SAMPLE

Video Edit with OpenCL resizing

1 Introduction

This sample demonstrates how to build the video edit pipeline using AMD's hardware-accelerated Media Foundation Transforms (MFTs). This sample also demonstrates how to write and use a custom MFT for resizing using DX11-OpenCL inter-operations (interops) and DX9-OpenCL interops.

2 Using the sample

2.1 Location

`$<installDirectory>\samples\mediaFoundation\videoEditOpenCl\`

2.2 Contents

Package contents

Folder: `$<installDirectory>\samples\mediaFoundation\videoEditOpenCl\src\`

File name	Description
`MftAsyncTransform.cpp`	Implementation of AsyncTransform class
`MftDx9DllMain.cpp`	DX9 resizer DLL functions
`MftDx9Resizer.cpp`	Class implementation for resizing functionality using DX9-OpenCL interop
`MftDx11DllMain.cpp`	DX11 resizer DLL functions
`MftDx11Resizer.cpp`	Class implementation for resizing functionality using DX11-OpenCL interop
`TranscodeSession.cpp`	Contains functions to build the video edit topology
`VideoEditConfig.cpp`	Contains functions to parse the configuration file
`VideoEditMain.cpp`	The sample's Main file (includes the entry point)

Folder:
`$<installDirectory>\samples\mediaFoundation\videoEditOpenCl\inc\`

File name	Description
`cl_d3d11.h`	OpenCL D3D11 header file
`cl_dx9_media_sharing.h`	OpenCL D3D9 header file
`ClassFactory.hpp`	Functions definitions of class factory
`ComRegistrar.h`	Header file for custom MFT registration functions
`MftDx9Transform.def`	Definition file for custom DX9 MFT DLL
`MftDx11Transform.def`	Definition file for custom DX11 MFT DLL
`MftAsyncTransform.h`	Header file for Async transform class
`MftDx9Resize.cl.h`	Kernel for D3D9 resizing
`MftDx9Resizer.h`	D3D9 resizer class declaration

File name	Description
MftDx9Transform.h	Header file for custom DX9 transform
MftDx11Resize.cl.h	Kernel for D3D11 resizing
MftDx11Resizer.h	D3D11 resizer class declaration
MftDx11Transform.h	Header file for custom DX11 transform
TranscodeSession.h	Header file for video edit session class
VideoEditApi.h	Header file for video edit sample
VideoEditConfig.h	Header file for configuration parsing functions
VideoEffect.h	Resizer's base structure

Folder:
$<*installDirectory*>\samples\mediaFoundation\videoEditOpenCl\config\

File name	Description
exampleConfig.cfg	Configuration file for H.264 encoder and resizer configuration parameters

Folder:
$<*installDirectory*>\samples\mediaFoundation\videoEditOpenCl\docs\

File name	Description
MediaSDK_MFT_videoEditOpenCL.pdf	Sample documentation

Folder:
$<*installDirectory*>\samples\mediaFoundation\videoEditOpenCl\build\windows\

File name	Description
mftDx9ResizerOpenClVs10.vcxproj	Visual Studio 10 project file for building custom D3D9 resizer MFT
mftDx9ResizerOpenClVs10.vcxproj.filters	Visual Studio 10 project filter file
mftDx9ResizerOpenClVs12.vcxproj	Visual Studio 12 project file for building custom D3D9 resizer MFT
mftDx9ResizerOpenClVs12.vcxproj.filters	Visual Studio 12 project filter file
mftDx11ResizerOpenClVs10.vcxproj	Visual Studio 10 project file for building custom D3D11 resizer MFT
mftDx11ResizerOpenClVs10.vcxproj.f ilters	Visual Studio 10 project filter file
mftDx11ResizerOpenClVs12.vcxproj	Visual Studio 12 project file for building custom D3D11 resizer MFT
mftDx11ResizerOpenClVs12.vcxproj.filters	Visual Studio 12 project filter file
videoEditOpenClVs10.sln	Visual Studio 10 solution file
videoEditOpenClVs10.vcxproj	Visual Studio 10 project file for the sample
videoEditOpenClVs10.vcxproj.filters	Visual Studio 10 project filter file
videoEditOpenClVs12.sln	Visual Studio 12 solution file
videoEditOpenClVs12.vcxproj	Visual Studio 12 project file for the sample
videoEditOpenClVs12.vcxproj.filters	Visual Studio 12 project filter file

2.3 Parameters Encoder-specific configuration parameters

Parameter name	Default value	Supported range	Remarks
encGOPSize	20		Max number of frames in a GOP (0=auto)
encMeanBitrate	3000000		Bitrate of encoded video (bits per second)
encMaxBitrate	4000000		Maximum bitrate of encoded video (used only for VBR) in bits per second
encBufferSize	2000000		VBR buffer size
encNumBFrames	1	0 - 3	Specifies the number of B frames to be inserted
encCompressionStandard	77	For supported values, see http://msdn.microsoft.com/en-us/library/windows/desktop/dd318776(v=vs.85).aspx	Compression standard
encRateControlMethod	1	eAVEncCommonRateControlMode_CBR = 0, eAVEncCommonRateControlMode_PeakConstrainedVBR = 1, eAVEncCommonRateControlMode_UnconstrainedVBR = 2, eAVEncCommonRateControlMode_Quality = 3	For more details, see http://msdn.microsoft.com/en-us/library/windows/desktop/dd388772(v=vs.85).aspx
encLowLatencyMode	0	1 - True 0 - False	Specifies whether the output stream should be structured so that the encoded stream has a low decoding latency.
encQualityVsSpeed	60	0 - Low quality faster encoding 100 - Higher quality, slower encoding	
encCommonQuality	50	0 to 100 0 - low quality 100 - highest quality	This parameter is used only when encRateControlMethod is set to eAVEncCommonRateControlMode_Quality. in this mode the encoder selects the bit rate to match the quality settings.

Resizer-specific configuration parameters

Parameter name	Default value	Supported range	Remarks
resizerProfile	1	0 - HD1080 (1920 x 1080) 1 - HD720 (1280 x 720) 2 - WVGA (800 x 480) 3 - VGA (640 x 480) 4 - QVGA (320 x 240)	

Common configuration parameters

Parameter name	Default value	Supported range	Remarks
useSWCodec	0	Enable=1 Disable=0	Enables/Disables the use of software Encoder and Decoder. If set to 0, hardware-based codecs are used to encode and decode the stream; otherwise software-based codecs are used.

2.4 Compile

1. Ensure that the following tools are present:
   - Microsoft Visual Studio 2010 or 2012
     If Windows Software Development Kit (SDK) is not installed, install it from http://msdn.microsoft.com/en-us/library/windows/desktop/hh852363.aspx.
2. Open one of the following solution files:
   `<installDirectory>\samples\mediaFoundation\videoEditOpenCl\build\windows\videoEditOpenClVs12.sln`
   `<installDirectory>\samples\mediaFoundation\videoEditOpenCl\build\windows\videoEditOpenClVs10.sln`
3. Build the sample:
   - Open the `videoEditOpenClVs10.sln` solution file with Microsoft Visual Studio 2010 Professional Edition or the `videoEditOpenClVs12.sln` solution file with Microsoft Visual Studio 2012 Professional Edition.
   - To build the solution, select `Build > Build Solution`.
     Ensure that all the projects under the solution are built.

     The executable `videoEditOpenCl.exe` is created in the following folders for 32-bit builds and 64-bit builds, respectively:
     `<installDirectory>\MediaSDK\samples\mediaFoundation\videoEditOpenCl\bin\x86\`
     `<installDirectory>\MediaSDK\samples\mediaFoundation\videoEditOpenCl\bin\x86_64\`

3 Run

The sample can be executed on an AMD platform that includes the VCE and UVD hardware blocks. AMD Catalyst Driver is required to execute the sample application.

On the command prompt, change to the directory that contains the executable, and execute the following command:
`videoEditOpenCl.exe -i <input.avi> -o <output.asf> -c <configFile> -l <0, 1, or 2>`

-l enables the logging. Setting "0" means no logging. "1" generates the log at the API level. "2" generates logs at the video edit session level.

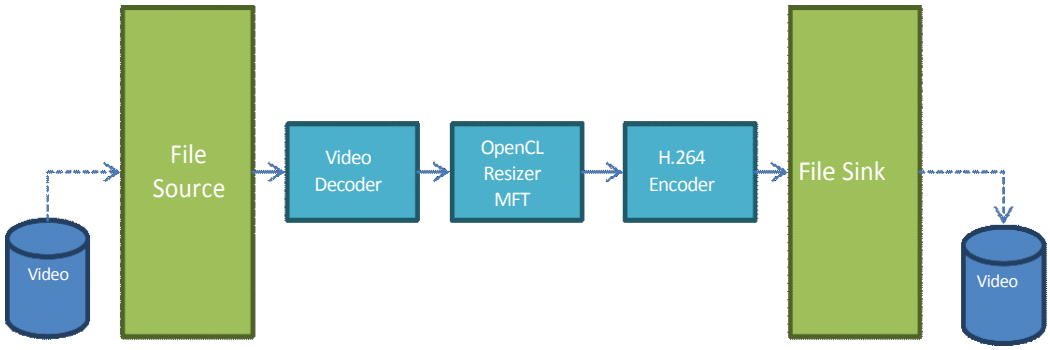
The `<installDirectory>\inc\ErrorCodes.h` file contains information about the error codes. You can also print the Microsoft error code using the `getMicrosoftErrorCode()` API, as shown in `VideoEditMain.cpp`. The Microsoft-specific error codes can be understood from the `Mferror.h` file that Microsoft provides as part of its Windows kits.

The configuration file associated with this sample is located in the following folder:
`<installDirectory>\samples\mediaFoundation\videoEditOpenCl\config\`

`Output.asf` will have the encoded resized (as per the configuration) video.

4 Implementation Details

The sample implements the following video edit pipeline. The OpenCL Resizer MFT resizes the decoded video frames to the output resolution specified in the configuration file by using the OpenCL kernel.



The pipeline uses data transfer inter-operations (interops) to eliminate the unnecessary copying of buffers from the GPU to the CPU and back to the GPU.

On Windows 8/8.1, the `DXGIDeviceManager` class and its associated object, `D3D11Texture2D`, are used. The video decoder MFT provides the `D3D11Texture2D` object with `MFVideoFormat_NV12`-decoded video as the output. The OpenCL Resizer MFT uses the

`DirectX 11-OpenCL` interop to obtain the OpenCL image from `D3D11Texture2D`, on which the resizer kernel is applied

On Windows 7, the `Direct3DDeviceManager` and its associated object, `Direct3DSurface9`, are used. The video Decoder MFT provides the `Direct3DSurface9` object with `MFVideoFormat_NV12`-decoded video as the output. The OpenCL Resizer MFT uses the `DirectX9-OpenCL` interop to obtain the OpenCL image from `Direct3DSurface9`, on which the resizer kernel is applied.

By default, the sample uses the `DirectX 11-OpenCL` interop path on Windows 8/8.1 and the `DirectX9-OpenCL` interop path on Windows 7. The `DirectX 11-OpenCL` interop is not supported on Windows 7.

5 Supported formats

The following file formats are supported:

- Input file/container formats: .avi, .mp4, .wmv
- Video decoders supported: H264, MPEG4 part II, WMV9, VC1
- Output file/container format .asf
- Video encoder supported: H264

6 Known limitations

The sample is currently supported on the following platforms:

- Windows 7 (DirectX 9)
- Windows 8/8.1 (DirectX 9 and DirectX 11)

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For AMD Accelerated Parallel Processing:

URL: developer.amd.com/appsdk
Developing: developer.amd.com/
Forum: developer.amd.com/openclforum