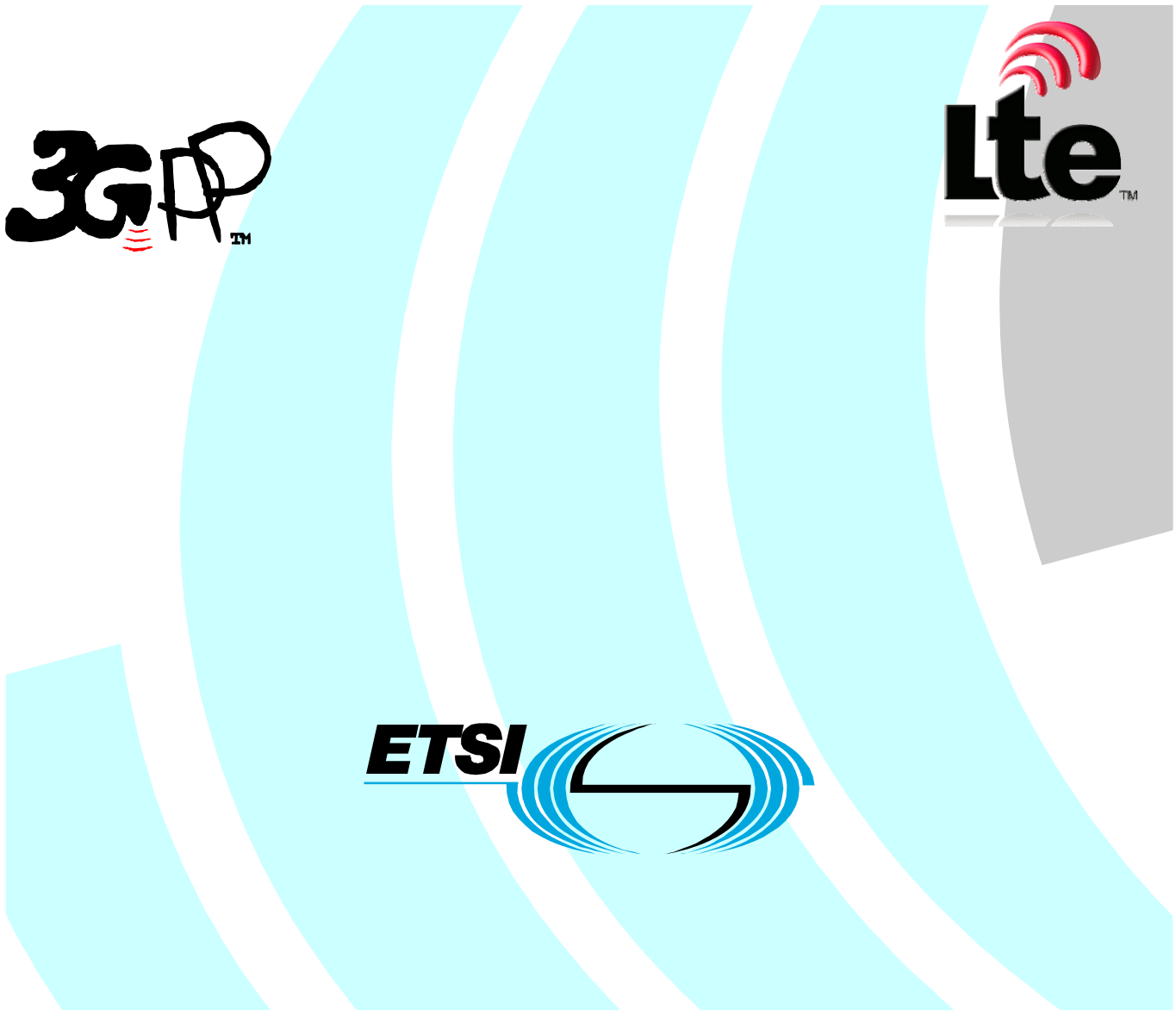
ETSI TS 126 411 V8.0.0 (2009-01)

Technical Specification

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
General audio codec audio processing functions;
Enhanced aacPlus general audio codec;
Fixed-point ANSI-C code
(3GPP TS 26.411 version 8.0.0 Release 8)



Reference
RTS/TSGS-0426411v800

Keywords
GSM, LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

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 Enhanced aacPlus codec [1].

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 26.401: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; General description".

[2] 3GPP TS 26.403: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Encoder specification; Advanced Audio Coding (AAC) part".

[3] 3GPP TS 26.404: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Encoder specification; Spectral Band Replication (SBR) part".

[4] 3GPP TS 26.405: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Encoder specification; Parametric stereo part".

[5] ISO/IEC 14496-3 (2001): "Information technology - Coding of audio-visual objects - Part 3: Audio".

[6] ISO/IEC 14496-3:2001/Amd.1:2003: "Bandwidth Extension".

[7] ISO/IEC 14496-3:2001/Amd.1:2003/Cor 1:2004.

[8] ISO/IEC 14496-3:2001/ Amd.2:2004: "Parametric coding for high quality audio".

[9] 3GPP TS 26.402: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Additional decoder tools".

[10] 3GPP TS 26.244: "Transparent end-to-end streaming service; 3GPP file format (3GP)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 26.401 [1], 3GPP TS 26.403 [2], 3GPP TS 26.404 [3], 3GPP TS 26.405 [4] and 3GPP TS 26.402 [9] apply.

3.2 Abbreviations

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

AAC	Advanced Audio Coding
aacPlus	Combination of MPEG-4 AAC and MPEG-4 Bandwidth extension (SBR)
ANSI	American National Standards Institute
Enhanced aacPlus	Combination of MPEG-4 AAC, MPEG-4 Bandwidth extension (SBR) and MPEG-4 Parametric Stereo
GSM	Global System for Mobile communications
I/O	Input/Output
MDCT	Modified Discrete Cosine Transform
QMF	Quadrature Mirror Filter
RAM	Random Access Memory
ROM	Read Only Memory
SBR	Spectral Band Replication

4 Fixed point ANSI-C code structure

This clause gives an overview of the structure of the fixed point ANSI-C code and provides an overview of the contents and organization of the C code attached to the present document.

The C code has been verified on the following systems:

- IBM PC/AT compatible computers with Windows XP, 2000 and Microsoft Visual C++ v.6.0 compiler.
- IBM PC/AT compatible computers with Linux OS and GCC v.3.3 compiler.

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

4.1 Contents of the fixed point ANSI-C source code

The C code distribution is organized in two directories for encoder and decoder and further into several subdirectories, reflecting the major building blocks of the Enhanced aacPlus codec. The file descriptions on root level as well as the directory structure is given as follows.

Table 1: Source code directory structure for the encoder (ETSI_aacPlusenc)

Directory	Description
README.txt	information on how to compile
Makefile	UNIX style encoder Makefile
enhAacPlusenc.dsw	Win32 MSVC 6.0 encoder workspace
enhAacPlusenc.dsp	Win32 MSVC 6.0 encoder makefile
src/	directory for the encoder frontend
etsiop_fastaacenc/	AAC encoder library
etsiop_resamplib/	resampler library
etsiop_sbrenclib/	SBR encoder library

Table 2: Source code directory structure for the decoder (ETSI_aacPlusdec)

Directory	Description
README.txt	information on how to compile
Makefile	UNIX style encoder Makefile
enhAacPlusdec.dsw	Win32 MSVC 6.0 decoder workspace
src/	directory for the decoder frontend
etsiop_aacdec	AAC decoder library
etsiop_sbrdeclib/	SBR decoder library

Table 3: Source code directory structure common for encoder and decoder

Directory	Description
etsiop_bitbuf/	bitstream reading/writing library
etsiop_ffrlib/	general purpose functionalities
etsioplib/	ETSI operators implementation
3g_lib/	precompiled libraries for audio and bitstream file format handling

The distributed files with suffix "c" contain the source code and the files with suffix "h" are the header files. Within the respective libraries, the RAM data is contained in "xxx_ram" files with suffix "c", the ROM data is contained in "xxx_rom" files with suffix "c". Makefiles are provided for the platforms in which the C code has been verified (listed above).

A list of source code files is given below:

Table 4: Encoder source code files

Directory	Module
src/	main.c
	mp4file.c
etsiop_fastaacenc/	qc_main.c
	aacenc.c
	ms_stereo.c
	spreading.c
	interface.c
	bit_cnt.c
	adj_thr.c
	quantize.c
	psy_configuration.c
	sf_estim.c
	tns_param.c
	grp_data.c
	pre_echo_control.c
	stprepro.c
	tns.c
	dyn_bits.c
	psy_main.c
	channel_map.c
	block_switch.c
	band_nrg.c
	transform.c
	bitenc.c
	line_pe.c
	stat_bits.c
etsiop_sbrenc/	qmf_enc.c
	ton_corr.c
	fram_gen.c
	env_bit.c
	env_est.c
	mh_det.c
	hybrid.c
	bit_sbr.c
	ps_bitenc.c
	sbr_main.c
	tran_det.c
	sbr_misc.c
	code_env.c
	nf_est.c
	freq_sca.c
	invf_est.c
	ps_enc.c
etsiop_resamplib/	downsample_FIR.c

Table 5: Decoder source code files

Directory	Module
src/	main.c
	fileifc.c
	spline_resampler.c
etsiop_aacdec/	aacdecoder.c
	streaminfo.c
	channelinfo.c
	stereo.c
	longblock.c
	shortblock.c
	pulsedata.c
	block.c
	pns.c
	imdct.c
	tns.c
	bitstream.c
	channel.c
	conceal.c
	datastream.c
etsiop_sbrdeclib/	env_dec.c
	aacpluscheck.c
	env_calc.c
	lpp_tran.c
	sbrdecoder.c
	sbr_dec.c
	sbr_crc.c
	hybrid.c
	ps_bitdec.c
	env_extr.c
	freq_sca.c
	ps_dec.c
	qmf_dec.c

Table 6: Common source code files

Directory	Module
etsiop_bitbuf/	bitbuffer.c
etsiop_ffrlib/	fft_32x32.c
	transcendent.c
	transcendent_enc.c
	intrinsics.c
	vector.c

4.2 Program execution

The Enhanced aacPlus codec is implemented in two programs:

- enhAacPlusEnc.exe.
- enhAacPlusDec.exe.

The programs should be called like:

- enhAacPlusEnc.exe <wav_file> <bitstream_file> <bitrate> <(m)ono/(s)tereo>.
- enhAacPlusDec.exe <bitstream_file> <wav_file> <mode> [error_pattern_file].

The audio files contain 16-bit linear encoded PCM samples with wav header, the bitstream files are of 3GPP type an the error pattern file is a ASCII file, see clause 5.

The encoder and decoder command line handling is also explained by running the applications without input arguments.

4.3 Memory requirements

The data types of variables and tables used in the fixed-point implementation are defined by the ETSI operator data types, the following types are used:

- Word8.
- Word16.
- Word32.
- Flag.

4.3.1 Constants and tables

This clause contains a listing of all constants and tables contributing to the ROM requirements of the encoder and decoder.

Table 7: Encoder constants and tables

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
LongWindowSine	Word16	1 024	aac_rom.c	Window coefficients
ShortWindowSine	Word16	128	aac_rom.c	Window coefficients
LongWindowKBD	Word16	1 024	aac_rom.c	Window coefficients
fftTwiddleTable	Word16	513	aac_rom.c	FFT twiddle coefficients
formfac_sqrttable	Word32	192	aac_rom.c	Lookup table for efficient sqrt implementation
mTab_3_4	Word32	1 024	aac_rom.c	Quantizer table, used for efficient pow () implementation
mTab_4_3	Word32	1 024	aac_rom.c	Inverse quantizer table, used for efficient pow () implementation
pow2tominusNover16	Word16	17	aac_rom.c	Lookup table for efficient pow() implementation
specExpMantTableComb_enc	Word32	128	aac_rom.c	Lookup table for efficient inverse quantizer implementation
specExpTableComb_enc	Word8	32	aac_rom.c	Lookup table for efficient inverse quantizer implementation
quantBorders	Word16	16	aac_rom.c	Lookup table for efficient distortion calculation implementation
quantBorders	Word16	12	aac_rom.c	Lookup table for efficient distortion calculation implementation
p_8000_mono_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_8000_stereo_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_8000_mono_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_8000_stereo_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_16000_mono_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_16000_stereo_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_16000_mono_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_16000_stereo_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_24000_mono_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_24000_stereo_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_24000_mono_short	TNS_CONFIG_T	5	aac_rom.c	TNS tuning parameters

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
	ABULATED			
p_24000_stereo_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_32000_mono_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_32000_stereo_long	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_32000_mono_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
p_32000_stereo_short	TNS_CONFIG_TABULATED	5	aac_rom.c	TNS tuning parameters
m_log2_table	Word32	64	aac_rom.c	Lookup table for efficient ld() implementation
tnsCoeff3	Word32	16	aac_rom.c	TNS filter coefficients
tnsCoeff3Borders	Word32	16	aac_rom.c	TNS filter borders
tnsCoeff4	Word32	32	aac_rom.c	TNS filter coefficients
tnsCoeff4Borders	Word32	32	aac_rom.c	TNS filter borders
tnsInfoTab	TNS_INFO_TAB	20	aac_rom.c	TNS bitrate to tuning mapping table
tnsMaxBandsTab	TNS_MAX_TAB_ENTRY	12	aac_rom.c	max. TNS bands per sampling rate table
huff_ltab1_2	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ltab3_4	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ltab5_6	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ltab7_8	Word16	64	aac_rom.c	Huffman codeword table AAC
huff_ltab9_10	Word16	169	aac_rom.c	Huffman codeword table AAC
huff_ltab11	Word16	289	aac_rom.c	Huffman codeword table AAC
huff_ltabscf	Word16	121	aac_rom.c	Huffman codeword table AAC
huff_ctab1	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ctab2	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ctab3	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ctab4	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ctab5	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ctab6	Word16	81	aac_rom.c	Huffman codeword table AAC
huff_ctab7	Word16	64	aac_rom.c	Huffman codeword table AAC
huff_ctab8	Word16	64	aac_rom.c	Huffman codeword table AAC
huff_ctab9	Word16	169	aac_rom.c	Huffman codeword table AAC
huff_ctab10	Word16	169	aac_rom.c	Huffman codeword table AAC
huff_ctab11	Word16	289	aac_rom.c	Huffman codeword table AAC
huff_ctabscf	Word32	242	aac_rom.c	Huffman codeword table AAC
sfb_16000_long_1024	Word8	22	aac_rom.c	Scalefactor band table
sfb_16000_short_128	Word8	8	aac_rom.c	Scalefactor band table
sfb_22050_long_1024	Word8	24	aac_rom.c	Scalefactor band table
sfb_22050_short_128	Word8	8	aac_rom.c	Scalefactor band table
sfb_24000_long_1024	Word8	24	aac_rom.c	Scalefactor band table
sfb_24000_short_128	Word8	8	aac_rom.c	Scalefactor band table
p4_13	Word32	26	sbr_rom.c	Hybrid filterbank coefficients
p8_13	Word32	26	sbr_rom.c	Hybrid filterbank coefficients
sbr_cos_twiddle_L32_enc	Word16	16	sbr_rom.c	QMF filterbank twiddle table
sbr_cos_twiddle_L64_enc	Word16	32	sbr_rom.c	QMF filterbank twiddle table
sbr_sin_twiddle_L32_enc	Word16	16	sbr_rom.c	QMF filterbank twiddle table
sbr_sin_twiddle_L64_enc	Word16	32	sbr_rom.c	QMF filterbank twiddle table
sbr_alt_sin_twiddle_L32_enc	Word16	17	sbr_rom.c	QMF filterbank twiddle table
sbr_alt_sin_twiddle_L64_enc	Word16	33	sbr_rom.c	QMF filterbank twiddle table
sbr_qmf_64_640_enc	Word16	330	sbr_rom.c	QMF window coefficients
aBookPsIidTimeCode	Word32	58	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIidFreqCode	Word32	58	sbr_rom.c	Huffman codeword table Parametric Stereo
aHybridResolution	Word16	3	sbr_rom.c	Number of hybrid bands in each QMF band
hiResBandBorders	Word8	11	sbr_rom.c	Borders of Parametric Stereo bins
groupBordersMix	Word8	15	sbr_rom.c	Borders of Parametric Stereo groups
bins2groupMap	Word32	58	sbr_rom.c	Mapping of Parametric Stereo bins to Parametric Stereo groups
iidQuantLeft	Word32	14	sbr_rom.c	IID quant values for Parametric Stereo

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
iidQuantRight	Word32	14	sbr_rom.c	IID quant values for Parametric Stereo
iccQuant	Word32	16	sbr_rom.c	ICC quant values for Parametric Stereo
v_Huff_envelopeLevelC10T	Word32	242	sbr_rom.c	Huffman codeword table SBR
v_Huff_envelopeLevelC10F	Word32	242	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceC10F	Word32	98	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceC10T	Word32	98	sbr_rom.c	Huffman codeword table SBR
v_Huff_envelopeLevelC11T	Word32	126	sbr_rom.c	Huffman codeword table SBR
v_Huff_NoiseLevelC11T	Word32	126	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceC11T	Word32	50	sbr_rom.c	Huffman codeword table SBR
bookSbrNoiseBalanceC11T	Word32	50	sbr_rom.c	Huffman codeword table SBR
v_Huff_envelopeLevelC11F	Word32	126	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceC11F	Word32	50	sbr_rom.c	Huffman codeword table SBR
aBookPsIidTimeLength	Word8	15	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIidFreqLength	Word8	15	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIccFreqLength	Word8	8	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIccTimeLength	Word8	8	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIccFreqCode	Word16	15	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIccTimeCode;	Word16	15	sbr_rom.c	Huffman codeword table Parametric Stereo
v_Huff_envelopeLevelL10T	Word8	61	sbr_rom.c	Huffman codeword table SBR
v_Huff_envelopeLevelL10F	Word8	61	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceL10F	Word8	25	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceL10T	Word8	25	sbr_rom.c	Huffman codeword table SBR
v_Huff_envelopeLevelL11T	Word8	32	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceL11T	Word8	13	sbr_rom.c	Huffman codeword table SBR
v_Huff_NoiseLevelL11T	Word8	32	sbr_rom.c	Huffman codeword table SBR
bookSbrNoiseBalanceL11T	Word8	13	sbr_rom.c	Huffman codeword table SBR
v_Huff_envelopeLevelL11F	Word8	32	sbr_rom.c	Huffman codeword table SBR
bookSbrEnvBalanceL11F	Word8	13	sbr_rom.c	Huffman codeword table SBR
logDualisTable	Word16	65	transcendent.c	Lookup table for efficient log() implementation
pow2Table	Word32	512	transcendent_enc.c	Lookup table for efficient pow() implementation
FIRenc_band24	Word16	49	downsample_FIR.c	FIR filter coefficients for 2:1 resampling
FIRenc_band63	Word16	127	downsample_FIR.c	FIR filter coefficients for 3:2 resampling
tuningTable	tuningTable	294	sbr_main.c	SBR tuning parameters
Sum		**11282**		

Table 8: Decoder constants and tables

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
tnsCoeff3	Word16	8	aac_rom.c	TNS filter coefficients
tnsCoeff4	Word16	16	aac_rom.c	TNS filter coefficients
trigData	Word16	513	aac_rom.c	Sine table, used for efficient sin(), cos()
OnlyLongWindowKBD	Word16	1 024	aac_rom.c	Window coefficients
OnlyShortWindowKBD	Word16	128	aac_rom.c	Window coefficients
OnlyLongWindowSine	Word16	1 024	aac_rom.c	Window coefficients
OnlyShortWindowSine	Word16	128	aac_rom.c	Window coefficients
sfb_48_1024	Word16	50	aac_rom.c	Scalefactor band table
sfb_48_128	Word16	15	aac_rom.c	Scalefactor band table
sfb_32_1024	Word16	51	aac_rom.c	Scalefactor band table
sfb_24_1024	Word16	49	aac_rom.c	Scalefactor band table
sfb_24_128	Word16	16	aac_rom.c	Scalefactor band table
sfb_16_1024	Word16	44	aac_rom.c	Scalefactor band table
sfb_16_128	Word16	16	aac_rom.c	Scalefactor band table
sfb_8_1024	Word16	41	aac_rom.c	Scalefactor band table
sfb_8_128	Word16	16	aac_rom.c	Scalefactor band table
HuffmanCodeBook_1	Word16	204	aac_rom.c	Huffman codeword table AAC

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
HuffmanCodeBook_2	Word16	156	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_3	Word16	156	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_4	Word16	152	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_5	Word16	164	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_6	Word16	160	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_7	Word16	124	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_8	Word16	124	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_9	Word16	336	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_10	Word16	328	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_11	Word16	544	aac_rom.c	Huffman codeword table AAC
HuffmanCodeBook_SCL	Word16	260	aac_rom.c	Huffman codeword table AAC
SamplingRateInfoTable	Word16	45	aac_rom.c	Sampling rate to scalefactor mapping table AAC
HuffmanCodeBooks	mixed	52	aac_rom.c	Huffman codeword table AAC
specExpTableComb	Word8	52	aac_rom.c	Lookup table for efficient exponent implementation
specExpTableMant	Word32	104	aac_rom.c	Lookup table for efficient exponent implementation, mantissa
InverseQuantTable	Word32	514	aac_rom.c	Lookup table for efficient inverse quantizer implementation
sgn_mask	Word16	3	aac_rom.c	Helper vector for efficient bit-wise operations in the TNS module
neg_mask	Word16	3	aac_rom.c	Helper vector for efficient bit-wise operations in the TNS module
tns_max_bands_tbl	Word8	18	aac_rom.c	max. TNS bands per sampling rate table
sbr_limGains_m	Word16	4	sbr_rom.c	SBR limiter gain values. mantissa
sbr_limGains_e	Word16	4	sbr_rom.c	SBR limiter gain values, exponent
sbr_limiterBandsPerOctaveDiv4	Word16	4	sbr_rom.c	Number of SBR limiter bands, divided by 4
sbr_smoothFilter	Word16	4	sbr_rom.c	Smoothing filter for gain values
sbr_invIntTable	Word16	49	sbr_rom.c	Table of 1/x function
sbr_randomPhase	Word16 or Word32	1 120	sbr_rom.c	Random numbers for SBR noise addition and PNS. Note: for mono only case the data type Word16 can be used.
sbr_qmf_64_640	Word16	340	sbr_rom.c	QMF window coefficients
sbr_ralt_sin_twiddle_L64	Word16	64	sbr_rom.c	FFT twiddle table
sbr_ralt_sin_twiddle_L32	Word16	32	sbr_rom.c	FFT twiddle table
sbr_cos_twiddle_L64	Word16	32	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_cos_twiddle_L32	Word16	16	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_sin_twiddle_L64	Word16	32	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_sin_twiddle_L32	Word16	16	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_alt_sin_twiddle_L64	Word16	33	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_cos_twiddle_ds_L32	Word16	32	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_sin_twiddle_ds_L32	Word16	32	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_t_cos_L32	Wrod16	32	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
sbr_t_sin_L32	Word16	32	sbr_rom.c	FFT twiddle table, obsolete for mono only decoder
pHybridResolution	Word16	3	sbr_rom.c	Parametric Stereo hybrid filterbank resolution
groupShift	Word16	6	sbr_rom.c	Parametric Stereo shift bits for uppermost frequency groups
aRevLinkDecaySer	Word16	3	sbr_rom.c	Parametric Stereo all-pass filter coefficients
aFractDelayPhaseFactorReQmf	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorImQmf	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorReSubQmf	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
aFractDelayPhaseFactorImSubQmf	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerReQmf0	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerReQmf1	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerReQmf2	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerImQmf0	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerImQmf1	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerImQmf2	Word16	20	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerReSubQmf0	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerReSubQmf1	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerReSubQmf2	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerImSubQmf0	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerImSubQmf1	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aFractDelayPhaseFactorSerImSubQmf2	Word16	10	sbr_rom.c	Parametric Stereo phase rotation factor
aaFractDelayPhaseFactorSerReQmf	Word16	3	sbr_rom.c	Parametric Stereo phase rotation factor
aaFractDelayPhaseFactorSerImQmf	Word16	3	sbr_rom.c	Parametric Stereo phase rotation factor
aaFractDelayPhaseFactorSerReSubQmf	Word16	3	sbr_rom.c	Parametric Stereo phase rotation factor
aaFractDelayPhaseFactorSerImSubQmf	Word16	3	sbr_rom.c	Parametric Stereo phase rotation factor
scaleFactors	Word16	15	sbr_rom.c	Parametric Stereo quantization table
scaleFactorsFine	Word16	31	sbr_rom.c	Parametric Stereo quantization table
alphas	Word16	8	sbr_rom.c	Parametric Stereo quantization table
p2_6	Word16	6	sbr_rom.c	Hybrid filterbank coefficients
p8_13	Word16	13	sbr_rom.c	Hybrid filterbank coefficients
sbr_start_freq_16	Word8	16	sbr_rom.c	SBR frequency scale index
sbr_start_freq_22	Word8	16	sbr_rom.c	SBR frequency scale index
sbr_start_freq_24	Word8	16	sbr_rom.c	SBR frequency scale index
sbr_start_freq_32	Word8	16	sbr_rom.c	SBR frequency scale index
sbr_start_freq_44	Word8	16	sbr_rom.c	SBR frequency scale index
sbr_start_freq_48	Word8	16	sbr_rom.c	SBR frequency scale index
sbr_frame_info1_16	Word16	18	sbr_rom.c	SBR frequency scale index
sbr_frame_info2_16	Word16	18	sbr_rom.c	SBR frequency scale index
sbr_frame_info4_16	Word16	18	sbr_rom.c	SBR frequency scale index
sbr_huffBook_EnvLevel10T	Word16	120	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvLevel10F	Word16	120	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvBalance10T	Word16	48	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvBalance10F	Word16	48	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvLevel11T	Word16	62	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvLevel11F	Word16	62	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvBalance11T	Word16	24	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_EnvBalance11F	Word16	24	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_NoiseLevel11T	Word16	62	sbr_rom.c	Huffman codeword table SBR
sbr_huffBook_NoiseBalance11T	Word16	24	sbr_rom.c	Huffman codeword table SBR
aRevLinkDelaySer	Word16	3	sbr_rom.c	Parametric Stereo all-pass delay line lengths
groupBorders	Word16	23	sbr_rom.c	Borders of Parametric Stereo groups
aGroupToBin	Word16	22	sbr_rom.c	Parametric Stereo group to bin table
aHybridToBin	Word16	12	sbr_rom.c	Parametric Stereo hybrid to bin table
aDelayToBin	Word16	20	sbr_rom.c	Parametric Stereo delay to bin table
aBookPsIidTimeDecode	Word16	28	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIidFreqDecode	Word16	28	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIccTimeDecode	Word16	14	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIccFreqDecode	Word16	14	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIidFineTimeDecode	Word16	60	sbr_rom.c	Huffman codeword table Parametric Stereo
aBookPsIidFineFreqDecode	Word16	60	sbr_rom.c	Huffman codeword table Parametric Stereo
sbr_defaultHeader	mixed	22	sbr_rom.c	Default SBR header data

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
logDualisTable	Word16	65	transcendent.c	Lookup table for efficient log() implementation
invTable	Word16	256	transcendent.c	Lookup table for efficient 1/x implementation
Sum		**10222**		

NOTE: All Parametric Stereo related coefficients which are marked as such in the table above are obsolete for mono only capable decoders.

4.3.2 Static memory

This clause contains a listing of all static buffers contributing to the RAM requirements of the encoder and decoder.

Table 9: Encoder static memory

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
mdctDelayBuffer	Word16	3200	aac_ram.c	Time domain input signal delay , only half the size for mono only encoder
quantSpec	Word16	2048	aac_ram.c	Quantized spectrum, only half the size for mono only encoder
scf	Word16	120	aac_ram.c	Scalefactors, only half the size for mono only encoder
maxValueInSfb	Word16	120	aac_ram.c	Max. value per calefactor, only half the size for mono only encoder
sideInfoTabLong	Word16	52	aac_ram.c	Table lookup for side information, long blocks
sideInfoTabShort	Word16	16	aac_ram.c	Table lookup for side information, short blocks
aacEncoder	AAC_ENCODER	6 851	aacenc.c	AAC encoder instance, can be reduced to 3809 for mono only encoder
sbr_QmfStatesAnalysis	Word16	1 280	sbr_ram.c	QMF filterbank states buffer, only half the size for mono only encoder
sbr_QmfStatesSynthesis	Word32	640	sbr_ram.c	QMF filterbank states buffer, not needed for mono only encoder
sbr_envYBuffer	Word32	8 192	sbr_ram.c	QMF band energy buffer, only half the size for mono only encoder
sbr_quotaMatrix	Word32	1 024	sbr_ram.c	Tonality values. only half the size for mono only encoder
sbr_thresholds	Word32	256	sbr_ram.c	Detector parameters, only half the size for mono only encoder
sbr_toncorrBuff	Word32	2 160	sbr_ram.c	Detector value buffer, only half the size for mono only encoder
ps_IccDataBuffer	Word32	40	sbr_ram.c	ICC buffer, not needed for mono only encoder
ps_IidDataBuffer	Word32	40	sbr_ram.c	Iid buffer, not needed for mono only encoder
ps_histQmfBuffer	Word16	1 608	sbr_ram.c	QMF history buffer, not needed for mono only encoder
sbr_freqBandTableLO	Word16	14	sbr_ram.c	SBR frequency band table, low resolution
sbr_freqBandTableHI	Word16	28	sbr_ram.c	SBR frequency band table, high resolution
sbr_v_k_master	Word16	28	sbr_ram.c	SBR frequency band table index
sbr_guideScfb	Word16	54	sbr_ram.c	Additional sine detection parameter, only half the size for mono only encoder
sbr_detectionVectors	Word16	108	sbr_ram.c	Additional sine detection parameter, only half the size for mono only encoder
sbr_prevEnvelopeCompensation	Word16	54	sbr_ram.c	Additional sine detection parameter, only half the size for mono only encoder
sbr_guideVectorDetected	Word16	108	sbr_ram.c	Additional sine detection parameter, only half the size for mono only encoder

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
sbrEncoder	SBR_ENCODER	3 650	sbr_main.c	SBR encoder instance, can be reduced to 1722 for mono only encoder
outputBuffer	Word8	384	main.c	Bitstream output buffer
inputBuffer	Word16	8 234	main.c	Time domain input signal buffer, only half the size for mono only encoder
up2Sampler	RESAMPLER_FIR_2_1	144	main.c	1:2 FIR resampler instance (includes states) , only half the size for mono only encoder
down3Sampler	RESAMPLER_FIR_3_2	208	main.c	3:2 FIR resampler instance (includes states) , only half the size for mono only encoder
down2Sampler	RESAMPLER_FIR_2_1	144	main.c	2:1 FIR resampler instance (includes states) , only half the size for mono only encoder
Sum		**40 643**		

Table 10: Decoder static memory

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
OverlapBuffer[nChan]	Word16	1 024	aac_ram.c	Delay buffer for overlap and add, only half the size for mono only decoder
AacDecoderInstance	AAC_DECODER_INSTANCE	12	aacdecoder.c	AAC decoder instance
sbrPayloadBuffer	Word8	540	aacdecoder.c	SBR payload buffer
StreamInfo	CStreamInfo	7	aac_ram.c	Bitstream information
AacDecoderStaticChannelInfo[nChan]	CaacDecoderStaticChannelInfo	16	aac_ram.c	Channel information, only half the size for mono only decoder
PnsStaticInterChannelData	Word16	2	aac_ram.c	PNS static data
ConcealmentInfoInstance	CConcealmentInfo	4 127	aac_ram.c	Concealment static data
sbr_CodecQmfStatesAnalysis	Word16	640	sbr_ram.c	QMF analysis filter bank states
sbr_GainSmooth_m	Word16	96	sbr_ram.c	Gain smoothing filter states, mantissa
sbr_GainSmooth_e	Word16	96	sbr_ram.c	Gain smoothing filter states, exponent
sbr_NoiseSmooth_m	Word16	96	sbr_ram.c	Noise level smoothing filter states, mantissa
sbr_QmfStatesSynthesis	Word16	2 560	sbr_ram.c	QMF synthesis filter bank states
sbr_OverlapBuffer	Word32	3 072	sbr_ram.c	SBR delay buffer, only half the size for mono only decoder
sbr_LpcFilterStatesReal	Word32	256	sbr_ram.c	LPC filter states
sbr_LpcFilterStatesImag	Word32	256	sbr_ram.c	LPC filter states, obsolete for mono only decoder
sbr_TransposerSettings	TRANSPOSER_SETTINGS	61	sbr_ram.c	Transposer configuration parameters
FreqBandData	FREQ_BAND_DATA	164	sbr_ram.c	SBR Frequency band information
PrevFrameData[nChan]	SBR_PREV_FRAME_DATA	120	sbr_ram.c	SBR previous frame data, only half the size for mono only decoder
sbrDecoderInstance	SBR_DECODER_INSTANCE	797	sbrdecoder.c	SBR decoder instance
timeData	Word16	4 096	main.c	Output buffer for time-domain signal, only half the size for mono only decoder
inBuffer	short	768	main.c	Input buffer for bitstream
splineResamplerInstance	SPLINE_RESAMPLER	19	spline_resampler.c	Spline resampler instance
Sum		**18825**		

4.3.3 Dynamic memory

This clause contains a listing of all dynamic buffers contributing to the RAM requirements of the encoder and decoder. Dynamic memory can be re-used outside of the encoder or decoder application.

Table 11: Encoder dynamic memory

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
ps_tempQmfBuffer	Word16	1 608	sbr_ram.c	NOTE: reused in AAC encoder, not needed for mono only encoder
sbr_envRBuffer	Word16	4 096	sbr_ram.c	NOTE: reused in AAC encoder
sbr_envIBuffer	Word16	4 096	sbr_ram.c	NOTE: reused in AAC encoder
sbr_transients	Word32	384	sbr_ram.c	NOTE: reused in AAC encoder
Sum		**10 184**		

Table 12: Decoder dynamic memory

Name	Data type	Size [16-bit word]	Allocated in Source File	Description
WorkBufferCore	Word32	4 096	aac_ram.c	NOTE: reused in SBR decoder
WorkBuffer1	Word32	4 096	sbr_ram.c	Work buffer, obsolete for mono only decoder
Sum		**8 192**		

4.3.4 Maximum stack size

This clause contains tables for the encoder and the decoder which describe the call stack that results in the maximum stack size usage.

Table 13: Encoder call stack

Function	Local variables / Function call arguments	Stack used [16-bit words]
main	AACENC_CONFIG config;	8
	Word16 error;	1
	Flag bEncodeMono;	1
	Word16 ch;	1
	Word16 dummy;	1
	Word32 bitrate;	2
	Word16 nChannelsAAC, nChannelsSBR;	2
	Word32 sampleRateAAC;	2
	Word32 frmCnt ;	2
	Word32 bandwidth;	1
	UWord16 numAncDataBytes;	2
	UWord8 ancDataBytes[MAX_PAYLOAD_SIZE];	64
	Word32 numSamplesRead;	2
	Flag bDingleRate;	1
	Flag useParametricStereo;	1
	Word32 coreWriteOffset;	2
	Word32 coreReadOffset;	2
	Word32 envWriteOffset;	2
	Word32 envReadOffset;	2
	Word32 writeOffset;	2
	struct AAC_ENCODER *hAacEnc;	1
	struct SBR_ENCODER *hEnvEnc;	1
	Flag bDoUpsample;	1
	Flag upsampleReadOffset;	1
	Word16 inSamples;	1
	Flag bDo32Resample;	1
	Word16 nRuns;	1
	Word16 *resamplerScratch;	1
	UWord32 numOutBytes;	2
	Word32 i;	2
	UWord8 *bitstreamOut;	1
	Word32 *nOutBytes;	1
	Word16 numEncSamples;	1
	Flag downsample;	1
		=116
AacEncEncode	struct AAC_ENCODER *aacEnc;	1
	Word16 *timeSignal;	1
	const UWord8 *ancBytes;	1
	UWord16 *numAncBytes;	1
	UWord8 *outBytes;	1
	Word32 *numOutBytes;	1
	ELEMENT_INFO *elInfo;	1
	Word16 globUsedBits;	1
	Word16 ancDataBytes, ancDataBytesLeft;	1
		=9
QCMain	QC_STATE* hQC,	1
	Word16 nChannels,	1
	ELEMENT_BITS* elBits,	1
	ATS_ELEMENT* adjThrStateElement,	1
	PSY_OUT_CHANNEL psyOutChannel[MAX_CHANNELS_PER_ELEM],	1
	PSY_OUT_ELEMENT* psyOutElement,	1
	QC_OUT_CHANNEL qcOutChannel[MAX_CHANNELS_PER_ELEM],	1
	QC_OUT_ELEMENT* qcOutElement,	1
	Word16 ancillaryDataBytes	1
	Word16 ch;	1
	Word16 logSfbFormFactor[MAX_CHANNELS_PER_ELEM][MAX_GROUPED_SFB];	120
	Word16 sfbNRelevantLines[MAX_CHANNELS_PER_ELEM][MAX_GROUPED_SFB];	120
	Word16 logSfbEnergy[MAX_CHANNELS_PER_ELEM][MAX_GROUPED_SFB];	120
	Word16 maxChDynBits[MAX_CHANNELS_PER_ELEM];	2
	Word16 chBitDistribution[MAX_CHANNELS_PER_ELEM];	2
		=375

Function	Local variables / Function call arguments	Stack used [16-bit words]
AdjustThresholds	ADJ_THR_STATE *adjThrState; ATS_ELEMENT *AdjThrStateElement; PSY_OUT_CHANNEL psyOutChannel[MAX_CHANNELS_PER_ELEM]; PSY_OUT_ELEMENT *psyOutElement; Word16 *chBitDistribution; Word16 logSfbEnergy[MAX_CHANNELS_PER_ELEM][MAX_GROUPED_SFB]; Word16 sfbNRelevantLines[MAX_CHANNELS][MAX_GROUPED_SFB]; const Word16 nChannels; QC_OUT_ELEMENT *qcOE; const Word16 avgBits; const Word16 bitresBits; const Word16 maxBitresBits; const Word16 maxBitFac; const Word16 sideInfoBits; Word16 noRedPe, grantedPe, grantedPeCorr; Word16 curWindowSequence; PE_DATA peData; Word16 bitFactor; Word16 ch;	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 667 1 1 =686
adaptThresholdsToPe	PSY_OUT_CHANNEL psyOutChannel[MAX_CHANNELS_PER_ELEM]; PSY_OUT_ELEMENT *psyOutElement; Word16 logSfbEnergy[MAX_CHANNELS_PER_ELEM][MAX_GROUPED_SFB]; PE_DATA *peData; const Word16 nChannels; const Word16 desiredPe; AH_PARAM *ahParam; MINSNR_ADAPT_PARAM *msaParam; Word16 noRedPe, redPe, redPeNoAH; Word16 constPart, constPartNoAH; Word16 nActiveLines, nActiveLinesNoAH; Word16 desiredPeNoAH; Word32 redVal, avgThrExp; Word16 iter;	1 1 1 1 1 1 1 1 4 2 2 1 4 1 =22
correctThresh	PSY_OUT_CHANNEL psyOutChannel[MAX_CHANNELS_PER_ELEM]; Word16 ahFlag[MAX_CHANNELS][MAX_GROUPED_SFB]; PE_DATA *peData; Word32 thrExp[MAX_CHANNELS][MAX_GROUPED_SFB]; const Word32 redVal; const Word16 nChannels; const Word32 deltaPe; Word16 ch, sfb, sfbGrp; PSY_OUT_CHANNEL *psyOutChan; PE_CHANNEL_DATA *peChanData; Word32 deltaSfbPe; Word32 sfbPeFactors[MAX_CHANNELS][MAX_GROUPED_SFB], normFactor; Word32 sfbEn, sfbThr; Word32 sfbThrReduced; Word32 thrFactor;	1 1 1 1 2 1 2 3 1 1 2 242 4 2 2 =266
ffr_pow2_xy	Word32 x, Word32 y; Word32 iPart,fPart; Word32 res; Word32 tmp, tmp2; Word16 shift, shift2; Word16 denom_h, denom_l;	4 4 2 4 2 2 =18
	Sum	**992**

Table 14: Decoder call stack

Function	Local variables / Function call arguments	Stack used [16-bit words]
main()	Flag endOfFile;	1
	Flag frameOk;	1
	Word32 i;	2
	Word32 written16;	2
	Word8 channelMode;	1
	struct BIT_BUF bitBuf, *hBitBuf;	14
	AACDECODER aacDecoderInfo;	1
	SBRBITSTREAM streamSBR;	6
	SBRDECODER sbrDecoderInfo;	1
	HANDLE_SPLINE_RESAMPLER splineResampler;	1
	Word32 frameSize;	2
	Word32 sampleRate, outputSampleRate;	4
	Word8 numChannels;	1
	Word8 numChannelsLast;	1
	Word16 numOutSamples;	1
	Flag bDownSample;	1
	Flag fosr16;	1
	Flag fosr8;	1
	Flag bBitstreamDownMix;	1
	Flag bValidMode;	1
		=44
applySBR()	SBRDECODER self,	1
	SBRBITSTREAM * Bitstr,	1
	Word16 *timeData,	1
	Word8 *numChannels,	1
	Flag SbrFrameOK,	1
	Flag bDownSample,	1
	Flag bBitstreamDownMix	1
	Word16 lr;	1
	Word32 i;	2
	Word8 prevStereo, prevPs;	2
	Word8 psPresent;	1
	Word8 stereo;	1
	Word8 readHeader;	1
	Word32 err;	2
	SBR_CHANNEL *SbrChannel;	1
	struct BIT_BUF bitBuf;	1
	HANDLE_SBR_HEADER_DATA hHeaderData;	1
	SBR_HEADER_STATUS headerStatus;	1
	Word32 codecFrameSize;	2
	SBR_SYNC_STATE initialSyncState;	1
	SBR_FRAME_DATA *hFrameDataLeft;	1
	SBR_FRAME_DATA *hFrameDataRight;	1
	Flag bUseLP;	1
		=28

Function	Local variables / Function call arguments	Stack used [16-bit words]
sbr_dec()	HANDLE_SBR_DEC hSbrDec, Word16 *timeIn, Word16 *timeOut, HANDLE_SBR_HEADER_DATA hHeaderData, HANDLE_SBR_FRAME_DATA hFrameData, HANDLE_SBR_PREV_ FRAME_DATA hPrevFrameData, HANDLE_PS_DEC hPS, HANDLE_SBR_QMF_FILTER_BANK hSynthesisQmfBankRight, SBR_SCALE_FACTOR *sbrScaleFactorRight, Flag applyProcessing, Flag bUseLP Word32 i, k; Word16 slot, reserve; Word16 saveLbScale; Word16 ov_len; Word32 *QmfBufferReal[MAX_ENV_COLS]; Word32 *QmfBufferImag[MAX_ENV_COLS]; Word32 *ptr; Word16 noCols;	1 1 1 1 1 1 1 1 1 1 1 1 4 2 1 1 38 38 1 1 = 98
cplxSynthesisQmfFiltering()	Word32 **qmfReal, Word32 **qmfImag, Word32 splitSlot, SBR_SCALE_FACTOR *sbrScaleFactor, Word16 *timeOut, HANDLE_SBR_QMF_FILTER_BANK qmfBank, HANDLE_PS_DEC hPS, Flag active, Flag bUseLP Word16 i, j; Word16 codScalefactor; Word16 sbrScalefactor; Word16 outScalefactor; Word16 scaleFactorLowBand; Word16 scaleFactorHighBand; Word16 *filterStates; Word32 qmfReal2[NO_SYNTHESIS_CHANNELS]; Word16 no_synthesis_channels; Word32 qmfRealTmp[NO_SYNTHESIS_CHANNELS]; Word32 *qmfImagTmp; Word8 env; Word16 scaleFactorLowBandSplit; Word32 *imagSlot;	1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 128 1 128 1 1 1 1 = 277
inverseModulationLP()	Word32 *qmfReal, Word32 *qmfReal2, const Word16 scaleFactorLowBand, const Word16 scaleFactorHighBand, HANDLE_SBR_QMF_FILTER_BANK synQmf Word16 i; const Word16 L; const Word16 M; Word32 timeOut[2*NO_SYNTHESIS_CHANNELS];	1 1 1 1 1 1 1 1 256 = 264
dct2 ()	Word32 *data, Word32 *tmp, Word32 L, HANDLE_SBR_QMF_FILTER_BANK qmfBank Word16 cosphi,sinphi; Word16 i; Word32 M; Word32 N; Word32 a1,a2,a3,a4,a5,a6;	1 1 2 1 2 1 2 2 12 = 24

Function	Local variables / Function call arguments	Stack used [16-bit words]
DSP_fft32x32s()	Word32 npoints Word32 *ptr_x	2 1 = 3
__fft32()	Word32 x[] Word16 m; Word32 re, ie, ro, io, rx, ix ; Word32 *x1m, *x2m, *x1mm, *x2mm;	2 1 12 4 = 19
__fft2x16()	Word32 x[] Word16 sch, sh, ch; Word32 a0, a1, a2, a3, a00, a10, a20, a30 ; Word32 tr, ti, vr, vi ; Word32 even [32], odd [32] ;	2 3 16 8 128 = 157
fixmul_32x16b()	Word32 a; Word16 b; Word16 hi1 lo1	2 1 2 = 5
	Sum	**919**

5 File formats

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

5.1 Audio input file (encoder input/decoder output)

The audio input files read by the encoder and written by the decoder are 16-bit PCM wave files. For convenient handling of wave files a precompiled audio-file format library is used.

5.2 Bitstream file format (encoder output/decoder input)

The encoder program writes and the decoder program reads raw frames packetized in access units as described by 3GPP TS 26.244 [10]. For packetization the ISO media library is used. A precompiled library is used.

5.3 Error pattern file (decoder input)

The decoder program can optionally process an additional input file which describes an error pattern. The format of the error pattern file is 1 character per line. Each line corresponds to one frame, where a "0" indicates that the respective frame has been transmitted without errors, while a "1" indicates that the corresponding frame has been lost and error concealment shall be applied by the decoder.

Annex A (informative): Weighted MOPS

The complexity numbers for the Enhanced aacPlus audio codec can be found in the following tables, the numbers have been derived using the "allcat.wav" item, which holds all the material from the selection test concatenated in one single item. For every test case the average and worst frame weighted MOPS figure has been derived. The worst case wMOPS figure over all test cases has been marked in **blue**.

The fixed-point C-code contains a pre-compiler directive named 'ACCOUNT_ETSIOP_OVERHEAD_SPLITWORD32'. If this pre-compiler directive is set during compilation, the complexity figures will be as given in Table 15. If this pre-compiler directive is not set during compilation, the complexity figures will be as given in Table 16. Regarding the details of the corresponding difference in the fixed-point C-code, please see the comments in the SplitWord32() function in intrinsics.c.

Table A.1: Weighted MOPS figures with ACCOUNT_ETSIOP_OVERHEAD_SPLITWORD32 set

	Test Case	Mono Encoder	Stereo Encoder	Decoder	Decoder, mono only
wMOPS [average / worst frame]	14m	26.51 / 28.87	26.50 / 31.61	19.15 / 21.20	14.73 / 16.80
	18s	---	61.38 / 65.25	35.18 / 38.04	15.14 / 17.39
	24m	29.51 / 34.28	29.51 / 34.26	20.98 / 23.84	15.93 / 18.74
	24s	---	63.47 / 68.17	37.35 / 40.98	15.93 / 18.72
	32s	---	64.61 / 71.02	**38.39 / 42.28**	16.47 / 19.60
	48s	---	**64.17 / 77.63**	32.65 / 38.46	21.96 / 26.83

Table A.2: Weighted MOPS figures with ACCOUNT_ETSIOP_OVERHEAD_SPLITWORD32 not set

	Test Case	Mono Encoder	Stereo Encoder	Decoder	Decoder, mono only
wMOPS [average / worst frame]	14m	23.80 / 25.41	23.79 / 29.04	15.86 / 17.73	12.38 / 14.38
	18s	---	51.14 / 53.92	29.35 / 32.07	12.77 / 14.95
	24m	26.52 /29.60	26.52 / 29.66	17.53 / 20.13	13.51 / 16.21
	24s	---	53.08 / 56.06	31.17 / 34.64	13.57 / 16.22
	32s	---	54.11 / 58.39	**32.08 / 35.75**	14.05 / 17.09
	48s	---	**57.46 / 65.89**	27.88 / 33.43	18.53 / 23.22

Annex B (informative): Change history

Change history							
Date	**TSG SA#**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2005-03	27	SP-050146			Approved at TSG SA#27	1.0.0	6.0.0
2005-06	28	SP-050251	001		Correction to C-code: Corrections to ETSI operator usage	6.0.0	6.1.0
2005-06	28	SP-050251	002		Correction to C-code: Avoid reading of non-initialized variable	6.0.0	6.1.0
2005-06	28	SP-050251	003		Correction to C-code: fixes for error concealment	6.0.0	6.1.0
2005-09	29	SP-050426	0004		Correction to C-code: enable 44.1 kHz input material encoding (fixed-point code)	6.1.0	6.2.0
2005-09	29	SP-050426	0005		Correction to C-Code: Bitrate dependancy of AAC block switching threshold	6.1.0	6.2.0
2005-09	29	SP-050426	0006		Correction to C-Code: prevent overflow in parameter estimation of Parametric Stereo IID value	6.1.0	6.2.0
2005-09	29	SP-050426	0007		Correction to C-Code: smoothing of SBR noise values	6.1.0	6.2.0
2005-09	29	SP-050426	0008		Correction to C-Code: removal of obsolete table (fixed-point code)	6.1.0	6.2.0
2005-09	29	SP-050426	0009		Correction to C-Code: rounding in TNS parcor coefficient calculation	6.1.0	6.2.0
2005-12	30	SP-050786	0011		Correction to C-code: encoder tuning table entry for 44.1 kHz was wrong	6.2.0	6.3.0
2005-12	30	SP-050786	0012		Correction to C-code: encoder bitrate switching simulation toolset	6.2.0	6.3.0
2006-09	33	SP-060595	0015	1	Correction of written specification: correct memory tables (fixed-point code)	6.3.0	6.4.0
2006-09	33	SP-060601	0013	2	Modification of C-code: Change of encoder bitrate border for Parametric Stereo usage (fixed-point code)	6.4.0	7.0.0
2006-12	34	SP-060849	0016		Correction to C-code: Correct the maximum possible SBR payload size (fixed point code)	7.0.0	7.1.0
2007-03	35	SP-070030	0018		Bugfix for the Enhanced aacPlus fixed point arithmetic library	7.1.0	7.2.0
2007-09	37	SP-070628	0020	1	Adding safety code to avoid encoder runtime-assertion	7.2.0	7.3.0
2007-09	37	SP-070628	0022	1	Prevent arithmetic overflow during estimation of perceptual entropy in AAC encoder	7.2.0	7.3.0
2007-09	37	SP-070628	0024	1	Prevent out-of-bounds memory access of encoder in parametric stereo mode	7.2.0	7.3.0
2008-12	42				Version for Release 8	7.3.0	8.0.0

History

Document history		
V8.0.0	January 2009	Publication