

**Table G-9 – Mapping of ( nX, nY ) to coeffTokenIdx and vice versa**

| nY | nX | 0 ≤ nC < 2 | 2 ≤ nC < 4 | 4 ≤ nC < 8 |
|----|----|------------|------------|------------|
| 1  | 9  | 34         | 32         | 33         |
| 2  | 9  | 31         | 33         | 29         |
| 3  | 9  | 24         | 26         | 19         |
| 0  | 10 | 37         | 39         | 40         |
| 1  | 10 | 38         | 36         | 37         |
| 2  | 10 | 35         | 37         | 34         |
| 3  | 10 | 28         | 30         | 27         |
| 0  | 11 | 41         | 42         | 44         |
| 1  | 11 | 42         | 40         | 41         |
| 2  | 11 | 39         | 41         | 38         |
| 3  | 11 | 36         | 34         | 35         |
| 0  | 12 | 45         | 43         | 47         |
| 1  | 12 | 46         | 44         | 45         |
| 2  | 12 | 43         | 45         | 42         |
| 3  | 12 | 40         | 38         | 39         |
| 0  | 13 | 50         | 47         | 49         |
| 1  | 13 | 49         | 48         | 48         |
| 2  | 13 | 47         | 49         | 46         |
| 3  | 13 | 44         | 46         | 43         |
| 0  | 14 | 54         | 51         | 53         |
| 1  | 14 | 51         | 54         | 50         |
| 2  | 14 | 52         | 52         | 51         |
| 3  | 14 | 48         | 50         | 52         |
| 0  | 15 | 58         | 55         | 57         |
| 1  | 15 | 55         | 56         | 54         |
| 2  | 15 | 56         | 57         | 55         |
| 3  | 15 | 53         | 53         | 56         |
| 0  | 16 | 61         | 59         | 61         |
| 1  | 16 | 59         | 60         | 58         |
| 2  | 16 | 60         | 61         | 59         |
| 3  | 16 | 57         | 58         | 60         |

#### G.9.2.2 Alternative parsing process for run information

Inputs to this process are bits from slice data and the number of non-zero transform coefficient levels  $\text{TotalCoeff}(\text{coeff\_token})$ .

Output of this process is a list of runs of zero transform coefficient levels preceding non-zero transform coefficient levels called run.

Let  $\text{maxCoeff}$  be a variable that is set equal to  $(\text{scan\_idx\_end} - \text{scan\_idx\_start} + 1)$ .

Initially, an index  $i$  is set equal to 0.

The variable `zerosLeft` is derived as follows.

- If the number of non-zero transform coefficient levels `TotalCoeff( coeff_token )` is equal to the maximum number of non-zero transform coefficient levels `maxCoeff`, a variable `zerosLeft` is set equal to 0.
- Otherwise (the number of non-zero transform coefficient levels `TotalCoeff( coeff_token )` is less than the maximum number of non-zero transform coefficient levels `maxCoeff`), `total_zeros` is decoded and `zerosLeft` is set equal to its value.

The VLC used to decode `total_zeros` is derived as follows:

- If `maxCoeff` is less than or equal to 4, one of the VLCs specified in Table 9-9(a) is used with `tzVlcIndex` being derived by

$$\text{tzVlcIndex} = \min( 3, \text{TotalCoeff}( \text{coeff\_token} ) + 4 - \text{maxCoeff} ) \quad (\text{G-363})$$

- Otherwise, if `maxCoeff` is greater than 4 and less than or equal to 8, one of the VLCs specified in Table 9-9(b) is used with `tzVlcIndex` being derived by

$$\text{tzVlcIndex} = \min( 7, \text{TotalCoeff}( \text{coeff\_token} ) + 8 - \text{maxCoeff} ) \quad (\text{G-364})$$

- Otherwise, if `maxCoeff` is greater than 8 and less than 15, VLCs from Table 9-7 and Table 9-8 are used with `tzVlcIndex` being derived by

$$\text{tzVlcIndex} = \min( 15, \text{TotalCoeff}( \text{coeff\_token} ) + 16 - \text{maxCoeff} ) \quad (\text{G-365})$$

- Otherwise (`maxCoeff` is greater than or equal to 15), VLCs from Table 9-7 and Table 9-8 are used with `tzVlcIndex` equal to `TotalCoeff( coeff_token )`.

The following procedure is then applied iteratively (`TotalCoeff( coeff_token ) – 1`) times.

1. The variable `run[ i ]` is derived as follows.
  - If `zerosLeft` is greater than zero, a value `run_before` is decoded based on Table 9-10 and `zerosLeft`. `run[ i ]` is set equal to `run_before`.
  - Otherwise (`zerosLeft` is equal to 0), `run[ i ]` is set equal to 0.
2. The value of `run[ i ]` is subtracted from `zerosLeft` and the result assigned to `zerosLeft`. The result of the subtraction shall be greater than or equal to 0.
3. The index  $i$  is incremented by 1.

Finally the value of `zerosLeft` is assigned to `run[ i ]`.

### **G.9.3 Alternative CABAC parsing process for slice data in scalable extension**

Subclause G.9.3.1 specifies the initialisation process for the alternative CABAC parsing process for slice data in scalable extension.

Subclause G.9.3.2 specifies the binarization process for the alternative CABAC parsing process for slice data in scalable extension.

Subclause G.9.3.3 specifies the decoding process flow for the alternative CABAC parsing process for slice data in scalable extension.

#### **G.9.3.1 Initialisation process**

Outputs of this process are the initialised CABAC context variables indexed by `ctxIdx`.

Table G-11 and Table G-12 contain the values of the variables  $n$  and  $m$  used in the initialisation of context variables that are assigned to syntax element `base_mode_flag`, `motion_prediction_flag_10`, `motion_prediction_flag_11`, and `residual_prediction_flag` in subclause G.7.3.4.1 and G.7.3.6. For all other syntax elements in subclauses G.7.3.4.1 and G.7.3.6 the initialisation process of context variables as specified in subclause 9.3.1 applies.

For each context variable, the two variables pStateIdx and valMPS are initialised. The two values assigned to pStateIdx and valMPS for the initialisation are derived from SliceQP<sub>Y</sub>, which is derived in Equation 7-28. Given the two table entries ( m, n ), the following applies

1.  $\text{preCtxState} = \text{Clip3}(1, 126, ((m * \text{Clip3}(0, 51, \text{SliceQP}_Y)) \gg 4) + n)$
2.  $\text{if}(\text{preCtxState} \leq 63) \{$   
 $\quad \text{pStateIdx} = 63 - \text{preCtxState}$   
 $\quad \text{valMPS} = 0$   
 $\quad \} \text{ else } \{$   
 $\quad \text{pStateIdx} = \text{preCtxState} - 64$   
 $\quad \text{valMPS} = 1$   
 $\quad \}$

In Table G-10, the ctxIdx for which initialisation is needed for each of the slice types EI, EP, and EB are listed. Also listed is the table number that includes the values of m and n needed for the initialisation. For EP and EB slices, the initialisation depends also on the value of the cabac\_init\_idc syntax element. Note that the syntax element names do not affect the initialisation process.

**Table G-10 - Association of ctxIdx and syntax elements for each slice type in the initialisation process**

|                                                                         | Syntax element            | Table      | Slice type |           |           |
|-------------------------------------------------------------------------|---------------------------|------------|------------|-----------|-----------|
|                                                                         |                           |            | EI         | EP        | EB        |
| macroblock_layer_in_scalable_extension()                                | base_mode_flag            | Table G-11 | 1024-1026  | 1024-1026 | 1024-1026 |
| mb_pred_in_scalable_extension() and sub_mb_pred_in_scalable_extension() | motion_prediction_flag_10 | Table G-12 |            | 1027      | 1027      |
|                                                                         | motion_prediction_flag_11 | Table G-12 |            | 1027      | 1027      |
| residual_in_scalable_extension()                                        | residual_prediction_flag  | Table G-12 |            | 1028-1029 | 1028-1029 |

**Table G-11 - Values of variables m and n for ctxIdx from 1024 to 1026**

| ctxIdx | EI slices |     | Value of cabac_init_idc (EP, EB slices) |    |   |    |   |    |
|--------|-----------|-----|-----------------------------------------|----|---|----|---|----|
|        |           |     | 0                                       |    | 1 |    | 2 |    |
|        | m         | n   | m                                       | n  | m | n  | m | n  |
| 1024   | -14       | 138 | 0                                       | 75 | 0 | 75 | 0 | 75 |
| 1025   | -22       | 140 | 2                                       | 65 | 2 | 65 | 2 | 65 |
| 1026   | -11       | 99  | 2                                       | 59 | 2 | 59 | 2 | 59 |

**Table G-12 - Values of variables m and n for ctxIdx from 1027 to 1029**

| ctxIdx | Value of cabac_init_idc |     |     |     |     |     |
|--------|-------------------------|-----|-----|-----|-----|-----|
|        | 0                       |     | 1   |     | 2   |     |
|        | m                       | n   | m   | n   | m   | n   |
| 1027   | -6                      | 67  | -6  | 67  | -6  | 67  |
| 1028   | -23                     | 104 | -23 | 104 | -23 | 104 |
| 1029   | -35                     | 106 | -35 | 106 | -35 | 106 |

### G.9.3.2 Binarization process

Input to this process is a request for a syntax element.

Output of this process is the binarization of the syntax element, maxBinIdxCtx, ctxIdxOffset, and bypassFlag.

Associated with each binarization or binarization part of a syntax element is a specific value of the context index offset (ctxIdxOffset) variable and a specific value of the maxBinIdxCtx variable as given in Table G-13.

The variable bypassFlag is set equal to 0.

The possible values of the context index ctxIdx are in the range 1024 to 1029, inclusive. The value assigned to ctxIdxOffset specifies the lower value of the range of ctxIdx assigned to the corresponding binarization or binarization part of a syntax element.

**Table G-13 - Syntax elements and associated types of binarization, maxBinIdxCtx, and ctxIdxOffset**

| Syntax element                                          | Type of binarization | maxBinIdxCtx | ctxIdxOffset |
|---------------------------------------------------------|----------------------|--------------|--------------|
| base_mode_flag                                          | FL, cMax=1           | 0            | 1024         |
| motion_prediction_flag_l0,<br>motion_prediction_flag_l1 | FL, cMax=1           | 0            | 1027         |
| residual_prediction_flag                                | FL, cMax=1           | 0            | 1028         |

### G.9.3.3 Decoding process flow

Input to this process is a binarization of the requested syntax element, maxBinIdxCtx, bypassFlag and ctxIdxOffset as specified in subclause G.9.3.2.

Output of this process is the value of the syntax element.

This process specifies how each bit of a bit string is parsed for each syntax element.

After parsing each bit, the resulting bit string is compared to all bin strings of the binarization of the syntax element and the following applies.

- If the bit string is equal to one of the bin strings, the corresponding value of the syntax element is the output.
- Otherwise (the bit string is not equal to one of the bin strings), the next bit is parsed.

While parsing each bin, the variable binIdx is incremented by 1 starting with binIdx being set equal to 0 for the first bin.

The parsing of each bin is specified by the following two ordered steps:

1. Given binIdx, maxBinIdxCtx and ctxIdxOffset, ctxIdx is derived as specified in subclause G.9.3.3.1.
2. Given ctxIdx, the value of the bin from the bitstream as specified in subclause 9.3.3.2 is decoded.

#### G.9.3.3.1 Derivation process for ctxIdx

Inputs to this process are binIdx, maxBinIdxCtx and ctxIdxOffset.

Output of this process is ctxIdx.

Table G-14 shows the assignment of ctxIdx increments (ctxIdxInc) to binIdx for all ctxIdxOffset values for the syntax elements base\_mode\_flag, motion\_prediction\_flag\_l0, motion\_prediction\_flag\_l1, and residual\_prediction\_flag.

The ctxIdx to be used with a specific binIdx is the sum of ctxIdxOffset and ctxIdxInc, which is found in Table G-14. When more than one value is listed in Table G-14 or Table 9-39 for a binIdx, the assignment process for ctxIdxInc for that binIdx is further specified in the subclauses given in parenthesis of the corresponding table entry.

All entries in Table G-14 labelled with "na" correspond to values of binIdx that do not occur for the corresponding ctxIdxOffset.

**Table G-14 - Assignment of ctxIdxInc to binIdx for the ctxIdxOffset values related to the syntax elements base\_mode\_flag and residual\_prediction\_flag**

| ctxIdxOffset | binIdx                           |    |    |    |    |    |     |
|--------------|----------------------------------|----|----|----|----|----|-----|
|              | 0                                | 1  | 2  | 3  | 4  | 5  | ≥ 6 |
| 1024         | 0,1,2<br>(subclause G.9.3.3.2.1) | na | na | na | na | na | na  |
| 1027         | 0                                | na | na | na | na | na | na  |
| 1028         | 0,1<br>(subclause G.9.3.3.2.2)   | na | na | na | na | na | na  |

#### **G.9.3.3.2 Assignment process of ctxIdxInc using neighbouring syntax elements**

Subclause G.9.3.3.2.1 specifies the derivation process of ctxIdxInc for the syntax element base\_mode\_flag.

Subclause G.9.3.3.2.2 specifies the derivation process of ctxIdxInc for the syntax element residual\_prediction\_flag.

##### **G.9.3.3.2.1 Derivation process of ctxIdxInc for the syntax element base\_mode\_flag**

Output of this process is ctxIdxInc.

The derivation process for neighbouring macroblocks specified in subclause 6.4.10.1 is invoked and the output is assigned to mbAddrA and mbAddrB.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

- If mbAddrN is available and base\_mode\_flag for the macroblock mbAddrN is equal to 1, condTermFlagN is set equal to 0.
- Otherwise (mbAddrN is not available or base\_mode\_flag for the macroblock mbAddrN is equal to 0), condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived by

$$\text{ctxIdxInc} = \text{condTermFlagA} + \text{condTermFlagB} \quad (\text{G-366})$$

##### **G.9.3.3.2.2 Derivation process of ctxIdxInc for the syntax element residual\_prediction\_flag**

Output of this process is ctxIdxInc.

Depending on base\_mode\_flag, the following applies.

- If base\_mode\_flag is equal to 1, ctxIdxInc is set equal to 0.
- Otherwise (base\_mode\_flag is equal to 0), ctxIdxInc is set equal to 1.

## **G.10 Profiles and levels**

### **G.10.1 Profiles**

#### **G.10.1.1 Scalable Baseline profile**

Bitstreams conforming to the Scalable Baseline profile shall obey the following constraints:

- a) The base layer bitstream as specified in subclause G.8.8.2 shall obey the following constraints.  
NOTE – In addition to the base layer constraints specified below in items i) through vi), the value of the syntax element constrained\_intra\_pred\_flag for picture parameter sets of the base layer stream is constrained as specified below in item k).
  - i) All constraints of the Baseline profile specified in subclause A.2.1 shall be obeyed.
  - ii) Sequence parameter sets should have profile\_idc equal to 66. Sequence parameter sets may have profile\_idc equal to 77 or 88. Sequence parameter sets shall not have profile\_idc equal to a value other than 66, 77, or 88.

- iii) Sequence parameter sets shall have `constraint_set0_flag`, `constraint_set1_flag`, and `constraint_set2_flag` equal to 1.
  - iv) Picture parameter sets shall have `num_slice_groups_minus1` equal to 0.
  - v) Picture parameter sets shall have `redundant_pic_cnt_present_flag` equal to 0.
  - vi) Arbitrary slice order is not allowed.
- b) Only I, P, EI, EP, and EB slices shall be present.
  - c) SVC sequence parameter sets shall have `chroma_format_idc` equal to 1.
  - d) SVC sequence parameter sets shall have `bit_depth_luma_minus8` equal to 0.
  - e) SVC sequence parameter sets shall have `bit_depth_chroma_minus8` equal to 0.
  - f) SVC sequence parameter sets shall have `separate_colour_plane_flag` equal to 0.
  - g) SVC sequence parameter sets shall have `qpprime_y_zero_transform_bypass_flag` equal to 0.
  - h) SVC sequence parameter sets shall have `frame_mbs_only_flag` equal to 1.
  - i) Picture parameter sets shall have `num_slice_groups_minus1` in the range of 0 to 7, inclusive.
  - j) When present, picture parameter sets shall have `slice_group_map_type` equal to 2.
  - k) A list of integer values specifying layer representation identifiers is derived by invoking the process specified in G.8.1.1 with the output being the list `dqIdList`. The variable `numDQEntries` is set equal to the number of elements in the list `dqIdList`. When `numDQEntries` is greater than 1, for any element `dqIdList[i]` with  $i = 0..(\text{numDQEntries} - 2)$ , when `tcoeff_level_prediction_flag` is equal to 0 for any layer representation with DQId in the set specified by `dqIdList[k]` with  $k = i..(\text{numDQEntries} - 1)$ , the picture parameter set that is referenced by the coded slice NAL units of the layer representation with DQId equal to `dqIdList[i]` shall have `constrained_intra_pred_flag` equal to 1.
  - l) Arbitrary slice order is not allowed.
  - m) For each present value of `dependency_id` greater than 0, one of the following constraints shall be obeyed.
    - `ScaledRefLayerPicWidthInSamplesL` is equal to `RefLayerPicWidthInSamplesL` and `ScaledRefLayerPicHeightInSamplesL` is equal to `RefLayerPicHeightInSamplesL`
    - `ScaledRefLayerPicWidthInSamplesL` is equal to  $(1.5 * \text{RefLayerPicWidthInSamples}_L)$  and `ScaledRefLayerPicHeightInSamplesL` is equal to  $(1.5 * \text{RefLayerPicHeightInSamples}_L)$
    - `ScaledRefLayerPicWidthInSamplesL` is equal to  $(2 * \text{RefLayerPicWidthInSamples}_L)$  and `ScaledRefLayerPicHeightInSamplesL` is equal to  $(2 * \text{RefLayerPicHeightInSamples}_L)$
  - n) For each present value of `dependency_id` greater than 0, all of the following constraints shall be obeyed.
    - $(\text{ScaledRefLayerLeftOffset} \% 16)$  is equal to 0
    - $(\text{ScaledRefLayerTopOffset} \% 16)$  is equal to 0
  - o) The level constraints specified in subclause G.10.2 shall be fulfilled.

Conformance of a bitstream to Scalable Baseline profile is specified by `profile_idc` equal to 83.

Decoders conforming to the Scalable Baseline profile at a specific level shall be capable of decoding all bitstreams in which all active SVC sequence parameter sets have `profile_idc` equal to 83 or (`profile_idc` equal to 86 and `constraint_set0_flag` equal to 1) or (`profile_idc` equal to 66, 77, or 88 and `constraint_set0_flag` equal to 1 and `constraint_set1_flag` equal to 1 and `constraint_set2_flag` equal to 1) and in which `level_idc` or (`level_idc` and `constraint_set3_flag`) for all active SVC sequence parameter sets represent a level less than or equal to the specified level.

#### **G.10.1.2 Scalable High profile**

Bitstreams conforming to the Scalable High profile shall obey the following constraints:

- a) The base layer bit-stream as specified in subclause G.8.8.2 shall obey the following constraints.
 

NOTE – In addition to the base layer constraints specified below in items i) through iii), the value of the syntax element `constrained_intra_pred_flag` for picture parameter sets of the base layer stream is constrained as specified below in item j).

  - i) All constraints of the High profile specified in subclause A.2.4 shall be obeyed.

- ii) Sequence parameter sets should have `profile_idc` equal to 100. Sequence parameter sets may have `profile_idc` equal to 66, 77, or 88 and `constraint_set1_flag` equal to 1. Sequence parameter sets shall not have `profile_idc` equal to a value other than 66, 77, 88, or 100.
- iii) The syntax element `direct_spatial_mv_pred_flag` shall be equal to 1.
- b) Only I, P, B, EI, EP, and EB slices shall be present.
- c) SVC sequence parameter sets shall have `chroma_format_idc` equal to 1.
- d) SVC sequence parameter sets shall have `bit_depth_luma_minus8` equal to 0.
- e) SVC sequence parameter sets shall have `bit_depth_chroma_minus8` equal to 0.
- f) SVC sequence parameter sets shall have `separate_colour_plane_flag` equal to 0.
- g) SVC sequence parameter sets shall have `qpprime_y_zero_transform_bypass_flag` equal to 0.
- h) Picture parameter sets shall have `redundant_pic_cnt_present_flag` equal to 0.
- i) Picture parameter sets shall have `num_slice_groups_minus1` equal to 0.
- j) A list of integer values specifying layer representation identifiers is derived by invoking the process specified in G.8.1.1 with the output being the list `dqIdList`. The variable `numDQEntries` is set equal to the number of elements in the list `dqIdList`. When `numDQEntries` is greater than 1, for any element `dqIdList[i]` with  $i = 0..(\text{numDQEntries} - 2)$ , when `tcoeff_level_prediction_flag` is equal to 0 for any layer representation with `DQId` in the set specified by `dqIdList[k]` with  $k = i..(\text{numDQEntries} - 1)$ , the picture parameter set that is referenced by the coded slice NAL units of the layer representation with `DQId` equal to `dqIdList[i]` shall have `constrained_intra_pred_flag` equal to 1.
- k) Arbitrary slice order is not allowed.
- l) The level constraints specified in subclause G.10.2 shall be fulfilled.

Conformance of a bitstream to Scalable High profile is specified by `profile_idc` equal to 86.

Decoders conforming to the Scalable High profile at a specific level shall be capable of decoding all bitstreams in which all active SVC sequence parameter sets have `profile_idc` equal to 86 or (`profile_idc` equal to 83 and `constraint_set1_flag` equal to 1) or `profile_idc` equal to 77 or `profile_idc` equal to 100 or (`profile_idc` not equal to 83 or 86 and `constraint_set1_flag` equal to 1) and in which `level_idc` or (`level_idc` and `constraint_set3_flag`) for all active SVC sequence parameter sets represent a level less than or equal to the specified level.

#### **G.10.1.3 Scalable High Intra profile**

Bitstreams conforming to the Scalable High Intra profile shall obey the following constraints:

- a) The base layer bit-stream as specified in subclause G.8.8.2 shall obey the following constraints:
  - i) All constraints of the High profile specified in subclause A.2.4 shall be obeyed.
  - ii) Sequence parameter sets should have `profile_idc` equal to 100 and `constraint_set3_flag` equal to 1. Sequence parameter sets may have `profile_idc` equal to 66, 77, or 88 and `constraint_set1_flag` equal to 1. Sequence parameter sets shall not have `profile_idc` equal to a value other than 66, 77, 88, or 100.
- b) All constraints of the Scalable High profile specified in subclause G.10.1.2 shall be obeyed.
- c) All pictures shall be IDR pictures.
- d) SVC sequence parameter sets shall have `num_ref_frames` equal to 0.
- e) When `vui_parameters_present_flag` is equal to 1 and `bitstream_restriction_flag` is equal to 1, SVC sequence parameter sets shall have `num_reorder_frames` equal to 0.
- f) When `vui_parameters_present_flag` is equal to 1 and `bitstream_restriction_flag` is equal to 1, SVC sequence parameter sets shall have `max_dec_frame_buffering` equal to 0.
- g) Picture timing SEI messages, whether present in the bitstream (by non-VCL NAL units) or conveyed equivalently by other means not specified in this Recommendation | International Standard, shall have `dpb_output_delay` equal to 0.
- h) The level constraints specified in subclause G.10.2 shall be fulfilled.

Conformance of a bitstream to Scalable High Intra profile is specified by `constraint_set3_flag` being equal to 1 with `profile_idc` equal to 86.

Decoders conforming to the Scalable High Intra profile at a specific level shall be capable of decoding all bitstreams in which all active SVC sequence parameter sets have profile\_idc equal to 86 or 100 and constraint\_set3\_flag equal to 1 and in which level\_idc for all active SVC sequence parameter sets represents a level less than or equal to the specified level.

## G.10.2 Levels

The following is specified for expressing the constraints in this subclause.

- Let access unit n be the n-th access unit in decoding order with the first access unit being access unit 0.
- Let picture n be the primary coded picture or the corresponding decoded picture of access unit n.

### G.10.2.1 Level limits common to Scalable Baseline, Scalable High, and Scalable High Intra profiles

The variable fR is derived as follows.

- If picture n is a frame, fR is set equal to (1 + 172).
- Otherwise (picture n is a field), fR is set equal to (1 + (172 \* 2)).

The variable dqIdMax is set equal to the maximum value of DQId for the layer representation of the access unit.

The variable refLayerDQId is set equal to the maximum value of ref\_layer\_dq\_id for the layer representation with DQId equal to dqIdMax.

A list of integer values specifying layer representation identifiers for the access unit is derived by invoking the process specified in subclause G.8.1.1 with the output being the list dqIdList. The variable numDQEntries is set equal to the number of elements in the list dqIdList.

A variable dependentDId is derived by the following pseudo-code.

```
dependentDId = 0
for( i = 0; i < numDQEntries; i++ )
    if( ( dqIdList[ i ] % 16 ) == 0 )
        dependentDId++
```

(G-367)

The variable svcPicSizeInMbs is derived as follows.

- If numDQEntries is less than 3, svcPicSizeInMbs is set equal to PicSizeInMbs for the layer representation with DQId equal to dqIdMax.
- Otherwise (numDQEntries is greater than 2), svcPicSizeInMbs is derived by applying the following ordered steps.
  1. svcPicSizeInMbs is set equal to PicSizeInMbs for the layer representation with DQId equal to dqIdMax.
  2. The variable refLayerMbs is set equal to 0.
  3. For each element dqIdList[ i ] with i = 2..(numDQEntries – 1), with refLayerPicSizeInMbs being the variable PicSizeInMbs for the layer representation, the variable refLayerMbs is modified by

refLayerMbs += refLayerPicSizeInMbs (G-368)

4. svcPicSizeInMbs is modified by

svcPicSizeInMbs += ( refLayerMbs + 1 ) >> 1 (G-369)

Bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles at a specific level shall obey the following constraints:

- a) The nominal removal time of access unit n (with n > 0) from the CPB as specified in subclause C.1.2, satisfies the constraint that  $t_{r,n}(n) - t_r(n-1)$  is greater than or equal to  $\text{Max}( \text{svcPicSizeInMbs} + \text{MaxMBPS}, \text{fR} )$ , where MaxMBPS is the value specified in Table A-1 that applies to picture n – 1 and svcPicSizeInMbs is derived for picture n – 1.
- b) The difference between consecutive output times of pictures from the DPB as specified in subclause C.2.2, satisfies the constraint that  $\Delta t_{o,dpb}(n) \geq \text{Max}( \text{svcPicSizeInMbs} + \text{MaxMBPS}, \text{fR} )$ , where MaxMBPS is the value specified in Table A-1 for picture n, and svcPicSizeInMbs is derived for picture n, provided that picture n is a picture that is output and is not the last picture of the bitstream that is output.

- c)  $\text{PicWidthInMbs} * \text{FrameHeightInMbs} \leq \text{MaxFS}$ , where  $\text{MaxFS}$  is specified in Table A-1.  $\text{PicWidthInMbs}$  and  $\text{FrameHeightInMbs}$  are the derived variables for the layer representation with  $\text{DQId}$  equal to  $\text{dQIdmax}$ .
- d)  $\text{PicWidthInMbs} \leq \sqrt{\text{MaxFS} * 8}$ , where  $\text{MaxFS}$  is specified in Table A-1.  $\text{PicWidthInMbs}$  and  $\text{FrameHeightInMbs}$  are the derived variables for the layer representation with  $\text{DQId}$  equal to  $\text{dQIdmax}$ .
- e)  $\text{PicHeightInMbs} \leq \sqrt{\text{MaxFS} * 8}$ , where  $\text{MaxFS}$  is specified in Table A-1.  $\text{PicWidthInMbs}$  and  $\text{FrameHeightInMbs}$  are the derived variables for the layer representation with  $\text{DQId}$  equal to  $\text{dQIdmax}$ .
- f)  $\text{max\_dec\_frame\_buffering} \leq \text{MaxDpbSize}$ , where  $\text{MaxDpbSize}$  is equal to  $\text{Min}(1024 * \text{MaxDPB} / (\text{PicWidthInMbs} * \text{FrameHeightInMbs} * 384), 16)$  and  $\text{MaxDPB}$  is specified in Table A-1.  $\text{PicWidthInMbs}$  and  $\text{FrameHeightInMbs}$  are the derived variables for the layer representation with  $\text{DQId}$  equal to  $\text{dQIdmax}$ .
- g) Vertical motion vector component range does not exceed  $\text{MaxVmvR}$  in units of luma frame samples, where  $\text{MaxVmvR}$  is specified in Table A-1.
- h) Horizontal motion vector range does not exceed the range of -2048 to 2047.75, inclusive, in units of luma samples.
- i) Number of motion vectors per two consecutive macroblocks in decoding order (also applying to the total from the last macroblock of a slice and the first macroblock of the next slice in decoding order) does not exceed  $\text{MaxMvsPer2Mb}$ , where  $\text{MaxMvsPer2Mb}$  is specified in Table A-1. The number of motion vectors for each macroblock is value of the variable  $\text{MvCnt}$  after the completion of the target macroblock decoding process specified in subclause G.8.1.5.6.
- j) Number of bits of `macroblock_layer()` and `macroblock_layer_in_scalable_extension()` data for any macroblock is not greater than  $128 + \text{RawMbBits}$ . Depending on `entropy_coding_mode_flag`, the bits of `macroblock_layer()` data are counted as follows.
  - If `entropy_coding_mode_flag` is equal to 0, the number of bits of `macroblock_layer()` data is given by the number of bits in the `macroblock_layer()` syntax structure for a macroblock.
  - Otherwise (`entropy_coding_mode_flag` is equal to 1), the number of bits of `macroblock_layer()` data for a macroblock is given by the number of times `read_bits(1)` is called in subclauses 9.3.3.2.2 and 9.3.3.2.3 when parsing the `macroblock_layer()` associated with the macroblock.
- k) The variable `dependentDId` specified at the beginning of this subclause shall not exceed 3.
- l) When `refLayerDQId` is greater than or equal to 0, with `PicSizeInMbs` being the variable `PicSizeInMbs` for the layer representation with  $\text{DQId}$  equal to  $\text{dQIdMax}$ , with `refLayerPicSizeInMbs` being the variable `PicSizeInMbs` for the layer representation with  $\text{DQId}$  equal to `refLayerDQId`, with `mbType` specifying the array `mbType` of the collective term `currentVars` after completion of the SVC decoding process specified in subclause G.8, with `refLayerMbType` specifying the array `mbType` of the collective term `currentVars` after completion of the base decoding process for layer representations specified in subclauses G.8.1.3.1 and G.8.1.3.2 for the layer representation with  $\text{DQId}$  equal to `refLayerDQId`, the following applies.
  - If `PicSizeInMbs` is less than or equal to 1620, with `numIBLMbs` being the number of macroblocks `mbAddr` with `mbAddr = 0..(PicSizeInMbs - 1)` for which `mbType[mbAddr]` is equal to `I_BL` and with `numRefLayerIntraMbs` being the number of macroblocks `refLayerMbAddr` with `refLayerMbAddr = 0..(refLayerPicSizeInMbs - 1)` for which `refLayerMbType[refLayerMbAddr]` is equal to `I_PCM`, `I_16x16`, `I_8x8`, `I_4x4`, or `I_BL`, the following constraint shall be obeyed.
 
$$\text{numRefLayerIntraMbs} \leq \text{Ceil}(1.5 * \text{numIBLMbs} * \text{refLayerPicSizeInMbs} / \text{PicSizeInMbs}) \quad (\text{G-370})$$
  - Otherwise (`PicSizeInMbs` is greater than 1620), with `numIBLMbs` being the number of macroblocks `mbAddr` in a particular slice of the layer representation with  $\text{DQId}$  equal to  $\text{dQIdMax}$  for which `mbType[mbAddr]` is equal to `I_BL`, with `refLayerSlices` be the set of slices of the layer representation with  $\text{DQId}$  equal to `refLayerDQId` that are required for decoding the particular slice of the layer representation with  $\text{DQId}$  equal to  $\text{dQIdMax}$ , and with `numRefLayerIntraMbs` being the number of macroblocks `refLayerMbAddr` in the slices of the set `refLayerSlices` for which `refLayerMbType[refLayerMbAddr]` is equal to `I_PCM`, `I_16x16`, `I_8x8`, `I_4x4`, or `I_BL`, the constraint specified in Equation G-370 shall be obeyed.
- m) When `ref_layer_dq_id` is greater than or equal to 0 for a particular layer representation, the value of `level_idc` in the SVC sequence parameter set that is referenced by the particular layer representation shall be greater than or equal to the value of `level_idc` for the SVC sequence parameter set that is referenced by the layer representation with  $\text{DQId}$  equal to `ref_layer_dq_id`.

Table A-1 specifies the limits for each level. Entries marked "-" in Table A-1 denote the absence of a corresponding limit.

A level to which the bitstream conforms shall be indicated by the syntax element `level_idc` as follows.

- If `level_idc` is equal to 9, the indicated level is level 1b.
- Otherwise (`level_idc` is not equal to 9), `level_idc` is equal to a value of ten times the level number specified in Table A-1.

### G.10.2.2 Profile specific level limits

The variable `dqIdMax` is set equal to the maximum value of `DQId` for the layer representation of the access unit.

A list of integer values specifying layer representation identifiers for the access unit is derived by invoking the process specified in subclause G.8.1.1 with the output being the list `dqIdList`. The variable `numDQEntries` is set equal to the number of elements in the list `dqIdList`.

The variable `numSVCSlices` is derived as specified by the following pseudo-code.

```
numSVCSlices = 0
for( i = 0; i < numDQEntries; i++ )
    numSVCSlices += number of slices in layer representation with DQId equal to dqIdList[ i ]
```

(G-371)

The variable `svcPicSizeInMbs` is derived as specified in subclause G.10.2.1.

The following constraints are specified.

- a) In bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles, the removal time of access unit 0 shall satisfy the constraint that the number of slices `numSVCSlices` in picture 0 is less than or equal to  $((\text{svcPicSizeInMbs} + \text{MaxMBPS} * (t_r(0) - t_{r,n}(0))) \div \text{SliceRate})$ , where `SliceRate` is the value that applies to picture 0 and `MaxMBPS` is the value specified in Table A-1. For Scalable Baseline profile, `SliceRate` is specified in Table G-15. For Scalable High and Scalable High Intra profiles, `SliceRate` is specified in Table A-4.
- b) In bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles, the difference between consecutive removal time of access units  $n$  and  $n-1$  (with  $n > 0$ ) shall satisfy the constraint that the number of slices `numSVCSlices` in picture  $n$  is less than or equal to  $\text{MaxMBPS} * (t_r(n) - t_r(n-1)) \div \text{SliceRate}$ , where `SliceRate` is the value that applies to picture  $n$  and `MaxMBPS` is the value specified in Table A-1. For Scalable Baseline profile, `SliceRate` is specified in Table G-15. For Scalable High and Scalable High Intra profiles, `SliceRate` is specified in Table A-4.
- c) In bitstreams conforming to the Scalable High or Scalable High Intra profiles, SVC sequence parameter sets shall have `direct_8x8_inference_flag` equal to 1 for the levels specified in Table A-4. In bitstreams conforming to Scalable Baseline profile, SVC sequence parameter sets shall have `direct_8x8_inference_flag` equal to 1.
- d) In bitstreams conforming to the Scalable High or Scalable High Intra profiles, SVC sequence parameter sets shall have `frame_mbs_only_flag` equal to 1 for the levels specified in Table A-4.
- e) In bitstreams conforming to the Scalable High or Scalable High Intra profiles, for all macroblocks `mbAddr` and macroblock partitions `mbPartIdx`, the value of `subMbType[ mbAddr ][ mbPartIdx ]` that is derived as specified in subclause G.8.1.5.1.1 shall not be equal to `B_Bi_8x4`, `B_Bi_4x8`, or `B_Bi_4x4` for the levels in which `MinLumaBiPredSize` is shown as 8x8 in Table A-4. In bitstreams conforming to the Scalable Baseline profile, for all macroblocks `mbAddr` and macroblock partitions `mbPartIdx`, the value of `subMbType[ mbAddr ][ mbPartIdx ]` that is derived as specified in subclause G.8.1.5.1.1 shall not be equal to `B_Bi_8x4`, `B_Bi_4x8`, or `B_Bi_4x4`.
- f) In bitstreams conforming to the Scalable Baseline profile,  $(xInt_{\max} - xInt_{\min} + 6) * (yInt_{\max} - yInt_{\min} + 6) \leq \text{MaxSubMbRectSize}$  in macroblocks coded with macroblock type equal to `P_8x8`, `P_8x8ref0` or `B_8x8` for all invocations of the process specified in subclause 8.4.2.2.1 used to generate the predicted luma sample array for a single reference picture list (reference picture list 0 or reference picture list 1) for each 8x8 sub-macroblock, where `NumSubMbPart( sub_mb_type ) > 1`, where `MaxSubMbRectSize` is specified in Table G-15 and
  - `xIntmin` is the minimum value of `xIntL` among all luma sample predictions for the sub-macroblock
  - `xIntmax` is the maximum value of `xIntL` among all luma sample predictions for the sub-macroblock
  - `yIntmin` is the minimum value of `yIntL` among all luma sample predictions for the sub-macroblock

- $yInt_{max}$  is the maximum value of  $yInt_L$  among all luma sample predictions for the sub-macroblock
- g) In bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles, for the VCL HRD parameters,  $BitRate[ SchedSelIdx ] \leq cpbBrVclFactor * MaxBR$  and  $CpbSize[ SchedSelIdx ] \leq cpbBrVclFactor * MaxCPB$  for at least one value of  $SchedSelIdx$ , where  $cpbBrVclFactor$  is specified in Table G-16,  $BitRate[ SchedSelIdx ]$  is specified by Equation E-37 and  $CpbSize[ SchedSelIdx ]$  is specified by Equation E-38 when  $vcl\_hrd\_parameters\_present\_flag$  is equal to 1.  $MaxBR$  and  $MaxCPB$  are specified in Table A-1 in units of  $cpbBrVclFactor$  bits/s and  $cpbBrVclFactor$  bits, respectively. The bitstream shall satisfy these conditions for at least one value of  $SchedSelIdx$  in the range 0 to  $cpb\_cnt\_minus1$ , inclusive.
- h) In bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles, for the NAL HRD parameters,  $BitRate[ SchedSelIdx ] \leq cpbBrNalFactor * MaxBR$  and  $CpbSize[ SchedSelIdx ] \leq cpbBrNalFactor * MaxCPB$  for at least one value of  $SchedSelIdx$ , where  $cpbBrNalFactor$  is specified in Table G-16,  $BitRate[ SchedSelIdx ]$  is specified by Equation E-37 and  $CpbSize[ SchedSelIdx ]$  is specified by Equation E-38 when  $nal\_hrd\_parameters\_present\_flag$  is equal to 1.  $MaxBR$  and  $MaxCPB$  are specified in Table A-1 in units of  $cpbBrNalFactor$  bits/s and  $cpbBrNalFactor$  bits, respectively. The bitstream shall satisfy these conditions for at least one value of  $SchedSelIdx$  in the range 0 to  $cpb\_cnt\_minus1$ , inclusive.
- i) In bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles, the sum of the  $NumBytesInNALunit$  variables for access unit 0 is less than or equal to  $384 * ( svcPicSizeInMbs + MaxMBPS * ( t_r(0) - t_{r,n}(0) ) ) \div MinCR$ , where  $MaxMBPS$  and  $MinCR$  are the values specified in Table A-1 that apply to picture 0 and  $PicSizeInMbs$  is the number of macroblocks in picture 0.
- j) In bitstreams conforming to the Scalable Baseline, Scalable High, or Scalable High Intra profiles, the sum of the  $NumBytesInNALunit$  variables for access unit  $n$  (with  $n > 0$ ) is less than or equal to  $384 * MaxMBPS * ( t_r(n) - t_{r,n}(n-1) ) \div MinCR$ , where  $MaxMBPS$  and  $MinCR$  are the values specified in Table A-1 that apply to picture  $n$ .
- k) In bitstreams conforming to Scalable Baseline profile, picture parameter sets shall have `entropy_coding_mode_flag` equal to 0 and `transform_8x8_mode_flag` equal to 0 for level 2.1 and below.
- l) In bitstreams conforming to Scalable Baseline, Scalable High, and Scalable High Intra profiles, when  $PicSizeInMbs$  is greater than 1620 for  $DQId$  equal to  $dQIdMax$ , the number of macroblocks in any coded slice shall not exceed  $MaxFS / 4$ , where  $MaxFS$  is specified in Table A-1.

**Table G-15 – Scalable Baseline profile level limits**

| Level number | SliceRate | MaxSubMbRectSize |
|--------------|-----------|------------------|
| 1            | -         | 576              |
| 1b           | -         | 576              |
| 1.1          | -         | 576              |
| 1.2          | -         | 576              |
| 1.3          | -         | 576              |
| 2            | -         | 576              |
| 2.1          | 22        | 576              |
| 2.2          | 22        | 576              |
| 3            | 22        | 576              |
| 3.1          | 60        | 1152             |
| 3.2          | 60        | 1152             |
| 4            | 60        | 1440             |
| 4.1          | 24        | 1440             |
| 4.2          | 24        | 1440             |
| 5            | 24        | -                |
| 5.1          | 24        | -                |

**Table G-16 – Specification of cpbBrVclFactor and cpbBrNalFactor**

| Profile                                                       | cpbBrVclFactor | cpbBrNalFactor |
|---------------------------------------------------------------|----------------|----------------|
| Scalable Baseline<br>Scalable High, or<br>Scalable High Intra | 1250           | 1500           |

#### **G.11 Byte stream format**

The specifications in Annex B apply.

#### **G.12 Hypothetical reference decoder**

The specifications in Annex C apply with substituting SVC sequence parameter set for sequence parameter set.

#### **G.13 Supplemental enhancement information**

The specifications in Annex D together with the extensions and modifications specified in this subclause apply.

##### **G.13.1 SEI payload syntax**

##### **G.13.1.1 Scalability information SEI message syntax**

| scalability_info( payloadSize ) {                                     | C | Descriptor |
|-----------------------------------------------------------------------|---|------------|
| temporal_id_nesting_flag                                              | 5 | u(1)       |
| priority_layer_info_present_flag                                      | 5 | u(1)       |
| priority_id_setting_flag                                              | 5 | u(1)       |
| num_layers_minus1                                                     | 5 | ue(v)      |
| for( i = 0; i <= num_layers_minus1; i++ ) {                           |   |            |
| layer_id[ i ]                                                         | 5 | ue(v)      |
| priority_id[ i ]                                                      | 5 | u(6)       |
| discardable_flag[ i ]                                                 | 5 | u(1)       |
| dependency_id[ i ]                                                    | 5 | u(3)       |
| quality_id[ i ]                                                       | 5 | u(4)       |
| temporal_id[ i ]                                                      | 5 | u(3)       |
| sub_pic_layer_flag[ i ]                                               | 5 | u(1)       |
| sub_region_layer_flag[ i ]                                            | 5 | u(1)       |
| iroi_division_info_present_flag[ i ]                                  | 5 | u(1)       |
| profile_level_info_present_flag[ i ]                                  | 5 | u(1)       |
| bitrate_info_present_flag[ i ]                                        | 5 | u(1)       |
| frm_rate_info_present_flag[ i ]                                       | 5 | u(1)       |
| frm_size_info_present_flag[ i ]                                       | 5 | u(1)       |
| layer_dependency_info_present_flag[ i ]                               | 5 | u(1)       |
| parameter_sets_info_present_flag[ i ]                                 | 5 | u(1)       |
| bitstream_restriction_info_present_flag[ i ]                          | 5 | u(1)       |
| exact_inter_layer_pred_flag[ i ]                                      | 5 | u(1)       |
| if( sub_pic_layer_flag[ i ]    iroi_division_info_present_flag[ i ] ) |   |            |
| exact_sample_value_match_flag[ i ]                                    | 5 | u(1)       |
| layer_conversion_flag[ i ]                                            | 5 | u(1)       |

|                                                                                    |   |       |
|------------------------------------------------------------------------------------|---|-------|
| <b>layer_output_flag[ i ]</b>                                                      | 5 | u(1)  |
| if (profile_level_info_present_flag[ i ])                                          |   |       |
| <b>layer_profile_level_idc[ i ]</b>                                                | 5 | u(24) |
| if( bitrate_info_present_flag[ i ] ) {                                             |   |       |
| <b>avg_bitrate[ i ]</b>                                                            | 5 | u(16) |
| <b>max_bitrate_layer[ i ]</b>                                                      | 5 | u(16) |
| <b>max_bitrate_layer_representation[ i ]</b>                                       | 5 | u(16) |
| <b>max_bitrate_calc_window[ i ]</b>                                                | 5 | u(16) |
| }                                                                                  |   |       |
| if( frm_rate_info_present_flag[ i ] ) {                                            |   |       |
| <b>constant_frm_rate_idc[ i ]</b>                                                  | 5 | u(2)  |
| <b>avg_frm_rate[ i ]</b>                                                           | 5 | u(16) |
| }                                                                                  |   |       |
| if( frm_size_info_present_flag[ i ]   <br>iroi_division_info_present_flag[ i ] ) { |   |       |
| <b>frm_width_in_mbs_minus1[ i ]</b>                                                | 5 | ue(v) |
| <b>frm_height_in_mbs_minus1[ i ]</b>                                               | 5 | ue(v) |
| }                                                                                  |   |       |
| if( sub_region_layer_flag[ i ] ) {                                                 |   |       |
| <b>base_region_layer_id[ i ]</b>                                                   | 5 | ue(v) |
| <b>dynamic_rect_flag[ i ]</b>                                                      | 5 | u(1)  |
| if( !dynamic_rect_flag[ i ] ) {                                                    |   |       |
| <b>horizontal_offset[ i ]</b>                                                      | 5 | u(16) |
| <b>vertical_offset[ i ]</b>                                                        | 5 | u(16) |
| <b>region_width[ i ]</b>                                                           | 5 | u(16) |
| <b>region_height[ i ]</b>                                                          | 5 | u(16) |
| }                                                                                  |   |       |
| }                                                                                  |   |       |
| if( sub_pic_layer_flag[ i ] )                                                      |   |       |
| <b>roi_id[ i ]</b>                                                                 | 5 | ue(v) |
| if( iroi_division_info_present_flag[ i ] ) {                                       |   |       |
| <b>iroi_grid_flag[ i ]</b>                                                         | 5 | u(1)  |
| if( iroi_grid_flag[ i ] ) {                                                        |   |       |
| <b>grid_width_in_mbs_minus1[ i ]</b>                                               | 5 | ue(v) |
| <b>grid_height_in_mbs_minus1[ i ]</b>                                              | 5 | ue(v) |
| } else {                                                                           |   |       |
| <b>num_rois_minus1[ i ]</b>                                                        | 5 | ue(v) |
| for( j = 0; j <= num_rois_minus1[ i ]; j++ ) {                                     |   |       |
| <b>first_mb_in_roi[ i ][ j ]</b>                                                   | 5 | ue(v) |
| <b>roi_width_in_mbs_minus1[ i ][ j ]</b>                                           | 5 | ue(v) |
| <b>roi_height_in_mbs_minus1[ i ][ j ]</b>                                          | 5 | ue(v) |
| }                                                                                  |   |       |
| }                                                                                  |   |       |
| }                                                                                  |   |       |
| if( layer_dependency_info_present_flag[ i ] ) {                                    |   |       |
| <b>num_directly_dependent_layers[ i ]</b>                                          | 5 | ue(v) |
| for( j = 0; j < num_directly_dependent_layers[ i ]; j++ )                          |   |       |
| <b>directly_dependent_layer_id_delta_minus1[ i ][ j ]</b>                          | 5 | ue(v) |

|                                                                  |   |       |
|------------------------------------------------------------------|---|-------|
| } else                                                           |   |       |
| layer_dependency_info_src_layer_id_delta[ i ]                    | 5 | ue(v) |
| if( parameter_sets_info_present_flag[ i ] ) {                    |   |       |
| num_seq_parameter_set_minus1[ i ]                                | 5 | ue(v) |
| for( j = 0; j <= num_seq_parameter_set_minus1[ i ]; j++ )        |   |       |
| seq_parameter_set_id_delta[ i ][ j ]                             | 5 | ue(v) |
| num_subset_seq_parameter_set_minus1[ i ]                         | 5 | ue(v) |
| for( j = 0; j <= num_subset_seq_parameter_set_minus1[ i ]; j++ ) |   |       |
| subset_seq_parameter_set_id_delta[ i ][ j ]                      | 5 | ue(v) |
| num_pic_parameter_set_minus1[ i ]                                | 5 | ue(v) |
| for( j = 0; j <= num_pic_parameter_set_minus1[ i ]; j++ )        |   |       |
| pic_parameter_set_id_delta[ i ][ j ]                             | 5 | ue(v) |
| } else                                                           |   |       |
| parameter_sets_info_src_layer_id_delta[ i ]                      | 5 | ue(v) |
| if( bitstream_restriction_info_present_flag[ i ] ) {             |   |       |
| motion_vectors_over_pic_boundaries_flag[ i ]                     | 5 | u(1)  |
| max_bytes_per_pic_denom[ i ]                                     | 5 | ue(v) |
| max_bits_per_mb_denom[ i ]                                       | 5 | ue(v) |
| log2_max_mv_length_horizontal[ i ]                               | 5 | ue(v) |
| log2_max_mv_length_vertical[ i ]                                 | 5 | ue(v) |
| num_reorder_frames[ i ]                                          | 5 | ue(v) |
| max_dec_frame_buffering[ i ]                                     | 5 | ue(v) |
| }                                                                |   |       |
| if( layer_conversion_flag[ i ] ) {                               |   |       |
| conversion_type_idc[ i ]                                         | 5 | ue(v) |
| for( j=0; j < 2; j++ ) {                                         |   |       |
| rewriting_info_flag[ i ][ j ]                                    | 5 | u(1)  |
| if( rewriting_info_flag[ i ][ j ] ) {                            |   |       |
| rewriting_profile_level_idc[ i ][ j ]                            | 5 | u(24) |
| rewriting_avg_bitrate[ i ][ j ]                                  | 5 | u(16) |
| rewriting_max_bitrate[ i ][ j ]                                  | 5 | u(16) |
| }                                                                |   |       |
| }                                                                |   |       |
| }                                                                |   |       |
| }                                                                |   |       |
| if( priority_layer_info_present_flag ) {                         |   |       |
| pr_num_dId_minus1                                                | 5 | ue(v) |
| for( i = 0; i <= pr_num_dId_minus1; i++ ) {                      |   |       |
| pr_dependency_id[ i ]                                            | 5 | u(3)  |
| pr_num_minus1[ i ]                                               | 5 | ue(v) |
| for( j = 0; j <= pr_num_minus1; j++ ) {                          |   |       |
| pr_id[ i ][ j ]                                                  | 5 | ue(v) |
| pr_profile_level_idc[ i ][ j ]                                   | 5 | u(24) |
| pr_avg_bitrate[ i ][ j ]                                         | 5 | u(16) |
| pr_max_bitrate[ i ][ j ]                                         | 5 | u(16) |
| }                                                                |   |       |
| }                                                                |   |       |
| }                                                                |   |       |
| }                                                                |   |       |

|                                                                    |   |      |
|--------------------------------------------------------------------|---|------|
| if( priority_id_setting_flag ) {                                   |   |      |
| PriorityIdSettingUriIdx = 0                                        |   |      |
| do                                                                 |   |      |
| <b>priority_id_setting_uri</b> [ PriorityIdSettingUriIdx ]         | 5 | b(8) |
| while( priority_id_setting_uri[ PriorityIdSettingUriIdx++ ] != 0 ) |   |      |
| }                                                                  |   |      |
| }                                                                  |   |      |

#### G.13.1.2 Layers not present SEI message syntax

|                                     |          |                   |
|-------------------------------------|----------|-------------------|
| layers_not_present( payloadSize ) { | <b>C</b> | <b>Descriptor</b> |
| <b>num_layers</b>                   | 5        | ue(v)             |
| for( i = 0; i < num_layers; i++ ) { |          |                   |
| <b>layer_id</b> [ i ]               | 5        | u(8)              |
| }                                   |          |                   |
| }                                   |          |                   |

#### G.13.1.3 Layer dependency change SEI message syntax

|                                                              |          |                   |
|--------------------------------------------------------------|----------|-------------------|
| layer_dependency_change ( payloadSize ) {                    | <b>C</b> | <b>Descriptor</b> |
| <b>num_layers_minus1</b>                                     | 5        | ue(v)             |
| for( i = 0; i <= num_layers_minus1; i++ ) {                  |          |                   |
| <b>layer_id</b> [ i ]                                        | 5        | u(8)              |
| <b>layer_dependency_info_present_flag</b> [ i ]              | 5        | u(1)              |
| if (layer_dependency_info_present_flag[ i ] ) {              |          |                   |
| <b>num_directly_dependent_layers</b> [ i ]                   | 5        | ue(v)             |
| for ( j = 0; j < num_directly_dependent_layers[ i ]; j++ )   |          |                   |
| <b>directly_dependent_layer_id_delta_minus1</b> [ i ][ j ]   | 5        | ue(v)             |
| } else {                                                     |          |                   |
| <b>layer_dependency_info_src_layer_id_delta_minus1</b> [ i ] | 5        | ue(v)             |
| }                                                            |          |                   |
| }                                                            |          |                   |
| }                                                            |          |                   |

#### G.13.1.4 Sub-picture scalable layer SEI message syntax

|                                         |          |                   |
|-----------------------------------------|----------|-------------------|
| sub_pic_scalable_layer( payloadSize ) { | <b>C</b> | <b>Descriptor</b> |
| <b>layer_id</b>                         | 5        | ue(v)             |
| }                                       |          |                   |

### G.13.1.5 Non-required layer representation SEI message syntax

|                                                                  |          |                   |
|------------------------------------------------------------------|----------|-------------------|
| non_required_layer_rep( payloadSize ) {                          | <b>C</b> | <b>Descriptor</b> |
| num_info_entries_minus1                                          | 5        | ue(v)             |
| for( i = 0; i <= num_info_entries_minus1; i++ ) {                |          |                   |
| entry_dependency_id[ i ]                                         | 5        | u(3)              |
| num_non_required_layer_rep_minus1[ i ]                           | 5        | ue(v)             |
| for( j = 0; j <= num_non_required_layer_rep_minus1[ i ]; j++ ) { |          |                   |
| non_required_layer_rep_dependency_id[ i ][ j ]                   | 5        | u(3)              |
| non_required_layer_rep_quality_id[ i ][ j ]                      | 5        | u(4)              |
| }                                                                |          |                   |
| }                                                                |          |                   |
| }                                                                |          |                   |

### G.13.1.6 Priority layer information SEI message syntax

|                                           |          |                   |
|-------------------------------------------|----------|-------------------|
| priority_layer_info( payloadSize ) {      | <b>C</b> | <b>Descriptor</b> |
| pr_dependency_id                          | All      | u(3)              |
| num_priority_ids                          | 5        | u(4)              |
| for( i = 0; i < num_priority_ids; i++ ) { |          |                   |
| alt_priority_id[ i ]                      | 5        | u(6)              |
| }                                         |          |                   |
| }                                         |          |                   |

### G.13.1.7 Scalable nesting SEI message syntax

|                                                            |          |                   |
|------------------------------------------------------------|----------|-------------------|
| scalable_nesting( payloadSize ) {                          | <b>C</b> | <b>Descriptor</b> |
| all_layer_representations_in_au_flag                       | 5        | u(1)              |
| if( all_layer_representations_in_au_flag == 0 ) {          |          |                   |
| num_layer_representations_minus1                           | 5        | ue(v)             |
| for( i = 0; i <= num_layer_representations_minus1; i++ ) { |          |                   |
| sei_dependency_id[ i ]                                     | 5        | u(3)              |
| sei_quality_id[ i ]                                        | 5        | u(4)              |
| }                                                          |          |                   |
| sei_temporal_id                                            | 5        | u(3)              |
| }                                                          |          |                   |
| while( !byte_aligned( ) )                                  |          |                   |
| sei_nesting_zero_bit /* equal to 0 */                      | 5        | f(1)              |
| do                                                         |          |                   |
| sei_message( )                                             | 5        |                   |
| while( more_rbsp_data( ) )                                 |          |                   |
| }                                                          |          |                   |

### G.13.1.8 Base layer temporal HRD SEI message syntax

|                                                                                       |          |                   |
|---------------------------------------------------------------------------------------|----------|-------------------|
| base_layer_temporal_hrd ( payloadSize ) {                                             | <b>C</b> | <b>Descriptor</b> |
| <b>num_of_temporal_layers_in_base_layer_minus1</b>                                    | 5        | ue(v)             |
| for( i = 0; i < num_of_temporal_layers_in_base_layer_minus1; i++){                    |          |                   |
| <b>temporal_id[ i ]</b>                                                               | 5        | u(3)              |
| <b>timing_info_present_flag[ i ]</b>                                                  | 5        | u(1)              |
| if( timing_info_present_flag[ i ] ) {                                                 |          |                   |
| <b>num_units_in_tick[ i ]</b>                                                         | 5        | u(32)             |
| <b>time_scale[ i ]</b>                                                                | 5        | u(32)             |
| <b>fixed_frame_rate_flag[ i ]</b>                                                     | 5        | u(1)              |
| }                                                                                     |          |                   |
| <b>nal_hrd_parameters_present_flag[ i ]</b>                                           | 5        | u(1)              |
| if( nal_hrd_parameters_present_flag[ i ] )                                            |          |                   |
| hrd_parameters()                                                                      |          |                   |
| <b>vcl_hrd_parameters_present_flag[ i ]</b>                                           | 5        | u(1)              |
| if( vcl_hrd_parameters_present_flag[ i ] )                                            |          |                   |
| hrd_parameters()                                                                      |          |                   |
| if( nal_hrd_parameters_present_flag[ i ]   <br>vcl_hrd_parameters_present_flag[ i ] ) |          |                   |
| <b>low_delay_hrd_flag[ i ]</b>                                                        | 5        | u(1)              |
| <b>pic_struct_present_flag[ i ]</b>                                                   | 5        | u(1)              |
| }                                                                                     |          |                   |
| }                                                                                     |          |                   |

### G.13.1.9 Quality layer integrity check SEI message syntax

|                                                   |          |                   |
|---------------------------------------------------|----------|-------------------|
| quality_layer_integrity_check( payloadSize ) {    | <b>C</b> | <b>Descriptor</b> |
| <b>num_info_entries_minus1</b>                    | 5        | ue(v)             |
| for( i = 0; i <= num_info_entries_minus1; i++ ) { |          |                   |
| <b>entry_dependency_id[ i ]</b>                   | 5        | u(3)              |
| <b>quality_layer_crc[ i ]</b>                     | 5        | u(16)             |
| }                                                 |          |                   |
| }                                                 |          |                   |

#### G.13.1.10 Redundant picture property SEI message syntax

| redundant_pic_property( payloadSize ) {                       | C | Descriptor |
|---------------------------------------------------------------|---|------------|
| <b>num_dId_minus1</b>                                         | 5 | ue(v)      |
| for( i = 0; i <= num_dId_minus1; i++ ) {                      |   |            |
| <b>dependency_id[ i ]</b>                                     | 5 | u(3)       |
| <b>num_qId_minus1[ i ]</b>                                    | 5 | ue(v)      |
| for( j = 0; j <= num_qId_minus1[ i ]; j++ ) {                 |   |            |
| <b>quality_id[ i ][ j ]</b>                                   | 5 | u(4)       |
| <b>num_redundant_pics_minus1[ i ][ j ]</b>                    | 5 | ue(v)      |
| for( k = 0; k <= num_redundant_pics_minus1[ i ][ j ]; k++ ) { |   |            |
| <b>redundant_pic_cnt_minus1[ i ][ j ][ k ]</b>                | 5 | ue(v)      |
| <b>pic_match_flag[ i ][ j ][ k ]</b>                          | 5 | u(1)       |
| if( !pic_match_flag[ i ][ j ][ k ] ) {                        |   |            |
| <b>mb_type_match_flag[ i ][ j ][ k ]</b>                      | 5 | u(1)       |
| <b>motion_match_flag[ i ][ j ][ k ]</b>                       | 5 | u(1)       |
| <b>residual_match_flag[ i ][ j ][ k ]</b>                     | 5 | u(1)       |
| <b>intra_samples_match_flag[ i ][ j ][ k ]</b>                | 5 | u(1)       |
| }                                                             |   |            |
| }                                                             |   |            |
| }                                                             |   |            |
| }                                                             |   |            |
| }                                                             |   |            |

#### G.13.1.11 Temporal level zero dependency representation index SEI message syntax

| tl0_dep_rep_index( payloadSize ) { | C | Descriptor |
|------------------------------------|---|------------|
| <b>tl0_dep_rep_idx</b>             | 5 | u(8)       |
| <b>effective_idr_pic_id</b>        | 5 | u(16)      |
| }                                  |   |            |

#### G.13.1.12 Temporal level switching point SEI message syntax

| tl_switching_point( payloadSize ) { | C | Descriptor |
|-------------------------------------|---|------------|
| <b>delta_frame_num</b>              | 5 | se(v)      |
| }                                   |   |            |

#### G.13.2 SEI payload semantics

Depending on payloadType, the corresponding SEI message semantics are extended as follows.

- If payloadType is equal to 3, 8, 19, 20, or 22, the following applies.
  - If the SEI message is not included in a scalable nesting SEI message, it applies to the layer representation of the current access unit with dependency\_id and quality\_id equal to 0.
  - Otherwise (the SEI message is included in a scalable nesting SEI message), it applies to all layer representations of the current access unit with DQId equal to (( sei\_dependency\_id[ i ] << 4 ) + sei\_quality\_id[ i ]) for any i in the range of 0 to num\_layer\_representations\_minus1, inclusive.

- Otherwise, if payloadType is equal to 2, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 21, 23, 25, 27, 34, or 35, the following applies.
  - If the SEI message is not included in a scalable nesting SEI message, it applies to the dependency representation of the current access unit with dependency\_id equal to 0.
  - Otherwise (the SEI message is included in a scalable nesting SEI message), the value of sei\_quality\_id[ i ] shall be equal to 0 for any i in the range of 0 to num\_layer\_representations\_minus1, inclusive, and it applies to all dependency representations of the current access unit with dependency\_id equal to sei\_dependency\_id[ i ] for any i in the range of 0 to num\_layer\_representations\_minus1, inclusive. When payloadType is equal to 10 for the SEI message that is included in a scalable nesting SEI message, the semantics for sub\_seq\_layer\_num of the sub-sequence information SEI message is modified as follows.  
**sub\_seq\_layer\_num** specifies the sub-sequence layer number of the current picture. When the current picture resides in a sub-sequence whose first picture in decoding order is an IDR picture, the value of sub\_seq\_layer\_num shall be equal to 0. For a non-paired reference field, the value of sub\_seq\_layer\_num shall be equal to 0. sub\_seq\_layer\_num shall be in the range of 0 to 255, inclusive.
- Otherwise, if payloadType is equal to 0 or 1, the following applies.
  - If the SEI message is not included in a scalable nesting SEI message, the following applies. When the SEI message and all other SEI messages with payloadType equal to 0 or 1 not included in a scalable nesting SEI message are used as the buffering period and picture timing SEI messages for checking the bitstream conformance according to Annex C and the decoding process specified in clauses 2-9 of this Recommendation | International Standard is used, the bitstream shall be conforming to this Recommendation | International Standard.
  - Otherwise (the SEI message is included in a scalable nesting SEI message), the following applies. When the SEI message and all other SEI messages with payloadType equal to 0 or 1 included in a scalable nesting SEI message with identical values of sei\_temporal\_id, sei\_dependency\_id[ i ], and sei\_quality\_id[ i ] are used as the buffering period and picture timing SEI messages for checking the bitstream conformance according to Annex C, the bitstream that would be obtained by invoking the bitstream extraction process as specified in subclause G.8.8.1 with tIdTarget equal to sei\_temporal\_id, dIdTarget equal to sei\_dependency\_id[ i ], and qIdTarget equal to sei\_quality\_id[ i ] shall be conforming to this Recommendation | International Standard.
- Otherwise (all remaining payloadType values), the corresponding SEI message semantics are not extended.

When an SEI NAL unit contains an SEI message with payloadType in the range of 24 to 35, inclusive, it shall not contain any SEI messages with payloadType less than 24 and the first SEI message in the SEI NAL unit shall have payloadType in the range of 24 to 35, inclusive.

When a scalable nesting SEI message (payloadType is equal to 30) is present in an SEI NAL unit, it shall be the only SEI message in the SEI NAL unit.

### G.13.2.1 Scalability information SEI message semantics

Scalability information SEI messages contain scalability information for subsets of the bitstream. Each of these subsets is referred to as a scalable layer and is associated with a layer identifier. A scalable layer represents a set of coded slice NAL units with the same values of dependency\_id, quality\_id, and temporal\_id and associated non-VCL NAL units. The bitstream subset that is required for decoding the scalable layer is referred to as the scalable layer representation or the representation of the scalable layer. Information such as bit rate, frame rate, and spatial resolution, among others, are signalled for the representation of each scalable layer as specified in the following.

Each scalable layer is associated with a layer identifier. A larger value of the layer identifier indicates a higher scalable layer. The scalable layer with layer identifier equal to 0 is the lowest scalable layer or scalable layer 0. Decoding of a scalable layer is independent of any higher scalable layer but may be dependent on a lower scalable layer. Therefore, the lowest scalable layer can be decoded independently, decoding of scalable layer 1 may be dependent on scalable layer 0, decoding of scalable layer 2 may be dependent on scalable layers 0 and 1, and so on. Decoding of a scalable layer requires the presence of the scalable layer itself and all the lower scalable layers the scalable layer directly or indirectly depends on. Scalable layer A is directly dependent on scalable layer B if there is at least one coded slice in scalable layer A that refers to one or more coded slices of scalable layer B through inter prediction or inter-layer prediction. Scalable layer A is indirectly dependent on layer B if scalable layer A is not directly dependent on scalable layer B but decoding of scalable layer A requires the presence of at one or more coded slices of the layer B.

NOTE 1 – For example, layer identifier values may be allocated as equal to  $(((\text{dependency\_id} \ll 4) + \text{quality\_id}) * 8 + \text{temporal\_id})$ .

When present, a scalability information SEI message shall appear in an access unit for which all the VCL NAL units have `nal_unit_type` equal to 5 or `idr_flag` equal to 1. The contents of the scalability information SEI message with the exception of the layer dependency information, for which a change may be signalled by the presence of one or more layer dependency change SEI messages, apply for all access units until, but excluding, the access unit that contains the next scalability information SEI message. The set of access units for which the current SEI message applies is referred to as the target access unit set. Layer dependency information for a particular scalable layer is specified by the syntax elements `num_directly_dependent_layers`, `directly_dependent_layer_id_delta_minus1`, and `layer_dependency_info_src_layer_id_delay`. When a scalability information SEI message specifies that a scalable layer A does not directly or indirectly depend on a scalable layer B, this relationship applies to the complete target access unit set. When a scalability information SEI message specifies that a scalable layer A does directly or indirectly depend on a scalable layer B, a following layer dependency change SEI message may specify that this dependency does not apply for a subset of the target access unit set.

**temporal\_id\_nesting\_flag** specifies whether inter prediction is additionally restricted for the target access unit set. Dependent on **temporal\_id\_nesting\_flag**, the following applies.

- If **temporal\_id\_nesting\_flag** is equal to 0, additional constraints may not obeyed.
- Otherwise (**temporal\_id\_nesting\_flag** is equal to 1), the following constraint shall be obeyed for layer representations with any combination of `dependency_id` and `quality_id` values present in the target access unit set.

For each access unit `auA` with `temporal_id` equal to `tIdA`, an access unit `auB` with `temporal_id` equal to `tIdB` and `tIdB` less than or equal to `tIdA` shall not be referenced by inter prediction when there exists an access unit `auC` with `temporal_id` equal to `tIdC` and `tIdC` less than `tIdB`, which follows the access unit `auB` and precedes the access unit `auA` in decoding order.

NOTE 2 – The syntax element **temporal\_id\_nesting\_flag** is used to indicate that temporal up-switching, i.e., switching from decoding of up to a specific `temporal_id` `tIdN` to decoding up to a `temporal_id` `tIdM` > `tIdN`, is always possible.

**priority\_layer\_info\_present\_flag** equal to 1 specifies that priority layer information SEI messages are present in the target access unit set and characteristic information of priority layers is present in the current scalability information SEI message. **priority\_layer\_info\_present\_flag** equal to 0 specifies that characteristic information of priority layers is not present in the current scalability information SEI message.

**priority\_id\_setting\_flag** equal to 1 specifies that the syntax element `priority_id_setting_uri` is present and the `priority_id` value calculation method description is known. **priority\_id\_setting\_flag** equal to 0 specifies that the syntax element `priority_id_setting_uri` is not present.

**num\_layers\_minus1** plus 1 specifies the number of scalable layers contained in the target access unit set. The value of **num\_layers\_minus1** shall be in the range of 0 to 2047, inclusive.

**layer\_id[i]** specifies the layer identifier of the *i*-th scalable layer specified in the SEI message. **layer\_id[i]** shall be in the range of 0 to 2047, inclusive.

From here to the end of this subclause, the scalable layer with layer identifier equal to the current value of **layer\_id[i]** is referred to as the current scalable layer, and the representation of the current scalable layer is referred to as the current scalable layer representation.

**priority\_id[i]** specifies that all VCL NAL units of the current scalable layer representation have a value of `priority_id` that is less than or equal to **priority\_id[i]**.

**discardable\_flag[i]** equal to 1 specifies that all VCL NAL units of the current scalable layer have `discardable_flag` equal to 1. **discardable\_flag[i]** equal to 0 specifies that the current scalable layer may contain a VCL NAL units with `discardable_flag` equal to 0.

**dependency\_id[i]**, **quality\_id[i]**, and **temporal\_id[i]** are equal to `dependency_id`, `quality_id`, and `temporal_id`, respectively, of the NAL units in the current scalable layer. The value of `dependency_id` is identical in all VCL NAL units of the current scalable layer. The value of `quality_id` is identical in all VCL NAL units of the current scalable layer. The value of `temporal_id` is identical in all VCL NAL units of the current scalable layer.

**sub\_pic\_layer\_flag[i]** equal to 1 specifies that the current scalable layer consists of sub-pictures. Each sub-picture consists of a proper subset of coded slices of a layer representation. **sub\_pic\_layer\_flag[i]** equal to 0 specifies that no sub-pictures are associated with the current scalable layer in this SEI message.

A scalable layer consisting of sub-pictures is also referred to as a sub-picture layer. Any coded slice that is not included in the scalable layer representation of a sub-picture layer shall not be referred to in inter prediction or inter-layer prediction for decoding of the sub-picture layer.

NOTE 3 – The mapping of each sub-picture to a scalable layer may be signalled by the sub-picture scalable layer information SEI message.

**sub\_region\_layer\_flag[i]** equal to 1 specifies that the current scalable layer consists of a region as specified later in the semantics of the scalability information SEI message. **sub\_region\_layer\_flag[i]** equal to 0 specifies that the current scalable layer does not represent a region as specified later in the semantics of the scalability information SEI message.

**iroi\_division\_info\_present\_flag[i]** equal to 1 specifies that the region that is represented by the current scalable layer is divided along macroblock boundaries into multiple rectangular regions-of-interest, referred to as interactive regions-of-interest (IROIs), and the IROI division information is explicitly signalled in the scalability information SEI message. **iroi\_division\_info\_present\_flag[i]** equal to 0 specifies that the IROI division information for the current scalable layer is not present in the scalability information SEI message.

All coded slices covering a subset of the region of an IROI form the IROI. Any coded slice that is not included in the scalable layer representation of the IROI shall not be referred to in inter prediction or inter-layer prediction for decoding of the IROI.

**profile\_level\_info\_present\_flag[i]** equal to 1 specifies that **profile\_idc**, **constraint\_set0\_flag**, **constraint\_set1\_flag**, **constraint\_set2\_flag**, **constraint\_set3\_flag**, and **level\_idc** applicable for the current scalable layer representation are specified by the value of **layer\_profile\_level\_idc[i]** as specified subsequently.

When **profile\_level\_info\_present\_flag[i]** is equal to 0, **profile\_idc**, **constraint\_set0\_flag**, **constraint\_set1\_flag**, **constraint\_set2\_flag**, **constraint\_set3\_flag**, and **level\_idc** applicable for the current scalable layer representation are not present in the scalability information SEI message.

**bitrate\_info\_present\_flag[i]** equal to 1 specifies that the bit rate information for the current scalable layer representation is present in the scalability information SEI message. **bitrate\_info\_present\_flag[i]** equal to 0 specifies that the bit rate information for the current scalable layer representation is not present in the scalability information SEI message.

**frm\_rate\_info\_present\_flag[i]** equal to 1 specifies that the frame rate information for the current scalable layer is present in the scalability information SEI message. **frm\_rate\_info\_present\_flag[i]** equal to 0 specifies that the frame rate information for the current scalable layer is not present in the scalability information SEI message.

**frm\_size\_info\_present\_flag[i]** equal to 1 specifies that the frame size information for the current scalable layer is present in the scalability information SEI message. **frm\_size\_info\_present\_flag[i]** equal to 0 specifies that the frame size information for the current scalable layer is not present in the scalability information SEI message.

**layer\_dependency\_info\_present\_flag[i]** equal to 1 specifies that the layer dependency information for the current scalable layer is present in the scalability information SEI message. **layer\_dependency\_info\_present\_flag[i]** equal to 0 specifies that **layer\_dependency\_info\_src\_layer\_id\_delta[i]** is present in the scalability information SEI message.

**parameter\_sets\_info\_present\_flag[i]** equal to 1 specifies that the values of **seq\_parameter\_set\_id** of the sequence parameter set and subset sequence parameter set and the values of **pic\_parameter\_set\_id** of the picture parameter sets that are referred to by the VCL NAL units of the current scalable layer representation are present in the scalability information SEI message. **parameter\_sets\_info\_present\_flag[i]** equal to 0 specifies that **parameter\_sets\_info\_src\_layer\_id\_delta[i]** is present in the scalability information SEI message.

**bitstream\_restriction\_info\_present\_flag[i]** equal to 1 specifies that the bitstream restriction information for the current scalable layer is present in the scalability information SEI message. **bitstream\_restriction\_info\_present\_flag[i]** equal to 0 specifies that the bitstream restriction information for the current scalable layer is not present in the scalability information SEI message.

**exact\_inter\_layer\_pred\_flag[i]** equal to 1 specifies that there is no mismatch in the inter-layer prediction reference signal when decoding the current scalable layer representation compared to the inter-layer prediction reference signal when the scalable layer representation was encoded. **exact\_inter\_layer\_pred\_flag[i]** equal to 0 specifies that a mismatch may exist.

NOTE 4 – The mismatch of the inter-layer prediction reference signal for encoding and decoding may be present, for example, because NAL units with **quality\_id** greater than 0 that were used for inter-prediction during the encoding process are not included in the current scalable layer representation.

**exact\_sample\_value\_match\_flag[i]** specifies the exact sample value match of the sub-picture layer or the IROIs within the current scalable layer.

When **sub\_pic\_layer\_flag[i]** is equal to 1, the following applies.

- Let **scalLayerRepA** be the set of all layer representations of the target access unit set that contain coded slices of the sub-picture layer.
- **exact\_sample\_value\_match\_flag[i]** equal to 1 specifies that the value of each reconstructed sample for decoding the sub-picture layer representation is exactly the same as the value of the sample at the same location that would be reconstructed when all the coded slices in **scalLayerRepA** were decoded.

- `exact_sample_value_match_flag[i]` equal to 0 specifies that the value of each reconstructed sample for decoding the sub-picture layer representation need not be the same as the value of the sample at the same location that would be reconstructed when all the coded slices in `scalLayerRepA` were decoded.

When `iroi_division_info_present_flag[i]` is equal to 1, for each IROI, the following applies.

- Let `scalLayerRepB` be the set of all the layer representations of the target access unit set that contain coded slices of the IROI.
- `exact_sample_value_match_flag[i]` equal to 1 specifies that the value of each reconstructed sample for decoding the IROI is exactly the same as the value of the sample at the same location that would be reconstructed when all the coded slices in `scalLayerRepB` were decoded.
- `exact_sample_value_match_flag[i]` equal to 0 specifies that the value of each reconstructed sample for decoding the IROI need not be the same as the value of the sample at the same location that would be reconstructed when all the coded slices in `scalLayerRepB` were decoded.

NOTE 5 – When `disable_deblocking_filter_idc` is equal to 2 in all slices in the target access unit set, `exact_sample_value_match_flag` should be equal to 1.

`layer_conversion_flag[i]` equal to 1 specifies that the representation of the current scalable layer can be converted into an alternative set of access units that conform to one or more of the profiles specified in Annex A and gives exactly the same decoding result as the current scalable layer representation and that this conversion can be done without full reconstruction and re-encoding. `layer_conversion_flag[i]` equal to 0 specifies that such a conversion of the current scalable layer representation may not be possible.

`layer_output_flag[i]` equal to 1 specifies that the decoding result for the current scalable layer representation is intended for output. `layer_output_flag[i]` equal to 0 specifies that the decoding result for the current scalable layer representation is not intended for output.

NOTE 6 – The decoding result for a scalable layer representation with `layer_output_flag[i]` equal to 0 may be inappropriate for output due to the low visual quality.

`layer_profile_level_idc[i]` specifies the profile and level compliancy of the representation of the current scalable layer. `layer_profile_level_idc[i]` is the exact copy of the three bytes comprised of `profile_idc`, `constraint_set0_flag`, `constraint_set1_flag`, `constraint_set2_flag`, `constraint_set3_flag`, reserved zero 4bits and `level_idc`, if these syntax elements were used to specify the profile and level compliancy of the representation of the current scalable layer as specified in Annex A and Annex G.

`avg_bitrate[i]` specifies the average bit rate of the representation of the current scalable layer. The average bit rate for the representation of the current scalable layer in bits per second is given by `BitRateBPS(avg_bitrate[i])` with the function `BitRateBPS()` being specified by

$$\text{BitRateBPS}(x) = (x \& 0x3fff) * 10^{(2+(x \gg 14))} \quad (\text{G-372})$$

All NAL units of the representation of the current scalable layer are taken into account in the calculation. The average bit rate is derived according to the access unit removal time specified in Annex C of this Recommendation | International Standard. In the following, `bTotal` is the number of bits in all NAL units of the representation of the current scalable layer succeeding a scalability information SEI message (including the bits of the NAL units of the representation of the current scalable layer in the access unit) and preceding the next access unit (in decoding order) including a scalability information SEI message (if present) or the end of the stream (otherwise). `t1` is the removal time (in seconds) of the current access unit, and `t2` is the removal time (in seconds) of the latest access unit (in decoding order) before the next scalability information SEI message (if present) or the end of the stream (otherwise).

The following conditions shall be fulfilled with `x` specifying the value of `avg_bitrate[i]`.

- If `t1` is not equal to `t2`, the following condition shall be true.

$$(x \& 0x3fff) == \text{Round}(bTotal \div ((t_2 - t_1) * 10^{(2+(x \gg 14))})) \quad (\text{G-373})$$

- Otherwise (`t1` is equal to `t2`), the following condition shall be true.

$$(x \& 0x3fff) == 0 \quad (\text{G-374})$$

`max_bitrate_layer[i]` specifies the maximum bit rate of the current scalable layer. The maximum bit rate of the current scalable layer in bits per second is given by `BitRateBPS(max_bitrate_layer[i])` with the function `BitRateBPS()` being specified in Equation G-372. The maximum bit rate of the current scalable layer is calculated based on a time window specified by `max_bitrate_calc_window[i]`.

**max\_bitrate\_layer\_representation[ i ]** specifies the maximum bit rate of the representation of the current scalable layer. The maximum bit rate of the representation of the current scalable layer in bits per second is given by  $\text{BitRateBPS}(\text{max\_bitrate\_layer\_representation}[ i ]) / 1000$  with the function  $\text{BitRateBPS}()$  being specified in Equation G-372. The maximum bit rate of the representation of the current scalable layer is calculated based on a time window specified by **max\_bitrate\_calc\_window[ i ]**.

**max\_bitrate\_calc\_window[ i ]** specifies the length of the time window in units of 1/100 second, based on which **max\_bitrate\_layer[ i ]** and **max\_bitrate\_layer\_representation[ i ]** are calculated.

**constant\_frm\_rate\_idc[ i ]** specifies whether the output frame rate of the representation of the current scalable layer is constant. If the value of **avg\_frm\_rate[ i ]** as specified in below is constant over all temporal segments of the scalable layer representation, the frame rate is constant; otherwise the frame rate is not constant. **constant\_frm\_rate\_idc[ i ]** equal to 0 specifies that the frame rate is not constant. **constant\_frm\_rate\_idc[ i ]** equal to 1 specifies that the frame rate is constant, and **constant\_frm\_rate\_idc[ i ]** equal to 2 specifies that the frame rate may be or may be not constant. The value of **constantFrameRate** shall be in the range of 0 to 2, inclusive.

**avg\_frm\_rate[ i ]** specifies the average frame rate, in units of frames per 256 seconds, of the representation of the current scalable layer. The semantics of **avg\_frm\_rate[ i ]** is identical to the semantics of **average\_frame\_rate** in sub-sequence layer characteristics SEI message when **accurate\_statistics\_flag** is equal to 1, except that herein the set of NAL units in the range of sub-sequence layers is replaced by the set of NAL units of the representation of the current scalable layer.

**frm\_width\_in\_mbs\_minus1[ i ]** and **frm\_height\_in\_mbs\_minus1[ i ]** specify the width and height, respectively, of the representation of the current scalable layer. When **frame\_mbs\_only\_flag** is equal to 0,  $(\text{frm\_height\_in\_mbs\_minus1}[ i ] + 1) \% 2$  shall be equal to 0.

When **sub\_pic\_layer\_flag[ i ]** is equal to 0, the width of a decoded picture of the representation of the current scalable layer is equal to **frm\_width\_in\_mbs\_minus1[ i ]** + 1 macroblocks, the height of a decoded picture of the representation of the current scalable layer is equal to  $(\text{frm\_height\_in\_mbs\_minus1}[ i ] + 1) / (1 + \text{field\_pic\_flag})$  macroblocks.

When **sub\_pic\_layer\_flag[ i ]** is equal to 1, the width of a decoded sub-picture of the representation of the current scalable layer is equal to **frm\_width\_in\_mbs\_minus1[ i ]** + 1 macroblocks, the height of a decoded sub-picture of the representation of the current scalable layer is equal to  $(\text{frm\_height\_in\_mbs\_minus1}[ i ] + 1) / (1 + \text{field\_pic\_flag})$  macroblocks.

The variable **FrmWidthInMbs[ i ]** is derived as equal to **frm\_width\_in\_mbs\_minus1[ i ]** + 1. The variable **FrmHeightInMbs[ i ]** is derived as equal to **frm\_height\_in\_mbs\_minus1[ i ]** + 1. The variable **FrmSizeInMbs[ i ]** specifies the frame size in unit of macroblocks and is derived as equal to **FrmWidthInMbs[ i ]** \* **FrmHeightInMbs[ i ]**.

**base\_region\_layer\_id[ i ]** specifies the layer identifier of scalable layer sLA. The area of each decoded picture of the current scalable layer representation represents a subset of the area of the corresponding decoded picture of the scalable layer representation of the scalable layer sLA. The current scalable layer represents a region within scalable layer sLA, and the scalable layer sLA is referred to as the base region.

**dynamic\_rect\_flag[ i ]** equal to 1 specifies that the region represented by the current scalable layer representation is a dynamically changing rectangular subset of the base region. **dynamic\_rect\_flag[ i ]** equal to 0 specifies that the region represented by the current scalable layer representation is a fixed rectangular subset of the base region and is specified by **horizontal\_offset[ i ]**, **vertical\_offset[ i ]**, **region\_width[ i ]** and **region\_height[ i ]**.

**horizontal\_offset[ i ]**, **vertical\_offset[ i ]**, **region\_width[ i ]** and **region\_height[ i ]** specify the position and size of the region represented by the current scalable layer. When **frame\_mbs\_only\_flag** is equal to 0,  $(\text{vertical\_offset}[ i ] + 1) \% 2$  and  $(\text{region\_height}[ i ] + 1) \% 2$  shall both be equal to 0.

When **sub\_pic\_layer\_flag[ i ]** is equal to 0, the horizontal and vertical offsets of the top-left pixel of the rectangular region represented by a decoded picture of the current scalable layer representation, relative to the top-left pixel of the base region, in units of luma samples of a decoded frame in the base region, are equal to **horizontal\_offset[ i ]** and **vertical\_offset[ i ]** / (1 + **field\_pic\_flag**), respectively, and the width and height of the rectangular region represented by a decoded picture of the current scalable layer representation, in units of luma samples of a decoded frame in the base region, are equal to **region\_width[ i ]** and **region\_height[ i ]** / (1 + **field\_pic\_flag**), respectively.

When **sub\_pic\_layer\_flag[ i ]** is equal to 1, the horizontal and vertical offsets of the top-left pixel of the rectangular region represented by a decoded sub-picture of the current scalable layer representation, relative to the top-left pixel of the base region, in units of luma samples of a decoded frame in the base region, are equal to **horizontal\_offset[ i ]** and **vertical\_offset[ i ]** / (1 + **field\_pic\_flag**), respectively, and the width and height of the rectangular region represented by a decoded picture of the current scalable layer representation, in units of luma samples of a decoded frame in the base region, are equal to **region\_width[ i ]** and **region\_height[ i ]** / (1 + **field\_pic\_flag**), respectively.

**roi\_id[ i ]** indicates the region-of-interest identifier of the region represented by the current scalable layer.

Let layerIdA and layerIdB with layerIdA less than layerIdB be the layer identifiers of the scalable layers A and B, respectively, both of which having sub\_pic\_layer\_flag[i] equal to 1, and roiIdA and roiIdB be the region-of-interest identifiers of the scalable layers A and B respectively. roiIdA shall not be greater than roiIdB.

**iroi\_grid\_flag[i]** specifies how the IROI division information is coded for the current scalable layer. iroi\_grid\_flag[i] equal to 1 specifies that all IROIs for the current scalable layer have the same size. iroi\_grid\_slice\_flag[i] equal to 0 specifies that the IROIs for the current scalable layer may have different sizes.

**grid\_width\_in\_mbs\_minus1[i]** and **grid\_height\_in\_mbs\_minus1[i]** specify the size of the IROI grids in the current scalable layer. When frame\_mbs\_only\_flag is equal to 0, (grid\_height\_in\_mbs\_minus1 + 1) % 2 shall be equal to 0.

The width and height of a decoded sub-picture of the IROI grids in the current scalable layer are equal to grid\_width\_in\_mbs\_minus1[i] + 1 and (grid\_height\_in\_mbs\_minus1[i] + 1) / (1 + field\_pic\_flag), respectively, in unit of macroblocks, with the following exceptions:

- When (FrmWidthInMbs[i] % (grid\_width\_in\_mbs\_minus1[i] + 1)) is not equal to 0, the decoded sub-picture of an IROI in the very right grid column of the current scalable layer have a width equal to (FrmWidthInMbs[i] % (grid\_width\_in\_mbs\_minus1[i] + 1)) macroblocks.
- When (FrmHeightInMbs[i] % (grid\_height\_in\_mbs\_minus1[i] + 1)) is not equal to 0, the decoded sub-picture of an IROI in the very bottom grid row of the current scalable layer have a height equal to (FrmHeightInMbs[i] % (grid\_height\_in\_mbs\_minus1[i] + 1)) / (1 + field\_pic\_flag) macroblocks.

**num\_rois\_minus1[i]** plus 1 specifies the number of IROIs in the current scalable layer.

**first\_mb\_in\_roi[i][j]** specifies the macroblock address of the first macroblock of the j-th IROI in the current scalable layer. The value of first\_mb\_in\_roi[i][j] shall be equal to the syntax element first\_mb\_in\_slice of the coded slices that belong to the current scalable layer and that cover the top-left macroblock of the j-th IROI. For each combination of i, j, and k, with k not equal to j, first\_mb\_in\_roi[i][j] shall be different then first\_mb\_in\_roi[i][k].

**roi\_width\_in\_mbs\_minus1[i][j]** and **roi\_height\_in\_mbs\_minus1[i][j]** specify the size of the IROIs in the current scalable layer. When frame\_mbs\_only\_flag is equal to 0, (roi\_height\_in\_mbs\_minus1 + 1) % 2 shall be equal to 0.

The width and height of a decoded sub-picture of the j-th IROI in the current scalable layer are equal to roi\_width\_in\_mbs\_minus1[i][j] + 1 and (roi\_height\_in\_mbs\_minus1[i][j] + 1) / (1 + field\_pic\_flag), respectively, in unit of macroblocks.

**num\_directly\_dependent\_layers[i]** specifies the number of scalable layers that the current scalable layer is directly dependent on. The value of num\_directly\_dependent\_layers shall be in the range of 0 to 255, inclusive.

**directly\_dependent\_layer\_id\_delta\_minus1[i][j]** plus 1 specifies the difference between layer\_id[i] and the layer identifier of the j-th scalable layer that the current scalable layer is directly dependent on. The layer identifier of the j-th directly dependent scalable layer is equal to (layer\_id[i] – directly\_dependent\_layer\_id\_delta\_minus1[i][j] – 1).

**layer\_dependency\_info\_src\_layer\_id\_delta[i]** greater than 0 specifies that the current scalable layer has the same layer dependency information as the layer with layer identifier equal to (layer\_id[i] – layer\_dependency\_info\_src\_layer\_id\_delta[i]). layer\_dependency\_info\_src\_layer\_id\_delta[i] equal to 0 specifies that the layer dependency information of the current scalable layer is not present in the SEI message.

**num\_seq\_parameter\_set\_minus1[i]** plus 1 specifies the number of different sequence parameter sets that are referred to by the VCL NAL units of the current scalable layer representation.

**seq\_parameter\_set\_id\_delta[i][j]** specifies the smallest value of the seq\_parameter\_set\_id of any sequence parameter set required for decoding the representation of the current scalable layer, if j is equal to 0. Otherwise (j is greater than 0), seq\_parameter\_set\_id\_delta[i][j] specifies the difference between the value of the seq\_parameter\_set\_id of the j-th required sequence parameter set and the value of the seq\_parameter\_set\_id of the (j – 1)-th required sequence parameter set for decoding the representation of the current scalable layer. The sequence parameter sets are logically ordered in ascending order of the value of seq\_parameter\_set\_id.

**num\_subset\_seq\_parameter\_set\_minus1[i]** plus 1 specifies the number of different subset sequence parameter sets that are referred to by the VCL NAL units of the current scalable layer representation.

**subset\_seq\_parameter\_set\_id\_delta[i][j]** specifies the smallest value of the seq\_parameter\_set\_id of any subset sequence parameter set required for decoding the representation of the current scalable layer, if j is equal to 0. Otherwise (j is greater than 0), subset\_seq\_parameter\_set\_id\_delta[i][j] specifies the difference between the value of the seq\_parameter\_set\_id of the j-th required subset sequence parameter set and the value of the seq\_parameter\_set\_id of the (j – 1)-th required subset sequence parameter set for decoding the representation of the current scalable layer. The subset sequence parameter sets are logically ordered in ascending order of the value of seq\_parameter\_set\_id.

**num\_pic\_parameter\_set\_minus1[ i ]** plus 1 specifies the number of different picture parameter sets that are referred to by the VCL NAL units of the current scalable layer representation.

**pic\_parameter\_set\_id\_delta[ i ][ j ]** specifies the smallest value of the **pic\_parameter\_set\_id** of any picture parameter set required for decoding the representation of the current scalable layer, if **j** is equal to 0. Otherwise (**j** is greater than 0), **pic\_parameter\_set\_id\_delta[ i ][ j ]** specifies the difference between the value of the **pic\_parameter\_set\_id** of the **j**-th required picture parameter set and the value of the **pic\_parameter\_set\_id** of the (**j** – 1)-th required picture parameter set for decoding the representation of the current scalable layer. The picture parameter sets are logically ordered in ascending order of the value of **pic\_parameter\_set\_id**.

**parameter\_sets\_info\_src\_layer\_id\_delta[ i ]** greater than 0 specifies that the values of **seq\_parameter\_set\_id** of the sequence parameter sets and subset sequence parameter sets and the values of **pic\_parameter\_set\_id** of the picture parameter sets that are referred to by the VCL NAL units of the current scalable layer representation are the same as those for the scalable layer representation with layer identifier equal to (**layer\_id[ i ]** – **parameter\_sets\_info\_src\_layer\_id\_delta[ i ]**).

**parameter\_sets\_info\_src\_layer\_id\_delta[ i ]** equal to 0 specifies that the values of **seq\_parameter\_set\_id** of the sequence parameter sets and subset sequence parameter sets and the values of **pic\_parameter\_set\_id** of the picture parameter sets that are referred to by the VCL NAL units of the current scalable layer are not present in the SEI message.

**motion\_vectors\_over\_pic\_boundaries\_flag[ i ]** specifies the value of **motion\_vectors\_over\_pic\_boundaries\_flag**, as specified in subclause E.2.1, for the current scalable layer. When the **motion\_vectors\_over\_pic\_boundaries\_flag[ i ]** syntax element is not present, **motion\_vectors\_over\_pic\_boundaries\_flag** value for the current scalable layer shall be inferred to be equal to 1.

**max\_bytes\_per\_pic\_denom[ i ]** specifies the **max\_bytes\_per\_pic\_denom** value, as specified in subclause E.2.1, for the current scalable layer. When the **max\_bytes\_per\_pic\_denom[ i ]** syntax element is not present, the value of **max\_bytes\_per\_pic\_denom** for the current scalable layer shall be inferred to be equal to 2.

**max\_bits\_per\_mb\_denom[ i ]** specifies the **max\_bits\_per\_mb\_denom** value, as specified in subclause E.2.1, for the current scalable layer. When the **max\_bits\_per\_mb\_denom[ i ]** is not present, the value of **max\_bits\_per\_mb\_denom** for the current scalable layer shall be inferred to be equal to 1.

**log2\_max\_mv\_length\_horizontal[ i ]** and **log2\_max\_mv\_length\_vertical[ i ]** specify the values of **log2\_max\_mv\_length\_horizontal** and **log2\_max\_mv\_length\_vertical**, as specified in subclause E.2.1, for the current scalable layer. When **log2\_max\_mv\_length\_horizontal[ i ]** is not present, the values of **log2\_max\_mv\_length\_horizontal** and **log2\_max\_mv\_length\_vertical** for the current scalable layer shall be inferred to be equal to 16.

NOTE 7 – The maximum absolute value of a decoded vertical or horizontal motion vector component is also constrained by profile and level limits as specified in Annex A.

**num\_reorder\_frames[ i ]** specifies the value of **num\_reorder\_frames**, as specified in subclause E.2.1, for the current scalable layer. The value of **num\_reorder\_frames[ i ]** shall be in the range of 0 to **max\_dec\_frame\_buffering**, inclusive. When the **num\_reorder\_frames[ i ]** syntax element is not present, the value of **num\_reorder\_frames** for the current scalable layer shall be inferred to be equal to **max\_dec\_frame\_buffering**.

**max\_dec\_frame\_buffering[ i ]** specifies the value of **max\_dec\_frame\_buffering**, as specified in subclause E.2.1, for the current scalable layer. The value of **max\_dec\_frame\_buffering[ i ]** shall be in the range of **num\_ref\_frames[ i ]** to **MaxDpbSize** (as specified in subclause A.3.1 or A.3.2), inclusive. When the **max\_dec\_frame\_buffering[ i ]** syntax element is not present, the value of **max\_dec\_frame\_buffering** for the current scalable layer shall be inferred to be equal to **MaxDpbSize**.

**conversion\_type\_idc[ i ]** equal to 0 specifies that **tcoeff\_level\_prediction\_flag** is equal to 1 for all the coded slices of the current scalable layer representation excluding those having **no\_inter\_layer\_pred\_flag** equal to 1.

**conversion\_type\_idc[ i ]** equal to 1 specifies that the **slice\_header\_restriction\_flag** in the sequence parameter sets referred to by the current scalable layer is equal to 1, and **slice\_skip\_flag** in all VCL NAL units with **no\_inter\_layer\_pred\_flag** equal to 0 in the current scalable layer representation is equal to 1.

**conversion\_type\_idc[ i ]** equal to 2 specifies that **slice\_header\_restriction\_flag** in the sequence parameter sets referred to by the current scalable layer is equal to 1, and **no\_inter\_layer\_pred\_flag** in all VCL NAL units in the current scalable layer is equal to 1.

The value of **conversion\_type\_idc[ i ]** shall be in the range of 0 to 2, inclusive.

**conversion\_type\_idc[ i ]** equal to 0 specifies that the information specified by the syntax elements **rewriting\_profile\_level\_idc[ i ][ j ]**, **rewriting\_avg\_bitrate[ i ][ j ]**, and **rewriting\_max\_bitrate[ i ][ j ]**, when present, is correct, though the method for converting the current scalable layer representation into an alternative set of access units that conforms to one or more profiles specified in Annex A and gives exactly the same decoding result as the current scalable layer representation is unspecified.

`conversion_type_idc[i]` equal to 1 specifies that the alternative set of access units obtained by applying the following operations in sequential order to the current scalable layer representation complies to one or more of the profiles specified in Annex A.

1. Remove all NAL units with `nal_unit_type` equal to 20 and `skip_slice_flag` equal to 1
2. Remove all NAL units with `nal_unit_type` equal to 14
3. Remove the NAL unit header SVC extension from NAL units with `nal_unit_type` equal to 20
4. For NAL units with `nal_unit_type` equal to 20 and `idr_flag` equal to 1, set `nal_unit_type` equal to 5
5. For NAL units with `nal_unit_type` equal to 20 and `idr_flag` equal to 0, set `nal_unit_type` equal to 1
6. Remove all SEI NAL units
7. For all picture parameter set NAL units referred to by NAL units with `nal_unit_type` equal to 1 or 5, change the value of `seq_parameter_set_id` to be equal to the `seq_parameter_set_id` in a subset sequence parameter set NAL unit with `profile_idc` equal to 83 or 86
8. Remove all NAL units with `nal_unit_type` equal to 7
9. For all NAL units with `nal_unit_type` equal to 15, set `nal_unit_type` equal to 7, remove all the syntax elements after the syntax structure `seq_parameter_set_data()` and before the `rsbsp_trailing_bits()` syntax structure, replace the three bytes starting from `profile_idc` as specified by `rewriting_profile_level_idc[i][entropy_coding_mode_flag]`, when present, and change RBSP trailing bits appropriately.

`conversion_type_idc[i]` equal to 2 indicates that the alternative bitstream obtained by applying the following operations in sequential order to the current scalable layer representation conforms to one or more of the profiles specified in Annex A.

1. Remove all NAL units with `nal_unit_type` equal to 14
2. Remove the NAL unit header SVC extension from NAL units with `nal_unit_type` equal to 20
3. For NAL units with `nal_unit_type` equal to 20 and `idr_flag` equal to 1, set `nal_unit_type` equal to 5
4. For NAL units with `nal_unit_type` equal to 20 and `idr_flag` equal to 0, set `nal_unit_type` equal to 1
5. Remove all SEI NAL units
6. For all NAL units with `nal_unit_type` equal to 15, set `nal_unit_type` equal to 7, remove all the syntax elements after the syntax structure `seq_parameter_set_data()` and before the `rsbsp_trailing_bits()` syntax structure, replace the three bytes starting from `profile_idc` as specified by `rewriting_profile_level_idc[i][entropy_coding_mode_flag]`, when present, and change RBSP trailing bits appropriately.

`rewriting_info_flag[i][j]` equal to 1 specifies that information is present about the alternative set of access units obtained by converting the current scalable layer representation. It is asserted that the signalled information is correct, though the method for constructing the alternative set of access units is unspecified. The variable `j` specifies the value of `entropy_coding_mode_flag` for the alternative set of access units, with values for `j` equal to 0 or 1 indicating use of the CAVLC or CABAC entropy coding methods, respectively.

`rewriting_profile_level_idc[i][j]` specifies the profile compliancy of the alternative set of access units for the current scalable layer representation after conversion. `rewriting_profile_level_idc[i]` is the exact copy of the three bytes consist of `profile_idc`, `constraint_set0_flag`, `constraint_set1_flag`, `constraint_set2_flag`, `constraint_set3_flag`, `reserved_zero_4bits`, and `level_idc`, if these syntax elements were used to specify the profile and level compliancy of the alternative set of access units obtained by converting the scalable layer representation as specified in Annex A.

`rewriting_avg_bitrate[i][j]` specifies the average bit rate of the alternative set of access units obtained by converting the representation of the current scalable layer. The average bit rate of the alternative set of access units in bits per second is given by `BitRateBPS(rewriting_avg_bitrate[i][j])` with the function `BitRateBPS()` being specified in Equation G-372.

`rewriting_max_bitrate[i][j]` specifies the maximum bit rate of the alternative set of access units obtained by converting the representation of the current scalable layer, in any one-second time window of access unit removal time as specified in Annex C. The maximum bit rate of the alternative set of access units in bits per second is given by `BitRateBPS(rewriting_max_bitrate[i][j])` with the function `BitRateBPS()` being specified in Equation G-372.

**pr\_num\_dId\_minus1** plus 1 specifies the number of dependency layers for which the priority layer characteristic information is present in the SEI message. A dependency layer consists of all the VCL NAL units of the target access unit set with a particular value of **dependency\_id** and the associated non-VCL NAL units.

**pr\_dependency\_id[i]** specifies the value of **dependency\_id** of the dependency layer for which the priority layer characteristic information is signalled by the following syntax elements.

**pr\_num\_minus1[i]** plus 1 specifies the number of priority layers corresponding to the dependency layer with **dependency\_id** equal to **pr\_dependency\_id[i]**. A priority layer represents the set of coded slice NAL units of a dependency layer that have a value of **priority\_id** that is less than or equal to a specified value of **priority\_id** and the associated non-VCL NAL units. The bitstream subset that is required for decoding the priority layer is referred to as priority layer representation. The value of **ql\_num\_minus1** shall be in the range of 0 to the greatest value of the syntax element **alt\_priority\_id[i]** in all priority layer information SEI messages in the target access unit set, inclusive.

**pr\_id[i][j]** specifies the identifier of the priority layer for the dependency layer with **dependency\_id** equal to **pr\_dependency\_id[i]**. The priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** consists of all VCL NAL units with a value of **alt\_priority\_id[i]**, specified by the priority layer information SEI message, less than or equal to **pr\_id[i][j]**, and the associated non-VCL NAL units.

**pr\_profile\_level\_idc[i][j]** specifies the profile and level compliancy of the priority layer representation for the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]**. **pr\_profile\_level\_idc[i]** is the exact copy of the three bytes consisting of **profile\_idc**, **constraint\_set0\_flag**, **constraint\_set1\_flag**, **constraint\_set2\_flag**, **constraint\_set3\_flag**, **reserved\_zero\_4bits**, and **level\_idc**, if these syntax elements were used to specify the profile and level compliancy of the bitstream subset of the priority layer representation for the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** as specified in Annex A or Annex G.

**pr\_avg\_bitrate[i][j]** specifies the average bit rate of the bitstream subset of the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]**. The average bit rate of the bitstream subset of the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** in bits per second is given by  $\text{BitRateBPS}(\text{pr\_avg\_bitrate}[i][j])$  with the function  $\text{BitRateBPS}()$  being specified in Equation G-372. The average bit rate is derived according to the access unit removal time specified in Annex C of the Recommendation | International Standard. In the following,  $b_{\text{Total}}$  is the number of bits in all NAL units of the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** succeeding a scalability information SEI message (including the bits of the NAL units of the quality layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** in the access unit) and preceding the next access unit (in decoding order) including a scalability information SEI message (if present) or the end of the stream (otherwise).  $t_1$  is the removal time (in seconds) of the current access unit, and  $t_2$  is the removal time (in seconds) of the latest access unit (in decoding order) before the next scalability information SEI message (if present) or the end of the stream (otherwise).

The following conditions shall be fulfilled with  $x$  specifying the value of **pr\_avg\_bitrate[i]**.

- If  $t_1$  is not equal to  $t_2$ , the condition specified in Equation G-373 shall be fulfilled.
- Otherwise ( $t_1$  is equal to  $t_2$ ), the condition specified in Equation G-374 shall be fulfilled.

**pr\_max\_bitrate[i][j]** specifies the maximum bit rate of the bitstream subset of the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** in any one-second time window of access unit removal time as specified in Annex C. The maximum bit rate of the priority layer identified by **pr\_dependency\_id[i]** and **pr\_id[i][j]** in bits per second is given by  $\text{BitRateBPS}(\text{pr\_max\_bitrate}[i][j])$  with the function  $\text{BitRateBPS}()$  being specified in Equation G-372.

**priority\_id\_setting\_uri[PriorityIdSettingUriIdx]** is the **PriorityIdSettingUriIdx**-th byte of a null-terminated string encoded in UTF-8 characters, specifying the universal resource identifier (URI) of the description of the method used to calculate the **priority\_id** values in the NAL unit headers for the target access unit set.

### G.13.2.2 Layers not present SEI message semantics

When present, this SEI message applies to the target access unit set that consists of the current access unit and all the subsequent access units, in decoding order, until the next layers not present SEI message, the next scalability information SEI message, or the end of the bitstream, whichever is the earliest in decoding order. The layers not present SEI message identifies the scalable layers indicated by the preceding scalability information SEI message that are not present in the target access unit set. A layers not present SEI message always refers to the most recent scalability SEI message in decoding order.

NOTE – Layers not present SEI messages do not have a cumulative effect.

**num\_layers** specifies the number of scalable layers that are not present in the target access unit set. **num\_layers** equal to 0 indicates that all scalable layers specified in the most recent scalability information SEI message in decoding order are present in the target access unit set. The value of **num\_layers** shall be in the range of 0 to 2047, inclusive.

**layer\_id[ i ]** specifies the identifier of a scalable layer that is not present in the target access unit set. The **layer\_id[ i ]** values refer to, and shall be among to, the **layer\_id[ i ]** values included in the most recent scalability information SEI message in decoding order. The value of **layer\_id[ i ]** shall be in the range of 0 to 2047, inclusive.

#### G.13.2.3 Layer dependency change SEI message semantics

Layer dependency change SEI message specifies a change of scalable layer dependency relationship. When present, the layer dependency change SEI message applies to the target access unit set that consists of the current access unit and all the subsequent access units, in decoding order, until the next layer dependency change SEI message, the next layers not present SEI message, the next scalability information SEI message, or the end of the bitstream, whichever is the earliest in decoding order. The presence of the layer dependency change SEI message specifies that the layer dependency relationships for the scalable layers identified by **layer\_id[ i ]** for all values of *i* changes for the target access unit set relative to the layer dependency relationship specifies by the most recent scalability information SEI message in decoding order. For a scalable layer with the layer identifier not equal to **layer\_id[ i ]** for any value of *i*, the layer dependency relationship remains the same as specified in the most recent scalability information SEI message in decoding order.

If, according to the layer dependency information indicated in the most recent scalability information SEI message in decoding order, a scalable layer A does not directly or indirectly depend on another scalable layer B, the layer dependency change SEI message shall not specify layer dependency relationship between layers A and B. A layer dependency change SEI message always refers to the most recent scalability information SEI message in decoding order.

NOTE – Layer dependency change SEI messages do not have a cumulative effect.

**num\_layers\_minus1** plus 1 specifies the number of scalable layers for which the layer dependency information changes for the target access unit set relative to the layer dependency relationship indicated by the most recent scalability information SEI message in decoding order. The value of **num\_layers\_minus1** is in the range of 0 to 2047, inclusive.

**layer\_id[ i ]** specifies the identifier of a scalable layer for which the layer dependency information changes. The **layer\_id[ i ]** values refer to, and shall be among to, the **layer\_id[ i ]** values included in the most recent scalability information SEI message in decoding order. The value of **layer\_id[ i ]** shall be in the range of 0 to 2047, inclusive.

**layer\_dependency\_info\_present\_flag[ i ]** equal to 1 specifies the presence of layer dependency information for the scalable layer with layer identifier equal to **layer\_id[ i ]** in the layer dependency change SEI message. **layer\_dependency\_info\_present\_flag[ i ]** equal to 0 specifies that the layer dependency information for the scalable layer with layer identifier equal to **layer\_id[ i ]** is not present in the layer dependency change SEI message, and is the same as the layer dependency information for the scalable layer indicated by **layer\_dependency\_info\_src\_layer\_id\_delta\_minus1[ i ]**.

**num\_directly\_dependent\_layers[ i ]** specifies the number of scalable layers that the scalable layer with layer identifier equal to **layer\_id[ i ]** is directly dependent on. The value of **num\_directly\_dependent\_layers** shall be in the range of 0 to 2047, inclusive.

**directly\_dependent\_layer\_id\_delta\_minus1[ i ][ j ]** specifies the difference between **layer\_id[ i ]** and the layer identifier of the *j*-th scalable layer that the scalable layer with layer identifier equal to **layer\_id[ i ]** is directly dependent on. The layer identifier of the directly dependent scalable layer is equal to (**layer\_id[ i ]** – **directly\_dependent\_layer\_id\_delta\_minus1[ i ][ j ]**).

**layer\_dependency\_info\_src\_layer\_id\_delta\_minus1[ i ]** specifies the layer identifier of the scalable layer that has the same layer dependency information as the scalable layer with layer identifier equal to **layer\_id[ i ]**. The layer identifier of the scalable layer that has the same layer dependency information as the scalable layer with layer identifier equal to **layer\_id[ i ]** is equal to (**layer\_id[ i ]** – **layer\_dependency\_info\_src\_layer\_id\_delta\_minus1[ i ]**).

#### G.13.2.4 Sub-picture scalable layer SEI message semantics

When present, the sub-picture scalable layer SEI message shall immediately succeed, in decoding order, a motion-constrained slice group set SEI message and both of these SEI messages shall be present in the same SEI NAL unit. The slice group set identified by the motion-constrained slice group set SEI message is referred to as the associated slice group set of the sub-picture layer information SEI message.

**layer\_id** specifies the layer identifier of the scalable layer to which the coded slice NAL units in the associated slice group set belong to. The value of **layer\_id** shall be in the range of 0 to 2047, inclusive.

#### G.13.2.5 Non-required layer representation SEI message semantics

**num\_info\_entries\_minus1** plus 1 specifies the number of the information entries following this syntax element. The value of **num\_info\_entries\_minus1** shall be in the range of 0 to 7, inclusive.

**entry\_dependency\_id[ i ]** specifies the dependency\_id value of the layer representation, which is referred to as the target layer representation in the following, for which the information of non-required layer representations is specified by the following syntax elements. The instances of entry\_dependency\_id[ i ] shall appear in the increasing order of the value of entry\_dependency\_id[ i ]. The quality\_id value of the target layer representation is equal to 0. A non-required layer representation for the target layer representation is not required in decoding of any other layer representation in the coded video sequence having the same dependency\_id value and quality\_id value as the target layer representation.

NOTE 1 – A layer representation having quality\_id greater than 0 has the same set of non-required layer representations as the layer representation having the same dependency\_id value and quality\_id equal to 0.

**num\_non\_required\_layer\_rep\_minus1[ i ]** plus 1 specifies the number of non-required layer representations for the target layer representation. The value of num\_non\_required\_layer\_rep\_minus1[ i ] shall be in the range of 0 to 127, inclusive.

**non\_required\_layer\_rep\_dependency\_id[ i ][ j ]** specifies the dependency\_id value of the j-th non-required layer representation for the target layer representation.

**non\_required\_layer\_rep\_quality\_id[ i ][ j ]** specifies the quality\_id value of the j-th non-required layer representation for the target layer representation. In addition, those layer representations that have dependency\_id equal to non\_required\_layer\_rep\_dependency\_id[ i ][ j ] and quality\_id greater than non\_required\_layer\_rep\_quality\_id[ i ][ j ] are also non-required layer representations for the target layer representation. Furthermore, additional non-required layer representations are derived as follows.

- When a layer representation having dependency\_id equal to A is not a non-required layer representation for the layer representation having dependency\_id equal to B wherein B is greater than or equal to A, then all the non-required layer representations for the layer representation having dependency\_id equal to A are also non-required layer representations for the layer representation having dependency\_id equal to B.
- When dIdMax is not equal to entry\_dependency\_id[ i ] for any value of i and dIdMax is greater than entry\_dependency\_id[ i ] for at least one value of i, the following applies.
  - Let dIdC be equal to the greatest value of entry\_dependency\_id[ i ] less than dIdMax.
  - The layer representation having dependency\_id equal to dIdMax shall have the same set of non-required layer representations as the layer representation having dependency\_id equal to dIdC and quality\_id equal to 0.

NOTE 2 – When dIdMax is less than entry\_dependency\_id[ i ] for all values of i, there is no non-required layer representation in the associated access unit for the layer representation having dependency\_id equal to dIdMax.

#### G.13.2.6 Priority layer information SEI message semantics

The information conveyed in this SEI message specifies alternative values for priority\_id for the NAL units with the given value of dependency\_id inside the current access unit. The alternative values for priority\_id specify priority layers.

**pr\_dependency\_id** specifies the value of dependency\_id for the NAL units inside the access unit for which the SEI message applies.

**num\_priority\_ids** specifies the number of layer representations within the dependency representation with dependency\_id equal to pr\_dependency\_id for which alternative values of priority\_id are specified by this SEI message.

**alt\_priority\_id[ i ]** specifies the alternative value for priority\_id for the NAL units with dependency\_id equal to pr\_dependency\_id and quality\_id equal to i inside the current access unit.

#### G.13.2.7 Scalable nesting SEI message semantics

A scalable nesting SEI message shall contain one or more SEI messages of payloadType not equal to 30. An SEI message contained in a scalable nesting SEI message is referred to as a nested SEI message. An SEI message not contained in a scalable nesting SEI message is referred to as a non-nested SEI message. The scope to which the nested SEI message applies is indicated by the syntax elements all\_layer\_representations\_in\_au\_flag, num\_layer\_representations\_minus1, sei\_dependency\_id[ i ], sei\_quality\_id[ i ], and sei\_temporal\_id, when present.

A buffering period SEI message and an SEI message of any other type shall not be nested in the same scalable nesting SEI message. When a buffering period SEI message is included in a scalable nesting SEI message, which is referred to as the SEI message under consideration, the following applies.

- Any SEI message preceding the SEI message under consideration in decoding order within the same access unit shall be a buffering period SEI message or a scalable nesting SEI message with a buffering period SEI message as payload.

- Additionally, the following applies.
  - If the value of `all_layer_representations_in_au_flag` in the SEI message under consideration is equal to 1, no scalable nesting SEI message shall precede the SEI message under consideration in decoding order within the same access unit.
  - Otherwise (the value of `all_layer_representations_in_au_flag` in the SEI message under consideration is equal to 0), for any scalable nesting SEI message that precedes the SEI message under consideration in decoding order within the same access unit, the values  $(sei\_dependency\_id[i] \ll 4) + sei\_quality\_id[i]$  for all present  $i$  shall be less than any of the values  $(sei\_dependency\_id[i] \ll 4) + sei\_quality\_id[i]$  for the SEI message under consideration.

A picture timing SEI message and an SEI message of any other type shall not be nested in the same scalable nesting SEI message.

`all_layer_representations_in_au_flag` equal to 1 specifies that the nested SEI message applies to all the layer representations of the access unit. `all_layer_representations_in_au_flag` equal to 0 specifies that the applicable scope of the nested SEI message is specified by the syntax elements `num_layer_representations_minus1`, `sei_dependency_id[i]`, `sei_quality_id[i]`, and `sei_temporal_id`.

`num_layer_representations_minus1` plus 1 specifies the number of layer representation to which the nested SEI message applies. When `num_layer_representations_minus1` is not present, it is inferred to be equal to  $(numSVCLayers - 1)$  with `numSVCLayers` being the number of layer representations that are present in the access unit.

`sei_dependency_id[i]` and `sei_quality_id[i]` specify the `dependency_id` and the `quality_id`, respectively, of the  $i$ -th layer representation to which the nested SEI message applies. When `num_layer_representations_minus1` is not present, the values of `sei_dependency_id[i]` and `sei_quality_id[i]` for  $i$  in the range of 0 to `num_layer_representations_minus1` (with `num_layer_representations_minus1` being the inferred value), inclusive, are inferred as follows.

- Let `setDQId` be the set of the values `DQId` for all layer representations that are present in the access unit.
- For  $i$  proceeding from 0 to `num_layer_representations_minus1`, inclusive, the following applies.
  - `sei_dependency_id[i]` and `sei_quality_id[i]` are inferred to be equal to  $(minDQId \gg 4)$  and  $(minDQId \& 15)$ , respectively, with `minDQId` being the smallest value (smallest value of `DQId`) in `setDQId`.
  - The smallest value (smallest value of `DQId`) of the set `setDQId` is removed from `setDQId` and thus the number of elements in `setDQId` is decreased by 1.

`sei_temporal_id` specifies the `temporal_id` of the bitstream subset to which the nested SEI message applies.

When `sei_temporal_id` is not present, it shall be inferred to be equal to `temporal_id` of the access unit.

When the nested SEI message is a buffering period SEI message or a picture timing SEI message, `sei_temporal_id` indicates the bitstream subset for which the nested buffering period SEI message or picture timing SEI message applies. For a buffering period SEI message or picture timing SEI message that is nested in a scalable nesting SEI message, `sei_dependency_id[i]`, `sei_quality_id[i]` and `sei_temporal_id` specify the greatest `dependency_id`, `quality_id`, and `temporal_id` values, respectively, of the bitstream subsets for which the nested buffering period SEI message or picture timing SEI message applies.

For a nested SEI message for which `payloadType` is in the range of 2 to 3, inclusive, or 6 to 35, inclusive, `sei_temporal_id` shall be equal to `temporal_id` of the access unit.

`sei_nesting_zero_bit` shall be equal to 0.

#### G.13.2.8 Base layer temporal HRD SEI message semantics

When present, this SEI message applies to the target access unit set that consists of the current access unit and all the subsequent access units, in decoding order, until the next base layer temporal HRD SEI message or the end of the bitstream, whichever is the earliest in decoding order.

`num_of_temporal_layers_in_base_layer_minus1` plus 1 specifies the number of bitstream subsets inside the target access unit set for which the following syntax elements apply.

`temporal_id[i]` specifies the `temporal_id` value of the  $i$ -th bitstream subset.

Let the  $i$ -th bitstream subset for the target access unit set that is obtained by invoking the bitstream extraction process as specified in subclause G.8.8.1 with `tIdTarget` equal to `temporal_id[i]`, `dIdTarget` equal to 0, and `qIdTarget` equal to 0 as input.

**timing\_info\_present\_flag[ i ]** specifies the value of **timing\_info\_present\_flag** that applies for the i-th bitstream subset.

**num\_units\_in\_tick[ i ]** specifies the value of **num\_units\_in\_tick** that applies for the i-th bitstream subset. **num\_units\_in\_tick[ i ]** shall be greater than 0.

**time\_scale[ i ]** specifies the value of **time\_scale** that applies for the i-th bitstream subset. **time\_scale[ i ]** shall be greater than 0.

**fixed\_frame\_rate\_flag[ i ]** specifies the value of **fixed\_frame\_rate\_flag** that applies for the i-th bitstream subset.

**nal\_hrd\_parameters\_present\_flag[ i ]** specifies the value of **nal\_hrd\_parameters\_present\_flag** that applies for the i-th bitstream subset.

**vcl\_hrd\_parameters\_present\_flag[ i ]** specifies the value of **vcl\_hrd\_parameters\_present\_flag** that applies for the i-th bitstream subset.

**low\_delay\_hrd\_flag[ i ]** specifies the value of **low\_delay\_hrd\_flag** that applies for the i-th bitstream subset.

**pic\_struct\_present\_flag[ i ]** specifies the value of **pic\_struct\_present\_flag** that applies for the i-th bitstream subset.

#### G.13.2.9 Quality layer integrity check SEI message semantics

**num\_info\_entries\_minus1** plus 1 specifies the number of syntax elements **entry\_dependency\_id[ i ]** and **quality\_layer\_crc[ i ]**. The value of **num\_info\_entries\_minus1** shall be in the range of 0 to 7, inclusive.

**entry\_dependency\_id[ i ]** specifies the **dependency\_id** value of the dependency representation for which **quality\_layer\_crc[ i ]** is indicated. The instances of **entry\_dependency\_id[ i ]** shall appear in increasing order of their values.

**quality\_layer\_crc[ i ]** specifies the cyclic redundancy check for all the VCL NAL units with **quality\_id** greater than 0 in the dependency representation.

**quality\_layer\_crc[ i ]** shall be equal to the value obtained by the following ordered steps.

1. Let the variable **qNalUnits[ ]** be the one-dimensional array of bytes that contains a concatenation, in network byte order, of the bytes of all the VCL NAL units with **quality\_id** greater than 0 in the dependency representation having **dependency\_id** equal to **entry\_dependency\_id[ i ]**, in decoding order.
2. Let the variable **pLen** be the number of bytes in **qNalUnits[ ]**.
3. The value of **quality\_layer\_crc[ i ]** shall be equal to the value of **crcVal** obtained by performing the following pseudo code process.

```
qNalUnits[pLen] = 0
qNalUnits[pLen + 1] = 0
crcVal = 0xFFFF
for( bitIdx = 0; bitIdx < ( pLen + 2 ) * 8; bitIdx++ ) {
    crcMsb = ( crcVal >> 15 ) & 1
    bitVal = ( qNalUnits[bitIdx >> 3] >> ( 7 - ( bitIdx & 7 ) ) ) & 1
    crcVal = ( ( ( crcVal << 1 ) + bitVal ) & 0xFFFF ) ^ ( crcMsb * 0x1021 )
}
```

#### G.13.2.10 Redundant picture property SEI message semantics

**num\_dId\_minus1** plus 1 specifies the number of the dependency representations for which the redundant picture properties are specified by the following syntax elements.

**dependency\_id[ i ]** specifies the **dependency\_id** value of the dependency representation for which the redundant picture properties are specified by the following syntax elements.

**num\_qId\_minus1[ i ]** plus 1 specifies the number of the layer representations inside the dependency representation having **dependency\_id** equal to **dependency\_id[ i ]** for which the redundant picture properties are specified by the following syntax elements.

**quality\_id[ i ][ j ]** specifies the **quality\_id** value of the layer representation having **dependency\_id** equal to **dependency\_id[ i ]** for which the redundant picture properties are specified by the following syntax elements.

**num\_redundant\_pics\_minus1[ i ][ j ]** plus 1 specifies the number of redundant coded pictures in the layer representation having **dependency\_id** equal to **dependency\_id[ i ]** and **quality\_id** equal to **quality\_id[ i ][ j ]** for which the redundant picture properties are specified by the following syntax elements.

**redundant\_pic\_cnt\_minus1**[i][j][k] plus 1 specifies the **redundant\_pic\_cnt** value of the redundant coded picture having **dependency\_id** equal to **dependency\_id**[i] and **quality\_id** equal to **quality\_id**[i][j] for which the redundant picture properties are specified by the following syntax elements.

The redundant coded picture having **dependency\_id** equal to **dependency\_id**[i], **quality\_id** equal to **quality\_id**[i][j], and **redundant\_pic\_cnt** equal to (**redundant\_pic\_cnt\_minus1**[i][j][k] + 1) is referred to as the target redundant coded picture. The layer representation having **dependency\_id** equal to **dependency\_id**[i], **quality\_id** equal to **quality\_id**[i][j], and **redundant\_pic\_cnt** equal to 0 is referred to as the primary layer representation.

**pic\_match\_flag**[i][j][k] equal to 1 specifies that the target redundant coded picture is an exact copy of the primary layer representation, with the only difference in the value of **redundant\_pic\_cnt**.

**mb\_type\_match\_flag**[i][j][k] equal to 1 specifies that the syntax element **mb\_type** for all macroblocks in the target redundant coded picture are identical to the corresponding syntax element in the primary layer representation.

**motion\_match\_flag**[i][j][k] equal to 1 specifies that the **mb\_type** of all inter-coded macroblocks and the associated reference indices and motion vectors for all macroblocks in the redundant coded picture are identical to the corresponding syntax elements in the primary layer representation.

**residual\_match\_flag**[i][j][k] equal to 1 specifies that the reconstructed residual sample values as specified in subclause G.8.5.3 for all inter-coded macroblocks in the redundant coded picture is identical or close to the reconstructed residual sample values for the corresponding macroblocks in the primary layer representation.

**intra\_samples\_match\_flag**[i][j][k] equal to 1 specifies that the reconstructed sample values for all intra coded macroblocks in the redundant coded picture are identical or close to the reconstructed sample values of the corresponding macroblocks in the primary layer representation.

#### G.13.2.11 Temporal level zero dependency representation index SEI message semantics

The temporal level zero dependency representation index SEI message provides a mechanism to detect whether a dependency representation with **temporal\_id** equal to 0 needed for decoding the current access unit is available when NAL unit losses are expected during transport.

Let **setOfDId** be a set of values that is derived as follows.

- If the temporal level zero dependency representation index SEI message is not included in a scalable nesting SEI message, **setOfDId** consists of exactly one value, which is set equal to 0.
- Otherwise (the temporal level zero dependency representation index SEI message is included in a scalable nesting SEI message), **setOfDId** consists of the values **sei\_dependency\_id**[i] for all i in the range of 0 to **num\_layer\_representations\_minus1**, inclusive, that are associated with the temporal level zero dependency representation index SEI message.

**tl0\_dep\_rep\_idx** specifies the temporal level zero index for the associated dependency representations, if **temporal\_id** is equal to 0. Otherwise (**temporal\_id** is greater than 0), **tl0\_dep\_rep\_idx** specifies the temporal level zero index of the associated dependency representations of the most recent access unit with **temporal\_id** equal to 0 in decoding order.

For each value of **dId** of the set **setOfDId**, the following constraints on **tl0\_dep\_rep\_idx** shall be obeyed.

- If the dependency representation with **dependency\_id** equal to **dId** contains a NAL unit with **nal\_unit\_type** equal to 5 or a NAL unit with **nal\_unit\_type** equal to 20 and **idr\_flag** equal to 1, **tl0\_dep\_rep\_idx** shall be equal to 0.
- Otherwise (the dependency representation with **dependency\_id** equal to **dId** does not contain a NAL unit with **nal\_unit\_type** equal to 5 or a NAL unit with **nal\_unit\_type** equal to 20 and **idr\_flag** equal to 1), the following applies.
  - Let **prevTL0DepRep** be the dependency representation with **dependency\_id** equal to **dId** of the most recent access unit with **temporal\_id** equal to 0 in decoding order that includes a dependency representation with **dependency\_id** equal to **dId**.
  - Let **prevTL0DepRepIdx** be equal to **tl0\_dep\_rep\_idx** of the dependency representation **prevTL0DepRep**.
  - Depending on **temporal\_id** of the current access unit, the following applies.
    - If **temporal\_id** of the current access unit is equal to 0, **tl0\_dep\_rep\_idx** shall be equal to (**prevTL0DepRepIdx** + 1) % 256.
    - Otherwise (**temporal\_id** of the current access unit is greater than 0), **tl0\_dep\_rep\_idx** shall be equal to **prevTL0DepRepIdx**.

NOTE – For the `tl0_dep_rep_idx` mechanism to be effectively used, transport mechanisms should ensure that the information is present in every packet that carries data for the particular values of `dependency_id`.

**effective\_idr\_pic\_id** specifies the latest value of `idr_pic_id` in decoding order present in this access unit or any preceding access unit for dependency representations indicated by `sei_dependency_id[i]`.

For each value of `dId` of the set `setOfDId`, the following constraints on **effective\_idr\_pic\_id** shall be obeyed.

- If the dependency representation with `dependency_id` equal to `dId` contains a NAL unit with `nal_unit_type` equal to 5 or a NAL unit with `nal_unit_type` equal to 20 and `idr_flag` equal to 1, **effective\_idr\_pic\_id** shall be equal to `idr_pic_id` of the dependency representation with `dependency_id` equal to `dId`.
- Otherwise (the dependency representation with `dependency_id` equal to `dId` does not contain a NAL unit with `nal_unit_type` equal to 5 or a NAL unit with `nal_unit_type` equal to 20 and `idr_flag` equal to 1), **effective\_idr\_pic\_id** shall be equal to `idr_pic_id` of the previous dependency representation in decoding order with `dependency_id` equal to `dId` that contains a NAL unit with `nal_unit_type` equal to 5 or a NAL unit with `nal_unit_type` equal to 20 and `idr_flag` equal to 1.

#### G.13.2.12 Temporal level switching point SEI message semantics

Temporal layer switching point SEI messages provide information on temporal layer switching points. If a layer representation is associated with a temporal layer switching point SEI message, then it is a temporal layer switching point as specified subsequently and constrained by `delta_frame_num`. Otherwise, the layer representation may or may not be a temporal layer switching point.

In the following, let `tId` be equal to the value of `temporal_id` associated with the access unit in which the temporal layer switching point SEI message is included.

NOTE 1 – Let `dId` be the value of `dependency_id` that a bitstream adaptation process has used to generate a bitstream subset `subBitstreamA` that contains dependency representations with `dependency_id` less than or equal to `dId` and `temporal_id` less than `tId` until the current access unit, exclusive. The bitstream adaptation process can infer from a temporal layer switching point SEI message whether or not the bitstream subset containing `subBitstreamA` and the dependency representations with `dependency_id` less than or equal to `dId` and `temporal_id` less than or equal to `tId` starting from the current access unit, inclusive, is conforming to this Recommendation | International Standard.

The temporal layer switching point SEI message shall not be present in access units for which `temporal_id` is equal to 0, or in access units that do not contain NAL units with `nal_unit_type` equal to 20 and in which no prefix NAL units are present.

The temporal layer switching point SEI message shall not be present in a supplemental enhancement information RBSP unless it is included within a scalable nesting SEI message.

The semantics apply independently to layer representations indicated by each pair of values of `sei_dependency_id[i]` and `sei_quality_id[i]` of the scalable nesting SEI message containing this SEI message.

Let the switch-to layer representation be the layer representation in the access unit containing this SEI message having `dependency_id` equal to `sei_dependency_id[i]` and `quality_id` equal to `sei_quality_id[i]`.

**delta\_frame\_num** indicates the difference of the `frame_num` values between the switch-to layer representation and the required layer representation as specified subsequently. Let `currFrameNum` be the `frame_num` of the switch-to layer representation and `requiredFrameNum` be the `frame_num` of the required layer representation. Let the variable `requiredFrameNum` be set equal to `currFrameNum` – `delta_frame_num`. Let the required layer representation be the layer representation of the previous access unit, in decoding order, having `frame_num` equal to `requiredFrameNum`, `dependency_id` equal to `sei_dependency_id[i]`, and `quality_id` equal to `sei_quality_id[i]`.

NOTE 2 – The decoding process for the switch-to layer representation may use the required layer representation and any subsequent layer representation, in decoding order, having `temporal_id` less than `tId` and `dependency_id` is less than or equal to `sei_dependency_id[i]` in the decoding process.

The switch-to layer representation and any subsequent layer representation in decoding order for which `temporal_id` is equal to `tId` and `dependency_id` is less than or equal to `sei_dependency_id[i]` shall not use any layer representation preceding the required layer representation in decoding order for which `temporal_id` is equal to `tId` – 1 and `dependency_id` is less than or equal to `sei_dependency_id[i]` in the decoding process.

The switch-to layer representation and any subsequent layer representation in decoding order for which `temporal_id` is equal to `tId` and `dependency_id` is less than or equal to `sei_dependency_id[i]` shall not use any layer representation preceding the access unit containing the switch-to layer representation in decoding order for which `temporal_id` is equal to `tId` and `dependency_id` is less than or equal to `sei_dependency_id[i]` in the decoding process.

NOTE 3 – All layer representations that follow the switch-to layer representation in decoding order, inclusive, for which temporal\_id is equal to tId and dependency\_id is less than or equal to sei\_dependency\_id[ i ], can be decoded, when all layer representations have been decoded starting from the required layer representation to the layer representation in the access unit preceding the access unit that includes the switch-to layer representation in decoding order, inclusive, for which temporal\_id is less than tId and dependency\_id is less than or equal to sei\_dependency\_id[ i ].

## G.14 SVC video usability information extension

The specifications in Annex E apply with substituting SVC sequence parameter set for sequence parameter set. Additionally, the following applies.

### G.14.1 SVC VUI parameters extension syntax

| svc_vui_parameters_extension() {                                                      | C | Descriptor |
|---------------------------------------------------------------------------------------|---|------------|
| <b>num_layers_minus1</b>                                                              | 0 | ue(v)      |
| for( i = 0; i <= num_layers_minus1; i++ ) {                                           |   |            |
| <b>dependency_id[ i ]</b>                                                             | 0 | u(3)       |
| <b>quality_id[ i ]</b>                                                                | 0 | u(4)       |
| <b>temporal_id[ i ]</b>                                                               | 0 | u(3)       |
| <b>timing_info_present_flag[ i ]</b>                                                  | 0 | u(1)       |
| if( timing_info_present_flag[ i ] ) {                                                 |   |            |
| <b>num_units_in_tick[ i ]</b>                                                         | 0 | u(32)      |
| <b>time_scale[ i ]</b>                                                                | 0 | u(32)      |
| <b>fixed_frame_rate_flag[ i ]</b>                                                     | 0 | u(1)       |
| }                                                                                     |   |            |
| <b>nal_hrd_parameters_present_flag[ i ]</b>                                           | 0 | u(1)       |
| if( nal_hrd_parameters_present_flag[ i ] )                                            |   |            |
| hrd_parameters()                                                                      |   |            |
| <b>vcl_hrd_parameters_present_flag[ i ]</b>                                           | 0 | u(1)       |
| if( vcl_hrd_parameters_present_flag[ i ] )                                            |   |            |
| hrd_parameters()                                                                      |   |            |
| if( nal_hrd_parameters_present_flag[ i ]   <br>vcl_hrd_parameters_present_flag[ i ] ) |   |            |
| <b>low_delay_hrd_flag[ i ]</b>                                                        | 0 | u(1)       |
| <b>pic_struct_present_flag[ i ]</b>                                                   | 0 | u(1)       |
| }                                                                                     |   |            |
| }                                                                                     |   |            |

### G.14.2 SVC VUI parameters extension semantics

**num\_layer\_minus1** plus 1 specifies the number of temporal\_id, dependency\_id, and quality\_id three-tuples for which timing information, NAL HRD parameters, VCL HRD parameters and the pic\_struct\_present\_flag may be present.

**dependency\_id[ i ]** indicates the i-th dependency layer.

**quality\_id[ i ]** indicates the i-th quality layer.

**temporal\_id[ i ]** indicates the i-th temporal layer.

The following syntax elements apply to the coded video sequence that is obtained by the sub-bitstream extraction process as specified in subclause G.8.8.1 with tIdTarget equal to temporal\_id[ i ], dIdTarget equal to dependency\_id[ i ], and qIdTarget equal to quality\_id[ i ] as the input and the i-th sub-bitstream as the output.

**timing\_info\_present\_flag[ i ]** specifies the timing\_info\_present\_flag value of the i-th sub-bitstream.

**num\_units\_in\_tick[ i ]** specifies the num\_units\_in\_tick value of the i-th sub-bitstream.

**time\_scale[ i ]** specifies the time\_scale value of the i-th sub-bitstream.

**fixed\_frame\_rate\_flag[ i ]** specifies the fixed\_frame\_rate\_flag value of the i-th sub-bitstream.

**`nal_hrd_parameters_present_flag[ i ]`** specifies the `nal_hrd_parameters_present_flag` value of the *i*-th sub-bitstream.

**`vcl_hrd_parameters_present_flag[ i ]`** specifies the `vcl_hrd_parameters_present_flag` value of the *i*-th sub-bitstream.

**`low_delay_hrd_flag[ i ]`** specifies the `low_delay_hrd_flag` value of the *i*-th sub-bitstream.

**`pic_struct_present_flag[ i ]`** specifies the `pic_struct_present_flag` value of the *i*-th sub-bitstream.



## SERIES OF ITU-T RECOMMENDATIONS

|                 |                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------|
| Series A        | Organization of the work of ITU-T                                                           |
| Series D        | General tariff principles                                                                   |
| Series E        | Overall network operation, telephone service, service operation and human factors           |
| Series F        | Non-telephone telecommunication services                                                    |
| Series G        | Transmission systems and media, digital systems and networks                                |
| <b>Series H</b> | <b>Audiovisual and multimedia systems</b>                                                   |
| Series I        | Integrated services digital network                                                         |
| Series J        | Cable networks and transmission of television, sound programme and other multimedia signals |
| Series K        | Protection against interference                                                             |
| Series L        | Construction, installation and protection of cables and other elements of outside plant     |
| Series M        | Telecommunication management, including TMN and network maintenance                         |
| Series N        | Maintenance: international sound programme and television transmission circuits             |
| Series O        | Specifications of measuring equipment                                                       |
| Series P        | Telephone transmission quality, telephone installations, local line networks                |
| Series Q        | Switching and signalling                                                                    |
| Series R        | Telegraph transmission                                                                      |
| Series S        | Telegraph services terminal equipment                                                       |
| Series T        | Terminals for telematic services                                                            |
| Series U        | Telegraph switching                                                                         |
| Series V        | Data communication over the telephone network                                               |
| Series X        | Data networks, open system communications and security                                      |
| Series Y        | Global information infrastructure, Internet protocol aspects and next-generation networks   |
| Series Z        | Languages and general software aspects for telecommunication systems                        |