Unlocking the Power of Vulkan Video Encode

Agenda

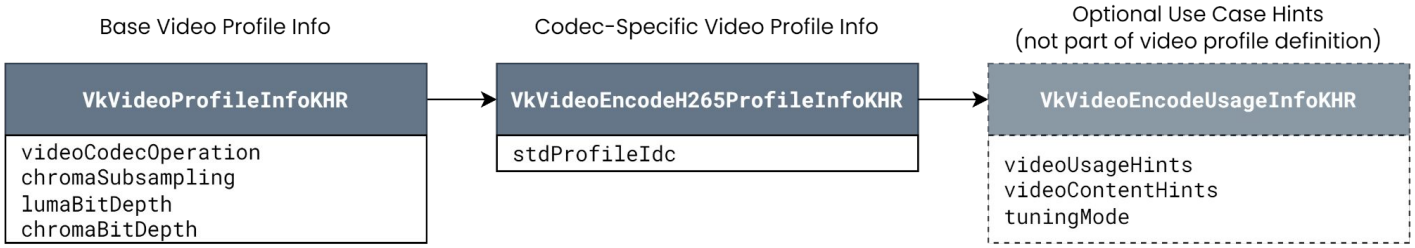
- **Vulkan Video Encode Introduction**
  - Daniel Rakos, RasterGrid
- **AV1 Encode**
  - Benjamin Cheng, AMD
- **Quantization Maps**
  - Srinath Kumarapuram, NVIDIA

Vulkan Video Encode Introduction

Vulkan Video Encode Introduction

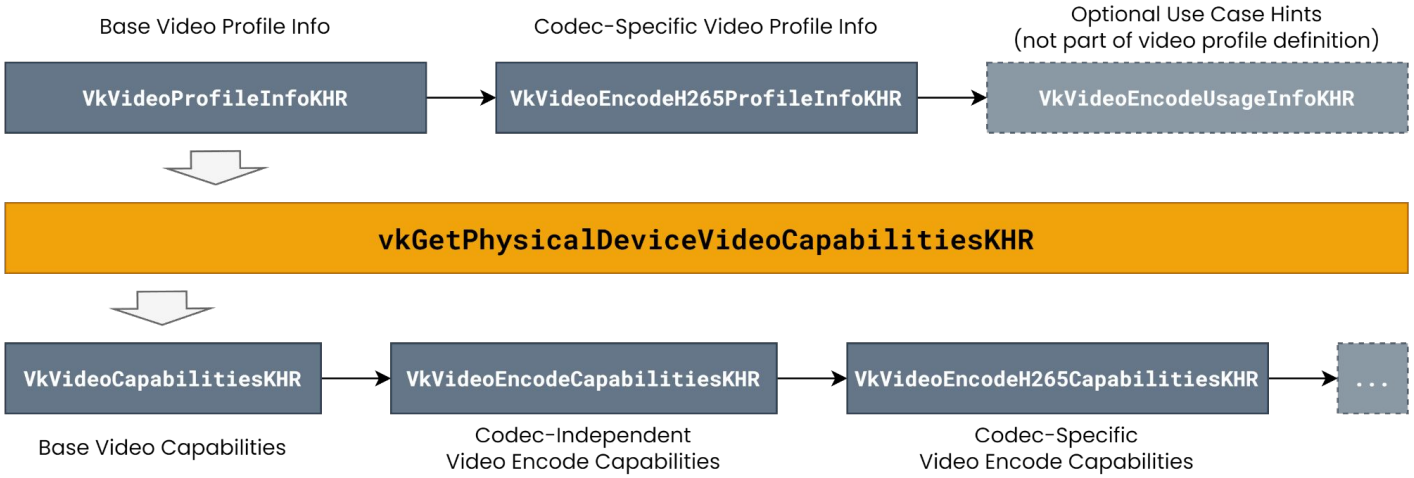
- **Video profiles**
  - how to construct them and how to query their capabilities / formats
- **Video sessions**
  - DPB slots and video picture resources
- **Video session parameters and their encode-specific aspects**
  - understanding overrides
  - video encode quality levels
- **Rate control**
  - available modes, codec-independent and codec-specific parameters
- **The anatomy of a video encode operation**
  - understanding codec-independent and codec-specific aspects

Video Profiles



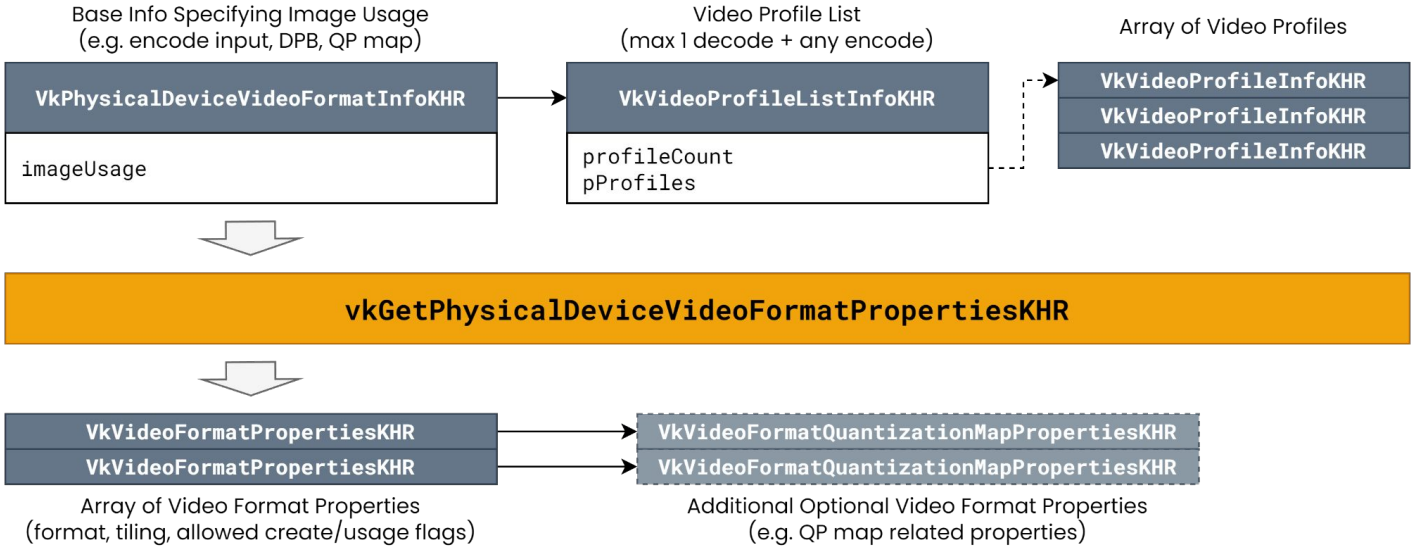
- **Includes different levels of information**
  - basic info
    - used codec (e.g. H.265 encode)
    - subsampling (e.g. 4:2:0)
    - bit depth (e.g. 8-bit)
  - codec-specific profile information
  - optional usage hints

Video Profile Capabilities



- vkGetPhysicalDeviceVideoCapabilitiesKHR
  - **input:** video profile info
  - **output:** codec-independent, codec-specific, and feature-specific capabilities

Video Profile Formats



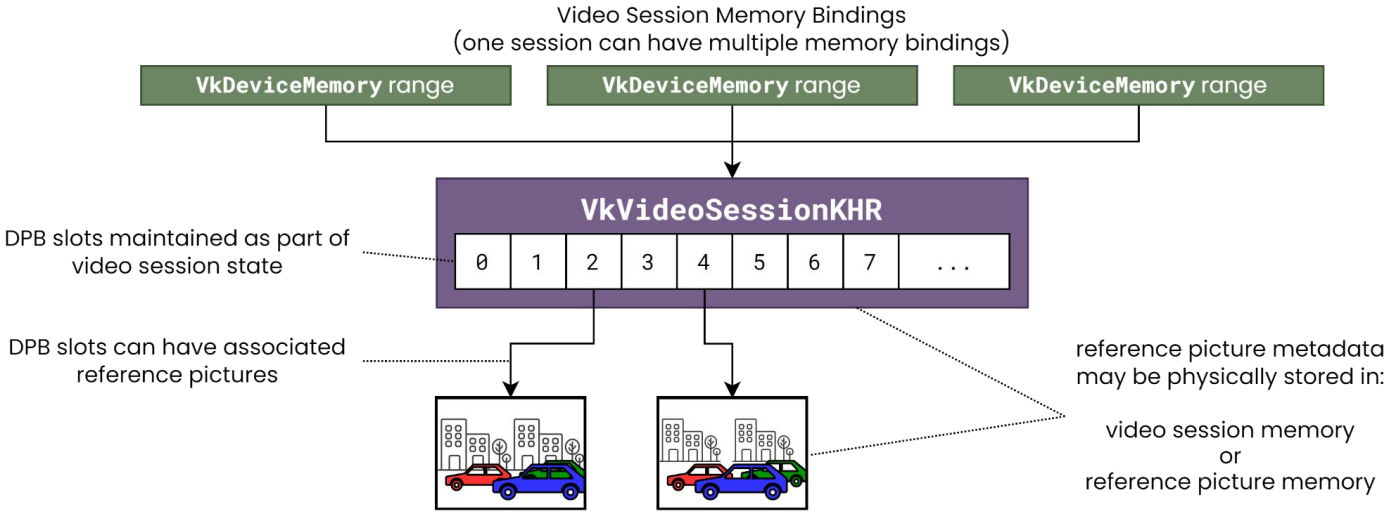
- vkGetPhysicalDeviceVideoFormatPropertiesKHR
  - **input:** image usage flags, video profile list
    - video profile list relevant for transcoding use cases
  - **output:** array of general and feature-specific format properties

Video Encode Resources

- **Video Encode Input Pictures - ENCODE_SRC usage bit**
  - VkImage with explicit video profile list or *video profile independent**
- **Video Reference (DPB) Pictures - ENCODE_DPB usage bit**
  - VkImage with explicit video profile list including the target video profile
- **Video Bitstream Buffer - ENCODE_DST usage bit**
  - VkBuffer with explicit video profile list or *video profile independent**
- **Video Encode Feedback Query Pool**
  - VkQueryPool with explicit video profile
  - used to retrieve e.g. the size of the encoded bitstream
- **Video Session and Video Session Parameters**

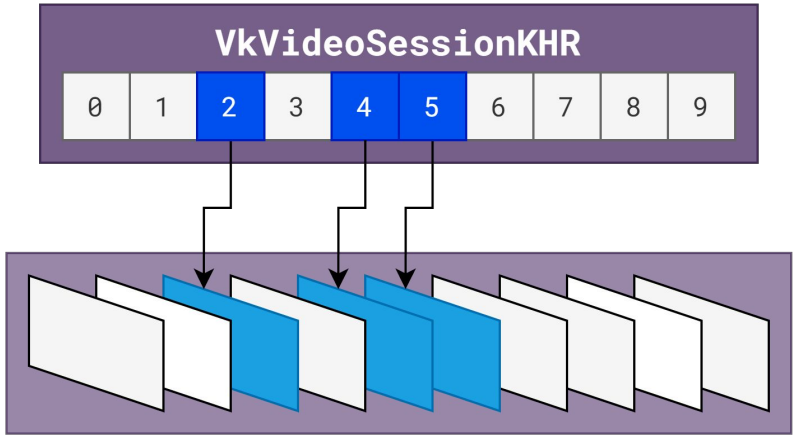
** video profile independent resources (except for DPB images) added by the VK_KHR_video_maintenance1 extension - see video profile compatibility*

Video Sessions



- **Maintains the state of the video stream**
  - state of the DPB slots and the associated video picture resources
    - reference metadata stored either in video session memory or picture memory
  - **encode-specific:** current rate control settings, quality level, etc.

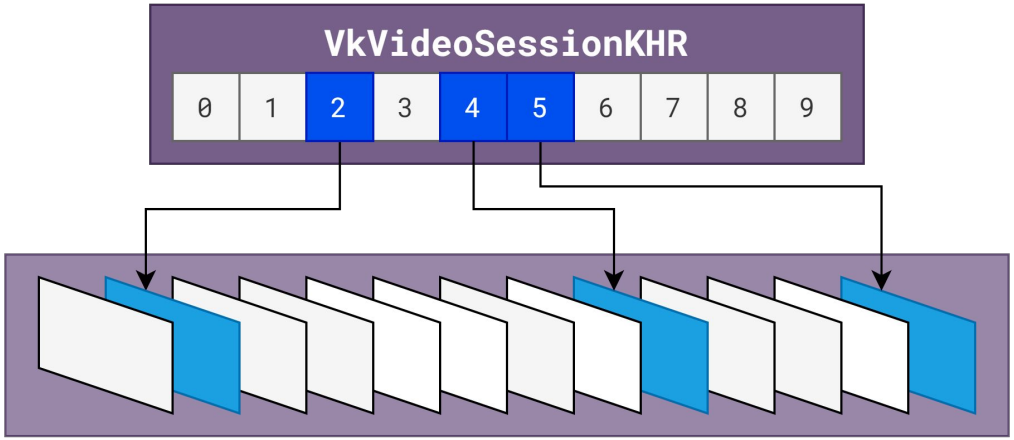
DPB Slots and Video Picture Resources



DPB image with multiple layers

- **Single DPB VkImage with multiple layers**
  - separate VkImageViews for individual video picture resources

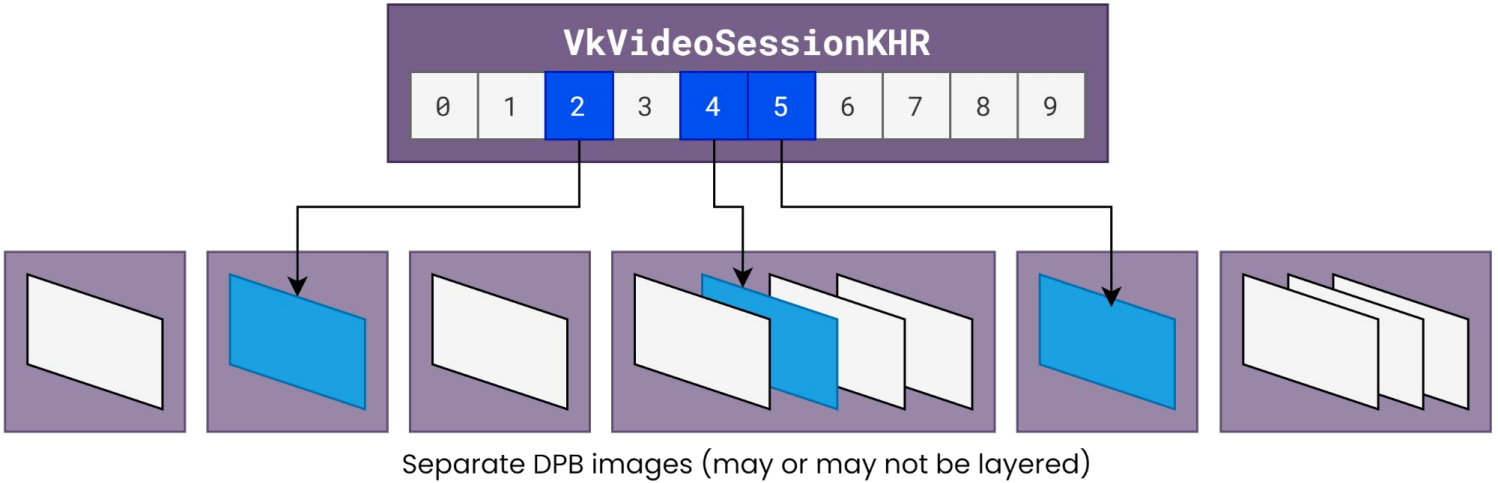
DPB Slots and Video Picture Resources



DPB image with multiple layers

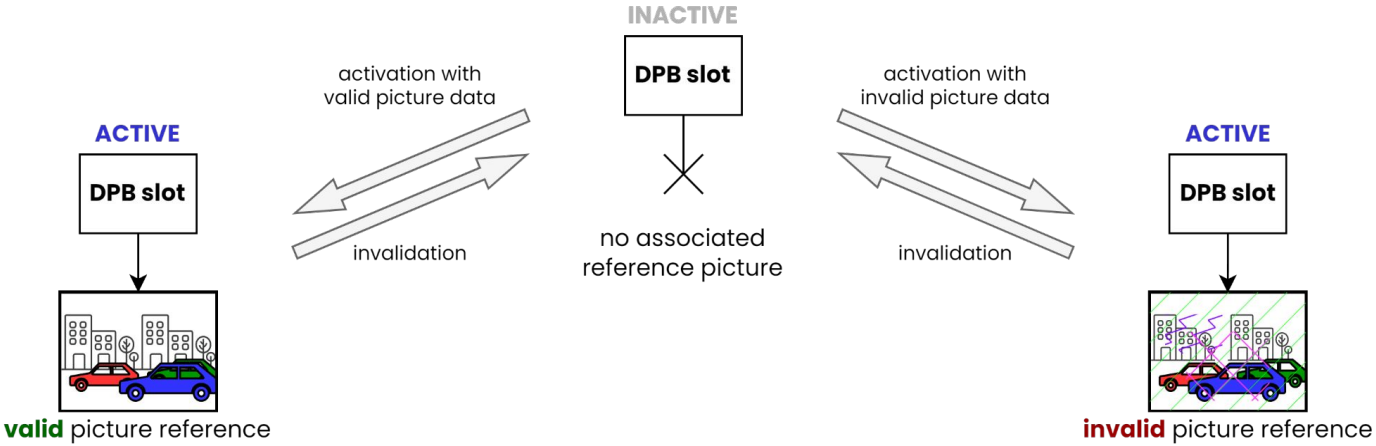
- **Single DPB VkImage with multiple layers**
  - separate VkImageViews for individual video picture resources
  - DPB slots may be associated with different picture resources over time

DPB Slots and Video Picture Resources



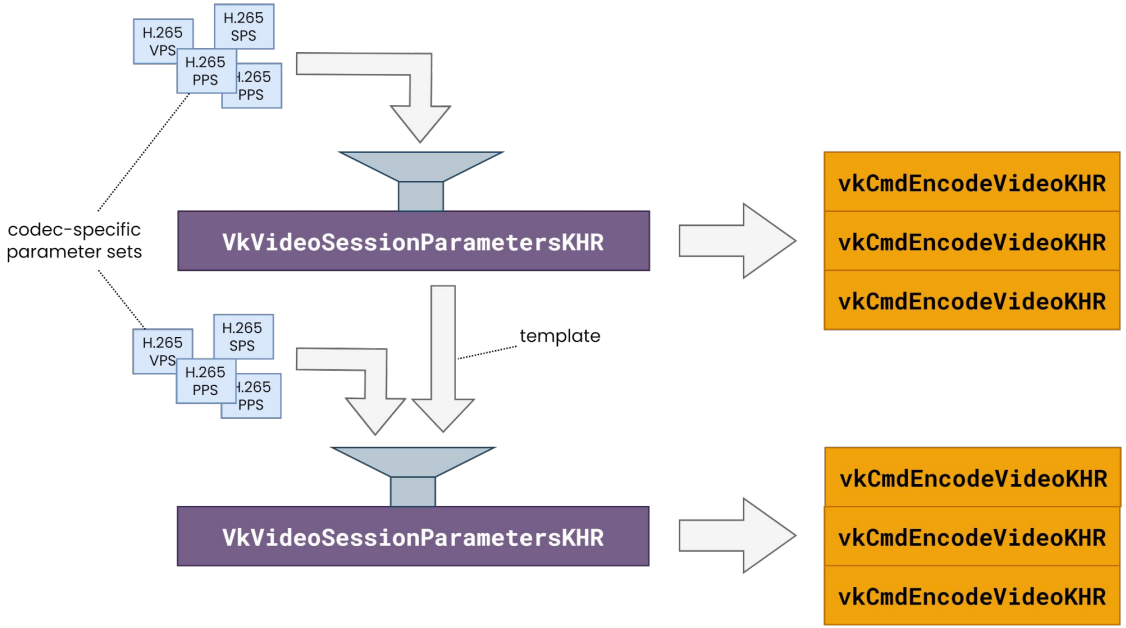
- **Separate VkImage objects for each DPB video picture resource**
  - requires VK_VIDEO_CAPABILITY_SEPARATE_REFERENCE_IMAGES_BIT_KHR

DPB State Management



- **Each DPB slot can be:**
  - *inactive* (does not hold any reference picture)
  - *active* (holds a reference picture)
    - with a *valid* picture reference (successful encode)
    - with an *invalid* picture reference (unsuccessful encode)

Video Session Parameters



- **Container objects holding codec-specific parameter sets**
  - e.g. H.264 SPS, PPS; H.265 VPS, SPS, PPS; AV1 sequence header
  - parsed once, used across multiple encode operations

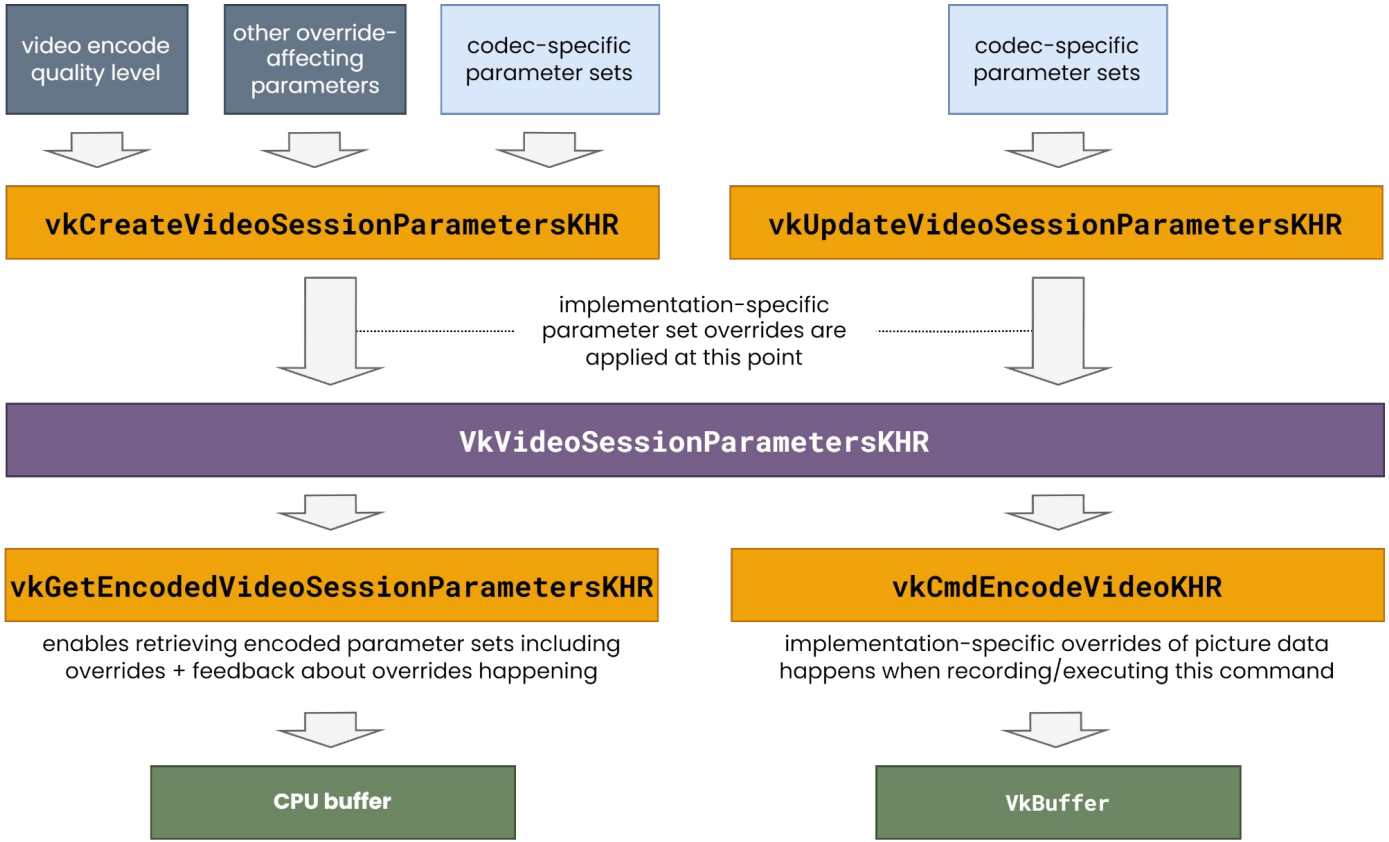
Video Encode Parameter Overrides

- **Encoders typically only support a limited subset of the coding tools**
  - too complex to expose all these implementation-specific limitations
    - instead, implementations are allowed to ***override*** unsupported combinations
    - some parameters are ***guaranteed*** (always or as reported by stdSyntaxFlags)
- **vkGetEncodedVideoSessionParametersKHR**
  - gives feedback about parameter set overrides
  - enables retrieving encoded parameter sets
- **Encoded picture parameters may also be overridden**
  - transparent to the application, but feedback may be available
    - VK_VIDEO_ENCODE_FEEDBACK_BITSTREAM_HAS_OVERRIDES_BIT_KHR
- **Optional (opt-in) overrides**
  - VK_VIDEO_SESSION_CREATE_ALLOW_ENCODE_PARAMETER_OPTIMIZATIONS_BIT_KHR

Video Encode Quality Levels

- **Different "presets" the encoder can be operated with**
  - optimized for specific use cases - different sets of coding tools used
  - vkGetPhysicalDeviceVideoEncodeQualityLevelPropertiesKHR
    - enables retrieving recommended settings for specific quality levels
- **Part of video session state**
  - can be changed with vkCmdControlVideoCodingKHR
    - VK_VIDEO_CODING_CONTROL_ENCODE_QUALITY_LEVEL_BIT_KHR
- **May affect overrides**
  - as they control the set of coding tools used
  - hence the quality level is an input to video session parameters creation
    - see quantization map texel size for another example

Video Session Parameters with Overrides



Video Encode Rate Control

- **Enables controlling bitrate / quality**
- **Supported modes**
  - DEFAULT - implementation-specific (default) mode
  - DISABLED - no rate control (also known as CQP)
  - CBR - constant bit rate
  - VBR - variable bit rate
- **Part of video session state**
  - can be changed with vkCmdControlVideoCodingKHR
    - VK_VIDEO_CODING_CONTROL_ENCODE_RATE_CONTROL_BIT_KHR
  - current state has to be specified to vkCmdBeginVideoCodingKHR
    - does **NOT** change the rate control state
    - only informs the driver about the current settings

Codec-Independent Rate Control Parameters

- **Support for multi-layer rate control**
  - enables different settings for each codec-specific temporal layer

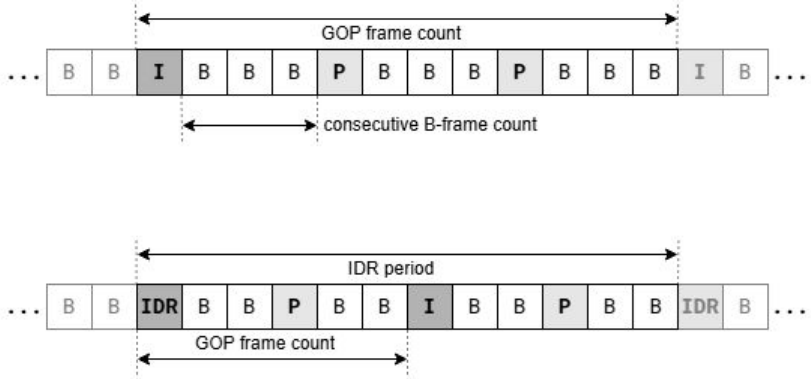
- **Global settings - VkVideoEncodeRateControlInfoKHR**
  - rate control mode
  - per rate control layer settings
  - virtual buffer parameters

- **Per-layer settings - VkVideoEncodeRateControlLayerInfoKHR**
  - average/max bitrate
  - framerate

Codec-Specific Rate Control Parameters

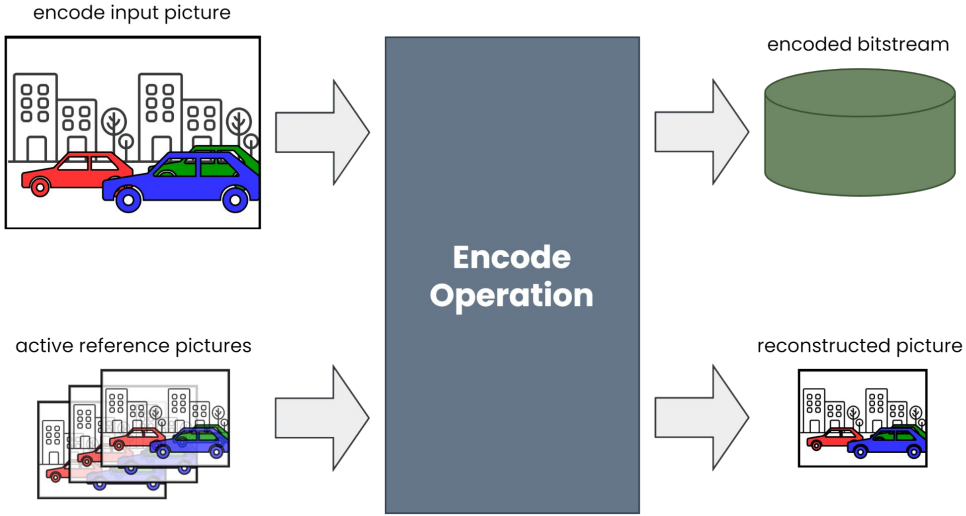
- **Optional**
  - but allows for better rate control decisions made by the implementation

- **Global settings**
  - group-of-pictures structure
    - frame count
    - consecutive B frames
    - IDR period
  - temporal layer count



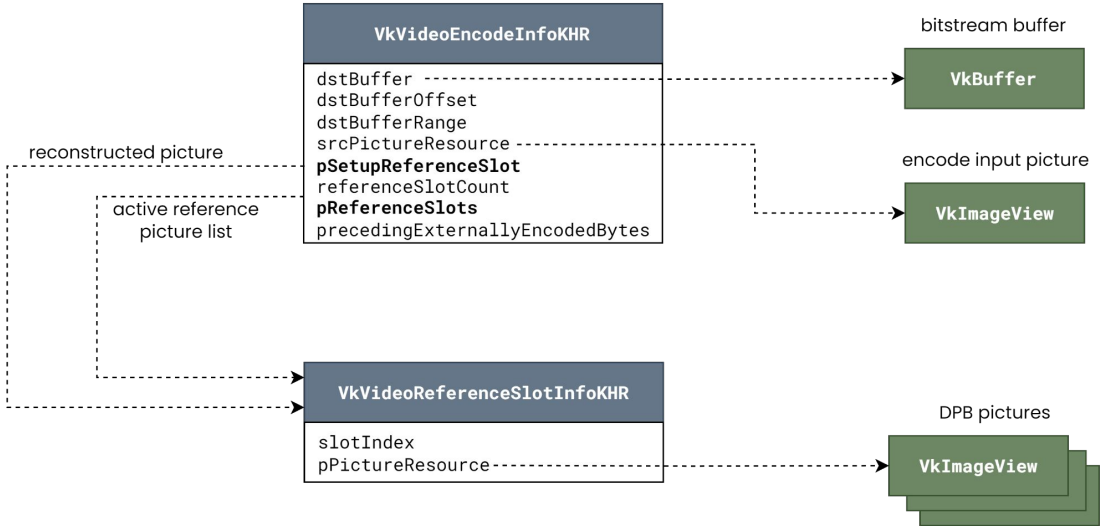
- **Per-layer settings**
  - e.g. min/max QP, maximum frame size

The Anatomy of a Video Encode Operation



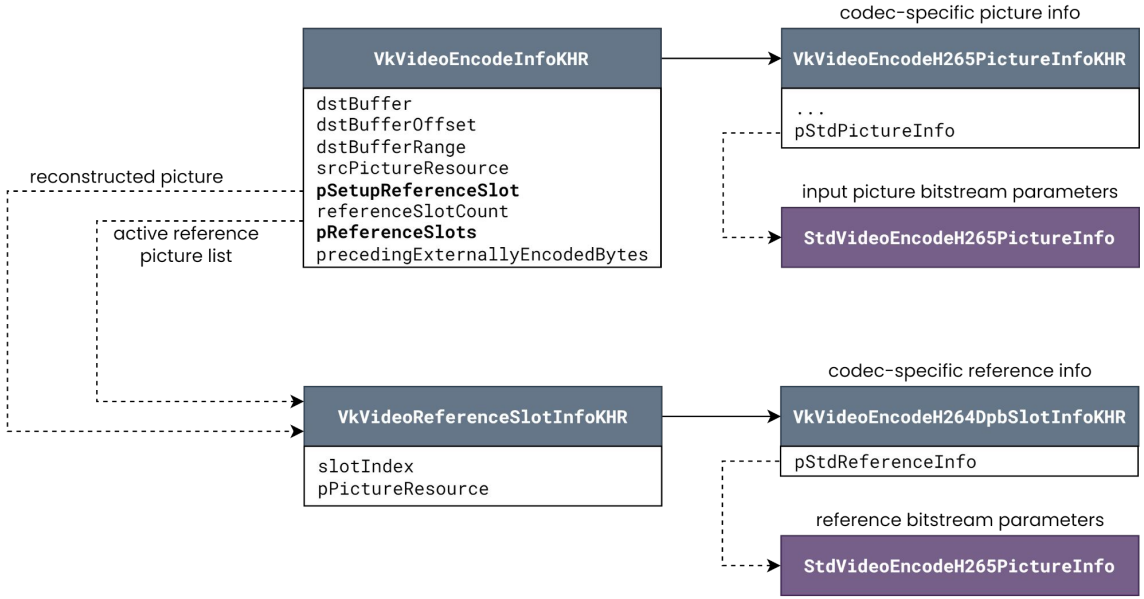
- **An encode operation performs the following:**
  - produce encoded bitstream for the encode input picture
  - use provided active reference pictures for prediction
  - reconstruct the encoded picture
    - optionally activate the DPB slot with it (if used as reference in the future)

The Anatomy of a Video Encode Operation



- **Reconstructed/reference picture resources must be bound**
  - as specified in the parameters of vkCmdBeginVideoCodingKHR
- **DPB slot has to be explicitly activated with the reconstructed picture**

The Anatomy of a Video Encode Operation



- **Codec-specific reference-to-DPB-slot mapping**
  - H.264 and H.265 - based on RefPicList0 and RefPicList1 arrays
  - AV1 - explicit reference name to DPB slot mapping

AV1 Encode

VK_KHR_video_encode_av1

Section Topics

- **AV1 Codec Overview**
  - (very) brief summary of codec features
  - new structures introduced
- **Vulkan AV1 Encode API params**
- **Overrides and Capabilities**
  - notable overrides (and guarantees)
  - highlight important capabilities to consider

AV1 Codec Overview

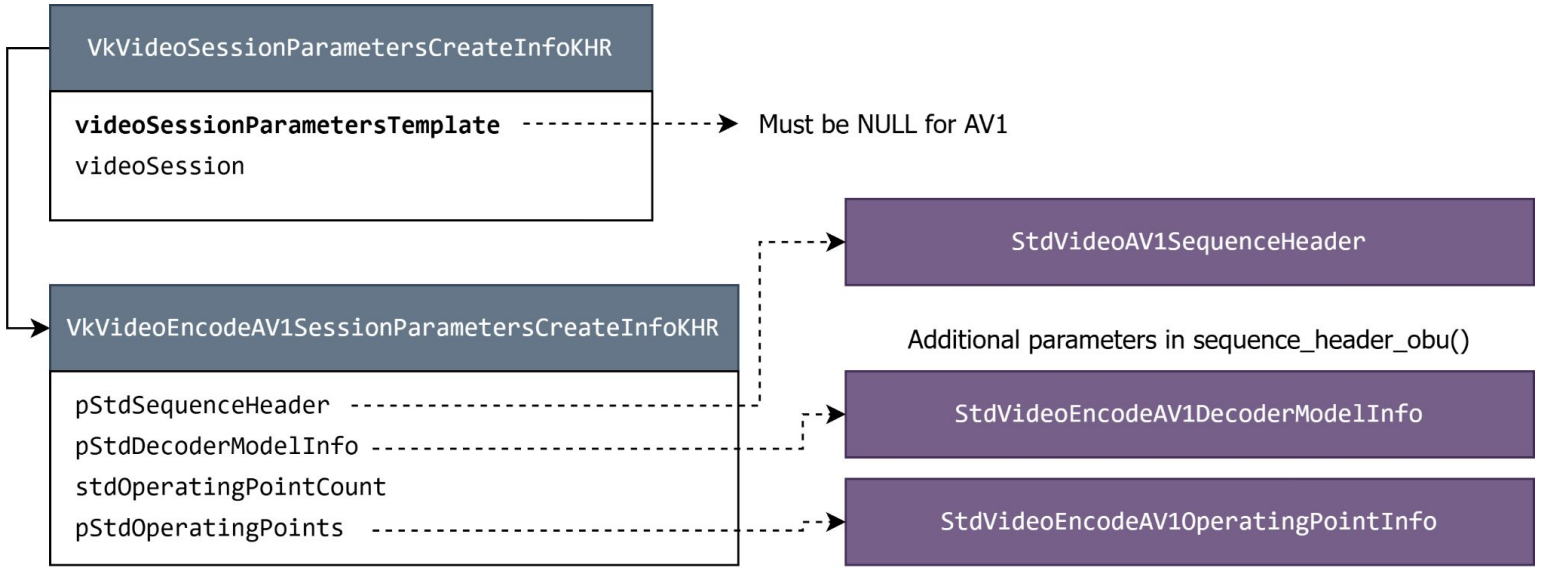
- **Relevance**
  - Open-source, royalty-free codec developed by the Alliance for Open Media
  - More efficient than H.264, can be more efficient than H.265*

- **Adoption**
  - Increasingly available HW support for both decode/encode

* Varies depending on encoder configurations and implementations

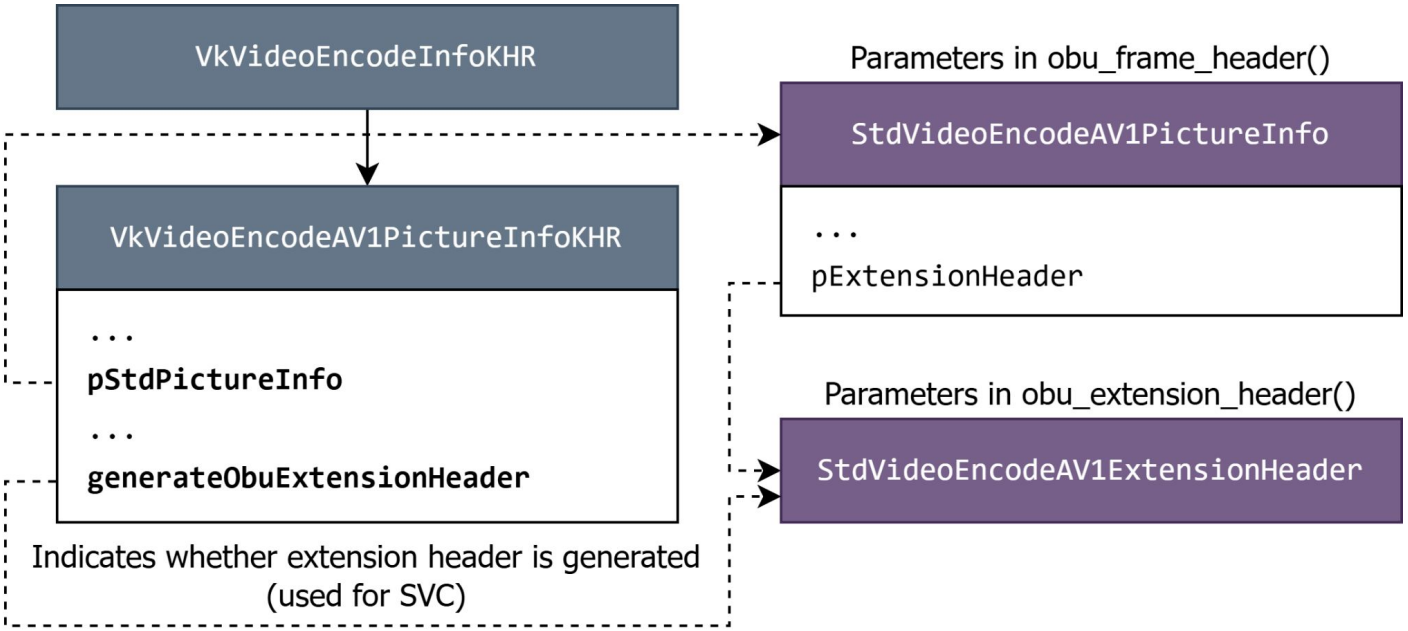
AV1 Codec Overview

- **Relevant parameters come in two places:**
  - Sequence Header (*sequence_header_obu()* in AV1 spec)



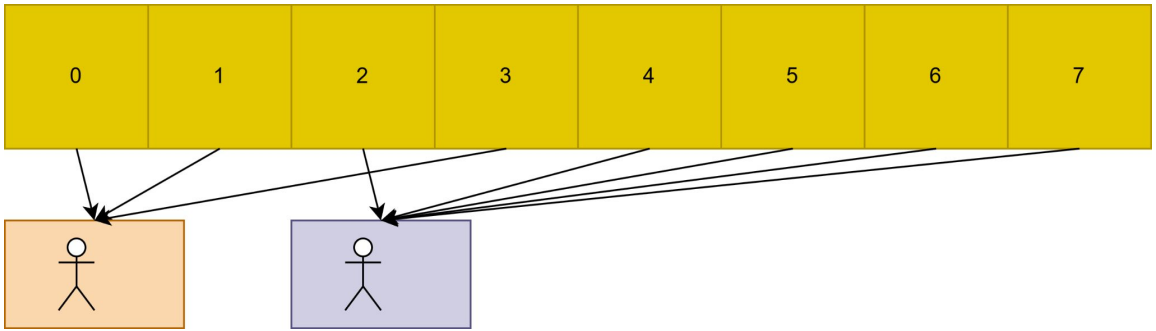
AV1 Codec Overview

- **Relevant parameters come in two places:**
  - Frame Header (*frame_header_obu()* in AV1 spec)



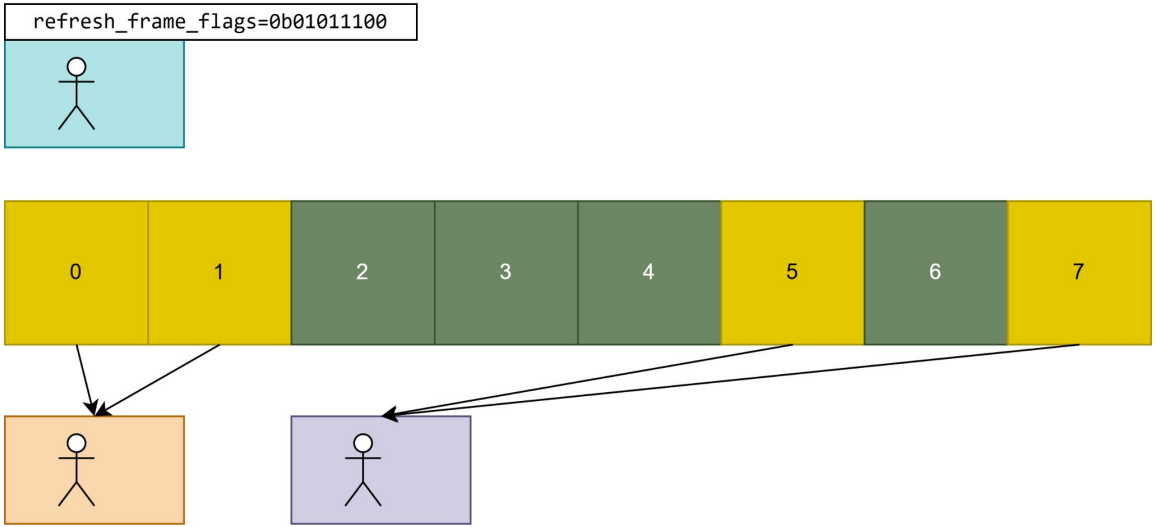
AV1 Codec Overview - References

- **Codec defines 8 *VBI* index slots (AV1 spec)**
  - Each of the slots points to a previously decoded picture
  - Multiple VBI index slots can point to the same picture



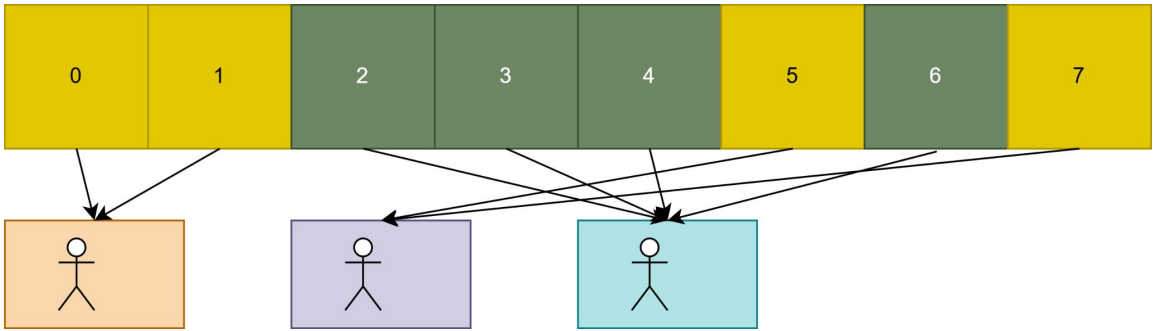
AV1 Codec Overview - References

- **When decoding a frame:**
  - *refresh_frame_flags* syntax indicates which VBI index slots are updated



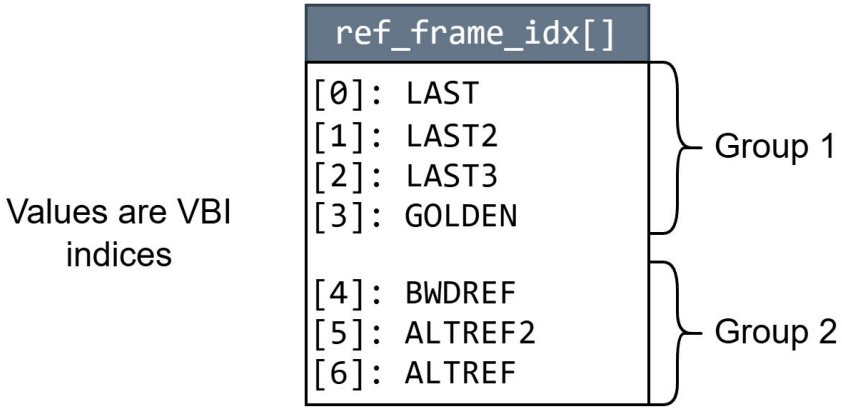
AV1 Codec Overview - References

- **When decoding a frame:**
  - *refresh_frame_flags* syntax indicates which VBI index slots are updated



AV1 Codec Overview - References

- **Each frame can reference up to 7 pictures, specified through ref_frame_idx**



- **Reminder:**
  - Multiple VBI entries may point to the same picture
  - Multiple reference names may point to the same VBI slot index (and also the same picture)

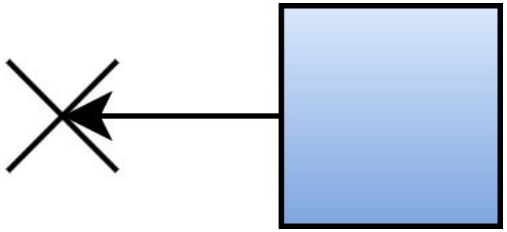
AV1 Codec Overview - Primary reference

- **Can reuse data from previously decoded frames (aside from pixel data)**
  - CDF data (entropy encoding probabilities)
  - Loop filter params
  - etc.

- ***primary_ref_frame* indicates the reference name to load this data from**
  - Allowed to specify NONE to improve error resilience

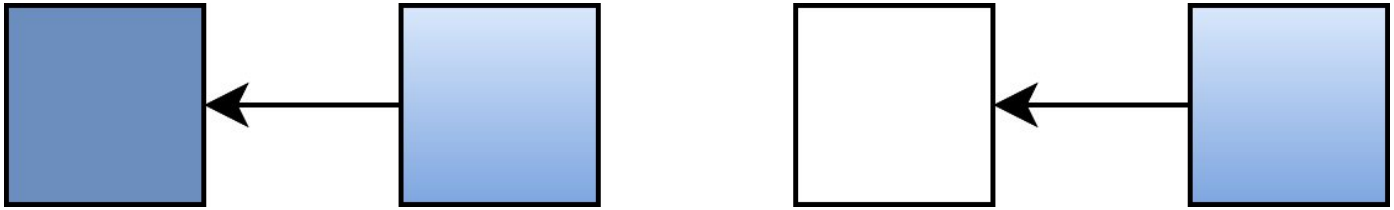
AV1 Codec Overview - Prediction Modes

- **Within a frame, blocks may be:**
  - Intra (no pixel data from another frame)



AV1 Codec Overview - Prediction Modes

- **Within a frame, blocks may be:**
  - Single Reference (pixel data from any reference frame)

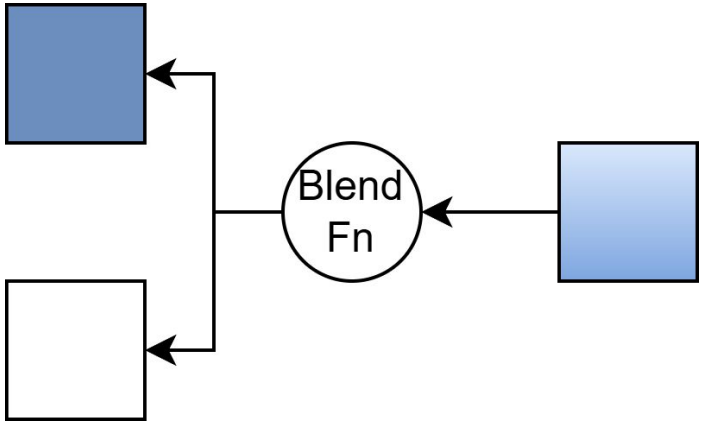


Despite the name, a frame using only single reference prediction can refer to multiple reference frames.

*Arrows indicate prediction direction, not data flow

AV1 Codec Overview - Prediction Modes

- **Within a frame, blocks may be:**
  - Unidirectional Compound (two reference frames - same group)
  - Bidirectional Compound (two reference frames - different group)



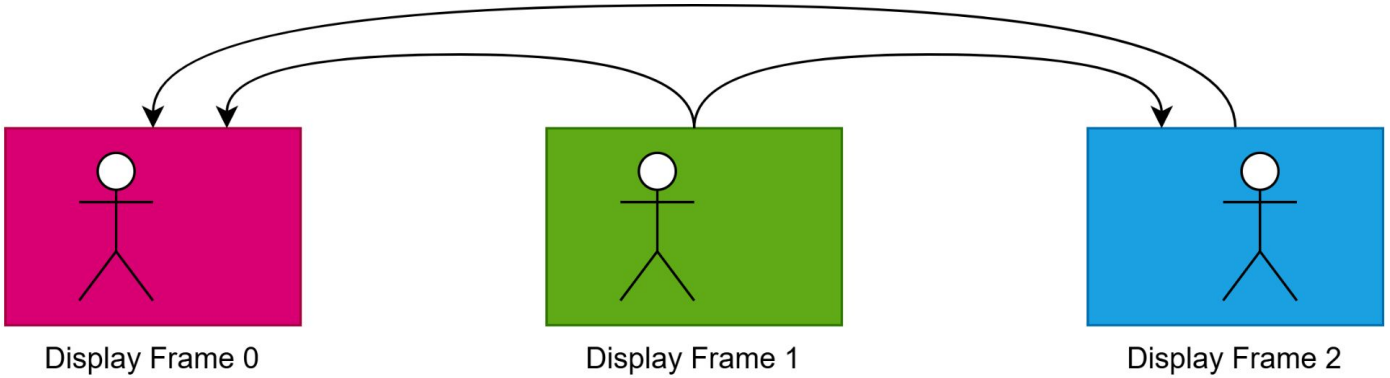
*Arrows indicate prediction direction, not data flow

AV1 Codec Overview - Prediction Modes

- **Within a frame, blocks may be:**
  - Intra (no pixel data from another frame)
  - Single Reference (pixel data from a single reference frame)
  - Unidirectional Compound (two reference frames - same group)
  - Bidirectional Compound (two reference frames - different group)

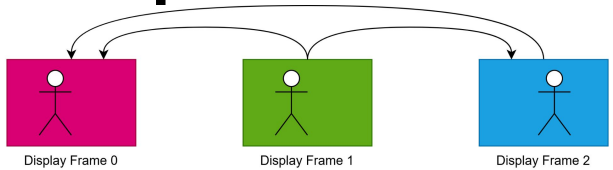
- **Block level selection of modes is encoder-dependent**
  - *frame_type* can restrict the modes to intra-only
  - *reference_select* indicates whether compound modes are enabled

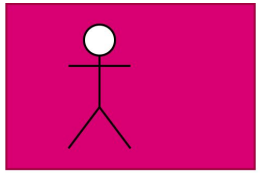
AV1 Codec Overview - Example sequence



In H.26X speak, this is an I, B, P, sequence.

AV1 Codec Overview - Example sequence



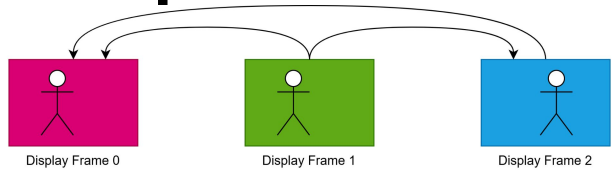


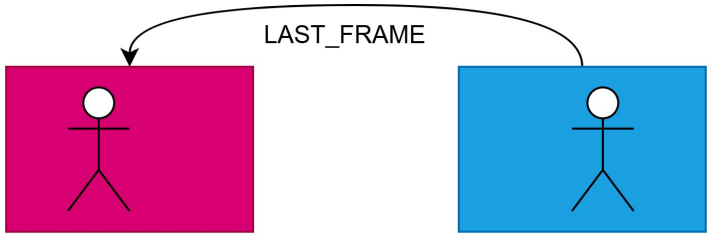
Frame 0, Display 0

`frame_type=KEY_FRAME`

VBI:
0, 0, 0, 0, 0, 0, 0, 0

AV1 Codec Overview - Example sequence





Frame 0, Display 0

`frame_type=KEY_FRAME`

Frame 1

`frame_type=INTER`

`show_frame=0`

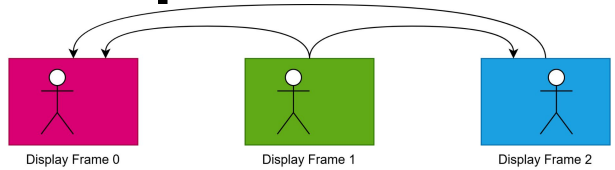
`refresh_frame_flags=0x1`

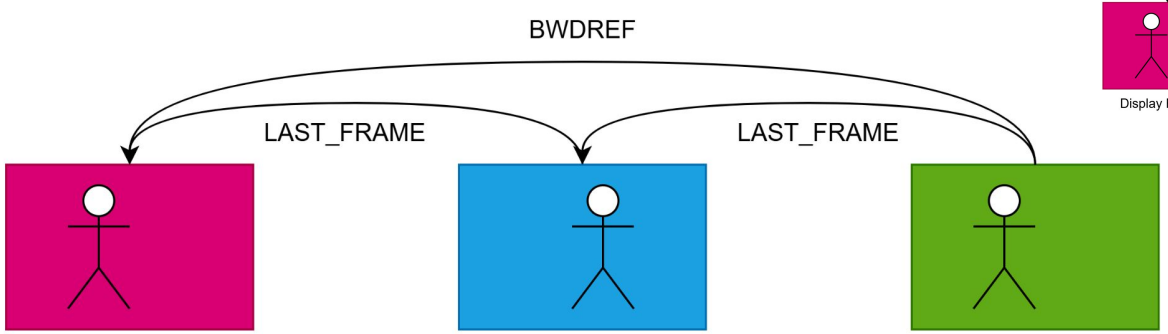
`ref_frame_idx[LAST] = 0`

VBI:
0, 0, 0, 0, 0, 0, 0, 0

VBI:
1, 0, 0, 0, 0, 0, 0, 0

AV1 Codec Overview - Example sequence





Frame 0, Display 0

`frame_type=KEY_FRAME`

VBI:
0, 0, 0, 0, 0, 0, 0, 0

Frame 1

`frame_type=INTER`

`show_frame=0`

`refresh_frame_flags=0x1`

`ref_frame_idx[LAST] = 0`

VBI:
1, 0, 0, 0, 0, 0, 0, 0

Frame 2, Display 1

`frame_type=INTER`

`show_frame=1`

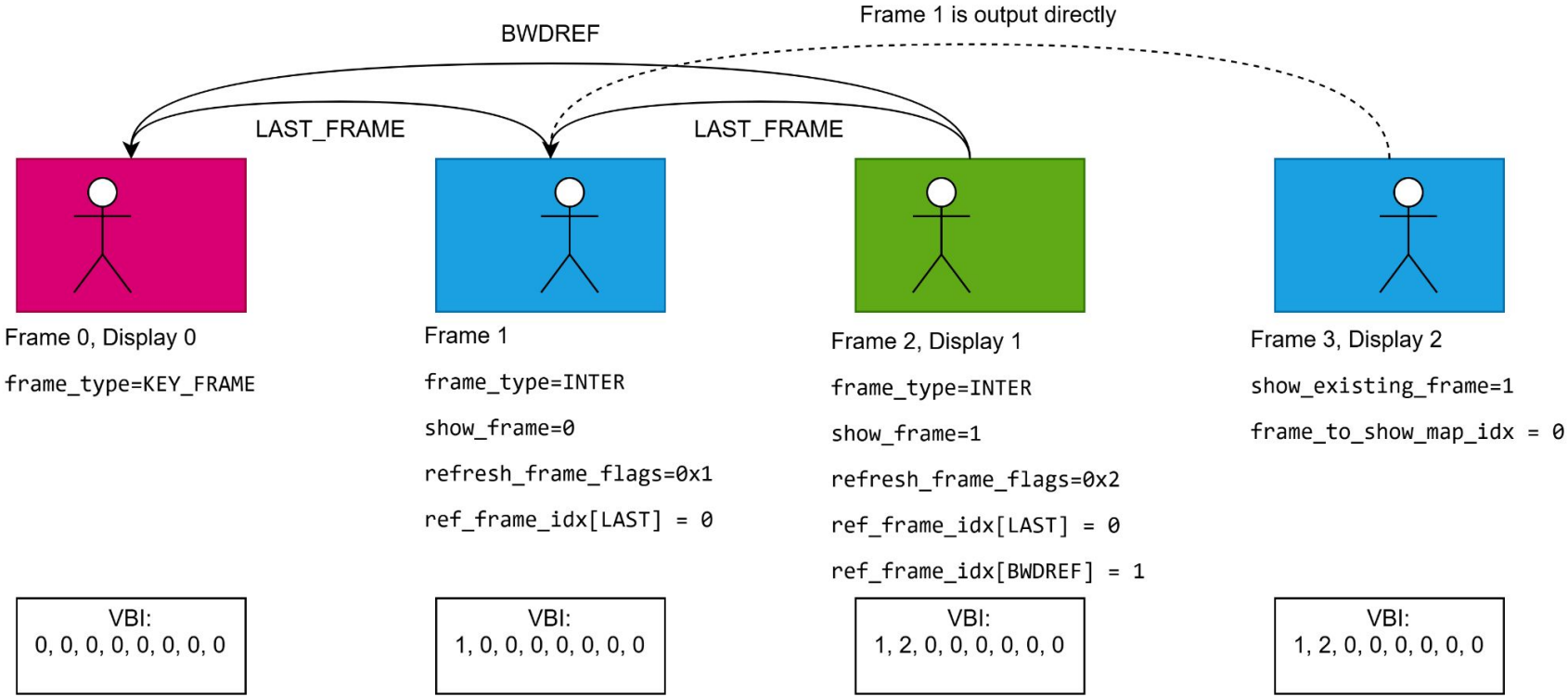
`refresh_frame_flags=0x2`

`ref_frame_idx[LAST] = 0`

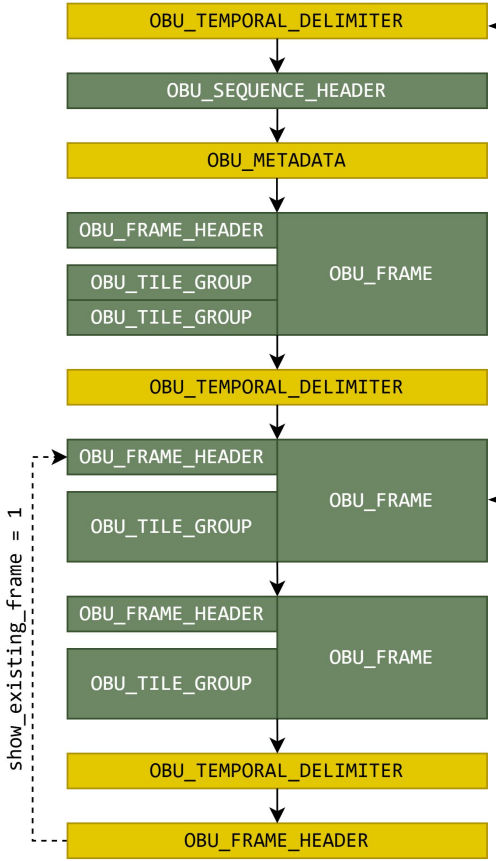
`ref_frame_idx[BWDREF] = 1`

VBI:
1, 2, 0, 0, 0, 0, 0, 0

AV1 Codec Overview - Example sequence



Vulkan AV1 Encode Bitstream Output



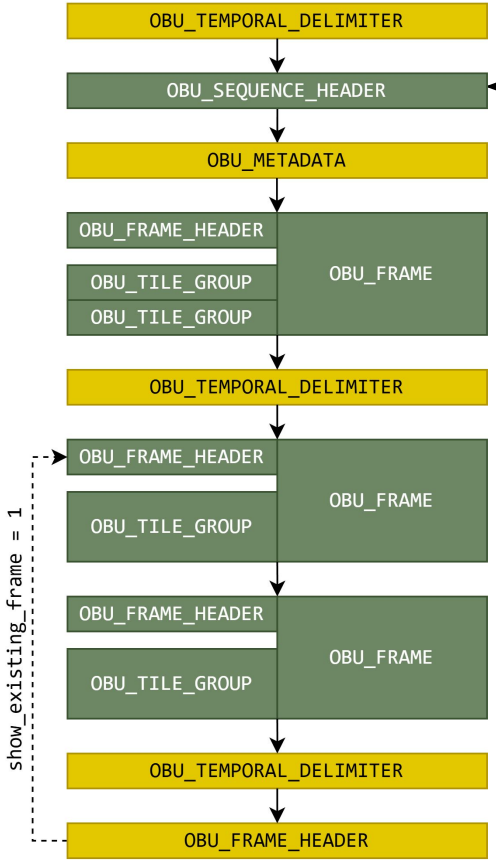
Signals the beginning of a new temporal unit.

Coded by application.

Each temporal unit has one shown frame.

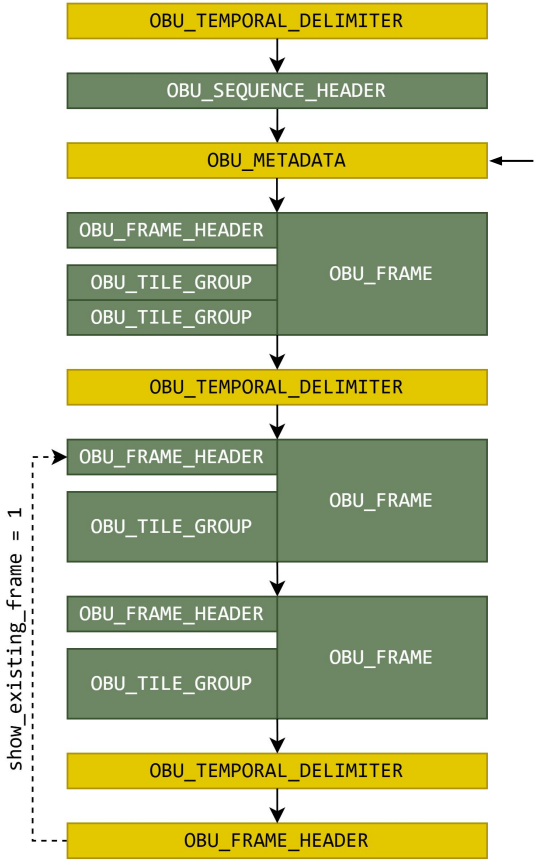
This frame header has show_frame=0.

Vulkan AV1 Encode Bitstream Output



The result of [vkGetEncodedVideoSessionParametersKHR]()

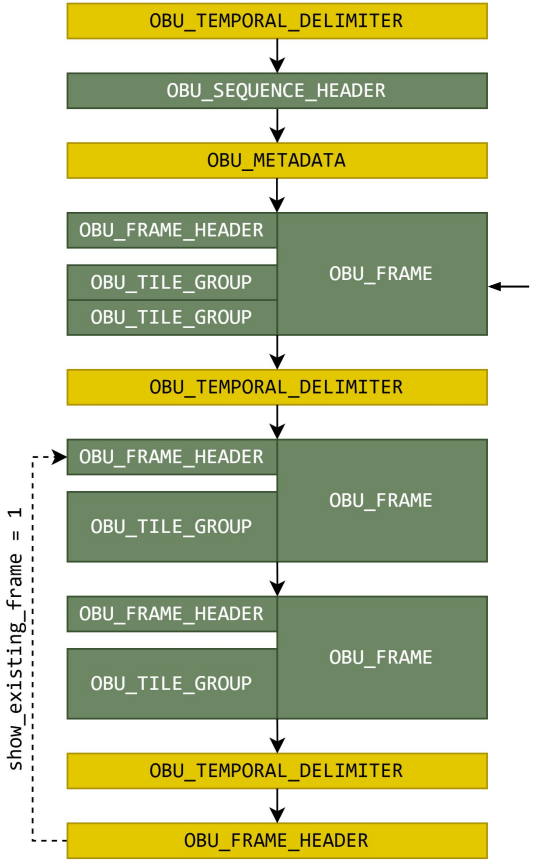
Vulkan AV1 Encode Bitstream Output



Optional metadata (i.e. HDR).

Coded by application.

Vulkan AV1 Encode Bitstream Output

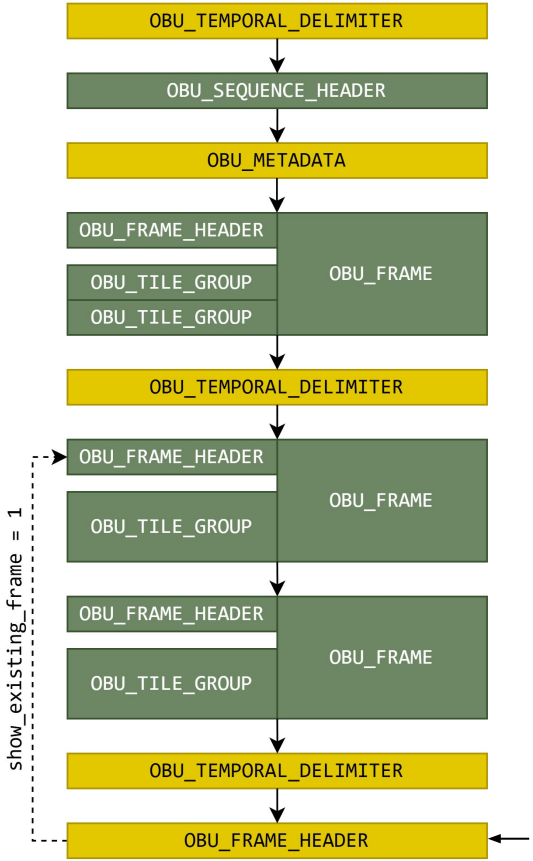


The bitstream output from an AV1 encode operation:

- A single *OBU_FRAME*
- *OBU_FRAME_HEADER* + *OBU_TILE_GROUP*(s)

Implementations are allowed to output either.

Vulkan AV1 Encode Bitstream Output



Frames with show_existing_frame=1 are coded by the application.

AV1 Encode Parameters - References

- **Vulkan DPB slot indices are associated with unique pictures**
  - Contrary to AV1 VBI slot indices
  - VBI management explicitly performed through Std parameters
  - Both Vulkan DPB slots and AV1 VBI need to be managed

- **VkVideoEncodeAV1PictureInfoKHR::referenceNameSlotIndices**
  - Array mapping reference names to Vulkan slot indices
  - Multiple reference names can point to the same slot index
  - Entries are allowed to be negative (unused)

- **VkVideoEncodeAV1PictureInfoKHR::primaryReferenceCdfOnly**
  - boolean indicating that the specified reference at *referenceNameSlotIndices[primary_ref_frame]* will only be used for CDF (and misc. data), and that no pixel data will be used

AV1 Encode Parameters

- VkVideoEncodeAV1PictureInfoKHR::predictionMode
  - INTRA_ONLY, SINGLE_REFERENCE, UNIDIRECTIONAL, BIDIRECTIONAL
  - Specifies which prediction modes the implementation should consider
  - Each mode has requirements for active *referenceNameSlotIndices*

  - Modes can change on the block-level
  - This is a cap on which mode is available, i.e.:
    - bidirectional means all modes are considered
    - unidirectional means that intra, single ref, unidirectional are considered

AV1 Encode Parameters

- VkVideoEncodeAV1PictureInfoKHR::rateControlGroup
  - Indicates directions of the reference frames (in display order)
    - Intra (no references)
    - Predictive (only previous frames)
    - Bipredictive (both previous and subsequent frames)

    } Mimics H.264/H.265 picture types
  - Independent from *predictionMode*, since AV1 does not have restrictions on reference directions
  - Will only be used for implementation defined rate control purposes

AV1 Encode Overrides

- In VK_KHR_video_encode_h26{4,5}, **we only had 2 guaranteed parameters:**
  - *primary_pic_type*/*pic_type* and *slice_type* (prior to VK_KHR_video_maintenance2)
- **For AV1 encode, we start with a much richer set**
  - We guaranteed many structures/parameters that are irrelevant to HW
    - i.e. StdVideoEncodeAV1DecoderModelInfo, OperatingPointInfo
  - Guarantees for *ref_frame_idx*, and related reference frame syntax to allow for predictable explicit management of VBI

- **Parameters specified in *StdVideoEncodeAV1PictureInfo::pTileInfo* are not guaranteed**
  - It's expected that drivers will try to follow where possible

AV1 Encode Overrides

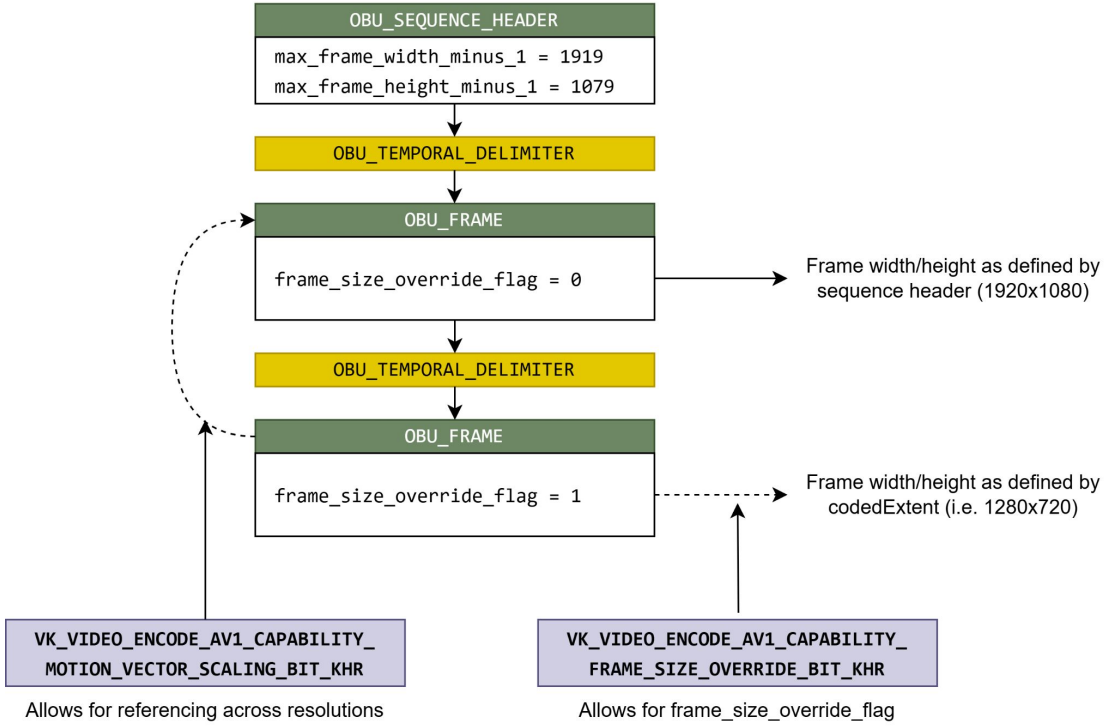
- **The effective output frame size may be overridden**
  - AV1 can only crop to arbitrary resolutions from an 8x8 alignment, but some HW have larger alignment requirements
  - Associated capability at VkVideoEncodeAV1CapabilitiesKHR::codedPictureAlignment
  - codedPictureAlignment = {16, 16}, codedExtent = { 1920, 1080 }
    - Output: 1920x1088
- **Use a side-channel (i.e. container cropping) to do cropping if necessary**
- **See Section 5.9 of the proposal for more details**

AV1 Encode Capabilities

- **Listed in VkVideoEncodeAV1CapabilitiesKHR**
- **For each prediction mode, we indicate:**
  - The maximum number of reference frames supported
  - Example: maxSingleReferenceCount
  - Which reference names those references can be bound to
  - Example: singleReferenceNameMask
  - For compound modes, further caps for each group are indicated:
    - maxUnidirectionalCompoundGroup1ReferenceCount
    - maxBidirectionalCompoundGroup1ReferenceCount
    - maxBidirectionalCompoundGroup2ReferenceCount

AV1 Encode Capabilities

- Frame size changes within a sequence



AV1 Encode Capabilities

- **CDF Only Reference**
  - VkVideoEncodeAV1PictureInfoKHR::primaryReferenceCdfOnly is only available with

    VK_VIDEO_ENCODE_AV1_CAPABILITY_PRIMARY_REFERENCE_CDF_ONLY_BIT

- **And more…**

Quantization Maps

Overview

- VK_KHR_video_encode_quantization_map: **finer control over picture quality**

- **Additional quality control on top of:**
  - Sequence-level quality control (rate control parameters, quality level)
  - Picture-level quality control

- **Specify one value per picture region to control quality**
  - Picture region: square/rectangle block of pixels
  - Specified value can be either direct or indirect control

Use Case

Input image

-4	-7	-6	-1	-6	-5	-10	-10	-12	-12	-12
-7	-4	-9	-6	-7	-4	-8	-12	-12	-12	-12
-5	-5	-5	-8	-3	-5	-5	-12	-12	-12	-12
-7	-12	-6	-8	-6	-4	-5	-6	2	2	2
-6	-7	-11	-11	-11	-8	-3	4	4	4	4
-9	-10	-11	-9	-12	-11	2	4	4	3	3
-12	-12	-12	-12	-12	1	4	4	2	4	4
-11	-1	-2	-6	0	4	3	3	2	2	2
3	-2	-2	-2	3	3	3	3	3	3	3

Quantization map

Output image

Quantization Map Types

- **Quantization delta map**
  - Direct control: each element is a delta quantizer

-5	-6	-6	-6	-5
-6	-6	-6	-6	-6
-4	-2	-2	-3	-4
2	0	0	0	2
3	4	4	4	3

- **Emphasis map**
  - Indirect control: each element is a normalized value
  - `VK_FORMAT_R8_UNORM` is a guaranteed format for this type

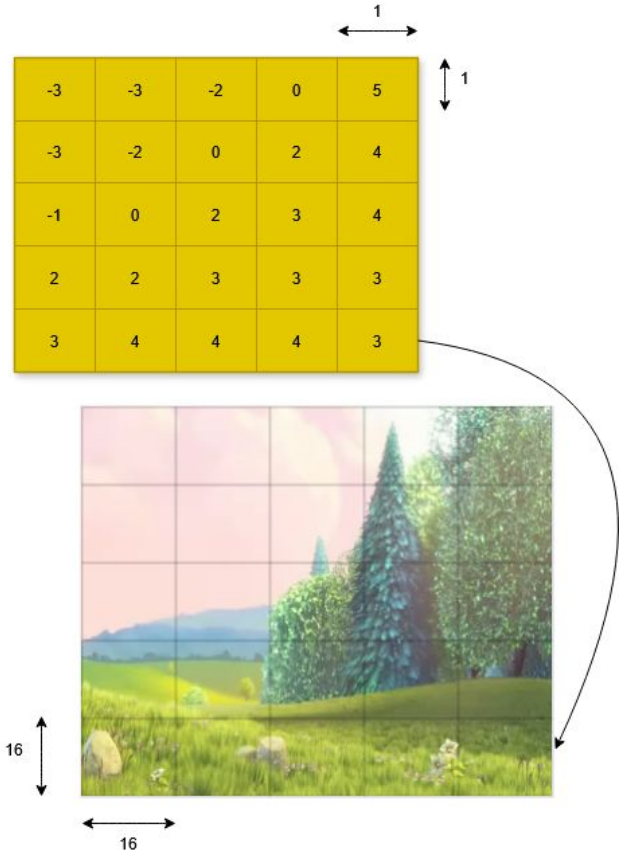
0.75	0.8	0.8	0.8	0.75
0.8	0.8	0.8	0.8	0.8
0.6	0.4	0.4	0.5	0.6
0.2	0.25	0.25	0.25	0.2
0.1	0	0	0	0.1

Note: no correlation between the values in the two maps

Key Concepts

- **Quantization map image**
  - 2D image
  - No multisampling
  - No mip levels
  - Single channel formats

- **Quantization map texel**
  - A texel in the quantization map image
  - Corresponds to a block of size `quantizationMapTexelSize` in the input image (see Format Properties)

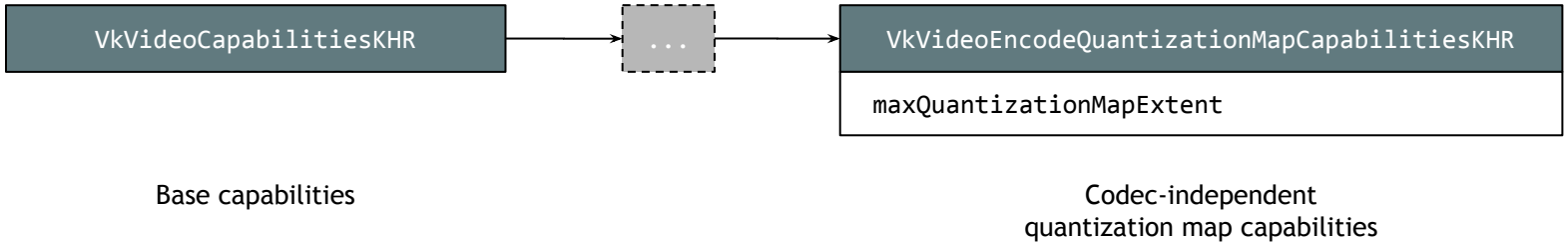


quantizationMapTexelSize = { 16, 16 }

Capabilities

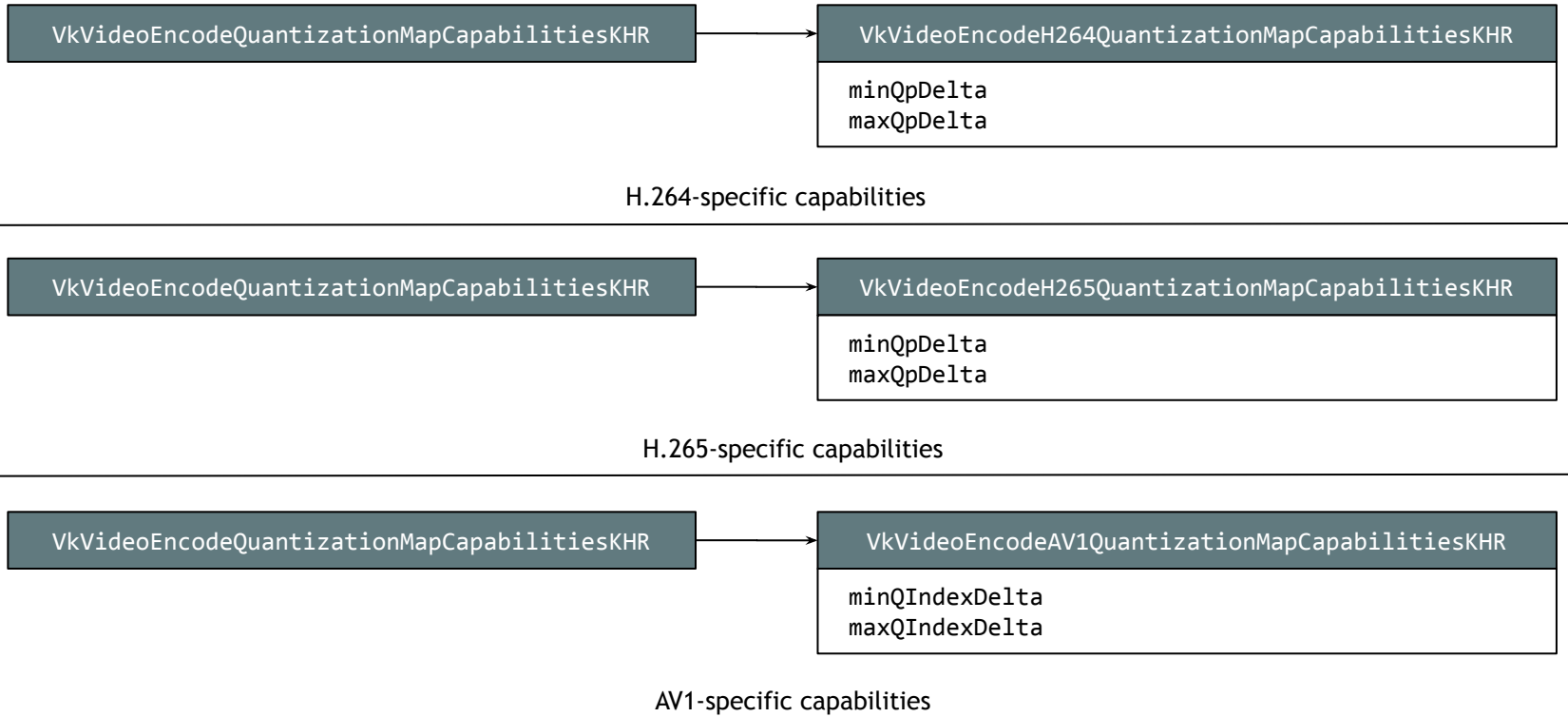
- Supported quantization map types
  - VK_VIDEO_ENCODE_CAPABILITY_QUANTIZATION_DELTA_MAP_BIT_KHR
  - VK_VIDEO_ENCODE_CAPABILITY_EMPHASIS_MAP_BIT_KHR

- Quantization map image capabilities

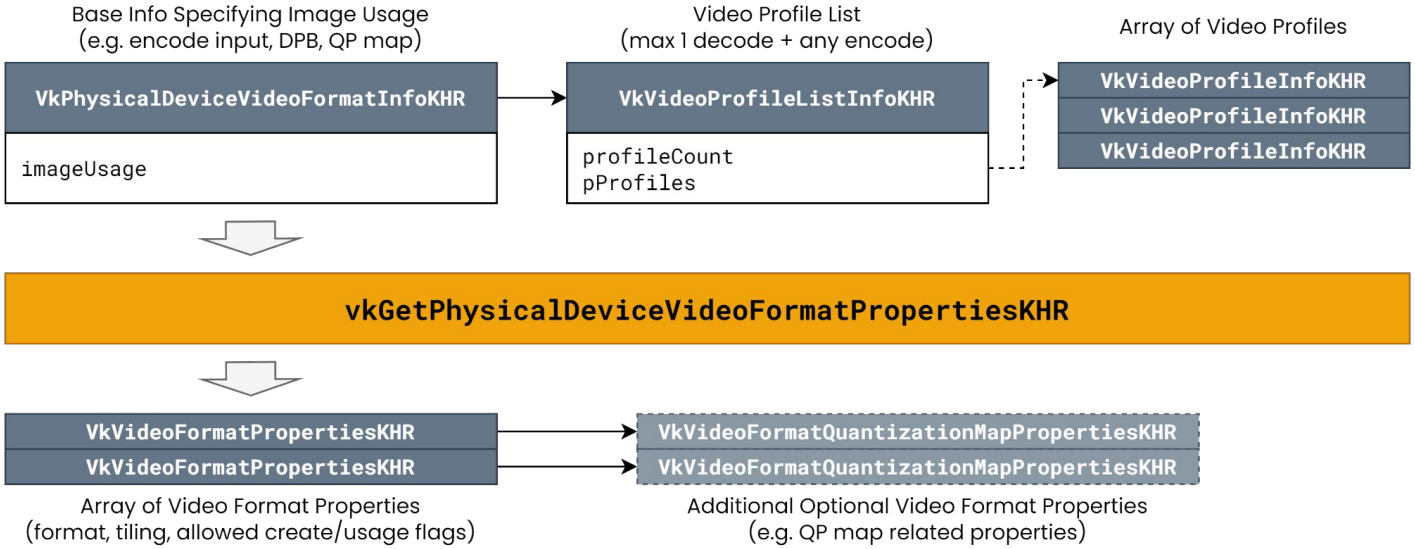


Capabilities

- Codec-specific capabilities for quantization delta maps

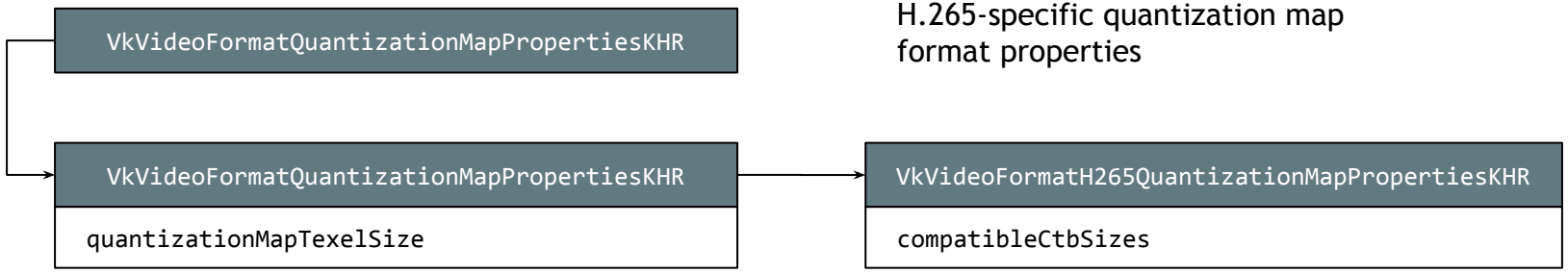


Format Properties

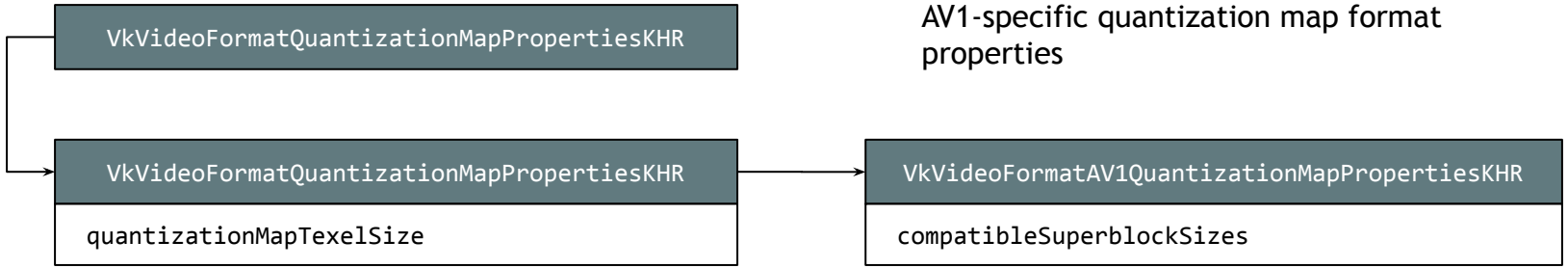


- VkVideoFormatQuantizationMapPropertiesKHR
  - quantizationMapTexelSize: the size of the rectangular block of pixels corresponding to an element in the quantization map

Format Properties



H.265-specific quantization map format properties



AV1-specific quantization map format properties

API Usage

- **Query capabilities**
  - Chain codec-specific quantization map structures
  - On return, check VkVideoEncodeCapabilitiesKHR::flags

- **Query format properties**
  - Inputs: the required imageUsage and profile list
  - Chain structures for quantization map format properties
  - On return, check chained output structures for suitable format(s) and corresponding properties

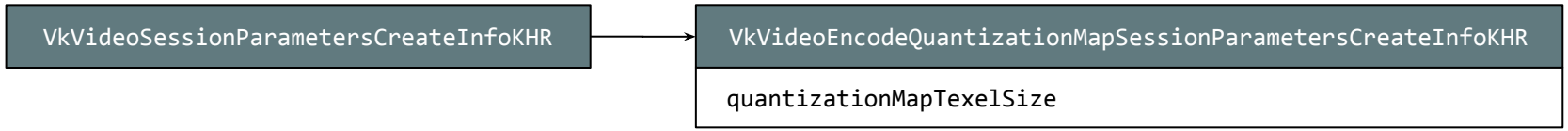
API Usage

- **Create quantization image(s) and corresponding image view(s)**
  - Inputs:
    - imageUsage and profile list from format properties query
    - Format returned in the format properties query

- **Create video session**
  - Specify flag corresponding to quantization map type

API Usage

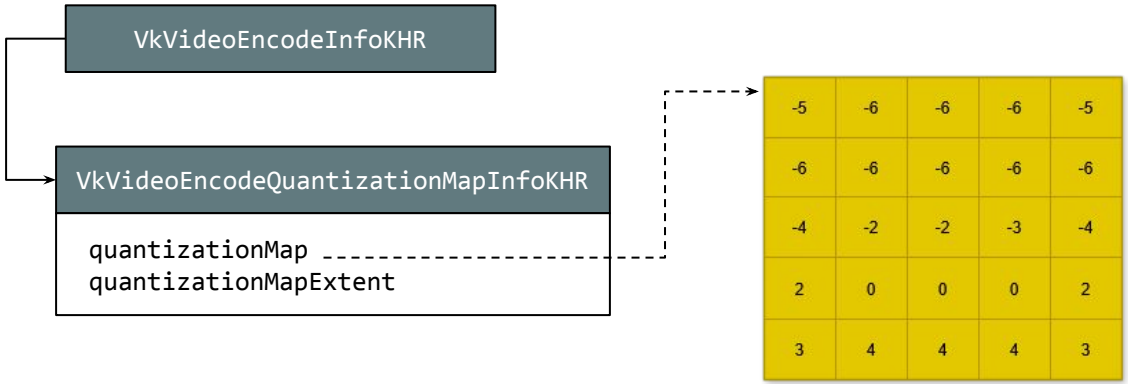
- **Create video session parameters**
  - Set the flag to use this object along with quantization maps
  - Specify the quantization map texel size to aid in parameter overrides



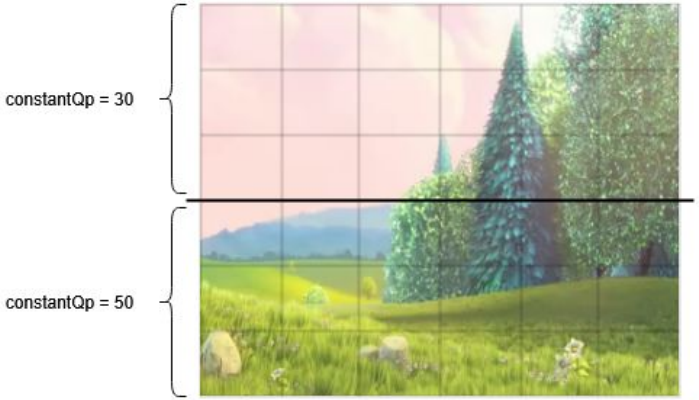
API Usage

- **Fill in the quantization map**
  - Populate from host, fill using compute shader etc.

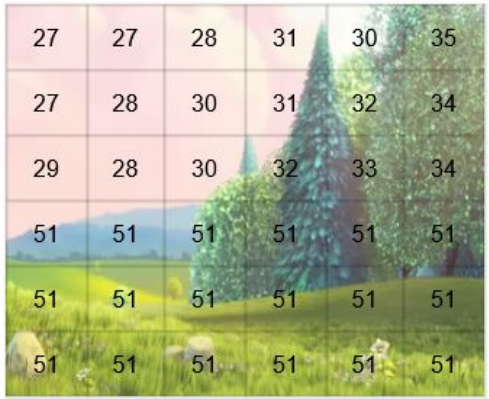
- **Encode with quantization map**
  - Specify quantization map along with the other encode parameters and resources



Quantization Maps



Input frame (2 slices)



-3	-3	-2	1	0	5
-3	-2	0	1	2	4
-1	-2	0	2	3	4
1	1	1	3	3	3
2	2	2	3	4	3
3	4	4	3	4	3

Quantization delta map

Thank you!

Questions?