

# Evolution of Hardware HEVC Encode on 10th Generation Intel® Core™ Processors

Achieve efficient, flexible, and faster media creation and distribution with high quality and high performance hardware encoder

## Authors

**Narifumi Iwamoto**

(narifumi.iwamoto@intel.com)

**Ragha Khandenahally**

(ragha.khandenahally@intel.com)

**Sonal Sharma**

(sonal.sharma@intel.com)

**Maggie Sun**

(maggie.sun@intel.com)

## Executive Summary

Video compression technologies play a key role in creation and distribution of high-quality video content. Cloud-based video distribution and video analytics workloads are growing exponentially. This paper provides the status quo of video encode quality of a key revision to Intel Codec IP available in 10th generation Intel® Core™ processors. The analysis includes video encode quality of HEVC<sup>1</sup> codecs using objective evaluation methodology.

## Challenges for Video Encoding

Video compression is a highly complex process defined by international standards. Software-based encoding takes a significant amount of time or consumes a lot of power, which has big impact on battery life in laptop or mobile use cases.

On the other hand, hardware-based encoding tends to have different challenges in terms of quality and configuration flexibility, as compared to software-based solutions. Intel has been a leader in striving for these difficulties for a decade starting with the 1st generation Intel® Core™ processor, and continuously improving encoding quality, performance, and configuration flexibility. In the 10th generation Intel Core processor, video quality of HEVC hardware encoding is dramatically improved by implementing new logic. This paper addresses HEVC hardware encoding quality evaluation delivered by 10th generation Intel Core processors by using industry standard methodology.

## Contents

- Executive Summary..... 1
- Challenges for Video Encoding ..... 1
- Quality Evaluation Methodology
  - Approach ..... 2
  - Target Encoders ..... 2
  - Configurations..... 2
  - Quantization Parameter..... 2
  - Video Sequences and Configurations..... 3
  - Tested Command Line and Configurations..... 3
- Quality Evaluation Result ..... 5
- Quality Evaluation Summary ..... 9

<sup>1</sup> G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T.

Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, December 2012.

## Quality Evaluation Methodology

### Approach

In order to conduct video quality assessment fairly, common conditions and software configurations need to be defined across encoders. We used standardized methodology that has been used by video coding standardization groups such as JCTVC.<sup>2</sup> Based on this methodology, we defined the following configurations and conditions and used HM Test Model<sup>3</sup> as an anchor encoder software.

### Target Encoders

We used three encoders for this quality assessment as shown in the following table.

Table 1: Target encoders

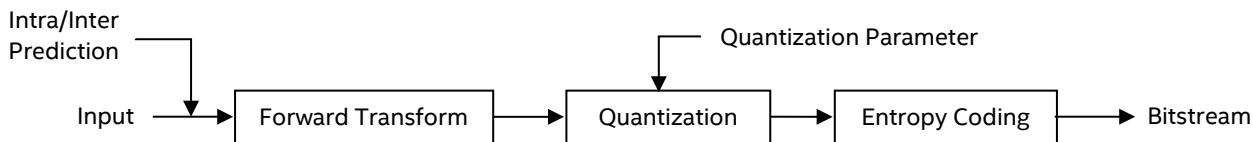
Target encoders	Components	Availability
HEVC Test Model 14.0 (HM14)	14.0	<a href="https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/archived/HM-14.0-dev/">https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/archived/HM-14.0-dev/</a>
10th generation Intel® Core™ processor and Intel® Media SDK sample encode	Intel® Core™ i7-1065G7 processor	
	Intel® Iris® Plus graphics driver 26.20.100.7372	<a href="https://downloadcenter.intel.com/">https://downloadcenter.intel.com/</a>
	Encoding sample version 8.4.27.31	<a href="https://github.com/Intel-Media-SDK/MediaSDK">https://github.com/Intel-Media-SDK/MediaSDK</a>
FFMPEG with libX265 library enabled	4.1.3 static	<a href="https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-4.1.3-win64-static.zip">https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-4.1.3-win64-static.zip</a> (2019-Apr-26 16:10)

### Configurations—Low Delay and Random Access

Two configurations are selected for this quality assessment to cover various encoding use cases. One is Low Delay (LD), which has forward reference only to cover low latency use cases such as video conferencing. In this configuration, frame re-ordering is not required at the time of decoding so that the encoder and decoder frame order matches. The other configuration is Random Access (RA), which includes both forward and backward references for better quality. Frame re-ordering is required at the decoding stage in this configuration.

### Quantization Parameter

Transformed compressed video signals are quantized by using the Quantization Parameter (Qp). Smaller Qp value indicates larger frame size; larger Qp value indicates smaller frame size.



We defined 10 different Qp values for all the intra frames of for this quality assessment to cover a wide range of bitrates. To avoid quality impact by rate controller implementations, we used constant Qp for each frame, as with standardized

<sup>2</sup> F. Bossen, "Common HM Test Conditions and Software Reference Configuration," Joint Collaborative Team on Video Coding, document JCTVC-L1100, Geneva, Switzerland, Jan. 2013.

<sup>3</sup> "High Efficiency Video Coding Test Model Software Version 16," Available at [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware), Last accessed on October 1, 2017.

methodology.

Table 2: Qp value and intra frame period definition.

Target	I Frame Qp Value	Intra Frame Period
High Bitrate	{ 18 / 21 / 23 }	Low Delay: 60 frame Random Access: 64 frame <sup>4</sup>
Medium Bitrate	{ 25 / 27 / 29 / 32 }	
Low Bitrate	{ 32 / 37 / 42 / 47 }	

Qp offset for P frame and B frame including GOP (group of pictures) structure are defined by each encoder based on its capability. Because configuration capability of GOP structure should be included for this quality assessment, we defined I frame Qp value and period as common conditions across target encoders.

### Video Sequences and Configurations

Input video sequences are defined in Table 3. All the sequences are available to the public at <https://media.xiph.org/video/derf/>.

Table 3: Video sequences and configuration definition.

	Sequence Name	Resolution	Frame Count	Frame Rate	Bit Depth	Configuration
1	ParkJoy	1920 x 1080	400	50	8	Main Profile Random Access
2	RushFieldCuts	1920 x 1080	400	30	8	Main Profile Random Access
3	Counter-Strike: Global Offensive*	1920 x 1080	400	60	8	Main Profile Random Access
4	Minecraft*	1920 x 1080	400	60	8	Main Profile Low Delay
5	CrowdRun 4K	3840 x 2160	400	50	8	Main Profile Random Access
6	CrowdRun 4K	3840 x 2160	400	50	8	Main Profile Low Delay

### Tested Command Line and Configurations

GOP structure is defined by each encoder based on its capability while locking I frame Qp value and its period, as defined in Table 2, for fair comparison. Following is the tested execution command line and configuration for each encoder.

HM14.0 Tested command line:

```
>TapEncoderStatic -c {encoder_lowdelay_main.cfg|encoder_randomaccess_main.cfg} -f NUM_OF_FRAMES -fs 0 -
fr FRAME_RATE --InputBitDepth=8|10 --OutputBitDepth=8|10 -wdt VIDEO_WIDTH -hgt VIDEO_HEIGHT -i
INPUT_FILE_NAME -q QP_VALUE -b OUT_FILE_NAME
```

<sup>4</sup> In 8 frames B pyramid GOP configuration, some of encoders require multiple of 8 for Intra period to encode, we defined 64 frames for Intra period.

HM14.0 Random Access configuration snippet. B-pyramid structure is used for higher quality:

```

##### Coding Structure #####
IntraPeriod           : 64          # Period of I-Frame ( -1 = only first)
DecodingRefreshType   : 0          # Random Accesss 0:none, 1:CRA, 2:IDR, 3:Recovery Point SEI
GOPSize               : 8          # GOP Size (number of B slice = GOPSize-1)
Frame1: B   8   1   0.442  0   0   0   4   4   -8 -10 -12 -16   0
Frame2: B   4   2   0.3536 0   0   0   2   3   -4 -6  4     1   4   5   1 1 0 0 1
Frame3: B   2   3   0.3536 0   0   0   2   4   -2 -4  2  6     1   2   4   1 1 1 1
Frame4: B   1   4   0.68   0   0   1   2   4   -1  1  3  7     1   1   5   1 0 1 1 1
Frame5: B   3   4   0.68   0   0   1   2   4   -1 -3  1  5     1  -2   5   1 1 1 1 0
Frame6: B   6   3   0.3536 0   0   0   2   4   -2 -4 -6  2     1  -3   5   1 1 1 1 0
Frame7: B   5   4   0.68   0   0   1   2   4   -1 -5  1  3     1   1   5   1 0 1 1 1
Frame8: B   7   4   0.68   0   0   1   2   4   -1 -3 -7  1     1  -2   5   1 1 1 1 0
    
```

HM14.0 Low Delay configuration snippet:

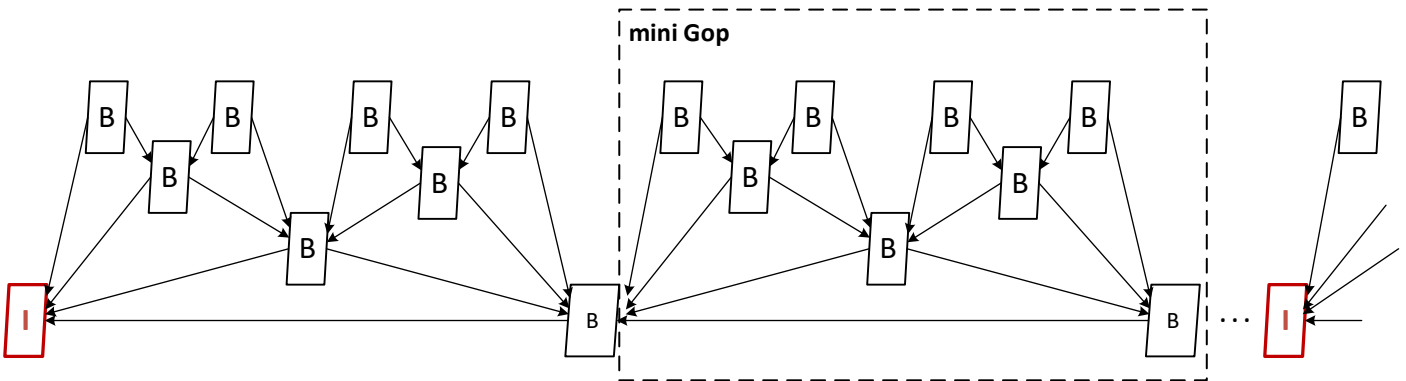
```

##### Coding Structure #####
IntraPeriod           : 60          # Period of I-Frame ( -1 = only first)
DecodingRefreshType   : 0          # Random Accesss 0:none, 1:CRA, 2:IDR, 3:Recovery Point SEI
GOPSize               : 4          # GOP Size (number of B slice = GOPSize-1)
Frame1: B   1   3   0.4624 0   0   0   4   4   -1 -5 -9 -13   0
Frame2: B   2   2   0.4624 0   0   0   4   4   -1 -2 -6 -10   1  -1  5   1 1 1 0 1
Frame3: B   3   3   0.4624 0   0   0   4   4   -1 -3 -7 -11   1  -1  5   0 1 1 1 1
Frame4: B   4   1   0.578   0   0   0   4   4   -1 -4 -8 -12   1  -1  5   0 1 1 1 1
    
```

**10th generation Intel® Core™ processor and Intel® Media SDK command line and configuration**

Intel Media SDK supports B-pyramid GOP structure for random access configuration for higher quality. The illustration of B-pyramid GOP structure like HM14.0 is shown in Figure 1. In B-pyramid configuration, each B frame inside a mini GOP has a Qp offset, and reference structure is carefully designed to achieve the highest quality while minimizing video stream size.

Figure 1: B-pyramid GOP structure supported by Intel® Media SDK.



Following is the command line to enable this feature followed by Low Delay configuration.

Command line for Random Access configuration:

```

>sample_encode.exe h265 -i INPUT_FILE_NAME -w VIDEO_WIDTH -h VIDEO_HEIGHT -u veryslow|medium|veryfast -
f FRAME_RATE -o OUT_FILE_NAME -hw -async 3 -g 64 -n NUM_OF_FRAMES -cqp -qpi
18|21|23|25|27|29|32|37|37|42|47 -r 8 -bref
    
```

Command line for Low Delay configuration:

```
>sample_encode.exe h265 -i INPUT_FILE_NAME -w VIDEO_WIDTH -h VIDEO_HEIGHT -u veryslow|medium|veryfast -f
FRAME_RATE -o OUT_FILE_NAME -hw -async 3 -g 60 -n NUM_OF_FRAMES -cqp -qpi
18|21|23|25|27|29|32|37|37|42|47 -preset conference
```

## FFMPEG and libx265 command line configuration

Following is the x265 software encoder configuration and command line used by this quality evaluation.

Command line for Low Delay configuration (8/10 bit):

```
>ffmpeg.exe -y -s VIDEO_WIDTHxVIDEO_HEIGHT -pix_fmt yuv420p|yuv420p10le -r FRAME_RATE -i INPUT_FILE_NAME
-frames:v NUM_OF_FRAMES -g 60 -bsf:v hevc_mp4toannexb -c:v libx265 -preset veryslow|medium|veryfast -
profile main|main10 -x265-params "qp=QP_VALUE:aq-mode=0:b-adapt=0:bframes=0:b-pyramid=1:tune=psnr:no-
scenecut=1:no-open-gop=1:input-depth=8|10:output-depth=8|10\" OUT_FILE_NAME
```

Command line for Random Access configuration (8/10 bit):

```
>ffmpeg.exe -y -s VIDEO_WIDTHxVIDEO_HEIGHT -pix_fmt yuv420p|yuv420p10le -r FRAME_RATE -i INPUT_FILE_NAME
-frames:v NUM_OF_FRAMES -g 64 -bf 7 -bsf:v hevc_mp4toannexb -c:v libx265 -preset
veryslow|medium|veryfast -profile main|main10 -x265-params "qp=QP_VALUE:aq-mode=0:b-
adapt=0:bframes=7:b-pyramid=1:tune=psnr:no-scenecut=1:no-open-gop=1:input-depth=8|10:output-
depth=8|10\" OUT_FILE_NAME
```

## Quality Evaluation Result

This section describes the common methodology used for quality evaluation for each video encoder. Bitrate, measured in bits per second (bps), is calculated by the following formula from each encoded video stream:

$$\text{Bitrate(bps)} : \text{Encoded video stream file size in byte} * 8 / ( \text{total frames count} / \text{frame rate} )$$

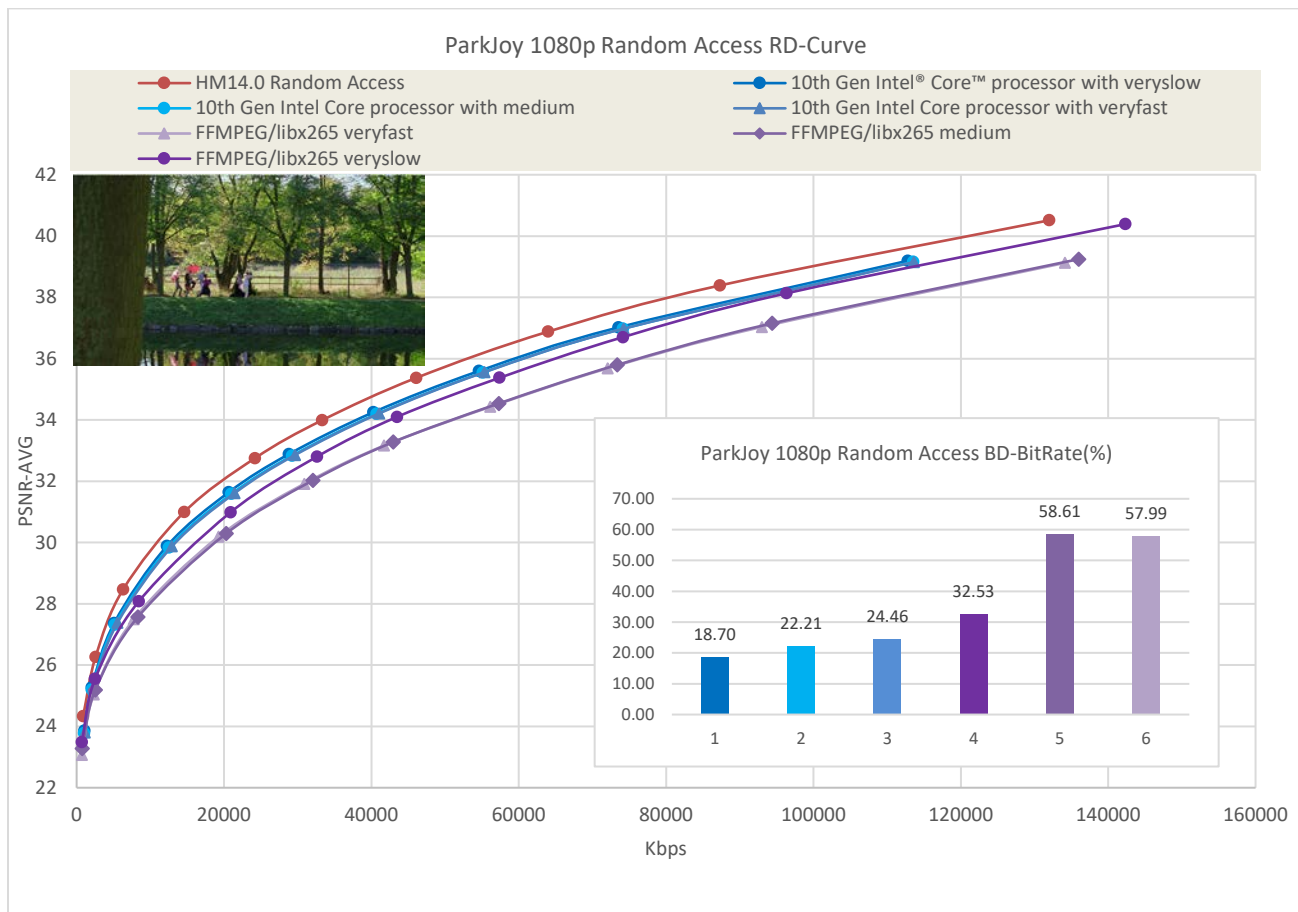
Also, average peak signal-to-noise ratio (PSNR) across video sequences is calculated against original uncompressed video by the following command:

```
> ffmpeg.exe -framerate %d -r FRAME_RATE -i INPUT_FILE_NAME_1 -vcodec rawvideo -pix_fmt
yuv420p|yuv420p10le -s:v VIDEO_WIDTHxVIDEO_HEIGHT -r FRAME_RATE -i INPUT_FILE_NAME_2 -frames:v
NUM_OF_FRAMES -lavfi psnr|ssim -f null
```

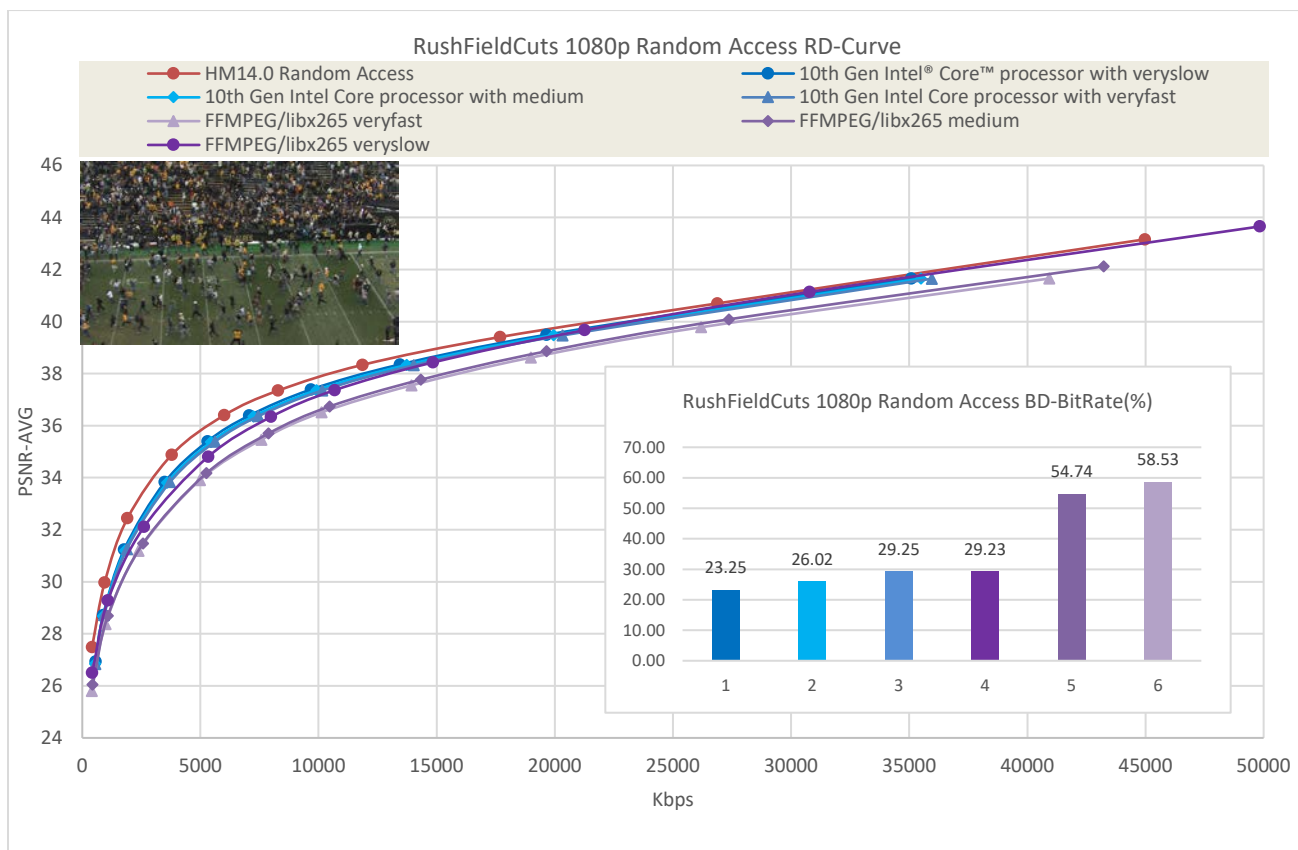
We plot all the bitrate and average PSNR values from each output stream on the graph to create the Rate Distortion Curve (RD-Curve) to compare quality between target encoders. In order to quantify compression gains, we also used the Bjontegaard Delta Bitrate (BD-Rate)<sup>5</sup>. This metric allows computing of the average distance between any two RD curves. The metric shows how much bitrate savings an encoder achieved with respect to a base encoder of the same quality. We used the HM14.0 encoder as a base quality; a lower bar indicates better quality.

<sup>5</sup> G. Bjontegaard, Calculation of average PSNR differences between RDcurves, Technical Report VCEG-M33, ITU-T SG16/Q6, Austin, Texas, USA, 2001.

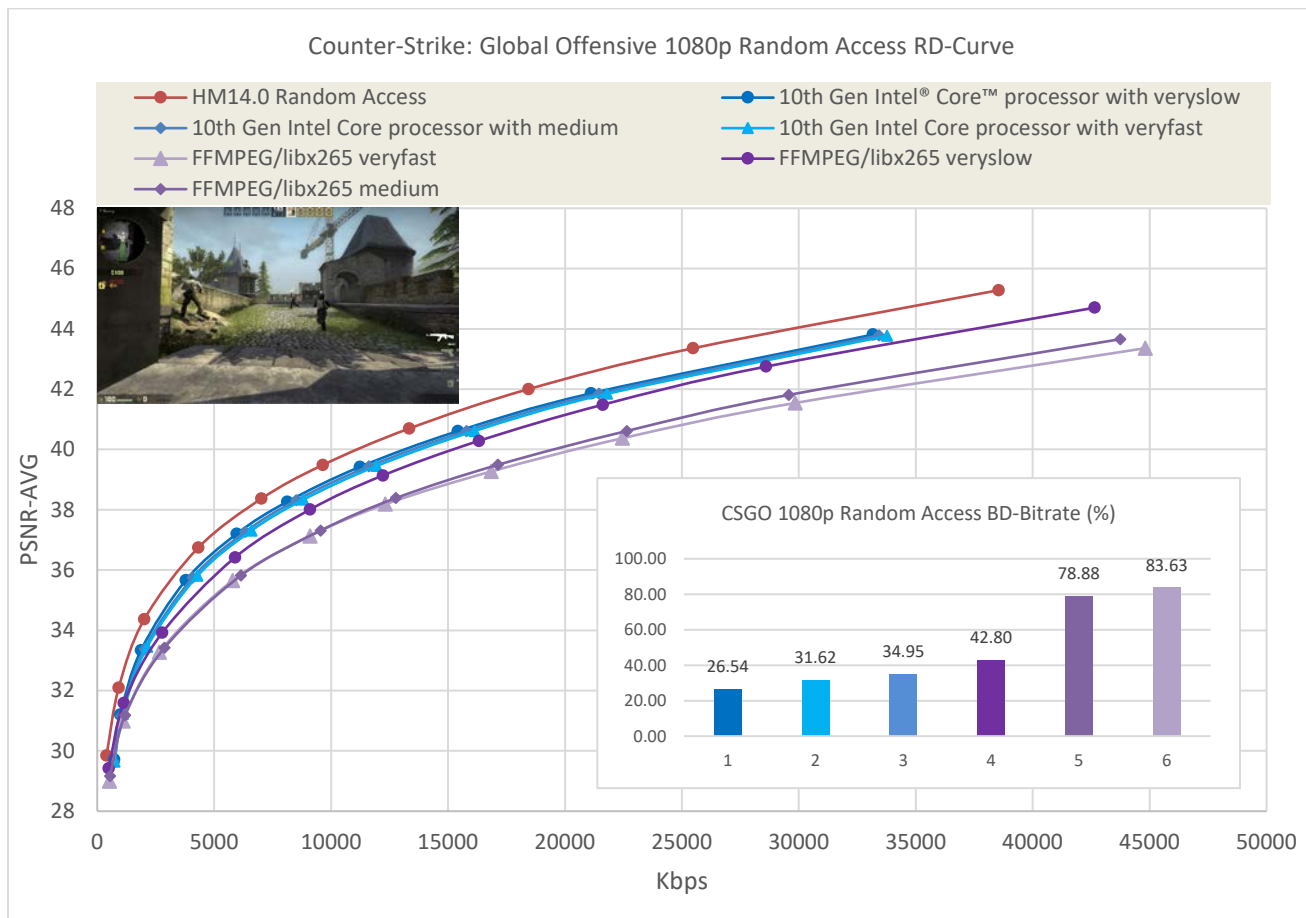
### ParkJoy 1080p - Random Access



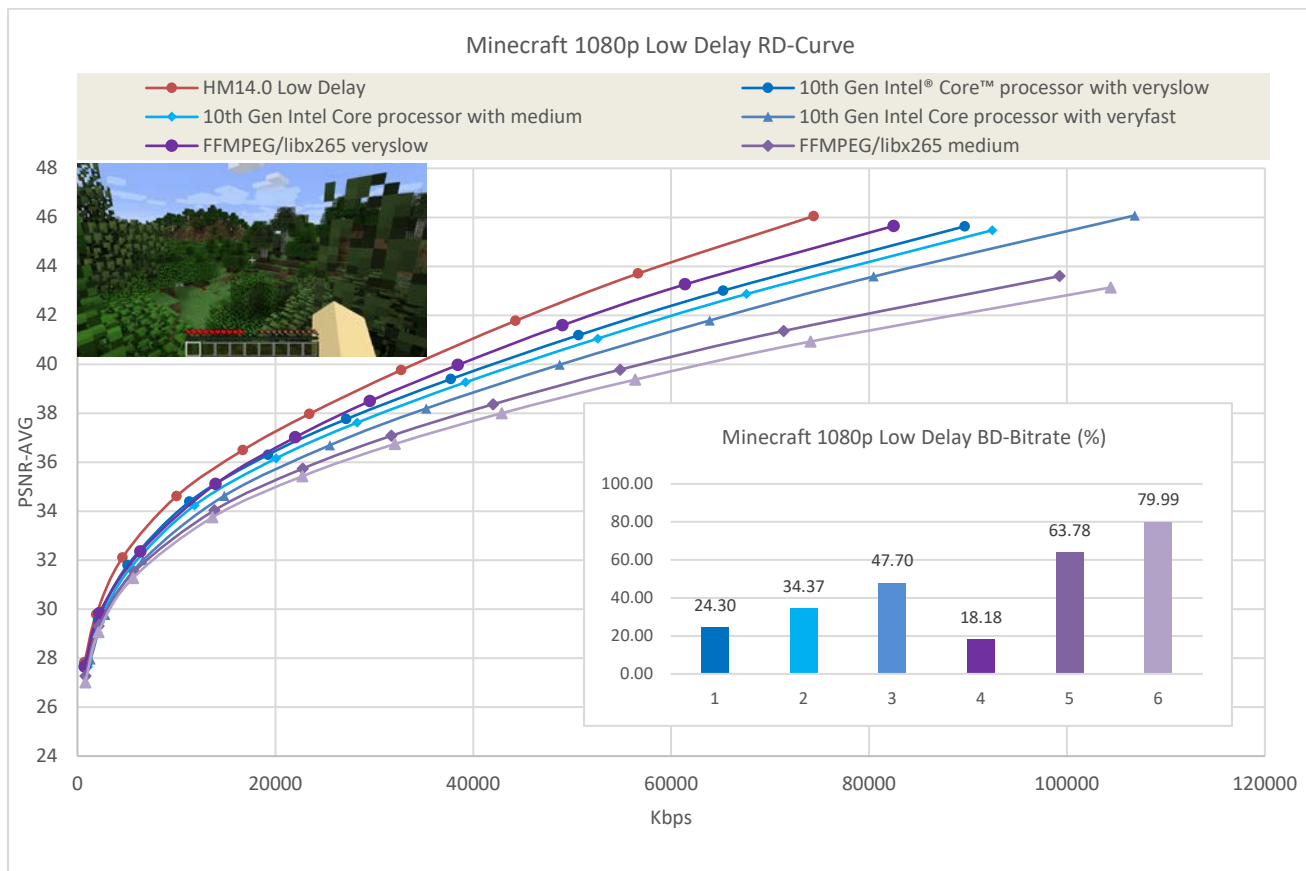
### RushFieldCuts 1080p - Random Access



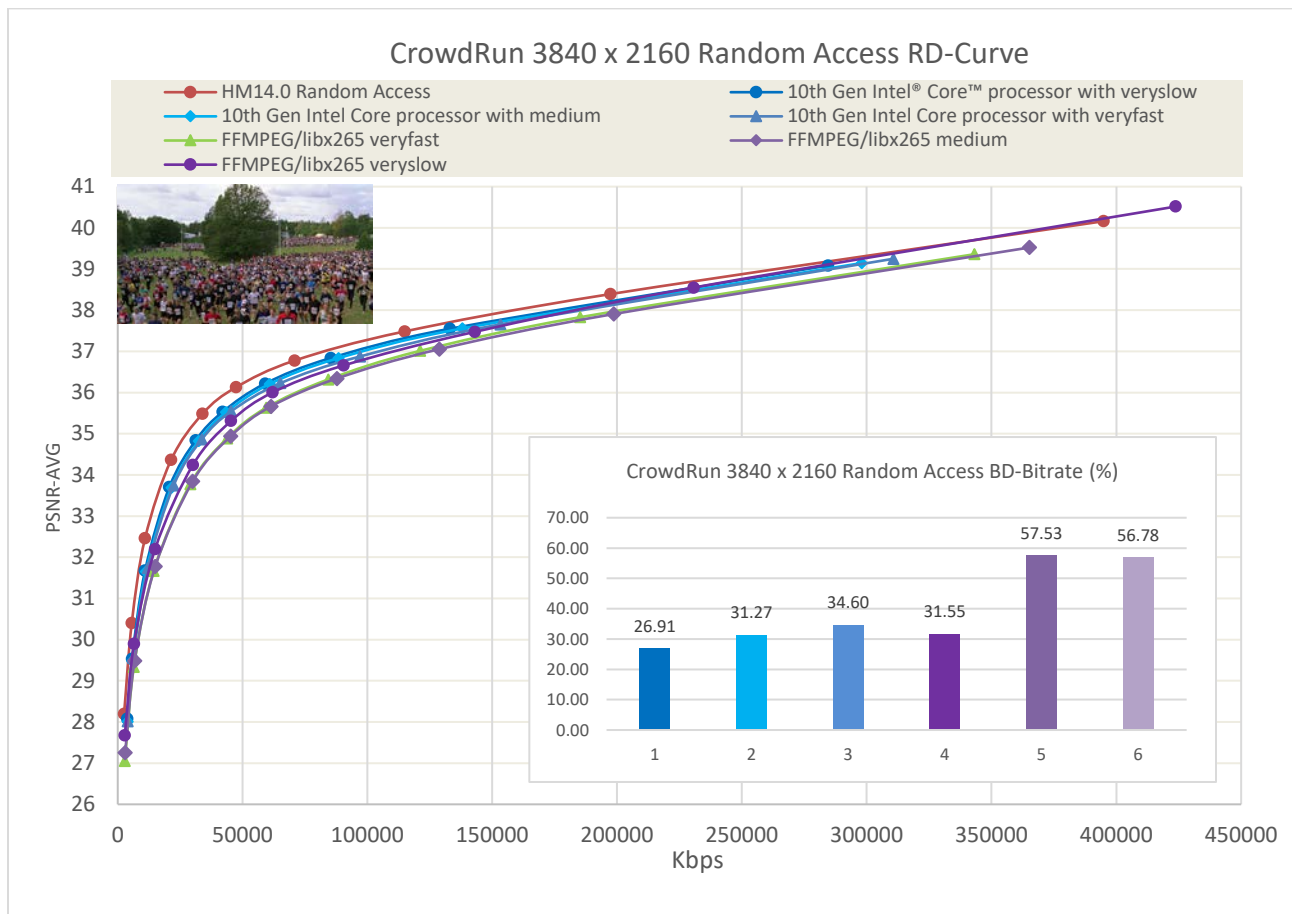
### Counter-Strike: Global Offensive 1080p – Random Access



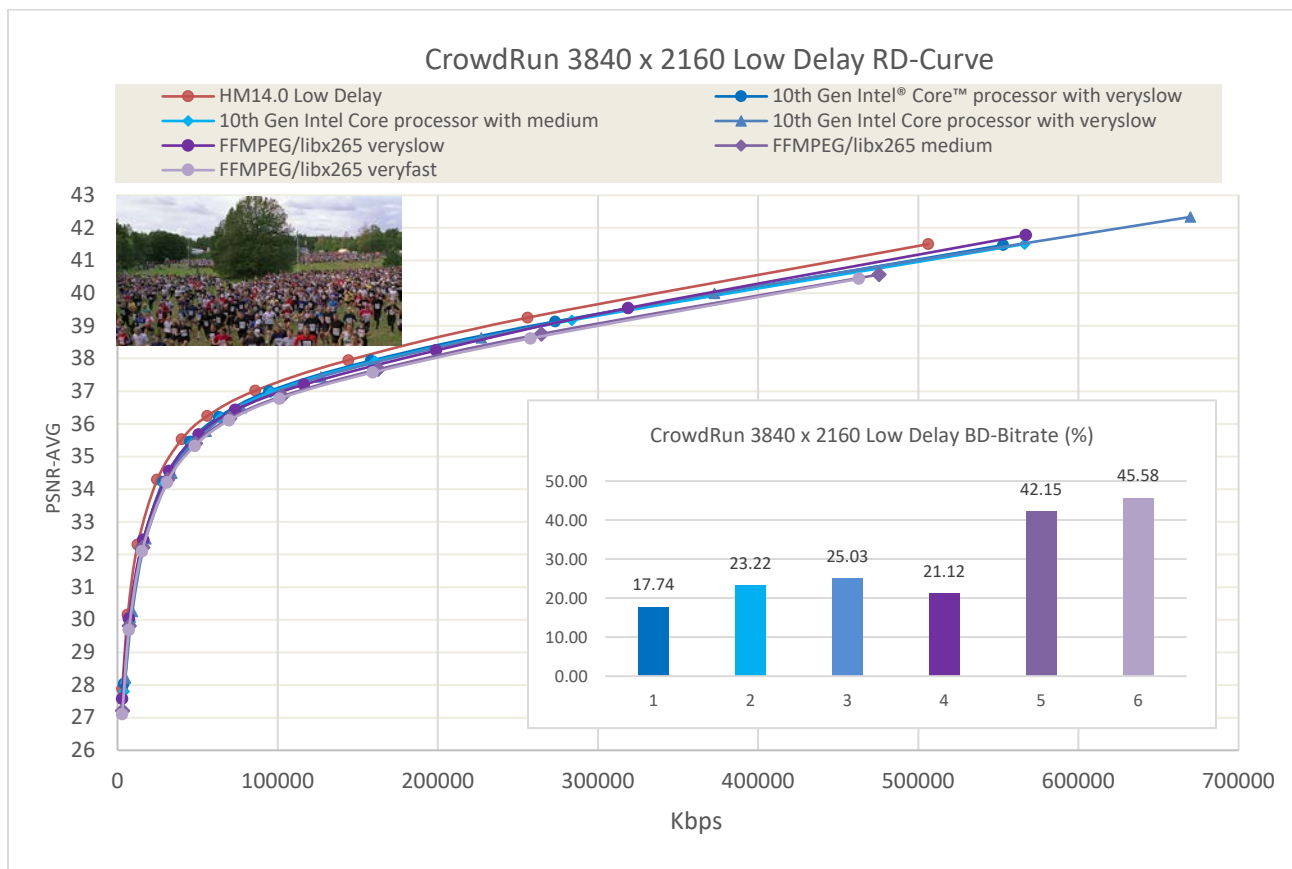
### Minecraft 1080p - Low Delay



CrowdRun 3840 x 2160 - Random Access



CrowdRun 3840 x 2160 - Low Delay



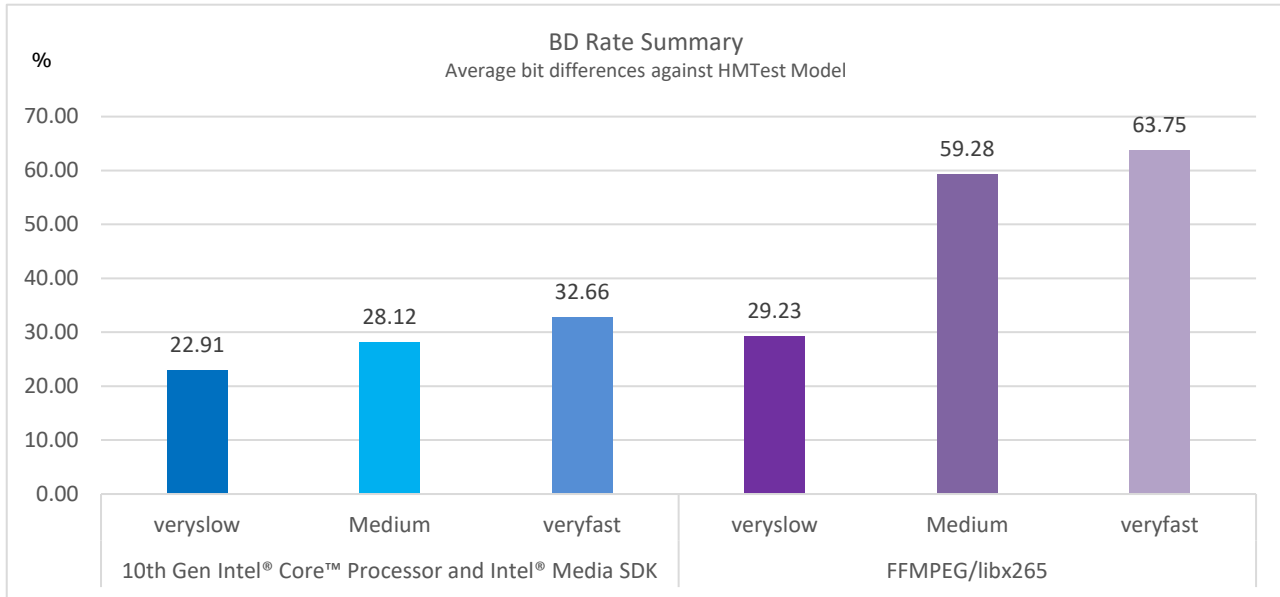


## Quality Evaluation Summary

The new 10th generation Intel Core processor hardware HEVC encoder delivers a superior quality HEVC video stream. It is very close to the HMTTest model, which is theoretically the best software HEVC encoder—only 22.9 percent to 32.6 percent bit differences at the same quality level across the six contents and configurations defined here. Also, each configuration shows reasonably smooth differences, as we expect.

Be ready to take advantage of this new feature and technology in your business.

Table 4: BD-Rate summary



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel® products described herein. You agree to grant Intel® a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at [intel.com](https://www.intel.com).

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting [www.intel.com/design/literature.htm](https://www.intel.com/design/literature.htm).

This sample source code is released under the [Intel Sample Source Code License Agreement](#).

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

© 2019 Intel Corporation