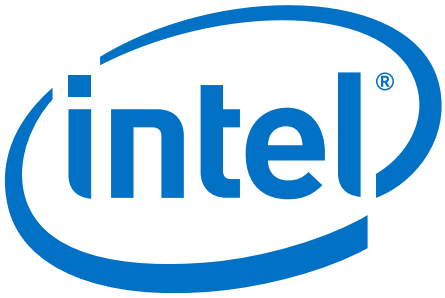

Intel® Stress Bitstreams and Encoder (Intel® SBE) 2016 – SHVC

User Guide

Revision 2.2.0
April 5, 2016

Contents

Revision History

Revision	Date	Description
2.2.0	04/05/2016	Initial revision

Chapter 1
Introduction

This document describes the use of Intel® Stress Bitstreams and Encoder (Intel® SBE) 2016 – SHVC (scalable extension of H.265/HEVC).

1.1 General Overview

Testing of a decoder is a complex multi-criteria task. Code coverage of 100% lines of a decoder code does not guarantee the decoder is fully compliant with video coding standard.

Intel® SBE SHVC Encoder tosses different combinations of syntax elements and coding configurations and writes them into standard-compliant bitstream. Intel® SBE SHVC Encoder neglects efficient motion estimation and mode decision that are not parts of video standard for the sake of flexibility.

Intel® SBE SHVC Encoder comes with the set of bitstreams. Each bitsream of the set is aimed to test some specific technology inherent to SHVC video coding standard. If decoder in test fails to correctly decode randomly generated stream then it makes sense to extend the test pool with the stream for future regression validation. Intel® SBE SHVC Encoder is a great extension of codec validation in addition to Intel® SBE SHVC Encoder test bitstreams.

1.2 Compliance Mode

Random encoder is highly configurable and flexible SHVC syntax encoder tool. In opposite to regular encoders it is not intended to achieve compression but is only designed to create a valid specification-compliant stream. Compliant streams contain only allowed combinations of syntax elements and their values to test decoder for unusual, stress or corner cases where developers usually relax requirements to decoders code for the sake of higher decoding speed. Decoder must be compatible with any of standard conformant bitstreams so its sloppy optimizations have to be carefully tracked for boundary cases where e.g. residual overflow may break visual quality of the picture.

Recommended validation process of the SHVC compliant decoder with Intel® SBE SHVC Encoder is portrayed on the Figure 1. It is up to user to change the flow and to react on pass and fail events and evenset up criteria of test passing.

Intel® SBE SHVC Encoder accepts source emphYUV file as an input and *parfile* describing testing settings such as features to utilize, fixed values, random values. Intel® SBE SHVC Encoder produces encoded bitstream and optionally writes *YUV file* with internal reconstruction data.

This file is used to validate that Intel® SBE SHVC Encoder generated conformant compressed file and that resulted bitstream is valid.

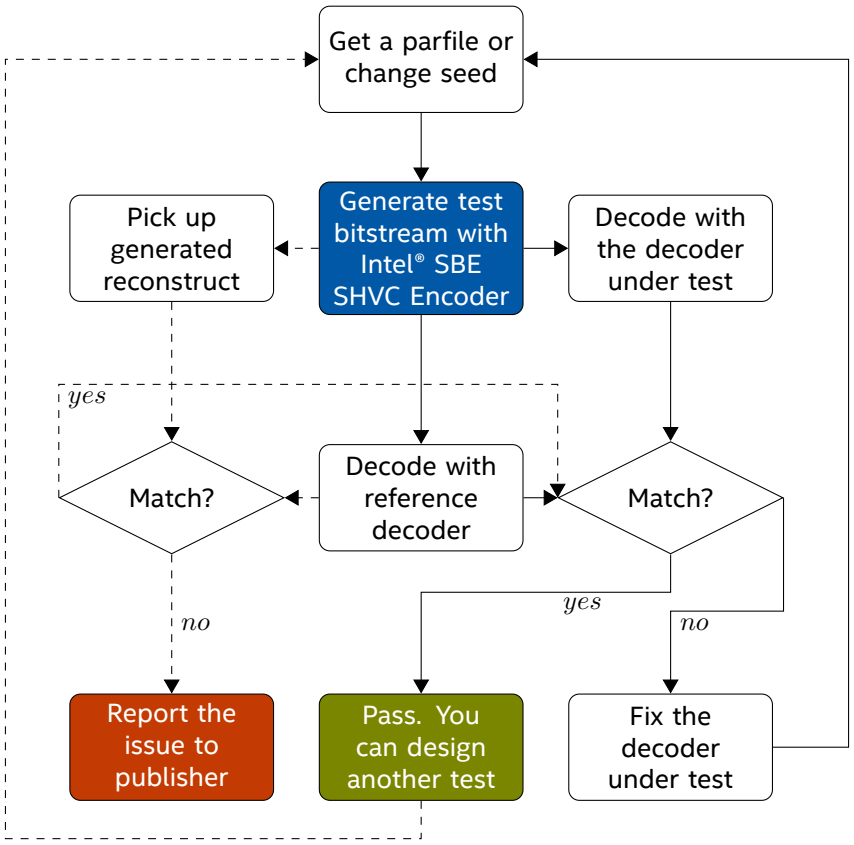

Figure 1.1 – Working with the Intel® SBE SHVC Encoder

If there is a mismatch between encoder reconstruct and reference decoder result, which by the way may mean that bug is inside reference SHM decoder, you are welcome to report to Software Publisher (your Intel contact) with the case configuration to request for the fix if it fits to your license agreement with Intel. We will always appreciate your feedback.

Intel® SBE SHVC Encoder has seed parameter defining initial state of random engine. Changing it allows user to re-use the same parfile to produce different streams with the same scope of randomization defined by parfile. The main purpose of this feature is extensive testing with all possible syntax element combinations. In addition, this feature can be used to create small bug reproducers (by setting frame number parameter `-f` to some small value) for the parfiles which are known to generate the streams causing failure of examined decoder.

To summarize, proposed workflow for compliance testing consists of the following steps:

1. Produce test stream by feeding Intel® SBE SHVC Encoder with a parfile and an optional input YUV file
2. Decode the stream with reference SHM decoder
3. Verify that Intel® SBE SHVC Encoder reconstruct matches reference decoder result
4. Decode the stream with your decoder and verify that its result matches reference decoder result
5. Increment seed parameter and go to the step 1

1.3 SHVC Specific

To start validation cycle with Intel® SBE SHVC Encoder make a decision regarding testing agenda and setup certain features and value range in random-encode parfile. Parfile is a JSON fromatted file, it contains flags and settings user can vary to test some desired parts of the decoder. Parfile can be used to limit features to a set which is currently implemented by the decoder under testing and focus on testing these features only.

Intel® SBE SHVC Encoder accepts an optional input YUV file and parfile and produces one bitstream per call. Therefore the best option for extensive testing is to modify input configuration file outside of random encoder.

For easier validation, Intel® SBE SHVC Encoder writes hashes of reconstructed pictures into special SHVC suffix SEI messages.

1.4 Known Limitations

Current version of Intel® SBE SHVC Encoder has following limitations:

- Encoder has no support of color gamut related features of SHVC
- Though 10-bit bitstream may be generated with Encoder, full support of 10-bit is not provided
- Encoder supports only small number of variants of AU and layers mixing
- Encoder supports up to 8 layers only
- Encoder does correct resampling for bitstreams of 2 layers only
- Interlayer references are not fully supported for 3 or more layers
- Encoder supports only 4 different coding structure configurations (GOP)
- Tiles are not supported
- Scaling lists are not supported
- POC reset feature with `poc_reset_idc` equal to 3 is not fully supported
- Temporal scalability is not supported
- VUI (including HRD) features are not fully supported though relevant syntax may be present
- Output layer sets are not fully supported
- Only HEVC base layer is supported
- Activation of headers is not supported
- DPB size syntax is not supported

Chapter 2

Usage of Various Features

This chapter describes different features of Intel® SBE SHVC Encoder

2.1 Command Line Options

Intel® Stress Random Encoder is a console application, which can be controlled by some command line options. Most options specified in command line may also be specified in parfile. Intel® SBE SHVC Encoder essentially needs parfile and will not work without it.

Command line option	Parfile parameter	Purpose
`-i<N> <INPUT>`	`InputFile`	Input YUV file for Nth layer
`-o<N> <OUTPUT>`	—	Output of reconstructed YUV file of Nth layer
`-f <FRAMES>`	`FramesToBeEncoded`	Number of frames to encode
`-ss <SEED>`	`seed`	Number to initialize pseudorandom engine
`-p <PARFILE>`	—	Parfile with encoding parameters

Table 2.1 – Command line options.

Usage example:

```
shvc_random_encoder.exe -b b.hevc -p test.json -o0 o0.yuv -o1 o1.yuv
```

2.2 Parfile Fields and Values

Parfile is a JSON file which has the following sections: Stream, VPS, SPS, PPS, Slice, CTB and SEI. Each section has a list of values. Each of them may have the following type: string, number, flag, range or probability P. String types are used for file names and tiers. Numbers specify values which do not change during stream generation process. Flags are also unchangeable values which can be “true” or “false” only. Range values are used for values which vary randomly within specified range.

Entries of a parfile control specific syntax elements of SHVC and other parameters of coding structure. The latter are borrowed from SHM reference software configuration files.

When parameters in a given parfile are incompatible with SHVC constraints Intel® SBE SHVC Encoder will print message that it cannot start encoding. Intel® SBE SHVC Encoder may adjust some values on its own discretion if it is still possible to encode bitstream after that. For example, if BL resolution is larger than EL resolution, Intel® SBE SHVC Encoder generates negative offsets for scaled EL window.

Probabilities of randomized parameters are specified as a set of relative weights for every possible value of a parameter. Thus, the entry

$$\texttt{discardable\_flag} : [1, 0]$$

should be read as “false” value will be set in 100% of tosses. If it had $[1, 1]$, Intel® SBE SHVC Encoder would toss “false” or “true” with equal probability.

Intel® SBE SHVC Encoder has no internal bitrate control, so if it is necessary to satisfy tier and level bitrate and compression rate requirements, user is expected to tune QP setting.

Generally random encoder produces large streams because unlike real encoder it aims to syntax generation, not to achieve any compression. But in most cases to test a decoder it is not necessary to produce streams with bitrate limited by tier and level requirements.

2.3 List of Parameters with Default Values

Parameter	Type	Default	Description
`seed`	number	1234556789	Pseudorandom generator initial value
`FramesToBeEncoded`	number	–	Number of frames to generate
`NumLayers`	range	[2, 2]	Number of layers
`Level<N>`	number	120	SHVC Level of layer N
`InputBitDepth<N>`	number	8	Layer N bitdepth
`FrameRate<N>`	number	24	Stream frame rate
`SourceWidth<N>`	number	—	Width of layer N
`SourceHeight<N>`	number	—	Height of layer N
`IntraPeriod<N>`	number	1	Intra period for layer N
`InputFile<N>`	string	—	Source YUV filename for layer
`Profile<N>`	number	1 or 7	Layer N SHVC profile
`DecodingRefreshType`	number	2	1 - CRA, 2 - IDR
`GOPSize`	number	1	Size of group of ordered pictures
`frame_type`	probability	[1, 0, 0]	Probabilities of frame types
`gop_structure`	probability	[1, 0, 0, 0]	Probabilities of one of GOP types in use

Table 2.2 – Stream section

Parameter	Type	Default	Description
`ScalabilityMask<N>`	probability	–	Scalability mask for layer N
`AdaptiveResolutionChange`	number	0	Adaptive resolution coding structure enabled
`NumSamplePredRefLayers<N>`	number	1	Number of ref layers for sample predicition
`SamplePredRefLayerIds<N>`	string	“0”	List of sample prediction refs for layer N
`NumMotionPredRefLayers<N>`	number	1	Number of motion prediction layers for layer N
`MotionPredRefLayerIds<N>`	string	“0”	List of motion prediction refs for layer N
`NumActiveRefLayers<N>`	number	1	Number of active refs for layer N
`PredLayerIds<N>`	string	“0”	List of prediciton layers of layer N
`NumLayerSets`	range	[2, 2]	Number of layer sets
`NumLayerInIdList<N>`	number	2	Number of layer in set
`LayerSetLayerIdList<N>`	string	“0 1”	List of layers in set
`NumAddLayerSets`	number	0	Number of layers in additional layer set
`NumOutputLayerSets`	number	2	Number of OLS
`NumOutputLayersInOutputLayerSet`	string	“1”	List of number of layers in OLS
`ListOfOutputLayers`	string	“1”	List of output layers
`ListOfProfileTierLevelOls`	string	“1 2”	List of PTLs in OLS

Table 2.3 – Layers section

Parameter	Type	Default	Description
discardable_flag	probability	[1, 0]	Tosses discardable flag
cross_layer_bla_flag	probability	[1, 0]	Tosses cross layer BLA flag
poc_reset_idc_3	probability	[1, 0]	Enables POC reset type 3 related syntax

Table 2.4 – Slices section

Parameter	Type	Default	Description
scaled_ref_layer_offset_present_flag	probability	[1, 0]	Enables EL window offsets
ref_region_offset_present_flag	probability	[1, 0]	Enables BL window offsets
resample_phase_set_present_flag	probability	[1, 0]	Enables resampling phases
phase_luma	range	[0, 31]	Set limits for luma phases
phase_chroma	range	[0, 63]	Set limits for chroma phases

Table 2.5 – PPS section

Parameter	Type	Default	Description
inter_view_mv_vert_constraint_flag	probability	[1, 0]	Tosses the specified flag
update_rep_format_flag	probability	[1, 0]	Enables format updated in SPS

Table 2.6 – SPS section

Parameter	Type	Default	Description
vps_max_layers_minus1	range	[2, 2]	Max number of layers
scalability_mask_flag	probability	[1, 1]	Tosses specified flag
vps_sub_layers_max_minus1_present_flag	probability	[1, 1]	Tosses specified flag
MaxTidRefPresentFlag	probability	[1, 1]	Enables max temporal id set
MaxTidIlRefPicsPlus1	range	[2, 2]	Sets max temporal id
default_ref_layers_active_flag	probability	[1, 1]	Tosses specified flag
splitting_flag	probability	[0, 1]	Tosses specified flag
vps_poc_lsb_aligned_flag	probability	[1, 0]	Tosses specified flag
direct_dep_type_len_minus2	range	[3, 3]	Limits dependency type
direct_dependency_all_layers_flag	probability	[1, 0]	Tosses specified flag
vps_non_vui_extension_length	range	[0, 4096]	Limits for specified length
vps_non_vui_extension_data_byte	range	[0, 255]	Limits for specified byte
vps_vui_present_flag	probability	[0, 1]	Tosses specified flag
chroma_and_bit_depth_vps_present_flag	probability	[0, 1]	Tosses specified flag
bit_rate_present_vps_flag	probability	[1, 1]	Tosses specified flag
pic_rate_present_vps_flag	probability	[1, 1]	Tosses specified flag
bit_rate_present_flag	probability	[1, 1]	Tosses specified flag
pic_rate_present_flag	probability	[1, 1]	Tosses specified flag
video_signal_info_idx_present_flag	probability	[0, 1]	Tosses specified flag
vps_num_video_signal_info	range	[1, 3]	Limits specified value
vps_video_signal_info_idx	range	[0, 15]	Limits specified value
tiles_not_in_use_flag	probability	[0, 1]	Tosses specified flag
direct_dependency_type	range	[0, 1]	Limits IL prediction types

Table 2.7 – VPS section

Legal Information

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.