Intel and Badaboom Video File Transcoding

Introduction

Intel® Quick Sync Video, built right into 2nd generation Intel® Core™ processors, is breakthrough hardware acceleration that lets the user complete in minutes what used to take hours. Create DVDs or Blu-ray discs, cover video files for your media plater, and convert video for upload to your favorite social networking sites – all in a flash.

Badaboom* uses Intel® Quick Sync Video technology to transcode video files in just minutes. Why do videos need to be transcoded? In order for a video to play back on a device such as a smartphone or a tablet, it needs to be formatted to correct specifications. With so many different devices out there, odds are low a video from a camcorder will automatically play on all of them. That’s where Badaboom comes in: it transcodes video files to play on hundreds of the most popular devices available today—and it does so quickly and easily.
Badaboom was built with four key cornerstones in mind, and these have been strengthened with its implementation of the Intel Media SDK:

- **Efficient use of system resources:** Badaboom takes advantage of Intel® HD Graphics 2000 and 3000 series chips, offloading the CPU and allowing users to continue other tasks without unnecessarily slowing down their system.
- **Transcoding speed:** Intel's Quick Sync Video technology enables Badaboom to transcode at several hundreds of frames per second.
- **Uncompromised video quality:** Intel's Quick Sync H.264 encoder ensures Badaboom will continue to produce high quality outputs.
- **Ease of use and flexibility:** Whether it is the default settings or advanced features, the Intel Media SDK supports the video processing parameters necessary for the flexible Badaboom interface to operate as users have come to expect.
Badaboom also utilizes video enhancement features, such as scaling, provided by Intel Quick Sync Video technology. More information about which video pre-processing filters are supported by the Media SDK can be found later in this paper.

The Intel® Media Software Development (MSDK) library exposes Intel Quick Sync Video technology to the application layer through three sets of functions: Decode, Video Pre Processing (VPP), and Encode. After Badaboom does any necessary demuxing of the input video file, the raw video frames are handed off to the Intel MediaSDK for the video processing. Badaboom performs video processing on a separate process from the demuxing, audio processing, and muxing, which allows it to integrate the Media SDK into its transcoding pipeline.

Badaboom uses asynchronous Decode, VPP and Encode functions to submit multiple frames to Intel MSDK to better and efficiently utilizes Intel® QSV.
Intel MSDK:

Mentioned earlier, the Intel Media SDK provides three sets of functions to access Intel QSV features provided by 2nd generation of Intel® Core processors. These functions are divided into following categories:

1. Decode
2. VPP
3. Encode

Intel MSDK Initialization:

The application initializes the MSDK session through MFXInit() function before it can use any MSDK functions. Each MSDK session can run exactly one instance of Decode, VPP and Encode functions. The application can start multiple sessions to process multiple frames in parallel. These multiple sessions can run independently or can be joined through to share the session data. The sessions can be joined through MFXJoinSession or MFXCloneSession.

```c
mfxSession session, clone;
// to initialize a session
status = MFXInit(MFX_IMPL_AUTO, 0, &session);

// to clone the session
status = MFXCloneSession(session, &clone);

// to close a session
Status = MFXClose(session);
```

MSDK Decode:

Media SDK Decode takes a compressed pure or elementary video stream as input and converts it to raw frames. The application can retrieve the decoded frames in display order or decoded order. The output frames are in NV12 color format. The input codecs supported by Intel® Media SDK are H.264, MPEG-2 and VC-1.

The Intel® Media SDK provides a very simple and easy to way to expose decode capabilities of Intel® Quick Sync Video technology. At high level, application needs to perform following steps:
1. Initialize Decode Parameters: The application needs to initialize decode parameters before initializing decode processing. The application can call mfxVideoDecode_DecodeHeader() to obtain decode parameters. This function parses the input bitstream and fills the decode parameters in mfxVideoParam structure.

2. Get Surface Allocation Requirements: If two components are going to share surfaces, the application needs to find out the surface allocation requirement. This can be done by calling MFXVideoDECODE_QueryIOSurf().

3. Decoder Initialization: The application needs to initialize the decoder using MFXVideoDECODE_Init(). At this point, the Intel® Media SDK is ready to decode the frames.

4. Frame Decoding: The application can decode the frames using MFXVideoDECODE_DecodeFrameAsync(). This is an asynchronous function and returns immediately without waiting for the result. Media SDK performs following four steps:
   a. Checks the input data
   b. Prepares a valid computation workload for hardware acceleration
   c. Submits the workload for processing
   d. Update the fields in the output structure as needed.
   The Media SDK returns a sync point after successful execution of asynchronous function. The application needs to use this sync point to synchronize the results using MFX_VideoCORE_SyncOperation().

5. De-initialize the Decoder: After all frames decoded and retrieved de-initialize the decoder using MFXVideoDECODE_Close().

Video Pre-Processing:

Intel® Quick Sync Video provides many Video Pre-Processing (VPP) filters:
- Deinterlacing/Inverse Telecine
- Cropping and Resizing
The application can access these filters using Intel® Media SDK functions. The VPP takes raw frames as input and produces raw frames as output. To utilize VPP filters of Intel® Quick Sync Video technology, the application needs to perform following steps:

1. Get Surface Allocation Requirement: The application can retrieve the surface allocation requirement by calling MFXVideoVPP_QueryIOSurf() function.
2. Initialize VPP: The application needs to initialize VPP before submitting frames for preprocessing. The initialization can be done using MFXVideoVPP_Init().
3. Preprocess Frames: To preprocess a frame, the application uses MFXVideoVPP_RunFrameVPPAsync() function to submit a frame for preprocessing. If number of frames consumed at input is more than the output frames, VPP returns MFX_ERR_MORE_DATA for additional inputs. If number of frames at input is less than the output, VPP returns either MFX_ERR_MORE_SURFACE (if more outputs are ready) or MFX_ERR_NONE (if one output is ready).
4. De-initialize the preprocessor: After video pre-processing is done, the applicant can deinitialize the preprocessor by calling MFXVideoVPP_Close().

**Intel Media SDK Encode:**

Media SDK Encode takes raw frames as input and compress them into a bitstream. Encode can process frames in Display order or in Encoded order. The encode supports variable bit rate (VBR)
and constant Quantization Parameter (QP). The Encode output consists of one frame of a bitstream with the timestamp passed from the input frame. To encode frames using Intel Quick Sync Video technology, the application needs to perform following steps:

1. **Get Surface Allocation Requirements:** If encoder is sharing surfaces with other components, then it needs to get the surface requirement. The application can retrieve this information using `MFXVideoEncode_QueryIOSurf()` function.

2. **Initialize Encoder:** The application needs to initialize the encoder using `MFXVideoEncode_Init()` function.

3. **Encode Frames:** To encode a frame, the application uses `MFXVideoENCODE_EncodeFrameAsync()` function. If the application wants to set a new encoding configuration, it needs to retrieve any cached frames in encoder using `MFXVideoENCODE_EncodeFrameAsync()` function with a “NULL” input frame pointer. After that, the application needs to call `MFXVideoENCODE_Reset()` function with the new configuration.

4. **De-initialize the Encoder:** After all frames are encoded and retrieved by the application, the application can de-initialize the encoder using `MFXVideoENCODE_Close()` function.

### About Elemental

Elemental Technologies is the leading provider of massively parallel video processing solutions for broadcast and online video customers. Elemental’s products use off-the-shelf, programmable graphics processing units (GPUs) for compute-intensive video processing and conversion tasks. The product line is ideal for digital media workflows that require video encoding for Internet and mobile delivery, faster-than-real-time transcoding, or support for adaptive bit rate distribution.

Elemental helps content owners and distributors deliver any video to any screen, at any time – all
at once...it’s Media Multiplied™. Founded in 2006, Elemental is headquartered in Portland, Oregon. To learn more about Elemental, visit www.elementaltechnologies.com.
Notices

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to:
http://www.intel.com/design/literature.htm
Intel, the Intel logo, and Xeon are trademarks of Intel Corporation in the U.S. and other countries.
*Other names and brands may be claimed as the property of others.
Copyright 20011 Intel Corporation. All rights reserved.