SMPTE REGISTERED DISCLOSURE DOCUMENT

Uncompressed Video Transport Over MPEG-2 Transport Stream

The attached document is a Registered Disclosure Document prepared by the sponsor identified below. It has been examined by the appropriate SMPTE Technology Committee and is believed to contain adequate information to satisfy the objectives defined in the Scope, and to be technically consistent.

This document is NOT a Standard, Recommended Practice or Engineering Guideline, and does NOT imply a finding or representation of the Society.

Every attempt has been made to ensure that the information contained in this document is accurate. Errors in this document should be reported to the proponent identified below, with a copy to eng@smpte.org.

Evertz (or the proponent) does not hold any patents essential to implementation of this RDD, and no license is required from Evertz (or proponent).

Evertz (or the proponent) is not aware of any patents held by other entities that are essential to the implementation of this RDD, but it is the responsibility of the implementer to conduct appropriate research and to obtain any necessary licenses.

All other inquiries in respect of this document should be addressed to the proponent identified below.

Proponent contact information:

Rakesh Jalali
Evertz Microsystems Ltd.
Email: rjalali@evertz.com

Approved
February 26, 2016

Table of Contents

Page

Introduction

This document describes a method of mapping uncompressed video to an MPEG-2 transport stream.

MPEG-2 transport streams convey one or more programs of coded data, and can be constructed from one or more elementary coded data streams, program streams, or other transport streams. This document specifies a method to transport uncompressed video streams within an MPEG-2 transport stream using existing SMPTE ST 302 for audio and SMPTE ST 2038 for ancillary data.

1 Scope

This document outlines the architecture and structure of changes to the existing broadly used MPEG-2 transport stream specification to accommodate efficient transmission of uncompressed video over MPEG-2 transport stream. It details how uncompressed video is packed into an elementary stream and PES stream. It outlines an encapsulation structure for carrying uncompressed video over transport streams and relies on existing SMPTE ST 302, SMPTE ST 2038, ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 for PAT, PMT and PID management structures.

2 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this Registered Disclosure Document (RDD). At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this RDD are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

SMPTE ST 291-1:2011, Ancillary Data Packet and Space Formatting

SMPTE ST 302:2007, Television — Mapping of AES3 Data into an MPEG-2 Transport Stream

SMPTE ST 2038:2008, Carriage of Ancillary Data Packets in an MPEG-2 Transport Stream

ITU-T Recommendation H.222.0 (10/14)/ISO/IEC 13818-1:2015, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 1: Systems

ITU-T Recommendation T.800 (11/15) | ISO/IEC 15444-1:2004, Information Technology — JPEG 2000 Image Coding System: Core Coding System

3 Video Co-ordinate Convention

All references to line numbers and pixel numbers shall follow this convention:

- The first line of the frame is line 0. Line numbers increment by 1 for each subsequent line of the frame. At the beginning of the next frame the line number reverts to 0.
- The first pixel of the line is pixel 0. Pixel numbers increment by 1 for each subsequent pixel of the line. At the beginning of the next line the pixel number reverts to 0.
- The active picture is right-justified. The last active pixel of a line is also the last pixel of that line. The horizontal blanking region starts at pixel number 0.

4 Program Association Table (PAT)

The Program Association Table shall conform with ITU-T Recommendation H.222.0 / ISO/IEC 13818-1.

5 Program Map Table (PMT) — Video Section

The Program Map Table shall conform to J2K video descriptor defined in ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 with the following exceptions:

- The **descriptor_tag** shall have a value of 224 to indicate an uncompressed video descriptor.
- The **stream_type** shall have a value of 0xEA to indicate an uncompressed video stream.
- The **profile_and_level** field shall be 0.
- The **vertical_size** field shall indicate the total number of active lines in the video frame.
- The **max_bit_rate** field may be 0.
- The **max_buffer_size** field may be 0.
- The **private_data_byte** section shall be formatted as described below

Table 1 – Private Data Bytes in Video Descriptor

Private_data_bytes() {		
total_horizontal_size	16	uimsbf
first_active_pixel	16	uimsbf
for (field = 0; field < 2; field++) {		
total_vertical_size[field]	16	uimsbf
active_vertical_size[field]	16	uimsbf
first_active_line[field]	16	uimsbf
first_extended_active_line[field]	16	uimsbf
}		
‘0000’	4	bslbf
component_size	4	uimsbf
‘000000’	6	bslbf
sample_structure	2	uimsbf
horizontal_sync_start	16	uimsbf
horizontal_sync_stop	16	uimsbf
for (field = 0; field < 2; field++) {		
vertical_sync_start[field]	16	uimsbf
vertical_sync_stop[field]	16	uimsbf
vertical_sync_horizontal_position[field]	16	uimsbf
}		
horizontal_sync_polarity	1	bslbf
vertical_sync_polarity	1	bslbf
Reserved	6	bslbf
}		

total_horizontal_size:
Shall specify the total number of pixels in an entire line including horizontal blanking.

first_active_pixel:
Shall specify the pixel number of the first active pixel in a line.

total_vertical_size[field]:
Shall specify the number of lines in the entire field including vertical blanking. The value of **total_vertical_size[1]** shall be 0 if video is progressive.

active_vertical_size[field]:
Shall specify the number of lines in the active part of the field. The value of **active_vertical_size[1]** shall be 0 if video is progressive.

first_active_line[field]:
Shall specify the line number of the first active line in the field. The value of **first_active_line[1]** shall be 0xFFFF if video is progressive.

first_extended_active_line[field]:

- For standard definition signals there may be video data signals (such as D-VITC or CEA 608 closed captions) present that is not formatted according to SMPTE ST 291-1 and cannot therefore be carried according to SMPTE ST 2038. This data is sometimes in a proprietary format. In order to convey this data in a simple manner, some of the lines immediately preceding the active region may be encapsulated as if they were part of the active region. The lines that are included shall be contiguous and adjacent to the first active line of the field.
- **first_extended_active_line[field]** shall indicate the first line that is encapsulated as if it is part of the vertical active region. All of the lines from starting line to the end of the vertical active region, inclusive, are treated as vertical active lines.

  first_extended_active_line[field] shall be the same value as **first_active_line[field]** if none of the lines in the vertical blanking region are to be treated as if they are vertical active lines.
- **first_extended_active_line[1]** shall be 0xFFFF if video is progressive.

component_size:
Shall specify the number of bits used to specify the value of each component. All components shall use the same number of bits.

sample_structure:
Shall specify the sampling structure of the components and the number of components.

0 = 4:2:2
1 = 4:4:4
2 = 4:4:4:4
3 = 4:2:2:4

horizontal_sync_start:
Shall specify the pixel number at which the horizontal sync starts.

horizontal_sync_stop:
Shall specify the pixel number at which the horizontal sync stops.

vertical_sync_start[field]:
Shall specify the line number in which the vertical sync starts for the field. For progressive rasters, **vertical_sync_start[1]** shall be 0xFFFF.

vertical_sync_stop[field]:
Shall specify the line number in which the vertical sync stops for the field. For progressive rasters, **vertical_sync_stop[1]** shall be 0xFFFF.

vertical_sync_horizontal_position[field]:
Shall specify the pixel number at which the vertical sync changes for the field. For progressive rasters, **vertical_sync_horizontal_position[1]** shall be 0xFFFF.

horizontal_sync_polarity:
Shall specify the polarity of the horizontal sync signal. A value of 1 shall indicate active-high polarity, a value of 0 shall indicate active-low polarity.

vertical_sync_polarity:
Shall specify the polarity of the vertical sync signal. A value of 1 shall indicate active-high polarity, a value of 0 shall indicate an active-low polarity.

reserved: This is reserved and shall be '0'.

6 Program Clock Reference (PCR)

The PCR may have its own exclusive PID or it may share the Video PID. Packet identifier(PID) is defined in ITU-T Recommendation H.222.0 / ISO/IEC 13818-1.

If the PCR shares the Video PID then it shall be sent in a packet with the **adaptation_field_control** '10' indicating the presence of the **adaptation_field** and the absence of any **data_bytes**.

7 Ancillary Data

Ancillary Data shall conform to SMPTE ST 2038.

8 Audio Data

Audio data shall conform to SMPTE ST 302, except for Sections 5.4, 6.9 and 6.10 of SMPTE ST 302. This is for simplicity of implementation and to allow for the transport of audio that is not associated with a particular video transport.

9 Video Data

Video Data formatting shall utilize the **transport_packet** format defined in ITU-T Recommendation H.222.0 / ISO/IEC 13818-1.

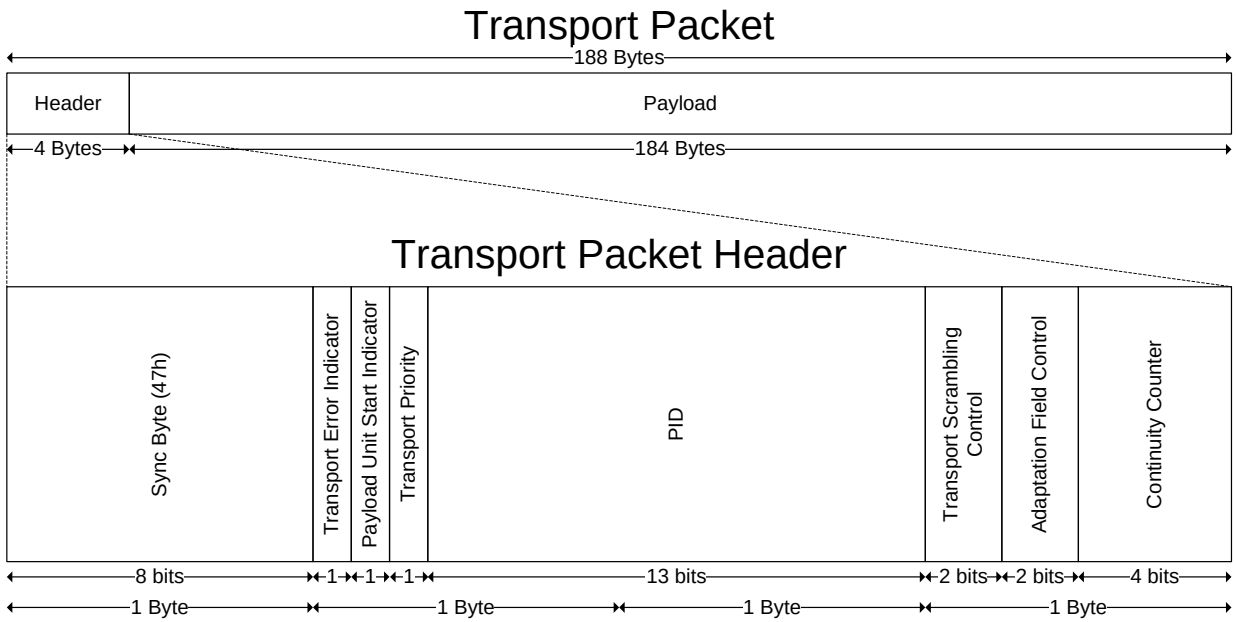


Figure 1 – Transport Packet

Figure 1 above details the standard transport stream packet layer as indicated in ITU-T Recommendation H.222.0 / ISO/IEC 13818-1.

Following exceptions apply:

payload_unit_start_indicator shall be 1 to indicate the start of a new video frame where the start of a new video frame is defined as the beginning of the first line of the frame.

adaptation_field_control shall be '01' indicating the absence of the **adaptation_field** and the presence of **data_bytes**.

PES_packet fields shall be formatted according to the uncompressed video **PES_packet** defined in Section 9.1.

9.1 Uncompressed Video PES_packet

The **PES_packet** shall conform to the **PES_packet** structure defined in ITU-T Recommendation H.222.0 / ISO/IEC 13818-1.

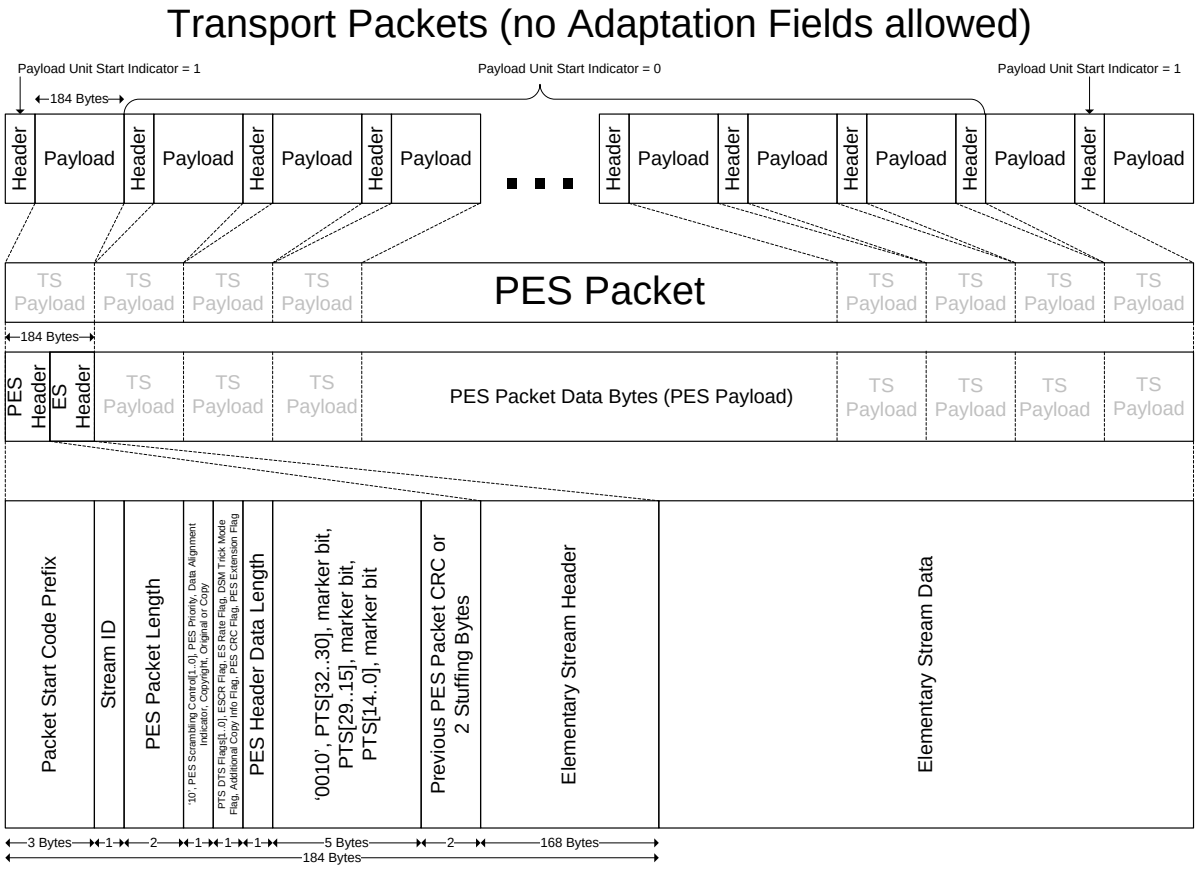


Figure 2 – Uncompressed Video PES Packet Structure

The following PES Header identifiers shall be set as indicated:

a. **`stream_id`** shall be 1011 1101 (private_stream_1).
b. **`PES_packet_length`** may be 0 as allowed by ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 for video elementary streams.
c. **`PTS_DTS_flag`** shall be '10' indicating the presence of the PTS field and the absence of the DTS field.
d. **`ESCR_flag`** shall be '0' indicating the absence of the ESCR fields.
e. **`ES_rate_flag`** shall be '0' indicating the absence of the **ES_rate_field**.
f. **`DSM_trick_mode_flag`** shall be '0' indicating the absence of the trick mode field.
g. **`additional_copy_info_flag`** shall be '0' indicating the absence of the **`additional_copy_info`** field.
h. **`PES_CRC_flag`** may be '0' or '1'. If '0' then there shall be exactly two **`stuffing_bytes`** preceding the **`PES_packet_data_bytes`**. If '1' then there shall be no **`stuffing_bytes`** present.
i. **`PES_extension_flag`** shall be '0' indicating the absence of an extension field.
j. **`PES_header_data_length`** shall be 7.
k. **`PTS`** shall refer to the time at the start of the video frame, when the first line begins
l. Exactly two **`stuffing_bytes`** shall be present if **`PES_CRC_flag`** is '0' and no **`stuffing_bytes`** shall be present **if `PES_CRC_flag`** is '1'.

The PES data payload **PES_packet_data_bytes** shall be formatted according to the **PES_packet_data_bytes** defined in Section 9.1.1.

9.1.1 PES_packet_data_bytes

The section describes the uncompressed active region video payload carried within the **PES_packet_data_bytes**. All references to line numbers and pixel numbers shall follow this convention:

The **PES_packet_data_byte** structure is represented by the structure below, and is composed of an **uncompressed_video_elementary_stream_header** as defined in Section 9.1.1.1 and **uncompressed_video_elementary_stream_data** as defined in Section 9.1.1.2.

Table 2 – PES Packet Data Bytes Structure

PES_packet_data_bytes() {		
Uncompressed_video_elementary_stream_header()	1344	bslbf
for (i = 0; i < N; i++) {		
Uncompressed_video_elementary_stream_data	8	bslbf
}		
}		

9.1.1.1 Uncompressed_video_elementary_stream_header (ES Header)

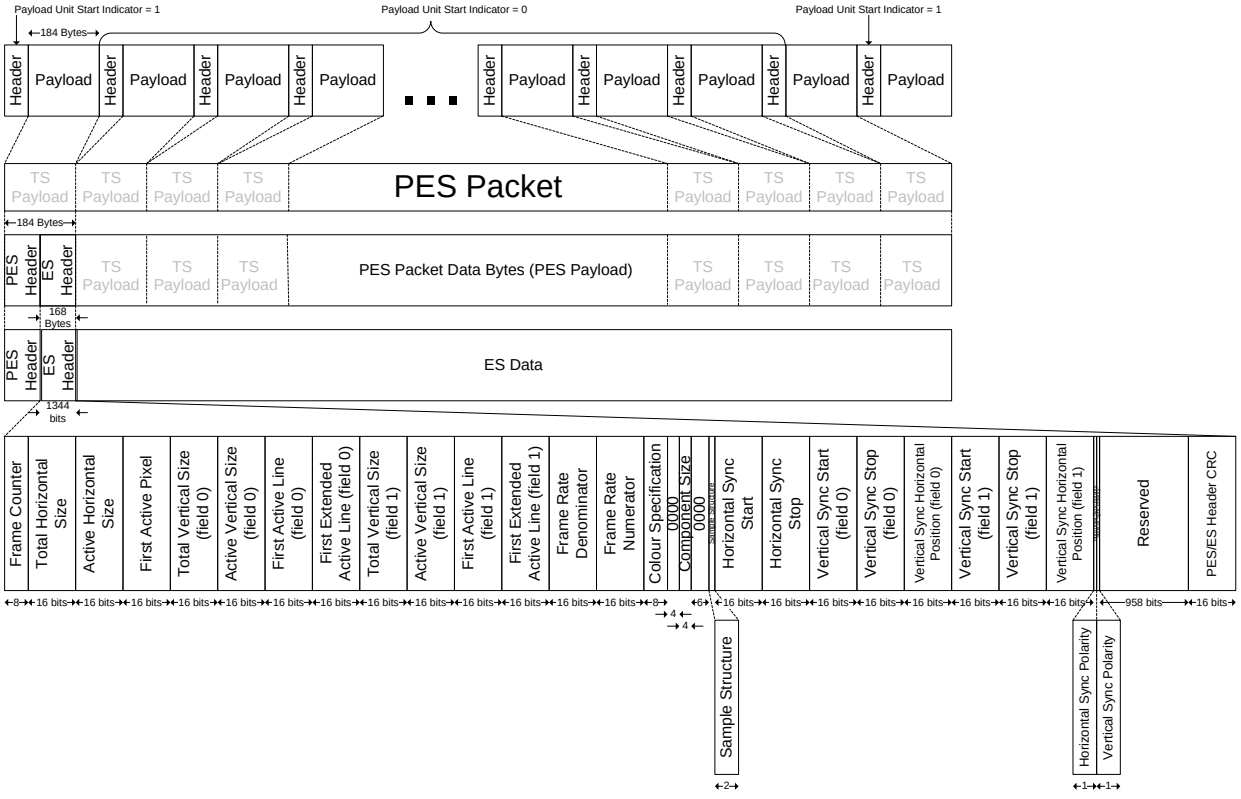


Figure 3 – Uncompressed Video Elementary Stream Header

The **uncompressed_video_elementary_stream_header** (ES Header) is sized such that it will exactly fill the transport_packet within which it co-exists with the PES_header. Its byte structure is defined below:

Table 3 – Uncompressed Video Elementary Stream Header Structure

Uncompressed_video_elementary_stream_header(){		
frame_counter	8	Uimsbf
total_horizontal_size	16	Uimsbf
active_horizontal_size	16	Uimsbf
first_active_pixel	16	Uimsbf
for (field = 0; field < 2; field++) {		
total_vertical_size[field]	16	Uimsbf
active_vertical_size[field]	16	Uimsbf
first_active_line[field]	16	Uimsbf
first_extended_active_line[field]	16	Uimsbf
}		
frame_rate_denominator	16	Uimsbf
frame_rate_numerator	16	Uimsbf
color_specification	8	Uimsbf
‘0000’	4	Uimsbf
component_size	4	Uimsbf
‘000000’	6	uimsbf
sample_structure	2	uimsbf
horizontal_sync_start	16	uimsbf
horizontal_sync_stop	16	uimsbf
for (field = 0; field < 2; field++) {		
vertical_sync_start[field]	16	uimsbf
vertical_sync_stop[field]	16	uimsbf
vertical_sync_horizontal_position[field]	16	uimsbf
}		
horizontal_sync_polarity	1	bslbf
vertical_sync_polarity	1	bslbf
reserved	958	uimsbf
PES_ES_header_CRC	16	uimsbf
}		

The **uncompressed_video_elementary_stream_header** provides essential information such as raster dimensions, color specification and frame rate. The following definitions apply:

frame_counter:
The frame_counter increments by 1 for each frame. This value returns to 0 after reaching the terminal value of 255.

total_horizontal_size:
Shall specify the number of pixels in an entire line including horizontal blanking.

active_horizontal_size:
Shall specify the number of pixels in the active part of a line.

first_active_pixel:
Shall specify the pixel number of the first active pixel in a line.

total_vertical_size[field]:
Shall specify the number of lines in the entire field including vertical blanking. If the video is progressive then for field 1 this value shall be 0.

active_vertical_size[field]:
Shall specify the number of lines in the active part of the field. If the video is progressive then for field 1 this value shall be 0.

first_active_line[field]:
Shall specify the first line number of the first active line in the field. If the video is progressive then for field 1, this value shall be 0xFFFF.

first_extended_active_line[field]:
For some signals (e.g. standard definition signals) there may be ancillary data present that is not formatted according to ST 291-1. This data is sometimes in a proprietary format. In order to convey this data in a simple manner, some of the lines immediately preceding the vertical active region may be encapsulated as if they were part of the vertical active region. The lines that are included shall be contiguous and shall be adjacent to the first active line of the field. **first_extended_active_line[field]** shall indicate the first line in the vertical blanking region that will be encapsulated as if it is part of the vertical active region. All of the lines beginning with this line to the end of the vertical active region are treated as vertical active lines. Shall be the same value as **first_active_line[field]** if none of the lines in the vertical blanking region are to be treated as if they are vertical active lines.

frame_rate_denominator:
This is identical to the DEN field specified in Table 2-99 of Section 2.6.80 of ITU-T Recommendation H.222.0 / ISO/IEC 13818-1

frame_rate_numerator:
This is identical to the NUM field specified in Table 2-99 of Section 2.6.80 of ITU-T Recommendation H.222.0 / ISO/IEC 13818-1

color_specification:
Shall specify the color format
0x00: Unspecified
0x01: IEC 61966-2-1:1999
0x02: ITU-R Recommendation BT.601-7
0x03: ITU-R Recommendation BT.709-6

0x04: Refer ITU-T Recommendation H.222.0 / ISO/IEC 13818-1
0x05: ISO 26428-1 (X'Y'Z')
0x06: ITU-R Recommendation BT.2020-2
0x07-0xFF: Reserved

component_size:
Shall specify the number of bits used to define the value of each component. All components shall use the same number of bits.

sample_structure
Shall specify the sampling structure and the number of components.
0 = 4:2:2, 1 = 4:4:4, 2 = 4:4:4:4, 3 = 4:2:2:4

horizontal_sync_start:
Shall specify the pixel number at which the horizontal sync starts.

horizontal_sync_stop:
Shall specify the pixel number at which the horizontal sync stops.

vertical_sync_start[field]:
Shall specify the line number in which the vertical sync starts for the field. For progressive rasters, **vertical_sync_start[1]** shall be 0xFFFF.

vertical_sync_stop[field]:
Shall specify the line number in which the vertical sync stops for the field. For progressive rasters, **vertical_sync_stop[1]** shall be 0xFFFF.

vertical_sync_horizontal_position[field]:
Shall specify the pixel number at which the vertical sync changes for the field. For progressive rasters, **vertical_sync_horizontal_position[1]** shall be 0xFFFF.

horizontal_sync_polarity:
Shall specify the polarity of the horizontal sync signal. A value of 1 shall indicate active-high polarity, a value of 0 shall indicate active-low polarity.

vertical_sync_polarity:
Shall specify the polarity of the vertical sync signal. A value of 1 shall indicate active-high polarity, a value of 0 shall indicate active-low polarity.

reserved:
This is reserved and shall be '0'.

PES_ES_header_CRC:
Shall contain the CRC value that yields a zero output of the 16 registers in the decoder similar to the one defined in ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 Annex A, but with the polynomial: $x^{16} + x^{12} + x^5 + 1$ after processing all of the bytes from the beginning of the **PES_packet** to the end of the **uncompressed_video_elementary_stream_header**

9.1.1.2 Uncompressed_video_elementary_stream_data (ES Data)

The **uncompressed_video_elementary_stream_data** (ES Data) contains the uncompressed video data.

An uncompressed video elementary stream data shall be composed of 184 byte **uncompressed_video_elementary_stream_units** (ES Unit)

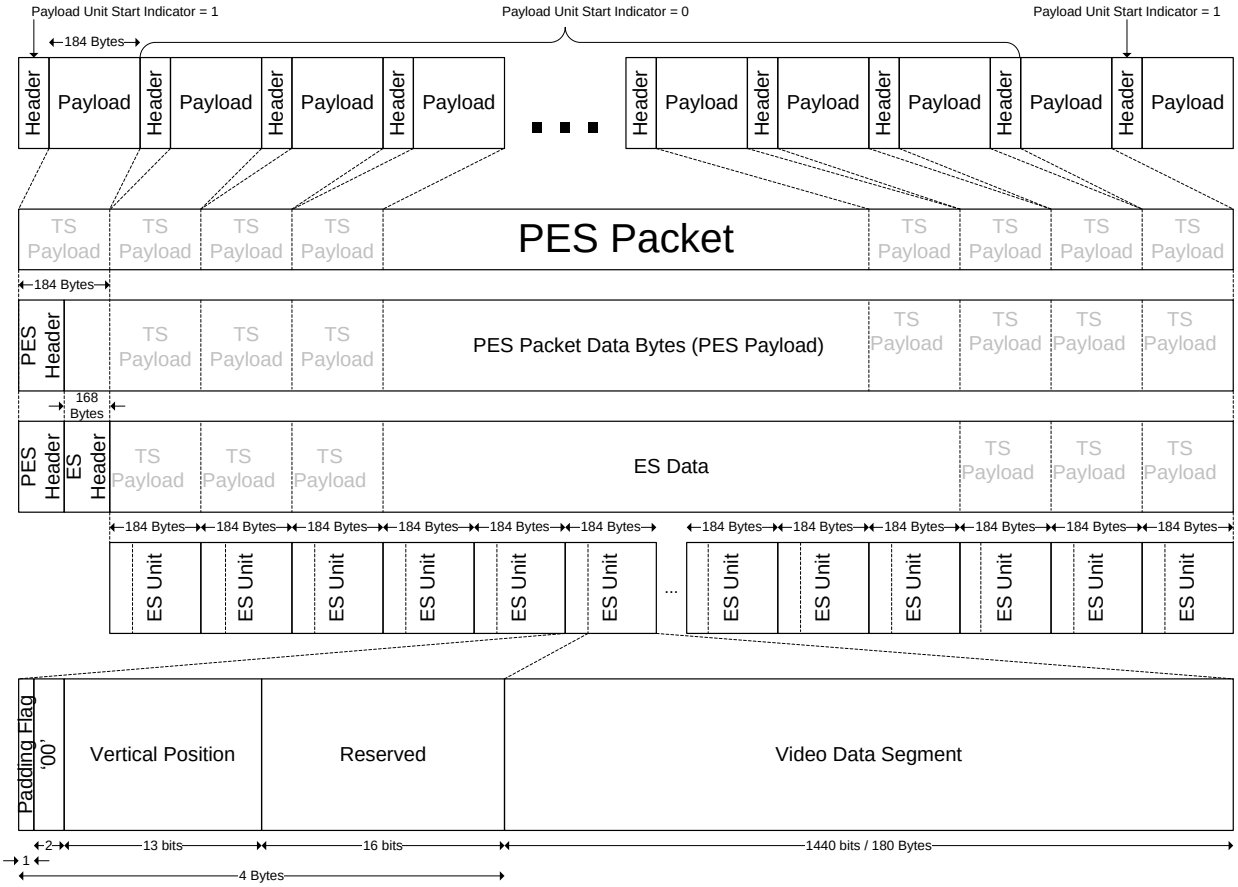


Figure 4 – Uncompressed Video Elementary Stream Data

When all of the **uncompressed_video_elementary_stream_units** are concatenated they form the entire **uncompressed_video_elementary_stream_data** for an **uncompressed_video_pes_packet**.

There shall be exactly one **uncompressed_video_elementary_stream_unit** per **transport_packet**. The **uncompressed_video_elementary_stream_unit** format is defined in Section 9.1.1.2.1.

9.1.1.2.1 Uncompressed_video_elementary_stream_unit

Table 4 – Uncompressed Video Elementary Stream Unit Structure

uncompressed_video_elementary_stream_unit() {		
padding_flag	1	bslbf
'00'	2	bslbf
vertical_position	13	uimsbf
reserved	16	bslbf
video_data_segment()	1440	bslbf
}		

padding_flag:
Shall specify that the latter part of this **uncompressed_video_elementary_stream_unit** contains **padding_bits** (see Figure 5). This value shall be 0 if the **uncompressed_video_elementary_stream_unit** contains no **padding_bits,** and it shall be 1 if the **uncompressed_video_elementary_stream_unit** contains **padding_bits**.

vertical_position:
Shall specify the line number of the first unit of data in the **video_data_segment** field.

reserved:
This is reserved and shall be '0'.

video_data_segment:
This value is a 1440 bit segment of all of the **video_data** for the entire frame. Further explanation of **video_data** is defined in Section 9.1.1.2.2.

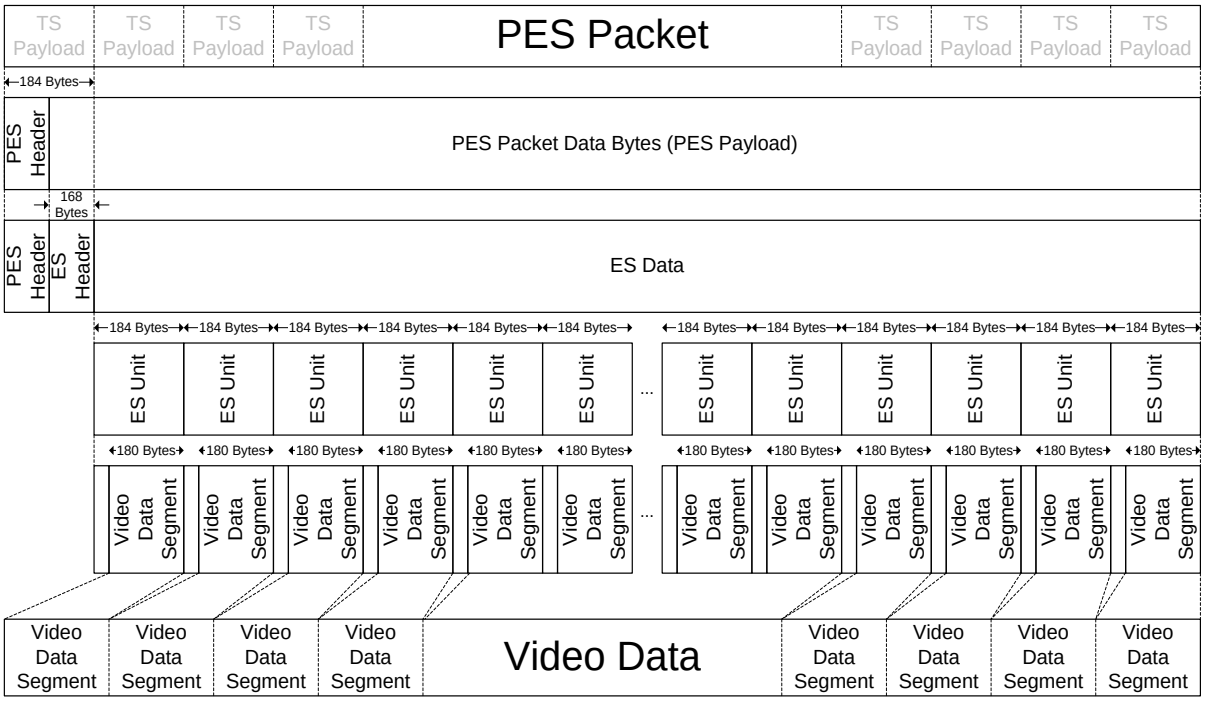


Figure 5 – Video Data

9.1.1.2.2 Video_data

The **video_data** is composed of atoms. The size of an atom depends on the **component size** and the **sample structure**.

If the **sample_structure** is 4:2:2 then the size of atom is 4 times the **component_size**.
If the **sample_structure** is 4:4:4 then the size of an atom is 3 times the **component_size**. If the **sample_structure** is 4:4:4:4 then the size of an atom is 4 times the c**omponent_size**.
If the **sample_structure** is 4:2:2:4 then the size of an atom is 6 times the c**omponent_size**.

Some common examples:

- For 4:2:2 10bit, atoms are 40 bits. (see Annex A)
- For 4:4:4 8bit, atoms are 24 bits.
- For 4:4:4 10bit, atoms are 30 bits.
- For 4:4:4 12bit, atoms are 36 bits.
- For 4:4:4:4 8bit, atoms are 32 bits.
- For 4:4:4:4 10bit, atoms are 40 bits.
- For 4:4:4:4 12bit, atoms are 48 bits.

For all of these examples the **video_data_segment** field contains a whole number of atoms.

The diagram in Figure 6 details how the atoms of **video_data** are put together to form a line of data and multiple lines put together into a frame of data. Table 5 details how the data is packaged to create a frame of data The raw data is packaged into video data segments and circulated over standard MPEG-2 Transport Stream.

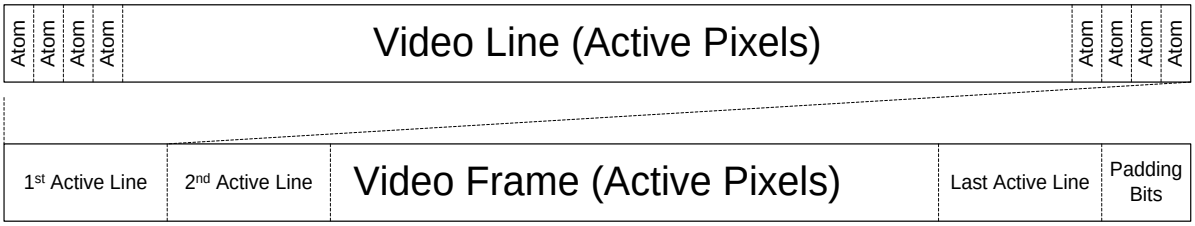


Figure 6 – Video Data packaged into video data segments

Table 5 – Video Data Structure

video_data() {		
for (line = 0; line < (total_vertical_size[0] + total_vertical_size[1]); line++)		
{		
if (((line >= first_extended_active_line[0]) && (line < (first_active_line[0] + active_vertical_size[0]))) || ((line >= first_extended_active_line[1]) && (line < (first_active_line[1] + active_vertical_size[1]))))		
{		
// Vertical Active		
for (pixel = first_active_pixel; pixel < (first_active_pixel + active_horizontal_size); pixel++)		
{		
pixel_data		bslbf
}		
if ((line == (first_active_line[0] + active_vertical_size[0] - 1)) || (line == (first_active_line[1] + active_vertical_size[1] - 1)))		
{		
for (i = 0; i < N; i++)		
{		
padding_bits		bslbf
}		
}		
}		
}		
}		

padding_bits:

Shall specify the bits that are used to achieve **transport_packet** alignment. At the last line of the vertical active region of each field, the final **video_data_segment** shall contain **padding_bits** in the precise quantity required to completely fill the final **transport_packet** of that line. The **padding_bits** may be any value. The **padding_bits** shall be ignored by the receiver.

pixel_data:

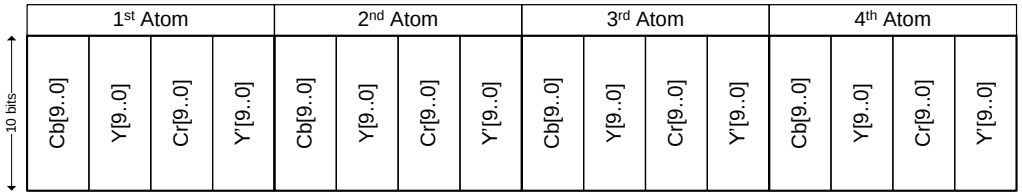
The pixels of the active picture are scanned horizontally from left to right across the horizontal active part of each line in the extended vertical active region. The **pixel_data** components shall be sent in the same order as its native serial format. For example, 4:2:2 SDI video components

are transmitted in the order Cb, Y, Cr, Y'. Likewise, the pixel_data would be sent as Cb, Y, Cr, Y'. The pixel_data shall be transmitted most significant bit first. For example, 10bit 4:2:2 SDI video shall be transmitted in this order: Cb[9], Cb[8], ... Cb[1], Cb[0], Y[9], Y[8], ... Y[1], Y[0], Cr[9], Cr[8], ... Cr[1], Cr[0], Y'[9], Y'[8], ... Y'[1], Y'[0].

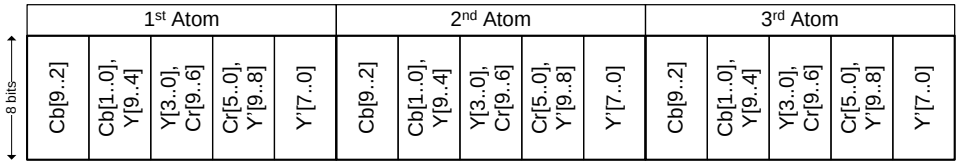
Annex A

4:2:2 10bit YCbCr Atoms (40 bits/atom)

SDI Format:



TS Format:



TS Format (detailed):

	1st Atom					2nd Atom				
	1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte
Most Significant (first) bit:	Cb[9]	Cb[1]	Y[3]	Cr[5]	Y'[7]	Cb[9]	Cb[1]	Y[3]	Cr[5]	Y'[7]
	Cb[8]	Cb[0]	Y[2]	Cr[4]	Y'[6]	Cb[8]	Cb[0]	Y[2]	Cr[4]	Y'[6]
	Cb[7]	Y[9]	Y[1]	Cr[3]	Y'[5]	Cb[7]	Y[9]	Y[1]	Cr[3]	Y'[5]
	Cb[6]	Y[8]	Y[0]	Cr[2]	Y'[4]	Cb[6]	Y[8]	Y[0]	Cr[2]	Y'[4]
	Cb[5]	Y[7]	Cr[9]	Cr[1]	Y'[3]	Cb[5]	Y[7]	Cr[9]	Cr[1]	Y'[3]
	Cb[4]	Y[6]	Cr[8]	Cr[0]	Y'[2]	Cb[4]	Y[6]	Cr[8]	Cr[0]	Y'[2]
	Cb[3]	Y[5]	Cr[7]	Y'[9]	Y'[1]	Cb[3]	Y[5]	Cr[7]	Y'[9]	Y'[1]
Least Significant (last) bit:	Cb[2]	Y[4]	Cr[6]	Y'[8]	Y'[0]	Cb[2]	Y[4]	Cr[6]	Y'[8]	Y'[0]