

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公开说明书

[21] 申请号 200480011555.0

[51] Int. Cl.

G11B 27/10 (2006.01)
G11B 27/034 (2006.01)
G11B 27/036 (2006.01)
G11B 27/038 (2006.01)
G11B 27/34 (2006.01)
G11B 20/10 (2006.01)

[43] 公开日 2006年5月31日

[11] 公开号 CN 1781153A

[51] Int. Cl. (续)

G11B 20/12 (2006.01)
G11B 20/14 (2006.01)
H04N 5/00 (2006.01)
G06F 3/033 (2006.01)

[22] 申请日 2004.4.27

[21] 申请号 200480011555.0

[30] 优先权

[32] 2003.4.28 [33] US [31] 60/465,972

[86] 国际申请 PCT/JP2004/006074 2004.4.27

[87] 国际公布 WO2004/098193 英 2004.11.11

[85] 进入国家阶段日期 2005.10.28

[71] 申请人 松下电器产业株式会社

地址 日本大阪府

[72] 发明人 约瑟夫·麦克罗森 冈田智之

小川智辉

[74] 专利代理机构 永新专利商标代理有限公司

代理人 王 英

权利要求书 5 页 说明书 46 页 附图 42 页

[54] 发明名称

记录介质和方法、再现装置和方法、程序和集成电路

[57] 摘要

记录介质储存通过多路复用视频流和图形流构成的 AVClip。图形流呈现由多个图像构成的运动图像，并且图形流包括再现将要与图像组合的图形的图形数据。图形流还包括窗口信息(WDS)，所述窗口信息规定用于呈现图形的窗口，并表示平面上的窗口的宽度、高度和位置，所述平面是将图形与图像组合起来的再现装置的平面存储器。

window_definition_segment

window_id
window_horizontal_position
window_vertical_position
window_width
window_height

1、一种用于储存数据的记录介质，所述记录介质包括：

通过多路复用视频流和图形流构成的数字流，其中所述视频流代表由多个图像构成的运动图像，该图形流包括：

表示将要与所述图像组合的图形的图形数据；和

规定在其中呈现图形的窗口的窗口信息，所述窗口信息表示平面上的窗口的宽度、高度和位置，该平面是将图形与图像组合起来的再现装置的平面存储器。

2、根据权利要求1的记录介质，其中窗口的宽度和高度设置成使该窗口的尺寸是该平面的 $1/x$ ，该平面对应每个图像的尺寸， x 是以窗口更新速度和图像显示速度之间的比例为基础的实数。

3、根据权利要求1的记录介质，其中：

该图形流包括含有裁剪信息的控制信息，所述裁剪信息规定在通过解码该图形数据获得的图形目标内的裁剪框；和

要在该窗口中呈现的图形是裁剪框内的图形目标的一部分。

4、根据权利要求3的记录介质，其中控制信息含有位置信息，所述位置信息规定在该窗口中呈现该裁剪框内的图形目标部分的位置。

5、根据权利要求4的记录介质，其中：

该图形流包括用于实现滚动、划入、划出、切入、切出显示效果之一的多条控制信息；和

该控制信息中的每条信息包括分别规定不同裁剪框和位置的裁剪信息和位置信息。

6、一种用于再现数字流的再现装置，所述数字流是通过多路复用视频流和图形流构成的，所述再现装置包括：

视频解码器，可操作地对视频流进行解码，以便获得由多个图像构成的运动图像；

图形解码器，可操作地呈现图形，以使该图形与所述图像同步地显示；和

平面存储器，其对应一个平面，并用于将所述图形与所述图像组合起来，其中：

该图形流包括窗口信息，该窗口信息规定所述平面的一部分作为窗口以在其中呈现所述图形；和

通过所述图形解码器呈现所述图形包括在所述平面存储器中的窗口中清除图形，并将图形写到所述平面存储器中的该窗口中。

7、根据权利要求6的再现装置，其中：

所述图形流包括压缩图形数据；和

所述图形解码器包括处理器和控制单元，该处理器可操作地对压缩图形数据进行解码，所述控制单元可操作地进行所述清除操作和所述写操作。

8、根据权利要求7的再现装置，其中：

该窗口的尺寸设置为所述平面的 $1/x$ ， x 是以窗口更新速度和视频流的显示速度之间的比例为基础的实数；和

以基于该窗口的更新速度和窗口的尺寸的传送速度进行由所述控制单元执行的所述写操作。

9、根据权利要求7的再现装置，其中：

所述图形解码器包括目标缓冲器，其可操作地储存由所述处理器解码的解压缩图形数据；

该图形流包括控制信息，该控制信息包括裁剪信息，该裁剪信息规定通过解码目标缓冲器中的该图形数据获得的图形目标内的裁剪框；

所述控制单元是可操作地裁剪在裁剪框内的一部分图形目标；并

将要与所述图像同步地显示的该图形是该裁剪框内的一部分图形目标。

10、根据权利要求 9 的再现装置，其中：

该控制信息含有位置信息，所述位置信息规定在所述窗口内呈现该裁剪框内的部分的位置；和

裁剪框内的该部分被写到由该位置信息规定的该位置上的窗口内。

11、根据权利要求 10 的再现装置，其中：

该图形流包括多条控制信息；

分别由在每条该控制信息中的裁剪信息和位置信息表示的裁剪部分和位置是不同的；并且

所述控制单元可操作以便根据每条控制信息中的裁剪信息和位置信息，通过进行该图形的清除和写来实现滚动、划入、划出、切入和切出显示效果之一。

12、根据权利要求 6 的再现装置，还包括构成双重缓冲器的两个平面存储器，其中通过将所显示图形从储存在所述双重缓冲器中的平面存储器之一中的内容切换到储存在另一所述平面存储器中的内容来显示图形。

13、一种在记录介质上进行记录的方法，所述方法包括：

产生应用数据；和

在记录介质中记录产生的应用数据，其中：

该应用数据包括通过多路复用视频流和图形流构成的数字流；

该视频流代表由多个图像构成的运动图像，该图形流包括：

代表要与该图像组合的该图形的图形数据；和

窗口信息，规定窗口以在其中呈现图形，该窗口信息表示平面上的该窗口的宽度、高度和位置，该平面是组合该图形与该图像的再现

装置的平面存储器。

14、一种用于使计算机再现数字流的程序，所述数字流是通过多路复用视频流和图形流构成的，所述程序包括：

代码，可操作地使该计算机对该视频流进行解码，以便获得由多个图像构成的运动图像；和

代码，可操作地使计算机呈现图形，以便与该图像同步地显示，其中：

该图形流包括窗口信息，该窗口信息规定该平面的一部分作为窗口以在其中呈现图形；和

可操作地使该计算机呈现该图形的所述代码包括可操作地使该计算机对用于组合该图形与该图像的平面存储器中的窗口中的图形执行清除，以及将该图形写到该平面存储器中的该窗口中的代码。

15、一种再现数字流的方法，所述数字流是通过多路复用视频流和图形流构成的，所述方法包括：

对该视频流进行解码，以便获得由多个图像构成的运动图像；和
呈现图形，以便与该图像同步地显示，其中：

该图形流包括窗口信息，用于规定一部分平面作为窗口以在其中呈现图形；和

所述图形的呈现包括用于组合该图形与该图像的平面存储器中的窗口中的该图形的清除和将该图形写到该平面存储器中的该窗口。

16、一种用于再现数字流的集成电路，所述数字流是通过多路复用视频流和图形流构成的，所述集成电路包括：

视频解码器，可操作地对该视频流进行解码，以便获得由多个图像构成的运动图像；

图形解码器，可操作地呈现图形，以便与该图像同步地显示；和
平面存储器，其对应平面并用于组合该图形与该图像，其中：

该图形流包括窗口信息，该窗口信息规定该平面的一部分作为窗

口以在其中呈现图形；和

通过所述图形解码器执行的图形的呈现包括该平面存储器中该窗口中的图形的清除和将图形写到该平面存储器中的该窗口。

记录介质和方法、再现装置和方法、程序和集成电路

技术领域

本发明涉及一种记录介质，如 BD-ROM，和再现器装置，特别涉及通过再现数字流而进行加字幕的技术，该数字流是通过多路复用视频流（video stream）和图形流（graphics stream）而构成的。

通过实施图形流实现的背景技术是用于允许不同语言区域的人们欣赏用他们本国语言以外的语言产生的影片的重要技术。常规的加字幕技术的例子是用于基于由欧洲通讯标准协会（European Telecommunications Standards Institute（ETSI））提出的 ETSI EN 300 743 标准的像素缓冲器（Pixel Buffer）的存储器分配方案。像素缓冲器是用于临时储存解压缩图形的存储器，再现装置将像素缓冲器中的图形写入被称为图形平面（Graphics Plane）的显示存储器中，因此显示该图形。在存储器分配方案中，在像素缓冲器中包括区域（region）的定义，并且对应该区域的一部分解压缩图形被写入图形平面中。例如，当字幕“Goodbye”包含在像素缓冲器中，并且定义该区域的位置和尺寸以便包括“Go”部分，则“Go”部分被写入图形平面中并显示在荧光屏上。同样。当定义该区域的位置和尺寸以便包括“Good”部分时，则“Good”部分显示在荧光屏上。

通过重复定义区域和写到图形平面，字幕“Goodbye”逐渐显示在荧光屏上，即首先是“Go”，然后是“Good”，接着是“Goodbye”，最后显示整个字幕“Goodbye.....”。通过利用这种方式提供字幕，可以实现划入（wipe-in）效果。

然而，ETSI EN 300 743 标准根本没有考虑到在写到图形平面的负荷很高时保证图形显示（graphics display）和图像显示（picture display）之间的同步。

被写到图形平面的图形没有被压缩，因而，用于写到图形平面的负荷增加，而图形的分辨率变得更高。当以 1920×1080 的分辨率提供

图形时，这个分辨率是被提出用于 BD-ROM 的标准分辨率，要写到图形平面的图形的大小高达 2 兆字节，并且需要从像素缓冲器到图形平面的图形数据传输的更高的带宽，以便与图像显示同步地提供大到 2 兆字节的图形。然而，要求高带宽用于数据传输以将 <RTI 图形写到图形平面会妨碍了降低制造再现装置成本的企图。通过使再现装置总是进行“合理的写入”，其中只有与前面的显示的不同之处被写到图形平面中，而可以降低写到图形平面所需的带宽。但是，要求再现装置总是进行“合理的写入”限制了可适用于再现装置的软件。

如上所述，写到图形平面的高负荷要求再现装置在高带宽下工作，或者进行合理的写入，结果是，限制了再现装置的产品研制。

发明内容

本发明的目的是提供一种记录介质，即使在将要写到图形平面中的数据量很大时，利用该记录介质也可以与图像显示同步地更新图形。

为了实现上述目的，根据本发明的记录介质的例子是用于储存数据的记录介质，所述记录介质包括：通过多路复用视频流和图形流构成的数字流，其中所述视频流代表由多个图像构成的运动图像，图形流包括：表示将要与这些图像组合的图形的图形数据；和规定一个窗口以在其中提供图形的窗口信息，窗口信息指示出在平面上的窗口的宽度、高度和位置，该平面是将图形与图像组合起来的再现装置的平面存储器（plane memory）。

通过规定对应每个图像的平面的一部分作为用于提供图形的窗口，而不必使再现装置为整个平面提供图形，并且再现装置只在有限尺寸的窗口中提供图形就足够了。由于不必在平面中在窗口以外提供图形，因此可以减少再现装置中的软件的负载。

此外，通过设置窗口的尺寸以便保证图形和图像之间的同步显示，可以使进行制作的制造者保证在任何种类的再现装置中的同步显示，即使在最坏情况下进行图形的更新。

而且，通过窗口信息设置窗口的位置和尺寸，可以调整制作中的

窗口的位置和尺寸,从而在观察屏幕时,使字幕不干扰对图像的观看。因此,即使屏幕上的图像随着时间流逝而改变时,也可保持图形的可视性,因此可以保持影片的质量。

更新图形时的最坏情况是指在最低效率操作下更新图形的情况,即全部清楚窗口并重新绘制窗口。当为了准备最坏情况而设置窗口的尺寸时,希望上述记录介质使得窗口的宽度和高度设置成使窗口的尺寸是该平面的 $1/x$, 该平面对应每个图像的尺寸, x 是基于窗口更新率和图像显示率之间的比例的实数。

通过用这种方式设置窗口尺寸,用于写到图形平面上所需的再现装置上的带宽设置为固定值。通过构成再现装置,以便满足这个带宽,可以实现图形和图像之间的同步显示,而与安装到再现装置上的软件无关。

如上所述,可以提供用于再现装置的结构的标准。只要传输速度设置成满足该最低标准,该再现装置的设计可以由开发者决定。因此,可以扩大再现装置的研制的可能性。

附图的简要说明

- 图 1 表示根据本发明记录介质的使用的例子;
- 图 2 表示 BD-ROM 的结构;
- 图 3 是示意性地表示 AVClip 的结构示意图;
- 图 4A 表示一个展示 (presentation) 图形流的结构;
- 图 4B 表示转换了功能段之后获得的 PES 信息包;
- 图 5 表示由各种功能段构成的逻辑结构;
- 图 6 表示字幕的显示位置和时元 (Epoch) 之间的关系;
- 图 7A 表示在目标定义段 (ODS) 中定义图形目标的语法;
- 图 7B 表示调色板 (Palette) 定义段的语法;
- 图 8A 表示窗口定义段 (WDS) 的语法;
- 图 8B 表示描绘组合段 (PCS) 的语法;
- 图 9 表示用于做字幕的显示设置的描述的例子;
- 图 10 表示 DS1 中的 WDS 和 PCS 的描述的例子;

图 11 表示 DS2 中的 PCS 的描述的例子；

图 12 表示 DS3 中的 PCS 的描述的例子；

图 13 是在进行切入/切出(Cut-In/Out)时的显示设置的描述的例子，并且是沿着时间轴表示的；

图 14 是在进行淡入/淡出(Fade-In/Out)时沿着时间轴表示的显示设置的描述的例子；

图 15 是在进行滚动 (Scrolling) 时沿着时间轴表示的显示设置的描述的例子；

图 16 是在进行划入/划出(Wipe-In/Out)时沿着时间轴表示的显示设置的描述的例子；

图 17 是对比两种情况的图：窗口具有四个图形目标，和窗口具有两个图形目标；

图 18 表示用于计算解码持续时间的算法的例子；

图 19 是图 18 的算法的流程图；

图 20A 和 B 是图 18 的算法的流程图；

图 21A 表示每个窗口具有目标定义段的情况；

图 21B 和 C 是表示在图 18 中所提到的数字之间的顺序的时序图；

图 22A 表示其中每个窗口具有两个目标定义段的情况；

图 22B 和 C 表示图 18 中所提到的数字之间的顺序的时序图；

图 23A 表示两个窗口中的每个窗口包括 ODS 的情况；

图 23B 表示解码周期 (2) 比清除周期 (1) 和写周期 (31) 的总和更长的情况；

图 23C 表示清除周期 (1) 和写周期 (31) 的总和比解码周期 (2) 长的情况；

图 24 表示在本说明书中的例子中所述的更新时间的偏移；

图 25A 表示为了进行上述更新而描述的四个显示设置；

图 25B 是表示在四个显示设置中包含的功能段的 DTS 和 PTS 的设置时序图；

图 26 表示根据本发明的再现装置的内部结构；

图 27 表示写速度 Rx、Rc、Rd、图形平面 8、编码数据缓冲器 13、

目标缓冲器 15 和组合缓冲器 16 的尺寸；

图 28 是表示由再现装置执行的流水线处理的时序图；

图 29 表示在完成图形平面的清除之前 ODS 的解码结束的情况的流水线处理中的时序图；

图 30 是表示功能段的负载操作过程的流程图；

图 31 表示一个多路复用的例子；

图 32 表示将 DS10 装载到编码数据缓冲器 13 的一种方式；

图 33 表示正常再现中的 DS1、DS10、DS20 的装载；

图 34 表示在图 33 所示正常再现中的 DS1、DS10 和 DS20 的装载；

图 35 表示通过图形控制器 17 执行的处理的流程图；

图 36 表示通过图形控制器 17 执行的处理的流程图；

图 37 表示通过图形控制器 17 执行的处理的流程图；

图 38 表示在 PDS 的 PTS 基础上的再现装置的流水线处理；

图 39 是表述再现装置的流水线处理中的 END 的意义的示意图；

图 40 表示根据第二实施例的再现装置的内部结构；

图 41 示意性地表示读出和写到构成双缓冲器的图形平面的操作；

图 42 是表示根据第三实施例的 BD-ROM 的制造工艺的流程图。

实施本发明的最佳方式

第一实施例

下面解释根据本发明的记录介质的第一实施例。

图 1 表示记录介质的使用的例子。在图中，BD-ROM 100 是根据本发明的记录介质。BD-ROM 100 用于给由再现装置 200、电视 300 和遥控器 400 构成的家庭影院系统提供电影作品数据。

根据本发明的记录介质是通过 BD-ROM 的应用层 (application layer) 的改进来制造的。图 2 表示 BD-ROM 的结构。

在该图中，在图的底部示出了 BD-ROM，并在 BD-ROM 上方示出了 BD-ROM 上的轨道。该轨道实际上在盘上是螺旋形状的，但是图中示出的是直线。该轨迹包括导入 (lead-in) 区、卷区 (volume area) 和导出区。这个图中的卷区具有物理层、文件系统层和应用层。在图

的顶部，使用目录结构示出了 BD-ROM 的应用格式。如图所示，BD-ROM 具有在根目录下的目录 BDMV，并且 BDMV 目录包含具有扩展名 M2TS (XXX.M2TS) 的用于储存 AVClip 的文件、具有扩展名 CLPI (XXX.CLPI) 的用于储存 AVClip 的管理信息的文件、和具有扩展名 MPLS (YYY.MPLS) 的用于为 AVClip 定义逻辑播放列表 (PL) 的文件。通过形成上述应用格式，可以制造根据本发明的记录介质。在每种类型有一个以上文件的情况下，优选在 BDMV 下提供命名为 STREAM、CLIPINF 和 PLAYLIST 的三个目录，以便在一个目录中储存具有相同扩展名的文件。具体地说，希望在 STREAM 中储存具有扩展名 MSTTS 的文件、在 CLIPINF 中储存具有扩展名 CLPI 的文件以及在 PLAYLIST 中储存具有扩展名 MPLS 的文件。

下面解释上述应用格式中的 AVClip (XXX.M2TS)。

AVClip (XXX.M2TS) 是通过多路复用视频流、至少一个音频流和展示图形流获得的 MPEG-TS 格式 (TS 是传输流) 的数字流。视频流表示影片的图像，音频流表示影片的声音，展示图形流 (presentation graphics stream) 表示影片的字幕。图 3 是示意性地表示 AVClip 的结构示意图。

AVClip (XXX.M2TS) 按照以下方式构成。由多个视频帧 (图像 pj1, pj2 和 pj3) 构成的每个视频流和由多个音频帧 (图的顶行) 构成的音频流被转换成一行 PES 信息包 (图的第二行)，然后被转换成一行 TS 信息包 (图的第三行)。展示图形流 (图的底行) 被转换成 PES 信息包 (图的倒数第二行)，然后转换成 TS 信息包 (图的倒数第三行)。三行 PS 信息包被多路复用，由此构成 AVClip (XXX.M2TS)。

在图中，只有一个展示图形流被多路复用。但是，在 BD-ROM 可对多种语言兼容的情况下，用于每种语言的展示图形流被多路复用从而构成 AVClip。用上述方式构成的 AVClip 被分割成一个以上的范围 (extent)，与普通计算机文件相同，并被储存在 BD-ROM 中的各区域中。

接着，解释展示图形流。图 4A 表示展示图形流的结构。顶行表示要被多路复用到 AVClip 的 TS 信息包行。从顶行数的第二行表示

构成图形流的 PES 信息包行。PES 信息包行是通过提取具有预定 PID 的 TS 信息包中的有效载荷并连接该提取的有效载荷而构成的。

从顶行数的第三行表示图形流的结构。该图形流由命名为展示组合段 (PCS)、窗口定义段 (WDS)、调色板定义段 (PDS)、目标定义段 (ODS) 和显示设置段 (END) END 的功能段构成。在上述功能段当中, PCS 被称为屏幕组合段, WDS、PDS、ODS 和 END 被称为定义段。PES 信息包和每个功能段一一对应, 或者一个对应多个。换言之, 一个功能段在被转换成一个 PES 信息包之后, 或者在被分割成多个片段并转换成一个以上 PES 信息包之后被记录在 BD-ROM 中。

图 4B 表示通过转换功能段获得的 PES 信息包。如图所示, PES 信息包由信息包报头和有效载荷构成, 并且有效载荷是功能段的主体。信息包报头包括对应该功能段的 DTS 和 PTS。下面将包含在信息包报头中的 DTS 和 PTS 称为功能段的 DTS 和 PTS。

上述各种功能段构成如图 5 所示的逻辑结构。图 5 表示由各种功能段构成的逻辑结构。在图中, 顶行表示时元 (epoch), 中间行表示显示设置 (DS), 底行表示功能段。

在构成图形流的所有多个功能段当中, 中间行所示的每个 DS 是构成一个屏幕的图形的一组功能段。图中的虚线表示底行中的功能段属于由该虚线标示的 DS, 并表示 PCS、WDS、PDS、ODS 和 END 等一串功能段构成一个 DS。该再现装置能够通过读取构成 DS 的功能段而产生一个屏幕的图形。

顶行所示的时元表示时间周期, 在一个时元中沿着 AVClip 再现的时间轴存储器管理在时间上是连续的。一个时元还表示分配给相同时间周期的一组数据。这里所称的存储器是储存一个屏幕的图形的图形平面, 和储存解压缩图形数据的目标缓冲器。存储器管理的连续性意味着在该时元中不会发生图形平面或目标缓冲器的闪烁, 并且只在图形平面上的预定矩形区域中进行图形的擦除和展现 (这里的闪烁表示擦除平面或缓冲器中的储存数据的所有内容)。在一个时元内矩形区域的尺寸和位置是固定的。只要只在图形平面上的预定矩形区域中进

行图形的擦除和展现，就可以保证图像和图形之间的同步再现。换言之，该时元是再现时间轴中的一个单元，并且在这个单元中，保证图像和图形同步地再现。当将在其中擦除和展现图形的该区域移动到不同位置时，必须在时间轴上定义一个点来移动该区域，并且该点之后的一段时间成为一个新的时元。在两个时元之间的区间不能保证同步再现。

在观看实际影片时，一个时元是在屏幕上的相同矩形区域中显示字幕的时间周期。图 6 表示字幕的位置和时元之间的关系。在该附图所示的例子中，示出的五个字幕“Actually…”、“I was hiding”、“my feeling”、“I always”和“loved you”根据影片中的图像而移动。具体地说，字幕“Actually…”、“I was hiding”和“my feeling”出现在屏幕的底部，而字幕“I always”和“loved ybu”显示在屏幕的顶部。当观看屏幕时，考虑到影片的可观赏性，矩形区域的位置是移动的以使字幕不遮挡图像。字幕出现在底部的时间周期是时元 1，字幕出现在顶部的下一时间周期是时元 2。时元 1 和 2 各自具有不同区域以在其中显示字幕。时元 1 中的区域是位于屏幕底部的窗口 1，并且时元 2 中的区域是位于屏幕顶部的窗口 2。存储器管理在每个时元 1 和 2 中是连续的，因而，在窗口 1 和 2 的字幕的显示与图像同步。

接着，将详细介绍显示设置 (DS)。

图 5 中的虚线 hkl1 和 hkl2 表示位于中间行的功能段属于哪个时元。一系列 DS “Epoch Start” (时元开始)、“采集点” (Acquisition Point) 和“正常情况” (Normal Case) 构成在顶部行的该时元。“Epoch Start”、“Acquisition Point”和“Normal Case”是 DS 的各个类型，并且“Acquisition Point”和“Normal Case”之间的顺序都没有关系，它们中任一个都可以先出现。

Epoch Start 是具有“新显示”的显示效果的 DS，其表示新时元的开始。因此，Epoch Start 包含用于显示屏幕的新元素所需的所有功能段。Epoch Start 设置在作为 AVClip 的跳跃操作 (skip operation) 的目标的位置上，如影片中的章节。

Acquisition Point 是具有“显示更新”的显示效果的 DS，并且其用

于呈现图形的内容与作为在前 DS 的 Epoch Start 是相同的。Acquisition Point 不设置在时元的开始点，但是包含显示屏幕的新成分所需的所有功能段。因此，当对 Acquisition Point 进行跳跃操作时，可以不出现故障地显示图形。因而，利用 Acquisition Point，可以在该时元中部构成一个屏幕。

Acquisition Point 设置在可以作为跳跃操作的目标的位置上。这种位置的例子是可以在进行时间检索时指定的位置。时间检索是响应于用户的输入时间的操作，从而在对应由用户规定的时间的再现点而开始再现。该时间是大致规定的，如相差 10 分钟或 10 秒钟，相应地，再现开始的点设置为例如 10 分钟间隔或 10 秒钟间隔。通过将 Acquisition Point 提供在再现可以开始的点上，可以在时间检索之后平滑地进行再现。

Normal Case 是具有“显示更新”的显示效果的 DS，并只包含不同于前一屏幕成分的元素。具体地说，当 DS_v 中的字幕与 DS_u 中的字幕相同但是屏幕在 DS_v 和 DS_u 中以不同方式显示时，DS_v 设置成只包括 PCS 并使 DS_v 为 Normal Case。由此，不必提供具有与前一 DS 中的 ODS 的内容相同内容的 ODS，从而可以减小 BD-ROM 中的数据大小。另一方面，由于作为 Normal Case 的 DS 只包含差异，因此单独使用 Normal Case 不可能构成该屏幕。

下面解释定义段 (ODS、WDS 和 PDS) 的细节。目标定义段 (ODS) 是定义图形目标的功能段。下面首先解释图形目标。记录在 BD-ROM 中的 AVClip 的卖点是其与高清晰度电视一样的高分辨率，因此图形目标的分辨率设置在 1920×1080 像素。由于 1920×1080 像素的高分辨率，可以在屏幕上清楚地显示字幕的具体字符格式。关于字幕的颜色，每个像素 (色差红 Cr、色差蓝 Cb、亮度 Y、和透明度 T) 的指标值的位长度是 8 位，因此可以从全色 (16777216 种颜色) 中选择用于这些字幕的任何 256 种颜色。由图形目标实现的字幕通过在透明背景上设置正文来呈现。

定义图形目标的 ODS 的语法示于图 7A 中。ODS 由表示字节是 ODS 的 segment_type (段_类型)、表示 ODS 的数据长度的

segment_length (段_长度)、唯一地表示对应该时元中的 ODS 的图形目标的 object_id (目标_id)、表示该时元内的 ODS 的版本的 object_version_number (目标_版本_号)、last_insequence_flag (最后_非序列_标记) 和作为对应一部分或全部图形目标的连续序列比特的 object_data_fragment。

object_id 用于唯一地识别对应该时元内的 ODS 的图形目标。图形流的该时元包含具有相同 ID 的一个以上的 ODS。具有相同 ID 的 ODS 也具有相同宽度和高度，并且被分配以目标缓冲器中的公共区域。在该公共区域中读取具有相同 ID 的一个 ODS 之后，用具有相同 ID 的下一个 ODS 对被读取的 ODS 进行重写。在进行视频流的再现时，通过用具有相同 ID 的下一 ODS 重写被读到目标缓冲器的 ODS，从而更新该 ODS 的图形。只在一个时元期间施加了尺寸约束，即，具有相同 ID 的图形目标的宽度和高度应该是相同的，并且不同时元中的图形目标可以具有不同尺寸。

下面介绍 last_sequence_flag(最后_序列_标记) 和 object_data_fragment(目标_数据_片段)。在有些情况下，由于 PES 信息包的有效载荷限制而不可能在一个 ODS 中储存构成字幕的解压缩图形。在这些情况下，图形分裂成一系列连续的片段，并且一个片段设置为 object_data_fragment。当一个图形目标作为一个以上片段被储存时，除了最后片段以外的每个片段都具有相同的尺寸。最后片段小于或等于在前片段的尺寸。承载这些片段的 ODS 按照相同的连续顺序出现在 DS 中，其中由 ODS 表示的序列的最后一个具有 last_sequence_flag。虽然 ODS 的上述语法是基于这样的假设，即，该多个片段从在前 PES 开始堆叠，但是这些片段可以堆叠成使得每个 PES 含有空白部分。

接着，解释调色板定义段 (PDS)。PDS 用于定义颜色转换用的调色板。图 7B 表示 PDS 的语法。PDS 由表示该段是 PDS 的 segment_type、表示 PDS 的数据长度的 segment_length、唯一地标识包含在 PDS 中的调色板的 palette_id(调色板_id)、表示该时元内的 PDS 的版本的 palette_version_number(调色板_版本_号)、和规定调色板的

条目号 (entry number) 的 palette_entry_id 构成。palette_entry_id(调色板_条目_id)表示色差红 (Cr_value)、色差蓝 (Cb_value)、亮度 (Y_value) 和透明度 (T_value)。

接下来, 介绍窗口定义段 (WDS)。

WDS 用于定义图形平面上的矩形区域。如上所述, 只有在图形平面上的某个区域内进行擦除和显现时, 存储器管理是连续的。图形平面上的该区域由 WDS 定义并被称为“窗口”。图 8A 示出了 WDS 的语法。如图所示, WDS 由表示该段是 WDS 的 segment_type、表示 WDS 的数据长度的 segment_length、唯一地标示图形平面上的该窗口的 window_id、规定图形平面上的窗口的左上角像素的水平地址的 window_horizontal_position、规定图形平面上的窗口的左上角像素的垂直地址的 window_vertical_position、规定图形平面上的窗口的宽度的 window_width 和规定图形平面上的窗口的高度的 window_height 构成。

下面介绍 window_horizontal_position、window_vertical_position、window_width、和 window_height 可以采取的数值的范围。用于这些值的坐标系在图形平面上的区域内, 并且该窗口的尺寸两维地由表示高度的 window_height 和表示宽度的 window_width 来表示。

window_horizontal_position 规定图形平面上的该窗口的左上角像素的水平地址, 并且处于 0 到 (window_width) - 1 的范围内。而且, window_vertical_position 规定图形平面上的该窗口的左上角像素的垂直地址, 并且处于 0 到 (window_height) - 1 的范围内。

window_width 规定图形平面上的该窗口的宽度。规定的宽度落入 1 到 (video_width) - (window_horizontal_position) 的范围内。此外, window_height 规定图形平面上的窗口的高度, 并且该规定的高度在 1 到 (video_height) - (window_vertical_position) 的范围内。

每一个时元内的图形平面上的窗口的位置和尺寸由 window_horizontal_position、window_vertical_position、window_width、和 window_height 定义。因而, 可以调节正在构成的窗口的位置和尺寸, 使得在观看影片时, 一个时元内的窗口出现在不遮挡图像的位置

上。由此，字幕的可见性变得更高。由于为每一个时元定义了 WDS，即使图像随着时间改变，也可以根据图像调节窗口的位置。结果是，影片的质量保持为与在字幕结合到影片的主体中的情况一样高。

接着，解释显示设置段（END）的结束。END 提供完成 DS 的传输的信息。在数据流中将 End 插入到紧跟在一个 DS 中的最后 IDS 之后，End 由表示该段是 END 的 `segment_type` 和表示 END 的数据长度的 `segment_length` 构成。END 不包括需要进一步解释的任何其它元素。

接着，下面将介绍展示成分段（PCS）。

PCS 是用于构成交互式显示的功能段。图 8B 表示 PCS 的语法。如图所示，PCS 由 `segment_type`、`segment_length`、`composition_number`、`Composition_state`、`palette_update_flag`、`palette_id` 和窗口信息 1-m 构成。

`composition_number` 通过 0-15 的范围内的数值表示 DS 中的图形更新。如果图形更新存在于时元的头部和 PCS 之间，则在每次发生图形更新时增加 `composition_number`。

`Composition_state` 表示其中含有 PCS 的 DS 的类型、Normal Case、Acquisition Point 或 Epoch Start。

`palette_update_flag` 表示 PCS 描述仅对调色板的显示更新。仅对调色板的显示更新表示只有调色板从相邻的在前调色板更新。如果进行仅对调色板的显示更新，则 `palette_update_flag` 字段设置为“1”。

`palette_id` 标示要在仅对调色板的显示更新中使用的调色板。

窗口信息 1-m 表示怎样控制包含该 PCS 的 DS 中的每个窗口。图 8B 中的虚线 wd1 用于详细地表述窗口信息 i 的内部语法。窗口信息 i 由 `object_id`、`window_id`、`object_cropped_flag`、`object_horizontal_position`、`object_vertical_position` 和 `cropping_rectangle` 信息 1-n 构成。

`object_id` 标示对应窗口信息 i 的窗口中的 ODS。

`window_id` 标示在 PCS 中分配了图形目标的窗口。最多两个图形目标可以分配给一个窗口。

`object_cropped_flag` 用于在目标缓冲器中在裁剪图形目标的显示和非显示之间进行切换。当 `object_cropped_flag` 设置为“1”，裁剪图形目标显示在目标缓冲器中，并如果设置为“0”，则不显示图形目标。

`object_horizontal_position` 规定图形平面中的图形目标的左上角像素的水平地址。

`object_vertical_position` 规定图形平面中的图形目标的左上角像素的垂直地址。

`cropping_rectangle` 信息 1-n 是在 `object_cropped_flag` 设置为“1”时使用的元素。虚线 wd2 是用于表示用于 `cropping_rectangle` 信息 i 的内部语法的细节。如虚线 wd2 所示，`cropping_rectangle` 信息 i 由四个字段构成：`object_cropping_horizontal_position`、`object_cropping_vertical_position`、`object_cropping_width` 和 `object_cropping_height`。

`object_cropping_horizontal_position` 规定在图形平面中显示图形目标期间使用的裁剪矩形的左上角角的水平地址。裁剪矩形是用于规定和裁剪一部分图形目标的裁剪框，并对应 ETSI EN 300 743 标准中的区域。

`object_cropping_vertical_position` 规定在图形平面中显现图形目标期间使用的裁剪矩形的左上角角的垂直地址。

`object_cropping_width` 规定该裁剪矩形的宽度。

`object_cropping_height` 规定该裁剪矩形的高度。

下面详细介绍 PCS 的具体例子。在该例中，在图像播放时，如图 6 所示的字幕“Actually…”、“I was hiding”和“my feelings.”通过 3 次写到图形平面而逐渐地出现。图 9 是用于实现这种字幕显示的说明的例子。图中的时元包括 DS1 (Epoch Start)、DS2 (Normal Case) 和 DS3 (Normal Case)。DS1 含有用于规定其中显示字幕的窗口的 WDS、用于规定台词“Actually… I was hiding my feelings.”和第一 PCS。DS2 含有第二 PCS，并且 DS3 含有第三 PCS。

图 10-12 表示在 DS 中含有的 WDS 和 PCS 的例子。图 10 表示 DS1 中的 PCS 的例子。

在图 10 中，WDS 的 `window_horizontal_position` 和 `window_vertical_position` 由 LP1、图形平面上的窗口的左上角像素的位置表示。`window_width` 和 `window_height` 分别表示该窗口的宽度和高度。

在图 10 中，`object_cropping_horizontal_position` 和 `object_cropping_vertical_position` 表示一个坐标系中的裁剪矩形的参考点 ST1，该坐标系的原点是图形目标的左上角像素。裁剪矩形是具有从 ST 到 `object_cropping_width` 的宽度和从 ST 到 `object_cropping_height` 的高度的区域（由粗线框所示的矩形）。裁剪图形目标位于虚线框 cp1 表示的矩形内，具有坐标系中的参考点，坐标系的原点位于图形平面中的 `object_horizontal_position` 和 `object_vertical_position`（图形目标的左上角像素）。由此，将字幕“Actually…”写到图形平面上的窗口，然后与电影图像组合并显示在屏幕上。

图 11 表示 DS2 中的 PCS 的例子。不解释 DS2 中的 WDS，因为 DS2 中的 WDS 与 DS1 中的 WDS 相同。DS2 中的裁剪信息的描述不同于图 10 所示的裁剪信息的描述。

在图 11 中，裁剪信息中的 `object_cropping_horizontal_position` 和 `object_cropping_vertical_position` 表示目标缓冲器中的“Actually… I was hiding my feelings.”中的字幕“I was hiding”的左上角像素。`object_cropping_width` 和 `object_cropping_height` 表示含有字幕“I was hiding”的矩形的宽度和高度。由此，字幕“I was hiding”被写到图形平面上，然后与电影图像组合并显示在屏幕上。

图 12 表示在 DS3 中的 PCS 的例子。由于 DS3 中的 WDS 与 DS1 中的 WDS 相同，因此不解释 DS3 中的 WDS。DS3 中的裁剪信息的描述不同于图 10 所示的裁剪信息的描述。

在图 12 中，裁剪信息中的 `object_cropping_horizontal_position` 和 `object_cropping_vertical_position` 表示目标缓冲器中的“Actually… I was hiding my feelings.”中的字幕“my feelings”的左上角像素。`object_cropping_width` 和 `object_cropping_height` 表示含有字幕“my

feelings”的矩形的宽度和高度。由此，字幕“my feelings”被写到图形平面的窗口中，然后与电影图像组合并显示在屏幕上。

如上所述，通过描述 DS1、DS2 和 DS3，可以实现在屏幕上显示字幕的效果。还可以实现其它效果，并且下面介绍用于实现其它效果的说明协议。

首先，解释用于切入/切出的说明协议。图 13 表示在进行切入/切出时对 DS 的描述的沿着时间轴表示的例子。

在图中，窗口 (x, y, u, v) 中的 x 和 y 分别表示 `window_vertical_position` 和 `window_horizontal_position` 的值，并且 u 和 v 分别表示 `window_width` 和 `window_height`。而且在图中，裁剪矩形 (a, b, c, d) 中的 a 和 b 分别表示 `object_cropping_vertical_position` 和 `object_cropping_horizontal_position` 的值，并且 c 和 d 分别表示 `object_cropping_width` 和 `object_cropping_height` 的值。显示设置 DS11、DS12 和 DS13 处于图中的再现时间轴上的点 $t11$ 、 $t12$ 和 $t13$ 上。

处于点 $t11$ 上的 DS11 包括其中 `Composition_state` 是“Epoch Start”和 `object_cropped_flag` 是“0”（`no_cropping_rectangle_visible`）的 PCS#0、具有在图形平面中的 $(100, 100)$ 上的宽度 $700 \times$ 高度 500 的窗口用的声明的 WDS#0、PDS#0、表示字幕“片头字幕 (Credits):”的 ODS#0 和 END。

处于点 $t12$ 的 DS12 包括 PCS#1，其 `Composition_state` 是“Normal Case”，并表示图形目标的裁剪操作是从目标缓冲器中的 $(0, 0)$ 开始，大小为 600×400 (`cropping_rectangle#0 (0, 0, 600, 400)`)，并且将裁剪图形目标定位在图形平面 (on Window#0 $(0, 0)$) 中的坐标 $(0, 0)$ 。

处于点 $t13$ 的 DS13 包括其 PCS#2，其 `Composition_state` 是“Normal Case”，并且其中 `object_cropped_flag` 设置为“0”，以便擦除裁剪图形目标 (`no_cropping_rectangle_visible`)。

利用上述的显示设置，“Credits:”在 $t11$ 不显示，而在 $t12$ 出现，然后再次在 $t13$ 不显示，从而实现了切入/切出效果。

其次，解释用于淡入/淡出效果的说明协议。图 14 表示在进行淡

入/淡出时 DS 的说明的沿着时间轴表示的例子。显示设置 DS21、DS22、DS23 和 DS24 处于图中的再现时间轴上的点 t21、t22、t23 和 t24。

处于点 t21 上的 DS21 包括 PCS#0、WDS#0、PDS#0、ODS#0 和 END，其中 PCS#0 的 Composition_state 是“Epoch Start”并表示目标缓冲器中的图形目标的裁剪操作从 (0, 0) 开始且大小为 600×400 (cropping_rectangle#0 (0, 0, 600, 400))，并且将裁剪图形目标定位在图形平面 (在 on Window#0 (0, 0)) 中的坐标 (0, 0)，WDS#0 具有对窗口位于图象平面的 (100, 100)，大小为宽度 700×高度 500 的声明，ODS#0 表示字幕“Fin”。

处于点 t22 上的 DS22 包括其 Composition_state 是“Normal Case”的 PCS#1 和 PDS#1。PDS#1 表示与 PDS#0 相同的 Cr 和 Cb 的值，但是由 PDS#1 表示的亮度比 PDS#0 中的亮度高。

处于点 t23 的 DS23 包括其 Composition_state 是“Normal Case”的 PCS#2、PDS#2 和 END。PDS#2 表示与 PDS#1 相同的 Cr 和 Cb 的值，但是由 PDS#2 表示的亮度比 PDS#1 中的亮度低。

处于点 t24 上的 DS24 包括其 Composition_state 是“Normal Case”和 object_cropped_flag 是“0” (no_cropping_rectangle_visible) 的 PCS 和 END。

每个 DS 规定不同于在前 DS 的 PDS，因而，在一个时元中以上 PCS 显现的图形目标的亮度逐渐变高或低。借此，可以实现淡入/淡出的效果。

下面将解释用于滚动的说明协议。图 15 表示在进行滚动时沿着时间轴表示的 DS 的说明的例子。显示设置 DS31、DS32、DS33 和 DS34 处于图中的再现时间轴上的点 t31、t32、t33 和 t34 上。

处于点 t31 上的 DS31 包括其 Composition_state 被设置为“Epoch Start”并且 object_cropped_flag 是“0” (no_cropping_rectangle_visible) 的 PCS#0、具有对在图形平面中的 (100, 100) 处的宽度 700×高度 500 的窗口的声明的 WDS#0、PDS#0、表示字幕“Credits: Company”的 ODS#0 和 END。

处于点 t32 的 DS32 包括 PCS#1, 其 Composition_state 是“Normal Case”, 并表示图形目标的裁剪操作是目标缓冲器中从 (0, 0) 开始大小为 600×400 (cropping_rectangle#0 (0, 0, 600, 400)), 并且将裁剪图形目标定位在图形平面中的坐标 (0, 0) 处 (on Window#0 (0, 0))。图形目标中的从 (0, 0) 开始的 600×400 的区域包括显示为两行的字幕“Credits: Company”的一部分“Credits”, 因此“Credits:”部分出现在图形平面上。

处于点 t33 的 DS33 包括 PCS#2, 其 Composition_state 是“Normal Case”, 并且表示图形目标的裁剪操作是目标缓冲器中从坐标 (0, 100) 开始的 600×400 大小 (cropping_rectangle#0 (0, 100, 600, 400)), 并且将裁剪的图形目标定位在图形平面中的坐标 (0, 0) 处 (on Window#0 (0, 0))。目标缓冲器中的从 (0, 100) 开始的 600×400 尺寸的区域包括两行中所示的字幕“Credits: Company”的“Credits:”部分和“Company”部分, 因此“Credits:”和“Company”部分出现在图形平面上的两行中。

处于点 t34 的 DS34 包括 PCS#3, 其 Composition_state 是“Normal Case”, 并表示图形目标的裁剪操作是从目标缓冲器 (cropping_rectangle#0 (0, 200, 600, 400)) 中从 (0, 200) 开始的 600×400 的尺寸, 并将裁剪图形目标定位在图形平面中的坐标 (0, 0) (on Window#0 (0, 0))。图形目标中的从 (0, 0) 开始的 600×400 的区域包括在两行中所示的字幕“Credits: Company”的“Company”部分, 因此“Company”部分出现在图形平面上。通过上述 PCS 说明, 可以向下滚动在两行中的该字幕。

最后, 将解释用于划入/划出效果的说明协议。图 16 表示沿着时间轴表示的进行划入/划出时的 DS 的说明的例子。显示设置 DS21、DS22、DS23 和 DS24 处于图中的再现时间轴上的点 t21、t22、t23 和 t24 上。

处于点 t51 上的 DS51 包括其 Composition_state 是“Epoch Start”和 object_cropped_flag 是“0” (no_cropping_rectangle_visible) 的 PCS#0、具有对在图形平面中的在 (100, 100) 处的宽度 700×高度

500 的窗口的声明的 WDS#0、PDS#0、表示字幕“Fin”的 ODS#0 和 END。

处于点 t52 上的 DS52 包括 PCS#1, 其 Composition_state 是“Normal Case” 和表示图形目标的裁剪操作是从目标缓冲器中的 (0, 0) 开始的 600×400SIZE (cropping_rectangle#0 (0, 0, 600, 400)), 并且在图形平面中的坐标 (0, 0) 上定位裁剪图形目标 (on Window#0 (0, 0))。目标缓冲器中从 (0, 0) 开始的 600×400 尺寸的区域包括字幕“Fin”, 因此在图形平面上出现字幕“Fin”。

处于点 t53 的 DS53 包括 PCS#2, 其 Composition_state 是“Normal Case”, 并且表示图形目标的裁剪操作将在目标缓冲器中从 (200, 0) 开始的 400×400SIZE (cropping_rectangle#0 (200, 0, 400, 400)), 并且将裁剪的图形目标定位在图形平面中的坐标 (200, 0) 处 (on Window#0 (200, 0))。由此, 用窗口中的坐标 (200, 0) 和 (400, 400) 表示的区域变为显示区域, 并且由坐标 (0, 0) 和 (199, 400) 表示的区域变为非显示区域。

处于点 t54 上的 DS54 包括 PCS#3, 其 Composition_state 是“Normal Case”, 并且表示图形目标的裁剪操作是在目标缓冲器中从 (400, 0) 开始的 200×400SIZE (cropping_rectangle#00 (400, 0, 200, 400)), 并且将裁剪的图形目标定位在图形平面 (on Window #0 (400, 0) 上) 中的坐标 (400, 0) 上。由此, 用窗口中的坐标 (0, 0) 和 (399, 400) 表示的区域变为非显示区域。

由此, 随着非显示区域变大, 显示区域变小, 因此实现了划入/划出效果。

如上所述, 使用相应的语言 (script) 可以实现各种效果, 如切入/切出、淡入/淡出、划入/划出和滚动, 因此可以做各种布置来呈现字幕。

为了实现上述效果的约束如下。为了实现滚动效果, 需要对窗口进行清除和重画的操作。以图 15 为例, 必须进行“窗口清除”, 从而在 t32 从图形平面中擦除图形目标“Credits:”, 然后进行“窗口重画”, 从而在 t32 和 t33 之间的间隔期间将“Credits:”的下部分和“Company”

的上部分写到图形平面上。假设该间隔与视频帧的间隔相同，则用于滚动效果所希望的目标缓冲器和图形平面之间的传送速度是重要的。

这里，介绍关于窗口可以是多大的约束。 R_c 是目标缓冲器和图形平面之间的传送速度。这里最坏的情况是要以速度 R_c 进行窗口清除和窗口重画。在这种情况下，需要以 R_c 的一半的速度 ($R_c/2$) 进行窗口清除和窗口重画中的每一个。

为了使窗口清除和窗口重画与视频帧同步，需要满足以下等式，

$$\text{窗口尺寸} \times \text{帧速度} = R_c/2$$

如果帧速度是 29.97， R_c 由如下关系式表示，

$$R_c = \text{窗口尺寸} \times 2 \times 29.97$$

在显示字幕时，窗口尺寸占图形平面的至少 25%到 33%。图形平面中的像素总量为 1920×1080 。假定每像素的索引比特长度是 8 位，则图形平面的总容量是 2 兆比特 ($= 1920 \times 1080 \times 8$)。

假定窗口尺寸是图形平面的总容量的 1/4，窗口尺寸变为 500K 比特 ($= 2M$ 比特/4)。通过将这个值代到上述关系式中，计算出 R_c 为 256 Mbps ($= 500K$ 比特 $\times 2 \times 29.97$)。如果用于窗口清除和窗口重画的速度可以是帧速度的一半或四分之一，即使 R_c 相同，也可以使窗口的尺寸成为两倍或四倍。

通过保持该窗口尺寸为图形平面的 25%到 33%，并以 256 Mbps 的传送速度显示字幕，无论要实现任何种类的显示效果，都可以保持图形和电影图像之间的同步显示。

接着，介绍窗口的位置、尺寸和面积。如上所述，窗口的位置和面积在一个时元内不变。窗口的位置和尺寸在一个时元内设置为相同，因为如果位置和尺寸改变则必须改变图形平面的目标写地址，并且改变地址所引起的操作将降低从目标缓冲器向图形平面的传送速度。

每个窗口的图形目标的数量有限制。提供该数量的限制是为了减少传送解码图形目标的操作。这里的操作是在设置图形目标的边缘的地址时产生的，并且边缘的数量越多，产生的操作越多。

图 17 表示相互比照的例子，在一个例子中一个窗口具有四个图形目标，另一个例子中一个窗口具有两个图形目标。具有四个图形目标的例子的边缘的数量是具有两个图形目标的例子的边缘数量的两倍。

在图形目标的数量没有限制的情况下，不知道在传送图形时会产生多少操作，因此用于传送的负载急剧增加和减少。另一方面，当窗口中的图形目标的最大数量是两个时，考虑最多有 4 个操作费来设置传送速度。因而，更容易设置最小传送速度的量。

接着，介绍具有 PCS 和 ODS 的 DS 如何分配到 AVClip 的时间轴。时元是一个时间周期，其中存储器管理沿着再现时间轴是连续的。由于时元由一个以上的 DS 构成，如何将 DS 分配到 AVClip 的再现时间轴是很重要的。AVClip 的再现时间轴是用于规定每片图像数据的解码和再现的定时的时间轴，所述图像数据构成被多路复用为 AVClip 的视频流。再现时间轴上的解码和再现定间用 90KHz 的精度表示。附在 DS 中的 PCS 和 ODS 上的 DTS 和 PTS 表示在再现时间轴上同步控制的定时。将显示设置分配到再现时间轴上意味着使用附在 PCS 和 ODS 上的 DTS 和 PTS 进行同步控制。

首先，下面将介绍如何使用固定到 ODS 上的 DTS 和 PTS 进行同步控制。

以 90KHz 的精度，DTS 表示在 ODS 的解码开始时的时刻，PTS 表示在解码结束时的时刻。

ODS 的解码不是立即完成的，而是具有一定的时间长度。响应于要清楚地表示解码持续时间的开始点和结束点的请求，ODS 的 DTS 和 PTS 分别表示在解码开始和结束时的时刻。

PTS 的值表示截止时间，因此必须使 ODS 的解码在由 PTS 表示的时间前完成，并且解压缩的图形目标被写到再现装置上的目标缓冲器。

DS_n 中的任何 ODS_j 的解码开始时间以 90KHz 的精度由 DTS

(DSn[ODS]) 来表示。将解码持续时间的最大长度加上 DTS (DSn[ODS]) 是 ODSj 的解码结束时的时间。

当 ODSj 的尺寸是“SIZE(DSn[ODS])”和 ODS 的解码速度是“Rd”时，用秒表示的用于解码所需的最大时间表示为“SIZE (DSn[ODS]) //Rd”。符号“//”表示进行除法并对十分位之后的位置四舍五入的操作符。

通过将最大时间周期转换成以 90KHz 的精度表示的数字并加到 ODSj 的 DTS 上，就计算得到了由 PTS 表示的解码结束时的时间 (90KHz)。

DSn 中的 ODSj 的 PTS 用以下关系式表示，

$$\text{PTS}(\text{DSn}[\text{ODSj}]) = \text{DTS}(\text{DSn}[\text{ODSj}]) + 90000 \times (\text{SIZE}(\text{DSn}[\text{ODSj}]) // \text{Rd})$$

此外，必须使两个连续的 ODS, ODSj 和 ODSj+1 之间的关系满足以下关系式，

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

接下来解释 PCS 的 DTS 和 PTS 的设置。

必须在 DS_n 中的第一 ODS(ODS1)的解码开始时间之前和在 DS_n 中的第一 PDS (PDS1) 变为有效时的时间 (PTS (DS_n[PDS1]) 之前，将 PCS 装载到再现装置上的目标缓冲器。因而，必须将 DTS 设置成满足以下关系式，

$$\text{DTS}(\text{DSn}[\text{PCS}]) \leq \text{DTS}(\text{DSn}[\text{ODS1}])$$

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

此外，DS_n 中的 PCS 的 PTS 用以下关系式表示，

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

“decodeduration (DSn)”表示对用于更新 PCS 的所有图形目标进行解码所持续的时间。解码持续时间不是固定值，但也不随再现装置和安装到再现装置上的器件或软件的状态而改变。当用于构成 DSn.PCSn 屏幕的目标是 DSn.PCSn.OBJ[j]时，decodeduration (DSn) 受到 (i) 清除窗口所需的时间、(ii) 用于解码 DSn.PCSn.OBJ 的解码持续时间和 (iii) 用于写 DSn.PCSn.OBJ 所需的时间的影响。当设置 Rd 和 Rc 时，decode_duration (DSn) 总是相同的。因此，通过计算正在执行的这些持续时间的长度来计算 PTS。

decode_duration 的计算是在图 18 所示的程序为基础进行的。图 19、20A 和 20B 是示意性表示该程序的算法的流程图。下面参照附图介绍关于 decode_duration 的计算。在图 19 所示的流程图中，首先，调用 PLANEINITIALZE 函数(图 19 中的步骤 S1)。PLANEINITIALZE 函数用于调用计算使图形平面初始化所需的时间周期的功能，用于再现 DS。在图 19 中的步骤 S1 中，使用参数 DSn、DSn.PCS.OBJ[0]和 decode_duration 调用该函数。

下面参照图 20A 介绍 PLANEINITIALZE 函数。在该图中，initialize_duration 表示 PLANEINITIALZE 函数的返回值的变量。

图 20 中的步骤 S2 是 if 语句，用于根据 DSn 中的 PCS 中的 page_state 是否表示 Epoch Start 而切换操作。如果 page_state 表示 Epoch Start (DSn.PCS.page_state==epoch_start, 图 18 中的步骤 S2=是)，则清除图形平面所需的时间周期设置为 initialize_duration (步骤 S3)。

当目标缓冲器和图形平面之间的传送速度 Rc 是如上所述的 256,000,000，并且图形平面的总尺寸设置为 video_width*video_height 时，清除所需的时间周期是“video_width*video_height//256000000”。当乘以 90.000Hz 以用 PTS 的时间精度表示时，清除图形平面所需的时间周期是“90000×video_width*video_height//256000000”。这个时间周期加到 initialize_duration 上。

如果 page_state 不表示 Epoch Start (步骤 S2=不), 则对于所有窗口, 清除由 WDS 定义的窗口[i]所需的时间周期加到 initialize_duration 上 (步骤 S4)。当目标缓冲器和图形平面之间的传送速度 Rc 是如上所述 256,000,000 和属于 WDS 的窗口[i]的总尺寸是 ΣSIZE (WDS.WIN[i] 时, 清除所需的时间周期是 “ ΣSIZE (WDS.WIN[i]//256000000 ”。当乘以 90.000Hz 以便以 PTS 的时间精度表示时, 清除属于 WDS 的窗口所需的时间周期是 “ $90000 \times \Sigma\text{SIZE}$ (WDS.WIN[i]//256000000 ”。这个时间周期加到 initialize_duration 上, 结果是, initialize_duration 被返回。上面是 PLANEINITIALZE 函数。

图 19 中的步骤 S5 用于根据 DS_n 中的图形目标的数量是 2 或 1 来切换操作, (图 18 中, if (DS_n.PCS.num_of_object==2, if (DS_n.PCS.num_of_object== 1), 和如果该数量是 1(步骤 S5), 用于解码图形目标的等待时间加到 decode_duration 上(步骤 S6)。等待时间的计算是通过调用 WAIT 函数 (图 18, decode_duration+=WAIT (DS_n, DS.PCS.OBJ[0],decode_duration) 来进行的。该函数是使用设置到 DS_n、DS_n.PCS.OBJ[0]、decode_duration 等参量来调用的, 并且返回值是 wait_duration。

图 20B 是表示 WAIT 函数的操作的流程图。

在该流程图中, 调用者 (invoker) 的 decode_duration 设置为 current_duration。object_definition_ready_time 是设置为 DS 的图形目标的 PTS 的变量。

current_time 是设置为 DS_n 中的 PCS 的 DTS 和 current_duration 的总值的变量。当 object_definition_ready_time 大于 current_time (步骤 S7 为是, 如果 (current_time<object_definition_ready_time)), 作为返回值的 wait_duration 设置为 object_definition_ready_time 和 current_time 之间的差值 (步骤 S8, wait_duration+=object_definition_ready_time-current_time) 。 decode_duration 设置为 WAIT 函数的返回值加到用于重画该窗口所需的时间周期上所得的时间周期, ($90000 * (\text{SIZE} (\text{DSn.WDS.WIN}[0])) // 256000000$)。

上面的解释是关于图形目标的数量是 1 的情况。在图 5 中的步骤 S5 中，判断是否图形目标的数量是 2。如果 DS_n 中图形目标的数量是大于 2（在图 18 中，if DS_n.PCS.num_of_object==2），则使用 PCS 中的 OBJ[0] 作为参量调用 WAIT 函数，并将返回值加到 decode_duration 上（步骤 S10）。

在随后的步骤 S11 中，判断是否 DS_n 的 OBJ[0] 所属的窗口与图形目标 [1] 所属的窗口相同（if (DS_n.OBJ[0]).window_id==DS_n.PCS.OBJ[1].window_id）。如果该窗口是相同的，则使用 OBJ[1] 作为参量调用 WAIT 函数，并将返回值 wait_duration 加到 decode_duration 上（步骤 S12），并且将重画 OBJ[1] 所属的窗口所需的时间（90000*（SIZE（DS_n.WDS.OBJ[0].window_id））//256000000）加到 decode_duration 上（步骤 S13）。

如果判断这些窗口是不同的（步骤 S11，“不同”），则将重画 OBJ[0] 所属的窗口所需的时间（90000*（SIZE（DS_n.WDS.OBJ[0].window_id））//256000000）加到 decode_duration 上（步骤 S15），使用 OBJ[1] 作为参量调用 WAIT 函数，将返回值 wait_duration 加到 decode_duration 上（步骤 S16），并将重画 OBJ[1] 所属的窗口所需的时间（90000*（SIZE（DS_n.WDS.OBJ[0].window_id））//256000000）加到 decode_duration 上（步骤 S17）。

通过上述算法计算 decode_duration。下面介绍设置 OCS 的 PTS 的具体方式。

图 21A 表示一个窗口中包含一个 ODS 的情况。图 21B 和 21C 是时序图，按照图 18 中显示的时间的顺序显示各个值。每个图中的底行“ODS 解码”和中间行“图形平面访问”表示在再现时同时进行的两个操作。上述算法是在假设这两个操作并列进行的情况下进行说明的。

图形平面访问包括清除周期（1）和写周期（3）。清除周期（1）表示清除整个图形平面所需的时间周期（90000×（图形平面的尺寸//256000000）），或者清除图形平面上的所有窗口所需的时间周期（Σ

($90000 \times (\text{窗口}[i] \text{的尺寸} // 256000000)$)。

写周期 (3) 表示显示整个窗口所需的时间周期 ($90000 \times (\text{窗口}[i] \text{的尺寸} // 256000000)$)。

此外, 解码周期 (2) 表示 ODS 的 DTS 和 PTS 之间的时间周期。

清除周期 (1)、解码周期 (2) 和写周期 (3) 的长度可以根据要清除的范围、要解码的 ODS 的尺寸、和要写到图形平面的图形目标的尺寸而变化。为了方便起见, 图中的解码周期 (2) 的开始点与清除周期 (1) 的开始点相同。

图 21B 表示解码周期 (2) 很长和 `decode_duration` 等于解码周期 (2) 和写周期 (3) 的总和的情况。

图 21C 表示清除周期 (1) 很长, 和 `decode_duration` 等于清除周期 (1) 和写周期 (3) 的总和的情况。

图 22A-22C 表示在一个窗口中包括两个 ODS 的情况。图 22B 和 22C 中的解码周期 (2) 表示用于解码两个图形所需的总时间周期。同样, 写周期 (3) 表示用于将两个图形写到图形平面中所需的总时间周期。

即使 ODS 的数量是 2, 也可以利用与图 21 的情况相同的方式计算 `decode_duration`。当用于解码两个 ODS 的解码周期 (3) 很长时, `decode_duration` 等于解码周期 (2) 和写周期 (3) 的总和, 如图 22B 所示。

当清除周期 (1) 很长时, `decode_duration` 等于清除周期 (1) 和写周期 (3) 的总和。

图 23A 表示两个窗口的每个窗口包括一个 ODS 的情况。与以前的情况相同, 当清除周期 (1) 比用于解码两个 ODS 的解码周期 (3) 长时, `decode_duration` 等于清除周期 (1) 和解码周期 (2) 的总和。然而, 当清除周期 (1) 比解码周期 (3) 短时, 可以在解码周期 (2) 结束之前写第一窗口。因而, `decode_duration` 不等于清除周期 (1) 和写周期 (3) 的总和, 或者解码周期 (2) 和写周期 (3) 的总和中的任何一个。

当用于解码第一 ODS 所需的时间周期是写周期 (3) 和用于解码

第二 ODS 所需的时间周期是写周期 (32) 时, 图 23B 表示了解码周期 (2) 比清除周期 (1) 和写周期 (31) 的总和长的情况。在这种情况下, `decode_duration` 等于解码周期 (2) 和写周期 (32) 的总和。

图 23C 表示清除周期 (1) 和写周期 (3) 的总和比解码周期 (2) 长的情况。在这种情况下, `decode_duration` 等于清除周期 (1)、写周期 (31) 和写周期 (32) 的总和。

预先从再现装置的模式得知图形平面的尺寸。而且, 窗口的尺寸和 ODS 的尺寸和数量也可以在创造时得知。因而, 可以发现 `decode_duration` 等于哪个时间周期的组合: 清除周期 (1) 和写周期 (3), 解码周期 (2) 和写周期 (3), 解码周期 (2) 和写周期 (32), 或者清除周期 (1)、写周期 (3) 和写周期 (32)。

通过在计算上述 `decode_duration` 的基础上设置 ODS 的 PTS, 可以以高精度与图像数据同步地显示图形。通过定义窗口和限制重画该窗口的区域使这种高精度的同步显示变为可能。因此, 将窗口的概念引入创作环境 (authoring environment) 中具有重大意义。

下面将介绍 DS_n 中的 WDS 的 DTS 和 PTS 的设置。WDS 的 DTS 可以设置成满足以下关系式,

$$DTS (DS_n[WDS]) \geq DTS (DS_n[PCS])$$

另一方面, DS_n 中的 WDS 的 OTS 表示开始写到图形平面的截止时间。由于写到图形平面上的窗口就足够了, 开始写到图形平面的时间通过从用于写 WDS 所需的时间周期减去由 PCS 的 PTS 表示的时间长度来确定。当 WDS 的总尺寸是 $\Sigma SIZE (WDS.WIN[i])$ 时, 用于清除和重画所需的时间周期是 “ $\Sigma SIZE (WDS.WIN[i]) // 256000000$ ”。当以 90.000KHz 的时间精度表示时, 该时间是 “ $90000 \times \Sigma SIZE (WDS.WIN[i]) // 256000000$ ”。

相应地, 可以通过以下关系式计算 WDS 的 PTS,

$$\text{PTS}(\text{DSn}[\text{WDS}]) = \text{PTS}(\text{DSn}[\text{PCS}]) - 90000 \times \Sigma \text{SIZE}(\text{WDS.WIN}[i]) \\ // 256000000$$

在 WDS 中表示的 PTS 是截止时间，并可以比 PTS 更早地开始写到图形平面。换言之，如图 23 所示，一旦对一个窗口中的要显现的 ODS 进行解码，通过解码获得的图形目标的写操作可以在此时开始。

如上所述，可以使用加到 WDS 的 DTS 和 PTS 将该窗口分配给 AVClip 的再现时间轴上的任何时间点。

下面参照图 24-25 所示的具体例子介绍在这些设置的基础上显示设置中的 DTS 和 PTS 的设置例子。该例子是关于通过四次写到图形平面来显示字幕和用于显示两个字幕“what is blu-ray”和“blu-ray is everywhere”中的每一个进行更新的情况。图 24 表示在该例中更新随时间的变化。直到点 t1 为止，显示“what”，在 t1 之后直到 t2 显示“what is”，然后在 t3 显示“what is blu-ray”。在已经出现第一字幕的整个句子之后，在 t4 显示第二字幕“blu-ray is everywhere”。

图 25A 表示为了进行上述更新所描述四个显示设置。DS1 包括用于控制在 t1 的更新的 PCS1.2、用于着色的 PDS1、对应字幕“what is blu-ray”的 ODS1、和作为 DS1 的结束代码的 END。

DS2 包括用于控制在 t2 的更新的 PCS1.2 和 END。DS3 包括用于控制在 t3 的更新的 PCS1.3 和 END。DS4 包括用于控制在 t2 的更新的 PCS2、用于颜色转换的 PDS2、对应字幕“blu-ray is everywhere”的 ODS2 和 END。

参见图 25B 中的时序图，解释四个显示设置中的每个功能段的 DTS 和 PTS 的设置。

时序图中的再现时间轴与图 24 中的时间轴相同。在图 25A 的时序图中，PTS (PCS1.1)、PTS (PCS1.2)、PTS (PCS1.3) 和 PTS (PCS2) 分别设置在用于显示“what”的显示点 t1、用于显示“what is”的显示点 t2、用于显示“what is blu-ray”的显示点 t3 和用于显示“blu-ray is everywhere”的显示点 t4。每个 PTS 按照上述那样设置，因为必须在每个字幕的显示点上控制，例如在每个 PCS 中所述的裁剪。

PTS (ODS1) 和 PTS (ODS2) 设置成表示通过分别从分别由 PTS (PCS1.1) 和 PTS (PCS2) 表示的点减去 `decode_duration` 计算的点, 这是因为 PTS (PCS) 必须设置成满足以下公式,

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

在图 25B 中, PTS (ODS2) 设置成表示点 t4 之前出现的点 t5, PTS (ODS1) 设置成表示在点 t1 之前出现的点 t0。

DTS (ODS1) 和 DTS (ODS2) 设置成表示通过从分别由 PTS (ODS1) 和 PTS (ODS2) 表示的点减去 `decode_duration` 计算的点, 这是因为 DTS (ODS) 必须设置成满足以下等式,

$$\text{PTS}(\text{DS}[\text{ODSj}]) = \text{DTS}(\text{DSn}[\text{ODSj}]) + 90000 \times (\text{SIZE}(\text{DSn}[\text{ODSj}]) // \text{Rd})$$

在图 25B 中, PTS (ODS2) 设置成表示在点 t0 之前出现的点 t5, PTS (ODS1) 设置成表示在点 t0 之前出现的点。这里满足由 $\text{DTS}(\text{ODS2}) = \text{PTS}(\text{ODS1})$ 表示的关系。

通过先显示的前一个 ODS 的 PTS 之后立即设置 ODS 的 PTS, 再现装置进行这样的操作, 其中将该 ODS 读出到存储器中以便重写前一个 ODS, 因此可以通过小尺寸存储器进行再现处理。通过实现这种再现处理, 可用于再现装置的存储器尺寸的选择变得更宽了。

PCS1.1 的 DTS 设置成为 $\text{DTS}(\text{PCS1.1}) = \text{DTS}(\text{ODS1})$, 因为 PCS1.1 的 DTS 的值可以是由 DTS (ODS1) 表示的点之前的任何点。

ODS1 的 PTS、ODS2 的 DTS 以及 PCS1.2、PCS1.3 和 PCS2 的 PTS 设置在点 0, 以便满足由以下关系式表示的关系,

$$\begin{aligned} \text{PTS}(\text{ODS1}) &= \text{DTS}(\text{ODS2}) = \text{PTS}(\text{PCS1.2}) \\ &= \text{PTS}(\text{PCS1.3}) = \text{PTS}(\text{PCS2}) \end{aligned}$$

这是因为用于 PCS1.2 和 PCS1.3 的 DTS 的值可以是在由 PTS (PCS1.3)表示的点之前的任何点,并且 PCS2 的 DTS 可以是由 DTS (PCS2)表示的点之前的任何点。

如上所述,通过同时读出一个以上的 PCS,在完成了在前 PCS 的更新后,就可以立即进行后一个 PCS 的更新。

PCS 的 DTS 和 PTS 以及 ODS 的 DTS 和 PTS 满足由上述公式表示的关系。因而,可以使这些值设置为 $DTS(ODS2) = PTS(ODS1)$ 或者 $PTS(ODS1) = DTS(ODS2) = PTS(PCS1.2) = PTS(PCS1.3) = DTS(PCS2)$ 。通过对时间戳的这种设置,可以调节解码时的负载增加或需要更多缓冲器的周期的长度。这种调节扩大了在再现期间的控制的可能性,并且有利于进行创造或制造再现装置的人。

上述显示设置 (PCS、WDS、PDS、ODS) 的数据结构是用编程语言中描述的分类结构的例子。进行创作的作者可以通过根据以 Blu-ray 盘预记录格式提供的语法来描述分类结构而在 BD-ROM 上获得该数据结构。

接着,下面介绍根据本发明的再现装置的实际例子。图 26 表示根据本发明的再现装置的内部结构。根据本发明的再现装置是在图中所示的内部结构基础上可工业上制造的。根据本发明的再现装置主要由以下三部分构成:系统 LSI、驱动器件、和微型计算机系统,并且可以通过将这三个部分安装到该装置的壳体和基板上工业上制造该再现装置。系统 SI 是集成电路,其中集成了用于执行该再现装置的功能的各个处理单元。按照上述方式制造的再现装置包括 BD 驱动 1、读取缓冲器 2、PID 滤波器 3、传送缓冲器 4a-4c、外围电路 4d、视频解码器 5、视频平面 6、音频解码器 7、图形平面 8、CLUT 单元 9、加法器 10、图形解码器 12、编码数据缓冲器 13、外围电路 13a、流图形处理器 14、目标缓冲器 15、组合缓冲器 16、和图形控制器 17。

BD 驱动 1 执行 BD-ROM 的装载/读取/弹出和访问 BD-ROM。

读取缓冲器 2 是用于按照先进先出的顺序储存从 BD-ROM 读取的 TS 信息包的 FIFO 存储器。

PID 滤波器 3 对从读取缓冲器 2 输出的一个以上 TS 信息包进行滤波。通过 PID3 进行的过滤是只将具有所希望 PID 的 TS 信息包写到传送缓冲器 4a-4c。对于通过 PID 滤波器 3 进行的滤波不需要缓冲，因而，在没有延迟的情况下将输入到 PID 滤波器 3 的 TS 信息包写到传送缓冲器 4a-4c 上。

传送缓冲器 4a-4c 是用于按照先进先出的顺序储存从 PID 滤波器 3 输出的 TS 信息包。从传送缓冲器 4a-4c 输出 TS 信息包的速度是速度 Rx。

外围电路 4d 是用于将从传送缓冲器 4a-4c 读出的 TS 信息包转换成功能段的布线逻辑电路。通过该转换获得的功能段被储存在编码数据缓冲器 13 中。

视频解码器 5 对从 PID 滤波器 3 输出的一个以上的 TS 信息包解码为解压缩图像，并写到视频平面 6。

视频平面 6 是用于运动图像的平面储存器。

音频解码器 7 对从 PID 滤波器 3 输出的 TS 信息包进行解码并输出解压缩音频数据。

图形平面 8 是具有用于一个屏幕的区域的平面储存器，并且能储存一个屏幕的解压缩图形。

CLUT 单元 9 根据由 PDS 表示的 Y、Cr 和 Cb 的值来转换储存在图形平面 8 中的解压缩图形的索引颜色。

加法器 10 将已经通过 CLUT 单元 9 进行了颜色转换的解压缩图形乘以由 PDS 表示的 T 值（透明度），加上被储存在每个像素的视频平面中的分解图像数据，然后获得和输出形成的图像。

图形解码器 12 解码图形流，从而获得分解图形，并将分解图形作为图形目标写到图形平面 8 中。通过解码图形流，字幕和菜单出现在屏幕上。图形解码器 12 包括编码数据缓冲器 13、外围电路 13a、流图形处理器 14、目标缓冲器 15、组合缓冲器 16 和图形控制器 17。

编码数据缓冲器 13 是存储功能段以及 DTS 和 PTS 的缓冲器。该功能段是通过从储存在传送缓冲器 4a-4c 中的传送流中的每个 TS 信息包除去 TS 信息报头部和 PES 信息报头部，并通过顺次设置有效载

荷获得的。在形成 PES 信息包之间的联系之后，储存除去的 TS 信息报头部和 PES 信息报头部中的 PTS 和 DTS。

外围电路 13a 是实现编码数据缓冲器 13 和流图形处理器 14 之间的传送、编码数据缓冲器 13 和组合缓冲器 16 之间的传送的布线逻辑电路。在传送操作中，当在当前时间是由 ODS 的 DTS 表示的时间时，将 ODS 从编码数据缓冲器 13 传送到流图形处理器 14。在当前时间是由 PCS 和 PDS 的 DTS 表示的时间时，将 PCS 和 PDS 传送到组合缓冲器 16。

流图形处理器 14 解码 ODS，并将通过解码获得的索引颜色的解压缩图形作为图形目标写到目标缓冲器 15。通过流图形处理器 14 进行的解码在对应 ODS 的 DTS 的时间开始，并在由对应 ODS 的 PTS 表示的解码结束时间之前结束。图形目标的解码速度 R_d 是流图形处理器 14 的输出速度。

目标缓冲器 15 是对应 ETSI EN 300 743 标准中的像素缓冲器的缓冲器，并且设置通过流图形处理器 14 进行的解码获得的图形目标。目标缓冲器 15 必须设置为与图形平面 8 的两倍或四倍一样大，这是因为在进行滚动效果的情况下，目标缓冲器 15 需要储存与图形平面的两倍或四倍一样大的图形目标。

组合缓冲器 16 是其中设置 PCS 和 PDS 的存储器。

图形控制器 17 解码设置在组合缓冲器 16 中的 PCS，并在该 PCS 基础上进行控制。用于进行控制的定时是以附加在 PCS 的 PTS 为基础的。

接着，下面介绍传送速度和构成 PID 滤波器 3、传送缓冲器 4a-4c、图形平面 8、CULT 单元 9、编码数据缓冲器 13 和图形控制器 17 的缓冲器尺寸的推荐值。图 27 表示写速度 R_x 、 R_c 和 R_d 、图形平面 8、编码数据缓冲器 13、目标缓冲器 15 和组合缓冲器 16 的尺寸。

目标缓冲器 15 和图形平面 8 之间的传送速度 R_c 是本实施例的再现装置中的最高速度，并由窗口尺寸和帧速度计算为 $256 \text{ Mbps} (=500 \text{ Kbytes} \times 29.97 \times 2)$ 。

与 R_c 不同，流图形处理器 14 和目标缓冲器 15 之间的传送速度

Rd（像素解码速度）不必在每个视频帧循环中被更新，并且 Rd 是 Rc 的 1/2 或 1/4 就足够了。因而，Rd 是 128 Mbps 或 64 Mbps。

传送缓冲器 4a-4c 和编码数据缓冲器 13 之间的传送缓冲器泄漏速度 Rx 是压缩状态下的 ODS 的传送速度。因而，乘以压缩率的传送速度 Rd 足以用于传送缓冲泄漏速度 Rx。假定 ODS 的压缩率是 25%，16 Mbps (=64 Mbps×25%) 就足够了。

图中所示的传送速度和缓冲器尺寸是最小标准，并且也可以设置为较高速度和较大尺寸。

在上述构成的再现装置中，每个元件执行流水线结构中的解码操作。

图 28 是表示通过再现装置进行的流水线处理的时序图。图中的第 5 行是 BD-ROM 中的显示设置，第 4 行表示从 PCS、WDS、PDS 和 ODS 到编码数据缓冲器 13 的读取周期。第 3 行表示由流图形处理器 14 执行的每个 ODS 的解码周期。第 1 行表示图形控制器 17 执行的操作。

附加于 ODS1 和 ODS2 的 DTS（解码开始时间）分别表示图中的 t31 和 t32。由于解码开始时间由 DTS 设置，因此每个 ODS 需要读出到编码数据缓冲器 13。因而，在对 ODS1 解码到编码数据缓冲器 13 的解码周期 dp1 之前完成 ODS1 的读取。而且，在对 ODS2 进行解码到编码数据缓冲器 13 的解码周期 dp2 之前完成 ODS2 的读取。

另一方面，附加于 ODS1 和 ODS2 的 PTS（解码结束时间）分别表示图中的 t32 和 t33。通过流图形处理器 14 执行的 ODS1 的解码是在 t32 之前完成，并且 ODS2 的解码在 t33 表示的时间之前完成。如上所述，在 ODS 的 DTS 表示的时间之前，流图形处理器 14 将 ODS 读到编码数据缓冲器 13，并且在 ODS 的 PTS 表示的时间之前，对读到编码数据缓冲器 13 的 ODS 进行解码，而且将解码的 ODS 写到目标缓冲器 15。

在图中第 1 行的周期 cd1 表示图形控制器 17 清除图形平面所需的周期。而且，周期 td1 表示将在目标缓冲器上获得的图形目标写到图形平面 8 所需的周期。WDS 的 PTS 表示开始写的截止时间，PCS 的

PTS 表示写的结束时间和用于显示的定时。在由 PCS 的 PTS 表示的时间，在图形平面 8 上获得构成交互式屏幕的解压缩图形。

CLUT 单元 9 进行解压缩图形的颜色转换和加法器 10 进行储存在视频平面 6 中的分解图形和分解图像的组合之后，获得组合图像。

在图形解码器 12 中，流图形处理器 14 连续地进行解码，同时图形控制器 17 进行图形平面 8 的清除。通过上述流水线处理，可以进行图形的即时显示。

在图 28 中，解释了在完成 ODS 的解码之前图形平面的清除结束的情况。图 29 表示在完成图形平面的清除之前 ODS 解码结束的情况中的流水线处理中的时序图。在这种情况下，不可能在完成 ODS 的解码时写到图形平面。当完成图形平面的清除时，就可以将通过解码获得的图形写到图形平面。

接着，下面介绍如何运行控制单元 20 和图形解码器 12。控制单元 20 是通过写一段程序并使一般的 CPU 执行该程序来运行的，该程序执行图 30 中所示的操作。下面参照图 30 介绍通过控制单元 20 执行的操作。

图 30 是表示功能段的装载操作的处理的流程图。在该流程图中，段 K 是可变的，表示在再现 AVClip 时读出的每个段（PCS、WDS、PDS 和 ODS）。忽略标记是确定段 K 是否被忽略或装载的标记。该流程图具有循环结构，其中首先忽略标记被初始化到 0，然后对每个段 K 重复执行步骤 S21-S24 和步骤 S27-S31（步骤 S25 和步骤 S26）。

步骤 S21 是用于判断段 K 是否是 PCS，如果段 K 是 PCS，则进行步骤 S27 和步骤 S28 中的判断。

步骤 S22 是用于判断忽略标记是否为 0。如果忽略标记为 0，则操作移动到步骤 S23，如果忽略标记是 1，则操作移动到步骤 S24。如果忽略标记是 0（步骤 S22 中为是），则在步骤 S23 中将段 K 装载到编码数据缓冲器 13 中。

如果忽略标记是 1（步骤 S22 中是否），则在步骤 S24 中忽略段 K。由此，由于步骤 S22 是否（步骤 S24），则忽略属于 DS 的其余所有功能段。

如上所述，段 K 是否忽略或装载是由忽略标记来确定的。步骤 S27-S31、S34 和 S35 是用于设置忽略标记的步骤。

在步骤 S27 中，判断段 K 的 `segment_type` 是否是 Acquisition Point。如果段 K 是 Acquisition Point，则操作移动到步骤 S28，如果段 K 是 Epoch Start 或 Normal Case，则操作移动到步骤 S31。

在步骤 S28 中，判断是否在前 DS 存在于图形解码器 12 中的任何缓冲器中（编码数据缓冲器 13、流图形处理器 14、目标缓冲器 15、和组合缓冲器 16）。在步骤 S27 中的判断为是时进行步骤 S28 中的判断。在前 DS 不存在于图形解码器 12 中的情况表示进行跳跃操作的情况。在这种情况下，从作为 Acquisition Point 的 DS 开始显示，因此操作转移到步骤 S30（步骤 S28 中是否）。在步骤 S30 中，忽略标记设置为 0，并且操作移动到步骤 S22。

在前 DS 存在于图形解码器 12 中的情况指的是进行正常再现的情况。在这种情况下，操作移动到步骤 S29（步骤 S28 中为是）。在步骤 S29 中，忽略标记设置为 1，并且操作移动到步骤 S22。

在步骤 S31 中，判断 PCS 的 `segment_type` 是否是 Normal Case。如果 PCS 是 Normal Case，则操作移动到步骤 S34，如果 PCS 是 Epoch Start，则在步骤 S30 中将忽略标记设置为 0。

在步骤 S34 中，与步骤 S28 中一样，判断在前 DS 是否存在于图形解码器 12 中的任何缓冲器中。如果在前 DS 存在，则忽略标记设置为 0（步骤 S30）。如果在前 DS 不存在，不可能获得足够的功能段来构成交互式屏幕，忽略标记设置为 1（步骤 S35）。

通过用上述方式设置忽略标记，在在前 DS 不存在于图形解码器 12 中时，忽略构成 Normal Case 的功能段。

用如图 31 所示的对 DS 进行多路复用的情况作为例子介绍如何进行 DS 的读取的方式。在图 31 的例子中，将三个 DS 与运动图像多路复用。DS1 的 `segment_type` 是 Epoch Start，DS10 的 `segment_type` 是 Acquisition Point，并且 DS20 的 `segment_type` 是 Normal Case。

假设在对三个 DS 和运动图像进行多路复用的 AVClip 中，执行由箭头 am1 表示的图像数据 pt10 跳跃操作，并且 DS10 最接近于跳跃

目标,因此 DS10 是在图 30 中的流程图中表示的 DS。尽管在步骤 S27 中判断 `segment_type` 是 Acquisition Point, 但是因为没有在前 DS 存在于编码数据缓冲器 13 中而将忽略标记设置为 0, 并且将 DS10 装载到再现装置的编码数据缓冲器 13, 如图 32 中的箭头 md1 所示。另一方面, 在跳跃目标处于 DS10 (图 31 中的箭头 am2 所示) 之后的情况下, DS20 将被忽略, 因为 DS20 是 Normal Case 显示设置, 并且因为在前 DS 不存在于编码数据缓冲器 13 中 (图 32 中的箭头 md2)。

图 33 表示正常再现中的 DS1、DS10 和 DS20 的装载。其 PCS 的 `segment_type` 是 Epoch Start 的 DS1 被原样装载到编码数据缓冲器 13 中 (步骤 S23)。然而, 由于其 PCS 的 `segment_type` 是 Acquisition Point 的 DS10 的忽略标记设置为 1 (步骤 S29), 因此构成 DS10 的功能段被忽略并且不装载到编码数据缓冲器 13 (图 34 中的箭头 rd2, 步骤 S24)。此外, 由于 DS20 的 PCS 的 `segment_type` 是 Normal Case, 因此将 DS20 装载到编码数据缓冲器 13 (图 34 中的箭头 rd3)。

接下来, 解释由图形控制器 17 执行的操作。图 35-37 表示由图形控制器 17 执行的操作的流程图。

步骤 S41-S44 是流程图的主程序的步骤并等待在步骤 S41-S44 中描述的任何事件发生。

步骤 S41 是判断当前再现时间是否是由 PCS 的 DTS 表示的时间, 如果该判断为是, 则执行步骤 S45-S53 中的操作。

步骤 S45 是判断 OCS 的 `Composition_state` 是否是 Epoch Start, 如果判断是 Epoch Start, 则在步骤 S46 中清除全部图形平面 8。如果判断是 Epoch Start 以外的其它情况, 则清除由 WDS 的 `window_horizontal_position`、`window_vertical_position`、`window_width` 和 `window_height` 表示的窗口。

步骤 S48 是在步骤 S46 或在步骤 S47 中执行清除之后进行的步骤, 并判断是否经过了由任何 ODSx 的 PTS 表示的时间。任何 ODSx 的解码能够在清除结束之前已经完成, 因为整个图形平面 8 的清除花费了时间。因此, 在步骤 S48 中, 判断清除结束之前是否任何 ODSx 的解码已经完成。如果判断为否, 则操作返回到主程序。如果已经经

过了由任何 ODSx 的 PTS 表示的时间，则执行步骤 S49-S51 中的操作。在步骤 S49 中，判断 object_cropped_flag 是否为 0，如果该标记为 0，则将图形目标设置为“**No-display(不显示)**”（步骤 S50）。

如果在步骤 S49 中判断该标记不是 0，则将根据 object_cropping_horizontal_position、object_cropping_vertical_position、cropping_width 和 cropping_height 而裁剪的目标写到图形平面 8 中的窗口中的由 object_cropping_horizontal_position 和 object_cropping_vertical_position 表示的位置上（步骤 S51）。通过上述操作，在该窗口中呈现一个或多个图形目标。

在步骤 S52 中，判断是否经过了对应另一 ODSy 的 PTS 的时间。在将 ODSx 写到图形平面 8 中时，如果已经完成了 ODSy 的解码，则 ODSy 变为 ODSx（步骤 S53），并且操作移动到步骤 S49。由此，也对另一 ODS 执行从步骤 S49-51 的操作。

接下来，下面参照图 36 介绍步骤 S42 和步骤 S54-S59。

在步骤 42 中，判断当前再现点是否处于 WDS 的 PTS。如果该判断是当前再现点处于 WDS 的 PTS，则在步骤 S54 中判断该窗口的数量是否是一个。如果该判断是两个，则操作返回到主程序。如果判断是一个，则执行步骤 S55-S59 的循环处理。在该循环处理中，对在该窗口中显示的两个图形目标的每个执行步骤 S55-S59 中的操作。在步骤 S57 中，判断 object_cropped_flag 是否表示 0。如果它表示 0，则不显示该图形（步骤 S58）。

如果它不表示 0，则将根据 object_cropping_horizontal_position、object_cropping_vertical_position、cropping_width 和 cropping_height 而裁剪的目标写到图形平面 8 中的窗口中的由 object_cropping_horizontal_position 和 object_cropping_vertical_position 表示的位置上（步骤 S59）。通过重复上述操作，在该窗口中呈现一个或多个图形目标。

在步骤 S44 中，判断当前再现点是否处于 PDS 的 PTS。如果该判断是当前再现点处于 PDS 的 PTS，则在步骤 S60 中判断是否 pallet_update_flag 是否是一。如果该判断是一，则将由 pallet_id 表示

的 PDS 设置在 CLUT 单元中（步骤 S61）。如果判断是 0，则跳过步骤 S61。

之后，CLUT 单元进行将与运动图像组合的图形平面 8 上的图形目标的颜色转换（步骤 S62）。

接着，下面参照图 37 介绍步骤 S43 和步骤 S64-S66。

在步骤 S43 中，判断当前再现点是否处于 ODS 的 PTS。如果判断当前再现点是处于 ODS 的 PTS，则在步骤 S63 中判断该窗口的数量是否是两个。如果判断是一个，则操作返回到主程序。如果判断是两个，则执行步骤 S64-S66 中的操作。在步骤 S64 中，判断 `object_cropped_flag` 是否表示 0。如果它表示 0，则不显示图形（步骤 S65）。

如果它不表示 0，则将根据 `object_cropping_horizontal_position`、`object_cropping_vertical_position`、`cropping_width` 和 `cropping_height` 而裁剪的目标写到图形平面 8 中的窗口中的由 `object_cropping_horizontal_position` 和 `object_cropping_vertical_position` 表示的位置上（步骤 S66）。通过重复上述操作，在每个窗口中呈现图形目标。

上面的说明是关于 PCS 的 DTS 和 PTS 以及属于 DS_n 的 ODS 的 DTS 和 PTS。没有说明 PDS 的 DTS 和 PTS 以及 END 的 DTS 和 PTS。首先，说明属于 DS_n 的 PD 的 DTS 和 PTS。

关于属于 DS_n 的 PDS，如果在第一 ODS 的解码开始点之后（DTS（DS_n[ODS1]））通过将 PDS 装载到组合缓冲器 16 中就足够了，在 CLUT 单元 9 中该 PDS 可由 PCS 获得（DTS（DS_n[PCS]））。因而，需要将 DS_n 中的每个 PDS（PDS1-PDS_{LAST}）的 PTS 的值设置成满足以下关系式，

$$\begin{aligned} \text{DTS (DS}_n\text{[PCS])} &\leq \text{PTS (DS}_n\text{[PDS1])} \\ \text{PTS (DS}_n\text{[PDS}_j\text{])} &\leq \text{PTS (DS}_n\text{[PDS}_{j+1}\text{])} \leq \text{PTS (DS}_n\text{[PDS}_{\text{LAST}}\text{])} \\ \text{PTS (DS}_n\text{[PDS}_{\text{LAST}}\text{])} &\leq \text{DTS (DS}_n\text{[ODS1])} \end{aligned}$$

注意在再现期间没有参考 PDS 的 DTS，将 ODS 的 DTS 设置为与 PDS 的 PTS 相同的值，以便满足 MPEG2 标准。

下面介绍在 DTS 和 PDS 设置成满足上述关系式时再现装置的流水线处理中的 DTS 和 PTS 的作用。图 38 表示在 PDS 的 PTS 基础上的再现装置的流水线。图 38 是以图 26 为基础的。图 38 中的第一行表示设置 CLUT 单元 9 中的 ODS。第一行下面是与图 26 中的第一到第五行相同的。在传输 PCS 和 WDS 之后和 ODS1 解码之前，将 PDS1-PDSLlast 设置到 CLUT 单元 9，因而对 CLUT 单元 9 的 PDS1-PDSLlast 的设置是在由 ODS1 的 DTS 表示的点之前设置的，如箭头 up2 和 up3 所示。

如上所述，在 ODS 的解码之前进行 PDS 的设置。

接着，介绍 DS_n 中的显示设置段的 END 的 PTS 的设置。属于 DS_n 的 END 表示 DS_n 的结束，因而必须使 END 的 PTS 表示 ODS₂ 的解码结束时间。解码结束时间由 ODS₂ (ODSLast) 的 PTS (PTS (DS_n (ODSLast)) 表示，因此要求 END 的 PTS 设置为满足以下等式的值，

$$DTS (DS_n[END]) = PTS (DS_n[ODSLast])$$

鉴于 DS_n 和属于 DS_{n+1} 的 PCS 之间的关系，在第一 ODS(ODS₁) 的装载时间之前将 DS_n 中的 PCS 装载到组合缓冲器 16，因此 END 的 PTS 应该是在 DS_n 中的 PCS 的装载时间之后和属于 DS_{n+1} 的 PCS 的装载时间之前。因而，需要使 END 的 PTS 满足下列关系，

$$DTS (DS_n[PCS]) \leq PTS (DS_n[END]) \leq DTS (DS_{n+1} [PCS])$$

另一方面，第一 ODS(ODS₁) 的装载时间是在最后 PDS(PDSLlast) 的装载时间之前，因此 END 的 PTS (PTS (DS_n[END])) 应该是在属于 DS_n 的 PDS 的装载时间 (PTS (DS_n[PDSLlast])) 之后。相应地，END 的 PTS 必须满足以下关系，

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

下面是关于再现装置的流水线处理中的 END 的 PTS 的重要性的重要性的说明。图 39 是表示再现装置的流水线处理中的 END 的重要性的示意图。图 39 是以 26 为基础的，图 39 中的每行基本上与图 26 中的相同，除了图 39 中的第一行表示组合缓冲器 16 的内容之外。此外，在图 39 中，示出了 2 个显示设置，DSn 和 DSn+1。DSn 中的 ODSLlast 是 A-ODS 的最后 ODSn，相应地，由 END 的 PTS 表示的点是在 DSn+1 中的 PCS 的 DTS 之前。

在 END 的 PTS 之前，可以发现在再现期间在 DSn 中装载 ODS 是何时完成的。

注意尽管在再现期间不参照 END 的 DTS，但是 END 的 DTS 设置为与 END 的 PTS 相同的值，以便满足 MPEG2 标准。

如上所述，根据本实施例，将一部分图形平面规定为用于显示图形的窗口，因此再现装置不必在整个平面呈现图形。再现装置可以只为预定尺寸的窗口中，如图形平面的 25%-33%，呈现图形。由于不需要呈现窗口中的图形以外的图形，因此用于再现装置中的软件的负载减少了。

即使在对例如图形平面的 1/4 进行图形更新的最坏情况下，可以通过再现装置以预定传送速度如 256Mbps 进行对图形平面的写操作，并且通过设置窗口的尺寸以便确保与图像的同步显示，从而能够与图像同步地显示图形。

因此，由于很容易保证同步显示，因此可以实现用于各种再现装置的高分辨率字幕显示。

第二实施例

在第一实施例中，该窗口的尺寸设置为整个图形平面的 1/4，并向图形平面进行写操作的写速度 Rc 设置为 256 Mbps，以便为每个视频帧更新图形。此外，通过设置更新速度为视频帧速度的 1/2 或 1/4，可以更新大尺寸的图形。然而，当更新速度是视频帧速度的 1/2 或 1/4，

需要经过 2 或 4 个帧来写到图形平面。当提供一个图形平面时，在写图形的 2 或 4 个帧期间图形的写处理对于用户来说是可见的。在这种情况下，不能有效地实现显示效果，如从一个图形到较大图形立刻的转换。因此，在第二实施例中，提供两个图形平面。图 40 表示根据第二实施例的再现装置的内部结构。与根据图 24 和 25 中的再现装置相比，图 40 中的再现装置的新特征在于图 40 中的再现装置具有两个图形平面（图中的图形平面 81 和图形平面 82），并且这两个图形平面构成双重缓冲器。因而，可以对图形平面之一写入，同时从另一图形平面进行读取。此外，根据第二实施例的图形控制器 17 切换在由 PCS 的 PTS 表示的点读出的图形平面。

图 41 示意性地表示读出和写到构成双重缓冲器的图形平面的操作。上面的行表示图形平面 81 的内容，下面的行表示图形平面 82 的内容。图中示出了从第一帧到第五帧的每帧的两个图形平面的内容（从左到右）。被粗线环绕的每帧的图形平面 81 和 82 的一部分是读出的目标。在图中，在图形平面 81 中含有脸谱标记，并且该脸谱标记将被处于目标缓冲器 15 中的太阳标记来代替。太阳标记的尺寸是 4 兆比特，这是目标缓冲器 15 的最大尺寸。

为了以写到图形平面的速度（ $R_c=256$ Mbps）将太阳标记写到图形平面 82 中，需要经过 4 帧，直到完成写操作，并且在第一帧期间只将太阳标记的 1/4 写到图形平面 82 中，在第二帧期间写 2/4，在第三帧期间写 3/4。然而，由于图形平面 81 是要在屏幕上显示的目标，因此将太阳标记写到该图形平面中的过程对于用户来说是不可见的。在第五帧，当显示的目标转换到图形平面 82 时，图形平面 82 的内容变得对于用户来说是可见的。因此，完成了从脸谱标记向太阳标记的转换。

如上所述，根据第二实施例，即使在用四帧将大尺寸图形写到图形平面时，也可以将屏幕中的显示立即转换到另一图形，因此在屏幕中一次显示如 Credits、电影概要或警告时是有用的。

第三实施例

第三实施例涉及 BD-ROM 的制造工艺。图 42 是表示根据第三实施例的 BD-ROM 的制造工艺。

BD-ROM 的制造包括用于制造材料和记录电影和声音的材料制造步骤 S201、用于使用创作装置产生应用格式的创作步骤 S202、和用于制造 BD-ROM 的主盘和冲压以便完成 BD-ROM 的冲压步骤 S203。

BD-ROM 的创作步骤包括如下的步骤 S204-S209。

在步骤 S204 中,描述 WDS 以便定义显示字幕的窗口,在步骤 S205 中,在一个时间周期中定义该窗口以便该窗口以相同尺寸出现在相同位置上,该时间周期设置为一个时元,并且描述每个时元的 PCS。

利用上述方式获得 OCS 之后,将作为用于字幕的资料的图形转换成 ODS,并在步骤 S206 中通过将 ODS 与 PCS、WDS、和 PDS 组合而获得显示设置。然后,在步骤 S207 中,将显示设置中的每个功能段分割成 PES 信息包,并通过附加时间戳而获得图形流。

最后,在步骤 S208 中,通过对具有分开产生的视频流和音频流的图形流进行多路复用而产生 AVClip。

获得 AVClip 之后,通过将 AVClip 调整成 BD-ROM 格式,完成了应用格式。

其它问题

上面的说明没有展示根据本发明的所有实施例。本发明也可以通过以下所示的修改例来实现。在本申请的权利要求书中所述的发明包括上述实施例以及修改例的扩展或普遍化。尽管所述扩展和普遍化的程度是以申请时的相关技术的技术水平的特性为基础的,但是根据本申请的权利要求的发明反映了解决常规技术中的技术问题的手段,因此本发明的范围不超过本领域技术人员为了解决常规技术中的技术问题而能认识到的技术范围。因此,根据本申请权利要求的发明实质上对应本发明的细节的说明。

(1) 在说明所有实施例时使用了 BD-ROM。然而,本发明的特征在于在介质中记录的图形流,并且这一特征不取决于 BD-ROM 的物理性能。能储存图形流的任何记录介质都可实现本发明。这种记录

介质的例子除了内置硬盘以外还包括：光盘，如 DVD-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R 和 CD-RW；磁性光盘，如 PD 和 MO；半导体存储卡，如小型快闪卡、智能介质、存储棒、多媒体卡、和 PCM-CIA 卡；和磁盘，如软盘、超级盘、Zip、和 Clikl，和可拆卸硬盘驱动器，如 ORB、Jaz、SparQ、SyJet、EZFley、和微驱动器。

(2) 在上面所有实施例中所描述的再现装置对在 BD-ROM 中记录的 AVClip 进行解码并将解码的 AVClip 输出到 TV。然而，也可以通过只包括 BD-ROM 驱动器的再现装置来实现本发明，其中 TV 设有其他元件。在这种情况下，再现装置和 TV 可以经 IEEE1394 而连接在一起，从而构成家庭网络。而且，尽管这些实施例中的再现装置是通过连接到 TV 而使用的，但是该再现装置也可以多合一 TV 和再现装置。此外，形成每个实施例的再现装置中的处理的主要部分的 LSI（集成电路）可以单独投入实现。这种再现装置和 LSI 都在本说明书中描述了，因此，无论采取怎样的工作示例，以根据第一实施例的再现装置的内部结构为基础制造再现装置能够实现本发明。而且，根据本发明的再现装置的是否作为礼物或收益的转送、出租和进口都被认为是实施本发明。通过店面显示和发布小册子向普通用户提供这种转送和出租也都被认为是实施了本发明。

(3) 由流程图所示的程序执行的信息处理是使用硬件资源实现的，因而，其处理示于每个流程图中的程序本身也是一个本发明。尽管所有上述实施例描述了根据本发明的程序内置于再现装置中，但是根据第一实施例的程序可以独立实施。独立实施该程序的例子包括 (i) 制造该程序，(ii) 作为礼物或利益转送这些程序，(iii) 出租这些程序，(iv) 进口这些程序，(v) 将这些程序经过交互式电子通信线提供给一般公众，(vi) 通过店面显示和小册子发布转送和出租给普通用户。

(4) 在每个流程图中顺序执行的步骤中的时间元素是本发明的实质特征，并且很清楚每个流程图所示的程序公开了再现的方法。通过依次执行每个步骤中的操作来执行由流程图表示的程序，从而获得

本发明的目的和实现效果，是实施根据本发明的记录方法。

(5) 在 BD-ROM 中记录时，希望给构成 AVClip 的每个信息包加上扩展报头。扩展报头是被称为 TP_EXTRA_HEADER 的 4 比特数据，其包括 arrival_time_stamp 和 copy_permission_indicator。具有 TP_EXTRA_HEADER 的 TS 信息包(以下称为具有 EX 的 TS 信息包)，其每 32 个信息包被分为一组，并且写到 3 个扇区。包括 32 个具有 EX 的 TS 信息包的组具有 6144 比特 (=32×192)，这是与 3 个扇区的 6144 比特 (=2048×3) 的尺寸相同的尺寸。被储存在 3 个扇区中的 32 个具有 EX 的 TS 信息包的组被称为对准单元。

当在通过 IEEE1394 连接的家庭网络中使用再现装置时，该再现装置通过以下传输程序传输对准单元。发送器从包含在对准单元中的 32 个具有 EX 的 TS 信息包的每个信息包获得 TP_EXTRA_HEADER，并在 DTCP 标准的基础上在解码之后输出 TS 信息包的主体。当输出 TS 信息包时，将等时信息包插在任何两个相继的 TS 信息包之间。插入点是以由 TP_EXTRA_HEADER 中的 arrival_time_stamp 表示的时间为基础的位置。伴随着 TS 信息包的输出，再现装置输出 DTCP_Descriptor。DTCP_DESCRIPTOR 表示用于复制允许的设置。通过描述 DTCP_DESCRIPTOR 以便表示禁止复制，当在通过 IEEE1394 标准连接的家庭网络中使用，不通过其他器件记录 TS 信息包。

(6) 上述实施例中的数字流是 AVClip。然而，数字流可以是 DVD-视频标准或 DVD-视频记录标准中的视频目标 (VOB)。VOB 是通过多路复用视频流和音频流获得的基于 ISO/IEC13818-L-标准的程序流。此外，AVClip 中的视频流也可以是以 MPEG4 或 WMV 标准为基础的。而且，音频流可以以线性-PCM、Dolby-AC3、MP3、MPEG-AAC 或 DTS 标准为基础的。

(7) 上述实施例中的电影可以通过对经过模拟广播传输的模拟图像信号进行编码而获得的，或者可以由经过数字广播传输的传送流构成的流数据。

还可以通过对记录在录像带中的模拟或数字图像信号进行编码来

获得内容。此外，这些内容还可以通过对从视频照相机中直接装载的模拟或数字图像信号进行编码来获得。而且，这些内容可以是通过配送服务器传输的数字作品。

(8) 第一和第二实施例中的图形目标是在运行长度限制编码基础上进行编码的栅格数据。采用运行长度限制编码用于压缩和编码图形目标，因为运行长度限制编码是最适合于压缩和解压缩字幕的。这些字幕具有水平方向上的长度变得相对长的特性，因而，通过使用运行长度限制编码可以获得高压缩率。此外，运行长度限制编码优选用于制成用来解码的软件，因为解压缩中的负载很低。此外，为了在字幕和图形目标之间共享解码用的装置结构，对图形目标采用与用于这些字幕的压缩/解压缩方法相同的方法。然而，使用运行长度限制编码不是本发明的主要部分，图形目标可以是 PNG 数据。而且，图形目标不需要是栅格数据，可以是矢量数据。此外，图形目标可以是透明图形。

(9) 通过 PCS 进行的显示效果的目标可以是根据再现装置的语言设置选择的用于字幕的图形。实现这种显示具有很高利用价值，因为可以通过根据再现装置的语言设置显示的字幕图形来实现在常规 DVD 中由运动图像实现的效果。

(10) 通过 PCS 获得的显示效果的目标可以是根据再现装置的显示设置而选择的用于字幕的图形。具体地说，用于各种显示模式的图形，如宽屏、全屏显示和邮政信箱被记录在 BD-ROM 中，并且再现装置根据与再现装置连接的 TV 的设置来选择任何记录设置。在这种情况下，对根据显示设置显示的字幕图形执行以 PCS 为基础的显示效果，这些字幕看上去更令人印象深刻和更专业化。实现这种显示器具有高实用性，因为可以通过根据再现装置的显示设置显示的字幕图形来实现与常规 DVD 中的运动图像本身中实现的效果相同的效果。

(11) 在第一实施例中，窗口尺寸设置为整个图形平面的 25%，以便将对图形平面的写速度 R_c 设置为在一帧中执行图形平面的清除和重画的速度。但是， R_c 可以设置成在垂直回扫周期期间完成清除和重画。假定垂直回扫周期是 1/29.93 秒的 25%， R_c 是 1Gbps。用这

种方式设置 Rc 具有很高的实用价值，因为可以更平滑地显示图形。

此外，除了在垂直回描周期期间的写操作之外，还可以与行扫描同步地进行写操作。由此，甚至可以在写速度 Rc 是 256 Mbps 时更平滑地显示图形。

(12) 在上述实施例中，将图形平面安装到再现装置上。但是，还可以将用于储存一行的解压缩像素的线缓冲器代替图形平面安装到再现装置上。转换成图像信号是通过行来进行的，因此可以单独利用线缓冲器来执行图像信号的转换。

(13) 在上述实施例中，作为图形的例子介绍了电影的文本字幕。但是，该图形可以包括例如构成商标的图案、字符和颜色的组合、国家饰章、国旗、国徽、国家政府使用的标记和用于监督或认证的大图章、国际组织的饰章、旗帜或徽章，或者特殊项目的原始标记。

(14) 在第一实施例中，假设水平地写字幕，用于呈现字幕的窗口被限定在屏幕的上侧或者屏幕的底部。但是，可以将窗口限定成在屏幕的左或右侧上显现，以便在屏幕的左侧和右侧上显示字幕。通过这种方式，可以改变文本方向并垂直显示字幕。

(15) 上述实施例中的 AVClip 构成电影。但是，AVClip 也可以用于卡拉 OK。在这种情况下，PCS 可以执行显示效果，使得字幕的颜色随着声音而改变。

参考数字

1 BD 驱动器；2 读缓冲器；3 PID 滤波器；4 TB 缓冲器；5 视频解码器；6 视频平面；7 音频解码器；8 图形平面；9 CULT 单元；10 加法器；12 图形解码器；13 编码数据缓冲器；14 流图形处理器；16 组合缓冲器；17 图形控制器；200 再现装置；300 TV；400 遥控器。

工业实用性

根据本发明的记录介质和再现装置能以显示效果显示字幕。因而，可以给市场上供应的电影增加更高的价值，并激励电影市场和消费产

品。因此，根据本发明的记录介质和再现装置在如电影工业和消费产品工业等工业领域具有很高工业实用性。

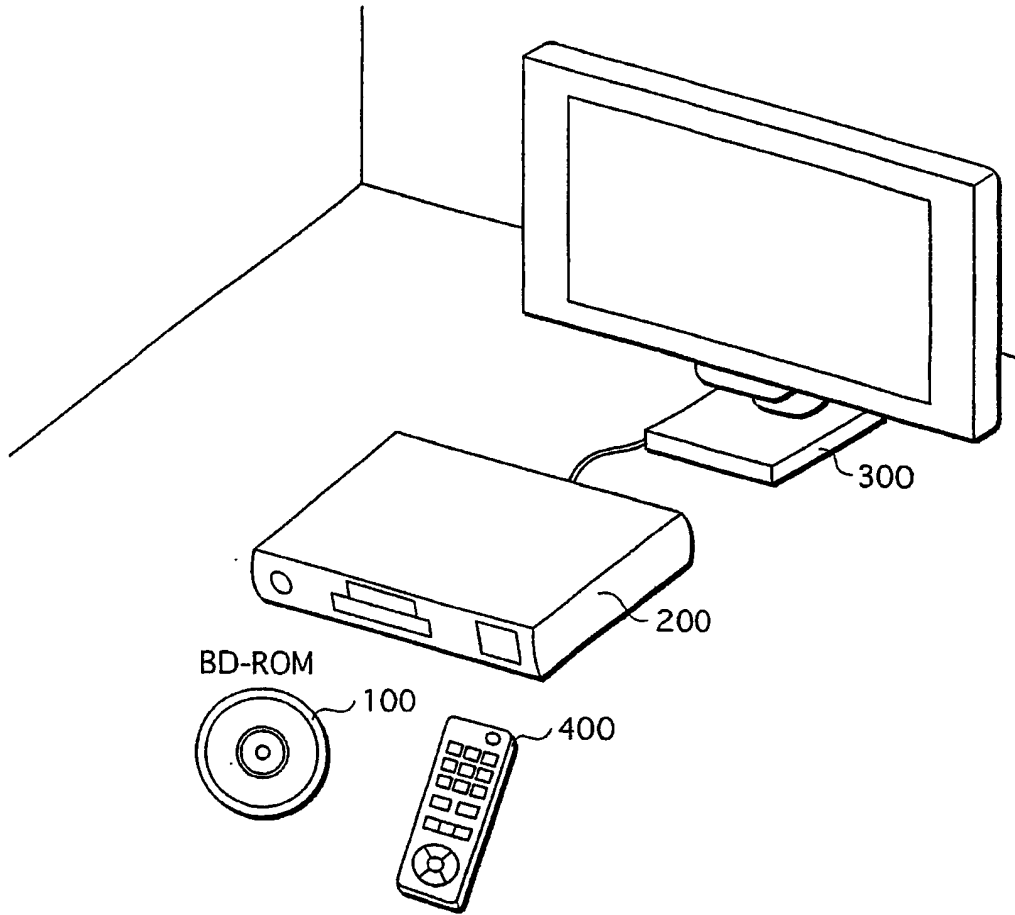
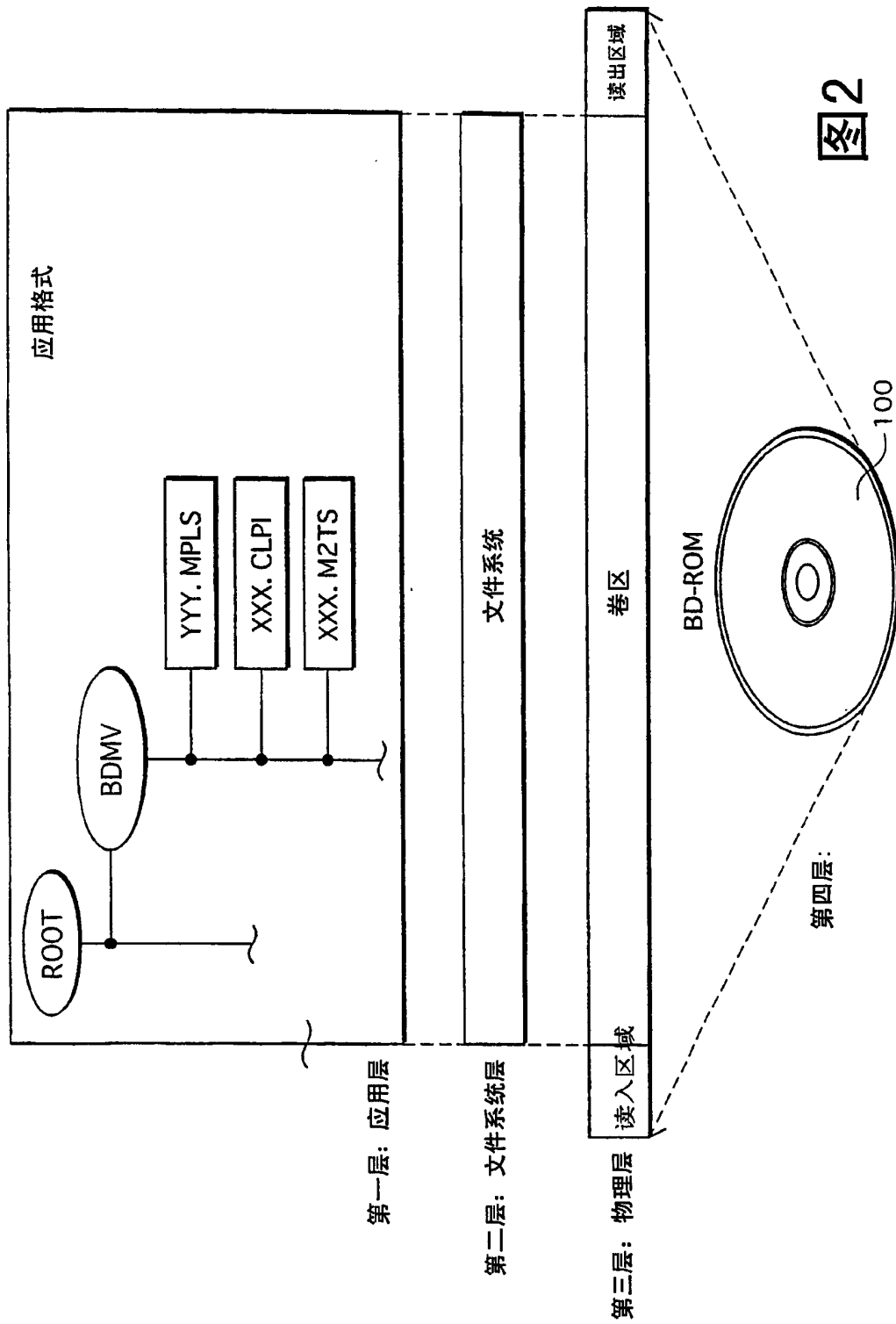


图1



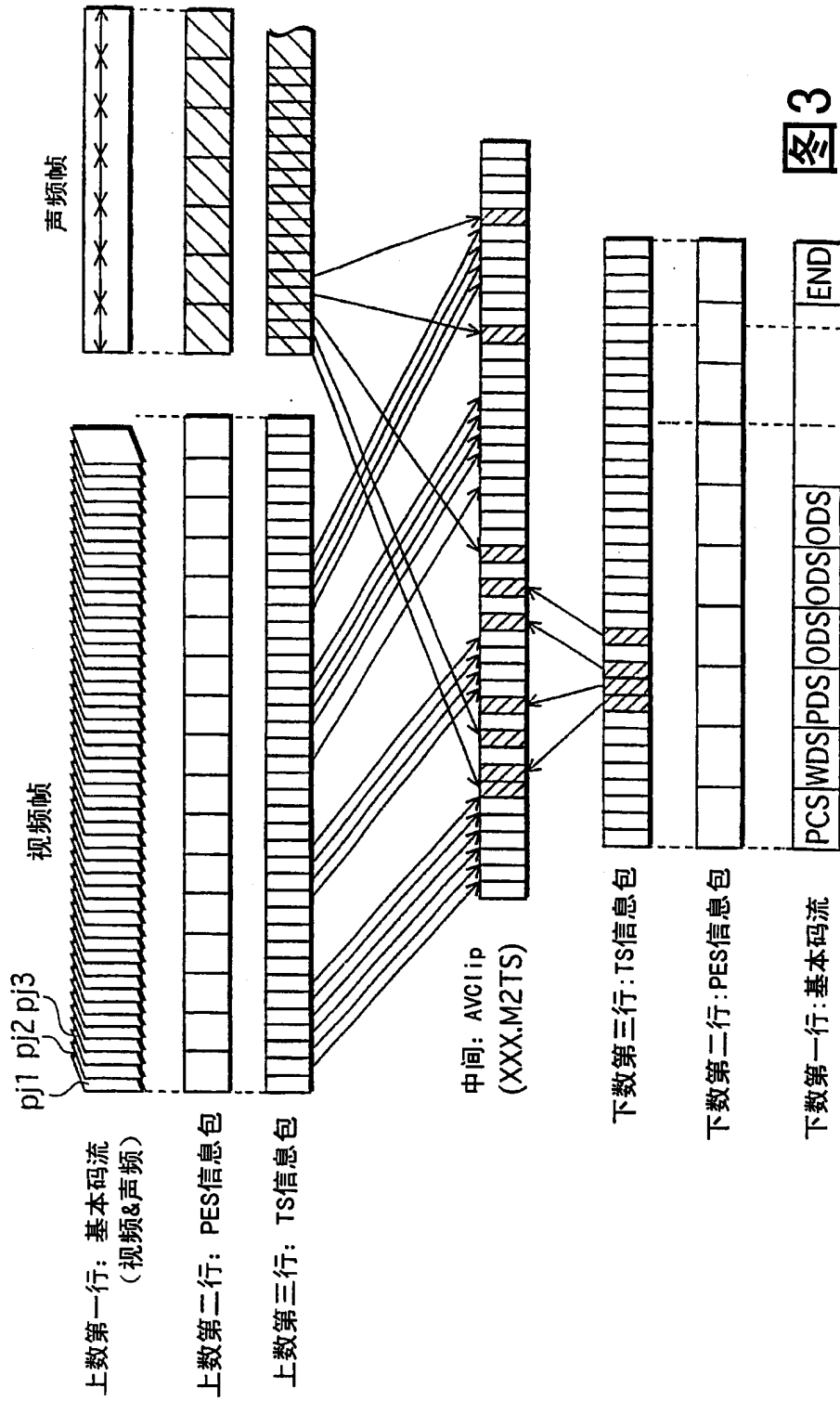


图 3

展示图形流

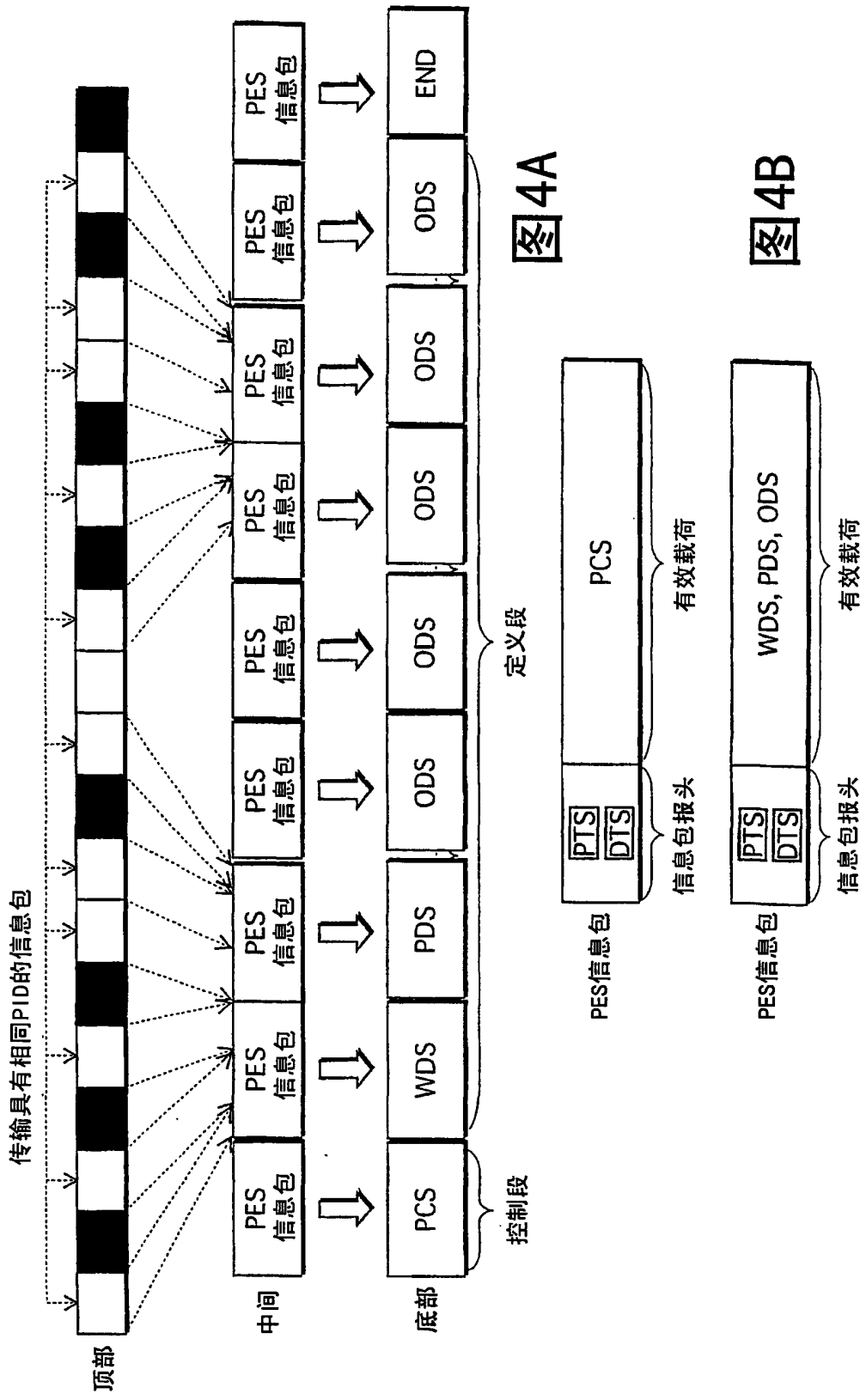


图4A

图4B

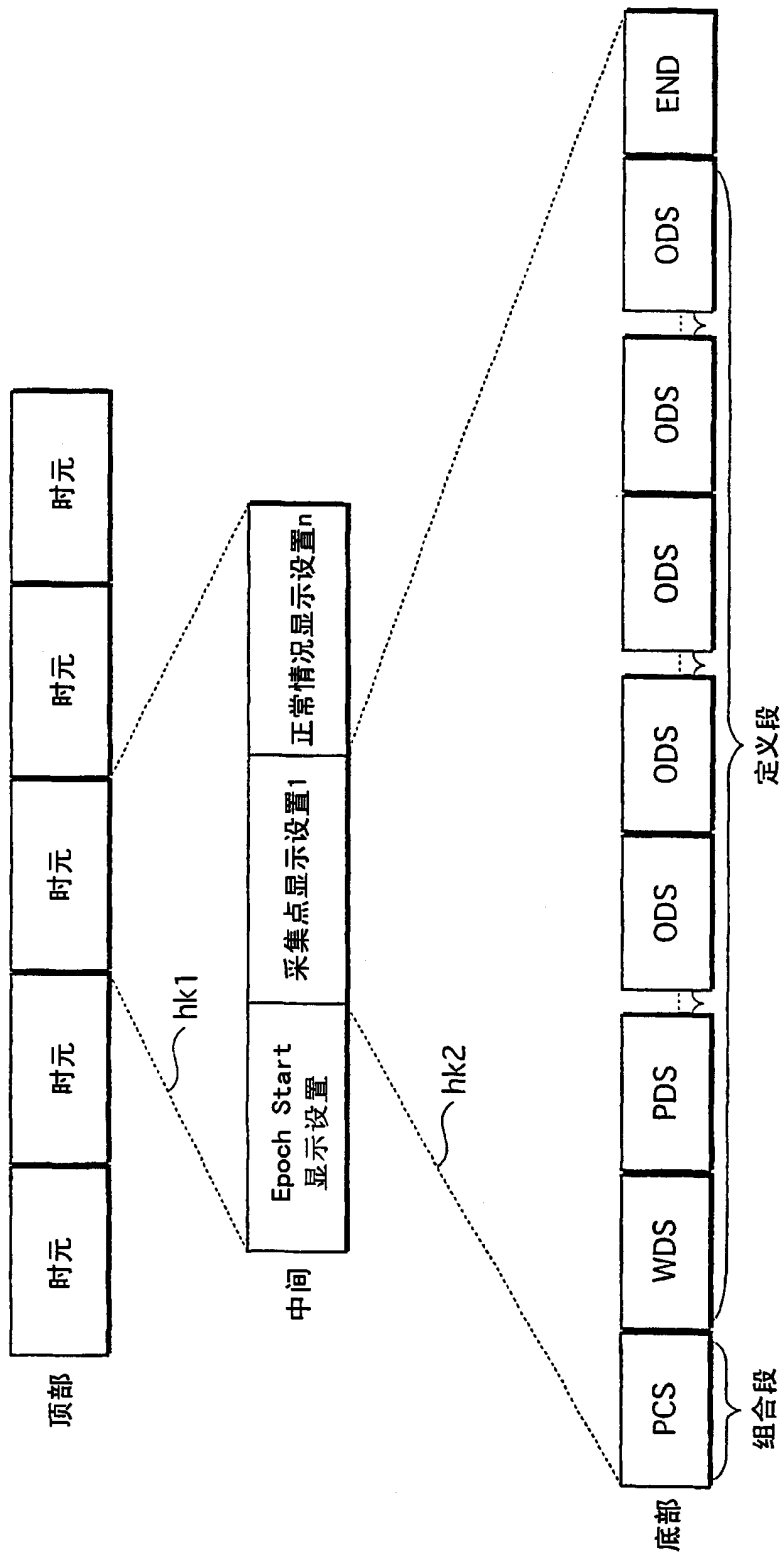


图5

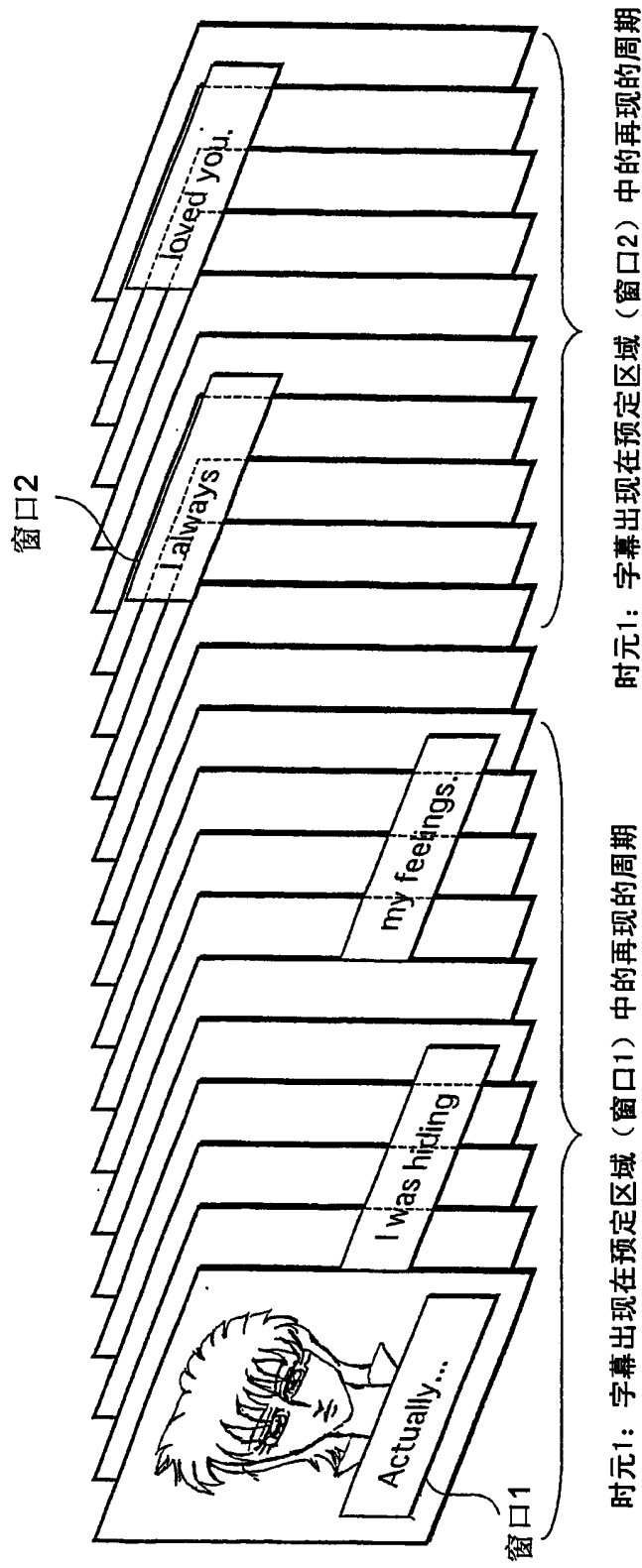


图6

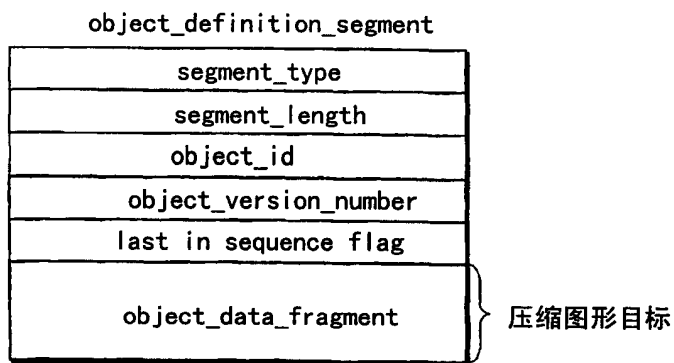


图7A

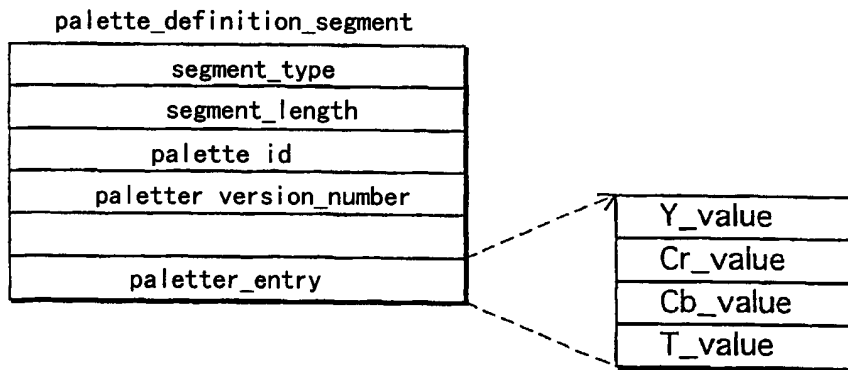


图7B

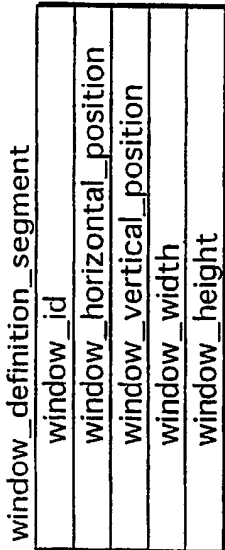


图 8A

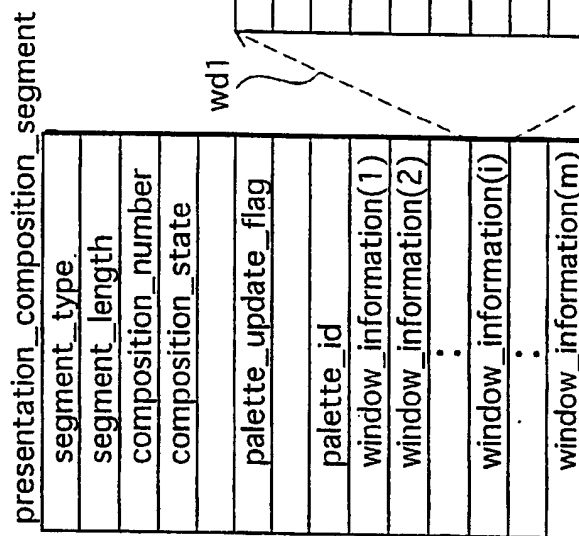
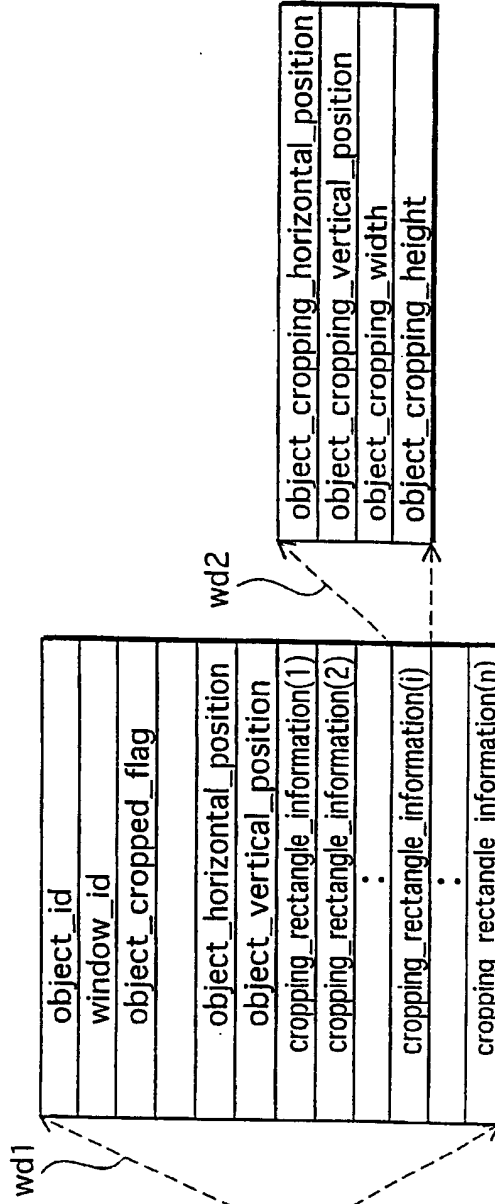


图 8B



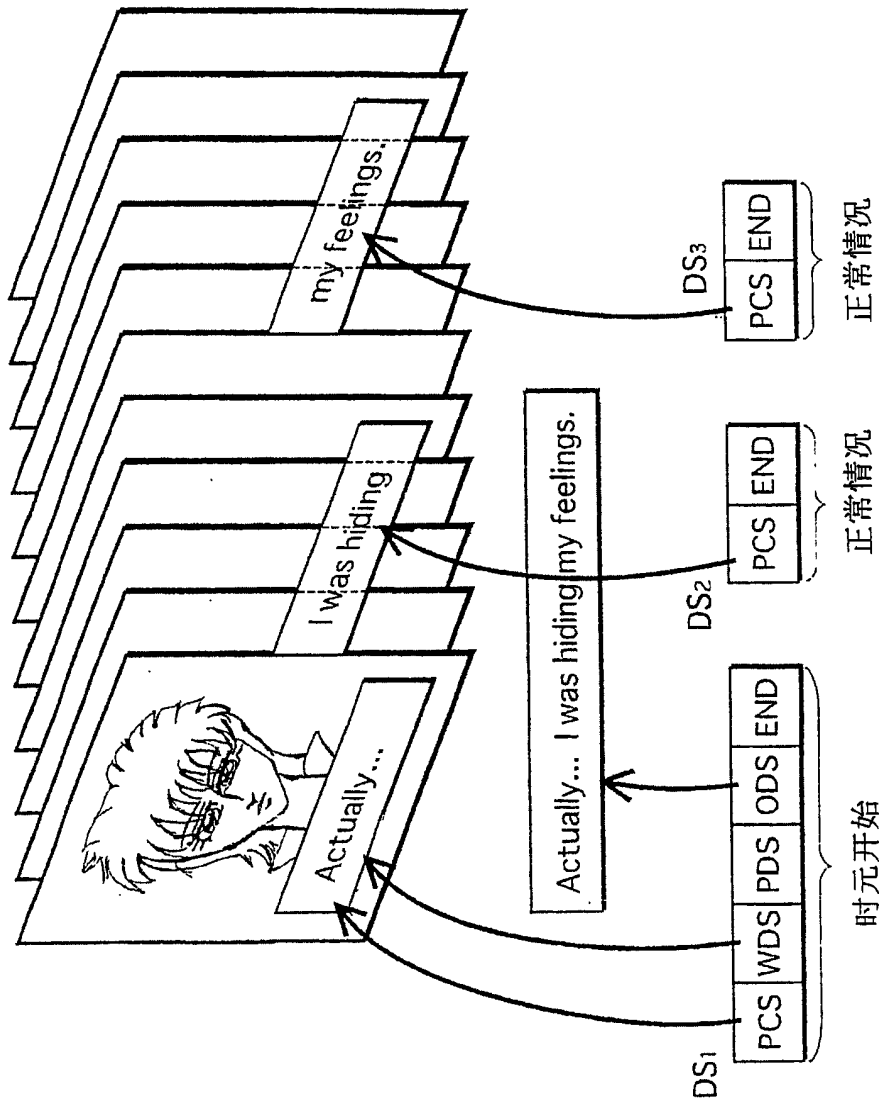


图9

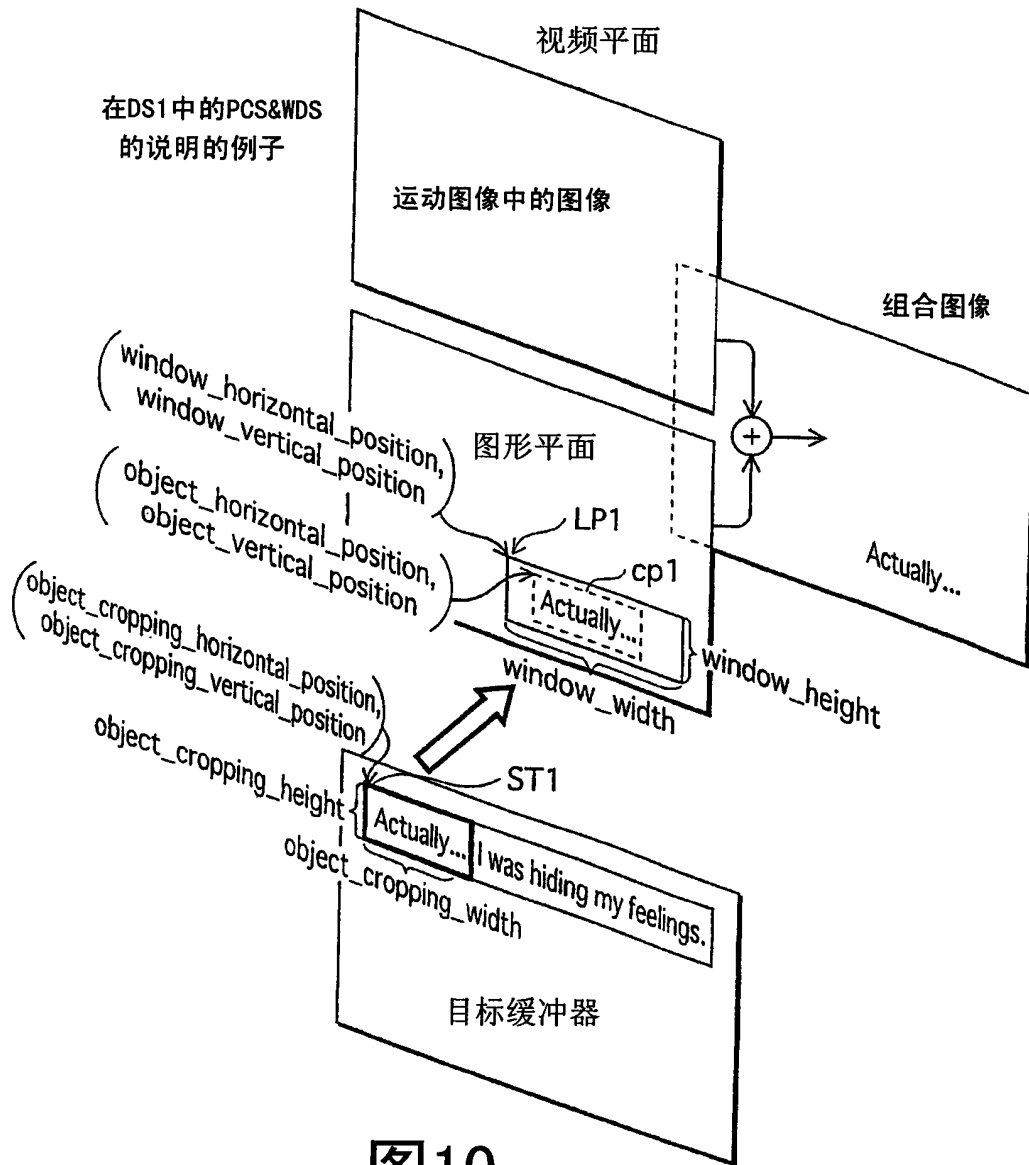


图10

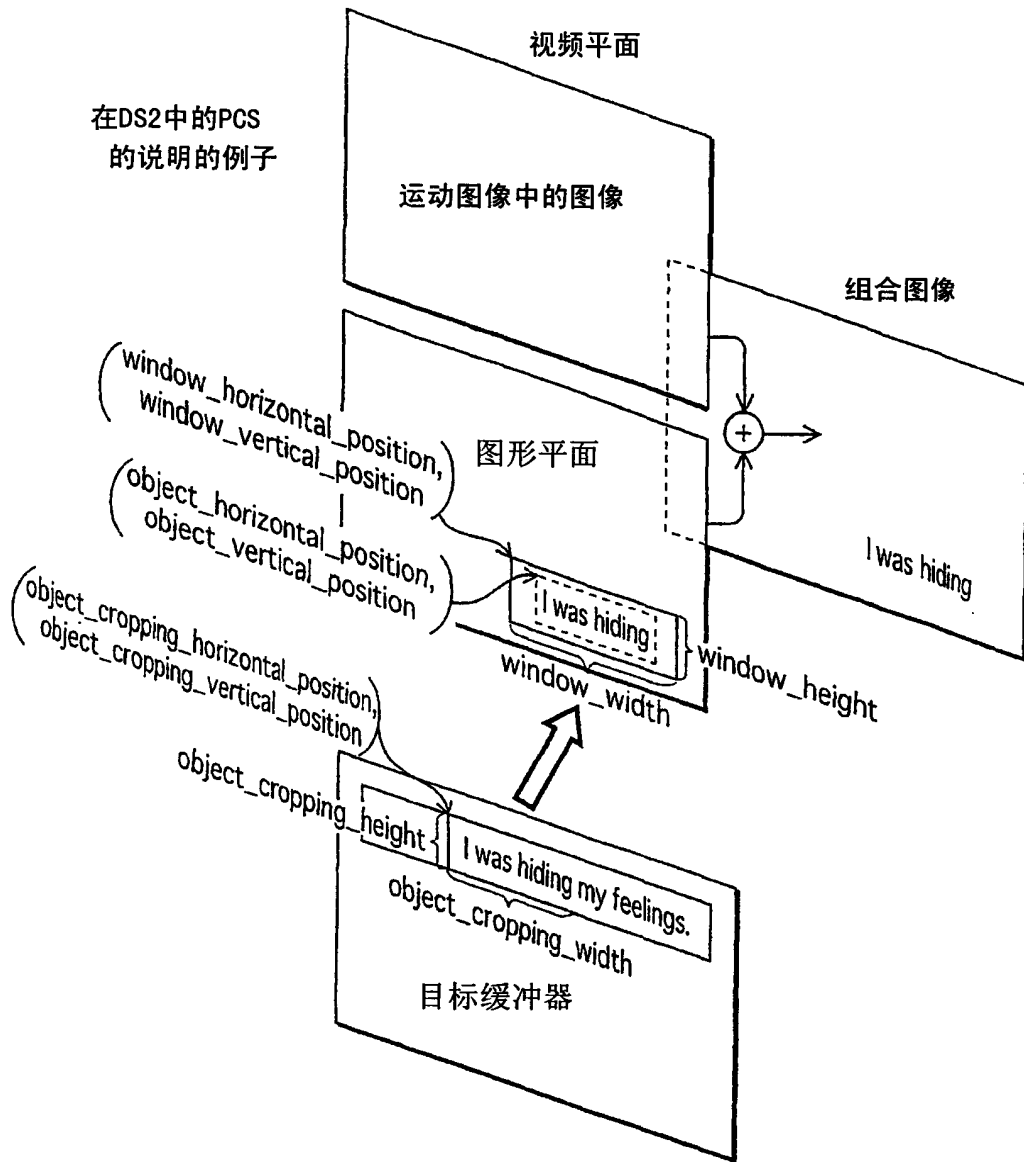


图11

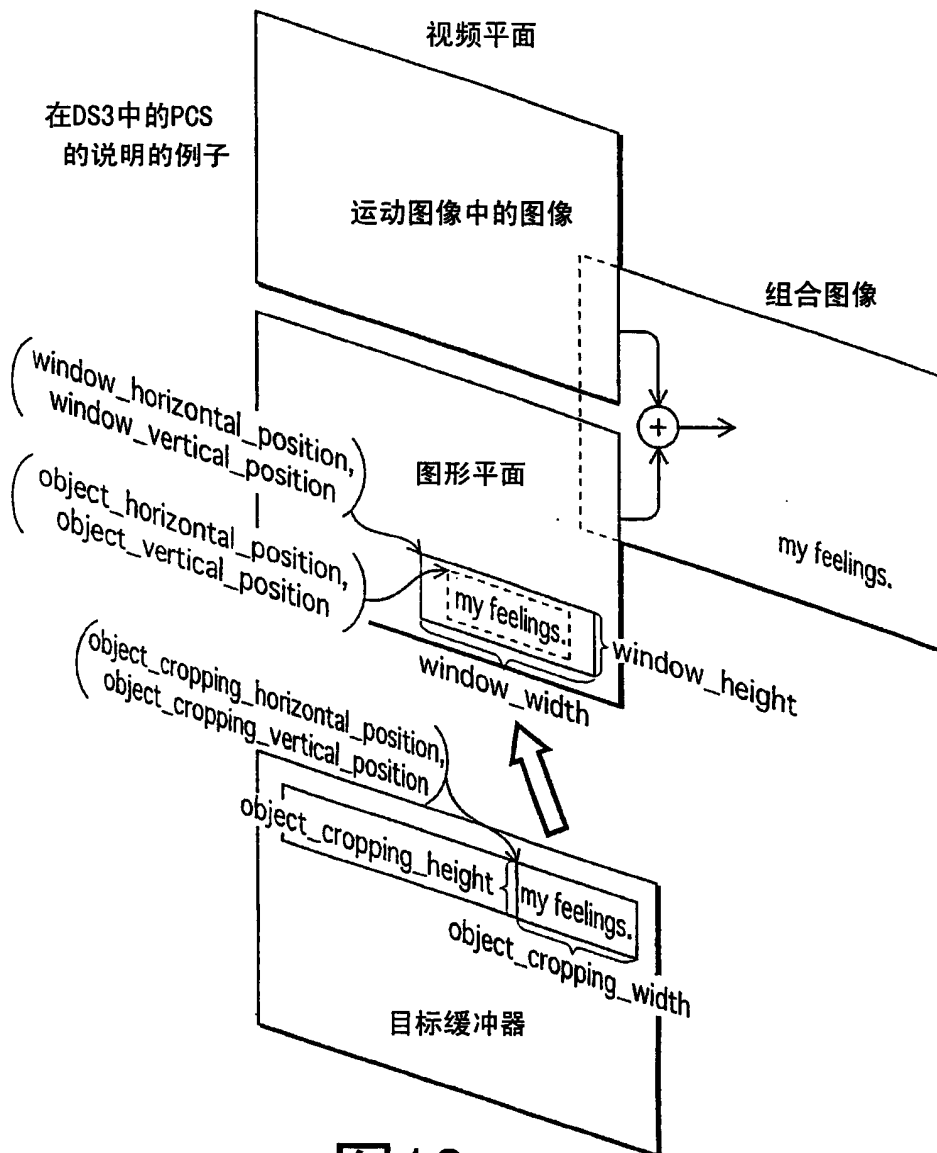


图12

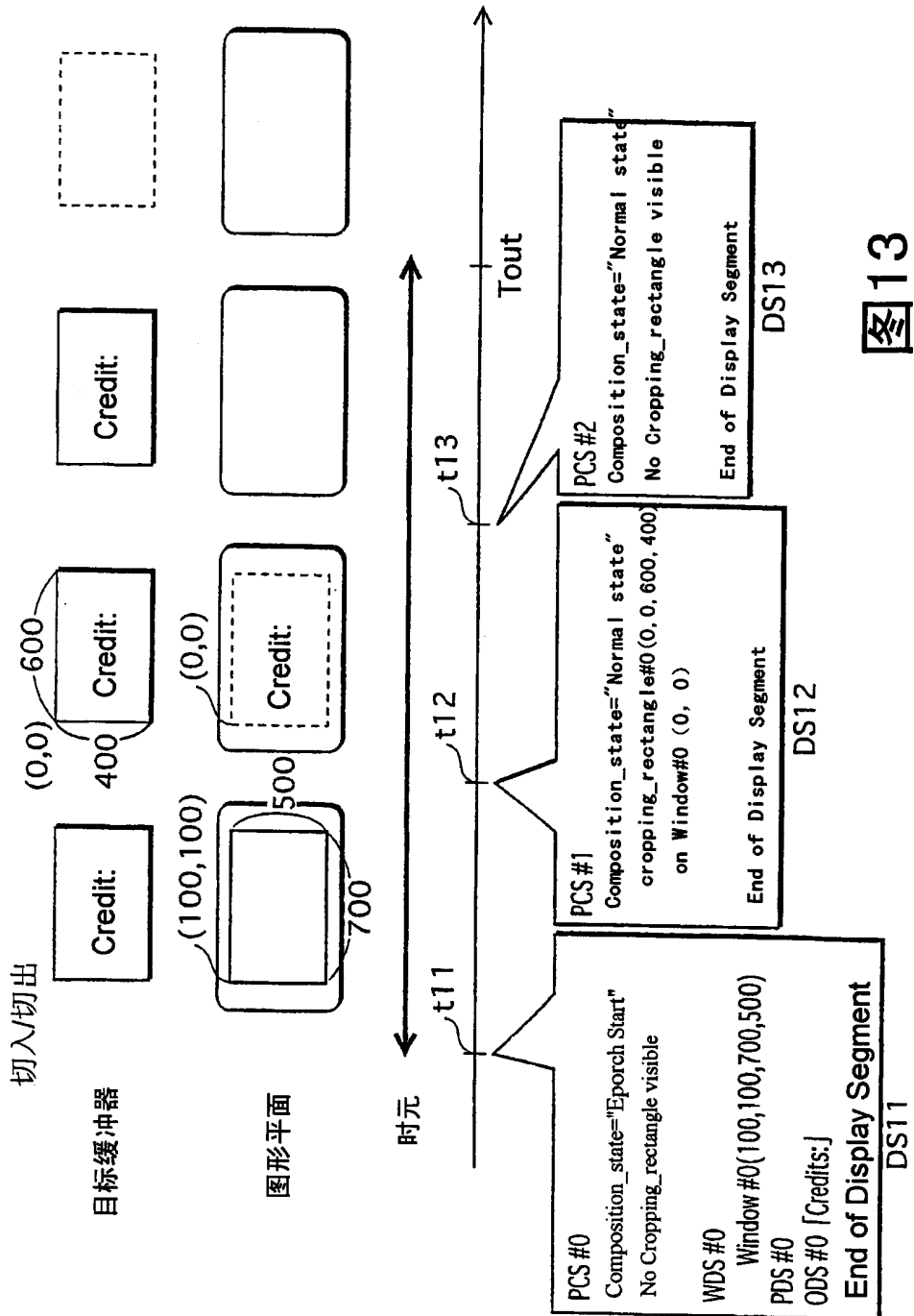


图13

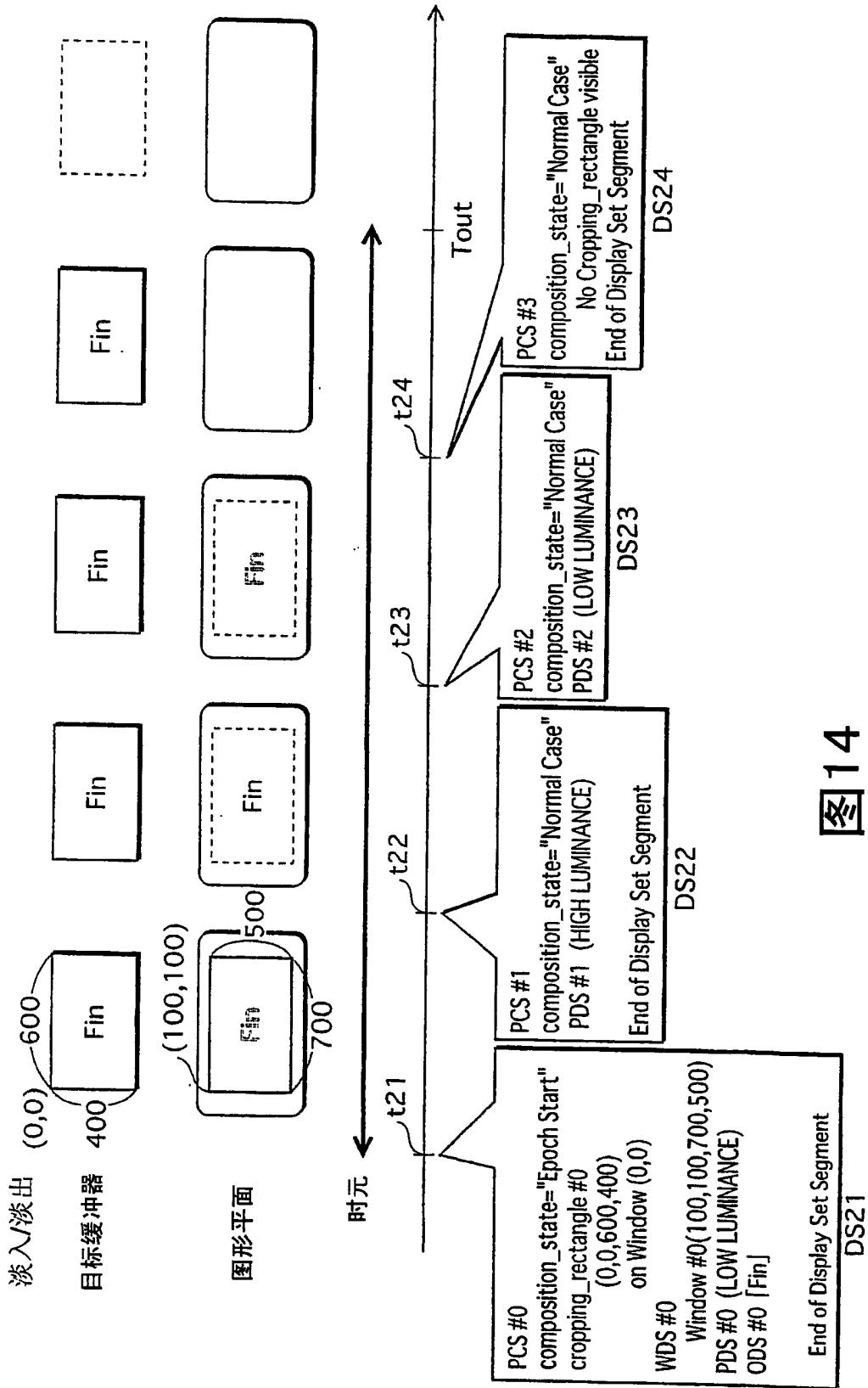


图14

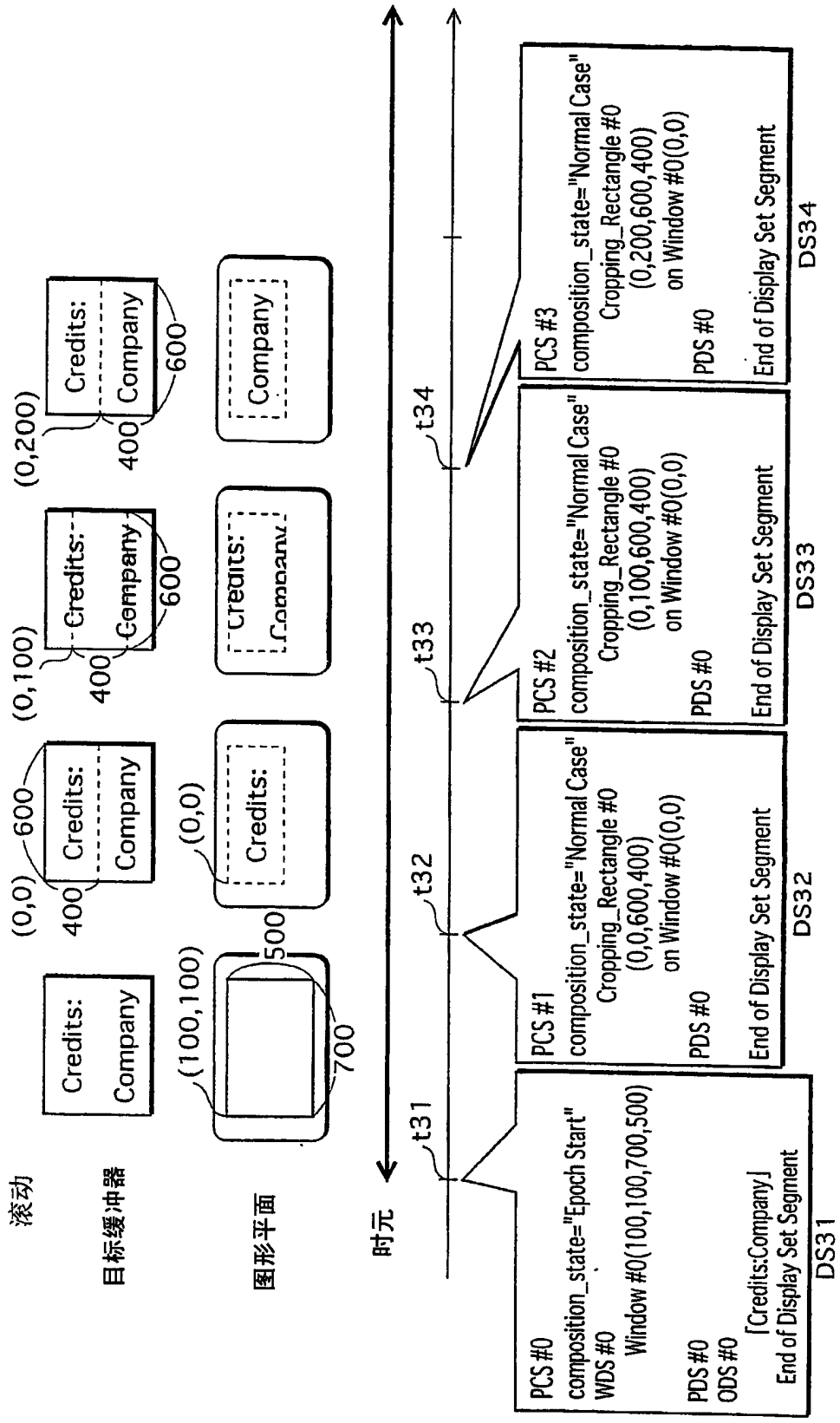


图15

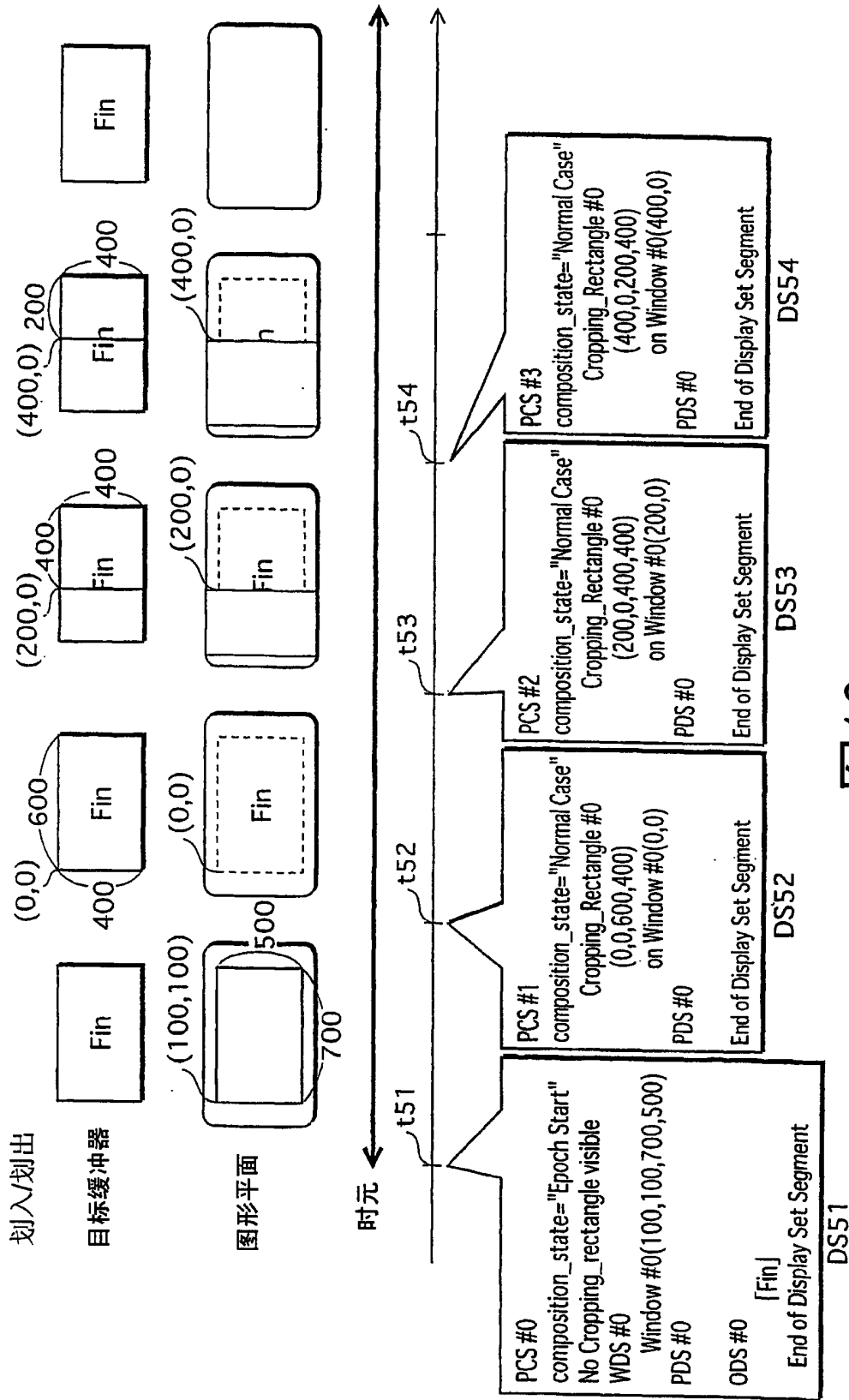
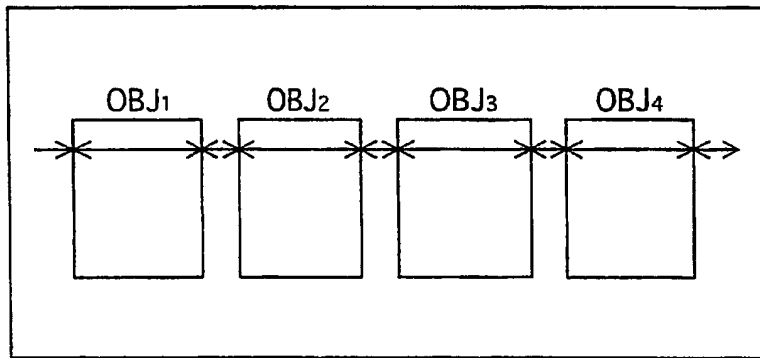
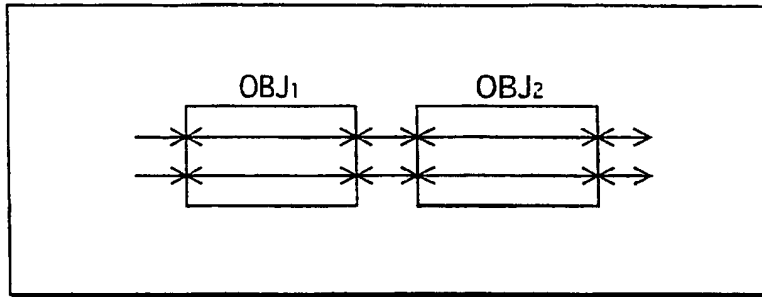


图16

目标缓冲器



X: 边缘

图17

```

    PTS( DSn[PCS] ) >= DTS( DSn[PCS] ) + DECODEDURATION( DSn )
其中：
    • DECODEDURATION( DSn ) 计算如下：
    decode_duration = 0 ;
    decode_duration += PLANEINITIALIZATIONTIME( DSn ) ;
    if( DSn.PCS.num_of_objects == 2 )
    {
        decode_duration += WAIT( DSn, DSn.PCS.OBJ[0], decode_duration ) ;
        if( DSn.PCS.OBJ[0].window_id == DSn.PCS.OBJ[1].window_id )
        {
            decode_duration += WAIT( DSn, DSn.PCS.OBJ[1], decode_duration ) ;
            decode_duration += 90000*( SIZE( DSn.PCS.OBJ[0].window_id )//256*106 ) ;
        }
        else
        {
            decode_duration += 90000*( SIZE( DSn.PCS.OBJ[0].window_id )//256*106 ) ;
            decode_duration += WAIT( DSn, DSn.PCS.OBJ[1], decode_duration ) ;
            decode_duration += 90000*( SIZE( DSn.PCS.OBJ[1].window_id )//256*106 ) ;
        }
    }
    else if( DSn.PCS.num_of_objects == 1 )
    {
        decode_duration += WAIT( DSn, DSn.PCS.OBJ[0], decode_duration ) ;
        decode_duration += 90000*( SIZE( DSn.PCS.OBJ[0].window_id )//256*106 ) ;
    }
    return decode_duration ;

    • PLANEINITIALIZATIONTIME( DSn ) 计算如下：
    initialize_duration = 0 ;
    if( DSn.PCS.composition_state == EPOCH_START )
    {
        initialize_duration = 90000*( 8*video_width*video_height//256*106 ) ;
    }
    else
    {
        for( i=0 ; i < WDS.num_windows ; i++ )
        {
            if( EMPTY( DSn.WDS.WIN[i], DSn ) )
                initialize_duration += 90000*( SIZE( DSn.WDS.WIN[i] )//256*106 ) ;
        }
    }
    return initialize_duration ;

    • WAIT( DSn, OBJ, current_duration ) 计算如下：
    wait_duration = 0 ;
    if( EXISTS( OBJ.object_id, DSn ) )
    {
        object_definition_ready_time = PTS( GET( OBJ.object_id, DSn ) ) ;
        current_time = DTS( DSn.PCS ) + current_duration ;
        if( current_time < object_definition_ready_time )
            wait_duration += object_definition_ready_time - current_time ;
    }
    return wait_duration ;

```

图 18

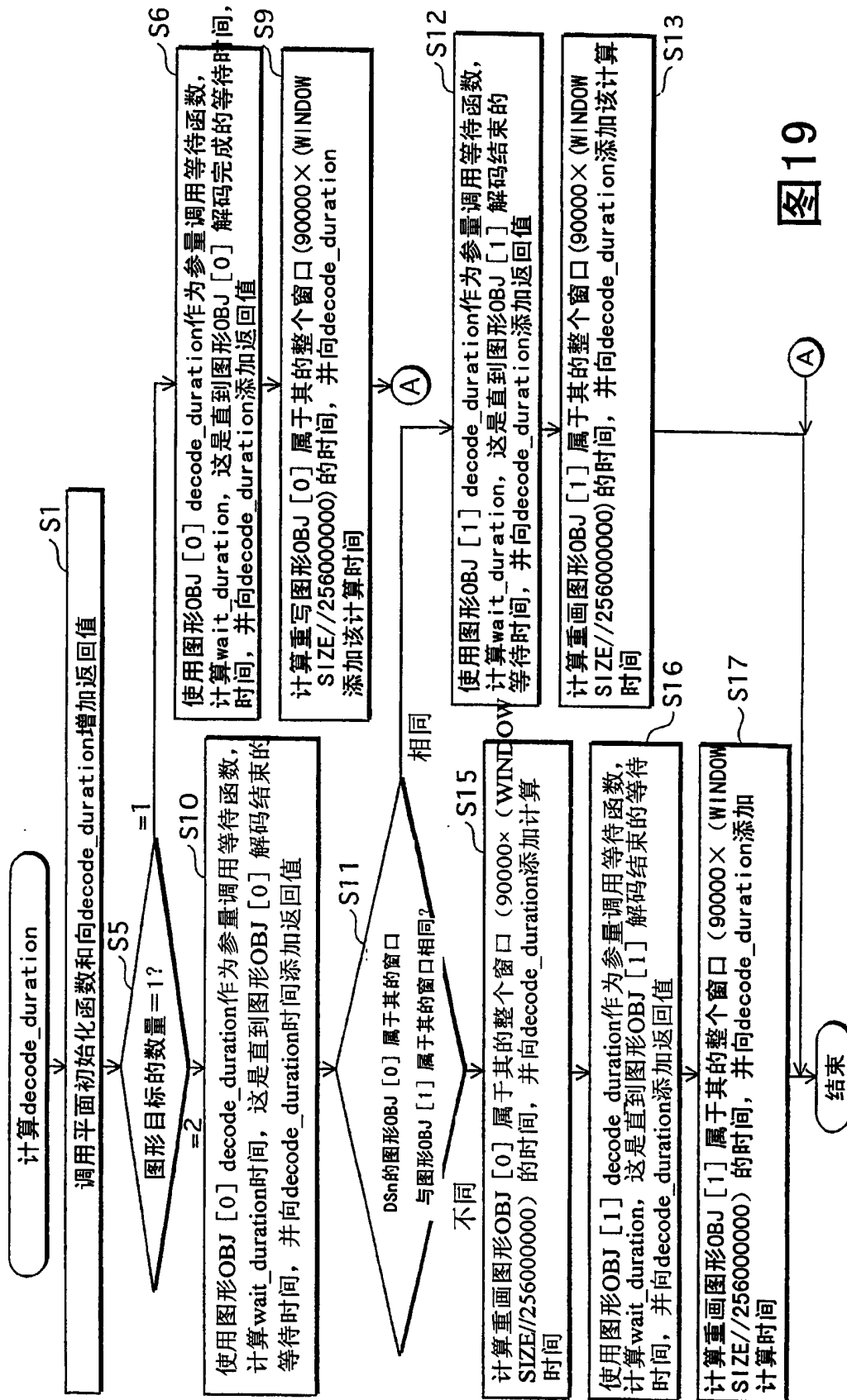


图19

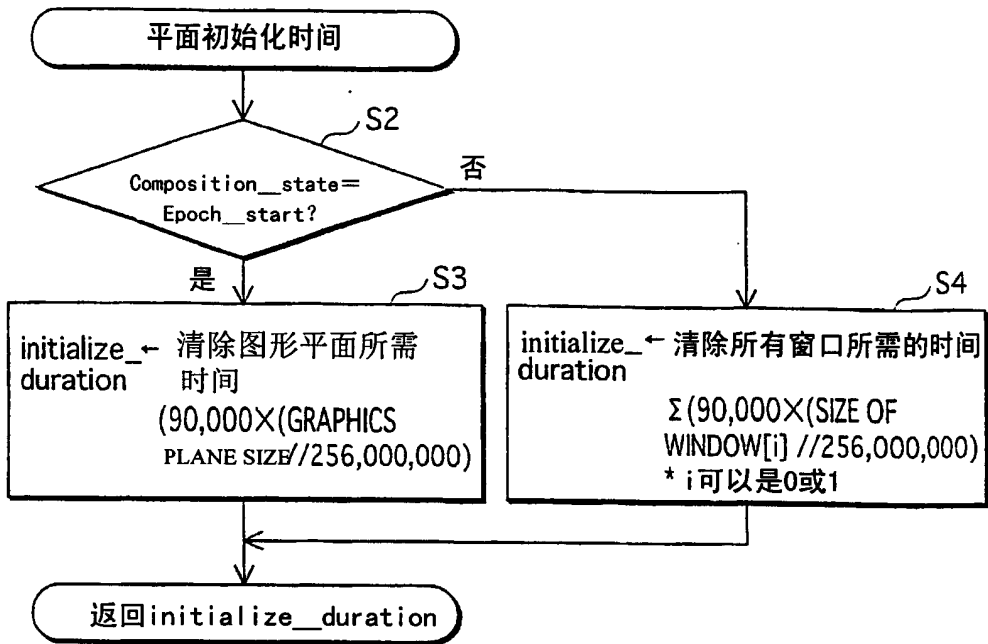


图20A

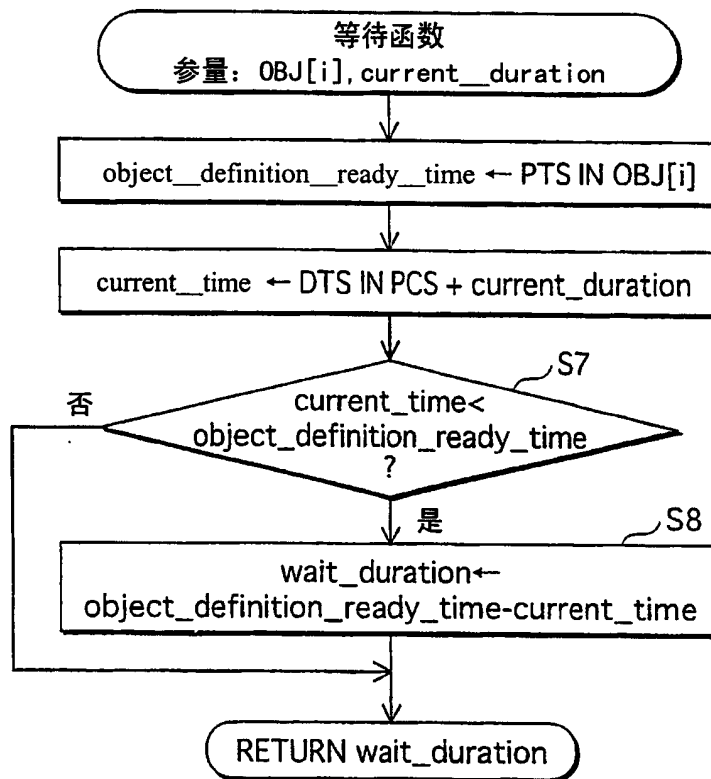
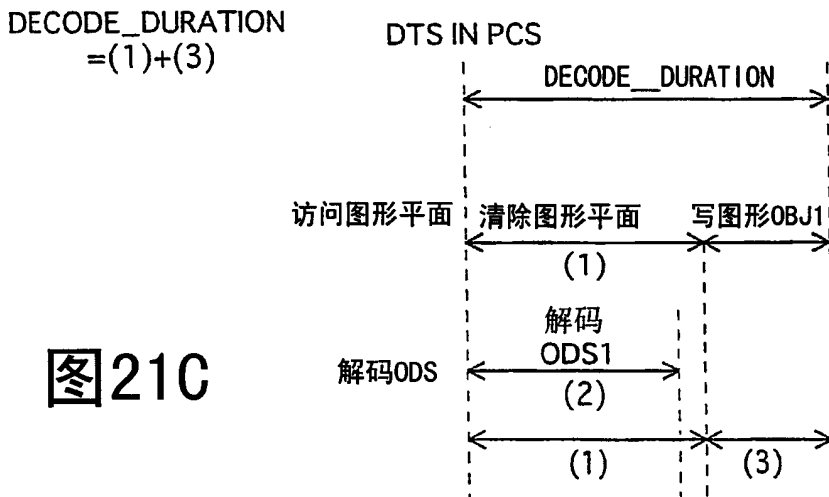
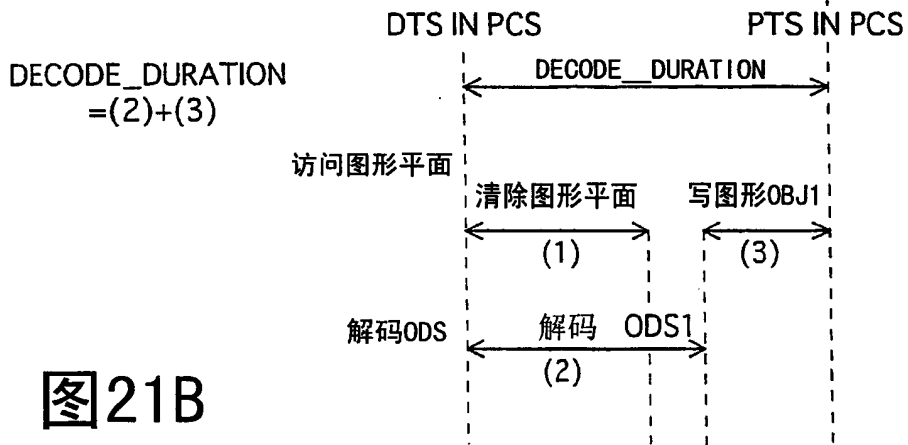
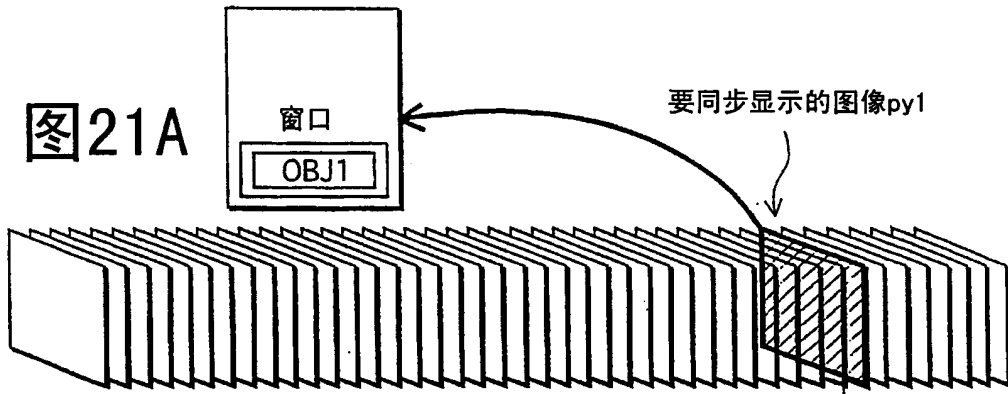


图20B



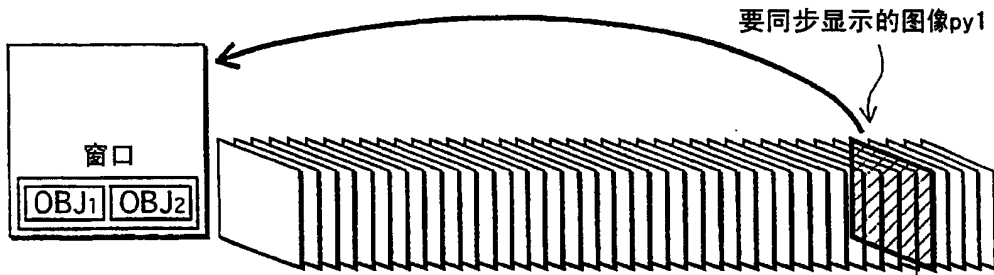


图22A

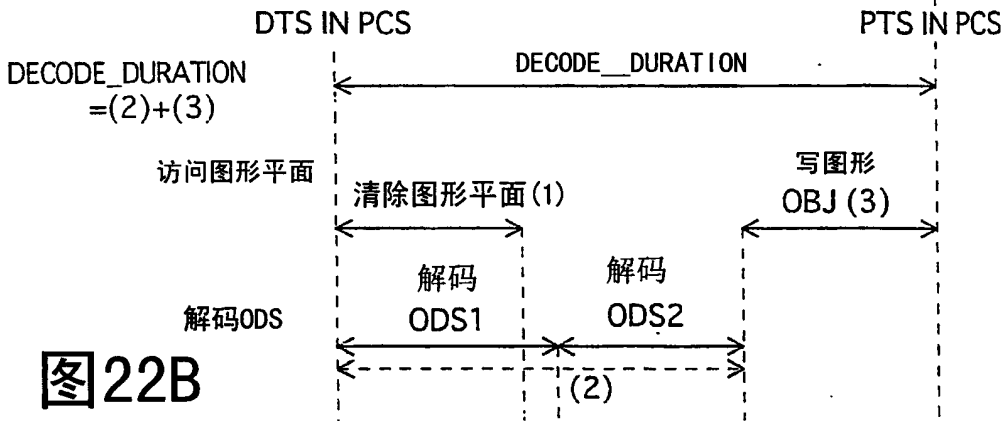


图22B

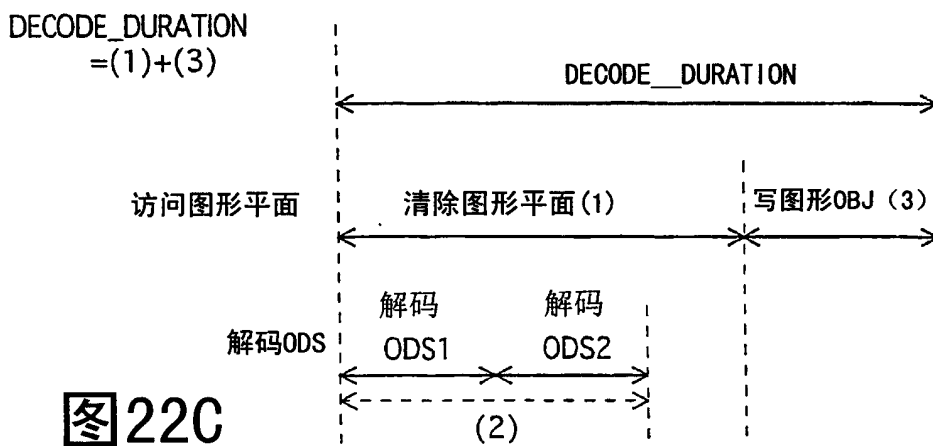


图22C

图 23A

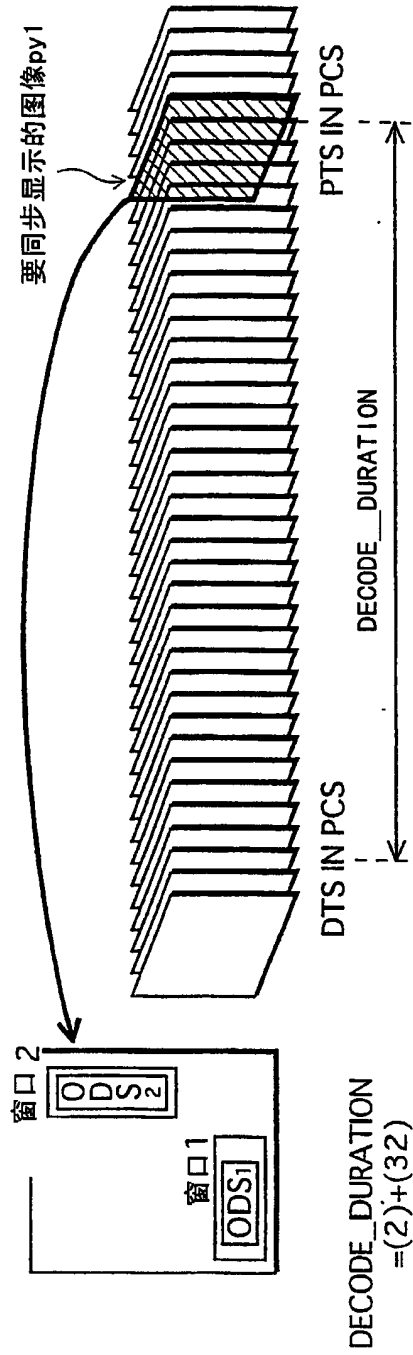


图 23B

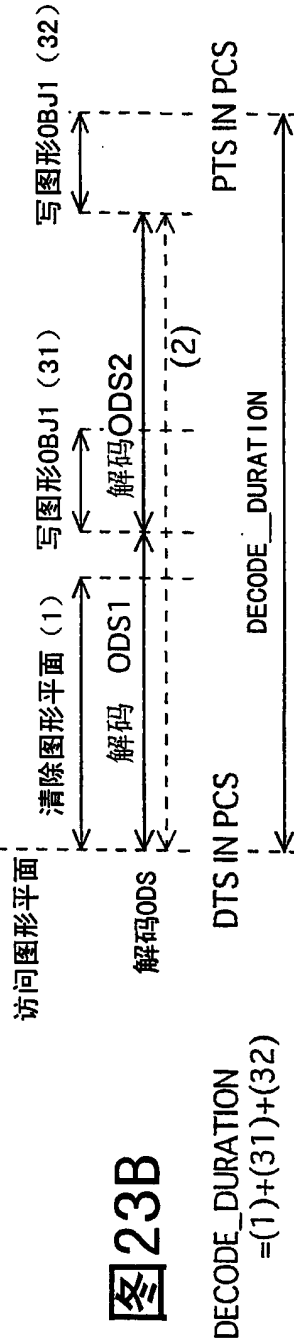
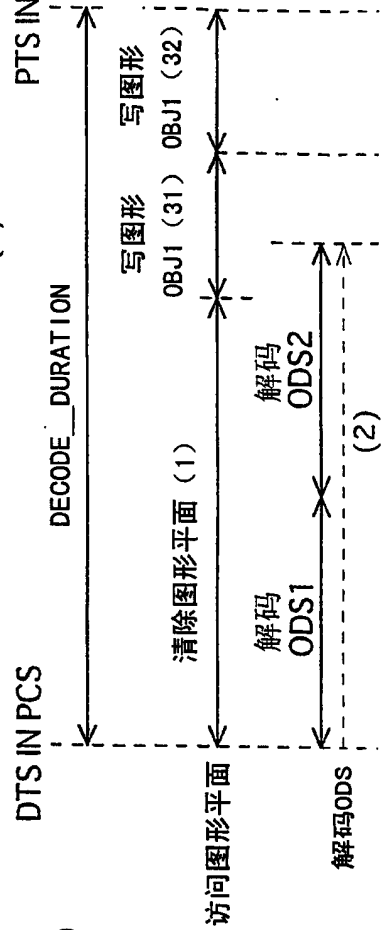


图 23C



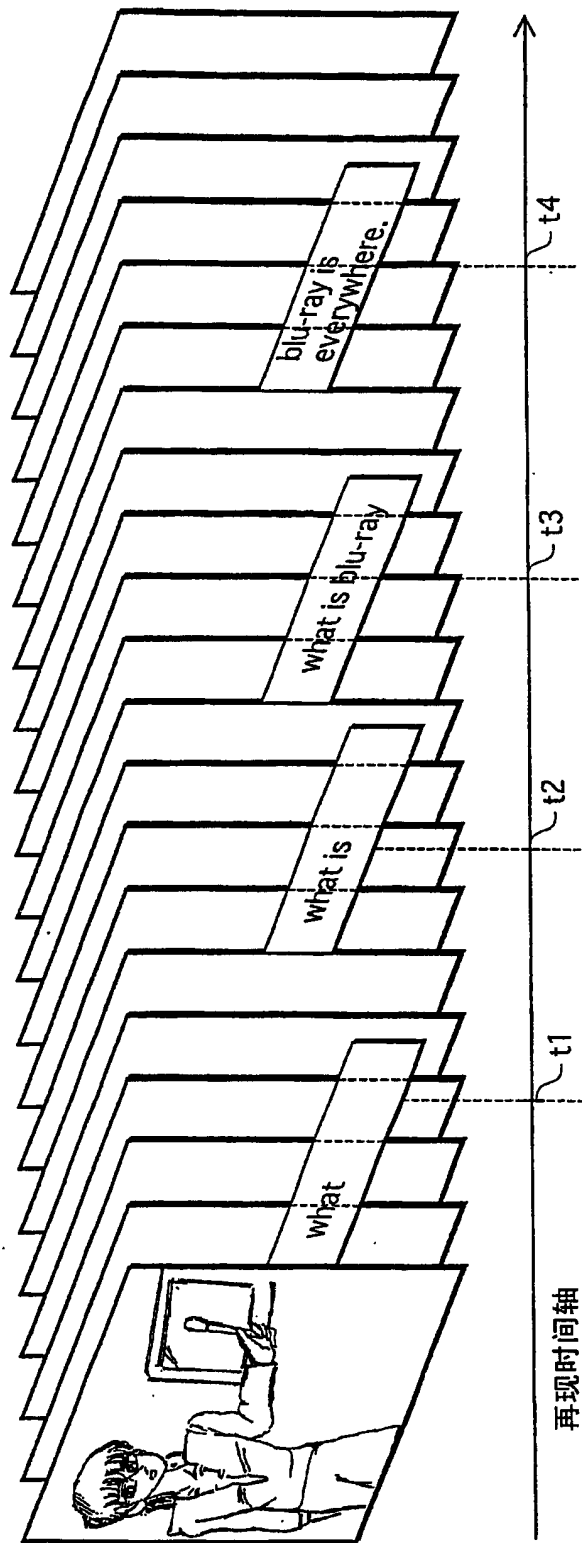
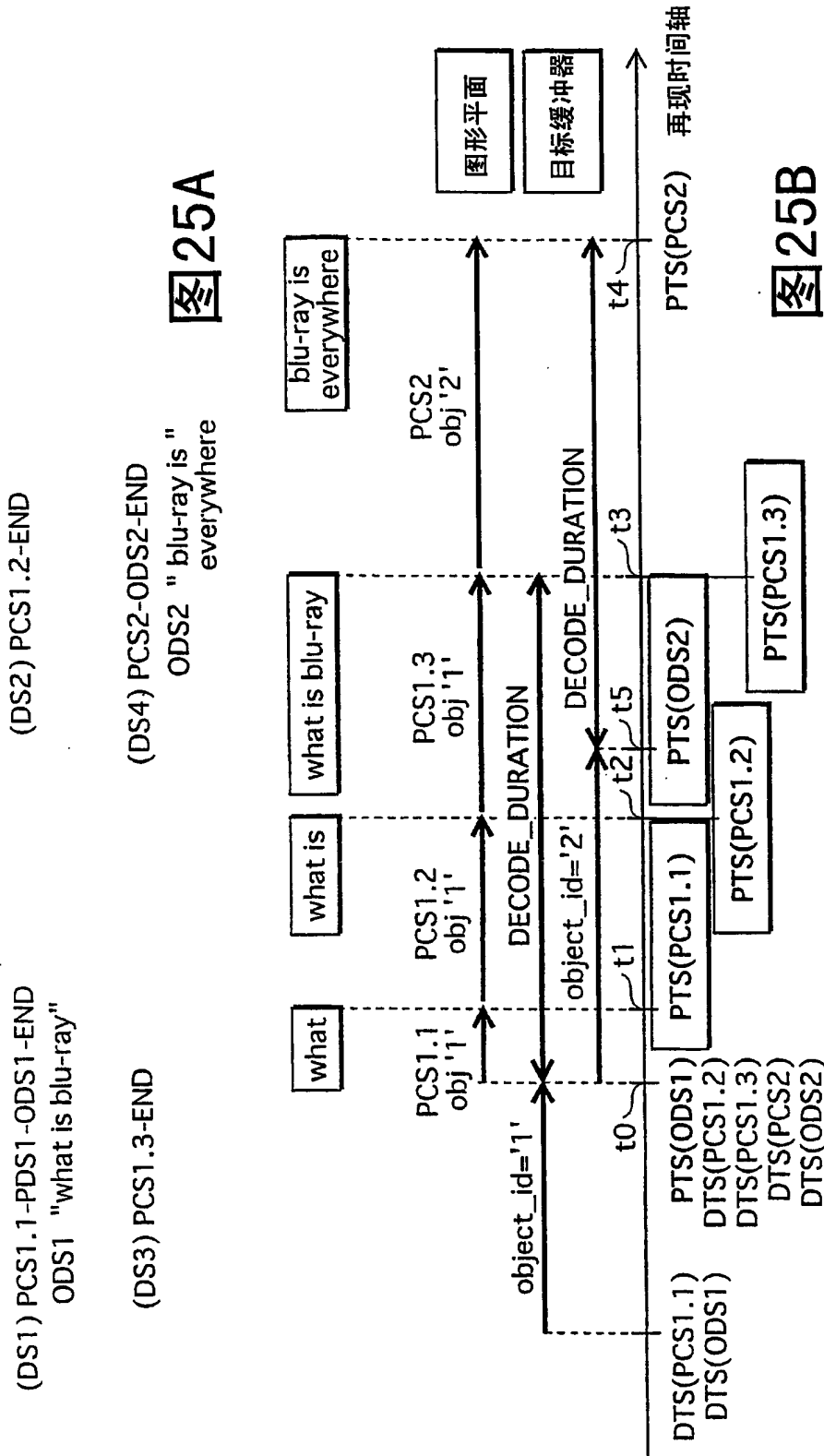


图24



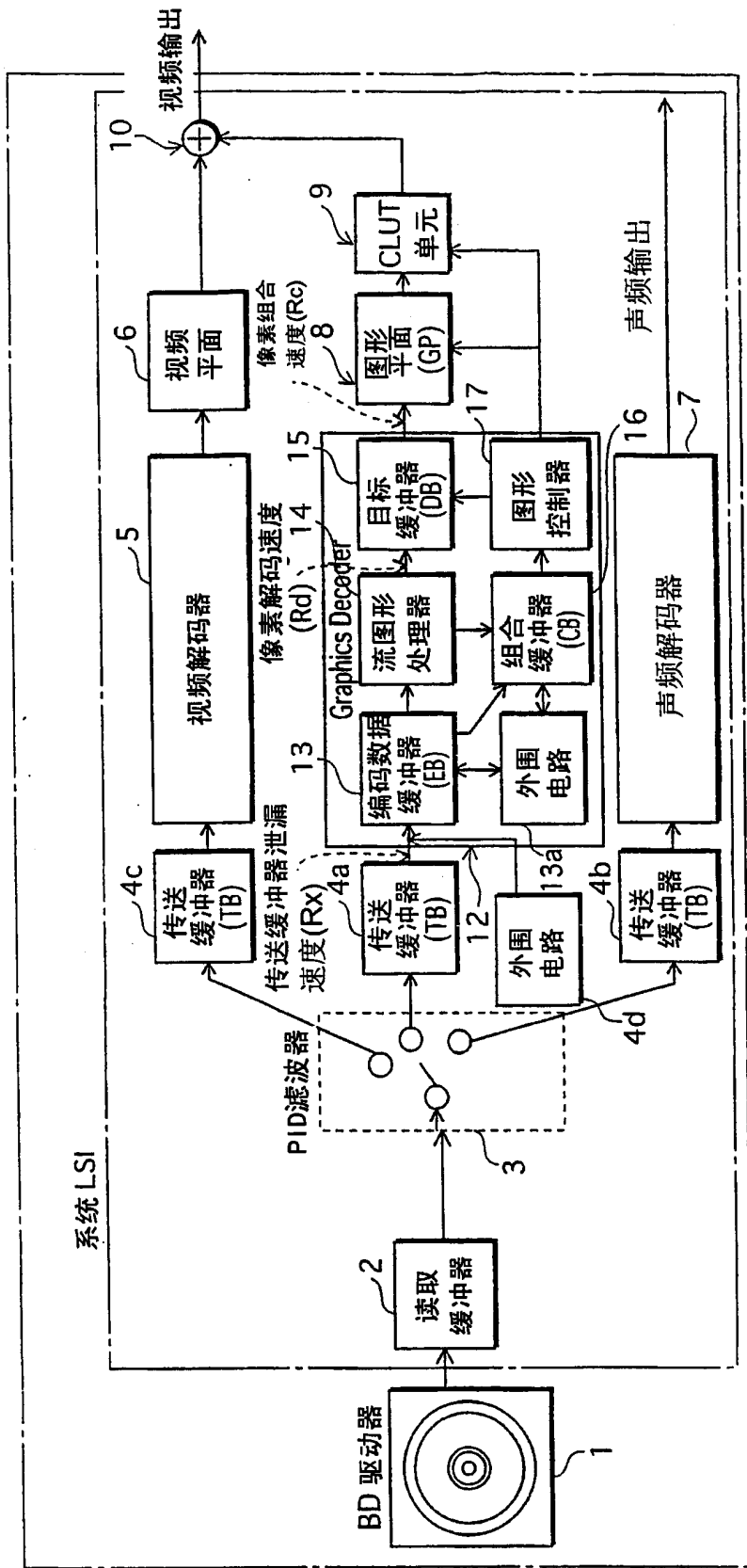


图 26

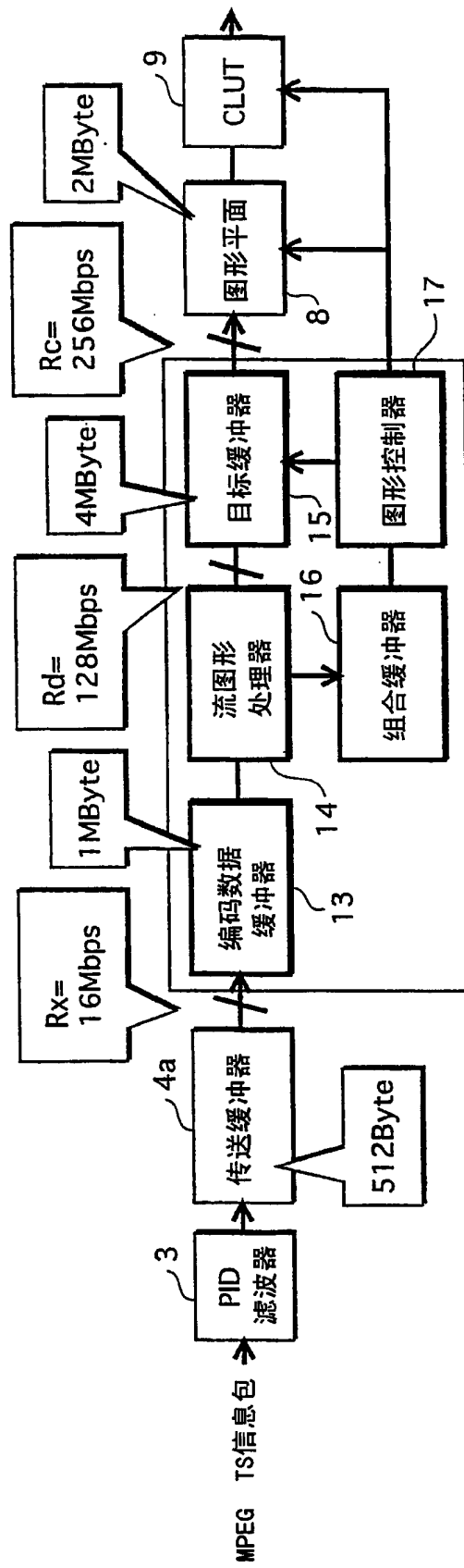


图27

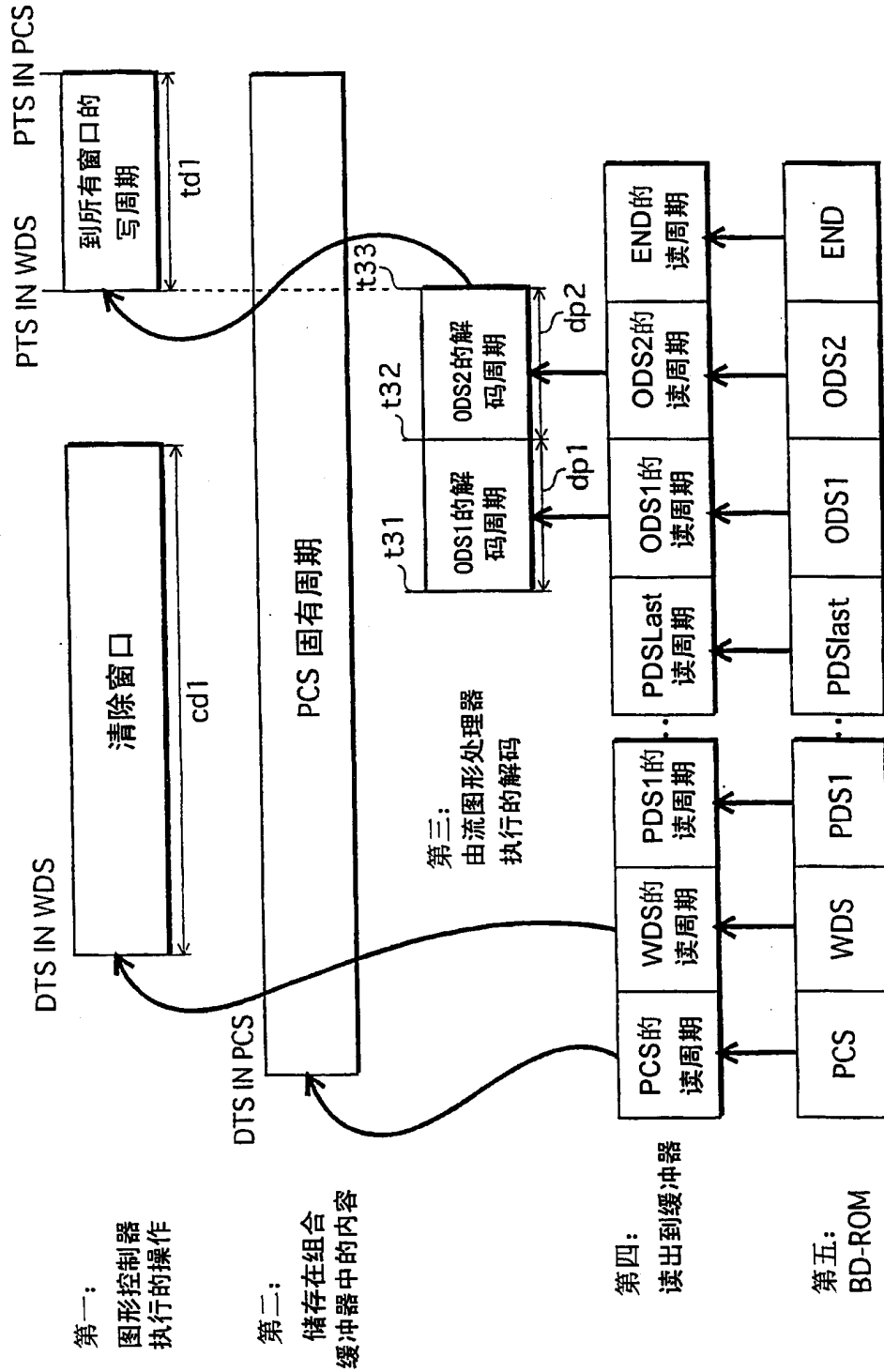


图28

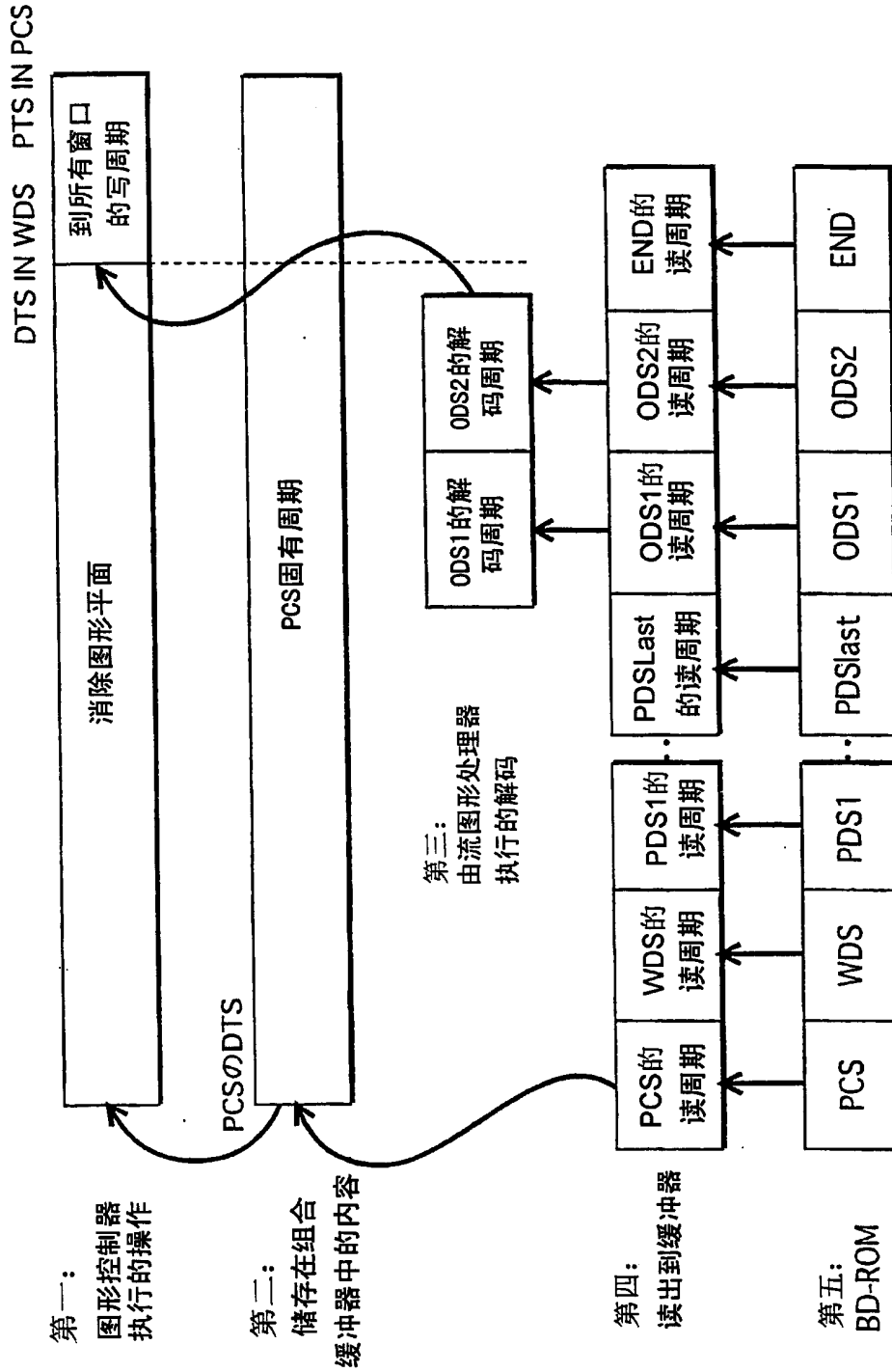


图29

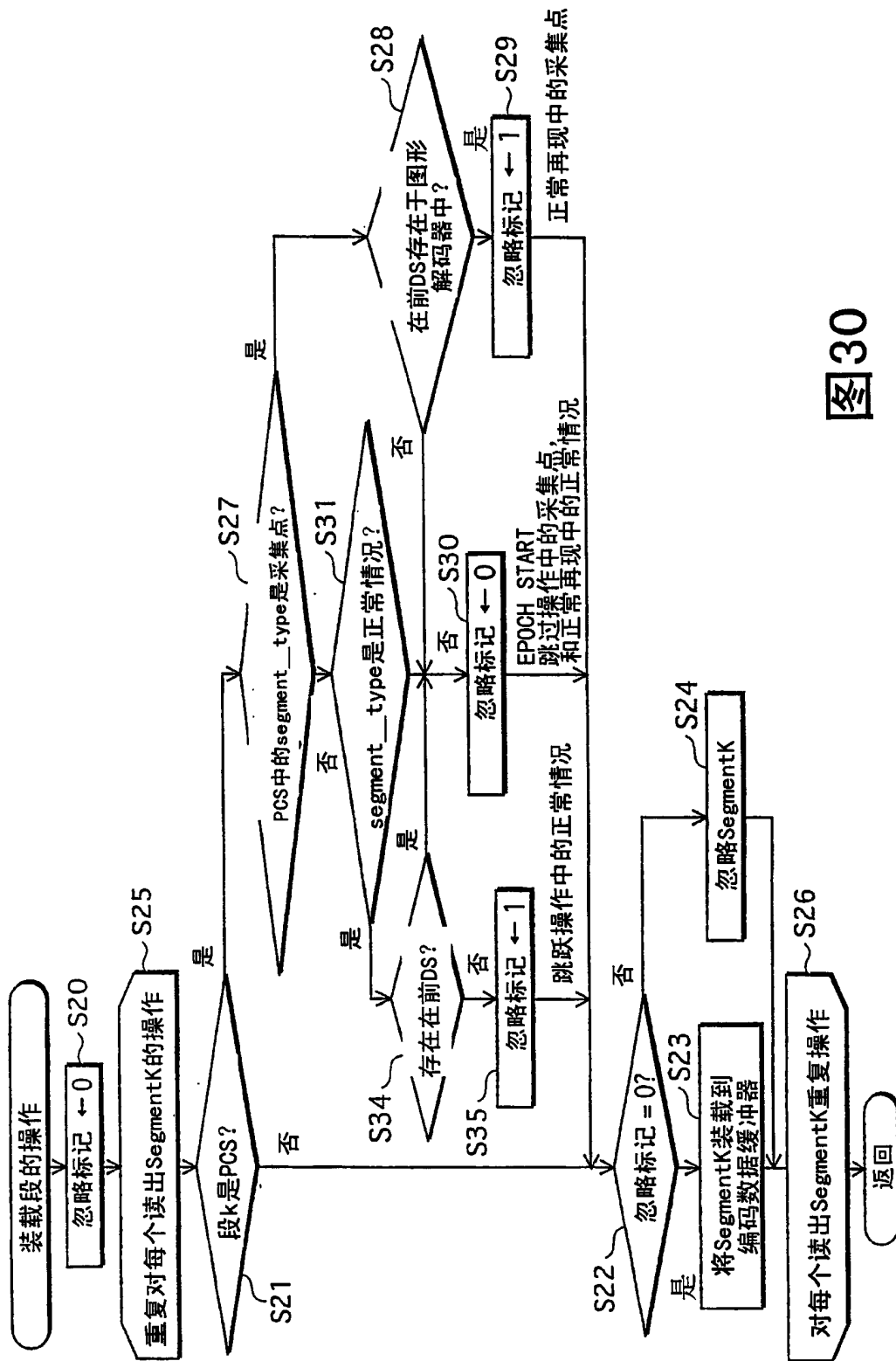


图 30

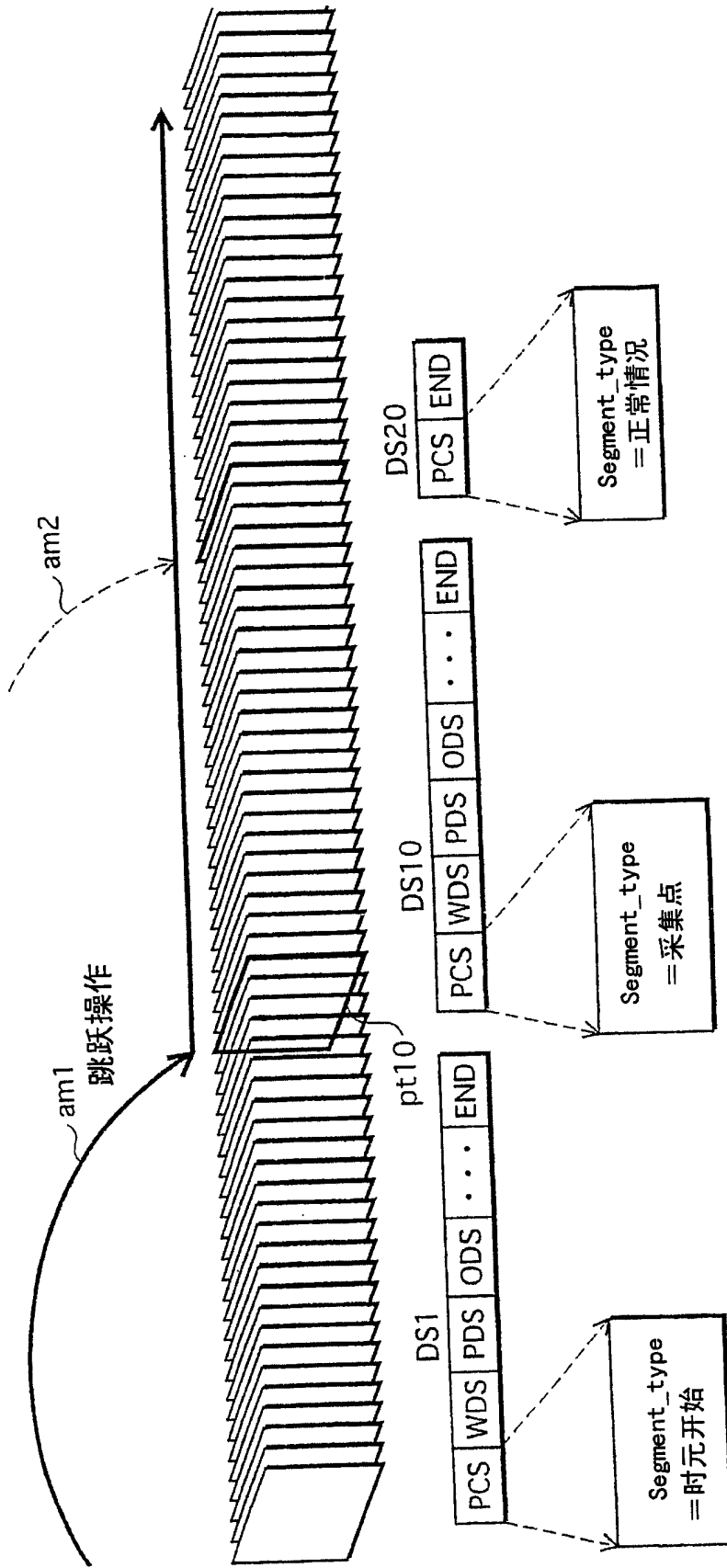


图31

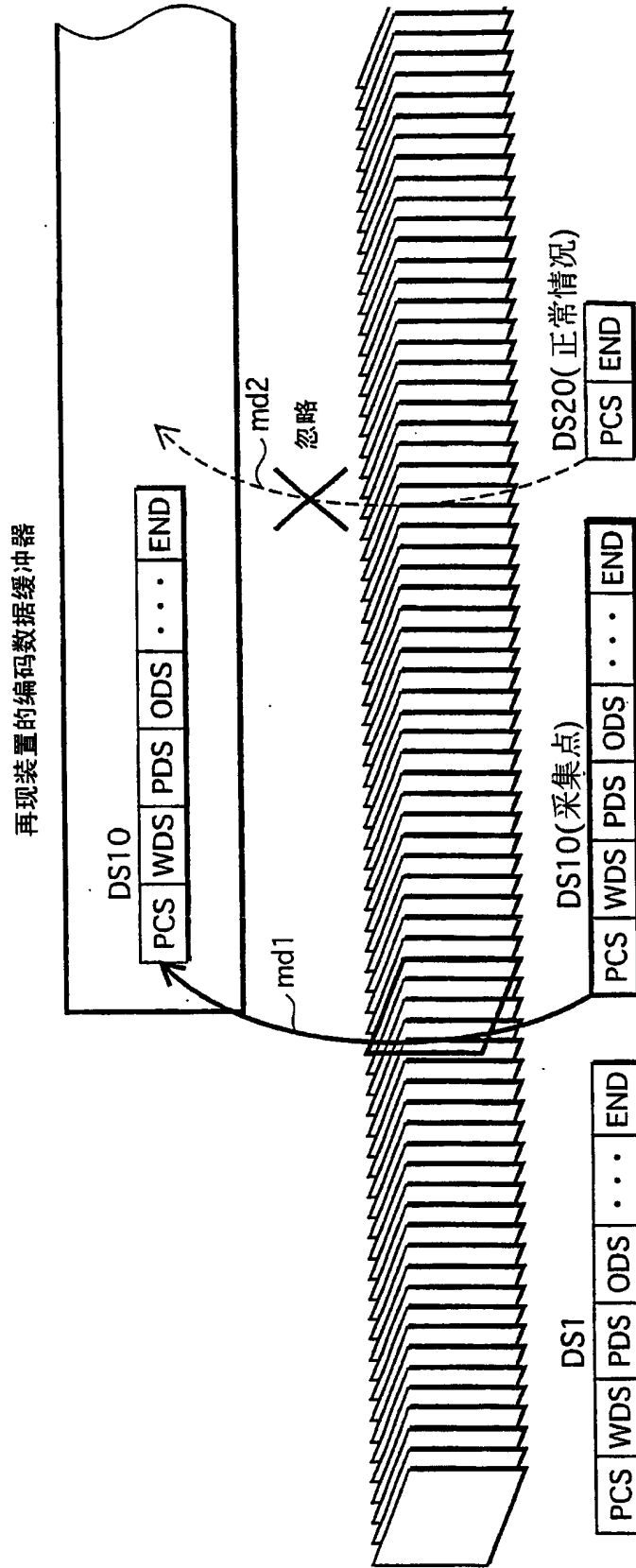


图32

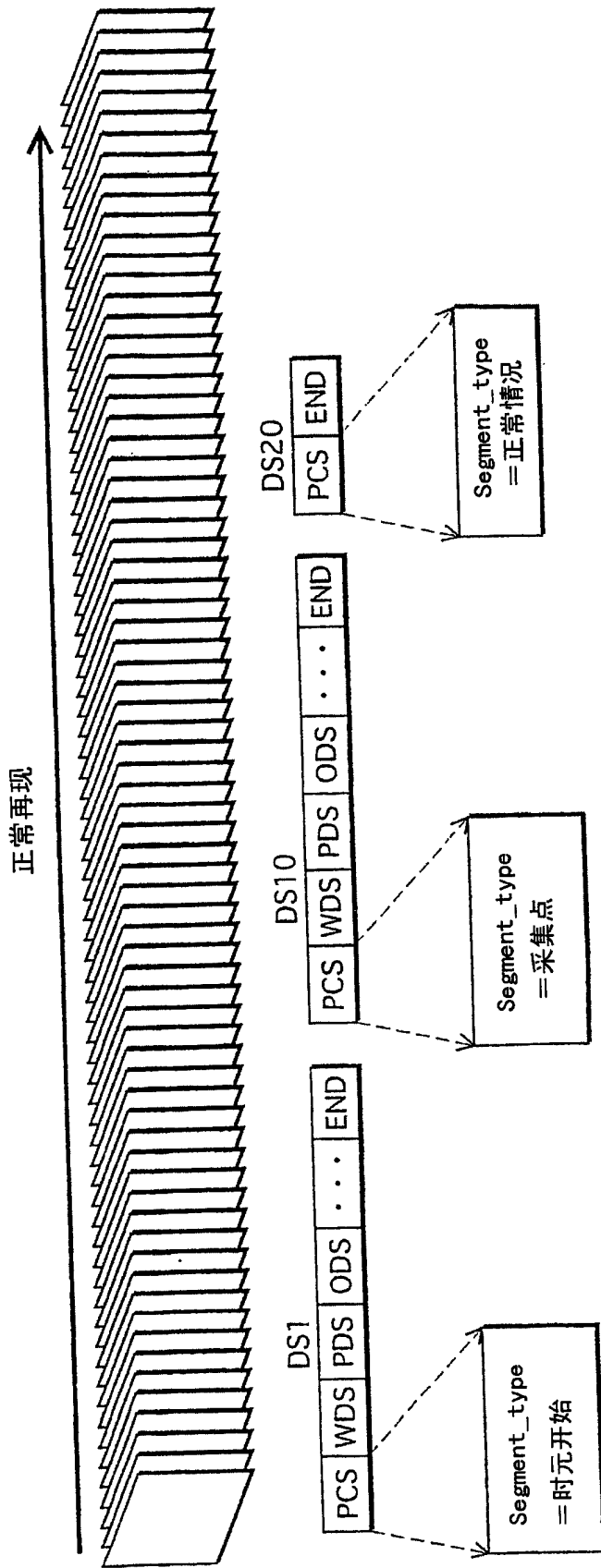


图 33

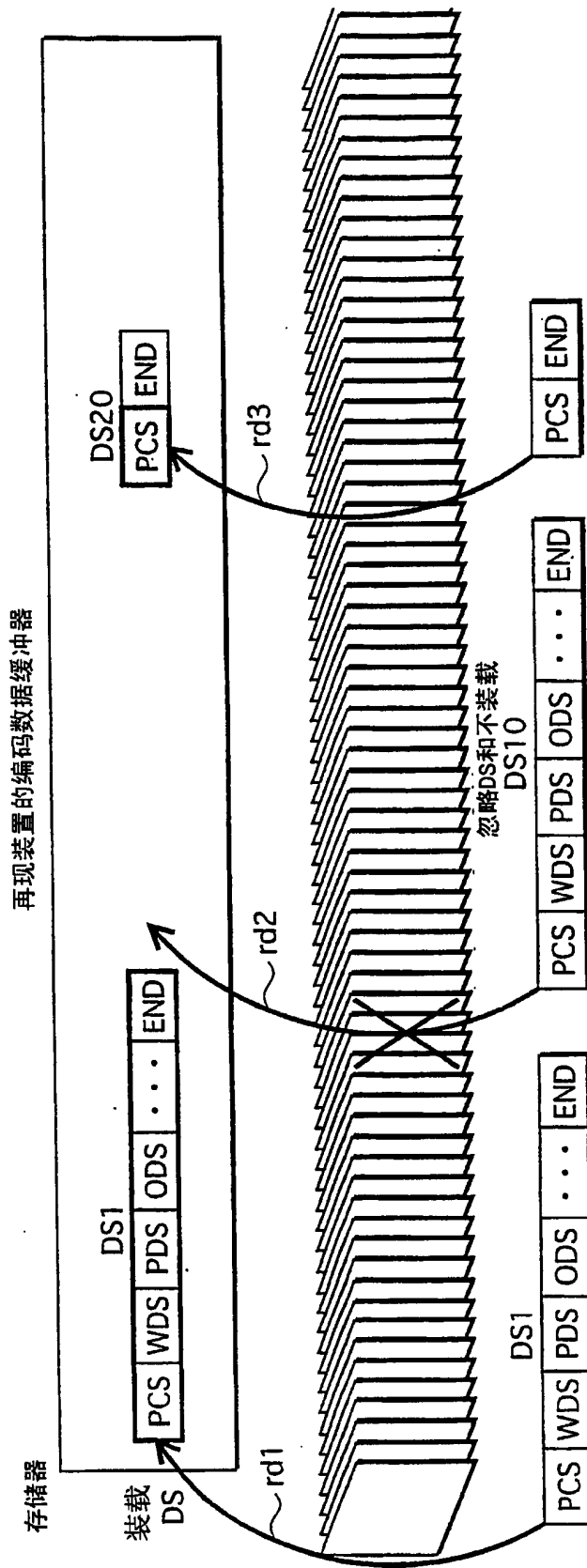


图34

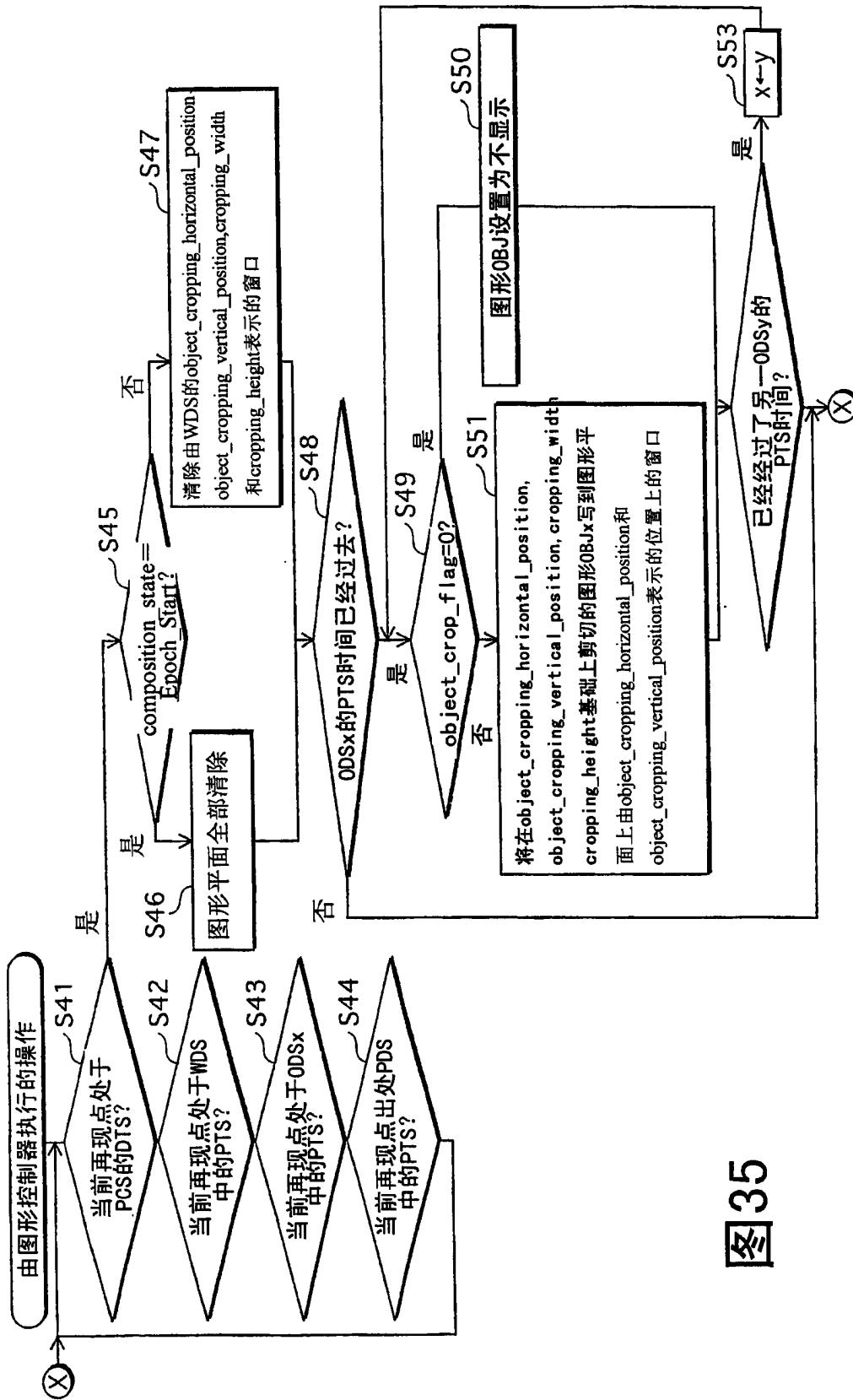


图35

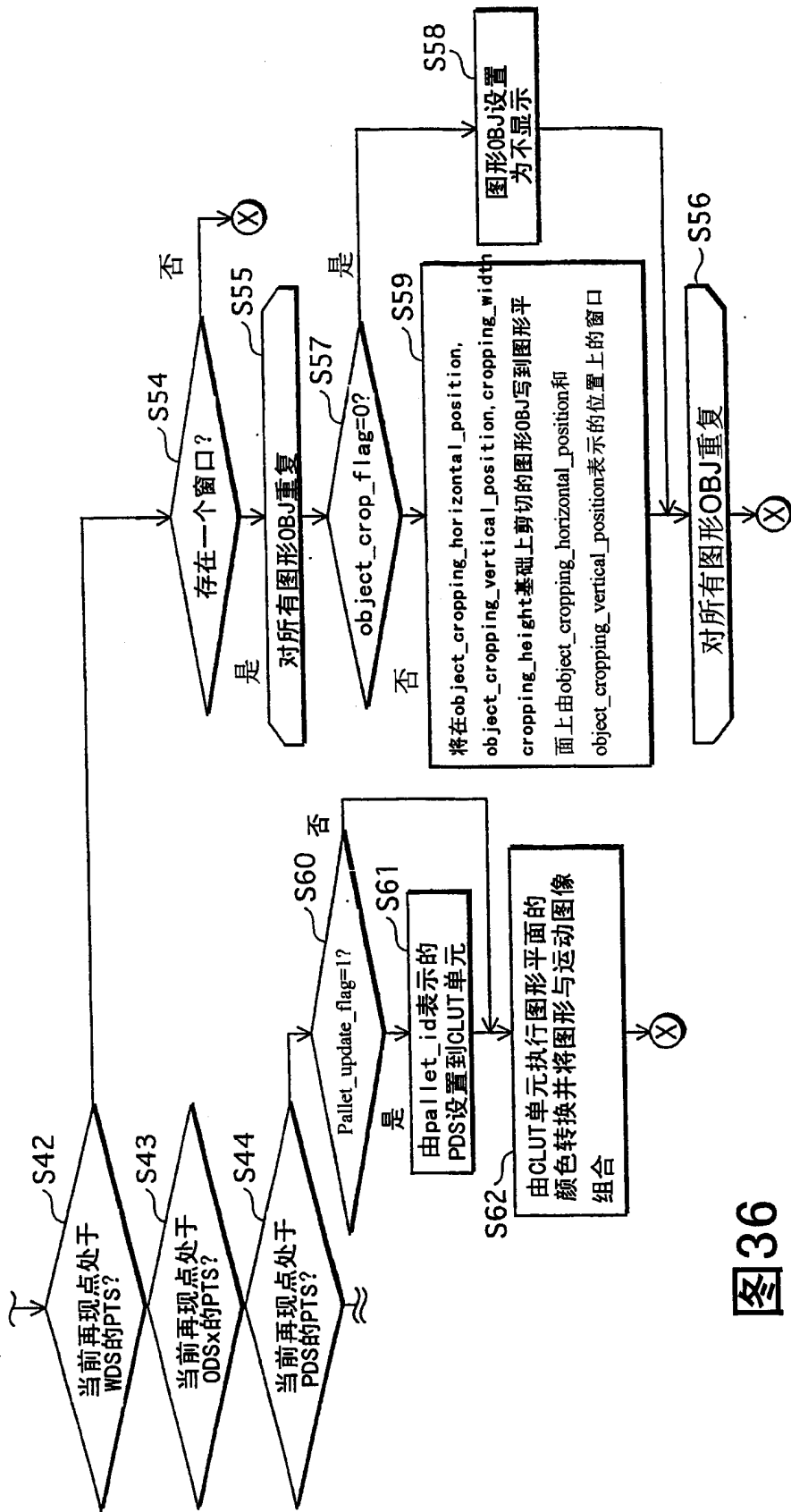


图36

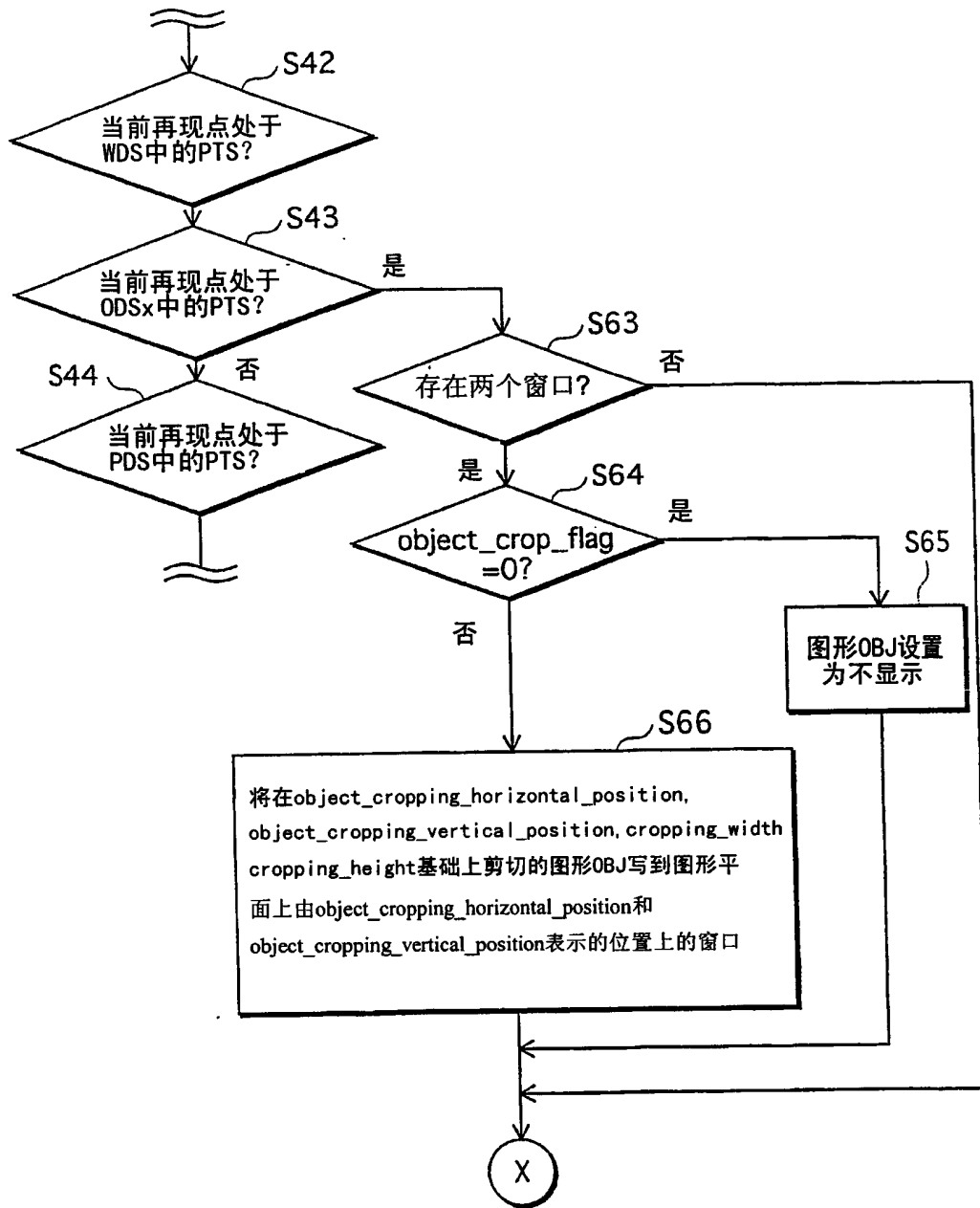


图37

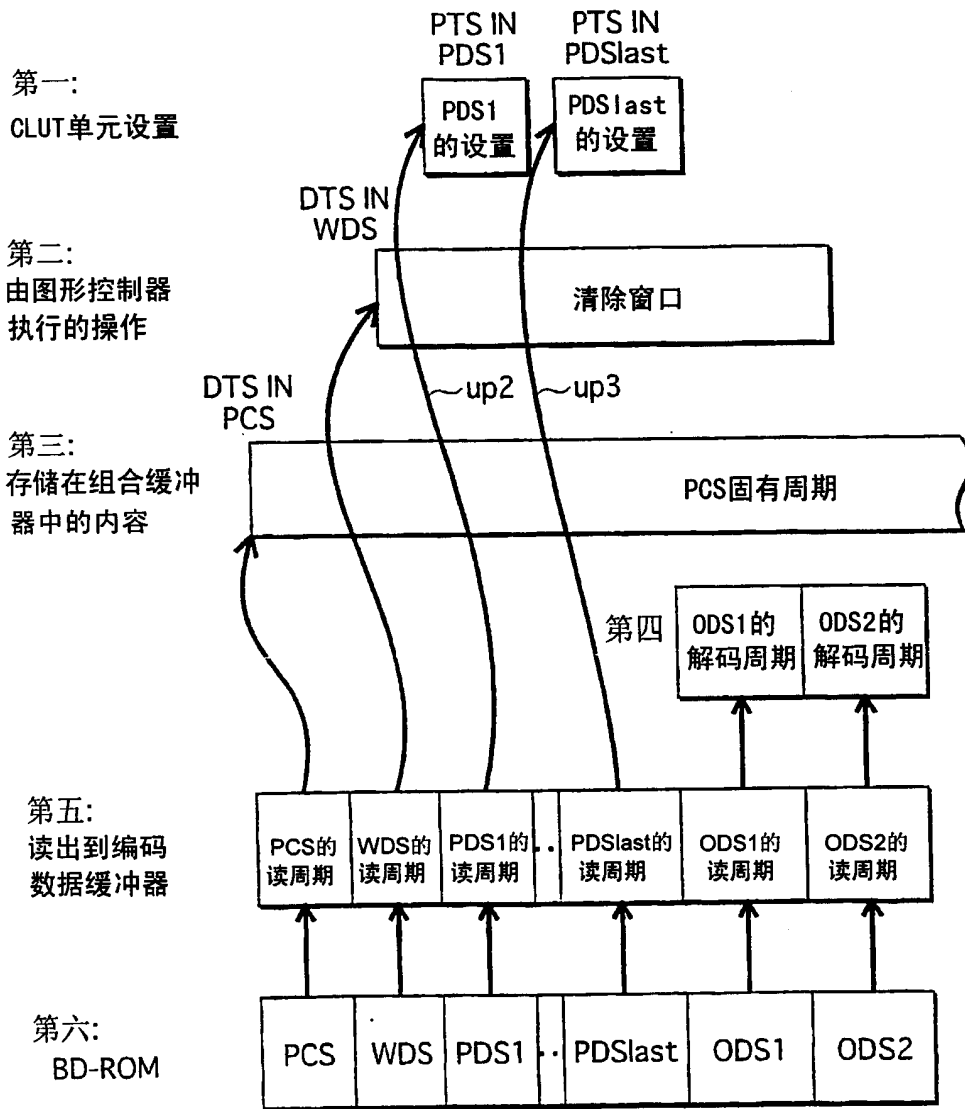


图38

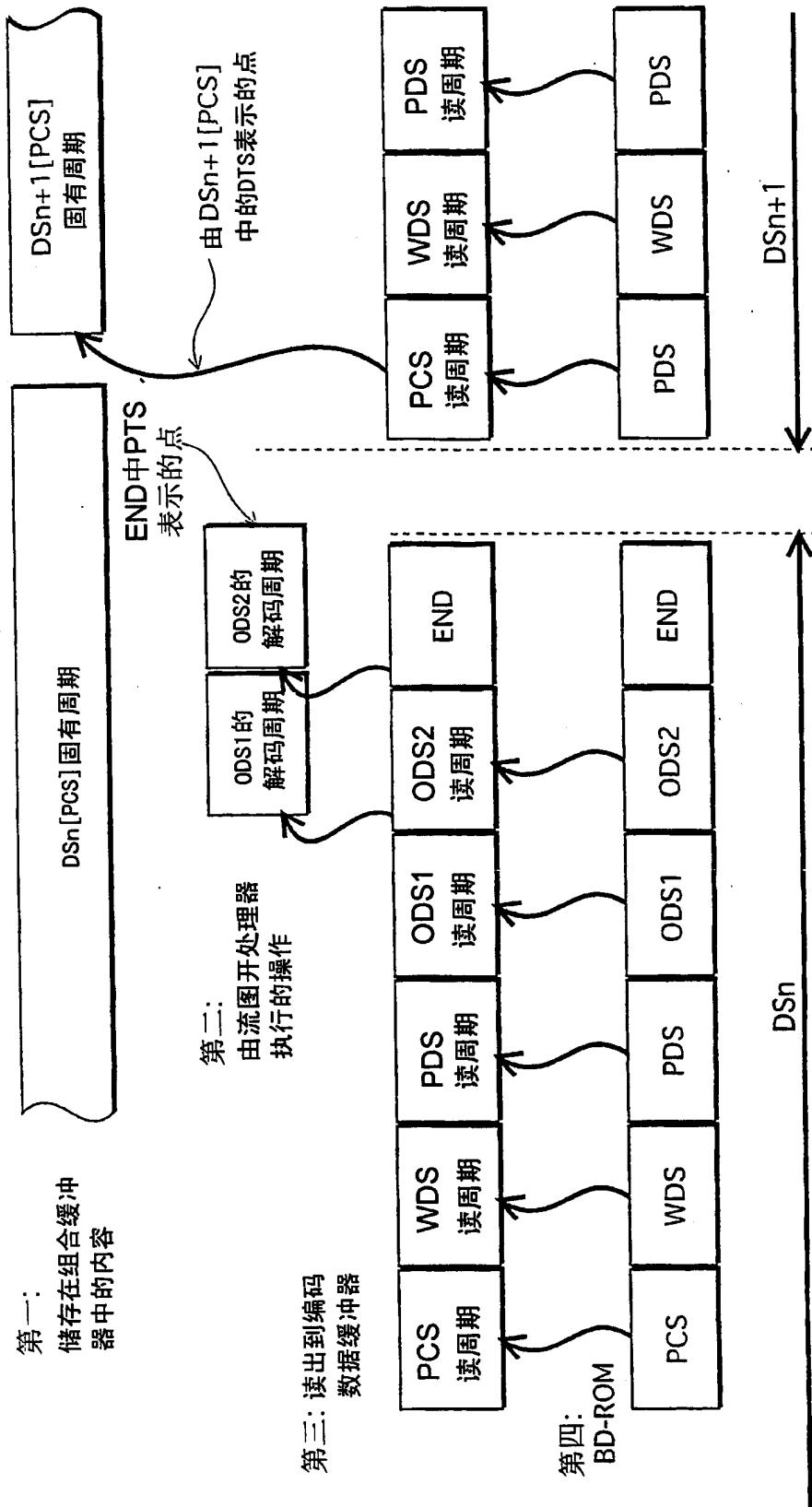


图 39

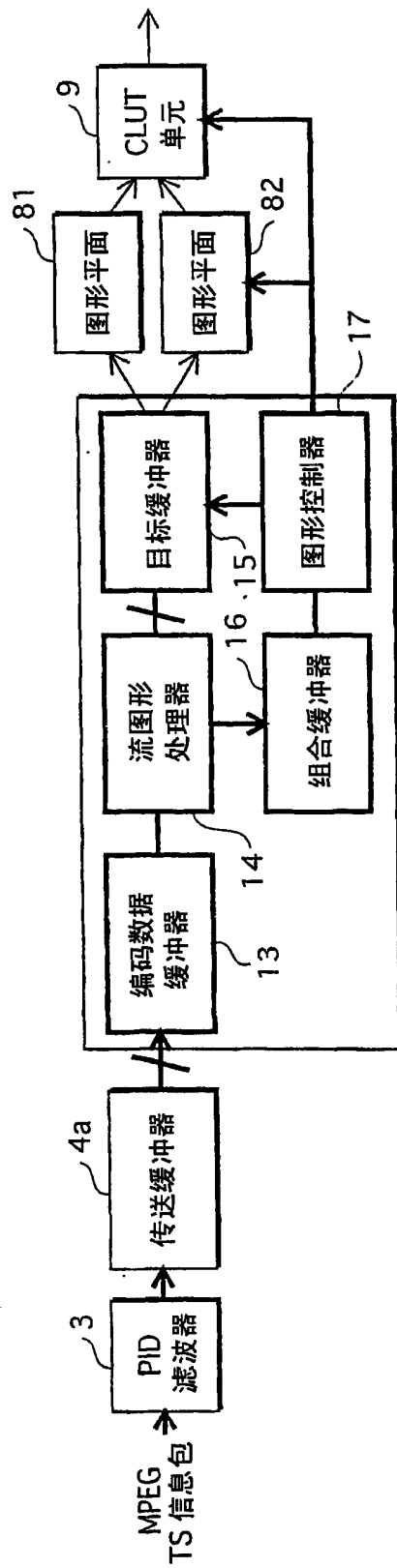


图40

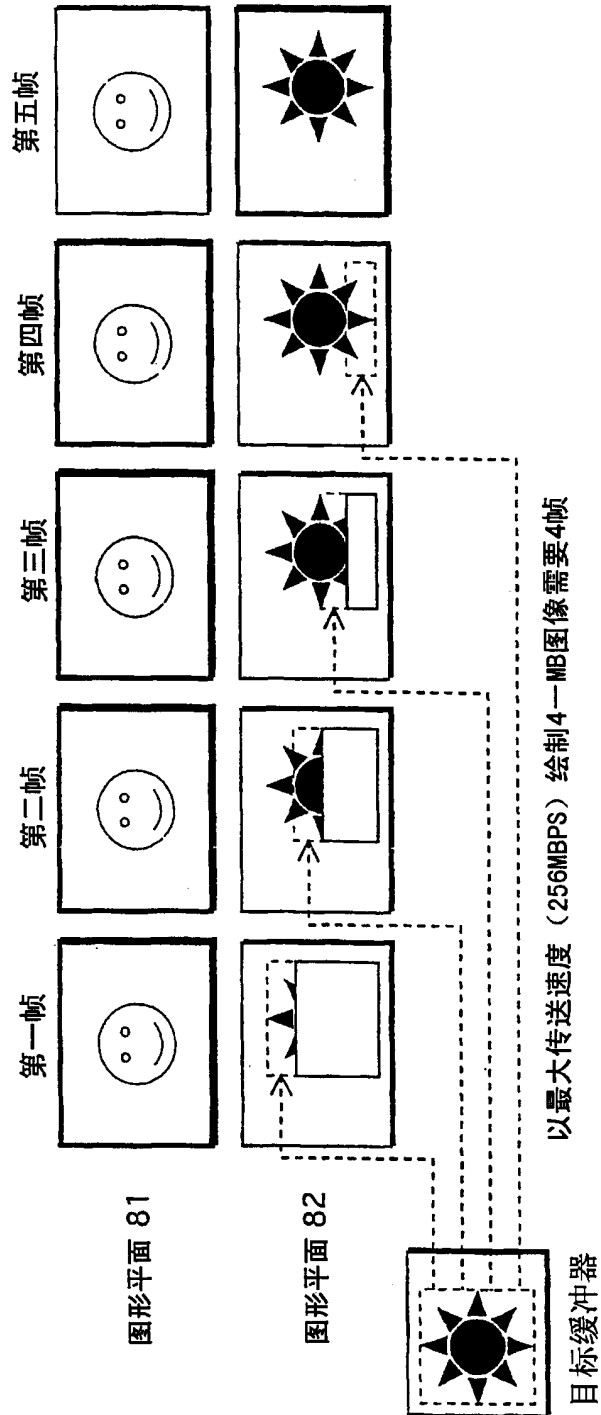


图41

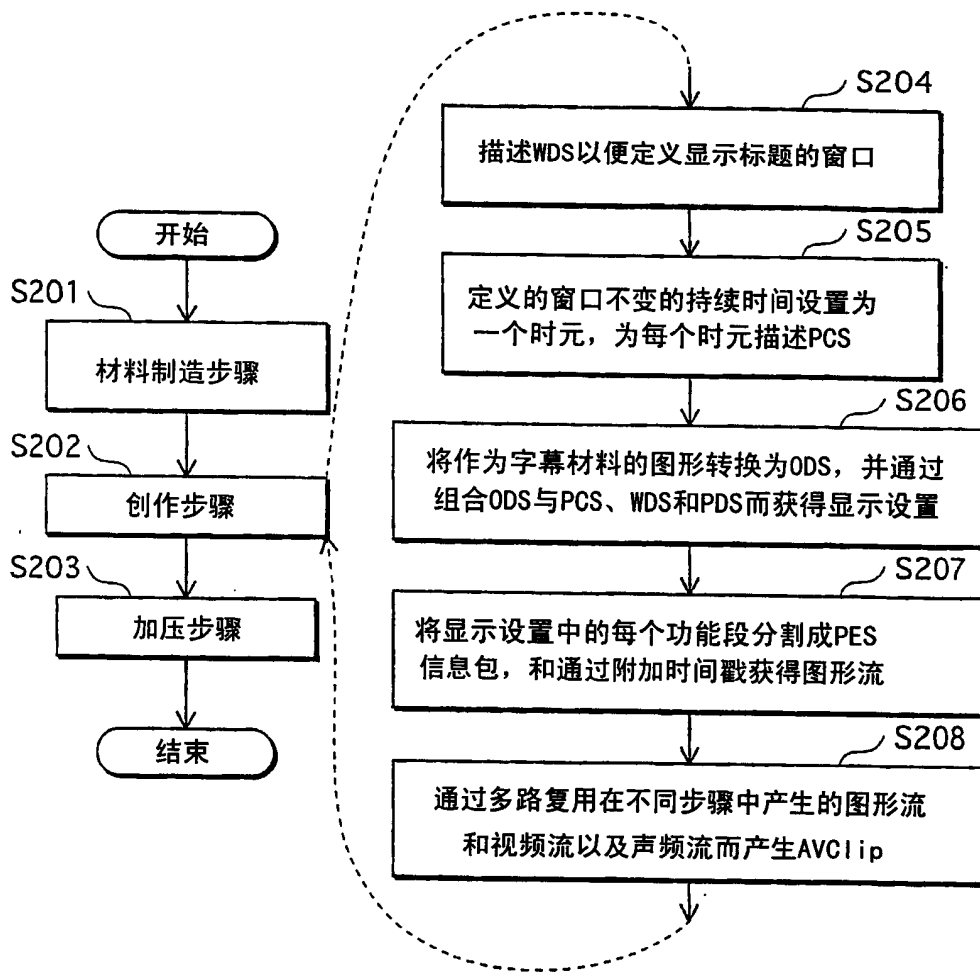


图42