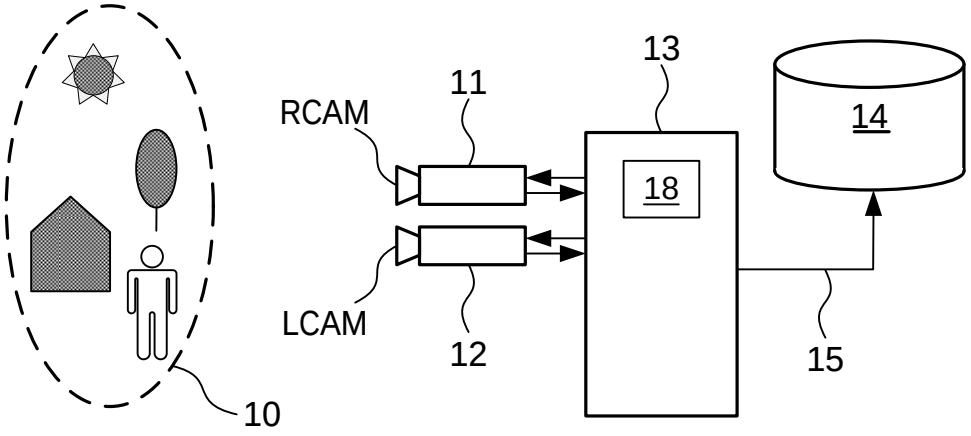


FIG. 1

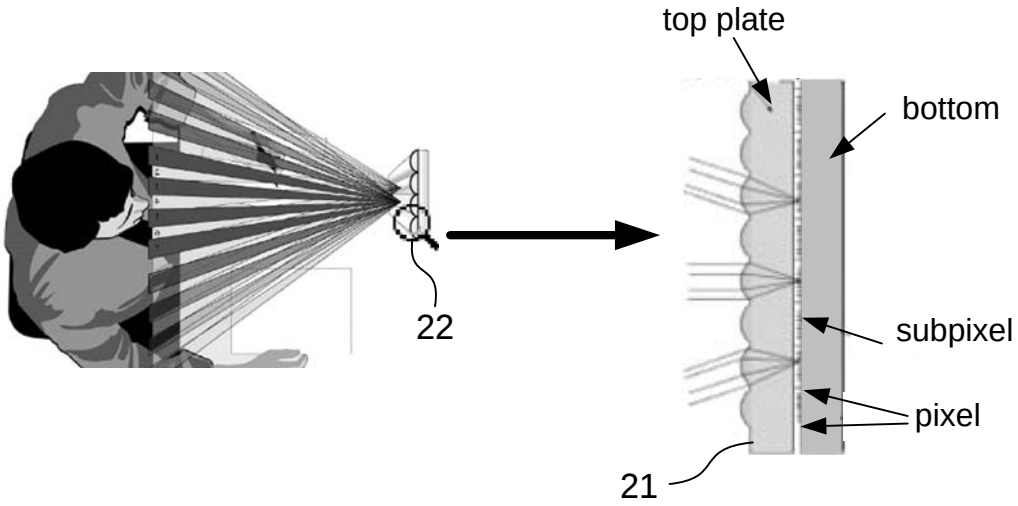


FIG. 2

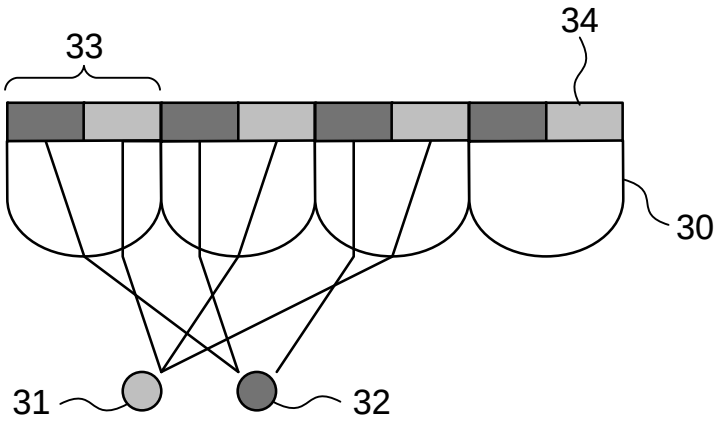


FIG. 3

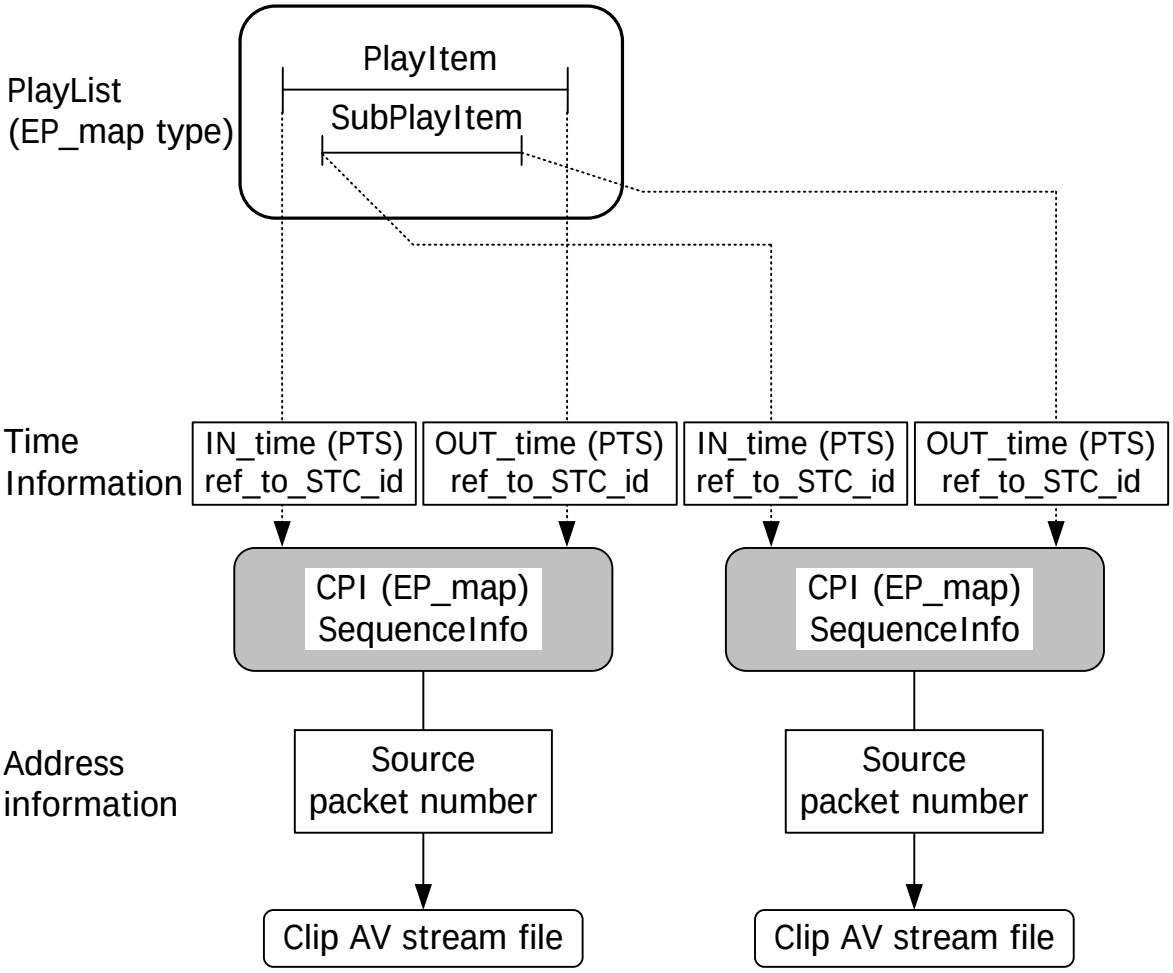

PlayList
(EP_map type)
PlayItem
SubPlayItem
Time
Information
IN_time (PTS)
ref_to_STC_id
OUT_time (PTS)
ref_to_STC_id
IN_time (PTS)
ref_to_STC_id
OUT_time (PTS)
ref_to_STC_id
CPI (EP_map)
SequenceInfo
CPI (EP_map)
SequenceInfo
Address
information
Source
packet number
Source
packet number
Clip AV stream file
Clip AV stream file

FIG. 4

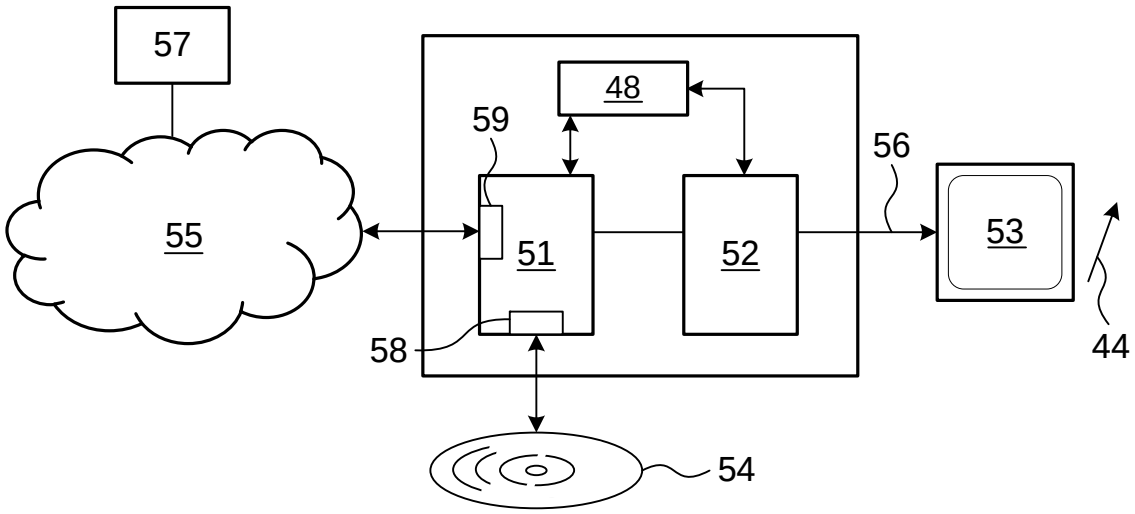

57
55
48
59
51
52
56
53
58
54
44

FIG. 5

CPI_type	Meaning
0	reserved
1	EP_map type
2	reserved
3 - 15	reserved

FIG. 6

CPI_type	Meaning
0	reserved
1	EP_map type
2	**EP_map_MVC**
3 - 15	reserved

FIG. 7

EP_stream_type	Meaning
0	reserved
1	**Video type 1**
2	reserved
3	**audio**
4 -7	reserved
8	**MVC_stream_type**

FIG. 8

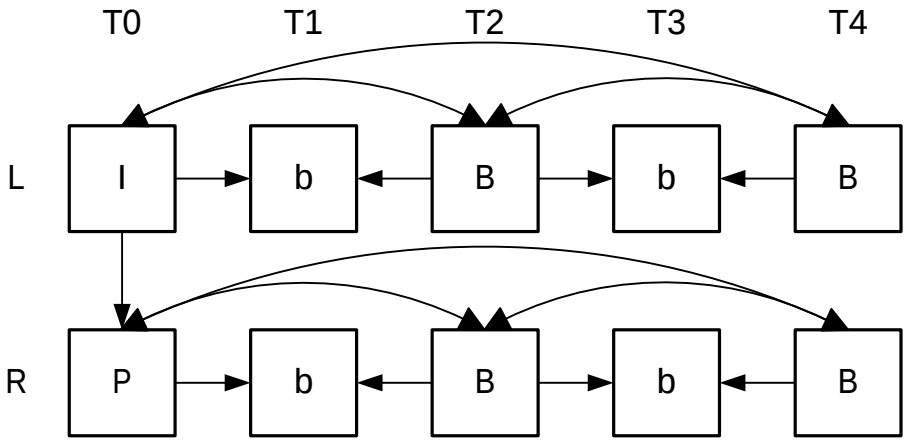


FIG. 9

Syntax	No. of bits	Mnemonic
EP_map(){		
reserved_for_word_align	8	bslbf
Number_of_stream_PID_entries	8	uimsbf
for (k=0; k<number_of_stream_PID_entries; k+ +) {		
stream_PID *[k]*	16	bslbf
reserved_for_word_align	10	bslbf
EP_stream_type *[k]*	4	bslbf
number_of_EP_coarse_entries *[k]*	16	uimsbf
number_of_EP_fine_entries *[k]*	18	uimsbf
EP_map_for_one_stream_PID_start_address *[k]*	32	uimsbf
}		
for (i=0; i<X; i++) {		
padding_word	16	bslbf
}		
for (k=0; k<number_of_stream_PID_entries; K++) {		
EP_map_for_one_stream_PID(EP_stream_type*[k]*, number_of_EP_coarse_entries*[k]*, number_of_EP_fine_entries*[k]*)		
for (i=0; i<Y[k]; i++) {		
padding_word		bslbf
}		
}		
}		

FIG. 10

Syntax	No. of bits	Mnemonic
EP_map_for_one_stream_PID(EP_stream_type, Nc,Nf) {		
EP_fine_table_strart_address	32	uimsbf
for (i=0; i<Nc; i++) {		
Ref_to_EP_fine_id*[i]*	18	uimsbf
PTS_EP_coarse*[i]*	14	uimsbf
SPN_EP_coarse*[i]*	32	uimsbf
MVC_ENTRY {		
TEMPORAL_VECTOR[i]{		
PTS_EP_coarse*[i]*		
SPN_EP_coarse*[i]*		
SPATIAL_VECTOR[i]{		
PTS_EP_coarse*[i]*		
SPN_EP_coarse*[i]*		
}		
}		
for (i=0; i<X; i++) {		
Padding_word	16	bslbf
}		

for (EP_fine_id = 0;
EP_fine_id < Nf;
EP_fine_id ++) {		

is_angle_change_piont*[EP_fine_id]*	1	bslbf
I_end_position_offset*[EP_fine_id]*	3	bslbf
PTS_EP_fine*[EP_fine_id]*	11	uimsbf
SPN_EP_fine*[EP_fine_id]*	17	uimsbf
MVC_ENTRY {		
TEMPORAL_VECTOR[i]{		
PTS_EP_fine*[EP_fine_id]*		
SPN_EP_fine*[EP_fine_id]*		
SPATIAL_VECTOR[i]{		
PTS_EP_fine*[EP_fine_id]*		
SPN_EP_fine*[EP_fine_id]*		
}		
}		

FIG. 11