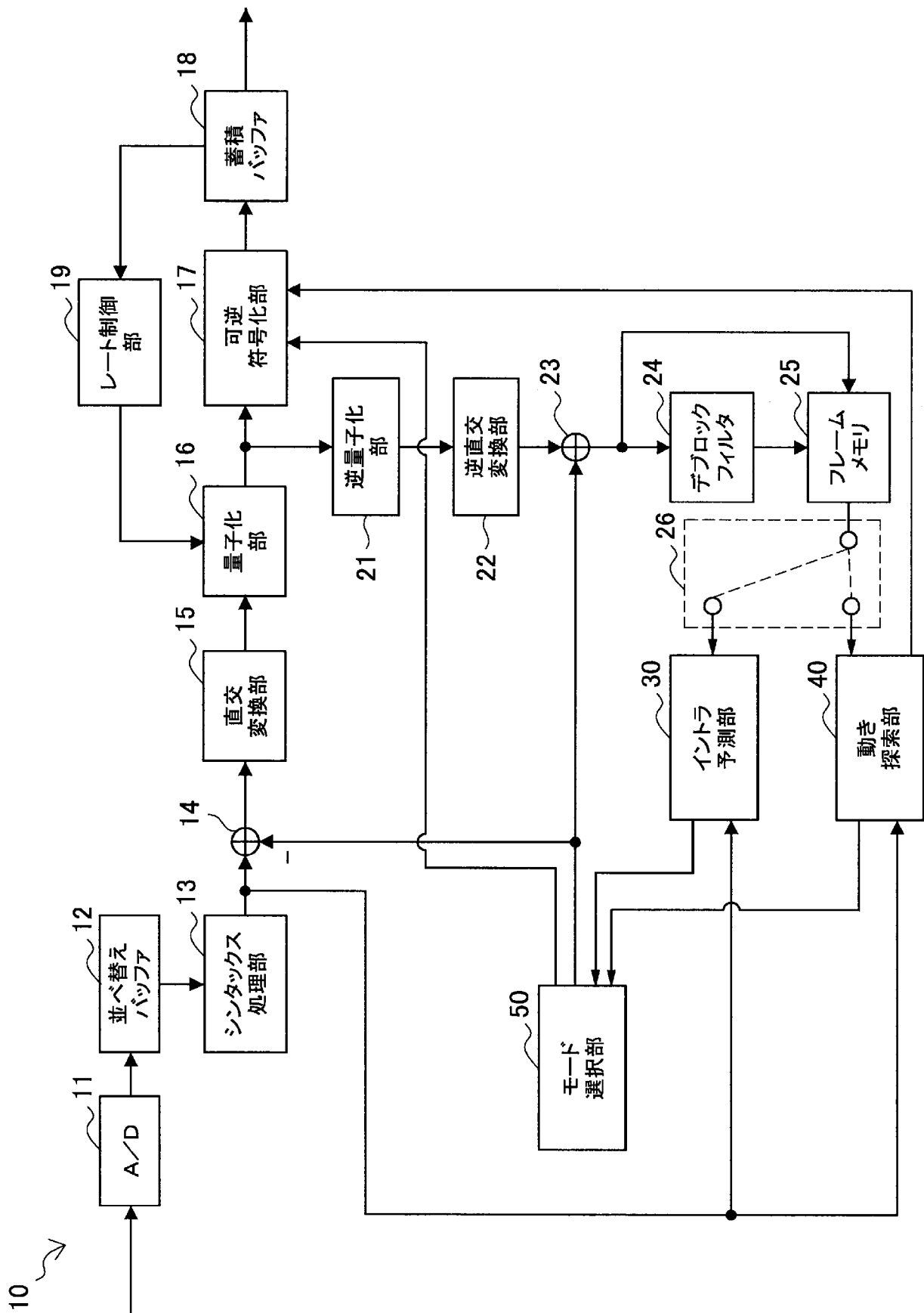
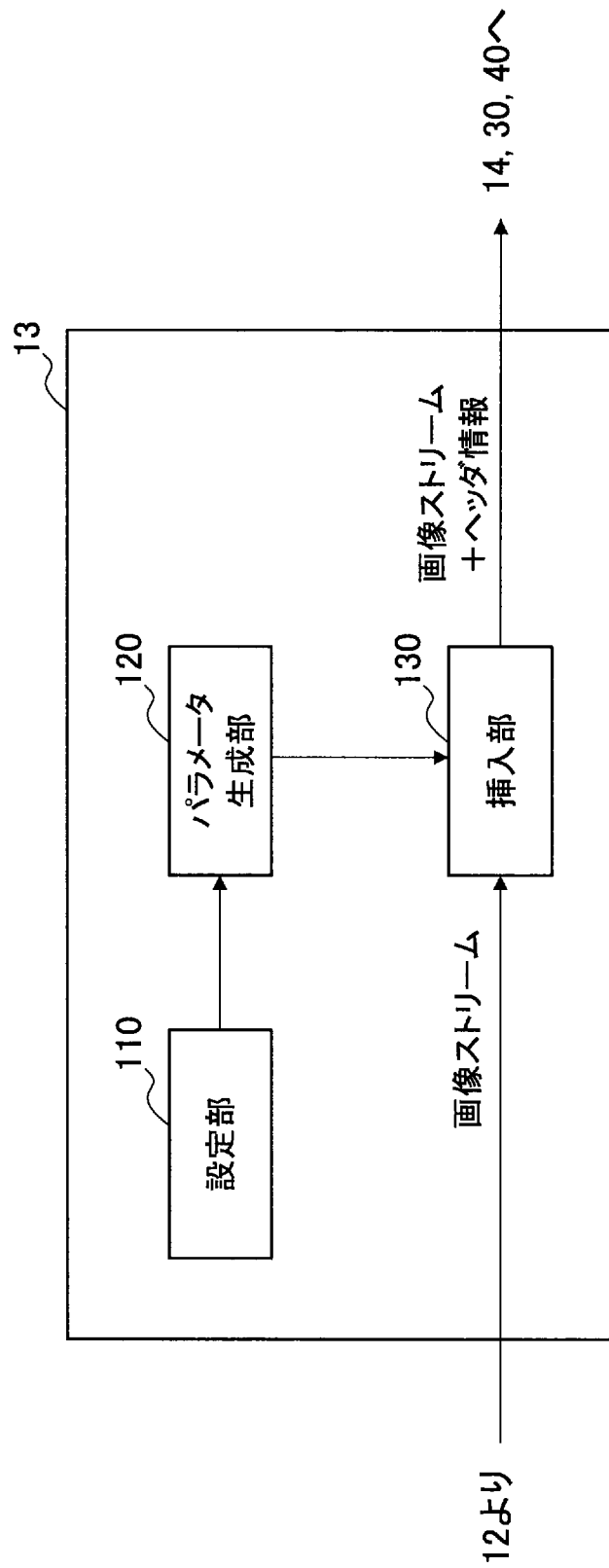


[図1]



[図2]



[図3]

パラメータ名	説明
QMPS ID	各QMPSを互いに識別するための識別子
生成モード存在 フラグ	生成モードの区分(pred_mode)が存在するか を示すフラグ
生成モード <pred_mode>	量子化行列生成処理のモードを表す区分 e.g.) 0:コピー, 1:軸指定, 2:全スキャン
コピーモードの場合(pred_mode = '0')	
ソースID	コピー元のQMPSを指定するQMPS ID ※既定の行列を使用する場合は自己のIDを指定
コピー元サイズ	コピー元の量子化行列のサイズ
コピー元タイプ	コピー元の量子化行列のタイプ
残差フラグ	残差の存在を表すフラグ
残差データ	残差行列を生成するためのデータ
軸指定モードの場合(pred_mode = '1')	
指定方式 フラグ	基準軸の指定方式を表すフラグ e.g.) 0:差分, 1:補間
基準軸データ	差分方式の場合の基準軸の要素のデータ
コーナーデータ	補間方式の場合のコーナーの要素のデータ
残差フラグ	残差の存在を表すフラグ
残差データ	残差行列を生成するためのデータ
全スキャンモードの場合(pred_mode = '2')	
差分データ	量子化行列の各要素について DPCM方式で算出された差分値の一次元配列

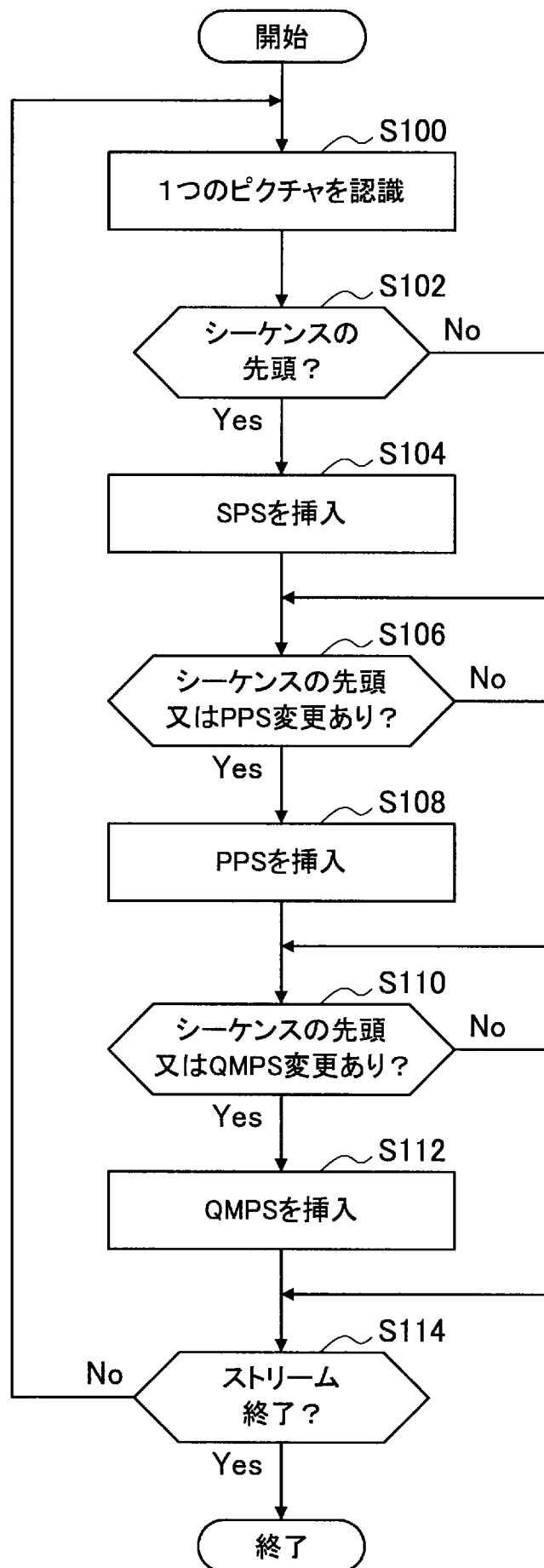
量子化行列パラメータセット

[図4]

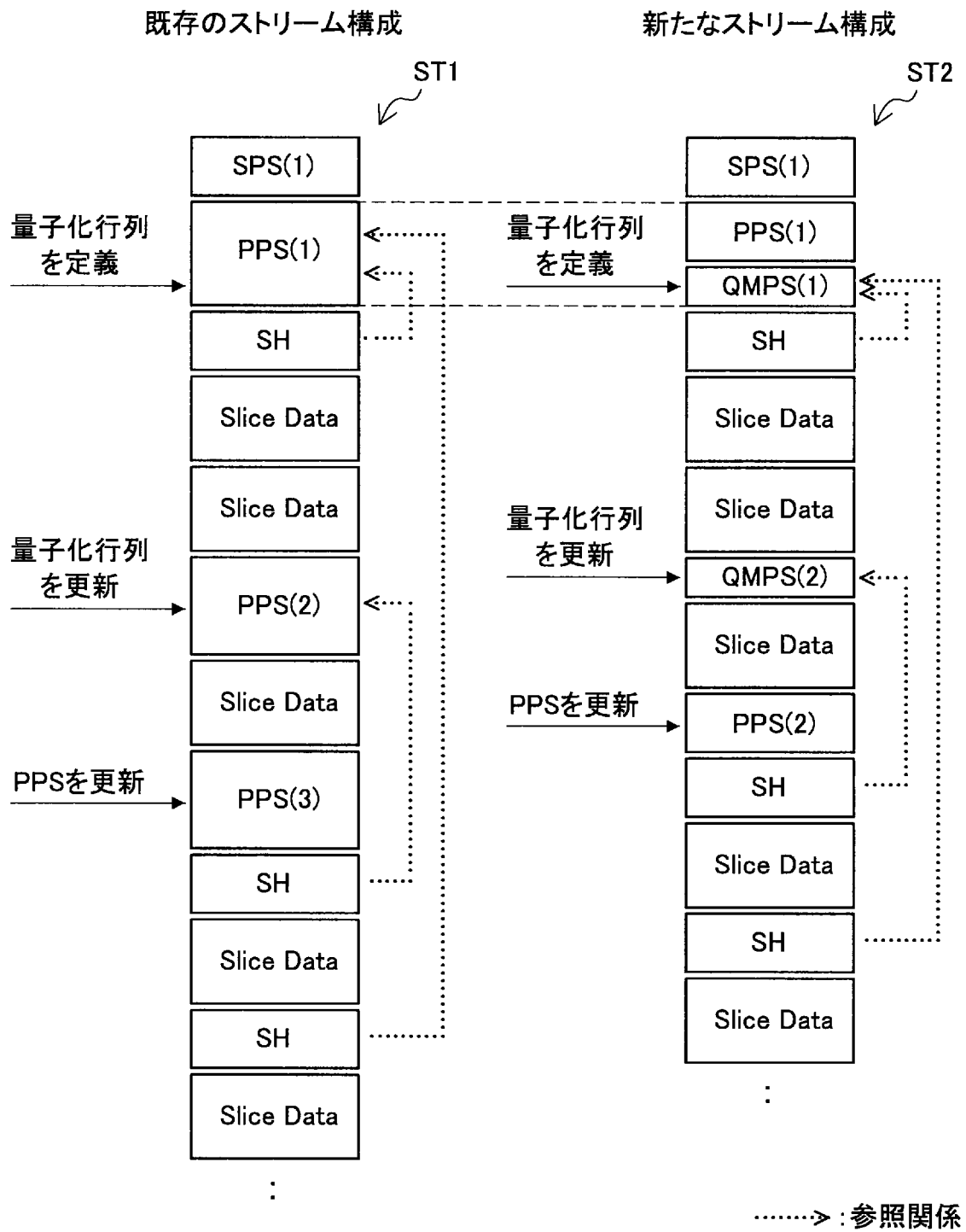
パラメータ名	説明
:	
スライスタイプ	スライスの符号化タイプ
PPS ID	当該スライスのために参照されるPPSのID
:	
QMPS ID存在フラグ	当該ヘッダ内にQMPS IDが存在するかを示すフラグ
QMPS ID	当該スライスのために参照されるQMPSのID
:	

スライスヘッダ

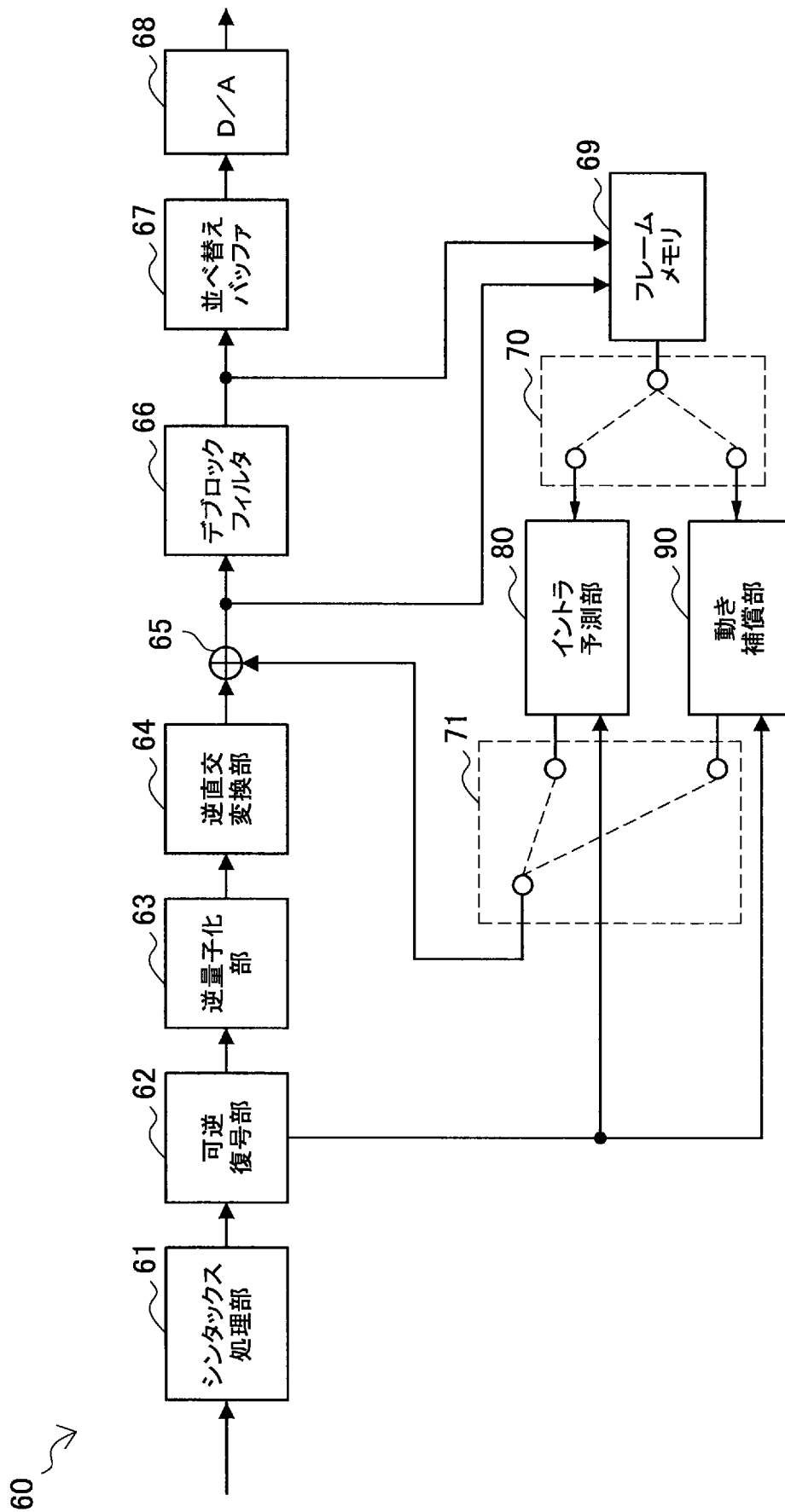
[図5]



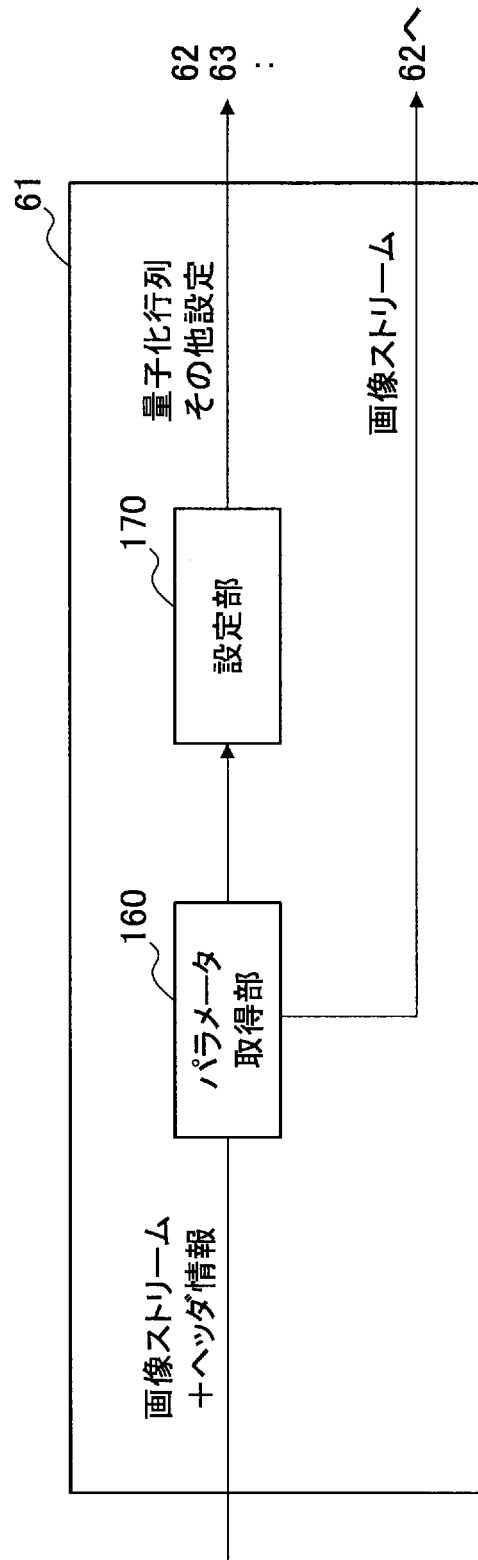
[図6]



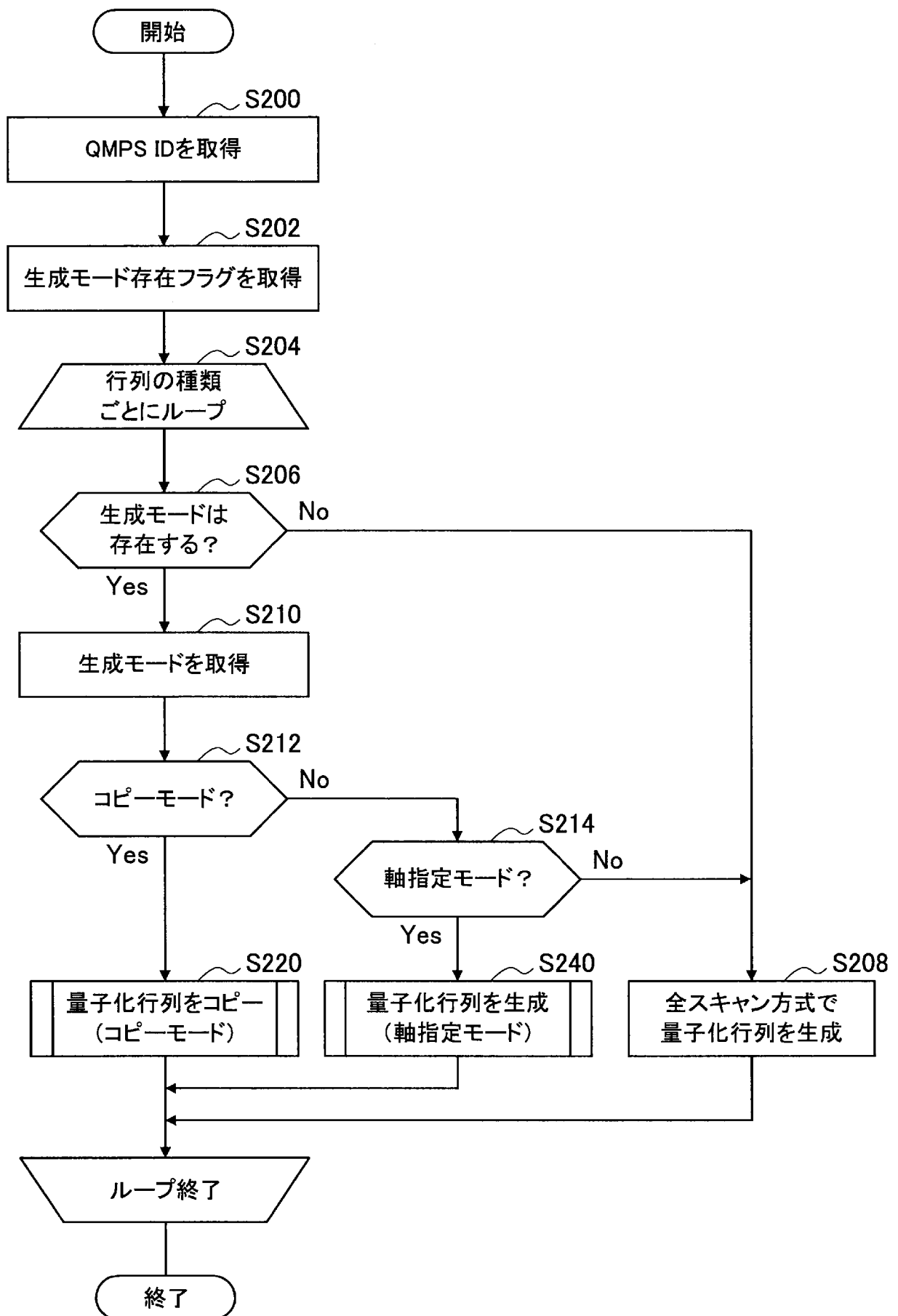
[図7]



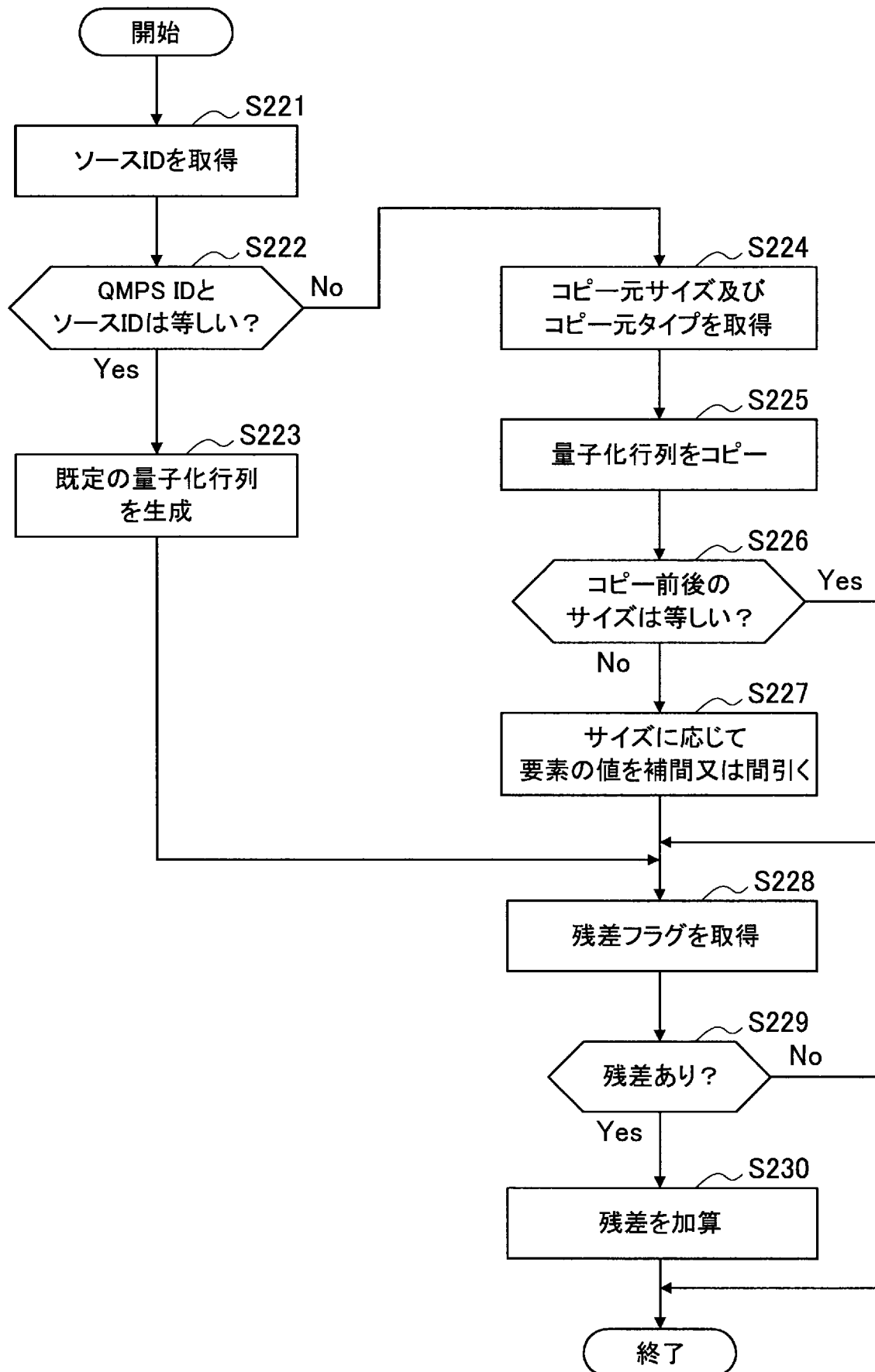
[図8]



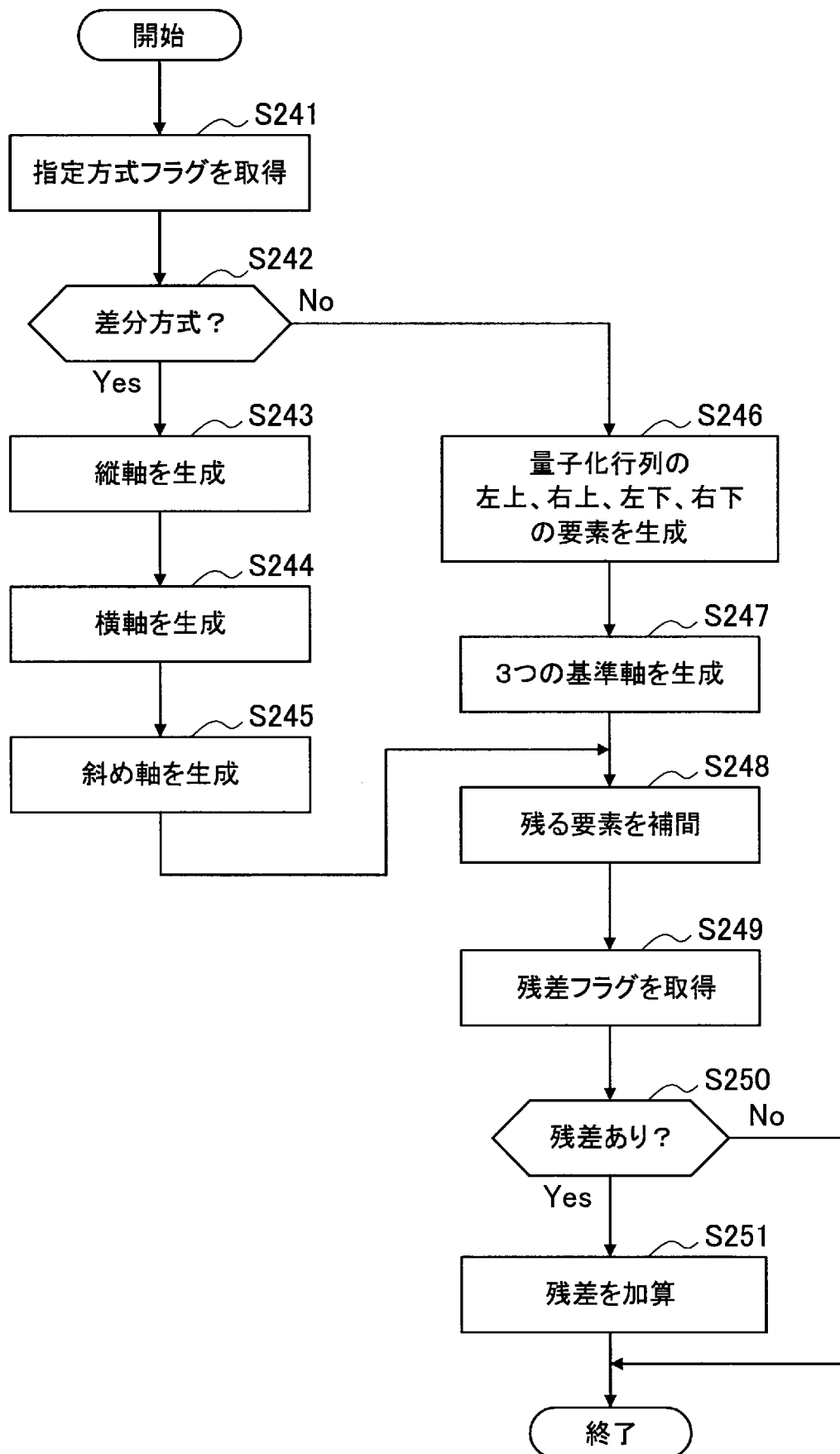
[図9]



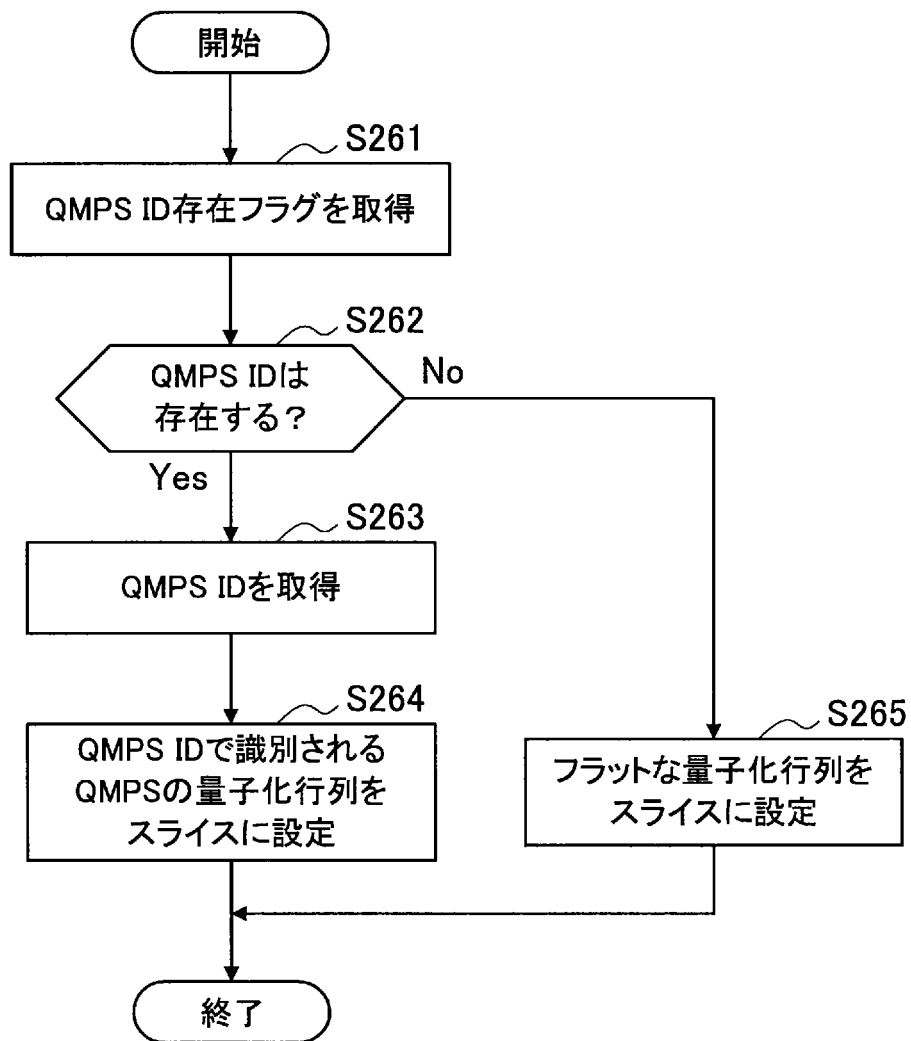
[図10]



[図11]



[図12]



[図13]

```

01 | QuantizaionMatrixParameterSet(){
02 |   quantization_matrix_paramter_id    ...QMPS ID
03 |   pred_present_flag                  ...生成モード存在フラグ
04 |   for(i=0;i<SizeIdc;i++){                // サイズごとにループ //
05 |     for(j=0;j<6;j++){                    // タイプごとにループ //
06 |       if(pred_present_flag){
07 |         pred_mode                      ...生成モード
08 |         if(pred_mode==0){                // コピーモード //
09 |           pred_qmps_id                  ...ソースID
10 |           pred_size_idc                ...コピー元サイズ
11 |           pred_matrix_id              ...コピー元タイプ

12 |           pred_matrix(j,i,pred_qmps_id,pred_size_idc,pred_matrix_id)
                                           // コピー //

13 |           residual_flag                  ...残差フラグ
14 |           if(residual_flag){
15 |             residual_matrix(i)           // 残差を加算 //
16 |           }
17 |         }else if(pred_mode==1){          // 軸指定モード //

           // 図14を参照 //

51 |         }else if(pred_mode==2){          // 全スキャンモード //
52 |           qmatrix(i,j)
53 |         }
54 |       }else{
55 |         qmatrix(i,j)
56 |       }
57 |     }
58 |   }
59 | }

```

[図14]

```

18 |   dpcm_flag                ...指定方式フラグ
19 |   if(dpcm_flag){                // 差分方式 //
20 |       nextcoef = 0
21 |       for(i=0,i<Size;i++){        // 縦軸 //
22 |           delta_coef
23 |           nextcoef = nextcoef + delta_coef
24 |           coef_vertical[i] = nextcoef
25 |       }
26 |       nextcoef = coef_horizontal[0] = coef_vertical[0]
27 |       copy_from_vertical
28 |       if(copy_from_vertical != 0){
29 |           for(i=1,i<Size;i++){    // 横軸 //
30 |               delta_coef
31 |               nextcoef = nextcoef + delta_coef
32 |               coef_horizontal[i] = nextcoef
33 |           }
34 |       }
35 |       nextcoef = coef_diagonal[0] = coef_vertical[0]
36 |       for(i=1,i<Size;i++){        // 斜め軸 //
37 |           delta_coef
38 |           nextcoef = nextcoef + delta_coef
39 |           coef_diagonal[i] = nextcoef
40 |       }
41 |   }else{                          // 補間方式 //
42 |       dc                    ...左上(DC成分)
43 |       horizontal_end        ...右上
44 |       vertical_end          ...左下
45 |       diagonal_end         ...右下
46 |   }
47 |   residual_flag            ...残差フラグ
48 |   if(residual_flag){
49 |       residual_matrix(i)          // 残差を加算 //
50 |   }

```

[図15]

```
01 | residual_matrix(){  
02 |   residual_dpcm_flag          ...残差指定方式フラグ  
03 |   if(residual_dpcm_flag){      // DPCM方式 //  
04 |     nextcoef = 0  
05 |     for(i=0,i<coefNum;i++){  
06 |       delta_coef  
07 |       nextcoef = nextcoef + delta_coef  
08 |       coef[i] = nextcoef  
09 |     }  
10 |   }else{                        // ランレングス方式 //  
11 |     next_pos = 0  
12 |     while(last_pos==0){  
13 |       run  
14 |       data  
15 |       last_pos  
16 |       coef[next_pos+run] = data  
17 |       next_pos += run+1  
18 |     }  
19 |   }  
20 | }
```

[図16]

```

01 | QuantizaionMatrixParameterSet(){
02 |   quantization_matrix_paramter_id    ...QMPS ID
03 |   use_default_only_flag
04 |   use_dqp_flag
05 |   if(use_default_only_flag == 0){
06 |     pred_present_flag                ...生成モード存在フラグ
07 |     Qscale0                          ...第1量子化スケール
08 |     Qscale1                          ...第2量子化スケール
09 |     Qscale2                          ...第3量子化スケール
10 |     Qscale3                          ...第4量子化スケール
11 |     for(i=0;i<4;i++){                  // サイズごとにループ //
12 |       for(j=0;j<6;j++){                // タイプごとにループ //
13 |         if(pred_present_flag){
14 |           pred_mode                    ...生成モード
15 |           if(pred_mode==0){              // コピーモード //
16 |             pred_qmps_id                ...ソースID
17 |             pred_size_idc              ...コピー元サイズ
18 |             pred_matrix_id            ...コピー元タイプ
19 |             residual_flag              ...残差フラグ
20 |             if(residual_flag){
21 |               residual_matrix()          // 残差を特定 //
22 |             }

// 図17, 図18を参照 //

77 |   }
78 | }
79 | }
80 | }

```

```

23 | }else if(pred_mode==1){ // 軸指定モード //
24 |     dpcm_flag           ...指定方式フラグ
25 |     if(dpcm_flag){ // 差分方式 //
26 |         nextcoef = use_dqp_flag ? -6 : 8
27 |         for(i=0,i<Size;i++){ // 縦軸 //
28 |             delta_coef
29 |             nextcoef = nextcoef + delta_coef
30 |             coef_vertical[i] = nextcoef
31 |         }
32 |         nextcoef = coef_horizontal[0] = coef_vertical[0]
33 |         copy_from_vertical
34 |         if(copy_from_vertical != 0){
35 |             for(i=1,i<Size;i++){ // 横軸 //
36 |                 delta_coef
37 |                 nextcoef = nextcoef + delta_coef
38 |                 coef_horizontal[i] = nextcoef
39 |             }
40 |         }
41 |         nextcoef = coef_diagonal[0] = coef_vertical[0]
42 |         for(i=1,i<Size;i++){ // 斜め軸 //
43 |             delta_coef
44 |             nextcoef = nextcoef + delta_coef
45 |             coef_diagonal[i] = nextcoef
46 |         }
47 |     }else{ // 補間方式 //
48 |         dc           ...左上(DC成分)
49 |         horizontal_end ...右上
50 |         vertical_end   ...左下
51 |         diagonal_end   ...右下
52 |     }
53 |     residual_flag       ...残差フラグ
54 |     if(residual_flag){
55 |         residual_matrix() // 残差を特定 //
56 |     }

```

[図18]

```
57 |         }else if(pred_mode==2){                                     // 全スキャンモード //
58 |             matrix_symmetry_flag
59 |             matrix_sign_flag
60 |             matrix_dpcm_run_flag      ...指定方式フラグ
61 |             vlc_table_data          ...VLCテーブル(データ)
62 |             if(matrix_dpcm_run_flag){
63 |                 vlc_table_run      ...VLCテーブル(ラン)
64 |             }
65 |             qmatrix()
66 |         }
67 |     }else{
68 |         matrix_symmetry_flag
69 |         matrix_sign_flag
70 |         matrix_dpcm_run_flag      ...指定方式フラグ
71 |         vlc_table_data          ...VLCテーブル(データ)
72 |         if(matrix_dpcm_run_flag){
73 |             vlc_table_run      ...VLCテーブル(ラン)
74 |         }
75 |         qmatrix()
76 |     }
```

[図19]

```
01 | qmatrix(){
02 |   if(!matrix_dpcm_run_flag){                                     // DPCM方式 //
03 |     nextcoef = use_dqp_flag ? -6 : 8
04 |     for(i=0;i<coefNum;i++){
05 |       delta_coef                                     <----- VLCテーブル(データ)で符号化
06 |       nextcoef = nextcoef + delta_coef
07 |       coef[i] = nextcoef
08 |     }
09 |   }else{                                                         // ランレンゲス方式 //
10 |     next_pos = 0
11 |     while(last_pos==0){
12 |       run                                             <----- VLCテーブル(ラン)で符号化
13 |       data                                           <----- VLCテーブル(データ)で符号化
14 |       coef[next_pos+run] = data
15 |       next_pos += run+1
16 |       if(uiSize <= next_pos){
17 |         last_pos = 1;
18 |       }else{
19 |         last_pos = 0;
20 |       }
21 |     }
22 |   }
23 | }
```

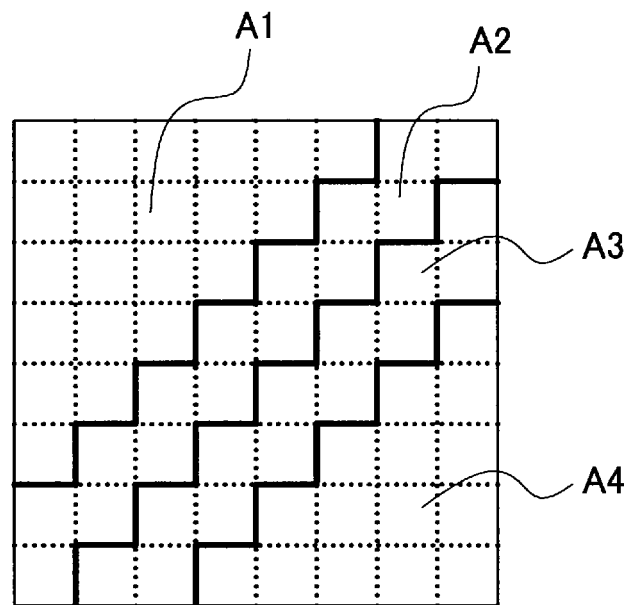
[図20]

```

01 | residual_matrix(){
02 |   residual_symmetry_flag
03 |   residual_dpcm_flag
04 |   residual_run_flag           ...残差指定方式フラグ
05 |   residual_sign_flag
06 |   if(!residual_run_flag){           // DPCM方式 //
07 |     vlc_table_data           ...VLCテーブル(データ)
08 |     nextcoef = 0
09 |     for(i=0,i<coefNum;i++){
10 |       delta_coef           <----- VLCテーブル(データ)で符号化
11 |       nextcoef = nextcoef + delta_coef
12 |       coef[i] = nextcoef
13 |     }
14 |   }else{           // ランレンゲス方式 //
15 |     vlc_table_data           ...VLCテーブル(データ)
16 |     vlc_table_run           ...VLCテーブル(ラン)
17 |     next_pos = 0
18 |     while(last_pos==0){
19 |       run           <----- VLCテーブル(ラン)で符号化
20 |       data           <----- VLCテーブル(データ)で符号化
21 |       coef[next_pos+run] = data
22 |       next_pos += run+1
23 |       if(uiSize <= next_pos){
24 |         last_pos = 1;
25 |       }else{
26 |         last_pos = 0;
27 |       }
28 |     }
29 |   }
30 | }

```

[図21]



8×8量子化行列の
量子化スケール設定領域の例

[図22]

領域 A1: Qscale0 = 1 (量子化せず)

領域 A2: Qscale1 = 2

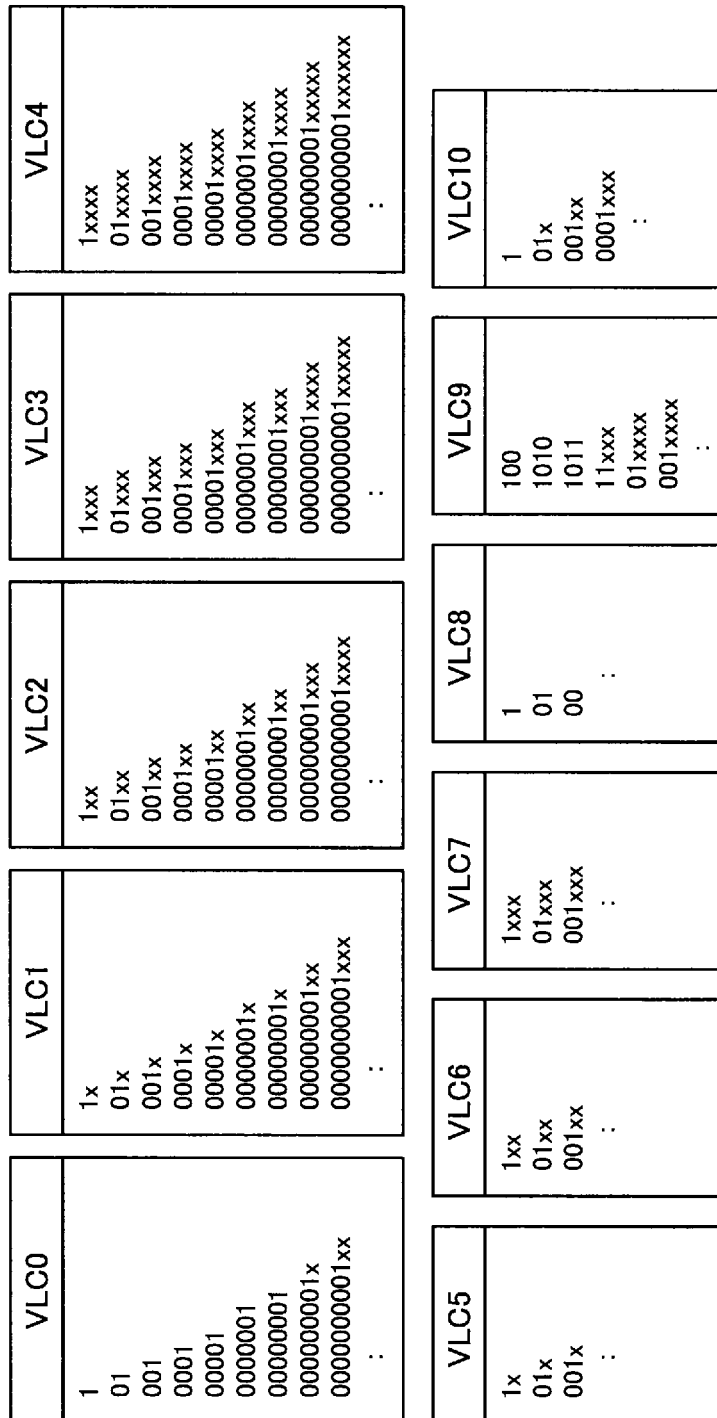
領域 A3: Qscale2 = 3

領域 A4: Qscale3 = 4

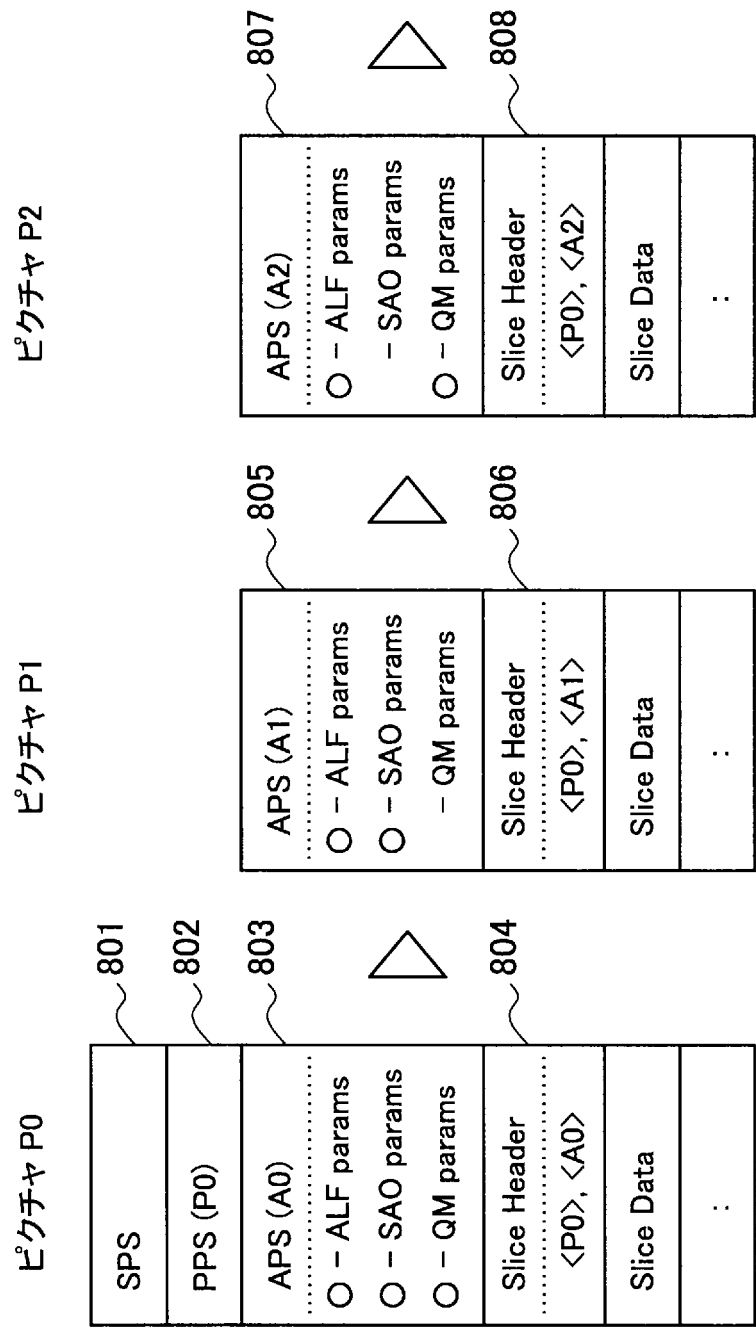
1	1	1	1	1	1	2	2
1	1	1	1	1	2	2	3
1	1	1	1	2	2	3	3
1	1	1	2	2	3	3	4
1	1	2	2	3	3	4	4
1	2	2	3	3	4	4	4
2	2	3	3	4	4	4	4
2	3	3	4	4	4	4	4

量子化スケールの設定例

[図23]



[図24]



(): パラメータセット識別子 < > : 参照用パラメータ ○ : パラメータに更新あり

[図25]

```
01 | aps_rbsp() {  
02 |     aps_id                                ←----- APS ID  
03 |     aps_sample_adaptive_offset_flag  
04 |     aps_adaptive_loop_filter_flag  
05 |     aps_qmatrix_flag  
06 |     if( aps_sample_adaptive_offset_flag || aps_adaptive_loop_filter_flag || aps_qmatrix_flag ) {  
07 |         aps_cabac_use_flag  
08 |         if( aps_cabac_use_flag ) {  
09 |             aps_cabac_init_idc  
10 |             aps_cabac_init_qp_minus26  
11 |         }  
12 |     }  
13 |     if( aps_adaptive_loop_filter_flag ) {  
14 |         alf_data_byte_count  
15 |         alf_param()  
16 |         byte_align()  
17 |     }  
18 |     if( aps_sample_adaptive_offset_flag ) {  
19 |         sao_data_byte_count  
20 |         byte_align ()  
21 |         sao_param()  
22 |         byte_align()  
23 |     }  
24 |     if(aps_qmatrix_flag) {  
25 |         qmatrix_data_byte_count  
26 |         byte_align()  
27 |         qmatrix_param()  
28 |     }  
29 |     rbsp_trailing_bits()  
30 | }
```

ALF関連パラメータ

SAO関連パラメータ

QM関連パラメータ

[図26]

```

01 | slice_header() {
02 |     lightweight_slice_flag
03 |     if( !lightweight_slice_flag ) {
04 |         slice_type
05 |         pic_parameter_set_id          <----- 参照PPS ID
06 |     }
07 |     if( sample_adaptive_offset_enabled_flag || adaptive_loop_filter_enabled_flag || qmatrix_enabled_flag )
08 |         aps_id                        <----- 参照APS ID
09 |         frame_num
10 |         if( IdrPicFlag ) idr_pic_id
11 |         if( pic_order_cnt_type == 0 ) pic_order_cnt_lsb
12 |         if( slice_type == P || slice_type == B ) {
13 |             num_ref_idx_active_override_flag
14 |             if( num_ref_idx_active_override_flag ) {
15 |                 num_ref_idx_l0_active_minus1
16 |                 if( slice_type == B ) num_ref_idx_l1_active_minus1
17 |             }
18 |         }
19 |         ref_pic_list_modification()
20 |         ref_pic_list_combination()
21 |         if( nal_ref_idc != 0 ) dec_ref_pic_marking()
22 |     }
23 |     if( entropy_coding_mode_flag && slice_type != I ) cabac_init_idc
24 |     first_slice_in_pic_flag
25 |     if( first_slice_in_pic_flag == 0 ) slice_address
26 |     if( !lightweight_slice_flag ) {
27 |         slice_qp_delta
28 |         if( deblocking_filter_control_present_flag ) {
29 |             disable_deblocking_filter_idc
30 |             if( disable_deblocking_filter_idc != 1 ) {
31 |                 slice_alpha_c0_offset_div2
32 |                 slice_beta_offset_div2
33 |             }
34 |         }
35 |         if( slice_type == B ) collocated_from_l0_flag
36 |         if( adaptive_loop_filter_enabled_flag && aps_adaptive_loop_filter_flag ) {
37 |             byte_align()
38 |             alf_cu_control_param()
39 |             byte_align()
40 |         }
41 |     }
42 | }

```

[図27]

```

01 |  aps_rbsp() {
02 |     aps_id                                ←----- APS ID
03 |     aps_sample_adaptive_offset_flag
04 |     aps_adaptive_loop_filter_flag
05 |     aps_qmatrix_flag
06 |     if( aps_sample_adaptive_offset_flag || aps_adaptive_loop_filter_flag || aps_qmatrix_flag ) {
07 |         aps_cabac_use_flag
08 |         if( aps_cabac_use_flag ) {
09 |             aps_cabac_init_idc
10 |             aps_cabac_init_qp_minus26
11 |         }
12 |     }
13 |     if( aps_adaptive_loop_filter_flag ) {
14 |         alf_data_byte_count
15 |         alf_param()
16 |         byte_align()
17 |     }
18 |     if( aps_sample_adaptive_offset_flag ) {
19 |         sao_data_byte_count
20 |         byte_align ()
21 |         sao_param()
22 |         byte_align()
23 |     }
24 |     if(aps_qmatrix_flag) {
25 |         ref_aps_id_present_flag           ←----- 過去参照ID存在フラグ
26 |         if(ref_aps_id_present_flag) {
27 |             ref_aps_id                   ←----- 過去参照ID
28 |         } else {
29 |             qmatrix_data_byte_count
30 |             byte_align()
31 |             qmatrix_param()
32 |         }
33 |     }
34 |     rbsp_trailing_bits()
35 | }

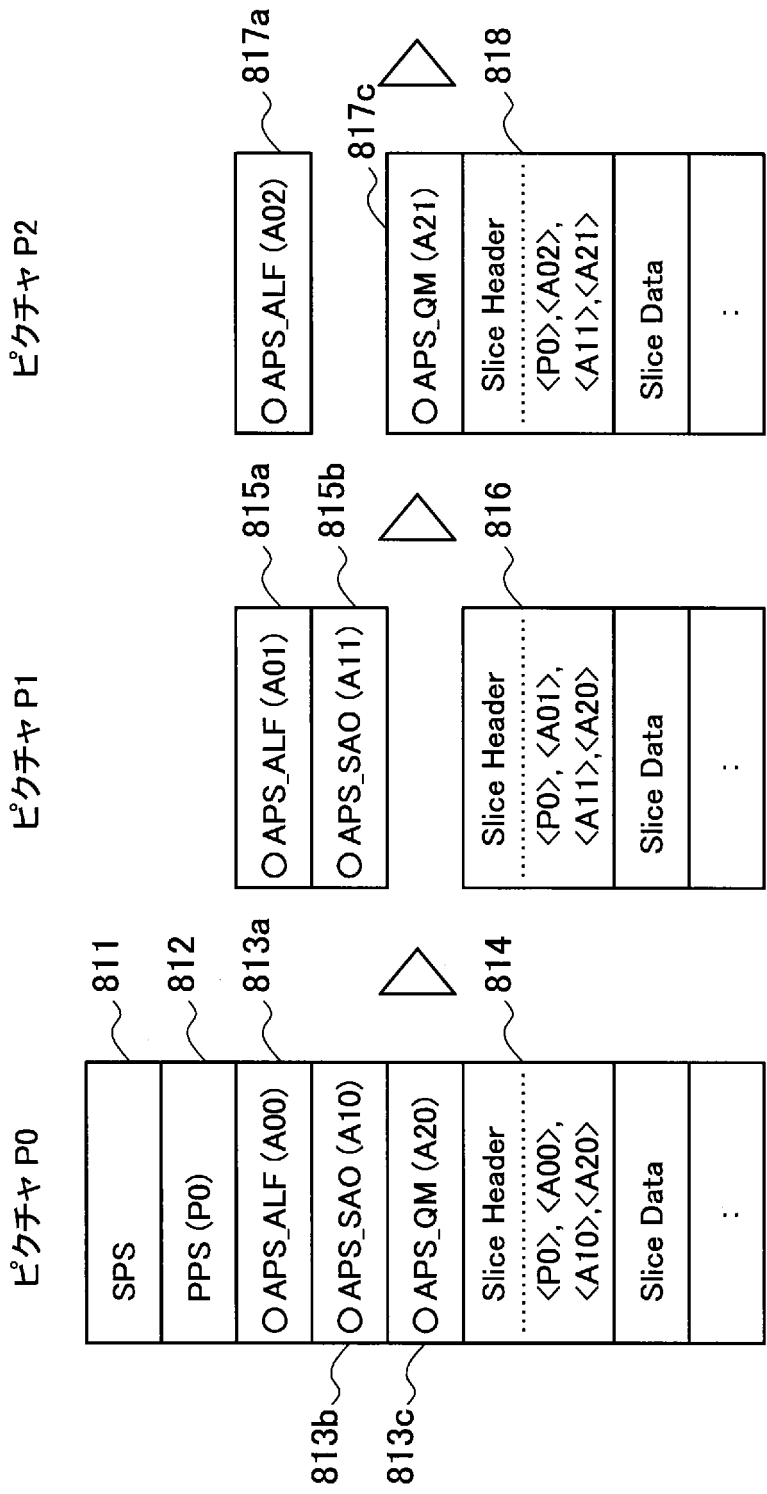
```

ALF関連パラメータ

SAO関連パラメータ

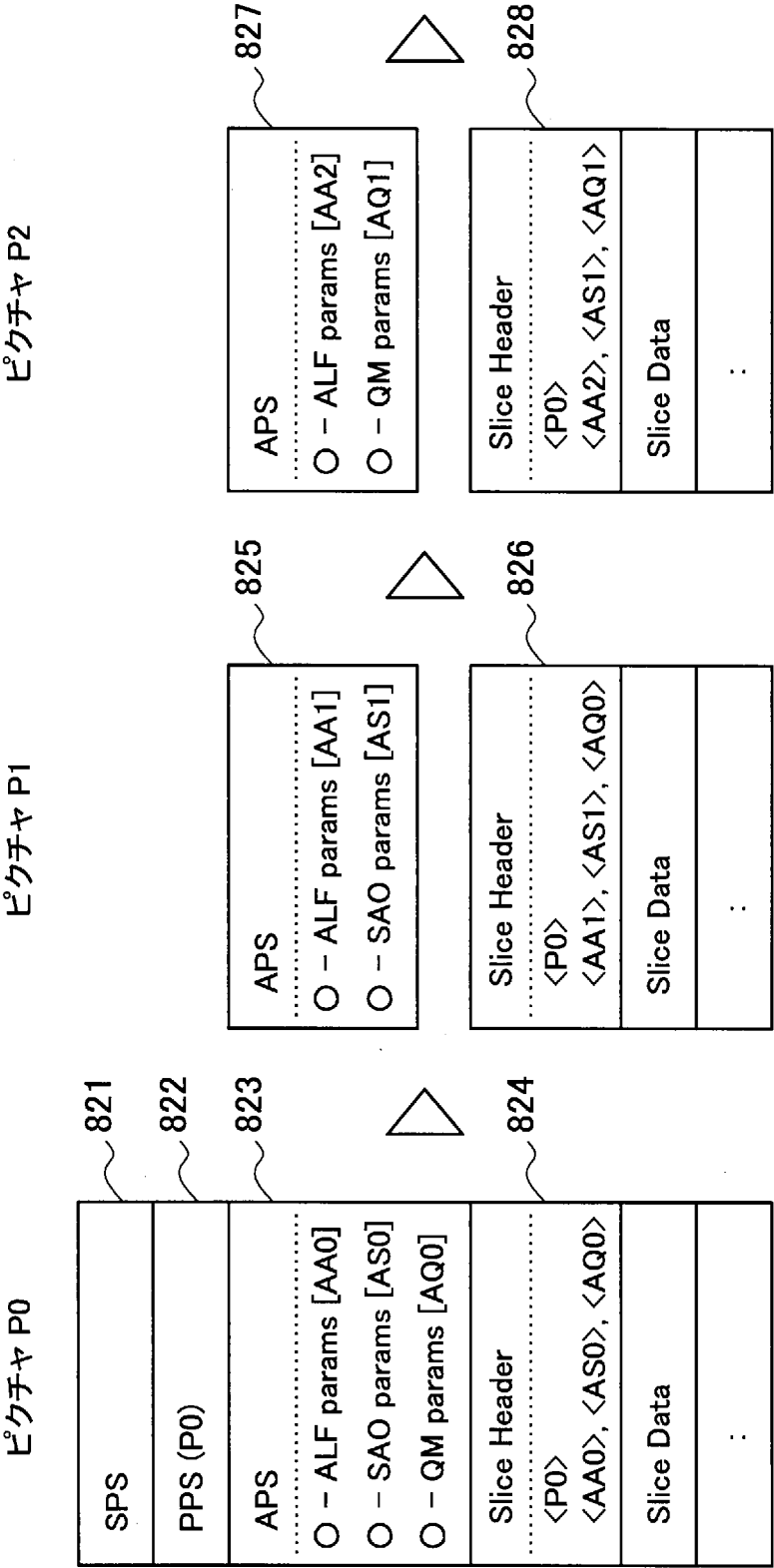
QM関連パラメータ

[図28]



(): パラメータセット識別子 < > : 参照用パラメータ ○ : パラメータに更新あり

[図29]



(): パラメータセット識別子 []: 補助識別子 < >: 参照用パラメータ ○: パラメータに更新あり

[図30]

```

01 | sub_rbsp() {
02 |     aps_adaptive_loop_filter_flag
03 |     aps_sample_adaptive_offset_flag
04 |     aps_qmatrix_flag
05 |     if( aps_sample_adaptive_offset_flag || aps_adaptive_loop_filter_flag || aps_qmatrix_flag ) {
06 |         aps_cabac_use_flag
07 |         if( aps_cabac_use_flag ) {
08 |             aps_cabac_init_idc
09 |             aps_cabac_init_qp_minus26
10 |         }
11 |     }
12 |     if( aps_adaptive_loop_filter_flag ) {
13 |         sub_alf_id
14 |         alf_data_byte_count
15 |         alf_param()
16 |         byte_align()
17 |     }
18 |     if( aps_sample_adaptive_offset_flag ) {
19 |         sub_sao_id
20 |         sao_data_byte_count
21 |         byte_align()
22 |         sao_param()
23 |         byte_align()
24 |     }
25 |     if(aps_qmatrix_flag) {
26 |         sub_qmatrix_id
27 |         qmatrix_data_byte_count
28 |         byte_align()
29 |         qmatrix_param()
30 |     }
31 |     rbsp_trailing_bits()
32 | }

```

グループ別存在フラグ

SUB_ALF ID

ALF関連パラメータ

SUB_SAO ID

SAO関連パラメータ

SUB_QM ID

QM関連パラメータ

[図31]

```

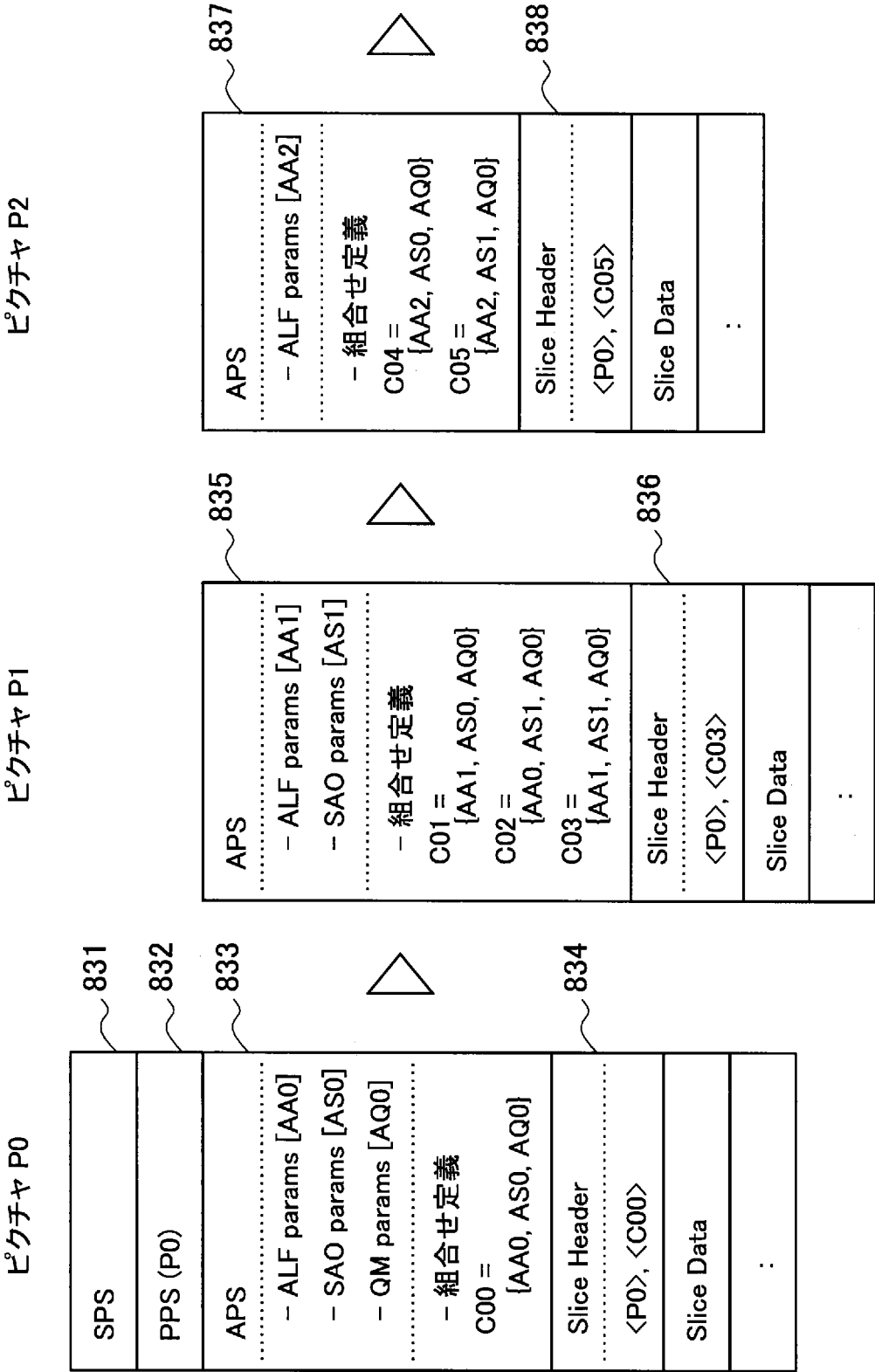
01 | slice_header() {
02 |     lightweight_slice_flag
03 |     if( !lightweight_slice_flag ) {
04 |         slice_type
05 |         pic_parameter_set_id          <----- 参照PPS ID
06 |     }
07 |     if( sample_adaptive_offset_enabled_flag || adaptive_loop_filter_enabled_flag || qmatrix_enabled_flag )
08 |         sub_alf_id                    <--- 参照SUB_ALF ID
09 |         sub_sao_id                    <----- 参照SUB_SAO ID
10 |         sub_qmatrix_id                <--- 参照SUB_QM ID
11 |         frame_num
12 |         if( IdrPicFlag ) idr_pic_id
13 |         if( pic_order_cnt_type == 0 ) pic_order_cnt_lsb
14 |         if( slice_type == P || slice_type == B ) {
15 |             num_ref_idx_active_override_flag
16 |             if( num_ref_idx_active_override_flag ) {
17 |                 num_ref_idx_l0_active_minus1
18 |                 if( slice_type == B ) num_ref_idx_l1_active_minus1
19 |             }
20 |         }
21 |         ref_pic_list_modification()
22 |         ref_pic_list_combination()
23 |         if( nal_ref_idc != 0 ) dec_ref_pic_marking()
24 |     }
25 |     if( entropy_coding_mode_flag && slice_type != I ) cabac_init_idc
26 |     first_slice_in_pic_flag
27 |     if( first_slice_in_pic_flag == 0 ) slice_address
28 |     if( !lightweight_slice_flag ) {
29 |         slice_qp_delta
30 |         if( deblocking_filter_control_present_flag ) {
31 |             disable_deblocking_filter_idc
32 |             if( disable_deblocking_filter_idc != 1 ) {
33 |                 slice_alpha_c0_offset_div2
34 |                 slice_beta_offset_div2
35 |             }
36 |         }
37 |         if( slice_type == B ) collocated_from_l0_flag
38 |         if( adaptive_loop_filter_enabled_flag && aps_adaptive_loop_filter_flag ) {
39 |             byte_align()
40 |             alf_cu_control_param()
41 |             byte_align()
42 |         }
43 |     }
44 | }

```

[図32]

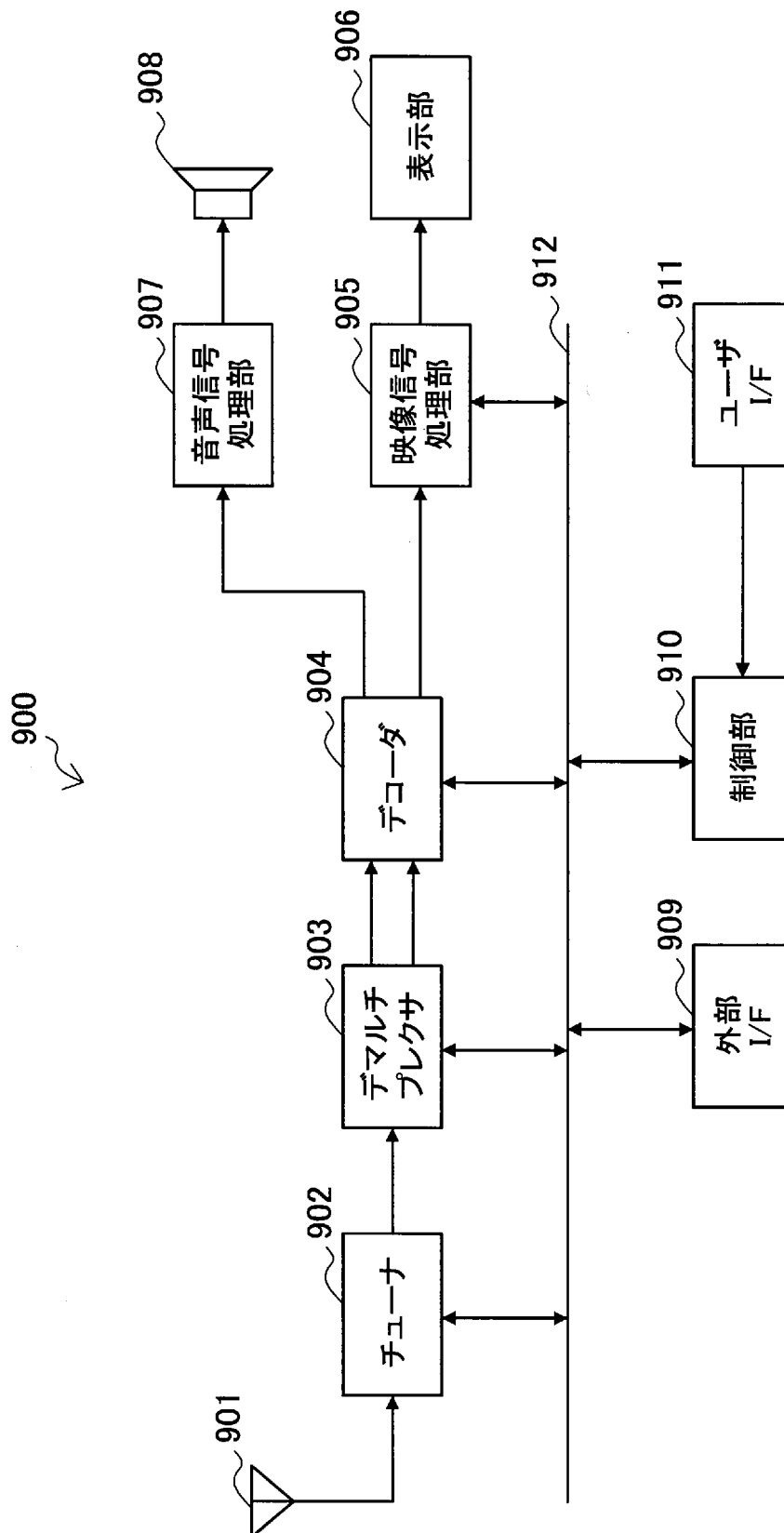
符号化ツール	パラメータの内容	更新頻度	データ サイズ	備考
適応ループフィルタ (ALF)	・フィルタ係数 ・CUごとのOn/Offフラグ	毎ピクチャ	非常に 大きい	・符号量の小さいBピクチャなどでは省略され得る ・過去に設定されたパラメータ値を再利用できる可能性は低い
サンプル適応オフセット (SAO)	・オフセットパターン ・オフセット値	毎ピクチャ	やや 大きい	・過去に設定されたパラメータ値を再利用できる可能性がある
量子化行列 (QM)	・ピクチャごとの量子化行列	毎ピクチャ/ ピクチャ種別毎 /GOP毎	大きい	・ピクチャ種別(I/P/B)ごとの再利用が可能である
適応補間フィルタ (AIF)	・サブピクセル位置ごとのフィルタ係数	毎ピクチャ	小さい	・ピクチャ種別(I/P/B)ごとの再利用が可能である

[図33]

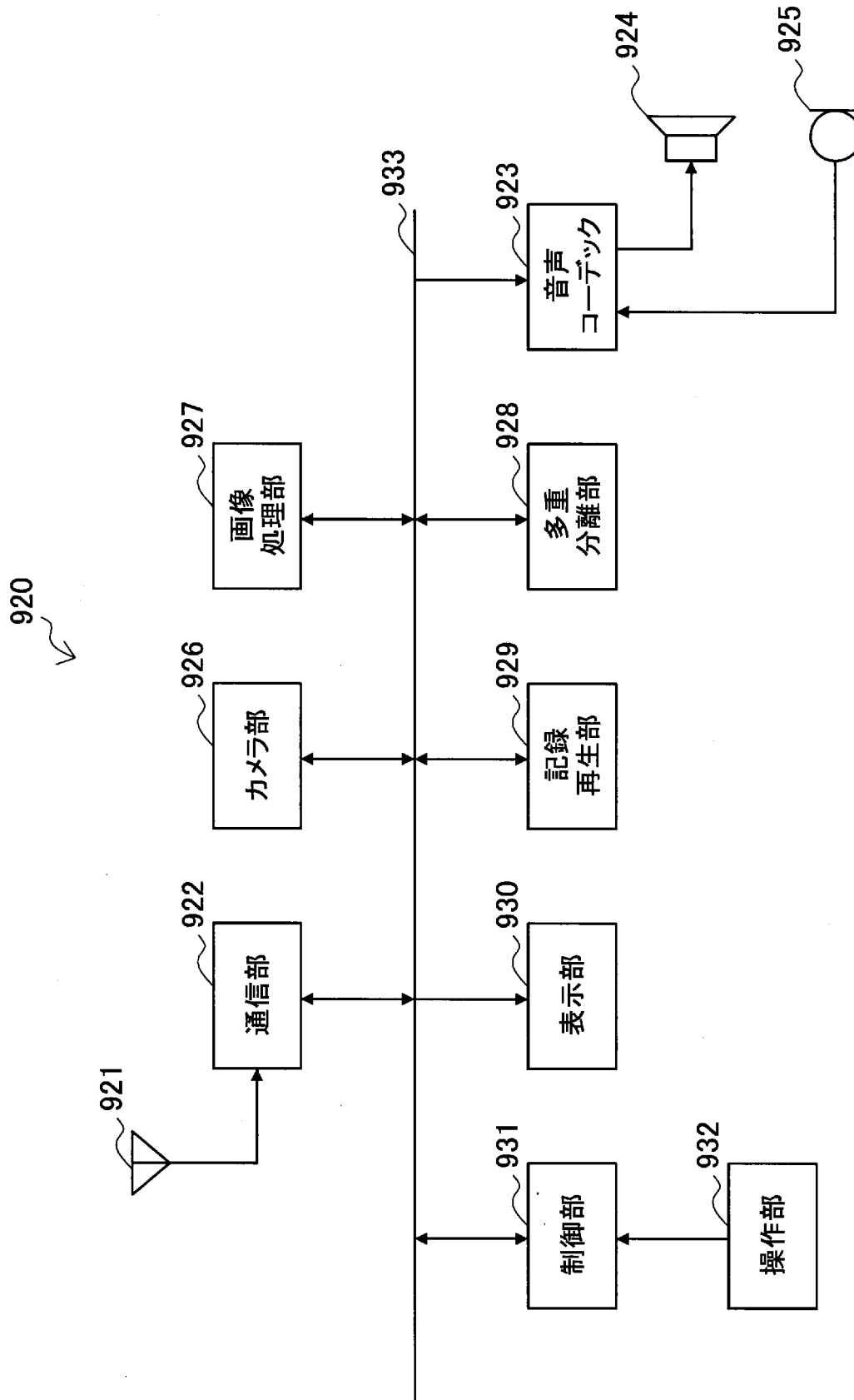


(): パラメータセット識別子 [] : 補助識別子 < > : 参照用パラメータ

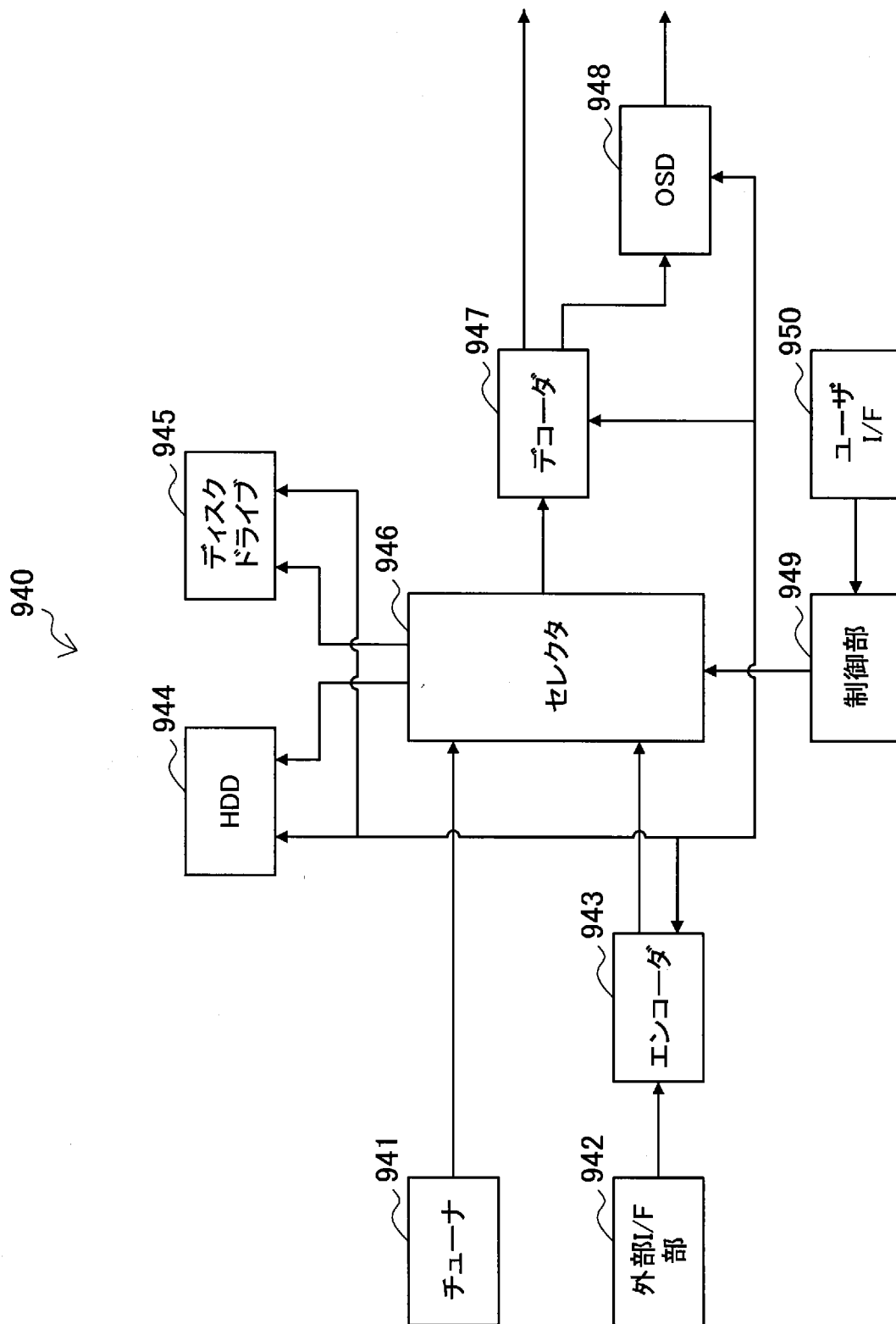
[図34]



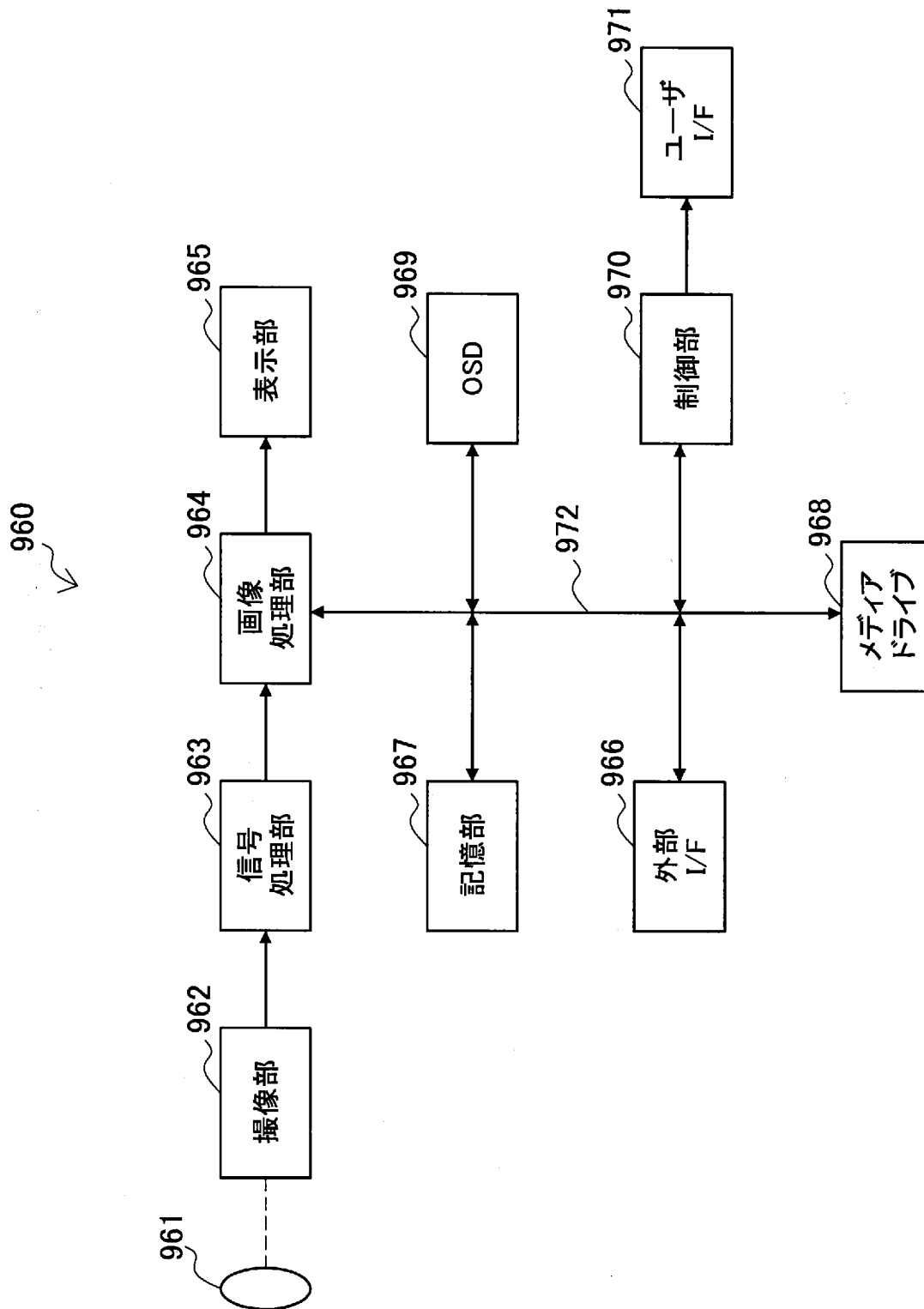
[図35]



[図36]



[図37]



[図38]

