ISO/IEC JTC 1/SC 29/WG 03 N0903

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Technologies under consideration on ISO/IEC 23090-13 VDI

**Status:** Approved

**Date of document:** 2023-01-20

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 9 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0903**

**April 2023, Antalya, TR**

|  |  |
| --- | --- |
| **Title** | **Technologies under consideration on ISO/IEC 23090-13 VDI** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **22646** |

# Introduction

This document collects technologies being under study for consideration in the development of the standard ISO/IEC 14496-34 Syntactic Description Language

# Table of content

[Introduction 1](#_Toc133590001)

[Table of content 1](#_Toc133590002)

[1. Improvements and fixes on the API extension definition (m62994) 1](#_Toc133590003)

[2. About a hypothetical reference systems decoder model 1](#_Toc133590004)

# Improvements and fixes on the API extension definition (m62994)

ISO/IEC 23090-13 defines extensions to video decoding engine (VDE) APIs in a generic manner to address the needs of immersive applications.

I has been identified some possible improvement an fixes on:

* additional structure definitions and their semantics
* editorial improvements
* improve definition of the API extensions

For references, the proposed text changes were provided in m62994 and are also attached with the present document.

# About a hypothetical reference systems decoder model

## A concrete use case

A concrete use case for using VDI is the usage of in an AR split rendering context. As an example, 4 bitstreams may be available

* A separate video stream for left and right eye
* An alpha channel for blending
* Depth information

The information then can directly processed in an XR Runtime that conforms to an OpenXR API. However, such an interface typically requires RGBA data, i.e. YUV data may be converted to RGB directly.

In order to be referenced in an external specification, VDI needs to be combined as follows:

* A *video decoding interface* shall be supported with the following parameters
  + “Profile”: sub-sampling, bit depth (example: 4:2:0, 10 bit)
  + Aggregate level capabilities: MB/s, bitrate, output constraints (level 6.1)
  + Number of instances (for example 16)
  + Codecs included: (AVC and HEVC), i.e. bitstreams may have different codecs
* Functionalities of the *video decoding interface*:
  + Decoding of conforming video bitstreams
  + Common coded picture buffer for a group of decoders
  + Associated access units across bitstreams provided at the same time to the coded picture buffer are provided synchronously at the output for a group of decoders
  + SEI and other metadata messages associated the access unit are provided synchronously at the decoder output
  + Conversion to colour spaces
* Signaling in content:
  + Max capabilities of each bitstream (regular profile/level signaling)
  + Group of coded bitstreams are associated to a group with max capabilities

## Summary of possible additions

On a high-level, the proposed additions are

* The VDI is only document from an API perspective.
* It addresses aggregate capabilities, but those are not related to a video decoding pipeline that included:
  + Scheduling of input data provided to the decoder
  + Memory and bandwidth of the decoder on the output
* The definition of a conforming set/group of bitstreams treated by one video decoding engine
* The definition of a conforming video decoding platform with multiple concurrent decoders
* The provisioning of signaling in order to annotate one bitstream or a set of bitstream for conformance
* The definition of a Hypothetical Reference Decoder for the Video Decoding interface
* The potential ability to add well-defined post-processing such as colour space transform
* The provisioning of associated data which each decoded picture, such as timing information or rendering poses

Concretely, in the HRD

* Extension to Hypothetical Reference Decoder of H.265/HEVC
  + Bitstream conformance according to C.2
  + Bitstream conformance aligned with C.4
    - Not a single bitstream but a group of bitstream
    - Their aggregate needs to within boundaries
  + Decoder conformance aligned with C.5
    - Not a single bitstream but a group of bitstream
    - Their aggregate needs to within boundaries
* Profile/level restrictions apply to a set of decoders running in parallel
* Additional efforts
  + Creation of test and conformance test vectors for VDI
  + Creation of a test regime for a multi-decoder functionality

## Starting point for a HRD-like model for VDI

### Example of HEVC HRD

No tutorial of the complex HEVC HRD is provided, but the basic functionalities of the HRD are provided in Figure 1.

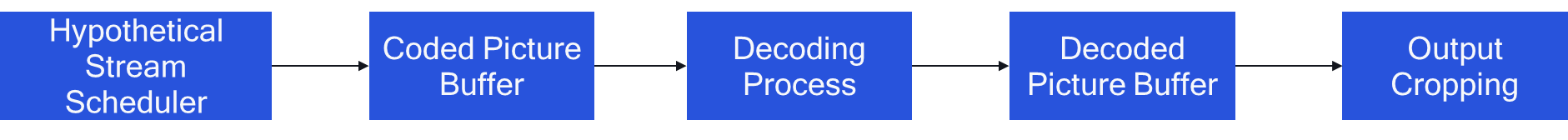


Figure 1 HEVC HRD

The HRD operates as follows:

* The HRD is initialized at decoding unit 0, with both the CPB and the DPB being set to be empty (the DPB fullness is set equal to 0).
* Data associated with decoding units that flow into the CPB according to a specified arrival schedule are delivered by the hypothetical stream scheduler (HSS).
* The data associated with each decoding unit are removed and decoded instantaneously by the instantaneous decoding process at the CPB removal time of the decoding unit.
* Each decoded picture is placed in the DPB.
* A decoded picture is removed from the DPB when it becomes no longer needed for inter prediction reference and no longer needed for output.

The bitstream is tested by the HRD for conformance as specified in clause C.1

All of the following conditions shall be fulfilled

* The initial removal delay needs to be such that each AU is available in the buffer when it is removed.
* The CPB shall never overflow, i.e. the total number of bits in the CPB shall be smaller than or equal to CBP size
* In case of no low delay, the CPB shall never underflow, i.e.
  + AuNominalRemovalTime[ n ] >= AuFinalArrivalTime[ n ]
  + DuNominalRemovalTime[ m ] >= DuFinalArrivalTime[ m ]
* In case of low delay, if DuNominalRemovalTime[ m ] < DuFinalArrivalTime[ m ], then AuNominalRemovalTime[ n ] < AuFinalArrivalTime[ n ]
* Profile/level specific requirements for cpb removal times ávoiding to process too many samples at the output.
* The number of decoded pictures in the DPB shall be less than or equal to sps\_max\_dec\_pic\_buffering\_minus1[ HighestTid ]
* All reference pictures shall be present in the DPB when needed for prediction.
* Difference of maxPicOrderCnt − minPicOrderCnt is restricted
* Profile/level specific requirements for decoded picture output interval
* The decoding units need to match the access unit delays, but this seems to be buggy in the spec
* If DpbOutputTime[ m ] is greater than DpbOutputTime[ n ], then PicOrderCntVal[ m ] > PicOrderCntVal[ n ]

To check conformance of a decoder, test bitstreams conforming to the claimed profile, tier and level, as specified in clause C.4 are delivered by a hypothetical stream scheduler (HSS) both to the HRD and to the decoder under test (DUT).

All cropped decoded pictures output by the HRD shall also be output by the DUT, each cropped decoded picture output by the DUT shall be a picture with PicOutputFlag equal to 1, and, for each such cropped decoded picture output by the DUT, the values of all samples that are output shall be equal to the values of the samples produced by the specified decoding process.

For output timing decoder conformance, the HSS operates as described above, with delivery schedules selected only from the subset of values of SchedSelIdx for which the bit rate and CPB size are restricted as specified in Annex A for the specified profile, tier and level or with "interpolated" delivery schedules as specified below for which the bit rate and CPB size are restricted as specified in Annex A.

The same delivery schedule is used for both the HRD and the DUT.

In summary, the HRD specification is as follows

* For a bitstream conforming to a profile/level/tier and for given HRD parameters under HSS operation, the following holds
  + CBP does never overflow
  + CBP does never underflow
  + Maximum sample processing rate at the decoder output is not exceeded
  + Number of decoded pictures that need to be stored in DPB is not exceeded
  + The decoded picture output interval is not below a specified limit
* For a decoder conforming to a profile/level/tier and for given HRD parameters under HSS operation, any bitstream conforming to the HRD constraints shall be decodable

### VDI HRD

### Key extensions

Assume there are N video streams that need to be decoded concurrently. Each of the N video streams follows some profile/level/tier requirements and HRD requirements and parameters specified along with the bitstream.

Three key models are to be considered

* Configuration 1: CPB and DPB run independently for each of the N decoders
  + Core Issue: aggregate CBP, decoding complexity and DBP must not exceed the limits of a single decoder HRD with the same parameters
* Configuration 2: CPB independently, but a combined DPB that ensures that the output of the N decoders can be provided with a proper timing in a synchronuous manner to the next level
  + Core Issues:
    - The core issue from configuration 1
    - the combined DPB not only follows the aggregate requirements, but also requires that all decoded pictures need to be stored in the DPB until the decoded pictures from all N decoders with a common timeline can be released
    - There are HRD parameters needed for each bitstream that may be different in case of operating multiple decoders in parallel
* Configuration 3: In addition to Config 2, also the Coded Picture buffer operates jointly.
  + Core Issues:
    - The core issues from configuration 2
    - The coded picture buffer operates sequentially on well defined decode timing of decoding units across different streams

### Option 1: Individual CBP and DPB

In Option 1, the CBP and the DPB for each stream are treated individually. This is the case when the decoder inputs and outputs are completely independent.

Treemap chart

Description automatically generated with medium confidence

Figure 3 Option 1: Individual CBP and DPB

The operation is as follows:

* Data associated with decoding units that flow into the CPB of each stream according to a specified arrival schedule are delivered by the common Hypothetical Stream Scheduler (HSS) that scheduled the N bitstreams for decoding each of the units. For each access unit
  + all data associated with an access unit is removed and decoded instantaneously by the instantaneous decoding process at CPB removal time of the access unit.
  + Each decoded picture is placed in the Decoded Picture Buffer (DPB) for being referenced by the decoding process of this stream as well as for output and cropping.
  + A decoded picture is removed from the DPB at the time that it becomes no longer needed for inter-prediction reference as well as the output time of the access unit.
* At any point time,
  + each of the individual streams conforms to the signaled profile/level/tier and HRD parameters of the individual stream
  + The sum of the CPB size conforms to common profile/level/tier signaling
  + The aggregate decoder processing speed (samples per seconds) conforms to common profile/level/tier signaling
  + The sum of the DPB size conforms to common profile/level/tier signaling
* Common HRD parameters for initial delay may be specified

### Option 2: Individual CBP – joint DPB

In Option 2, which is likely the most relevant one, a set of bit streams has individual input, but there are dependencies on the DPBs.

Treemap chart

Description automatically generated with low confidence

Figure 4 Individual CBP – joint DPB

The operation is as follows

* Data associated with decoding units that flow into the CPB of each stream according to a specified arrival schedule are delivered by the common Hypothetical Stream Scheduler (HSS) that scheduled the N bitstreams for decoding each of the units. For each access unit
  + all data associated with an access unit is removed and decoded instantaneously by the instantaneous decoding process at CPB removal time of the access unit.
  + Each decoded picture is placed in the Decoded Picture Buffer (DPB) for being referenced by the decoding process of this stream as well as for output and cropping.
  + A decoded picture is removed from the DPB at the time that it becomes no longer needed for inter-prediction reference as well as the output time of the access unit is the largest of all decoded pictures remaining in the group of decoders
* At any point time,
  + The same requirements as for configuration 1
  + The common DPB size conforms to common profile/level/tier signaling
* Again common HRD parameters for initial delay may be specified

### Option 3: Common CPB and DPB

In Option 3, also the coded picture buffer is shared, i.e. data can only be removed according to a sequence of decoding units shared across bitstreams.

[Editor’s note: The figure below has been identified to be update to align on the text, the left side (Coded Picture Buffer VDI) should show a combined CPB]

Treemap chart

Description automatically generated with low confidence

Figure 5 Common CPB and DPB

The operation is as follows:

* Data associated with decoding units that flow into the CPB of each stream according to a specified arrival schedule are delivered by the common Hypothetical Stream Scheduler (HSS) that scheduled the N bitstreams for decoding each of the units. The addition of each decoding unit is done according to the common HSS. For each access unit
  + all data associated with an access unit is removed and decoded instantaneously by the instantaneous decoding process at CPB removal time of the access unit.
  + Each decoded picture is placed in the Decoded Picture Buffer (DPB) for being referenced by the decoding process of this stream as well as for output and cropping.
  + A decoded picture is removed from the DPB at the time that it becomes no longer needed for inter-prediction reference as well as the output time of the access unit is the largest of all decoded pictures remaining in the group of decoders
* At any point time,
  + The same requirements as for configuration 1
  + The common DPB size conforms to common profile/level/tier signaling
  + The common CPB operation is conforms to common profile/level/tier signaling
* Again common HRD parameters for initial delay may be specified

### Bitstream conformance

The bitstream conformance is defined as follows

* A set of bitstreams of coded data conforming to the specification shall fulfil all requirements specified in this clause
* The set of bitstreams is tested by the VDI HRD for conformance
* The set of bitstreams may be annotated with
  + Common profile/level/tier information
  + Common HRD parameters, in particular for initial delay
  + Whether they have a common or separate CPB
  + Whether they have a common or separate DPB
* The set of bitstreams may be even be different profiles or they may be different codecs even. In this case some common terminology across codecs needs to be defined.

### Bitstream Generation Scenarios

There are a couple of scenarios that may be considered for bitstream generation

* Scenario 1: joint generation
  + Streams are encoded with VDI decoding in mind
  + Overall HRD parameters need to be defined
  + Overall a set of encoders is controlled ensure that the common HRD parameters are maintained
* Scenario 2: individual generation
  + Streams are encoded independently
  + Each stream is annotated with individual profile/level/tier and HRD parameters
  + Additional information may be provided for each bitstream to support joint decoding (for example decoded pictures)
  + From individual annotation, a common HRD operation may be derived by the decoder
* Scenario 3: Stream scheduler picks streams according to annotation parameters.
* Scenario 4: Extension to scenario 2 for which each bitstream includes encoding metadata such that HRD parameters can be derived on the fly at the decoding instance.