Contents for Subpart 3

Subpart 3: Speech Coding - CELP

3.1 Scope

3.1.1 General description of the CELP decoder

This subclause provides a brief overview of the CELP (Code Excited Linear Prediction) decoder. A basic block diagram of the CELP decoder is given in Figure 3.1.

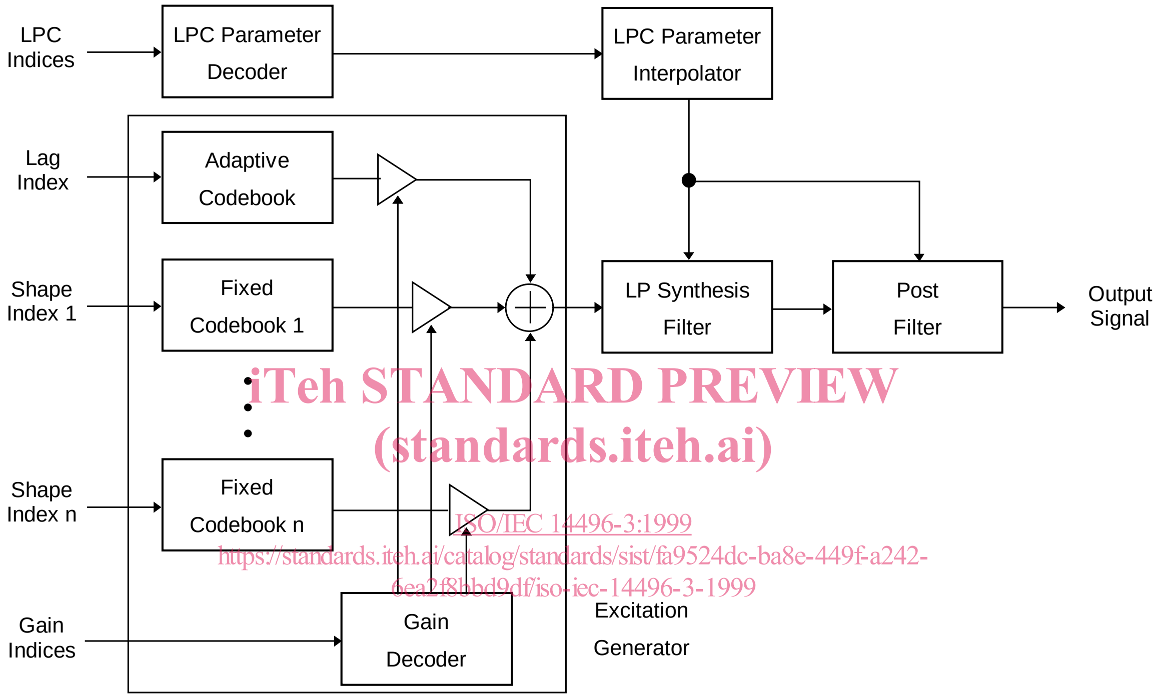

Figure 3.1 – Block diagram of a CELP decoder

The CELP decoder primarily consists of an excitation generator and a synthesis filter. Additionally, CELP decoders often include a post-filter. The excitation generator has an adaptive codebook to model periodic components, fixed codebooks to model random components and a gain decoder to represent a speech signal level. Indices for the codebooks and gains are provided by the encoder. The codebook indices (pitch-lag index for the adaptive codebook and shape index for the fixed codebook) and gain indices (adaptive and fixed codebook gains) are used to generate the excitation signal. It is then filtered by the linear predictive synthesis filter (LP synthesis filter). Filter coefficients are reconstructed using the LPC indices, then are interpolated with the filter coefficients of successive analysis frames. Finally, a post-filter can optionally be applied in order to enhance the speech quality.

3.1.2 Functionality of MPEG-4 CELP

MPEG-4 CELP is a generic coding algorithm with new functionalities. Conventional CELP coders offer compression at a single bitrate and are optimized for specific applications. Compression is one of the functions provided by MPEG-4 CELP, enabling the use of one basic coder for various applications. It provides scalability in bitrate and bandwidth, as well as the ability to generate bitstreams at arbitrary bitrates. The MPEG-4 CELP coder supports two sampling rates, namely, 8 and 16 kHz. The associated bandwidths are 100 – 3400 Hz for 8 kHz sampling rate and 50 – 7000 Hz for 16 kHz sampling rate.

3.1.2.1 Configuration of the MPEG-4 CELP coder

Two different tools can be used to generate the excitation signal. These are the Multi-Pulse Excitation (MPE) tool or the Regular-Pulse Excitation (RPE) tool. MPE is used for speech sampled at 8 kHz or 16 kHz. RPE is only used for speech sampled at 16 kHz. The two possible coding modes are summarized in Table 3.1.

Table 3.1 – Coding modes in the MPEG-4 CELP coder

Coding Mode	Excitation tool	Sampling rate
I	RPE	16 kHz
II	MPE	8, 16 kHz

3.1.2.2 Features of the MPEG-4 CELP coder

The MPEG-4 CELP coder offers the following functionality, depending on the coding mode.

Table 3.2 – Functionality of the MPEG-4 CELP coder

Coding Mode	Functionality
I	Multiple bitrates, FineRate Control
II	Multiple bitrates, Bitrate Scalability, Bandwidth Scalability, FineRate Control

Multiple bitrates: The available bitrates depend on the coding mode and the sampling rate. The following fixed bitrates are supported:

Table 3.3 – Fixed bitrates for the mode I coder

Bitrates for the 16 kHz sampling rate (bit/s)
14400, 16000, 18667, 22533

Table 3.4 – Fixed bitrates for the mode II coder

Bitrates for the 8 kHz sampling rate (bit/s)	Bitrates for the 16 kHz sampling rate (bit/s)
3850, 4250, 4650, 4900, 5200, 5500, 5700, 6000, 6200, 6300, 6600, 6900, 7100, 7300, 7700, 8300, 8700, 9100, 9500, 9900, 10300, 10500, 10700, 11000, 11400, 11800, 12000, 12200	10900, 11500, 12100, 12700, 13300, 13900, 14300, 14700, 15900, 17100, 17900, 18700, 19500, 20300, 21100, 13600, 14200, 14800, 15400, 16000, 16600, 17000, 17400, 18600, 19800, 20600, 21400, 22200, 23000, 23800

Fine Rate Control: Enables fine step bitrate control (permitting variable bitrate operation). This is achieved purely by controlling the transmission rate of the LPC parameters using a combinations of the two bitstream elements **interpolation_flag** and **LPC_present** flag. Using FineRate Control it is possible to vary the ratio of LPC-frames to total frames between 50% and 100%. This enables the bitrate to be decreased with respect to the anchor bitrate, as defined in the Semantics.

Bitrate Scalability: Bitrate scalability is provided by adding enhancement layers. Enhancement layers can be added with a step of 2000 bit/s for signals sampled at 8 kHz or 4000 bit/s for signals sampled at 16 kHz. A maximum of three enhancement layers may be combined with any bitrate chosen from Table 3.4.

Bandwidth Scalability: Bandwidth scalability to cover both sampling rates is achieved by incorporating a bandwidth extension tool in the CELP coder. This is an enhancement tool, supported in Mode II, which may be added if scalability from the 8 kHz sampling rate to the 16 kHz sampling rate is required. A complete coder with bandwidth scalability consists of a core CELP coder for the 8 kHz sampling rate and the bandwidth extension tool to provide a single layer of scalability. The core CELP coder for the 8 kHz sampling rate can comprise several layers. It should be noted that an 8 kHz sampling rate coder with this tool is not the same as a 16 kHz sampling rate coder. Both configurations (8 kHz sampling rate coder with bandwidth scalability and 16 kHz sampling rate coder) offer greater intelligibility and naturalness of decoded speech than does the 8 kHz coder alone because they expand the bandwidth to 7 kHz. The additional bitrate required for the bandwidth scalability tool can be selected from 4 discrete steps for each core layer bitrate as shown in Table 3.5.

Table 3.5 – Bitrates for the bandwidth scalable mode

Bitrate of the core layer (bit/s)	Additional bitrate (bit/s)
3850 - 4650	+9200, +10400, +11600, +12400
4900 - 5500	+9467, +10667, +11867, +12667
5700 - 10700	+10000, +11200, +12400, +13200
11000 - 12200	+11600, +12800, +14000, +14800

3.1.2.3 Algorithmic delay of MPEG-4 CELP modes

The algorithmic delay of the CELP coder comes from the frame length and an additional look ahead length. The frame length depends on the coding mode and the bitrate. The look ahead length, which is an informative parameter, also depends on the coding mode. The delays presented below are applicable to the modes where FineRate Control is off. When FineRate Control is on, additional one-frame delay is introduced. Bandwidth scalability in the mode II coder requires an additional look ahead of 5 ms due to down-sampling.

Table 3.6 – Delay and frame length for the mode I coder of the 16 kHz sampling rate

Bitrate for Mode I (bit/s)	Delay (ms)	Frame Length (ms)
14400	26.25	15
16000	18.75	10
18667	26.56	15
22533	26.75	15

Table 3.7 – Delay and frame length for the mode II coder of the 8 kHz sampling rate

Bitrate for Mode II (bit/s)	Delay (ms)	Frame Length (ms)
3850, 4250, 4650	45	40
4900, 5200, 5500, 6200	35	30
5700, 6000, 6300, 6600, 6900, 7100, 7300, 7700, 8300, 8700, 9100, 9500, 9900, 10300, 10500, 10700	25	20
11000, 11400, 11800, 12000, 12200	15	10

Table 3.8 – Delay and frame length for the mode II coder of the 16 kHz sampling rate

Bitrate for Mode II (bit/s)	Delay (ms)	Frame Length (ms)
10900, 11500, 12100, 12700, 13300, 13900, 14300, 14700, 15900, 17100, 17900, 18700, 19500, 20300, 21100	25	20
13600, 14200, 14800, 15400, 16000, 16600, 17000, 17400, 18600, 19800, 20600, 21400, 22200, 23000, 23800	15	10

3.2 Definitions

3.2.1 adaptive codebook: An approach to encode the long-term periodicity of the signal. The entries of the codebook consists of overlapping segments of past excitations.
3.2.2 bandwidth scalability: The possibility to change the bandwidth of the signal during transmission.
3.2.3 bitrate scalability: The possibility to transmit a subset of the bitstream and still decode the bitstream with the same decoder.
3.2.4 CELP: Code Excited Linear Prediction
3.2.5 demultiplexing: Splitting one bitstream into several.
3.2.6 excitation: The excitation signal represents the input to the LPC module. The signal consists of contributions that cannot be covered by the LPC model.
3.2.7 enhancement layer(s): The part(s) of the bitstream that is possible to drop in a transmission and still decode the bitstream.
3.2.8 fine rate control: The possibility to change the bitrate by, under some circumstances, skipping transmission of the LPC indices.

3.2.9 fixed codebook: The fixed codebook contains excitation vectors for the speech synthesis filter. The contents of the codebook are non-adaptive (i.e. fixed).
3.2.10 index: Number indicating the quantized value(s).
3.2.11 LPC: Linear Predictive Coding.
3.2.12 LSP: Line Spectral Pairs.
3.2.13 MPE: Multi Pulse Excitation.
3.2.14 multiplexing: Combining several bitstreams into one.
3.2.15 post-filter: This filter is applied to the output of the synthesis filter to enhance the perceptual quality of the reconstructed speech.
3.2.16 RPE: Regular Pulse Excitation.
3.2.17 unvoiced frame: Frame containing unvoiced speech which looks like random noise with no periodicity.
3.2.18 variable bitrate: The ability to permit a variable number of bits corresponding to one coded frame.
3.2.19 vector quantizer: Tool that quantizes several values to one index.
3.2.20 voiced frame: A voiced speech segment is known by its relatively high energy content, but more importantly it contains periodicity which is called the pitch of the voiced speech.

3.3 Bitstream syntax

CelpSpecificConfig()

CELP Base Layer

The CELP core in the unscalable mode or as the base layer in the scalable mode requires the following CelpSpecificConfig():

```
class CelpSpecificConfig (uint(4) samplingFrequencyIndex ) {
        CelpHeader (samplingFrequencyIndex);
}
```


CELP Enhancement Layer

The CELP core is used for both bitrate and bandwidth scalable modes. In the bitrate scalable mode, the enhancement layer requires no CelpSpecificConfig(). In the bandwidth scalable mode, the enhancement layer has the following CelpSpecificConfig():

```
class CelpSpecificConfig() {
        CelpBWSenhHeader();
}
```

Transmission of CELP bitstreams

Each scalable layer of an MPEG-4 CELP audio bitstream is transmitted in an Elementary Stream. In an SL-PDU payload, the following dynamic data for CELP Audio has to be included:

CELP Base Layer

```
slPduPayload {
        CelpBaseFrame();
}
```

CELP Enhancement Layer

To parse and decode the CELP enhancement layer, information decoded from the CELP base layer is required. For the bitrate scalable mode, the following data for the CELP enhancement layer has to be included:

```
slPduPayload {
        CelpBRSenhFrame();
}
```

For the bandwidth scalable mode, the following data for the CELP enhancement layer has to be included:

```
slPduPayload {
        CelpBWSenhFrame();
}
```

In case bitrate scalability and bandwidth scalability are both used simultaneously, first all bitrate enhancement layers have to be conveyed prior to the bandwidth scalability layer.

3.3.1 Header syntax

Table 3.9 – Syntax of CelpHeader()

Syntax	No. of bits	Mnemonic
CelpHeader (samplingFrequencyIndex)		
{		
ExcitationMode	**1**	**uimsbf**
SampleRateMode	**1**	**uimsbf**
FineRateControl	**1**	**uimsbf**
if (ExcitationMode==RPE) {		
RPE_Configuration	**3**	**uimsbf**
}		
if (ExcitationMode==MPE) {		
MPE_Configuration	**5**	**uimsbf**
NumEnhLayers	**2**	**uimsbf**
BandwidthScalabilityMode	**1**	**uimsbf**
}		
}		

Table 3.10 – Syntax of CelpBWSenhHeader()

Syntax	No. of bits	Mnemonic
CelpBWSenhHeader ()		
{		
BWS_configuration	**2**	**uimsbf**
}		

3.3.2 Frame syntax

Table 3.11 – Syntax of CelpBaseFrame()

Syntax	No. of bits	Mnemonic
CelpBaseFrame()		
{		
Celp_LPC()		
if (ExcitationMode==MPE) {		
MPE_frame()		
}		
if ((ExcitationMode==RPE)&&(SampleRateMode==16kHz)) {		
RPE_frame()		
}		
}		

Table 3.12 – Syntax of CelpBRSenhFrame()

Syntax	No. of bits	Mnemonic
CelpBRSenhFrame()		
{		
for (subframe=0; subframe<nrof_subframes; subframe++) {		
shape_enh_positions [subframe][enh_layer]	**4, 12**	**uimsbf**
shape_enh_signs [subframe][enh_layer]	**2, 4**	**uimsbf**
gain_enh_index [subframe][enh_layer]	**4**	**uimsbf**
}		
}		

Table 3.13 – Syntax of CelpBWSenhFrame()

Syntax	No. of bits	Mnemonic
CelpBWSenhFrame()		
{		
BandScalable_LSP()		
for (subframe=0; subframe<nrof_subframe_bws; subframe++) {		
shape_bws_delay [subframe]	**3**	**uimsbf**
shape_bws_positions [subframe]	**22, 26, 30, 32**	**uimsbf**
shape_bws_signs [subframe]	**6, 8, 10, 12**	**uimsbf**
gain_bws_index [subframe]	**11**	**uimsbf**
}		
}		

3.3.3 LPC syntax

Table 3.14 – Syntax of Celp_LPC()

Syntax	No. of bits	Mnemonic
Celp_LPC()		
{		
if (FineRateControl == ON){		
interpolation_flag	**1**	**uimsbf**
LPC_Present	**1**	**uimsbf**
if (LPC_Present == YES) {		
LSP_VQ()		
}		
} else {		
LSP_VQ()		
}		
}		

Table 3.15 – Syntax of LSP_VQ()

Syntax	No. of bits	Mnemonic
LSP_VQ()		
{		
if (SampleRateMode==8kHz) {		
NarrowBand_LSP()		
} else {		
WideBand_LSP()		
}		
}		

Table 3.16 – Syntax of NarrowBand_LSP()

Syntax	No. of bits	Mnemonic
NarrowBand_LSP()		
{		
lpc_indices [0]	**4**	**uimsbf**
lpc_indices [1]	**4**	**uimsbf**
lpc_indices [2]	**7**	**uimsbf**
lpc_indices [3]	**6**	**uimsbf**
lpc_indices [4]	**1**	**uimsbf**
}		

Table 3.17 – Syntax of BandScalable_LSP()

Syntax	No. of bits	Mnemonic
BandScalable_LSP()		
{		
lpc_indices [5]	**4**	**uimsbf**
lpc_indices [6]	**7**	**uimsbf**
lpc_indices [7]	**4**	**uimsbf**
lpc_indices [8]	**6**	**uimsbf**
lpc_indices [9]	**7**	**uimsbf**
lpc_indices [10]	**4**	**uimsbf**
}		

Table 3.18 – Syntax of WideBand_LSP()

Syntax	No. of bits	Mnemonic
WideBand_LSP()		
{		
lpc_indices [0]	**5**	**uimsbf**
lpc_indices [1]	**5**	**uimsbf**
lpc_indices [2]	**7**	**uimsbf**
lpc_indices [3]	**7**	**uimsbf**
lpc_indices [4]	**1**	**uimsbf**
lpc_indices [5]	**4**	**uimsbf**
lpc_indices [6]	**4**	**uimsbf**
lpc_indices [7]	**7**	**uimsbf**
lpc_indices [8]	**5**	**uimsbf**
lpc_indices [9]	**1**	**uimsbf**
}		

3.3.4 Excitation syntax

Table 3.19 – Syntax of RPE_frame()

Syntax	No. of bits	Mnemonic
RPE_frame()		
{		
for (subframe = 0; subframe < nrof_subframes; subframe++) {		
shape_delay [subframe]	**8**	**uimsbf**
shape_index [subframe]	**11,12**	**uimsbf**
gain_indices [0][subframe]	**6**	**uimsbf**
gain_indices [1][subframe]	**3,5**	**uimsbf**
}		
}		

Table 3.20 – Syntax of MPE_frame()

Syntax	No. of bits	Mnemonic
MPE_frame()		
{		
signal_mode	**2**	**uimsbf**
rms_index	**6**	**uimsbf**
for (subframe=0; subframe<nrof_subframes; subframe++) {		
shape_delay [subframe]	**8, 9**	**uimsbf**
shape_positions [subframe]	**14 ... 32**	**uimsbf**
shape_signs [subframe]	**3 ... 12**	**uimsbf**
gain_index [subframe]	**6, 7**	**uimsbf**
}		
}		

3.4 Semantics

This clause describes the semantics of the syntactic elements. Bitstream elements are shown in **bold**-face and the help variables that appear in the syntax and are needed to extract the bitstream elements are shown in *italics*.

ExcitationMode A one-bit identifier representing whether the Multi-Pulse Excitation tool or the Regular-Pulse Excitation tool is used.

Table 3.21 – Description of ExcitationMode

ExcitationMode	ExcitationID	Description
0	MPE	MPE tool is used
1	RPE	RPE tool is used

SampleRateMode A one-bit identifier representing the sampling rate. Two sampling rates are supported.

Table 3.22 – Description of SampleRateMode

SampleRateMode	SampleRateID	Description
0	8kHz	8 kHz Sampling rate
1	16kHz	16 kHz Sampling rate

FineRateControl A one-bit flag indicating whether fine rate control in very fine steps is enabled or disabled.

Table 3.23 – Description of FineRateControl

FineRateControl	RateControlID	Description
0	OFF	Fine-rate control is disabled
1	ON	Fine-rate control is enabled

FineRate Control enables the bitrate to be decreased with respect to its anchor bitrate. When transmitting the LPC parameters in every frame, the anchor bitrate will be obtained. The lowest bitrate possible bitrate for each configuration can be obtained by transmitting the LPC parameters in 50% of the frames.

RPE_Configuration This is a 3-bit identifier which configures the MPEG-4 CELP coder using the Regular-Pulse Excitation tool. This parameter directly determines the set of allowed bitrates (Table 3.24) and the number of subframes in a CELP frame (Table 3.25).

Table 3.24 – Rate allocation for the 16 kHz mode I coder

RPE_Configuration	Fixed bitrate FineRate Control OFF (bit/s)	Min. bitrate FineRate Control ON, 50% LPC (bit/s)	Max. bitrate FineRate Control ON, 100% LPC (bit/s)
0	14400	13000	14533
1	16000	13900	16200
2	18667	17267	18800
3	22533	21133	22667
4 ... 7	Reserved		

MPE_Configuration This is a 5-bit field that configures the MPEG-4 CELP coder using the Multi-Pulse Excitation tool. This parameter determines the variables *nrof_subframes* and *nrof_subframes_bws*. This parameter also specifies the number of bits for **shape_positions[**i**]**, **shape_signs[**i**]**, **shape_enh_positions[**i**][**j**]** and **shape_enh_signs[**i**][**j**]**.

nrof_subframes is a help parameter, specifying the number of subframes in a CELP frame, and is used to signal how many times the excitation parameters must be read. For the Regular-Pulse Excitation tool operating at the sampling rate of 16 kHz, this variable is dependent on the **RPE_Configuration** as follows:

Table 3.25 – Definition of nrof_subframes for the 16 kHz mode I coder

RPE_Configuration	nrof_subframes
0	6
1	4
2	8
3	10
4 ... 7	Reserved

For the Multi-Pulse Excitation tool, it is derived from the **MPE_Configuration** depending on the sampling rate as follows:

Table 3.26 – Definition of nrof_subframes for the 8 kHz mode II coder

MPE_Configuration	nrof_subframes
0,1,2	4
3,4,5	3
6 ... 12	2
13 ... 21	4
22 ... 26	2
27	4
28 ... 31	reserved

Table 3.27 – Definition of nrof_subframes for the 16 kHz mode II coder

MPE_Configuration	nrof_subframes
0 ... 6	4
8 ... 15	8
16 ... 22	2
24 ... 31	4
7, 23	reserved

NumEnhLayers This is a two-bit field specifying the number of enhancement layers that are used.

Table 3.28 – Definition of nrof_enh_layers

NumEnhLayers	nrof_enh_layers
0	0
1	1
2	2
3	3

BandwidthScalabilityMode This is a one-bit identifier that indicates whether bandwidth scalability is enabled. This mode is only valid when **ExcitationMode** = MPE.

Table 3.29 – Description of BandwidthScalabilityMode

BandwidthScalabilityMode	ScalableID	Description
0	OFF	Bandwidth scalability is disabled
1	ON	Bandwidth scalability is enabled

BWS_Configuration This is a two-bit field that configures the bandwidth extension tool. This identifier is only valid when **BandwidthScalabilityMode** = ON. This parameter specifies the number of bits for **shape_bws_positions[**i**]**, **shape_bws_signs[**i**]**.

nrof_subframes_bws This parameter, which is a help variable, represents the number of subframes in the bandwidth extension tool and is derived from the **MPE_Configuration** as follows:

Table 3.30 – Definition of nrof_subframes_bws

MPE_Configuration	nrof_subframes_bws
0,1,2	8
3,4,5	6
6 ... 12	4
13 ... 21	4
22 ... 26	2
27	not valid
28 ... 31	reserved

interpolation_flag This is a one-bit flag. When set, it indicates that the LPC parameters for the current frame must be derived using interpolation.

Table 3.31 – Description of interpolation_flag

interpolation_flag	InterpolationID	Description
0	OFF	LPC coefficients of the frame do not have to be interpolated
1	ON	LPC coefficients of the frame must be retrieved by interpolation

LPC_Present This bit indicates whether LPC parameters are attached to the current frame. These LPC parameters are either of the current frame or the next frame.

Table 3.32 – Description of LPC_Present

LPC_Present	LPCID	Description
0	NO	Frame does not carry LPC data
1	YES	Frame carries LPC data

Together, the **interpolation_flag** and the **LPC_Present** flag describe how the LPC parameters are to be derived.

Table 3.33 – LPC decoding process described by interpolation_flag and LPC_Present flag

interpolation_flag	LPC_Present	Description
1	1	LPC Parameters of the current frame must be extracted using interpolation. The current frame carries LPC Parameters belonging to the next frame.
1	0	RESERVED
0	1	LPC Parameters of the current frame are present in the current frame.
0	0	LPC Parameters of the previous frame must be used in the current frame.

lpc_indices[] These are multi-bit fields representing LPC coefficients. These contain information needed to extract the LSP coefficients. The exact extraction procedure is described in the Decoding Process.

shape_delay[subframe**]** This bit field represents the adaptive codebook lag. The decoding of this field depends on **ExcitationMode** and **SampleRateMode**.

Table 3.34 – Number of bits for shape_delay[]

ExcitationMode	SampleRateMode	shape_delay[] (bits)
RPE	16 kHz	8
MPE	8 kHz	8
MPE	16 kHz	9

shape_index[subframe**]** This index contains information needed to extract the fixed codebook contribution from the regular pulse codebook. The number of bits consumed by this field depends on the bitrate (derived from the **RPE_configuration**).

Table 3.35 – Number of bits for shape_index[]

RPE_Configuration	number of bits representing shape_index[]
0	11
1	11
2	12
3	12
4 ... 7	Reserved

gain_indices[0**][**subframe**]** These bit fields specify the adaptive codebook gain in the RPE tool using 6 bits. It is read from the bitstream for every subframe.

gain_indices[1**][**subframe**]** These bit fields specify the fixed codebook gain in the RPE tool. It is read from the bitstream for every subframe. The number of bits read to represent this field depends on the subframe number. For the first subframe this is 5 bits, while for the remaining subframes it is 3 bits.

gain_index[subframe**]** This 6 or 7-bit field represents the gains for the adaptive codebook and the multi-pulse excitation for the 8 kHz or 16 kHz sampling rate, respectively.

gain_enh_index[subframe**]** This 4-bit field represents the gain for the enhancement multi-pulse excitation in the CELP coder at 8 kHz.

gain_bws_index[subframe**]** This 11-bit field represents the gains for the adaptive codebook and two multi-pulse excitation in the bandwidth extension tool.

signal_mode This 2-bit field represents the type of signal. This information is used in the MPE tool. The gain codebooks are switched depending on this information.

Table 3.36 – Description of signal_mode

signal_mode	Description
0	Unvoiced
1,2,3	Voiced

rms_index This parameter indicates the rms level of the frame. This information is only utilized in the MPE tool.

shape_positions[subframe**], shape_signs[**subframe**]** These bit fields represent the pulse positions and the pulse signs for the multi-pulse excitation. The length of the bit field is dependent on **MPE_Configurations**.

Table 3.37 – Definitions of shape_positions[] and shape_signs[] for speech sampled at 8 kHz

MPE_Configuration	shape_positions[] (bits)	shape_signs[] (bits)
0	14	3
1	17	4
2	20	5
3	20	5
4	22	6
5	24	7
6	22	6
7	24	7
8	26	8
9	28	9

10	30	10
11	31	11
12	32	12
13	13	4
14	15	5
15	16	6
16	17	7
17	18	8
18	19	9
19	20	10
20	20	11
21	20	12
22	18	8
23	19	9
24	20	10
25	20	11
26	20	12
27	19	6
28 ... 31	reserved	

Table 3.38 – Definitions of shape_positions[] and shape_signs[] for speech sampled at 16 kHz

MPE_Configuration	shape_positions[] (bits)	shape_signs[] (bits)
0, 16	20	5
1, 17	22	6
2, 18	24	7
3, 19	26	8
4, 20	28	9
5, 21	30	10
6, 22	31	11
7, 23	reserved	reserved
8, 24	11	3
9, 25	13	4
10, 26	15	5
11, 27	16	6
12, 28	17	7
13, 29	18	8
14, 30	19	9
15, 31	20	10

shape_enh_positions[subframe**][], shape_enh_signs[**subframe**][]** These bit fields represent the pulse positions and the pulse signs for the multi-pulse excitation in each enhancement layer. The length of the bit field is dependent on **MPE_Configuration**.

Table 3.39 – Definition of shape_enh_positions[][] and shape_enh_signs[][] for speech sampled at 8 kHz

MPE_Configuration	shape_enh_positions[][] (bits)	shape_enh_signs[][] (bits)
0 ... 12	12	4
13 ... 26	4	2
27	not valid	
28 ... 31	reserved	

Table 3.40 – Definition of shape_enh_positions[][] and shape_enh_signs[][] for speech sampled at 16 kHz

MPE_Configuration	shape_enh_positions[][] (bits)	shape_enh_signs[][] (bits)
0 ... 6, 16 ... 22	12	4
8 ... 15, 24 ... 31	4	2
7, 23	reserved	

shape_bws_delay[subframe**]** This 3-bit field is utilized in decoding the adaptive codebook for the bandwidth extension tool. This value indicates the differential lag from the lag described in **shape_delay[]**.

shape_bws_positions[subframe**], shape_bws_signs[**subframe**]** These fields represent the pulse positions and the pulse signs for the multi-pulse excitation in the bandwidth extension tool. The length of the bit field is dependent on **BWS_Configuration**.

Table 3.41 – Definition of shape_bws_positions[] and shape_bws_signs[]

BWS_Configuration	shape_bws_positions[] (bits)	shape_bws_signs[] (bits)
0	22	6
1	26	8
2	30	10
3	32	12

3.5 MPEG-4 CELP Decoder tools

This clause provides a brief description of the functionality, parameter definition and the decoding processes of the tools supported by the MPEG-4 CELP core. The description of each tool comprises three parts and an optional part containing tables used by the tool:

1. Tool description: a short description of the functionality of the tool is given together with its interface.
2. Definitions: the input and output parameters as well as help elements of the tool are described here. Each element is either bold or italic. Bold names indicate that the element is read from the bitstream, italic names indicate auxiliary elements. If elements are already used by another tool, a reference to the previous definition is given.
3. Decoding process: The decoding process is explained here in detail with the aid of mathematical equations and pseudo-C code.
4. Tables: this optional fourth part contains tables that are used by the tool.

3.5.1 General Introduction to the MPEG-4 CELP decoder tool-set

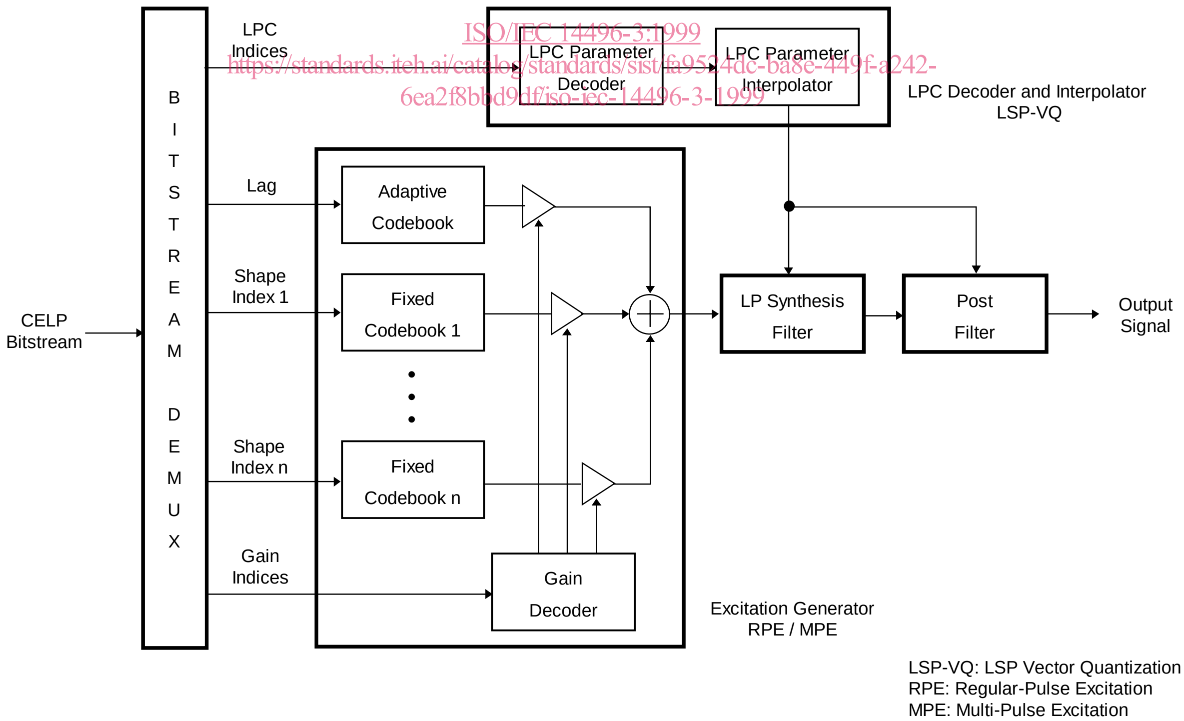

Figure 3.2 – Block diagram illustrating the tools used in the MPEG-4 CELP coder

Figure 3.2 illustrates the MPEG-4 CELP decoder. It operates at a sampling rate of 8 or 16 kHz. One or more individual blocks have been grouped together (outlined in bold), forming the tools available for MPEG-4 CELP decoding. The following tools are supported: