

AV1 Layer-specific OBUs: Summary and Recommendations

Wan-Teh Chang <wtc@google.com>

Presented by: Adrian Grange <agrange@google.com>
2024-04-15

eleft pull request 281

By eleft (Alex Eleftheriadis), merged on Sep 9, 2019

RTC SG clarifications on high-level syntax:

<https://github.com/AOMediaCodec/av1-spec/pull/281>

[Commit 86fb0ac](#)

OBU header

```
obu_header() {                                     Type
    ..
    obu_extension_flag                             f(1)
    ..
    if ( obu_extension_flag == 1 )
        obu_extension_header()
}

obu_extension_header() {                           Type
    temporal_id                                    f(3)
    spatial_id                                      f(2)
    extension_header_reserved_3bits                f(3)
}
```

Layers

Layer

A set of tile group OBUs with identical **spatial_id** and identical **temporal_id** values.

Base layer

The layer with **spatial_id** and **temporal_id** values equal to 0.

Enhancement layer

A layer with either **spatial_id** greater than 0 or **temporal_id** greater than 0.

The drop_obu() condition

```
if ( obu_type != OBU_SEQUENCE_HEADER &&
    obu_type != OBU_TEMPORAL_DELIMITER &&
    OperatingPointIdc != 0 &&
    obu_extension_flag == 1 )
{
    inTemporalLayer = (OperatingPointIdc >> temporal_id ) & 1
    inSpatialLayer = (OperatingPointIdc >> ( spatial_id + 8 ) ) & 1
    if ( !inTemporalLayer || ! inSpatialLayer ) {
        drop_obu( )
        Return
    }
}
```

Consequences of the drop_obu() condition

- **Temporal delimiter** and **sequence header** OBUs are treated differently from other types of OBUs
 - The first pull request generalized this distinction as the notion of **layer-specific OBUs**
- If **OperatingPointIdc** is 0, all OBUs are processed
- If **OperatingPointIdc** is not 0, the operating point bitmask has an effect only on those OBUs with **obu_extension_flag equal to 1**

Summary of the eleft pull request (1/3)

- Introduce the notion of layer-specific OBUs (6.2.2)
- For each OBU type, indicate whether it is layer-specific (table in 6.2.2)
- For each metadata OBU type, indicate whether it is layer-specific (table in 6.7.1)
- New requirement: **OBUs that are not layer-specific must have the obu_extension_flag set to 0.** (6.2.2)
- Clarify that temporal_id and spatial_id default to 0 only for layer-specific OBUs. (6.2.3)
 - Implication: non-layer-specific OBUs don't have temporal_id and spatial_id. If a non-layer-specific OBU has an obu_extension_header(), the decoder ignores the temporal_id and spatial_id in it.

Summary of the eleft pull request (2/3)

6.4.1. General sequence header OBU semantics

Fix the ambiguous, incomplete **requirement of bitstream conformance**

Errata 1

- If `operating_point_idc[op]` is not equal to 0 for any value of `op` from 0 to `operating_points_cnt_minus_1`, it is a requirement of bitstream conformance that `obu_extension_flag` is equal to 1.

eleft pull request

- If `operating_point_idc[op]` is not equal to 0 for any value of `op` from 0 to `operating_points_cnt_minus_1`, it is a requirement of bitstream conformance that `obu_extension_flag` is equal to 1 **for all layer-specific OBUs in the coded video sequence.**

Summary of the elefth pull request (3/3)

Metadata OBUs (6.7.1)

- Clarify the **layer scope** of metadata OBUs
 - Which layers?
 - obu_extension_flag=0: all operating points
 - obu_extension_flag=1: the specific layer identified in the OBU extension header
 - Starting from the point the metadata OBU appears in the bitstream
 - Does not say where the scope ends
- For each metadata type, indicate whether it is layer-specific
- Among the currently-defined metadata types, only METADATA_TYPE_ITUT_T35 may be layer-specific (depends on the payload)

Recommendations for the eleft pull request

- Keep the notion of layer-specific OBUs
- Keep the clarifications
- To prevent existing bitstreams from becoming non-compliant, downgrade the two new requirements to recommendations (“should”)
 - See the pull request <https://github.com/AOMediaCodec/av1-spec/pull/350>

First change in pull request 350

Section 6.2.2. OBU header semantics

Errata 1:

<None>

Pull request 350:

OBUs that are not layer-specific **should** have the `obu_extension_flag` set to 0.

Second change in pull request 350

6.4.1. General sequence header OBU semantics

Errata 1:

- If `operating_point_idc[op]` is not equal to 0 for any value of `op` from 0 to `operating_points_cnt_minus_1`, **it is a requirement of bitstream conformance that `obu_extension_flag` is equal to 1.**

Pull request 350:

- If `operating_point_idc[op]` is not equal to 0 for any value of `op` from 0 to `operating_points_cnt_minus_1`, `obu_extension_flag` **should be equal to 1 for all layer-specific OBUs in the coded video sequence.**