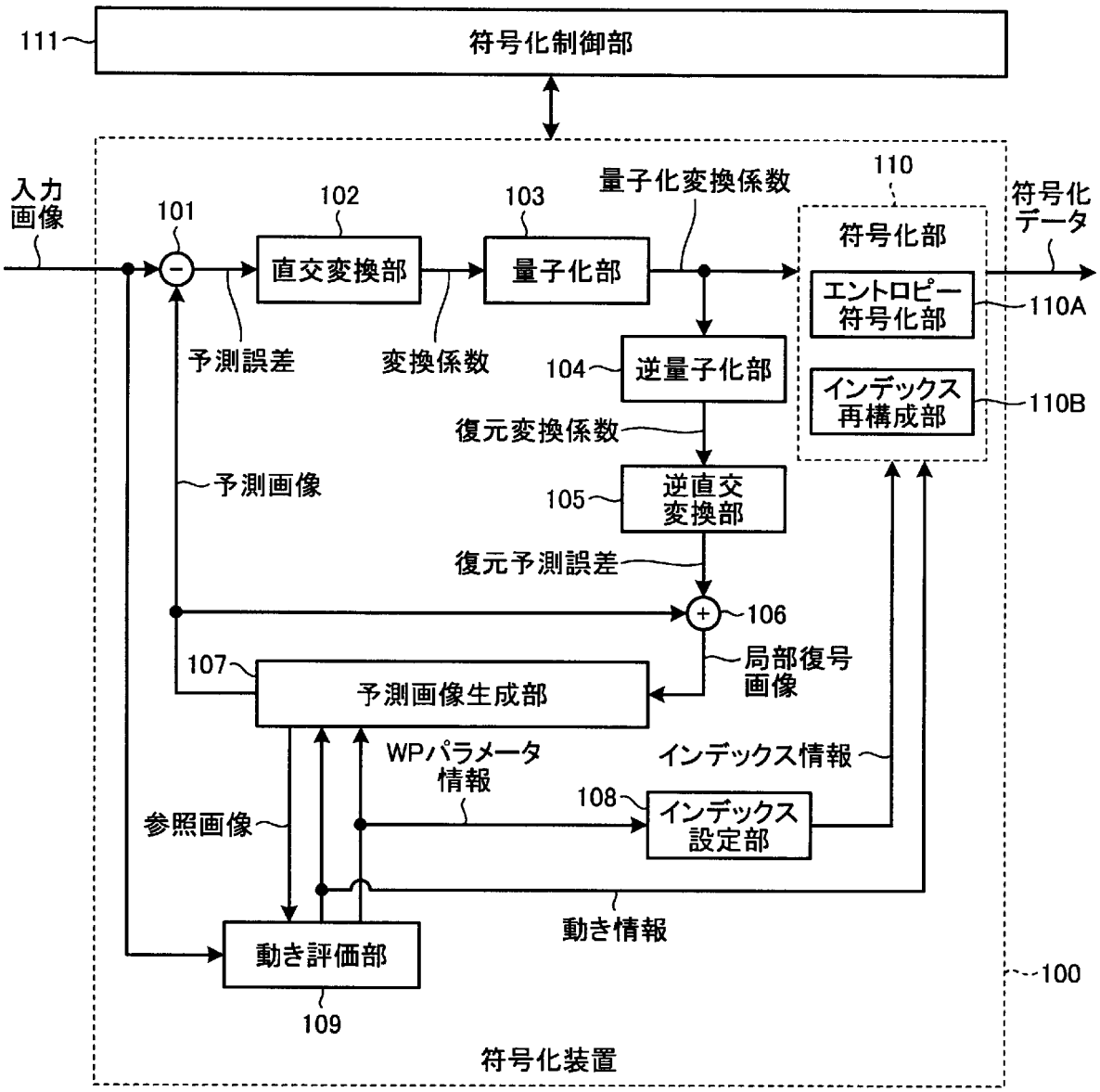
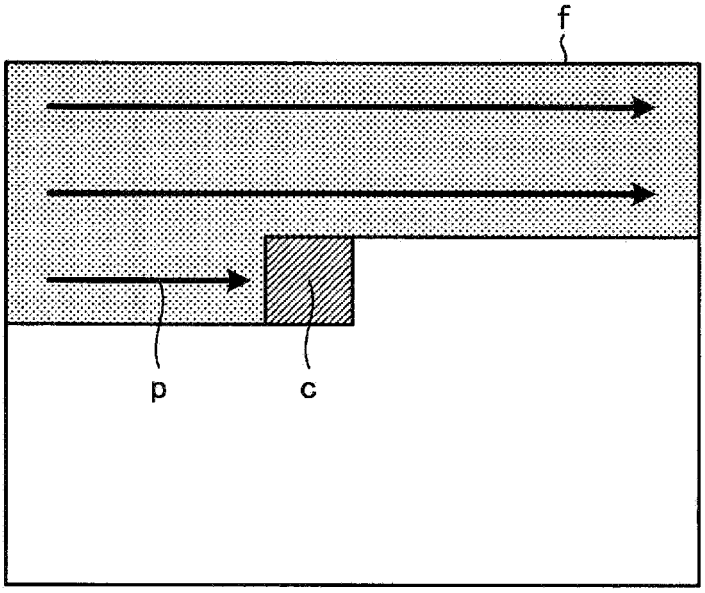
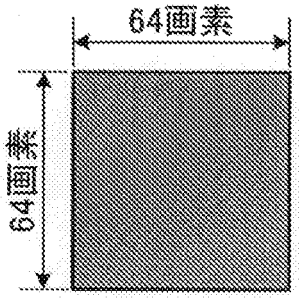
[図1]


111
符号化制御部
入力画像
101
102
直交変換部
103
量子化部
量子化変換係数
110
符号化部
符号化データ
エントロピー符号化部
110A
インデックス再構成部
110B
予測誤差
変換係数
104
逆量子化部
復元変換係数
予測画像
105
逆直交変換部
復元予測誤差
106
局部復号画像
107
予測画像生成部
WPパラメータ情報
インデックス情報
108
インデックス設定部
参照画像
動き情報
動き評価部
109
100
符号化装置

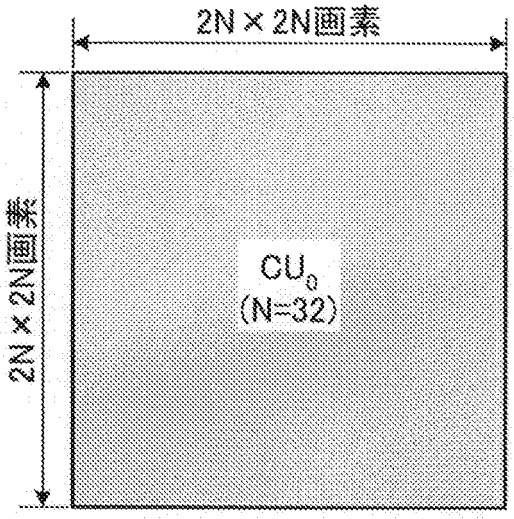
[図2]



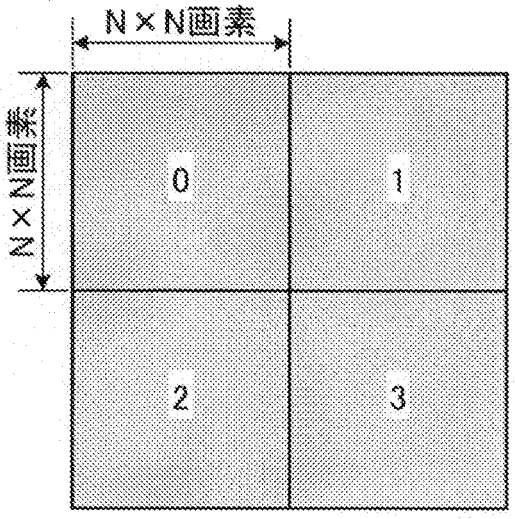
[図3A]



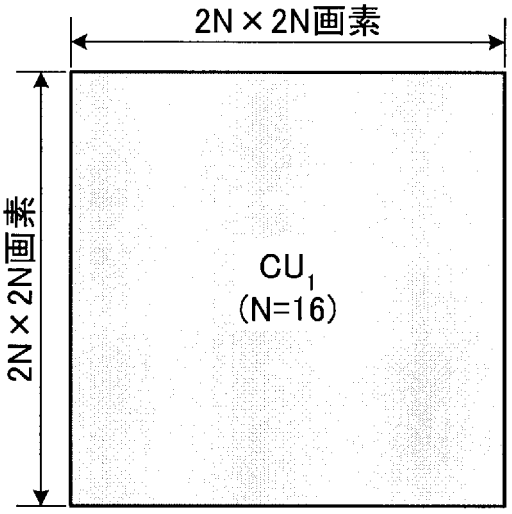
[図3B]



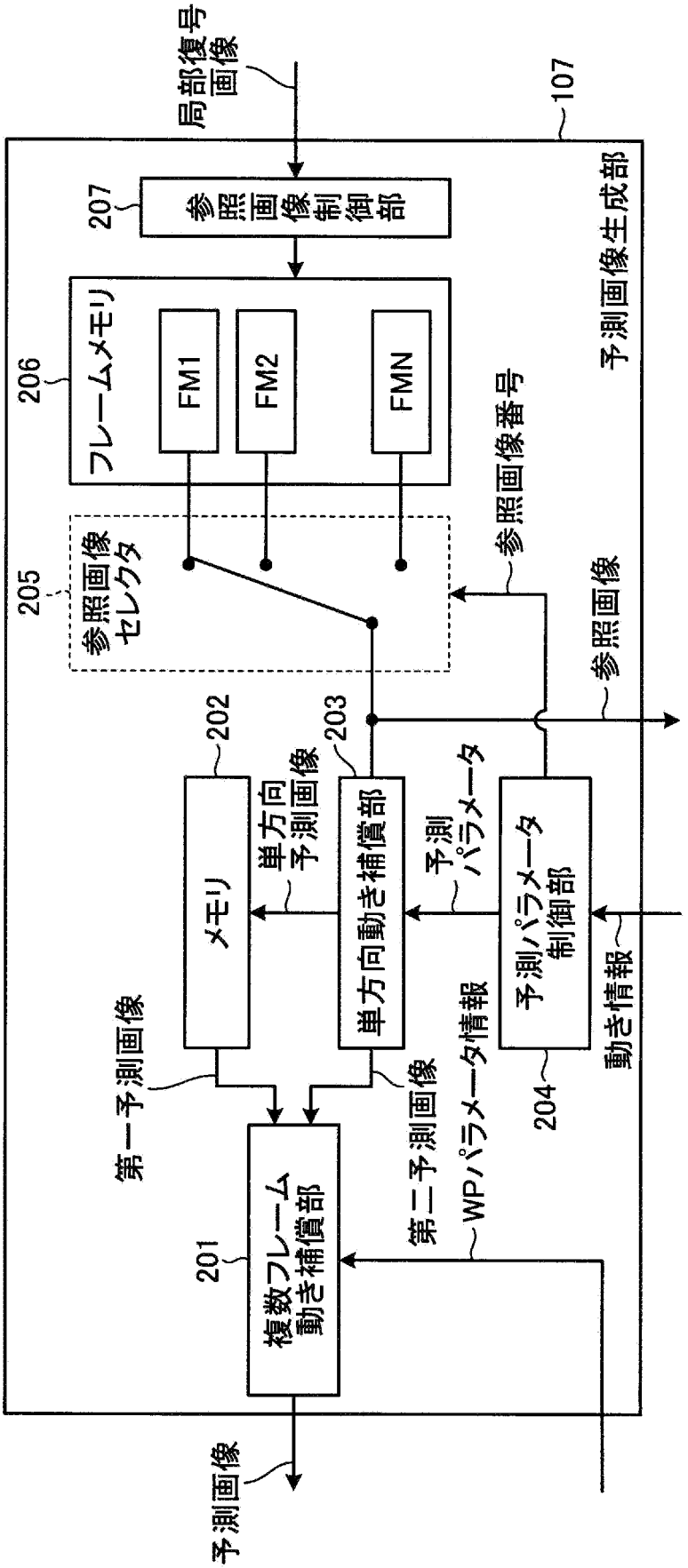
[図3C]



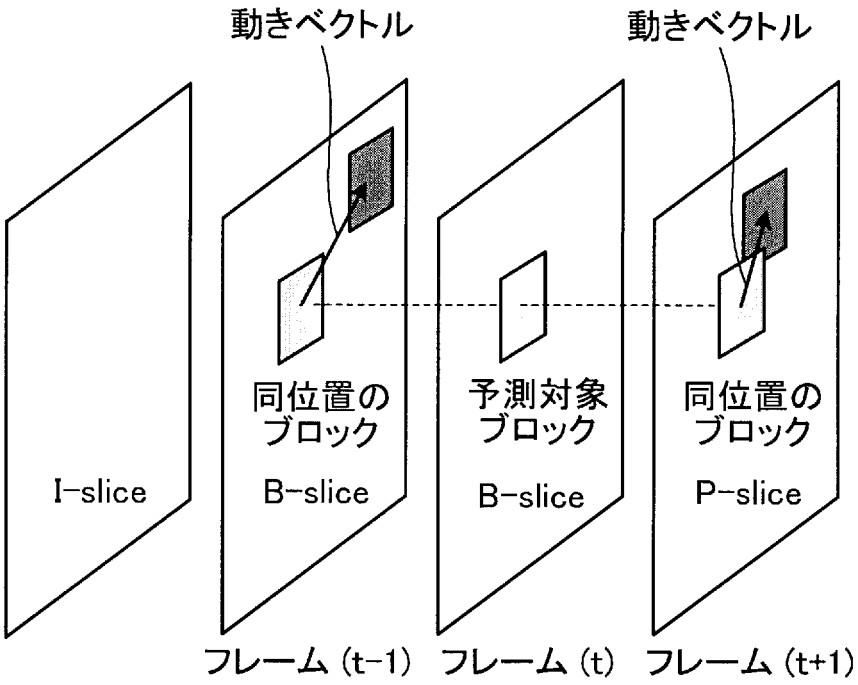
[図3D]



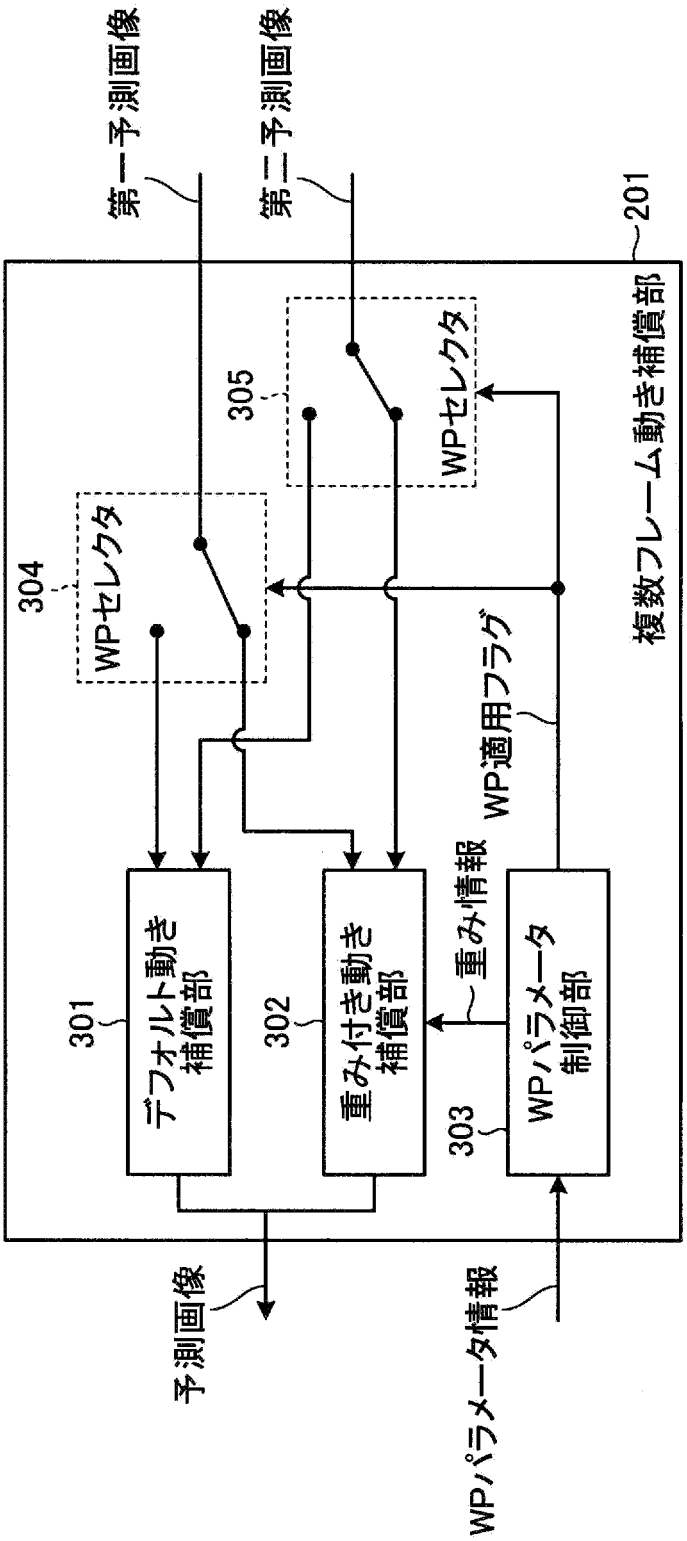
[図4]


局部復号画像
107
予測画像生成部
207
参照画像制御部
206
フレームメモリ
FM1
FM2
FMN
参照画像番号
205
参照画像セレクタ
参照画像
202
メモリ
単方向予測画像
203
単方向動き補償部
予測パラメータ
予測パラメータ制御部
動き情報
第一予測画像
第二予測画像
WPパラメータ情報
204
201
複数フレーム動き補償部
予測画像

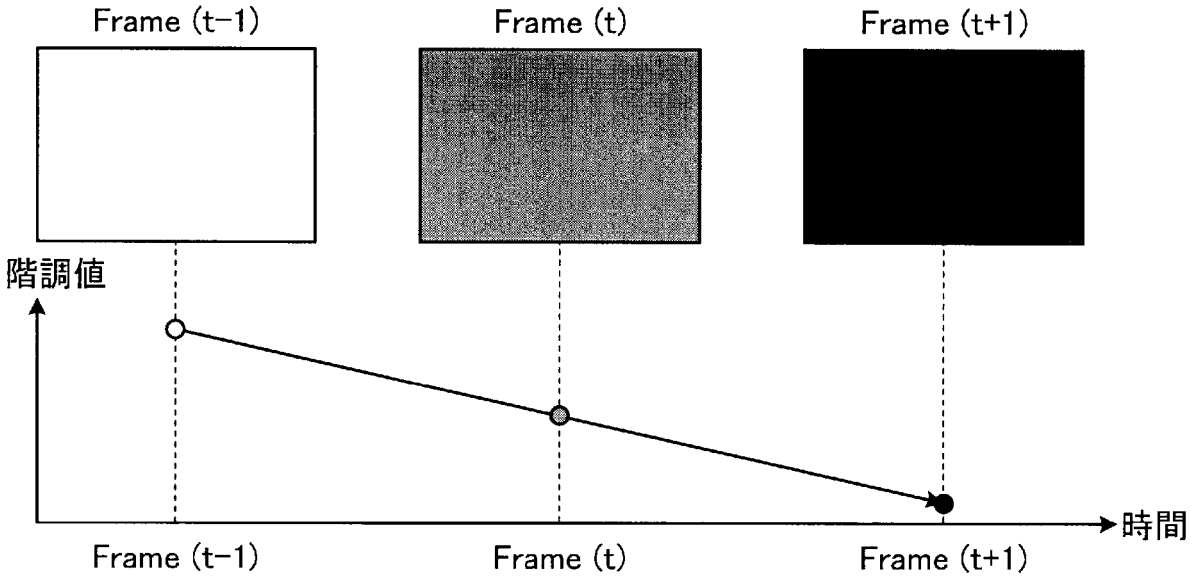
[図5]


動きベクトル
動きベクトル
同位置の
ブロック
予測対象
ブロック
同位置の
ブロック
I-slice
B-slice
B-slice
P-slice
フレーム (t-1)
フレーム (t)
フレーム (t+1)

[図6]


第一予測画像
第二予測画像
201
複数フレーム動き補償部
305
WPセレクタ
304
WPセレクタ
WP適用フラグ
重み情報
301
デフォルト動き補償部
302
重み付き動き補償部
303
WPパラメータ制御部
予測画像
WPパラメータ情報

[図7]



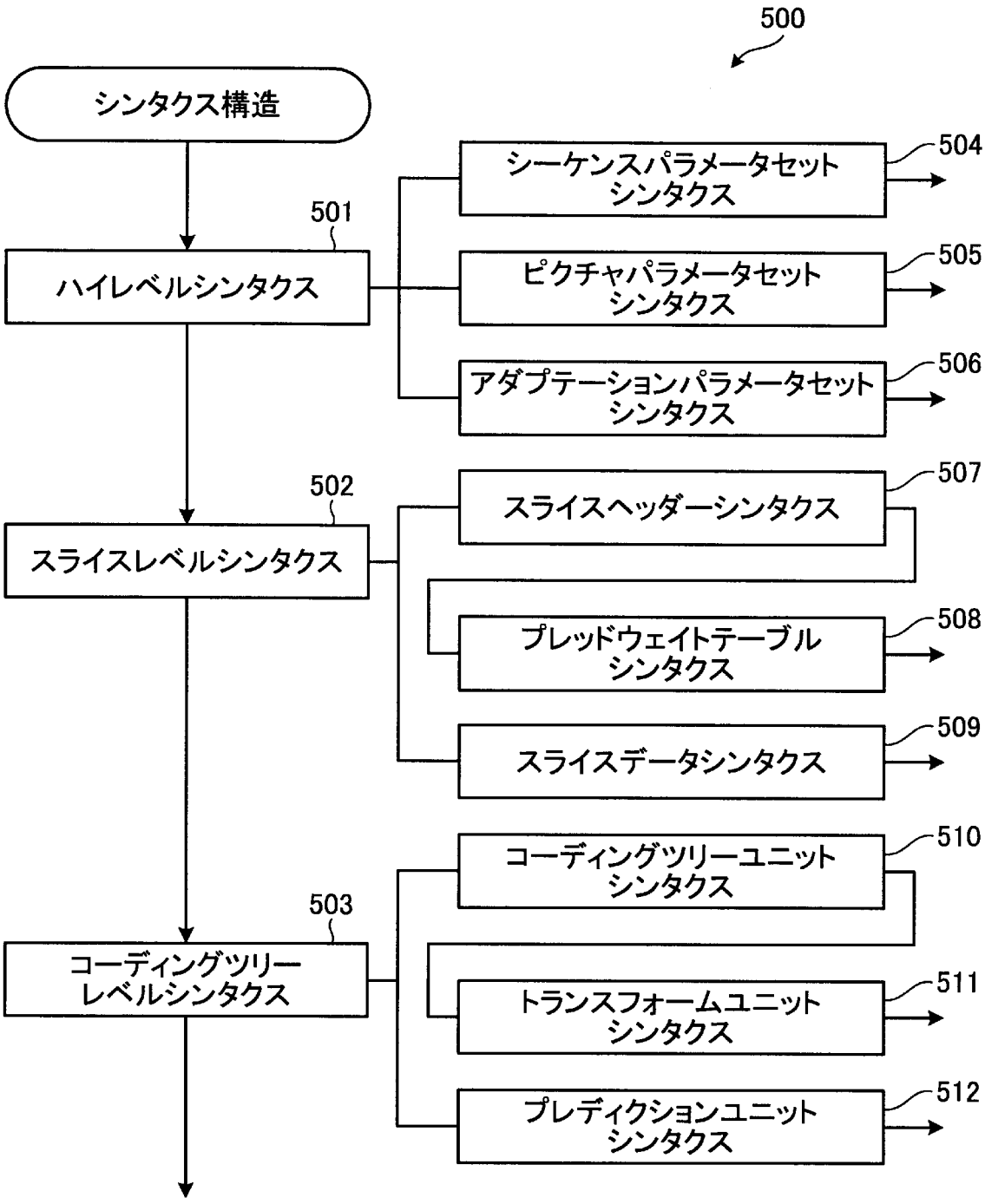
[図8A]

リスト番号	0						
参照番号	0	1	2	3	4	・・・	N
WP適用フラグ	WP_flag[0]	WP_flag[1]	WP_flag[2]	WP_flag[3]	WP_flag[4]	・・・	WP_flag[N]
重み係数	Weight[0]	Weight[1]	Weight[2]	Weight[3]	Weight[4]	・・・	Weight[N]
オフセット	Offset[0]	Offset[1]	Offset[2]	Offset[3]	Offset[4]	・・・	Offset[N]

[図8B]

リスト番号	1						
参照番号	0	1	2	3	4	・・・	N
WP適用フラグ	WP_flag[0]	WP_flag[1]	WP_flag[2]	WP_flag[3]	WP_flag[4]	・・・	WP_flag[N]
重み係数	Weight[0]	Weight[1]	Weight[2]	Weight[3]	Weight[4]	・・・	Weight[N]
オフセット	Offset[0]	Offset[1]	Offset[2]	Offset[3]	Offset[4]	・・・	Offset[N]

[図9]


500
シンタクス構造
501
ハイレベルシンタクス
502
スライスレベルシンタクス
503
コーディングツリーレベルシンタクス
シーケンスパラメータセットシンタクス
504
ピクチャパラメータセットシンタクス
505
アダプテーションパラメータセットシンタクス
506
スライスヘッダーシンタクス
507
プレッドウェイトテーブルシンタクス
508
スライスデータシンタクス
509
コーディングツリーユニットシンタクス
510
トランスフォームユニットシンタクス
511
プレディクションユニットシンタクス
512

[図10]

pic_parameter_set_rbsp( ) {	Descriptor
pic_parameter_set_id	u(8)
seq_parameter_set_id	u(8)
...	
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
weighted_pred_flag	u(1)
weighted_bipred_idc	u(2)
...	
rbsp_trailing_bits( )	
}	

[図11]

slice_header( ) {	Descriptor
...	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
...	
if(slice_type == P || slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
...	
if((weighted_pred_flag && (slice_type == P)) || (weighted_bipred_idc == 1 && slice_type == B))	
pred_weight_table( )	
...	
}	

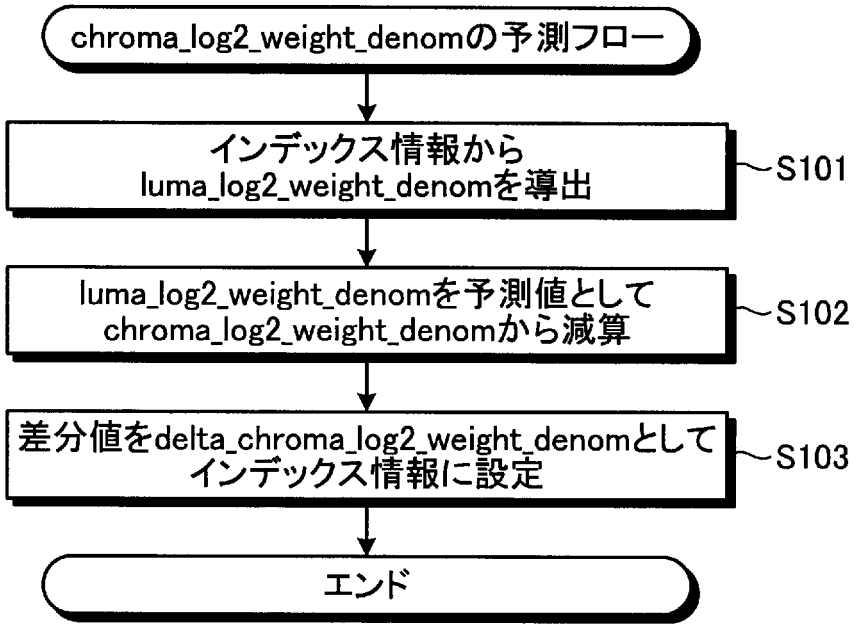
[図12]

pred_weight_table( ) {	Descriptor
luma_log2_weight_denom	ue(v)
if(chroma_format_idc != MONO_IDX)	
chroma_log2_weight_denom	ue(v)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {	
luma_weight_l0_flag	u(1)
if(luma_weight_l0_flag) {	
luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if (chroma_format_idc != MONO_IDX) {	
chroma_weight_l0_flag	u(1)
if(chroma_weight_l0_flag)	
for(j =0; j < COLOR_COMP − 1; j++) {	
chroma_weight_l0[i][j]	se(v)
chroma_offset_l0[i][j]	se(v)
}	
}	
}	
if(slice_type = = B_SLICE)	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {	
luma_weight_l1_flag	u(1)
if(luma_weight_l1_flag) {	
luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_format_idc != MONO_IDX) {	
chroma_weight_l1_flag	u(1)
if(chroma_weight_l1_flag) {	
for(j = 0; j < COLOR_COMP − 1; j++) {	
chroma_weight_l1[i][j]	se(v)
chroma_offset_l1[i][j]	se(v)
}	
}	
}	
}	
}	
}	

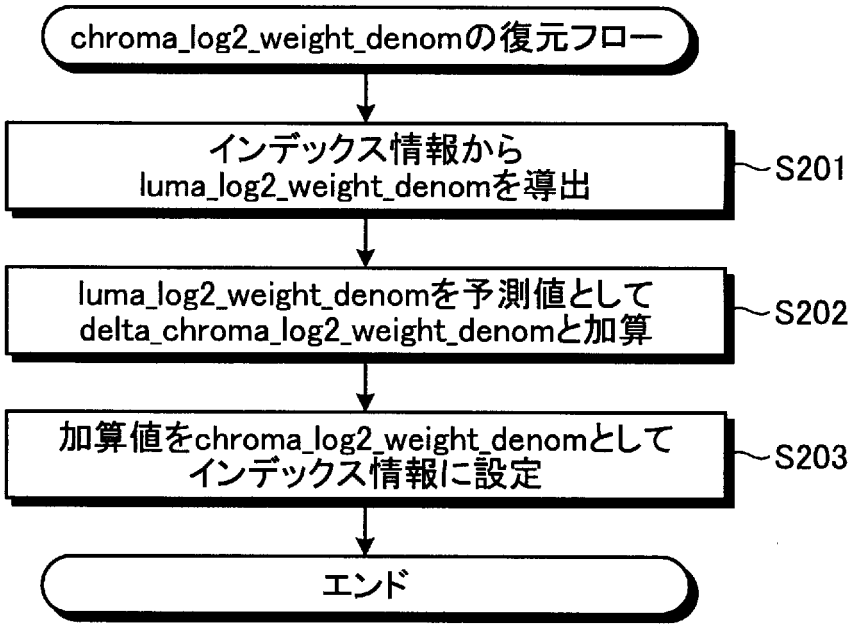
[図13]

pred_weight_table( ) {	Descriptor
luma_log2_weight_denom	ue(v)
if(chroma_format_idc != MONO_IDX)	
delta_chroma_log2_weight_denom	ue(v)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {	
luma_weight_l0_flag	u(1)
if(luma_weight_l0_flag) {	
delta_luma_weight_l0[i]	se(v)
delta_luma_offset_l0[i]	se(v)
}	
if (chroma_format_idc != MONO_IDX) {	
chroma_weight_l0_flag	u(1)
if(chroma_weight_l0_flag)	
for(j =0; j < COLOR_COMP − 1; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
}	
if(slice_type = = B_SLICE)	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {	
luma_weight_l1_flag	u(1)
if(luma_weight_l1_flag) {	
delta_luma_weight_l1[i]	se(v)
delta_luma_offset_l1[i]	se(v)
}	
if(chroma_format_idc != MONO_IDX) {	
chroma_weight_l1_flag	u(1)
if(chroma_weight_l1_flag) {	
for(j = 0; j < COLOR_COMP − 1; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	
}	
}	
}	

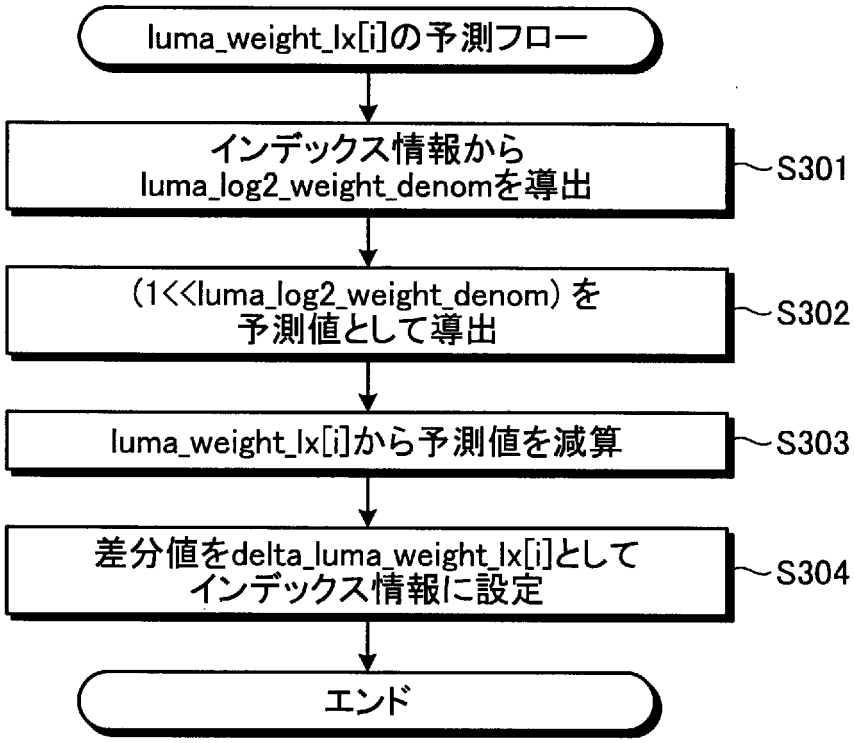
[図14]


chroma_log2_weight_denomの予測フロー
インデックス情報から
luma_log2_weight_denomを導出
S101
luma_log2_weight_denomを予測値として
chroma_log2_weight_denomから減算
S102
差分値をdelta_chroma_log2_weight_denomとして
インデックス情報に設定
S103
エンド

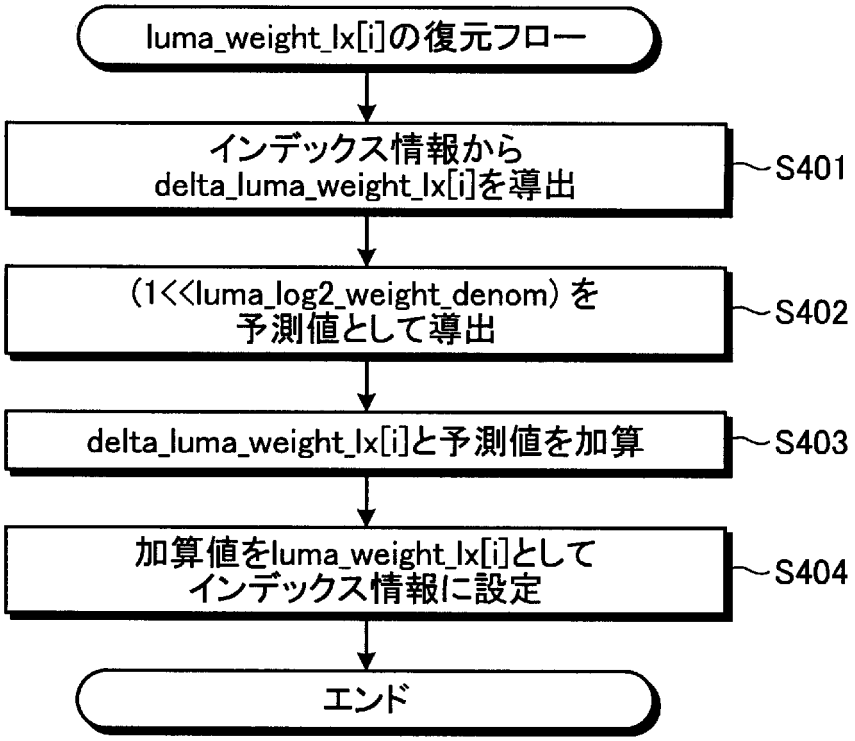
[図15]


chroma_log2_weight_denomの復元フロー
インデックス情報から
luma_log2_weight_denomを導出
S201
luma_log2_weight_denomを予測値として
delta_chroma_log2_weight_denomと加算
S202
加算値をchroma_log2_weight_denomとして
インデックス情報に設定
S203
エンド

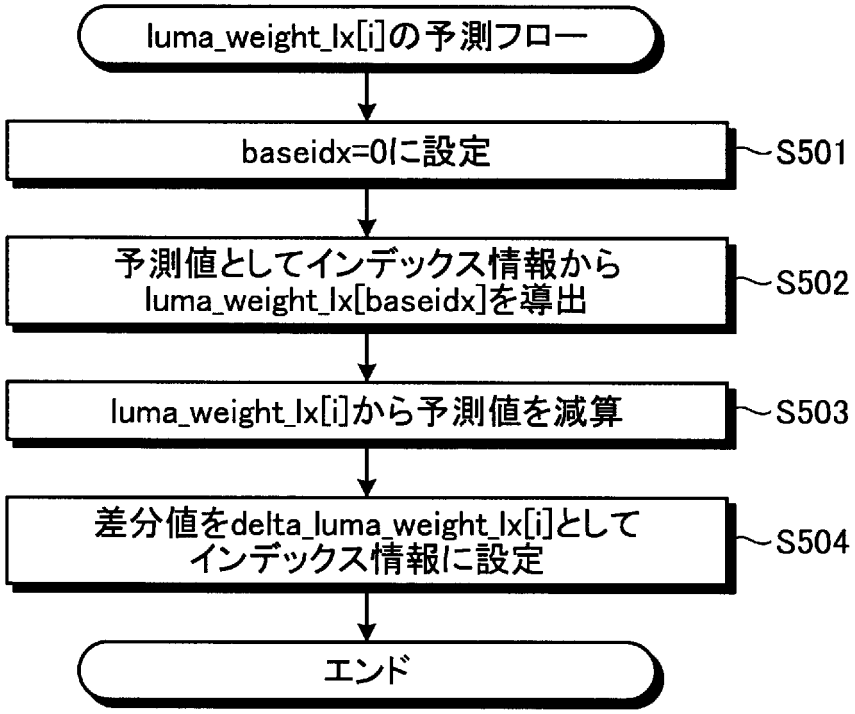
[図16]


luma_weight_lx[i]の予測フロー
インデックス情報から
luma_log2_weight_denomを導出
S301
(1<<luma_log2_weight_denom) を
予測値として導出
S302
luma_weight_lx[i]から予測値を減算
S303
差分値をdelta_luma_weight_lx[i]として
インデックス情報に設定
S304
エンド

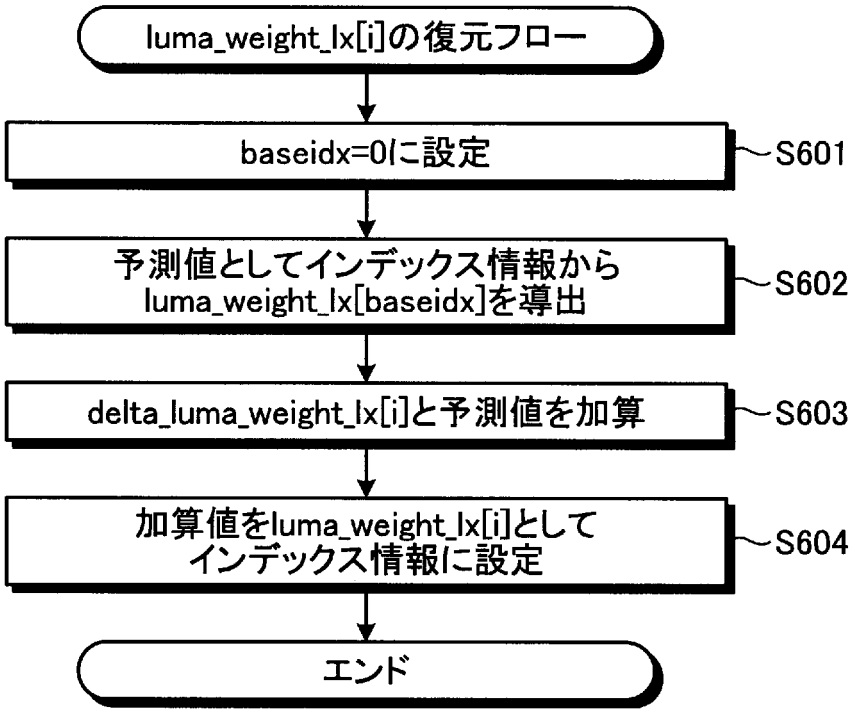
[図17]


luma_weight_lx[i]の復元フロー
インデックス情報から
delta_luma_weight_lx[i]を導出
S401
(1<<luma_log2_weight_denom) を
予測値として導出
S402
delta_luma_weight_lx[i]と予測値を加算
S403
加算値をluma_weight_lx[i]として
インデックス情報に設定
S404
エンド

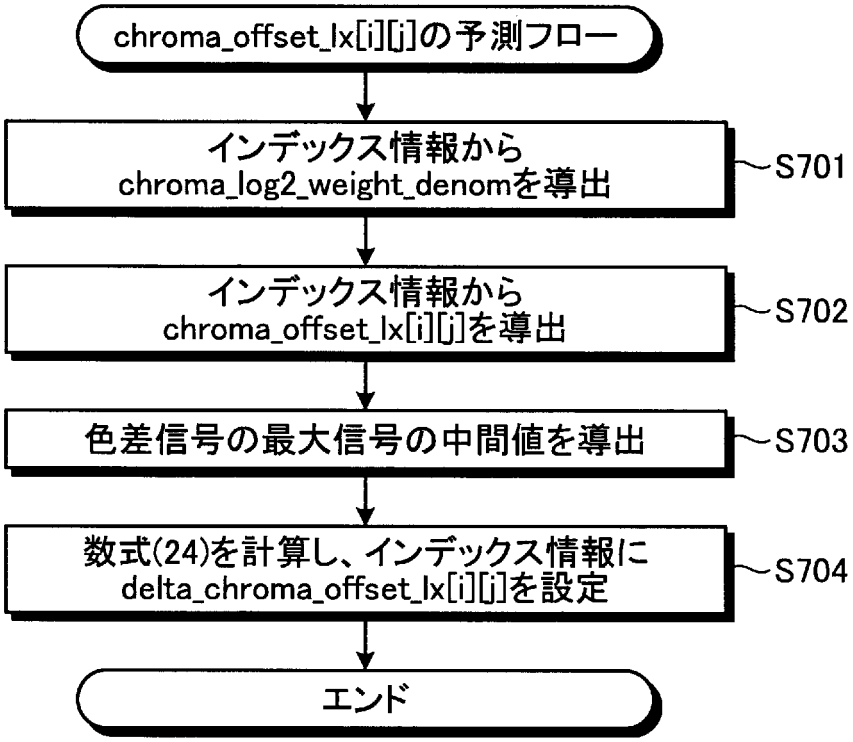
[図18]


luma_weight_lx[i]の予測フロー
baseidx=0に設定
S501
予測値としてインデックス情報から
luma_weight_lx[baseidx]を導出
S502
luma_weight_lx[i]から予測値を減算
S503
差分値をdelta_luma_weight_lx[i]として
インデックス情報に設定
S504
エンド

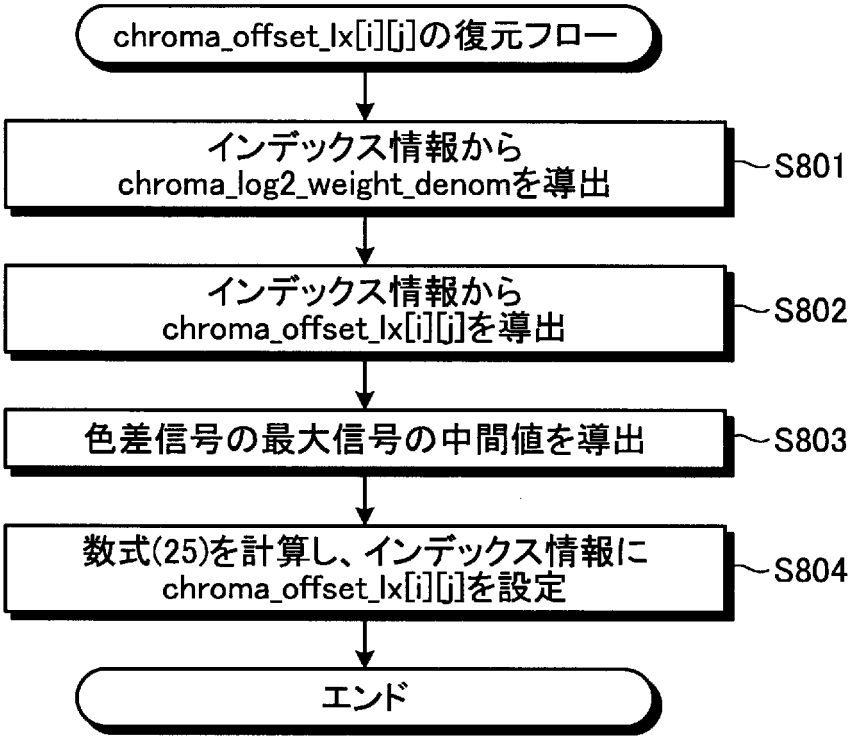
[図19]


luma_weight_lx[i]の復元フロー
baseidx=0に設定
S601
予測値としてインデックス情報から
luma_weight_lx[baseidx]を導出
S602
delta_luma_weight_lx[i]と予測値を加算
S603
加算値をluma_weight_lx[i]として
インデックス情報に設定
S604
エンド

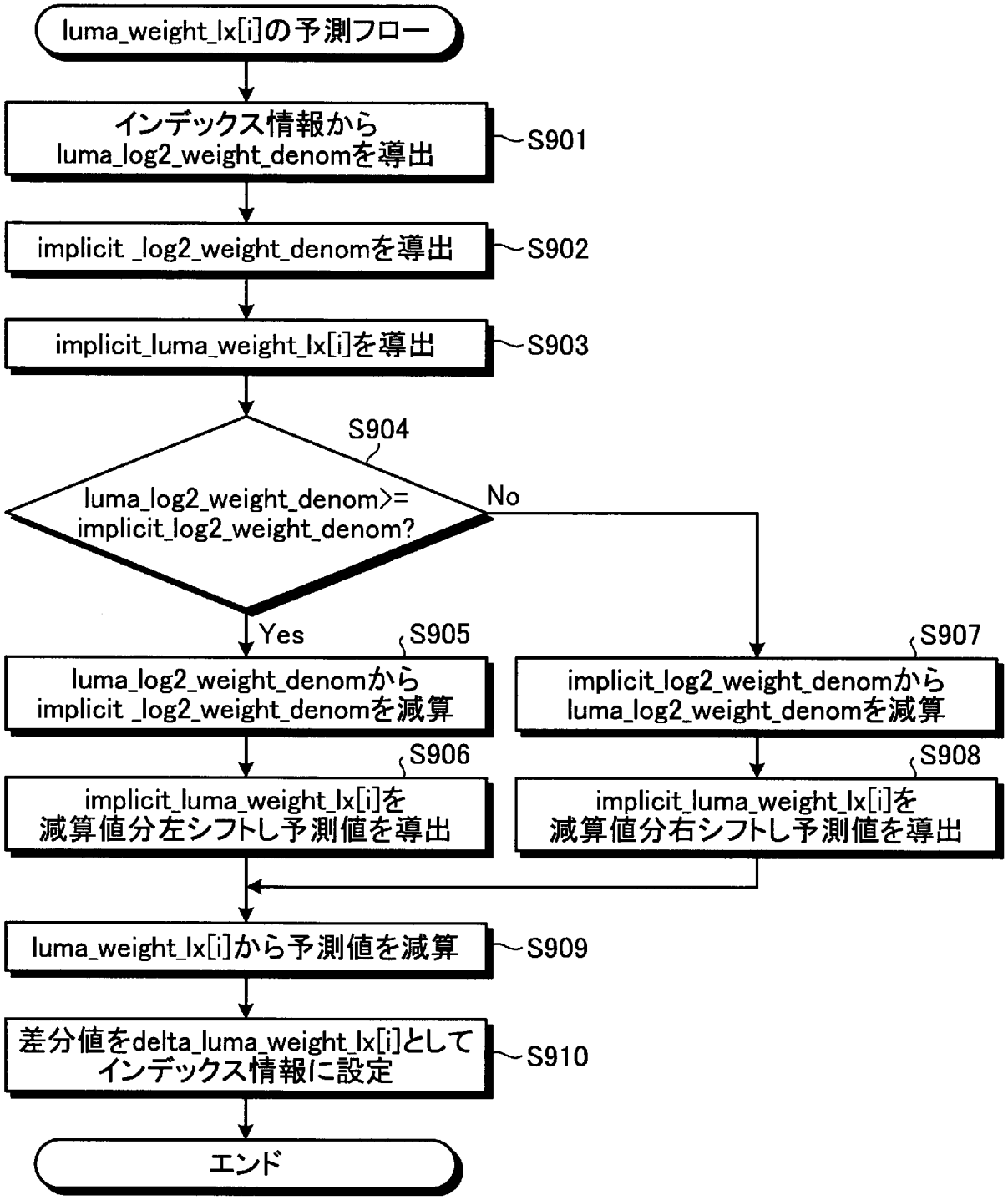
[図20]


chroma_offset_lx[i][j]の予測フロー
インデックス情報から
chroma_log2_weight_denomを導出
S701
インデックス情報から
chroma_offset_lx[i][j]を導出
S702
色差信号の最大信号の中間値を導出
S703
数式(24)を計算し、インデックス情報に
delta_chroma_offset_lx[i][j]を設定
S704
エンド

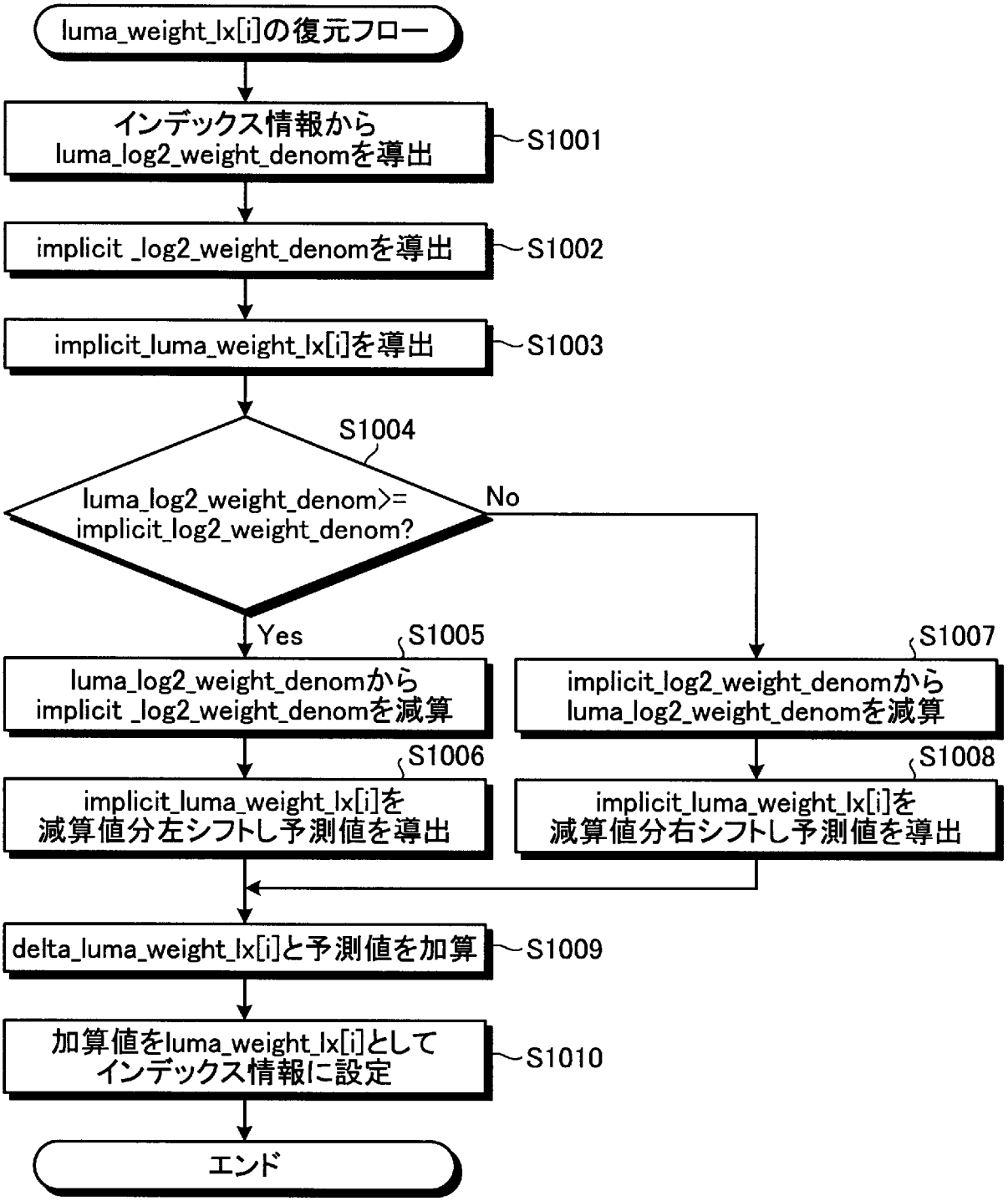
[図21]


chroma_offset_lx[i][j]の復元フロー
インデックス情報から
chroma_log2_weight_denomを導出
S801
インデックス情報から
chroma_offset_lx[i][j]を導出
S802
色差信号の最大信号の中間値を導出
S803
数式(25)を計算し、インデックス情報に
chroma_offset_lx[i][j]を設定
S804
エンド

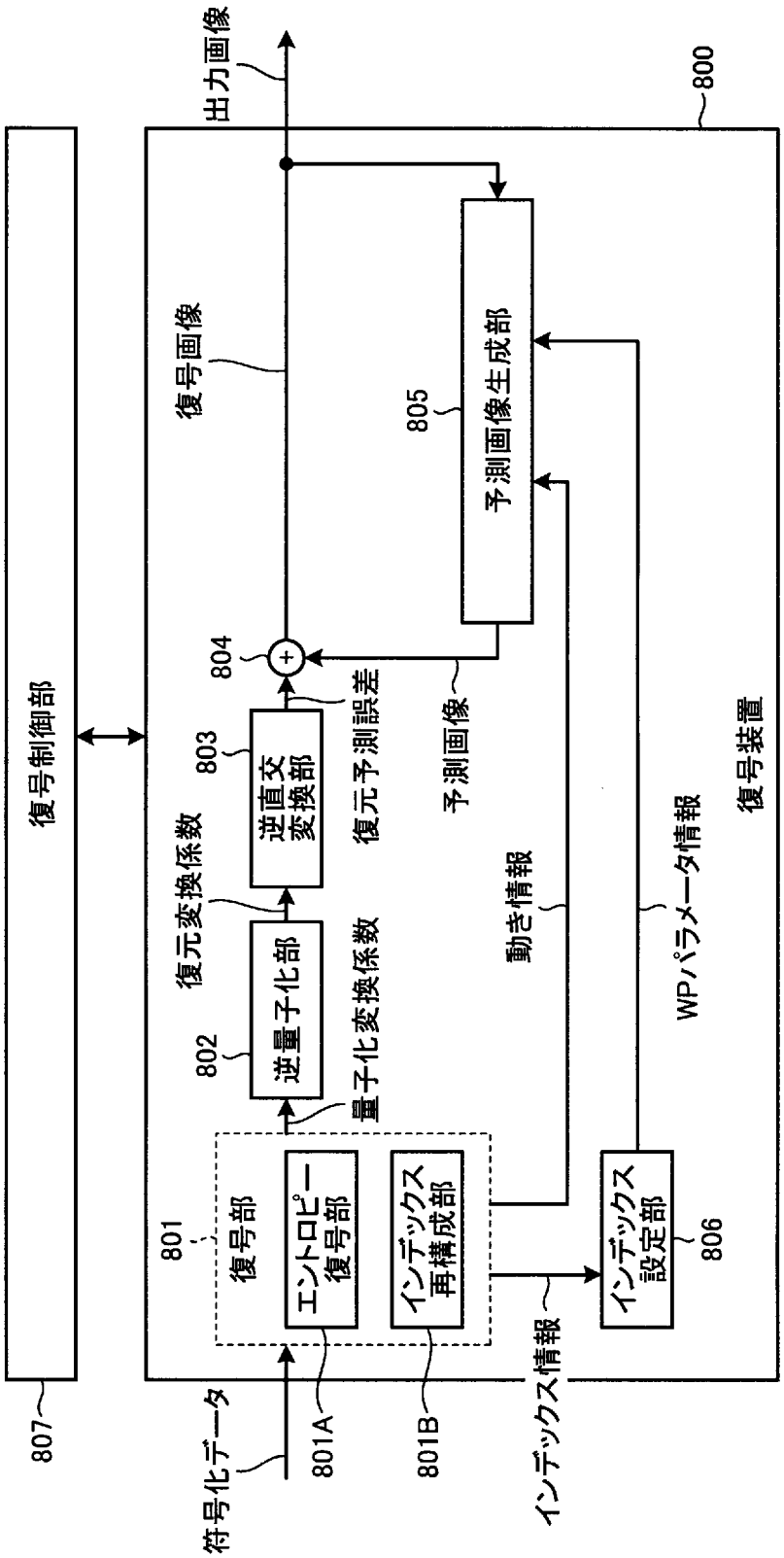
[図22]


luma_weight_lx[i]の予測フロー
インデックス情報から
luma_log2_weight_denomを導出
S901
implicit _log2_weight_denomを導出
S902
implicit_luma_weight_lx[i]を導出
S903
S904
luma_log2_weight_denom>=
implicit_log2_weight_denom?
No
Yes
S905
luma_log2_weight_denomから
implicit _log2_weight_denomを減算
S906
implicit_luma_weight_lx[i]を
減算値分左シフトし予測値を導出
S907
implicit_log2_weight_denomから
luma_log2_weight_denomを減算
S908
implicit_luma_weight_lx[i]を
減算値分右シフトし予測値を導出
luma_weight_lx[i]から予測値を減算
S909
差分値をdelta_luma_weight_lx[i]として
インデックス情報に設定
S910
エンド

[図23]


luma_weight_lx[i]の復元フロー
インデックス情報から
luma_log2_weight_denomを導出
S1001
implicit _log2_weight_denomを導出
S1002
implicit_luma_weight_lx[i]を導出
S1003
S1004
luma_log2_weight_denom>=
implicit_log2_weight_denom?
No
Yes
S1005
luma_log2_weight_denomから
implicit _log2_weight_denomを減算
S1006
implicit_luma_weight_lx[i]を
減算値分左シフトし予測値を導出
S1007
implicit_log2_weight_denomから
luma_log2_weight_denomを減算
S1008
implicit_luma_weight_lx[i]を
減算値分右シフトし予測値を導出
delta_luma_weight_lx[i]と予測値を加算
S1009
加算値をluma_weight_lx[i]として
インデックス情報に設定
S1010
エンド

[図24]


出力画像
800
復号制御部
復号画像
805
予測画像生成部
804
803
逆直交変換部
復元予測誤差
予測画像
復元変換係数
802
逆量子化部
量子化変換係数
動き情報
WPパラメータ情報
復号装置
801
復号部
エントロピー復号部
インデックス再構成部
インデックス設定部
806
インデックス情報
807
符号化データ
801A
801B