**Joint Collaborative Team on Video Coding (JCT-VC)**
**of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**
6th Meeting: Torino, 14-22 July, 2011

Document: JCTVC-F256

*Title:* Improving Deblocking filter efficiency

*Status:* Input Document to JCT-VC

*Purpose:* Proposal

*Author(s) or Contact(s):* Mangesh Sadafale
Texas Instruments Inc., India
Tel:
Email: mangesh@ti.com

*Source:* Texas Instruments Inc.

# 1 Abstract

In this contribution we are proposing two changes to HM-3.0 Deblocking filter.

1> In HM 3.0 the chroma filter is highly quantize and applied to only Intra CU. Also it's implementation is different from that of Luma filter. Here we are proposing a change in Chroma filter to improve it's harmony with Luma-Weak filter and improve BD-rate.

2> In HM-3.0, vertical and horizontal filter decision is made parallel, but it's usage is still sequential. This has resulted in extra stage and extra storage with no BD-rate improvement. This overhead in the design decreases the efficiency of the Deblocking filter in LCU based processing. Hence we propose the removal of it in HM-4.0.

INTEL-1045
9,641,841



Date Saved: 2011-07-16

# 2 Problem Statement

## *2.1 Improving Chroma Filter*

There are different types of filter used in Deblocking of HEVC (JVT-VC HM-3.0) for Luma and Chroma. Based on pixel strength there are two types of filters (Strong/Weak) applied to Luma. Where as for Chroma there is one filter which is applied only if edge belongs to Intra CU (Coding-Unit).

Following are three different filters used in HM 3.0 Deblocking.

1: Luma Strong Filter

```
piSrc[-iOffset] = Clip(( m1 + 2*m2 + 2*m3 + 2*m4 + m5 + 4) >> 3 );
piSrc[0] = Clip(( m2 + 2*m3 + 2*m4 + 2*m5 + m6 + 4) >> 3 );
piSrc[-iOffset*2] = Clip(( m1 + m2 + m3 + m4 + 2)>>2);
piSrc[ iOffset] = Clip(( m3 + m4 + m5 + m6 + 2)>>2);
piSrc[-iOffset*3] = Clip(( 2*m0 + 3*m1 + m2 + m3 + m4 + 4 )>>3);
piSrc[ iOffset*2] = Clip(( m3 + m4 + m5 + 3*m6 + 2*m7 +4 )>>3);
```

2. Luma Weak Filter

```
Int delta = Clip3(-tc, tc, ((13*(m4-m3) + 4*(m5-m2) - 5*(m6-m1)+16)>>5) );
piSrc[-iOffset] = Clip(m3+delta);
piSrc[0] = Clip(m4-delta);
piSrc[-iOffset*2] = Clip(m2+delta/2);
piSrc[ iOffset] = Clip(m5-delta/2);
```

3. Chroma Filter

```
delta = Clip3(-tc,tc, (((( m4 - m3 ) << 2 ) + m2 - m5 + 4 ) >> 3) );
piSrc[-iOffset] = Clip(m3+delta);
piSrc[0] = Clip(m4-delta);
```

If we check above three filters, Chroma filter is the weakest filter (lowest filter-taps) with highly quantized coefficients which is applied to only Intra-edge and has no similarity with Luma filter results in more logic in hardware.

## *2.2 Parallel Decision*

Parallel decision was introduced in HM-3.0 to dissolve dependency in filter decision which was present in HM-2.0. But, Parallel decision is effective for those sections of pixels where both vertical and horizontal filter is applied. Over and above it will be effective only when vertical-edge is applied with strong filter, with weak filter the pixels used for horizontal decision are not modified.

Following table gives the filtered pixels count for two case 1> strong-filter to both edge 2> weak filter to both edge. From table we can see the impact of having sequential processing of vertical and horizontal decision and filter is small. Even if we apply Filter to all 8x8-edge of Luma and chroma, 189MHz is enough with 1-line/clock hardware. Thus performance (execution time) has no impact because of sequential dependency. So from performance point of view there is no improvement because of parallel decision.

**Table 2 : Filtered Pixels**

| TuSize | % pixel | VER_EDGE & HOR_EDGE | VER_EDGE | HOR_EDGE | No-Filter |
|---|---|---|---|---|---|
| | # pixels modifed by Strong Filter | | | | |
| 8 | 56.3% | 36 | 12 | 12 | 4 |
| 16 | 14.1% | 36 | 60 | 60 | 100 |
| 32 | 3.5% | 36 | 156 | 156 | 676 |
| | # pixels modifed by Weak Filter | | | | |
| 8 | 6.3% | 4 | 12 | 12 | 36 |
| 16 | 1.6% | 4 | 28 | 28 | 196 |
| 32 | 0.4% | 4 | 60 | 60 | 900 |

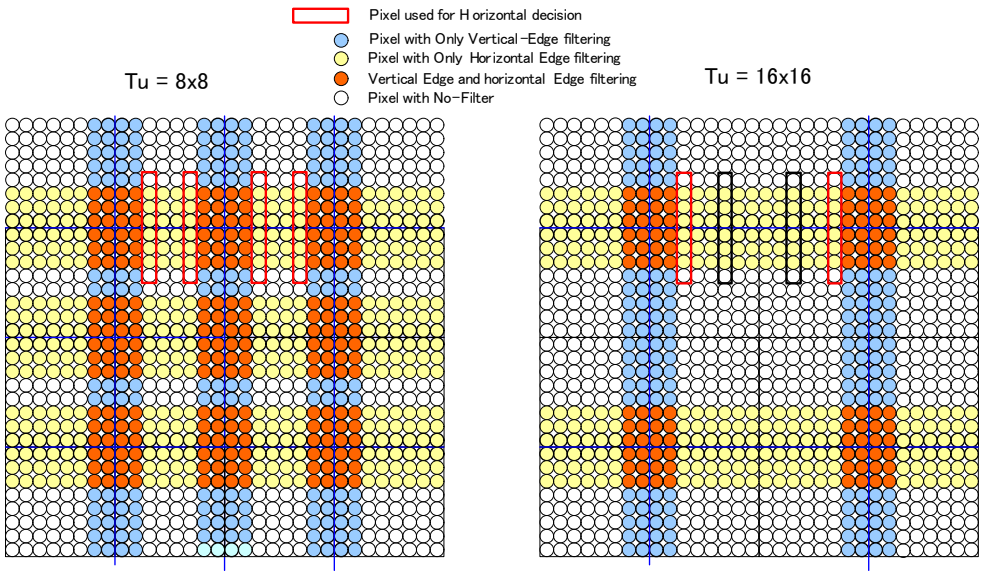


**Figure 1 : With Strong Filter**

The Parallel decision use unfiltered data to decide filter's ON/OFF and filter type (Stong/Weak) for horizontal edge. With this, decisions of filter On/Off and filters type are parallel, but it's usage for horizontal-edge has to be done only after completing vertical-edge. So, we have to store (yellow highlighted code) filter-decision (1-bit per 8-line) and filter-type (1-bit per line) used by all vertical edges in LCU. For Top-LCU, we have to store data for the filter-decision (two Pel per 8-line used in xCalcD()) and filter-type (two Pel "m3j" and "m0j" per line) for bottom-row of LCU. Because of extra memory storage and memory access, it's going to increase memory footprint, memory management and decrease performance. Figure 1 shows the extra stages and extra storage required in HM-3.0 as compare to HM-2.0.

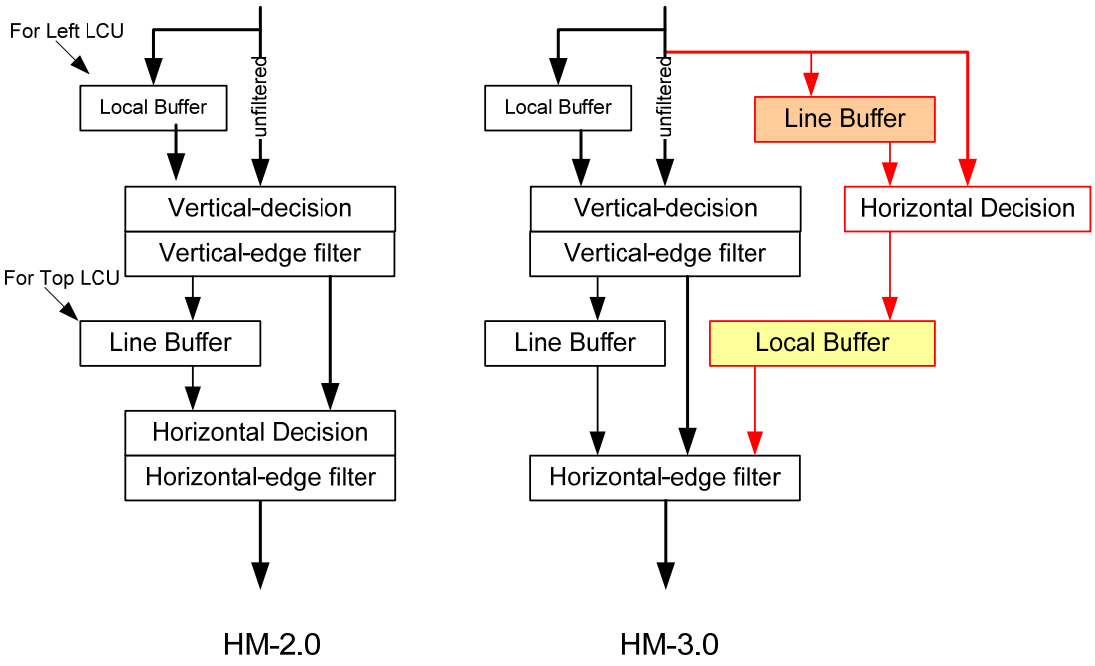


**Figure 2 : Difference in HM-2.0 and HM-3.0**

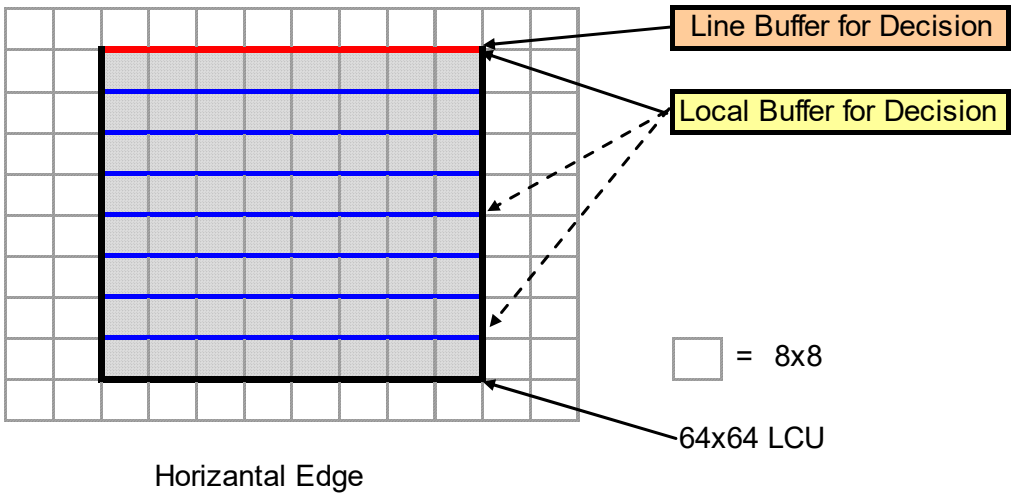


**Figure 3 : LCU based processing with all TuSize = 8x8**

If we take example of LCU based processing with CUSize=64 with TuDepth= 4, then we have 64 horizantal-edge and 64-vertical-edge (8-line/per edge).

For all internal edges, we need 64-bits to store filter ON/OFF decision of 64 Horizontal-edge. Also we will need 512-bit (64*8) to store strong/weak filter decision for each vertical line.

For Top-Edge (Red -> bottom of current LCU) we will need to store data in line buffer for Filter On/Off decision and Strong/Weak filter for complete row which will be use while processing lower LCU.

| Buffer Type | Data Type | bits/LCU | bits/line | Note |
|---|---|---|---|---|
| Local | ON/OFF | 64 | NA | 1-bit/8-line |
| | Strong/Weak | 512 | NA | 1-bit/1-line |
| Line | ON/OFF | 80 | 5,120 | (Pel+2)/8-line |
| | Strong/Weak | 1,024 | 65,536 | 2*Pel/1-line |
| | Total | 1,680 | 70,656 | |

**Table 1: Storage requirement for 4Kx2K**

- **Local Storage of ON/OFF decision per 8-line**

```
Int iD =
xCalcD(piTmpSrcJudge+iSrcStep*(iIdx*uiPelsInPart+iBlkIdx*DEBLOCK_SMALLEST_BLOCK+2),iOffset) +
xCalcD(piTmpSrcJudge+iSrcStep*(iIdx*uiPelsInPart+iBlkIdx*DEBLOCK_SMALLEST_BLOCK+5),iOffset);

        if (iD < iBeta)  // ON/OFF decision to be stored
```

- **Local Storage of Strong/Weak filter decision per 1-line**

```
  Pel m4j  = piSrcJudge[0];
  Pel m3j  = piSrcJudge[-iOffset];
  Pel m7j  = piSrcJudge[ iOffset*3];
  Pel m0j  = piSrcJudge[-iOffset*4];

  d_strong = abs(m0j-m3j) + abs(m7j-m4j);

  if ((d_strong<(beta>>3))&&(iD<(beta>>2)) && (abs(m3j-m4j) < ((tc*5+1)>>1))) //Strong/Weak
```

- **Storage for ON/OFF decision per 8-line of TOP-Edge in line buffer**

```
__inline Int TComLoopFilter::xCalcD( Pel* piSrc, Int iOffset)
{
  return abs( piSrc[-iOffset*3] - 2*piSrc[-iOffset*2] + piSrc[-iOffset] ) +
         abs( piSrc[0] - 2*piSrc[iOffset] + piSrc[iOffset*2] );
}
```

- **Storage for Strong/Weak filter decision per 1-line of TOP-Edge in line buffer**

```
  Pel m3j  = piSrcJudge[-iOffset];
  Pel m0j  = piSrcJudge[-iOffset*4];
```

Parallel decision is adding huge storage, memory access/latency and sequential dependency with no gain in BD-rate. Parallel decision in HM-3.0 is questionable; hence we propose the removal of it in HM-4.0.

# 3 Proposal

## *3.1 Improving Chroma Filter*

Here we have two proposals:

### 3.1.1 Increasing Chroma filter precision

We are proposing to increase filter coefficient precision by 2-bit. It will make Chroma filter sub-set of Luma weak filter and this will improve the BD-Rate also.

*Chroma HM-3.x* : `delta = Clip3(-tc,tc, ((((m4 - m3)<<2) + m2 - m5 + 4 )>>3) );`

*Chroma proposed* : `delta = Clip3(-tc,tc,((13*(m4-m3) - 4*(m5-m2) + 16)>>5) );`

*Luma Weak filter* : `delta = Clip3(-tc,tc,((13*(m4-m3) + 4*(m5-m2) - 5*(m6-m1)+16)>>5));`

This change has no impact on software as total number of operations remains same. In Hardware this will reduce the implementation complexity by sharing same logic for Luma-weak filter and Chroma filter.

### 3.1.2 Apply Chroma filter to INTER CU

Here we are proposing to apply chroma filter to INTER CU (same as AVC/H.264). If we apply filter to all edges of Luma and Chroma in 4Kx2K@60fps, 189MHz will be enough with 1-line/clock. With 2-line/clock 95MHz will be enough, it's not a big impact for hardware. In software, the extra computation is a fraction of the total required computation. As a result BD-rate gain to extra computation is very high. Also this extra gain is without any impact on Encoder.

Condition under which chroma filter is applied is as given below. This condition is also subset of Luma Strong/Weak selection.

```
Int d_inter = abs(m0-m3) + abs(m7-m4);
  if ( (bs>2) || ((d_inter < (beta>>3)) && (abs(m3-m4) < ((tc*5+1)>>1))) )  {
```

## *3.2 Remove Parallel Decision*

Here proposal is to remove parallel decision and make it same as HM-2.0. Which will decrease memory requirement and it's sequential dependency, improve performance and gives non-zero BD-rate improvement.

# 4 Required Changes in Working Draft

## *4.1 Chroma Filter changes*

In Section 8.6.1.4.2 (Filtering process for chroma block edge) following change is required

*2. If bS is greater than 0, the following ordered steps apply:*

In section 8.6.1.4.4 (Filtering process for a chroma sample) "E603_d5" needs following changes :

*When the variable bS is greater than 0, the filtered sample values $p_0'$ and $q_0'$ are derived by*

- *If bS greater then "2" or $|p_3 - p_0| + |q_0 - q_3|$ is less than $(\beta >> 3)$ and $|p_0 - q_0|$ is less than $(5*t_C + 1) >> 1$, the following filtering is applies :*

$$\Delta = Clip3(-tc,tc,((13*(m4-m3) - 4*(m5-m2) + 16)>>5) ); \quad (8\text{-}461)$$

## *4.2 parallel decision*

Revert back the section " 8.6.1.4 Filtering process for coding unit" to "JCTVC-E603-d5"

# 5 Simulation Results in HM-3.0

## 5.1 Chroma Filter changes

| | Intra | | | Intra LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | 0.0 | -0.4 | -0.2 | 0.0 | -0.6 | -0.4 |
| Class B | 0.0 | -0.2 | -0.1 | 0.0 | -0.7 | -0.5 |
| Class C | 0.0 | -0.2 | -0.2 | 0.0 | -0.7 | -0.6 |
| Class D | 0.0 | -0.2 | -0.1 | 0.0 | -0.6 | -0.5 |
| Class E | 0.0 | 0.1 | 0.1 | 0.0 | 0.1 | -0.1 |
| All | 0.0 | -0.2 | -0.1 | 0.0 | -0.5 | -0.5 |

| | Random access | | | Random access LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | -0.1 | -0.6 | -0.6 | 0.1 | -0.7 | -0.4 |
| Class B | 0.0 | -0.6 | -0.5 | 0.0 | -0.7 | -0.6 |
| Class C | 0.0 | -0.6 | -0.5 | 0.0 | -0.6 | -0.6 |
| Class D | 0.0 | -0.5 | -0.5 | 0.0 | -0.4 | -0.6 |
| Class E | | | | | | |
| All | 0.0 | -0.6 | -0.5 | 0.0 | -0.6 | -0.5 |

| | Low delay | | | Low delay LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | | | | | | |
| Class B | 0.0 | -0.7 | -1.0 | 0.0 | -0.9 | -1.0 |
| Class C | 0.0 | -0.8 | -0.8 | 0.0 | -0.5 | -0.8 |
| Class D | 0.0 | -0.2 | -0.4 | 0.0 | -0.4 | -0.6 |
| Class E | 0.1 | -0.8 | -1.4 | 0.0 | -1.9 | -1.5 |
| All | 0.0 | -0.6 | -0.9 | 0.0 | -0.9 | -1.0 |

## 5.2 Removing Parallel Decision

| | Intra | | | Intra LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Class B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Class C | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Class D | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Class E | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| All | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

| | Random access | | | Random access LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | 0.0 | -0.2 | 0.2 | 0.0 | -0.1 | 0.2 |
| Class B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Class C | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.1 |
| Class D | 0.0 | 0.0 | -0.1 | 0.0 | 0.0 | 0.0 |
| Class E | | | | | | |
| All | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

| | Low delay | | | Low delay LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | | | | | | |
| Class B | 0.0 | 0.2 | 0.1 | 0.0 | 0.0 | -0.3 |
| Class C | 0.0 | -0.1 | -0.1 | 0.0 | 0.1 | -0.1 |
| Class D | 0.0 | -0.1 | 0.2 | 0.0 | 0.1 | -0.1 |
| Class E | 0.0 | 0.2 | 0.2 | 0.1 | -0.2 | -0.7 |
| All | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | -0.3 |

## *5.3 Removing Parallel Decision and Chroma Filter changes*

| | Intra | | | Intra LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | 0.0 | -0.4 | -0.2 | 0.0 | -0.6 | -0.4 |
| Class B | 0.0 | -0.2 | -0.1 | 0.0 | -0.7 | -0.5 |
| Class C | 0.0 | -0.2 | -0.2 | 0.0 | -0.7 | -0.6 |
| Class D | 0.0 | -0.2 | -0.1 | 0.0 | -0.6 | -0.5 |
| Class E | 0.0 | 0.1 | 0.1 | 0.0 | 0.1 | -0.1 |
| All | 0.0 | -0.2 | -0.1 | 0.0 | -0.5 | -0.5 |
| Enc Time[%] | 105% | | | 102% | | |
| Dec Time[%] | 104% | | | 101% | | |

| | Random access | | | Random access LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | 0.0 | -0.6 | -0.6 | 0.0 | -0.7 | -0.4 |
| Class B | 0.0 | -0.6 | -0.5 | 0.0 | -0.7 | -0.6 |
| Class C | 0.0 | -0.4 | -0.5 | 0.0 | -0.6 | -0.7 |
| Class D | 0.0 | -0.5 | -0.5 | 0.0 | -0.7 | -0.5 |
| Class E | | | | | | |
| All | 0.0 | -0.6 | -0.5 | 0.0 | -0.7 | -0.6 |
| Enc Time[%] | 100% | | | 102% | | |
| Dec Time[%] | 99% | | | 101% | | |

| | Low delay | | | Low delay LC | | |
|---|---|---|---|---|---|---|
| | Y BD-rate | U BD-rate | V BD-rate | Y BD-rate | U BD-rate | V BD-rate |
| Class A | | | | | | |
| Class B | -0.1 | -0.8 | -0.8 | 0.0 | -1.0 | -1.1 |
| Class C | -0.1 | -0.8 | -0.8 | -0.1 | -0.6 | -0.8 |
| Class D | 0.0 | -0.3 | 0.0 | 0.0 | -0.4 | -0.5 |
| Class E | 0.0 | -1.2 | -1.3 | -0.1 | -2.4 | -1.9 |
| All | 0.0 | -0.7 | -0.7 | 0.0 | -1.0 | -1.0 |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 100% | | | 99% | | |

# 6 Patent rights declaration(s)

Texas Instruments Inc. may have IPR 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).