|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  34th Meeting: Marrakech, MA, 12–18 Jan. 2019 | Document: JCTVC-AH0027 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **HEVC: SEI manifest SEI message implementation in reference software** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Discussion and Agreement | | |
| *Author(s) or Contact(s):* | **Thomas Stockhammer**  **Waqar Zia**  **Ece Öztürk** | Tel: Email: | +49 1725702667  [tsto@qti.qualcomm.com](mailto:tsto@qti.qualcomm.com)  zia@nomor.de |
| *Source:* | Qualcomm Incorporated,  Nomor Research |  |  |

# Introduction

In this document, information about the contribution made – in accordance with the HEVC specification [1] – to the reference HEVC codec software [2], specifically “SEI Manifest SEI Messages” and “SEI Prefix Indication Messages” is provided.

This document:

* gives information on the implemented SEI messages
* provides the details of the implementation
* explains the future outlook regarding this work

This work is done as a contribution to the reference HEVC codec software, which is hosted at [2]. The work comprises of the abovementioned SEI messages being implemented in this software.

# Background

SEI messages provides information to decoding, display processes, although they are optional and hence, they do not need to be processed by the decoder. Defined by the payload type, they are located in access units. This document focuses on two SEI messages, namely:

* SEI Manifest SEI messages
* SEI Prefix Indication SEI messages

SEI Manifest SEI messages carries data about the SEI messages that are expected to be present or not present and that are necessary or unnecessary for the decoding or display processes if expected to be present.

SEI Prefix Indication SEI messages conveys prefix indication information in which it states that SEI messages of a particular payload type are expected to be present starting with the provided data.

# Implementation

## Environment

The information about the implementation environment is provided in Table 1 below.

Table 1: Implementation environment information.

|  |  |
| --- | --- |
| **Category** | **Information** |
| OS | Ubuntu 16.04 |
| Compiler | g++ 5.4.0 |
| Content | ShakeNDry:   * 1080p, 8 bit-depth, YUV format, Raw container * Taken from <http://ultravideo.cs.tut.fi/#testsequences> |

## Current Status

The syntax and semantic for the SEI messages are provided in Section D2.45, D2.46 and D3.45, D3.46 in the HEVC specification, respectively [1]. Accordingly, both messages are integrated into the software, following the examples of other existing SEI message implementations. The difference information of the software files is provided in the attached zip folder.

Currently, encoding and decoding of both of the messages are possible. Given the configuration information to the encoder, SEI messages will be added to the bitstream and will afterwards be parsed by the decoder.

## Introduced Changes to the Software

As aforementioned the difference files are provided in the zip file attached with this document. According to these the following header and source files in the Fraunhofer HHI HEVC codec software have been modified:

* Common:
  + Source/Lib/TLibCommon/SEI.h
    - Additional two classes for the two SEI messages
    - Defining the payload type for the two SEI messages
  + Source/Lib/TLibCommon/SEI.cpp
    - Adding the SEI messages to prefix SEI definition
  + cfg/sei/sei\_manifest.cfg
    - Creating the example configuration file for using sei manifest sei message
  + cfg/sei/sei\_prefix\_indication.cfg[[1]](#footnote-1)
    - Creating the example configuration file for using sei prefix indication sei message
* Encoder-related:
  + Source/App/TAppEncoder/TAppEncTop.cpp
    - Adding the parameter setting function calls for the two SEI messages
  + Source/App/TAppEncoder/TAppEncCfg.h
    - Adding the parameters defined for the two SEI messages to application configuration class
  + Source/App/TAppEncoder/TAppEncCfg.cpp
    - Adding the functions and helper options for setting the parameter values provided in the configuration files
  + Source/Lib/TLibEncoder/TEncCfg.h
    - Defining the get and set functions for the sei message class parameters
  + Source/Lib/TLibEncoder/SEIEncoder.h
    - Adding the function definitions that initiate the respective sei message objects with the configuration parameter values
  + Source/Lib/TLibEncoder/SEIEncoder.cpp
    - Adding the functions that initiate the respective sei message objects with the configuration parameter values
  + Source/Lib/TLibEncoder/TEncGOP.cpp
    - Adding the function call for initiating the sei object generation with the configuration parameter values
  + Source/Lib/TLibEncoder/SEIwrite.h
    - Defining the function that writes the SEI message to the bitstream
  + Source/Lib/TLibEncoder/SEIwrite.cpp
    - Adding the function that writes the SEI message to the bitstream and the call to this function
* Decoder-related:
  + Source/Lib/TLibDecoder/SEIread.h
    - Defining the parsing function for the two SEI messages
  + Source/Lib/TLibDecoder/SEIread.cpp
    - Adding the parsing function for the two SEI messages and the call to this function

## How to Use

For using these SEI messages, appropriate configuration files need to be provided to the encoder application. There are example configuration files provided for these messages; so, if desired these files can be used either directly or for reference to create different configuration files. Some example command line arguments are provided below:

* Enable SEI Manifest SEI messages
  + TAppEncoderStatic -i <*Input\_File*> -c <*Config\_file\_location*/sei\_manifest.cfg> -o <*Output\_File*>
* Enable SEI Prefix Indication SEI messages
  + TAppEncoderStatic -i <*Input\_File*> -c <*Config\_file\_location*/sei\_prefix\_indication.cfg> -o <*Output\_File*>
* Enable both SEI Manifest SEI messages and SEI Prefix Indication SEI messages
  + TAppEncoderStatic -i <*Input\_File*> -c <*Config\_file\_location*/sei\_manifest.cfg> -c <*Config\_file\_location*/sei\_prefix\_indication.cfg> -o <*Output\_File*>

# Next Steps

Next steps are thorough check of their locations in the access units and their relation to each other. Some examples Depending on this:

* The related function calls might need to be moved to somewhere else
* Additional assert functions might need to be added

# Proposal

SVN Diff is provided as associated resource, to be integrated in the reference software after review.

# References

[1] – ISO/IEC 23008-2 DIS 4th Edition – *High efficiency coding and media delivery in heterogeneous environments — Part 2: High Efficiency Video Coding*.

[2] – Fraunhofer HHI – *HEVC codec reference software*.

1. The SEISpiSeiPrefixDataBit corresponding to the configuration for sei\_prefix\_data\_bit parameter in the conguration file for SEI prefix indication SEI message should be provided as follows:

   *NUM\_BITS1 BIT11 BIT12 … BIT1NUM\_BITS1 NUM\_BITS2 BIT21 BIT22 … BIT2NUM\_BITS2 NUM\_BITS3 BIT31 BIT32 … BIT3NUM\_BITS3 … NUM\_BITSN BITN1 BITN2 … BITNNUM\_BITSN*

   , where N is equal to the num\_sei\_prefix\_indication\_minus1 plus 1, NUM\_BITSX is equal to X-th num\_bits\_in\_prefix\_indications\_minus1 plus 1, and BITXY corresponds to the Y-th prefix data bit in the X-th prefix indication message.

   This method is chosen since it adds no additional change or complexity to the software. [↑](#footnote-ref-1)