

FIG. 1

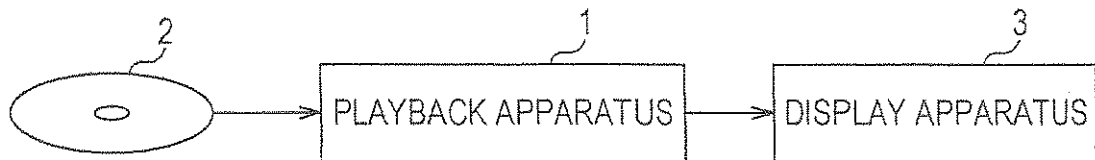


FIG. 2

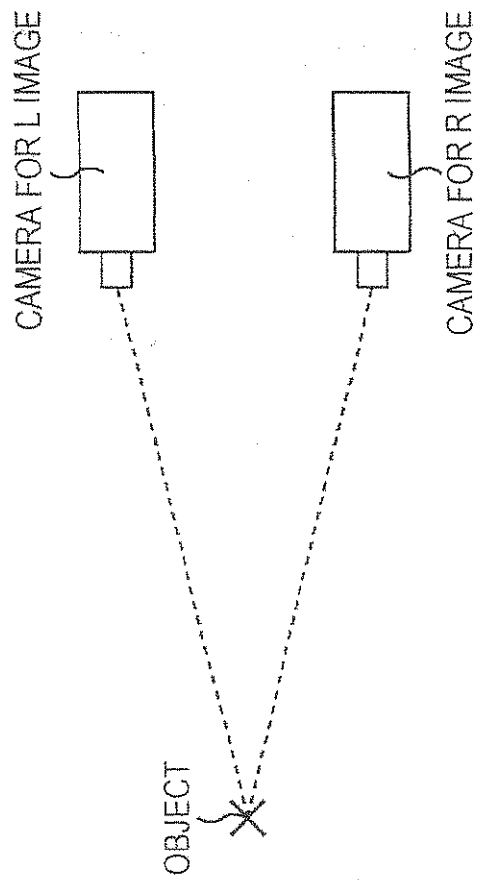


FIG. 3

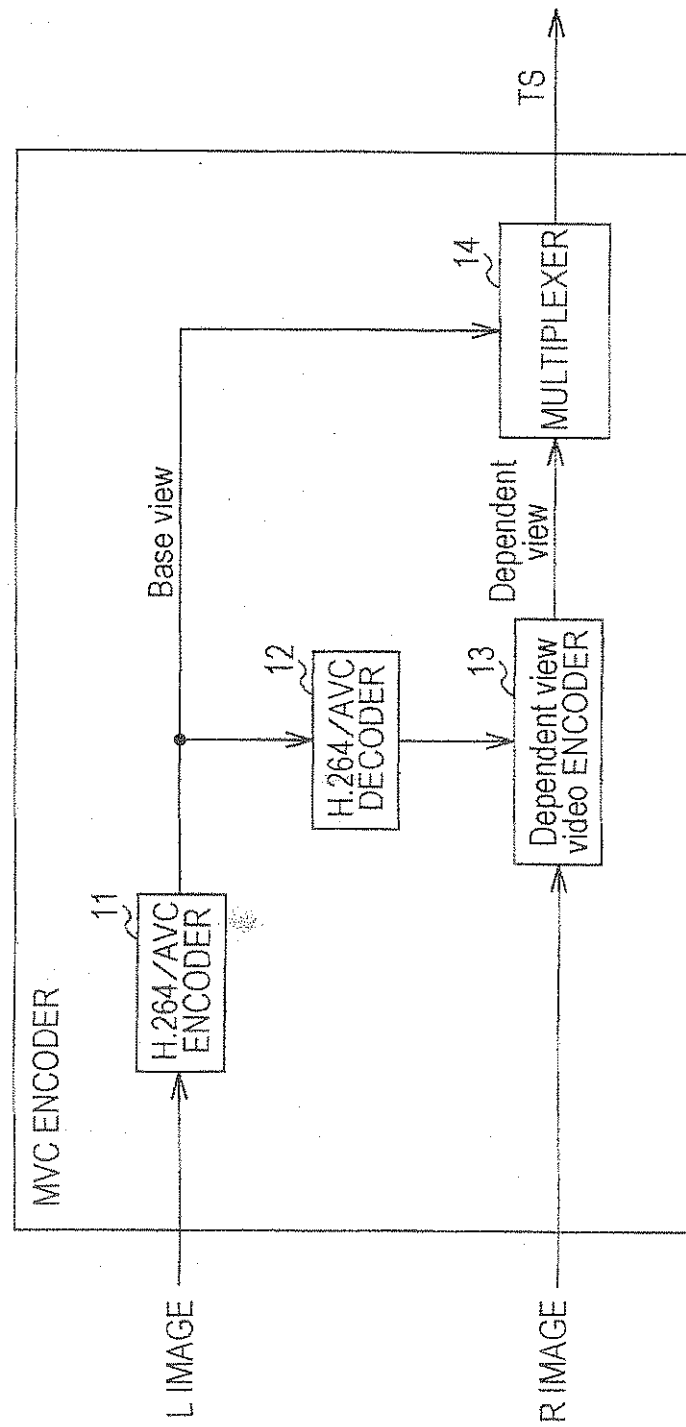


FIG. 4

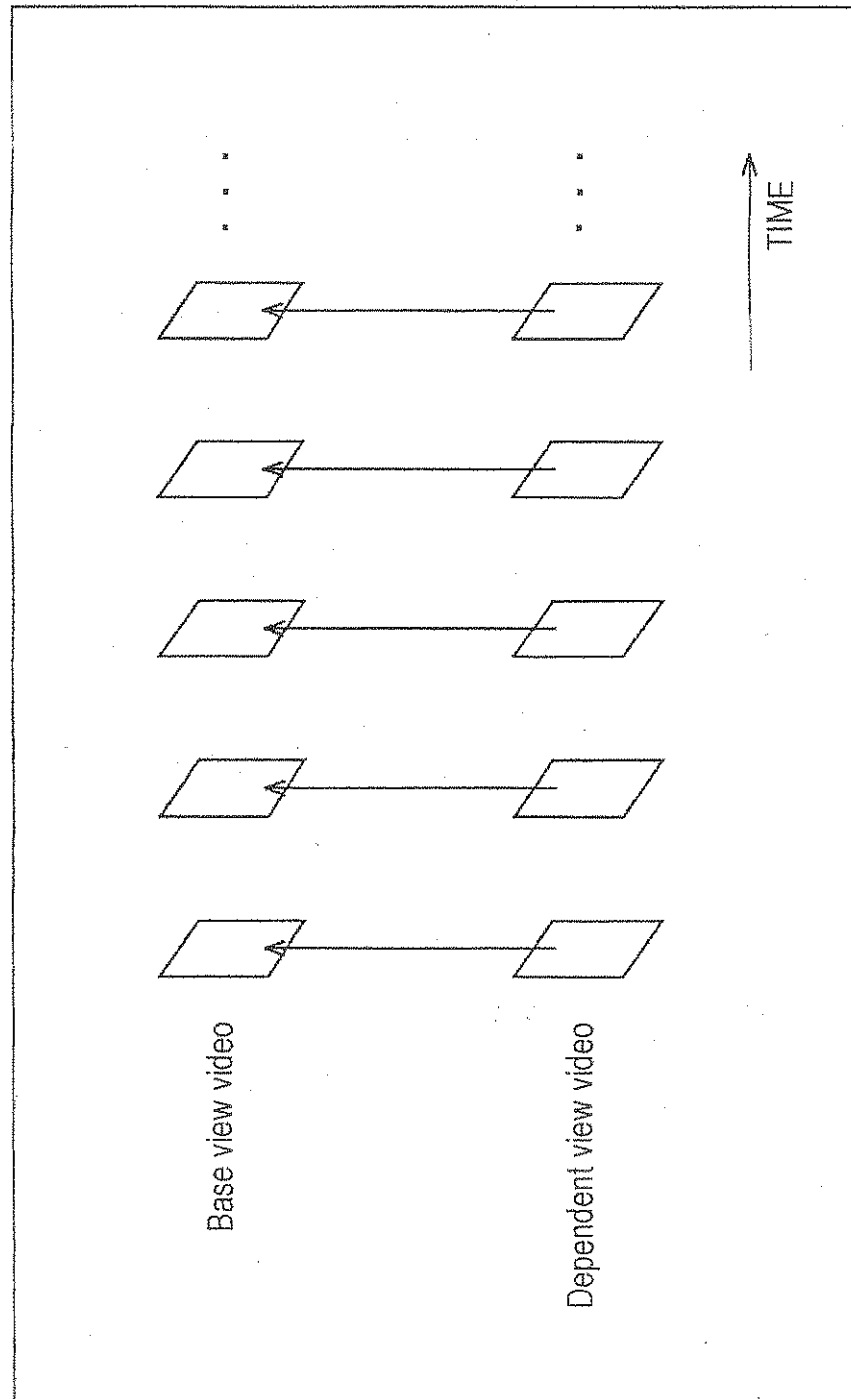


FIG. 5

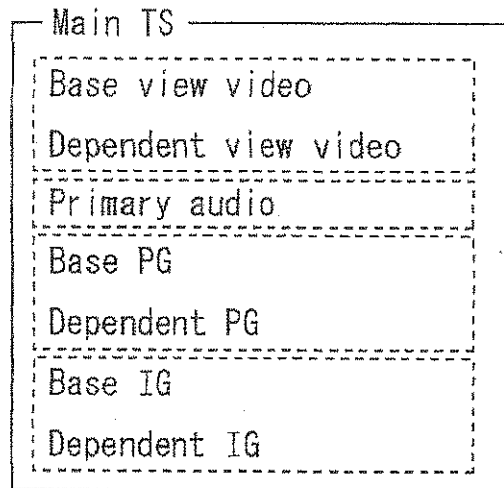


FIG. 6

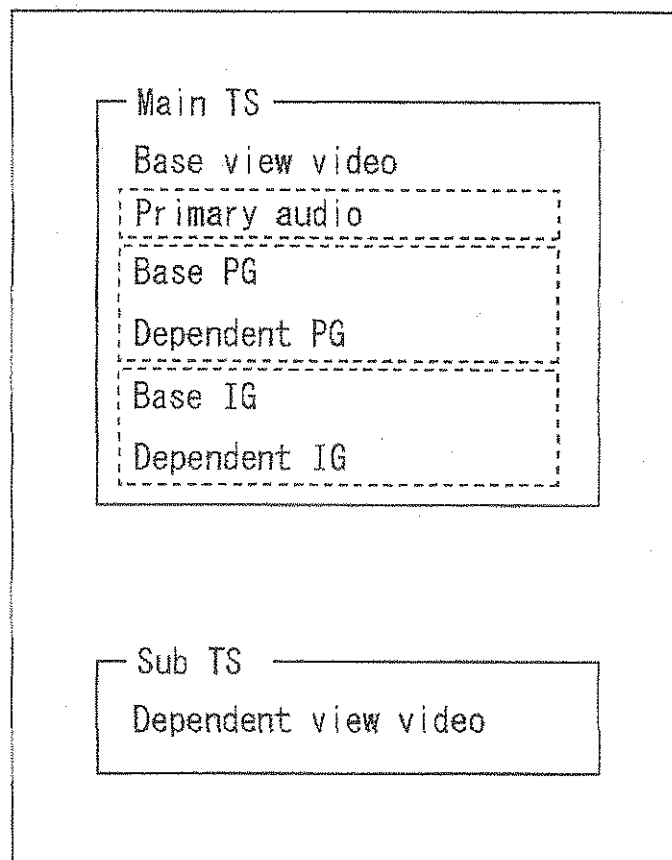


FIG. 7

5/38

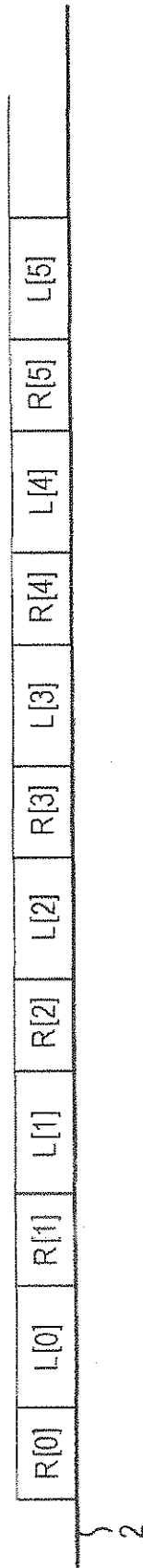


FIG. 9

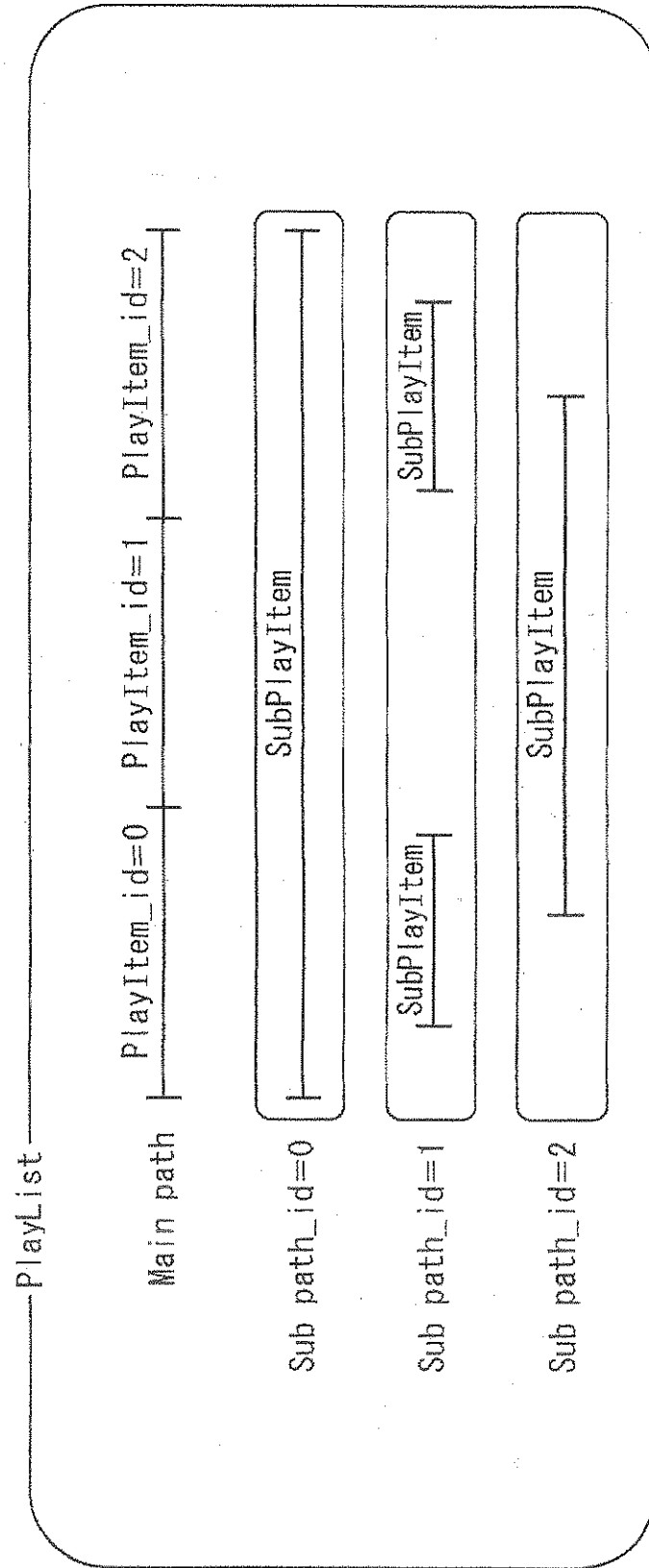


FIG. 10

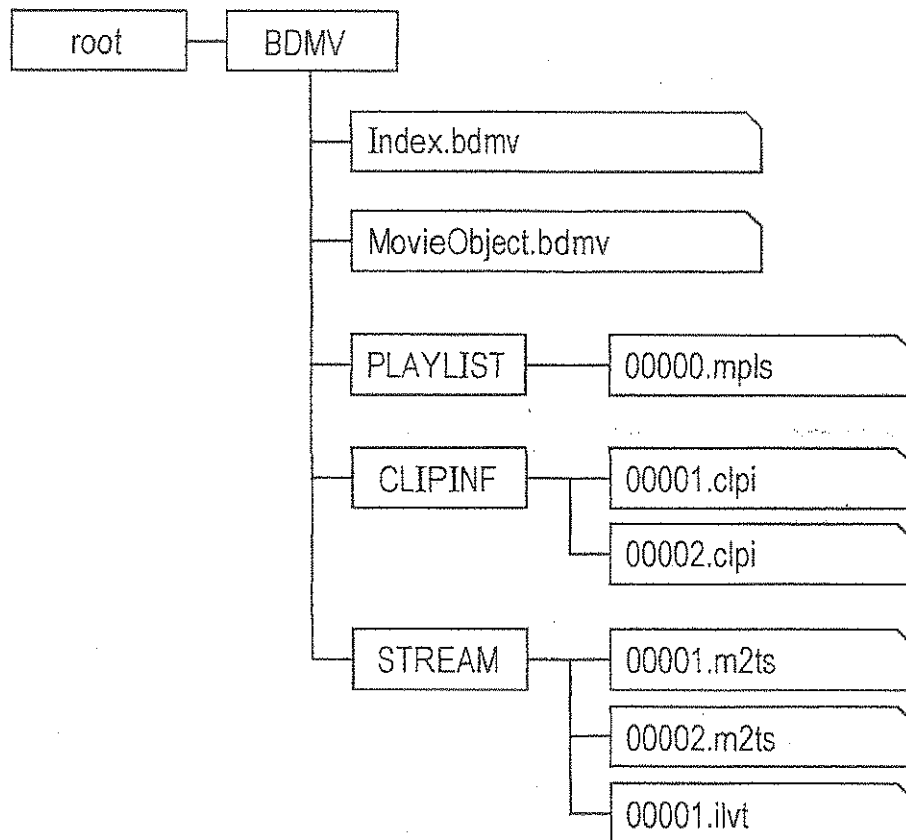


FIG. 11

```

1: yyyyy.mpls {
2:   AppInfoPlayList();
3:   PlayList();
4:   PlayListMark();
5:   ExtensionData();
6: }

7: AppInfoPlayList() {
8:   ...
9:   MVC_flag;
10:  ...
11: }

12: PlayList() {
13:   number_of_PlayItems;
14:   number_of_SubPaths;
15:   for (PlayItem_id=0;
16:     PlayItem_id<number_of_PlayItems;
17:     PlayItem_id++) {
18:     PlayItem();
19:   }
20:   for (SubPath_id=0;
21:     SubPath_id<number_of_SubPaths;
22:     SubPath_id++) {
23:     SubPath();
24:   }
25: }

26: PlayItem() {
27:   Clip_Information_file_name;
28:   MVC_file_type;
29:   reserved;
30:   IN_time;
31:   OUT_time;
32:   STN_table() {ref_to_L_video_PID;}
33: }

34: SubPath() {
35:   SubPath_type;
36:   number_of_SubPlayItems;
37:   for (i=0; i<number_of_SubPlayItems; i++){
38:     SubPlayItem();
39:   }
40: }

41: SubPlayItem() {
42:   Clip_Information_file_name;
43:   SubPlayItem_IN_time;
44:   SubPlayItem_OUT_time;
45: }

46: ExtensionData() {
47:   SubPath_entries_extension();
48:   STN_table_extension();
49: }

```

FIG. 12

MVC_flag	Meaning
0	THIS PlayList DOES NOT INCLUDE INFORMATION REGARDING MVC PLAYBACK.
1	THIS PlayList INCLUDES INFORMATION REGARDING MVC PLAYBACK.

FIG. 13

MVC_file_type	Meaning
0	BOTH OF L video and R video ARE SUPPLIED FROM ONE TS REPRESENTED BY Clip_Information_file_name.
1	FIRST TS file INCLUDING L VIDEO REPRESENTED BY Clip_Inforamtion_file_name AND SECOND TS file INCLUDING R video ARE INTERLEAVED ON BD-ROM disc.
2	BOTH OR ONE OF FIRST TS file INCLUDING L VIDEO REPRESENTED BY Clip_Inforamtion_file_name AND SECOND TS file INCLUDING R video ARE SUPPLIED FROM LOCAL STORAGE.
3	Reserved

FIG. 14

SubPath_type	Meaning
0	reserved
1	reserved
2	Out-of-mux and primary audio presentation path of the Browsable slideshow (The audio presentation path using the SubPath is not synchronized with the main path using PlayItems in the PlayList.)
3	Out-of-mux and interactive graphics menu (The interactive graphics menu using the SubPath is not synchronized with the main path using PlayItems in the PlayList.)
4	Out-of-mux and text subtitle presentation path (The text subtitle presentation path using the SubPath is synchronized with the main path using PlayItems in the PlayList.)
5	Out-of-mux and AV synchronized type of one or more elementary streams path (Primary audio/PG/IG/Secondary audio path.) Out-of-mux and AV synchronized type of Picture-in-Picture presentation path which contains one or more elementary stream paths. (The elementary streams used by the path are multiplexed in a separate Clip from other Clip used by PlayItem. The path using the SubPath is synchronized with the main path using PlayItems in the PlayList.)
6	Out-of-mux and AV non-synchronized type of Picture-in-Picture presentation path which contains one or more elementary stream paths. (The elementary streams used by the path are multiplexed in a separate Clip from other Clip used by PlayItem. The path using the SubPath is not synchronized with the main path using PlayItems in the PlayList.)
7	In-mux type and AV synchronized type of Picture-in-Picture presentation path which contains one or more elementary stream paths. (The elementary streams used by the path is multiplexed in the same Clip used by PlayItem. The path using the SubPath is synchronized with the main path using PlayItems in the PlayList.)
8-255	reserved

FIG. 15

SubPath_entries_extension()

Syntax	No. of bits	Mnemonic
SubPath_entries_extension() {		
length	32	uimbsf
number_of_SubPath_extensions	16	uimbsf
for (SubPath_id=number_of_SubPaths;		
SubPath_id<number_of_SubPaths		
+number_of_SubPath_extensions;		
SubPath_id++) {		
SubPath_extension()		
}		
}		

FIG. 16

SubPath_extension()

Syntax	No. of bits	Mnemonic
SubPath_extension() {		
length	32	uimsbf
reserved_for_future_use	8	bslbf
SubPath_type_extension	8	bslbf
reserved_for_future_use	24	bslbf
number_of_SubPlayItems	8	uimsbf
for (i=0; i<number_of_SubPlayItems; i++) {		
SubPlayItem(i)		
}		
}		

FIG. 17

SubPath_type_ extension	Meaning
0-7	N/A
8	OUT-of-mux and R (dependent) video path SUB PLAYBACK PATH OF TS INCLUDING MVC R (dependent) video, THE TS BEING A FILE DIFFERENT FROM TS FOR MAIN PLAYBACK PATH OF PlayItem.
others	reserved

FIG. 18

STN_table_extension

Syntax	No. of bits	Mnemonic
STN_table_extension() {		
for (PlayItem_id=0; PlayItem_id<number_of_PlayItems;		uimsbf
PlayItem_id++) {		
length	16	uimsbf
ref_to_R_video_info() {		
stream_entry()		
stream_attributes()		
}		
}		

FIG. 19

stream_entry()

Syntax	No. of bits	Mnemonic
stream_entry() {		
length	8	uimsbf
type	8	bslbf
if (type==1) {		
ref_to_stream_PID_of_mainClip	16	uimsbf
reserved_for_future_use	48	bslbf
} else if (type==2) {		
ref_to_SubPath_id	8	uimsbf
ref_to_subClip_entry_id	8	uimsbf
ref_to_stream_PID_of_subClip	16	uimsbf
reserved_for_future_use	32	bslbf
}		
}		

FIG. 20

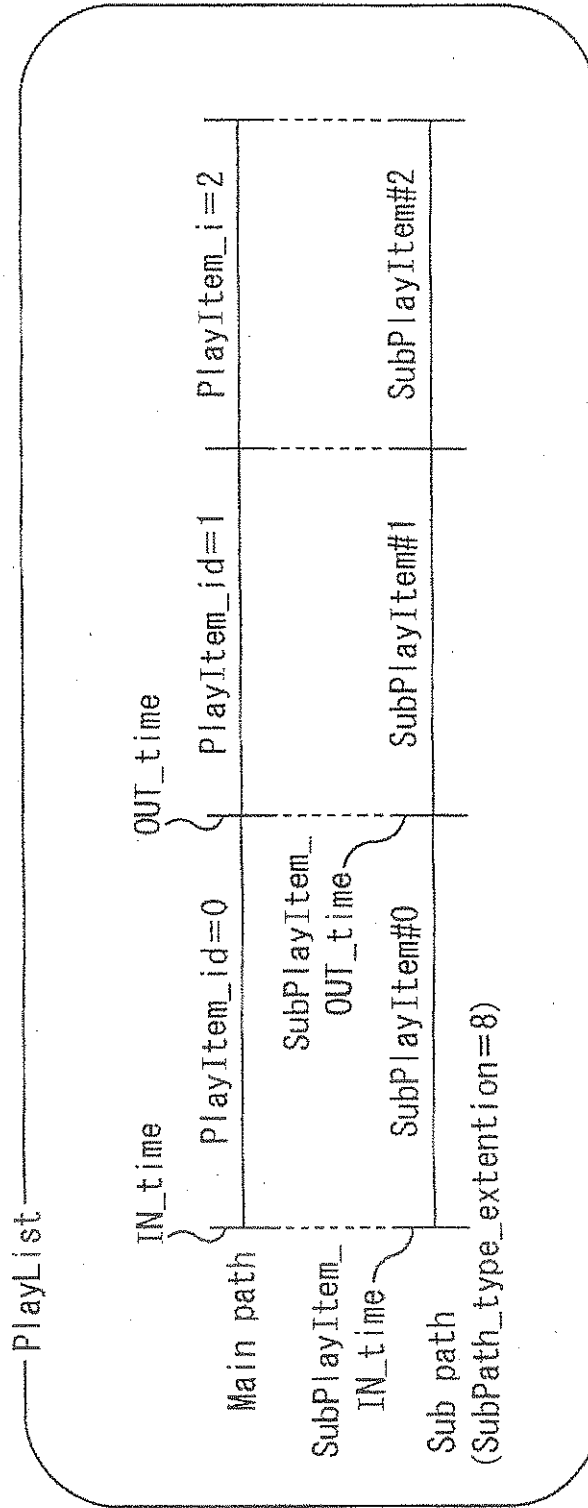


FIG. 21

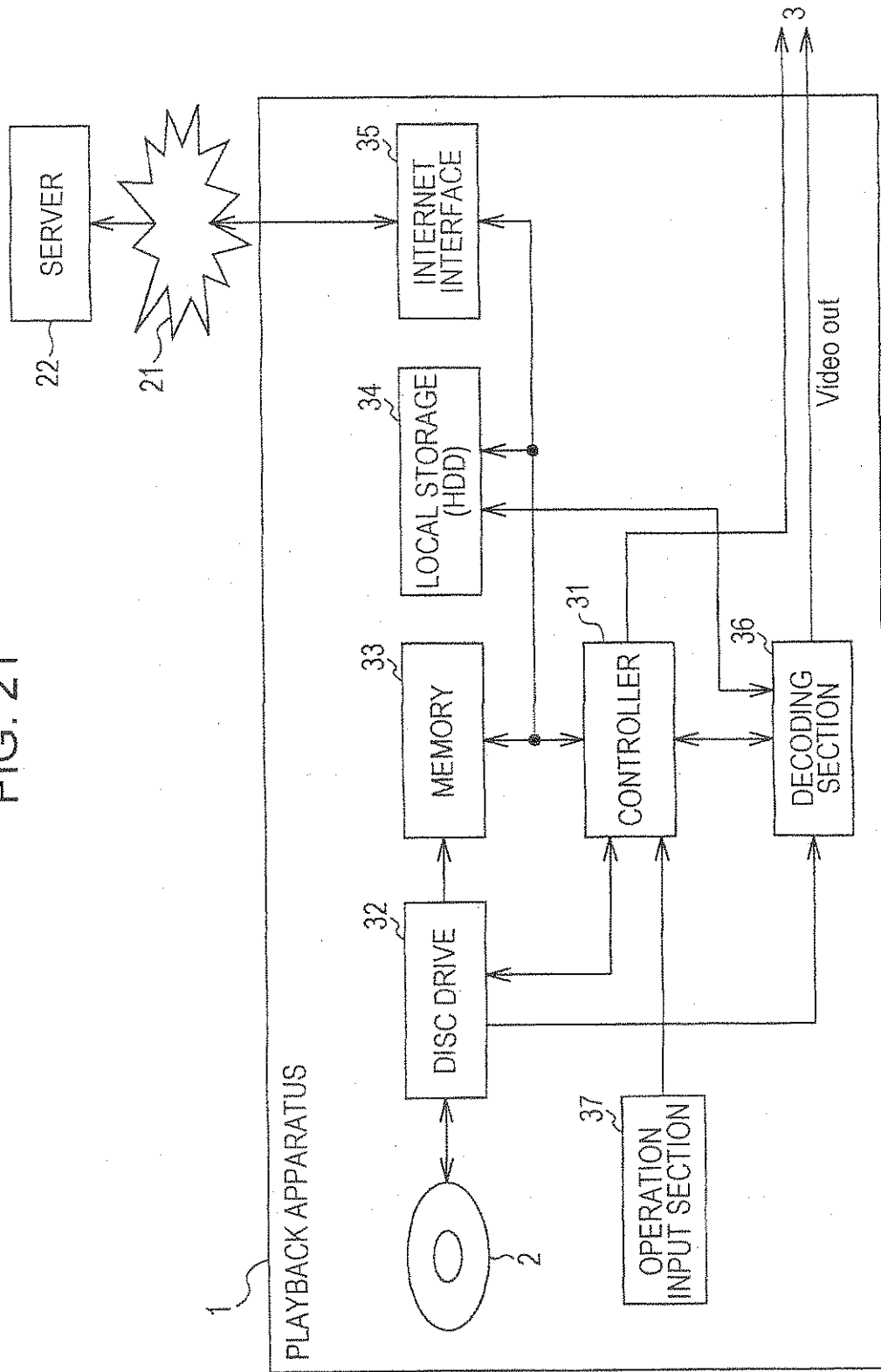


FIG. 22

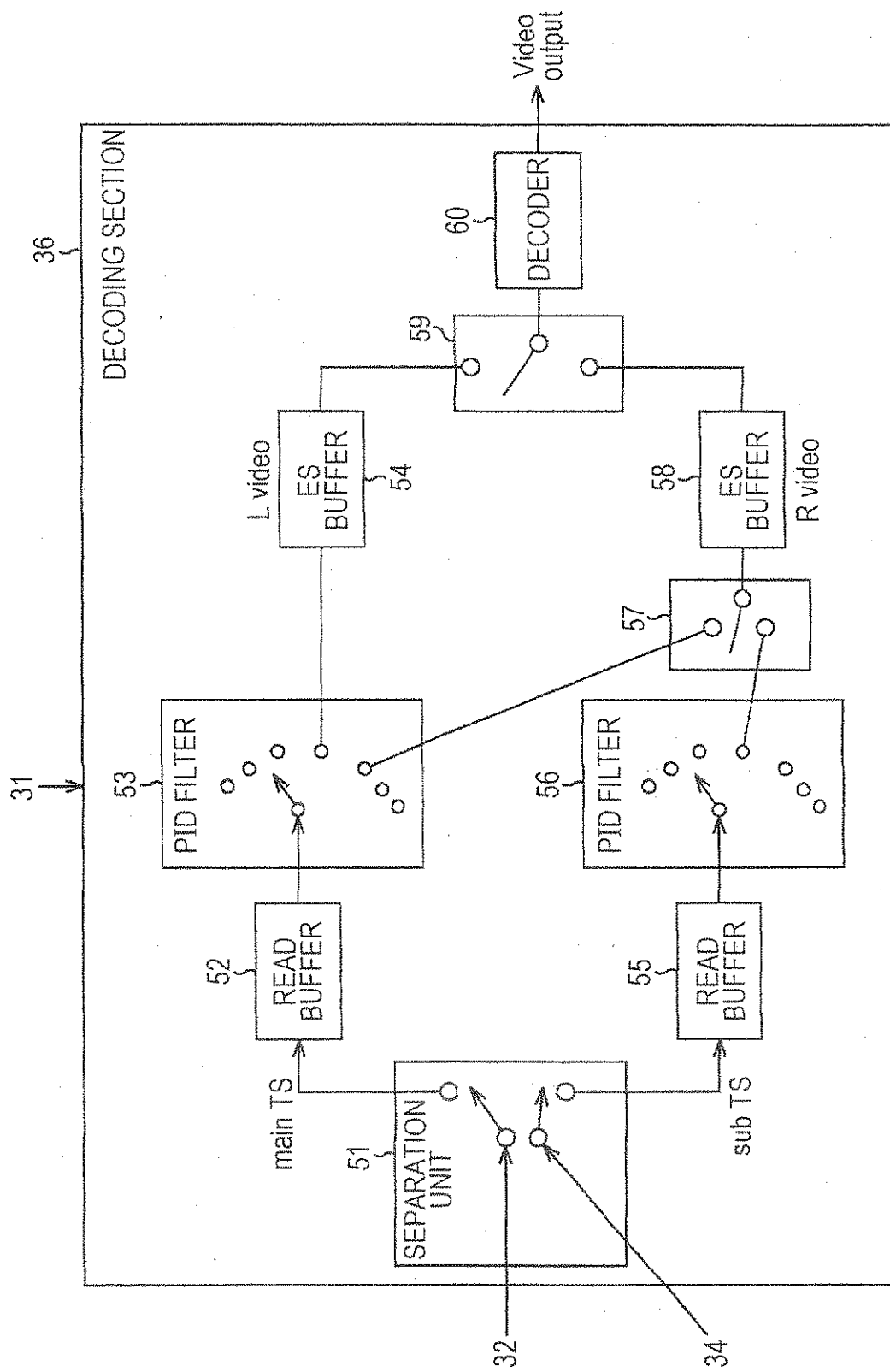


FIG. 23

```

00000.mpls{
  AppInfoPlayList() {MVC_flag=1}
  PlayList() {
    PlayItem() {
      "00001" ; // ref to 00001.clpi
      MVC_file_type=1;
      IN_time; OUT_time;
      STN_table() {ref_to_L_video_PID;}
    }
  }
  ExtensionData() {
    SubPath_entries_extension() {
      SubPath_extension() { //SubPath_id=0 IS ASSIGNED TO THIS SubPath
        SubPath_type_extension=8;
        SubPlayItem() {
          "00002" ; // ref to 00002.clpi
          SubPlayItem_IN_time; SubPlayItem_OUT_time;
        }
      }
    }
    STN_table_extension() {type=2; ref_to_SubPath_id=0
    ref_to_R_video_PID;}
  }
}

```

FIG. 24

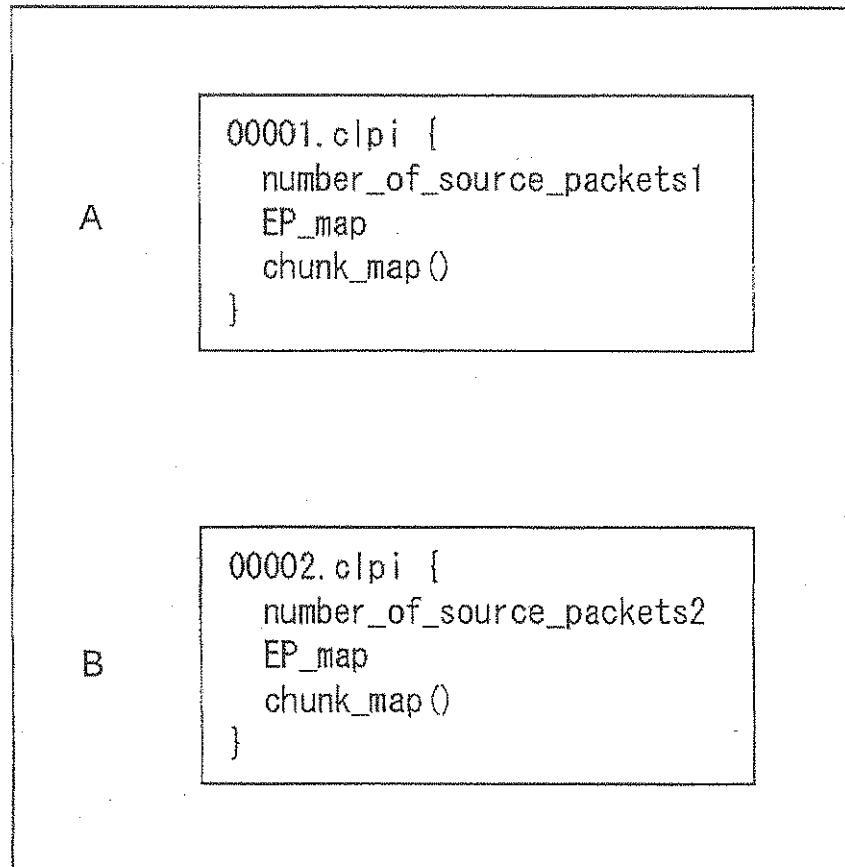


FIG. 25

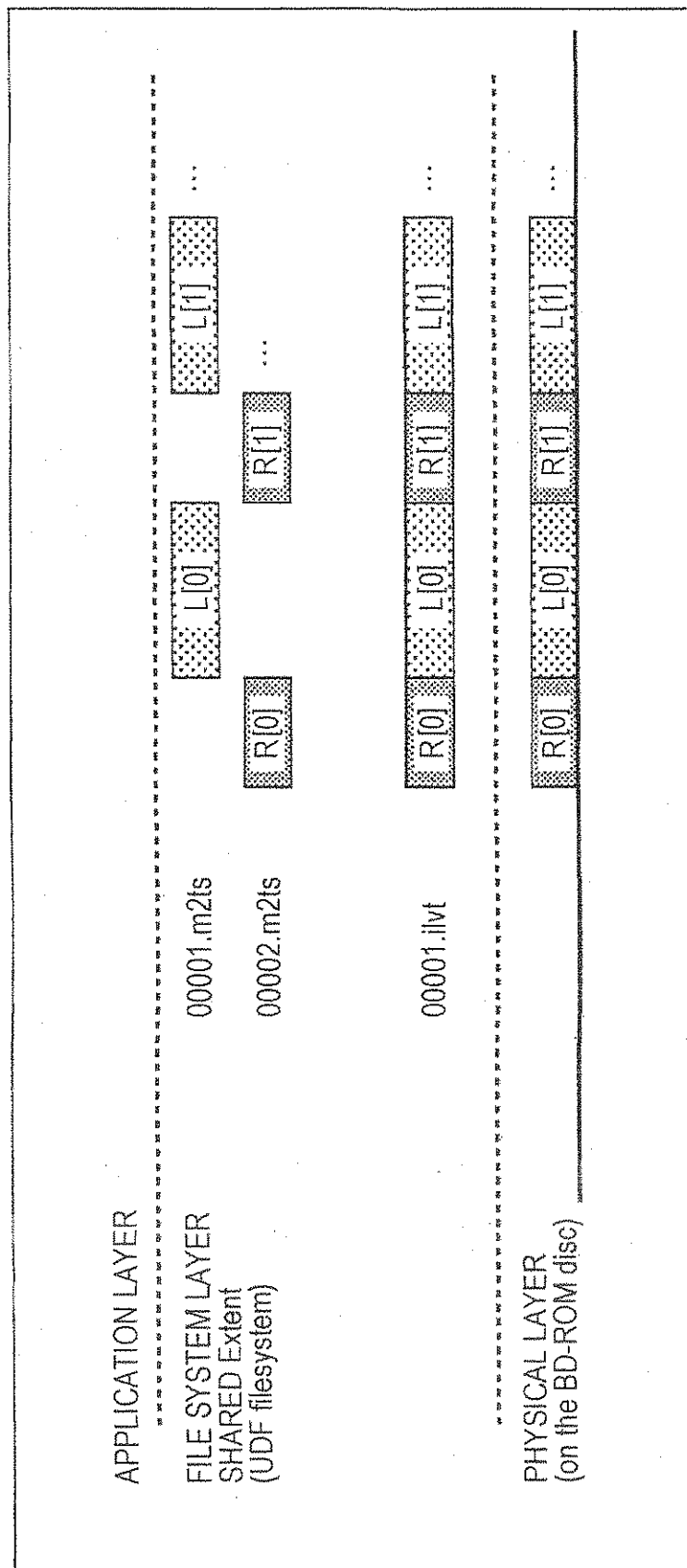


FIG. 26

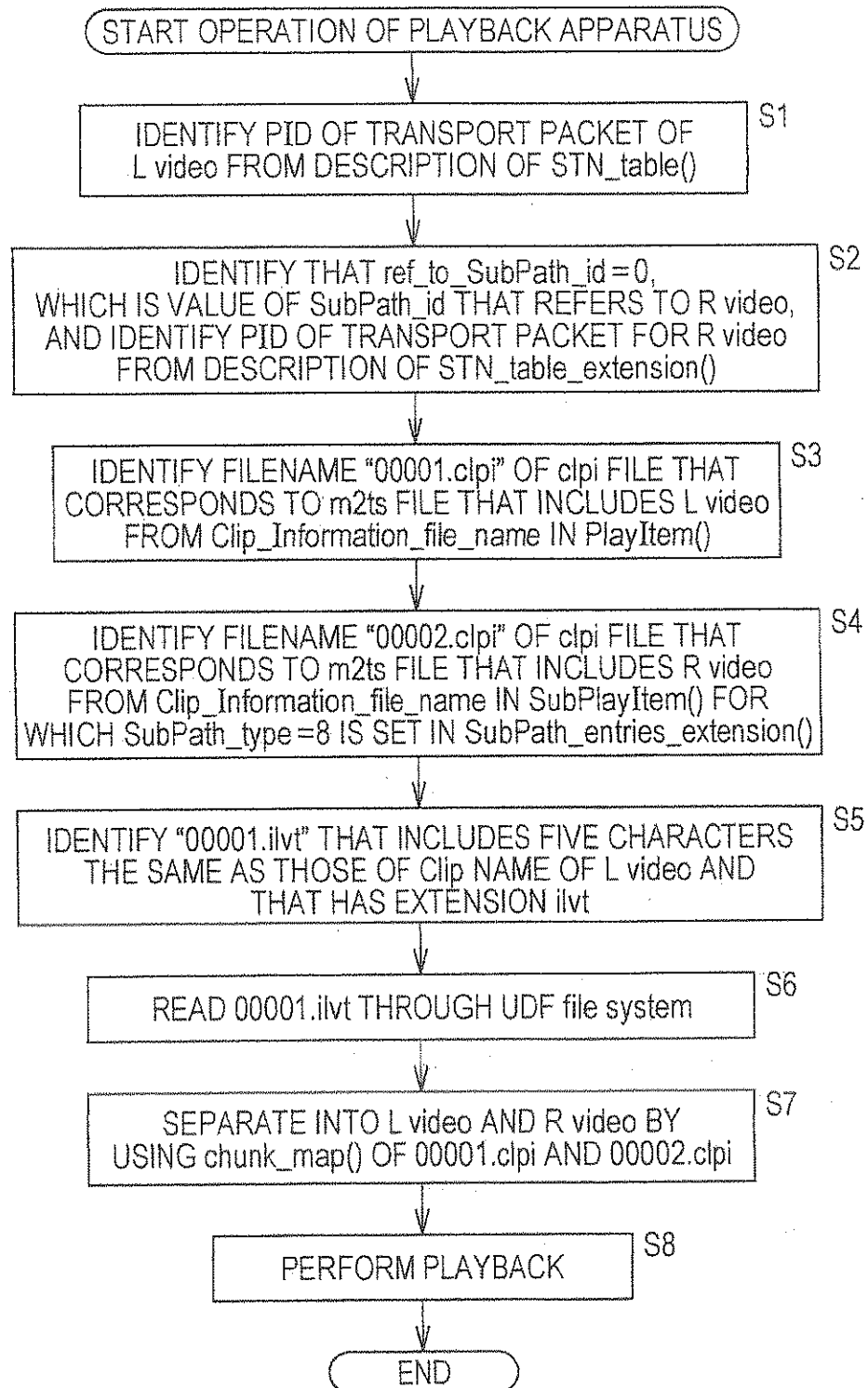


FIG. 27

```

chunk_map() {
    number_of_chunks
    for (i=0; i<number_of_chunks; i++) {
        SPN_chunk_start[i]
    }
}

```

FIG. 28

A

```

00001.clpi {
    number_of_source_packets1
    EP_map
    chunk_map() {
        n+1,
        0, a1, a2, ..., an
    }
}

```

B

```

00002.clpi {
    number_of_source_packets2
    EP_map
    chunk_map() {
        n+1,
        0, b1, b2, ..., bn
    }
}

```

FIG. 29

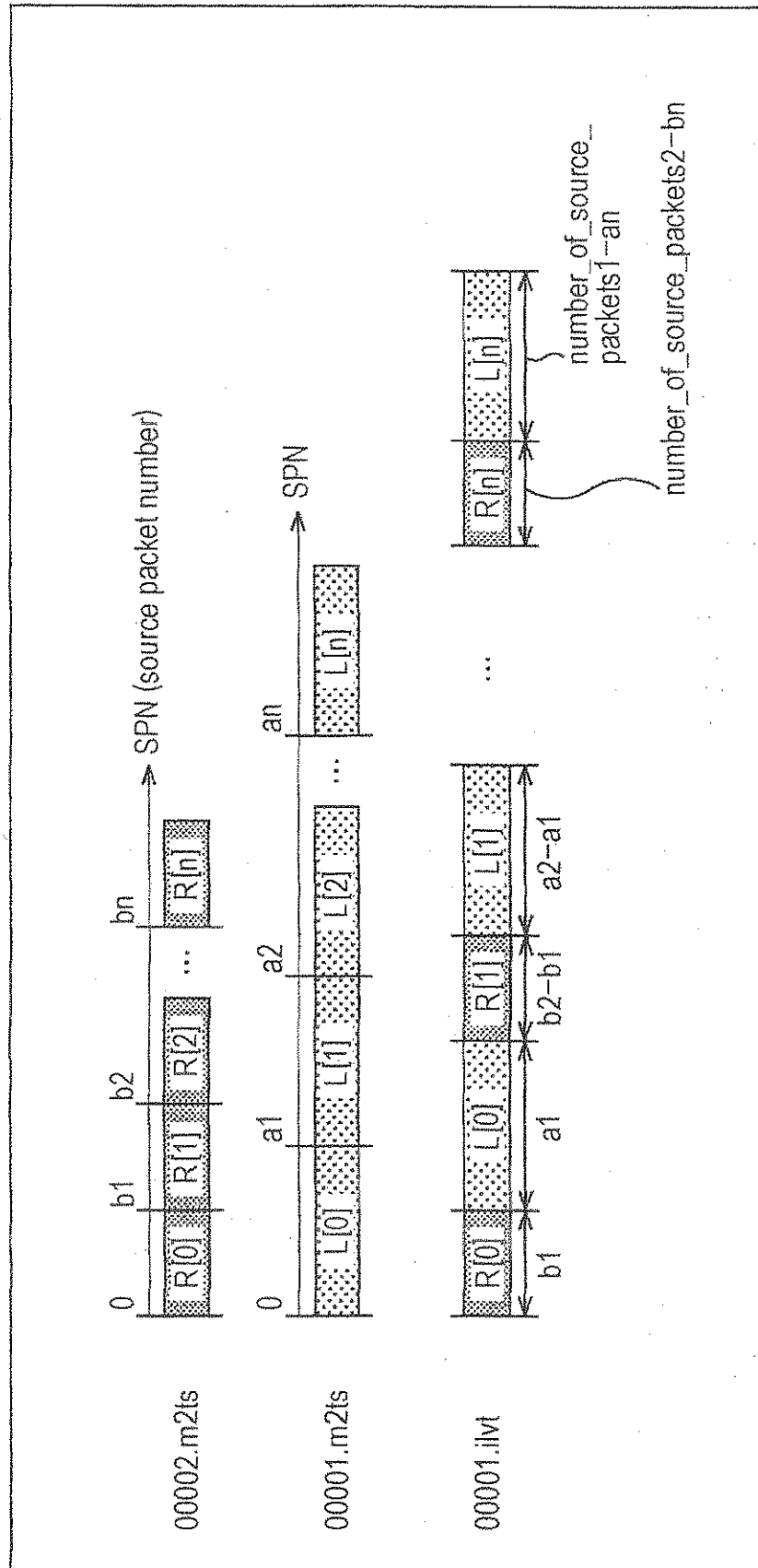


FIG. 30

```
EP_map() {  
    number_of_EP_entries  
    for (i=0; i< number_of_EP_entries; i++) {  
        PTS_EP_start[i]  
        SPN_EP_start[i]  
    }  
}
```

FIG. 31

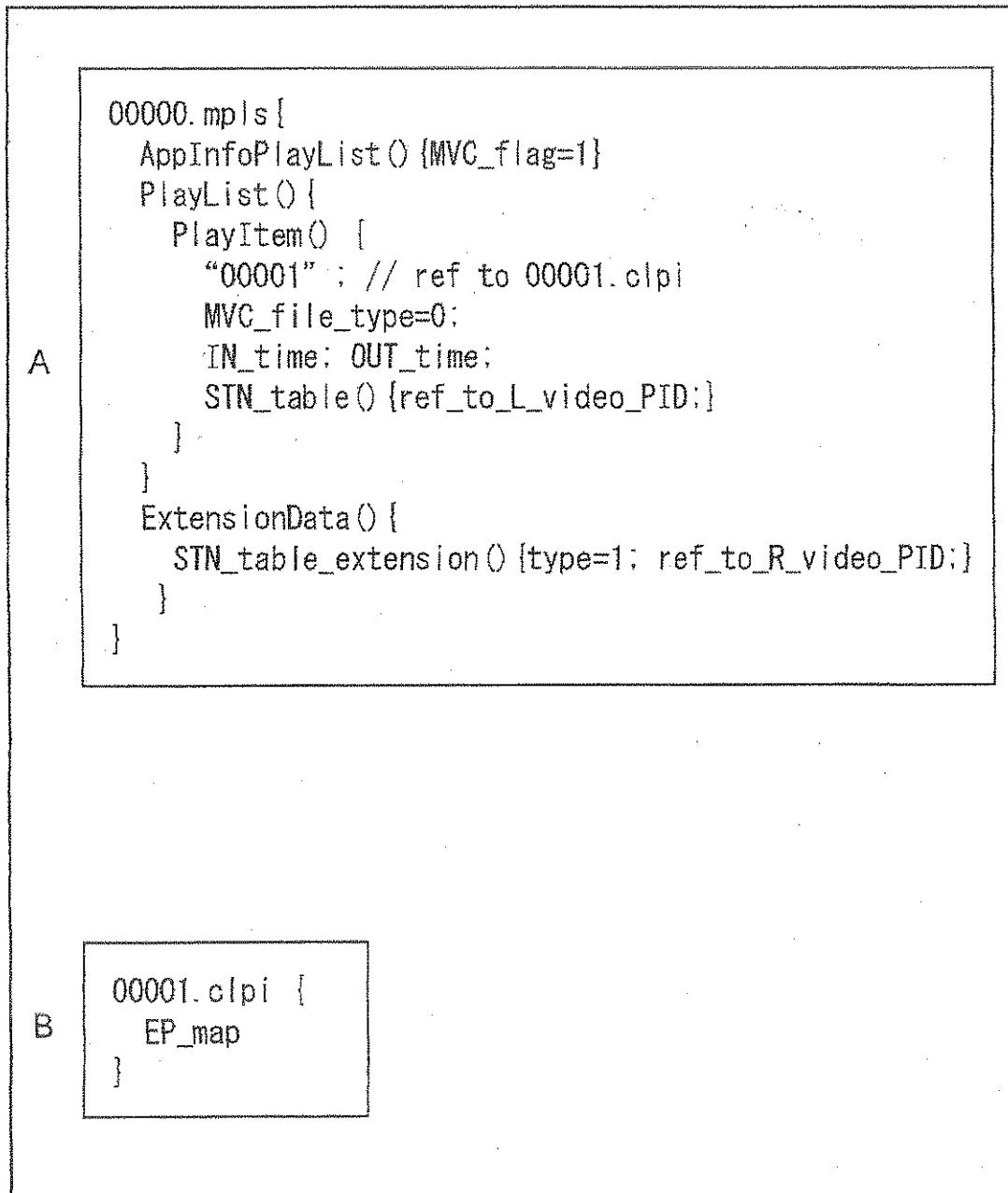


FIG. 32

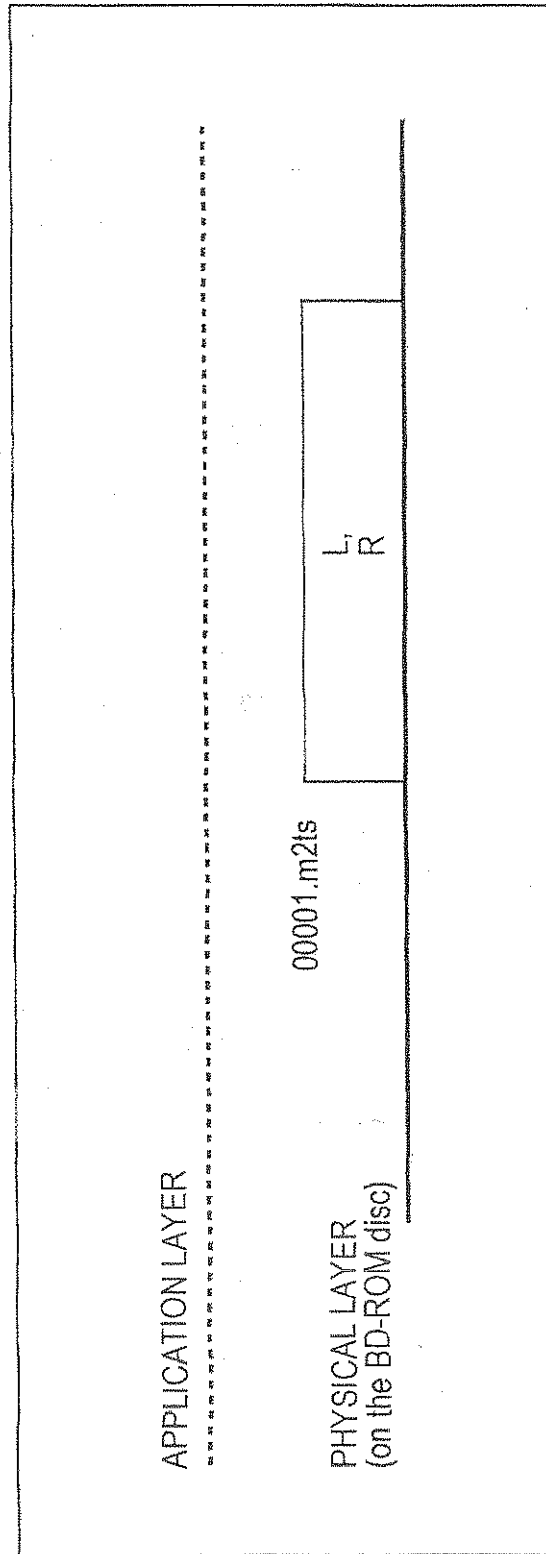


FIG. 33

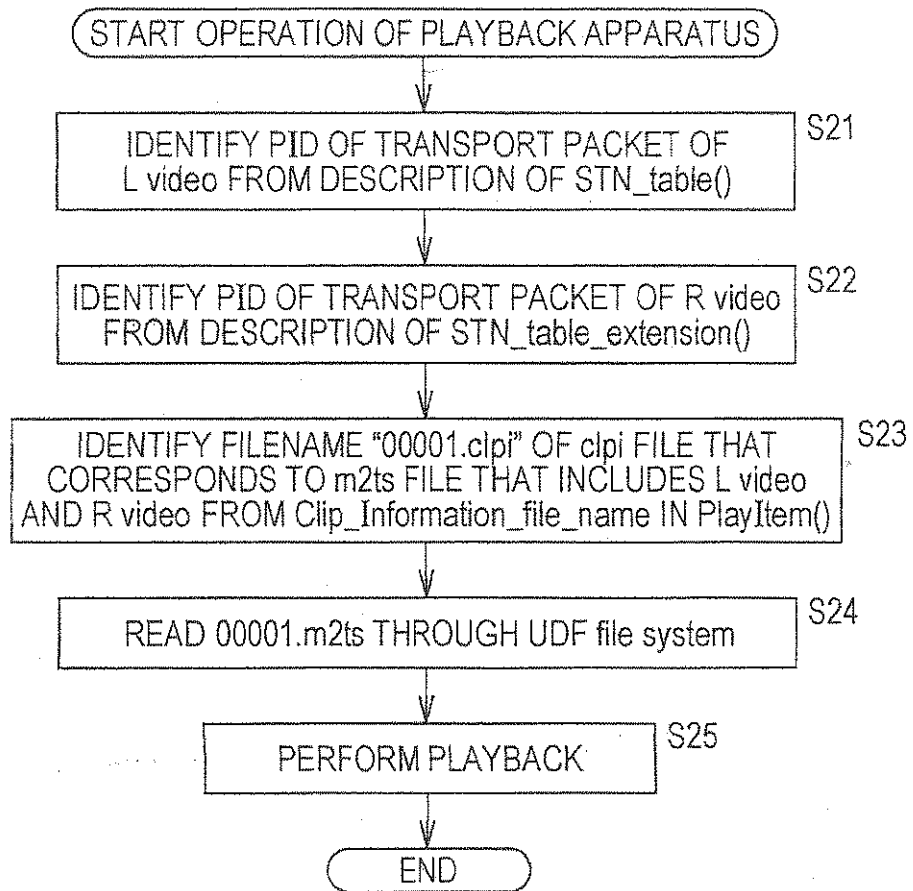


FIG. 34

```

00000.mpls{
  AppInfoPlayList() {MVC_flag=1}
  PlayList(){
    PlayItem() {
      "00001" : // ref to 00001.clpi
      MVC_file_type=2;
      IN_time; OUT_time;
      STN_table() {ref_to_L_video_PID;}
    }
  }
  ExtensionData() {
    SubPath_entries_extension() {
      SubPath_extension() { //SubPath_id=0 IS ASSIGNED TO THIS SubPath
        SubPath_type_extension=8;
        SubPlayItem() {
          "00003" : // ref to 00003.clpi
          SubPlayItem IN_time; SubPlayItem OUT_time;
        }
      }
    }
    STN_table_extension() {type=2; ref_to_SubPath_id=0
    ref_to_R_video_PID;}
  }
}

```

FIG. 35

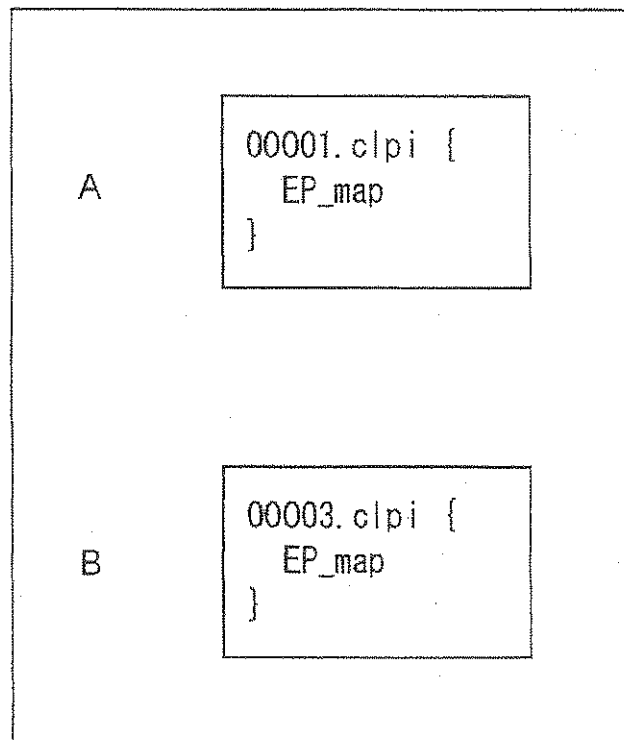


FIG. 36

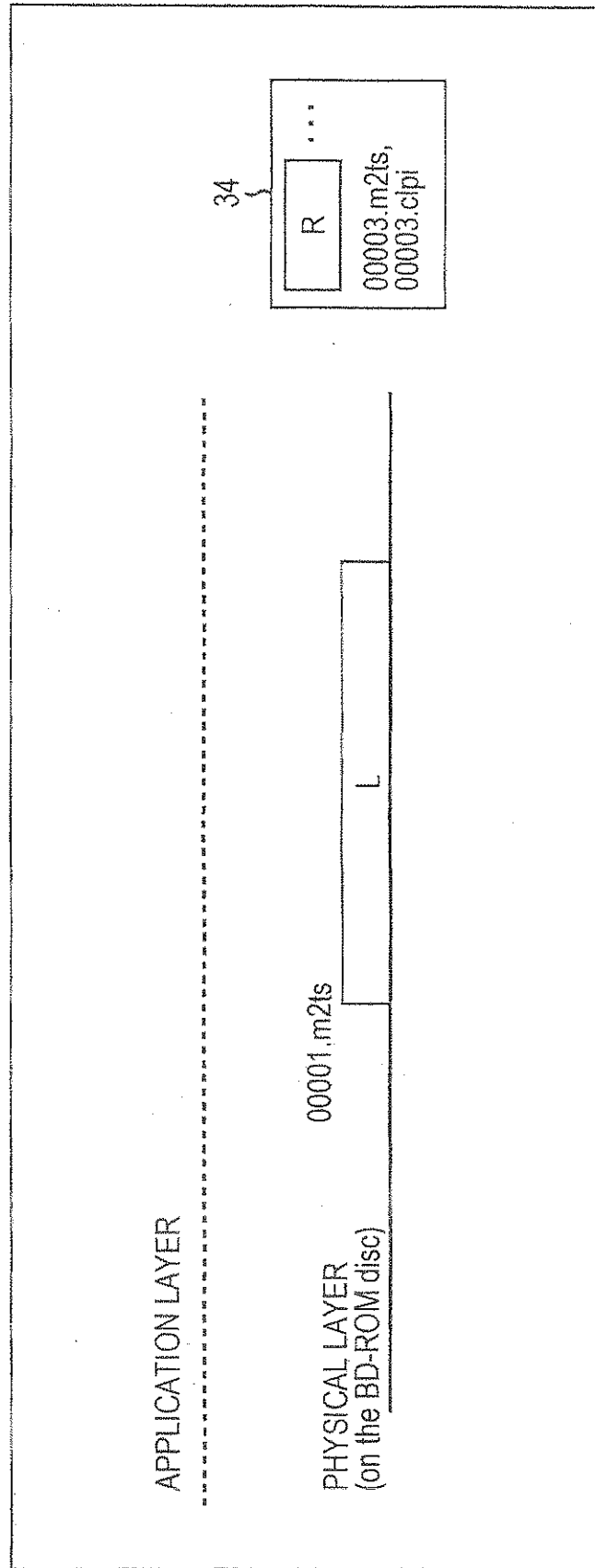


FIG. 37

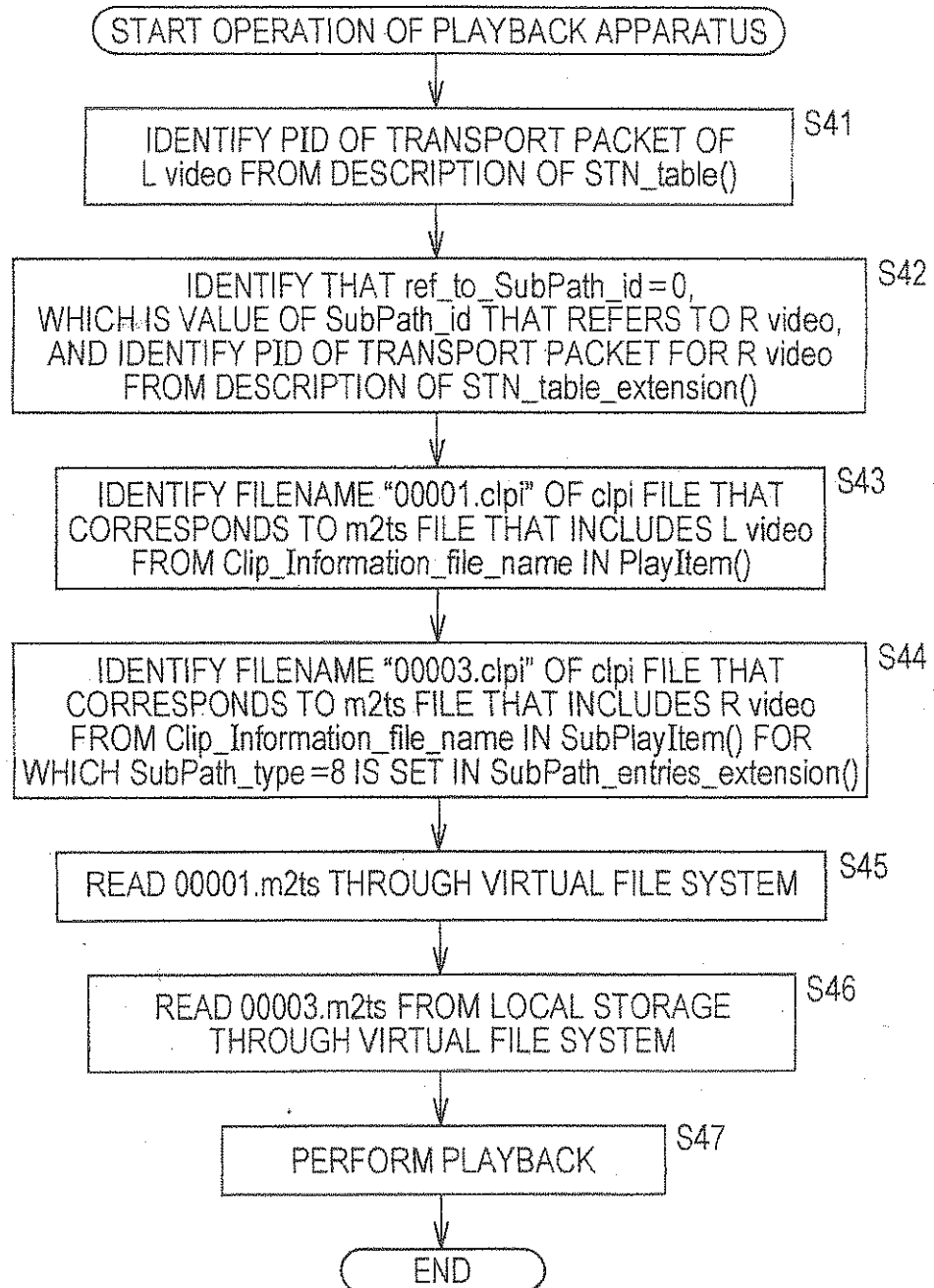


FIG. 38

```

00000.mpls[
  AppInfoPlaylist() {MVC_flag=1}
  Playlist() {
    PlayItem() {
      "00001" ; // ref to 00001.clpi
      MVC_file_type=1;
      IN_time; OUT_time;
      STN_table() {ref_to_L_video_PID;}
    }
  }
  ExtensionData() {
    SubPath_entries_extension() {
      SubPath() {
        SubPath_type=8;
        SubPlayItem() {
          "00002" ; // ref to 00002.clpi
          IN_time; OUT_time;
        }
      }
    }
    STN_table_extension() {type=2; ref_to_SubPath_id;
    ref_to_R_video_PID;}
  }
}

```

FIG. 39

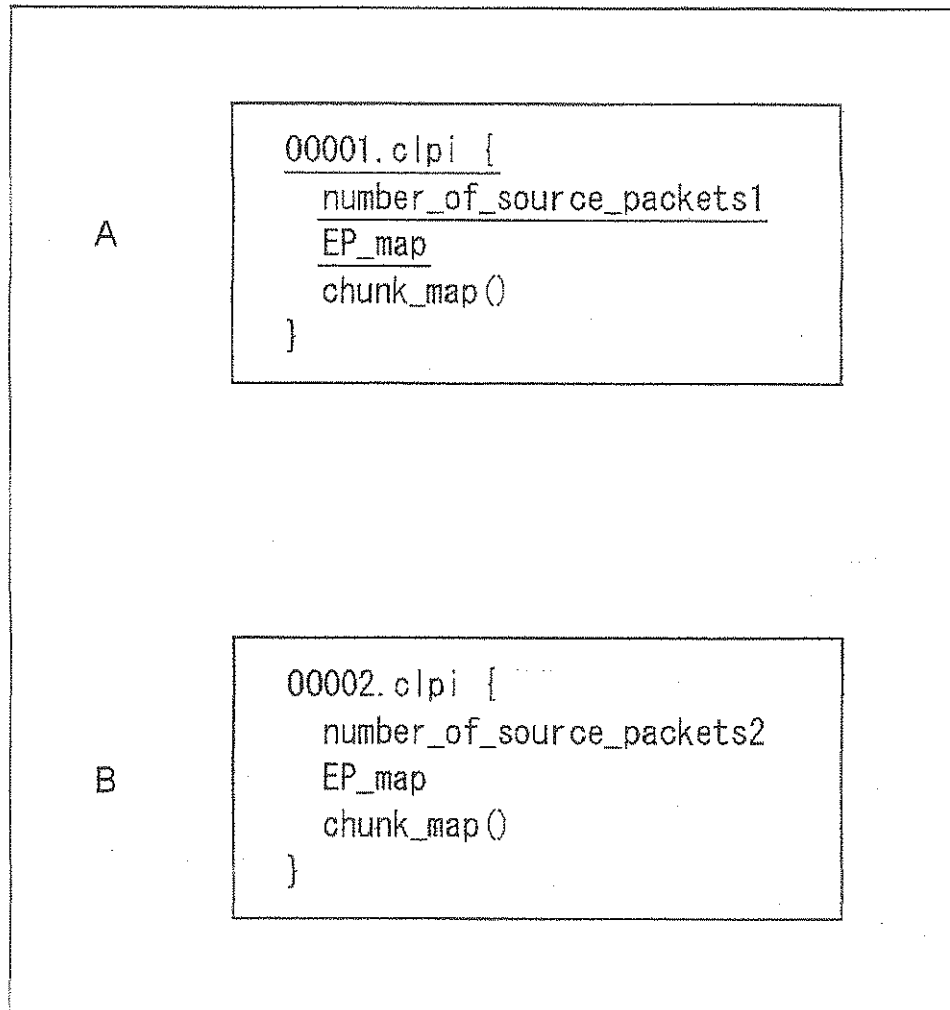


FIG. 40

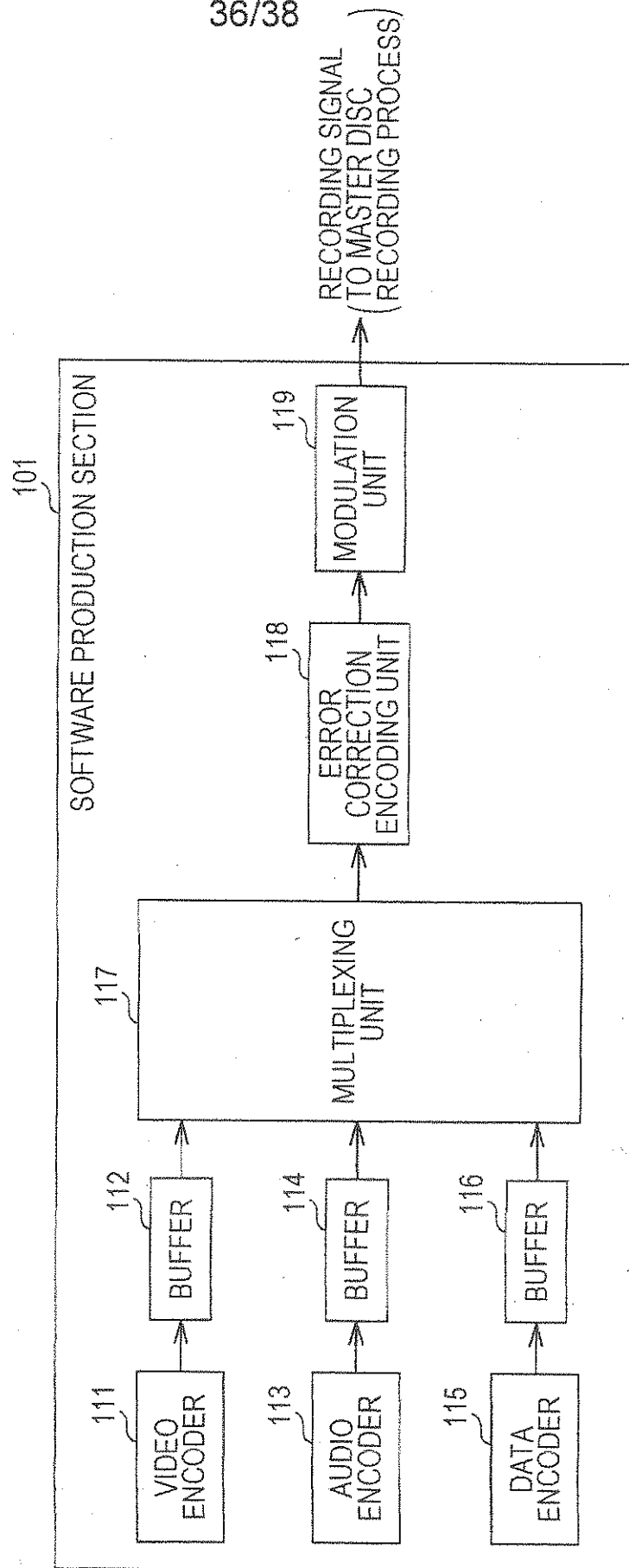


FIG. 41

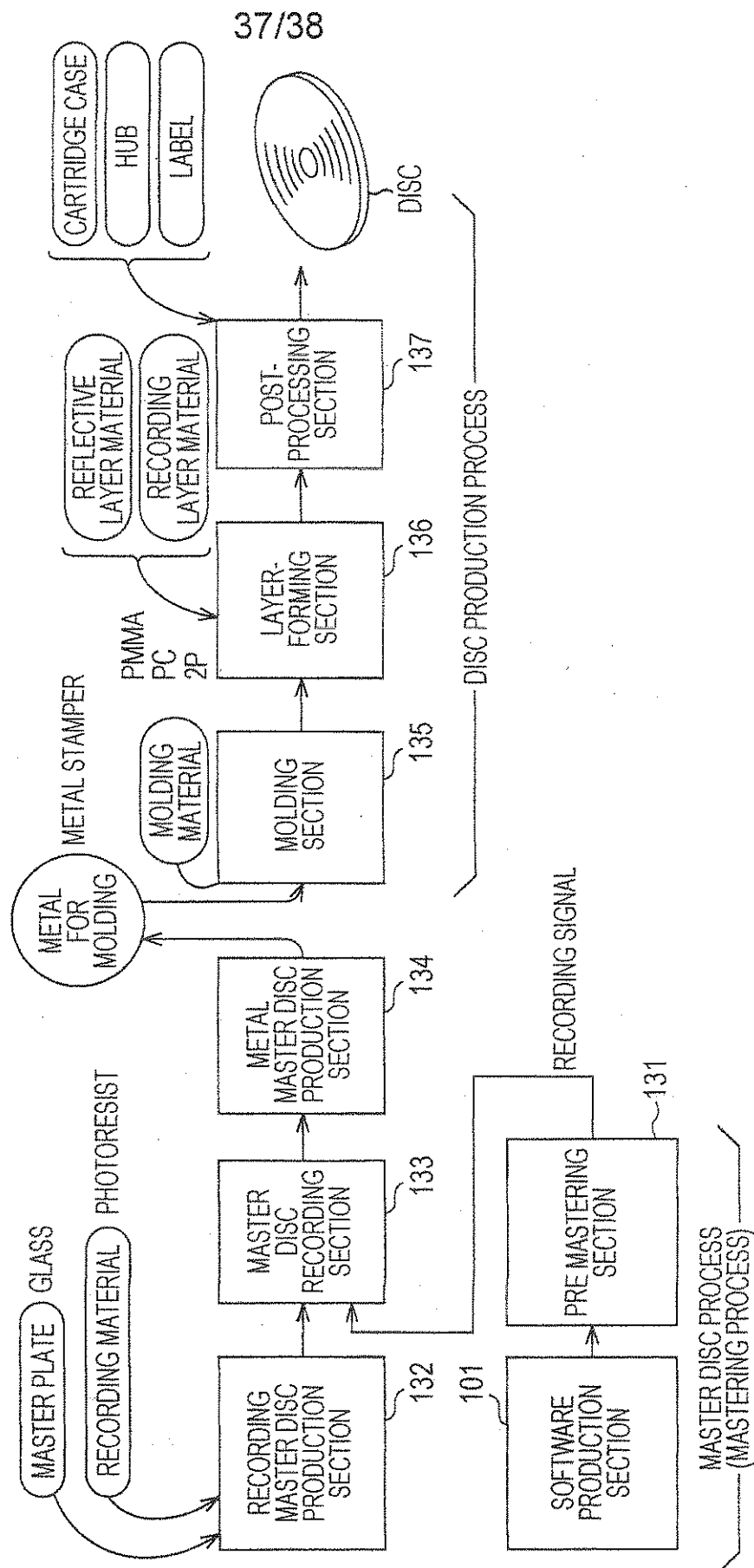


FIG. 42

