i]`のクリアに要する時間を`initialize_duration`に加えるという処理を全ての`window`について繰り返す(ステップ S 4)。オブジェクトバッファ - グラフィックスプレーン間の転送レート`Rc`を上述したように256,000,000とし、WDSに属するウィンドウ`[i]`の総サイズを `SIZE(WDS.WIN[i])`とすると、クリアに要する時間(秒)は、「`SIZE(WDS.WIN[i])//256,000,000`」になる。これに、90.000Hzを乗じPTSの時間精度で表現すれば、WDSに属する全ウィンドウのクリアに要する時間は「`90000 × SIZE(WDS.WIN[i])//256,000,000`」になる。この時間を、`initialize_duration`に加えて、`initialize_duration`を戻り値としてリターンする。以上が`PLANEINITIALIZATINTIME`関数である。

#### 【 0 0 8 2 】

図 1 9 のステップ S 5 は、`DSn`に属するグラフィクスオブジェクトの数が2であるか、1であるかで処理を切り換えるものであり(図 1 8 の`if(DSn.PCS.num_of_object==2),if(DSn.PCS.num_of_object==1)`)、もし " 1 " であるなら(ステップ S 5 = 1)、グラフィクスオブジェクトのデコードを行うための待ち時間を`decode_duration`に加える(ステップ S 6)。この待ち時間の算出は、`WAIT`関数の呼出(図 1 8 の`decode_duration +=WAIT(DSn,DSn.PCS.OBJ[0],decode_duration)`)で実現される。この関数呼出にあたっての引数は`DSn,DSn.PCS.OBJ[0],decode_duration`であり、待ち時間を示す`wait_duration`が戻り値として返される

10

20

30

40

50

。

## 【 0 0 8 3 】

図 2 0 ( b ) は、WAIT関数の処理手順を示すフローチャートである。

このWAIT関数においてcurrent\_durationとは、呼出元のdecode\_durationが設定される。object\_define\_ready\_timeは、Display Setのグラフィクスオブジェクト[i]のPTSが設定される変数である。

current\_timeとは、DSnのPCSのDTSに、current\_durationを足した値が設定される変数である。このcurrent\_timeよりobject\_define\_ready\_timeが大きい場合(ステップS 7 がYes、if(current\_time < object\_define\_ready\_time))、戻り値たるwait\_durationは、object\_define\_ready\_timeとcurrent\_timeとの差分が設定されることになる(ステップS 8、wait\_duration += object\_define\_ready\_time - current\_time)。以上がWait関数である。ステップS 6 においてdecode\_durationには、このwait関数の戻り値と、window再描画に必要な時間を足し合わせた時間(90,000\*(SIZE(DSn.WDS.WIN[0]))//256,000,000)が設定されることになる。

10

## 【 0 0 8 4 】

以上はDSnのグラフィクスオブジェクトが1つである場合の処理である。図 1 9 のステップS 5 は、グラフィクスオブジェクトの数が2であるかどうかを判定している。DSnのグラフィクスオブジェクトが2以上であれば(図 1 8 のif(DSn.PCS.num\_of\_object==2))、PCSにおけるOBJ[0]を引数として、wait関数を呼び出し、その戻り値をdecode\_durationに加算する(ステップS 1 0)。

20

## 【 0 0 8 5 】

続くステップS 1 1 は、DSnのOBJ[0]が属するwindowが、グラフィクスオブジェクト[1]が属するwindowと同一かどうかの判定であり(if(DSn.PCS.OBJ[0].window\_id == DSn.PCS.OBJ[1].window\_id)、もし同一であるなら、OBJ[1]を引数にしてwait関数を呼び出し、その戻り値wait\_durationを、decode\_durationに加算する(ステップS 1 2)。OBJ[0]が属するwindowの再描画に必要な時間(90,000\*(SIZE(DSn.WDS.OBJ[0].window\_id)//256,000,000)をdecode\_durationに加える(ステップS 1 3)。

## 【 0 0 8 6 】

もし属するwindowが異なるなら(ステップS 1 1 で " 異なる " )、OBJ[0]が属するwindowの再描画に必要な時間(90,000\*(SIZE(DSn.WDS.OBJ[0].window\_id)//256,000,000)をdecode\_durationに加える(ステップS 1 5)。OBJ[1]を引数にしてwait関数を呼び出し、その戻り値wait\_durationを、decode\_durationに加算する(ステップS 1 6)。その後、OBJ[1]が属するwindowの再描画に必要な時間(90,000\*(SIZE(DSn.WDS.OBJ[1].window\_id)//256,000,000)をdecode\_durationに加える(ステップS 1 7)。

30

## 【 0 0 8 7 】

以上のアルゴリズムにより、Decode\_Durationは算出される。このPCSのPTSが、具体的にどのように設定されるかについて説明する。

図 2 1 ( a ) は、1つのwindowに1つのODSが存在するケースを想定した図である。図 2 1 ( b ) ( c ) は、図 1 8 で引用した各数値の時間的な前後関係を示すタイミングチャートである。本チャートには3つの段があり、これらの段のうち、" グラフィックスプレーンアクセス " の段、" ODSデコード " の段は、再生時にパラレルになされる2つの処理を示す。上述したアルゴリズムは、これらの2つの処理のパラレル実行を想定している。

40

## 【 0 0 8 8 】

グラフィックスプレーンアクセスは、クリア期間(1)、書き込み期間(3)からなる。このクリア期間(1)は、グラフィックスプレーン全体のクリアに要する期間(90,000×(グラフィックスプレーンのサイズ//256,000,000))、グラフィックスプレーンにおける全windowのクリアに要する時間( (90,000×(window[i]のサイズ//256,000,000)))のどちらかを意味する。

## 【 0 0 8 9 】

書き込み期間(3)は、window全体描画に要する期間(90,000×(windowサイズ//256,000,0

50

00))を意味する。

一方、デコード期間(2)は、ODSのDTSからPTSまでに示される期間を意味する。これらクリア期間(1)～書き込み期間(3)は、クリアすべき範囲、デコードすべきODSのサイズ、グラフィックスプレーンに書き込むべきグラフィクスオブジェクトのサイズにより変化し得る。本図では、説明の簡略化のため、ODSのデコード期間(2)の始点は、クリア期間(1)の始点と同一であると仮定している。

#### 【0090】

図21(b)はデコード期間(2)が長くなるケースであり、Decode\_Durationはデコード期間(2)＋書き込み期間(3)になる。

図21(c)は、クリア期間(1)が長くなるケースであり、Decode\_Durationはクリア期間(1)＋書き込み期間(3)の期間がDecode\_Durationになる。

図22(a)～(c)は、1つのwindowに2つのODSが存在するケースを想定した図である。本図(b)(c)におけるデコード期間(2)は、2つのグラフィクスのデコードに要する期間の総和を意味する。グラフィクス書込期間(3)も、2つのグラフィクスをグラフィックスプレーンに書き込む期間の総和を意味する。

ODSが2つになっているものの、図21と同様に考えればDecode\_Durationを算出することができる。2つのODSをデコードするためのデコード期間(2)が長い場合は、図22(b)に示すようにDecode\_Durationはデコード期間(2)＋書き込み期間(3)に算出されることになる。

#### 【0091】

クリア期間(1)が長い場合は、図22(c)に示すように、Decode\_Durationはクリア期間(1)＋書き込み期間(3)となる。

図23(a)は、2つのwindowのそれぞれに、ODSが1つずつ存在するケースを想定している。この場合でもクリア期間(1)が、2つのODSをデコードするための総デコード期間(2)より長い場合、Decode\_Durationはクリア期間(1)＋書き込み期間(3)になる。問題は、クリア期間(1)がデコード期間(2)より短くなるケースである。この場合デコード期間(2)の経過を待たずに、1つ目のwindowへの書き込みは可能になる。そのためクリア期間(1)＋書き込み期間(3)、デコード期間(2)＋書き込み期間(3)の長さにはならない。ここで1つ目のODSのデコードに要する期間を書込期間(31)、2つ目のODSのデコードに要する期間を書込期間(32)とする。図23(b)は、デコード期間(2)がクリア期間(1)＋書込期間(31)より長くなるケースを示す。この場合Decode\_Durationは、デコード期間(2)＋書込期間(32)になる。

#### 【0092】

図23(c)は、クリア期間(1)＋書込期間(31)がデコード期間(2)より長くなるケースを示す。この場合Decode\_Durationはクリア期間(1)＋書込期間(31)＋書込期間(32)になる。

グラフィックスプレーンのサイズは、プレーヤモデルから予め判明しており、またwindowのサイズ、ODSのサイズ、個数もオーサリングの段階で判明しているので、これらからDecode\_Durationがクリア期間(1)＋書き込み期間(3)、デコード期間(2)＋書き込み期間(3)、デコード期間(2)＋書込期間(32)、クリア期間(1)＋書込期間(31)＋書込期間(32)のどれかになる。こうしたDecode\_Duration算出を基にPCSのPTSを設定すれば、ピクチャデータとの同期表示を高い時間精度で実現することができる。このような高精度な同期制御は、windowを定義し、クリア・再描画する範囲を、このwindowに限定することで成り立っているので、オーサリングの環境に、このwindowの概念を導入したことの意義は大きい。

#### 【0093】

続いてDSnにおけるWDSのDTS、PTSの設定について説明する。WDSのDTSは、以下の数式を満たすように設定すればよい。

$$DTS(DSn[WDS]) \quad DTS(DSn[PCS])$$

一方、DSnにおけるWDSのPTSは、グラフィックスプレーンに対する書き込みを開始すべきデッドラインを示す。グラフィックスプレーンへの書き込みは、ウィンドウだけで足り

10

20

30

40

50

るので、PCSのPTSに示される時刻から、WDSの書き込みに要する期間を差し引けば、グラフィックスプレーンへの書き込みを開始すべき時刻は定まる。WDSの総サイズを  $SIZE(WDS.WIN[i])$  とすれば、これのクリア及び再描画に要する時間は、『  $SIZE(WDS.WIN[i]) // 256,000,000$  』になる。そして、これを90.000KHzの時間精度で表現すると『  $90000 \times SIZE(WDS.WIN[i]) // 256,000,000$  』になる。

#### 【0094】

故に以下の数式から、WDSのPTSを算出すればよい。

$$PTS(DSn[WDS]) = PTS(DSn[PCS]) - 90000 \times SIZE(WDS.WIN[i]) // 256,000,000$$

このWDSに示されるPTSはデッドラインなので、これより早い時点からグラフィックスプレーンの書き込みを開始してもよい。つまり図23に示すように、2つのウィンドウのうち、1つのウィンドウに表示させるべきODSのデコードが完了したなら、その時点からデコードにより得られたグラフィックスオブジェクトの書き込みを開始してもよい。

10

#### 【0095】

このようにWDSに付加されたDTS、PTSを用いてAVClipの再生時間軸の任意の時点に、ウィンドウを割り当てることができる。

以上が再生時間軸に対するDisplay Setの割り当てについての説明である。

Display Setに対するDTS、PTSの設定について、図24～図25の具体例を交えながら説明する。図24は、4回のグラフィックスプレーンへの書き込みにより字幕を表示することを想定した具体例である。この具体例は、2つの字幕『what is blu-ray』、『blu-ray is everywhere』を1つずつ表示させるというアップデートを想定している。図24は、本具体例が想定しているアップデートの、時間的変遷がどのようなものであるかを示す図である。時点t1では『what』までを表示し、続く時点t2では『what is』まで、時点t3では『what is blu-ray』全体を表示させる。字幕の全貌を明らかにした上で時点t4において別の字幕『blu-ray is everywhere』を表示させるのである。

20

#### 【0096】

図25(a)は、かかるアップデートを実現するため、記述された4つのDisplay Setを示す図である。DS1は、時点t1におけるアップデートを制御するPCS1.2と、彩色のためのPDS1と、字幕『what is blu-ray』に相当するODS1と、DS1の終了コードであるENDとからなる。

DS2は、時点t2におけるアップデートを制御するPCS1.2と、ENDとからなる。DS3は、時点t3におけるアップデートを制御するPCS1.3と、ENDとからなる。DS4は、時点t2のアップデートを制御するためのPCS2と、色変換のためのPDS2と、字幕『blu-ray is everywhere』に相当するODS2と、ENDとからなる。

30

#### 【0097】

これら4つのDisplay Setに属する各機能セグメントのDTS、PTSをどのように設定するかを、図25(b)のタイミングチャートを参照して説明する。

このタイミングチャートの再生時間軸は、図24の再生時間軸と同じものである。このタイミングチャートにおいて、PTS(PCS1.1)、PTS(PCS1.2)、PTS(PCS1.3)、PTS(PCS2)は、それぞれ『what』の表示時点t1、『what is』の表示時点t2、『what is blu-ray』の表示時点t3、『blu-ray is everywhere』の表示時点t4を示すよう設定される。これは、それぞれの字幕の表示時点において各PCSに記述された制御(Cropなど)を実行する必要があるからである。

40

#### 【0098】

PTS(ODS1)、PTS(ODS2)は、PTS(PCS1.1)及びPTS(PCS2)の時点からDecode\_Durationを差し引いた時点を示すよう設定されている。PTS(PCS)は、数式  $PTS(DSn[PCS]) - DTS(DSn[PCS]) + decodeduration(DSn)$  を満たすよう設定する必要があるからである。

図25(b)では、時点t4より手前の時点t5を示すよう、PTS(ODS2)は設定されており、時点t1より手前の時点t0を示すよう、PTS(ODS1)は設定されている。DTS(ODS1)、DTS(ODS2)は、PTS(ODS1)及びPTS(ODS2)の時点からデコード期間を差し引いた時点を示すよう設定されている。DTS(ODS)は、数式  $PTS(DS[ODSj]) = DTS(DSn[ODSj]) + 90,000 \times (SIZE(DSn[ODSj])$

50

)//Rd)を満たすよう設定する必要があるからである。図25(b)では、時点t5より手前の時点t0を示すよう、PTS(ODS2)は設定されており、時点t0より手前の時点を示すよう、PTS(ODS1)は設定されている。

#### 【0099】

ここでDTS(ODS2)=PTS(ODS1)の関係が満たされることがわかる。これにより再生装置の実装によっては、ODS1がオブジェクトバッファに読み出された後、このオブジェクトバッファ上のODS1を上書きする形で、ODS2を読み出すことができる。あるODSのPTSを、先に表示すべきODSのPTSに設定しておけば、古いODSを上書きする形で、新しいODSをメモリに読み出すという動作を再生装置が行うことにより、小メモリ規模での再生処理を実現することができる。かかる再生処理の実現により、再生装置におけるメモリ規模の選択の幅は広がる。

10

#### 【0100】

PCS1.1のDTSは、DTS(PCS1.1)=DTS(ODS1)になるように設定されている。これは、PCS1.1のDTS値は、DTS(ODS1)により示される時点以前であるなら、どの時点でもよいからである。

ODS1のPTS値、ODS2のDTS値、PCS1.2、PCS1.3、PCS2のPTS値は、PTS(ODS1)=DTS(ODS2)=PTS(PCS1.2)=PTS(PCS1.3)=DTS(PCS2)の関係を満たすよう、何れも時点t0に設定されている。

#### 【0101】

これは、PCS1.2、PCS1.3のDTS値は、PTS(PCS1.2)、PTS(PCS1.3)以前であれば、どの時点でもよく、またPCS2のDTS値も、DTS(PCS2)以前であれば、どの時点でもよいという自由度があるからである。これにより、複数のPCSをまとめて読み出しおき、前のPCSによるアップデートの実行が終われば続くPCSによるアップデートを即座に実行することができる。

20

#### 【0102】

PCSのDTS・PTS、ODSのDTS・PCSは、上述の数式で示したような不等号の関係さえ最低限満たせばよいので、DTS(ODS2)=PTS(ODS1)になるように設定したり、PTS(ODS1)=DTS(ODS2)=PTS(PCS1.2)=PTS(PCS1.3)=DTS(PCS2)になるように設定することも可能になる。こうしたタイムスタンプの設定により、デコード負荷が発生する期間やバッファの占有量が増える期間を、調整することができる。かかる調整が可能になるので、再生時になしえる再生制御の幅が広がり、オーサリングを行う技術者及び再生装置を製造する技術者にとってのメリットは大きい。これで、DTS、PTS設定の具体例の説明を終える。

30

#### 【0103】

以上説明したDisplay Set(PCS,WDS,PDS,ODS)のデータ構造は、プログラミング言語で記述されたクラス構造体のインスタンスであり、オーサリングを行う制作者は、Blu-ray Disc Prerecording Formatに規定された構文に従い、クラス構造体を記述することで、BD-ROM上のこれらのデータ構造を得ることができる。以上が本発明に係る記録媒体の実施形態である。続いて本発明に係る再生装置の実施形態について説明する。図26は、本発明に係る再生装置の内部構成を示す図である。本発明に係る再生装置は、本図に示す内部に基づき、工業的に生産される。本発明に係る再生装置は、主としてシステムLSIと、ドライブ装置、マイコンシステムという3つのパーツからなり、これらのパーツを装置のキャビネット及び基板に実装することで工業的に生産することができる。システムLSIは、再生装置の機能を果たす様々な処理部を集積した集積回路である。こうして生産される再生装置は、BDドライブ1、Read Buffer 2、PIDフィルタ3、Transport Buffer 4a,b,c、周辺回路4d、ビデオデコーダ5、ビデオプレーン6、オーディオデコーダ7、グラフィクスプレーン8、CLUT部9、加算器10、グラフィクスデコーダ12、Coded Data Buffer 13、周辺回路13a、Stream Graphics Processor 14、Object Buffer 15、Composition Buffer 16、Graphical Controller 17から構成される。

40

#### 【0104】

BD-ROMドライブ1は、BD-ROMのローディング/リード/イジェクトを行い、BD-ROMに対

50

するアクセスを実行する。

Read Buffer 2 は、FIFOメモリであり、BD-ROMから読み出されたTSパケットが先入れ先出し式に格納される。

PIDフィルタ 3 は、Read Buffer 2 から出力される複数TSパケットに対してフィルタリングを施す。PIDフィルタ 3 によるフィルタリングは、TSパケットのうち、所望のPIDをもつもののみをTransport Buffer 4 a,b,cに書き込むことでなされる。PIDフィルタ 3 によるフィルタリングでは、バッファリングは必要ではない。従って、PIDフィルタ 3 に入力されたTSパケットは、時間遅延なく、Transport Buffer 4 a,b,cに書き込まれる。

【 0 1 0 5 】

Transport Buffer 4 a,b,cは、PIDフィルタ 3 から出力されたTSパケットを先入れ先出し式に格納しておくメモリである。このTransport Buffer 4 a,b,cからTSパケットが取り出される速度を速度Rxとする 10

周辺回路 4 dは、Transport Buffer 4 a,b,cから読み出されたTSパケットを、機能セグメントに変換する処理を行うワイアロジックである。変換により得られた機能セグメントはCoded Data Buffer 1 3に格納される。

【 0 1 0 6 】

ビデオデコーダ 5 は、PIDフィルタ 3 から出力された複数TSパケットを復号して非圧縮形式のピクチャを得てビデオプレーン 6 に書き込む。

ビデオプレーン 6 は、動画用のプレーンメモリである。

オーディオデコーダ 7 は、PIDフィルタ 3 から出力されたTSパケットを復号して、非圧縮形式のオーディオデータを出力する。 20

【 0 1 0 7 】

グラフィクスプレーン 8 は、一画面分の領域をもったプレーンメモリであり、一画面分の非圧縮グラフィクスを格納することができる。

CLUT部 9 は、グラフィクスプレーン 8 に格納された非圧縮グラフィクスにおけるインデックスカラーを、PDSに示されるY,Cr,Cb値に基づき変換する。

加算器 1 0 は、CLUT部 9 により色変換された非圧縮グラフィクスに、PDSに示されるT値(透過率)を乗じて、ビデオプレーン 6 に格納された非圧縮状態のピクチャデータと画素毎に加算し、合成画像を得て出力する。

【 0 1 0 8 】

グラフィクスデコーダ 1 2 は、グラフィクスストリームをデコードして、非圧縮グラフィクスを得て、これをグラフィクスオブジェクトとしてグラフィクスプレーン 8 に書き込む。グラフィクスストリームのデコードにより、字幕やメニューが画面上に現れることになる。このグラフィクスデコーダ 1 2 は、Coded Data Buffer 1 3、周辺回路 1 3 a、Stream Graphics Processor 1 4、Object Buffer 1 5、Composition Buffer 1 6、Graphical Controller 1 7 から構成される。 30

【 0 1 0 9 】

Coded Data Buffer 1 3 は、機能セグメントがDTS、PTSと共に格納されるバッファである。かかる機能セグメントは、Transport Buffer 4 a,b,cに格納されたトランスポートストリームの各TSパケットから、TSパケットヘッダ、PESパケットヘッダを取り除き、ペイロードをシーケンシャルに配列することにより得られたものである。取り除かれたTSパケットヘッダ、PESパケットヘッダのうち、PTS/DTSは、PESパケットと対応付けて格納される。 40

【 0 1 1 0 】

周辺回路 1 3 aは、Coded Data Buffer 1 3 - Stream Graphics Processor 1 4 間の転送、Coded Data Buffer 1 3 - Composition Buffer 1 6 間の転送を実現するワイアロジックである。この転送処理において現在時点がODSのDTSに示される時刻になれば、ODSを、Coded Data Buffer 1 3 からStream Graphics Processor 1 4 に転送する。また現在時刻がPCS、PDSのDTSに示される時刻になれば、PCS、PDSをComposition Buffer 1 6 に転送するという処理を行う。 50



## 【 0 1 1 1 】

Stream Graphics Processor 1 4 は、ODSをデコードして、デコードにより得られたインデックスカラーからなる非圧縮状態の非圧縮グラフィックスをグラフィクスオブジェクトとしてObject Buffer 1 5 に書き込む。このStream Graphics Processor 1 4 によるデコードは、ODSに関連付けられたDTSの時刻に開始し、ODSに関連付けられたPTSに示されるデコード終了時刻までに終了する。上述したグラフィックオブジェクトのデコードレートRdは、このStream Graphics Processor 1 4 の出力レートである。

## 【 0 1 1 2 】

Object Buffer 1 5 は、ETSI EN 300 743標準規格におけるピクセルバッファに相当するバッファであり、Stream Graphics Processor 1 4 のデコードにより得られたグラフィクスオブジェクトが配置される。Object Buffer 1 5 は、グラフィクスプレーン 8 の2倍 / 4 倍の大きさに設定せねばならない。何故ならScrollingを実現する場合を考えると、グラフィクスプレーン 8 の2倍、4倍のグラフィクスオブジェクトを格納しておかねばならないからである。

10

## 【 0 1 1 3 】

Composition Buffer 1 6 は、PCS、PDSが配置されるメモリである。

Graphical Controller 1 7 は、Composition Buffer 1 6 に配置されたPCSを解読して、PCSに基づく制御をする。この制御の実行タイミングは、PCSに付加されたPTSの値に基づく。以上が再生装置の構成要素である。

続いて、PIDフィルタ 3、Transport Buffer 4 a,b,c、グラフィクスプレーン 8、CLUT部 9、Coded Data Buffer 1 3 ~ Graphical Controller 1 7 を構成するための、転送レート、バッファサイズの推奨値について説明する。図 2 7 は、書込レートRx,Rc,Rd、グラフィクスプレーン 8、Coded Data Buffer 1 3、Object Buffer 1 5、Composition Buffer 1 6 のサイズを示す図である。

20

## 【 0 1 1 4 】

Object Buffer 1 5 - グラフィクスプレーン 8 間の転送レートRcは、本装置において最も高い転送レートであり、ウィンドウサイズ、フレームレートから256Mbps(=500Kバイト × 29.97 × 2)と算出される。

Stream Graphics Processor 1 4 - Object Buffer 1 5 間の転送レートRd(Pixel Decoding Rate)は、Rcとは異なり、ビデオフレームの周期によるアップデートは要求されずRcの1/2, 1/4でよい。故に128Mbps, 64Mbpsになる。

30

## 【 0 1 1 5 】

Transport Buffer 4 a,b,c - Coded Data Buffer 1 3 間のTransport Buffer LeakレートRxは、圧縮状態たるODSの転送レートである。従ってTransport Buffer Leakレートは、Rdに圧縮率を乗じた転送レートでよい。ODSの圧縮率を25%と仮定すれば、16Mbps(=64Mbps × 25%)で足りる。

この図に示す転送レート、バッファ規模はあくまでもミニマムスタンダードであり、これより大きい値での実装を否定している訳ではない。

## 【 0 1 1 6 】

以上のように構成された再生装置において、各構成要素はパイプライン式にデコード処理を行うことができる。

40

図 2 8 は、再生装置によるパイプライン処理を示すタイミングチャートである。第 5 段目は、BD-ROMにおけるDisplay Setを示し、第 4 段目は、Coded Data Buffer 1 3 へのPCS、WDS、PDS、ODSの読出期間を示す。第 3 段目は、Stream Graphics Processor 1 4 による各ODSのデコード期間を、第 2 段目はComposition Buffer 1 6 の格納内容を、第 1 段目はGraphical Controller 1 7 の処理内容を示す。

## 【 0 1 1 7 】

ODS1,2に付与されたDTS(デコード開始時刻)は、図中のt31,t32の時点を示している。デコード開始時刻がDTSに規定されているので、各ODSは、自身のDTSに示される時刻までにCoded Data Buffer 1 3 に読み出されなければならない。そのためODS1の読み出しは、Code

50

d Data Buffer 1 3 へのODS1のデコード期間dp1の直前までに完了している。Coded Data Buffer 1 3 へのODS2の読み出しは、ODS2のデコード期間dp2の直前までに完了している。

#### 【 0 1 1 8 】

一方、ODS1,2に付与されたPTS(デコード終了時刻)は、図中のt32,t33の時点を示している。Stream Graphics Processor 1 4 によるODS1のデコードはt32までに完了し、ODS2のデコードは、t33に示される時刻までに完了する。以上のように、Stream Graphics Processor 1 4 は、各ODSのDTSに示される時刻までに、ODSをCoded Data Buffer 1 3 に読み出し、Coded Data Buffer 1 3 に読み出されたODSを、各ODSのPTSに示される時刻までに、デコードしてObject Buffer 1 5 に書き込む。

#### 【 0 1 1 9 】

本図の第 1 段目における期間cd1は、Graphics Controller 1 7 がグラフィクスプレーン 8 をクリアするのに要する期間である。また期間td1は、Object Buffer 1 5 上にえられたグラフィクスオブジェクトを、グラフィクスプレーン 8 に書き込むのに要する期間である。WDSのPTSは、この書き込みの開始にあたってのデッドラインを示し、PCSのPTSはこの書き込みの終了時点及び表示タイミングを示す。このPCSのPTSに示される時点になれば、対話画面を構成する非圧縮グラフィクスがグラフィクスプレーン 8 上に得られることになる。この非圧縮グラフィクスの色変換をCLUT部 9 に行わせ、ビデオプレーン 6 に格納されている非圧縮ピクチャとの合成を加算器 1 0 に行わせれば、合成画像が得られることになる。

#### 【 0 1 2 0 】

グラフィクスデコーダ 1 2 において、Graphics Controller 1 7 がグラフィクスプレーン 8 のクリアを実行している間においても、Stream Graphics Processor 1 4 のデコードは継続して行われる。以上のようなパイプライン処理により、グラフィクスの表示を迅速に実施することができる。

図 2 8 では、グラフィックスプレーンのクリアが、ODSのデコードより早く終わる場合を想定したが、図 2 9 は、ODSのデコードが、グラフィックスプレーンのクリアより早く終わる場合を想定したパイプライン処理を示すタイミングチャートである。この場合、ODSのデコードが完了した段階では、グラフィックスプレーンへの書き込みを実行することができず、グラフィックスプレーンのクリアが完了した時点で、デコードにより得られたグラフィクスをグラフィックスプレーンに書き込むことができる。

#### 【 0 1 2 1 】

再生装置におけるバッファ占有量の時間的遷移について図 3 0 を参照しながら説明する。図 3 0 は、図 2 6 におけるCompositionバッファ 1 6、Object Buffer 1 5、Coded Data バッファ 1 3、グラフィクスプレーン 8 の時間的遷移を示すタイミングチャートである。本図は、第 1 段目から第 4 段目までに、グラフィクスプレーン 8、Object Buffer 1 5、Coded Data バッファ 1 3、Compositionバッファ 1 6 における占有量の時間的遷移を示している。この占有量の時間的遷移は、横軸を時間軸とし、縦軸を占有量とした折れ線グラフの表記で表現している。

#### 【 0 1 2 2 】

図 3 0 の第 4 段目は、Compositionバッファ 1 6 における占有量の時間的遷移を示す。本図に示すようにCompositionバッファ 1 6 の時間的遷移は、Coded Data バッファ 1 3 から出力されPCSが格納されることによる単調増加vf0を含む。

第 3 段目は、Coded Data バッファ 1 3 における占有量の時間的遷移を示す。本図に示すようにCoded Data バッファ 1 3 の時間的遷移は、ODSが格納されることによる単調増加Vf1、Vf2と、格納されたODSが順次Stream Graphicsプロセッサ 1 4 により取り出されることによる単調減少Vg1、Vg2とを含む。単調増加Vf1、Vf2の傾きは、Transportバッファ 4 a,b,cからCoded Data バッファ 1 3 への出力レートRxに基づき、単調減少Vg1、Vg2の傾きは、Stream Graphicsプロセッサ 1 4 によるデコードであり、瞬時に実行される。つまりODSに対するデコードは瞬時に行われ、Stream Graphicsプロセッサ 1 4 は、デコードにより得られた非圧縮グラフィクスを保持する。Stream Graphicsプロセッサ 1 4 からObject Buffer 1

10

20

30

40

50

5 への伝送路の書込レートは128Mbpsであるため、この書込レートにより、Object Buffer 1 5 の占有量は増加する。

【 0 1 2 3 】

第 2 段目は、Object Buffer 1 5 における占有量の時間的遷移を示す。本図に示すようにObject Buffer 1 5 の時間的遷移は、Stream Graphicsプロセッサ 1 4 から出力されたODSが格納されることによる単調増加Vh1,Vh2を含む。単調増加Vh1,Vh2の傾きは、Stream Graphicsプロセッサ 1 4 からObject Buffer 1 5 への転送レートRcに基づく。第 3 段目の単調減少が生じる期間及びの第 2 段目の単調増加が生ずる期間が、デコード期間である。このデコード期間の始期は、ODSのDTSに示されており、デコード期間の終期は、ODSのPTSに示されている。このODSのPTSに示される期間までに、非圧縮のグラフィクスがobject buffer 1 5 に格納されれば、ODSに対するデコードは完了したことになる。ODSのPTSに示される期間までに、非圧縮のグラフィクスがobject buffer 1 5 に格納されることが、必須の要件であり、このデコード期間における単調増加、単調減少はどのようなものであってもよい。

【 0 1 2 4 】

第 1 段目は、グラフィクスプレーン 8 における占有量の時間的遷移を示す。本図に示すようにグラフィクスプレーン 8 の時間的遷移は、Object Buffer 1 5 から出力されたデコード済みODSが格納されることによる単調増加Vf3を含む。単調増加Vf3の傾きは、Object Buffer 1 5 からグラフィクスプレーン 8 への転送レートRdに基づく。この単調増加の終期は、ODSのPTSに示されている。

【 0 1 2 5 】

ODSに付与されたDTS、PTS、ICSに付与されたDTS、PTS、そして図 2 7 に示した各バッファのサイズ、転送レートを用いれば、本図のようなグラフを作図することにより、BD-ROMにて供給すべきAVClipの再生時において、各バッファの状態がどのように変化するかが、オーサリングの段階で明らかになる。

このバッファ状態の遷移は、DTS、PTSに示される値を書き換えることで、調整することが可能なので、再生装置側のデコードのスペックを越えるような復号負荷の発生を回避したり、再生にあたってのバッファオーバーフローの回避することができる。そのため再生装置の開発にあたってのハードウェア、ソフトウェアの実装が簡易になる。

【 0 1 2 6 】

以上が再生装置の内部構成である。続いて制御部 2 0 及びグラフィクスデコーダ 1 2 を、どのようにして実装するかについて説明する。制御部 2 0 は、図 3 1 の処理手順を行うプログラムを作成し、汎用CPUに実行させることにより実装可能である。以降、図 3 1 を参照しながら、制御部 2 0 の処理手順について説明する。

図 3 1 は、機能セグメントのロード処理の処理手順を示すフローチャートである。本フローチャートにおいてSegmentKとは、AVClipの再生時において、読み出されたSegment(PCS,WDS,PDS,ODS)のそれぞれを意味する変数であり、無視フラグは、このSegmentKを無視するかロードするかを切り換えるフラグである。本フローチャートは、無視フラグを0に初期化した上で、ステップS 2 1 ~ S 2 4、ステップS 2 7 ~ S 3 1 の処理を全てのSegmentKについて繰り返すループ構造を有している(ステップS 2 5、ステップS 2 6)。

【 0 1 2 7 】

ステップS 2 1 は、SegmentKがPCSであるか否かの判定であり、もしSegmentKがPCSであれば、ステップS 2 7、ステップS 2 8 の判定を行う。

ステップS 2 2 は、無視フラグが1かどうかの判定である。無視フラグが0であるならステップS 2 3 に移行し、1であるならステップS 2 4 に移行する。無視フラグが1であれば(ステップS 2 2 でYes)、ステップS 2 3 においてSegmentKをCoded Data Buffer 1 3 にロードする。

【 0 1 2 8 】

無視フラグが0に設定されていれば(ステップS 2 2 がNo)、ステップS 2 4 においてSegmentKが無視される。これによりDSに属する機能セグメントは全て、ステップS 2 2 がNo

10

20

30

40

50

になって、無視されることになる(ステップS 2 4)。

このように、SegmentKが無視されるか、ロードされるかは、無視フラグの設定により決まる。ステップS 2 7 ~ S 3 1、S 3 4、S 3 5は、この無視フラグを設定する処理である。

#### 【0 1 2 9】

ステップS 2 7は、PCSにおけるComposition\_stateがAcquisition Pointであるか否かの判定である。SegmentKがAcquisition Pointであるなら、ステップS 2 8に移行し、SegmentKがもしEpoch StartがNormal Caseであるなら、ステップS 3 1に移行する。

ステップS 2 8は、先行するDSがグラフィクスデコーダ1 2内のどれかのバッファ(Coded Data Buffer 1 3、Stream Graphicsプロセッサ1 4、Object Buffer 1 5、Composition Buffer 1 6)に存在するかどうかの判定であり、ステップS 2 7がYesである場合に実行される。グラフィクスデコーダ1 2内にDSが存在しないケースとは、頭出しがなされたケースをいう。この場合、Acquisition PointたるDSから、表示を開始せねばならないので、ステップS 3 0に移行する(ステップS 2 8でNo)。

#### 【0 1 3 0】

ステップS 3 0は、無視フラグを0に設定し、ステップS 2 2に移行する。

グラフィクスデコーダ1 2内にDSが存在するケースとは、通常再生がなされたケースをいう。この場合、ステップS 2 9に移行する(ステップS 2 8でYes)。ステップS 2 9は、無視フラグを1に設定し、ステップS 2 2に移行する。

ステップS 3 1は、PCSにおけるComposition\_stateがNormal Caseであるか否かの判定である。もしNormal Caseであるなら、ステップS 3 4に移行する。SegmentKがEpoch Startであるなら、ステップS 3 0において無視フラグを0に設定する。ステップS 3 4は、ステップS 2 8と同じであり、先行するDSがグラフィクスデコーダ1 2内に存在するかどうかの判定を行う。もし存在するなら、無視フラグを0に設定する(ステップS 3 0)。存在しないなら、元々、対話画面を構成する十分な機能セグメントが得られないため、無視フラグを1に設定する(ステップS 3 5)。かかるフラグ設定により、先行するDSがグラフィクスデコーダ1 2に存在しない場合、Normal Caseを構成する機能セグメントは無視されることになる。

#### 【0 1 3 1】

DSが、図3 2のように多重化されている場合を想定して、DSの読み出しがどのように行われるかを説明する。図3 2の一例では、3つのDSが動画と多重化されている。この3つのDSのうち、初めのDS1は、Composition\_stateがEpoch\_Start、DS10はAcquisition Point、DS20は、Normal Caseである。

かかる3つのDSが、動画と多重化されているAVClipにおいて、ピクチャデータpt10からの頭出しが矢印am1に示すように行われたものとする。この場合、頭出し位置に最も近いDS10が、図3 1のフローチャートの対象となる。ステップS 2 7においてComposition\_stateはAcquisition Pointと判定されるが、先行するDSはCoded Data Buffer 1 3上に存在しないため、無視フラグは0に設定され、このDS10が図3 3の矢印md1に示すように再生装置のCoded Data Buffer 1 3にロードされる。一方、頭出し位置がDS10の存在位置より後である場合(図3 2の矢印am2)、DS20は、Normal CaseのDisplay Setであり、先行するDS20はCoded Data Buffer 1 3に存在しないので、このDisplay Setは、無視されることになる(図3 3の矢印md2)。

#### 【0 1 3 2】

図3 4のように通常再生が行われた場合のDS1,10,20のロードは、図3 4に示すものとなる。3つのDSのうち、PCSのComposition\_stateがEpoch StartであるDS1は、そのままCoded Data Buffer 1 3にロードされるが(ステップS 2 3)、PCSのComposition\_stateがAcquisition PointであるDS10については、無視フラグが1に設定されるため(ステップS 2 9)、これを構成する機能セグメントはCoded Data Buffer 1 3にロードされず無視される(図3 5の矢印rd2,ステップS 2 4)。またDS20については、PCSのComposition\_stateはNormal Caseなので、Coded Data Buffer 1 3にロードされる(図3 5の矢印rd3)。

## 【 0 1 3 3 】

続いてGraphical Controller 1 7 の処理手順について説明する。図 3 6 ~ 図 3 8 は、Graphical Controller 1 7 の処理手順を示すフローチャートである。

ステップ S 4 1 ~ ステップ S 4 4 は、本フローチャートのメインルーチンであり、ステップ S 4 1 ~ ステップ S 4 4 に規定した何れかの事象の成立を待つ。

ステップ S 4 1 は、現在の再生時点がPCSのPTS時刻になっているか否かの判定であり、もしなっていれば、ステップ S 4 5 ~ ステップ S 5 3 の処理を行う。

## 【 0 1 3 4 】

ステップ S 4 5 は、PCSのcomposition\_stateが、Epoch\_Startであるか否かの判定であり、もしEpoch\_Startであるなら、ステップ S 4 6 においてグラフィクスプレーン 8 を全クリアする。それ以外であるなら、ステップ S 4 7 においてWDSのwindow\_horizontal\_position、window\_vertival\_position、window\_width、window\_heightに示されるwindowをクリアする。

10

## 【 0 1 3 5 】

ステップ S 4 8 は、ステップ S 4 6 又はステップ S 4 7 でのクリア後の実行されるステップであり、任意のODSxのPTS時刻が既に経過しているか否かの判定である。つまり、グラフィクスプレーン 8 全体のクリアにあたっては、そのクリア時間に長時間を費するので、あるODS(ODSx)のデコードが既に完了していることもある。ステップ S 4 8 はその可能性を検証している。もし経過していないなら、メインルーチンにリターンする。どれかのODSのデコード時刻を経過しているなら、ステップ S 4 9 ~ ステップ S 5 1 を実行する。ステップ S 4 9 は、object\_crop\_flagが0を示しているか否かの判定であり、もし0を示しているなら、グラフィクスオブジェクトを非表示とする(ステップ S 5 0 )。

20

## 【 0 1 3 6 】

もし0を示していないなら、object\_cropping\_horizontal\_position、object\_cropping\_vertival\_position、cropping\_width、cropping\_heightに基づきクロップされたグラフィクスオブジェクトを、グラフィクスプレーン 8 のwindowにおいて object\_cropping\_horizontal\_position、object\_cropping\_vertival\_positionに示される位置に書き込む(ステップ S 5 1 )。以上の処理により、ウィンドウに1つ以上のグラフィクスオブジェクトが描かれることになる。

## 【 0 1 3 7 】

ステップ S 5 2 は、別のODSyのPTS時刻が経過しているか否かの判定である。ODSxをグラフィクスプレーン 8 に書き込んでいる際、別のODSのデコードが既に完了していれば、このODSyをODSxにして(ステップ S 5 3 )、ステップ S 4 9 に移行する。これにより、別のODSに対しても、ステップ S 4 9 ~ S 5 1 の処理が繰り返し行われる。

30

次に図 3 7 を参照して、ステップ S 4 2、ステップ S 5 4 ~ ステップ S 5 9 について説明する。

## 【 0 1 3 8 】

ステップ S 4 2 は、現在の再生時点がWDSのPTSであるか否かの判定であり、もしWDSのPTSであるなら、ステップ S 5 4 においてウィンドウが1つであるか否かを判定し、もし2つであれば、メインルーチンにリターンする。ウィンドウが1つであるなら、ステップ S 5 5 ~ ステップ S 5 9 のループ処理を行う。このループ処理は、ウィンドウに表示される2つのグラフィクスオブジェクトのそれぞれについて、ステップ S 5 7 ~ ステップ S 5 9 を実行するというものである。ステップ S 5 7 は、object\_crop\_flagが0を示しているか否かの判定であり、もし0を示しているなら、グラフィクスオブジェクトを非表示とする(ステップ S 5 8 )。

40

## 【 0 1 3 9 】

もし0を示していないなら、object\_cropping\_horizontal\_position、object\_cropping\_vertival\_position、cropping\_width、cropping\_heightに基づきクロップされたグラフィクスオブジェクトを、グラフィクスプレーン 8 のwindowにおいて object\_cropping\_horizontal\_position、object\_cropping\_vertival\_positionに示される位置に書き込む(ステッ

50

ブ S 5 9)。以上の処理を繰り返せば、ウィンドウに1つ以上のグラフィクスオブジェクトが描かれることになる。

#### 【 0 1 4 0 】

ステップ S 4 4 は、現在の再生時点がPCSのPTSに示される時点であるかの判定であり、もしそうであるなら、ステップ S 6 0 においてPallet\_update\_flagが1を示しているか否かを判定する。もし1を示しているなら、pallet\_idに示されるPDSをCLUT部に設定する(ステップ S 6 1)。0を示しているなら、ステップ S 6 1 をスキップする。

その後、グラフィクスプレーン 8 におけるグラフィクスオブジェクトの色変換をCLUT部に行わせて、動画像と合成する(ステップ S 6 2)。

#### 【 0 1 4 1 】

10

次に図 3 8 を参照して、ステップ S 4 3、ステップ S 6 4 ~ ステップ S 6 6 について説明する。

ステップ S 4 3 は、現在の再生時点がODSのPTSであるか否かの判定であり、もしODSのPTSであるなら、ステップ S 6 3 においてウィンドウが2つであるか否かを判定し、もし1つであれば、メインルーチンにリターンする。ウィンドウが2つであるなら、ステップ S 6 4 ~ ステップ S 6 6 を行う。ステップ S 6 4 は、object\_crop\_flagが0を示しているか否かの判定であり、もし示しているなら、グラフィクスオブジェクトを非表示とする(ステップ S 6 5)。

#### 【 0 1 4 2 】

もし0を示していないなら、object\_cropping\_horizontal\_position、object\_cropping\_vertival\_position、cropping\_width、cropping\_heightに基づきクロップされたグラフィクスオブジェクトを、グラフィクスプレーン 8 のwindowにおいて object\_cropping\_horizontal\_position、object\_cropping\_vertival\_positionに示される位置に書き込む(ステップ S 6 6)。以上の処理を繰り返せば、各ウィンドウにグラフィクスオブジェクトが描かれることになる。

20

#### 【 0 1 4 3 】

以上、DSnに属するPCSのDTS,PTS,ODSのDTS、PTSの設定について説明したが、PDSのDTS、PTS、ENDのDTS、PTSについては説明していない。先ず、DSnに属する各PDSのDTS,PTSの設定について説明する。

DSnに属する各PDSは、PCSがComposition Buffer 1 6 にロードされる時点(DTS(DSn[PCS]))から、最初のODSのデコード開始時点(DTS(DSn[ODS1]))までに、CLUT部 9 において、有効になればよい。このことからDSnに属する各PDS(PDS1 ~ PDSlast)のPTS値は、以下の関係を満たす値に、設定されねばならない。

30

$DTS(DSn[PCS]) \leq PTS(DSn[PDS1])$

$PTS(DSn[PDSj]) \leq PTS(DSn[PDSj+1]) \leq PTS(DSn[PDSlast])$

$PTS(DSn[PDSlast]) \leq DTS(DSn[ODS1])$

尚、PDSにおいてDTSは再生時に参照されないが、MPEG2規格を満たすため、PDSのDTSは、そのPTSと同じ値に設定される。

以上の関係を満たすようDTS、PDSが設定された場合、再生装置のパイプラインにおいてこれらDTS、PTSがどのような役割をもつかについて説明する。図 3 9 はPDSにおけるPTSに基づく、再生装置におけるパイプラインを示す図である。この図 3 9 は、図 2 8 をベースにして作図されている。図 3 9 において第 1 段目は、CLUT部 9 へのPDS設定を示している。以降の段は、図 2 8 の第 1 ~ 第 5 段目と同じである。PDS1 ~ lastのCLUT部 9 への設定は、ODS1のデコード開始と同時になされる(矢印up2,up3)。

40

#### 【 0 1 4 4 】

続いてDSnに属するEND of Display SetSegmentのPTSの設定について説明する。DSnに属するENDは、DSnの終わりを示すものだから、ODS2のデコード終了時刻を示せばよい。このデコード終了時刻は、ODS2(ODSlast)のPTS( $PTS(DSn[ODSlast])$ )に示されているので、ENDのPTSは、以下の式に示される値に設定されねばならない。

$PTS(DSn[END]) = PTS(DSn[ODSlast])$

50

DSn, DSn+1に属するPCSとの関係で考えれば、DSnにおけるPCSは、最初のODS(ODS1)のロード時刻以前に、Composition Buffer 16にロードされるから、ENDのPTSは、DSnに属するPCSのロード時刻(DTS(DSn[PCS]))以降、DSn+1に属するPCSのロード時刻(DTS(DSn+1[PCS]))以前でなければならない。そのためENDのPTSは、以下の式の間を満たす必要がある。

$$DTS(DSn[PCS]) \leq PTS(DSn[END]) \leq DTS(DSn+1[PCS])$$

一方、最初のODS(ODS1)のロード時刻は、最後のPDS(PDSlast)のロード時刻以前であるから、ENDのPTS(PTS(DSn[END]))は、DSnに属するPDSのロード時刻以降(PTS(DSn[PDSlast]))でなければならない。そのためENDのPTSは、以下の式の間を満たす必要がある。

$$PTS(DSn[PDSlast]) \leq PTS(DSn[END])$$

10

続いて再生装置のパイプラインにおいて、ENDのPTSが、どのような意味合いをなすのかについて説明する。図40は、再生装置のパイプライン動作時における、ENDの意味合いを示す図である。本図は、図28の第2段目以降をベースに作図しており、第2段以降の意味合いは図28と同一である。また本図では、DSn, DSn+1という2つのDisplay Setを描いている。DSnにおいてODSlastになるのは、A-ODSsの最後のODSnであるので、ENDのPTSは、このODSnのPTSを示すよう設定されている。そして、このENDのPTSに示される時点は、DSn+1のPCSのDTSにより示される時点より早いものになっている。

#### 【0145】

このENDのPTSにより、再生時にあたっては、DSnについてのODSのロードが、どの時点で完了するのかを知得することができる。

20

尚、ENDにおいてDTSは再生時に参照されないが、MPEG2規格を満たすため、PDSのDTSは、そのPTSと同じ値に設定される。

以上のように本実施形態によれば、グラフィックスプレーンの一部をグラフィックスの表示のためのウィンドウとして指定するので、再生装置は、プレーン全体のグラフィックス描画を行う必要はない。グラフィックスプレーンの25%~33%など、ある限られた大きさのウィンドウに対してのみ、グラフィックス描画を行えばよい。グラフィックスプレーンのうち、ウィンドウ以外の部分の描画を省くことができるので、再生装置側のソフトウェアの負担は遥かに軽くなる。

#### 【0146】

グラフィックスプレーンの1/4等、グラフィックスのアップデートがワーストケースで行われる場合であっても、ピクチャとの同期が保証されるような大きさにウィンドウを定めておけば、再生装置は、256Mbps等、ある決まった転送レートでグラフィックスプレーンの書き込みを行えば、ピクチャとの同期を実現することができる。

30

同期保証が容易になるので、高い解像度での字幕表示を多くの再生装置において実現することができる。

#### 【0147】

##### (第2実施形態)

第1実施形態は、専ら、字幕として表示されるためのグラフィックスについて説明した。これに対し第2実施形態は、対話的に表示されるグラフィックスについての実施形態とする。

本発明に係る記録媒体の実施行為のうち、生産行為についての形態について説明する。第2実施形態に係る記録媒体も、第1実施形態同様、BD-ROMの応用層に対する改良により実施することができる。図41は、第2実施形態に係るAVClipがどのように構成されているかを模式的に示す図である。

40

#### 【0148】

AVClipは(中段)、複数のビデオフレーム(ピクチャpj1,2,3)からなるビデオストリーム、複数のオーディオフレームからなるオーディオストリームを(上1段目)、PESパケット列に変換し(上2段目)、更にTSパケットに変換し(上3段目)、同じくインタラクティブグラフィックスストリーム(下1段目)を、PESパケット列に変換し(下2段目)、更にTSパケットに変換して(下3段目)、これらを多重化することで構成される。

#### 【0149】

50

続いてインタラクティブグラフィクスストリームについて説明する。インタラクティブグラフィクスストリームには、PCSの代わりにICS(Interactive Composition Segment)があり、WDSは存在しない。PDS(Palette Definition Segment)、ODS(Object\_Definition\_Segment)、END(END of Display Set Segment)と呼ばれる機能セグメントが存在する点は、プレゼンテーショングラフィクスストリームと同じである。

#### 【 0 1 5 0 】

これらの機能セグメントにより定義される対話画面は、画面上にGUI部品を配置することにより作成される対話画面である。図 4 2 ( a ) は、インタラクティブグラフィクスストリームにより実現される対話画面を示す図である。この対話画面は、ボタンA～ボタンDという4つのGUI部品を含む。インタラクティブグラフィクスストリームによる対話性とは、これらのGUI部品(ボタン)の状態をユーザ操作に応じて変化させることをいう。GUI部品(ボタン)における状態には、図 4 2 ( a ) に示すノーマル状態bt1、セレクトッド状態bt2、アクティブ状態bt3といったものがある。ノーマル状態とは、単に表示されているに過ぎない状態である。これに対しセレクトッド状態とは、ユーザ操作によりフォーカスが当てられているが、確定に至っていない状態をいう。アクティブ状態とは、確定に至った状態をいう。これらのボタンの状態は、第1実施形態に示したリモコン400のキーの押下により、変化させることができる。

#### 【 0 1 5 1 】

図 4 2 ( b ) は、対話画面に対する操作をユーザから受け付けるためのリモコン400におけるキーを示す図である。本図に示すようにリモコン400は、MoveUpキー、MoveDownキー、MoveRightキー、MoveLeftキーが設けられている。

MoveUpキーは、対話画面においてあるボタンがセレクトッド状態である場合、このボタンより上にあるボタンをセレクトッド状態に設定するためのキーである。MoveDownキーは、このボタンより下にあるボタンをセレクトッド状態に設定するためのキー、MoveRightキーは、このボタンより右にあるボタンをセレクトッド状態に設定するためのキー、MoveLeftキーは、このボタンより左にあるボタンをセレクトッド状態に設定するためのキーである。

#### 【 0 1 5 2 】

Activatedキーは、セレクトッド状態にあるボタンをアクティブ状態(アクティベート)するためのキーである。「0」～「9」の数値キーは、該当する数値が割り当てられたボタンをセレクトッド状態にするキーである。「+10」キーとは、これまで入力された数値に10をプラスするという操作を受け付けるキーである。尚、「0」キー、「+10」キーは、何れも2桁以上の数値の入力を受け付けるものなので、「0」キー、「+10」キーは、どちらかが具備されていればよい。

#### 【 0 1 5 3 】

ノーマル状態、セレクトッド状態、アクティブ状態といった各状態は、複数の非圧縮状態のグラフィクスから構成される。ボタンの各状態を表現する個々の非圧縮グラフィクスを「グラフィクスオブジェクト」という。あるボタンの1つの状態を、複数の非圧縮グラフィクスで表現しているのは、各ボタンの1つの状態をアニメーション表示することを念頭に置いているからである。

#### 【 0 1 5 4 】

以上が本実施形態が想定している対話画面についての説明である。続いて本実施形態におけるDefinition Segment(ODS、PDS)の改良について説明する。ODS、PDSは、第1実施形態と同様のデータ構造を有している。差違は、ODSについての『object\_ID』にある。第2実施形態におけるODSは、複数ODSにより定義される複数グラフィックスオブジェクトを用いてアニメーションを構成する。そしてアニメーションを構成するにあたって、一連のODSに、連番たる『object\_ID』を付加する。以上がODS、PDSについての説明である。

#### 【 0 1 5 5 】

続いてICSについて説明する。Interactive Composition Segmentは、対話的な画面を構成する機能セグメントである。Interactive Composition Segmentは、図 4 3 に示すデー

10

20

30

40

50



タ構造で構成される。本図に示すようにICSは、『segment\_type』と、『segment\_length』と、『composition\_number』と、『composition\_state』と、『command\_update\_flag』と、『Composition\_timeout\_PTS』と、『Selection\_timeout\_PTS』と、『UO\_Mask\_Table』と、『animation\_frame\_rate\_code』と、『default\_selected\_button\_number』と、『default\_activated\_button\_number』と、『ボタン情報群(button info(1)(2)(3)・・・)』とからなる。

#### 【0156】

『Composition\_Number』は、ICSが属するDSにおいて、Updateがなされることを示す0から15までの数値である。

『composition\_state』は、本ICSから始まるDSが、Normal Caseであるか、Acquisition Pointであるか、Epoch Startであるかを示す。 10

『command\_update\_flag』は、本ICS内のボタンコマンドは、前のICSから変化しているかを否かを示す。例えば、あるICSが属するDSが、Acquisition Pointであれば、このICSは、原則1つ前のICSと同じ内容になる。しかしcommand\_update\_flagをオンに設定しておけば、1つ前のICSと違うボタンコマンドをICSに設定しておくことができる。本フラグは、グラフィックスオブジェクトは流用するが、コマンドは変更したい場合に有効となる。

#### 【0157】

『Composition\_timeout\_PTS』は、ボタンによる対話画面の終了時刻を記述する。終了時刻において対話画面の表示は、もはや有効ではなく表示されない。Composition\_timeout\_PTSは、動画データにおける再生時間軸の時間精度で記述しておくことが好ましい。 20

『Selection\_Time\_out\_PTS』は、有効なボタン選択期間の終了時点を記述する。Selection\_Time\_out\_PTSの時点において、Default\_activated\_button\_numberにより特定されるボタンがアクティベートされる。Selection\_Time\_out\_PTSは、Composition\_timeout\_PTSの時間と等しいかそれより短い。Selection\_Time\_out\_PTSはビデオフレームの時間精度で記述される。

#### 【0158】

『UO\_Mask\_Table』は、ICSに対応するDisplay Setにおけるユーザ操作の許可/不許可を示す。このマスクフィールドが不許可に設定されていれば、再生装置に対するユーザ操作は無効になる。

『animation\_frame\_rate\_code』は、アニメーション型ボタンに適用すべきフレームレートを記述する。アニメーションフレームレートは、本フィールドの値を用いて、ビデオフレームレートを割ることにより与えられる。本フィールドが00なら、各ボタンのグラフィックスオブジェクトを定義するODSのうち、Start\_Object\_id\_xxxにて特定されるもののみが表示され、アニメーションされない。 30

#### 【0159】

『default\_selected\_button\_number』は、対話画面の表示が始まったとき、デフォルトとしてセレクトッド状態に設定すべきボタン番号を指示する。本フィールドが"0"であれば、再生装置のレジスタに格納されたボタン番号のボタンが自動的にアクティブ状態に設定される。もしこのフィールドが非ゼロであれば、このフィールドは、有効なボタンの値を意味する。 40

#### 【0160】

『default\_activated\_button\_number』は、Selection\_Time\_out\_PTSにより定義された時間の前に、ユーザがどのボタンもアクティブ状態にしなかったとき、自動的にアクティブ状態に設定されるボタンを示す。default\_activated\_button\_numberが"FF"であれば、Selection\_Time\_out\_PTSにより定義される時刻において、現在セレクトッド状態にあるボタンが自動的に選択される。このdefault\_activated\_button\_numberが00であれば、自動選択はなされない。00,FF以外の値であれば本フィールドは、有効なボタン番号として解釈される。

#### 【0161】

『ボタン情報(Button\_info)』は、対話画面において合成される各ボタンを定義する情 50

報である。図中の引き出し線hp1はICSにより制御されるi番目のボタンについてのボタン情報iの内部構成をクローズアップしている。以降、ボタン情報iを構成する情報要素について説明する。

『button\_number』は、ボタンiを、ICSにおいて一意に識別する数値である。

【0162】

『numerically\_selectable\_flag』は、ボタンiの数値選択を許可するか否かを示すフラグである。

『auto\_action\_flag』は、ボタンiを自動的にアクティブ状態にするかどうかを示す。auto\_action\_flagがオン(ビット値1)に設定されれば、ボタンiは、セレクト状態になる代わりにアクティブ状態になる。auto\_action\_flagがオフ(ビット値0)に設定されれば、ボタンiは、選択されたとしてもセレクト状態になるにすぎない。

10

【0163】

『object\_horizontal\_position』、『object\_vertical\_position』は、対話画面におけるボタンiの左上画素の水平位置、垂直位置を示す。

『upper\_button\_number』は、ボタンiがセレクト状態である場合においてMOVEUPキーが押下された場合、ボタンiの代わりに、セレクト状態にすべきボタンの番号を示す。もしこのフィールドにボタンiの番号が設定されていれば、MOVEUPキーの押下は無視される。

【0164】

『lower\_button\_number』、『left\_button\_number』、『right\_button\_number』は、ボタンiがセレクト状態である場合においてMOVE Down キー,MOVE Left キー,MOVE Right キーが押下された場合、ボタンiの押下の代わりに、セレクト状態にすべきボタンの番号を示す。もしこのフィールドにボタンiの番号が設定されていれば、これらのキーの押下は無視される。

20

【0165】

『start\_object\_id\_normal』は、ノーマル状態のボタンiをアニメーションで描画する場合、アニメーションを構成する複数ODSに付加された連番のうち、最初の番号がこのstart\_object\_id\_normalに記述される。

『end\_object\_id\_normal』は、ノーマル状態のボタンiをアニメーションで描画する場合、アニメーションを構成する複数ODSに付加された連番たる『object\_ID』のうち、最後の番号がこのend\_object\_id\_normalに記述される。このEnd\_object\_id\_normalに示されるIDが、start\_object\_id\_normalに示されるIDと同じである場合、このIDにて示されるグラフィックスオブジェクトの静止画が、ボタンiの絵柄になる。

30

【0166】

『repeated\_normal\_flag』は、ノーマル状態にあるボタンiのアニメーション表示を反復継続させるかどうかを示す。

『start\_object\_id\_selected』は、セレクト状態のボタンiをアニメーションで描画する場合、アニメーションを構成する複数ODSに付加された連番のうち、最初の番号がこのstart\_object\_id\_selectedに記述される。

【0167】

『end\_object\_id\_selected』は、セレクト状態のボタンをアニメーションで描画する場合、アニメーションを構成する複数ODSに付加された連番たる『object\_ID』のうち、最後の番号がこのend\_object\_id\_selectedに記述される。

40

このEnd\_object\_id\_selectedに示されるIDが、start\_object\_id\_selectedに示されるIDと同じである場合、このIDにて示されるグラフィックスオブジェクトの静止画が、ボタンiの絵柄になる。

【0168】

『repeat\_selected\_flag』は、セレクト状態にあるボタンiのアニメーション表示を、反復継続するかどうかを示す。start\_object\_id\_selectedと、end\_object\_id\_selectedとが同じ値になるなら、本フィールド00に設定される。

50

『start\_object\_id\_activated』は、アクティブ状態のボタンiをアニメーションで描画する場合、アニメーションを構成する複数ODSに付加された連番のうち、最初の番号がこのstart\_object\_id\_activatedに記述される。

#### 【 0 1 6 9 】

『end\_object\_id\_activated』は、アクティブ状態のボタンをアニメーションで描画する場合、アニメーションを構成する複数ODSに付加された連番たる『object\_ID』のうち、最後の番号がこのend\_object\_id\_activatedに記述される。

続いてボタンコマンドについて説明する。

『ボタンコマンド(button\_command)』は、ボタンiがアクティブ状態になれば、実行されるコマンドである。

#### 【 0 1 7 0 】

以上がICSの内部構成である。ICSによる対話制御の具体例について以下説明する。本具体例は、図4-4のようなODS、ICSを想定している。図4-4は、あるDSnに含まれるODSと、ICSとの関係を示す図である。このDSnには、ODS11~19,21~29,31~39,41~49が含まれているものとする。これらのODSのうち、ODS11~19は、ボタンAの各状態を描いたものであり、ODS21~29は、ボタンBの各状態を描いたもの、ODS31~39は、ボタンCの各状態を描いたもの、ODS41~49は、ボタンDの各状態を描いたものとする(図中の括弧)を参照)。そしてICSにおけるbutton\_info(1),(2),(3),(4)にて、これらのボタンA~ボタンDの状態制御が記述されているものとする(図中の矢印bh1,2,3,4参照)。

#### 【 0 1 7 1 】

このICSによる制御の実行タイミングが、図4-5に示す動画のうち、任意のピクチャデータpt1の表示タイミングであれば、ボタンA~ボタンDからなる対話画面tm1が、このピクチャデータpt1に合成(gs1)されて表示されることになる(gs2)。動画の中身に併せて、複数ボタンからなる対話画面が表示されるので、ICSによりボタンを用いたリアルな演出が可能になる。

#### 【 0 1 7 2 】

図4-7に示すボタンA~ボタンDの状態遷移を実行する場合のICSの記述例を図4-6に示す。図4-7における矢印hh1,hh2は、button\_info(1)のneighbor\_info()による状態遷移を象徴的に表現している。button\_info(1)のneighbor\_info()におけるlower\_button\_numberは、ボタンCに設定されているため、ボタンAがセレクト状態になっている状態で、MOVEDownキー押下のU0が発生すれば(図4-7のup1)、ボタンCがセレクト状態になる(図4-7のsj1)。button\_info(1)のneighbor\_info()におけるright\_button\_numberは、ボタンBに設定されているため、ボタンAがセレクト状態になっている状態で、MOVERightキー押下のU0が発生すれば(図4-7のup2)、ボタンBがセレクト状態になる(図4-7のsj2)。

#### 【 0 1 7 3 】

図4-7における矢印hh3は、button\_info(3)のneighbor\_info()による状態遷移の制御を示す。button\_info(3)のneighbor\_info()におけるupper\_button\_numberは、ボタンAに設定されているため、ボタンCがセレクト状態になっている状態で(up3)、MOVEUpキー押下のU0が発生すれば、ボタンAがセレクト状態に戻る。

続いてボタンA~ボタンDの絵柄について説明する。ODS11,21,31,41が図4-8に示す絵柄であるものとする。そしてボタンAに割り当てられたODS11~19は、図4-9のような絵柄であるものとする。ICSにおけるbutton\_info(1)のnormal\_state\_info()におけるstart\_object\_id\_normal,end\_object\_id\_normalは、ODS11~13を指定しているため、ボタンAのノーマル状態は、ODS11~13によるアニメーションで表現される。またbutton\_info(1)のselected\_state\_info()におけるstart\_object\_id\_selected,end\_object\_id\_selectedは、ODS14~16を指定しているため、ボタンAのセレクト状態は、ODS14~16で表現される。ユーザがこのボタンAをセレクト状態にすることにより、ボタンAの絵柄たる肖像は、ODS11~13によるものから、ODS14~16によるものへと変化する。ここでnormal\_state\_info()、selected\_state\_info()におけるrepeat\_normal\_flag,repeat\_select\_flagを1にしてお

10

20

30

40

50

けば、ODS11～13によるアニメーション、ODS14～16によるアニメーションは、図中の「(A)」、「(A)」、「(B)」、「(B)」,に示すように、アニメーション表示は反復継続する。

#### 【0174】

アニメーション描画が可能な複数ODSが、ボタンA～ボタンDに割り当てられており、これらによる制御がICSに記述されていれば、ユーザ操作に併せてキャラクタの表情が変わるような、リアルなボタンの状態制御を実現することができる。

続いてDisplay SetにおけるODSの順序について説明する。Display Setに属するODSは、ボタンの1つの状態を表すようICSにて指定されていることは、上述した通りである。ODSは、こうした指定、つまり、ボタンのどの状態を示すかという指定に応じて、Display Setにおける順序が決められる。

#### 【0175】

詳しくいうとDisplay SetにおいてODSは、ノーマル状態を表すもの(1)、セレクトッド状態を表すもの(2)、アクティブ状態を示すもの(3)というように、同じ状態を表すもの同士がグループ化される。このボタンの1つの状態を表すグループをbutton-stateグループという。そしてこれらbutton-stateグループを、ノーマル状態 セレクトッド状態 アクティブ状態というように並べる。このようにボタンのどの状態を表すかに応じて、ODSの順序を決めるとというのが、Display SetにおけるODSの順序である。

#### 【0176】

図50は、Display Setに属するODSの順序を示す図である。本図の第2段目に、Display Setにおける3つのbutton-stateグループを示す。本図においてノーマル状態を描くODSの集合(ODSs for Normal state)、ボタンのセレクトッド状態を描くODSの集合(ODSs for Selected state)、ボタンのアクティブ状態を描くODSの集合(ODSs for Actioned state)が示されている。そしてこれらbutton-stateグループの順序は、ノーマル状態 セレクトッド状態 アクティブ状態というように並べられている。これはODSのうち、対話画面の初期表示を構成するものを早く読み出させ、アップデート後の画面表示を構成するものの読み出しを後にするという配慮である。

#### 【0177】

図50の第1段目は、これらbutton-stateグループにより描かれるグラフィクスオブジェクトAn,Bn,Cn,Dn,As,Bs,Cs,Ds,Aa,Ba,Ca,Daを示す。本図におけるAn,Bn,Cn,Dnにおける添字nは各ボタンのノーマル状態を表し、As,Bs,Cs,Dsにおける添字sは各ボタンのセレクトッド状態を表す。Aa,Ba,Ca,Daにおける添字aは各ボタンのアクティブ状態を表す。図50の第2段目は、第1段目のグラフィクスオブジェクトが属するbutton-stateグループを示す。尚、本図におけるODS1～ODSnという表記は、「1」、「n」というような同じ番号が付されているが、これらN-ODSs,S-ODSs,A-ODSsに属するODSは別々のものである。以降、同様の表記の図は同じ意味であるとする。

#### 【0178】

図51は、図50のbutton-stateグループが配置された対話画面における状態遷移を示す図である。

本図における対話画面は、「初期表示」、「1stユーザアクションによる更新表示」、「2ndユーザアクションによる更新表示」という複数の状態をもつ。図中の矢印は、状態遷移のトリガとなるユーザアクションを表す。この図を参照すると、4つのボタンA,B,C,Dはそれぞれノーマル状態、セレクトッド状態、アクティブ状態という状態をもっている。このうち初期表示に必要なのは、3つのノーマル状態を描くグラフィクスオブジェクトと、1つのセレクトッド状態を描くグラフィクスオブジェクトであることがわかる。

#### 【0179】

デフォルトセレクトッドボタンが未確定であり、ボタンA～ボタンDのうち、どのボタンがセレクトッド状態になるかが動的に変わる場合であっても、各ボタンのノーマル状態、セレクトッド状態を表すグラフィクスオブジェクトのデコードが完了すれば、初期表示を実現することができる。このことを意識して、本実施形態では、各状態に対応するbutton

-stateグループを、図50の第2段目に示すようにノーマル状態 セレクトッド状態 アクティブ状態の順に配列している。かかる配列により、アクティブ状態を構成するODSの読み出しやデコードが未完であっても、初期表示を実現することができ、Display Setの読み出し開始から初期表示の完了までの期間を短くすることができる。

#### 【0180】

続いて図48、図49に示したODSを、どのような順序で配列させるかについて説明する。図52は、Display SetにおけるODSの順序を示す図である。本図においてODSs for Normal stateは、ODS11~13, ODS21~23, ODS31~33, ODS41~43から構成されている。またODSs for Selected stateは、ODS14~16, ODS24~26, ODS34~36, ODS44~46から構成され、ODSs for Actioned stateは、ODS17~19, ODS27~29, ODS37~39, ODS47~49から構成されている。ODS11~13は、図49に示したような、キャラクターの表情変化を描くものであり、ODS21~23, ODS31~33, ODS41~43も同様なので、これらのODSを先頭のbutton-stateグループに配置することにより、Display Setの読み出しの途中であっても、初期表示の準備を整えることができる。これによりアニメーションを取り入れた対話画面を、遅滞なく実行することができる。

10

#### 【0181】

続いて複数のボタン状態からの多重参照されるODSの順序について説明する。多重参照とは、あるODSについてのobject\_idがICSにおける2以上のnormal\_state\_info, selected\_state\_info, activated\_state\_infoにより指定されていることをいう。かかる多重参照を行えば、あるボタンのノーマル状態を描くグラフィクスオブジェクトを用いて、他のボタンのセレクトッド状態を描くことができ、グラフィクスオブジェクトの絵柄を共用することができる。かかる共用により、ODSの数を少なくすることができる。多重参照されるODSについては、どのbutton-stateグループに属するかが問題になる。

20

#### 【0182】

つまりあるボタンのノーマル状態と、別のボタンのセレクトッド状態とが1つのODSで描かれている場合、このODSは、ノーマル状態に対応するbutton-stateグループに属するか、セレクトッド状態に対応するbutton-stateグループに属するかが問題となる。

この場合ODSは、複数状態のうち、最も早く出現する状態に対応するbutton-stateグループだけ1回のみ配置される。

#### 【0183】

30

あるODSがノーマル状態、セレクトッド状態で多重参照されるなら、ノーマル状態に対応するbutton-stateグループ(N-ODSs)にこのODSは配置され、セレクトッド状態に対応するbutton-stateグループ(S-ODSs)には配置されない。また別のODSがセレクトッド状態、アクティブ状態で多重参照されるなら、セレクトッド状態に対応するbutton-stateグループ(S-ODSs)にこのODSは配置され、アクティブ状態に対応するbutton-stateグループ(A-ODSs)には配置されない。このように多重参照されるODSは、最も早く出現する状態に対応するbutton-stateグループ内に一回だけ配置される。以上が多重参照されるODSの順序についての説明である。

#### 【0184】

S-ODSsにおける、ODSの順序について説明する。S-ODSsにおいて、どのODSが先頭に位置するかは、デフォルトセレクトッドボタンが静的に確定しているか、動的であるかによって違う。静的に確定したデフォルトセレクトッドボタンとは、ICSにおけるdefault\_selected\_button\_numberに00以外の有効な値が設定され、この値で指示されるボタンのことをいう。default\_selected\_button\_numberが有効な値を示しており、尚且つデフォルトセレクトッドボタンを表すODSが、N-ODSsに無い場合は、デフォルトセレクトッドボタンを表すODSが、S-ODSsの先頭に配置される。

40

#### 【0185】

default\_selected\_button\_numberが値00を示している場合、デフォルトでセレクトッド状態に設定されるボタンは、再生装置側の状態によって動的に変化する。

値0を示すよう、default\_selected\_button\_numberを設定しておくのは、例えば、Displ

50

ay Setが多重されているAVClipが、複数再生経路の合流点になっているようなケースである。先行する複数再生経路がそれぞれ第1、第2、第3章であり、合流点にあたるDisplay Setが第1章、第2章、第3章に対応するボタンを表示させるものである場合、default\_selected\_button\_numberにおいて、デフォルトでセレクト状態とすべきボタンを決めてしまうのは、おかしい。

#### 【0186】

第1章からの到達時には第2章にあたるボタン、第2章からの到達時には第3章にあたるボタン、第3章からの到達時には第4章にあたるボタンというように、このDisplay Setに到達するまでに、先行する複数再生経路のうち、どの再生経路を経由するかによって、セレクト状態とすべきボタンを変化させるのが理想的である。先行する再生経路によって、セレクト状態とすべきボタンが変わるようなケースにおいて、default\_selected\_button\_numberは無効を示すよう、0に設定される。どの再生経路を経由するかによって、セレクト状態とすべきボタンを変化するから、特定のODSをbutton-stateグループの先頭に配置するというような配慮は行わない。

#### 【0187】

図53は、default\_selected\_button\_numberが"=0"である場合と、"=ボタンB"である場合とでS-ODSsにおいてODSの並びがどのように変わるかを示す図である。本図において破線ss1は、default\_selected\_button\_numberがボタンBを示している場合に、S-ODSsにおけるODSの配列がどのようになるかを示しており、破線ss2は、default\_selected\_button\_numberが=0を示している場合に、S-ODSsにおけるODSの配列がどのようになるかを示している。この図の表記からもわかるように、default\_selected\_button\_numberがボタンBを示している場合、ボタンBのセレクト状態を示すODSBsがS-ODSsの先頭に配され、その他のボタンのODSは、後回しにされている。一方、default\_selected\_button\_numberが"=0"である場合、ボタンAのセレクト状態を表すODSAsが先頭に配置されている。このようにdefault\_selected\_button\_numberが有効かどうかは、S-ODSs内の順序に大きな変動をもたらす。

#### 【0188】

以上がODSの順序についての説明である。続いてこれらICS、ODSを有したDisplay Setが、AVClipの再生時間軸上にどのように割り当てられるかについて説明する。ODSについてのDTS、PTSは、第1実施形態に示した数式に基づき設定すればよい。ICSにおけるDTS、PTSは、第1実施形態に示したものと違ったものになる。以降ICSの、DTS値、PTS値について説明する。

#### 【0189】

ICSのPTSは、Epoch開始直後であればDSnの初期表示を構成するODSのうち、デコード時刻が最も遅いODSのPTS値(1)、グラフィクスプレーンのクリアに要する時間(2)、ODSのデコードにより得られたグラフィクスオブジェクトをグラフィクスプレーンに書き込む書込時間(3)を足した値以降に設定される。一方Acquisition Pointであれば、ODSのPTS値(1)にプレーン書込期間(3)を足した値(ODSのPTS値(1)+プレーン書込期間(3))以降に設定される。

#### 【0190】

ICSにおいてdefault\_selected\_button\_numberが指定されている場合は、全てのボタンのノーマル状態を描画するODSのデコードと、デフォルトボタンのセレクト状態を描画するODSのデコードさえ完了すれば、初期表示を行うことができる。初期表示における複数ボタンのセレクト状態を描画するODSを、S-ODSsと呼び、そのうちデコード時刻が最も早いもの(この場合、デフォルトボタンを描画するもの)をS-ODSsfirstと呼ぶ。このS-ODSsfirstのPTS値を、デコード時刻が最も遅いODSのPTS値として、ICSのPTSの基準に用いる。

#### 【0191】

ICSにおいてdefault\_selected\_button\_numberが指定されていない場合は、どのボタンがセレクト状態になるかわからないから、全ボタンのノーマル状態、セレクト状態

10

20

30

40

50

態を描画する準備が整なわないと、初期表示の準備が完了しない。初期表示における複数ボタンのセレクト状態を描画するS-ODSsのうち、デコード時刻が最も遅いものをS-ODSs<sub>last</sub>と呼ぶ。このS-ODSs<sub>last</sub>のPTS値を、デコード時刻が最も遅いODSのPTS値として、ICSのPTSの基準値に用いる。

#### 【0192】

S-ODSs<sub>first</sub>のデコード終了時刻をPTS(DSn[S-ODSs<sub>first</sub>])とすると、PTS(DSn[ICS])は、PTS(DSn[S-ODSs<sub>first</sub>])に、グラフィックスプレーンのクリアに要する時間(2)、ODSのデコードにより得られたグラフィックスオブジェクトをグラフィックスプレーンに書き込む書込時間(3)を足した値になる。

グラフィックスプレーン内において描画可能な矩形領域の横幅をvideo\_width、縦幅をvideo\_heightとし、グラフィックスプレーンへの書込レートを128Mbpsとすると、グラフィックスプレーンのクリアに要する時間は、 $8 \times \text{video\_width} \times \text{video\_height} // 128,000,000$ と表現される。これを90KHzの時間精度で表現すれば、グラフィックスプレーンのクリア時間(2)は $90,000 \times (8 \times \text{video\_width} \times \text{video\_height} // 128,000,000)$ になる。

#### 【0193】

ICSに含まれる全ボタン情報により、指定されるグラフィックスオブジェクトの総サイズをSIZE(DSn[ICS.BUTTON[i]])とし、グラフィックスプレーンへの書込レートを128Mbpsとすると、グラフィックスプレーンへの書き込みに要する時間は、 $\text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) // 128,000,000$ と表現される。これを90KHzの時間精度で表現すれば、グラフィックスプレーンのクリア時間(2)は $90,000 \times (\text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) // 128,000,000)$ になる。

#### 【0194】

ここでSIZE(DSn[ICS.BUTTON[i]])は、各ボタンを表すグラフィックスオブジェクトのうち、最初に表示されるもののサイズの総和を意味する。このSIZE(DSn[ICS.BUTTON[i]])はデフォルトセレクトボタンが確定している場合と、動的に変わる場合とで、違う値になる。デフォルトセレクトボタンが静的に確定している場合、SIZE(DSn[ICS.BUTTON[i]])は、デフォルトセレクトボタンのセレクト状態を表す複数ODSのうち最初に表示されるもの、デフォルトセレクトボタン以外のボタンのノーマル状態を表す複数ODSのうち、最初に表示されるものの総和になる。

#### 【0195】

続いてデフォルトセレクトボタンが動的に変わる場合、どのボタンがデフォルトセレクトボタンになるかはわからないから、書き込み時間が最も長くなるケースを想定しせねばならない。ここで、任意のボタンxのノーマル状態における最初の一枚を表すグラフィックスオブジェクト(ODSn1)、及び、ボタンxのセレクト状態における最初の一枚を表すグラフィックスオブジェクト(ODSs1)のうち、サイズが大きいもの(Max(ODSn1,ODSs1))を、1つのボタンxにおいて最初に表示すべきグラフィックスオブジェクトと考える。

#### 【0196】

このMax(ODSn1,ODSs1)を、全てのボタンについて足し合わせた結果が、SIZE(DSn[ICS.BUTTON[i]])になる。

図54(a)(b)は、N-ODSsにボタンA~Dを構成する複数ODSが含まれており、S-ODSsにボタンA~Dを構成する複数ODSが含まれている場合、SIZE(DSn[ICS.BUTTON[i]])がどのような値になるかを示す図である。ここでdefault\_selected\_button\_numberが有効な値を示している場合、SIZE(DSn[ICS.BUTTON[i]])は太い枠で示す4つのODSのサイズの総和となる。"As1"は、ボタンAのセレクト状態を表す複数のODSのうち最初に表示されるODSである。"Bn1"、"Cn1"、"Dn1"は、ボタンB~ボタンDのノーマル状態を表す複数のODSのうち最初に表示されるODSを示す。これらのサイズをsize()で表すと、SIZE(DSn[ICS.BUTTON[i]])は、size(As1)+size(Bn1)+size(Cn1)+size(Dn1)になる。

#### 【0197】

一方、default\_selected\_button\_numberが"=0"であるなら、An1,As1のうち大きいODS、Bn1,Bs1のうち大きいODS、Cn1,Cs1のうち大きいODS、Dn1,Ds1のうち大きいODSの和が

10

20

30

40

50

SIZE(DSn[ICS.BUTTON[i]])になる。

故に SIZE(DSn[ICS.BUTTON[i]])は、

$$\begin{aligned} & \text{SIZE(DSn[ICS.BUTTON[i]])} \\ &= \max(\text{size(An1)}, \text{size(As1)}) + \max(\text{size(Bn1)}, \text{size(Bs1)}) \\ & \quad + \max(\text{size(Cn1)}, \text{size(Cs1)}) + \max(\text{size(Dn1)}, \text{size(Ds1)}) \end{aligned}$$

になる。

以上の数式を用いることにより、Epoch Start開始直後のPTS(DSn[ICS])は、以下の数式のように表現される。

$$\begin{aligned} & \text{PTS(DSn[ICS])} \quad \text{PTS(DSn[S-ODSsfirst])} \\ & \quad + 90,000 \times (8 \times \text{video\_width} \times \text{video\_height} // 128,000,000) \\ & \quad + 90,000 \times (\text{SIZE(DSn[ICS.BUTTON[i]])} // 128,000,000) \end{aligned} \quad 10$$

一方、default\_selected\_button\_numberが無効である場合、PTS(DSn[S-ODSsfirst])をPTS(DSn[S-ODSslast])に置き換えればよい。つまり算出式は、以下の通りになる。

$$\begin{aligned} & \text{PTS(DSn[ICS])} \quad \text{PTS(DSn[S-ODSslast])} \\ & \quad + 90,000 \times (8 \times \text{video\_width} \times \text{video\_height} // 128,000,000) \\ & \quad + 90,000 \times (\text{SIZE(DSn[ICS.BUTTON[i]])} // 128,000,000) \end{aligned}$$

以上のようにしてPTS、DTSを設定することにより、同期表示を実現する場合の一例を図55に示す。本図において動画における任意のピクチャデータpy1の表示タイミングで、ボタンを表示させる場合を想定する。この場合、ICSのPTS値は、このピクチャデータの表示時点になるよう設定せねばならない。

【0198】

そしてICSのPTSから、画面のクリア期間cd1、グラフィクスオブジェクトの転送期間td1を差し引いた時刻に、DSnの初期表示を構成するODSのうち、デコード時刻が最も遅いODSのデコードが完了せねばならないから、図中の時点(1)に、ODSのPTS値が設定しなければならない。更に、ODSのデコードには期間dd1を要するから、このPTSより期間dd1だけ早い時点に、このODSのDTS値を設定せねばならない。

【0199】

図55において、動画と合成されるODSは1つだけであり、単純化されたケースを想定している。動画と合成されるべき対話画面の初期表示が、複数のODSで実現される場合、ICSのPTS及びDTS、ODSのPTS、DTSは図56のように設定せねばならない。

図56は、対話画面の初期表示が複数ODSにて構成され、デフォルトセレクトッドボタンが静的に確定している場合のDTS、PTSの設定を示す図である。初期表示を実現するODSのうち、デコードが最も遅いS-ODSsfirstのデコードが図中の期間dd1の経過時に終了するなら、このS-ODSsfirstのPTS(DSn[S-ODSsfirst])は、期間dd1の経過時を示すよう設定される。

【0200】

更に、初期表示の実現には、画面クリアを行い、デコードされたグラフィクスオブジェクトを転送せねばならないから、このPTS(DSn[S-ODSsfirst])の値に画面クリアに要する期間(90,000 × (8 × video\_width × video\_height // 128,000,000))、デコードされたグラフィクスオブジェクトの転送期間(90,000 × (SIZE(DSn[ICS.BUTTON[i]]) // 128,000,000))を足した時点以降を、ICSのPTS(DSn[ICS])として設定せねばならない。

【0201】

図57は、対話画面の初期表示が複数ODSにて構成され、デフォルトセレクトッドボタンが未定である場合のDTS、PTSの設定を示す図である。初期表示を実現するS-ODSsのうち、デコードが最も遅いS-ODSslastのデコードが図中の期間dd2の経過時に終了するなら、このS-ODSslastのPTS(DSn[S-ODSslast])は、期間dd2の経過時を示すように設定される。

更に、初期表示の実現には、画面クリアを行い、デコードされたグラフィクスオブジェクトを転送せねばならないから、このPTS(DSn[S-ODSslast])の値に画面クリアに要する期間(90,000 × (8 × video\_width × video\_height // 128,000,000))、デコードされたグラフィクスオブジェクトの転送期間(90,000 × (SIZE(DSn[ICS.BUTTON[i]]) // 128,000,000))を

10

20

30

40

50



足した時点以降を、ICSのPTS(DSn[ICS])として設定せねばならない。以上がICSによる同期制御である。

#### 【0202】

尚、ICSのPTSによる同期制御は、再生時間軸上のあるタイミングでボタンを表示するという制御のみならず、再生時間軸上のある期間でPopupメニューの表示を可能とする制御を含む。Popupメニューとは、リモコン400に設けられたメニューキーの押下でPopup表示されるメニューであり、このPopup表示が、AVClipにおけるあるピクチャデータの表示タイミングで可能になることも、ICSのPTSによる同期制御である。Popupメニューを構成するODSは、ボタンを構成するODSと同様、ODSのデコードが完了し、デコードにより得られたグラフィックスオブジェクトがグラフィックスプレーンに書き込まれる。このグラフィックスプレーンへの書き込みが完了していなければ、ユーザからのメニューコールに応ずることはできない。そこでPopupメニューの同期表示にあたって、ICSのPTSに、Popup表示が可能になる時刻を示しておくのである。

#### 【0203】

以上が本発明に係る記録媒体の実施形態である。続いて本発明に係る再生装置の実施形態について説明する。第2実施形態に係る再生装置の内部構成は、第1実施形態に示したものと略同一になる。そしてこれらの構成において、Object Buffer15、Graphicalコントローラ17に改良が加えられている。以下、Object Buffer15、Graphicalコントローラ17の改良について説明する。

#### 【0204】

第2実施形態に係るObject Buffer15には、Stream Graphicsプロセッサ14のデコードにより得られたグラフィックスオブジェクトであって、対話画面を構成するためのボタンが配置される。図58は、Object Buffer15の格納内容をグラフィックスプレーン8と対比して示す図である。この格納内容は、図48、図49に示した具体例のODSが、Object Buffer15に書き込まれる場合を想定している。図48、図49の具体例は、4つのボタンのアニメーションを、36個のODS(ODS11~49)により実現するものであったが、このアニメーションの全てのコマを表すODSが、このObject Buffer15に格納される。一方グラフィックスプレーン8には、このObject Buffer15に格納された個々のODSの表示位置が規定される。この表示位置は、各ボタン情報のButton\_horizontal\_position、Button\_vertical\_positionにより定義されるものであり、Object Buffer15に格納されている複数ODSのそれぞれを、1コマずつ転送してグラフィックスプレーン8の表示位置に書き込んでゆくことにより、アニメーションは実現される。

#### 【0205】

第2実施形態に係るGraphicsコントローラ17は、Compositionバッファ16に配置されたICSを解釈して、ICSに基づく制御をする。この制御の実行タイミングは、ICSに付加されたPTSの値に基づく。このGraphicsコントローラ17のうち、重要なものは対話画面の初期表示時、更新時における書込処理である。以降、Graphicsコントローラ17による対話画面の初期表示時、更新時における書込処理を、図59を参照しながら説明する。図59は、初期表示時におけるGraphicsコントローラ17の処理を示す図である。本図に示すように、ボタンAにおけるボタン情報のButton\_horizontal\_position、Button\_vertical\_positionに規定される表示位置に、ボタンAのS-ODSsに属するODSを書き込み、ボタンB,C,Dにおけるボタン情報のButton\_horizontal\_position、Button\_vertical\_positionに規定される表示位置に、ボタンB,C,DのN-ODSsに属するODSを書き込むようGraphicsコントローラ17は制御を行う(図中の矢印w1,w2,w3,w4は、この書き込みを象徴的に示す)。この書き込みにより、図51に示した初期表示を作成することができる。ここで注目すべきは、対話画面の初期表示にあたっては、全てのODSが必要ではなく、デフォルトセレクトッドボタンのS-ODSsに属するODS、それ以外のボタンのN-ODSsに属するODSさえObject Buffer15に存在すれば、対話画面の初期表示は完了する点である。このことから、Object Buffer15に格納されるべき複数ODSのうち、デフォルトセレクトッドボタンのS-ODSsに属するODS、それ以外のボタンのN-ODSsに属するODSのデコードさえ完了すれば、Graphicsコン

トローラ 17 は対話画面の初期表示のための書き込みを開始することができる。

#### 【0206】

図 60 は、1stUserAction(MoveRight)による対話画面更新時における Graphics コントローラ 17 の処理を示す図である。本図に示すように、ボタン B のボタン情報におけるボタン情報の Button\_horizontal\_position、Button\_vertical\_position に規定される表示位置に、ボタン B の S-ODSs に属する ODS を書き込み、ボタン A のボタン情報におけるボタン情報の Button\_horizontal\_position、Button\_vertical\_position に規定される表示位置に、ボタン A の N-ODSs に属する ODS を書き込むよう Graphics コントローラ 17 は制御を行う(図中の矢印 w5, w6, w7, w8 は、この書き込みを象徴的に示す)。この書き込みにより、図 51 に示したような状態遷移を実現することができる。ボタン C, D は対話画面の初期表示時と同じくノーマル状態のままであるが、アニメーションを継続するため、グラフィクスプレーン 8 への書き込みが継続してなされている。

10

#### 【0207】

同じく、1stUserAction が、MoveDown, Activated である場合の、対話画面更新時における Graphics コントローラ 17 の処理を図 61、図 62 に示す。対話画面更新にあたっては、デフォルトセレクトボタン以外のボタンについての S-ODSs や、A-ODSs も必要になり、全ての ODS が Object Buffer 15 に格納されていることが望まれる。以上が Graphics コントローラ 17 の処理内容である。

#### 【0208】

以上のように構成された再生装置において、各構成要素は第 1 実施形態同様、パイプライン式にデコード処理を行う。

20

図 63 は、再生装置によるパイプライン処理を示すタイミングチャートである。第 4 段目は、BD-ROM における Display Set を示し、第 3 段目は、Coded Data バッファ 13 への ICS、PDS、ODS の読出期間を示す。第 2 段目は、Stream Graphics プロセッサ 14 による各 ODS のデコード期間を示す。第 1 段目は、Graphics コントローラ 17 による処理期間を示す。各 ODS のデコード開始時刻は、図中の DTS11, DTS12, DTS13 に示されている。Coded Data バッファ 13 への N-ODSs に属する最初の ODS (N-ODSs[ODS1]) の格納は DTS11 までに完了し、Coded Data バッファ 13 への N-ODSs に属する最後の ODS (N-ODSs[ODSn]) の格納は、DTS12 に示される時刻までに完了する。このように各 ODS は、自身の DTS に示される時刻までに Coded Data バッファ 13 への読み出しが完了している。

30

#### 【0209】

一方、各 ODS のデコード終了時刻は、図中の PTS11, PTS12, PTS13 に示されている。Stream Graphics プロセッサ 14 による N-ODSs (ODS1) のデコードは PTS11 までに完了し、N-ODSs (ODSn) のデコードは、PTS12 に示される時刻までに完了する。以上のように、各 ODS の DTS に示される時刻までに、ODS を Coded Data バッファ 13 に読み出し、Coded Data バッファ 13 に読み出された ODS を、各 ODS の PTS に示される時刻までに、デコードして Object Buffer 15 に書き込む。これらの処理を、1つの Stream Graphics プロセッサ 14 は、パイプライン式に行う。

#### 【0210】

デフォルトセレクトボタンが静的に確定している場合、対話画面の初期表示に必要なグラフィクスオブジェクトが Object Buffer 15 上で全て揃うのは、ノーマル状態に対応する button-state グループ、セレクト状態に対応する button-state グループの先頭 ODS のデコードが完了した時点である。本図でいえば、PTS13 に示される時点で、対話画面の初期表示に必要なグラフィクスオブジェクトは全て揃う。

40

#### 【0211】

本図の第 1 段目における期間 cd1 は、Graphics コントローラ 17 がグラフィクスプレーン 8 をクリアするのに要する期間である。また期間 td1 は、Object Buffer 15 上にえられたグラフィクスオブジェクトのうち、対話画面の最初の一枚を構成するグラフィクスオブジェクトを、グラフィクスプレーン 8 に書き込むのに要する期間である。グラフィクスプレーン 8 における書込先は、ICS における button\_horizontal\_position, button\_vertical\_

50

positionに示されている場所である。つまりODSのPTS13の値に、画面クリアの期間cd1と、デコードにより得られたグラフィクスオブジェクトの書込期間td1とを足し合わせれば、対話画面を構成する非圧縮グラフィクスがグラフィクスプレーン 8 上に得られることになる。この非圧縮グラフィクスの色変換をCLUT部 9 に行わせ、ビデオプレーン 6 に格納されている非圧縮ピクチャとの合成を加算器 10 に行わせれば、合成画像が得られることになる。

#### 【0212】

Display Setに含まれる全てのODSをデコードした上で初期表示を行う場合と比較すると、セレクトッド状態に対応するbutton-stateグループ、アクティブ状態に対応するbutton-stateグループのデコード完了を待つことなく、初期表示は可能になるので、図中の期間hy1だけ、初期表示の実行は早められることになる。

10

尚、本図におけるODS1～ODSnという表記は、「1」、「n」というような同じ番号が付されているが、これらN-ODSs,S-ODSs,A-ODSsに属するODSは別々のものである。以降、同様の表記の図は同じ意味であるとする。

#### 【0213】

グラフィクスデコーダ 12 において、Graphicsコントローラ 17 がグラフィクスプレーン 8 のクリアやグラフィクスプレーン 8 への書き込みを実行している間においても、Stream Graphicsプロセッサ 14 のデコードは継続して行われる(第2段目のODSnのデコード期間,ODS1のデコード期間,ODSnのデコード期間n,)。Graphicsコントローラ 17 によるグラフィクスプレーン 8 のクリアやグラフィクスプレーン 8 への書き込みが行われている間に、残りのODSに対するデコードは、継続してなされるので、残りのODSのデコードは早く完了する。残りのODSのデコードが早く完了することにより対話画面を更新するための準備は早く整うので、これら残りのODSを用いた対話画面更新も、ユーザ操作に即応することができる。以上のようなパイプライン処理により、対話画面の初期表示、更新の双方を迅速に実施することができる。

20

#### 【0214】

図63ではデフォルトセレクトッドボタンが静的に確定している場合を想定したが、図64は、デフォルトセレクトッドボタンが動的に変わる場合の、再生装置によるパイプライン処理を示すタイミングチャートである。デフォルトセレクトッドボタンが動的に変わる場合、button-stateグループに属する全てのODSをデコードして、グラフィクスオブジェクトをグラフィクスプレーン 8 に得れば、初期表示に必要なグラフィクスオブジェクトは全て揃う。Display Setに含まれる全てのODSをデコードした上で初期表示を行う場合と比較すると、アクティブ状態に対応するbutton-stateグループのデコード完了を待つことなく、初期表示は可能になる。そのため図中の期間hy2だけ、初期表示の実行は早められることになる。

30

#### 【0215】

図65は、グラフィクスプレーン 8、Object Buffer 15、Coded Dataバッファ 13、Compositionバッファ 16 における占有量の時間的遷移を示すタイミングチャートである。本図における占有量の表記は、図30に準ずる。第2実施形態では、N-ODSs,S-ODSs,A-ODSsを構成するODSがデコードの対象になっているため、第2段目、第3段目に示される単調増加、単調減少の数が増えている。これ以外は、図30と同一である。ODSに付与されたDTS、PTS、ICSに付与されたDTS、PTS、そして図27に示した各バッファのサイズ、転送レートを用いれば、第1実施形態と同様、図65に示したようなグラフを作図することにより、BD-ROMにて供給すべきAVClipの再生時において、各バッファの状態がどのように変化するかが、オーサリングの段階で明らかになる。このバッファ状態の遷移は、DTS、PTSに示される値を書き換えることで、調整することが可能なので、再生装置側のデコーダのスペックを越えるような復号負荷の発生を回避したり、再生にあたってのバッファオーバーフローの回避することが本実施形態でも可能になる。そのため再生装置の開発にあたってのハードウェア、ソフトウェアの実装が簡易になる。

40

#### 【0216】

50

以上が、第2実施形態に係る再生装置のパイプライン動作である。続いて第2実施形態に係る再生装置を実現するためのソフトウェアに対する改良について説明する。

図66は、機能セグメントのロード処理の処理手順を示すフローチャートである。本図は、図31のフローチャートをベースにして作図している。図31との差違は、ステップS29以降に、ステップS36、ステップS37を追加した点である。

#### 【0217】

ステップS36は、command\_update\_flagが1であるか否かの判定である。もし1であるなら(ステップS36でYes)、ボタン情報のボタンコマンドのみをCoded Dataバッファ13にロードし、それ以外を無視する(ステップS37)。もし0であるなら、ステップS22に移行する。これによりAcquisition Pointを示すICSは無視されることになる(ステップS24)。

10

#### 【0218】

DSが、図67のように多重化されている場合を想定して、DSの読み出しがどのように行われるかを説明する。図67の一例では、3つのDSが動画と多重化されている。この3つのDSのうち、初めのDS1は、Segment\_TypeがEpoch\_Startであり、Command\_update\_flagが0に設定され、LinkPL(PL#5)というボタンコマンドを含む。

DS10は、DS1のduplicateであり、Segment\_TypeはAcquisition Point、Command\_update\_flagが0に設定され、LinkPL(PL#5)というボタンコマンドを含む。

#### 【0219】

DS20は、DS1のInheritであり、Segment\_TypeはAcquisition Pointになっている。DS1から変化があるのはボタンコマンドであり(LinkPL(PL#10))、これを示すべくCommand\_update\_flagが1に設定されている。

20

かかる3つのDSが、動画と多重化されているAVClipにおいて、ピクチャデータpt10からの頭出しms1が行われたものとする。この場合、頭出し位置に最も近いDS10が、図66のフローチャートの対象となる。ステップS27においてComposition\_stateはAcquisition Pointと判定されるが、先行するDSはグラフィクスデコーダ12内に存在しないため、無視フラグは0に設定され、このDS10が再生装置のCoded Dataバッファ13にロードされる(図68のhs1)。一方、Display Setの存在位置より後から頭出しが行われた場合(ms2)、Display Set10に後続するDisplay Set20(図68のhs2)がCoded Dataバッファ13に読み出される。

30

#### 【0220】

図69のように通常再生が行われた場合のDS1,10,20のロードは、図70に示すものとなる。3つのDSのうち、ICSのSegment\_TypeがEpoch\_StartであるDS1は、そのままCoded Dataバッファ13にロードされるが(ステップS23)、ICSのSegment\_TypeがAcquisition PointであるDS10については、無視フラグが1に設定されるため(ステップS29)、これを構成する機能セグメントはCoded Dataバッファ13にロードされず無視される(ステップS24)。またDS20については、ICSのSegment\_TypeはAcquisition Pointであるが、Command\_update\_flagが1に設定されているので、ステップS36がYesになり、ボタンコマンドのみがロードされて、Coded Dataバッファ13上のDSのうち、ICS内のボタンコマンドのみをこれに置き換えられる(ステップS37)。しかし無視フラグは依然として1を示しているため、このボタンコマンド以外は、ロードされることなく無視される。

40

#### 【0221】

DSによる表示内容は同じであるが、DS20への到達時には、ボタンコマンドは、DSのLinkPL(#5)からLinkPL(#10)に置き換えられている。かかる置き換えにより、再生進行に伴い、ボタンコマンドの内容が変化するという制御が可能になる。続いてGraphicsコントローラ17の処理手順について説明する。図71はGraphicsコントローラ17の処理手順のうち、メインルーチンにあたる処理を描いたフローチャートである。本フローチャートは、タイムスタンプ同期処理(ステップS35)、アニメーション表示処理(ステップS36)、U0処理(ステップS37)という3つの処理を繰り返し実行するというものである。Graphicalコントローラ17の処理手順は、第1実施形態の図36～図38に示したものから、

50

図 7 1 ~ 図 7 8 に示すものへと大きく変化している。

#### 【 0 2 2 2 】

図 7 2 は、タイムスタンプによる同期制御の処理手順を示すフローチャートである。本フローチャートは、ステップ S 4 1、S 4 3 ~ ステップ S 4 7 の何れかの事象が成立しているかどうかを判定し、もしどれかの事象が成立すれば、該当する処理を実行してメインルーチンにリターンするというサブルーチンを構成する。

ステップ S 4 1 は、現在の再生時点が S-ODSfirst の PTS に示される時刻であるか、S-ODSslast の PTS に示される時刻であるかの判定であり、もしそうであるなら、ステップ S 4 2 において期間 を算出する。期間 とは、グラフィクスプレーンのクリアに要する時間 (2)、ODS のデコードにより得られたグラフィクスオブジェクトをグラフィクスプレーンに書き込む書込時間 (3) を足し合わせた期間である。

10

#### 【 0 2 2 3 】

ステップ S 4 2 において Graphical コントローラ 1 7 は、ICS の Segment\_Type を参照し、もし Segment\_Type が Epoch Start なら、プレーンクリア期間 (2) + プレーン書込期間 (3) をとする。Acquisition Point ならプレーン書込期間 (3) を とする。またプレーン書込期間 (3) の算出にあたっては、default\_selected\_button\_number が有効な値であれば図 5 4 ( a ) の計算で、default\_selected\_button\_number が = 0 であれば図 5 4 ( b ) の計算で算出する。こうして期間 を算出した後、ループ処理にリターンする。

#### 【 0 2 2 4 】

ステップ S 4 3 は、現在の再生時点が ICS の PTS- に示される時刻であるかの判定であり、もしそうであれば、グラフィクスプレーン 8 への書き込み処理を行って (ステップ S 5 1)、メインルーチンにリターンする。

20

ステップ S 4 5 は、現在の再生時点が ICS の PTS であるかの判定である。もしそうであれば、グラフィクスプレーン 8 の格納内容の出力を開始させる。この格納内容の出力先は、CLUT 部 9 であり、CLUT 部 9 により色変換がなされた上で、対話画面はビデオプレーン 6 の格納内容と合成される。これにより初期表示が実行される (ステップ S 5 2)。そして変数 animation(p) (p=1, 2, 3 · · · n) を 0 を設定して (ステップ S 5 3)、メインルーチンにリターンする。ここで変数 animation(p) とは、ボタン (p) のアニメーション表示を実行するにあたって、今何コマ目を表示しているかを示すグローバル変数 (複数フローチャートにわたって有効になる変数) である。ステップ S 5 3 では、全てのボタンについてのボタン (p) が、0 に設定されることになる。

30

#### 【 0 2 2 5 】

ステップ S 4 6、ステップ S 4 7 は、ICS に記述された時間情報に現在の再生時点が到達したかどうかの判定である。

ステップ S 4 6 は、現在の再生時点が selection\_TimeOut\_PTS に示される時刻であるかの判定であり、もしそうであれば、default\_activated\_button\_number で指定されるボタンをアクティベートする処理を行い、メインルーチンにリターンする (ステップ S 5 4)。

#### 【 0 2 2 6 】

ステップ S 4 7 は、現在の再生時点が Composition\_TimeOut\_PTS であるかの判定であり、もしそうであれば、画面クリアを行ってメインルーチンにリターンする (ステップ S 5 5)。以上がタイムスタンプによる同期処理である。この同期処理において、ステップ S 5 1、ステップ S 5 4 の処理手順は、サブルーチン化されている。ステップ S 5 1 のサブルーチンの処理手順を、図 7 3 を参照しながら説明する。

40

#### 【 0 2 2 7 】

図 7 3 は、メニューの初期表示をグラフィクスプレーン 8 へ書き込む処理の処理手順を示すフローチャートである。ステップ S 6 4 は、ICS における Segment\_type が Epoch Start であるか否かの判定であり、もし Epoch Start であればステップ S 6 5 においてグラフィクスプレーン 8 をクリアしてから、ステップ S 6 6 ~ ステップ S 7 3 の処理を行う。グラフィクスプレーン 8 のクリアに要する期間が、図 5 6、図 5 7 の期間 cd1 である。もし Epoch Start でなければステップ S 6 5 をスキップしてステップ S 6 6 ~ ステップ S 7 3 の処

50

理を行う。

【0228】

ステップS66～ステップS73は、ICSにおける各ボタン情報について繰り返されるループ処理を形成している(ステップS66、ステップS67)。本ループ処理において処理対象になるべきボタン情報をボタン情報(p)という。

ステップS67は、default\_selected\_button\_numberによる指定が有効であるか否かの判定であり、ステップS68は、button\_info(p)はdefault\_selected\_button\_numberにより指定されたデフォルトセレクトッドボタンに対応するボタン情報であるかの判定である。

【0229】

デフォルトセレクトッドボタンに対応するボタン情報でないなら、button\_info(p)のnormal\_state\_infoに指定されているstart\_object\_id\_normalのグラフィクスオブジェクトを、グラフィクスオブジェクト(p)としてObject Buffer15から特定する(ステップS69)

デフォルトセレクトッドボタンに対応するボタン情報であるなら、button\_info(p)のselected\_state\_infoに指定されているstart\_object\_id\_selectedのグラフィクスオブジェクトを、グラフィクスオブジェクト(p)としてObject Buffer15から特定して(ステップS70)、ボタン(p)をカレントボタンにする(ステップS71)。カレントボタンとは、現在表示中の対話画面において、セレクトッド状態になっているボタンであり、再生装置はこのカレントボタンの識別子を、PSR(10)として格納している。

【0230】

ステップS69、ステップS70を経ることでグラフィクスオブジェクト(p)が特定されれば、button\_info(p)のbutton\_horizontal\_position,button\_vertical\_positionに示されるグラフィクスプレーン8上の位置に、グラフィクスオブジェクト(p)を書き込む(ステップS72)。かかる処理をICSにおける各ボタン情報について繰り返せば、各ボタンの状態を表す複数グラフィクスオブジェクトのうち、最初のグラフィクスオブジェクトがグラフィクスプレーン8上に書き込まれることになる。Object Buffer15上の少なくとも初期表示に必要なグラフィクスオブジェクトについて、かかる処理を実行するのに要する期間が、図56、図57の期間td1である。以上がステップS51の詳細である。

【0231】

default\_selected\_button\_numberが"=0"であり、デフォルトセレクトッドボタンが動的に変わる場合は、ステップS67がNoになり、ステップS73においてbutton\_info(p)は、カレントボタンに対応するbutton\_infoであるか否かを判定する。もしそうであれば、ステップS70に、異なるならステップS69に移行する。

続いてステップS54のサブルーチンの処理手順を、図74を参照しながら説明する。

【0232】

図74は、デフォルトセレクトッドボタンのオートアクティベートの処理手順を示すフローチャートである。まずdefault\_activated\_\_button\_numberが0であるか、FFであるかどうかを判定し(ステップS75)、00であれば何の処理も行わずメインルーチンにリターンする。FFであれば、カレントボタンiをアクティブ状態に遷移する(ステップS77)。そしてカレントボタンiに対応する変数animation(i)を0に設定してメインルーチンにリターンする(ステップS78)。

【0233】

00でも、FFでもなければ、default\_activated\_\_button\_numberで指定されるボタンをカレントボタンとし(ステップS76)、カレントボタンiをアクティブ状態に遷移し(ステップS77)、カレントボタンiに対応する変数animation(i)を0に設定してメインルーチンにリターンする(ステップS78)。

以上の処理により、セレクトッド状態のボタンは、所定時間の経過時においてアクティブ状態に遷移させられることになる。以上が、図74のフローチャートの全容である。

【0234】

続いて、メニューにおけるアニメーション(ステップS 3 6)について説明する。図 7 5 は、アニメーション表示の処理手順を示すフローチャートである。

ここで初期表示は、各button\_infoのnormal\_state\_infoにおけるstart\_object\_id\_normal、selected\_state\_infoにおけるstart\_object\_id\_selectedで指定されているグラフィクスオブジェクトを、グラフィクスプレーン 8 に書き込まれることにより実現した。アニメーションとは、ステップS 3 5 ~ステップS 3 7 のループ処理が一巡する度に、各ボタンにおける任意のコマ(qコマ目にあるグラフィクスオブジェクト)をこのグラフィクスプレーン 8 に上書きする処理である。この更新は、button\_infoのnormal\_state\_info、selected\_state\_infoで指定されているグラフィクスオブジェクトを、一枚ずつグラフィクスプレーン 8 に書き込んでメインルーチンにリターンすることでなされる。ここで変数qとは、各ボタン情報のbutton\_infoのnormal\_state\_info、selected\_state\_infoで指定されている個々のグラフィクスオブジェクトを指定するための変数である。

#### 【 0 2 3 5 】

このアニメーション表示を実現するための処理を、図 7 5 を参照しながら説明する。尚本フローチャートは、記述の簡略化を期するため、ICSのrepeat\_normal\_flag、repeat\_selected\_flagが繰り返し要と設定されているとの前提で作図している。

ステップS 8 0 は初期表示が済んでいるか否かの判定であり、もし済んでいなければ何の処理も行わずにリターンする。もし済んでいればステップS 8 1 ~ステップS 9 3 の処理を実行する。ステップS 8 1 ~ステップS 9 3 は、ICSにおける各button\_infoについて、ステップS 8 3 ~ステップS 9 3 の処理を繰り返すというループ処理を構成している(ステップS 8 1、ステップS 8 2)。

#### 【 0 2 3 6 】

ステップS 8 3 は、button\_info(p)に対応する変数animation(p)を変数qに設定する。こうして、変数qは、button\_info(p)に対応する、現在のコマ数を示すことになる。

ステップS 8 4 は、button\_info(p)が、現在セレクト状態にあるボタン(カレントボタン)に対応するbutton\_infoであるか否かの判定である。

カレントボタン以外のボタンならば、button\_info(p).normal\_state\_infoにおけるstart\_object\_id\_normalに変数qを足した識別子をID(q)とする(ステップS 8 5)。

#### 【 0 2 3 7 】

カレントボタンに対応するボタンであれば、ステップS 8 6 の判定を行う。

ステップS 8 6 は、カレントボタンがアクティブ状態であるかの判定であり、もしそうであれば、ステップS 8 7 においてbutton\_info(p).actioned\_state\_infoにおけるstart\_object\_id\_actionedに変数qを足した識別子をID(q)とする。そしてbutton\_info(p)に含まれるボタンコマンドのうち、1つを実行する(ステップS 8 8)。

#### 【 0 2 3 8 】

カレントボタンがアクティブ状態でなければ、button\_info(p).selected\_state\_infoにおけるstart\_object\_id\_selectedに変数qを足した識別子をID(q)とする(ステップS 8 9)。

こうしてID(q)が決まれば、Object Buffer15に存在する、ID(q)を有するグラフィクスオブジェクト(p)を、button\_info(p)のbutton\_horizontal\_position,button\_vertical\_positionに示されるGraphics Plane8上の位置に書き込む(ステップS 9 0)。

#### 【 0 2 3 9 】

以上のループ処理により、カレントボタンのセレクト状態(若しくはアクティブ状態)及びその他のボタンのノーマル状態を構成する複数グラフィクスオブジェクトのうち、q枚目のものがグラフィクスプレーン 8 に書き込まれることになる。

ステップS 9 1 は、start\_object\_id\_normal + qがend\_object\_id\_normalに達したか否かの判定であり、もし達しないなら変数qをインクリメントした値を変数animation(p)に設定する(ステップS 9 2)。もし達したなら変数animation(p)を0に初期化する(ステップS 9 3)。以上の処理は、ICSにおける全てのbutton\_infoについて繰り返される(ステップS 8 1、ステップS 8 2)。全てのbutton\_infoについて、処理がなされれば、メインルー

10

20

30

40

50

チンにリターンする。

#### 【0240】

以上のステップS80～ステップS93により対話画面における各ボタンの絵柄は、ステップS35～ステップS37が一巡する度に新たなグラフィクスオブジェクトに更新される。ステップS35～ステップS37の処理が何度も反復されれば、いわゆるアニメーションが可能になる。アニメーションにあたって、グラフィクスオブジェクトコマの表示間隔は、animation\_frame\_rate\_codeに示される値になるようにGraphicsコントローラ17は時間調整を行う。

#### 【0241】

尚、ステップS88においてbutton\_info(p)に含まれるボタンコマンドを1つずつ実行したが、アクティブ状態に対応するグラフィクスオブジェクトを一通り表示した後に、button\_info(p)に含まれるボタンコマンドをまとめて実行してもよい。以上でアニメーション表示処理についての説明を終わる。続いてメインルーチンのステップS37におけるU0処理の処理手順について図76を参照しながら説明する。

#### 【0242】

図76は、U0処理の処理手順を示すフローチャートである。本フローチャートは、ステップS100～ステップS103の何れかの事象が成立しているかどうかを判定し、もしどれかの事象が成立すれば、該当する処理を実行してメインルーチンにリターンする。ステップS100は、U0maskTableが"1"に設定されているかどうかの判定であり、もしに設定されていれば、何の処理も行わずに、メインルーチンにリターンする。

#### 【0243】

ステップS101は、MoveUP/Down/Left/Rightキーが押下されたかどうかの判定であり、もしこれらのキーが押下されれば、カレントボタンを変更して(ステップS104)、カレントボタンのauto\_action\_flagが01かどうかを判定する(ステップS108)。もし違うならメインルーチンにリターンする。もしそうであるなら、ステップS105に移行する。

#### 【0244】

ステップS102は、activatedキーが押下されたかどうかの判定であり、もしそうであれば、カレントボタンiをアクティブ状態に遷移する(ステップS105)。その後、変数animation(i)を0に設定する(ステップS106)。

ステップS103は、数値入力であるかどうかの判定であり、もし数値入力であれば、数値入力処理を行って(ステップS107)、メインルーチンにリターンする。図76の処理手順のうち、ステップS104、ステップS107はサブルーチン化されている。このサブルーチンの処理手順を示したのが図77、図78である。以降これらのフローチャートについて説明する。

#### 【0245】

図77は、カレントボタンの変更処理の処理手順を示すフローチャートである。先ず初めに、カレントボタンのneighbor\_infoにおけるupper\_button\_number, lower\_button\_number, left\_button\_number, right\_button\_numberのうち、押下されたキーに対応するものを特定する(ステップS110)。

そしてカレントボタンをボタンiとし、新たにカレントボタンになるボタンをボタンjとする(ステップS111)。ステップS112は、ステップS111で特定されたボタンjが、ボタンiと一致しているかどうかの判定である。もし一致していれば、何の処理も行わずにメインルーチンにリターンする。もし一致しなければ、ボタンjをカレントボタンにして(ステップS113)、変数animation(i), 変数animation(j)を0に設定した上でメインルーチンにリターンする(ステップS114)。

#### 【0246】

図78は、数値入力処理の処理手順を示すフローチャートである。入力された数値に合致するbutton\_numberを有したButton\_info.jが存在するかどうかの判定を行い(ステップS121)、Button\_info.jにおけるnumerically\_selectable\_flagは1であるかどうかの判

10

20

30

40

50



定を行う(ステップS 1 2 2)。ステップS 1 2 1 及びステップS 1 2 2 がYesなら、カレントボタンをノーマル状態に遷移させ、ボタンjをカレントボタンにして(ステップS 1 2 3)、変数animation(i),変数animation(j)を0に設定した上で(ステップS 1 2 4)、Button info.jのauto\_action\_flagは1であるかを判定する(ステップS 1 2 5)。1でないならメインルーチンにリターンする。

#### 【0 2 4 7】

1であるなら、ステップS 1 2 6 においてカレントボタンをアクティブ状態に遷移した上でメインルーチンにリターンする。

ステップS 1 2 1 ~ S 1 2 2 のどちらかがNoなら、そのままメインルーチンにリターンする。

10

以上が同期表示を行う場合のGraphicsコントローラ17の処理手順である。Popup表示のように、ユーザ操作をトリガとした対話画面表示を行う場合、Stream Graphicsプロセッサ14、Graphicsコントローラ17は以下のような処理を行う。つまり、同期表示の場合と同様の処理を行う。これにより、グラフィックスプレーン8にはグラフィックスオブジェクトが得られる。このようにグラフィックスオブジェクトを得た後、現在の再生時点が、ICSに付加されたPTSに示される時点を経過するのを待つ。そしてこの再生時点の経過後、UOコントローラ18がメニューコールを示すUOを受け付けば、グラフィックスプレーン8に格納されたグラフィックスオブジェクトを合成させるよう、CLUT部9に出力する。UOに同期して、かかる出力を行えば、メニューコールの押下に応じたPopup表示を実現することができる。

20

#### 【0 2 4 8】

以上、DSnに属するICSのPTS,ODSのDTS、PTSの設定について説明したが、ICSのDTSや、PDSのDTS、PTS、ENDのDTS、PTSについては説明していない。以下、これらのタイムスタンプについて説明する。第2実施形態ではWDSは存在しないので、ICSは、DSnにおける最初のODS(ODS1)のデコード開始時点(DTS(DSn[ODS1]))以前、及び、DSnにおける最初のPDS(PDS1)が有効になる時点(PTS(DSn[PDS1]))以前に、Compositionバッファ16にロードされねばならない。よって以下の式の関係を満たす値に、設定されねばならない。

#### 【0 2 4 9】

$$DTS(DSn[ICS]) \leq DTS(DSn[ODS1])$$

$$DTS(DSn[ICS]) \leq PTS(DSn[PDS1])$$

30

続いてDSnに属する各PDSのDTS,PTSの設定について説明する。

DSnに属する各PDSは、ICSがCompositionバッファ16にロードされる時点(DTS(DSn[ICS]))から、最初のODSのデコード開始時点(DTS(DSn[ODS1]))までに、CLUT部9において、有効になればよい。このことからDSnに属する各PDS(PDS1 ~ PDSlast)のPTS値は、以下の関係を満たす値に、設定されねばならない。

$$DTS(DSn[ICS]) \leq PTS(DSn[PDS1])$$

$$PTS(DSn[PDSj]) \leq PTS(DSn[PDSj+1]) \leq PTS(DSn[PDSlast])$$

$$PTS(DSn[PDSlast]) \leq DTS(DSn[ODS1])$$

続いてDSnに属するEND of Display SetSegmentのPTSの設定について説明する。DSnに属するENDは、DSnの終わりを示すものだから、DSnに属する最後のODS(ODSlast)のデコード終了時刻を示せばよい。このデコード終了時刻は、ODSlastのPTS(PTS(DSn[ODSlast]))に示されているので、ENDのPTSは、以下の式に示される値に設定されねばならない。

$$PTS(DSn[END]) = PTS(DSn[ODSlast])$$

DSn,DSn+1に属するICSとの関係で考えれば、DSnにおけるICSは、最初のODS(ODS1)のロード時刻以前に、Compositionバッファ16にロードされるから、ENDのPTSは、DSnに属するICSのロード時刻(DTS(DSn[ICS]))以降、DSn+1に属するICSのロード時刻(DTS(DSn+1[ICS]))以前でなければならない。そのためENDのPTSは、以下の式の関係を満たす必要がある。

$$DTS(DSn[ICS]) \leq PTS(DSn[END]) \leq DTS(DSn+1[ICS])$$

一方、最初のODS(ODS1)のロード時刻は、最後のPDS(PDSlast)のロード時刻以後である

50

から、ENDのPTS(PTS(DSn[END]))は、DSnに属するPDSのロード時刻以降(PTS(DSn[PDSlast]))でなければならない。そのためENDのPTSは、以下の式の間を満たす必要がある。

$$\text{PTS}(\text{DSn}[\text{PDSlast}]) \leq \text{PTS}(\text{DSn}[\text{END}])$$

DTS,PTSが設定されたICS,PDS,ODSをAVClipに組み込んでおくので、ある動画の一コマが画面に現れたタイミングに、特定の処理を再生装置に実行させるという対話制御、つまり動画内容と緻密に同期した対話制御の記述に便利である。またICS,PDS,ODSは、AVClip自身に多重化されているので、再生制御を行いたい区間が数百個であっても、それらに対応するICS,PDS,ODSの全てをメモリに格納しておく必要はない。ICS,PDS,ODSはビデオパケットと共にBD-ROMから読み出されるので、現在再生すべき動画区間に対応するICS,PDS,ODSをメモリに常駐させ、この動画区間の再生が終われば、ICS,PDS,ODSをメモリから削除して、次の動画区間に対応するICS,PDS,ODSをメモリに格納すればよい。ICS,PDS,ODSは、AVClipに多重化されるので、たとえICS,PDS,ODSの数が数百個になってもメモリの搭載量を必要最低限にすることができる。

10

#### 【0250】

以上のように本実施形態によれば、アニメーションを実現するためのODSが360枚存在しており、ボタン部材が3つの状態をもっている場合、ODSは、120枚+120枚+120枚というように、3つのbutton-stateグループにグルーピングされる。そして個々のbutton-stateグループは、早く現れる状態に対応するもの程、前に置かれ、遅く現れる状態に対応するもの程、後に置かれる。このため、再生時にあたって、早く現れる状態に対応するbutton-stateグループの再生装置へのロードは早く行われ、遅く現れる状態に対応するbutton-stateグループのロードは、後回しにされる。早く現れる状態に対応するbutton-stateグループのロードは早い時期になされるので、360枚のものODSの読み出し/デコードは未完であっても、全体の約1/3~2/3のODSの読み出し/デコードが完了していれば、初期表示のための準備は整う。全体の約1/3~2/3のODSの読み出し/デコードの完了時点で、初期表示のための処理を開始させることができるので、たとえ読み出し/デコードすべきODSが大量にあっても、初期表示の実行は遅滞することはない。このため、アニメーションを伴った楽しい対話画面の表示を、迅速に実行することができる。

20

#### 【0251】

##### (第3実施形態)

本実施形態は、BD-ROMの製造工程に関する実施形態である。図79は、第1実施形態に示したPCSを作成するための製造工程を示す図である。

30

BD-ROMの制作工程は、動画収録、音声収録等の素材作成を行う素材制作工程S201、オーサリング装置を用いて、アプリケーションフォーマットを生成するオーサリング工程S202、BD-ROMの原盤を作成し、プレス・貼り合わせを行って、BD-ROMを完成させるプレス工程S203を含む。

#### 【0252】

これらの工程のうち、BD-ROMを対象としたオーサリング工程は、以下のステップS204~ステップS210を含む。

ステップS204において制御情報、ウィンドウ定義情報、パレット定義情報、グラフィクスを記述し、ステップS205では、制御情報、ウィンドウ定義情報、パレット定義情報、グラフィクスを機能セグメントに変換する。そしてステップS206において同期したいピクチャが出現するタイミングに基づき、PCSのPTSを設定し、ステップS207では、PTS[PCS]の値に基づき、DTS[ODS],PTS[ODS]を設定する。ステップS208において、DTS[ODS]の値に基づき、DTS[PCS],PTS[PDS],DTS[WDS],PTS[WDS]を設定し、ステップS209では、プレーヤモデルにおける各バッファの占有量の時間的遷移をグラフ化する。ステップS210では、グラフ化された時間的遷移がプレーヤモデルの制約を満たすか否かを判定し、もし満たさないなら、ステップS211において各機能セグメントのDTS,PTSを書き換える。もし満たすならステップS212においてグラフィクスストリームを生成し、ステップS213においてグラフィクスストリームを別途生成されたビデオストリーム、オーディオストリームと多重してAVClipを得る。以降、AVClipをBD-ROMのフォーマ

40

50

ットに適合させることにより、アプリケーションフォーマットが完成する。

【0253】

以上が、第1実施形態に係る記録媒体の製造工程である。第2実施形態に係る記録媒体の製造工程は、図80に示すものとなる。図79と比較すると、図80は、ステップS204～ステップS208が、ステップS304～ステップS308に置き換えられている。

以降ステップS304～ステップS308について説明する。ステップS304において制御情報、パレット定義情報、グラフィクスを記述し、ステップS305において、制御情報、パレット定義情報、グラフィクスを機能セグメントに変換する。ステップS306では、同期したいピクチャが出現するタイミングに基づき、ICSのPTSを設定する。そしてステップS307では、PTS[ICS]の値に基づき、DTS[ODS],PTS[ODS]を設定し、ステップS308においてDTS[ODS]の値に基づき、DTS[ICS],PTS[PDS]を設定する。

【0254】

(備考)

以上の説明は、本発明の全ての実施行為の形態を示している訳ではない。下記(A)(B)(C)(D)・・・の変更を施した実施行為の形態によっても、本発明の実施は可能となる。本願の請求項に係る各発明は、以上に記載した複数の実施形態及びそれらの変形形態を拡張した記載、ないし、一般化した記載としている。拡張ないし一般化の程度は、本発明の技術分野の、出願当時の技術水準の特性に基づく。

【0255】

(A)全ての実施形態では、本発明に係る記録媒体をBD-ROMとして実施したが、本発明の記録媒体は、記録されるグラフィクスストリームに特徴があり、この特徴は、BD-ROMの物理的性質に依存するものではない。グラフィクスストリームを記録しうる記録媒体なら、どのような記録媒体であってもよい。例えば、DVD-ROM,DVD-RAM,DVD-RW,DVD-R,DVD+RW,DVD+R,CD-R,CD-RW等の光ディスク、PD,MO等の光磁気ディスクであってもよい。また、コンパクトフラッシュ(登録商標)カード、スマートメディア、メモリスティック、マルチメディアカード、PCM-CIAカード等の半導体メモリカードであってもよい。フレキシブルディスク、SuperDisk,Zip,Clik!等の磁気記録ディスク(i)、ORB,Jaz,SparQ,SyJet,EZFly,マイクロドライブ等のリムーバブルハードディスクドライブ(ii)であってもよい。更に、機器内蔵型のハードディスクであってもよい。

【0256】

(B)全ての実施形態における再生装置は、BD-ROMに記録されたAVClipをデコードした上でTVに出力していたが、再生装置をBD-ROMドライブのみとし、これ以外の構成要素をTVに具備させてもよい、この場合、再生装置と、TVとをIEEE1394で接続されたホームネットワークに組み入れることができる。また、実施形態における再生装置は、テレビと接続して利用されるタイプであったが、ディスプレイと一体型となった再生装置であってもよい。更に、各実施形態の再生装置において、処理の本質的部分をなすシステムLSI(集積回路)のみを、実施としてもよい。これらの再生装置及び集積回路は、何れも本願明細書に記載された発明であるから、これらの何れの態様であろうとも、第1実施形態に示した再生装置の内部構成を元に、再生装置を製造する行為は、本願の明細書に記載された発明の実施行為になる。第1実施形態に示した再生装置の有償・無償による譲渡(有償の場合は販売、無償の場合は贈与になる)、貸与、輸入する行為も、本発明の実施行為である。店頭展示、カタログ勧誘、パンフレット配布により、これらの譲渡や貸渡を、一般ユーザに申し出る行為も本再生装置の実施行為である。

【0257】

(C)各フローチャートに示したプログラムによる情報処理は、ハードウェア資源を用いて具体的に実現されていることから、上記フローチャートに処理手順を示したプログラムは、単体で発明として成立する。全ての実施形態は、再生装置に組み込まれた態様で、本発明に係るプログラムの実施行為についての実施形態を示したが、再生装置から分離して、第1実施形態に示したプログラム単体を実施してもよい。プログラム単体の実施行為に

は、これらのプログラムを生産する行為(1)や、有償・無償によりプログラムを譲渡する行為(2)、貸与する行為(3)、輸入する行為(4)、双方向の電子通信回線を介して公衆に提供する行為(5)、店頭、カタログ勧誘、パンフレット配布により、プログラムの譲渡や貸渡を、一般ユーザに申し出る行為(6)がある。

【0258】

(D)各フロ - チャ - トにおいて時系列に実行される各ステップの「時」の要素を、発明を特定するための必須の事項と考える。そうすると、これらのフロ - チャ - トによる処理手順は、再生方法の使用形態を開示していることがわかる。各ステップの処理を、時系列に行うことで、本発明の本来の目的を達成し、作用及び効果を奏するよう、これらのフロ - チャ - トの処理を行うのであれば、本発明に係る記録方法の実施行為に該当することはいうまでもない。

10

【0259】

(E)BD-ROMに記録するにあたって、AVClipを構成する各TSパケットには、拡張ヘッダを付与しておくことが望ましい。拡張ヘッダは、TP\_extra\_headerと呼ばれ、『Arrival\_Time\_Stamp』と、『copy\_permission\_indicator』とを含み4バイトのデータ長を有する。TP\_extra\_header付きTSパケット(以下EX付きTSパケットと略す)は、32個毎にグループ化されて、3つのセクタに書き込まれる。32個のEX付きTSパケットからなるグループは、6144バイト(=32×192)であり、これは3個のセクタサイズ6144バイト(=2048×3)と一致する。3個のセクタに収められた32個のEX付きTSパケットを"Aligned Unit"という。

【0260】

20

IEEE1394を介して接続されたホームネットワークでの利用時において、再生装置は、以下のような送信処理にてAligned Unitの送信を行う。つまり送り手側の機器は、Aligned Unitに含まれる32個のEX付きTSパケットのそれぞれからTP\_extra\_headerを取り外し、TSパケット本体をDTCP規格に基づき暗号化して出力する。TSパケットの出力にあたっては、TSパケット間の随所に、isochronousパケットを挿入する。この挿入箇所は、TP\_extra\_headerのArrival\_Time\_Stampに示される時刻に基づいた位置である。TSパケットの出力に伴い、再生装置はDTCP\_Descriptorを出力する。DTCP\_Descriptorは、TP\_extra\_headerにおけるコピー許否設定を示す。ここで「コピー禁止」を示すようDTCP\_Descriptorを記述しておけば、IEEE1394を介して接続されたホームネットワークでの利用時においてTSパケットは、他の機器に記録されることはない。

30

(F)各実施形態におけるデジタルストリームは、BD-ROM規格のAVClipであったが、DVD-Video規格、DVD-Video Recording規格のVOB(Video Object)であってもよい。VOBは、ビデオストリーム、オーディオストリームを多重化することにより得られたISO/IEC13818-1規格準拠のプログラムストリームである。またAVClipにおけるビデオストリームは、MPEG4やWMV方式であってもよい。更にオーディオストリームは、Linear-PCM方式、Dolby-AC3方式、MP3方式、MPEG-AAC方式、dts方式であってもよい。

【0261】

(G)各実施形態における映画作品は、アナログ放送で放送されたアナログ映像信号をエンコードすることにより得られたものでもよい。デジタル放送で放送されたトランスポートストリームから構成されるストリームデータであってもよい。

40

またビデオテープに記録されているアナログ/デジタルの映像信号をエンコードしてコンテンツを得ても良い。更にビデオカメラから直接取り込んだアナログ/デジタルの映像信号をエンコードしてコンテンツを得ても良い。他にも、配信サーバにより配信されるデジタル著作物でもよい。

【0262】

(H)第1実施形態～第2実施形態に示したグラフィックスオブジェクトは、ランレングス符号化されたラスタデータである。グラフィックスオブジェクトの圧縮・符号化方式にランレングス符号方式を採用したのは、ランレングス符号化は字幕の圧縮・伸長に最も適しているためである。字幕には、同じ画素値の水平方向の連続長が比較的長くなるという特性があり、ランレングス符号化による圧縮を行えば、高い圧縮率を得ることができる。

50

また伸長のための負荷も軽く、復号処理のソフトウェア化に向いている。デコードを実現する装置構成を、字幕・グラフィックスオブジェクト間で共通化する目的で、字幕と同じ圧縮・伸長方式をグラフィックスオブジェクトに採用している。しかし、グラフィックスオブジェクトにランレングス符号化方式を採用したというのは、本発明の必須事項ではなく、グラフィックスオブジェクトはPNGデータであってもよい。またラスターデータではなくベクタデータであってもよい、更に透明な絵柄であってもよい。

【0263】

(I)PCSによる表示効果の対象は、装置側の言語設定に応じて選ばれた字幕のグラフィックスであってもよい。これにより、現状のDVDにおいて動画像本体で表現していたような文字を用いた表示効果を、装置側の言語設定に応じて表示された字幕グラフィックスで実現することができ、実用上の価値は大きい。

10

(J)PCSによる表示効果の対象は、装置側のディスプレイ設定に応じて選ばれた字幕グラフィックスであってもよい。つまり、ワイドビジョン、パンスキャン、レターボックス用といった様々な表示モード用のグラフィックスがBD-ROMに記録されており、装置側は自身に接続されたテレビの設定に応じてこれらの何れかを選んで表示する。この場合、そうして表示された字幕グラフィックスに対し、PCSに基づく表示効果をほどこすので、見栄えがよくなる。これにより、動画像本体で表現していたような文字を用いた表示効果を、装置側のディスプレイ設定に応じて表示された字幕で実現することができ、実用上の価値は大きい。

【0264】

20

(K)第1実施形態ではグラフィックスプレーンへの書込レート $R_c$ は、1ビデオフレーム内にグラフィックスプレーンクリア及び再描画が可能になるよう、windowのサイズを全体の25%に定めたが、これらクリア・再描画が垂直帰線期間に完遂するよう、 $R_c$ を定めても良い。垂直帰線期間は1/29.93秒の25%と仮定すると、 $R_c$ は1Gbpsになる。 $R_c$ をこのように設定することでグラフィックス表示はスムーズになされるので、実用上の効果は大きい。

【0265】

また垂直帰線期間での書き込みに加え、ラインスキャンに同期した書き込みを併用してもよい。これにより、 $R_c=256\text{Mbps}$ の書込レートであっても、スムーズな表示の実現が可能になる。

(L)各実施形態において再生装置には、グラフィックスプレーンを実装したが、このグラフィックスプレーンに代えて、一ライン分の非圧縮画素を格納するラインバッファを具備してもよい。映像信号への変換は水平行(ライン)毎に行われるので、このラインバッファさえ具備していれば、この映像信号への変換は行なえるからである。

30

【0266】

(M)グラフィックスたる字幕は、映画の台詞を表す文字列であるとして説明を進めたが、商標を構成するような図形、文字、色彩の組合せや、国の紋章、旗章、記章、国家が採用する監督/証明用の公の記号・印章、政府間国際機関の紋章、旗章、記章、特定商品の原産地表示を含んでいてもよい。

(N)第1実施形態では、字幕を画面の上側、下側に横書きで表示するものとして、ウィンドウをグラフィックスプレーンの上側、下側に定義したが、字幕を画面の右側、左側に表示するものとして、ウィンドウをグラフィックスプレーンの右側、左側に定義してもよい。こうすることにより、日本語字幕を縦書きで表示することができる。

40

【0267】

(O)各実施形態におけるAVClipは、映画作品を構成するものであったが、AVClipはカラオケを実現するものであってもよい、そしてこの場合、PCSは歌の進行に応じて、字幕の色を変え、という表示効果を実現してもよい。

(P)複数再生経路の合流点においてデフォルトセレクトボタンが変わるようなケースでは、複数再生経路のそれぞれの経路時に、固有の値を再生装置側のレジスタに設定するよう、動的シナリオにおいて再生制御を記述しておくことが望ましい。そして、そのレジスタの設定値に応じたボタンをセレクト状態に設定するよう再生手順を記述してお

50

けば、どの再生経路を経由するかによって、セレクトッド状態とすべきボタンを変化させることができる。

【産業上の利用可能性】

【0268】

本発明に係る記録媒体、再生装置は、表示効果を伴う字幕表示やアニメーションを伴う対話表示を実現することができるので、より付加価値が高い映画作品を市場に供給することができ、映画市場や民生機器市場を活性化させることができる。故に本発明に係る記録媒体、再生装置は、映画産業や民生機器産業において高い利用可能性をもつ。

【図面の簡単な説明】

【0269】

【図1】本発明に係る記録媒体の、使用行為についての形態を示す図である。

【図2】BD-ROMの構成を示す図である。

【図3】AVClipがどのように構成されているかを模式的に示す図である。

【図4】(a)プレゼンテーショングラフィクスストリームの構成を示す図である。(b)機能セグメントを変換することで得られるPESパケットを示す図である。

【図5】様々な種別の機能セグメントにて構成される論理構造を示す図である。

【図6】字幕の表示位置と、Epochとの関係を示す図である。

【図7】(a)ODSによるグラフィクスオブジェクトの定義を示す図である。(b)PDSのデータ構造を示す図である。

【図8】(a)WDSのデータ構造を示す図である。(b)PCSのデータ構造で構成される

。【図9】字幕表示を実現するためのDisplay Setの記述例である。

【図10】DS1におけるWDS、PCSの記述例を示す図である。

【図11】DS2におけるPCSの記述例を示す図である。

【図12】DS3におけるPCSの記述例を示す図である。

【図13】時間軸に沿った連続写真的な表記で、Cut-In/Outを実行する際のDisplay Setの記述例を示す図である。

【図14】時間軸に沿った連続写真的な表記で、Fade-In/Outを実行する際のDisplay Setの記述例を示す図である。

【図15】時間軸に沿った連続写真的な表記で、Scrollingがどのように行われるかを記述した図である。

【図16】時間軸に沿った連続写真的な表記で、Wipe-In/Outがどのように行われるかを記述した図である。

【図17】ウィンドウ内にグラフィクスオブジェクトが4つ存在する場合と、2つの存在する場合とを対比して示す図である。

【図18】decode\_durationの計算アルゴリズムの一例を示す図である。

【図19】図18のプログラムのアルゴリズムを図式化したフローチャートである。

【図20】(a)(b)図18のプログラムのアルゴリズムを図式化したフローチャートである。

【図21】(a)1つのwindowに1つのODSが存在するケースを想定した図である。(b)(c)図18で引用した各数値の時間的な前後関係を示すタイミングチャートである。

【図22】(a)1つのwindowに2つのODSが存在するケースを想定した図である。(b)(c)図18で引用した各数値の時間的な前後関係を示すタイミングチャートである。

【図23】(a)2つのwindowのそれぞれに、ODSが1つずつ存在するケースを想定したタイミングチャートである。(b)デコード期間(2)がクリア期間(1)+書込期間(31)より長くなるケースを示すタイミングチャートである。(c)クリア期間(1)+書込期間(31)がデコード期間(2)より長くなるケースを示すタイミングチャートである。

【図24】アップデートの、時間的変遷の具体例を示す図である。

【図25】(a)図24に示したアップデートを実現するため、記述された4つのDisplay Setを示す図である。(b)図25(a)におけるDisplay Setに属する各機能セグメ

10

20

30

40

50

ントのDTS、PTSがどのように設定されているかを示すタイミングチャートである。

【図26】本発明に係る再生装置の内部構成を示す図である。

【図27】書込レートRx,Rc,Rd、グラフィックスプレーン8、Coded Data Buffer13、Object Buffer15、Composition Buffer16のサイズを示す図である。

【図28】再生装置によるパイプライン処理を示すタイミングチャートである。

【図29】ODSのデコードが、グラフィックスプレーンのクリアより早く終わる場合を想定したパイプライン処理を示すタイミングチャートである。

【図30】機能セグメントのロード処理の処理手順を示すフローチャートである。

【図31】多重化の一例を示す図である。

【図32】DS10が再生装置のCoded Data Buffer13にロードされる様子を示す図である

10

。【図33】通常再生が行われる場合を示す図である。

【図34】図33のように通常再生が行われた場合のDS1,10,20のロードを示す図である

。【図35】Graphical Controller17の処理手順を示すフローチャートである。

【図36】Graphical Controller17の処理手順を示すフローチャートである。

【図37】Graphical Controller17の処理手順を示すフローチャートである。

【図38】PDSにおけるPTSに基づく、再生装置におけるパイプラインを示す図である。

【図39】再生装置のパイプライン動作時における、ENDの意味合いを示す図である。

【図40】第2実施形態に係る再生装置の内部構成を示す図である。

20

【図41】ダブルバッファによるグラフィックスプレーン読み出し/書き込みを模式的に示す図である。

【図42】(a)(b)第2実施形態に係る対話画面についての説明図である。

【図43】Interactive Composition Segmentのデータ構造を示す図である。

【図44】あるDSnに含まれるODSと、ICSとの関係を示す図である。

【図45】任意のピクチャデータpt1の表示タイミングにおける画面合成を示す図である

。【図46】ICSにおけるボタン情報の設定例を示す図である。

【図47】ボタンA~ボタンDの状態遷移を示す図である。

【図48】ODS11,21,31,41の絵柄の一例を示す図である。

30

【図49】ボタンA用のODS11~19の絵柄の一例を示す図である。

【図50】Display Setに属するODSの順序及びbutton-stateグループを示す図である。

【図51】図50のbutton-stateグループが配置された対話画面における状態遷移を示す図である。

【図52】Display SetにおけるODSの順序を示す図である。

【図53】default\_selected\_button\_numberが"=0"である場合と、"=ボタンB"である場合とでS-ODSsにおいてODSの並びがどのように変わるかを示す図である。

【図54】(a)(b)N-ODSsにボタンA~Dを構成する複数ODSが含まれており、S-ODSsにボタンA~Dを構成する複数ODSが含まれている場合、SIZE(DSn[ICS.BUTTON[i]])がどのような値になるかを示す図である。

40

【図55】ICSによる同期表示時のタイミングを示す図である。

【図56】対話画面の初期表示が複数ODSにて構成され、default\_selected\_button\_numberが有効である場合のDTS、PTSの設定を示す図である。

【図57】対話画面の初期表示が複数ODSにて構成され、default\_selected\_button\_numberが無効である場合のDTS、PTSの設定を示す図である。

【図58】Object Buffer15の格納内容をグラフィックスプレーン8と対比して示す図である。

【図59】初期表示時におけるGraphicsコントローラ17の処理を示す図である。

【図60】1stUserAction(MoveRight)による対話画面更新時におけるGraphicsコントローラ17の処理を示す図である。

50

【図 6 1】1stUserAction(MoveDown)による対話画面更新時におけるGraphicsコントローラ 17 の処理を示す図である。

【図 6 2】1stUserAction(Activated)による対話画面更新時におけるGraphicsコントローラ 17 の処理を示す図である。

【図 6 3】再生装置によるパイプライン処理を示すタイミングチャートである。

【図 6 4】デフォルトセレクトボタンが動的に変わる場合の、再生装置によるパイプライン処理を示すタイミングチャートである。

【図 6 5】Compositionバッファ 16、Object Buffer 15、Coded Dataバッファ 13、グラフィクスプレーン 8 における占有量の時間的遷移を示すタイミングチャートである。

【図 6 6】Segmentのロード処理の処理手順を示すフローチャートである。

10

【図 6 7】多重化の一例を示す図である。

【図 6 8】DS10が再生装置のCoded Dataバッファ 13 にロードされる様子を示す図である。

【図 6 9】通常再生が行われる場合を示す図である。

【図 7 0】図 6 9 のように通常再生が行われた場合のDS1,10,20のロードを示す図である。

【図 7 1】Graphicsコントローラ 17 の処理手順のうち、メインルーチンにあたる処理を描いたフローチャートである。

【図 7 2】タイムスタンプによる同期制御の処理手順を示すフローチャートである。

【図 7 3】グラフィクスプレーン 8 の書込処理の処理手順を示すフローチャートである。

20

【図 7 4】デフォルトセレクトボタンのオートアクティベートの処理手順を示すフローチャートである。

【図 7 5】アニメーション表示の処理手順を示すフローチャートである。

【図 7 6】U0処理の処理手順を示すフローチャートである。

【図 7 7】カレントボタンの変更処理の処理手順を示すフローチャートである。

【図 7 8】数値入力処理の処理手順を示すフローチャートである。

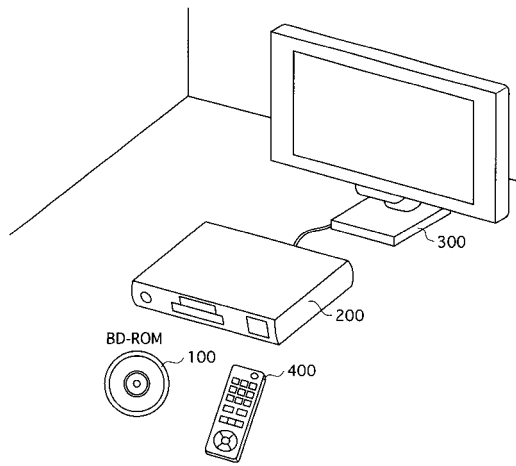
【図 7 9】第 1 実施形態に示したPCSが記録されたBD-ROMを製造するための製造工程を示す図である。

【図 8 0】第 2 実施形態に示したICSが記録されたBD-ROMを製造するための製造工程を示す図である。

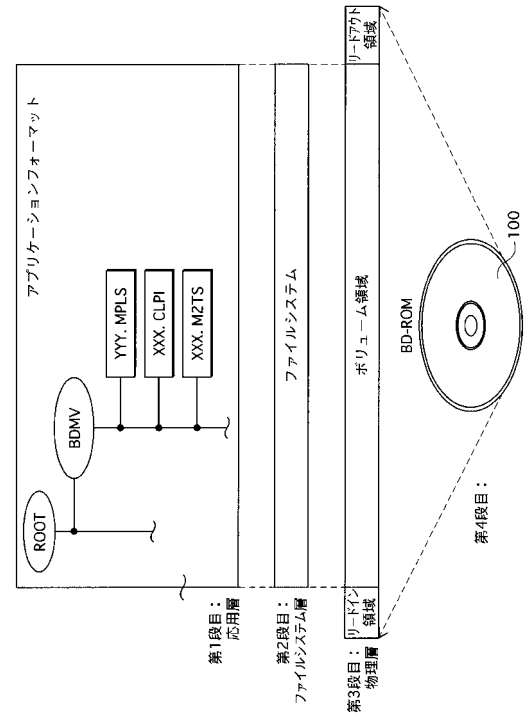
30



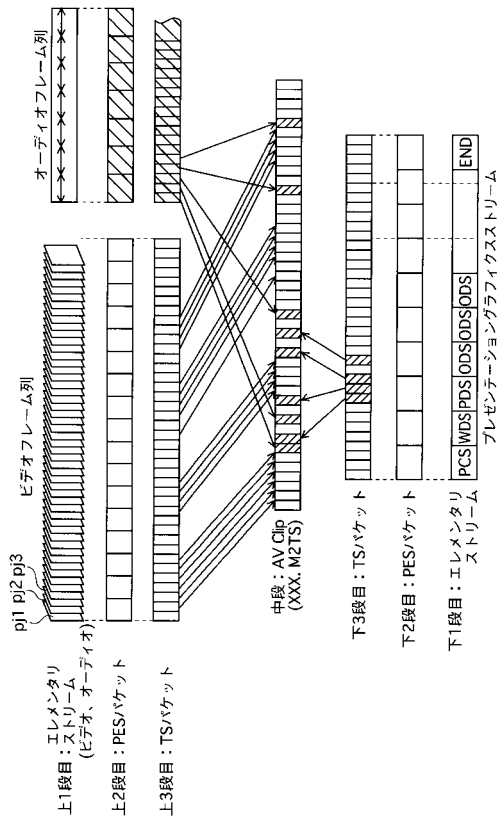
【図 1】



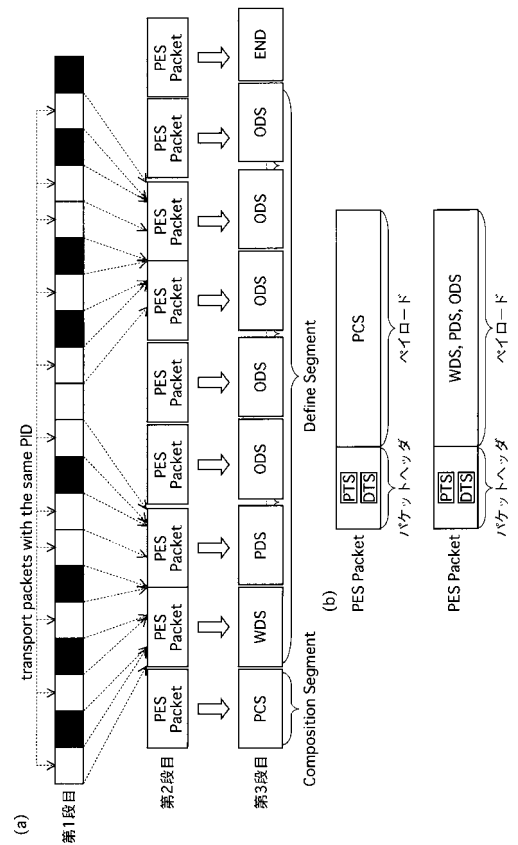
【図 2】



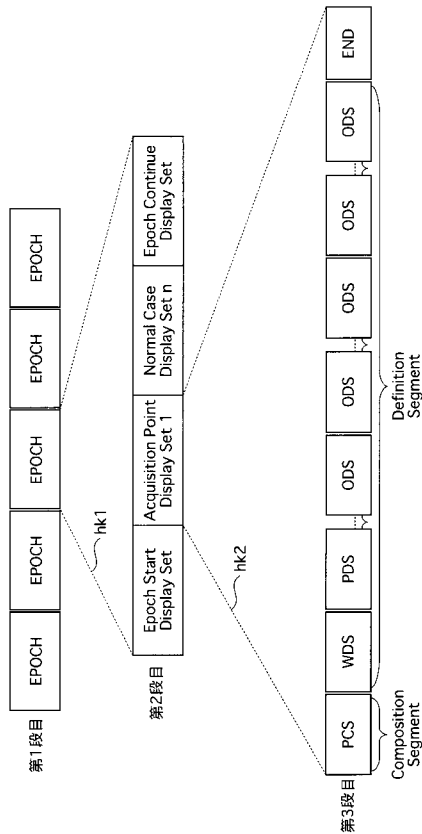
【図 3】



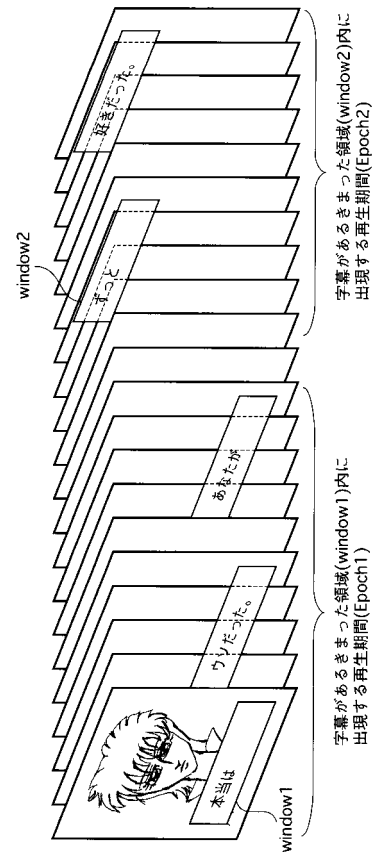
【図 4】



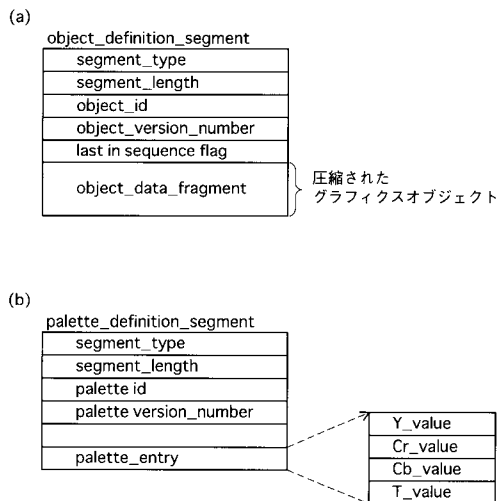
【 図 5 】



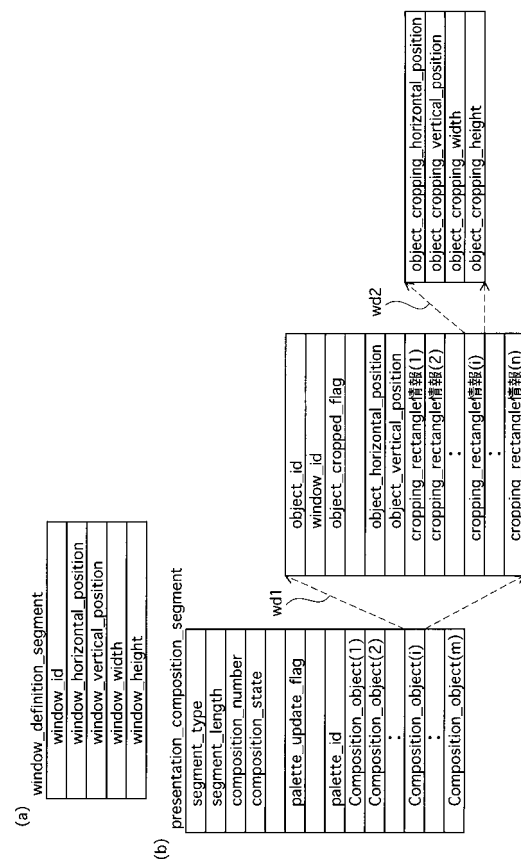
【 図 6 】



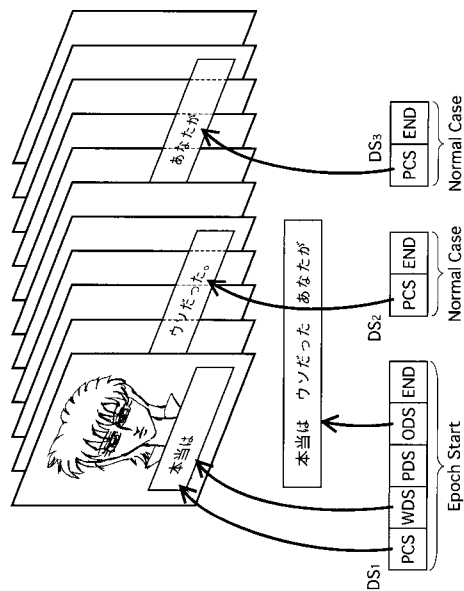
【圖 7】



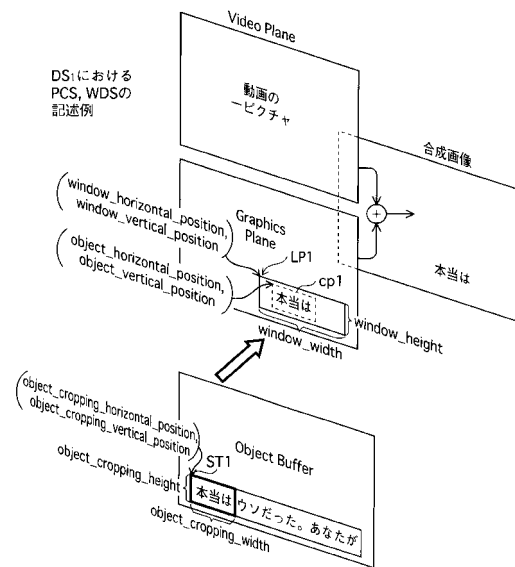
【圖 8】



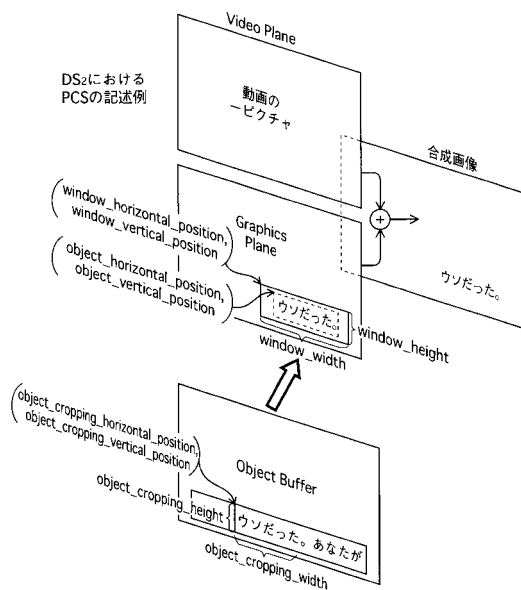
【図 9】



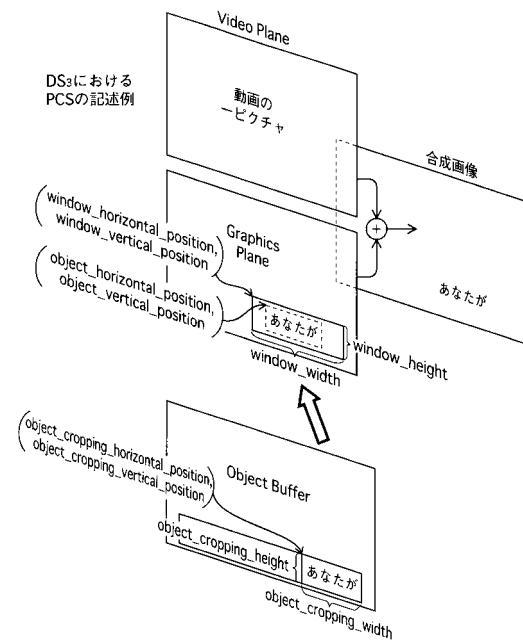
【図 10】



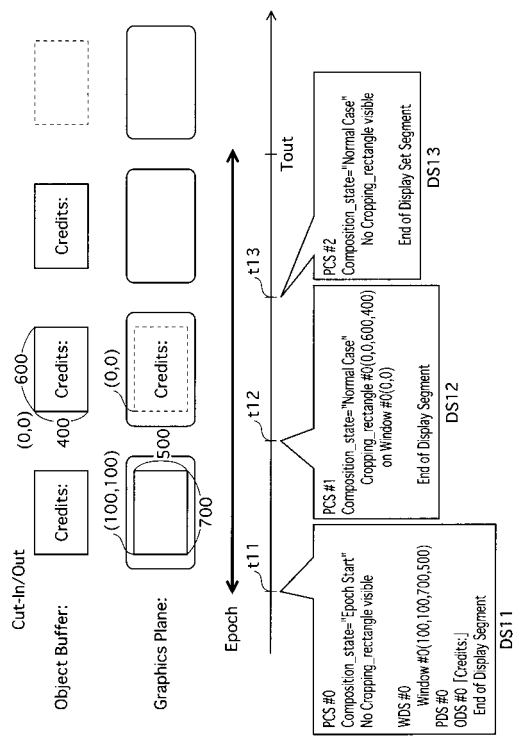
【図 11】



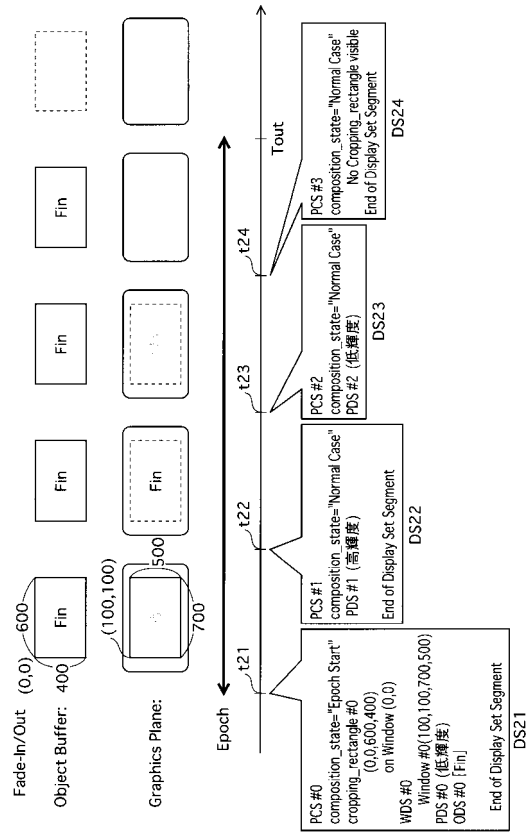
【図 12】



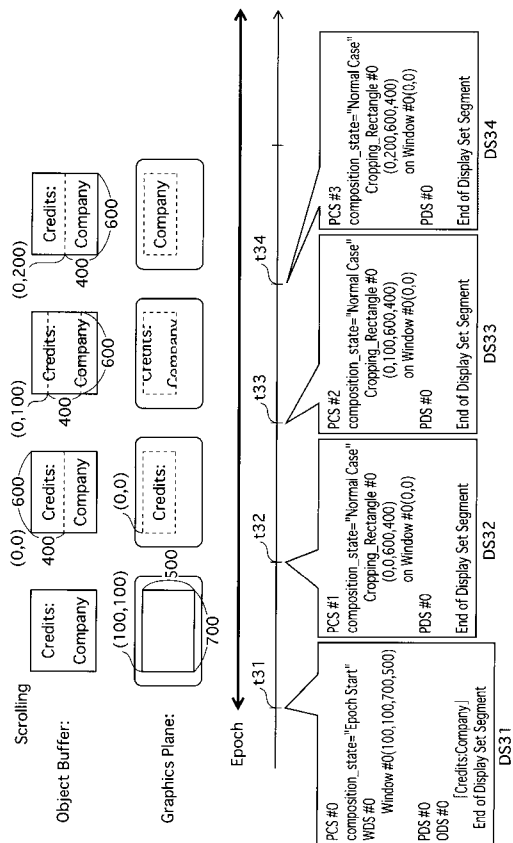
【図 13】



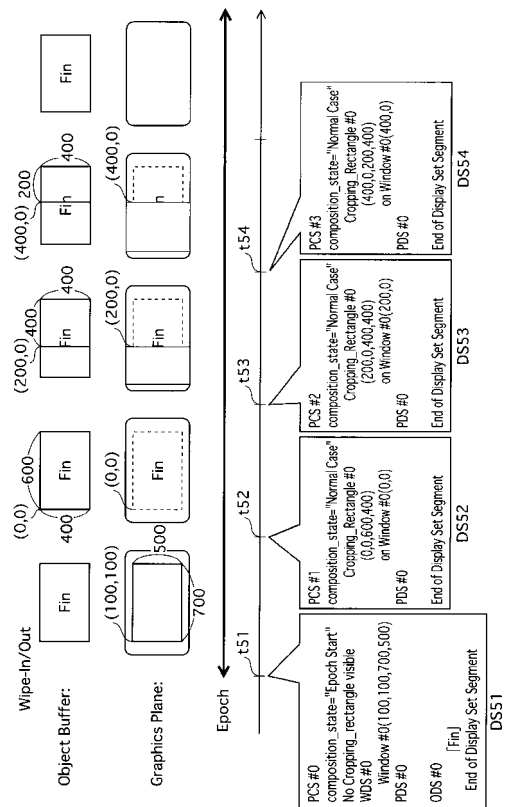
【図 14】



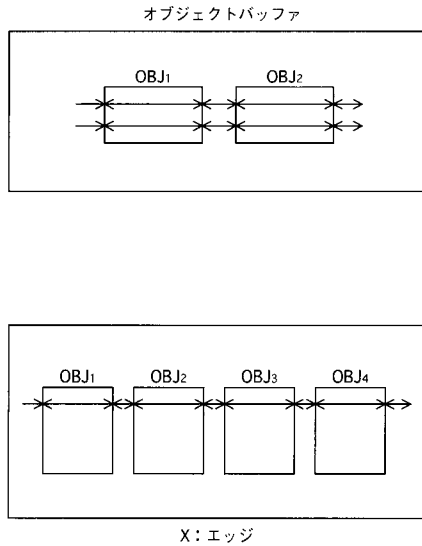
【図 15】



【図 16】



【図 17】



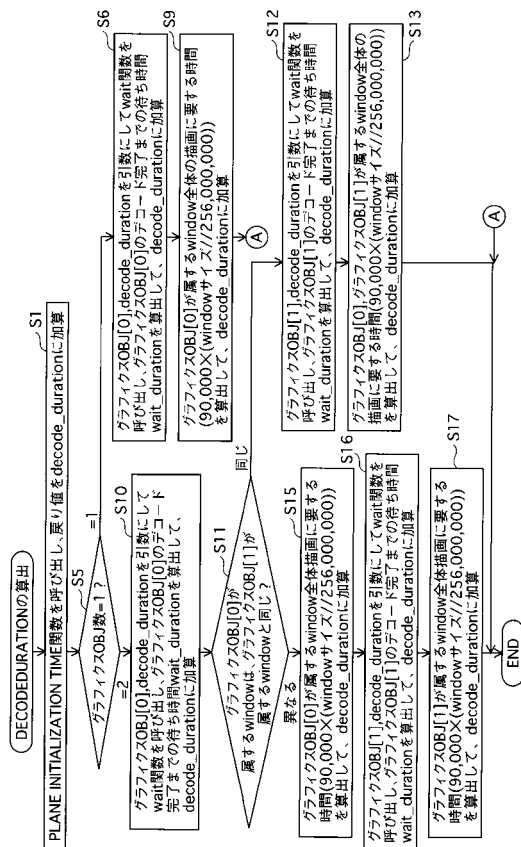
【図 18】

PTS( DSn[PCS]) >= DTS( DSn[PCS]) + DECODEDURATION( DSn )

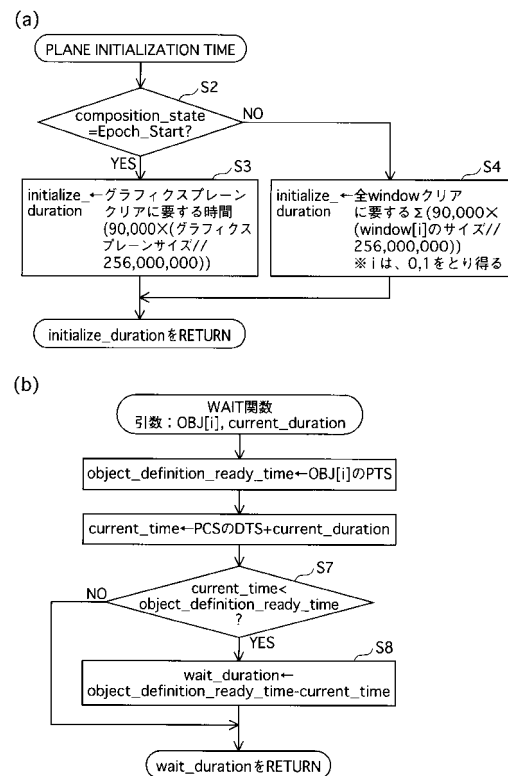
Where:

- DECODEDURATION( DSn ) is calculated as follows:  
 $\text{decode\_duration} = 0;$   
 $\text{decode\_duration} += \text{PLANEINITIALIZATIONTIME}( \text{DSn} );$   
 if( DSn.PCS.num\_of\_objects == 2 )  
 {  
 $\text{decode\_duration} += \text{WAIT}( \text{DSn}, \text{DSn.PCS.OBJ}[0], \text{decode\_duration} );$   
 if( DSn.PCS.OBJ[0].window\_id == DSn.PCS.OBJ[1].window\_id )  
 {  
 $\text{decode\_duration} += \text{WAIT}( \text{DSn}, \text{DSn.PCS.OBJ}[1], \text{decode\_duration} );$   
 $\text{decode\_duration} += 90000 * (\text{SIZE}( \text{DSn.PCS.OBJ}[0].window\_id) // 256 * 10^6 );$   
 }  
 }  
 else  
 {  
 $\text{decode\_duration} += 90000 * (\text{SIZE}( \text{DSn.PCS.OBJ}[0].window\_id) // 256 * 10^6 );$   
 $\text{decode\_duration} += \text{WAIT}( \text{DSn}, \text{DSn.PCS.OBJ}[1], \text{decode\_duration} );$   
 $\text{decode\_duration} += 90000 * (\text{SIZE}( \text{DSn.PCS.OBJ}[1].window\_id) // 256 * 10^6 );$   
 }  
 }  
 else if( DSn.PCS.num\_of\_objects == 1 )  
 {  
 $\text{decode\_duration} += \text{WAIT}( \text{DSn}, \text{DSn.PCS.OBJ}[0], \text{decode\_duration} );$   
 $\text{decode\_duration} += 90000 * (\text{SIZE}( \text{DSn.PCS.OBJ}[0].window\_id) // 256 * 10^6 );$   
 }  
 return decode\_duration ;
- PLANEINITIALIZATIONTIME( DSn ) is calculated as follows:  
 $\text{initialize\_duration} = 0;$   
 if( DSn.PCS.composition\_state == EPOCH\_START )  
 {  
 $\text{initialize\_duration} = 90000 * ( 8 * \text{video\_width} * \text{video\_height} // 256 * 10^6 );$   
 }  
 else  
 {  
 for( i=0; i < WDS.num\_windows; i++ )  
 {  
 if( EMPTY( DSn.WDS.WIN[i], DSn ) )  
 {  
 $\text{initialize\_duration} += 90000 * (\text{SIZE}( \text{DSn.WDS.WIN}[i] ) // 256 * 10^6 );$   
 }  
 }  
 }  
 return initialize\_duration ;
- WAIT( DSn, OBJ, current\_duration ) is calculated as follows:  
 $\text{wait\_duration} = 0;$   
 if( EXISTS( OBJ.object\_id, DSn ) )  
 {  
 $\text{object\_definition\_ready\_time} = \text{PTS}( \text{GET}( \text{OBJ.object\_id}, \text{DSn} ) );$   
 $\text{current\_time} = \text{DTS}( \text{DSn.PCS} ) + \text{current\_duration};$   
 if( current\_time < object\_definition\_ready\_time )  
 {  
 $\text{wait\_duration} += \text{object\_definition\_ready\_time} - \text{current\_time};$   
 }  
 }  
 return wait\_duration ;

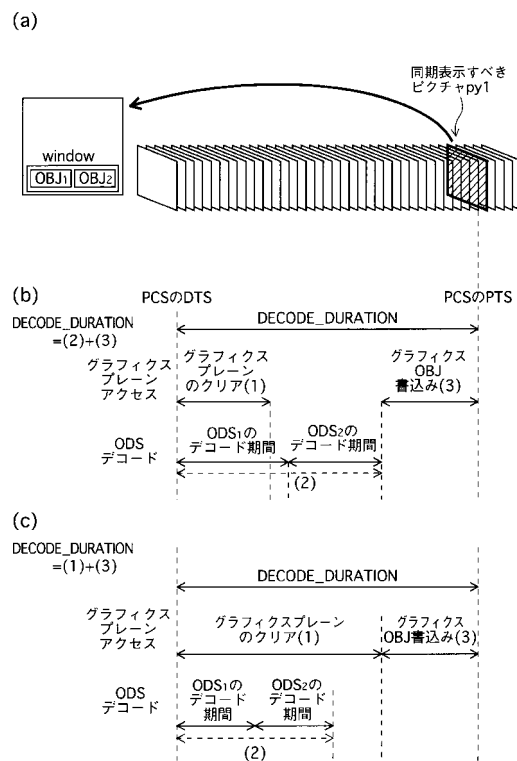
【図 19】



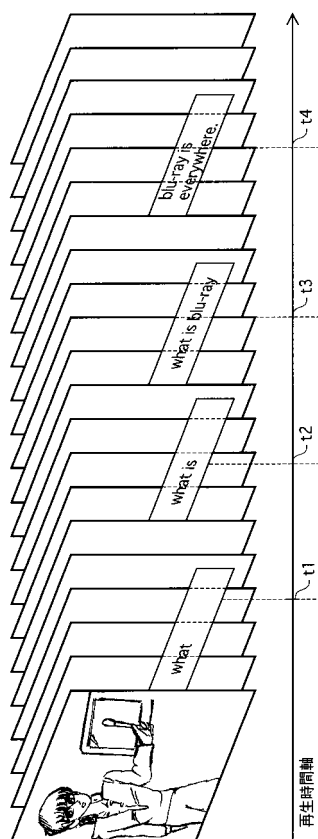
【図 20】



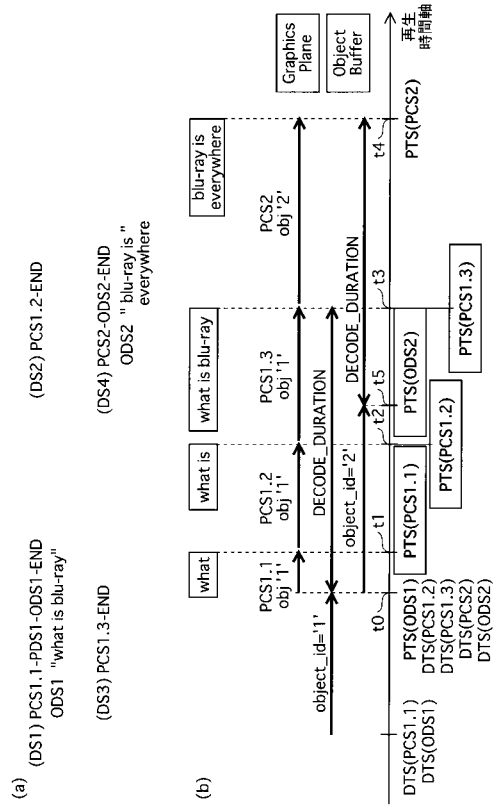
【 ㊤ 2 2 】



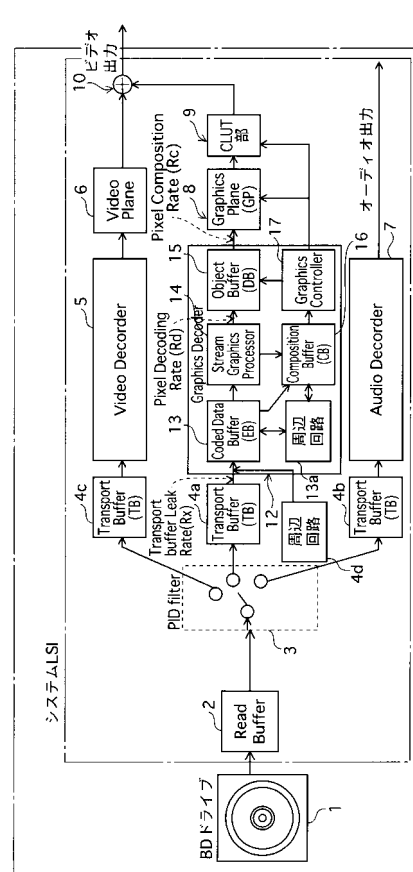
【 図 2 4 】



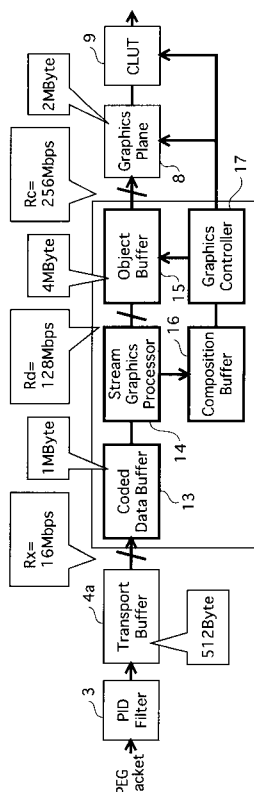
【図 25】



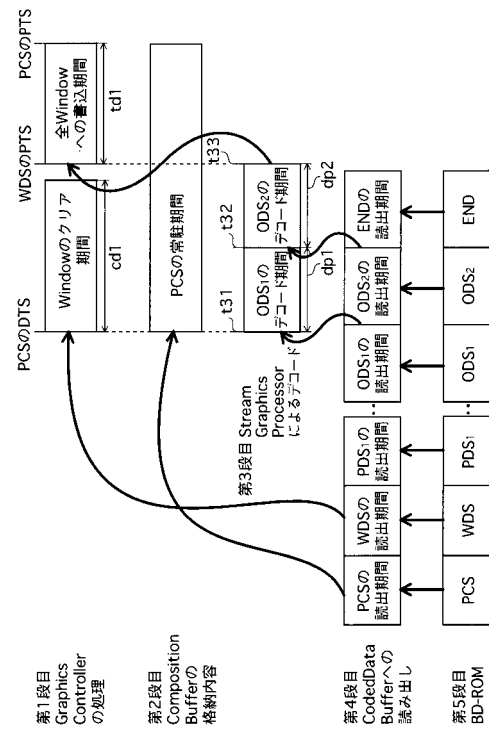
【図 26】



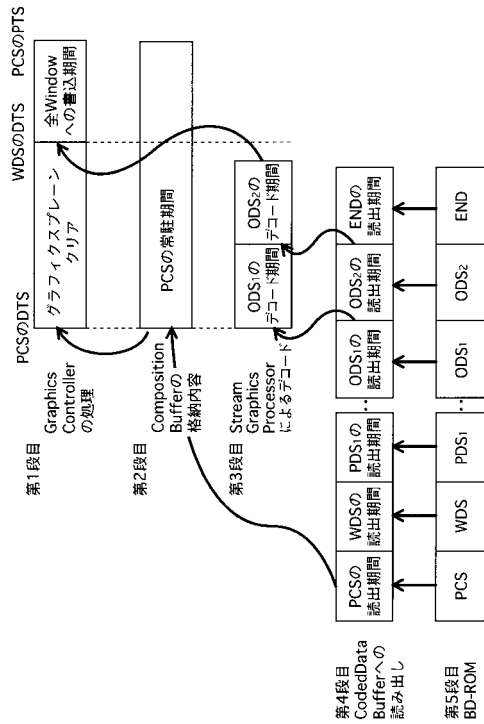
【図 27】



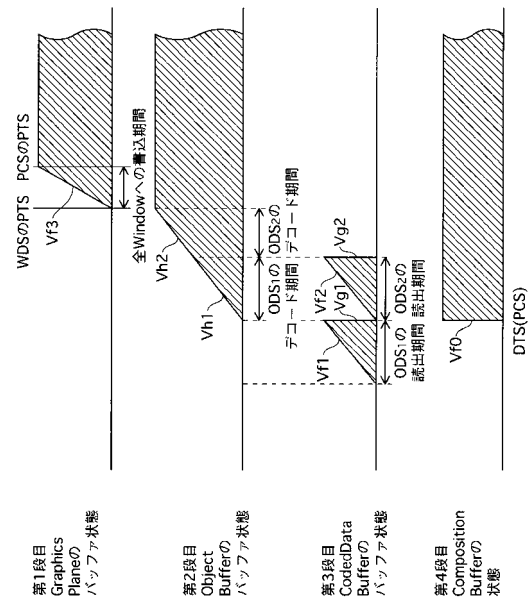
【図 28】



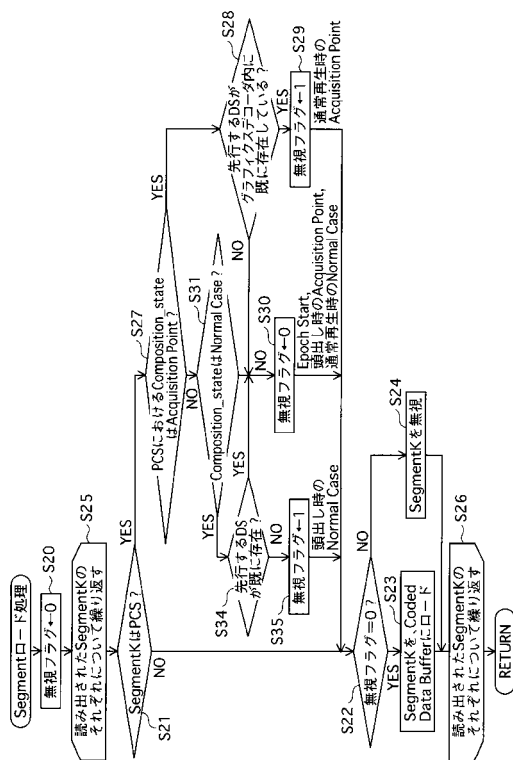
【 図 2 9 】



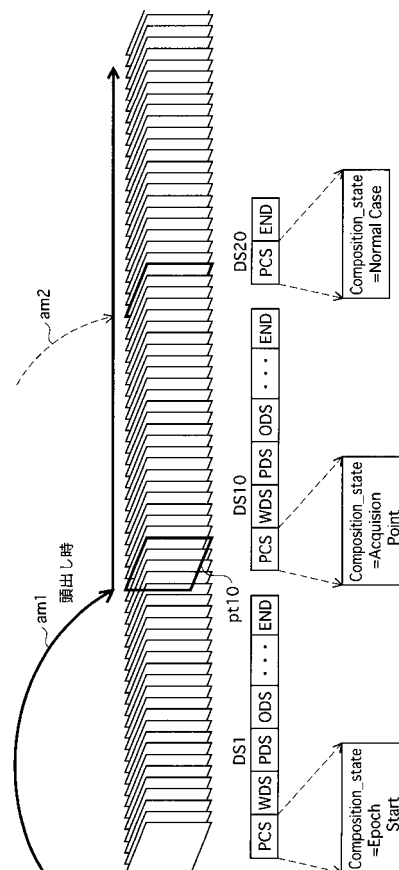
【 図 3 0 】



【 図 3 1 】

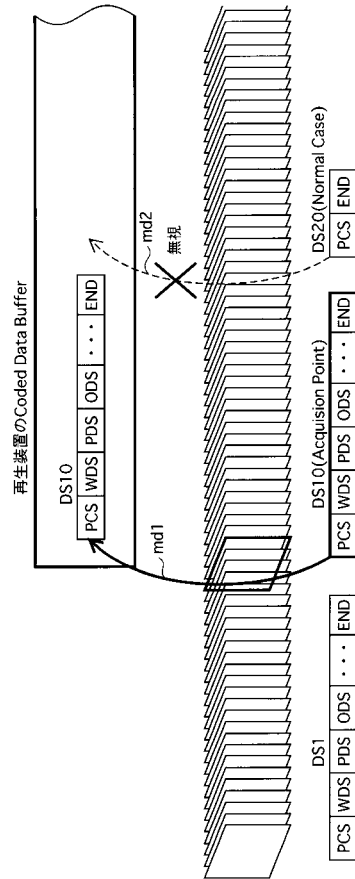


【 図 3 2 】

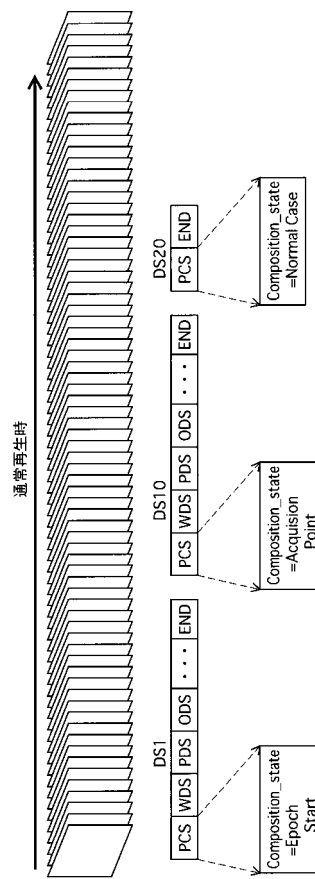




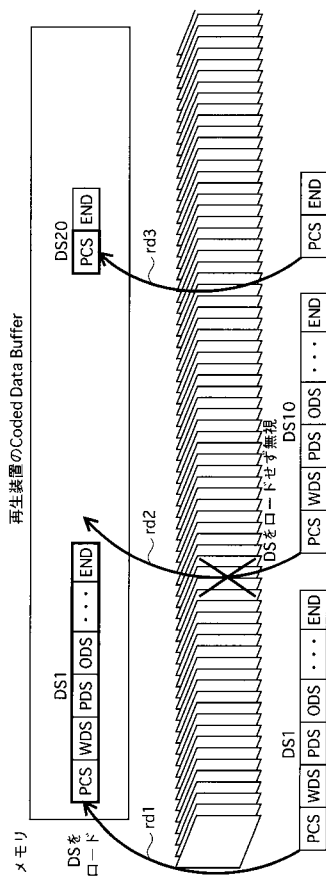
【 図 3 3 】



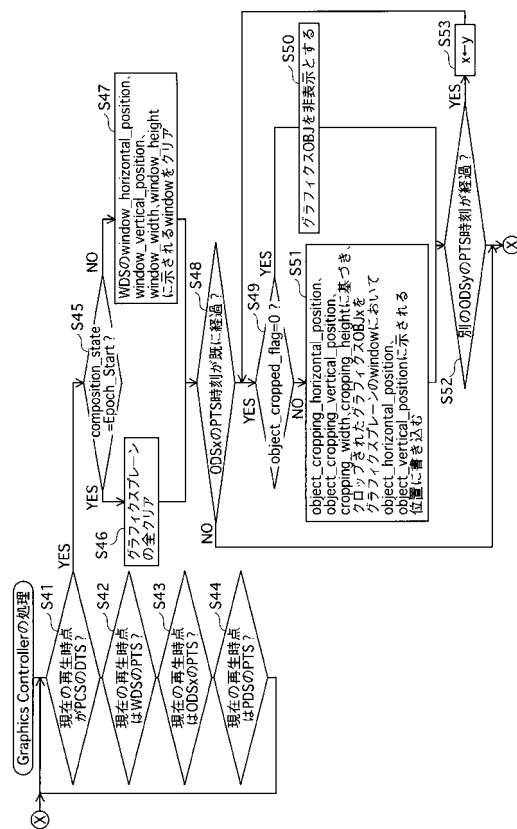
【 図 3 4 】



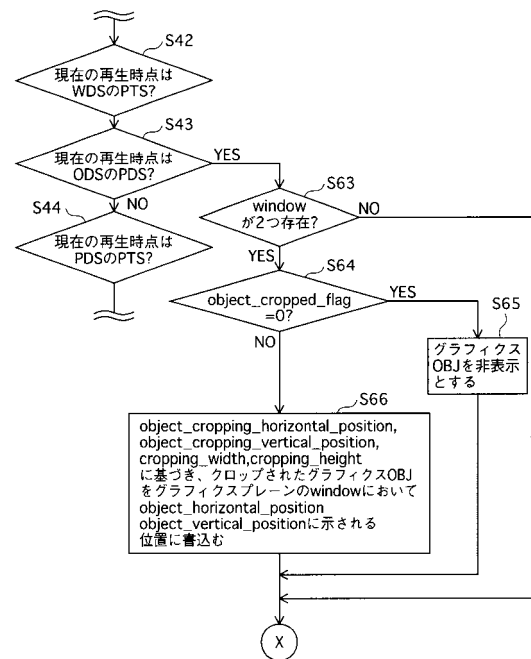
【 図 3 5 】



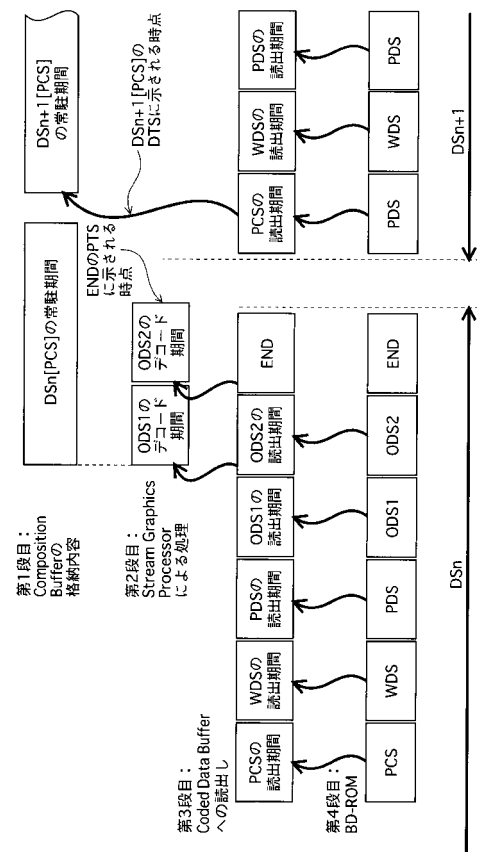
【 図 3 6 】



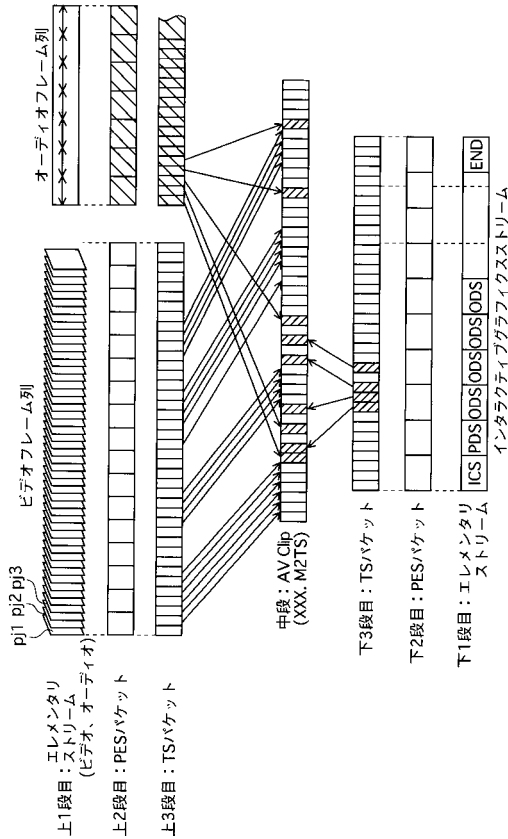
【 図 3 8 】



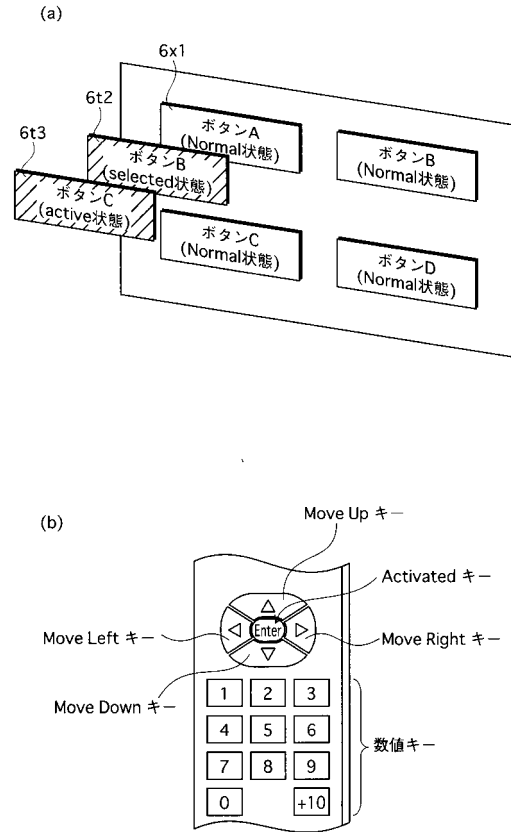
【 図 4 0 】



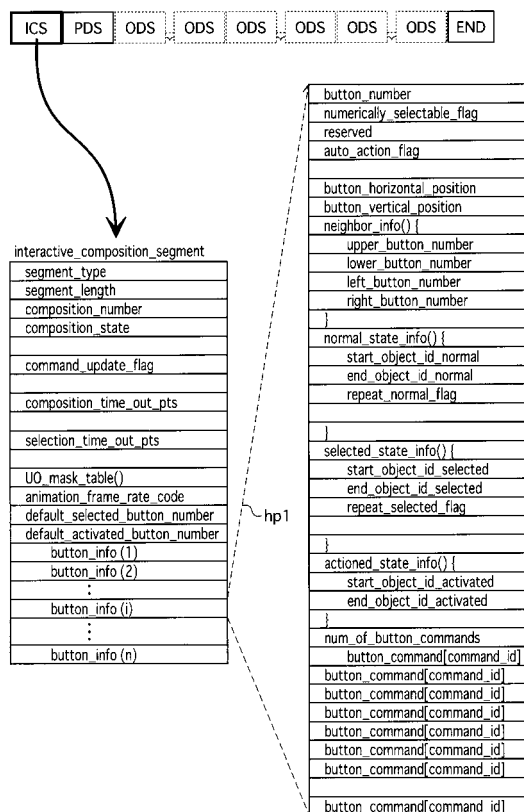
【図 4 1】



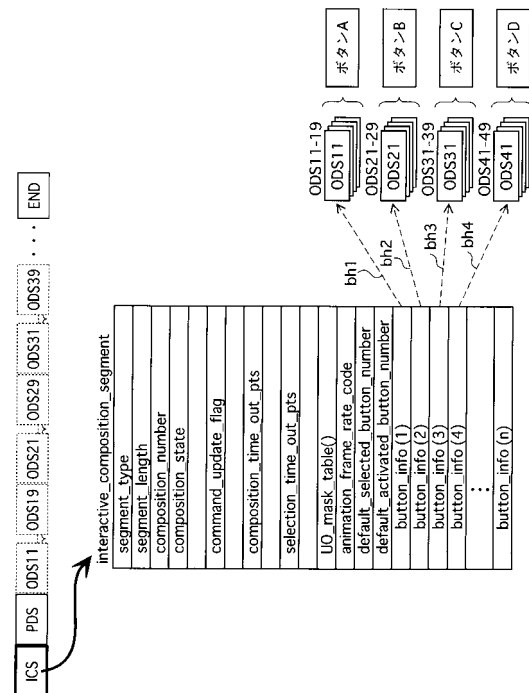
【図 4 2】



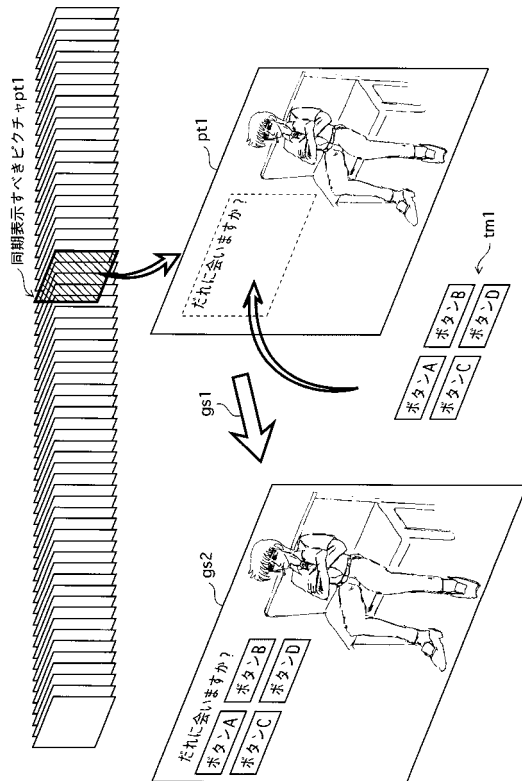
【図 4 3】



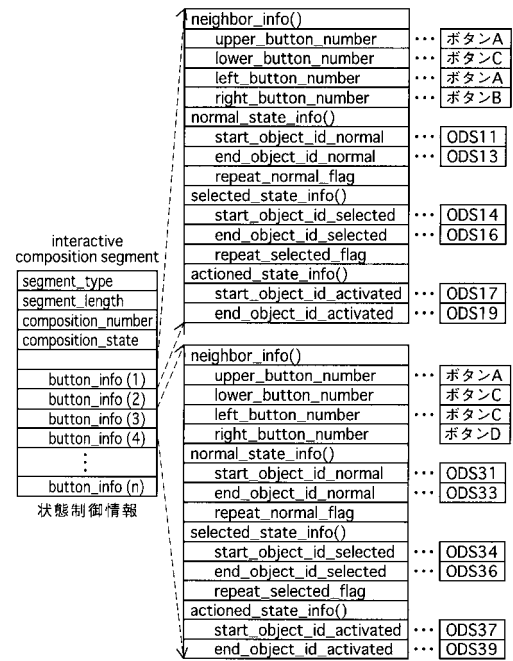
【図 4 4】



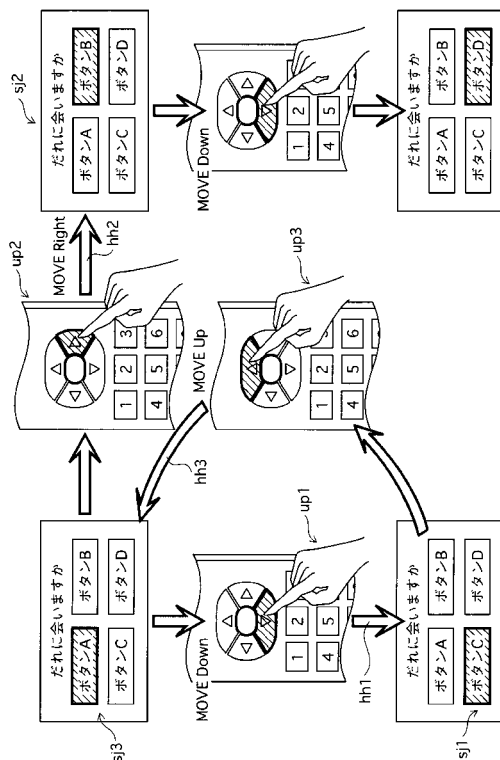
【図 45】



【図 46】



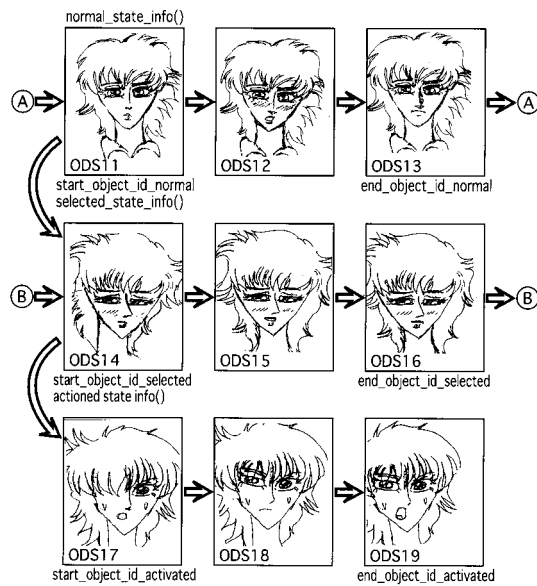
【図 47】



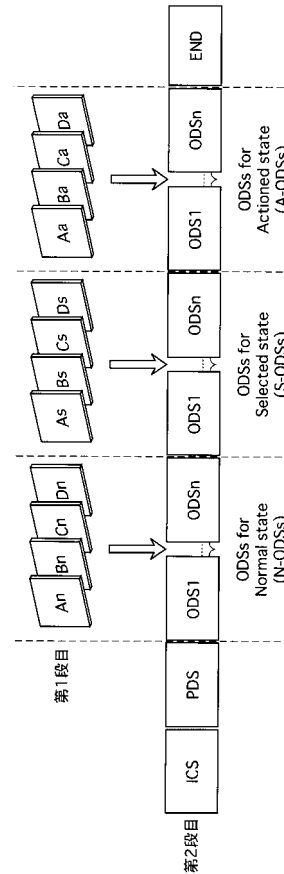
【図 48】



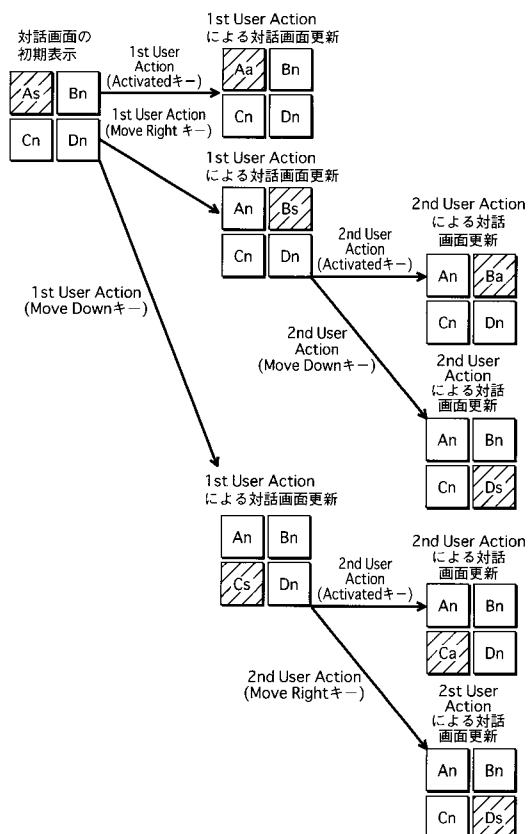
【図 49】



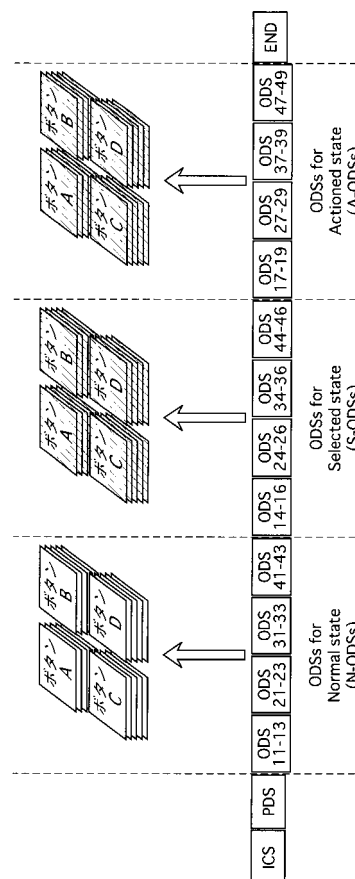
【図 50】



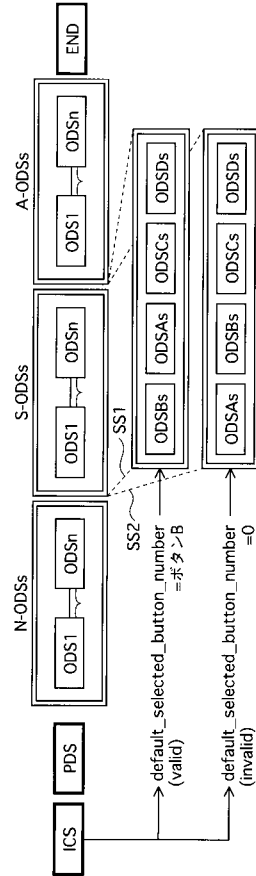
【図 51】



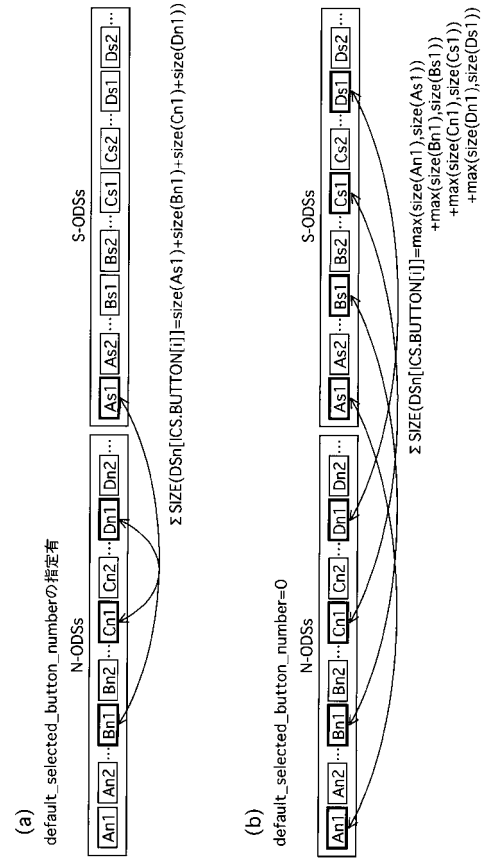
【図 52】



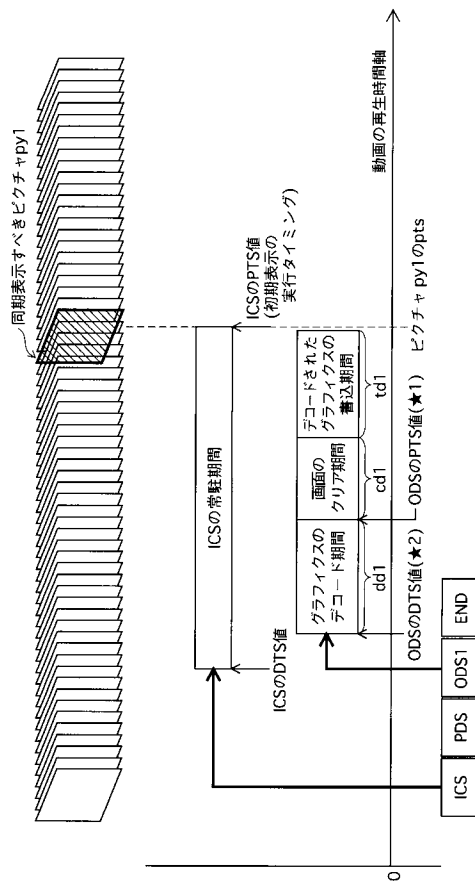
【図 5 3】



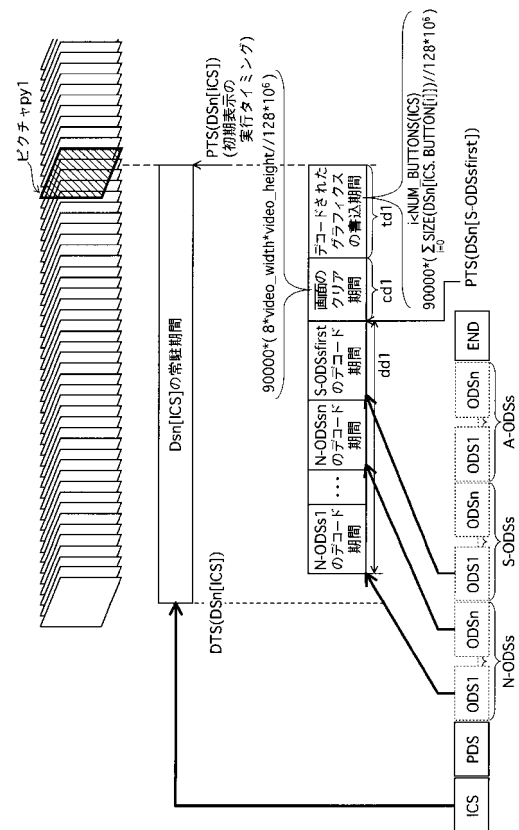
【図 5 4】



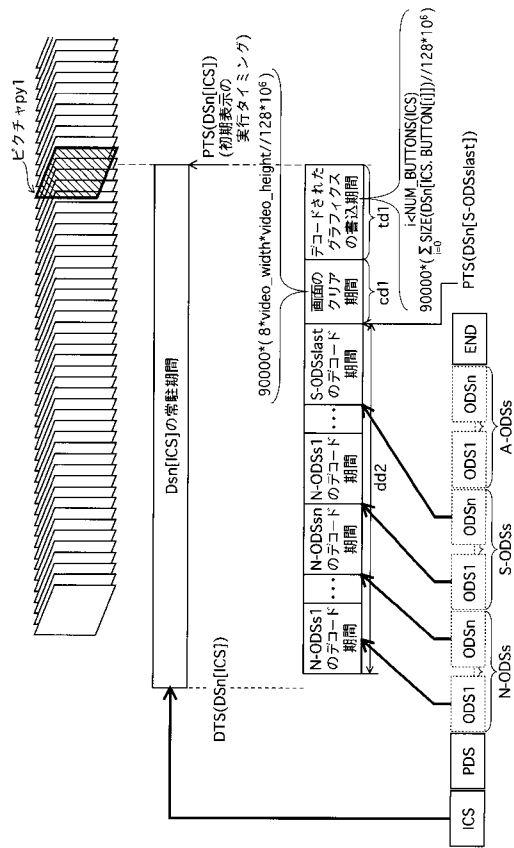
【図 5 5】



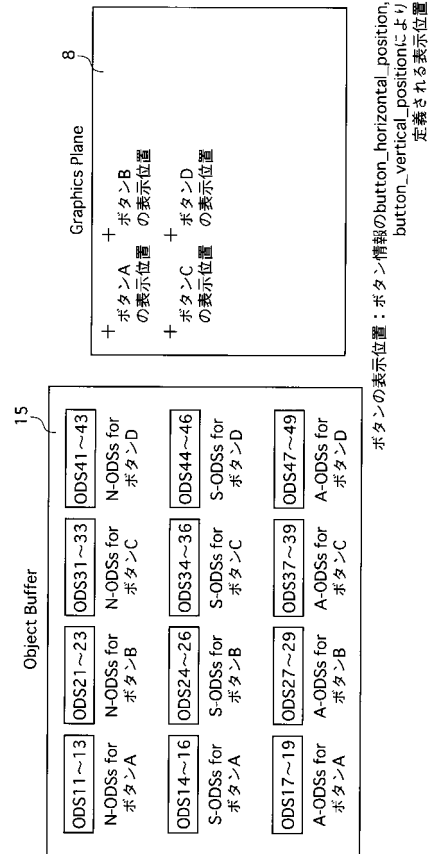
【図 5 6】



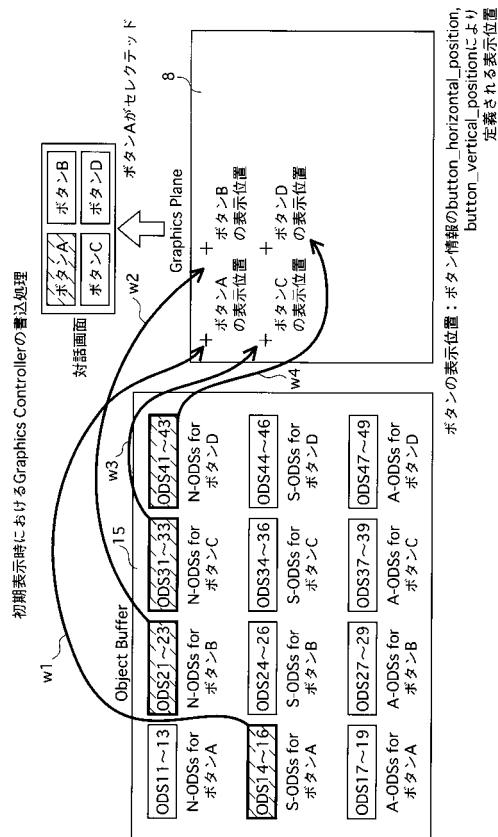
【 図 5 7 】



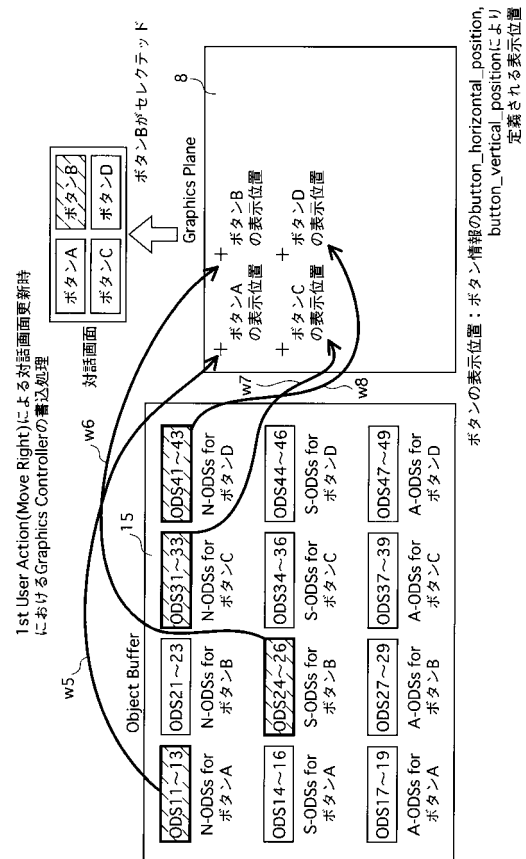
【 図 5 8 】



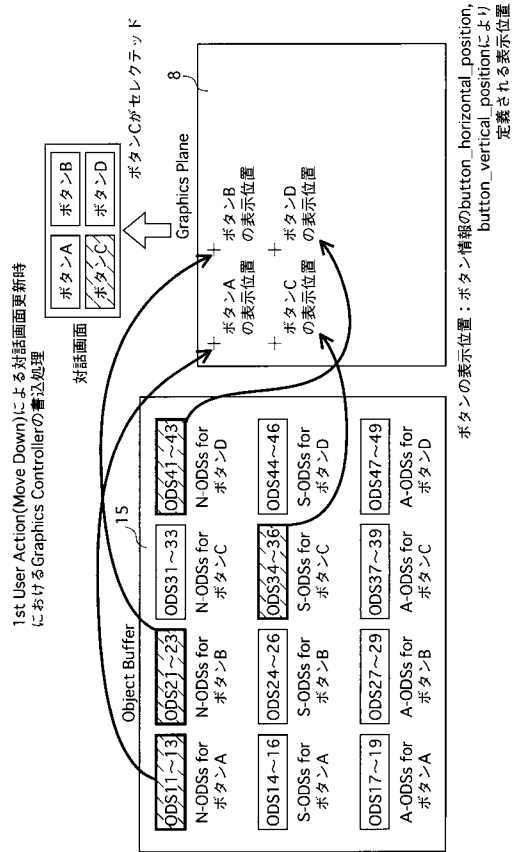
【 図 5 9 】



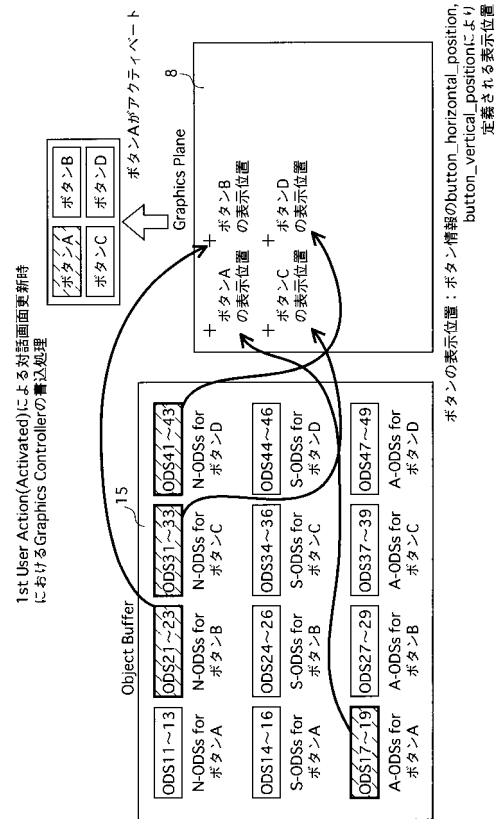
【 図 6 0 】



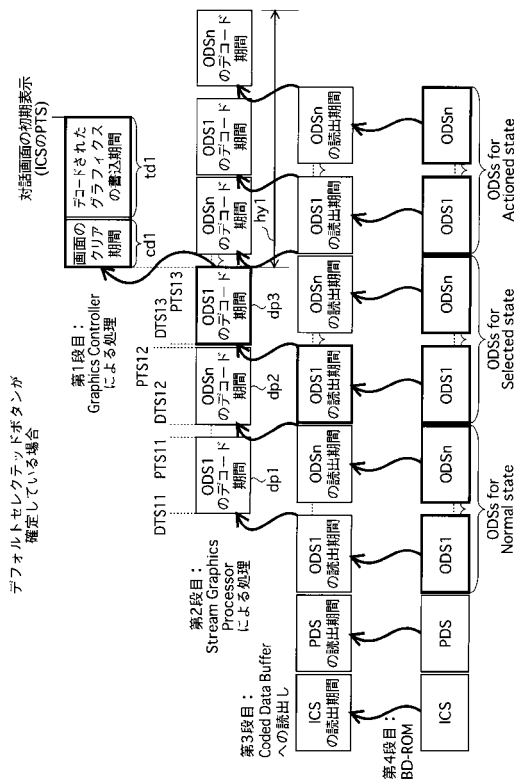
【 図 6 1 】



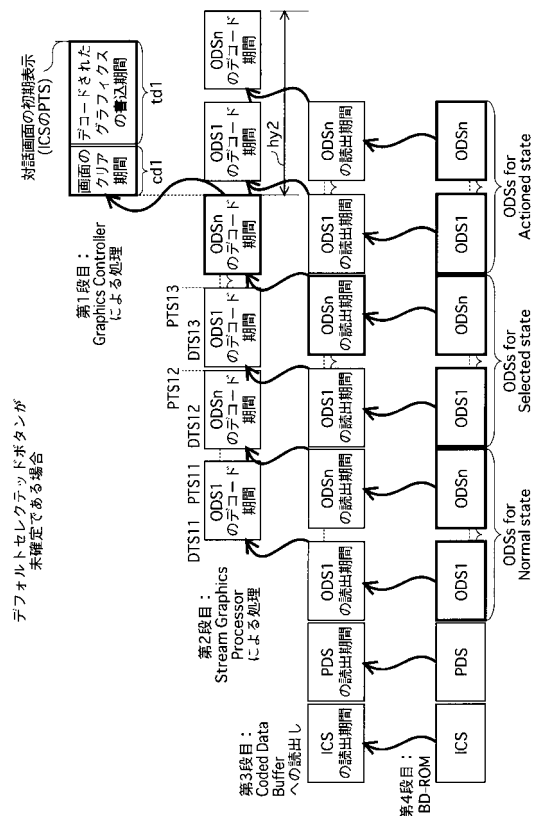
【 図 6 2 】



【 図 6 3 】

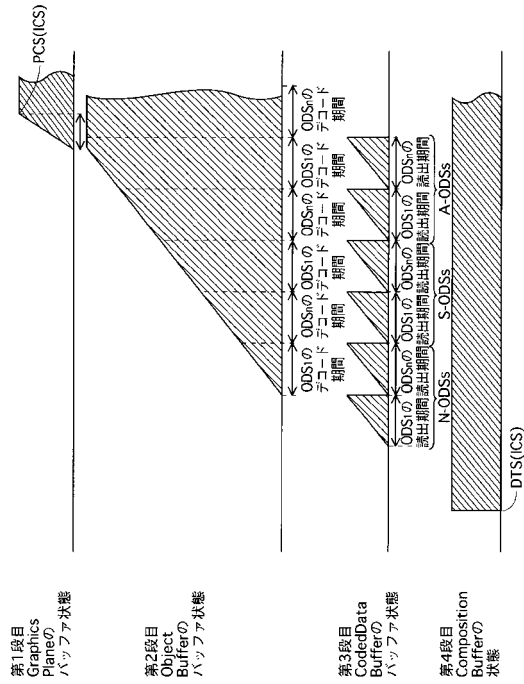


【 図 6 4 】

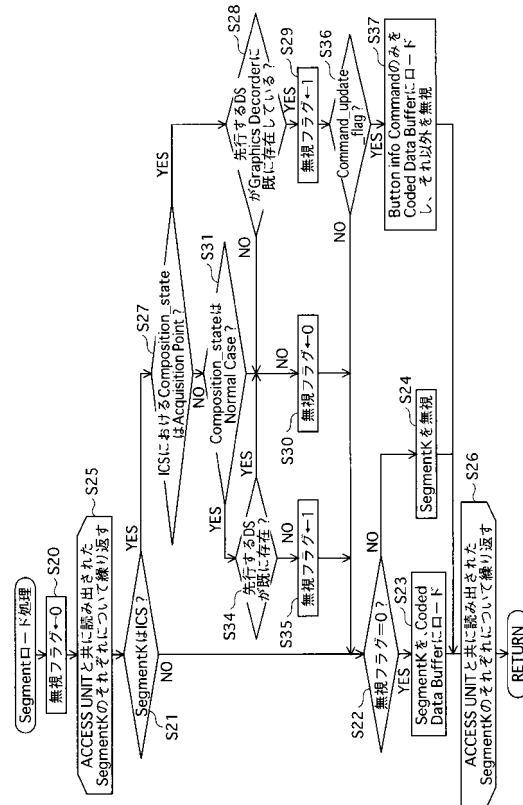




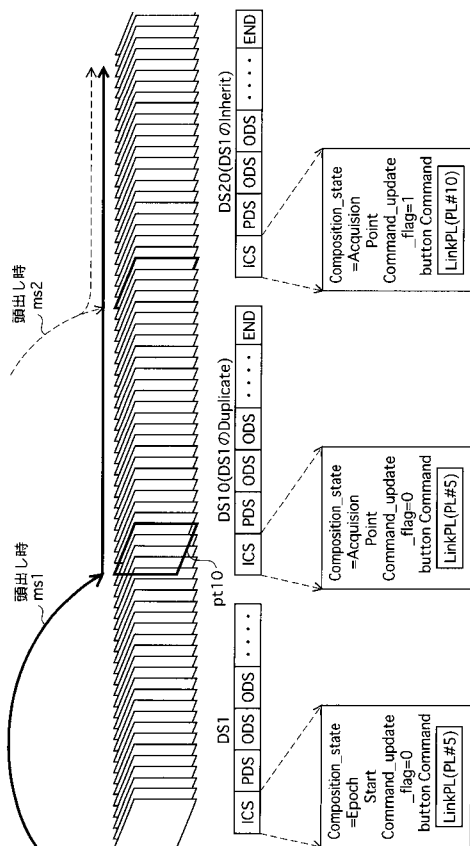
【図 65】



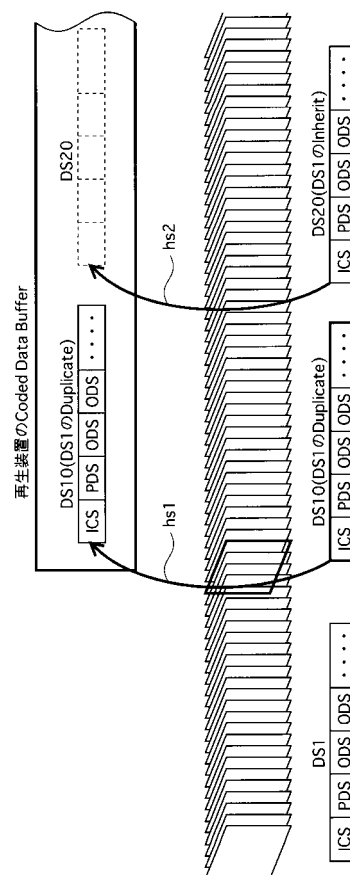
【図 66】



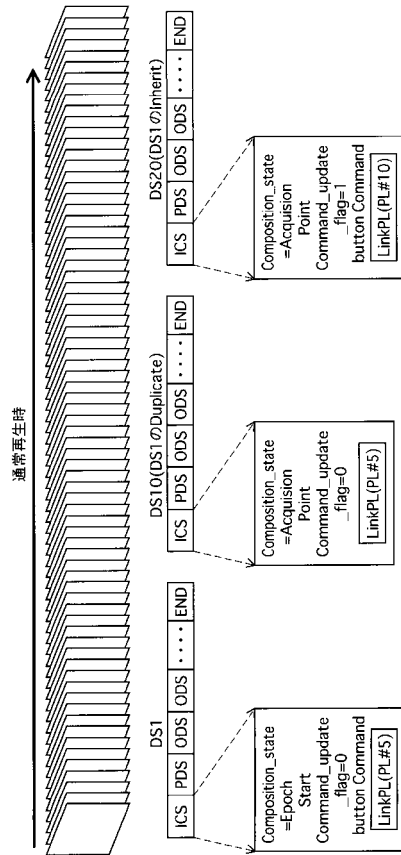
【図 67】



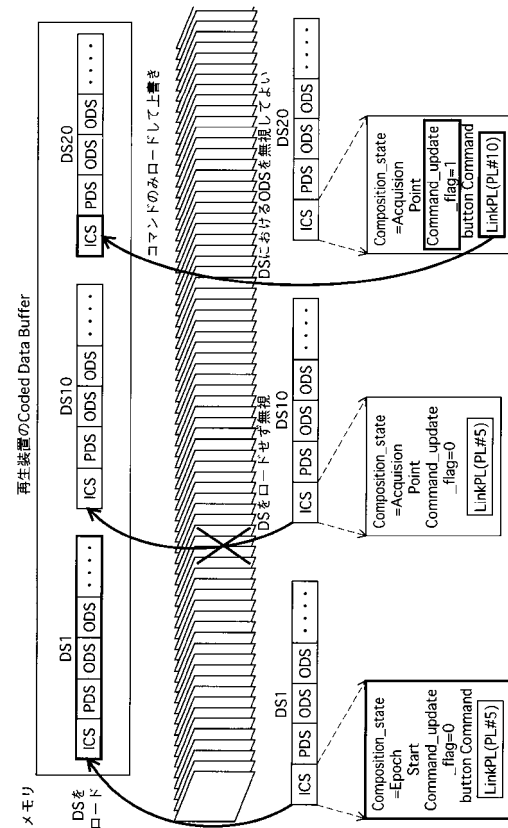
【図 68】



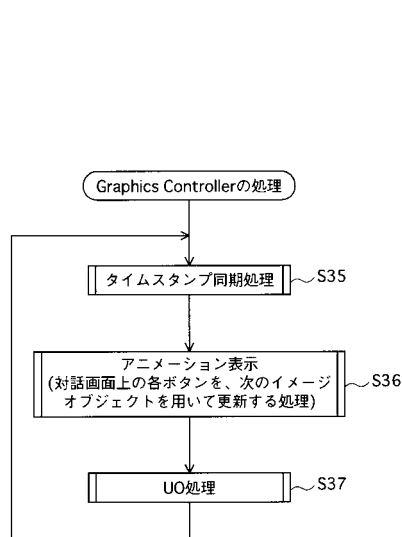
【図 69】



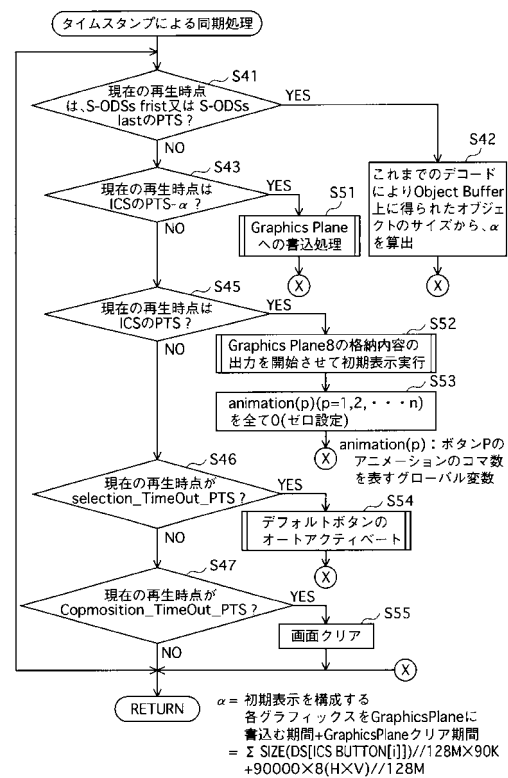
【図 70】



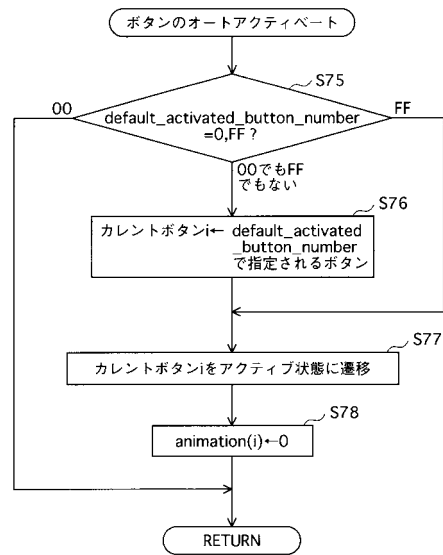
【図 71】



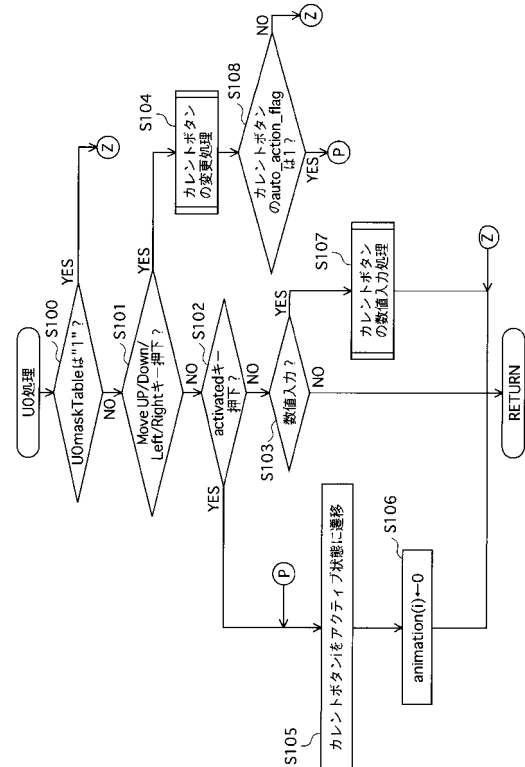
【図 72】



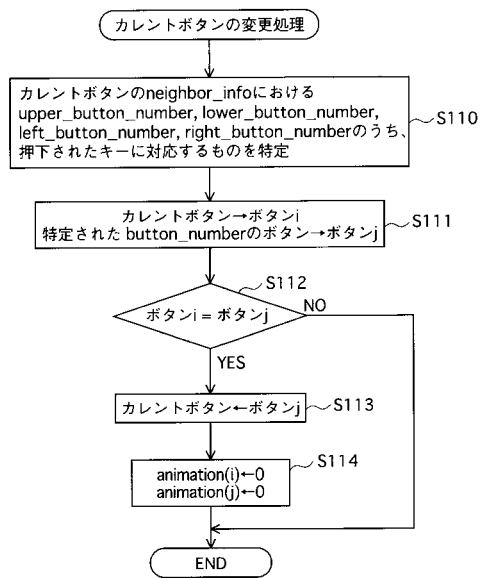
【 図 7 4 】



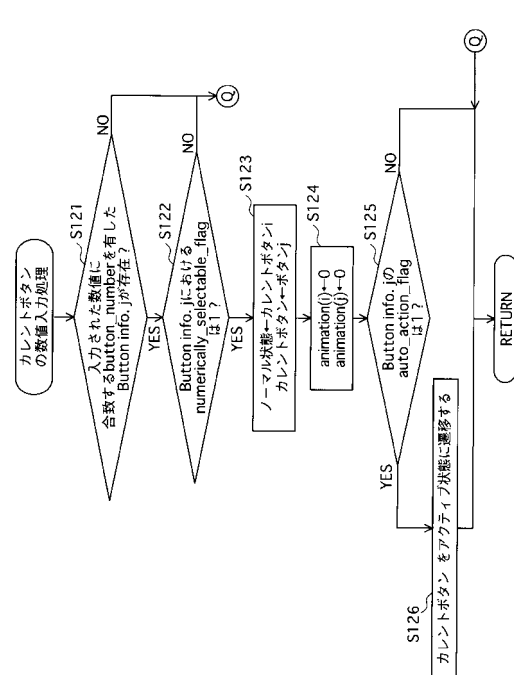
【 図 7 6 】



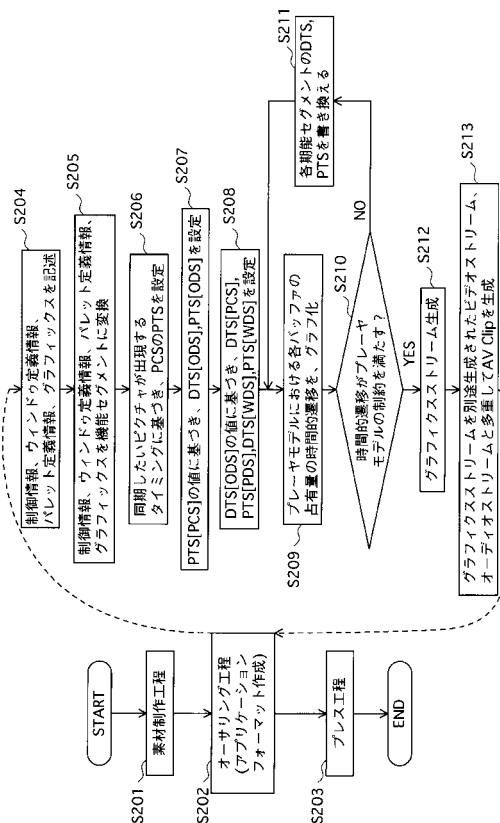
【図 77】



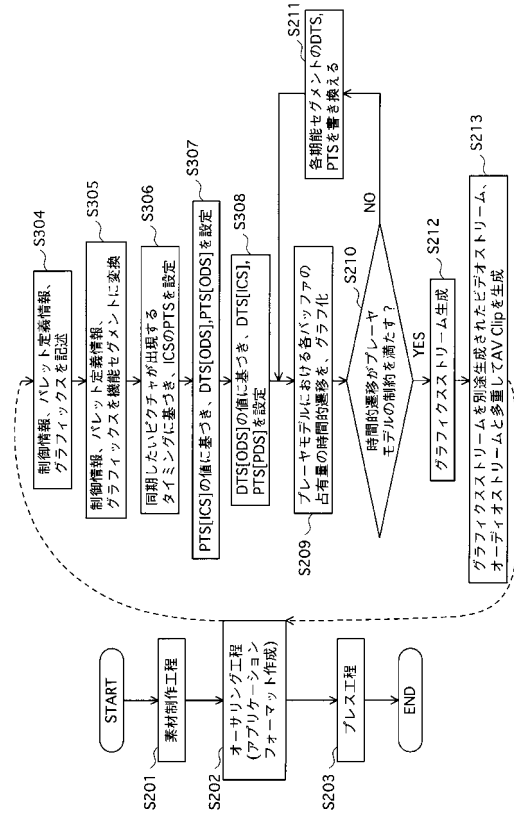
【図 78】



【図 79】



【図 80】



---

フロントページの続き

(51) Int.Cl. F I  
**H 0 4 N 5/85 (2006.01)** H 0 4 N 5/91 D  
H 0 4 N 5/85 Z

審査官 井上 正

(56) 参考文献 特開 2 0 0 3 - 1 1 6 1 0 0 ( J P , A )  
特開平 0 7 - 2 8 8 7 7 6 ( J P , A )  
特開平 0 6 - 1 2 1 2 6 3 ( J P , A )

(58) 調査した分野(Int.Cl. , D B 名)  
G11B 20/10-20/16、27/00-27/06、  
H04N 5/76- 5/956