Advanced Micro Devices

Advanced Media Framework – h.264 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 (VCE), which provides hardware accelerated H.264 video encoding functionality.

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

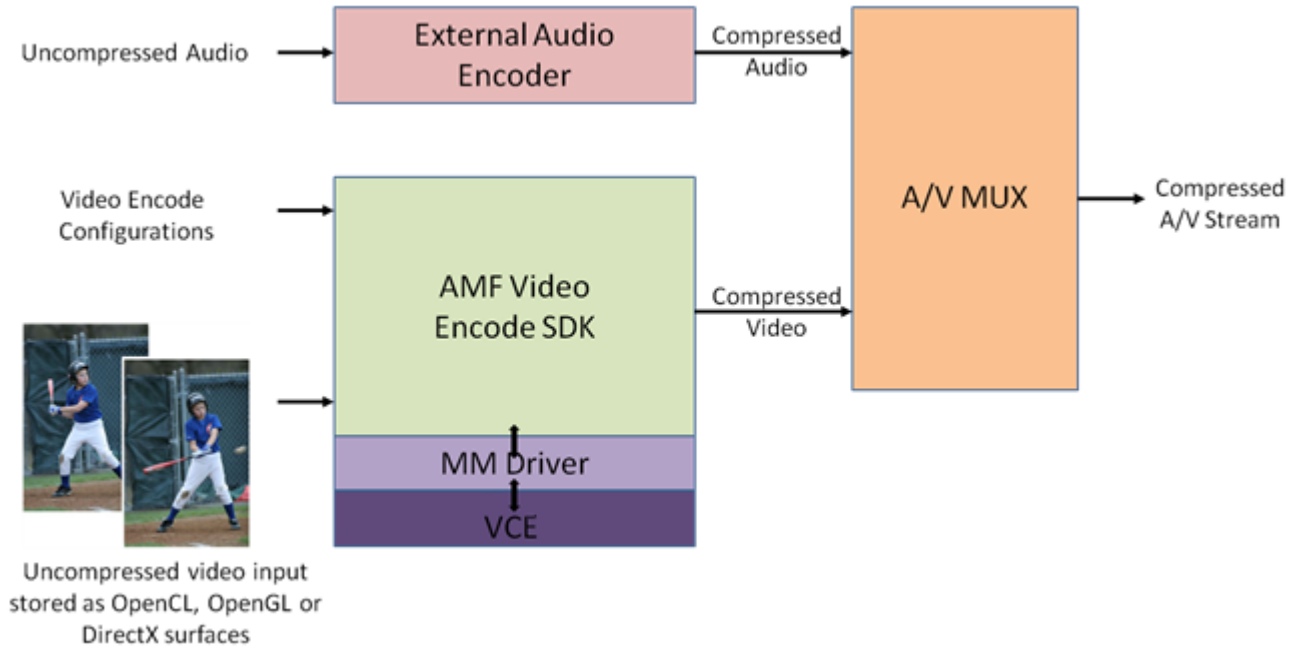

Figure 1 — System overview of the AMF Video Encode SDK

The AMF Video Encoder Component compresses RAW uncompressed video to an H.264 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, with scalable video coding (SVC) support.

2 AMF Video Encoder VCE-AVC Component

The AMF Video Encoder VCE-AVC component provides hardware accelerated AVC/SVC encoding using AMD's VCE.

To instantiate the AMF Video Encoder component, call the *AMFFactory::CreateComponent* method passing *AMFVideoEncoderVCE_AVC* or *AMFVideoEncoderVCE_SVC* component IDs defined in the *public/include/components/VideoEncoderVCE.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.

In the Transcoding mode the encoder needs to accept at least 3 input frames before any output is produced. In low latency modes output becomes available as soon as the first submitted frame is encoded.

2.2 Encode Parameters

Annex A provides the detailed description of encoding parameters (i.e., encoder properties) exposed by the Video Encoder VCE-AVC 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, 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.

2.2.4 SVC Properties

For define frame-rate SVC parameters per layer the next format must be used:

```
TL<Temporal_Layer_Number>.QL<Quality_Layer_Number>.<Parameter_name>
```

For example, to configure "Target bitrate" for temporal layer 2 and quality layer 1 the next parameter name uses:

```
"TL2.QL0.TargetBitrate"
```

Remark: quality layers are not supported on VCE 1.0. "`QL0`" must be used for quality layers.

3 Sample Applications

The AMF Encoder Sample application show how to setup and use the AMF Video Encoder VCE-AVC 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 VCEEncoderRaw application:

Category	Name	Values	Description
Miscellaneous parameters	ApplyTo	Frame number	Forces all subsequent configuration parameters to be applied to a specific frame
	Input	File name, relative or absolute path	Input file with frames (YUV420, NV12 or BGRA)
	Output	File name, relative or absolute path	Output H.264 file for encoded data
	DX9	Flag (without any values)	Forces Direct3D 9 (default Direct3D 11)
	OpenCL	Flag (without any values)	Forces OpenCL
	MTMode	Flag (without any values)	Enables creating or reading from file of frames in separate thread.
	PerfStat	Flag (without any values)	Enables showing a performance statistic

Additional parameters of VCEEncoderD3D application:

Category	Name	Values	Description
Miscellaneous parameters	Frames	Number of frames to be encoded	Number of frames to be encoded
	ApplyTo	Frame number	Forces all subsequent configuration parameters to be applied to a specific frame
	Output	File name, relative or absolute path	Output H.264 file for encoded data
	DX9	Flag (without any values)	Use Direct3D 9 (default Direct3D 11) for rendering
	DX9EX	Flag (without any values)	The same as DX9 but using Device9Ex instead Device9
	OpenGL	Flag (without any values)	Use OpenGL for rendering
	Windowmode	Flag (without any values)	Shows rendering window for D3D sample application

	MTMode	Flag (without any values)	Enables creating or reading from file of frames in separate thread. Doesn't work for OpenGL.
	PerfStat	Flag (without any values)	Enables showing a performance statistic

3.2 Command line example

3.2.1 Transcoding application (TranscodingHW.exe)

```
VCEEncoderRaw.exe -input input.h264 -output out.h264 -width 1280 -height 720 -usage transcoding -rateControlMethod cbr -targetBitrate 500000 -targetBitrate 100000
```

This command transcodes H264 elementary stream to H.264 video. Encoder is created with "Transcoding" usage.

3.2.2 D3D application (VCEEncoderD3D.exe)

```
VCEEncoderD3D.exe -output VideoSample_1024x768.h264 -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

Background color coding:

- light orange – new parameters that haven't been implemented yet.
- grey – hidden parameters.

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

Category	Name	Values	Description
Encoder static parameters	Usage	0, 1, 2, 3 (Transcoding, UltraLowLatency, LowLatency, Webcam)	Selects the AMF usage (see Section 1.2)
	Profile	66, 77, 100 (Baseline, Main, High)	Selects the H.264 profile
	ProfileLevel	1, 1.1, 1.2, 1.3, 2, 2.1, 2.2, 3, 3.1, 3.2, 4, 4.1, 4.2	Selects the H.264 profile level
	MaxOfLTRFrames	0 … 2	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, B-pictures and Intra-refresh features are not supported. • The actual maximum number of LTRs allowed depends on H.264 Annex A Table A-1 Level limits, which defines dependencies between the H.264 Level number, encoding resolution, and DPB size. The DPB size limit impacts the maximum number of LTR allowed.
Encoder resolution parameters	Width	64 – 1920	Frame width in pixels
	Height	64 – 1200	Frame height in pixels
Encoder rate-control	TargetBitrate	10 000 - 100 000 000 bit/s	Sets the target bitrate
	PeakBitrate	10 000 - 100 000 000 bit/s	Sets the peak bitrate

Category	Name	Values	Description
parameters	RateControlMethod	0, 1, 2, 3 (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 Remarks: • When SVC encoding is enabled, all Rate-control parameters (with some restrictions) can be configured differently for a particular SVC-layer. An SVC-layer is denoted by an index pair [SVC-Temporal Layer index][SVC-Quality Layer index]. E.g. The bitrate may be configured differently for SVC-layers [0][0] and [1][0]. • We restrict all SVC layers to have the same Rate Control method. Some RC parameters are not enabled with SVC encoding (e.g. all parameters related to B-pictures).
	RateControlSkipFrameEnable	True/False (On/Off)	Enables skip frame for rate control
	MinQP	0 – 51	Sets the minimum QP
	MaxQP	0 – 51	Sets the maximum QP
	QPI	0 – 51	Sets the constant QP for I-pictures. Remarks: • Only available for CQP rate control method.
	QPP	0 – 51	Sets the constant QP for P-pictures. Remarks: • Only available for CQP rate control method.
	QPB	0 – 51	Sets the constant QP for B-pictures. Remarks: • Only available for CQP rate control method.
	FrameRate	1*FrameRateDen … 120* FrameRateDen	Frame rate numerator
	VBVBufferSize	1000 – 100 000 000	Sets the VBV buffer size in bits
	InitialVBVBufferFullness	0 - 64	Sets the initial VBV buffer fullness
	EnforceHRD	True/False (On/Off)	Disables/enables constraints on QP variation within a picture to meet HRD requirement(s)
	MaxAUSize	0 - 100 000 000 bits	Maximum AU size in bits
	BPicturesDeltaQP*	-10 ... 10	Selects the delta QP of non-reference B pictures with respect to I pictures
	ReferenceBPicturesDeltaQP*	-10 ... 10	Selects delta QP of reference B pictures with respect to I pictures
Encoder picture-control parameters	HeaderInsertionSpacing	0 … 1000	Sets the headers insertion spacing
	IDRPeriod	0 ... 1000	Sets IDR period. IDRPeriod= 0 turns IDR off
	DeBlockingFilter	True/False (On/Off)	Turns on/off the de-blocking filter
	IntraRefreshMBsNumberPerSlot	0 - #MBs per frame	Sets the number of intra-refresh macro-blocks per slot
	SlicesPerFrame	1 - #MBs per frame	Sets the number of slices per frame

Category	Name	Values	Description
	BPicturesPattern*	0, 1, 2, 3	Sets the number of consecutive B-pictures in a GOP. BPicturesPattern = 0 indicates that B-pictures are not used
	BReferenceEnable*	True/False (On/Off)	Enables or disables using B-pictures as references
Encoder miscellaneous parameters	ScanType	0, 1 (Progressive, Interlaced)	Selects progressive or interlaced scan
	QualityPreset	0, 1, 2 (Balanced, Quality, Speed)	Selects the quality preset
Encoder motion estimation parameters	HalfPixel	True/False (On/Off)	Turns on/off half-pixel motion estimation
	QuarterPixel	True/False (On/Off)	Turns on/off quarter-pixel motion estimation
Encoder SVC parameters (only webcam usage)	NumOfTemporalEnhansmentLayers	0 … MaxOfTemporalEnhansmentLayers	Change the number of temporal enhancement layers. The maximum number allowed is set by the corresponding create parameter. Remarks: • Actual modification of the number of temporal enhancement layers will be delayed until the start of the next temporal GOP. • B-pictures and Intra-refresh features are not supported with SVC.
Encoder SVC per-layer parameters (only webcam usage)	TL<TL_Num>. QL<QL_Num>. <Parameter_name>	Parameter-specific values	Configures SVC frame-rate parameter per SVC layer. • TL_Num — temporal layer number • QL_Num — quality layer number • Parameter_name — frame rate parameter name (look to frame-rate parameters on this table). Remarks: • Quality layers are not supported on VCE 1.0. “QL0” must be used for quality layers.

* this feature is not supported by VCE 1.0

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

Category	Name	Values	Description
Frame per-submission parameters	InsertSPS	True/False (On/Off)	Inserts SPS
	InsertPPS	True/False (On/Off)	Inserts PPS
	InsertAUD	True/False (On/Off)	Inserts AUD
	PictureStructure	0, 1, 2, 3 (None, Frame, TopField, BottomField)	Picture structure
	ForcePictureType	0, 1, 2, 3, 4, 5* (NONE, SKIP, IDR, I, P, B*)	Forces the picture type
	EndOfSequence	True/False (On/Off)	End of sequence
	EndOfStream	True/False (On/Off)	End of stream

Category	Name	Values	Description
	MarkCurrentWithLTRIndex	-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. • When LTR is used with SVC encoding, only base temporal layer pictures can be coded as LTR. In this case, the request to mark the current picture as LTR would be delayed to the next base temporal layer picture if the current picture is in an enhancement layer. If the user submits multiple requests to mark current as LTR between base temporal layer pictures, then only the last request is applied.
	ForceLTRReferenceAllowedBitfield	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	OutputDataType	0, 1, 2, 3* (IDR, I, P, B*)	Type of encoded data
	MarkedLTRIndex	-1 … (MaxOfLTRFrames -1)	Marked as LTR Index. If != -1, then this picture was coded as a long-term reference with this LTR Index.
	ReferencedLTRIndexBitfield	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).

* this feature is not supported by VCE 1.0

Table A-3. Default value of parameters.

Type	Name	Transcoding	Ultra low latency	Low latency	Webcam
Static Parameters (Set at creation time)	Profile	Main	Main	Main	Main
	ProfileLevel	4.2	4.2	4.2	4.2
	MaxOfLTRFrames	0	0	0	0

Type	Name	Transcoding	Ultra low latency	Low latency	Webcam
Rate control	TargetBitrate	20 mbps	6 mbps	10 mbps	10 mbps
	PeakBitrate	20 mbps	6 mbps	10 mbps	10 mbps
	MinQP	18	22	22	22
	MaxQP	51	51	51	51
	QPI	22	22	22	22
	QPP	22	22	22	22
	QPB	22	22	22	22
	FrameRate	30 fps	60 fps	60 fps	30 fps
	VBVBufferSize	20 mbits	110 kbits	1 mbits	1 mbits
	InitialVBVBufferFullness	64	64	64	64
	EnforceHRD	false	true	true	true
	MaxAUSize	0	0	0	0
	FillerDataEnable	false	false	false	false
	BPicturesDeltaQP*	+4	0	+4	+4
	ReferenceBPicturesDeltaQP*	+2	0	+2	+2
Picture Control	HeaderInsertionSpacing**	0	0	0	0
	IDRPeriod	30	300	300	30
	DeBlockingFilter	true	false	false	false
	IntraRefreshNumMBsPerSlot*	0	255	255	0
	SlicesPerFrame	1	1	1	1
	BPicturesPattern*	3	0	0	0
	BReferenceEnable*	true	false	true	true
	ScanType	0	0	0	0
	QualityPreset	Balanced	Speed	Speed	Speed
Motion estimation	HalfPixel	1	1	1	1
	QuarterPixel	1	1	1	1
SVC	NumOfTemporalEnhansmentLayers	disable	disable	disable	0
Per-submission parameters	InsertSPS	0	0	0	0
	InsertPPS	0	0	0	0
	PictureStructure	0	0	0	0
	ForcePictureType	0	0	0	0
	InsertAUD	false	false	false	false
	EndOfSequence	false	false	false	false
	EndOfStream	false	false	false	false
	MarkCurrentWithLTRIndex	-1	-1	-1	-1
	ForceLTRReferenceAllowedBitfield	0x0	0x0	0x0	0x0

* `BPicturesDeltaQP`, `ReferenceBPicturesDeltaQP`, `IntraRefreshNumMBsPerSlot`, `BPicturesPattern` and `BReferenceEnable` parameters are available only when:

- `MaxOfReferenceFrames` is greater than 1
- `NumOfLTR` is 0 (LTR is not used)

** HeaderInsertionSpacing: Every IDR frame has SPS and PPS regardless of default value of HeaderInsertionSpacing per VCE logic.