Advanced Micro Devices

Advanced Media Framework – HEVC Video Encoder

Programming Guide

Disclaimer

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information.

Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

AMD, the AMD Arrow logo, ATI Radeon™, CrossFireX™, LiquidVR™, TrueAudio™ and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Windows™, Visual Studio and DirectX are trademark of Microsoft Corp.

Copyright Notice

Notice Regarding Standards. AMD does not provide a license or sublicense to any Intellectual Property Rights relating to any standards, including but not limited to any audio and/or video codec technologies such as MPEG-2, MPEG-4; AVC/H.264; HEVC/H.265; AAC decode/FFMPEG; AAC encode/FFMPEG; VC-1; and MP3 (collectively, the "Media Technologies"). For clarity, you will pay any royalties due for such third party technologies, which may include the Media Technologies that are owed as a result of AMD providing the Software to you.

MIT license

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

1 Introduction

1.1 Scope

This document provides a complete description of the AMD Advanced Media Framework (AMF) Video Encoder Component. This component exposes the AMD Video Compression Engine, which provides hardware accelerated HEVC video encoding functionality.

Figure 1 provides a system overview of the AMF Video Encoder Component.

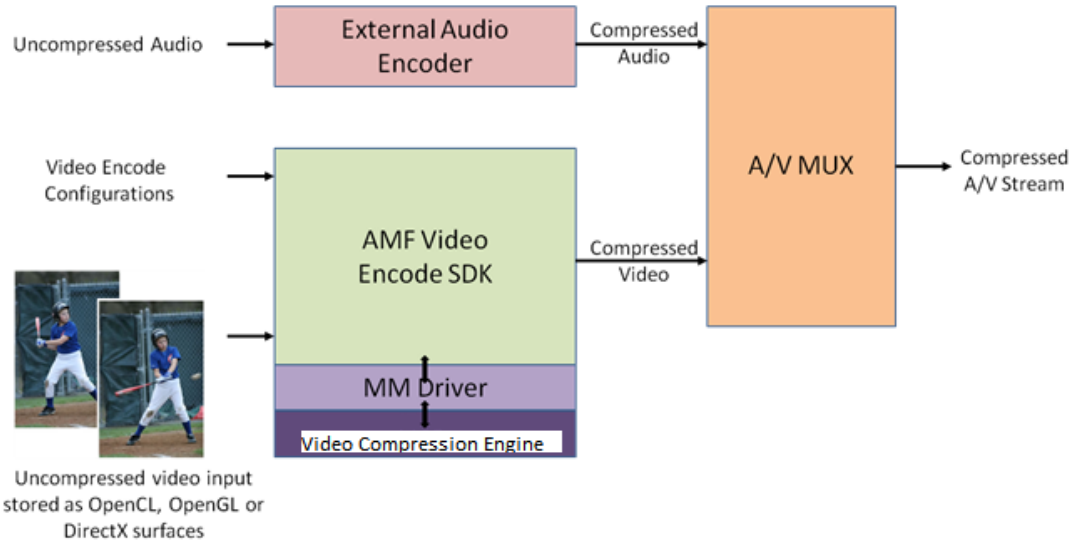

Figure 1 — System overview of the AMF Video Encode SDK

The AMF Video Encoder Component compresses RAW uncompressed video to an HEVC elementary bitstream.

The component does not provide a mechanism to handle audio compression, or stream multiplexing.

The component provides four different sets of pre-defined usages, which provide a convenient way for developers to configure the encoder to match the intended application use case. Advanced developers can also adjust encoding parameters to tailor the behavior to their specific application requirements.

1.2 Pre-defined Encoder Usages

The following table provides a brief overview of the encoding usage modes that have been defined:

Usage Mode	Intended use-cases	Comments
Transcoding	Transcoding, video editing	Favor compression efficiency and throughput over latency.
Ultra-low latency	Video game streaming	Optimize for extremely low latency use cases (e.g. cap the number of bits per frame), to enable high-interactivity applications.
Low Latency	Video collaboration, remote desktop	Optimize for low latency scenarios, but allow occasional bitrate overshoots to preserve quality.
Webcam	Video conferencing	Optimize for a low-latency video conferencing scenario.

2 AMF Video Encoder UVD/VCN-HEVC Component

The AMF Video Encoder HEVC component provides hardware accelerated HEVC encoding using AMD's IP.

To instantiate the AMF Video Encoder component, call the *AMFFactory::CreateComponent* method passing *AMFVideoEncoderHW_HEVC* component IDs defined in the */include/components/VideoEncoderHEVC.h* header.

2.1 Input Submission and Output Retrieval

The AMF Video Encoder component accepts *AMFSurface* objects as input and produces *AMFBuffer* objects for output.

2.2 Encode Parameters

Annex A provides the detailed description of encoding parameters (i.e., encoder properties) exposed by the Video Encoder HEVC component for the following four usages:

- Transcoding mode,
- Ultra-low latency mode,
- Low Latency mode, and
- Webcam mode.

All properties are accessed using the *AMFPropertyStorage* interface of the Encoder object.

2.2.1 Static Properties

Static properties (e.g., profile, tier, level, usage) must be defined before the `Init()` function is called, and will apply until the end of the encoding session.

2.2.2 Dynamic Properties

All dynamic properties have default values. Several properties can be changed subsequently and these changes will be flushed to encoder only before the next `Submit()` call.

2.2.3 Frame Per-Submission Properties

Per submission properties are applied on a per frame basis. They can be set optionally to force a certain behavior (e.g., force frame type to IDR) by updating the properties of the `AMFSurface` object that is passed through the `AMFComponent::Submit()` call.

3 Sample Applications

The AMF Encoder Sample application show how to setup and use the AMF Video Encoder HEVC Component to encode video frames that are loaded from disk or rendered by the DirectX 3D engine.

3.1 List of Parameters

Sample applications support almost all visible encoder parameters (except PictureStructure, EndOfSequence, EndOfStream) and few additional parameters.

Additional parameters of TranscodeHW application:

Category	Name	Values	Description
Miscellaneous parameters	Frames	Number of frames to be encoded	Number of frames to render
	Codec	HEVC or H265	Specify codec type
	Input	File name, relative or absolute path	Input file with frames (AVC or HEVC)
	Output	File name, relative or absolute path	Output HEVC file for encoded data
	Engine	DX9, DX11	Specify Engine type
	AdapterID	Number	Index of GPU adapter

Additional parameters of VCEEncoderD3D application:

Category	Name	Values	Description
Miscellaneous parameters	Frames	Number of frames to be encoded	Number of frames to be encoded
	Codec	HEVC or H265	Specify codec type
	Output	File name, relative or absolute path	Output HEVC file for encoded data
	Render	DX9, DX9EX, DX11, OpenGL, OpenCL, etc.	Specify render type
	AdapterID	Number	Index of GPU adapter
	Windowmode	Flag (without any values)	Shows rendering window for D3D sample application
	FullScreen	Flag (without any values)	Enables full screen
	QueryInstanceCount	Flag (without any values)	If the flag is set, the number of independent VCE instances will be queried and printed
	UseInstance	0 to (number of instances -1)	If there are more than one VCE instances, allow to force which instance to use.

3.2 Command line example

3.2.1 Transcoding application (TranscodeHW.exe)

```
TranscodeHW.exe -input input.h264 -output out.h265 –codec HEVC -width 1280 -height 720 -Usage transcoding -RateControlMethod cbr -TargetBitrate 100000
```

This command transcodes H264 elementary stream to H.265 video. Encoder is created with “Transcoding” usage.

3.2.2 D3D application (VCEEncoderD3D.exe)

```
VCEEncoderD3D.exe -output VideoSample_1024x768.h265 –codec HEVC -width 1024 -height 768 -Usage transcoding -RateControlMethod cbr -TargetBitrate 500000 -frames 400
```

This command encodes 400 frames through D3D renderer and creates an output file with the encoded data. Encoder is created with “Transcoding” usage. Initial configuration sets bitrate to a value of 500kbits/sec.

Annex A: Encoding & frame parameters description

Table A-1. The description of encoder's parameters.

Category	Name	Values	Description
Encoder static parameters	AMF_VIDEO_ENCODER_HEVC_USAGE	Transcoding, UltraLowLatency, LowLatency, Webcam	Selects the AMF usage (see Section 1.2)
	AMF_VIDEO_ENCODER_HEVC_PROFILE	Main	Selects the HEVC profile
	AMF_VIDEO_ENCODER_HEVC_TIER	Main, High	Selects the HEVC tier
	AMF_VIDEO_ENCODER_HEVC_PROFILE_LEVEL	1, 2, 2.1, 3, 3.1, 4, 4.1, 5, 5.1, 5.2, 6, 6.1, 6.2	Selects the HEVC ProfileLevel
	AMF_VIDEO_ENCODER_HEVC_MAX_LTR_FRAMES	0 … 16	The number of long-term references controlled by the user. Remarks: • When == 0, the encoder may or may not use LTRs during encoding. • When > 0, the user has control over all LTR. • With user control of LTR, Intra-refresh features are not supported. • The actual maximum number of LTRs allowed depends on H.265 (HEVC) Annex A Table A-4 Level limits, which defines dependencies between the H.265 Level number, encoding resolution, and DPB size. The DPB size limit impacts the maximum number of LTR allowed.
	AMF_VIDEO_ENCODER_HEVC_MAX_NUM_REFRAMES	0 … 16	Maximum number of reference frames
Encoder resolution parameters	AMF_VIDEO_ENCODER_HEVC_FRAMESIZE	Width: 192 – 4096 Height: 128 – 2176	Frame width/Height in pixels, maximum value is hardware-specific, should be queried through *AMFCaps*
	AMF_VIDEO_ENCODER_HEVC_ASPECT_RATIO	Default 1:1	Pixel aspect ratio
Encoder rate-control parameters	AMF_VIDEO_ENCODER_HEVC_TARGET_BITRATE	>0	Sets the target bitrate, bit/s based on use case
	AMF_VIDEO_ENCODER_HEVC_PEAK_BITRATE	>= TargetBitrate	Sets the peak bitrate

Category	Name	Values	Description
	AMF_VIDEO_ENCODER_HEVC_RATE_CONTROL_METHOD	CQP, CBR, VBR, VBR_LAT	Selects the rate control method: • CQP – Constrained QP, • CBR - Constant Bitrate, • VBR - Peak Constrained VBR, • VBR_LAT - Latency Constrained VBR
	AMF_VIDEO_ENCODER_HEVC_RATE_CONTROL_SKIP_FRAME_ENABLE	True/False	Enables skip frame for rate control
	AMF_VIDEO_ENCODER_HEVC_MIN_QP_I	0 – 51	Sets the minimum QP for I frame
	AMF_VIDEO_ENCODER_HEVC_MAX_QP_I	0 – 51	Sets the maximum QP for I frame
	AMF_VIDEO_ENCODER_HEVC_MIN_QP_P	0 – 51	Sets the minimum QP for P frame
	AMF_VIDEO_ENCODER_HEVC_MAX_QP_P	0 – 51	Sets the maximum QP for P frame
	AMF_VIDEO_ENCODER_HEVC_QP_I	0 – 51	Sets the constant QP for I-pictures. Remarks: Only available for CQP rate control method.
	AMF_VIDEO_ENCODER_HEVC_QP_P	0 – 51	Sets the constant QP for P-pictures. Remarks: Only available for CQP rate control method.
	AMF_VIDEO_ENCODER_HEVC_FRAMERATE	1*FrameRateDen … 60* FrameRateDen	Frame rate numerator/denominator
	AMF_VIDEO_ENCODER_HEVC_VBV_BUFFER_SIZE	>0	Sets the VBV buffer size in bits based on use case
	AMF_VIDEO_ENCODER_HEVC_INITIAL_VBV_BUFFER_FULLNESS	0 - 64	Sets the initial VBV buffer fullness
	AMF_VIDEO_ENCODER_HEVC_ENFORCE_HRD	True/False	Disables/enables constraints on QP variation within a picture to meet HRD requirement(s)
	AMF_VIDEO_ENCODER_HEVC_RATE_CONTROL_PREANALYSIS_ENABLE	True/False	Pre-analysis assisted rate control
	AMF_VIDEO_ENCODER_HEVC_ENABLE_VBAQ	True/False	By default, disable VBAQ
	AMF_VIDEO_ENCODER_HEVC_FILLER_DATA_ENABLE	True/False	Enable filler data for CBR usage
Encoder picture-control parameters	AMF_VIDEO_ENCODER_HEVC_MAX_AU_SIZE	0 – 100 000 000 bits	Maximum AU size in bits
	AMF_VIDEO_ENCODER_HEVC_HEADER_INSERTION_MODE	NONE, GOP aligned, IDR aligned	Sets the headers insertion mode
	AMF_VIDEO_ENCODER_HEVC_GOP_SIZE	0 ... 1000	The period to insert IDR/CRA in fixed size mode. 0 means only insert the first IDR/CRA (infinite GOP size)

Category	Name	Values	Description
	AMF_VIDEO_ENCODER_HEVC_NUM_GOPS_PER_IDR	1 – 65535	Determines the frequency to insert IDR as start of a GOP. 0 means no IDR will be inserted except for the first picture in the sequence.
	AMF_VIDEO_ENCODER_HEVC_DE_BLOCKING_FILTER_DISABLE	True/False	Disable/enable the de-blocking filter
	AMF_VIDEO_ENCODER_HEVC_SLICES_PER_FRAME	1 - #CTBs per frame	Sets the number of slices per frame
Encoder miscellaneous parameters	AMF_VIDEO_ENCODER_HEVC_QUALITY_PRESET	Balanced, Quality, Speed	Selects the quality preset
Encoder motion estimation parameters	AMF_VIDEO_ENCODER_HEVC_MOTION_HALF_PIXEL	True/False	Turns on/off half-pixel motion estimation
	AMF_VIDEO_ENCODER_HEVC_MOTION_QUARTERPIXEL	True/False	Turns on/off quarter-pixel motion estimation

Table A-2. The description of frame's and encoded data parameters.

Category	Name	Values	Description
Frame per-submission parameters	AMF_VIDEO_ENCODER_HEVC_INSERT_HEADER	True/False	Inserts SPS, PPS and VPS
	AMF_VIDEO_ENCODER_HEVC_INSERT_AUD	True/False	Inserts AUD
	AMF_VIDEO_ENCODER_HEVC_FORCE_PICTURE_TYPE	NONE, IDR, I, P	Forces the picture type
	AMF_VIDEO_ENCODER_HEVC_END_OF_SEQUENCE	True/False	End of sequence
	AMF_VIDEO_ENCODER_HEVC_MARK_CURRENT_WITH_LTR_INDEX	-1 … (MaxOfLTRFrames - 1)	If != -1, the current picture is coded as a long-term reference with the given index. Remarks: • When the user controls N LTRs (using the corresponding Create parameter), then the LTR Index the user can assign to a reference picture varies from 0 to N-1. By default, the encoder will "use up" available LTR Indices (i.e. assign them to references) even if the user does not request them to be used. • If the user submits multiple requests to mark current as LTR between base temporal layer pictures, then only the last request is applied.

Category	Name	Values	Description
	AMF_VIDEO_ENCODER_HEVC_FORCE_LTR_REFERENCE_BITFIELD	Bitfield (MaxOfLTRFrames (max possible 16 bits))	Force LTR Reference allowed bitfield. If == 0, the current picture should predict from the default reference. If != 0, the current picture should predict from one of the LTRs allowed by the bitfield (bit# = LTR Index#). Remarks: • E.g. if Bit#0 = 1, then the existing LTR with LTR Index = 0 may be used for reference. The bitfield may allow more than one LTR for reference, in which case the encoder is free to choose which one to use. This bitfield also disallows existing LTRs not enabled by it from current/future reference. • E.g. if Bit#1 = 0, and there is an existing reference with LTR Index = 1, then this LTR Index will not be used for reference until it is replaced with a newer reference with the same LTR Index.
Encoded data parameters	AMF_VIDEO_ENCODER_HEVC_OUTPUT_DATA_TYPE	I, P	Type of encoded data
	AMF_VIDEO_ENCODER_HEVC_OUTPUT_MARKED_LTR_INDEX	-1 … (MaxOfLTRFrames - 1)	Marked as LTR Index. If != -1, then this picture was coded as a long-term reference with this LTR Index.
	AMF_VIDEO_ENCODER_HEVC_OUTPUT_REFERENCED_LTR_INDEX_BITFIELD	Bitfield (MaxOfLTRFrames (max possible 16 bits))	Referenced LTR Index bitfield. If != 0, this picture was coded to reference long-term references. The enabled bits identify the LTR Indices of the referenced pictures (e.g. if Bit #0 = 1, then LTR Index 0 was used as a reference when coding this picture).

Table A-3. Default value of parameters.

Type	Name	Transcoding	Ultra low latency	Low latency	Webcam
Static Parameters (Set at creation time)	AMF_VIDEO_ENCODER_HEVC_PROFILE	Main	Main	Main	Main
	AMF_VIDEO_ENCODER_HEVC_PROFILE_LEVEL	6.2	6.2	6.2	6.2
	AMF_VIDEO_ENCODER_HEVC_TIER	Main	Main	Main	Main
	AMF_VIDEO_ENCODER_HEVC_MAX_LTR_FRAMES	0	0	0	0
Rate control	AMF_VIDEO_ENCODER_HEVC_RATE_CONTROL_METHOD	PEAK_CONSTRAINED_VBR	LATENCY_CONSTRAINED_VBR	PEAK_CONSTRAINED_VBR	PEAK_CONSTRAINED_VBR
	AMF_VIDEO_ENCODER_HEVC_FRAMERATE	30 fps	30 fps	30 fps	30 fps
	AMF_VIDEO_ENCODER_HEVC_VBV_BUFFER_SIZE	20 mbits	735 kbits	4 mbits	2 mbits
	AMF_VIDEO_ENCODER_HEVC_INITIAL_VBV_BUFFER_FULLNESS	64	64	64	64
	AMF_VIDEO_ENCODER_HEVC_RATE_CONTROL_PREANALYSIS_ENABLE	false	false	false	false
	AMF_VIDEO_ENCODER_HEVC_ENABLE_VBAQ	false	false	false	false
	AMF_VIDEO_ENCODER_HEVC_TARGET_BITRATE	20 mbps	20 mbps	20 mbps	20 mbps
	AMF_VIDEO_ENCODER_HEVC_PEAK_BITRATE	30 mbps	20 mbps	20 mbps	20 mbps
	AMF_VIDEO_ENCODER_HEVC_MIN_QP_I	18	22	22	22
	AMF_VIDEO_ENCODER_HEVC_MAX_QP_I	46	48	48	48
	AMF_VIDEO_ENCODER_HEVC_MIN_QP_P	18	22	22	22
	AMF_VIDEO_ENCODER_HEVC_MAX_QP_P	46	48	48	48
	AMF_VIDEO_ENCODER_HEVC_QP_I	26	26	26	26
	AMF_VIDEO_ENCODER_HEVC_QP_P	26	26	26	26
	AMF_VIDEO_ENCODER_HEVC_ENFORCE_HRD	false	true	false	false
	AMF_VIDEO_ENCODER_HEVC_MAX_AU_SIZE	0	0	0	0
	AMF_VIDEO_ENCODER_HEVC_FILLER_DATA_ENABLE	false	false	false	false
	AMF_VIDEO_ENCODER_HEVC_RATE_CONTROL_SKIP_FRAME_ENABLE	false	true	true	true
Picture Control	AMF_VIDEO_ENCODER_HEVC_HEADER_INSERTION_MODE	0	0	0	0
	AMF_VIDEO_ENCODER_HEVC_GOP_SIZE	30	300	300	30
	AMF_VIDEO_ENCODER_HEVC_NUM_GOPS_PER_IDR	1	1	1	1
	AMF_VIDEO_ENCODER_HEVC_DE_BLOCKING_FILTER_DISABLE	false	false	false	false
	AMF_VIDEO_ENCODER_HEVC_SLICES_PER_FRAME	1	1	1	1
	AMF_VIDEO_ENCODER_HEVC_QUALITY_PRESET	Balanced	Speed	Speed	Speed
Motion estimation	AMF_VIDEO_ENCODER_HEVC_MOTION_HALF_PIXEL	1	1	1	1
	AMF_VIDEO_ENCODER_HEVC_MOTION_QUARTERPIXEL	1	1	1	1
Per-submission parameters	AMF_VIDEO_ENCODER_HEVC_INSERT_HEADER	0	0	0	0
	AMF_VIDEO_ENCODER_HEVC_FORCE_PICTURE_TYPE	0	0	0	0
	AMF_VIDEO_ENCODER_HEVC_INSERT_AUD	false	false	false	false
	AMF_VIDEO_ENCODER_HEVC_END_OF_SEQUENCE	false	false	false	false
	AMF_VIDEO_ENCODER_HEVC_MARK_CURRENT_WITH_LTR_INDEX	-1	-1	-1	-1
	AMF_VIDEO_ENCODER_HEVC_FORCE_LTR_REFERENCE_BITFIELD	0x0	0x0	0x0	0x0