|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San José, CA, USA, 1–10 February, 2012 | Document: JCTVC-H0462 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Spatial Motion Vector Predictor Mismatch and Possible Unintended Behavior** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Information & Proposal | | |
| *Author(s) or Contact(s):* | Brian Heng Broadcom Corporation 5300 California Ave Irvine, CA 92617 USA  Peisong Chen Broadcom Corporation 16340 West Bernardo Dr. San Diego, CA 92127 USA  Wade Wan  Broadcom Corporation 5300 California Ave Irvine, CA 92617 USA | Tel: Email:  Tel:  Email:  Tel:  Email: | +1-949-926-5388 [bheng@broadcom.com](mailto:bheng@broadcom.com)  +1-858-521-4835  [peisongc@broadcom.com](mailto:peisongc@broadcom.com)  +1-949-926-6387  [wwan@broadcom.com](mailto:wwan@broadcom.com) |
| *Source:* | Broadcom Corporation | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This contribution describes a mismatch between the HEVC working draft (WD5 d6) and HEVC reference software (HM-5.0) related to spatial motion vector predictors. A possibly unintentional behaviour related to the derivation of spatial motion vector predictors in both the working draft and the reference software is presented. Two possible fixes are presented as alternative solutions to the current mismatch.

# Problem Statement

## Derivation of spatial motion vector predictor candidates as described in HEVC WD5 d6

One step in the HEVC motion vector prediction process involves selecting two spatial neighbor candidates: a left neighbor vector (mvLXA) and a top vector (mvLXB). These vectors come from the locations shown in Figure 1 (A0, A1, B0, B1, B2).



Figure 1 – Spatial motion vector neighbours

This document focuses only on the case where A0 and A1 are either unavailable or intra coded. In other words, isScaledFlagLX is set equal to 0 in the following step of Section 8.4.2.1.7 from WD5.

3. ) If one of the following conditions is true, the variable isScaledFlagLX is set equal to 1.

* the prediction unit covering luma location ( xA0, yA0 ) is available and PredMode is not MODE\_INTRA.
* the prediction unit covering luma location ( xA1, yA1 ) is available and PredMode is not MODE\_INTRA.

In the WD description, the derivation of the top motion vector candidate mvLXB includes the following two steps:

The motion vector mvLXB and the availability flag availableFlagLXB are derived in the following ordered steps:

1. For ( xBk, yBk ) from ( xB0, yB0 ) to ( xB2, yB2 ) where xB0 = xP +nPSW, xB1 = xB0 - MinPuSize , and xB2 =  xP - MinPuSize, the following applies repeatedly until availableFlagLXB is equal to 1:

* If the prediction unit covering luma location ( xBk, yBk ) is available, PredMode is not MODE\_INTRA, predFlagLX[ xBk ][ yBk ] is equal to 1, and the reference index refIdxLX[ xBk ][ yBk ] is equal to the reference index of the current prediction unit refIdxLX, availableFlagLXB is set equal to 1 and the motion vector mvLXB is set equal to the motion vector mvLX[ xBk ][ yBk ], refIdxB is set equal to refIdxLX[ xBk ][ yBk ] and ListB is set equal to LX.
* Otherwise if the prediction unit covering luma location ( xBk, yBk ) is available, PredMode is not MODE\_INTRA, predFlagLY[ xBk ][ yBk ] (with Y = !X) is equal to 1 and PicOrderCnt( RefPicListLY( refIdxLY[ xBk ][ yBk ] ) ) is equal to PicOrderCnt( RefPicListLX( refIdxLX ) ), availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLY[ xBk ][ yBk ], refIdxB is set equal to refIdxLY[ xBk ][ yBk ],  and ListB is set equal to LY.

1. When isScaledFlagLX is equal to 0, availableFlagLXB is set equal to 0 and for ( xBk, yBk ) from ( xB0, yB0 ) to ( xB2, yB2 ) where xB0 = xP +nPSW, xB1 = xB0 - MinPuSize , and xB2 =  xP - MinPuSize, the following applies repeatedly until availableFlagLXB is equal to 1:

* If the prediction unit covering luma location ( xBk, yBk ) is available, PredMode is not MODE\_INTRA, predFlagLX[ xBk ][ yBk ] is equal to 1, availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLX[ xBk ][ yBk ], refIdxB is set equal to refIdxLX[ xBk ][ yBk ], ListB is set equal to LX.
* Otherwise if the prediction unit covering luma location ( xBk, yBk ) is available, PredMode is not MODE\_INTRA, predFlagLY[ xBk ][ yBk ] (with Y = !X) is equal to 1, availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLY[ xBk ][ yBk ], refIdxB is set equal to refIdxLY[ xBk ][ yBk ], ListB is set equal to LY.

Step 4 above will search B0, B1, B2 to find the first neighbor vector that has the same reference picture as the current vector.

Step 5 will then search B0, B1, B2 again to find any neighbor vector (whether it points to the same reference picture or not) and will scale that vector if necessary.

Assume that a valid vector was selected in Step 4, and as mentioned above, assume that isScaledFlagLX is equal to 0. The text marked in red above will proceed to reset the availability flag availableFlagLXB that was set in Step 4. Therefore, Step 5 will proceed to overwrite mvLXB.

We assume it was not the intended behavior to discard the vector found in Step 4 in favor of a second vector found in Step 5. Therefore this description seems incorrect.

## Derivation of spatial motion vector predictor candidates as implemented in HM5.0

The following code from HM 5.0 TComDataCU::fillMvpCand implements the process described in the WD Step 4 above:

// Above predictor search

bAdded = xAddMVPCand( pInfo, eRefPicList, iRefIdx, uiPartIdxRT, MD\_ABOVE\_RIGHT);

if (!bAdded)

{

bAdded = xAddMVPCand( pInfo, eRefPicList, iRefIdx, uiPartIdxRT, MD\_ABOVE);

}

if(!bAdded)

{

bAdded = xAddMVPCand( pInfo, eRefPicList, iRefIdx, uiPartIdxLT, MD\_ABOVE\_LEFT);

}

The following code from HM 5.0 TComDataCU::fillMvpCand implements the process described in the WD Step 5 above:

bAdded = bAddedSmvp;

if (pInfo->iN==2) bAdded = true;

if(!bAdded)

{

bAdded = xAddMVPCandOrder( pInfo, eRefPicList, iRefIdx, uiPartIdxRT, MD\_ABOVE\_RIGHT);

if (!bAdded)

{

bAdded = xAddMVPCandOrder( pInfo, eRefPicList, iRefIdx, uiPartIdxRT, MD\_ABOVE);

}

if(!bAdded)

{

bAdded = xAddMVPCandOrder( pInfo, eRefPicList, iRefIdx, uiPartIdxLT, MD\_ABOVE\_LEFT);

}

}

The code in red will again reset the availability flag that was set in Step 4. The HM code will then proceed to find a second scaled vector. However, this vector will be added to candidate list. So, the HM code will use two top vector candidates with one non-scaled and one scaled.

Based on our understanding of this code, the HM implementation does not match the WD description, and neither one seems to match the intended behavior.

# Suggested Fix 1

Our understanding is that the original intent was to find one and only one top vector candidate. To accomplish this, we suggest modifying the description so that the process stops searching for scaled top candidates as soon as a non-scaled version was found. For example:

1. When isScaledFlagLX is equal to 0 and availableFlagLXB is equal to 0, for ( xBk, yBk ) from ( xB0, yB0 ) to ( xB2, yB2 ) where xB0 = xP +nPSW, xB1 = xB0 - MinPuSize , and xB2 =  xP - MinPuSize, the following applies repeatedly until availableFlagLXB is equal to 1:

* If the prediction unit covering luma location ( xBk, yBk ) is available, PredMode is not MODE\_INTRA, predFlagLX[ xBk ][ yBk ] is equal to 1, availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLX[ xBk ][ yBk ], refIdxB is set equal to refIdxLX[ xBk ][ yBk ], ListB is set equal to LX.
* Otherwise if the prediction unit covering luma location ( xBk, yBk ) is available, PredMode is not MODE\_INTRA, predFlagLY[ xBk ][ yBk ] (with Y = !X) is equal to 1, availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLY[ xBk ][ yBk ], refIdxB is set equal to refIdxLY[ xBk ][ yBk ], ListB is set equal to LY.

Similarly, the HM implementation could be modified as follows

bAdded = bAdded | bAddedSmvp;

# Suggested Fix 2

If it is preferable to extend the original intent and select two top candidates in cases where isScaledFlagLX is equal to 0, we suggest that it would be more logical to give priority to the non-scaled candidates (Step 4) first before moving to the scaled candidates (Step 5).

As currently implemented in HM 5.0, the software will find the first non-scaled vector, ignore any remaining non-scaled vectors, and immediately start searching for a scaled vector. It seems suboptimal to ignore those additional non-scaled candidates in favor of selecting a scaled vector instead.

For example, the HM code could be modified as follows to accomplish this:

// Above predictor search

xAddMVPCand( pInfo, eRefPicList, iRefIdx, uiPartIdxRT, MD\_ABOVE\_RIGHT);

if(pInfo->iN < 2)

{

xAddMVPCand( pInfo, eRefPicList, iRefIdx, uiPartIdxRT, MD\_ABOVE);

}

if(pInfo->iN < 2)

{

xAddMVPCand( pInfo, eRefPicList, iRefIdx, uiPartIdxLT, MD\_ABOVE\_LEFT);

}

This way, if two or more neighbor vectors used the same reference picture as the current vector, those non-scaled vectors would be added to the list before moving on to consider any scaled vectors.

# Conclusion

This contribution recommends fixing the inconsistency between the working draft and the reference software on the derivation of spatial motion vector predictors. There appears to be unintentional behavior when the left neighboring blocks of the current block are either unavailable or coded in intra modes with the methods described both in the working draft and the reference software. Therefore, two alternative solutions are suggested to solve this inconsistency.

# References

1. [B. Bross](mailto:benjamin.bross@hhi.fraunhofer.de), [W.-J. Han](mailto:wjhan.han@samsung.com), [J.-R. Ohm](mailto:ohm@ient.rwth-aachen.de), [G. J. Sullivan](mailto:garysull@microsoft.com), [T. Wiegand](mailto:thomas.wiegand@hhi.fraunhofer.de), “WD5: Working Draft 5 of High-Efficiency Video Coding,” Document of Joint Collaborative Team on Video Coding, JCTVC-G1103\_d5, Jan. 2012.

# Patent rights declaration(s)

**Broadcom, Inc. 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).**