Dolby TrueHD (MLP)
high-level bitstream description

7 February 2018

Copyright

For information, contact:

Dolby Laboratories, Inc.
1275 Market Street
San Francisco, CA 94103-1410 USA
Telephone 415-558-0200
Fax 415-863-1373
http://www.dolby.com

Trademarks

Dolby and the double-D symbol are registered trademarks of Dolby Laboratories. Following are trademarks of Dolby Laboratories:

- Dolby®
- Dolby Atmos®
- Dolby Audio™
- Dolby Cinema™
- Dolby Theatre®
- Dolby Vision™
- Dolby Voice®
- Feel Every Dimension in Dolby™
- Feel Every Dimension™
- Feel Every Dimension in Dolby Atmos™
- Dolby Digital Plus™
- Dolby Advanced Audio™
- Dolby Home Theater®
- Dialogue Intelligence™
- Dolby Digital Plus Home Theater™
- MLP Lossless™
- Pro Logic®
- Surround EX™
- Dolby Digital Plus Advanced Audio™
- Dolby Fidelio™
- Dolby AccessLink™
- Dolby CaptiView™
- Dolby CineAsset™
- Dolby CineAsset Player™

All other trademarks remain the property of their respective owners.

Contents

1 Introduction

This document provides a high-level description of the bitstream format used by the Dolby® TrueHD (MLP) lossless audio coding system.

1.1 Overview

Dolby TrueHD is a lossless coding system for use on high-quality digital audio data originally represented as linear PCM. It has the following features:

- Good compression of both peak and average data rates.
- Easy transcoding between fixed-rate and variable-rate data streams.
- Automatic savings on signals that do not use all of the available bandwidth, for example, low frequency effects.
- Modest decoding requirements.

The reduction in peak data rate is equivalent to reducing the wordwidth by 4 bits or more with 48-kHz-sampled signals, or by 8 bits with 96-kHz-sampled signals. Thus 24-bit 96-kHz audio is effectively compressed to 16 bits, making it possible to deliver 6 channels of 24-bit 96-kHz audio within a 9.6-Mbps stream and 8 channels of 24-bit 96-kHz audio within a 13.8-Mbps stream.

Dolby TrueHD is flexible in its multichannel operation. It contains provision for low-cost and portable applications whereby a 2-channel decoder with small MIPS and memory requirements can recover an $\{L_0, R_0\}$ mix from a multichannel bitstream.

The Dolby TrueHD bitstream can be represented in different ways according to the carrier, transport or storage format.

Dolby TrueHD allows transmission and storage of up to 32 channels, each with precision of up to 24 bits, at sampling rates of between 44.1 kHz and 192 kHz.

1.2 About this documentation

This document:

- Is intended for implementers who need to parse a Dolby TrueHD bitstream to determine basic configuration parameters that are useful for processes such as transport format multiplexing or packaging.
- Describes the syntax and semantics of the Dolby TrueHD bitstream syntax used for Blu-ray Disc applications, referred to in this document as the FBA syntax. The FBB syntax as used in DVD-Audio applications is not described.
- Does not address details of the core audio encoding and decoding algorithm, and describes only the bitstream parameters needed to parse the bitstream structure and understand the high-level configuration of the bitstream and audio presentations contained therein.

1.3 Resources

Table 1 lists documents that supplement the information in this document.

Table 1: Supplemental Resources

Document/Specification
Recommendation ITU-R BS.2051-1: Advanced sound system for programme production
SMPTE 428-3-2006: "D-Cinema Distribution Master - Audio Channel Mapping and Channel Labeling".

1.4 Channel abbreviations

This table lists the channel notations used in this document. All references to ITU channel names are with regard to Recommendation ITU-R BS.2051-1.

Channel name	Abbreviation	ITU R BS.2051 name
Left	L	FLc
Right	R	FRc
Centre	C	FC
Low-Frequency Effects	LFE	LFE1
Left Surround	Ls	SiL
Right Surround	Rs	SiR
Left Back	Lb	BL
Right Back	Rb	BR
Top Front Left	Tfl	TpFL
Top Front Right	Tfr	TpFR
Top Back Left	Tbl	TpBL
Top Back Right	Tbr	TpBR
Top Side Left	Tsl	TpSiL
Top Side Right	Tsr	TpSiR
Top Front Center	Tfc	TpFC
Top Centre	Tc	TpC
Low-Frequency Effects 2	LFE2	LFE2
Back Centre	Cb	BC
Left Wide	Lw	FL
Right Wide	Rw	FR
Left Screen	Lsc	Lsc
Right Screen	Rsc	Rsc
Left Surround Direct	Lsd	Not specified
Right Surround Direct	Rsd	Not specified

Note: Left Surround Direct and Right Surround Direct are specified in SMPTE 428-3-2006: "D-Cinema Distribution Master - Audio Channel Mapping and Channel Labeling.

This table lists terms and definitions for stereo and mono audio.

Stereo	L, R speaker feeds
Lt, Rt	Signals encoded for matrix-surround decoding, but also usable directly as L, R speaker feeds
Lbin, Rbin	Left and Right signals encoded for binaural (headphone) playback
Mono	A single channel signal. When carried in a stereo presentation, the signal is encoded identically in the two channels for use as L, R speaker feeds

1.5 Contacting Dolby

Support services are available to address any questions and to provide advice about implementing Dolby technology.

For any questions regarding the design, porting, or testing of an Implementation, contact Dolby at implementationsupport@dolby.com. By utilizing Dolby expertise, especially during the design process, many problems that might require design revisions before an Implementation is fully approved can be prevented.

If you have comments or feedback about this document, send us an email at documentation@dolby.com.

2 Bitstream organization

2.1 External and internal structure

The Dolby TrueHD bit stream has two levels of organization, external and internal. Its external organization is designed so that an external system can handle it easily and its internal structure is designed to code audio efficiently.

The external structure is based on access units and MLP Syncs. The internal structure is based on blocks, some of which contain restart headers.

2.2 Basic definitions

The audio stream to be encoded is divided into a series of audio frames. Each audio frame contains the sampled data for one presentation unit. The duration of a presentation unit is 1/1200 second for audio sampled at multiples of 48 kHz and 1/1102.5 second for audio sampled at multiples of 44.1 kHz.

An audio frame will therefore, for example, have a duration of 40 multichannel samples at 48 kHz, 80 multichannel samples at 96 kHz and 160 multichannel samples at 192 kHz.

2.3 External organization: Access Units and MLP Syncs

An **access unit** contains an **MLP Sync** followed by a segment from each substream.

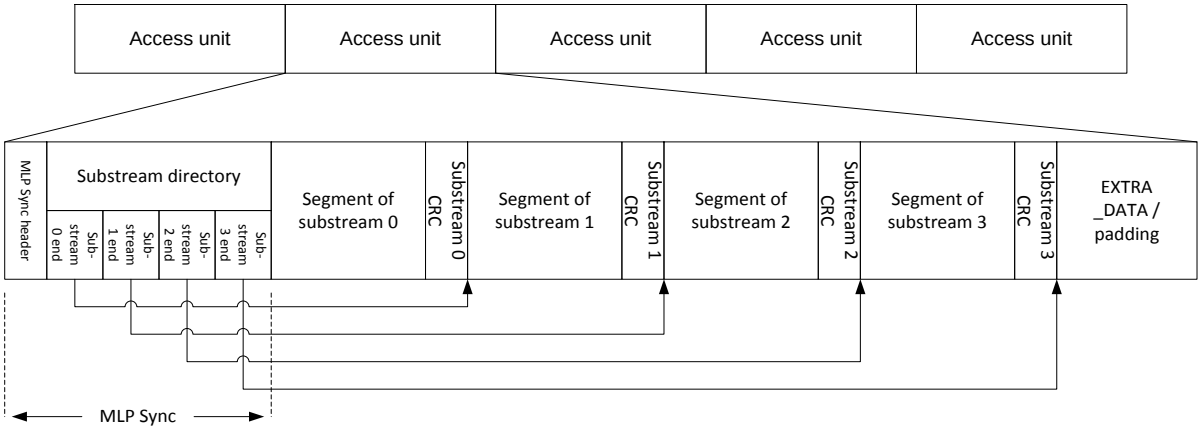

Figure 1: A section of a Dolby TrueHD bit stream

In Figure 1, the second access unit is expanded to show its internal structure, including segments of 3 substreams.

There are two types of MLP Sync: the major sync and the minor sync. Each sync contains a header and some directory information, as shown in Figure 1. The major sync has an expanded header containing all the information required to start full decoding of the stream, whereas a minor sync's header consists of no more than size and time stamps. To minimize data rate overheads, most access units are introduced by minor syncs, major syncs occurring typically once per 128 access units.

Any padding required to pad the stream to a fixed data rate is placed at the end of each access unit.

2.4 Internal organization: Blocks and Restart Blocks

A substream consists of a sequence of blocks. A block may start with a block header (block_header). This may optionally be preceded by a restart header (restart_header), in which case the block is called a **restart block** and its beginning is a **restart point**.

A block contains the encoded data for a number of audio samples of the channels conveyed in one substream. A block contains the data for at least eight samples and for at most one audio frame.

Restart points are the points at which decoding can be started, or restarted after an error. The restart header includes relevant initialization information for the decoder.

Block headers permit selective updating of the decoding parameters so that an encoder can optimize its compression characteristics in response to changes in signal statistics.

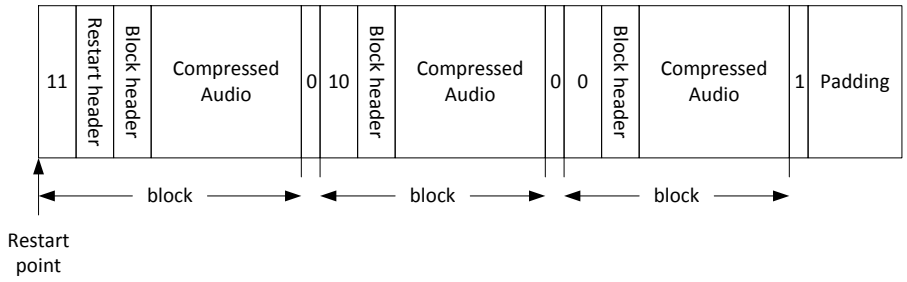

Figure 2: Substream segment

Figure 2 illustrates a substream segment, including a restart block, a block with a block header and a block without a block header. These possibilities are signaled by the bit patterns '11', '10' and '0', as shown in the block() syntax. A '1' bit after a block indicates that it is the last block in a substream segment and is followed by padding.

Related information

block() syntax on page 18

2.5 Relationship of external to internal organization

For current media delivery applications, the internal structure of MLP is synchronous both with the external structure and with the audio frames.

Each audio frame is encoded as between one and four complete blocks. These blocks are concatenated and padded to a multiple of 16 bits to produce a substream segment (see Figure 2) that is inserted into an access unit (see Figure 1). Each access unit thus contains the encoded data for one audio frame. The last block in a substream segment is followed by padding to a 16-bit boundary.

Most of the access units start with minor syncs. Major syncs occur at intervals of no fewer than eight access units and no more than 128 access units.

A substream segment starts with a block containing a restart header if and only if it is contained in an access unit that begins with a major sync. Consequently, restart points are synchronous between substreams and occur at intervals of no fewer than eight access units, except at the start of a Dolby TrueHD bitstream where the first two access units are permitted to contain major syncs, and consequently the substream segments in these first two access units start with a block containing a restart header.

2.6 Data rate smoothing using FIFO buffering

Lossless compression of audio data results in a bitstream that has an inherently variable data rate. Easy-to-compress sections of audio have a low data rate, and hard-to-compress sections will have a higher data rate. Dolby TrueHD uses a FIFO buffer synchronized between the encoder and decoder to minimize the differences between the lowest and highest data rates of the bitstream.

To achieve this, easy-to-compress sections of audio are delivered to the decoder faster than they are needed, pre-caching the FIFO buffer with audio data. This means that when a difficult-to-compress section of audio is encountered, the data can be delivered to the decoder more slowly than the inherent peak data rate of the data, as the FIFO buffer empties of the pre-cached data.

Figure 3 shows FIFO buffering in the stream. In this case each access unit decodes to one presentation unit. At the beginning and the end of the section the audio is easy to compress and the access units are small enough to be read from the delivery format during one presentation unit. Here the delay through the FIFO is one presentation unit or less.

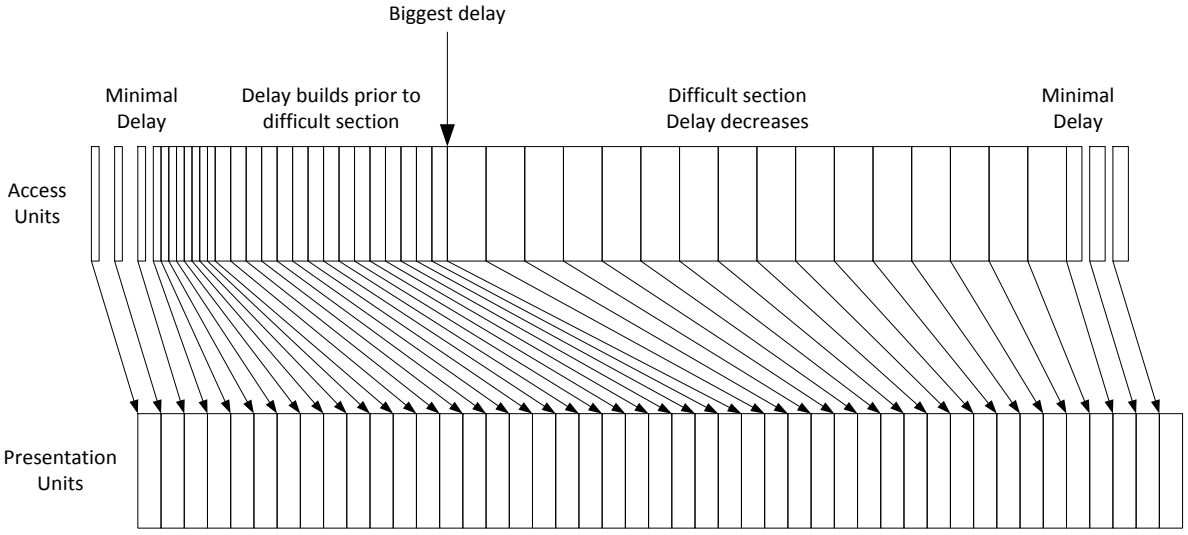

Figure 3: FIFO operation in MLP

Note: The varying sizes of each access unit is represented by the width of each access unit. The FIFO delay between receipt of access units and output of presentation units is represented by the deviation of the arrow from vertical.

In the difficult section the access units are spaced out in order not to violate any constraint on peak data rate. As an access unit cannot arrive after its presentation unit has been delivered, the access units for the earlier parts of the difficult section are advanced in time. FIFO delay is thus at its greatest at the beginning of a section in which every access unit is too big to be delivered during one presentation unit.

2.7 FIFO timing and management

Here we present a base model for FIFO timing and management. Different implementation strategies may be used.

Each access unit is transferred instantaneously at a time given by its input_timing field. Each substream within a major sync access unit begins with a restart header that contains an output_timing field. For current applications of Dolby TrueHD, output_timing are the same for all substreams. When the time equals output_timing, the amount of data needed to decode one audio frame is removed from the FIFO.

Subsequent access units are dealt with in the same way. Each is transferred to the FIFO buffer at its input_timing. Access units with minor syncs do not contain restart headers and so do not have output_timing fields, but the output timing can be deduced by incrementing a counter by one audio frame worth of audio samples.

It is the encoder's responsibility to ensure that the FIFO buffer does not under-run or over-run. For the applications of Dolby TrueHD supported by this document, the minimum required FIFO buffer size is 120,000 bytes.

2.7.1 Peak data rate limitation

The effective data rate from the delivery format is defined as

rate = **size[n]**/(**input_timing[n+1]** - **input_timing[n]**)

where

rate is the data rate in bits per sample.

size[n] is the number of bits in the **nth** access unit.

input_timing is the input timing of the access unit, in samples (the unwrapped values are used, not the values in the stream, which are wrapped to 16 bits).

It is the encoder's responsibility to ensure that **rate** does not exceed **peakDataRate** where **peakDataRate** = 18 Mbit/s for FBA streams. Note that this condition does not apply to the last access unit of the bitstream.

2.8 Error checking and recovery

Multiple-level error checking is supported by including check words both at the access unit level and at the substream level. Parsers may omit some of the checks in the interests of economy. In particular, CRC computations on audio data, which can become expensive, may be omitted and reliance placed on the simpler parity checks that are also included within the stream. Although some errors may go undetected in this case, there is provision for a decoder to ensure that these errors cannot produce 'bangs' substantially above the current level of the audio.

The error checking, protection and recovery provisions include:

- A CRC on each major sync.
- A CRC *and* a parity check on each segment of audio data carried in an access unit.
- Within the substream, a CRC on each restart header and block header.

3 Bitstream syntax

3.1 Access Unit

A Dolby TrueHD bit stream consists of a sequence of **access units**. An **access unit** consists of an MLP Sync (from label **mlp_sync** to label **start** in the syntax below) followed by a segment of each substream. A **major sync** is the same as a **minor sync**, but with additional information contained in the substructure major_sync_info(). Major syncs can be distinguished by the fact that the bitfield of 32 bits starting at bit offset 32 from the start of the access unit always equals `0xF8726FBA`, whereas the corresponding bitfield of a minor sync can never take this value.

The substream_directory contains end pointers for all the substream segments.

Each expansion of substream_segment() provides the substream segment for one substream. For example, if there are two substreams there are two such expansions, referring to substreams 0 and 1 respectively. If there are three substreams there are three such expansions, referring to substreams 0, 1 and 2 respectively.

3.2 Syntax description language

The notation of the bit stream syntax is derived mainly from **C**.

Entries in bold, for example

```
check_nibble                                    v(4)            4.1.1
```

signify the occurrence of the corresponding variable in the bit stream. The second column of each syntax section specifies the encoding (according to the notation in section 3.2.1) and the third column gives (where applicable) a reference to a section containing more information about the variable.

Substructures are defined using the **C** procedure syntax:

```
name(<optional arguments>) {
   .
   .
}
```

The substructure is expanded by the C call syntax name(<arguments>).

Iteration is expressed using the usual **C** for, while and do...while statements. Conditional expansion uses the if (. . .) ... and if (. . .) ... else ... constructs.

The frequently required procedure of reading a flag from the stream and performing an <action> if the flag is TRUE (binary '**1**') is expressed as

```
if (flag)                                       b(1)
   <action>
```

which can be regarded as shorthand for

```
flag                                            b(1)
if (flag) {
   <action>
}
```

Flags in while statements obey similar rules.

Labels are flush against the left margin and are indicated as comments in the code; for example

```
/* start */
```

They are regarded as markers within the expanded bit stream, not as textual markers within the syntax description. When used as values in formulae, they represent the offset (in 16-bit words) from the start of the access unit.

3.2.1 Bitfield encoding

The encoded data can be thought of as a serial bit stream containing items of data that occupy fields of varying width, e.g. three bits to specify a number of channels in the range [0...7]. The bits occupying each field are presented most significant bit first.

Because fields can be of variable length, they can in general straddle word boundaries. MLP Sync structures and restart points are, however always aligned with 16-bit word boundaries.

Values are encoded in signed or unsigned bitfields and are referred to in the syntax description using the following notation:

b(1) Boolean 1-bit: 1 = `TRUE`, 0 = `FALSE`

v(**n**) Bitfield taking **n** bits, with arbitrary representation

s(**n**) Signed integer taking **n** bits (**n** ≥ 1), represented in 2's complement

u(**n**) Unsigned integer taking **n** bits (**n** ≥ 0, u(0)=0)

pad(**n**) Pad with **n** zero bits

reserved bitfields are encoded as v(n) and should be ignored by parsers.

3.3 Syntax specification

3.3.1 access_unit()

Syntax	Encoding	Section
access_unit() {		
/* mlp_sync */		
check_nibble	v(4)	4.1.1
access_unit_length	u(12)	4.1.2
input_timing	u(16)	4.1.3
if(major_sync) {		
major_sync_info()		4.2
}		
/* substream_directory */		
for (i = 0; i < substreams; i++) {		
extra_substream_word[i]	b(1)	4.5.1
restart_nonexistent[i]	b(1)	4.5.2
crc_present[i]	b(1)	4.5.3
reserved	v(1)	
substream_end_ptr[i]	u(12)	4.5.4
if(extra_substream_word[i]) {		
/* dynamic_range_control */		
drc_gain_update[i]	s(9)	
drc_time_update[i]	u(3)	
reserved[i]	v(4)	
}		
}		
/* start */		
for (i = 0; i < substreams; i++) {		
/* substream_start */		
substream_segment(i)		4.6
if (crc_present[i]) {		
substream_parity[i]	u(8)	4.6.6
substream_CRC[i]	u(8)	4.6.7
}		
} /* substream_end[i] */		
/* extra_start */		
EXTRA_DATA()		4.8
/* unit_end */		
}		

3.3.2 major_sync_info()

Syntax	Encoding	Section
major_sync_info() {		
format_sync	v(32)	4.2.1
format_info	v(32)	4.2.2
signature	v(16)	4.2.3
flags	v(16)	4.2.4
reserved	v(16)	
variable_rate	b(1)	4.2.5
peak_data_rate	u(15)	4.2.6
substreams	u(4)	4.2.7
reserved	v(2)	
extended_substream_info	u(2)	4.2.8
substream_info	v(8)	4.2.9
channel_meaning()		4.3
major_sync_info_CRC	u(16)	4.2.10
}		

3.3.3 channel_meaning()

Syntax	Encoding	Section
channel_meaning() {		
reserved	v(6)	
2ch_control_enabled	b(1)	
6ch_control_enabled	b(1)	
8ch_control_enabled	b(1)	
reserved	v(1)	
drc_start_up_gain	s(7)	
2ch_dialogue_norm	u(6)	4.3.1
2ch_mix_level	u(6)	4.3.2
6ch_dialogue_norm	u(5)	4.3.3
6ch_mix_level	u(6)	4.3.4
6ch_source_format	v(5)	4.3.5
8ch_dialogue_norm	u(5)	4.3.6
8ch_mix_level	u(6)	4.3.7
8ch_source_format	v(6)	4.3.8
reserved	v(1)	
extra_channel_meaning_present	b(1)	4.3.9
if (extra_channel_meaning_present) {		
extra_channel_meaning_length	u(4)	4.3.10
extra_channel_meaning_data()		3.3.4
padding	pad(0…15)	
}		
}		

3.3.4 extra_channel_meaning_data()

Syntax	Encoding	Section
extra_channel_meaning_data() {		
if substream_info & 0x80 {		
16ch_channel_meaning()		3.3.5
reserved	v(((extra_channel_meaning_length+1)*16) - (16ch_channel_meaning_length_in_bits) - 4)	
}		
else {		
reserved	v(((extra_channel_meaning_length+1)*16) - 4)	
}		
}		

3.3.5 16ch_channel_meaning()

Syntax	Encoding	Section
16ch_channel_meaning() {		
16ch_dialogue_norm	u(5)	4.4.1
16ch_mix_level	u(6)	4.4.2
16ch_channel_count	u(5)	4.4.3
dyn_object_only	b(1)	4.4.4
if (dyn_object_only) {		
lfe_present	b(1)	4.4.5
}		
else {		
16ch_content_description	v(4)	4.4.6
if (16ch_content_description & 0x1) {		
chan_distribute	b(1)	
reserved	b(1)	
lfe_only	b(1)	4.4.7
if (lfe_only == 0) {		
reserved	b(1)	set to ‘1’
16ch_channel_assignment	v(10)	4.4.8
}		
}		
if (16ch_content_description & 0x2) {		
16ch_intermediate_spatial_format	v(3)	4.4.9
}		
if (16ch_content_description & 0x4) {		
16ch_dynamic_object_count	u(5)	4.4.10
}		
}		
}		

3.3.6 substream_segment()

Syntax	Encoding	Section
substream_segment() {		
do {		
block()		4.7
}		
while (**!last_block_in_segment**)	b(1)	4.6.1
padding	pad(0 … 15)	
if (last_access_unit_in_stream) {		
terminatorA	v(18)	4.6.1
zero_samples_indicated	b(1)	4.6.3
if (zero_samples_indicated) {		
zero_samples	u(13)	4.6.4
}		
else {		
terminatorB	v(13)	4.6.5
}		
}		
}		

3.3.7 block()

Syntax	Encoding	Section
block() {		
if (**block_header_exists**) {	b(1)	
if (**restart_header_exists**) {	b(1)	4.7.1
restart_header()		4.7.2
}		
block_header()		
}		
if (error_protect) {		
block_data_bits	u(16)	
}		
block_data()		
if (error_protect) {		
block_header_CRC	u(8)	
}		
}		

3.3.8 restart_header()

Syntax	Encoding	Section
restart_header() {		
restart_sync_word	v(14)	4.7.2
output_timing	u(16)	4.7.2
min_chan	u(4)	4.7.2
max_chan	u(4)	4.7.2
max_matrix_chan	u(4)	4.7.2
dither_shift	u(4)	
dither_seed	u(23)	
max_shift	s(4)	
max_lsbs	u(5)	
max_bits	u(5)	
max_bits	u(5)	
error_protect	b(1)	4.7.2
lossless_check	u(8)	
reserved	v(16)	
for (ch = 0; ch ≤ max_matrix_chan; ch++) {		
ch_assign[ch]	u(6)	
}		
restart_header_CRC	u(8)	4.7.2
}		

3.3.9 EXTRA_DATA()

Syntax	Encoding	Section
EXTRA_DATA() {		
length_check_nibble	v(4)	4.8.2
EXTRA_DATA_length	u(12)	4.8.3
data()		
EXTRA_DATA_padding	pad(variable)	4.8.4
EXTRA_DATA_parity	v(8)	4.8.5
}		

4 Description of bit stream elements

In the lists below, each variable is followed by its encoding, as given in the bit stream syntax.

4.1 MLP Sync

4.1.1 check_nibble - v(4)

The **check_nibble** field contains a 4-bit check sum. In a minor sync, the value of the **check_nibble** field is calculated so that the exclusive-OR of all the 4-bit nibbles in the minor sync is 0xF. In a major sync the calculation shall include only the items that also appear in a minor sync, i.e. the calculation shall exclude the expansion of major_sync_info(), which is protected by its own CRC.

4.1.2 access_unit_length - u(12)

The 12-bit **access_unit_length** field shall indicate the total length of the complete access unit, expressed in 16-bit words.

4.1.3 input_timing - u(16)

The 16-bit **input_timing** field shall indicate the time at which the access unit is input to the decoder, expressed in sample periods and modulo 65536.

Related Information

FIFO timing and management on page 11.

4.2 major_sync_info

4.2.1 format_sync - v(32)

The 32-bit **format_sync** field is a synchronization word that is provided close to the start of a major sync so that the major sync can be recognized without additional navigation information.

The value of the **format_sync** field shall be 0xF8726FBA (FBA bit stream syntax).

4.2.2 format_info - v(32)

The 32-bit **format_info** field defines the channel usage and sampling rate of the bit stream, and is interpreted as shown in this table.

Table 2: Meaning of format_info

format_info field	Number of bits	Number of bytes
audio_sampling_frequency	4	1
6ch_multichannel_type	1	
8ch_multichannel_type	1	
reserved	2	
2ch_presentation_channel_modifier	2	3
6ch_presentation_channel_modifier	2	
6ch_presentation_channel_assignment	5	
8ch_presentation_channel_modifier	2	
8ch_presentation_channel_assignment	13	

The syntax and semantics of each parameter are defined in and shall be consistent with the information carried in channel_meaning(). The most significant bit of the **format_info** field (bit 31) is the most significant bit of the audio_sampling_frequency field.

audio_sampling_frequency - u(4)

The 4-bit audio_sampling_frequency field specifies the sampling rate of the bit stream. The field is interpreted as shown in this table.

Table 3: Meaning of audio_sampling_frequency

Value	Sampling Rate
0000	48 kHz
0001	96 kHz
0010	192 kHz
1000	44.1 kHz
1001	88.2 kHz
1010	176.4 kHz
Others	reserved

2ch_presentation_channel_modifier - v(2)

The **2ch_presentation_channel_modifier** field specifies the channel assignments and content type of the 2-channel presentation. The field is interpreted as shown in this table.

Table 4: 2-channel presentation content type

2ch_presentation_channel_modifier bits 1-0	Content type
00	Stereo
01	Lt/Rt
10	Lbin/Rbin
11	Mono

6ch_multichannel_type - b(1)

The 1-bit **6ch_multichannel_type** field describes the multi-channel structure of the 6-ch presentation. The value of this field is set to '0' indicating a standard loudspeaker layout. A value of '1' has a reserved meaning.

6ch_presentation_channel_modifier - v(2)

The **6ch_presentation_channel_modifier** field specifies the channel assignments and content type of the 6-channel presentation. The field is interpreted as shown in the two following tables.

Table 5: 6-ch presentation content type for 2-channel content

6ch_presentation_channel_modifier bits 1-0	Content type
00	Stereo
01	Lt/Rt
10	Lbin/Rbin
11	Mono

Note: The 6ch_presentation_channel_modifier field specifies the channel assignments and content type of the 6-channel presentation. The field is interpreted as shown in the two following tables.

Table 5 shall only apply when the 6-channel presentation contains exactly two channels.

Table 6: 6-ch presentation content type for content containing Surrounds (Ls/Rs)

6ch_presentation_channel_ modifier bits 1-0	Application
00	Not indicated
01	Not Dolby Surround EX, Dolby Pro Logic IIx or Dolby Pro Logic IIz encoded
10	Dolby Surround EX / Dolby Pro Logic IIx encoded
11	Dolby Pro Logic IIz encoded

Note: When there are three or more encoded channels, the **6ch_presentation_channel_modifier** shall only have meaning if the Surround (Ls/Rs) channels are present, in which case Table 6 shall apply.

6ch_presentation_channel_assignment - v(5)

The presence or absence of channels is flagged in the bit field **6ch_presentation_channel_assignment**. Some loudspeakers are added singly and others as pairs. The field is interpreted as shown in this table.

Table 7: Channel assignments for the 6-channel presentation

6ch_presentation_ channel_assignment bit	Channels present if bit = 1 Note 1, 2	Number of channels Note 2
0	L/R	2
1	C	1
2	LFE	1
3	Ls/Rs	2
4	Tfl/Tfr	2

Note 1: When bit = 1 the channel(s) exist(s); when bit = 0 there is no assignment.

Note 2: Bit allocations shall not cause the number of channels allocated to exceed six.

8ch_multichannel_type - b(1)

The 1-bit **8ch_multichannel_type** field describes the multi-channel structure of the 8-ch presentation. The value of this field is set to '0' indicating a standard loudspeaker layout. A value of '1' has a reserved meaning.

8ch_presentation_channel_modifier - v(2)

The **8ch_presentation_channel_modifier** field specifies the channel assignments and content type of the 8-channel presentation. The field is interpreted as shown in the two following tables.

Table 8: Channel applications for 2-channel content

8ch_presentation_ channel_modifier bits 1-0	Application
00	Stereo
01	Lt/Rt
10	Lbin/Rbin
11	Mono

Note: The 8ch_presentation_channel_modifier field specifies the channel assignments and content type of the 8-channel presentation. The field is interpreted as shown in the two following tables.

Table 8 shall only apply when the presentation contains exactly two channels with a channel assignment of Main (Left/Right).

Table 9: Channel applications for content containing Surrounds (Ls/Rs) and no other surrounds

8ch_presentation_ channel_modifier bits 1-0	Application
00	Not indicated
01	Not Dolby Surround EX, Dolby Pro Logic IIx or Dolby Pro Logic IIz encoded
10	Dolby Surround EX / Dolby Pro Logic IIx encoded
11	Dolby Pro Logic IIz encoded

Note: When there are three or more encoded channels the 8ch_presentation_channel_modifier shall only have meaning if the Surround (Ls/Rs) channels are the only surround channels present, in which case Table 9 shall apply.

8ch_presentation_channel_assignment - v(13)

The channel assignment of the 8-channel presentation is described using the 13-bit **8ch_presentation_channel_assignment field**. Some loudspeakers are assigned singly and others as pairs. The field is interpreted in one of two ways, depending on the value of bit 11 of the flags field, as shown in the two following tables.

Table 10: Channel assignments for the 8-channel presentation when bit 11 of flags = 0

8ch_presentation_ channel_assignment bit	Exists Note 1	Number of channels Note 2
0	L/R	2
1	C	1
2	LFE	1
3	Ls/Rs	2
4	Tfl/Tfr	2
5	Lsc/Rsc	2
6	Lb/Rb	2
7	Cb	1
8	Tc	1
9	Lsd/Rsd	2
10	Lw/Rw	2
11	Tfc	1
12	LFE2	1

Table 11: Channel assignments for the 8-channel presentation when bit 11 of flags = 1

8ch_presentation_ channel_assignment bit	Exists Note 1	Number of channels Note 2
0	L/R	2
1	C	1
2	LFE	1
3	Ls/Rs	2
4	Tsl/Tsr	2
5 - 12	Reserved	N/A

Note 1: When bit = 1 the channel(s) exist(s); when bit = 0 there is no assignment.

Note 2: Bit allocations shall not cause the number of channels allocated to exceed eight.

Related Information

channel_meaning on page 26

4.2.3 signature - v(16)

The 16-bit **signature** field provides further confirmation of a valid major sync. The value of the **signature** field is 0xB752.

4.2.4 flags - v(16)

The 16-bit **flags** field is a bit field that is used to provide high-level information about the Dolby TrueHD bit stream configuration and content. The bit assignment of the **flags** field is shown in this table.

Table 12: Bit assignment and meaning of flags field

Bit number	Meaning when bit = 1	Restrictions
15 (MSB)	The FIFO delay is constant for the entire duration of the audio bit stream.	
14 to 12	reserved	
11	8ch_presentation_channel_assignment is interpreted according to Table 11.	May be set to '1' only when there are more than six channels in the 8-channel presentation.
10 to 0	reserved	

4.2.5 variable_rate - b(1)

If the data rate of the Dolby TrueHD bit stream is constant for the duration of the bit stream, the 1-bit **variable_rate** field is set to '0', otherwise it is set to '1'.

4.2.6 peak_data_rate - u(15)

The 15-bit **peak_data_rate** field shall specify the maximum data rate of the bit stream, measured over the entire duration of the bit stream, in units of $^1/_{16}$ bit per sample period. In a fixed-rate stream, **peak_data_rate** shall specify the constant data rate in units of $^1/_{16}$ bit per sample period.

4.2.7 substreams - u(4)

The 4-bit **substreams** field shall indicate the number of substreams syntactically present within the bit stream. Parsers should use the value of **substreams** to navigate within the access_unit().

4.2.8 extended_substream_info - u(2)

The 2-bit **extended_substream_info** field is used in combination with the value of the substream_info field to determine which substream or substreams carry the 16-channel presentation.

Table 13: Meaning of extended_substream_info

extended_substream_info value	Meaning
00	16-channel presentation is carried in substream 3
01	16-channel presentation is carried in substreams 2 and 3
10	16-channel presentation is carried in substreams 1, 2 and 3
11	16-channel presentation is carried in substreams 0, 1, 2 and 3

4.2.9 substream_info - v(8)

The 8-bit **substream_info** field indicates which substream or substreams carry the 2, 6 and 8-channel presentations. Bit 7 of the **substream_info** field is the most significant bit of the field.

- Bits 0 and 1 are reserved and both bits are set to '0'.
- 2-channel presentation: The 2-channel presentation is carried in substream 0. The number of channels is either 1 or 2.
- 6-channel presentation: The 6-channel presentation is either a copy of the 2-channel presentation, or is carried in substream 1 or a combination of substreams 0 and 1, according to the values of bits 2 and 3 of substream_info as specified in this table.

Table 14: Meaning of bits 3-2 of substream_info

Bits 3-2	Meaning
00	Illegal
01	6-ch presentation is a copy of the 2-ch presentation carried in substream 0
10	6-ch presentation is carried in substream 1
11	6-ch presentation is carried in substreams 0 and 1

- 8-channel presentation: The 8-channel presentation is either a copy of the 2-channel presentation, or is a copy of the 6-channel presentation, or is carried in substream 2 or a combination of substreams 0, 1 and/or 2 according to the values of bits 6-4 of substream_info, as specified in this table.

Table 15: Meaning of bits 6-4 of substream_info

Bits 6-4	Meaning
000	Illegal
001	8-ch presentation is a copy of the 2-ch presentation carried in substream 0
010	8-ch presentation is a copy of the 6-ch presentation carried in substream 1
011	8-ch presentation is a copy of the 6-ch presentation carried in substreams 0 and 1
100	8-ch presentation is carried in substream 2
101	8-ch presentation is carried in substreams 0 and 2
110	8-ch presentation is carried in substreams 1 and 2
111	8-ch presentation is carried in substreams 0, 1 and 2

Bit 7 of **substream_info** is used to indicate whether the 16-channel presentation is present in the bitstream. If bit 7 is set to '0', no 16-channel presentation exists in the bitstream.

If bit 7 is set to '1', the extended_substream_info field specifies which substream(s) carry the 16-ch presentation, and channel meaning data for the 16-channel presentation is present in extra_channel_meaning (as specified in 16ch_channel_meaning).

For bitstreams that contain more than 3 substreams, the range of permitted values for the **substream_info** field are restricted by the value of the extended_substream_info field. These restrictions are as defined in this table.

Table 16: Permitted values of substream_info for extended_substream_info field values of 00 to 11

extended_substream_info field value	Permitted values for bits 6-4 of substream_info	Permitted value for bits 2-3 of substream_info
00	100-111	01-11
01	100	10, 11
10	110	10
11	111	11

4.2.10 major_sync_info_CRC - u(16)

The 16-bit **major_sync_info_CRC** field is a CRC that protects major_sync_info. It is computed from all the preceding bits generated by the major_sync_info() syntax, modulo the polynomial:

$x^{16} + x^5 + x^3 + x^2 + 1$

(The shift register is cleared before the computation is made).

4.3 channel_meaning

4.3.1 2ch_dialogue_norm - u(6)

The 6-bit **2ch_dialogue_norm** field shall indicate the average dialogue level of the 2-channel presentation in LKFS. Values of 1 to 63 are interpreted as -1 LKFS to -63 LKFS. The reserved value of 0 is interpreted as -31 LKFS. The value of the **2ch_dialogue_norm** field is used to adjust the final sound reproduction level when decoding the 2-channel presentation from the Dolby TrueHD bit stream, so as to normalize the average dialogue level of the content to -31 LKFS. The gain applied is equal to (Value - 31) dB.

4.3.2 2ch_mix_level - u(6)

The 6-bit **2ch_mix_level** field is used to indicate the absolute acoustic sound pressure level of the 2-channel presentation during the final audio mixing session or at the original recording location. Values of 0 to 63 are interpreted as peak mixing levels of 70 to 133 dB in 1 dB steps.

The peak mixing level is defined as the acoustic level of a sine wave in a single channel whose peaks reach full scale. The absolute SPL value may be measured by means of pink noise with an RMS value of -20 or -30 dB with respect to the peak RMS sine wave level.

4.3.3 6ch_dialogue_norm - u(5)

The 5-bit **6ch_dialogue_norm** field shall indicate the average dialogue level of the 6-channel presentation in LKFS. Values of 1 to 31 are interpreted as -1 LKFS to -31 LKFS. The reserved value of 0 is interpreted as -31 LKFS. The value of the **6ch_dialogue_norm** field is used to adjust the final sound reproduction level when decoding the 6-channel presentation from the Dolby TrueHD bit stream, so as to normalize the average dialogue level of the content to -31 LKFS. The gain applied is equal to (Value - 31) dB.

4.3.4 6ch_mix_level - u(6)

The 6-bit **6ch_mix_level** field is used to indicate the absolute acoustic sound pressure level of the 6-channel presentation during the final audio mixing session or at the original recording location. Values of 0 to 63 are interpreted as peak mixing levels of 70 to 133 dB, in 1 dB steps.

The peak mixing level is defined as the acoustic level of a sine wave in a single channel whose peaks reach full scale. The absolute SPL value may be measured by means of pink noise with an RMS value of -20 or -30 dB with respect to the peak RMS sine wave level.

4.3.5 6ch_source_format - v(5)

The 5-bit **6ch_source_format** field may be used to describe hierarchical source information. The default value of 00000b indicates that the content is not of hierarchical origin.

4.3.6 8ch_dialogue_norm - u(5)

The 5-bit **8ch_dialogue_norm** field shall indicate the average dialogue level of the 8-channel presentation in LKFS. Values of 1 to 31 are interpreted as -1 LKFS to -31 LKFS. The reserved value of 0 is interpreted as -31 LKFS. The value of the **8ch_dialogue_norm** field is used to adjust the final sound reproduction level when decoding the 8-channel presentation from the Dolby TrueHD bit stream, so as to normalize the average dialogue level of the content to -31 LKFS. The gain applied is equal to (Value - 31) dB.

4.3.7 8ch_mix_level - u(6)

The 6-bit **8ch_mix_level** field is used to indicate the absolute acoustic sound pressure level of the 8-channel presentation during the final audio mixing session or at the original recording location. Values of 0 to 63 are interpreted as peak mixing levels of 70 to 133 dB in 1 dB steps.

The peak mixing level is defined as the acoustic level of a sine wave in a single channel whose peaks reach full scale. The absolute SPL value may be measured by means of pink noise with an RMS value of -20 or -30 dB with respect to the peak RMS sine wave level.

4.3.8 8ch_source_format - v(6)

The 6-bit **8ch_source_format** field may be used to describe hierarchical source information. The default value of 000000b indicates that the content is not of hierarchical origin.

4.3.9 extra_channel_meaning_present - b(1)

The 1-bit **extra_channel_meaning_present** field is set to '1' if further data are present after the first 64 bits of channel_meaning.

4.3.10 extra_channel_meaning_length - u(4)

The 4-bit **extra_channel_meaning_length** field shall indicate the length, in 16-bit words, of the expansion of **extra_channel_meaning()**, comprising **extra_channel_meaning_length** and **extra_channel_meaning_data**. The value of **extra_channel_meaning_length** is one less than the length of this expansion.

4.4 16ch_channel_meaning

4.4.1 16ch_dialogue_norm - u(5)

The 5-bit **16ch_dialogue_norm** field shall indicate the average dialogue level of the 16-channel presentation in LKFS. Values of 1 to 31 are interpreted as -1 LKFS to -31 LKFS. The reserved value of 0 is interpreted as -31 LKFS. The value of the **16ch_dialogue_norm** field is used to adjust the final sound reproduction level when decoding the 16-channel presentation from the Dolby TrueHD bit stream, so as to normalize the average dialogue level of the content to -31 LKFS. The gain applied is equal to (Value - 31) dB.

4.4.2 16ch_mix_level - u(6)

The 6-bit **16ch_mix_level** field indicates the absolute acoustic sound pressure level of the 16-channel presentation during the final audio mixing session or at the original recording location. Each 6-bit code represents a value in the range 0 to 63 dB.

The peak mixing level is 70 plus the value of **16ch_mix_level** dB SPL, thus falling into the range 70 to 133 dB SPL. It is defined as the acoustic level of a sine wave in a single channel whose peaks reach full scale. The absolute SPL value may be measured by means of pink noise with an RMS value of -20 or -30 dB with respect to the peak RMS sine wave level.

4.4.3 16ch_channel_count - u(5)

The 5-bit **16ch_channel_count** field shall specify the total number of audio channels present in the 16-channel presentation. The value of the **16ch_channel_count** field is one less than the number of channels present in the 16-channel presentation. As the maximum number of channels is limited to 16, Bit 4 (MSB) of the **16ch_channel_count** field shall be set to '0'.

4.4.4 dyn_object_only - b(1)

The 1-bit **dyn_object_only** field provides an efficient way to indicate that all full bandwidth channels of the 16-channel presentation are dynamic object audio channels. A value of '1' shall indicate that all full bandwidth channels of the 16-channel presentation are dynamic object audio channels. A value of '0' indicates that not all audio channels in the 16-channel presentation are

dynamic object audio channels, and the assignment of each channel is defined by the 16ch_content_description field and subsequent fields in 16ch_channel_meaning.

4.4.5 lfe_present - b(1)

The 1-bit **lfe_present** field is used to indicate that a 16-channel presentation consisting of dynamic audio objects also contains an LFE channel. A value of '1' shall indicate that the first channel in the 16-channel presentation contains an LFE channel and all subsequent channels are full bandwidth dynamic object audio channels. A value of '0'shall indicate that an LFE channel is not present, and all channels of the 16-channel presentation are full bandwidth dynamic object audio channels.

4.4.6 16ch_content_description - v(4)

The 4-bit **16ch_content_description** field is a bit field that is used to describe the types of audio channels that make up the 16-channel presentation when the 16-channel presentation contains channels that are not dynamic objects. The bit assignment of the **16ch_content_description** is specified in this table.

Table 17: 16ch_content_description meaning

16ch_content_description bits	Meaning
0000	Illegal
0001	16-ch presentation consists of loudspeaker feeds
0010	16-ch presentation consists of Intermediate Spatial Format audio
0011	16-ch presentation consists of loudspeaker feeds followed by Intermediate Spatial Format audio
0100	Reserved
0101	16-ch presentation consists of loudspeaker feeds followed by dynamic objects
0110	16-ch presentation consists of Intermediate Spatial Format audio followed by dynamic objects
0111	16-ch presentation consists of loudspeaker feeds followed by Intermediate Spatial Format audio followed by dynamic objects
1000 - 1111	Reserved

The total number of loudspeaker feed, intermediate spatial format and dynamic object channels is equal to the number of channels specified by the value of 16ch_channel_count.

4.4.7 lfe_only - b(1)

If bit 0 of the 16ch_content_description field is set to '1', the 1-bit **lfe_only** field is used when the only loudspeaker feed channel in the 16-channel presentation is an LFE channel and all other channels are intermediate spatial format channels (if bit 1 of the 16ch_content_description field is set to '1') and/or dynamic object channels (if bit 2 of the 16ch_content_description field is set to '1'). A value of '1' shall indicate that the first channel in the 16-channel presentation is an LFE channel, and that this is the only loudspeaker feed channel in the 16-channel presentation. All subsequent channels carry full bandwidth intermediate spatial format channels and/or dynamic objects. A value of '0' shall indicate that loudspeaker feeds other than an LFE channel are present in the 16-channel presentation and the channel assignment of these loudspeaker feeds are specified by the 16ch_channel_assignment field.

4.4.8 16ch_channel_assignment - v(10)

If bit 0 of the 16ch_content_description field is set to '1', and the 16-channel presentation contains loudspeaker feed channels other than an LFE channel, then the 10-bit **16ch_channel_assignment** field is used to indicate the assignment of the channels to loudspeaker feeds. Each bit in the field represents a different channel or channel pair present in the 16-channel presentation, as shown in this table.

Table 18: 16ch_channel_assignment Information

16ch_channel_assignment Bit	Channel(s) present when bit = 1	Number of Channels
0	L/R	2
1	C	1
2	LFE	1
3	Ls/Rs	2
4	Lb/Rb	2
5	Tfl/Tfr	2
6	Tsl/Tsr	2
7	Tbl/Tbr	2
8	Lw/Rw	2
9	LFE2	1

When bit = 1 the channel(s) exist(s); when bit = 0 there is no assignment. Channels in the bitstream shall follow in the order they are defined above. For example, if a program has a channel assignment that includes C and LFE, then the corresponding order of channels input to an encoder and output from a decoder is C followed by LFE.

4.4.9 16ch_intermediate_spatial_format - v(3)

If bit 1 of the 16ch_content_description field is set to '1', the 3-bit **16ch_intermediate_spatial_format** field shall indicate the intermediate spatial format of the 16-channel presentation. The **16ch_intermediate_spatial_format** field is set and interpreted as shown in this table.

Table 19: 16ch_intermediate_spatial_format

16ch_intermediate_ spatial_format value	Intermediate Spatial Format Indicated - Channels Present (in M.U.L.Z. layer order)	Number of Channels
000b - 001b	Reserved	N/A
010b	BH7.3.0.0 - M1 M2 M3 M4 M5 M6 M7 U1 U2 U3	10
011b	BH9.5.0.0 - M1 M2 M3 M4 M5 M6 M7 M8 M9 U1 U2 U3 U4 U5	14
100b	BH7.5.3.0 - M1 M2 M3 M4 M5 M6 M7 U1 U2 U3 U4 U5 L1 L2 L3	15
101b - 111b	Reserved	N/A

Intermediate Spatial Format audio channels shall follow in the bitstream in the order in which they are specified in Table 19. For example, if a program has a 16ch_intermediate_spatial _format value of 010b, then the corresponding order of channels input to an encoder and output from a decoder is M1 M2 M3 M4 M5 M6 M7 U1 U2 U3.

4.4.10 16ch_dynamic_object_count - u(5)

The 5-bit **16ch_dynamic_object_count field** specifies the number of dynamic objects (i.e. objects with potentially varying properties such as position) present in the 16-channel presentation. The value of the **16ch_dynamic_object_count** field shall be one less than the number of dynamic objects present in the 16-channel presentation.

When the 16-channel presentation consists of a mixture of loudspeaker feeds, intermediate spatial format audio channels and/or dynamic objects, the full channel assignment of the 16-channel presentation is given by the combination of the lfe_only, 16ch_channel_assignment, 16ch_intermediate_spatial_format, and 16ch_dynamic_object_count fields. For example, if the fields in 16ch_channel_meaning are set to the following values:

- 16ch_content_description: 0111b

- lfe_only: 0b
- 16ch_channel_assignment: 0000000110b (indicating C and LFE channels)
- 16ch_intermediate_spatial_format: 010b (indicating a 10-channel intermediate spatial format of M1 M2 M3 M4 M5 M6 M7 U1 U2 U3)
- 16ch_dynamic_object_count: 00001b indicating 2 dynamic objects

The overall channel assignment of the 16-channel presentation is C LFE M1 M2 M3 M4 M5 M6 M7 U1 U2 U3 Obj1 Obj2 (where Obj1 and Obj2 are the two dynamic objects). Channels are input to an encoder, and output from a decoder, in the order specified by this channel assignment.

4.5 Substream directory

4.5.1 extra_substream_word - b(1)

If the 1-bit **extra_substream_word[i]** field is set to '1', an additional 16-bit word shall follow the substream_end_ptr[i] field in the bit stream.

4.5.2 restart_nonexistent - b(1)

The 1-bit **restart_nonexistent[i]** field is set to '1' unless the access unit starts with a major sync. If the **restart_nonexistent[i]** field is set to '0', the first block within each substream segment shall contain a restart_header. An access unit with a major sync shall have a restart_header in the first block within each substream_segment. The encoder shall ensure that the value of the first 16 bits after substream_directory in a **minor_sync** is not 0xF872.

4.5.3 crc_present - b(1)

If the 1-bit **crc_present[i]** field is set to '1', the substream_parity field and substream_CRC field shall follow substream_segment [i] in the bitstream.

4.5.4 substream_end_ptr - u(12)

The 12-bit **substream_end_ptr[i]** field shall indicate the offset of substream_end[i] relative to start. All pointers and offsets within the substream are in units of 16-bit words.

4.6 Substream segment

4.6.1 last_block_in_segment - b(1)

The 1-bit **last_block_in_segment** field shall follow each block in the substream segment. A value of '1' shall indicate that the block that immediately precedes the **last_block_in_segment** field is the last block in the substream segment, and this block is padded to the next 16-bit word boundary. A value of '0' shall indicate that one or more additional blocks follow in the bitstream.

For substreams with a restart_sync_word value of 0x31EA, the number of blocks that may be present in each substream segment is limited to a maximum of four.

For substreams with a restart_sync_word value of 0x31EB and 0x31EC, the number of blocks that may be present in each substream segment is limited to a maximum of two.

4.6.2 terminatorA - v(18)

The 18-bit **terminatorA** field is present in each substream segment of the final access unit of the Dolby TrueHD bitstream. The value of the terminatorA field shall be set to 0x348D3

Note that it is not guaranteed that the bit pattern 0x348D3 will not occur at a non-terminal position within an Dolby TrueHD bit stream, for example as part of the encoded audio data. It is not, however, possible for an expansion of the block() syntax to result in a bit pattern that begins with 0x348D3.

4.6.3 zero_samples_indicated - b(1)

If additional sample periods are needed to complete the last access unit in the bit stream, the 1-bit **zeros_samples_indicated** field is set to '1'.

4.6.4 zero_samples - u(13)

The 13-bit **zero_sample**s field shall indicate the number of sample periods which have been added to the input audio material to complete the access unit.

4.6.5 terminatorB - v(13)

If the zeros_samples_indicated field is set to '0', the substream segment is completed with the 13-bit **terminatorB** field. If present, the value of the **terminatorB** field shall be set to 0x1234.

4.6.6 substream_parity - u(8)

The 8-bit **substream_parity** field is a parity check. The value of the **substream_parity** field is equal to the exclusive-OR of all the bytes in the expansion of substream_segment(), exclusive-ORed with the constant 0xA9 (the purpose of the latter being to force the check to fail in the event of the stream consisting entirely of zeroes).

4.6.7 substream_CRC - u(8)

The 8-bit **substream_CRC** field shall contain a CRC computed from all the bits in the expansion of substream_segment(), modulo the polynomial

$x^8 + x^6 + x^5 + x + 1$

where the relevant shift register is initialized to 0xA2 before the computation.

The value of **substream_CRC** is calculated using the following algorithm:

1. Set substream_CRC $\leftarrow$ `0xA2`.
2. For each bit in substream_segment(), taken in sequence,
   set substream_CRC $\leftarrow$ (substream_CRC << 1) + bit.
   If substream_CRC & `0x100` $\neq$ 0, set substream_CRC $\leftarrow$ substream_CRC $\oplus$ `0x163`.

4.7 Block

4.7.1 restart_header_exists - b(1)

If the 1-bit **restart_header_exists** field is set to '1', the restart_header shall follow in the bitstream.

4.7.2 restart_header

restart_sync_word - v(14)

The 14-bit **restart_sync_word** is used to distinguish between different types of substream, as shown in this table.

Table 20: Permitted values of restart_sync_word

Substream number	Permitted restart_sync_word value
0	0x31EA
1	0x31EA or 0x31EB
2	0x31EB
3	0x31EC

Note that due to the presence of the 1-bit block_header_exists and restart_header_exists fields within the block() syntax, a substream segment that starts with a restart header will begin with the 16-bit pattern 0xF1EA, 0xF1EB, or 0xF1EC.

output_timing - u(16)

The 16-bit **output_timing** field is used in combination with the input_timing field to control the timing of the decoding process, as described in FIFO timing and management. The value of the **output_timing** field is equal to the sample number of the first sample in the block containing the restart header, stored modulo 65536. The **output_timing** field is 16-bit aligned within the stream.

min_chan - u(4)

The 4-bit **min_chan** field indicates the minimum channel number carried by the substream. The value of the **min_chan** field is one less than the minimum channel number carried by the substream.

max_chan - u(4)

The 4-bit **max_chan** field indicates the maximum channel number carried by the substream. The value of the **max_chan** field is one less than the maximum channel number carried by the substream. The output channels of the matrixing operation are numbered from 0 to **max_chan**.

error_protect - b(1)

The encoder may include additional error protection within the substream. If the 1-bit **error_protect** field is set to '1', the block_data_bits and block_header_CRC fields are present in the bit stream.

restart_header_CRC - u(8)

The 8-bit **restart_header_CRC** field is a CRC that provides a check for all fields that make up the restart header. The value of the restart_header_CRC field is calculated using all fields in the restart header up to (but not including) the **restart_header_CRC** itself. The shift register is initialised to zero before the calculation. The following polynomial is used for the calculation:

$x^8 + x^4 + x^3 + x^2 + 1$.

4.8 EXTRA_DATA

4.8.1 Overview

EXTRA_DATA, if present, allows for extensions to Dolby TrueHD. The **EXTRA_DATA** field can also include padding inserted by an encoder or transcoder in order to produce a fixed-rate stream.

EXTRA_DATA starts at **extra_start**, which is at the same position in the stream as **substream_end[substreams-1]**. Thus if **substream_end_ptr[substreams-1]** equals **unit_end-start** there is no **EXTRA_DATA**. **EXTRA_DATA** shall always occupy a complete number of 16-bit words. If the first word of **EXTRA_DATA** is zero, the **EXTRA_DATA** consists entirely of padding (and may be discarded). If **EXTRA_DATA** is present, the first 16-bit word shall consist of the **length_check_nibble** and **EXTRA_DATA_length** parameters:

4.8.2 length_check_nibble - v(4)

The 4-bit **length_check_nibble** field is a check which is calculated so that the exclusive-OR of **length_check_nibble** and the three following 4-bit nibbles that make up the EXTRA_DATA_length parameter shall equal 0xF.

4.8.3 EXTRA_DATA_length - u(12)

The 12-bit **EXTRA_DATA_length** field shall indicate the total length, in 16-bit words, of **EXTRA_DATA**, including the length_check_nibble and **EXTRA_DATA_length** fields. The value of the **EXTRA_DATA_length** field is one less than the length of **EXTRA_DATA**.

4.8.4 EXTRA_DATA_padding - pad(variable).

The **EXTRA_DATA_padding** field ensures that the expansion of **EXTRA_DATA** is aligned to a 16-bit boundary. The **EXTRA_DATA_padding** field can also be used to fill out an access unit to a specified

size. The size of the **EXTRA_DATA_padding** field is variable and is equal to (EXTRA_DATA_length * 16) – 24 bits.

4.8.5 EXTRA_DATA_parity - v(8)

The 8-bit **EXTRA_DATA_parity** field is a parity check. The value of the field is equal to the exclusive-OR of all the bytes calculated over all the bytes in the expansion of **EXTRA_DATA** (excluding length_check_nibble, EXTRA_DATA_length, and the **EXTRA_DATA_parity** fields), exclusive-ORed with the constant 0xA9 (the purpose of the latter being to force the check to fail in the event of the stream consisting entirely of zeroes).

5 Dolby TrueHD file formats

This section specifies the format of a Dolby TrueHD file and describes the different data frame types that can be present in a file. Dolby TrueHD files use the .mlp file extension.

5.1 Introduction

A Dolby TrueHD file consists of a sequence of Dolby TrueHD access units. The file may include a 16-byte header that contains a SMPTE timestamp. Each Dolby TrueHD access unit begins with an MLP Sync.

- The MLP Sync must be a **major sync** to permit decoding from the start of the file.
- The maximum interval between access units containing a **major sync** is 128 access units.
- The minimum interval between access units containing a **major sync** is eight access units, except at the start of the Dolby TrueHD file, where the first two access units are both permitted to contain a major sync.

The initial 64 bits of an encoded Dolby TrueHD file determines the file type (or file format). One of the file types is simply a sequence of Dolby TrueHD access units. The other file format includes a header that contains a 16-byte SMPTE timestamp before the first access unit.

5.2 Dolby TrueHD file format specification

The following pseudocode shows the format of a Dolby TrueHD file

Syntax

```
Dolby_TrueHD_file () {
   if (word2==0xF872 && word3==0x6FBA) {
      /* file begins with major sync access unit */
      while (!EOF) {
         access_unit()
      }
   }
   else if (word0==0x0110) {
      /* file begins with SMPTE timestamp */
      timestamp()
      while (!EOF) {
         access_unit()
      }
   }
} /* end of Dolby TrueHD file */
```

5.3 Dolby TrueHD file without SMPTE timestamp header

This file type consists of a sequence of Dolby TrueHD access units. The first access unit in the file is an access unit containing a major sync. A major sync is present in the first access unit of the file when the 32 bits of the file starting at bit offset 32 are equal to `0xF8726FBA`, indicating that the format_sync field is present.

5.4 Dolby TrueHD file with SMPTE timestamp header

This file type begins with a 16-byte header containing a SMPTE timestamp. The syntax of the SMPTE timestamp header is specified in SMPTE ST 339. For convenience, it is reproduced here.

Syntax	Encoding
`timestamp() {`	
`  start_syncword`	`v(16)`
`  hours`	`bcd(16)`
`  minutes`	`bcd(16)`
`  seconds`	`bcd(16)`
`  frames`	`bcd(16)`
`  samples`	`u(16)`
`  reserved1`	`v(10)`
`  framerate`	`v(4)`
`  reserved2`	`v(1)`
`  dropframe`	`b(1)`
`  reserved3`	`v(16)`
`}`	

Note: The value of the **start_syncword** is always `0x0110`. The **hours**, **minutes**, **seconds** and **frames** fields of the timestamp are encoded using the packed binary coded decimal (bcd) format.

Following the timestamp, the file is identical to a Dolby TrueHD file that does not include the timestamp, and consists of a sequence of Dolby TrueHD access units. The first access unit of the file, immediately following the timestamp, is an access unit containing a major sync.

Note: The **minutes** and **seconds** fields of the SMPTE timestamp begin at bit offset 32 of a Dolby TrueHD file that includes a timestamp. In a Dolby TrueHD file that does not include a timestamp, the format_sync field begins at bit offset 32. However, the **minutes** and **seconds** fields can never take the value of format_sync which is always `0xF8726FBA`. This ensures that misdetection of the Dolby TrueHD file type cannot occur.

6 Glossary

The following list provides definitions for a number of important terms used throughout this document:

access unit

The minimum portion of the audio serial bit stream capable of being fully decoded. See the syntax for access_unit() for the precise definitions.

CRC

Cyclic Redundancy Check

downmixing

Combining (or mixing down) the content of n original channels to produce m channels, where m<n.

DRC

Dynamic Range Compression

FIFO

First In First Out

full bandwidth (fbw) channel

An audio channel capable of full audio bandwidth. All channels except the LFE channel are fbw channels.

low frequency effects (LFE) channel

An optional single channel of limited (<120 Hz) bandwidth, which is intended to be reproduced at a level +10 dB with respect to the fbw channels. The optional LFE channel allows high sound pressure levels to be provided for low frequency sounds.

LKFS

Loudness, K-weighted, relative to full scale.

LSB

Least Significant Bit

MLP

Meridian Lossless Packing. The lossless compression algorithm used by Dolby TrueHD to compress audio data.

PCM

Pulse Code Modulation

reserved

An element that is set aside for use by a future version.

substream

A subcomponent of the overall Dolby TrueHD bitstream that carries a portion of the overall audio presentation, as specified by the associated semantics.