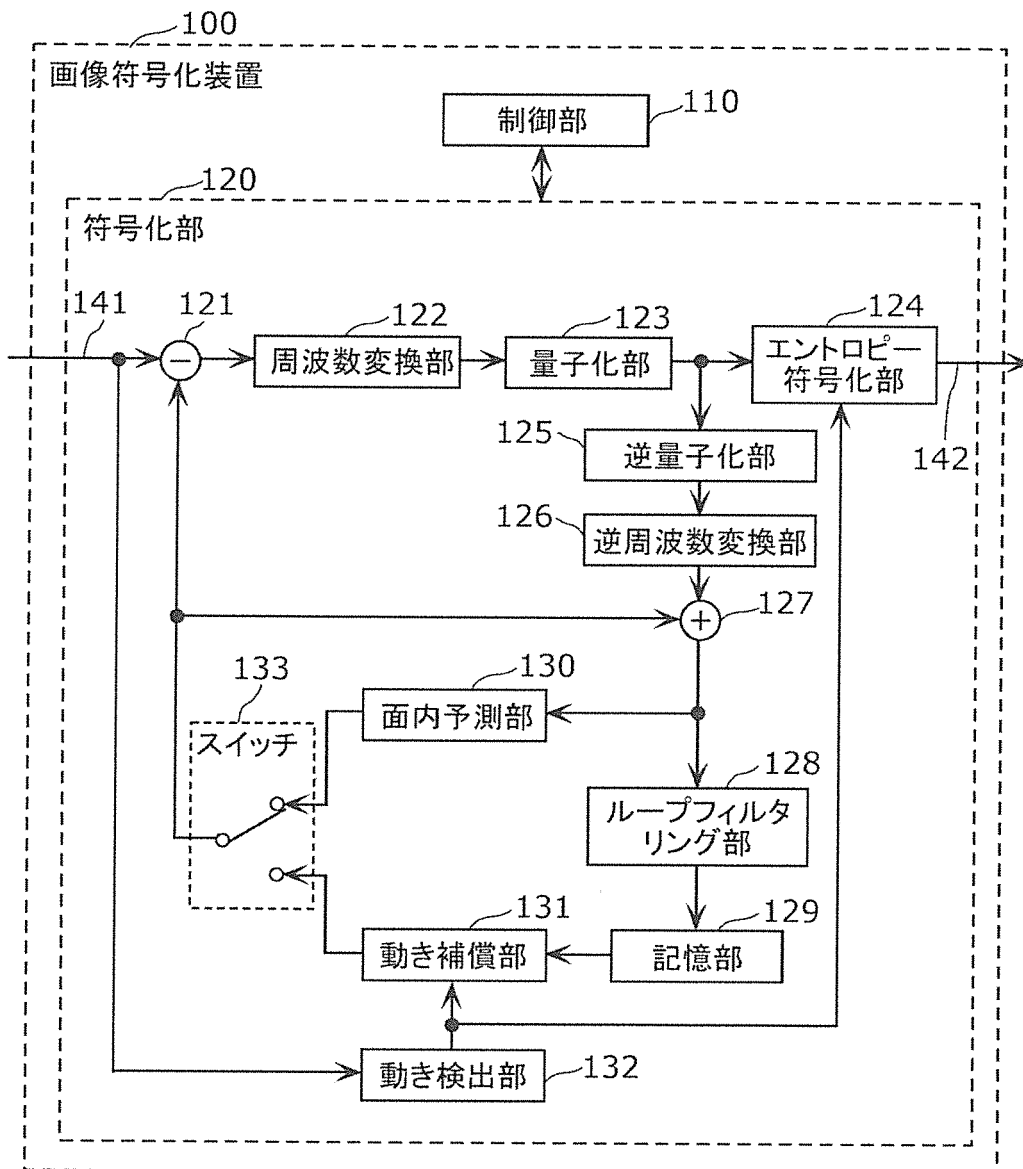
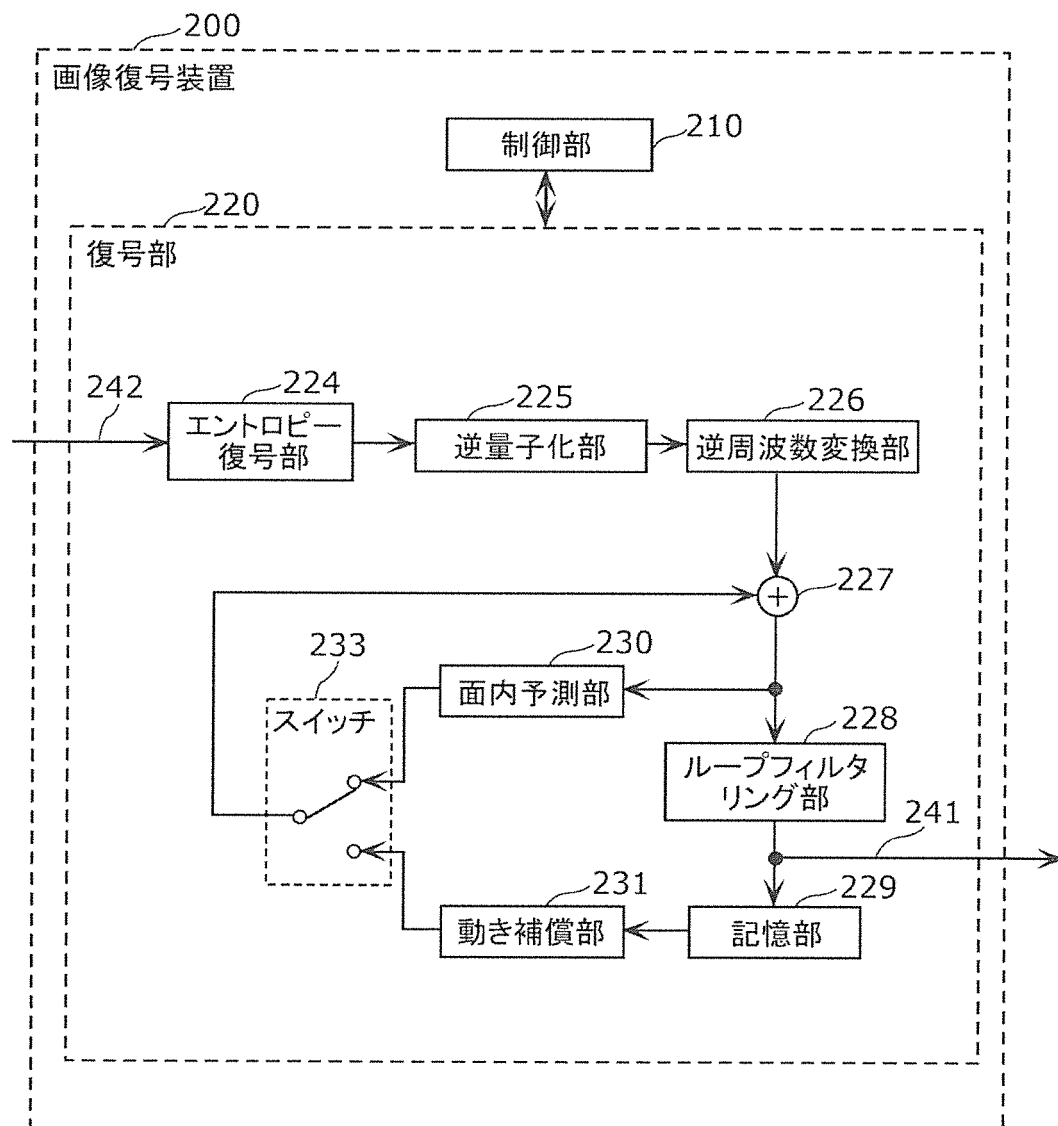


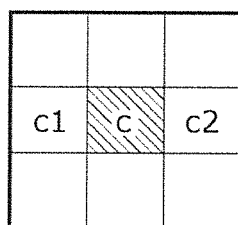
[図1]



[図2]



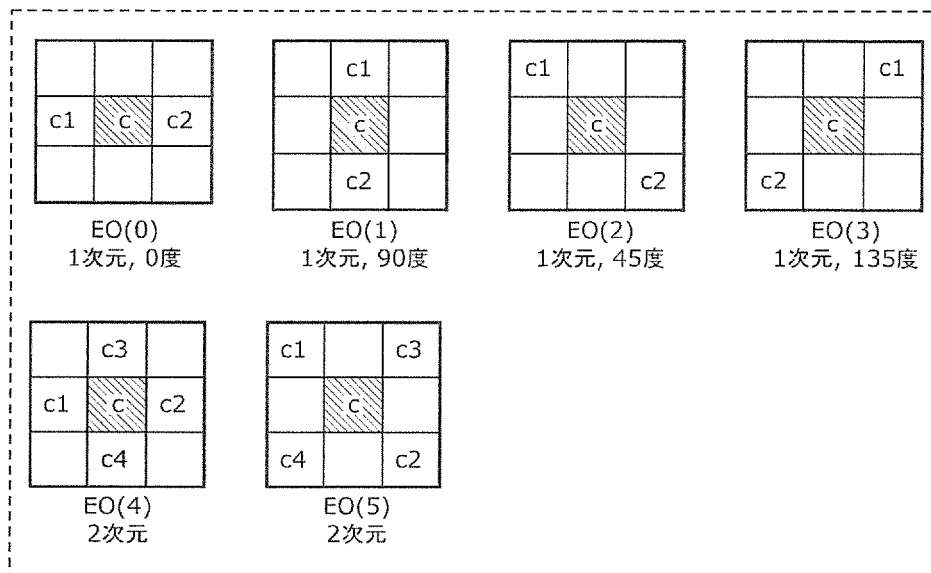
[図3]



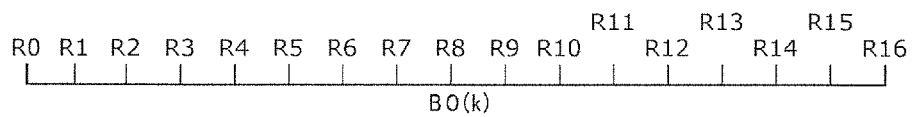
[図4]

クラス	条件	エッジ形状
1	$c < c1 \ \&\& \ c < c2$	
2	$(c < c1 \ \&\& \ c == c2) \ \ (c == c1 \ \&\& \ c < c2)$	
3	$(c > c1 \ \&\& \ c == c2) \ \ (c == c1 \ \&\& \ c > c2)$	
4	$c > c1 \ \&\& \ c > c2$	
0	上記のどれでもない	

[図5]



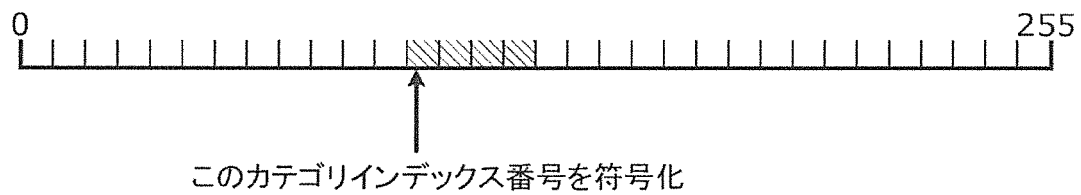
[図6]



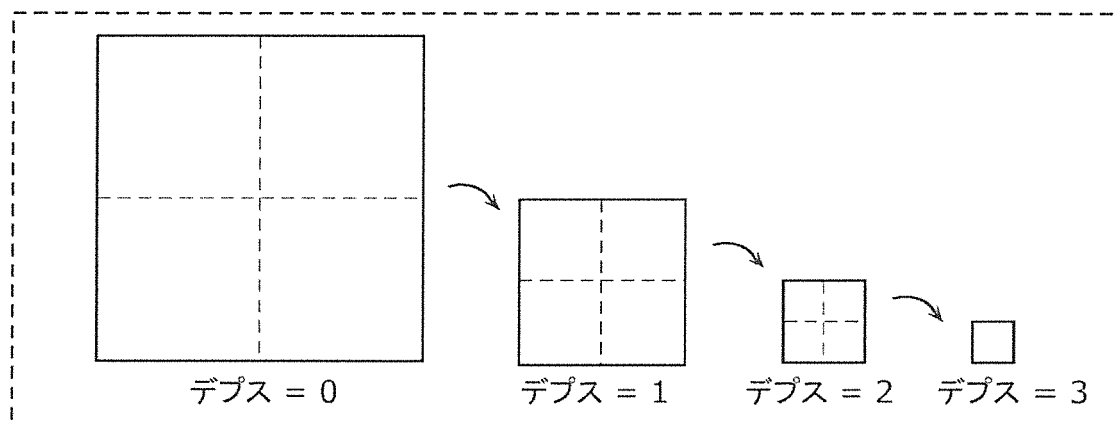
[図7]

クラス	条件	クラス	条件
1	$R0 \leq c < R1$	9	$R8 \leq c < R9$
2	$R1 \leq c < R2$	10	$R9 \leq c < R10$
3	$R2 \leq c < R3$	11	$R10 \leq c < R11$
4	$R3 \leq c < R4$	12	$R11 \leq c < R12$
5	$R4 \leq c < R5$	13	$R12 \leq c < R13$
6	$R5 \leq c < R6$	14	$R13 \leq c < R14$
7	$R6 \leq c < R7$	15	$R14 \leq c < R15$
8	$R7 \leq c < R8$	16	$R15 \leq c \leq R16$

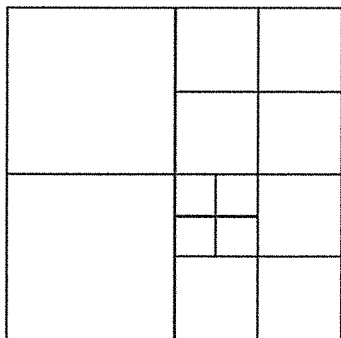
[図8]



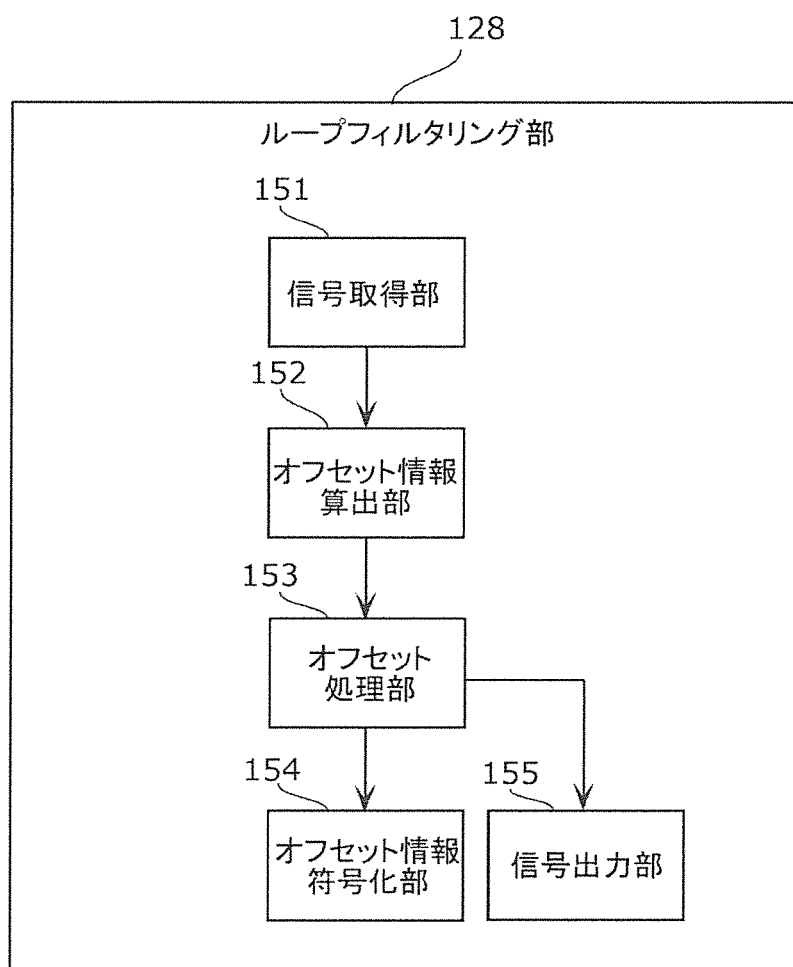
[図9A]



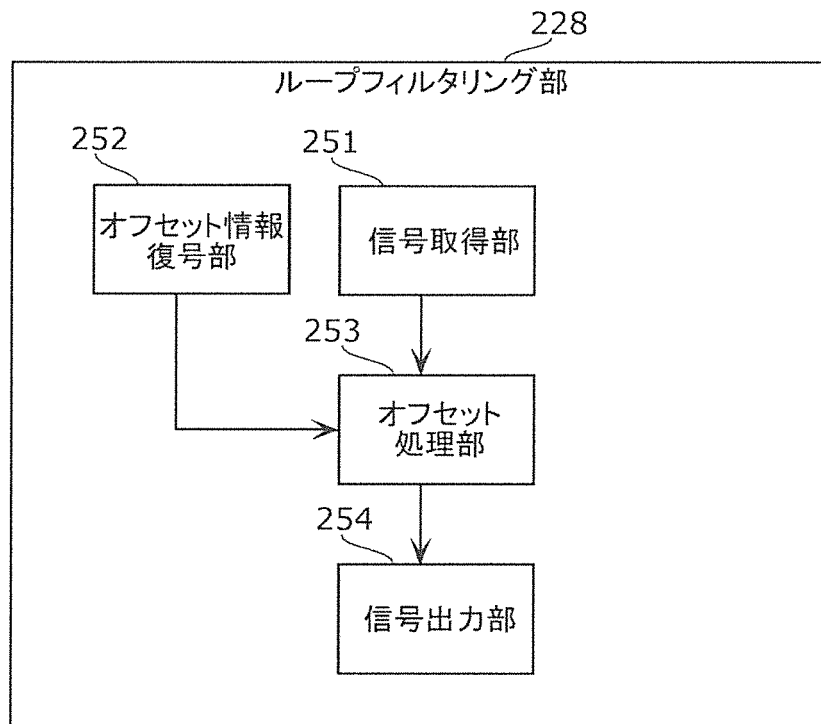
[図9B]



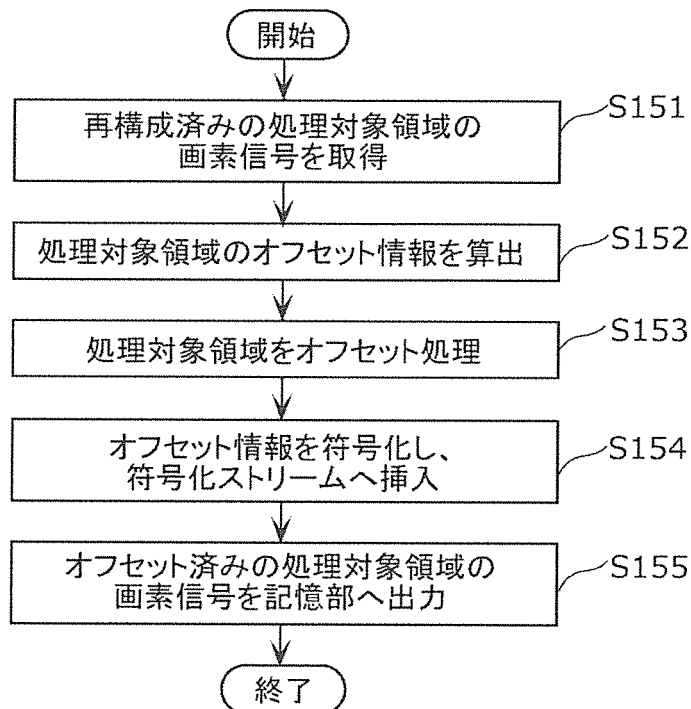
[図10]



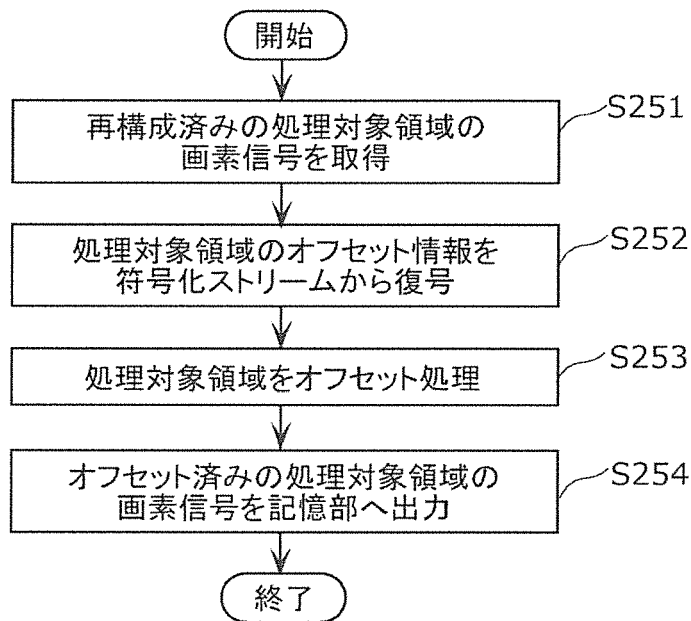
[図11]



[図12]



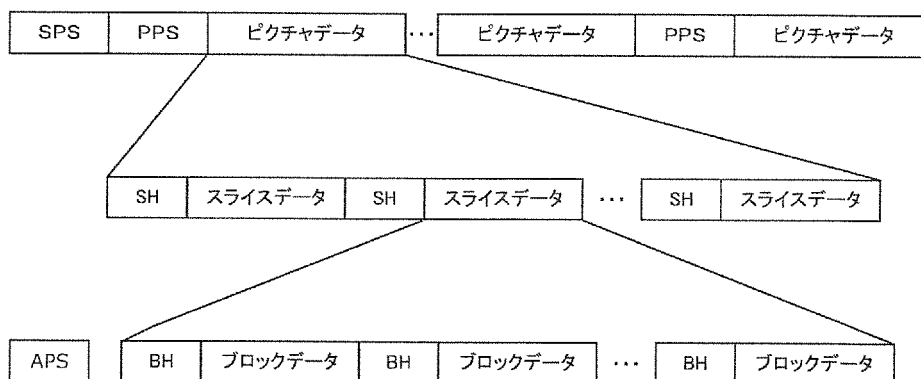
[図13]



[図14]

インデックス	ビット割り当て	画素分類方法
0	0	オフセット処理しない
1	10	エッジオフセット(0)
2	110	エッジオフセット(1)
3	1110	エッジオフセット(2)
4	11110	エッジオフセット(3)
5	11111	バンドオフセット

[図15]



[図16A]

aps_sao_param(){	Descriptor
sao_cb_enable_flag	u(1)
sao_cr_enable_flag	u(1)
sao_num_lcu_in_width_minus1	ue(v)
sao_num_lcu_in_height_minus1	ue(v)
sao_one_luma_unit_flag	u(1)
if(sao_one_luma_unit_flag)	
sao_offset_vlc(0, 0, 0)	
if(sao_cb_enable_flag){	
sao_one_cb_unit_flag	u(1)
if(sao_one_cb_unit_flag)	
sao_offset_vlc(0, 0, 1)	
}	
if(sao_cr_enable_flag){	
sao_one_cr_unit_flag	u(1)
if(sao_one_cr_unit_flag)	
sao_offset_vlc(0, 0, 2)	
}	
for(ry = 0; ry <= sao_num_lcu_in_height_minus1; ry++){	
for(rx = 0; rx <= sao_num_lcu_in_width_minus1; rx++){	
if(aps_sample_adaptive_offset_flag && !sao_one_luma_unit_flag){	
if(ry > 0 && rx == 0)	
sao_repeat_row_flag[0]	u(1)
sao_unit_vlc(rx, ry, 0)	
}	
if(sao_cb_enable_flag && !sao_one_cb_unit_flag){	
if(ry > 0 && rx == 0)	
sao_repeat_row_flag[1]	u(1)
sao_unit_vlc(rx, ry, 1)	
}	
if(sao_cr_enable_flag && !sao_one_cr_unit_flag){	
if(ry > 0 && rx == 0)	
sao_repeat_row_flag[2]	u(1)
sao_unit_vlc(rx, ry, 2)	
}	
}	
}	
}	

[図16B]

sao_unit_vlc(rx, ry, cIdx){	Descriptor
if(!sao_repeat_row_flag[cIdx]){	
if(rx == 0 run[cIdx][rx][ry] < 0)	
if(ry == 0){	
sao_run_diff	u(v)
saoRun[cIdx][rx][ry] = sao_run_diff	
} else {	
sao_run_diff	se(v)
saoRun[cIdx][rx][ry] = sao_run_diff + saoRun[cIdx][rx][ry - 1]	
}	
saoRun[cIdx][rx + 1][ry] = saoRun[cIdx][rx][ry] - 1	
if(rx == 0 saoRun[cIdx][rx][ry] < 0)	
if(ry > 0)	
sao_merge_up_flag	u(1)
if(!sao_merge_up_flag)	
sao_offset_vlc(rx, ry, cIdx)	
} else	
saoRun[cIdx][rx][ry] = saoRun[cIdx][rx][ry - 1]	
}	

[図16C]

sao_offset_vlc(rx, ry, cIdx){	Descriptor
sao_type_idx [cIdx][rx][ry]	ue(v)
if(sao_type_idx[cIdx][rx][ry] == 5){	
sao_band_position [cIdx][rx][ry]	u(5)
for(i = 0; i < 4; i++)	
sao_offset [cIdx][rx][ry][i]	se(v)
} else if(sao_type_idx[cIdx][rx][ry] != 0)	
for(i = 0; i < 4; i++)	
sao_offset [cIdx][rx][ry][i]	ue(v)
}	

[図17A]

slice_data(){	Descriptor
CtbAddrRS = SliceCtbAddrRS	
CtbAddrTs = CtbAddrRStoTS[CtbAddrRs]	
moreDataFlag = 1	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlagIdx = -1	
do {	
xCtb = InverseRasterScan(CtbAddrRS, CtbSize, CtbSize,	
pic_width_in_luma_samples, 0)	
yCtb = InverseRasterScan(CtbAddrRS, CtbSize, CtbSize,	
pic_width_in_luma_samples, 1)	
NumPCMBBlock = 0	
CtbAddrInSlice = CtbAddrRS - (slice_address >> SliceGranularity)	
AddrUp = CtbAddrRS - PicWidthInCtbs	
if(slice_sao_interleaving_flag){	
if(slice_sample_adaptive_offset_flag)	
sao_unit_cabac(xCtb, yCtb, 0)	
if(sao_cb_enable_flag)	
sao_unit_cabac(xCtb, yCtb, 1)	
if(sao_cr_enable_flag)	
sao_unit_cabac(xCtb, yCtb, 2)	
}	
moreDataFlag = coding_tree(xCtb, yCtb, Log2CtbSize, 0)	
CtbAddrTS++	
if(moreDataFlag && ((tiles_or_entropy_coding_sync_idc == 1 &&	
TileId[CtbAddrTS] != TileId[CtbAddrTS - 1])	
(tiles_or_entropy_coding_sync_idc == 2 &&	
num_substream_minus1 > 0 &&	
CtbAddrTs/PicWidthInCtbs <= num_substream_minus1 &&	
CtbAddrTS % PicWidthInCtbs == 0))) {	
rbsp_trailing_bits()	
if(nextbits(24) == 0x000002){	
entry_point_marker_two_3bytes	f(24)
tile_idx_minus_1	u(v)
}	
}	
} while(moreDataFlag)	
}	

[図17B]

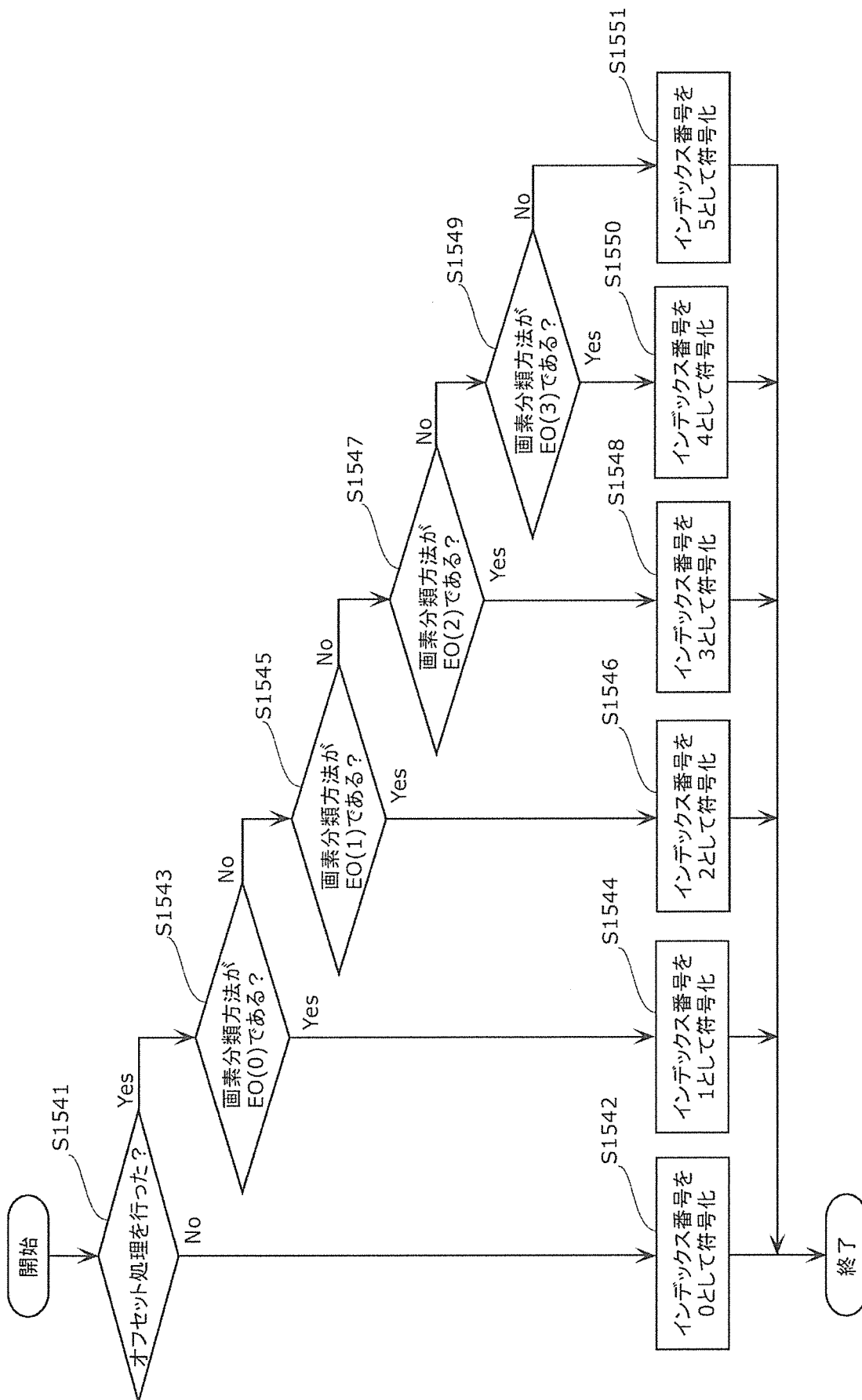
sao_unit_cabac(rx, ry, cIdx){	Descriptor
if(rx > 0) {	
if(CtbAddrInSlice!=0 &&	
TileId[CtbAddrTS] == TileId[CtbAddrRStoTS[CtbAddrRS - 1]])	
sao_merge_left_flag	ae(v)
}	
if(!sao_merge_left_flag) {	
if(ry > 0) {	
if(((CtbAddrTS - CtbAddrRStoTS[CtbAddrRS-PicWidthInCtbs])	
<=CtbAddrInSlice) &&	
(TileId[CtbAddrTS]==	
TileId[CtbAddrRStoTS[CtbAddrRS- PicWidthInCtbs]]))	
sao_merge_up_flag	ae(v)
}	
if(!sao_merge_up_flag)	
sao_offset_cabac(rx, ry, cIdx)	
}	
}	

[図17C]

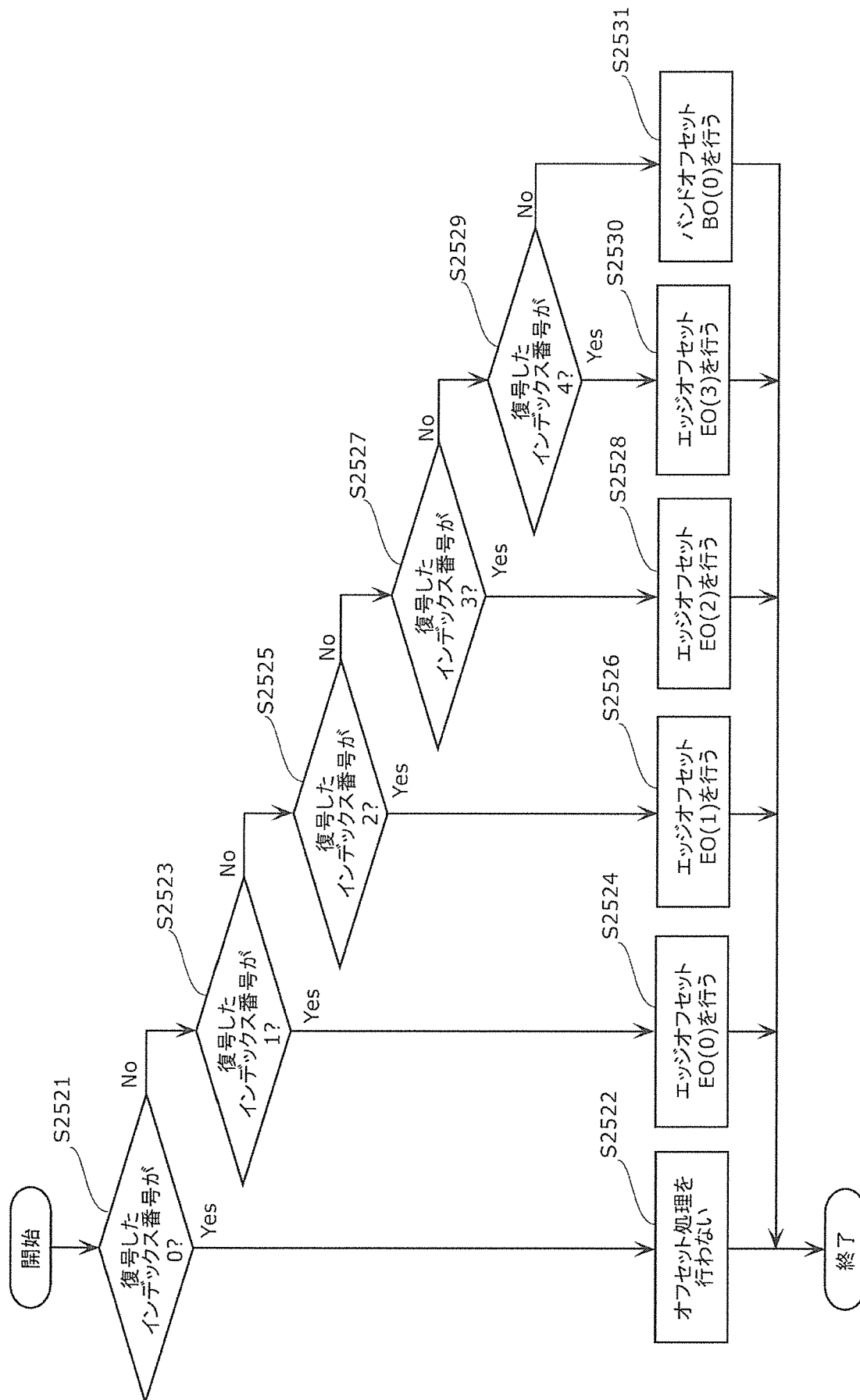
sao_offset_cabac(rx, ry, cIdx) {	Descriptor
sao_type_idx[cIdx][rx][ry]	ae(v)
if(sao_type_idx[cIdx][rx][ry] == 5)	
sao_band_position[cIdx][rx][ry]	ae(v)
if(sao_type_idx[cIdx][rx][ry] != 0) {	
for(i = 0; i < 4; i++)	
sao_offset[cIdx][rx][ry][i]	ae(v)
}	
if(sao_type_idx[cIdx][rx][ry] == 5) {	
for(i = 0; i < 4; i++) {	
if(sao_offset[cIdx][rx][ry] != 0)	
sao_offset_sign[cIdx][rx][ry][i]	ae(v)
}	
}	
}	

[図18]

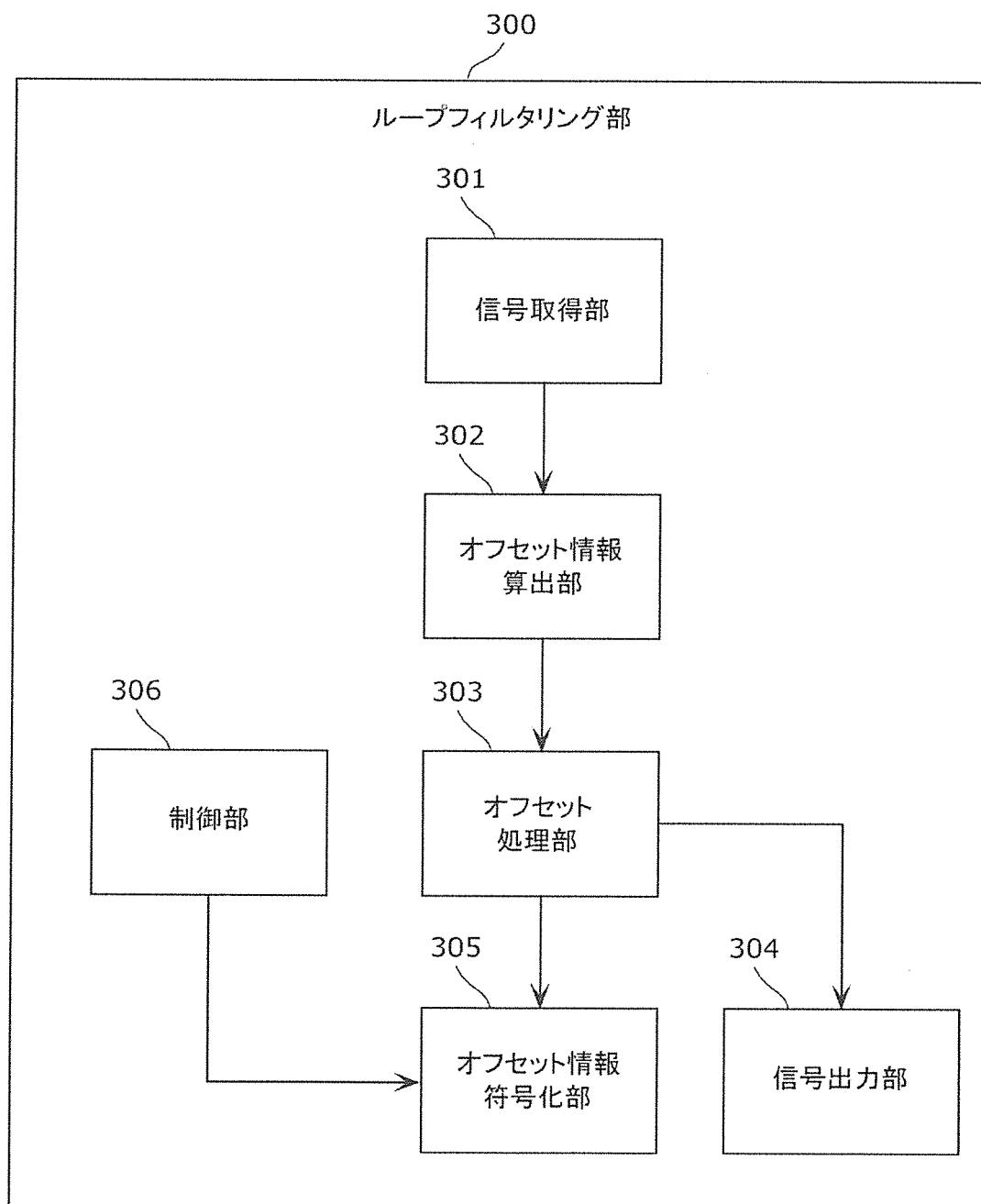
[図19]



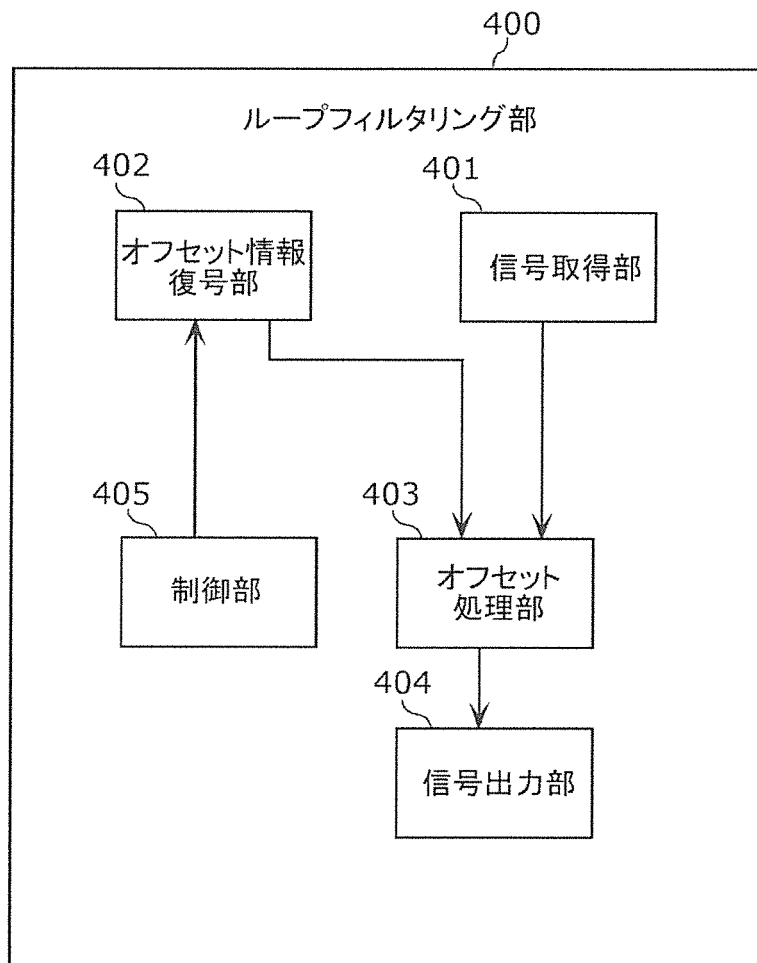
[図20]



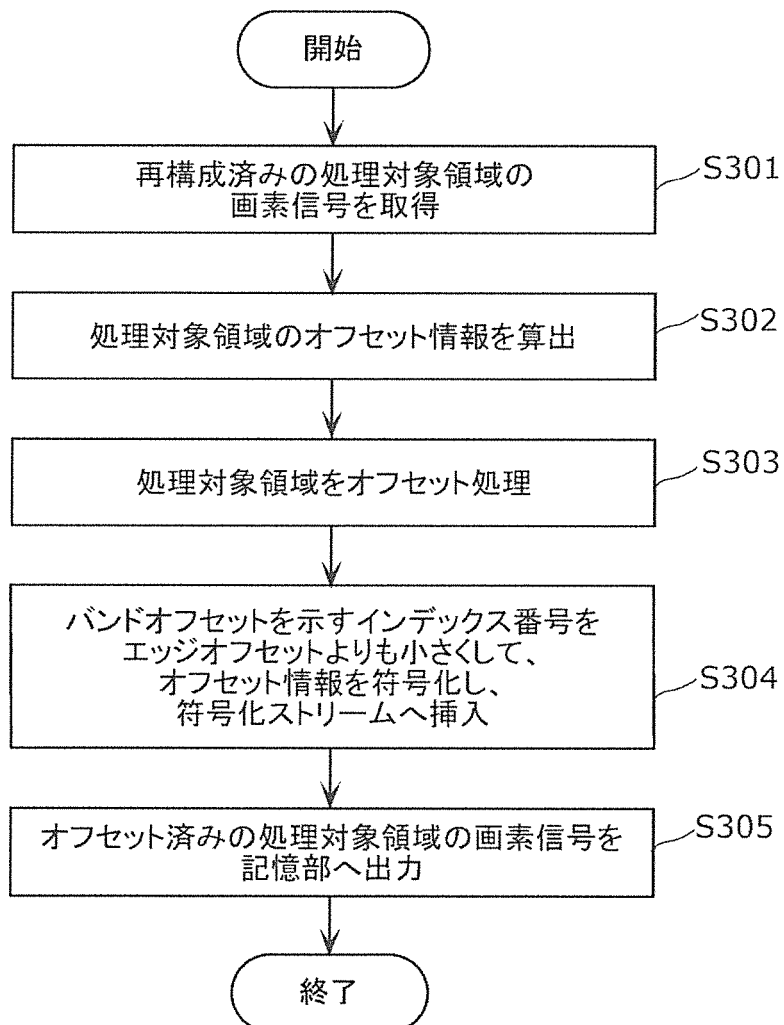
[図21]



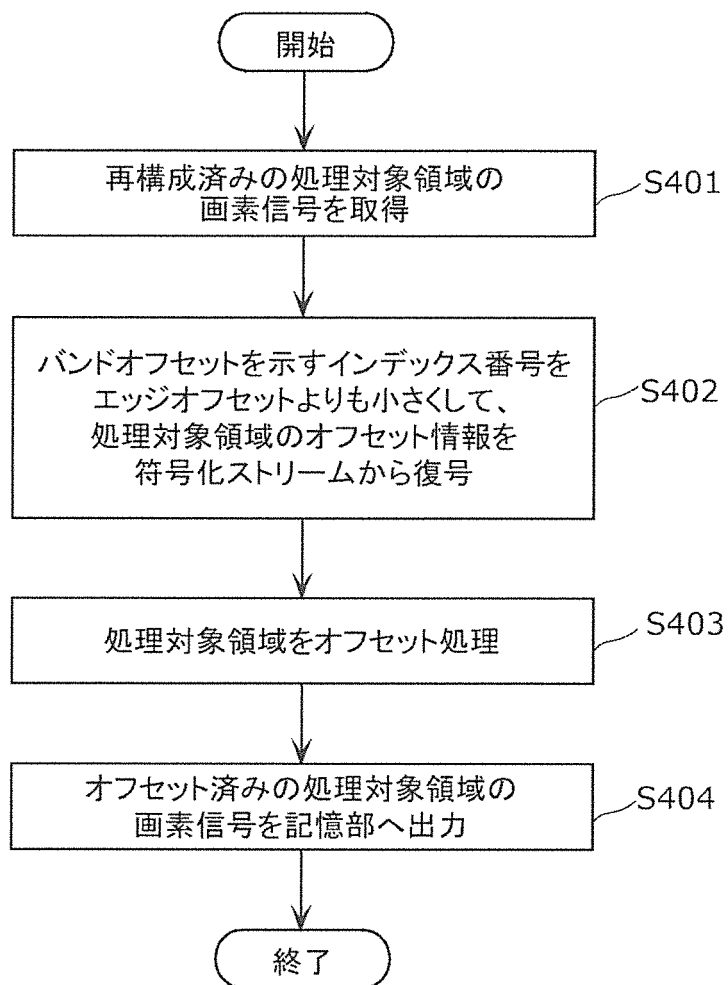
[図22]



[図23]



[図24]



[図25]

インデックス	ビット割り当て	画素分類方法
0	0	オフセット処理しない
1	10	バンドオフセット
2	110	エッジオフセット(0)
3	1110	エッジオフセット(1)
4	11110	エッジオフセット(2)
5	11111	エッジオフセット(3)

[図26]

インデックス	ビット割り当て	オフセット処理フラグ
0	0	オフセット処理しない
1	1	オフセット処理する

インデックス	ビット割り当て	画素分類方法
0	0	バンドオフセット
1	10	エッジオフセット(0)
2	110	エッジオフセット(1)
3	1110	エッジオフセット(2)
4	1111	エッジオフセット(3)

[図27A]

sao_unit_vlc(rx, ry, cIdx) {	Descriptor
if(!sao_repeat_row_flag[cIdx]) {	
if(rx == 0 run[cIdx][rx][ry] < 0)	
if(ry == 0) {	
sao_run_diff	u(v)
saoRun[cIdx][rx][ry] = sao_run_diff	
} else {	
sao_run_diff	se(v)
saoRun[cIdx][rx][ry] = sao_run_diff + saoRun[cIdx][rx][ry - 1]	
}	
saoRun[cIdx][rx + 1][ry] = saoRun[cIdx][rx][ry] - 1	
if(rx == 0 saoRun[cIdx][rx][ry] < 0)	
sao_on_flag オフセット処理フラグ	u(1)
if (sao_on_flag) {	
if (ry > 0)	
sao_merge_up_flag	u(1)
if(!sao_merge_up_flag)	
sao_offset_vlc(rx, ry, cIdx)	
}	
} else	
saoRun[cIdx][rx][ry] = saoRun[cIdx][rx][ry - 1]	
}	

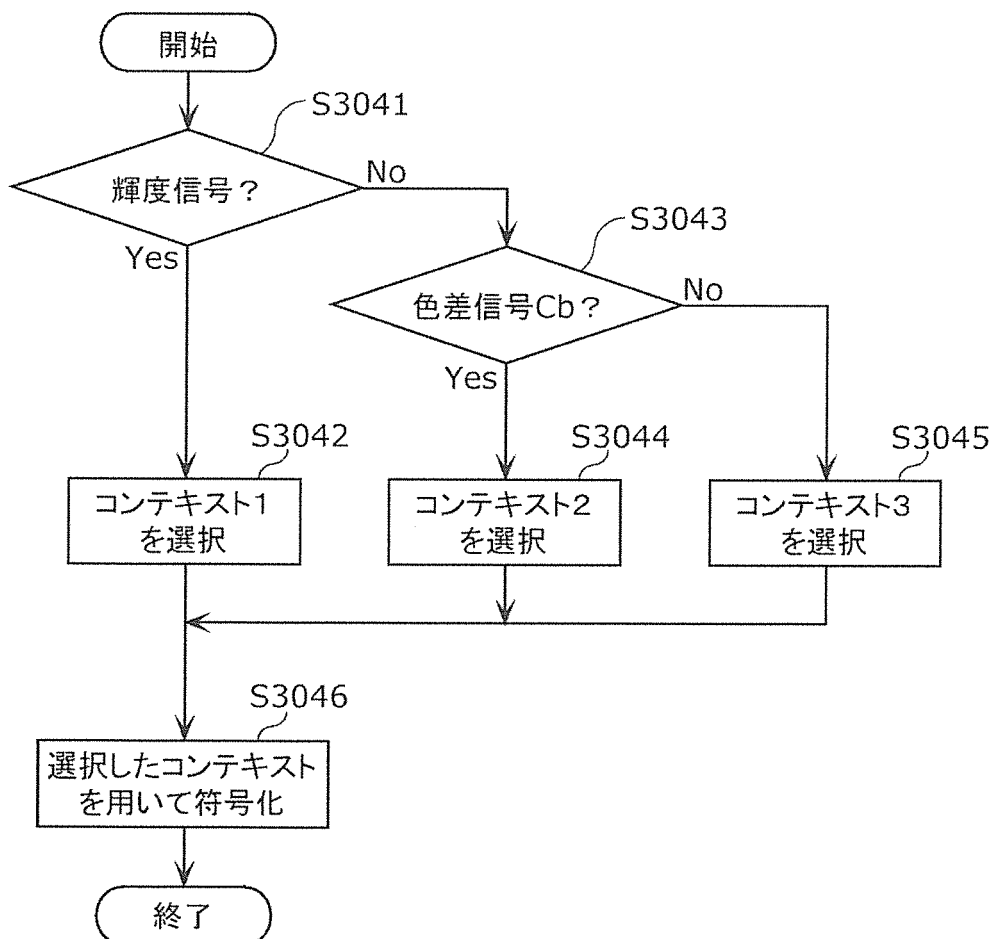
[図27B]

sao_offset_vlc(rx, ry, cIdx) {		
sao_type_idx[cIdx][rx][ry]	画素分類方法 インデックス番号	Descriptor
if(sao_type_idx[cIdx][rx][ry] == 5) {		ue(v)
sao_band_position[cIdx][rx][ry]		u(5)
for(i = 0; i < 4; i++)		
sao_band_offset[cIdx][rx][ry][i]		se(v)
} else		
for(i = 0; i < 4; i++)		
sao_edge_offset[cIdx][rx][ry][i]		ue(v)
}		

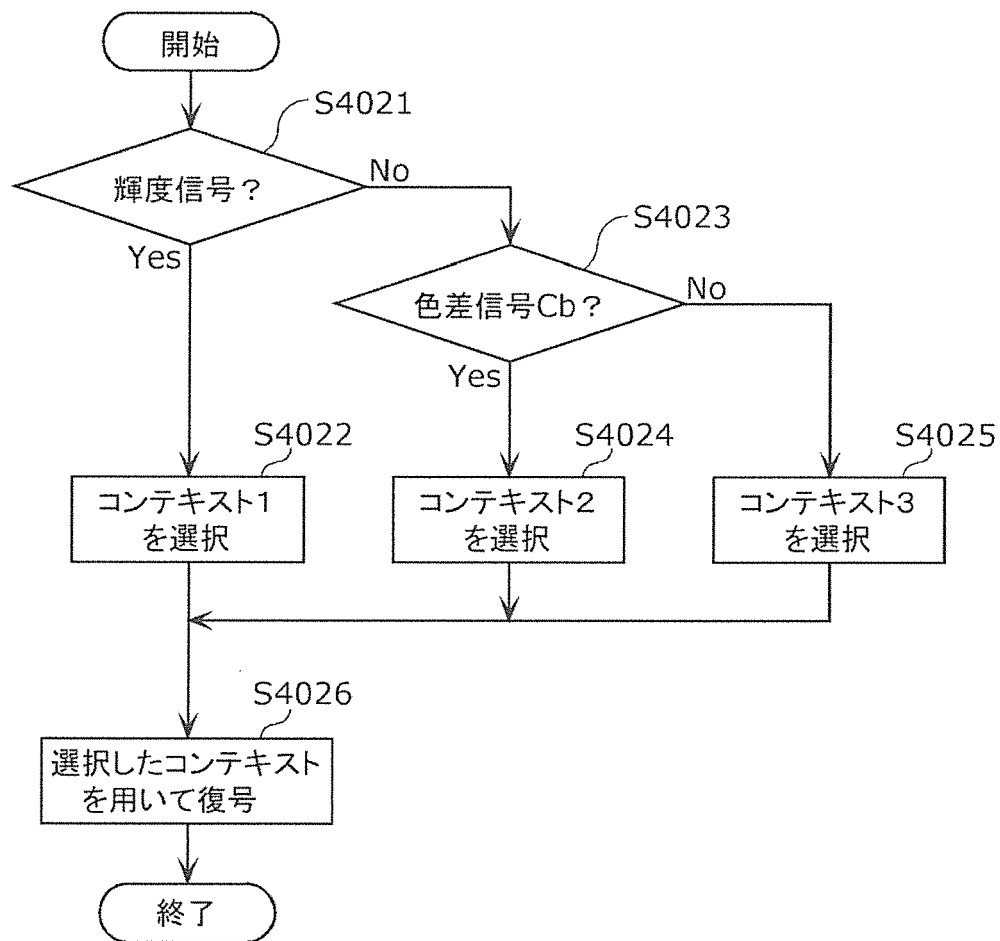
[図27C]

シンタックス要素	ctxIdxTable、 ctxIdxOffset	binIdx				
		0	1	2	3	>= 4
sao_on_flag	Table 9-5'	0	cIdx	na	na	na
		3	cIdx	na	na	na
		6	cIdx	na	na	na

[図27D]



[図27E]



[図28A]

sao_unit_cabac(rx, ry, cIdx){	Descriptor
sao_on_flag オフセット処理フラグ	u(1)
if(sao_on_flag) {	
if(rx > 0) {	
if(CtbAddrInSlice != 0 &&	
TileId[CtbAddrTS] == TileId[CtbAddrRStoTS[CtbAddrRS - 1]])	
sao_merge_left_flag	ae(v)
}	
if(!sao_merge_left_flag) {	
if(ry > 0) {	
if(((CtbAddrTS - CtbAddrRStoTS[CtbAddrRS - PicWidthInCtbs])	
<= CtbAddrInSlice) &&	
(TileId[CtbAddrTS] ==	
TileId[CtbAddrRStoTS[CtbAddrRS - PicWidthInCtbs]]))	
sao_merge_up_flag	ae(v)
}	
if(!sao_merge_up_flag)	
sao_offset_cabac(rx, ry, cIdx)	
}	
}	
}	

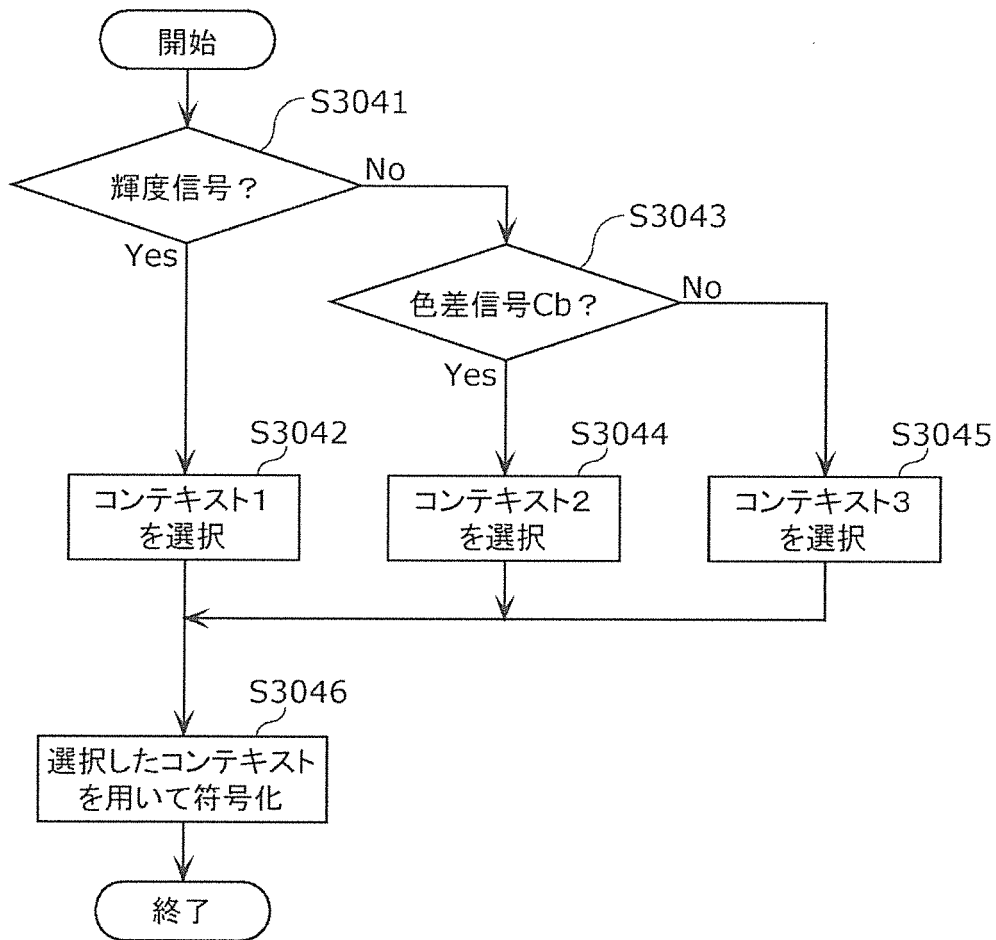
[図28B]

sao_offset_cabac(rx, ry, cIdx) {		
sao_type_idx[cIdx][rx][ry]	画素分類方法 インデックス番号	ae(v)
if(sao_type_idx[cIdx][rx][ry] != 0) {		
for(i = 0; i < 4; i++)		
sao_offset[cIdx][rx][ry][i]		ae(v)
}		
if(sao_type_idx[cIdx][rx][ry] == 5) {		
for(i = 0; i < 4; i++) {		
if(sao_offset[cIdx][rx][ry] != 0)		
sao_offset_sign[cIdx][rx][ry][i]		ae(v)
}		
sao_band_position[cIdx][rx][ry]		ae(v)
}		
}		

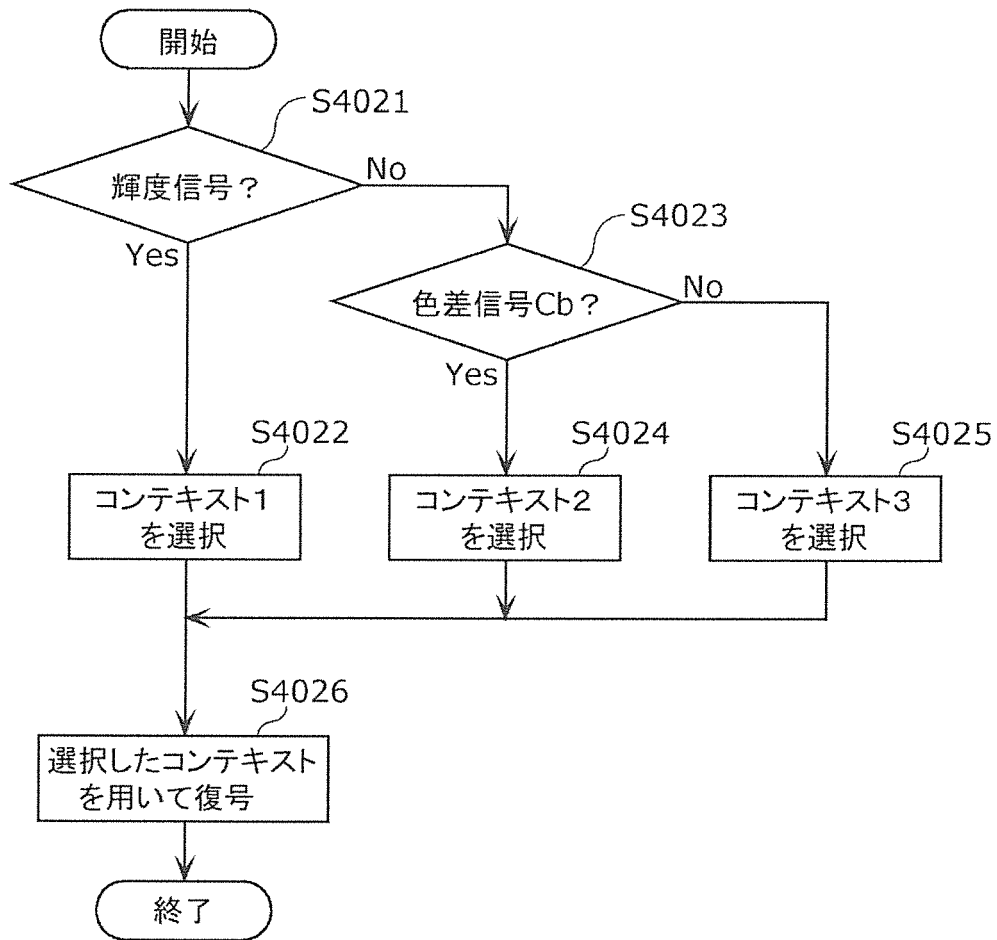
[図28C]

シンタックス要素	ctxIdxTable、 ctxIdxOffset		binIdx				
			0	1	2	3	>=4
sao_on_flag	Table 9-5'	0	cIdx	na	na	na	na
		3	cIdx	na	na	na	na
		6	cIdx	na	na	na	na
sao_merge_left_flag	Table 9-5	0	cIdx	na	na	na	na
		3	cIdx	na	na	na	na
		6	cIdx	na	na	na	na
sao_merge_up_flag	Table 9-6	0	0	na	na	na	na
		1	0	na	na	na	na
		2	0	na	na	na	na
sao_type_idx	Table 9-7	0	0	1	1	1	1
		2	0	1	1	1	1
		4	0	1	1	1	1
sao_offset	Table 9-8	0	0	1	1	1	1
		2	0	1	1	1	1
		4	0	1	1	1	1
sao_offset_sign	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_band_position	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na

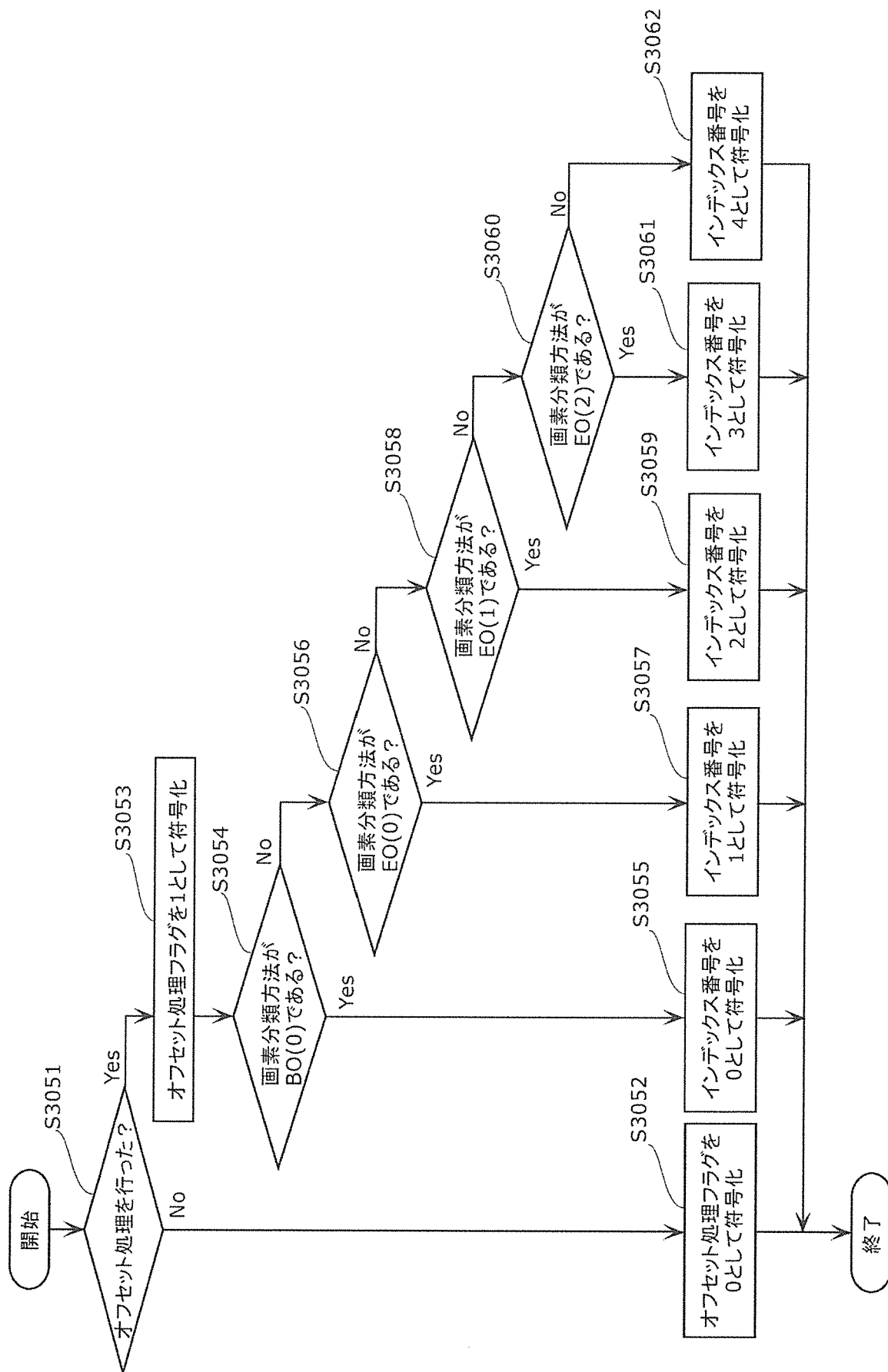
[図28D]



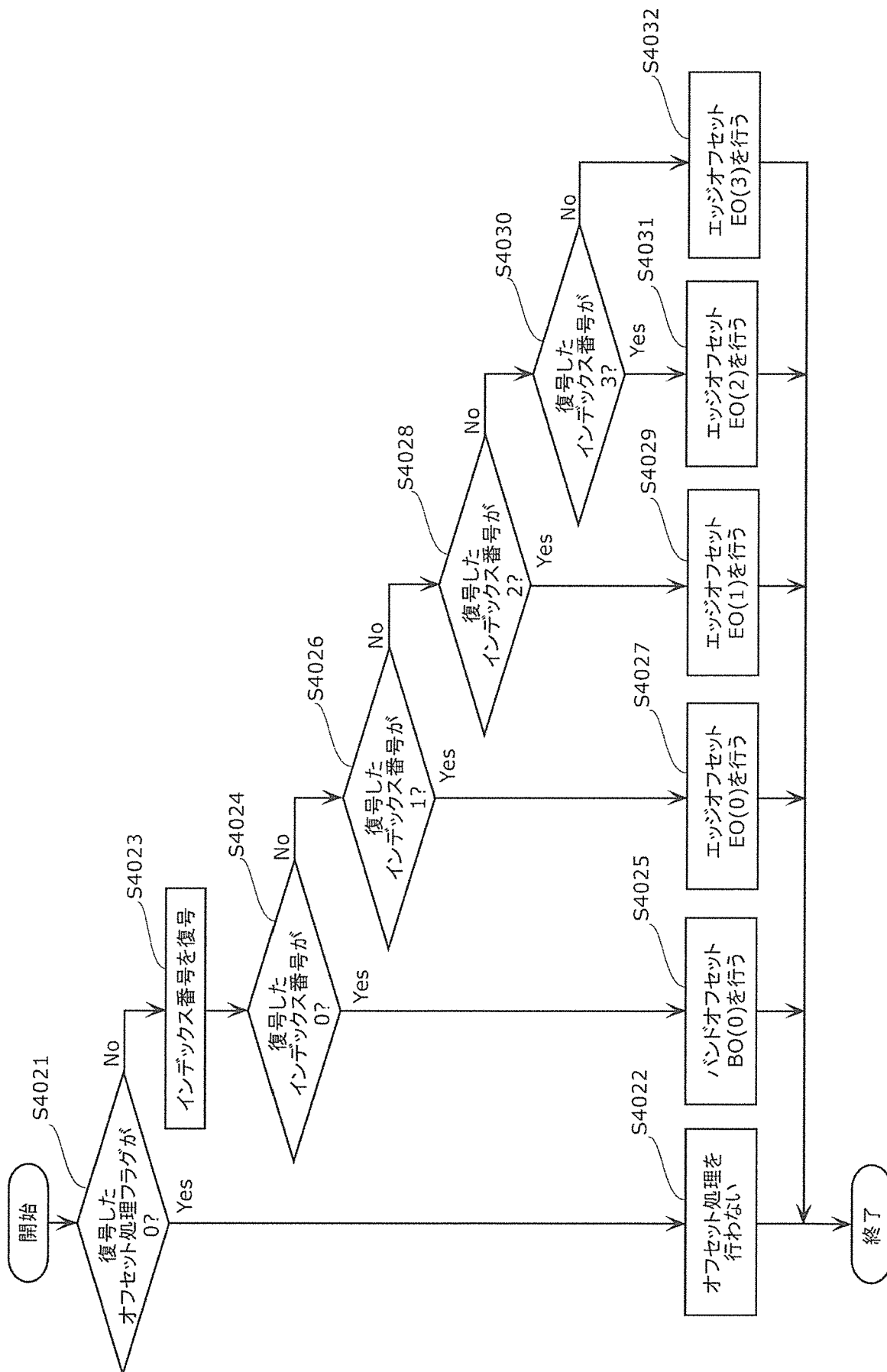
[図28E]



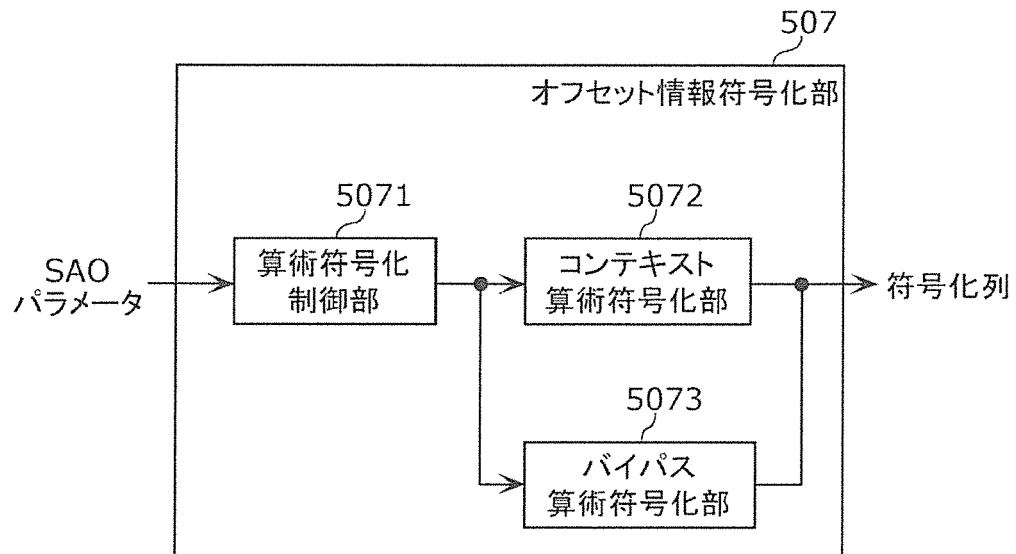
[図29]



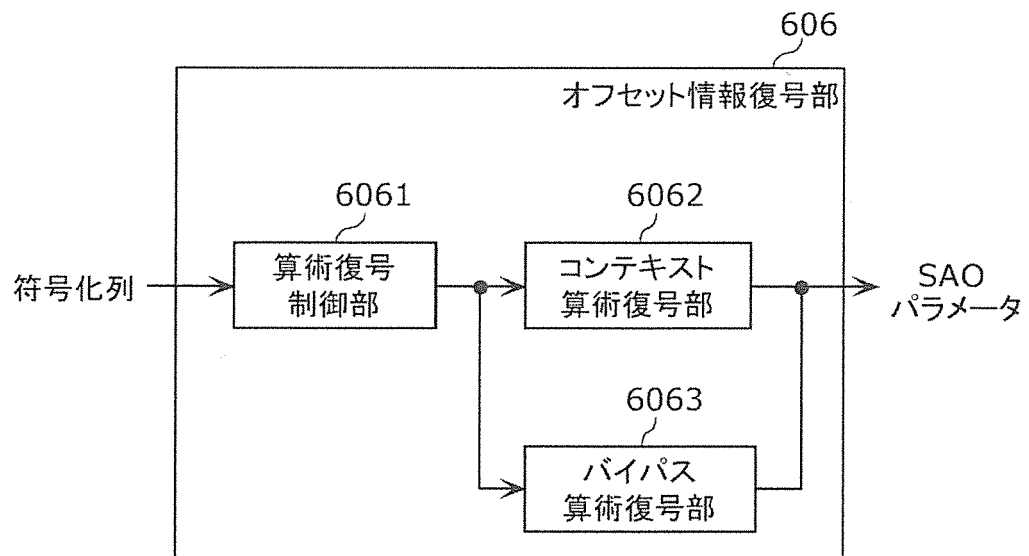
[図30]



[図31]



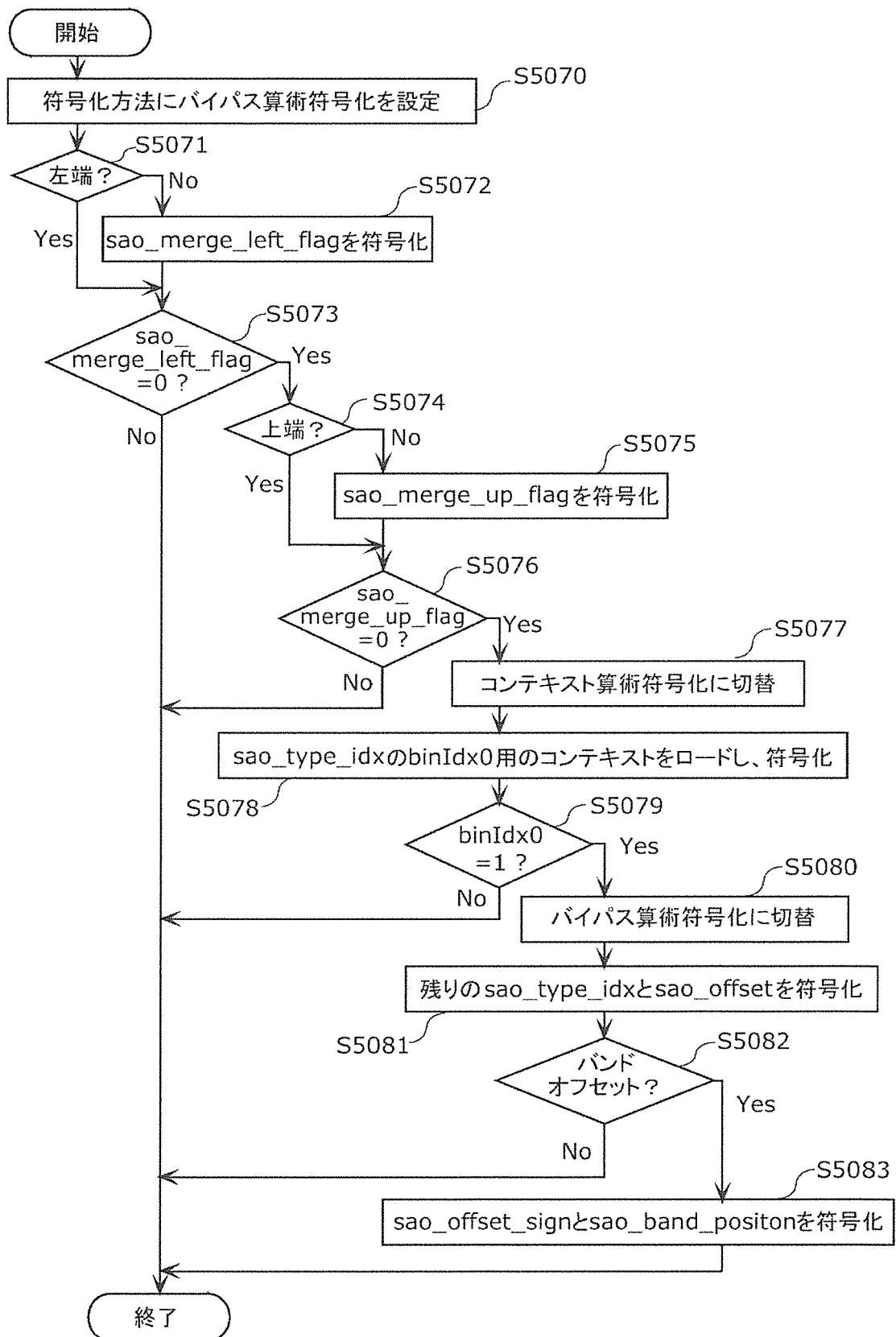
[図32]



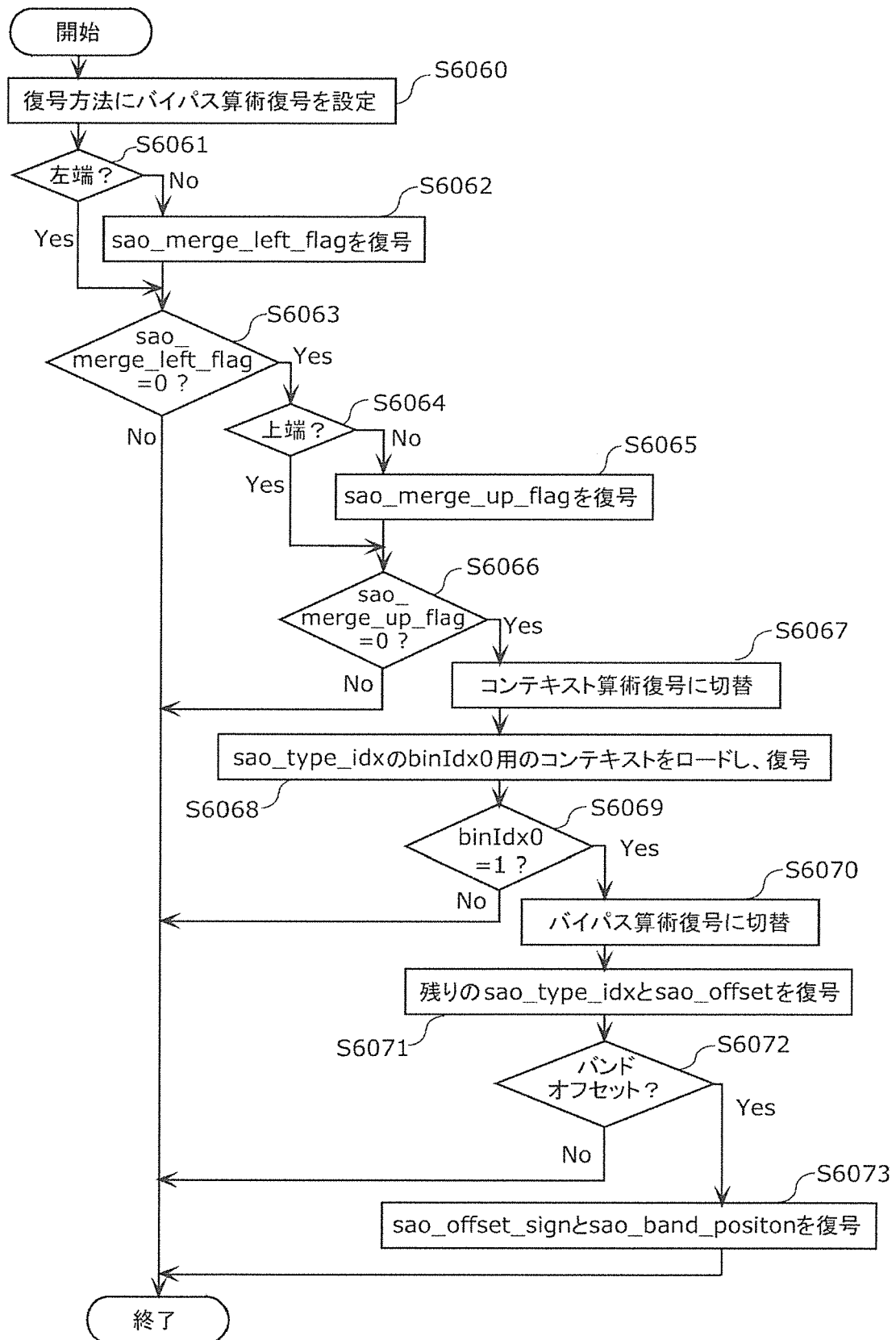
[図33]

シンタックス要素	ctxIdxTable- ctxIdxOffset		binIdx				
			0	1	2	3	>=4
sao_merge_left_flag	na	na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
sao_merge_up_flag	na	na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
sao_type_idx	Table 9-X	0	0	na(uses De- code Bypass)	na	na	na
		1	0	na(uses De- code Bypass)	na	na	na
		2	0	na(uses De- code Bypass)	na	na	na
sao_offset	na	na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
sao_offset_sign	na	na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
sao_band_position	na	na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na
		na	na(uses De- code Bypass)	na	na	na	na

[図34]



[図35]



[図36A]

シンタックス要素	ctxIdxTable、 ctxIdxOffset		binIdx				
			0	1	2	3	>=4
sao_on_flag	Table 9-X	0	0	na	na	na	na
		1	0	na	na	na	na
		2	0	na	na	na	na
sao_merge_left_flag	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_merge_up_flag	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_type_idx	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_offset	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_offset_sign	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_band_position	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na

[図36B]

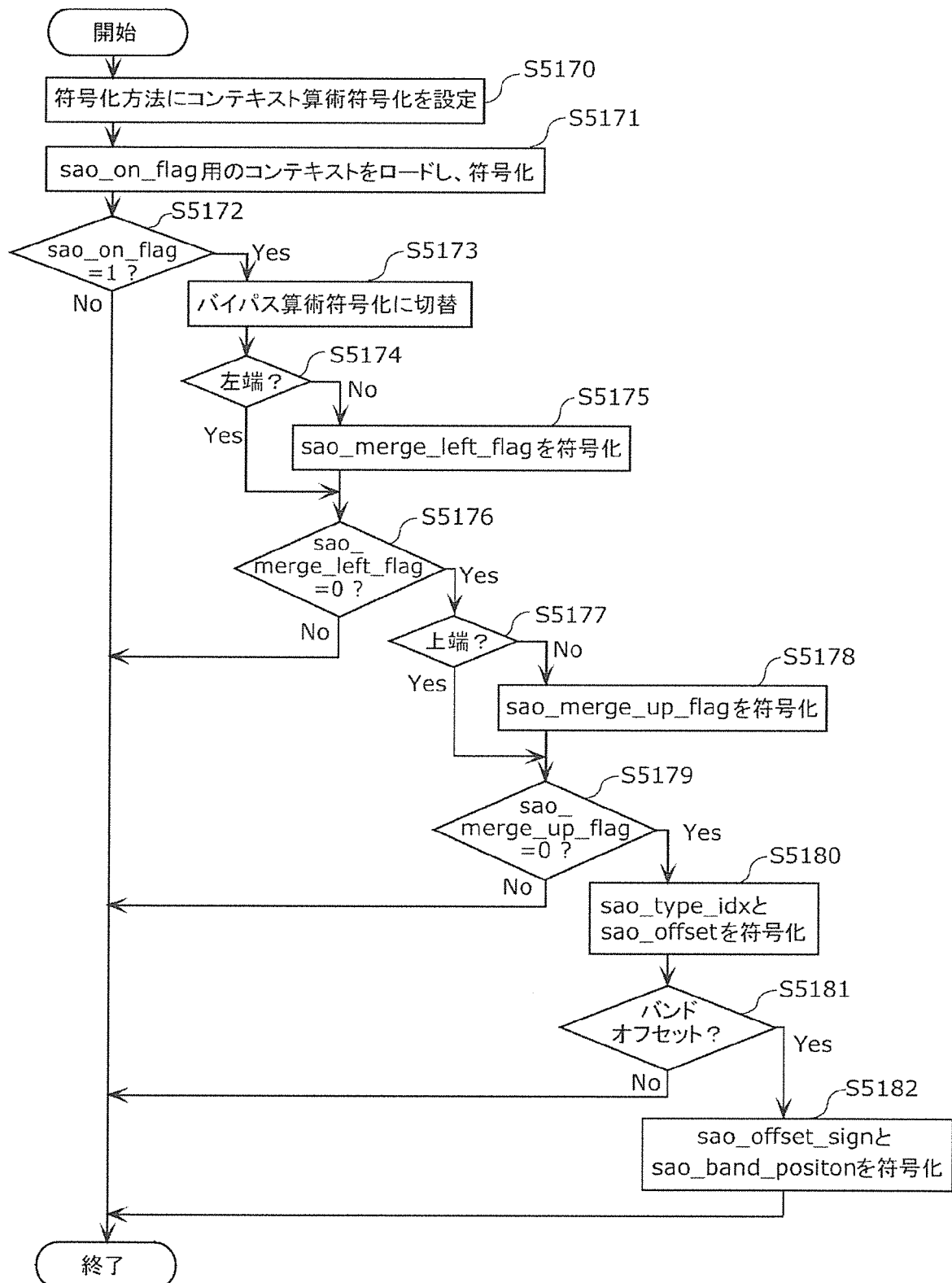
インデックス	ビット割り当て	オフセット処理フラグ
0	0	オフセット処理しない
1	1	オフセット処理する

インデックス	ビット割り当て	画素分類方法
0	0	エッジオフセット(0)
1	10	エッジオフセット(1)
2	110	エッジオフセット(2)
3	1110	エッジオフセット(3)
4	1111	バンドオフセット

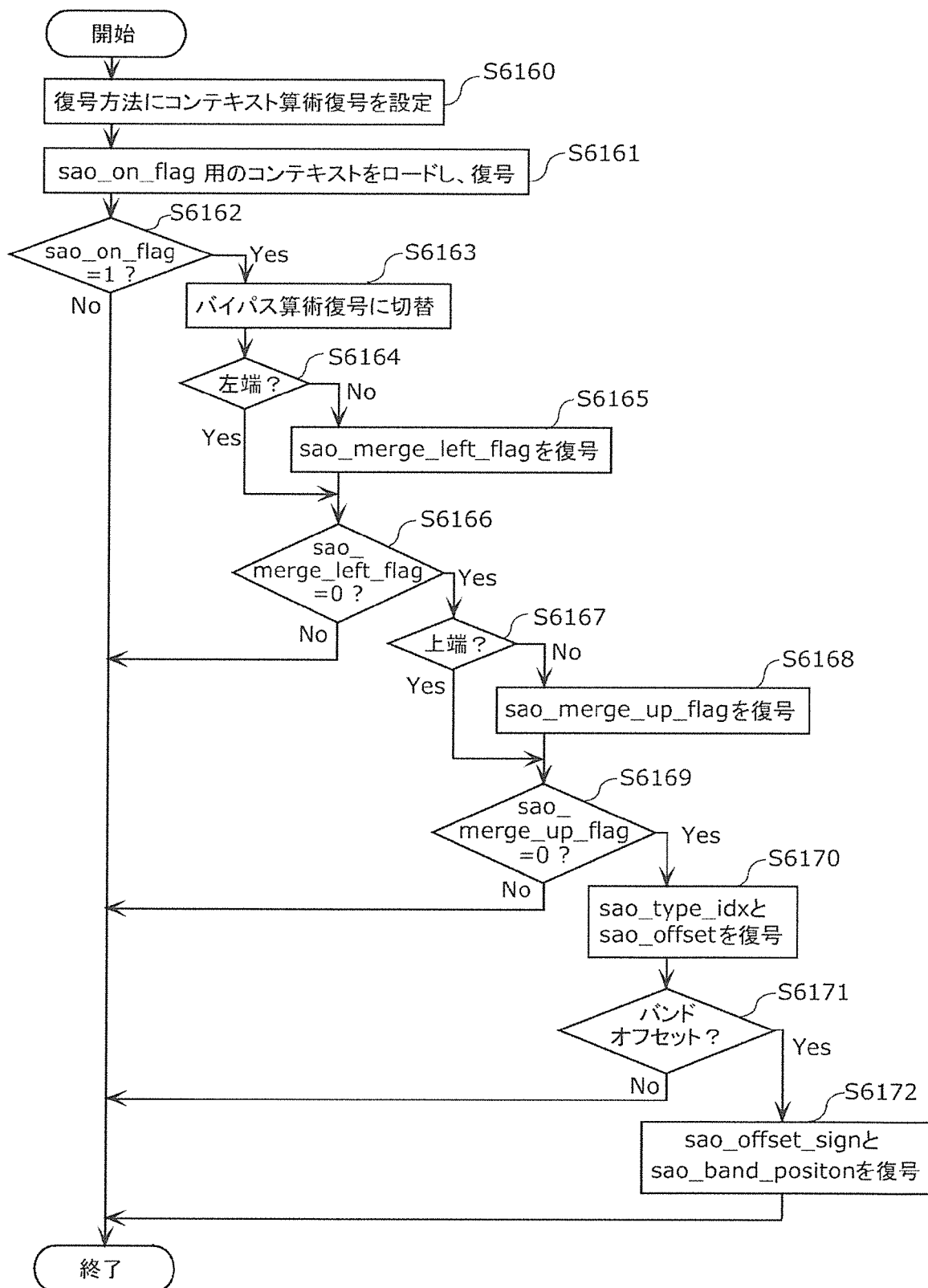
[図36C]

Configuration	BD-rate
All Intra Main	0.1%
Random Access Main	-0.1%
Low Delay B Main	0.0%
All Intra HE10	0.1%
Random Access HE10	0.0%
Low delay B HE10	0.0%

[図37]



[図38]



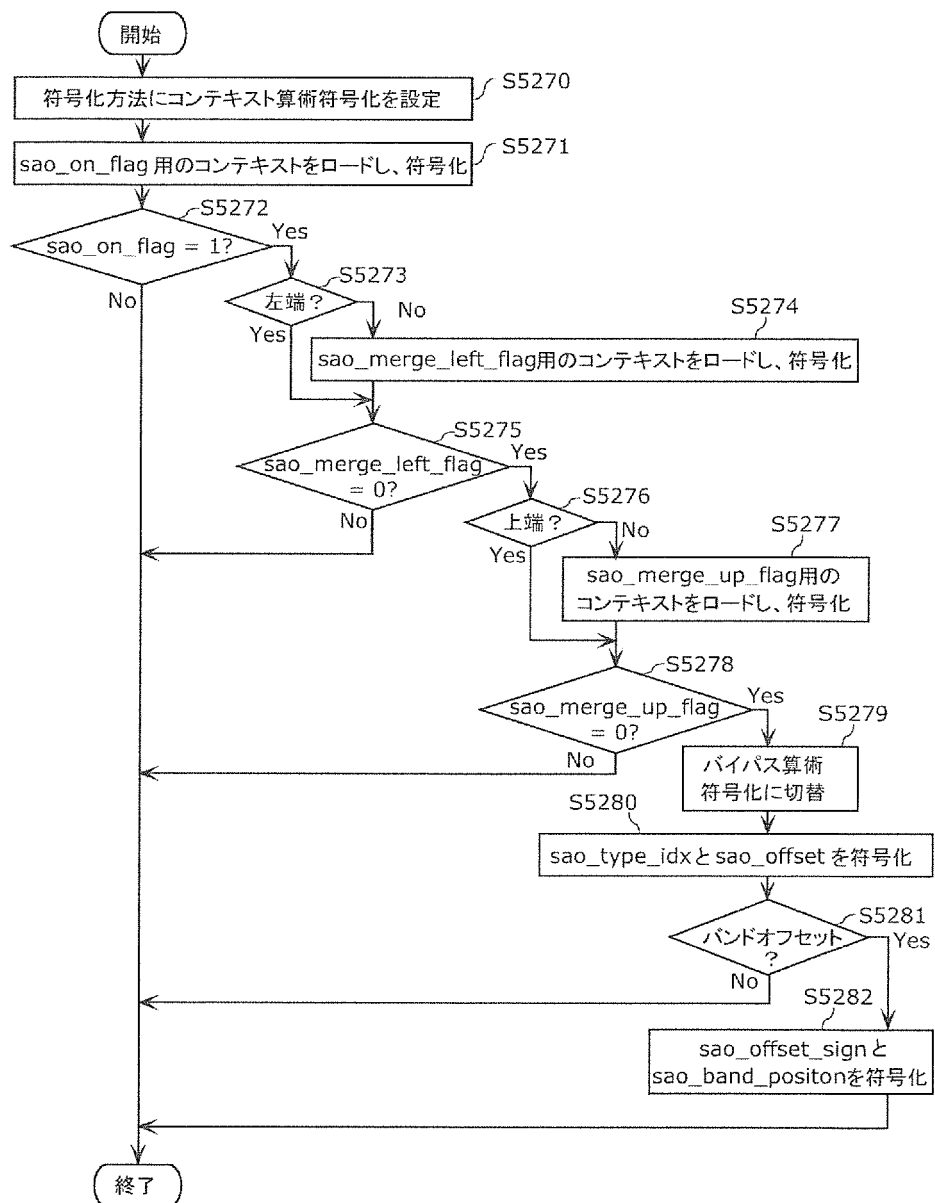
[図39A]

シンタックス要素	ctxIdxTable、 ctxIdxOffset		binIdx				
			0	1	2	3	>=4
sao_on_flag	Table 9-X	0	0	na	na	na	na
		1	0	na	na	na	na
		2	0	na	na	na	na
sao_merge_left_flag	Table 9-XX	0	0	na	na	na	na
		1	0	na	na	na	na
		2	0	na	na	na	na
sao_merge_up_flag	Table 9-XXX	0	0	na	na	na	na
		1	0	na	na	na	na
		2	0	na	na	na	na
sao_type_idx	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_offset	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_offset_sign	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
sao_band_position	na	na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na
		na	na(uses Decode Bypass)	na	na	na	na

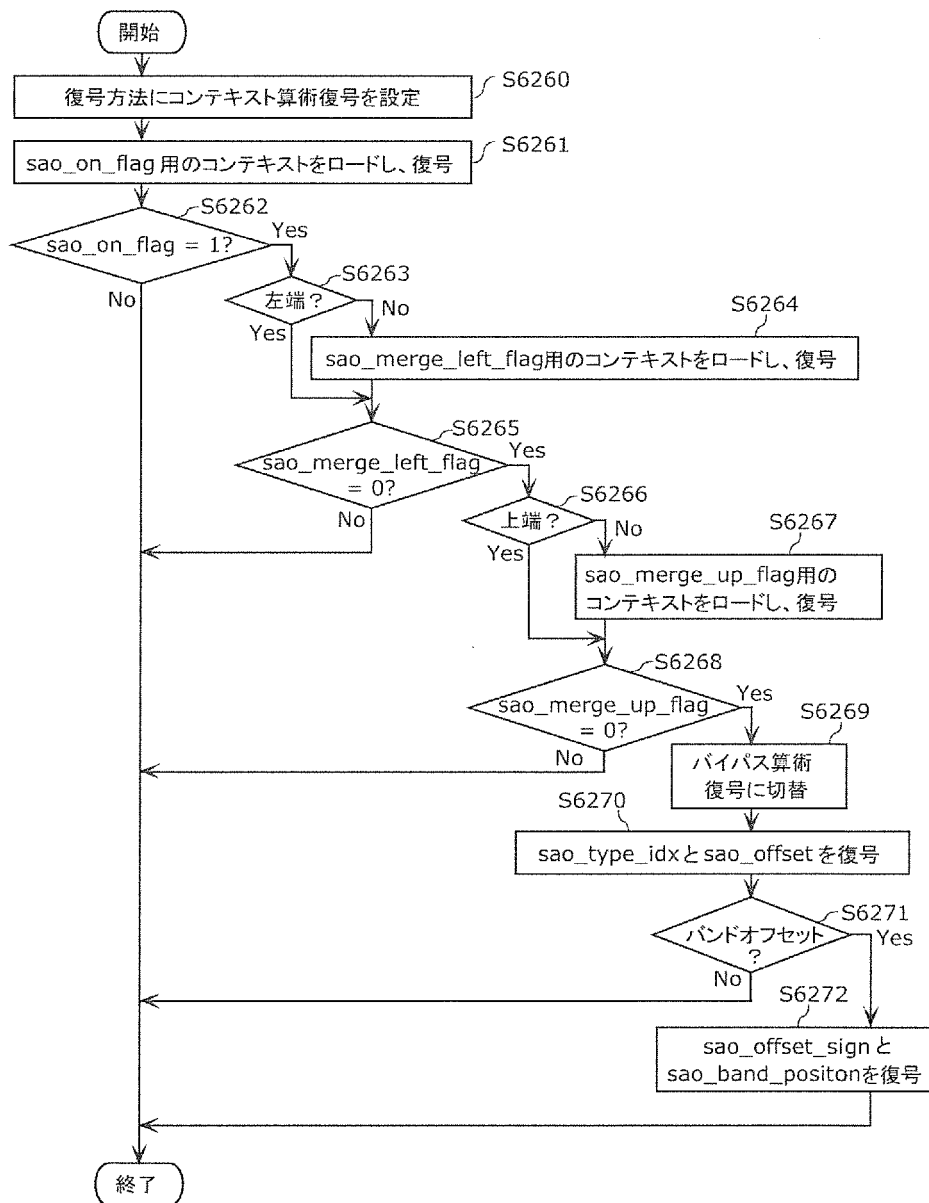
[図39B]

Configuration	BD-rate
All Intra Main	0.1%
Random Access Main	0.0%
Low Delay B Main	0.0%
All Intra HE10	0.1%
Random Access HE10	-0.1%
Low delay B HE10	0.0%

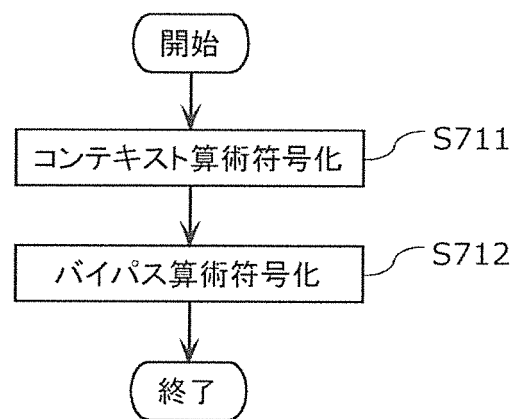
[図40]



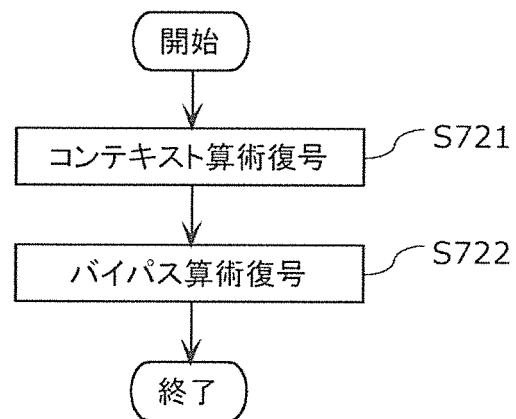
[図41]



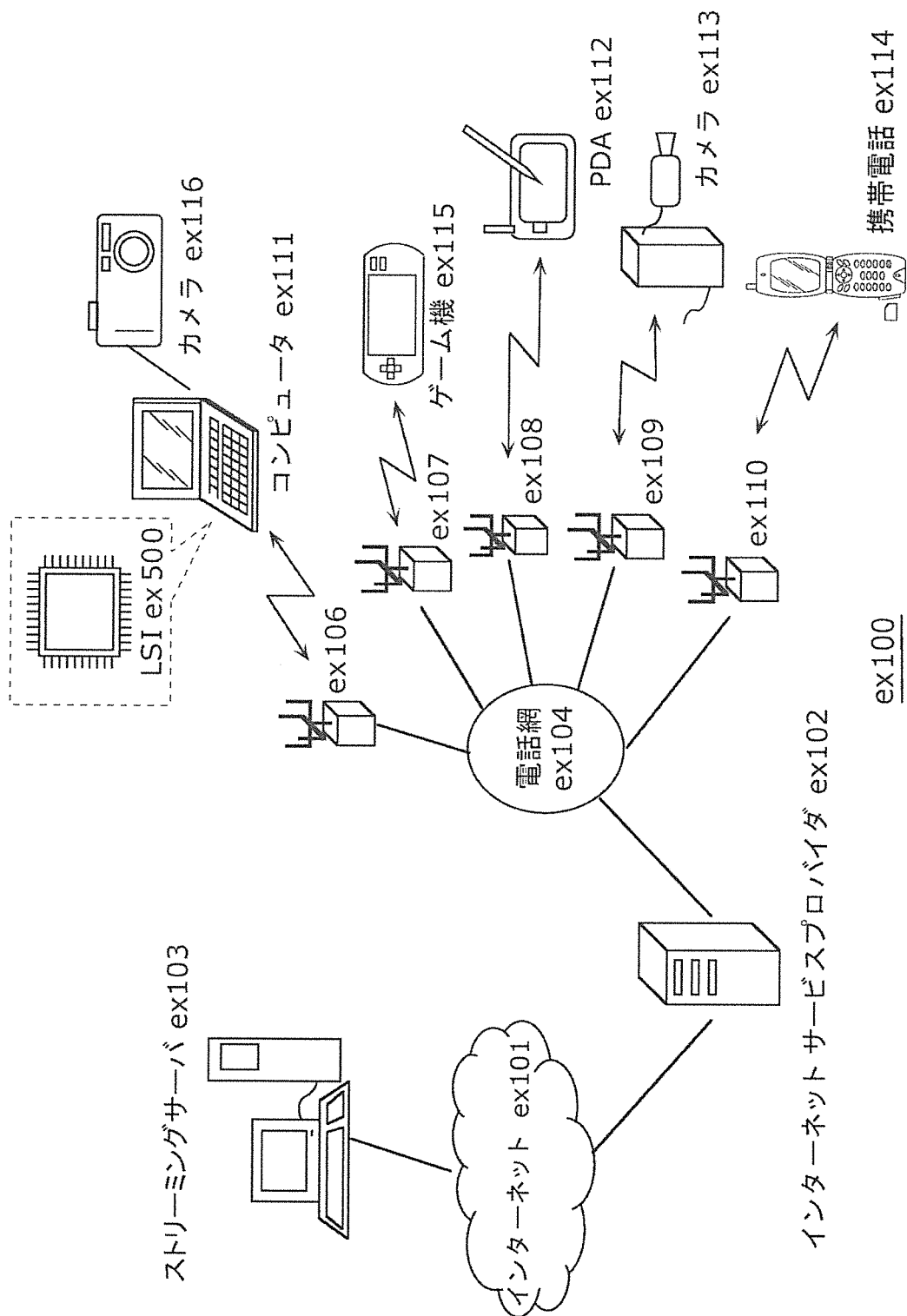
[図42]



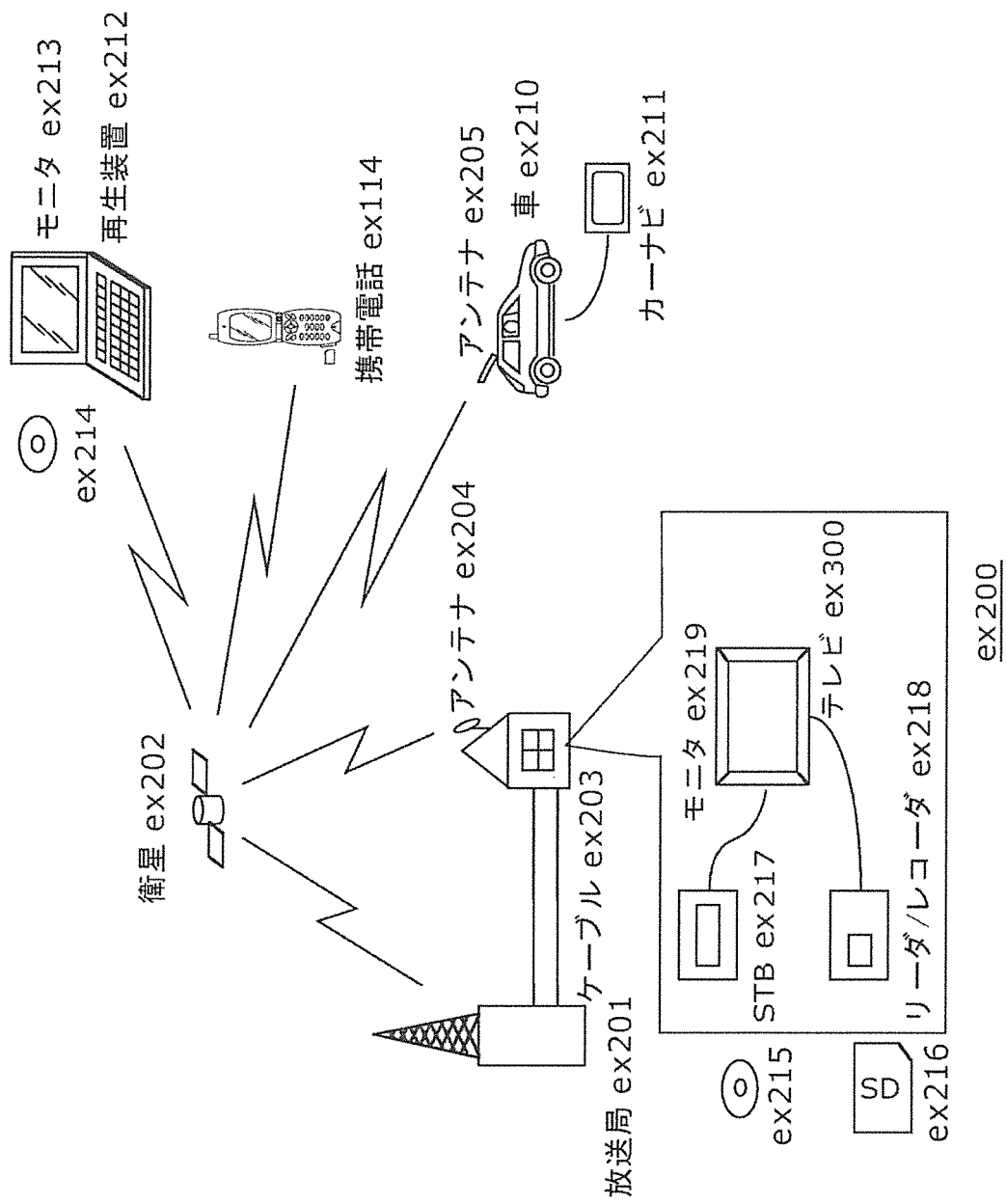
[図43]



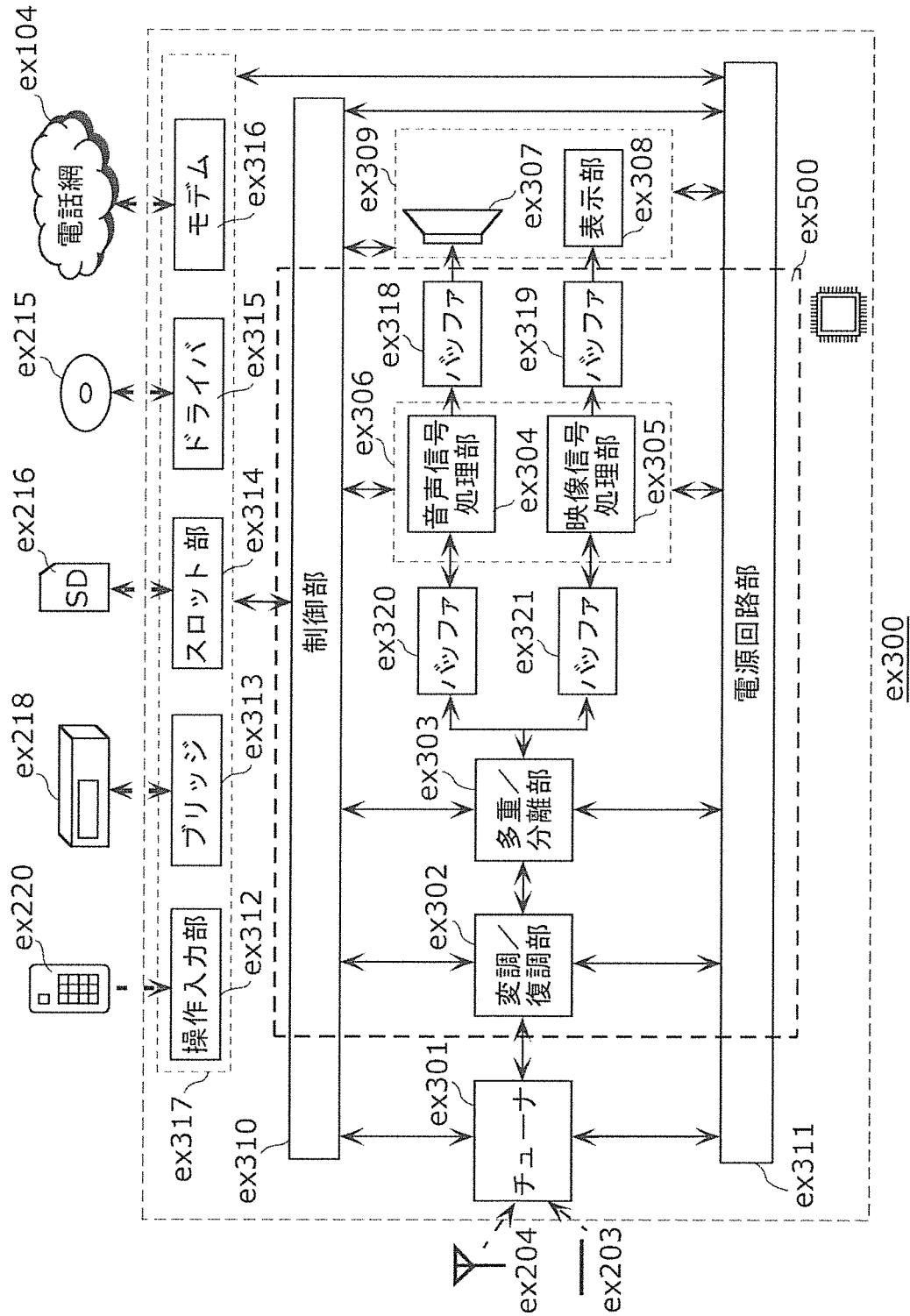
[図44]



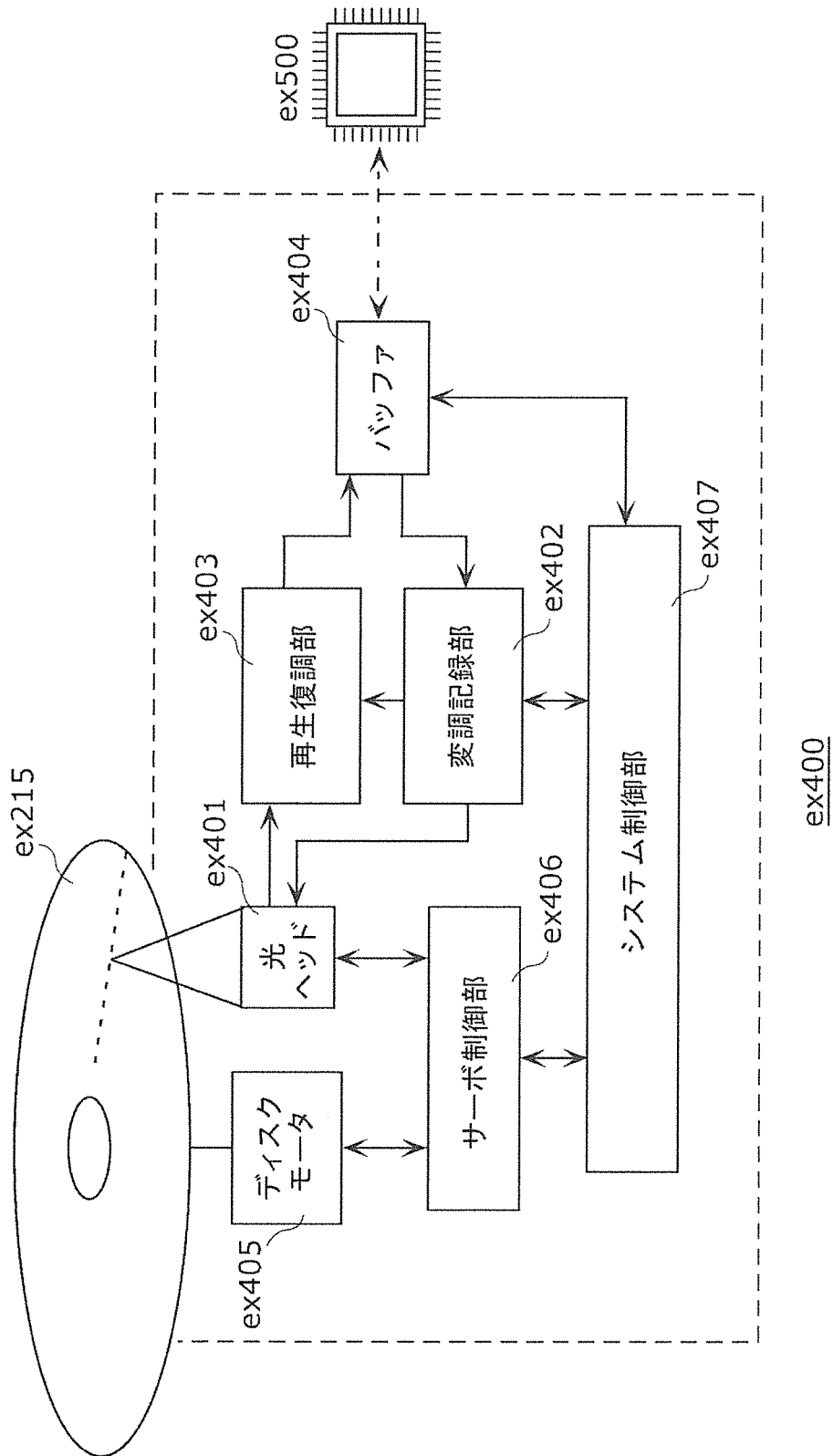
[図45]



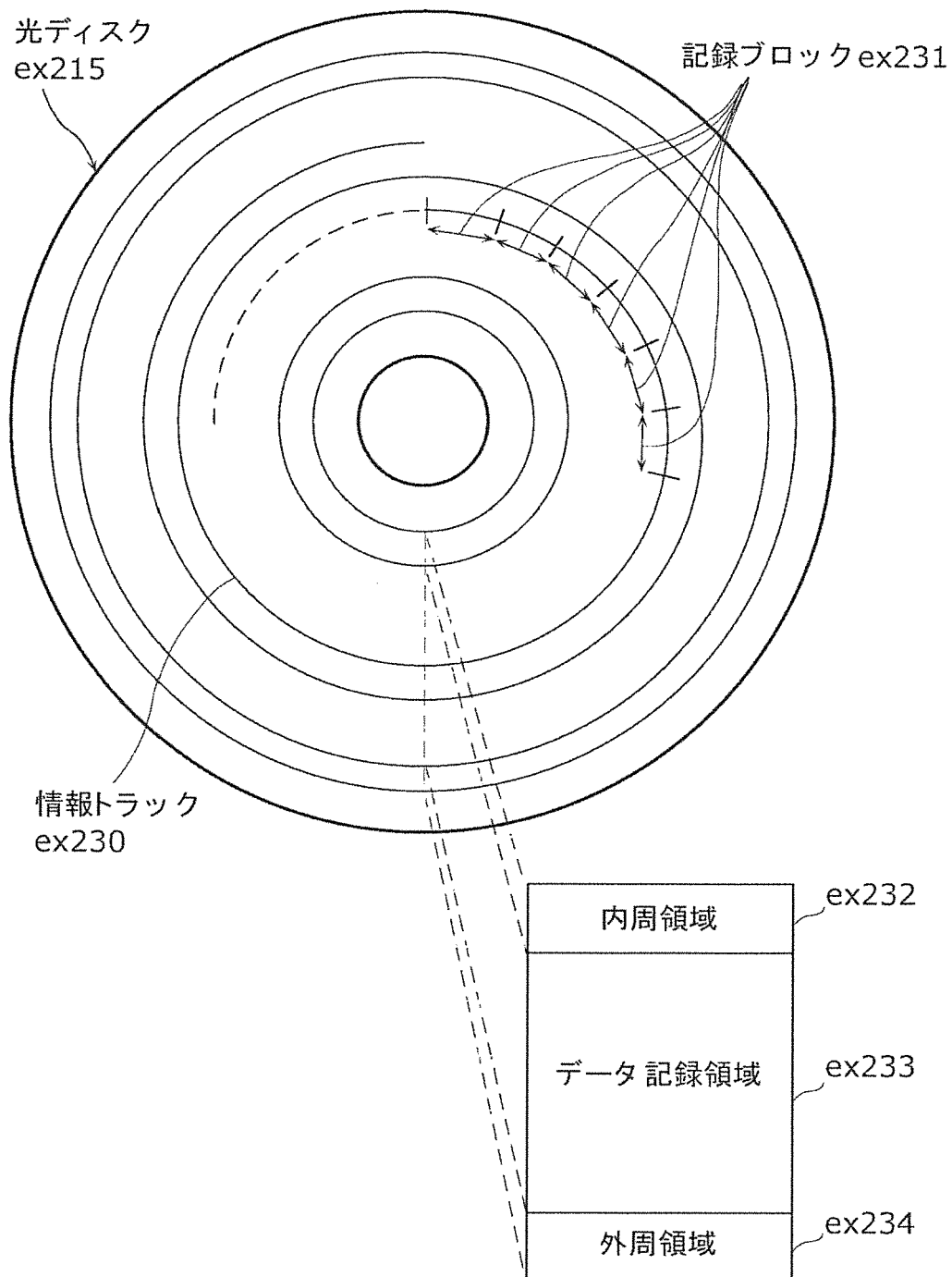
[図46]



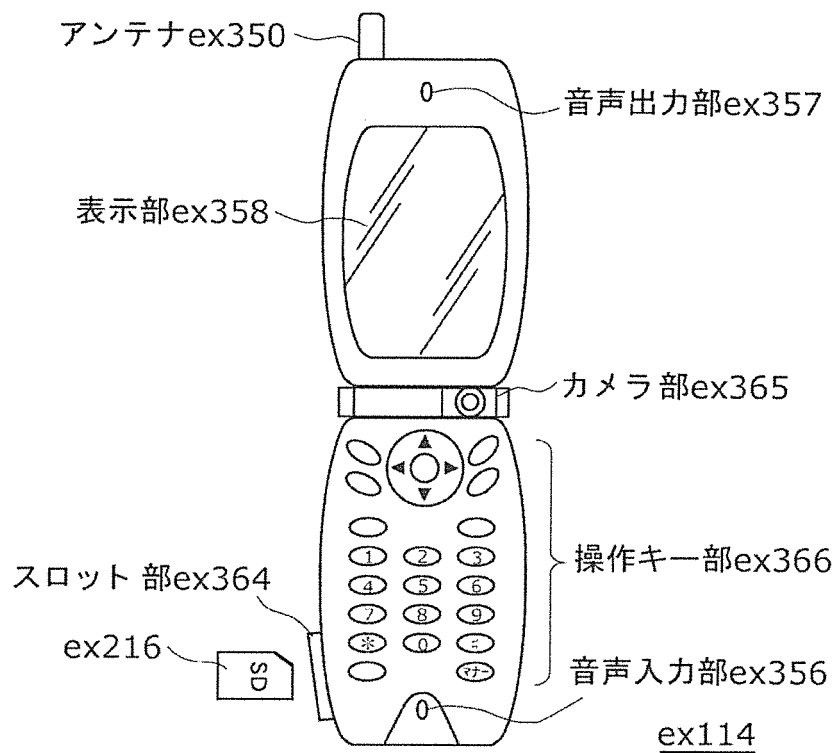
[図47]



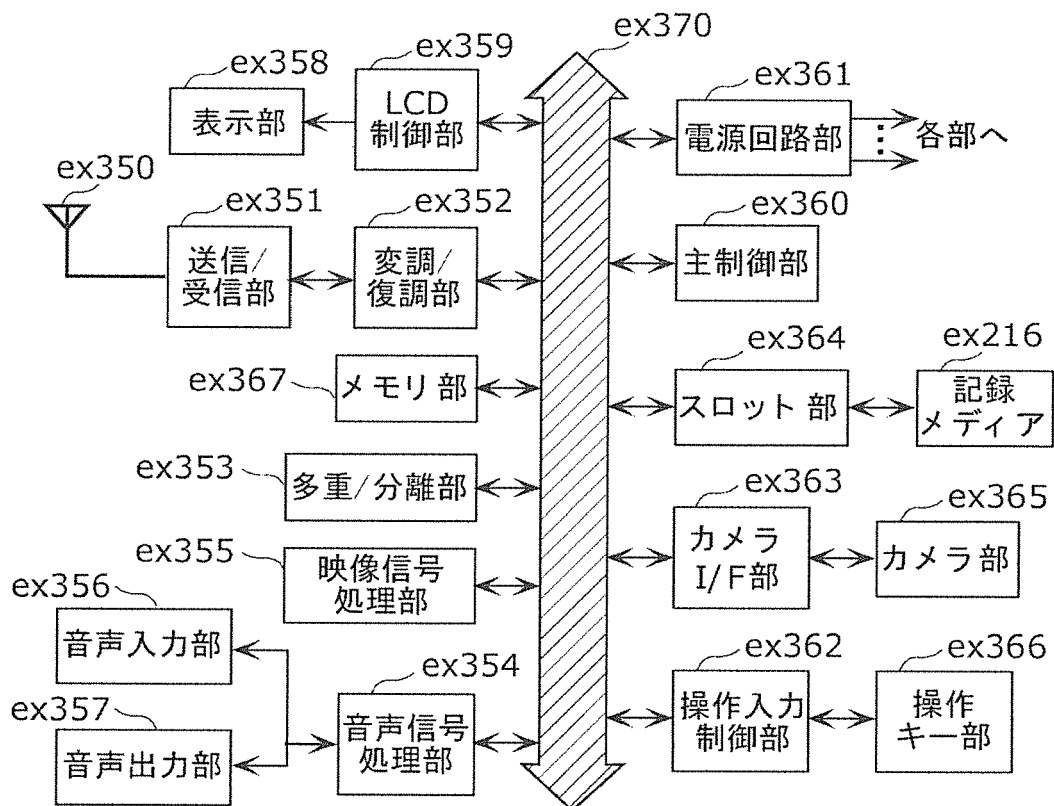
[図48]



[図49A]



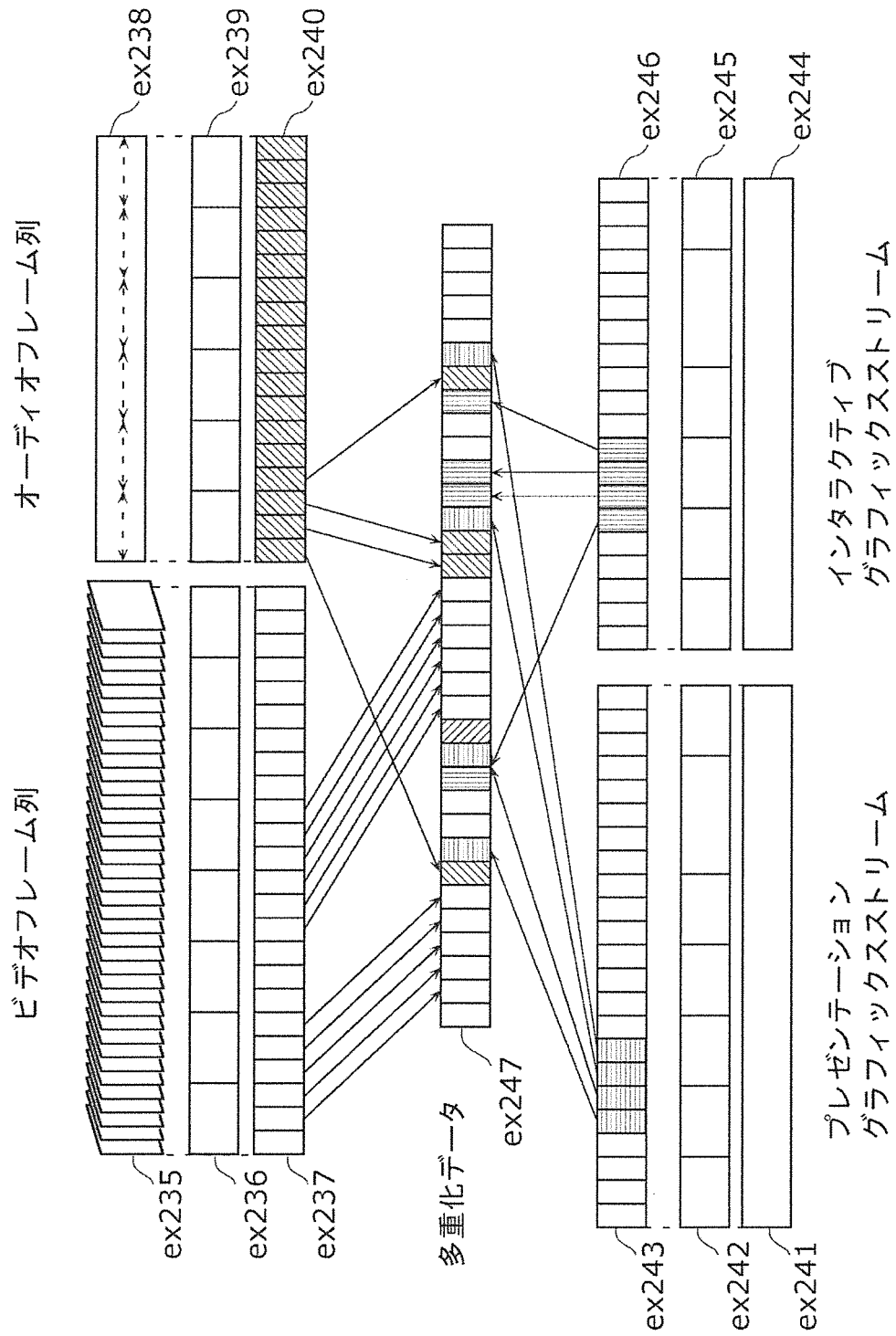
[図49B]



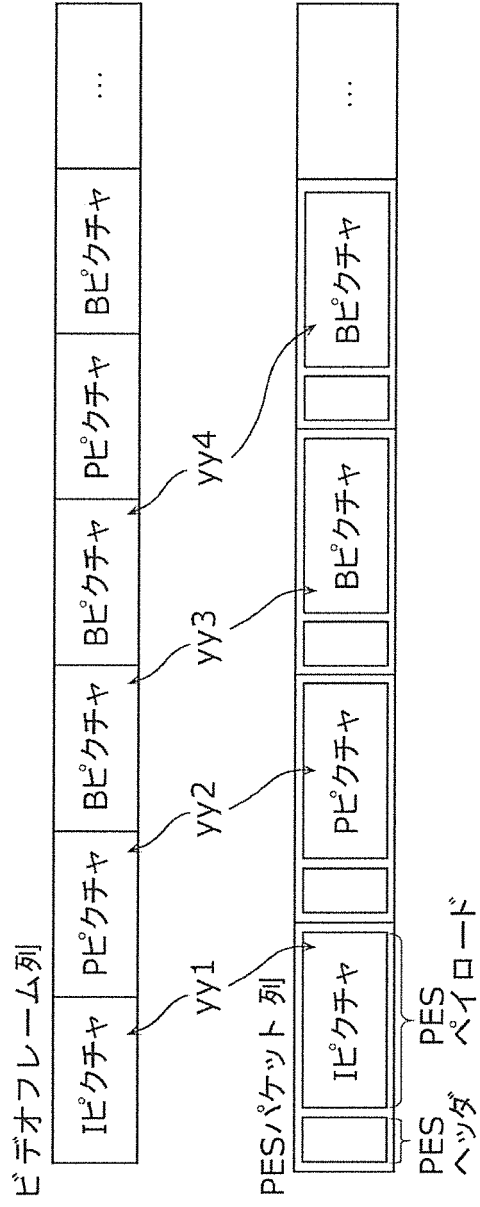
[図50]

ビデオストリーム(PID=0x1011 主映像)
オーディオストリーム(PID=0x1100)
オーディオストリーム(PID=0x1101)
プレゼンテーショングラフィックスストリーム(PID=0x1200)
プレゼンテーショングラフィックスストリーム(PID=0x1201)
インタラクティブグラフィックスストリーム(PID=0x1400)
ビデオストリーム(PID=0x1B00 副映像)
ビデオストリーム(PID=0x1B01 副映像)

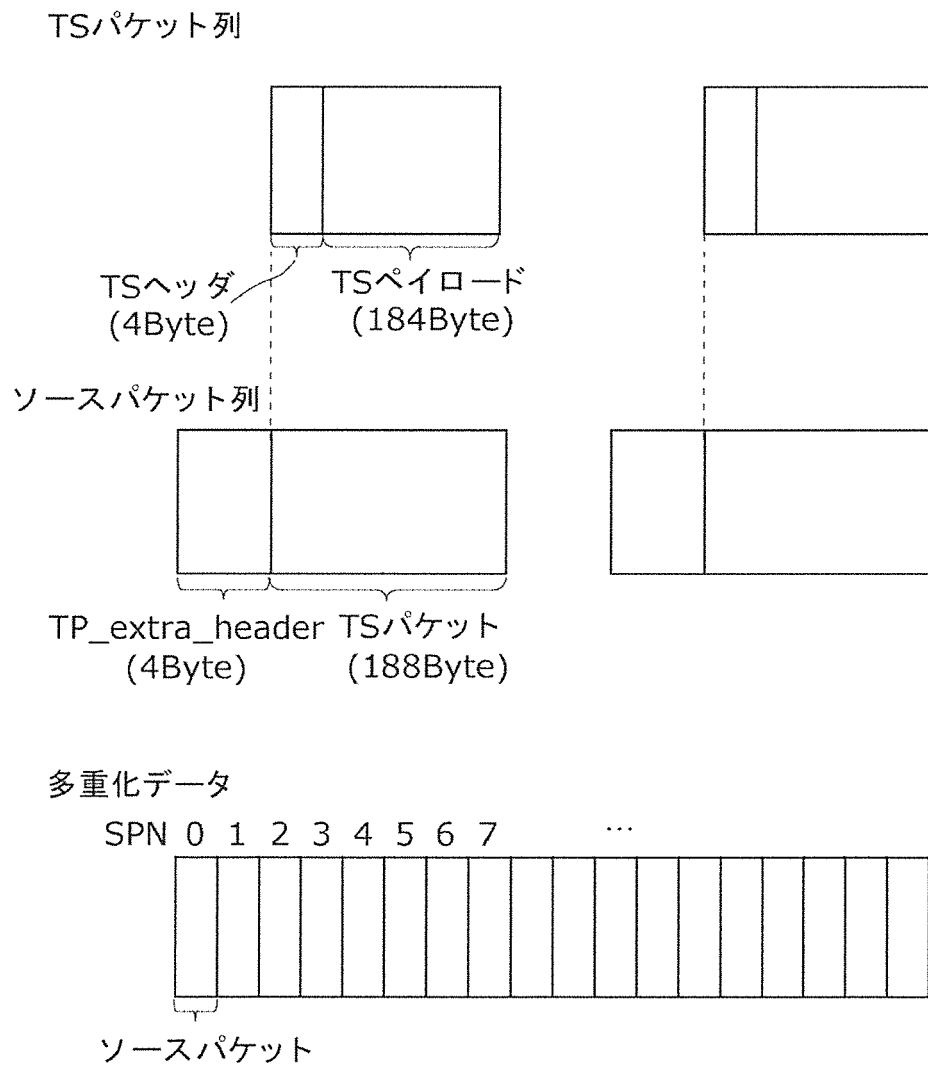
[図51]



[図52]

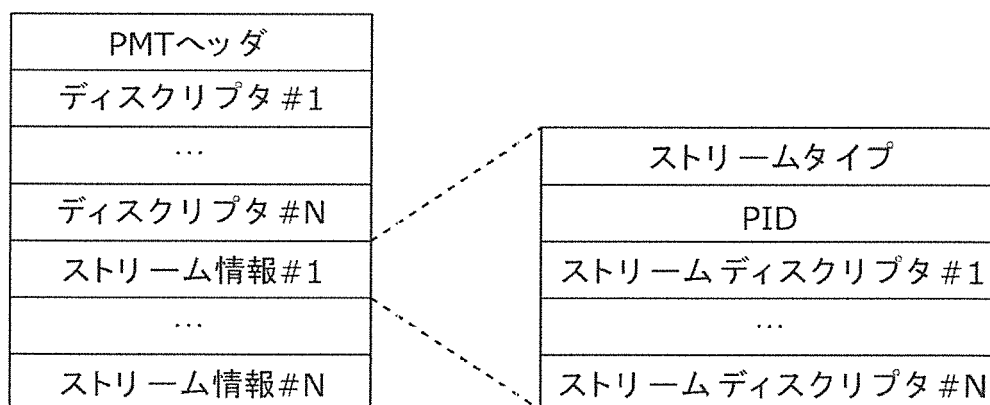


[図53]

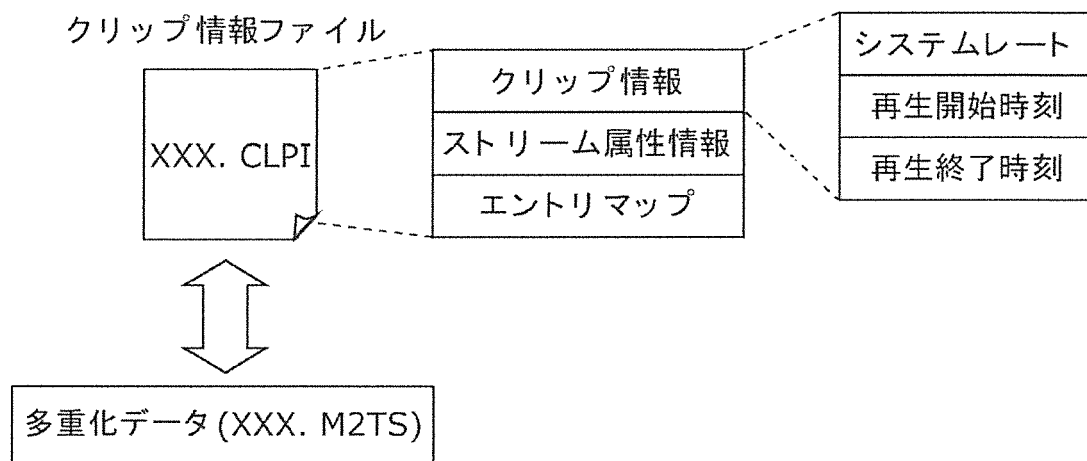


[図54]

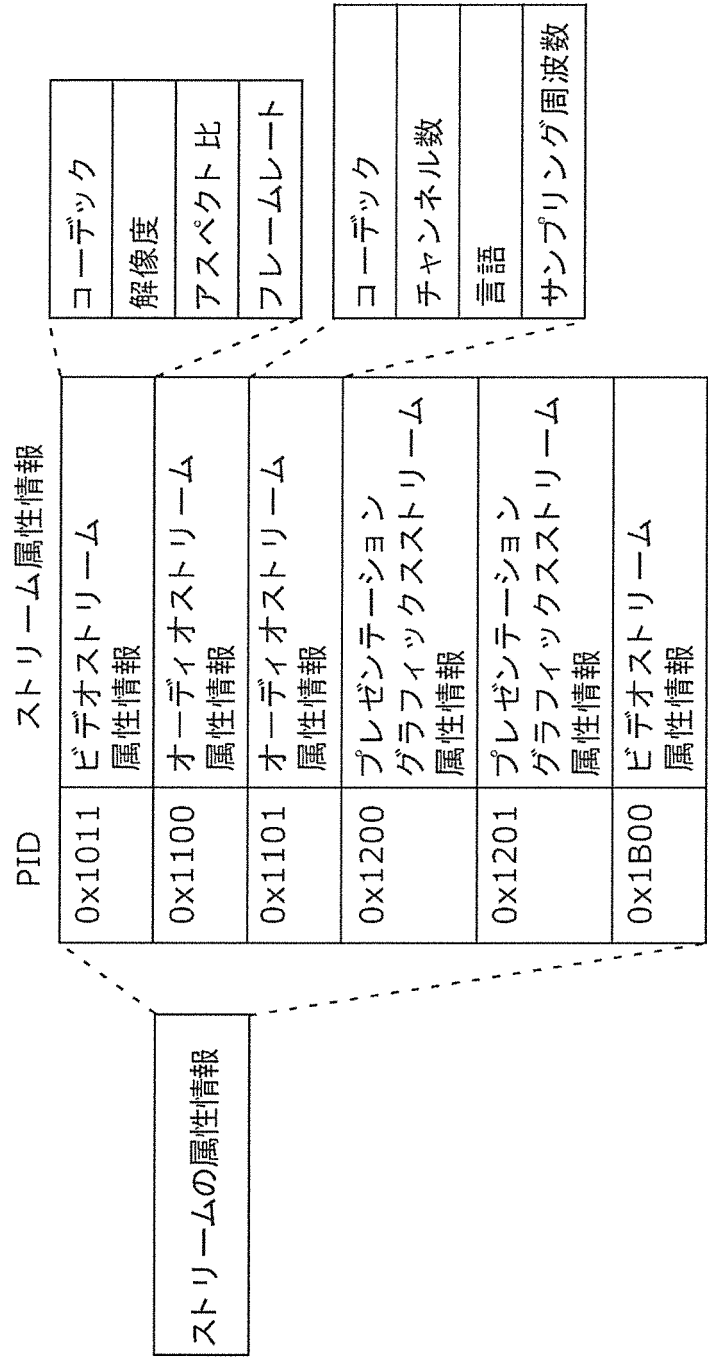
PMTのデータ構造



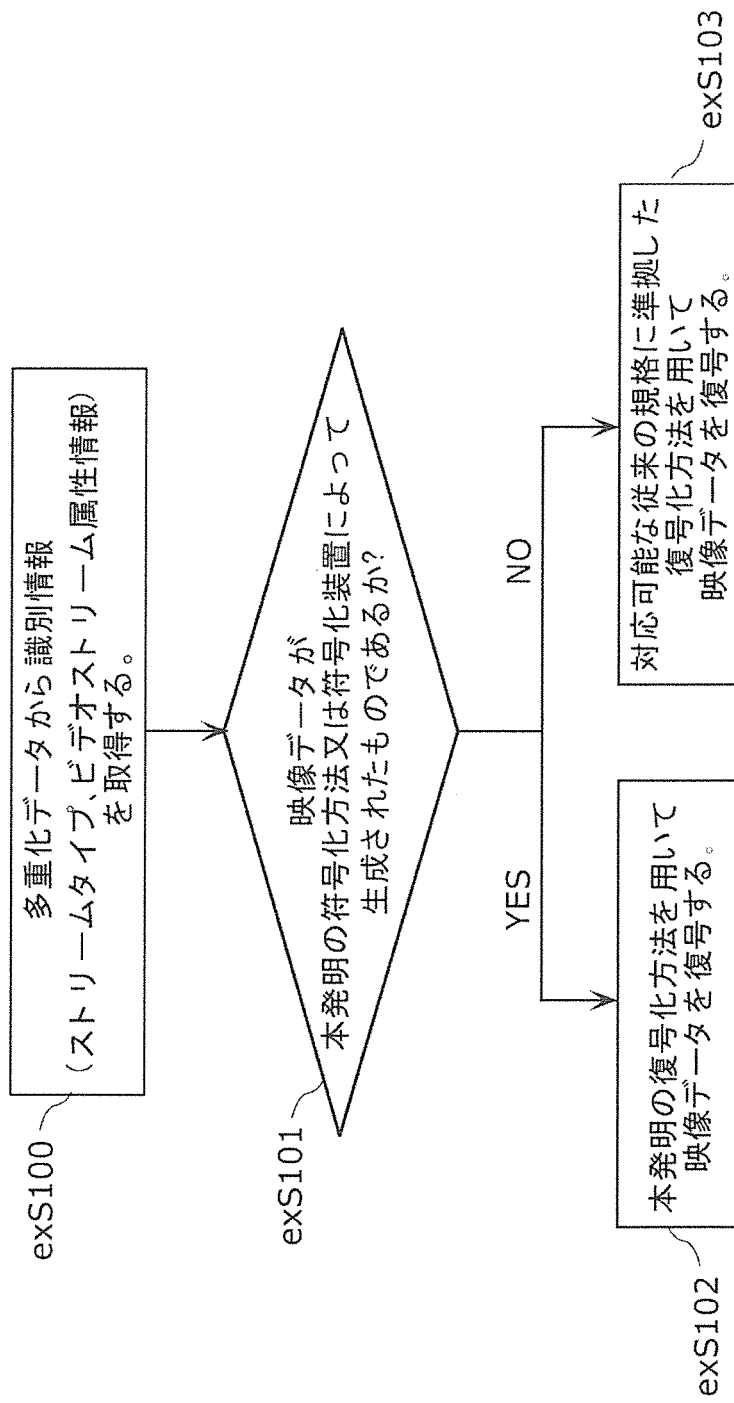
[図55]



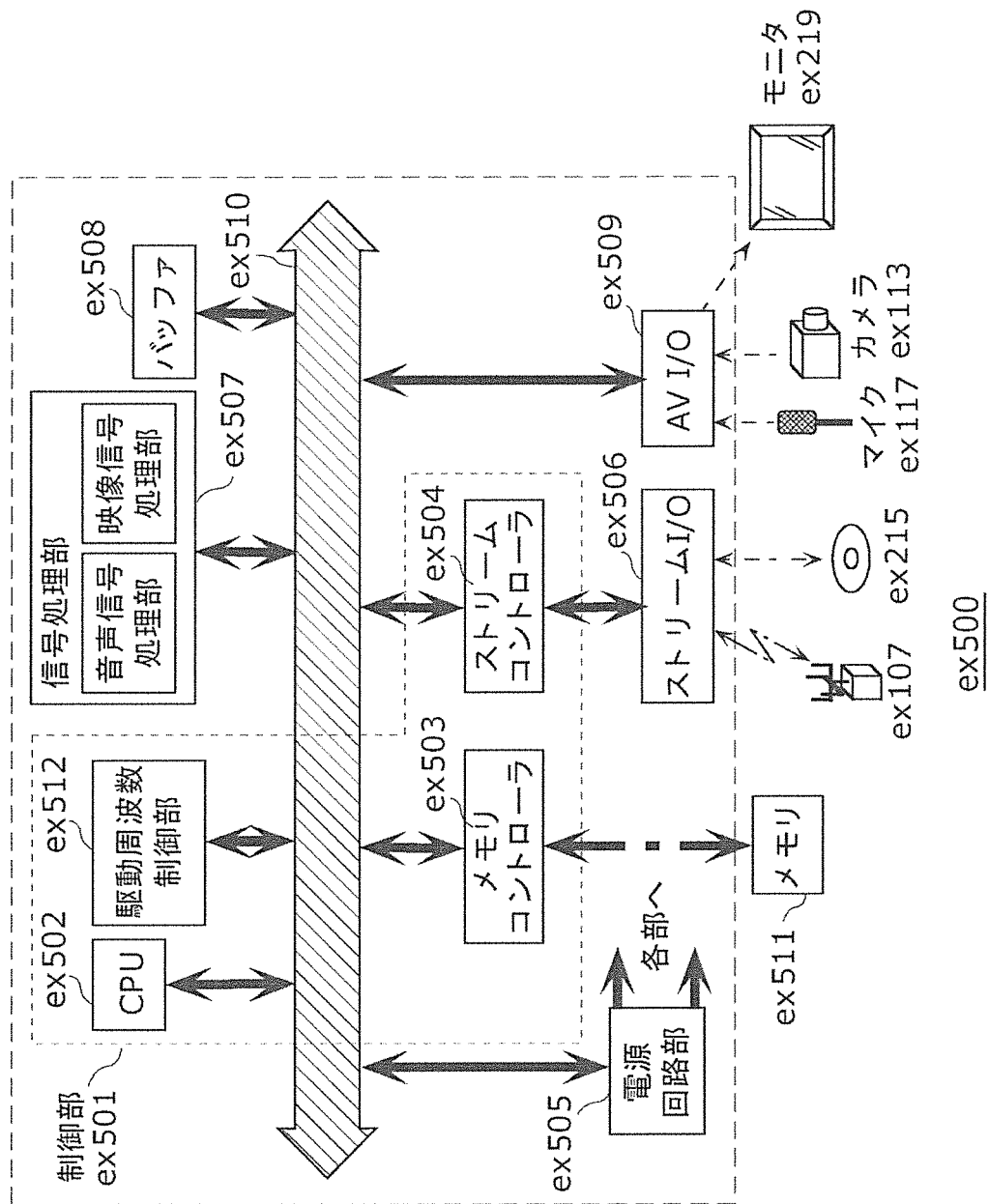
[図56]



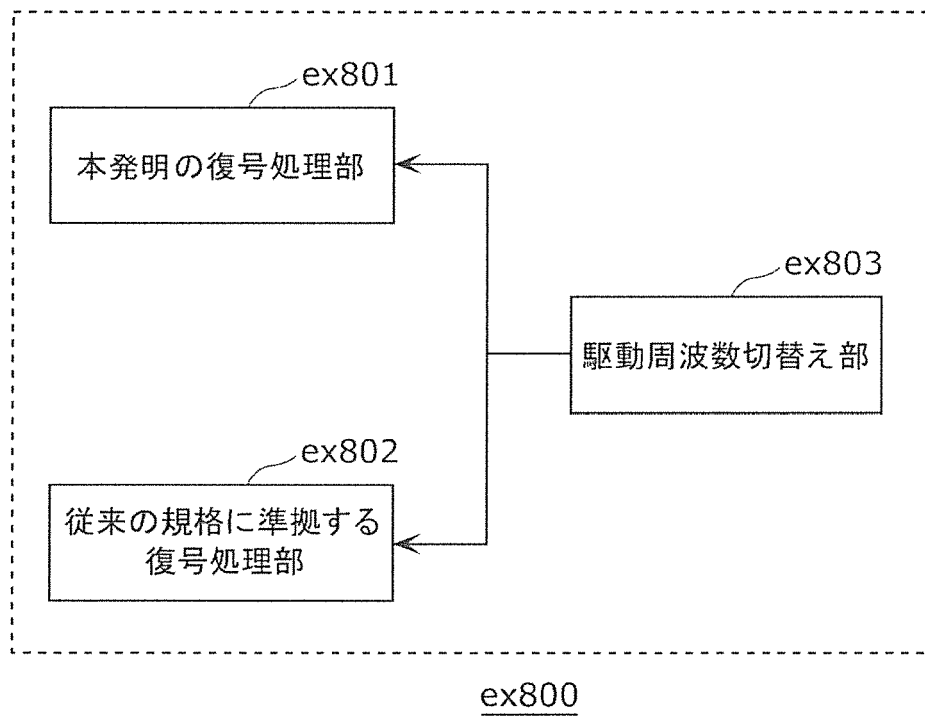
[図57]



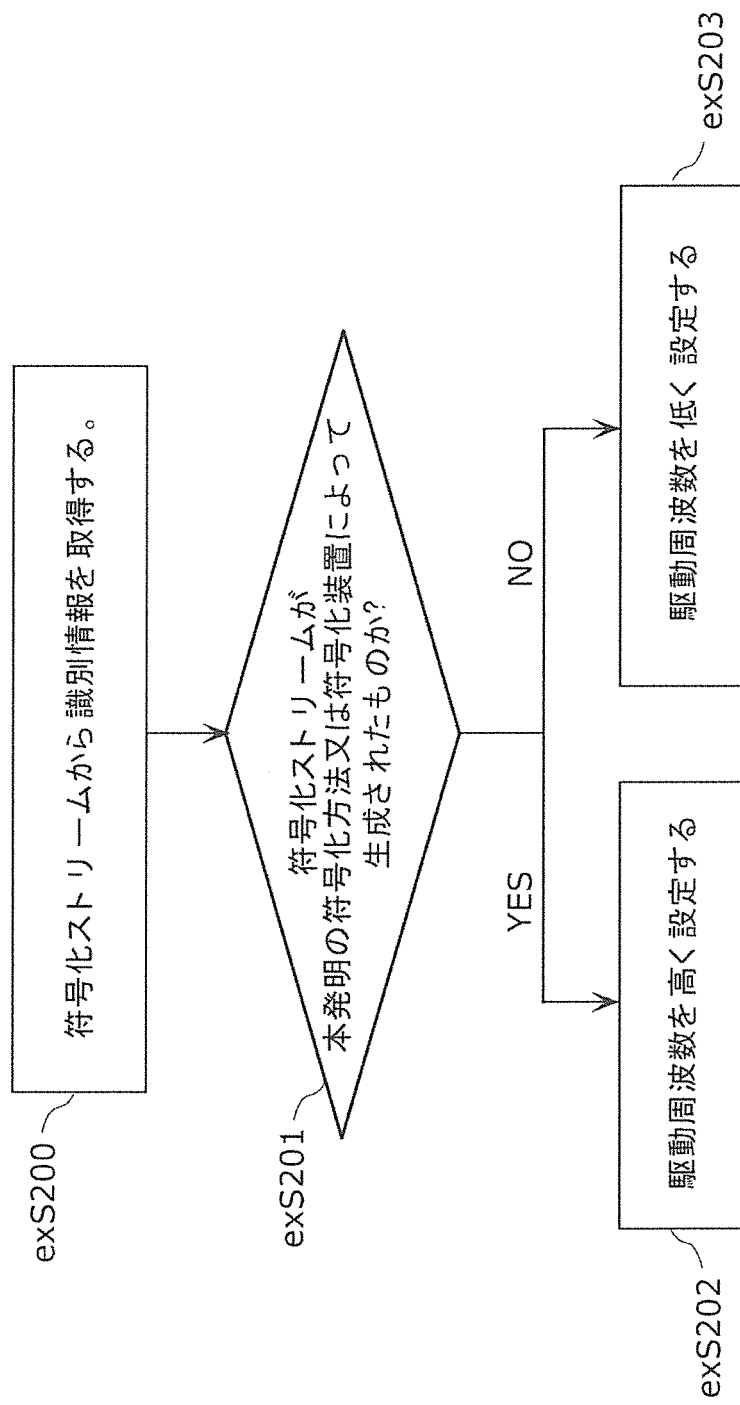
[図58]



[図59]



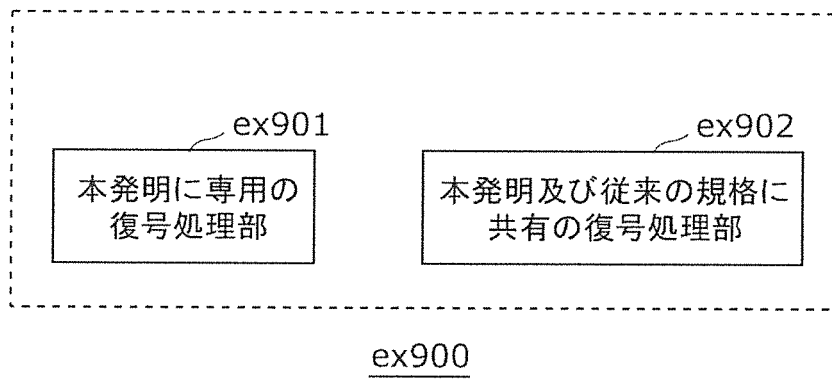
[図60]



[図61]

対応規格	駆動周波数
MPEG4.AVC	500MHz
MPEG2	350MHz
...	...

[図62A]



[図62B]

