3GPP TS 26.273 V1.0.0 (2004-06)

Technical Specification

**3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
ANSI-C code for the Fixed-point
Extended AMR Wideband codec
(Release 6)**

Keywords

AMR-WB, AMR-WB+, audio CODEC, Extended
AMR Wideband codec

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

3GPP

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x   the first digit:

1   presented to TSG for information;

2   presented to TSG for approval;

3   or greater indicates TSG approved document under change control.

y   the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z   the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document contains an electronic copy of the ANSI-C code for the Fixed-point Extended Adaptive Multi-Rate Wideband codec. Alternatively, floating-point ANSI-C code is specified in 3GPP TS 26.304 [1]. The floating-point codec/encoder/decoder specified in this document or the floating-point codec/encoder/decoder specified in [1] may be used depending on if the implementation platform is better suited for a floating-point or a fixed-point implementation. It has been verified that the fixed-point and floating-point codecs interoperate with each other without any artifacts.

The fixed-point ANSI-C code in the present document defines, besides the floating-point c-code specified in [1], one valid reference implementation of the Extended Adaptive Multi-Rate Wideband transcoder (3GPP TS 26.290 [2]). Standard conformance it is enforced by meeting the conformance criteria defined in [3].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1]   3GPP TS 26.304: "ANSI-C code for the Floating-point Extended AMR Wideband codec".

[2]   3GPP TS 26.290: " Audio codec processing functions; Extended AMR Wideband codec; Transcoding functions ".

[3]   3GPP TS 26.xxx: "3GPP audio codecs, Conformance".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions are given in TS 26.290 [1].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AMR-WB+	Extended Adaptive Multi-Rate WideBand
ANSI	American National Standards Institute
ETS	European Telecommunication Standard
GSM	Global System for Mobile communications
I/O	Input/Output
RAM	Random Access Memory
ROM	Read Only Memory

4 C code structure

This clause gives an overview of the structure of the bit-exact C code and provides an overview of the contents and organization of the C code attached to this document.

The C code has been verified on the following systems:

- IBM PC/AT compatible computers with Windows 2000 SP4 and Microsoft Visual C++ v.6.0 compiler.

ANSI-C was selected as the programming language because portability was desirable.

4.1 Contents of the C source code

The C code distribution has the files divided in six different directories, all present in the directory *c-code*. The directories are: *common_fx, decoder_fx, encoder_fx, lib_amr_fx include_fx* and *basic_op* . The distributed files with suffix "c" contain the source code and the files with suffix "h" are the header files.

Project and workspace files are provided in the directory *MSVC*.

4.2 Program execution

The Extended Adaptive Multi-Rate Wideband codec is implemented in two programs:

- (*encoder*) audio encoder;

- (*decoder*) audio decoder.

The programs should be called like:

- encoder [encoder options] –if <audio input file> -of <parameter file>;

- decoder [decoder options] –if <parameter file> -of <audio output file>.

The input files contain one or two channels of 16-bit linear encoded PCM audio samples stored in the *wav* file format and the parameter files contain encoded audio data and some additional flags.

The encoder and decoder options will be explained by running the applications without input arguments. See the file readme.txt for more information on how to run the *encoder* and *decoder* programs.

4.3 Code hierarchy

Tables 1 and 2 are call graphs that show the functions used in the audio codec.

Each column represents a call level and each cell a function. The functions contain calls to the functions in rightwards neighboring cells. The time order in the call graphs is from the top downwards as the processing of a frame advances. All standard C functions: printf(), fwrite(), etc. have been omitted. Also, no basic operations (add(), L_add(), mac(), etc.) or double precision extended operations (e.g. L_Extract()) appear in the graphs. The initialization of the static RAM (i.e. calling the _init functions) is also omitted.

The basic operations are not counted as extending the depth, therefore the deepest level in this software is level 6.

Table 1: Encoder call structure

TBA

Table 2: Decoder call structure

TBA

4.5 Variables, constants and tables

The data types of variables and tables used in the fixed point implementation are signed integers in 2's complement representation, defined by:

- **Word16** 16 bit variable;

- **Word32** 32 bit variable.

4.5.1 Description of constants used in the C-code

This subclause contains a listing of all global constants defined in cnst.h.

Table 5: Global constants

Constant	Value	Description
TBA		

4.5.2 Description of fixed tables used in the C-code

This section contains a listing of all fixed tables sorted by source file name and table name. All table data is declared as **Word16**.

Table 6: Fixed tables

File	Table name	Length	Description
			TBA

4.5.3   Static variables used in the C-code

In this section two tables that specify the static variables for the encoder and decoder respectively are shown. All static variables are declared within a C **struct.**

Table 7: Encoder static variables

Struct name	Variable	Type[Length]	Description
TBA			

Table 8: Decoder static variables

Struct name	Variable	Type[Length]	Description
TBA			

5   File formats

This section describes the file formats used by the encoder and decoder programs.

5.1   Audio file (encoder input / decoder output)

Audio files read by the encoder must be formatted as 16 bits PCM wave (*.wav) files. The decoder output is written as a 16 bit PCM wave file (*.wav).

Note that the decoder, with proper command line switch, can produce a mono wave file from a stereo bit-stream.

5.2   Parameter bitstream file (encoder output / decoder input)

TBA

Annex A (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2004-06	SP-24	SP-040425	-	-	Draft presented to TSG SA for information	-	1.0.0