

Title: Decoding a 10-bit sequence using an 8-bit decoder

Status: Input Document to JCT-VC

Purpose: Proposal

Author(s): David Flynn
Gaëlle Martin-Cocher
Dake He

dflynn@blackberry.com
gmartincocher@blackberry.com
dhe@blackberry.com

Source: Research In Motion Ltd.

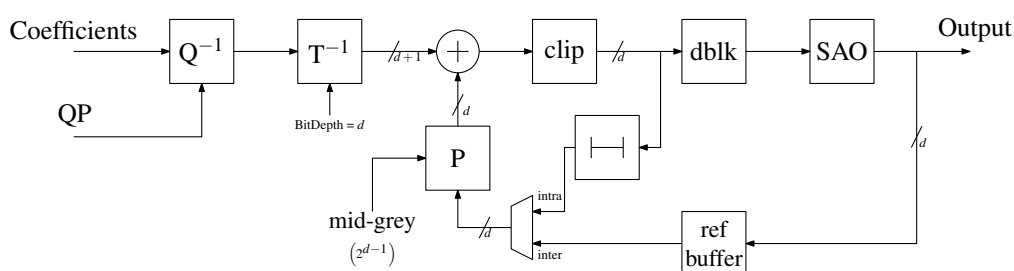
Abstract

HEVC, like other well-known codecs, contains profiles that accommodate the decoding of non-8-bit video sequences. In particular, HEVC contains a 10-bit profile that permits the carriage of 10-bit video sequences. However, for some implementations, the extra resources required for conformance to a higher bit-depth operating point can be prohibitive, yet there exist circumstances where a decoder may wish to offer best-effort decoding of a bitstream. Two techniques are presented that facilitate the decoding of 10-bit video bitstreams using an 8-bit decoder. The first uses the normal 8-bit reconstruction process, while the second involves addition of a rounding process during the reconstruction process. PSNR losses using low-delay main-10 bitstreams average at a 6 dB and 2.5 dB respectively. Further results are presented and specification text is provided for a possible annex.

1 Background

One such class of decoder that is particularly sensitive to the issue of decoding bitstreams at a bit-depth greater than 8-bits is the software decoder. Many general purpose processors can only perform operations at certain word sizes, eg 8-, 16-, 32-, or 64-bit, and the memory layout for video data used by software is strongly linked to the word size. For 8-bit video, it is obvious that it is very efficient to store individual samples in memory. However, for 10-bit video, rather than increasing the storage requirement by 25%, the next available word size (16-bit) doubles the storage and memory bandwidth requirements. While it is possible to use novel data packing formats to reduce the impact, the burden of continually packing and unpacking the data can be considerable for general purpose microprocessors¹

Figure 1 – Abstract decoder signal processing path in HEVC



The abstract signal processing path of an HEVC decoder operating at a nominal bit-depth d is shown in Figure 1. Considering the case when $d = 10$, the intra and inter prediction processes, P , produce 10-bit unsigned predicted samples that are combined with an (inverse quantised and inverse transformed) 11-bit signed residual, resulting in after clipping, 10-bit unsigned reconstructed samples. The in-loop filters for deblocking and SAO do not alter the bit-depth of the reconstructed samples (although internal bit-widths may differ). Finally the reconstructed samples are stored in a buffer available for future prediction and output.

¹This technique is certainly used in many hardware implementations

2 Method–1: 8-bit decoding by adjusting inverse transform scaling

In general, the magnitude of samples produced by the reconstruction process is governed by the magnitude of the coefficients, the QP for inverse quantisation², the scaling factor in the last stage of the inverse transform, the default values used for prediction when reference samples are not available, and the clipping limits.

Many of these aspects are controlled via the derived BitDepthX, adjusting the effective value of this variable will from the same bitstream produce a different output bit-depth. An initial modification to the HM decoder to provide 8-bit decoding in this manner was performed using the following alterations:

- The values RestrictedBitDepthY and RestrictedBitDepthC are set to 8.
- Unpacking operations that use the bit-depth as a binarization process parameter, in particular SAO, are performed using the appropriate bitstream native bit-depth.
- Unpacked SAO offsets are adjusted by a factor of $2^{\text{RestrictedBitDepthX}} \div 2^{\text{BitDepthX}}$.
- All other reconstruction processes are performed with RestrictedBitDepthX substituted for BitDepthX, where $X=\{Y,C\}$. ie, as if the bitstream was signalled as 8-bit.

To examine the modified decoder behaviour in unfavourable conditions, the HM-10 low-delay-B main-10 sequences [1] were decoded and the PSNR measured against the original input sequences. Table 1 illustrates the per-sequence losses when compared against the PSNR of the normal decoder. Figure 4 shows how the PSNR loss evolves as the sequence progresses. Visual inspection of the decoded video shows the following traits:

- Intra pictures experience DC drift increasing towards the bottom right.
- Drift (most noticeable in colour or saturation) evolves over the sequence.
- The distortion introduced can be significant (in excess of 11dB) and can be clearly discernible on a video sequence in a single stimulus test.
- The distortion becomes more significant as QP decreases.

Table 1 – PSNR loss associated with the method of section 2 averaged over 10s sequences at QPs 22/27/32/37 using low-delay main-10 configuration

Sequence	Luma PSNR	Chroma PSNR
BQMall	-8.3/-7.2/-2.7/-2.3	-8.8/-6.0/-2.4/-4.5
BQSquare	-4.5/-1.9/-1.3/-0.3	-4.9/-2.8/-1.1/-1.2
BQTerrace	-12.3/-6.3/-4.0/-2.0	-10.0/-3.9/-1.7/-2.2
BasketballDrillText	-10.8/-9.7/-3.2/-2.2	-6.9/-5.1/-1.4/-1.1
BasketballDrill	-9.5/-10.3/-3.3/-3.2	-7.6/-5.8/-1.7/-1.1
BasketballDrive	-10.8/-8.2/-5.9/-3.3	-11.4/-8.3/-6.0/-4.5
BasketballPass	-6.0/-3.3/-0.8/-0.3	-4.0/-1.6/-0.5/-0.8
BlowingBubbles	-6.7/-2.1/-2.0/-0.3	-7.6/-4.7/-1.9/-1.5
Cactus	-13.6/-9.6/-7.5/-5.3	-12.1/-8.3/-4.4/-4.9
ChinaSpeed	-10.9/-8.3/-5.0/-1.4	-10.3/-9.1/-4.9/-3.7
FourPeople	-15.1/-12.7/-7.2/-3.9	-12.2/-8.0/-5.0/-2.0
Johnny	-9.4/-9.8/-8.5/-1.7	-15.8/-6.1/-4.0/-3.3
Kimono1	-12.5/-8.7/-3.5/-8.5	-10.8/-6.9/-4.7/-6.2
KristenAndSara	-9.3/-7.7/-14.7/-2.9	-15.0/-5.2/-5.0/-3.2
ParkScene	-15.2/-8.2/-8.0/-6.2	-12.3/-5.4/-4.4/-2.7
PartyScene	-7.9/-7.7/-3.8/-1.5	-8.2/-5.5/-1.4/-2.0
PeopleOnStreet	-17.3/-13.2/-8.6/-4.4	-16.3/-11.6/-7.7/-6.9
RaceHorses	-6.2/-5.7/-2.0/-0.2	-4.7/-4.0/-0.5/-1.2
RaceHorsesC	-10.7/-7.3/-6.8/-1.7	-7.6/-8.5/-1.8/-2.5
SlideEditing	-7.4/-7.3/-2.8/-0.5	-6.3/-1.7/-1.0/-0.6
SlideShow	-5.2/-3.1/-1.0/-0.3	-6.6/-2.6/-1.4/-1.1
Traffic	-17.2/-15.0/-9.6/-8.4	-12.3/-10.7/-6.7/-7.0

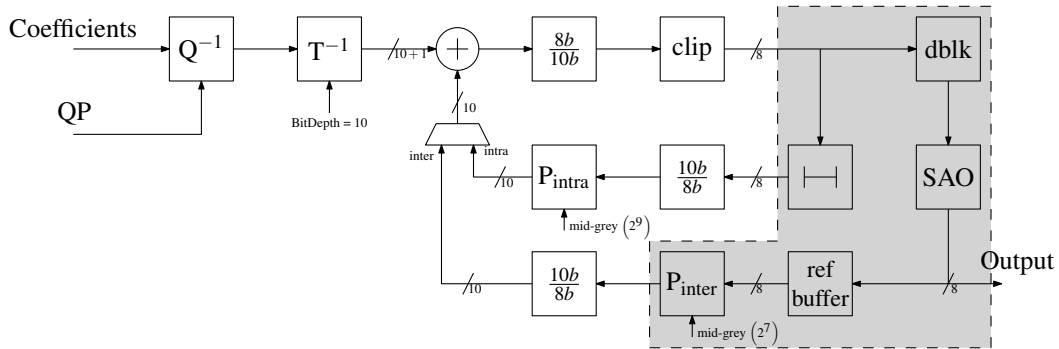
²The quantisation process itself is bit-depth independent

In a more complete implementation, the derivation of QpBdOffsetY would require the use of BitDepthY not RestrictedBitDepthY, otherwise there is the potential for the derivation of QP to wrap incorrectly. This effect cannot be observed using the constant QP anchor sequences.

3 Method–2: Hybrid 8-bit–10-bit decoding using rounding in picture construction process

For an 8-bit decoder, many calculations are already computed using 16-bit or larger words. For example, the inverse transformed residual is a 16-bit signed value, and that the combination of the residual and prediction must occur at a minimum of 10-bits due to the signed nature of the residual (assuming a reasonable residual and that addition with saturation is unavailable). On software systems, this must be implemented using the next available word size, typically 16-bit. Therefore there may exist a better trade-off as to the position of conversion from a 10-bit system to an 8-bit system, especially if the reconstruction quality can be improved.

Figure 2 – Illustrative signal coding path of hybrid 8-bit–10-bit decoder



An opportune position to perform such a conversion is immediately following the addition of the prediction and residual, prior to clipping. The proposed system is illustrated in figure 2.

If the reconstruction loop now performs what is an effective 10- to 8-bit conversion, it will also be necessary to perform a corresponding 8- to 10-bit conversion. We propose that this is performed on the samples used for intra prediction without significant impacting the cost of intra prediction, and in the case of inter prediction it is performed after motion compensation so as not to increase the cost of sub-pel filtering. The method for increasing the bit-width is a left shift operation.

Since significant drift was observed in the intra pictures produced by the method of section 2, it is worth noting that the method essentially implements a truncation of the residual. The effect of this is to introduce a systematic less-than-unity gain into the reconstruction loop, through consistent under-correction of the prediction, an error that accumulates due to the nature of intra prediction. The choice of rounding method is essential to reducing the drift in intra pictures. The obvious choice for implementing rounding, rounding half values towards positive infinity ($\lfloor (x + 2^{n-1}) \div 2^n \rfloor$), as favoured in other parts of the HEVC specification also contains a systematic bias resulting in a loop gain greater than one. An alternative rounding technique, as used in IEEE 754 is to round half values towards the nearest even value³. For sufficiently well distributed values, this method is unbiased for both positive and negative numbers. Such a method can be implemented using $(x + (1 \ll (n-1)) - 1 + ((x > n) \& 1)) \gg n$.

A modification to the HM decoder that provides 8-bit decoding using this method implemented the following changes to the decoding process:

- The values RestrictedBitDepthY and RestrictedBitDepthC are set to 8.
Note: the value of BitDepthX as used below is as normally derived from the bitstream.
- Unpacking operations that use the bit-depth as a binarization process parameter, in particular SAO, are performed using the appropriate bitstream native bit-depth.

³also known as bankers' rounding

- Unpacked SAO offsets are adjusted to by a factor of $2^{\text{RestrictedBitDepthX}} \div 2^{\text{BitDepthX}}$.
- Inverse quantisation, including the setting of QpBdOffset, and the inverse transformation of residuals are performed using BitDepthX.
- The process for generating intra prediction samples involves multiplying the samples used for intra prediction by $2^{\text{BitDepthX} - \text{RestrictedBitDepthX}}$. The intra prediction process is then performed using BitDepthX.
- The process for generating inter prediction samples involves multiplying the resulting samples by $2^{\text{BitDepthX} - \text{RestrictedBitDepthX}}$. The inter prediction process is performed using RestrictedBitDepthX.
- The picture reconstruction process involves adding the prediction samples to the residual samples and then reducing their magnitude using $(x + 1 + ((x > 2) \& 1)) \gg 2$ prior to clipping, with clipping limits set to $[0, 2^{\text{RestrictedBitDepthX}} - 1]$.
- All other reconstruction processes are performed with RestrictedBitDepthX substituted for BitDepthX, where $X = \{Y, C\}$.

Table 2 – PSNR loss associated with method of section 3 averaged over 10s sequences at QPs 22/27/32/37 using low-delay main-10 configuration

Sequence	Luma PSNR	Chroma PSNR
BQMall	-7.1/-3.5/-1.1/-0.6	-3.7/-2.7/-1.5/-1.2
BQSquare	-3.9/-1.7/-0.3/-0.2	-4.7/-2.0/-0.8/-0.6
BQTerrace	-6.3/-2.6/-0.8/-1.3	-4.2/-3.2/-1.7/-1.0
BasketballDrillText	-7.2/-2.8/-0.6/-0.4	-5.4/-2.9/-1.0/-0.4
BasketballDrill	-6.5/-3.1/-0.5/-0.5	-5.3/-4.1/-1.5/-0.3
BasketballDrive	-3.9/-2.3/-1.7/-0.7	-5.8/-2.6/-2.3/-0.9
BasketballPass	-4.0/-1.3/-0.3/-0.2	-2.8/-1.0/-0.4/-0.3
BlowingBubbles	-3.5/-1.3/-0.5/-0.3	-4.6/-1.7/-1.4/-0.8
Cactus	-6.2/-2.3/-1.5/-1.5	-5.1/-4.1/-1.5/-0.8
ChinaSpeed	-7.7/-3.0/-2.8/-0.6	-8.5/-4.0/-2.7/-1.2
FourPeople	-8.9/-4.2/-1.5/-3.5	-4.3/-4.3/-2.3/-0.8
Johnny	-7.2/-2.4/-2.1/-1.4	-7.8/-4.1/-2.3/-1.2
Kimono1	-5.6/-1.7/-1.4/-1.1	-2.7/-1.8/-2.6/-1.0
KristenAndSara	-5.4/-2.6/-0.9/-2.3	-7.8/-3.4/-4.0/-1.4
ParkScene	-5.6/-2.0/-1.0/-0.5	-4.6/-2.8/-0.8/-0.7
PartyScene	-3.3/-2.9/-0.6/-0.4	-3.4/-1.7/-1.0/-0.9
PeopleOnStreet	-3.6/-1.2/-1.2/-0.7	-4.4/-3.2/-3.5/-1.3
RaceHorses	-3.9/-1.4/-0.8/-0.3	-5.1/-0.9/-0.5/-0.7
RaceHorsesC	-2.4/-1.3/-0.8/-0.4	-4.2/-2.9/-1.9/-0.7
SlideEditing	-6.1/-2.0/-1.5/-0.3	-1.9/-0.7/-0.3/-0.2
SlideShow	-6.5/-1.9/-0.6/-0.3	-4.4/-2.4/-1.1/-0.5
Traffic	-6.2/-5.1/-2.3/-1.1	-4.5/-3.0/-2.4/-0.8

After performing the same analysis as per section 2, objective results show a significant reduction in losses compared to the first method. Visual inspection shows that the gross DC drift observed using the first method is avoided when using this method, to the extent that with a single stimulus test it can be difficult observe a degradation.

4 Further results

The enclosed spreadsheet includes results for all sequences and all QPs. For each test point it shows PSNR figures averaged for the entire sequence, and two additional results averaged over the first and last ten frames respectively.

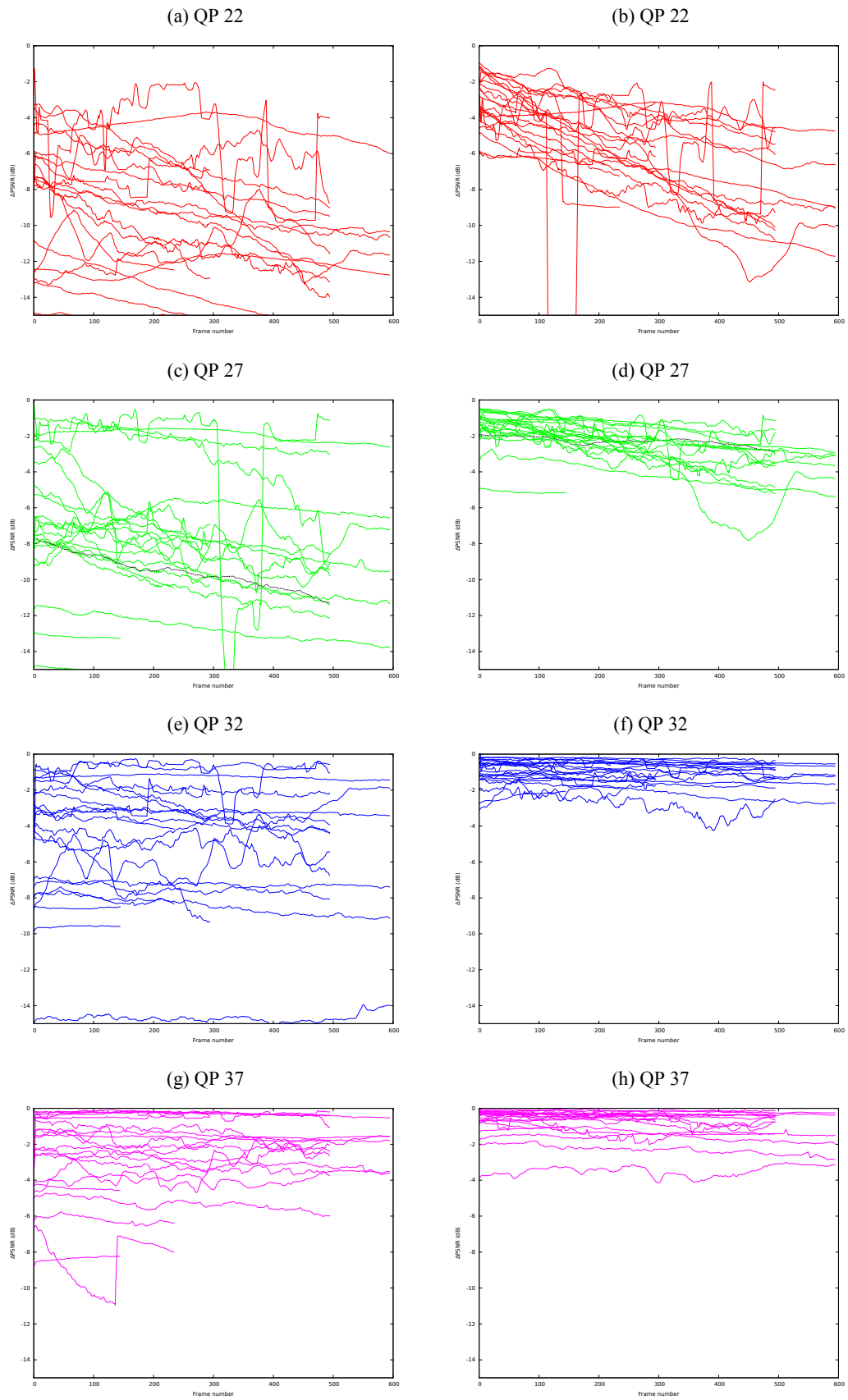
Figure 3 illustrates the typical drift effect observed when using the two methods compared to the vanilla HM-10 decode after decoding the ten-second 500-frame sequence Cactus encoded using the low-delay main-10 configuration at QP=27. The figure shows the first intra and the last inter frame of the sequence. Video sequences using both methods are available at the meeting for viewing.

Figure 4 illustrates the temporal performance of the two methods with plots of the per-frame PSNR (subsequent to a five tap median filter to remove variation due to the GOP structure). The Cactus sequence used in figure 3 is marked in black. Considering all of the sequences, in most cases the performance of the second method is superior to the first method. In circumstances where this doesn't appear to be the case, such as for SlideShow at around frames 115–160 for QP=22, careful inspection shows a peculiar uncorrected large magnitude intra prediction error that is only evident when performing a dual stimulus test.

Figure 3 – Typical distortion introduced by 8-bit decoding of 10-bit sequences. Sequence cactus QP=27, first frame (left), last frame (right). (a,b) Method-1, (c,d) Method-2, (e,f) HM-10.



Figure 4 – Plot of frame-number versus ΔPSNR for all sequences at particular QPs using Method-1 (left) and Method-2 (right).



5 Proposed specification text

Annex F

Best-effort decoding of 10-bit sequences using an 8-bit decoder

(This annex forms an integral part of this Recommendation | International Standard)

F.1 General

This annex specifies the recommended behaviour for decoders conformant with the Main profile that also provide a best-effort decoding capability for bit streams conformant to High profile that would otherwise not be decodable.

NOTE: The use of this recommended behaviour does not guarantee perfect reconstruction of the video sequence and an element of drift is likely.

F.2 Modified decoding process

A decoder implementing the modified decoding process as specified in this annex shall perform the ordinary decoding process as specified in this Recommendation | International Standard unless otherwise modified below.

The variables $\text{RestrictedBitDepth}_Y$ and $\text{RestrictedBitDepth}_C$ are each set to the value 8.

The following mathematical functions are defined:

$$\text{RClip1}_Y(x) = \text{RClip1Idx}(x, 0) \quad (\text{F-1})$$

$$\text{RClip1}_C(x) = \text{RClip1Idx}(x, 1) \quad (\text{F-2})$$

$$\text{RClip1Idx}(x, \text{cIdx}) = \text{Clip3}(0, (1 \ll \text{RestrictedBitDepthOf}(\text{cIdx})) - 1, x) \quad (\text{F-3})$$

$$\text{BitDepthOf}(\text{cIdx}) = \begin{cases} \text{BitDepth}_Y & ; \text{cIdx} = 0 \\ \text{BitDepth}_C & ; \text{cIdx} = 1, 2 \end{cases} \quad (\text{F-4})$$

$$\text{RestrictedBitDepthOf}(\text{cIdx}) = \begin{cases} \text{RestrictedBitDepth}_Y & ; \text{cIdx} = 0 \\ \text{RestrictedBitDepth}_C & ; \text{cIdx} = 1, 2 \end{cases} \quad (\text{F-5})$$

$$\text{RoundToEven}(x, \text{cIdx}) = \text{RoundToEvenShift}(x, \text{BitDepthOf}(\text{cIdx}) - \text{RestrictedBitDepthOf}(\text{cIdx})) \quad (\text{F-6})$$

$$\text{RoundToEvenShift}(x, \text{shift}) = x + (1 \ll (\text{shift} - 1)) - 1 + ((x \gg \text{shift}) \& 1) \gg \text{shift} \quad (\text{F-7})$$

The sample adaptive offset semantics as specified in subclause 7.4.9.3 are modified as follows. Subsequent to the ordinary derivation of SaoOffsetVal , each value is modified as follows:

$$\begin{aligned} \text{SaoOffsetVal}[\text{cIdx}][\text{rx}][\text{ry}][i] = \\ \text{SaoOffsetVal}[\text{cIdx}][\text{rx}][\text{ry}][i] \gg (\text{BitDepthOf}(\text{cIdx}) - \text{RestrictedBitDepthOf}(\text{cIdx})) \end{aligned} \quad (\text{F-8})$$

The generation of unavailable pictures as specified in subclause 8.3.3.2 shall be performed using $\text{RestrictedBitDepth}_Y$ in place of the variable BitDepth_Y and $\text{RestrictedBitDepth}_C$ in place of the variable BitDepth_C .

The general decoding process for coding units coded in intra prediction mode as specified in subclause 8.4.1 when $\text{pcm_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, shall be performed using $\text{RestrictedBitDepth}_Y$ in place of the variable BitDepth_Y and $\text{RestrictedBitDepth}_C$ in place of the variable BitDepth_C .

The general intra sample prediction process as specified in subclause 8.4.4.2.1 is modified as follows. Subsequent to the ordinary derivation of $p[x][y]$ when the sample $p[x][y]$ is marked as "available for intra prediction", the value of $p[x][y]$ is modified as follows:

$$p[x][y] = p[x][y] \ll (\text{BitDepthOf}(\text{cIdx}) - \text{RestrictedBitDepthOf}(\text{cIdx})) \quad (\text{F-9})$$

The general decoding process for coding units coded in inter prediction mode as specified in subclause 8.5.1 is modified as follows. Subsequent to the invocation of the inter prediction process as specified in subclause 8.5.2, the three arrays predSamples_L , predSamples_{Cb} and predSamples_{Cr} are modified as follows:

$$\begin{aligned} \text{predSamples}_L[x][y] &= \text{predSamples}_L[x][y] \ll (\text{BitDepth}_Y - \text{RestrictedBitDepth}_Y) \\ \text{with } x, y &= 0..nCbS_L \end{aligned} \quad (\text{F-10})$$

$$\begin{aligned} \text{predSamples}_{Cb}[x][y] &= \text{predSamples}_{Cb}[x][y] \ll (\text{BitDepth}_C - \text{RestrictedBitDepth}_C) \\ \text{with } x, y &= 0..nCbS_C \end{aligned} \quad (\text{F-11})$$

$$\begin{aligned} \text{predSamples}_{Cr}[x][y] &= \text{predSamples}_{Cr}[x][y] \ll (\text{BitDepth}_C - \text{RestrictedBitDepth}_C) \\ \text{with } x, y &= 0..nCbS_C \end{aligned} \quad (\text{F-12})$$

The luma sample interpolation process as specified in subclause 8.5.3.3.2 shall be performed using $\text{RestrictedBitDepth}_Y$ in place of the variable BitDepth_Y .

The chroma sample interpolation process as specified in subclause 8.5.3.3.3 shall be performed using $\text{RestrictedBitDepth}_C$ in place of the variable BitDepth_C .

The weighted sample prediction process as specified in subclause 8.5.3.3.4.1 shall be performed using $\text{RestrictedBitDepth}_Y$ in place of the variable BitDepth_Y and $\text{RestrictedBitDepth}_C$ in place of the variable BitDepth_C .

The picture construction process prior to the in-loop filter process as specified in subclause 8.6.5 is modified as follows. The derivation of $\text{recSamples}[][]$ using equation (8-280) is replaced by the following:

$$\begin{aligned} \text{recSamples}[x_{\text{Curr}} + i][y_{\text{Curr}} + j] &= \\ &\text{RClip1Idx}(\text{RoundToEven}(\text{predSamples}[i][j] + \text{resSamples}[i][j], \text{cIdx}), \text{cIdx}) \\ \text{with } i &= 0..n_{\text{CurrS}} - 1, j = 0..n_{\text{CurrS}} - 1 \end{aligned} \quad (\text{F-13})$$

The decision process for luma block edges as specified in subclause 8.7.2.5.3 shall be performed using $\text{RestrictedBitDepth}_Y$ in place of the variable BitDepth_Y .

The decision process for chroma block edges as specified in subclause 8.7.2.5.5 shall be performed using $\text{RestrictedBitDepth}_C$ in place of the variable BitDepth_C .

The filtering process for a luma sample as specified in subclause 8.7.2.5.7 shall be performed using RClip1_Y in place of the function Clip1_Y .

The filtering process for a chroma sample as specified in subclause 8.7.2.5.8 shall be performed using RClip1_C in place of the function Clip1_C .

The sample adaptive offset coding tree block modification process as specified in subclause 8.7.3.2 shall be performed using $\text{RestrictedBitDepth}_Y$ in place of BitDepth_Y and $\text{RestrictedBitDepth}_C$ in place of BitDepth_C .

F.3 Interpretation of SEI message semantics

TBD

F.4 Interpretation of VUI semantics

TBD

Research in Motion Limited may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

References

- [1] F. Bossen, “Common hm test conditions and software reference configurations [missing],” JCTVC-L1100, JCT-VC, Jan. 2013.