CLAIMS

1. A method implemented by a decoder, the method comprising:

receiving, by a receiver of the decoder, a bitstream comprising one or more layers of coded pictures and a video parameter set (VPS), wherein the VPS includes an output layer set (OLS) mode identification code (ols_mode_idc) specifying, when the ols_mode_idc is equal to zero, that a total number of OLSs specified by the VPS is equal to a number of layers specified by the VPS, that an i-th OLS includes layers with layer indices from zero to i, inclusive, and that for each OLS only the highest layer in the each OLS is an output layer;

determining, by a processor of the decoder, an output layer based on the ols_mode_idc in the VPS; and

decoding, by a processor of the decoder, a coded picture from the output layer to produce a decoded picture.

2. The method of claim 1, wherein the VPS includes a VPS maximum layers minus one (vps_max_layers_minus1) plus 1 specifies the maximum allowed number of layers in each CVS referring to the VPS.

3. The method of any of claims 1-2, wherein the total number of OLSs (TotalNumOlss) is equal to vps_max_layers_minus1 plus one when the ols_mode_idc is equal to zero or when the ols_mode_idc is equal to one.

4. The method of any of claims 1-3, wherein a number of layers in an i-th OLS (NumLayersInOls[i]) and a layer identifier (ID) in an OLS (LayerIdInOLS[i][j]) specifying a network abstraction layer (NAL) unit header layer identifier (nuh_layer_id) value of a j-th layer in the i-th OLS are derived as follows:

```
NumLayersInOls[ 0 ] = 1
LayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
for( i = 1, i < TotalNumOlss; i++ ) {
        if( each_layer_is_an_ols_flag ) {
                NumLayersInOls[ i ] = 1
                LayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
        } else if( ols_mode_idc  = =  0  ||  ols_mode_idc  = =  1 ) {
                NumLayersInOls[ i ] = i + 1
```

```
        for( j = 0; j < NumLayersInOls[ i ]; j++ )
            LayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
```

where vps_layer_id[i] is an i-th VPS layer identifier, TotalNumOlss is a total number of OLSs specified by the VPS, and an each_layer_is_an_ols_flag is an each layer is an OLS flag that specifies whether at least one OLS contains more than one layer.

5. A method implemented by an encoder, the method comprising:

encoding, by a processor of the encoder, a bitstream comprising one or more layers of coded pictures;

encoding into the bitstream, by the processor, a video parameter set (VPS), wherein the VPS includes an output layer set (OLS) mode identification code (ols_mode_idc) specifying, when the ols_mode_idc is equal to zero, that a total number of OLSs specified by the VPS is equal to a number of layers specified by the VPS, that an i-th OLS includes layers with layer indices from zero to i, inclusive, and that for each OLS only the highest layer in the each OLS is an output layer; and

storing, by a memory coupled to the processor, the bitstream for communication toward a decoder.

6. The method of claim 5, wherein the VPS includes a VPS maximum layers minus one (vps_max_layers_minus1) plus 1 specifies the maximum allowed number of layers in each CVS referring to the VPS .

7. The method of any of claims 5-6, wherein the total number of OLSs (TotalNumOlss) is equal to vps_max_layers_minus1 plus one when the ols_mode_idc is equal to zero or when the ols_mode_idc is equal to one.

8. The method of any of claims 5-7, wherein a number of layers in an i-th OLS (NumLayersInOls[i]) and a layer identifier (ID) in an OLS (LayerIdInOLS[i][j]) specifying a network abstraction layer (NAL) unit header layer identifier (nuh_layer_id) value of a j-th layer in the i-th OLS are derived as follows:

```
NumLayersInOls[ 0 ] = 1
LayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
for( i = 1, i < TotalNumOlss; i++ ) {
    if( each_layer_is_an_ols_flag ) {
```

```
            NumLayersInOls[ i ] = 1
            LayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
        } else if( ols_mode_idc  = =  0  ||  ols_mode_idc  = =  1 ) {
            NumLayersInOls[ i ] = i + 1
            for( j = 0; j < NumLayersInOls[ i ]; j++ )
                LayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
```

where vps_layer_id[i] is an i-th VPS layer identifier, TotalNumOlss is a total number of OLSs specified by the VPS, and an each_layer_is_an_ols_flag is an each layer is an OLS flag that specifies whether at least one OLS contains more than one layer.

9. A video coding device comprising:

a processor, a receiver coupled to the processor, a memory coupled to the processor, and a transmitter coupled to the processor, wherein the processor, receiver, memory, and transmitter are configured to perform the method of any of claims 1-8.

10. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of claims 1-8.

11. A decoder comprising:

a receiving means for receiving a bitstream comprising one or more layers of coded pictures and a video parameter set (VPS), wherein the VPS includes an output layer set (OLS) mode identification code (ols_mode_idc) specifying, when the ols_mode_idc is equal to zero, that a total number of OLSs specified by the VPS is equal to a number of layers specified by the VPS, that an i-th OLS includes layers with layer indices from zero to i, inclusive, and that for each OLS only the highest layer in the each OLS is an output layer;

a determining means for determining an output layer based on the ols_mode_idc in the VPS;

a decoding means for decoding a coded picture from the output layer to produce a decoded picture; and

a forwarding means for forwarding the decoded picture for display as part of a decoded video sequence.

12. The decoder of claim 11, wherein the decoder is further configured to perform the method of any of claims 2-4.

13. An encoder comprising:

an encoding means for:

encoding a bitstream comprising one or more layers of coded pictures; and

encoding into the bitstream a video parameter set (VPS), wherein the VPS includes an output layer set (OLS) mode identification code (ols_mode_idc) specifying that a total number of OLSs specified by the VPS is equal to a number of layers specified by the VPS, an i-th OLS includes layers with layer indices from zero to i, inclusive, and for each OLS only the highest layer in the each OLS is an output layer when the ols_mode_idc is equal to zero; and

a storing means for storing the bitstream for communication toward a decoder.

14. The encoder of claim 13, wherein the encoder is further configured to perform the method of any of claims 6-8.