White Paper
Abhishek Girotra
Graphics SW TME
Intel Corporation

Hardware-Accelerated Video Decode on the Intel® Atom™ Processor with the Intel® System Controller Hub US15W Chipset Platform

August 2009

322503

Executive Summary

This paper presents the complexities of the video decode software ecosystem and the platform elements that constitute the hardware accelerated video decoding on the Intel® Atom™ processor with the Intel® System Controller Hub (SCH) US15W platform across multiple operating systems.

In addition to explaining the inner workings of video decode software stack, the paper discusses benefits of hardware accelerated video decode and provides the audience an introduction to various parameters that govern the video playback performance.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. www.intel.com/embedded/edc.

§

Contents

Background

With increase in connectivity of embedded devices to the Internet, demand for watching high definition videos online or on locally stored media content has exploded. Due to the intense computing requirements of video processing, Intel has introduced an ultra low power platform which pairs the Intel® Atom™ processor with the Intel® System Controller Hub (SCH) US15W. This platform leverages dedicated hardware accelerators to decode high definition video in real-time while doing so with a very low power requirement. Getting the best possible results for video decoding is relatively complex, due to the many variables of media content types, device drivers, players, and perceptions of the users. This paper will discuss the complexities of the video decode software ecosystem and the platform elements that constitute hardware accelerated video decoding on the Intel® US15W-based platforms across multiple operating systems.

The Intel® Atom™ Processor With the Intel® SCH US15W Platform

The Intel® Atom™ processor paired with the Intel® SCH US15W chipset is a small form factor graphics platform.

The Intel® SCH US15W is a single chip solution which combines the functionality of the I/O Controller Hub and Memory Controller Hub. US15W features Intel® GMA500 graphics system hardware with advanced 3D features, a memory controller and extensive I/O capabilities such as USB 2.0, Secure Digital Input/output (SDIO) and PCI Express*. Additionally, it supports Intel® HD Audio and hardware accelerated video decode which is the topic of interest. Intel® US15W chipset supports a 433/500 MHz CMOS front side bus (FSB), dual independent display and up to 2GB of memory in a single channel with one or two ranks.

The Intel® GMA500 graphics system hardware is supported by two graphics driver solutions: the Intel® GMA 500 driver and the Intel® Embedded Graphics Driver. This white paper focuses only on the Intel® Embedded Graphics Driver. For more information about the Intel® Embedded Graphics Driver please visit

http://edc.intel.com/Software/Downloads/IEGD/

Video Decode Capability in the Intel® SCH US15W Chipset

The Intel® SCH US15W chipset featuring hardware video acceleration relieves the decode burden from the Intel® Atom™ processor and reduces power consumption of the system. Full hardware acceleration of H.264 [1], MPEG2 [3], VC1 [4], MPEG4 [2] and WMV9 [5] is supported, which eliminates the need for "software decode" and off-loads the host processor.

Profile and Levels

Table 1 summarizes the video codec profile/level/maximum frame rate/maximum bit rate supported by Intel® GMA500 core for each CODEC standard accelerated in hardware.

Note: The ability to play video at the stated resolution and frame rates is subject to the capabilities of the rest of the software stack which will be discussed later.

Table 1. Supported Profile/Level/Max bit rate on the Intel® SCH US15W

Format	Profile	Levels	Max Resolution	Max Frame Rate	Max Bit Rate (Mbps)
H.264	Baseline Profile	L1,L1.2,L1.3,L2,L2.2,L3 [1]			
H.264	Main Profile	L1,L1.2,L1.3,L2,L2.2,L3,L3.1,L3.2, L4.1			
H.264	High Profile	L1,L1.2,L1.3,L2,L2.2,L3,L3.1,L3.2, L4.1	1080i 720p	30 fps 60 fps	45
MPEG-4	Simple Profile	L0,L1,L2,L3			
MPEG-4		DivX HD [3]			
MPEG-4	Advanced Simple Profile	L0,L1,L2,L3,L5 [2]	720p	30 fps	8
VC-1	Simple Profile	LL,ML			
VC-1	Main Profile	LL,ML,HL			

Format	Profile	Levels	Max Resolution	Max Frame Rate	Max Bit Rate (Mbps)
VC-1	Advanced Profile	L0,L1,L2,L3	1080p 720p	30 fps 60 fps	40
WMV9	Simple Profile	LL,ML			
WMV9	Main Profile	LL,ML,HL	1080p 720p	30 fps 60 fps	20
MPEG-2	Simple Profile	Main Level			
MPEG-2	Main Profile	Low, Main, High1440, High Level	1080i 720p	30 fps 60 fps	80

NOTES:

1. L3.1 may be supported where the toolset use is those common to both Baseline and Main Profile.
2. At L5 resolution, only Simple Profile Toolset is supported.
3. DivX is based on MPEG4 Advanced simple profile but ignores the levels defined by MPEG4. There are two variants of DivX. The "certified" version does not require GMC or quarter pixel motion compensation prediction. The "non-certified" does support these features.

Entry Point

Figure 1. Generic Video Processing Pipeline

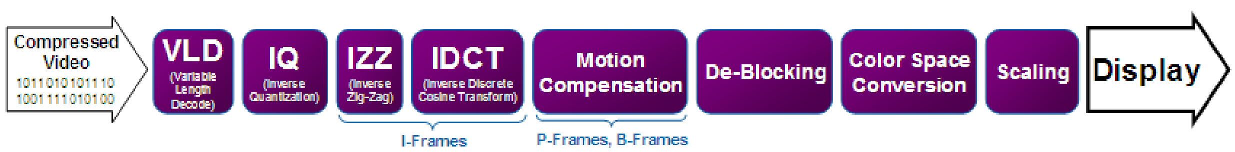


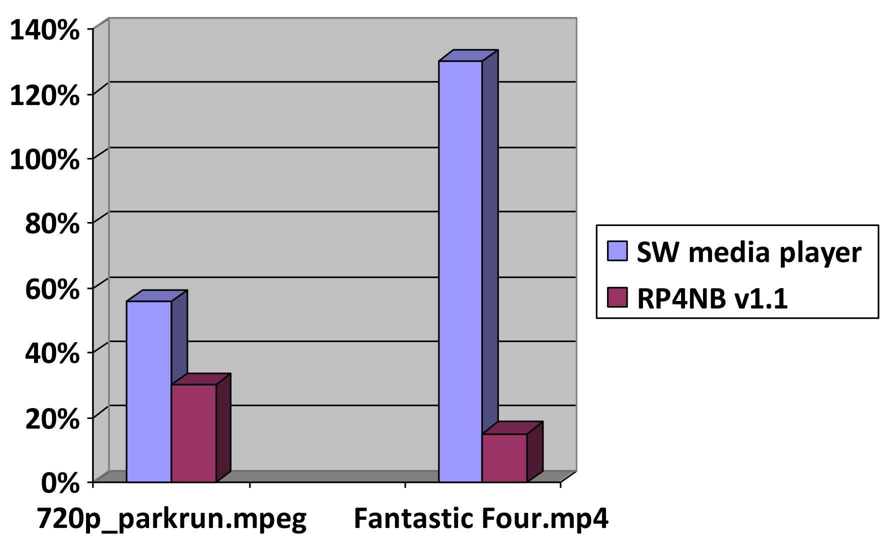
Figure 1 shows generic video decoding steps. A compressed video bit-stream goes through multiple processing steps in series where the bit-stream gets decompressed and rearranged into displayable pixel data. The processing stages as shown in Figure 1 can either be implemented in software or offloaded for processing to hardware. The piece of software module that deals with video processing is called host decoder software or codec. In case of hardware accelerated video decode, chipsets usually supports multiple entry points at which software can offload processing tasks to hardware. Such entry points include Variable Length Decoding (VLD), Inverse Discrete Cosine Transform (IDCT), motion compensation (MC), inverse quantization, scaling and other functions as shown in Figure 1. Intel® GMA500 features VLD and MC as two main entry points for MPEG-2, H.264, VC-1, WMV9 and MPEG-4 CODEC standards.

The VLD entry point takes full advantage of the hardware acceleration available in the chipset whereas the MC entry point offloads only the motion compensation work to hardware accelerator. Higher CPU Utilization will be observed during Video Playback utilizing the MC entry point vs. the VLD entry point.

Hardware vs. Software Decode

This section is for illustration purposes only; Figure 2 draws the comparison between CPU utilization of a SW media player (identity hidden on purpose) and RP4NB v1.1 under Ubuntu* 8.04 for the Mobile and Embedded (UME) operating system. The SW media player does not have a host software decoder or codec that can take advantage of hardware acceleration on the Intel® Atom™ processor with the Intel® SCH US15W chipset platform; the RP4NB v1.1 media player supports hardware-accelerated video decode on the Intel® Atom™ processor with the Intel® SCH US15W chipset platform system under test with a 1.6 GHz Intel® Atom™ processor.

Figure 2. CPU Utilization of SW vs. HW-accelerated Decode on the Intel® Atom™ Processor with the Intel® SCH US15W Chipset Platform



Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations.

Table 2 shows the streams that were used as test vectors to observe CPU utilization of media players. The table also contains stream characteristics such as format, resolution, profile, level and bit-rate. The results showed HW-accelerated video decode significantly reduced CPU utilization. In case of the MPEG-2 test video stream, 2x improvements were seen. H264 video decode in this case exhibited 8x improvements in CPU utilization of media playback. Performance of hardware accelerated video decode may vary with changes in stream characteristics.

Table 2. Test Stream Characteristics

Stream Name	Format	Profile	Level	Resolution	Bit Rate	Container Format
720p_parkrun.mpeg	MPEG-2	Main	High	1280x720	19.5 Mbps	MPEG-PS
Fantastic Four.mp4	AVC (H.264)	High	4.1	1280x544	5.8 Mbps	MPEG-4

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations.

Media Content

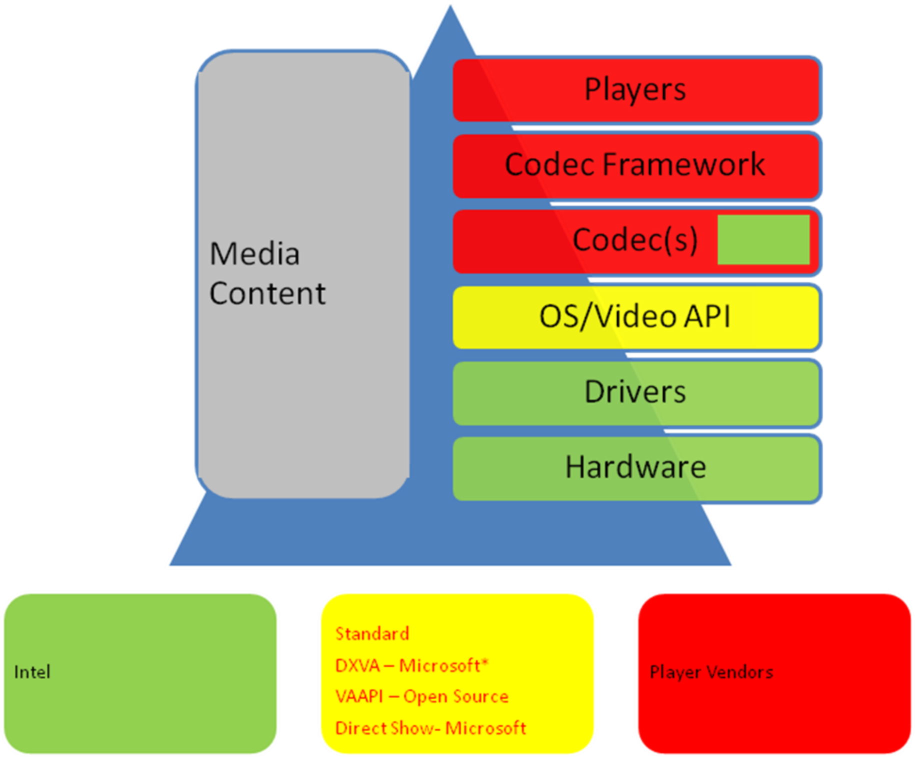
A container file [6] is a metadata file that interleaves audio and video codec data along with transport layer syntax. There are many popular multimedia containers like 3GP, MOV, ASF, MP4, TS, MPEG, etc. The differences in container formats arise due to their popularity, the support they provide for newer audio/video codec(s) and usage models. Due to the availability of a vast array of file formats, special care has to be taken to make sure the media player application and host decoder software installed is capable of decoding the content you choose.

You can use tools such as media analyzers to read audio and video file formats and get more detailed information on codec, profile/levels, and bit rate of video stream embedded in the file. There are several media analyzer tools available for free and for commercial use. MediaInfo* tool http://mediainfo.sourceforge.net/en is one such example. If you determine that the media content is not supported by the media player of your choice, then software trans-coders are other tools that come in handy. Trans-coding software can be used to convert file format under question to file format of interest with a choice of maintaining integrity of underlying video and audio elementary stream data. Ready to compromise with digital quality of video, you can also choose to re-encode elementary data from one video standard to other, e.g., MPEG-4 to MPEG-2 trans-coding.

Video Decode Software Stack

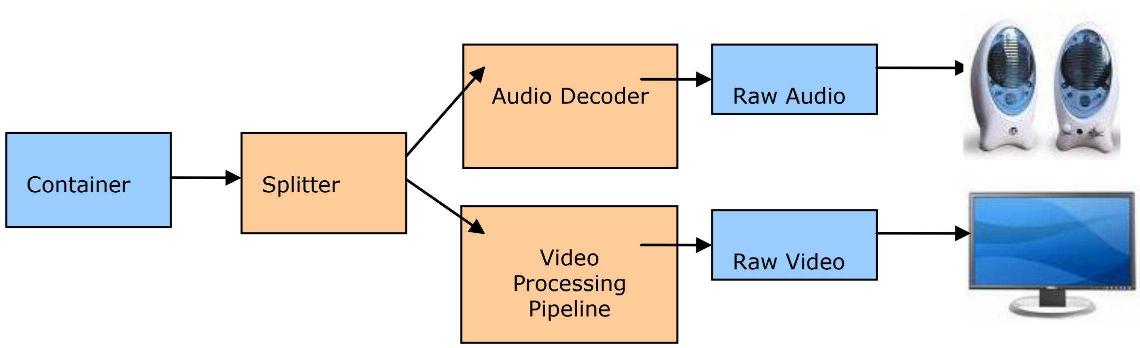
Figure 3 shows all the media ecosystem components that come together to successfully take advantage of hardware accelerated video decode.

Figure 3. Media Ecosystem Components



Applications such as media players parse the media content for a specific container format before it split or de-multiplex the payload data and un-wrap the video bit-stream. The raw compressed video bit-stream is known as an "elementary video bit-stream". A splitter is part of the host decoder software that is responsible for parsing the container format and splitting the payload into elementary streams. Media players would need to rely on different splitters for different container formats. Figure 4 describes the high level data flow during media decode and presentation process.

Figure 4. High Level Data Flow During Media Decode and Presentation



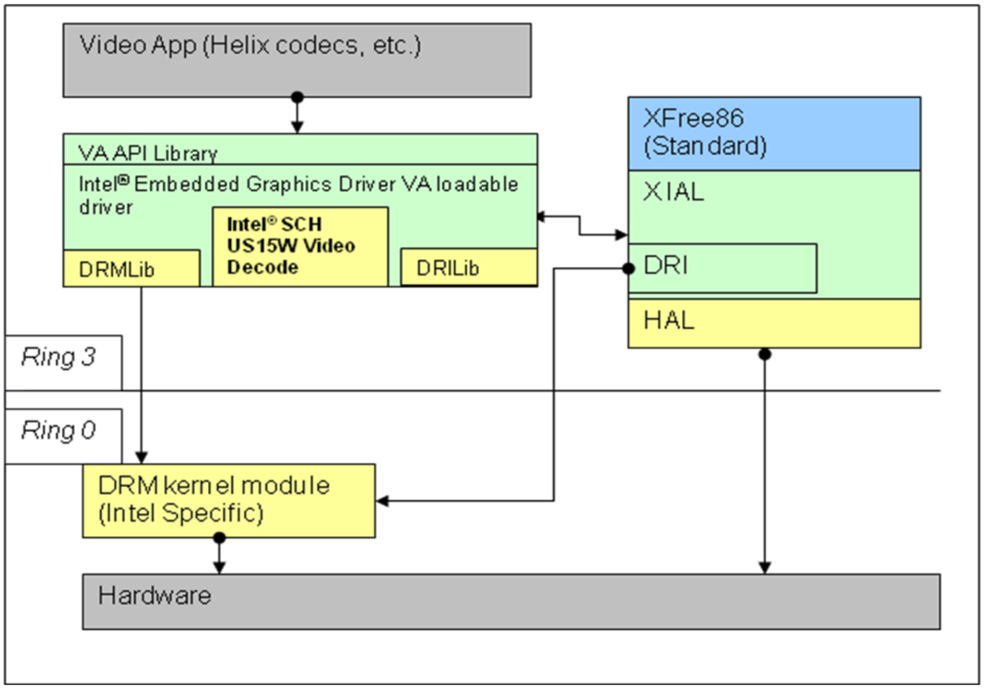
Depending on the format of the elementary video stream, the media player selects the appropriate host software decoder or codec present in the codec framework that is capable of decoding the elementary video stream. Codec(s) may be responsible for decode state machine tracking, compressed buffer management, reference frame buffer management, and actual mathematical decoding workload. Codec(s), if not completely implemented in software, may utilize the hardware acceleration capability of the hardware and hand over the compressed video bit-stream to the hardware decode engine with the help of the graphics and video hardware driver. An example of the above functionality would be the .ASF file format. ASF is a standard container for Microsoft WMA and WMV audio/video codec(s). After unwrapping the .ASF file, the host decoder software may detect the WMV9 video codec, and it can choose to use the capabilities of the Intel® US15W chipset to accelerate WMV9 decoding.

In all operating systems, standard API or infrastructure exists that abstract interfaces allow host decoder software to talk to underlying graphics drivers to offload video processing tasks. In cases when API does not exist, you would need to create codec(s) with private non-standard interfaces to access the graphics driver to accelerate the video decoding using hardware.

Linux*

VAAPI is an industry standard API created by Intel that is targeted to be hardware-abstracted so that player vendors outside Intel can support this API. In Linux* the VAAPI [11] is used to enable hardware-accelerated video decode at various entry points (VLD, MC etc.) for coding standards supported by the Intel® SCH US15W chipset. VAAPI defines the scheme to pass various types of data buffers from the application to Intel® SCH US15W hardware for decoding.

Figure 5. Video Driver Architecture in Linux*

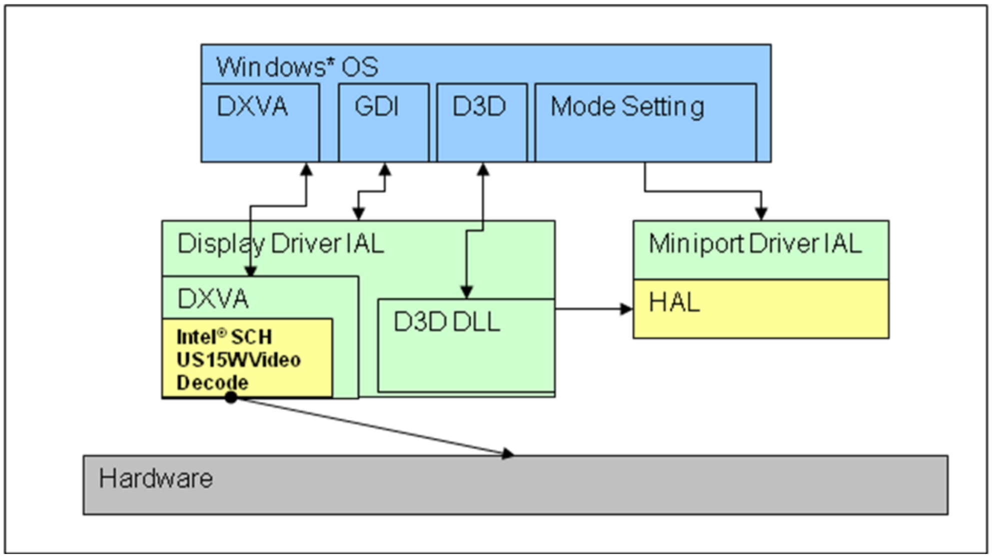


There are two important mega tasks that graphics drivers help with: decoding and drawing of frames on the screen. Decoding of buffers passed by the codec/application to the VAAPI is carried out by driver components which conform to Direct Rendering Infrastructure (DRI) (see Figure 5). In DRI systems DRI clients, XServer [7] and the kernel compete for direct access to video and graphics hardware. XServer might like to do 2D rendering, the client might read or write frame buffers and the kernel dispatches DMA buffers to hardware. The DRM kernel module provides a hardware lock to synchronize hardware accesses. The DRI client (a video driver module) fills the buffers provided by X system with commands and requests for the DRM to send buffers to the video decode hardware engine. Once decoding is done, XVideo extensions in X provide capabilities for 2D drawing and hardware overlay features along with color conversion and scaling of video.

Windows* XP

In Windows* XP, DirectX Video Acceleration (DXVA) [12] framework provided by Microsoft is equivalent to VAAPI specification for Linux. The DXVA specification defines how information is exchanged between user-mode codec/application and kernel mode Intel® Embedded Graphics Driver decode/display driver.

Figure 6. Video driver architecture in Windows* XP



DXVA consists of API for software decoders and Device Driver Interface (DDI) for Intel® Embedded Graphics Driver Decode/Display drivers (see Figure 6). In a video decode pipeline some of the video decoding processing happen on host CPU and computationally expensive tasks are offloaded to hardware acceleration for off-host processing. If there is a need for acceleration at VLD or MC entry points, user mode software codec(s) can use API provided by DXVA and Intel® Embedded Graphics Driver can use DDI to implement acceleration operations. Software codec(s) has responsibility to provide software modules as a backup if graphics driver has not implemented hardware acceleration capabilities for certain operations.

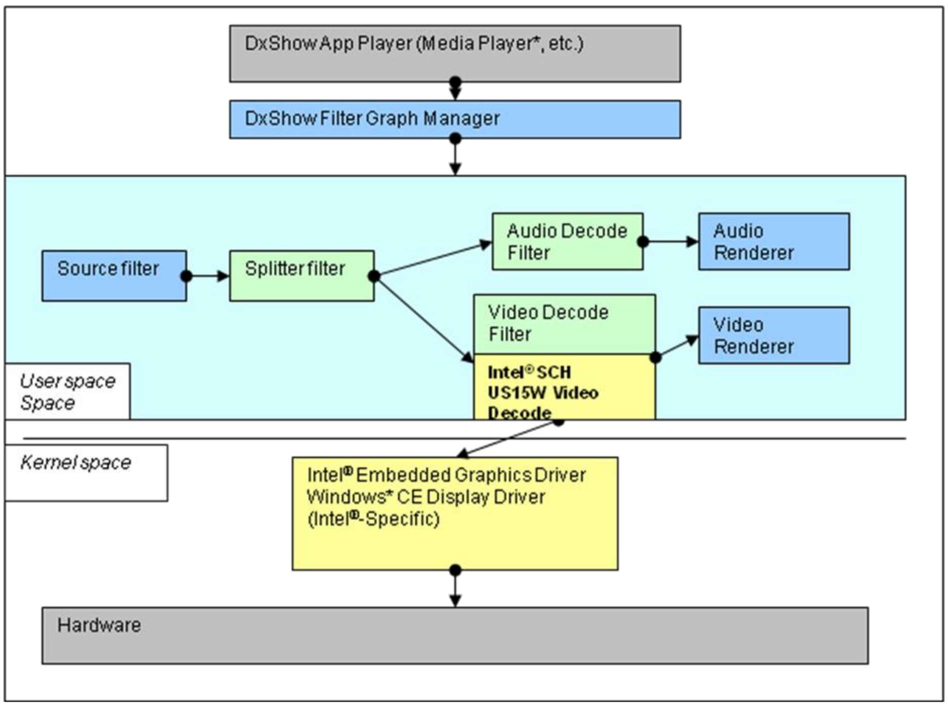
DXVA works in conjunction with the video rendering model provided by Direct Show framework. Decoded video is handed over to overlay rendering or VMR-7/9 [13] for display by the Intel® Embedded Graphics Driver which implements optional DDI for video hardware overlay and DDI for DirectDraw/Direct3D for VMR-7/9 [14] provided by the DirectShow framework.

The Intel® Embedded Graphics Driver for Windows XP currently does not support hardware acceleration for the MPEG-4 standard because no extensions are currently defined for DXVA to support MPEG-4. Furthermore, the Intel® Embedded Graphics Driver currently supports the MC entry point only for VC-1 hardware acceleration.

Windows* CE

In Windows* CE, as shown in Figure 7, the Microsoft* Direct Show multimedia framework does not provide a standardized video decoding acceleration API like the DXVA. Intel provides special audio-video codec software libraries based on Direct Show multimedia framework that offload video decode processing to hardware. Intel® Embedded Graphics Driver Windows CE Direct Show filters are provided to customers in the form of DLLs that interfaces (at this point Intel only supports private non-published interfaces) with the Intel® Embedded Graphics Driver display and decode driver present in Windows CE.

Figure 7. Video driver architecture in Windows* CE



The Intel® Embedded Graphics Driver Windows CE Decode and Display drivers are responsible for communicating with the Intel® SCH US15W chipset and facilitating with data and command passing for decode and display video on screen. Splitter filters that help in de-multiplexing audio and video streams are also provided as part of the Intel® Embedded Graphics Driver package.

Currently Intel® Embedded Graphics Driver codec filters work with Intel® Embedded Graphics Driver splitter filters only. VC-1 Direct Show filters are currently not available as part of the Intel® Embedded Graphics Driver Direct Show filter package. Therefore, VC-1/WMV9 video decode has to be done with the help of software codec only.

Video Decode Performance

Video decode playback performance is judged in many ways. There are certain HD video benchmarking tests that focus on silicon performance in terms of video noise and spatial quality of video. For graphics drivers, performance is measured by focusing on CPU utilization, average frames per second being displayed, and Audio/Video Sync measurements, while video is being played in order to guarantee an acceptable user experience. Next we would discuss how the above parameters relate to driver performance.

- **CPU utilization (% lower is better)**: The host CPU will always be involved at some level in a video decoding system. The use of the hardware decoding engine will help minimize the use of the host CPU by the host software decoder. However, CPU utilization cannot be used as the sole indicator of performance since the quality is also important. For example, a solution that plays back a video with low CPU utilization but only displays two frames per second is not generally an acceptable solution.

- **Average frames per second (higher is better)**: A video is typically encoded with a certain frame rate often but is not always aligned to the profile and level dictated by the specification. A video decoding system should ideally play back the video at the full frame rate at which the video stream was encoded. If average frames per second being measured is within certain confidence interval of ideal frame rate then video playback is generally acceptable.

- **Audio/Video synchronization**: Due to difference in processing delays, audio and video streams decoded at different times can go out of synchronization. Different studies are conducted to determine the minimum amount of a/v sync errors that can be detected by end users. The a/v sync standard is based on notion that light travels faster than the sound, which make people more tolerant of audio lagging video than audio leading the video. One such standard example is ATSC implementation which suggests that sound program should never lead video by more than 15 milliseconds and should never lag the video by more than 45 milliseconds [8]. Therefore, to make user experience pleasurable, graphics driver performance should take into account that no audio playback aberrations are present and audio is not out of sync of video under a defined range.

Video decode performance numbers are dependent on a lot of parameters. A general rule of thumb is that VLD entry point offloads more processing from CPU to GPU than MC entry point, which results in lower CPU utilization numbers. Sometimes the media players themselves play a huge role in the user experience. Some players opt not to use hardware video overlay capabilities supported by the Intel® Embedded Graphics Driver. Hardware overlay provides dedicated memory buffer for rendering video data. In the case of shared video memory among multiple applications, each application has to monitor itself to avoid writing outside its allocated display space and track the movement of the display window by the user. As a result of the monitoring, significant computing resources are wasted. Due to dedicated video buffers provided by the hardware overlay, the media player application does not have to worry about the constant monitoring of the video window therefore leading to lower CPU utilization. In the case where the media player chooses to use hardware video overlay, better video decode performance is observed due to freed up CPU resources.

We have seen certain players when using overlay do not clean up memory allocation after a stream has finished playing, which sometimes results in corruption of video or loss of overlay if the stream is played second time. Another example of a media player-centric problem is if an application or host decoder software is not fully optimized to run high definition, the content then risks exceeding the video pipeline memory utilization more than the memory capacity that was allocated for the graphics driver operation. This results in application crashes, "out of memory" errors or other undesirable behaviors.

Video Decode Ecosystem

Linux*

Any media player can use the hardware acceleration capability provided by the Intel® SCH US15W chipset for H264, VC-1, MPEG-2, WMV9, and MPEG-4 video standards on Linux*. To successfully offload video processing to the GPU, the application and the host software decoder should be available to communicate with VAAPI. Intel has enabled codec frameworks like Helix* and FFMPEG*. Helix and FFMPEG bundle media player and host decoder software or codec(s) that makes use of VAAPI. Some examples of media players that have been enabled include RP4NB [9], RP4MID [9], Helix Player, Splay* and MPlayer* [10]. Customers can choose to develop their own players and codec(s) compatible with VAAPI v0.29 supported by the Intel® Embedded Graphics Driver. While developing your own codec(s) you should make sure codec(s) are compliant to ISO/ITU standards, which define the syntax, semantics and rules for developing codec(s) in order to be able to decode media content conforming to these standards.

Windows* XP

Cyberlink* PowerDVD Player for the Intel® Atom™ Processor with the Intel® SCH US15W chipset platform that can be used to hardware-accelerate DVD and stored media file playback for Intel® SCH US15W-supported video standard formats via DXVA v1.0. Microsoft* Windows* Media Player 11 comes bundled with VC-1/WMV9 codec that uses the MC entry point for VC-1 hardware acceleration. As mentioned earlier, the MPEG-4 hardware accelerated decoding is currently not supported in the player and the graphics driver. If you choose to develop your own players and codec(s) compatible with DXVA1.0, they should be compliant with Microsoft* DXVA1.0 specification and ISO/ITU CODEC standards.

Windows* CE

The Windows* CE media player is used for Intel® Embedded Graphics Driver testing with Direct Show Filters for MPEG-4, MPEG-2 and H.264 provided by Intel. These Direct Show Filters are provided to customers under evaluation license along with the Intel® Embedded Graphics Driver. Customers can choose to develop their own player and codec(s) for going to production under which case they should be compliant with ISO/ITU CODEC standards.

See Table 3 which describes operating systems, players, codec(s) against which the Intel® Embedded Graphics Driver v 10.1 Gold* has been validated.

Table 3. List of Media Players and Codec(s) Tested with the Intel® Embedded Graphics Driver

OS	Player and Codec Combination	Standards	Entry Point
Linux*	RP4NB v 1.1 Gold with codec(s)	MPEG4/VC-1/H.264/WMV9/MPEG-2[1]	VLD
Linux	splay-plugin-atlas-01.2.0 with Menlow codec 1.8.8.22	MPEG4/VC-1/H.264/WMV9/MPEG-2	VLD
Linux	Helix-player-1.0.9 with Menlow codec 1.8.8.22	MPEG4/VC-1/H.264/WMV9/MPEG-2	VLD
Linux	Mplayer with FFMPEG codec [10]	MPEG4/VC-1/H.264/WMV9/MPEG-2	VLD
Windows* XP	PowerDVD Ultra 8.0 with codec version patch 2810a	H.264/MPEG-2	VLD
Windows XP	Windows Media 11 with VC-1 codec	VC-1/WMV9	MC
Windows* CE	Windows CE Player with IEGD Direct Show Filters	MPEG-4/H.264/MPEG-2	VLD

NOTES:

1. RealNetworks* provides MPEG-2 codec as an option.

Summary

This paper described the need for hardware accelerated video decode in embedded devices. We discussed the Intel® SCH US15W chipset hardware video decode capabilities in detail. This paper familiarizes you with the video decode software stack flow under multiple operating systems architecture. We also concluded that observed video decode system performance is dependent on a multitude of variables and this paper revealed those hidden parameters.

Under the circumstances where video playback performance does not reach expected levels, you should first troubleshoot the problem using some of the steps outlined below:

- Experiment with the same video clip on a higher performance platform and ensure video quality issues like video corruption and blocking artifacts are not embedded in the video clip as it was originally encoded.
- Analyze the video clip using media analyzer tools and make sure the video codec/profile/levels of the media clip are compatible with the hardware-supported feature set.
- Ensure the choice of media player is able to decode the container format of interest and that you have a host decoder software or codec(s) installed which can talk to graphics driver through video acceleration API(s) or video acceleration framework specific to the operating system under consideration.
- Be aware that video quality does not improve with a higher video profile. Video quality also depends on the bit rate of the stream and how the encoder encoded it. Always look for obvious errors like video corruption, blocking artifacts, etc. before calling it a graphics driver issue.
- Also be aware that CPU utilization is subjective and varies from one OS to the other. Combined with OS architecture, other factors play a significant role in CPU utilization such as software vs. hardware-accelerated video decode, VLD vs. MC video processing offloading, use of hardware video overlay, etc.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel’s tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. http://intel.com/embedded/edc.

References

1. ITU-T Rec. H.264 / ISO/IEC 11496-10, "*Advanced Video Coding*", Final Committee Draft, Document JVTE022, September 2002

2. ISO/IEC 14496-2, *"Information Technology – Coding of audio-visual objects – Part2: Visual"*, 3rd Edition, 2004

3. ISO/IEC 13818-2, "*Information Technology – Generic coding of moving pictures and associated audio information: video*", 1995

4. SMPTE 421M, "*VC-1 Compressed Video Bitstream Format and Decoding Process*", Final Draft SMPTE Standard, 2006

5. Srinivasan et. al, "*Windows Media Video 9: overview and applications",* pp. 851-875*,* Signal Processing: Image Communication, vol. 19, October 2004

6. "*Comparison of Container Formats",* <http://en.wikipedia.org/wiki/Comparison_of_container_formats>

7. "*X.org Foundation",* <http://www.x.org/wiki/>

8. *ATSC Implementation Subcommittee Finding: Relative Timing of Sound and Vision for Broadcast Operations* (2003). (Doc. IS-191, 26 June, 2003)

9. "*Real Player for Mobile Devices",* <https://helixcommunity.org/products/realplayer_for_mobile_devices/>

10. Splitted-desktop patches to add VA API support to FFMPEG and MPlayer, <http://www.splitted-desktop.com/~gbeauchesne/mplayer-vaapi/>

11. Video Acceleration API, <http://www.freedesktop.org/wiki/Software/vaapi>

12. "*Introduction to DirectX VA",* <http://msdn.microsoft.com/en-us/library/ms799545.aspx>

13. "*Video Mixing Renderer Filter 7",* <http://msdn.microsoft.com/en-us/library/dd407343(VS.85).aspx>

14. "*Video Mixing Renderer Filter 9",* <http://msdn.microsoft.com/en-us/library/dd407344(VS.85).aspx>

Acronyms

API	Application Programming Interface
A/V	Audio/Video
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
DDI	Device Driver Interface
DLL	Dynamic Link Library
DXVA	DirectX Video Acceleration
FSB	Front Side Bus
GMA	Graphics Media Accelerator
GPU	Graphics Processing Unit
HD	High Definition
ICH	I/O Controller Hub
IDCT	Inverse Discrete Cosine Transform
MC	Motion Compensation
MCH	Memory Controller Hub
PCI	Peripheral Component Interconnect
RP4MID	RealPlayer for Mobile Internet Device
RP4NB	RealPlayer for Notebook
SCH	System Controller Hub
SDIO	Secure Digital Input Output
USB	Universal Serial Bus
VAAPI	Video Acceleration API
VLD	Variable Length Decoding
VMR	Video Mixing Renderer

Author

Abhishek Girotra is a Graphics Software Technical Marketing Engineer with Embedded and Communications Group (ECG) at Intel Corporation.