

AHG16/AHG8: PROPOSED CLEANUP FOR REFERENCE WRAPAROUND (JVET-N0070)



Brian Heng, Minhua Zhou, and Wade Wan

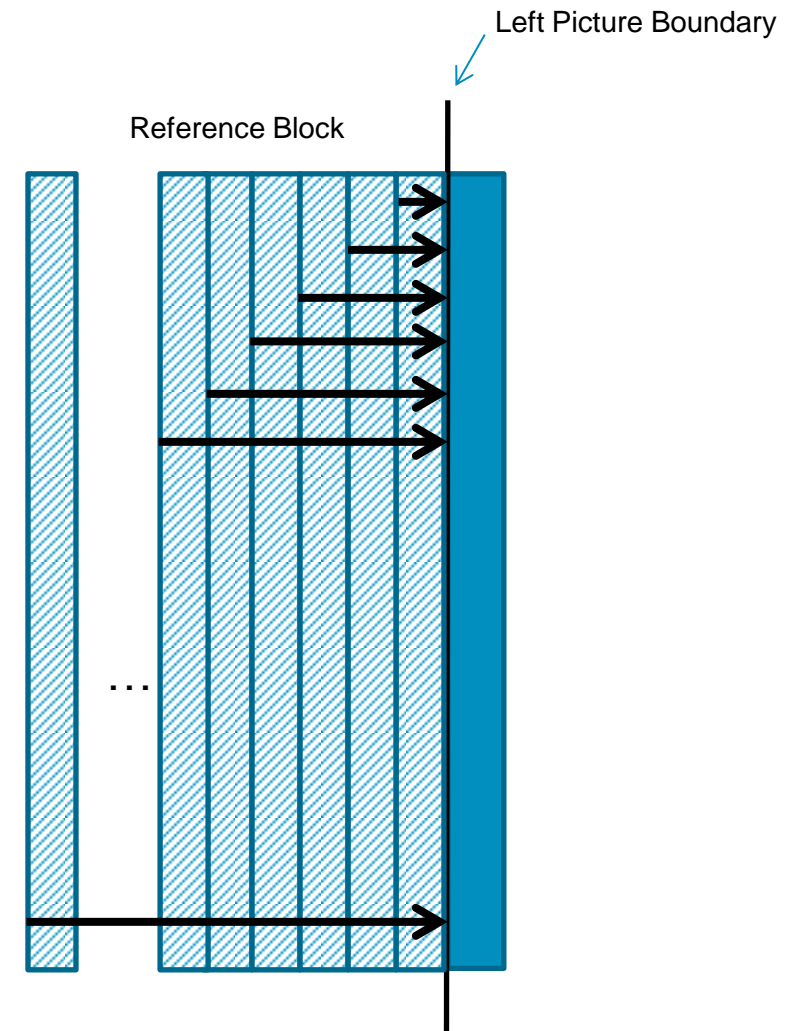
Broadcom Inc.

Background

- To address the following corner cases
 - Multiple wraparounds caused by small wraparound offsets (down to 4)
 - Repeated wraparounds caused by extremely narrow picture widths (e.g. down to one CTU)
 - Multiple wraparounds caused by extremely large motion vectors

Multiple Wraparounds Caused by Small Wraparound Offsets

- The current definition of wraparound offset
 - “**sps_ref_wraparound_offset_minus1** plus 1 specifies the offset used for computing the horizontal wrap-around position in units of `MinCbSizeY` luma samples. The value of `ref_wraparound_offset_minus1` shall be in the range of 0 to $(\text{pic_width_in_luma_samples} / \text{MinCbSizeY}) - 1$, inclusive.”
 - The minimum wraparound offset can be as small as 4 luma samples (i.e. `sps_ref_wraparound_offset_minus1=0`)
 - For example, for a 135x135 reference patch at left picture boundary the wraparounds would occur up to 34 times (each time it is shifted to the right by the wraparound by 4 samples)
 - **Proposed solution:**
 - Impose a restriction that the **sps_ref_wraparound_offset_minus1** shall be in the range of $(\text{CtbSizeY} / \text{MinCbSizeY}) + 1$ to $(\text{pic_width_in_luma_samples} / \text{MinCbSizeY}) - 1$, inclusive.
 - The minimum wraparound becomes `CtbSizeY+8`. Even the largest reference patch will wrap at most one time.



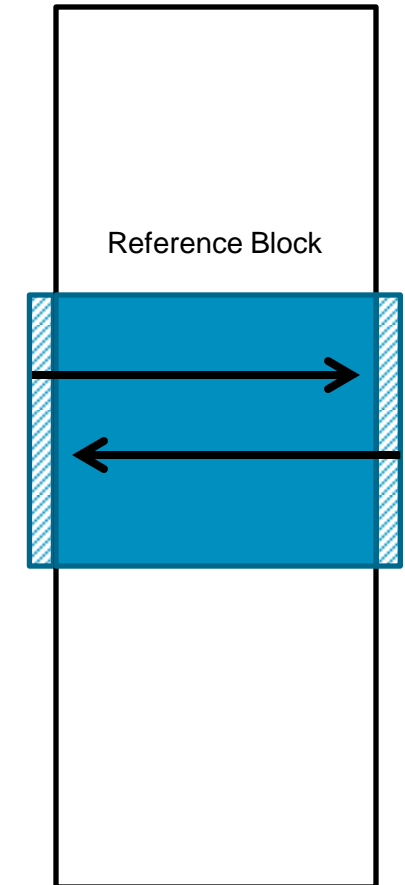
Narrow Pictures

- For reference picture width less than $(\text{CtbSizeY} + 8)$, a reference patch can extend beyond both the left and right sides of the picture, and wrapping is required in opposite directions.
 - Yet another corner case that would need to be handled by decoders.
- Proposed solution:**
 - Increasing the minimum wraparound offset to $\text{CtbSizeY} + 8$ (previous slide) also indirectly prevents this case.
 - Specifically, there is no valid wraparound offset when $(\text{CtbSizeY} / \text{MinCbSizeY}) + 1 > (\text{pic_width_in_luma_samples} / \text{MinCbSizeY}) - 1$.
 - Disable wraparound if this occurs.

“**sps_ref_wraparound_enabled_flag** equal to 1 specifies that horizontal wrap-around motion compensation is applied in inter prediction. **sps_ref_wraparound_enabled_flag** equal to 0 specifies that horizontal wrap-around motion compensation is not applied. **When **sps_ref_wraparound_enabled_flag** is not present, it is inferred to be equal to 0.**”

...
if ((CtbSizeY/MinCbSizeY + 1) <= (pic_width_in_luma_samples / MinCbSizeY - 1)) {
sps_ref_wraparound_enabled_flag
if(sps_ref_wraparound_enabled_flag)
sps_ref_wraparound_offset_minus1
}
...

Reference Picture



Motion Vector Modulus Wrapping

- Motion vectors are not restricted in how far outside the picture they can point outside the picture (besides the maximum vector range defined in the standard, -2^{17} to $2^{17} - 1$, inclusive).
- For small wraparound offsets with vectors pointing very far off the left or right edge of the picture, the vector can wrap around potentially thousands of times, by using e.g. the current SW implementation method:

```
while( mvX > iHorMax ) {  
    mvX -= ( sps.getWrapAroundOffset() << iMvShift );  
}  
while( mvX < iHorMin ) {  
    mvX += ( sps.getWrapAroundOffset() << iMvShift );  
}
```

- **Proposed solution: Replace modulo operation with single conditional offset**

If `sps_ref_wraparound_enabled_flag` equals 0:

$$xInt_i = \text{Clip3}(0, \text{picW} - 1, xInt_L + i)$$

Otherwise:

$$\text{ClipH}(o, W, x) = \begin{cases} x + o & ; \quad x < 0 \\ x - o & ; \quad x > W - 1 \\ x & ; \quad \text{otherwise} \end{cases}$$

$$xInt_i = \text{Clip3}(0, \text{picW} - 1, \text{ClipH}((\text{sps_ref_wraparound_offset_minus1} + 1) * \text{MinCbSizeY}, \text{picW}, (xInt_L + i)))$$

- Any reference location that would have wrapped around more than once will get clipped within the picture boundary instead.

Summary of Proposal

- 1) Impose a restriction that **sps_ref_wraparound_offset_minus1** shall be in the range of $(\text{CtbSizeY}/\text{MinCbSizeY}) + 1$ to $(\text{pic_width_in_luma_samples} / \text{MinCbSizeY}) - 1$, inclusive.
 - Avoid multiple wraparounds caused by small wraparound offsets
- 2) Disable wraparound for narrow pictures by not signalling **sps_ref_wraparound_enabled_flag** in SPS (inferred to be 0 instead) if $(\text{CtbSizeY}/\text{MinCbSizeY} + 1) > (\text{pic_width_in_luma_samples} / \text{MinCbSizeY} - 1)$
 - Avoid multiple wraparounds in opposite directions for narrow pictures
- 3) Replace the wrapping modulus operation “ClipH” with a conditional offset:

$$\text{ClipH}(o, W, x) = \begin{cases} x + o & ; \quad x < 0 \\ x - o & ; \quad x > W - 1 \\ x & ; \quad \text{otherwise} \end{cases}$$

and add a clipping operation:

$$\text{xInt}_i = \text{Clip3}(0, \text{picW} - 1, \text{ClipH}((\text{sps_ref_wraparound_offset_minus1} + 1) * \text{MinCbSizeY}, \text{picW}, (\text{xInt}_L + i)))$$

- Any reference location that would have wrapped around more than once, because of large motion vectors, will get clipped within the picture boundary instead.