**Source: IVAS-11 Editor (Dolby Laboratories Inc)[[1]](#footnote-1)**

**Title: IVAS Non-BE Conformance (IVAS-11)**

**Version: 0.0.1**

**Agenda item: 7.5 - IVAS\_Codec\_Ph2**

**Document for: Agreement**

**Revision history:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Meeting** | **Subject/Comment** | **Old** | **New** |
| **2025-05-22** | **SA4#132** | **Initial version including S4-250854 and time plan from S4-250xxx.** | **N/A** | **0.0.1** |
|  |  |  |  |  |
|  |  |  |  |  |

Key Acronyms

HOA Higher Order Ambisonics

IVAS Immersive Voice and Audio Services

ISM Independent Streams with Metadata

MASA Metadata-Assisted Spatial Audio

MC Multi Channel

FFT Fast Fourier Transform

EVS Enhanced Voice Services

WMOPS Weighted Million Operations Per Second

MCPS Million Cycles Per Second

WID Work Item description

BE Bit-exact

Non-BE Non Bit-exact

# Introduction

During SA #107, IVAS phase 2 WID objectives were updated [1] to develop the conformance process (including tools and test vectors) of IVAS for non-BE implementations as highlighted below.

* *Enhancements to codec conformance test procedures and criteria. Under this objective, a conformance process (including tools and test vectors) of IVAS for non-bit-exact floating-point implementations shall be developed, aligned with the conclusions of TR 26.843. More specifically, this work includes the following steps:*
	+ *Specification of tools and conformance test vectors suitable for performing IVAS conformance testing.*
	+ *Definition of relevant testing processes and criteria, based on the latest IVAS floating-point reference code.*
		- *The conformance testing processes and criteria shall be tight enough to ensure equivalent quality and interoperability with*
			* *implementation of the floating-point reference code that meets the bit-exact conformance requirements for this code specified in TS 26.252 and*
			* *implementation of the fixed-point reference code that meets the bit-exact conformance requirements for this code specified in TS 26.252.*
		- *At the same time, the coverage of the conformance testing processes and criteria shall be sufficient to avoid interoperability issues between implementations found conformant based on the non-bit-exact criteria and conformant implementations based on the bit-exact criteria.*
	+ *Investigation on the applicability of the testing processes and criteria, which includes:*
		- *Robustness testing with validation that inadequate optimizations are properly detected while adequate optimizations still pass the criteria.*
	+ *Definition of a mandatory non-bit-exact IVAS conformance testing process to be included in TS 26.252 using the tools, conformance criteria and conformance test vectors developed under this work item objective.*

To come up with conformance criteria, similar process can be followed as EVS non-BE conformance [3]. However, given that IVAS supports immersive input/output formats, certain changes may be required to the conformance methods and criteria to ensure equivalent quality as compared to the reference outputs. In this permanent document, the objective analysis data is gathered by running IVAS source code on various platforms and compilers using multiple compiler optimizations.

# Objective Methods

## Description

IVAS codec internally uses core coding switches similar to EVS and may encode an audio frame using LPC based coding or MDCT transform based coding depending on the outcome of various classifiers in the source code. Moreover, stereo format employs switching mechanism to switch between various stereo downmix techniques at low bitrates. Due to these dynamic switching mechanisms, the output of IVAS encoder+decoder may vary significantly even if there is a minor precision change at the encoder. For this reason, in the conformance criteria, it may be desired to use objective tools that rely on perceptual quality analysis. The impact of floating point precision change across different platforms and compilers may be minimal in case of IVAS external renderer and certain modes of IVAS decoder and hence signal based objective methods may be desirable to define conformance criteria for such cases.

Various signal based and perceptual based methods that can potentially be used for IVAS non-BE conformance are described below. All the methods/tools described below are used for mono audio comparison. For output formats with more than one channel (i.e. all output configurations except MONO), the tools perform a pairwise comparison of each channel from the test and reference audio samples.

## Signal based methods

### Overview

### Segmental Signal-to-Noise Ratio (SSNR)

The Segmental Signal-to-Noise Ratio is evaluated as defined in as defined in ISO/IEC 14496-4. The segment length is 20 ms and the minimum SSNR of all the segments across all channels is reported. This objective measure has been extensively used in IVAS fixed point development process and can be considered for IVAS non-BE conformance as well. The results with SSNR method with IVAS floating point code are present in clause 3.3.2.

### RMS error threshold

This tool is described in [3] and can be considered for IVAS non-BE conformance as well. This tool is yet to be tested with IVAS tests set described in clause 3.

## Perceptually based methods

### Overview

### Maximum Loudness Difference (MLD)

MLD is described in [3]. For MLD, the maximum number among all the channels is reported. This objective measure has been extensively used in IVAS fixed point development process and can be considered for IVAS non-BE conformance as well. The results with MLD method with IVAS floating point code are present in clause 3.3.1.

### MOS-LQO Validation

This tool is described in [3] and can be considered for IVAS non-BE conformance as well. This tool is yet to be tested with IVAS tests set described in clause 3.

### Perceptual Evaluation of Audio Quality (PEAQ)

PEAQ is fully complaint to ITU-R BS.1387 and can be considered for IVAS non-BE conformance as well. This tool is yet to be tested with IVAS tests set described in clause 3.

IVAS floating point Non-BE conformance

## Test vector set and test running method

### Test vector set

* IVAS encoder and decoder tests: Short test vector (STV) set from [6].
* IVAS external renderer tests: Test set from [6].
* IVAS Split renderer tests: Test set from [6].
* The test set used for all three test types mentioned above has the following characteristics:
	+ Input audio and accompanying metadata spanning all formats.
	+ Speech and mixed speech/music content
	+ Length: 1-30 seconds, average 10.5 seconds

### Test running method and command lines

* Pytest framework is used to run the STV set.
* For IVAS encoder + decoder tests, the following steps were followed:
1. First references were generated using reference platform and compiler settings as mentioned in clause 4. The following pytest command was used to generate reference outputs

*python -m pytest tests/codec\_be\_on\_mr\_nonselection -v -n auto --update\_ref 1 --create\_ref --keep\_files*

1. The reference outputs were stored on test Operating systems/platform using the same directory/file structure as on the reference system.
2. Then IVAS floating source code is compiled on test platform and compilation setting.
3. Then test outputs were generated using following pytest command

*python -m pytest tests/codec\_be\_on\_mr\_nonselection -v -n auto --keep\_files --create\_cut*

* For IVAS external renderer tests, the following steps were followed:
1. First references were generated using reference platform and compiler settings as mentioned in clause 4. The following pytest command was used to generate reference outputs

*python -m pytest tests/renderer/test\_renderer.py -v -n auto --update\_ref 1 --create\_ref --keep\_files*

1. The reference outputs were stored on test Operating systems/platform using the same directory/file structure as on the reference system.
2. Then IVAS floating source code is compiled on test platform and compilation setting.
3. Then test outputs were generated using following pytest command

*python -m pytest tests/renderer/test\_renderer.py -v -n auto --keep\_files --create\_cut*

* For IVAS split renderer tests, the following steps were followed:
1. First references were generated using reference platform and compiler settings as mentioned in clause 4. The following pytest command was used to generate reference outputs

*python -m pytest tests/split\_rendering/test\_split\_rendering.py -v -n auto --update\_ref 1 --create\_ref --keep\_files*

1. The reference outputs were stored on test Operating systems/platform using the same directory/file structure as on the reference system.
2. Then IVAS floating source code is compiled on test platform and compilation setting.
3. Then test outputs were generated using following pytest command

*python -m pytest tests/split\_rendering/test\_split\_rendering.py-v -n auto --keep\_files --create\_cut*

1. Reference encoded bitstreams were used for split rendering tests.

## Reference platform and compiler settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Processor | Platform | Compiler | Compiler settings and optimization flags |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Windows 11, SDK: 10.0.17763.0 | Microsoft Visual StudioToolset: Visual Studio 2017 (v141)ToolsVersion="15.0" | Debug, Win32, Od (disabled), no optimizations |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | Linux, RHEL 8.4 | GCC 9.4.0 | O0, makefile [8]Default setting with make -j |

**Table 3-2.1 Reference platform and compiler settings**

## Results

### MLD

#### IVAS encoder + decoder tests

Total number of tests: 627

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref used** | **Processor** | **Platform** | **Compiler** | **Compiler settings and optimization flags** | **MLD** | **% tests < 5 MLD** |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Windows 11, SDK: 10.0.17763.0 | Microsoft Visual StudioToolset: Visual Studio 2017 (v141)ToolsVersion="15.0" | Release, Win32, O2 (Favor speed) | 1.5984988 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | O0, Makefile [8]Default setting with make -j | 5.37 | 99.84 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | Makefile [8]Modifications: O3 optimization | 3.7 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O2 optimization | 3.7 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 optimization | 3.7 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: GCC and Ofast optimization | 16.534607 | 86.9 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 and -ffast\_math optimization | 20.8804 | 85.9 |

**Table 3-3.1 Results with IVAS encoder and decoder using MLD**

#### IVAS external renderer tests

Total number of tests: 658

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref used** | **Processor** | **Platform** | **Compiler** | **Compiler settings and optimization flags** | **MLD** | **% tests < 5 MLD** |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Windows 11, SDK: 10.0.17763.0 | Microsoft Visual StudioToolset: Visual Studio 2017 (v141)ToolsVersion="15.0" | Release, Win32, O2 (Favor speed) | 0.002 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | O0, Makefile [8]Default setting with make -j | 0.59 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | Makefile [8]Modifications: O3 optimization | 0.59 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O2 optimization | 0.59 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 optimization | 0.59 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: GCC and Ofast optimization | 0.59 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 and -ffast\_math optimization | 0.59 | 100 |

**Table 3-3.2 Results with IVAS external renderer using MLD**

#### IVAS split renderer tests

Total number of tests: 658

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref used** | **Processor** | **Platform** | **Compiler** | **Compiler settings and optimization flags** | **MLD** | **% tests < 5 MLD** |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | O0, Makefile [8]Default setting with make -j | 0.9 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | Makefile [8]Modifications: O3 optimization | 0.9 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O2 optimization | 0.9 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 optimization | 0.9 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: GCC and Ofast optimization | 11.858449 | 97.4 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 and -ffast\_math optimization | 12.11 | 97.5 |

**Table 3-3.3 Results with IVAS split renderer using MLD**

### SSNR

#### IVAS encoder + decoder tests

Total number of tests: 627

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref used** | **Processor** | **Platform** | **Compiler** | **Compiler settings and optimization flags** | **SSNR (dB)** | **% tests > 50 dB SSNR** |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Windows 11, SDK: 10.0.17763.0 | Microsoft Visual StudioToolset: Visual Studio 2017 (v141)ToolsVersion="15.0" | Release, Win32, O2 (Favor speed) | 58.83 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | O0, Makefile [8]Default setting with make -j | 46.81 | 99.84 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | Makefile [8]Modifications: O3 optimization | 33.35 | 99.84 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O2 optimization | 33.35 | 99.84 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 optimization | 33.35 | 99.84 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: GCC and Ofast optimization | 2.66 | 24 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 and -ffast\_math optimization | 10.13 | 26 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | llvm-v18.1.6d1rh84-O0 | Makefile [8]Modifications: Clang and O0 | inf | 100 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | gcc v9.4.0 -O3 | Makefile [8]Modifications: GCC and O3 | 44.7 | 99.82 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | llvm-v18.1.6d1rh84-O3 | Makefile [8]Modifications: Clang and O3 | 44.7 | 99.82 |
| 2 | 5-stage HiFi5s DSP | Cadence xt-clang | xt-clang RJ-2024.3 tools | Makefile [8]Modifications: xt-clang and O0 | WIP | WIP |

**Table 3-3.4 Results with IVAS encoder and decoder using SSNR**

#### IVAS external renderer tests

Total number of tests: 658

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref used** | **Processor** | **Platform** | **Compiler** | **Compiler settings and optimization flags** | **SSNR** | **% tests > 50 dB SSNR** |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Windows 11, SDK: 10.0.17763.0 | Microsoft Visual StudioToolset: Visual Studio 2017 (v141)ToolsVersion="15.0" | Release, Win32, O2 (Favor speed) | 96.5 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | O0, Makefile [8]Default setting with make -j | 81.58 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | Makefile [8]Modifications: O3 optimization | 85.62 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O2 optimization | 85.62 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 optimization | 85.62 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: GCC and Ofast optimization | 67.08 | 100 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 and -ffast\_math optimization | 65.43 | 100 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | llvm-v18.1.6d1rh84-O0 | Makefile [8]Modifications: Clang and O0 | inf | 100 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | gcc v9.4.0 -O3 | Makefile [8]Modifications: GCC and O3 | 81.98 | 100 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | llvm-v18.1.6d1rh84-O3 | Makefile [8]Modifications: Clang and O3 | 81.98 | 100 |
| 2 | 5-stage HiFi5s DSP | Cadence xt-clang | xt-clang RJ-2024.3 tools | Makefile [8]Modifications: xt-clang and O0 | 83.3 | 100 |

**Table 3-3.5 Results with IVAS external renderer using SSNR**

#### IVAS split renderer tests

Total number of tests: 658

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref used** | **Processor** | **Platform** | **Compiler** | **Compiler settings and optimization flags** | **SSNR** | **% tests > 50 SSNR** |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | O0, Makefile [8]Default setting with make -j | 45.82 | 97.1 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | GCC 11.3 | Makefile [8]Modifications: O3 optimization | 45.82 | 97.1 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O2 optimization | 45.82 | 97.1 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 optimization | 45.82 | 97.1 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: GCC and Ofast optimization | 28.62 | 67.9 |
| 1 | Intel(R) Xeon(R) W-1290 CPU | Linux, Ubuntu 22.04.1, x86\_64 | Clang 14.0.0 | Makefile [8]Modifications: Clang and O3 and -ffast\_math optimization | 28.27 | 65.2 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | llvm-v18.1.6d1rh84-O0 | Makefile [8]Modifications: Clang and O0 | Inf | 100 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | gcc v9.4.0 -O3 | Makefile [8]Modifications: GCC and O3 | 38.27 | 97.26 |
| 2 | Intel(R) Xeon(R) Gold 6152 CPU | RHEL 8.4 | llvm-v18.1.6d1rh84-O3 | Makefile [8]Modifications: Clang and O3 | 38.27 | 97.26 |
| 2 | 5-stage HiFi5s DSP | Cadence xt-clang | xt-clang RJ-2024.3 tools | Makefile [8]Modifications: xt-clang and O0 | WIP | WIP |

**Table 3-3.6 Results with IVAS split renderer using SSNR**

## Observations

IVAS floating point code was run with STV test set on various platforms with different compiler settings. IVAS reference outputs were generated using reference platform and compiler settings mentioned in Table 4.1. With the objective methods used so far, i.e., MLD and SSNR, the following observations can be made:

* In the IVAS encoder+decoder tests, low MLD (MLD < 5 in all cases except 1 test) values are observed with O0, O2 and O3 compiler optimization settings on various platforms with both GCC and clang compiler. MLD >10 values are observed with compiler optimization flags -Ofast and -ffast-math.

With O0, O2 and O3 compiler optimization settings on various platforms with both GCC and clang compiler, about 99.84 % tests report the worst case SSNR > 50 dB (only 1 test case out of 627 tests report SSNR less than 50 dB). With -Ofast and -ffast-math optimization SSNR performance can be considered poor.

* In the IVAS external renderer tests, very low MLD (< 1) and high SSNR (> 65 dB) can be observed across platforms and compilers.
* In the IVAS split renderer tests, very low MLD (< 1) is observed across platforms and compilers. MLD >10 values are observed with -ffast-math and -Ofast optimization flags. Without -Ofast and fast-math optimization, 97% of tests report the worst case SSNR of greater than 50 dB.
* More experiments are needed to draw conclusions on non-BE conformance criteria
* Cases with -Ofast and -ffast-math should be debugged further wherein perceptual differences can be heard in informal listening in the test runs that correspond to high MLD values.

# IVAS Fixed point Non-BE conformance

## Test vector set and test running method

## Results

## Observations

# Time plan

The time plan of the IVAS BASOP verification is covered in [9].

References

1. SP-250262: “Revised WID on EVS Codec Extension for Immersive Voice and Audio Services, Phase 2”
2. TS 26.444: “Test sequences (EVS)”
3. TR 26.843: “Study on non-bit-exact conformance criteria and tools for floating-point EVS codec”
4. TS 26.443: “Codec for Enhanced Voice Services (EVS); ANSI C code (floating-point)”
5. TS 26.258: “Codec for Immersive Voice and Audio Services (IVAS); C code (floating-point)”
6. IVAS floating point source code: “[IVAS Codec Public Collaboration / IVAS Codec · GitLab](https://forge.3gpp.org/rep/ivas-codec-pc/ivas-codec)”
7. IVAS-10 IVAS BASOP Verification (IVAS-10), S4-242071
8. IVAS floating point Makefile: “[Makefile · main · IVAS Codec Public Collaboration / IVAS Codec · GitLab](https://forge.3gpp.org/rep/ivas-codec-pc/ivas-codec/-/blob/main/Makefile?ref_type=heads)”
9. IVAS Permanent document IVAS-2b: IVAS\_Codec\_Ph2 Project Plan
1. Rishabh Tyagi, Dolby Laboratoires Inc. ; email : rishabh.tyagi@dolby.com [↑](#footnote-ref-1)