|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  11th Meeting: Shanghai, CN, 10-19 Oct., 2012 | Document: JCTVC-K0035 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Description of scalable video coding technology proposal by Qualcomm (configuration 1)** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jianle Chen Krishna Rapaka Xiang Li Vadim Seregin Liwei Guo Marta Karczewicz Geert Van der Auwera Joel Sole Xianglin Wang Chengjie Tu Ying Chen  5775 Morehouse Dr. San Diego, CA 92121, USA | Tel: Email: | +1-858-651-8028  [cjianle@qti.qualcomm.com](mailto:cjianle@qti.qualcomm.com) [krapaka@qti.qualcomm.com](mailto:krapaka@qti.qualcomm.com) [lxiang@qti.qualcomm.com](mailto:lxiang@qti.qualcomm.com)  [vseregin@qti.qualcomm.com](mailto:vseregin@qti.qualcomm.com) [liweig@qti.qualcomm.com](mailto:liweig@qti.qualcomm.com)  [martak@qti.qualcomm.com](mailto:martak@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This proposal is Qualcomm’s response to the joint call-for-proposal on scalable video coding extension of HEVC issued by MPEG and ITU-T. The proposed solution is developed based on HM6.1 and the base layer coding is kept unchanged. Under multi-loop decoding framework, coding tools are proposed to improve the coding efficiency of the enhancement layer by utilizing base layer information, including reconstructed pixel samples.

Compared to HEVC simulcast, the luma BD-rate saving (based on actual total rate) is 25.6%, 33.5%, 23.0%, 31.1%, and 28.2 %, respectively, for all intra 2x, all intra 1.5x, random access 2x, random access 1.5x, and random access SNR cases.

The luma BD-rate saving (based on actual enhancement layer rate) is 36.7%, 55.5%, 33.0%, 50.2%, and 41.4 %, respectively, for all intra 2x, all intra 1.5x, random access 2x, random access 1.5x, and random access SNR cases.

# Contents

[Abstract 1](#_Toc336871989)

[Contents 2](#_Toc336871990)

[1 Introduction 3](#_Toc336871991)

[1.1 Definitions 3](#_Toc336871992)

[2 Algorithm description 3](#_Toc336871993)

[2.1 CTU/TU/PU partitioning 3](#_Toc336871994)

[2.2 Up-sampling filters 4](#_Toc336871995)

[2.2.1 Fixed filters 4](#_Toc336871996)

[2.2.2 Adaptive filter 5](#_Toc336871997)

[2.3 Inter-layer prediction 6](#_Toc336871998)

[2.3.1 Intra-BL prediction 6](#_Toc336871999)

[2.3.2 Intra prediction 6](#_Toc336872000)

[2.3.3 Intra residual prediction 6](#_Toc336872001)

[2.3.4 Inter residual prediction 7](#_Toc336872002)

[2.3.5 Syntax Elements Prediction 8](#_Toc336872003)

[2.4 In-loop filtering 9](#_Toc336872004)

[2.4.1 De-blocking filter 9](#_Toc336872005)

[2.4.2 SAO 9](#_Toc336872006)

[2.5 Spatial transforms 10](#_Toc336872007)

[2.6 Quantization 10](#_Toc336872008)

[2.7 Motion/mode/parameter coding 10](#_Toc336872009)

[2.8 Entropy coding of residue / transform coefficients 10](#_Toc336872010)

[2.9 High-level syntax 11](#_Toc336872011)

[3 Syntax and semantics description 11](#_Toc336872012)

[3.1 Coding tree syntax 11](#_Toc336872013)

[3.2 Coding tree semantics 11](#_Toc336872014)

[3.3 Coding unit syntax 13](#_Toc336872015)

[3.4 Coding unit semantics 14](#_Toc336872016)

[4 Compression performance discussion 14](#_Toc336872017)

[4.1 Encoding setting 14](#_Toc336872018)

[4.1.1 QP settings 14](#_Toc336872019)

[4.2 Category 1 (HEVC base layer) 15](#_Toc336872020)

[4.2.1 BD-rate over simulcast based on EL+BL actual rate 15](#_Toc336872021)

[4.2.2 BD-rate over simulcast based on EL+BL target rate 15](#_Toc336872022)

[4.2.3 BD-rate savings over simulcast based on EL-only actual rate 16](#_Toc336872023)

[4.2.4 BD-rate savings over simulcast based on EL-only target rate 16](#_Toc336872024)

[5 Complexity analysis 17](#_Toc336872025)

[5.1 Encoding time and measurement methodology 17](#_Toc336872026)

[5.2 Decoding time and measurement methodology and comparison vs. anchor bitstreams decoded by HM 17](#_Toc336872027)

[5.3 Description of computing platform used to determine encoding and decoding times reported in sections 5.1 and 5.2 17](#_Toc336872028)

[5.4 Expected memory usage of encoder 17](#_Toc336872029)

[5.5 Expected memory usage of decoder 17](#_Toc336872030)

[5.6 Complexity characteristics of encoder motion estimation and partitioning selection in enhancement layer(s) 18](#_Toc336872031)

[5.7 Complexity characteristics of decoder motion compensation in enhancement layer(s) 18](#_Toc336872032)

[5.8 Complexity characteristics of encoder intra-frame prediction type and partitioning selection in enhancement layer(s) 18](#_Toc336872033)

[5.9 Complexity characteristics of decoder intra-frame prediction operation in enhancement layer(s) 18](#_Toc336872034)

[5.10 Complexity characteristics of up-sampling filters and transforms specific in enhancement layer(s) 18](#_Toc336872035)

[5.11 Complexity characteristics of quantization and inverse quantization in enhancement layer(s) 18](#_Toc336872036)

[5.12 Complexity characteristics of encoder entropy coding operation in enhancement layer(s) 18](#_Toc336872037)

[5.13 Complexity characteristics of decoder entropy decoding operation in enhancement layer(s) 18](#_Toc336872038)

[5.14 Degree of capability for encoder parallel processing 18](#_Toc336872039)

[5.15 Degree of capability for decoder parallel processing 19](#_Toc336872040)

[6 Software implementation description 19](#_Toc336872041)

[7 Highlighted aspects discussion 19](#_Toc336872042)

[8 Closing remarks 19](#_Toc336872043)

[9 References 19](#_Toc336872044)

[10 Patent rights declaration 19](#_Toc336872045)

# Introduction

This proposal covers the following categories (tick box ⌧)

⌧ Category 1 (HEVC base layer) spatial scalability

⌧ Category 1 (HEVC base layer) intra-only spatial scalability

⌧ Category 1 (HEVC base layer) SNR scalability

🞏 Category 2 (AVC base layer) spatial scalability

⌧ The proposal obeys the constraints under section 5 of the CfP (if box is not ticked, explain cases where constraints are violated)

## Definitions

|  |  |
| --- | --- |
| HM6.1: | HEVC test model 6.1, main profile |
| HEVC SVC: | HEVC scalable video coding extension |
| BL: | base layer, the reference layer of the current layer. In this proposal, the base layer is HEVC compatible layer, with layer\_id equal to 0. |
| EL: | enhancement layer, a layer which is not the HEVC compatible base layer. |
| collocated BL block: | collocated block of the current block in the BL picture |

# Algorithm description

To obtain high coding efficiency for the enhancement layer, a framework of multi-loop decoding is employed so that all the information in the base layer, including reconstructed pixel samples, can be used to code the enhancement layer.

As requested in the call-for-proposals [1], base layer pictures are coded with HM6.1 [2, 3]. For enhancement layer coding, the following coding tools are proposed on top of the HEVC tools:

* **Up-sampling filters**: A process to generate full EL resolution picture by up-sampling or filtering the reconstructed BL picture.
* **Intra-BL prediction**: A coding tool that uses collocated BL block to predict EL samples.
* **Inter-layer residual prediction**: A coding tool that predicts EL predicted residues based on base layer information. This method is applied to both intra and inter Coding Units (CUs).
* **Inter-layer syntax prediction**: A coding tool that employs BL syntax elements, such as CU-split, motion parameter, and prediction mode, to predict EL syntax elements.
* **Switchable DCT/DST**: A coding tool that employs additional transforms (DCT and DST) to improve the transform efficiency of inter-layer predicted residues.

Note that most tools can be easily extended to single-loop decoding framework. Detailed descriptions of the newly added coding tools are provided in the following sub-sections.

## CTU/TU/PU partitioning

Asymmetric motion partition (AMP) is applied for enhancement layer coding.

## Up-sampling filters

The basic function of the up-sampling process is to generate a picture that has the resolution of the current layer picture from the BL picture. The sample position mapping for the up-sampling process used in this proposal is similar to that in H.264/AVC SVC. The fractional sample position, denoted as phase, is quantized in units of 1/16-pel. Therefore, an up-sampling filter set may employ up to sixteen filters. The zero phase offset is used for the top-left sample position of a picture to synchronize with the down-sampling process used to generate the test sequences [4].

In this proposal, different filter sets are used for each of the three inter-layer prediction modes described in section 2.2. Separate sets of fixed filters are designed for intra and inter residual prediction modes. Additionally, an adaptive filter set may be used for intra-BL mode. Note that the filtering process is also applied to the integer sample positions. As a result, filtering process is used in both SNR and spatial scalability cases.

### Fixed filters

An eight-tap filter set shown in Table 1 is used for the luma component in intra residual prediction. These filters are designed in a similar manner as the HEVC luma motion compensation interpolation filters. It should be noted that the filter coefficients for phases 4 and 8 exactly match the quarter-pel and half-pel HEVC luma interpolation filters. This filter set may also be used in the intra-BL prediction.

Table 1 Luma filter for inter-layer intra prediction modes

|  |  |
| --- | --- |
| Phase | Filter coefficients |
| 0/16 | { 0, 0, 0, 64, 0, 0, 0, 0 } |
| 1/16 | { 0, 1, -3, 63, 4, -2, 1, 0 } |
| 2/16 | { 0, 2, -6, 61, 9, -3, 1, 0 } |
| 3/16 | { -1, 3, -8, 60, 13, -4, 1, 0 } |
| 4/16 | { -1, 4, -10, 58, 17, -5, 1, 0 } |
| 5/16 | { -1, 4, -11, 53, 25, -8, 3, -1 } |
| 6/16 | { -1, 4, -11, 50, 29, -9, 3, -1 } |
| 7/16 | { -1, 4, -11, 45, 34, -10, 4, -1 } |
| 8/16 | { -1, 4, -11, 40, 40, -11, 4, -1 } |

The filter set applied for the luma component in inter residual prediction is shown in Table 2. This filter set is designed to significantly suppress the high frequency components to reduce the noise presents in the residual pictures. It should be noted that only filters with phases 0, 5, 8 and 11 are involved in the filtering process for SNR, spatial 1.5x and spatial 2.0x scalability cases. For the integer position (phase 0), filters with different frequency responses are used, respectively, in the SNR, spatial (1.5x and 2x) scalability cases.

Table 2 Luma filter for inter residual prediction mode

|  |  |
| --- | --- |
| Phase | Filter coefficients |
| 0/16 (SNR) | { -1, -2, 17, 36, 17, -2, -1, 0 } |
| 0/16 (2x) | { 0, -5, 16, 42, 16, -5, 0, 0 } |
| 8/16 (2x) | { 0, -4, 6, 30, 30, 6, -4, 0 } |
| 0/16 (1.5x) | { 0, -2, 8, 52, 8, -2, 0, 0 } |
| 5/16 (1.5x) | { 1, -5, 6, 38, 29, -3, -3, 1 } |

A four-tap filter set shown in Table 3 is used for the chroma component up-sampling process. The filter is applied to all the inter-layer prediction modes.

Table 3 Chroma filter

|  |  |
| --- | --- |
| Phase | Filter coefficients |
| 0/16 | { 0, 64, 0, 0 } |
| 1/16 | { -2, 62, 4, 0 } |
| 2/16 | { -2, 58, 0, -2 } |
| 3/16 | { -4, 56, 14, -2 } |
| 4/16 | { -4, 54, 16, -2 } |
| 5/16 | { -6, 52, 20, -2 } |
| 6/16 | { -6, 48, 26, -4 } |
| 7/16 | { -4, 42, 30, -4 } |
| 8/16 | { -4, 36, 36, -4 } |

### Adaptive filter

Compared to the fixed filter set, an adaptive filter set can effectively reduce the distortion between the original EL picture and the up-sampled/filtered BL picture, thereby improving the inter-layer prediction efficiency. In this proposal, an adaptive filter set is employed to process the luma component in the intra-BL mode. The filter set is applied in both SNR and spatial scalability cases. To simplify the filter derivation process, the number of filter taps is fixed to 8.

The filter set for intra-BL mode can be switched between adaptive filter set and the fixed filter from Table 1. A flag is signaled to the decoder at picture level to indicate the choice of the filter.

#### Filter derivation and determination

The adaptive filter parameters are derived at the encoder side and signaled to the decoder for each picture. The vertical filter and the horizontal filter are derived sequentially. First, the vertical filter is derived by solving

where is the vertical auto-correlation matrix of the original EL picture and is the vertical cross-correlation between the original EL picture and the reconstructed BL picture. In spatial scalability case, the original EL picture is horizontally down-sampled to match the width of the BL picture and used as the input to this derivation process.

The derived filter is used to vertically up-sample/filter the BL picture and the resulting picture is used to derive the horizontal filter.

The horizontal filter is derived by solving

where is the horizontal auto-correlation matrix of the original EL picture and is the horizontal cross-correlation between the original EL picture and the vertically up-sampled/filtered BL picture.

The BL picture is up-sampled/filtered using both the derived filter set and the fixed filter set from Table 1. The SSE and SATD between the filtered picture and the original EL picture are calculated. When both SSE and SATD for the derived filter are smaller than those for the fixed filter, the derived filter is used in intra-BL prediction for the current picture. Otherwise, the fixed filter set from Table 1 is used in intra-BL prediction. A flag is signaled to the decoder at picture level to indicate the choice of the filter. If the flag is one, the filter coefficients are also signaled.

#### Encoding of filter parameters

The filter coefficients are quantized to six bits, with the constraint that the sum of the coefficients is equal to 64. The quantized coefficients are predicted before coding. The horizontal filter coefficients are predicted by the corresponding fixed filter in Table 1. The vertical filter coefficients are predicted by the horizontal filter coefficients. The magnitude of the predicted residue for each coefficient is coded with an Exponential-Golomb code. A sign flag is signaled if the predicted residue is non-zero.

The coefficients sum of a filter is known to be equal to 64. Only seven coefficients instead of all eight coefficients need to be signaled for a filter, and the remaining one can be derived by subtracting the sum of the coded coefficients from 64. In this proposal, the coefficient that is supposed to be the largest one in the filter is chosen to be derived. As shown in Figure 1, let with represents the coefficients of a filter. The coefficient which is closest to the pixel position is supposed to be the largest one. Therefore the coefficient is not signaled for a filter with the phase smaller than 8, and the coefficient is chosen for the remained filters.

The filter coefficients overhead can be further reduced by taking into account the symmetry of the filter coefficients. For the filters with the phase equals to 0 or 8, only three coefficients are signaled and the remaining coefficients are all derived at the decoder side.

For the filter with phase equal to 0, the remaining coefficients are derived as

and the coefficient is derived in the same manner as done for the other filters.

For the filter with the phase equal to 8, the remaining coefficients are derived as

,



Figure 1 Up-sampling filter coefficients

## Inter-layer prediction

### Intra-BL prediction

Under the framework of multi-loop decoding, intra-BL mode indicates that the pixels in the collocated BL block are used as the prediction for the EL block.

Two CU level modes, intra-BL and intra-BL skip, are added. For CUs in intra-BL skip mode, no further information is signaled.

### Intra prediction

The intra prediction process of HEVC except that for the intra DC mode is used as it is for EL picture coding. The intra DC Prediction is modified to use the average of the collocated BL block’s pixel values to form the prediction signal instead of using its spatial neighboring pixels as defined in HEVC.

As part of inter-layer syntax prediction in this proposal, the intra prediction mode of the collocated BL block is used to refine the final Most Probable Mode (MPM) list. The intra prediction mode of the collocated BL block, iColBaseDir, is added to the final MPM list, candModeList[ x ] x=0..2, as shown below:

if iColBaseDir is not one of the candModeList[ x ] x=0..2 and iColBaseDir is not equal to Intra\_DC then,

candModeList[0] = iColBaseDir  
 candModeList[1] = candModeList[0]  
 candModeList[2] = candModeList[1]

The total number of candidates in the MPM list is limited to three as that in HEVC.

### Intra residual prediction

In the intra residual prediction mode, as shown in Figure 2, the difference between the pixels of current neighbors and that of collocated BL neighbors are used to generate a difference prediction based on the intra prediction mode. The generated difference prediction signal is added to the collocated BL block signal to form the final prediction.

The intra prediction method for difference signal remains unchanged with respect to HEVC except for the planar mode. After intra prediction is performed to get the difference prediction signal, the bottom-right portion of the difference prediction for the planar mode is set to zero (that is for each pixel(x, y) satisfying the condition (x + y) >= N-1, where N is the width of the current block).

The intra residual prediction mode is indicated by a flag *intra\_resi\_pred\_flag* at CU level.

Due to the high frequency nature of the difference signals, HEVC mode dependent intra smoothing process is disabled in intra residual prediction mode.



Figure 2 Intra residual prediction

In intra residual prediction, the MPM candidate list derivation process of HEVC is modified. The MPM derivation process is refined in a way that the horizontal or vertical mode has higher priority than the DC mode. Table 4 illustrates only the refinements in MPM list derivation process.

Table 4 MPM list refinement for intra residual prediction

|  |  |  |  |
| --- | --- | --- | --- |
|  | candModeList[0] | candModeList[1] | candModeList[2] |
| (LeftIntraDir == AboveIntraDir) &&  (LeftIntraDir == Not Angular) | Planar | Vertical | Horizontal |
| LeftIntraDir != AboveIntraDir | Angular | Planar | If Mode[0] == Vertical DirSet then Horizontal else Vertical |
| Planar | Angular | If Mode[1] == Vertical DirSet then Horizontal else Vertical |

### Inter residual prediction

In this technique, inter predicted residue of the collocated BL block is used to predict that of the current block. This method is applied to inter CUs, and skip mode CUs. The framework of this method is shown in Figure 3 where the case of uni-prediction is illustrated.

Let **B**e and **B**b denote the current block in the EL picture and its collocated BL block, respectively. Let **P**e0 denote the temporal prediction for the block **B**e obtained by using the motion vector MVe0 (where sub-index 0 refers to reference list 0). Similarly, **P**b0 represents the temporal prediction for the block **B**b obtained by using the same motion vector MVe0 in the BL reference picture.

Then, the inter predicted residue of the BL block is obtained as

Considering the temporal prediction **P**e0 for block **B**e, the final uni-prediction **P** for block **B**e is

,

where *w* is a the weighting factor, which takes the values 0, 0.5, or 1.

When extending this method to the bi-predictional case, the final prediction **P** for **B**e is formed as

where sub-index 1 indicates reference list 1.

The weighting factor *w* is signaled at CU level as a weighting index. The weighting index 0, 1 and 2 are used to indicate the weighting factors 0, 0.5 and 1, respectively. A truncated unary code in regular CABAC mode is used for the signaling.



**Figure 3 Inter residual prediction for uni-prediction**

### Syntax elements prediction

#### Inter-layer motion parameter prediction

As a part of inter-layer syntax prediction in this proposal, base layer MVs are used in enhancement layer coding.

In HEVC, motion vectors are compressed after being coded and such compressed motion vectors can be later used as TMVP for coding other frames. In this proposal, the uncompressed base layer MVs are used in enhancement layer coding. In other words, motion vector compression for a base layer is performed after it is used for inter-layer prediction.



Figure 4 Center3 block

More specifically, the base layer MVs used in inter-layer prediction are obtained at a location co-located to the Center3 block in EL as shown in Figure 4, which is the same as that used for TMVP derivation in HEVC.

Additionally, when the spatial resolutions of BL and EL pictures are different, the base layer MVs are scaled according to the spatial resolution ratio between the two layers. For example, when the ratio is 3/2, motion vector scaling is performed according to the equation.

In this proposal, base layer MVs are used for both merge mode and AMVP mode in enhancement layer coding. The derived base layer MV is inserted as the first candidate in merge list and added after the TMVP candidate in AMVP list.

Additionally, validity check is performed on BL reference index. If its reference picture is not present in the EL reference picture list, the base layer MV is marked as “unavailable” and is not used in merge candidate list. For a bi-directional MV candidate from base layer, this reference index validity check is done for each reference picture list respectively. If base layer MV is marked as “unavailable” in one reference list, the bi-directional MV candidate is converted into uni-directional before using. And, if base layer MV is marked as “unavailable” in both reference lists, the candidate is not used in merge list.

For merge list construction at an enhancement layer, a new type of artificial merge candidates is generated by adding offset values to the first MV candidate in the merge candidate list. Such candidates are called offset MV candidates. If a merge candidate list is empty, zero MV with zero reference indexes is inserted prior to generating offset MV candidates. In this proposal, the four possible offset values of include , , and , with and representing the offset value for the horizontal and vertical components of a motion vector respectively. Such offset MV candidates are placed right after combined MV candidates in an EL merge candidate list. The maximum number of merge and AMVP motion vector candidates is kept the same as in HEVC, i.e. five for merge and two for AMVP.

#### Inter-layer CU split prediction

Due to the fact that CU split information between corresponding BL and EL blocks is often correlated, the split information of an enhancement layer CU can be conditionally inferred in this proposal.

It is observed that in general small size CUs are chosen much less frequently in an EL picture than in its BL picture. The enhancement layer CUs tend to have a larger size than their collocated base layer CUs. As a result, under certain conditions the *split\_cu\_flag* for an enhancement layer CU can be inferred to save signaling overhead.

Let CUcur, CUleft, CUtop, and CUtopright represent a current CU and its left, top and top-right CUs at an EL picture. Let CUbase represent its collocated base layer CU. Let *Depcur*, *Depleft*, *Depto*p, *Deptopright* and *Depbase* represent the depth value of each of the CUs respectively. According to this proposal, *split\_cu\_flag* is inferred as false for CUcur when the following three conditions are all met.

* CUleft, CUtop, and CUtopright are all available

When *split\_cu\_flag* is inferred, it is not coded in bitstream. Otherwise, if any of the three conditions is not met, split\_cu\_flag is explicitly signaled in bitstream in the same way as that in HEVC.

## In-loop filtering

### De-blocking filter

The de-blocking filter process for the EL is as in HEVC, except for setting the filtering strength for additional intra-BL, intra-BL skip and intra residual prediction modes. In case both blocks are coded with intra-BL skip mode, the edge is not processed; otherwise, if at least one of the blocks is coded as normal intra mode, the filtering strength is set to 2; otherwise, if at least one of the blocks is coded in intra-BL or intra residual prediction mode, then the filtering strength for the luma edge is set to 1, and the filtering strength is set to 2 for the chroma edge.

### SAO

This part is kept unchanged from HEVC.

## Spatial transforms

The residue characteristics can be different from those of the HEVC residues when inter-layer prediction is used. In addition to the two transforms in HEVC (i.e., DCT Type-2 and DST Type-7), additional DCT and DST transforms are applied to the enhancement layer to achieve better residue energy compaction.

For inter residual prediction blocks, if the weighting factor *w* is equal to 1 and the CU size is equal to or smaller than 16×16, DST Type-3 transform is used for the luma component.

For intra BL blocks, the encoder selects one of the three candidate transforms for the luma component. The selection is signaled using *transform\_selection\_index* at CU level with truncated unary codes. The three transforms and their associated indices are shown in Table 5.

The transform for chroma components is DCT type-2, that is, the same as in the HEVC main profile.

Table 5: Transform candidates for intra BL mode

|  |  |
| --- | --- |
| *transform\_selection\_index* | Transform |
| 0 | DST Type-3 |
| 1 | DCT Type-2 |
| 2 | DCT Type-8 |

DCT Type-2 is the transform used in HM6.1. The transform matrix coefficients of the other transforms are derived using the following formulas and rounding to the nearest integer values.

* DST Type-3:

.

* DCT Type-8:

.

In the above formulas, *N* is the transform size, for *n=N-1* and otherwise.

## Quantization

This part is kept unchanged from HEVC.

## Motion/mode/parameter coding

This part is kept unchanged from HEVC

## Entropy coding of residue / transform coefficients

The entropy coding of the residue in enhancement layer is almost the same as in the HEVC main profile with two exceptions. First, the condition assigning single context to all high-frequency coefficients is removed. Second, the scan order assignment is modified.

The coefficients scanning patterns used for the enhancement layer are the same as in the HEVC main profile, wherein three coefficient scanning orders (diagonal, vertical and horizontal) are applied to TUs of size 4×4 and 8×8; and only diagonal scan is applied to larger TU sizes.

The selection of the scan order for 4×4 and 8×8 TUs is defined as follows:

* First, horizontal and vertical scans are selected for intra TUs based on prediction directions (according to the mode dependent coefficient scan in the HEVC main profile).
* Second, for all other 4×4 and 8×8 TUs (i.e., non-intra blocks and intra blocks that the scan selected in the first stage is neither horizonal scan nor vertical scan), the scan order is selected based on the horizontal and vertical gradients of reconstructed pixel values, , of the collocated BL block. Specifically, for the collocated BL block, its horizontal gradient and vertical gradient are computed using following equations:

If both gradients are smaller than a threshold, the diagonal scan is selected. Otherwise, the selected scan is horizontal scan if ; vertical scan if ; and diagonal scan in other cases.

The scan order is derived at both the encoder and the decoder, and thus, no signaling is needed.

## High-level syntax

Similar to SVC, pictures in different layers (namely layer representations in SVC) of the same time instance are coded into one access unit, in an ascending order of layer\_id. reserved\_zero\_5bits in NAL unit header is renamed as layer\_id.

Each layer has its own sequence parameter set and thus picture parameter set. Sub-bitstream extraction based on layer\_id to produce an operation point that has layers with lower layer\_id values is possible. In fact, such a sub-bitstream extractor is implemented in this response.

# Syntax and semantics description

This proposal inserts few flags and indices in the coding unit level syntax to indicate whether additional coding tools are applied for that unit. The modified part of the HEVC syntax (based on HEVC text specification draft 6 [2]) is highlighted in the following syntax tables.

The syntax design is mainly based on the HEVC syntax/semantics, thus the syntax structure and the related semantics of HEVC are reused with additions and modifications, highlighted in yellow. In sub-sections that provide semantics of only the newly introduced syntax elements, the text is not highlighted.

## Coding tree syntax

|  |  |
| --- | --- |
| coding\_tree( x0, y0, log2CbSize, cbDepth ) { | Descriptor |
| if( x0 + ( 1 << log2CbSize ) <= pic\_width\_in\_luma\_samples &&  y0 + ( 1 << log2CbSize ) <= pic\_height\_in\_luma\_samples &&  MinCbAddrZS[ x0 >> Log2MinCbSize ][ y0 >> Log2MinCbSize ] >=   SliceCbAddrZS &&  log2CbSize > Log2MinCbSize && NumPCMBlock = = 0 && InferNoSplit ==0) |  |
| **split\_coding\_unit\_flag[** x0 **][** y0 **]** | ae(v) |
| **…** |  |
| } |  |
| return moreDataFlag |  |
| } |  |

## Coding tree semantics

**split\_coding\_unit\_flag**[ x0 ][ y0 ] specifies whether a coding unit is split into coding units with half horizontal and vertical size. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

When not present, split\_coding\_unit\_flag[ x0 ][ y0 ] is derived as follows:

– If variable InferNoSplit is equal to 1, the value of split\_coding\_unit\_flag[ x0 ][ y0 ] is inferred to be equal to 0.

Otherwise if log2CbSize is greater than Log2MinCbSize, the value of split\_coding\_unit\_flag[ x0 ][ y0 ] is inferred to be equal to 1.

– Otherwise (log2CbSize is equal to Log2MinCbSize), the value of split\_coding\_unit\_flag[ x0 ][ y0 ] is inferred to be equal to 0.

Variable InferNoSplit indicates whether the value of split\_coding\_unit\_flag can be inferred or not. Let DepCur, DepBase, DepLeft, DepTop, and DepTopRight represent the depth of current CU, collocated base layer CU, left CU, top CU, and top right CU, respectively. InferNoSplit is derived as follows

InferNoSplit = ( DepCur>=DepBase ) && ( DepCur  > max ( DepLeft, DepTop, DepTopRight ))

when a CU is not available, the related depth is set to log2\_diff\_max\_min\_coding\_block\_size.

## Coding unit syntax

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| CurrCbAddrTS = MinCbAddrZS[ x0 >> Log2MinCbSize ][ y0 >> Log2MinCbSize ] |  |
| **intra\_bl\_skip\_flag**[ x0 ][  y0 ] | ae(v) |
| if (!intra\_bl\_skip\_flag[ x0 ][  y0 ]) { | ae(v) |
| if( slice\_type != I ) |  |
| **skip\_flag[** x0 **][** y0 **]** | ae(v) |
| if( skip\_flag[ x0 ][ y0 ]){ |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| **inter\_resi\_weight\_idx[** x0 **][** y0 **]** | ae(v) |
| } |  |
| else { |  |
| **intra\_bl\_flag** | ae(v) |
| if (intra\_bl\_flag) |  |
| **transform\_selection\_index** | ae(v) |
| else { | ae(v) |
| if( slice\_type != I ) |  |
| **pred\_mode\_flag** | ae(v) |
| if( PredMode == MODE\_INTRA) |  |
| **intra\_resi\_pred\_flag** | ae(v) |
| if( PredMode != MODE\_INTRA | | log2CbSize = = Log2MinCbSize ) |  |
| **part\_mode** | ae(v) |
| x1 = x0 + ( ( 1 << log2CbSize ) >> 1 ) |  |
| y1 = y0 + ( ( 1 << log2CbSize ) >> 1 ) |  |
| x2 = x1 − ( ( 1 << log2CbSize ) >> 2 ) |  |
| y2 = y1 − ( ( 1 << log2CbSize ) >> 2 ) |  |
| x3 = x1 + ( ( 1 << log2CbSize ) >> 2 ) |  |
| y3 = y1 + ( ( 1 << log2CbSize ) >> 2 ) |  |
| if( PartMode = = PART\_2Nx2N ) |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| else if( PartMode = = PART\_2NxN ) { |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x0, y1 , log2CbSize ) |  |
| } else if( PartMode = = PART\_Nx2N ) { |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x1, y0 , log2CbSize ) |  |
| } else if( PartMode = = PART\_2NxnU ) { |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x0, y2 , log2CbSize ) |  |
| } else if( PartMode = = PART\_2NxnD ) { |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x0, y3 , log2CbSize ) |  |
| } else if( PartMode = = PART\_nLx2N ) { |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x2, y0 , log2CbSize ) |  |
| } else if( PartMode = = PART\_nRx2N ) { |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x3, y0 , log2CbSize ) |  |
| } else { /\* PART\_NxN \*/ |  |
| prediction\_unit( x0, y0 , log2CbSize ) |  |
| prediction\_unit( x1, y0 , log2CbSize ) |  |
| prediction\_unit( x0, y1 , log2CbSize ) |  |
| prediction\_unit( x1, y1 , log2CbSize ) |  |
| } |  |
| if(PredMode == MODE\_INTER) |  |
| **inter\_resi\_weight\_idx** | ae(v) |
| } |  |
| if( !pcm\_flag ) |  |
| transform\_tree( x0, y0, x0, y0, log2CbSize, log2CbSize, log2CbSize, 0, 0 ) |  |
| } |  |
| } |  |
| } |  |

## Coding unit semantics

The specifications in sub-clause 7.3.6 apply with the following additions.

**intra\_bl\_skip\_flag**[ x0 ][  y0 ] equal to 1 specifies that the current coding unit is coded in intra-BL mode, and no more syntax elements are parsed after intra\_bl\_skip\_flag[ x0 ][ y0 ]. intra\_bl\_skip\_flag [ x0 ][ y0 ] equal to 0 specifies that more syntax elements are to be parsed. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

**intra\_bl\_flag** equal to 1 specifies that the current coding unit is coded in intra-BL mode. intra\_bl\_flag equal to 0 specifies that coding unit is not coded in intra-BL mode.

When intra\_bl\_flag is not present, it is inferred to be equal to 0.

**intra\_resi\_pred\_flag** equal to 1 specifies that the current coding unit is coded with intra residual prediction mode. intra\_resi\_pred\_flag equal to 0 specifies that the current coding unit is not coded with intra residual prediction mode.

When intra\_resi\_pred\_flag is not present, it is inferred to be equal to 0.

**inter\_resi\_weight\_idx** specifies weighting factor of inter residual prediction for the current coding unit. The value of inter\_resi\_weight\_idx shall be in the range of 0 to 2, inclusive. When not present, inter\_resi\_weight\_idx is inferred to be equal to 0. The weighting factor is derived as inter\_resi\_weight\_idx\*0.5.

When inter\_resi\_weight\_idx is not present, it is inferred to be equal to 0.

**transform\_selection\_index** specifies the transform matrix used for the luma component when intra-BL mode is used. The value of transform\_selection\_index shall be in the range of 0 to 2, inclusive.

# Compression performance discussion

## Encoding setting

### QP settings

#### Enhancement layer QP hierarchy

For the EL in random access configuration, the picture QP is derived according to the picture-level Lagrange multiplier, which is similar to the method in HM8.0 [5,6].

Assuming that *λ* is the Lagrange multiplier for an EL picture, the picture *QP* is derived as

where is the QP used in the CfP [1] for anchor EL bitstream generation.

#### QP change for rate matching

To match the target bitrate, the EL QP changes once during one encoding test based on the ‘floating initial QP’ setting of HM61[3].

A ‘floating initial QP’ includes two parts: integer part and fractional part. The integer part is actually the input QP of the encoding. The fractional part indicates the picture after which (in terms of displaying order) the picture QP is increased by one.

For example, a floating initial QP equal to 32.25 for a 600-frame sequence indicates that the input QP for coding this sequence is 32 and that picture QPs after picture number will be increased by one.

## Category 1 (HEVC base layer)

The proposed methods are tested according to the HEVC SVC CfP [1]. As requested in the CfP, four different types of BD-rate results compared to simulcast are provided in subsections 4.2.1 to 4.2.4. The floating QP mentioned in subsection 4.1.1.1is used to meet the target rate defined in the CfP. There is no rate point above the target rate.

### BD-rate over simulcast based on EL+BL actual rate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HEVC 2x** | | | **All Intra HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -28.0% | -24.9% | -25.7% |  |  |  |
| Class B | -23.2% | -20.7% | -20.8% | -33.5% | -30.9% | -31.2% |
| **Overall** | -25.6% | -22.8% | -23.3% | -33.5% | -30.9% | -31.2% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC 2x** | | | **Random Access HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -24.8% | -16.1% | -17.5% |  |  |  |
| Class B | -21.3% | -15.4% | -15.3% | -31.1% | -23.9% | -24.1% |
| **Overall** | -23.0% | -15.7% | -16.4% | -31.1% | -23.9% | -24.1% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC SNR** | | |  |  |  |
|  | Y | U | V |  |  |  |
| Class A+ | -30.2% | -20.8% | -21.6% |  |  |  |
| Class B | -26.3% | -16.4% | -14.9% |  |  |  |
| **Overall** | -28.2% | -18.6% | -18.3% |  |  |  |

On average, 29.6% (All Intra) and 27.5% (Random Access) luma BD-rate reduction (based on EL+ BL actual rate) is obtained over simulcast. Clearly, the proposed methods are much more efficient than simulcast.

### BD-rate over simulcast based on EL+BL target rate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HEVC 2x** | | | **All Intra HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -27.7% | -24.6% | -25.4% |  |  |  |
| Class B | -22.9% | -20.3% | -20.4% | -33.4% | -30.7% | -31.1% |
| **Overall** | -25.3% | -22.5% | -22.9% | -33.4% | -30.7% | -31.1% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC 2x** | | | **Random Access HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -24.6% | -15.8% | -17.2% |  |  |  |
| Class B | -21.0% | -15.1% | -15.1% | -31.0% | -23.7% | -23.9% |
| **Overall** | -22.8% | -15.5% | -16.1% | -31.0% | -23.7% | -23.9% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC SNR** | | |  |  |  |
|  | Y | U | V |  |  |  |
| Class A+ | -30.0% | -20.6% | -21.4% |  |  |  |
| Class B | -26.0% | -16.0% | -14.6% |  |  |  |
| **Overall** | -28.0% | -18.3% | -18.0% |  |  |  |

On average, 29.3% (All Intra) and 27.3% (Random Access) luma BD-rate reduction is obtained over simulcast when the EL+BL target rate instead of the EL+BL actual rate is used in the BD-rate calculation. Due to rate matching, the actual rate is close to the target rate so that the BD-rate reduction in this subsection is similar to that in the previous subsection.

### BD-rate savings over simulcast based on EL-only actual rate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HEVC 2x** | | | **All Intra HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -40.1% | -36.9% | -37.8% |  |  |  |
| Class B | -33.3% | -30.5% | -30.5% | -55.5% | -52.6% | -52.9% |
| **Overall** | -36.7% | -33.7% | -34.1% | -55.5% | -52.6% | -52.9% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC 2x** | | | **Random Access HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -35.6% | -25.5% | -27.2% |  |  |  |
| Class B | -30.4% | -23.7% | -23.6% | -50.2% | -42.5% | -42.7% |
| **Overall** | -33.0% | -24.6% | -25.4% | -50.2% | -42.5% | -42.7% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC SNR** | | |  |  |  |
|  | Y | U | V |  |  |  |
| Class A+ | -44.0% | -33.3% | -34.2% |  |  |  |
| Class B | -38.8% | -27.2% | -25.5% |  |  |  |
| **Overall** | -41.4% | -30.3% | -29.8% |  |  |  |

On average, 46.1% (All Intra) and 41.5% (Random Access) luma BD-rate reduction is obtained over simulcast when the EL-only actual rate is used in the BD-rate calculation. Compared to the results based on the EL+BL actual rate, the gain by this measure is higher.

### BD-rate savings over simulcast based on EL-only target rate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HEVC 2x** | | | **All Intra HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -39.6% | -36.4% | -37.3% |  |  |  |
| Class B | -32.8% | -30.0% | -30.0% | -55.3% | -52.4% | -52.7% |
| **Overall** | -36.2% | -33.2% | -33.7% | -55.3% | -52.4% | -52.7% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC 2x** | | | **Random Access HEVC 1.5x** | | |
|  | Y | U | V | Y | U | V |
| Class A+ | -35.2% | -25.1% | -26.8% |  |  |  |
| Class B | -30.1% | -23.4% | -23.2% | -50.0% | -42.3% | -42.4% |
| **Overall** | -32.6% | -24.2% | -25.0% | -50.0% | -42.3% | -42.4% |
|  |  |  |  |  |  |  |
|  | **Random Access HEVC SNR** | | |  |  |  |
|  | Y | U | V |  |  |  |
| Class A+ | -43.7% | -33.0% | -33.8% |  |  |  |
| Class B | -38.4% | -26.8% | -25.0% |  |  |  |
| **Overall** | -41.1% | -29.9% | -29.4% |  |  |  |

On average, 45.8% (All Intra) and 41.2% (Random Access) luma BD-rate reduction is obtained over simulcast when the EL target rate is used in the BD-rate calculation. Similarly, the BD-rate reduction is close to that based on the EL actual rate due to the rate matching.

# Complexity analysis

## Encoding time and measurement methodology

Encoding time TSVC of the proposed SVC encoder is compared to the sum of anchor low resolution encoding time (TAL) and high resolution encoding time (TAH). The geometric mean of encoding time ratio is summarized in the following table.

|  |  |
| --- | --- |
| Test conditions | Geometric mean of ) |
| AI 2x | 2.64 |
| AI 1.5x | 2.27 |
| RA 2x | 1.32 |
| RA 1.5x | 1.21 |
| RA SNR | 1.25 |

## Decoding time and measurement methodology and comparison vs. anchor bitstreams decoded by HM

Decoding time TSVC of the proposed SVC decoder is compared to the sum of anchor low resolution decoding time (TAL) and high resolution decoding time (TAH). The geometric mean of encoding time ratio is summarized in the following table. As requested in CfP, YUV output was enabled and reference input was disabled during decoding test. Note the YUV output of base layer is disabled for both the proposed SVC decoder and anchor decoder.

|  |  |
| --- | --- |
| Test conditions | Geometric mean of ) |
| AI 2x | 1.50 |
| AI 1.5x | 1.42 |
| RA 2x | 1.65 |
| RA 1.5x | 1.68 |
| RA SNR | 1.31 |

## Description of computing platform used to determine encoding and decoding times reported in sections 5.1 and 5.2

The encoding test was conducted on 64-bit Linux cluster with Intel Xeon E5 2680 16-core CPUs (2.7GHz). Each cluster host has 64 GB memory. All hosts shared the same network drive. The encoder binary was compiled with gcc 4.1.2 64-bit. Both the proposed encoder and anchor encoder are tested in the same type of platform. Therefore, the reported encoding time is reliable.

The decoding time was measured on a single Windows PC with Intel Xeon W3530 4 cores CPU (2.8GHz), 6GB memory and SATA hard disk with 7200 RPM. Both the proposed decoder and anchor decoder are compiled with 64-bit Intel(R) C++ Compiler XE 12.1.5.344.

## Expected memory usage of encoder

For each layer, the encoder stores temporal reference pictures in the decoded picture buffer (DPB) in a way similar to HEVC single layer encoding. In addition, up-sampled/filtered BL pictures are also stored in DPB.

## Expected memory usage of decoder

For each layer, the decoder stores temporal reference pictures in the decoded picture buffer (DPB) in a way similar to HEVC single layer decoding. In addition, the up-sampled/filtered BL pictures are also stored in the DPB.

## Complexity characteristics of encoder motion estimation and partitioning selection in enhancement layer(s)

The proposed EL encoder employs the same motion estimation and partitioning selection as HM6.1.

## Complexity characteristics of decoder motion compensation in enhancement layer(s)

Compared to HEVC single layer decoding, additional motion compensations are introduced by inter residual prediction and multi-hypothesis motion prediction. If the difference picture between the reconstructed EL pictures and the BL pictures is stored in the DPB buffer, there would be no additional motion compensation process for inter residual prediction mode

## Complexity characteristics of encoder intra-frame prediction type and partitioning selection in enhancement layer(s)

The partitioning selection procedure of the proposed EL encoder is the same as in HM6.1. The intra prediction mode selection method of HM6.1 is applied in the same way for both normal intra prediction and residue intra prediction. An additional RD check is performed for intra-BL and intra-BL skip modes.

## Complexity characteristics of decoder intra-frame prediction operation in enhancement layer(s)

The intra frame prediction is similar to the HEVC and is applied to both normal intra prediction and residue intra prediction for the EL. For the intra residual prediction, BL picture samples need to be added to the intra prediction results of the difference in order to form the final prediction.

## Complexity characteristics of up-sampling filters and transforms specific in enhancement layer(s)

Up to three 8-tap up-sampling filters are employed (for intra-BL, intra residual prediction and inter residual prediction modes). The position mapping process of the up-sampling filters is the same as the method used in H.264/AVC SVC. The complexity of each separable 2D 8-tap up-sampling filter process is similar to that of the HEVC motion compensation interpolation filter.

Additional DCT and DST transforms have the same complexity as those in the HEVC.

## Complexity characteristics of quantization and inverse quantization in enhancement layer(s)

The complexity of quantization and inverse quantization in the EL is identical to those in the HEVC.

## Complexity characteristics of encoder entropy coding operation in enhancement layer(s)

The complexity of encoder entropy coding operation in the EL is similar to that of the HEVC. The additional operation is in the scan order derivation for 4×4 and 8×8 blocks, which needs to calculate the horizontal and vertical gradients of BL block.

## Complexity characteristics of decoder entropy decoding operation in enhancement layer(s)

The complexity of decoder entropy coding operation in the EL is similar to that of the HEVC. The additional operation is in the scan order derivation for 4×4 and 8×8 blocks, which needs to calculate the horizontal and vertical gradients of BL samples.

## Degree of capability for encoder parallel processing

Degree of capability for encoder parallel processing in enhancement layer(s) is similar to HEVC.

## Degree of capability for decoder parallel processing

Degree of capability for decoder parallel processing in enhancement layer(s) is similar to HEVC .

# Software implementation description

The proposed codec was developed based on HEVC test model HM6.1 and implemented with standard C++ programming language.

The software will be made available to interested parties to enable studying of developed algorithms in more detail.

# Highlighted aspects discussion

# Closing remarks

# References

[1] ISO/IEC JTC1/SC29/WG11 and ITU-T SG 16, “Joint Call for Proposals on Scalable Video Coding Extensions of High Efficiency Video Coding (HEVC)”, ISO/IEC JTC 1/SC 29/WG 11 (MPEG) Doc. N12957, Stockholm, Sweden, July 2012.

[2] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, “High Efficiency Video Coding (HEVC) text specification draft 6”, JCTVC-H1003, San Jose, USA, 1-10 Feb, 2012.

[3] HM 6.1, reference software of HEVC, https://hevc.hhi.fraunhofer.de/svn/svn\_HEVCSoftware/tags/HM-6.1.

[4] J. Dong, Y. He, Y. Ye, “J Downsampling filters for anchor generation for scalable extensions of HEVC”, ISO/IEC JTC 1/SC 29/WG 11 (MPEG) Doc. M24499, Geneva, Switzerland, Apr. 2012.

[5] B. Li, L. Li, J. Zhang, J.-Z. Xu, H. Li, “Encoding with fixed Lagrange multipliers”, JCTVC-J0242, Stockholm, Sweden, 11–20 July 2012.

[6] HM 8.0, reference software of HEVC, <https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-8.0>.

# Patent rights declaration

**Qualcomm Incorporated 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).**