FFMPEG QSV

Legal Notices and Disclaimers

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. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

Intel may make changes to specifications and product descriptions at any time, without notice.

Intel, processors, chipsets, and desktop boards 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.

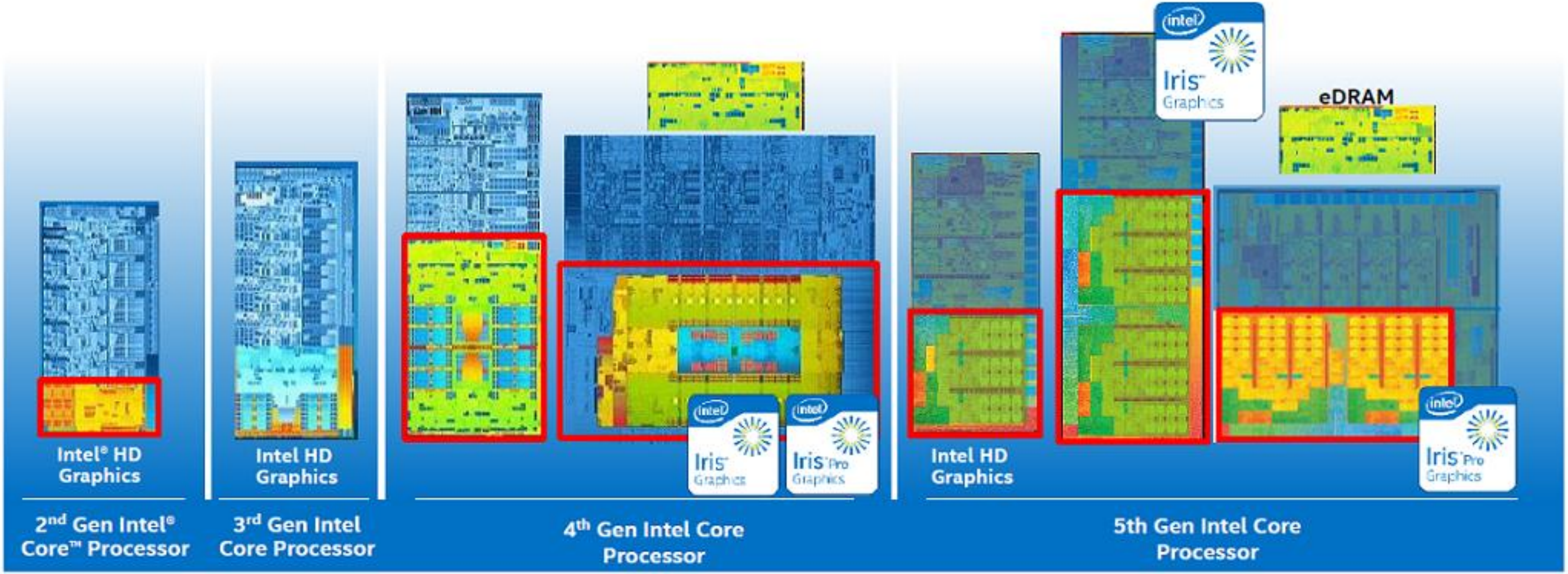
Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://intel.com.

Some results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

Copyright © Intel Corporation 2017

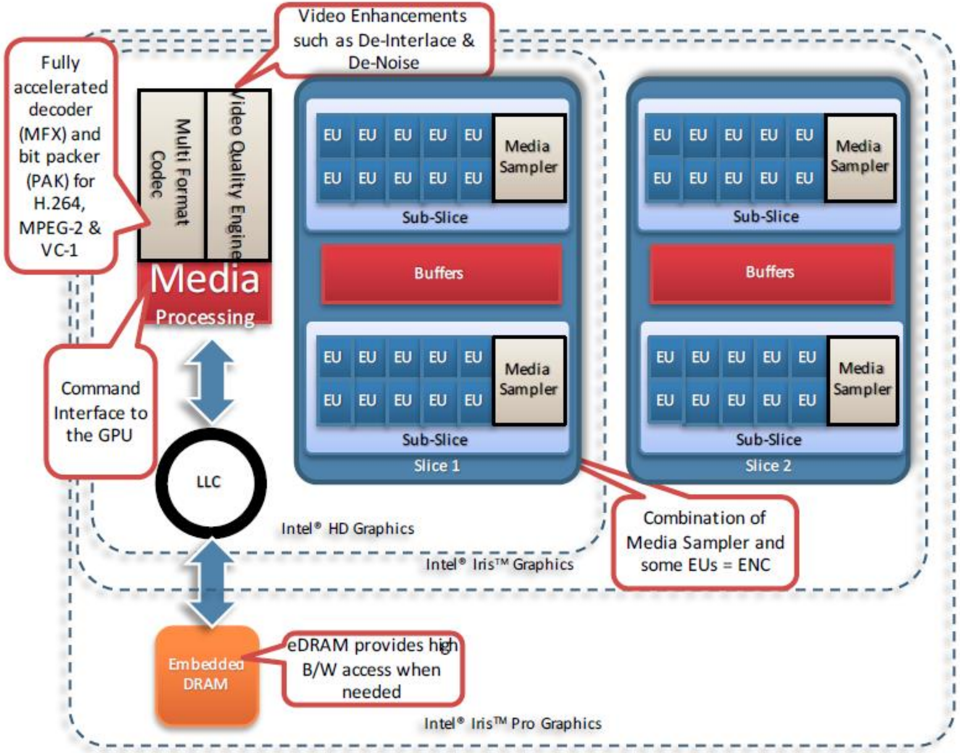
Graphics is a Key Investment and Still Growing



Graphics is Key Part of Everyone's Future

Intel® HD Graphics/Iris™ Graphics Architecture

- Starting with the Ivy Bridge family of Intel® Core™ processors, Intel® HD Graphics moved to a sliced architecture.
  - Each slice contains two sub-slices of Execution Units (EUs) and a media sampler with shared buffers/memory between them.
- Intel® HD Graphics (GT2) has a single slice.
- Intel® Iris™ Graphics (GT3) adds another full slice.
- Intel® Iris™ Pro Graphics (GT3e) adds dedicated eDRAM on-chip to provide very fast caching of textures in the GPU.
- Intel® Iris™ Graphics (GT4e) has 3 full slices with eDRAM.



Fully accelerated decoder (MFX) and bit packer (PAK) for H.264, MPEG-2 & VC-1

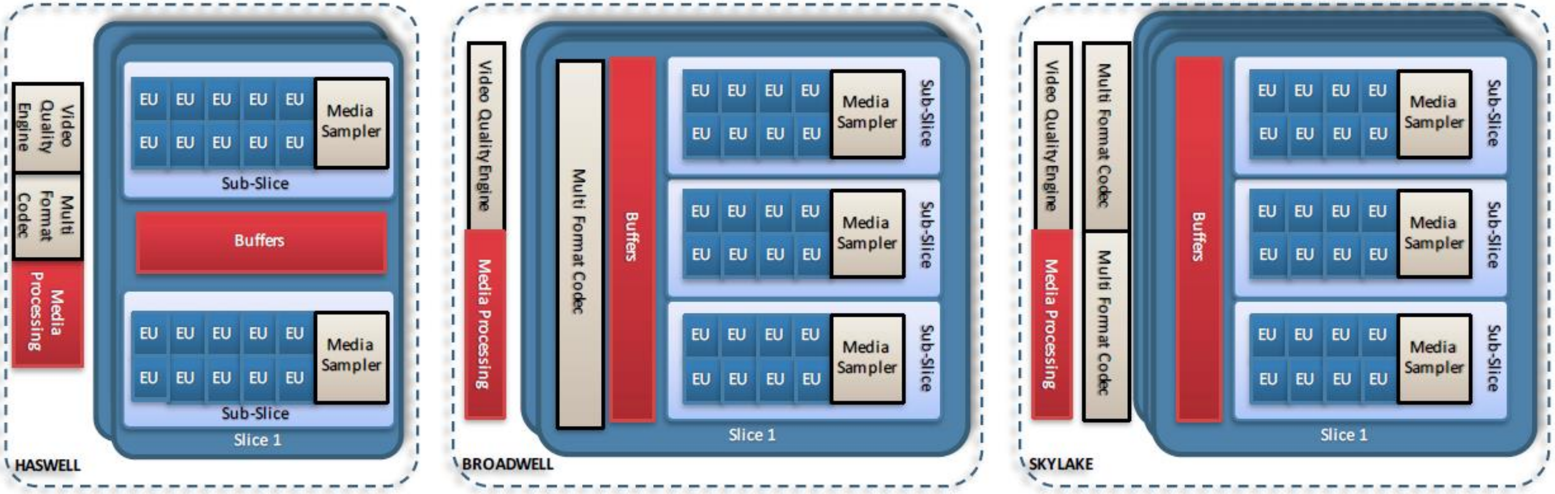
Video Enhancements such as De-Interlace & De-Noise

Command Interface to the GPU

Combination of Media Sampler and some EUs = ENC

eDRAM provides high B/W access when needed

Haswell to Skylake GPU change



Broadwell increases the number of sub-slices per slice from 2 to 3

- Total EU count increases from 40 in Haswell Gt3 in Broadwell GT3
- Provides 50% more Motion Engine capability per Slice
- Increases performance at higher quality modes
- Additional Multi Format Codec (in second slice increases decoder and entropy coding capacity for certain SKUs
- Hardware VP8 Decoder*

SkyLake adds 3rd Slice

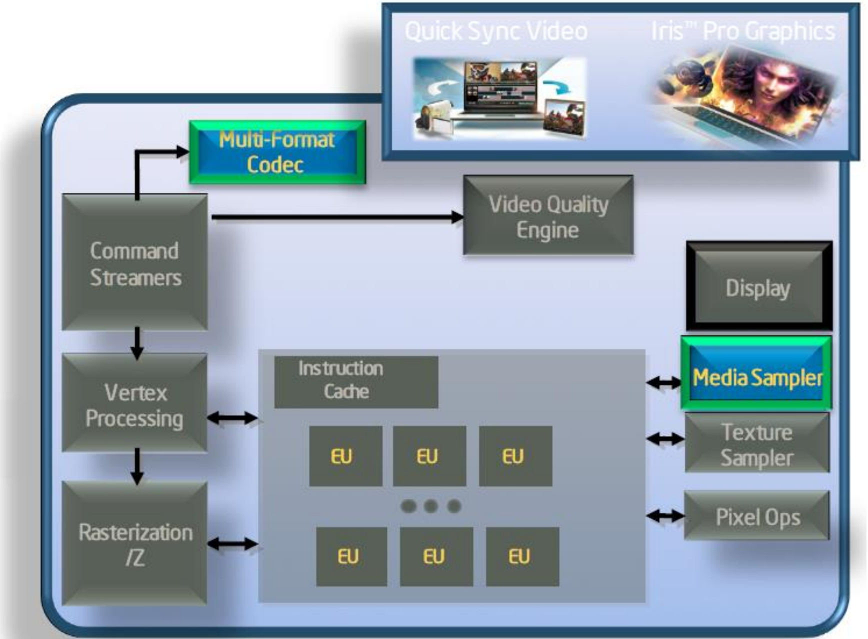
- Total EU count increases to 72
- 100%+ increase in Motion Engine capability over Haswell
- Hardware Decoder for HEVC
- PAK updated for HEVC encoder CABAC**
- Multi-Format Legacy decoder (MFL)** Hardware for MPEG4p2, DivX, Xvid, AVS and MPEG2/1

* Subject to software enabling planning

* *Only available in first Multi Format Codec

Quick Sync Video Acceleration

- Unique 2-Stage approach to deliver high performance and flexibility
- 7 Encoder Target Usages (TU) to deliver flexible performance and quality
- Additional optimization for video conferencing and screen content quality
- New single-stage low power low latency streaming encoding to support Wireless Display, Video conferencing, camera recording usages*



Quick Sync Video

Iris™ Pro Graphics

Multi-Format Codec

Command Streamers

Video Quality Engine

Display

Vertex Processing

Instruction Cache

EU EU EU

EU EU EU

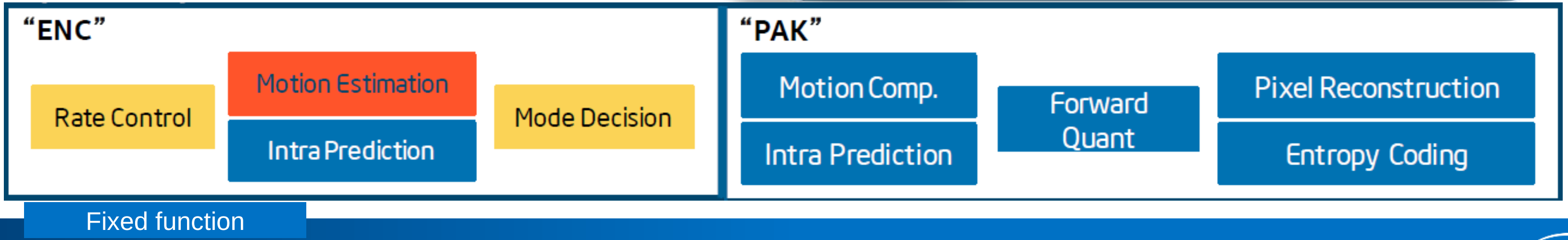
Media Sampler

Texture Sampler

Pixel Ops

Rasterization /Z

* Not in all SKUs



"ENC"

Rate Control

Motion Estimation

Intra Prediction

Mode Decision

"PAK"

Motion Comp.

Intra Prediction

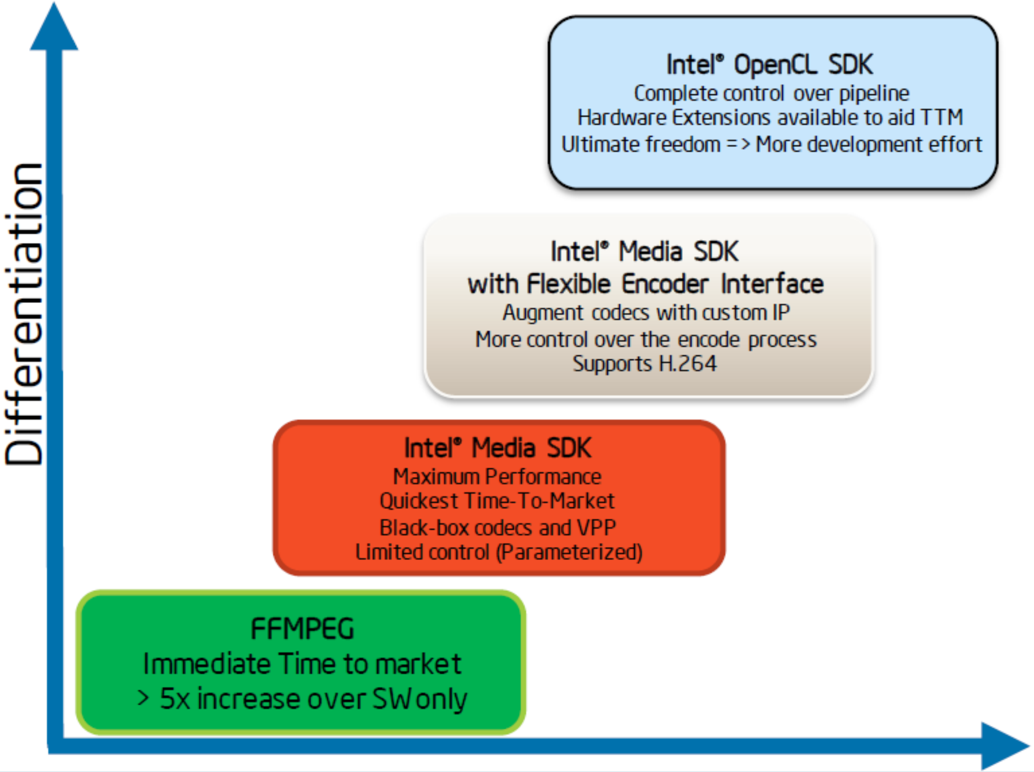
Forward Quant

Pixel Reconstruction

Entropy Coding

Fixed function

Software Strategy


Differentiation
Intel® OpenCL SDK
Complete control over pipeline
Hardware Extensions available to aid TTM
Ultimate freedom => More development effort
Intel® Media SDK
with Flexible Encoder Interface
Augment codecs with custom IP
More control over the encode process
Supports H.264
Intel® Media SDK
Maximum Performance
Quickest Time-To-Market
Black-box codecs and VPP
Limited control (Parameterized)
FFMPEG
Immediate Time to market
> 5x increase over SW only

There are several ways to harness the power of Intel HD Graphics:

– **FFMPEG**

The industry leading OpenSource Media Encoder now QSV enabled in the Main Branch (as of release 2.8)

– **Intel Media SDK**

Our time-to-market solution for customers where the featureset meets their requirements.

– **Intel Media SDK with Flexible Encoder Interface**

The FEI extension to the Intel Media SDK API provides low-level hooks into the H.264 encodr pipeline for more feedback and control. Designed for customers who need to tweak/augment the encoder process.

– **Intel OpenCL SDK**

The Intel OpenCL SDK enables customers to offload portions of their own codec/video fitler implementations to the GPU. This enables programmable access to the Executions Units (Eus) and other GPU blocks via OpenCL extensions

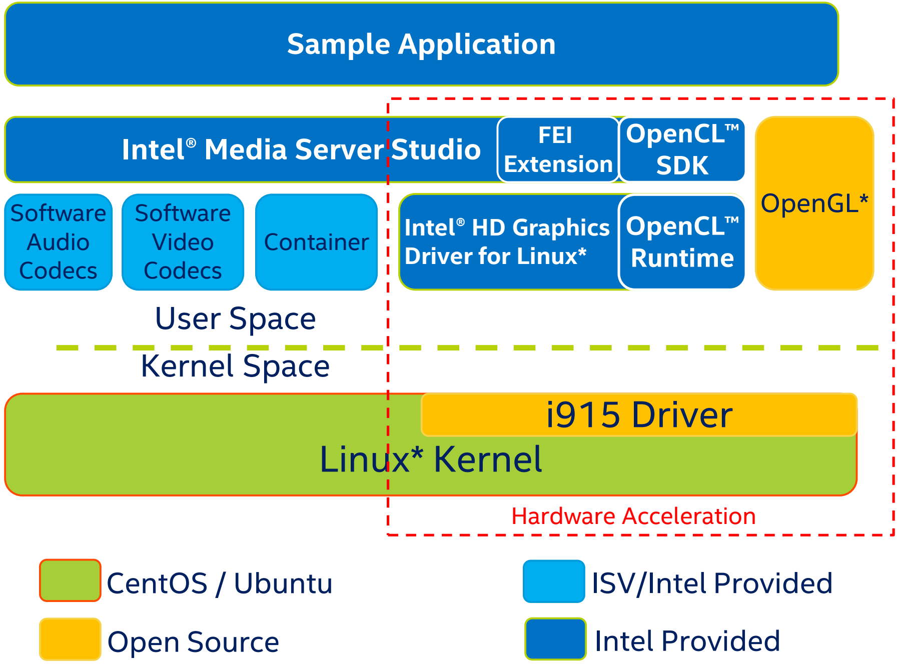
Intel® Media Server Studio

Media Tools	Community Edition	Essentials Edition	Professional Edition
Intel® Media SDK	✓	✓	✓
Media and Graphics Drivers	✓	✓	✓
Code Samples	✓	✓	✓
OpenCL TM Code Builder	✓	✓	✓
Metrics Monitor – Linux Only	✓	✓	✓
Intel® Premier Support		✓	✓
HEVC Decoder & Encoder			✓
Audio Decoder & Encoder			✓
Video Quality Caliper Tool			✓
Intel® VTune™ Amplifier XE			✓
Premium Telecine interlace Reverser			✓
GPU Assist API			✓

Media Solutions Portal: https://software.intel.com/en-us/media-solutions-portal

Media Server Studio Software Stack

(Linux Stack Shown)*



Sample Application

Intel® Media Server Studio

FEI Extension

OpenCL™ SDK

OpenGL*

Software Audio Codecs

Software Video Codecs

Container

Intel® HD Graphics Driver for Linux*

OpenCL™ Runtime

User Space

Kernel Space

i915 Driver

Linux* Kernel

Hardware Acceleration

🟩 CentOS / Ubuntu

🟨 Open Source

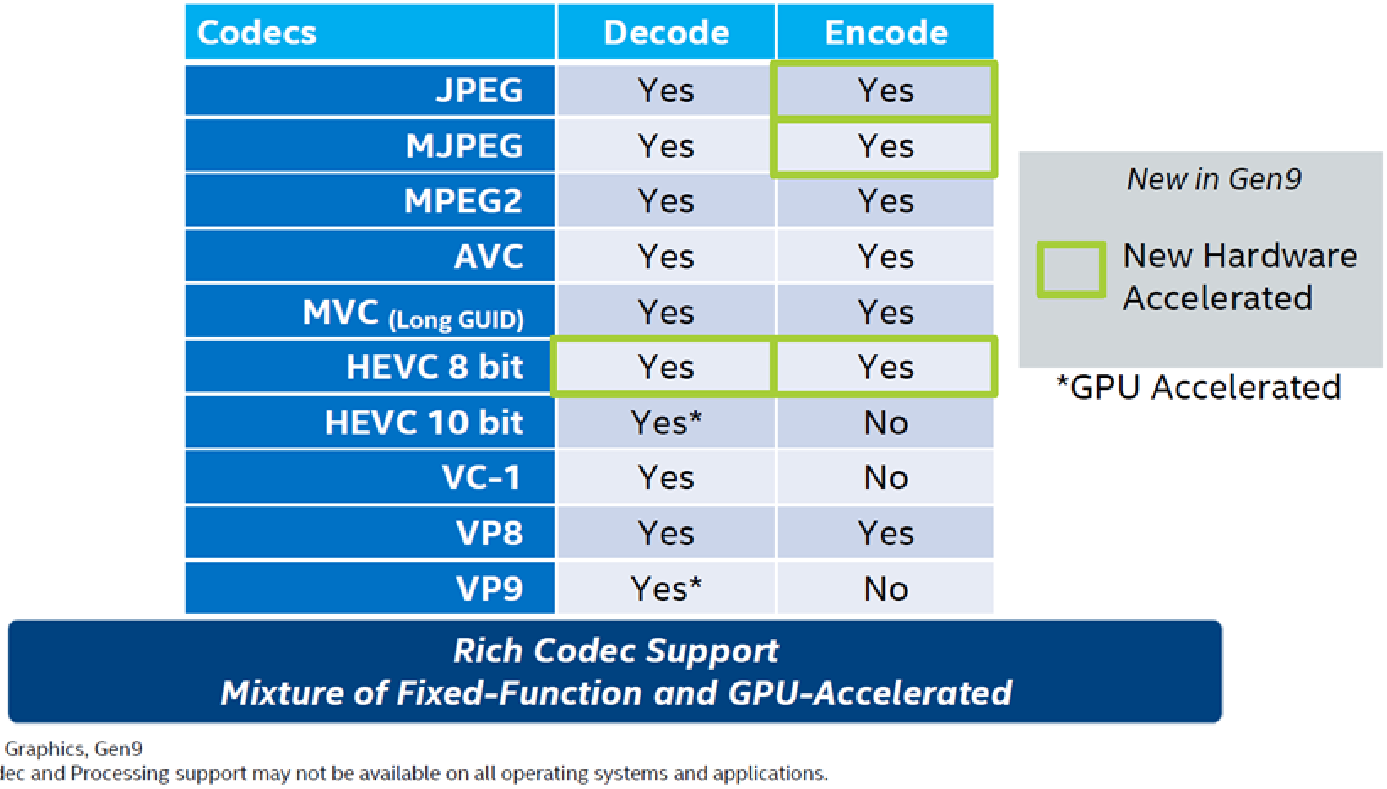
🟦 ISV/Intel Provided

🟦 Intel Provided

- **Modular Intel® Media Server Studio architecture**
  - Pluggable infrastructure for third-party codecs and containers
- **Flexible Encoder Infrastructure (FEI)**
  - Flexible encode pipeline enables differentiation and finer control of AVC encoding
- **Codecs**
  - HEVC (SW), AVC, MPEG-2, JPEG/MJPEG decode
- **Video Pre Processing**
  - Deinterlacing, Resizing, Cropping, Composition and Alpha Blending, Color Conversion, Denoising, Frame-rate conversion
- **OpenCL™ and OpenGL***
  - Allows for pixel processing in the middle of the transcode pipeline via OpenCL™ kernels
  - Leverage GPGPU capabilities to implement custom codecs/filters
  - Zero-copy surface sharing between Intel® Media Server Studio and OpenCL™/OpenGL*
- **Virtualization**
  - GVT-d & GVT-g supported for Xen today and KVM soon.

Note that some features are unique to certain generation of Intel processors

Codec Support



Codecs	Decode	Encode
JPEG	Yes	Yes 🟩
MJPEG	Yes	Yes 🟩
MPEG2	Yes	Yes
AVC	Yes	Yes
MVC (Long GUID)	Yes	Yes
HEVC 8 bit	Yes 🟩	Yes 🟩
HEVC 10 bit	Yes*	No
VC-1	Yes	No
VP8	Yes	Yes
VP9	Yes*	No

New in Gen9

🟩 New Hardware Accelerated

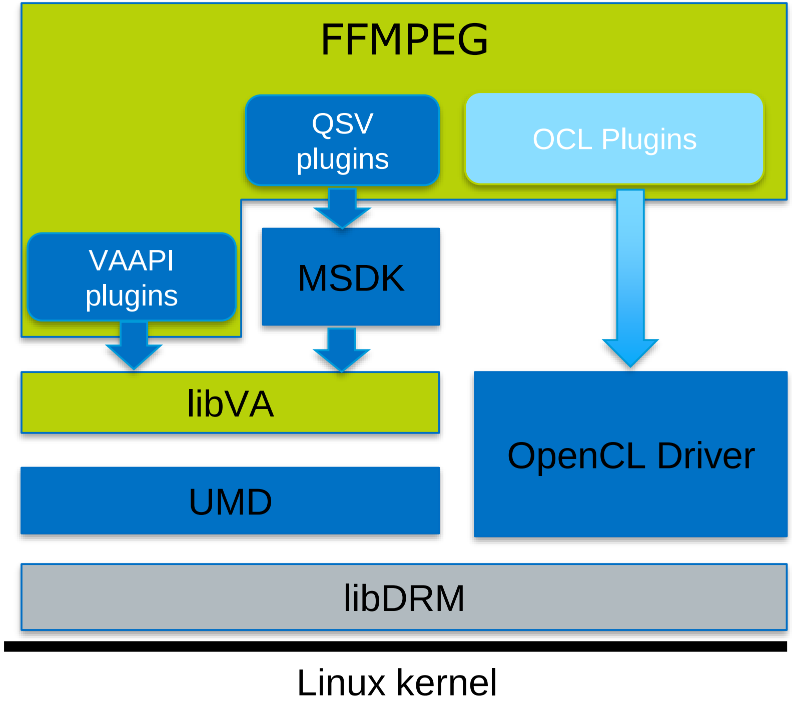
*GPU Accelerated

Rich Codec Support
Mixture of Fixed-Function and GPU-Accelerated

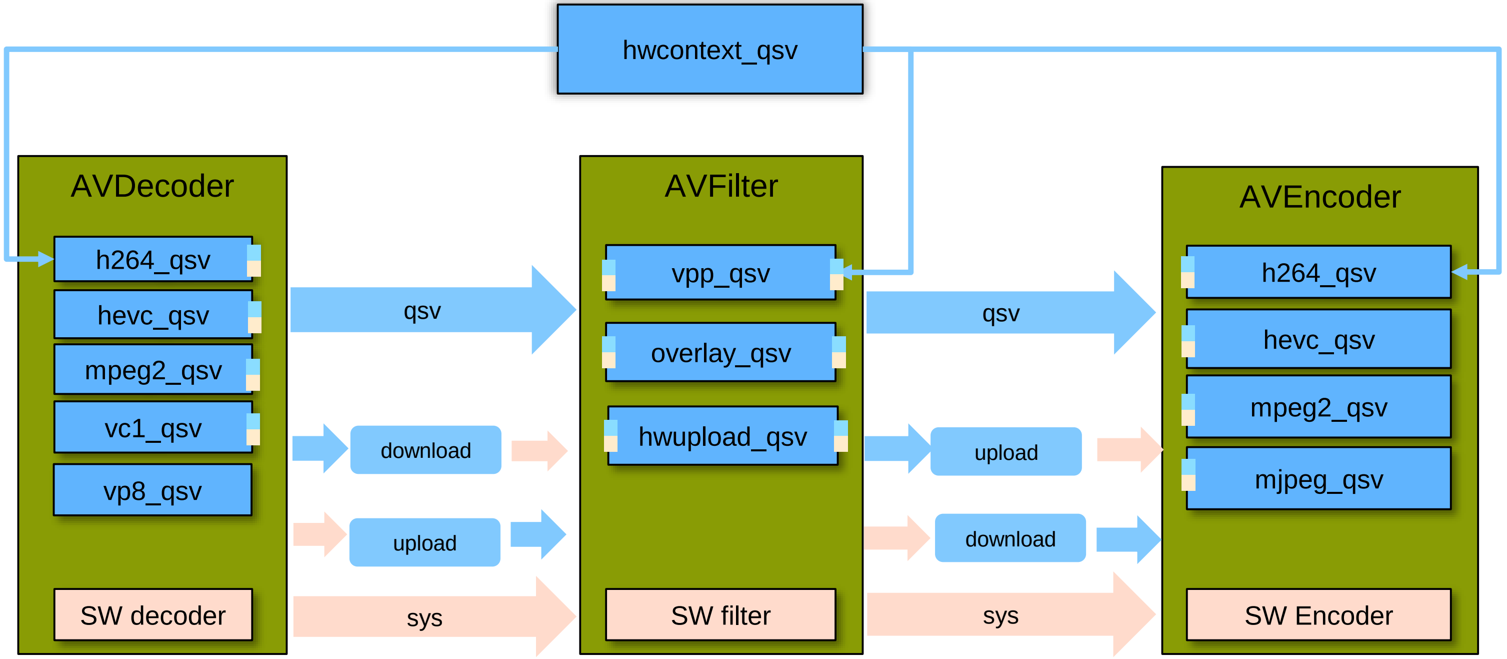
Intel® Processor Graphics, Gen9
Note: Media Codec and Processing support may not be available on all operating systems and applications.

FFMPEG + Intel Media Solution

- FFmpeg is most popular open source framework; it can help deploy Intel media solution in a quickest way
- FFmpeg QSV plugins are based on MSS; it has widely accepted by customers.
- VAAPI is lower level API; FFMPEG VAAPI plugins provides more flexible solution for customers.
- Integrate 3rd-party OCL video processing Library to enrich the solution.



Integrate MSDK into FFMPEG



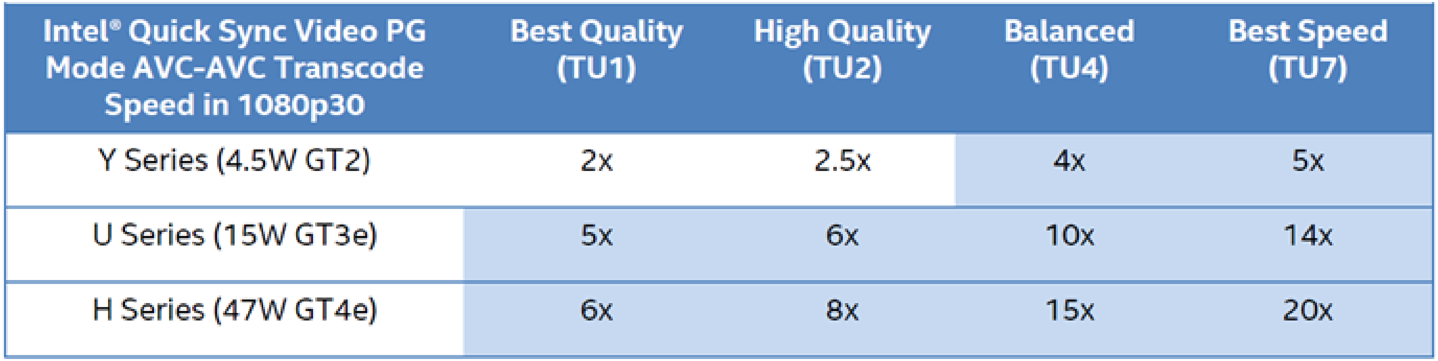
Note:

- hwcontext_qsv will manage the devices context, video surface allocation, format translation (download & upload which must be config explicitly)
- "-hwaccel qsv" must be added when using pure HW pipeline, and must make sure all modules is qsv-enabled.
- Arrow means YUV buffer

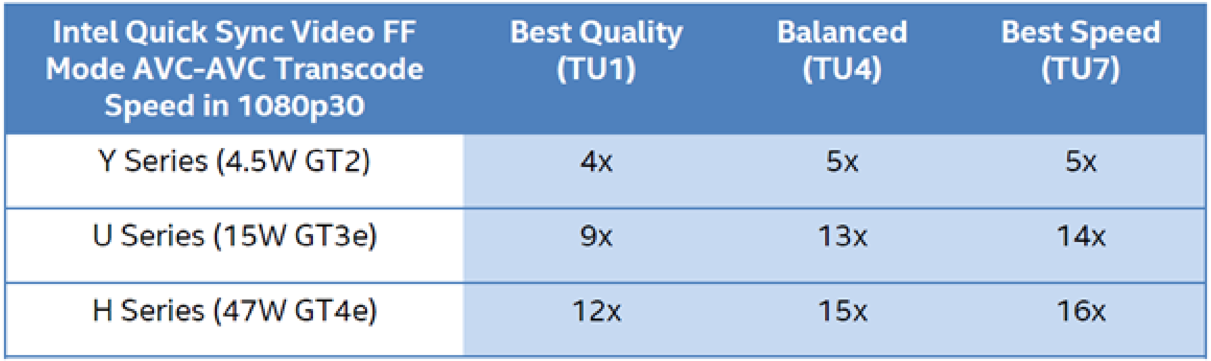
FFMPEG+QSV VS. MSS

	MSS	FFMPEG + QSV
Deliverables	MSS delivers a set of libraries and tools; user have to secondary development based on it. And it is only for video.	FFMPEG is a popular multimedia open framework, QSV is only part of them with GPU acceleration. It provides uniform interfaces and executable for user.
Supported Codec	Basically, FFMPEG QSV integrated all codecs supported (listed in previous page) by MSS into FFMPEG.	
Codec Features*	FFMPEG QSV supported feature has been listed in following pages. FFMPEG QSV haven't enable all of them supported by MSS which mostly needs to be configure runtime such as MBQP, etc. For long-term plan, FFMPEG QSV modules can make use of this feature to enhance current implementation.	
Video Processing	Provides VPP interface in library; secondary development is must to integrate the features.	Currently, 2 filters are developed; and all of features supported by MSS are integrated in these filters and exposed to user by options. FFMPEG QSV has secondary development based on MSS, e.g. overlay_qsv filter.
Memory Management	Developer must manage its own Memory	FFMPEG provides basic memory management utils: uniform implementation for system memory, hwcontext architecture for hw accelerator memory
Stream compatible	User must deal with it in their own application based on MSS API returns.	FFMPEG provide error handling, a/v sync mechanism based on the framework. FFMPEG QSV modules also have extra work to compatible with such FFMPEG's mechanism; To improve compatibility, FFMPEG QSV module will use SW module to parse stream in some case.
Pipeline	User must implement their own mux/demux or other necessary modules to work with MSS modules	FFMPEG QSV is implemented based on MSS and add special logical to make each module can work with FFMPEG's other modules. Add support for different transcoding usage such as 1:N;
Perf & quality	Almost the same performance & quality between MSS sample application and ffmpeg qsv; see FFMPEG QSV perf /quality data in next pages	

SKL Gen9 AVC Transcode Capability



Intel® Quick Sync Video PG Mode AVC-AVC Transcode Speed in 1080p30	Best Quality (TU1)	High Quality (TU2)	Balanced (TU4)	Best Speed (TU7)
Y Series (4.5W GT2)	2x	2.5x	4x	5x
U Series (15W GT3e)	5x	6x	10x	14x
H Series (47W GT4e)	6x	8x	15x	20x

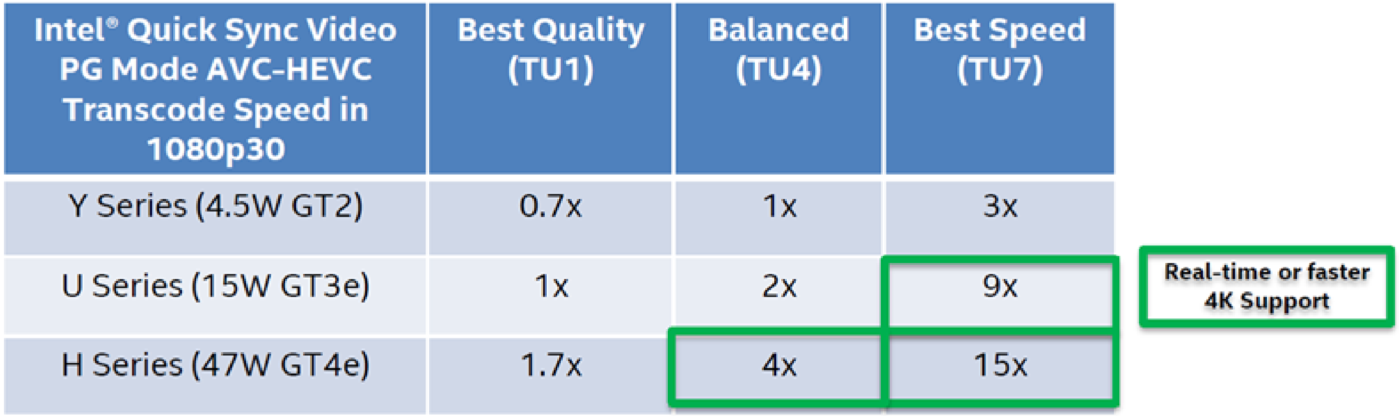


Intel Quick Sync Video FF Mode AVC-AVC Transcode Speed in 1080p30	Best Quality (TU1)	Balanced (TU4)	Best Speed (TU7)
Y Series (4.5W GT2)	4x	5x	5x
U Series (15W GT3e)	9x	13x	14x
H Series (47W GT4e)	12x	15x	16x

Real-time or faster 4K Support

SKL Gen9 HEVC Transcode Capability

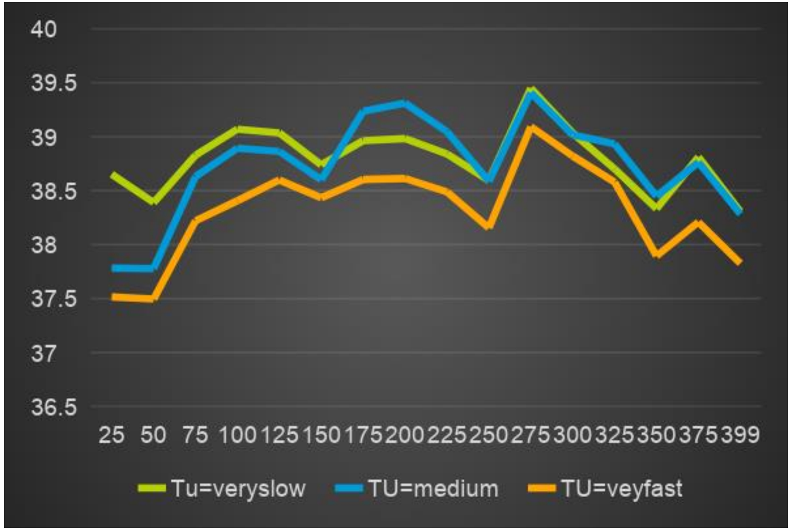
- Quality optimized for HD 1080p
- Tuned to enable 4Kp60 encode speed



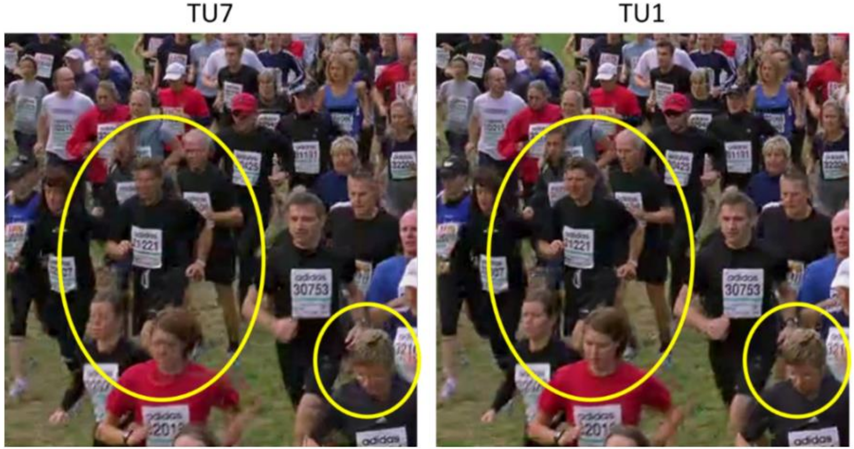
Intel® Quick Sync Video PG Mode AVC-HEVC Transcode Speed in 1080p30	Best Quality (TU1)	Balanced (TU4)	Best Speed (TU7)
Y Series (4.5W GT2)	0.7x	1x	3x
U Series (15W GT3e)	1x	2x	9x
H Series (47W GT4e)	1.7x	4x	15x

Real-time or faster 4K Support

Quality: FFMPEG + QSV Integration



PSNR Comparison



Subject Quality Comparison

- ❑ Preset behavior is highly input depended.
- ❑ TU=Medium is a good balance, relatively small quality loss for ~2x speedup.
- ❑ The quality loss for TU=veryfast is large relative to the speed improvement vs TU1/veryslow
- ❑ The subject quality of veryslow show great improvement compared to veryfast

Source Code and Build

Source Code

- Github: *git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg*

BUILD

- Check whether MSS is installed correctly: vainfo, and MSDK samples(sample_decode, sample_encode) can works well.
- Configure with “--enable-libmfx”
- Details as : https://github.com/Intel-FFmpeg-Plugin/Intel_FFmpeg_plugins/wiki/Intel-FFmpeg-QSV-Plugins#prerequisite

Usage: Decoder

Supported Codec: H264, HEVC, VP8, VC1

Supported Options:

Options	Description	Values	codec
async_depth	Internal parallelization depth, the higher the value the higher the latency. Higher value with performance improvement	Default is 4.	All supported decoder
load_plugin	A user plugin to load in an internal session. The value is ignored if load_plugins isn't set	none hevc_sw hevc_hw Default is hevc_hw on Linux	Hevc only
load_plugins	A :-separate list of hexadecimal plugin UIDs to load in an internal session	The value can be found in /opt/intel/mediasdk/plugins/plugins.cfg	Hevc only

Example:

```
./ffmpeg -c:v hevc_qsv -load_plugin hevc_hw -i out-hevc.mp4 -c:v hevc_qsv -b:v 5M -maxrate 5M -load_plugin hevc_hw -c:a copy -y out-hevc2.mp4
```

Encoder Usage

CBR	./ffmpeg -hwaccel qsv -c:v h264_qsv -y -i input.stream -c:v h264_qsv –b:v 2M –maxrate 2M out.stream
VBR	./ffmpeg -hwaccel qsv -c:v h264_qsv -y -i input.stream -c:v h264_qsv –b:v 2M –maxrate 5M out.stream
Look Ahead	./ffmpeg -hwaccel qsv -c:v h264_qsv -y -i input.stream -c:v h264_qsv –b:v 2M -look_ahead 1 –look_ahead_depth 40 -look_ahead_downsampling 2x out.stream
H264_qsv encoder examples	./ffmpeg -hwaccel qsv -c:v h264_qsv -y -i input.stream -c:v h264_qsv -preset veryfast -profile main -level 3.0 -g 30 -bf 4 -BRefControl bRefPyramid -refs 12 -look_ahead 1 -global_quality 30 -maxrate 300k -rc_init_occupancy 0 -f mp4 -r 30 encoding1.mp4
mjpeg_qsv examples	./ffmpeg -hwaccel qsv -c:v mjpeg_qsv -y -i input.stream -c:v mjpeg_qsv -quality 80 out.stream
hevc_qsv examples	Default HW plugin: ./ffmpeg -c:v hevc_qsv -y -i input.stream -c:v hevc_qsv -b:v 3M out.stream sw: ./ffmpeg –c:v hevc_qsv –load_plugin hevc_sw –i input.stream -c:v hevc_qsv –load_plugins [GUID] –b:v 3M out.hevc

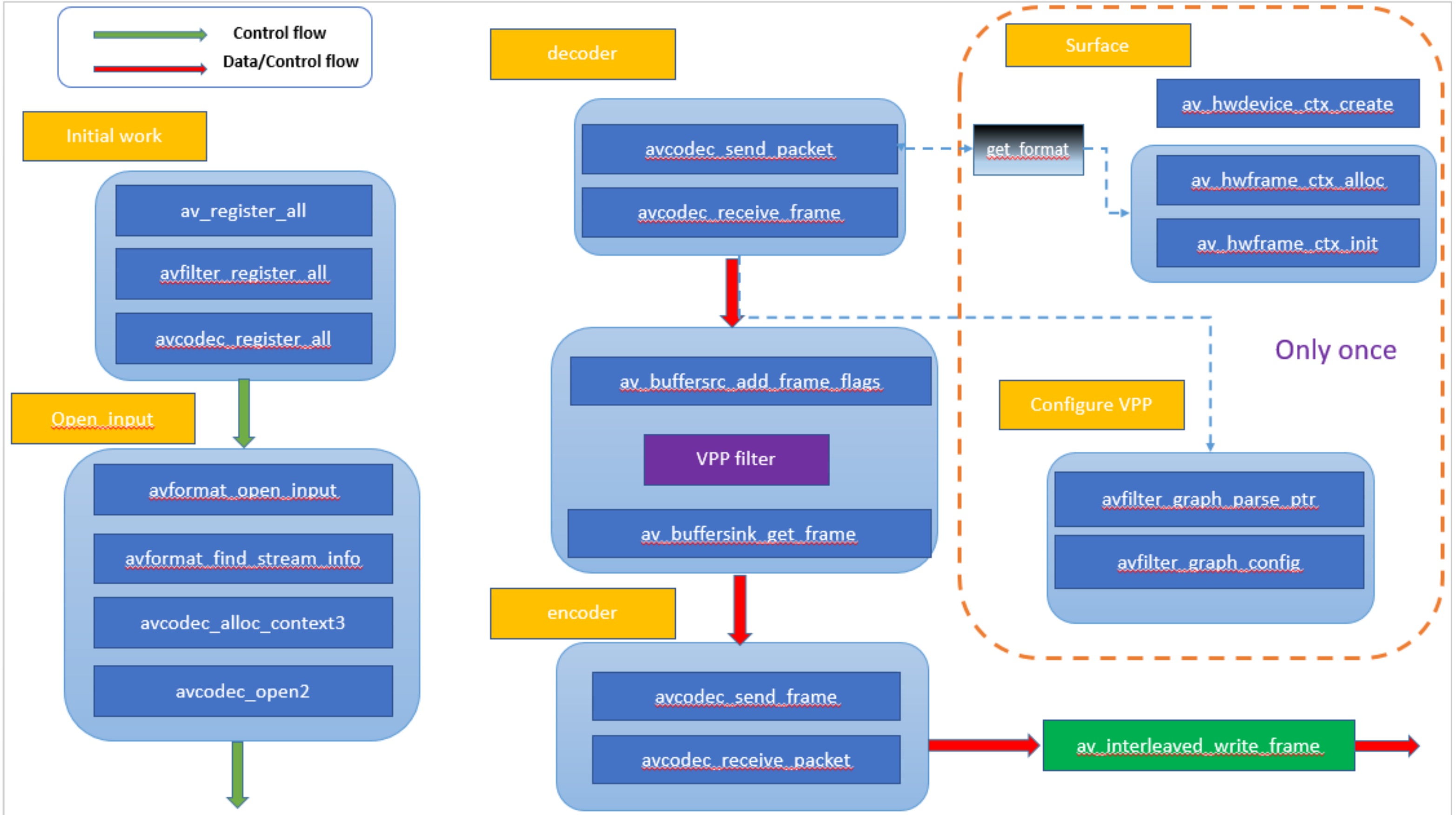
VPP USAGE:

QSV VPP filter	
sw decoder + mfxvpp + sw encoder	ffmpeg -i src.stream -vf vpp_qsv=deinterlace=1:w=iw/2:h=ih/2:framerate=2997/100 -y out.mp4
sw decoder + mfxvpp + mfxenc (partial video memory)	ffmpeg -i src.stream -vf hwupload_qsv=nv12,vpp_qsv=framerate=25 -c:v h264_qsv -y out.mp4
video memory pipeline	ffmpeg -hwaccel qsv -c:v h264_qsv -i src.stream -vf vpp_qsv=deinterlace=1 -c:v h264_qsv -y out.mp4
video memory mfxdec + mfxvpp + sw encoder	ffmpeg -i src.stream -filter_complex “movie=logo.png[a];[0:v][a]overlay_qsv=x_expr=10:y_expr=10” -y out.mp4
QSV Overlay filter	
video memory mfxdec + mfxvpp + sw encoder	ffmpeg -hwaccel qsv -c:v h264_qsv -i src.stream -vf vpp_qsv=w=iw/2,hwdownload,format=nv12 -c:v libx264 -y out.mp4
video memory pipeline, one input is system-mem (mostly for picture), another video in video-memory	ffmpeg -hwaccel qsv -c:v h264_qsv -i src.stream -filter_complex “movie=logo.png,hwupload[a];[0:v][a]overlay_qsv=x_expr=10:y_expr=10” -c:v h264_qsv -y out.mp4 //
both input are using video memory	ffmpeg -hwaccel qsv -c:v h264_qsv -i src.stream -hwaccel qsv -c:v mpeg2_qsv -i logo.str -filter_complex “overlay_qsv=x_expr=10:y_expr=10” -c:v h264_qsv -y out.mp4 //
Multiple videos composition	ffmpeg -init_hw_device qsv -filter_hw_device qsv0 -c:v h264_qsv -i ~/Bird.mp4 -filter_complex "split=4,overlay_qsv=4:manual:x_expr=array(0\,W/2\,0\,W/2):y_expr=array(0\,0\,H/2\,H/2):w_expr=iw(0)/2:h_expr=ih(0)/2" -c:v libx264 -f null – ffmpeg -hwaccel qsv -c:v h264_qsv -i ~/Bird.mp4 -hwaccel qsv -c:v h264_qsv -i ~/Cars.mp4 -filter_complex overlay_qsv=layout=grid -c:v h264_qsv -an -y out.mp4

Other Usage

1:N transcoding	ffmpeg -hwaccel qsv -c:v h264_qsv -i src.stream -filter_complex vpp_qsv=framerate=25,split[a][b] -map [a] \ -c:v h264_qsv -y out1.mp4 \ -c:v mpeg2_qsv -y out2.mp4
	ffmpeg -hwaccel qsv -c:v h264_qsv -i src.stream -vf vpp_qsv=deinterlace=1 \ -c:v h264_qsv -y out_deinterlaced.mp4 \ -c:v mpeg2_qsv -map “0:v” –y recorded_mp2v.mp4
	./ffmpeg-3.2.2 -nostats -loglevel fatal -hwaccel qsv -c:v h264_qsv -y -i <infile> \ -c:v h264_qsv -vf vpp_qsv=w=468:h=468:framerate=30 -maxrate 400k -bufsize 800k -threads 1 -preset 7 -f null - \ -c:v h264_qsv -vf vpp_qsv=w=720:h=720:framerate=30 -maxrate 800k -bufsize 1600k -threads 1 -preset 7 -f null

QSV FFMPEG API Usage



Future Work

- Support more codecs, such as vp8 encoding, vp9 decoding/encoding
- More flexible solution:
  - Windows: dxva decoding + qsv vpp + qsv encoding
  - Linux: vaapi decoding + vaapi vpp + qsv encoding
- More features : HDR, 10 bit HEVC, 10 bit transcoding

Backup

reference

https://en.wikipedia.org/wiki/FFmpeg

git clone git://source.ffmpeg.org/ffmpeg.git ffmpeg

http://ffmpeg.org/

http://ffmpeg.org/developer.html#Contributing

http://ffmpeg.org/legal.html

http://www.oschina.net/question/tag/ffmpeg

Encoder: Common Options

Example to show qsv private options : ffmpeg --help encoder=h264_qsv

Encoder Parameters (QSV special)	
preset	Target Usage: veryfast / faster / fast / medium / slow / slower/ veryslow
async_depth	Maximum processing parallelism, default is 4
avbr_accuracy	Accuracy of the AVBR ratecontrol
avbr_convergence	Convergence of the AVBR ratecontrol
rdo	Enable rate distortion optimization, (0, 1)
max_frame_size	Maximum encoded frame size in bytes
max_slice_size	Maximum encoded slice size in bytes
bitrate_limit	Toggle bitrate limitation; default is on
mbbrc	Setting this flag enables macroblock level bitrate control that generally improves subjective visual quality; cannot work with LA.
b_strategy	Strategy to choose between I/P/B frames; 0: MFX_B_REF_OFF 1: MFX_B_REF_PYRAMID
cavlc	Enable CAVLC; default is 0

Encoder Parameters (Shared with)	
level	codec level
g	GOP size in frames
flags	AV_CODEC_FLAG_CLOSED_GOP Indicate using closed GOP AV_CODEC_FLAG_INTERLACED_DCT Force frame encoded as top-field-first AV_CODEC_FLAG_QSCALE Force to use QCP rate control mode
bf	Number of B frames between 2 I/P
slice	number of slices
refs	Number of reference frames to consider for motion compensation
b:v	bitrate in bits/s
maxrate	maximum bitrate in bits/s
rc_init_occupancy	number of bits which should be loaded into the rc buffer before decoding starts
i_quant_factor	QP factor between P- and I-frames
i_quant_offset	QP offset between P- and I-frames
b_quant_factor	QP factor between P- and B-frames
sample_aspect_ratio	sample aspect ratio

Encoder: h264 - To be changed

Encoder Options (1)	
idr_interval	Distance (in I-frames) between IDR frames
buffer_size	BufferSizeInKB represents the maximum possible size of any compressed frame
pic_timing_sei	Insert picture timing SEI with pic_struct_syntax element, default is 1
single_sei_nal_unit	Put all the SEI messages into on NALU
max_dec_frame_buffering	Maxumum number of frames buffered in the DPB; default is 0
look_ahead, look_ahead_depth	Use VBR algorithm with look ahead, if it is 1, look_ahead_depth should be set: [0 .. 100]. Default look_ahead is disabled
look_ahead_downsampling	This option controls down sampling in look ahead bitrate control mode; it will affect quality. [unknown, off, 2x], default is unknown
resetreflist	Set this flag to reset the reference list to non-IDR I-frames of a GOP sequence; [0,1] default is 0
refpicmarkrep	Set this flag to write the reference picture marking repetition SEI message into the output bitstream; [0,1] default is 0
fieldoutput	Set this flag to instruct the AVC encoder to output bitstreams immediately after the encoder encodes a field; [0,1] default is 0
audelimiter	Set this flag to insert the Access Unit Delimiter NAL; [0,1] default is 0
vuinalhrdparam	Set this flag to insert NAL HRD parameters in the VUI header; [0,1] default is 0
framepicture	Set this flag to encode interlaced fields as interlaced frames; [0,1] default is 0
int_ref_type	Intra refresh type; [none, vertical]
int_ref_cycle_size	Number of frames in the intra refresh cycle
recovery_point_sei	Insert recovery point SEI messages

Encoder Options(2)	
int_ref_qp_delta	QP difference for the refresh MBs
maxQPI, minQPI, maxQPP, minQPP, maxQPB, minQPB	maxumum/ minimum allowed QP value for I/P/B frame, valid range: 1-51; 0 is default value, no limitation on QP;cannot work with LA.
trellis	Trellis quantization; [off, I, P, B] default: off
profile	[unknown, baseline, main, high] default: unknown
a53cc	Use A53 Closed Captions (if available) ; [0,1] default is 0
repeatPPS	The default is on and set flag will off the repetition; [0,1] default is 0
numMbperslice	This option specifies suggested slice size in number of macroblocks
fixedframerate	This option sets fixed_frame_rate_flag in VUI; [0,1] default is 0
disableVUI	This option sets fixed_frame_rate_flag in VUI; [0,1] default is 0
bufferPeriodSEI	This option controls insertion of buffering period SEI in the encoded bitstrea
enableMAD	Turn ON this flag to enable per-frame reporting of MAD; [0,1] default is 0
userawref	Set flag to use raw frames for reference instead reconstructed frames; [0,1] default is 0
numSlicei	The number of slices for I
winmaxavg	Specifies the maximum bitrate averaged over a sliding window for MFX_RATECONTROL_LA/MFX_RATECONTROL_LA_HRD
winsize	Specifies sliding used for MFX_RATECONTROL_LA/MFX_RATECONTROL_LA_HRD window size in frames
qvbrquality	Specifies quality factor used for MFX_RATECONTROL_QVBR
direct_bias_adj	Set flag to enable the ENC mode decision algorithm to bias to fewer B Direct/Skip types; [0,1] default is 0
mv_cost_sf	MV cost scaling ratio; [0,3] default is 0
force_idr	If forcing key-frames, force them as IDR frames; [0,1] default is 0

Encoder: HEVC, mjpeg, mpeg2

Encoder Options (hevc)	
load_plugin	A user plugin to load in an internal session. The value is ignored if load_plugins isn't set; [none, hevc_sw, hevc_hw], default is hevc_hw
load_plugins	A :-separate list of hexadecimal plugin UIDs to load in an internal session
profile	[uknown, main, main10, mainsp] default is unknown

Encoder Options (mjpeg)	
quality	Specifies the image quality; [0, 100], default is 90

Encoder Options (MPEG2)	
profile	[uknown, simple, main, high] default is unknown

VPP: vpp_qsv

VPP_QSV Options	
async_depth	Maximum processing parallelism, default is 4
deinterlace	deinterlace mode: 0=off, 1=bob, 2=advanced
denoise	denoise level [0, 100]
detail	detail enhancement level [0, 100]
dpic	dest pic struct: 0=tff, 1=progressive [default], 2=bff
framerate	output framerate， double
procamp	
procamp	Enable ProcAmp
hue	ProcAmp hue, [-180.0, 180.0] default 0.0
saturation	ProcAmp saturation, [0.0. 10.0] default 1.0
contrast	ProcAmp contrast, [0.0. 10.0] default 1.0
brightness	ProcAmp brightness, [-100.0, 100.0] default 0.0
Crop	
cw	set the width crop area expression
ch	set the height crop area expression
cx	set the x crop area expression
cy	set the y crop area expression
Scale	
width / w	Output video width
height / h	Output video height

VPP: overlay_qsv To-be changed

overlay_qsv options	
nb_inputs	number of inputs
layout	Layout mode: grid, manual, overlay
x_expr	each win's x position
y_expr	each win's y position
w_expr	each win's width
h_expr	each win's height
a_expr	each win's alpha
w	Overlay width , default overlay_iw
h	Overlay height, default overlay_ih*w/overlay_iw
alpha	Overlay global alpha [0, 255], default 255
eof_action	Action to be taken when encountering EOF from overlay input, [repeat, endall], default is repeat

QEUSTIONS:

experience
what's inside™