Universal Serial Bus Device Class Definition for Audio/Video Devices

(Developed pursuant to USB-IF IP agreement for Video Display and subsequently renamed as above)

AVFormat 1 – Video over Bulk

Release 1.0

December 07, 2011

Scope of This Release

This document is the Release 1.0 of this AVFormat 1 – Video over Bulk Definition.

Contributors

Jim Hunkins	AMD
Jason Hawken	AMD
Kenneth Ma	Broadcom
Hans van Antwerpen	Cypress Semiconductor
Jitendra Kulkarni	Cypress Semiconductor
Dan Ellis	DisplayLink
Trevor Hall	DisplayLink
Alec Cawley	DisplayLink
David Dolby	Dolby Labs
David Roh	Dolby Labs
Tsehao Lee	Grain Media
Pierre Bossart	Intel
David Harriman	Intel
Abdul R. Ismail (Chair)	Intel
J.P. Giacalone	Intel
Steve McGowan	Intel
Sridharan Ranganathan	Intel
Yoav Nissim	Jungo
Ygal Blum	Jungo
Max Basler	Littelfuse
Paul E. Berg	MCCI
Crisitian Chis	MCCI
John Garney	MCCI
Geert Knapen (Editor)	MCCI
Tomi Heinonen	Nokia Corporation
Richard Petrie	Nokia Corporation
Yoram Rimoni	Qualcomm, Inc.
Mark Bohm	SMSC
John Sisto	SMSC
Morgan Monks	SMSC
Bruno Paillard	Soft-dB

Will Harris	Texas Instruments
Grant Ley	Texas Instruments

Please send comments via electronic mail to av-chair@usb.org

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Scope

The *AVFormat 1 Definition v1.0* is part of the *AV Device Class Specification v1.0* release. The organization of all the documents that comprise the *AV Device Class Specification* is described in detail in the AV Device Class Overview & AVFunction Definition document.

This document describes the AV Formats that are used by an AVData FrameBuffer Entity inside an AVFunction to transmit or receive a stream of Video Data via the Control Bulk Pair. For detailed information on the AVData FrameBuffer Entity, refer to [AVFUNCTION]. For details on Audio-Only transport formats over isochronous endpoints, refer to [AVFORMAT_2].

The scope of this specification includes:

- Definitions of video commands and payloads supported by the USB AV Device Class
- Supported Video Payloads including:
  - Display resolutions
  - Compressed and uncompressed display pixel formats
- Mechanisms to support Video re-synchronization on the sink side
- Mechanisms to provide link protection to copy-protected Video data

1.2. Purpose

The purpose of this document is to provide all necessary information a designer needs to build AVFunctions that use the Video Data Formats described here on at least one of its AVData FrameBuffer Entities.

1.3. Related Documents

- [USB2.0] – Universal Serial Bus Specification, Revision 2.0, April 27, 2000 (referred to in this document as the USB 2.0 Specification) (available at: http://www.usb.org/developers/docs).
- [USB3.0] – Universal Serial Bus 3.0 Specification, Revision 1.0 (including errata and ECN's through May 1, 2011), June 6, 2011 (referred to in this document as the USB 3.0 Specification) (available at: http://www.usb.org/developers/docs.)
- [AUDIO1.0] – Universal Serial Bus Device Class Definition for Audio Devices, Release 1.0, March 18, 1998 (available at: http://www.usb.org/developers/devclass_docs).
- [FORMATS1.0] – Universal Serial Bus Device Class Definition for Audio Data Formats, Release 1.0, March 18, 1998 (available at: http://www.usb.org/developers/devclass_docs).
- [TERMTYPES1.0] – Universal Serial Bus Device Class Definition for Terminal Types, Release 1.0, March 18, 1998 (available at: http://www.usb.org/developers/devclass_docs).
- [AUDIO2.0] – Universal Serial Bus Device Class Definition for Audio Devices, Release 2.0, May 31, 2006 (available at: http://www.usb.org/developers/devclass_docs).
- [FORMATS2.0] – Universal Serial Bus Device Class Definition for Audio Data Formats, Release 2.0, May 31, 2006 (available at: http://www.usb.org/developers/devclass_docs).
- [TERMTYPES2.0] – Universal Serial Bus Device Class Definition for Terminal Types, Release 2.0, May 31, 2006 (available at: http://www.usb.org/developers/devclass_docs).
- [USBCS] – Universal Serial Bus Device Class Definition for Content Security Devices – Content Security Framework, Revision 2.0 (available at: http://www.usb.org/developers/devclass_docs).
- [USBCSM-5] – Universal Serial Bus Device Class Definition for Content Security Devices – Content Security Method 5 – High-bandwidth Digital Content Protection 2.1 (HDCP 2.1) Implementation, Revision 1.0 (available at: http://www.usb.org/developers/devclass_docs).
- USBECNIAD – USB Engineering Change Notice: Interface Association Descriptors (available at: http://www.usb.org/developers/docs).
- [USBIADDCC] – USB Interface Association Descriptor Device Class Code and Use Model, Revision 1.0, July 23, 2003 (available at: http://www.usb.org/developers/whitepapers).

- [USBLANGIDS] – Universal Serial Bus Language Identifiers (LANGIDs), Revision 1.0, March 29, 2000 (available at: http://www.usb.org/developers/docs).
- [AVFUNCTION] – Universal Serial Bus Device Class Definition for Audio/Video Devices – AV Device Class Overview & AVFunction Definition, Release 1.0, December 07, 2011 (available at: http://www.usb.org/developers/devclass_docs).
- [AVFORMAT_1] – Universal Serial Bus Device Class Definition for Audio/Video Devices – AVFormat 1 – Video over Bulk, Release 1.0, December 07, 2011 (available at: http://www.usb.org/developers/devclass_docs).
- [AVFORMAT_2] – Universal Serial Bus Device Class Definition for Audio/Video Devices – AVFormat 2 – Isochronous Audio, Release 1.0, December 07, 2011 (available at: http://www.usb.org/developers/devclass_docs).
- [AVFORMAT_3] – Universal Serial Bus Device Class Definition for Audio/Video Devices – AVFormat 3 – Uncompressed Full Frame Isochronous Video, Release 1.0, December 07, 2011 (available at: http://www.usb.org/developers/devclass_docs).
- [AVSCHEMA] – Available at: http://avschemas.usb.org/v1/avschema.xsd
- [BDP] – Universal Serial Bus Device Class Definition for Audio/Video Devices – Basic Device Profile, Release 1.0 (available at: http://www.usb.org/developers/whitepapers).
- [ANSIS1_11] – ANSI S1.11-2004 (R2009) standard (available at: http://www.ansi.org).
- [IEC11172_3] – MPEG-1 standard ISO/IEC 11172-3:1993 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 3: Audio (available at http://www.iec.ch).
- [IEC13818_1] – MPEG-2 standard ISO/IEC 13818:2000 Information technology – Generic coding of moving pictures and associated audio information – Part 1: Systems (available at http://www.iec.ch).
- [IEC13818_3] – MPEG-2 standard ISO/IEC 13818:1998 Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio" (available at http://www.iec.ch).
- [AC_3] – Digital Audio Compression Standard (AC-3, Enhanced AC-3), ETSI TS 102 366 (available at http://www.etsi.org).
- [IEEE_754] – ANSI/IEEE-754 floating-point standard (available at http://www.ieee.org).
- [IEC60958] – ISO/IEC 60958 International Standard: Digital Audio Interface and Annexes (available at: http://www.iec.ch).
- [IEC61937] – ISO/IEC 61937 standard (available at: http://www.iec.ch).
- [ETSI_TS_102_114] – ETSI Specification TS 102 114, "DTS Coherent Acoustics; Core and Extensions" (available at http://www.etsi.org).
- [HDCP2.1] – High-bandwidth Digital Content Protection System. Interface Independent Adaptation. Revision 2.1, July 18, 2011 (available from http://www.digital-cp.com).
- [MLP] – DVD Specifications for High Definition Video: MLP Reference Information.
- [IEC14496_3] – MPEG-4 Standard ISO/IEC 14496-3 – Information Technology – Coding of audio-visual objects – Part 3: Audio (available at http://www.iec.ch).
- [IEC14496_10] – MPEG-4 Standard ITU-T H.264 and ISO/IEC 14496-10:2004 – Information Technology – Coding of audio-visual objects – Part 10: Advanced Video Coding. Second Edition 2004-10-01 (available at http://www.iec.ch).
- [WMA] – Audio compression format from Microsoft. For technical and licensing information, contact Microsoft directly (http://www.microsoft.com/windows/windowsmedia/default.aspx).
- [HDMI] – The official High Definition Multimedia Interface website http://www.hdmi.org.
- [RFC5646] – Tags for Identifying Languages, September 2009 (available at http://www.rfc-editor.org/rfc/rfc5646.txt).
- [KHRONOS] – The Khronos Group, Open Standards for Media Authoring and Acceleration (available at http://www.khronos.org).
- [CEA-861-E] – A DTV Profile for Uncompressed High Speed Digital Interfaces, March 2008 (available at http://www.cea.org).
- [VESA] – Video Electronics Standards Association (available at http://www.vesa.org).
- [IEC10918_4] – JPEG Standard ISO/IEC 10918-4 – Information technology – Digital compression and coding of continuous-tone still images: Registration of JPEG profiles, SPIFF profiles, SPIFF tags, SPIFF colour spaces, APPn markers, SPIFF compression types and Registration Authorities (REGAUT). First Edition 1999-08-15 (available at http://www.iec.ch).

1.4. Terms and Abbreviations

This section defines terms used throughout this document. For additional terms that pertain to the Universal Serial Bus, see Chapter 2, "Terms and Abbreviations," in [USB2.0] and [USB3.0].

Table 1-1: Terms and Abbreviations

Term	Description
2D Video	A VideoStream consisting of a single VideoChannel, providing a monoscopic video experience.
3D2 Video	A VideoStream consisting of 2 separate VideoChannels, providing a stereoscopic 3-dimensional video experience. One VideoChannel is intended for the left eye, and the other VideoChannel is intended for the right eye.
3Dn Video	A VideoStream consisting of n separate VideoChannels, providing a multiscopic 3-dimensional video experience. This type of VideoStream usually requires a sophisticated lenticular system in front of the display to ensure that the viewer only sees 2 (one for the left eye, one for the right eye) out of the n channels at the same time. Depending on the position of the viewer with respect to the display, a different set of 2 VideoChannels is made visible to the viewer.
AAC	Advanced Audio Coding.
AC-3	Audio compression standard from Dolby Labs.
AudioBundle	AudioChannels are physically organized into AudioTracks, and AudioTracks are then organized into the AudioBundle where each AudioChannel has an AudioChannel Type associated with it. Very similar to the AudioCluster concept, but the physical characteristics of the audio stream, such as AudioFrame Rate and bit resolution, are retained in the AudioBundle.
(Logical) AudioChannel	A logical transport medium for a single audio channel. Makes abstraction of the physical Properties and formats of the connection. Is identified by AudioChannel Type.
AudioChannel Type	The spatial location of an AudioChannel. Uniquely identifies the AudioChannel. Examples are Front Left channel (FL), Back Right channel (BR), etc.
AudioCluster	The group of all the audio-only logical AudioChannels in an AVCluster.
AudioControl Property	A parameter of an AudioControl. Examples are Current, Next, Range Properties of a Volume Control.
AudioControl	A logical object that is used to manipulate a specific audio Property. Examples are Volume Control, Mute Control, etc.
AudioSample	The basic representation of an audio signal (one channel), sampled (digitized) at a specific moment in time.
AudioSlot	A collection of AudioSubSlots, each containing an AudioSample of a different physical audio channel, taken at the same moment in time.
AudioStream	A concatenation of a potentially very large number of AudioSlots ordered according to ascending time where the AudioSamples are formatted according to one of the Audio Formats described in this specification and where the AudioChannels are organized in an AudioBundle.
AudioStreamConfig (AudioStream Configuration)	An XML Description that provides all the information necessary to fully characterize an AudioStream. Includes an AudioBundle, AudioFrame, and AudioSample component.
AudioStreamConfigList	A list of AudioStreamConfig Descriptions used to indicate the different AudioStream Configurations an AVData Audio Streaming Interface supports.
AudioSubSlot	Holds a single AudioSample of a single audio channel.
AV Description Document (AVDD)	An XML-formatted document that provides a complete description of all the AV class-specific elements and features of the AVFunction.
AV Interface Association	A grouping of a single AVControl Interface, and zero or more AVData Streaming Interfaces that together constitute a complete interface to an AVFunction.
AV Profile	A set of limitations and restricting rules, bundled in a single document that applies to the AV Device Class Definition. A Profile defines a specific type of AVFunction that is guaranteed to interoperate with all Controllers that support that Profile.
AVBundle	A group of physical VideoChannels AudioChannels and MetadataChannels that carry tightly related (from a content perspective) synchronous video, audio, and metadata information over a USB pipe.
AVCluster	A group of logical VideoChannels AudioChannels and MetadataChannels that carry tightly related (from a content perspective) synchronous video, audio, and metadata information over an interconnect within the AVCore.

Term	Description
AVCore	That part of the AVFunction that contains most of the building blocks (Terminals, Units) that makes up most of the AVControl functionality of the AVFunction. Explicitly excludes the AVData Streaming Interfaces that are part of the AVFunction.
AVControl Interface	A USB interface used to access the AVControls inside an AVFunction.
AVData Entity	An addressable Entity representing a source of AV data flowing into or a sink for AV data flowing out of the AVCore (including AVData Streaming Interfaces).
AVData Streaming Interface	A USB interface used to transport AVStreams into or out of the AVFunction.
AVFunction	An independent part of a USB device that deals with AV-related functionality. Includes the AVCore and all AVData Entities.
AVHeader	The Header present in the AudioPayload or VideoPayload of a Command or Notify Message. Also, the Header present in every SIP.
AVStream	A generic name for either a VideoStream or an AudioStream. See VideoStream and AudioStream.
AVStream Configuration	A generic name for either a VideoStream Configuration or an AudioStream Configuration. See VideoStream Configuration and AudioStream Configuration.
BDP	Basic Device Profile. See [BDP]
BIB	Bus Interval Boundary ([USB3.0]).
CBP	Control Bulk Pair. A pair of one Bulk IN and one Bulk OUT endpoint in the AVControl Interface that is used for AV class-specific messaging.
CBP Idle Condition	A condition when an AVFunction determines that it will not have data available on its CBP Bulk IN pipe for a certain amount of time (AV_CBP_IDLE_TIME) and informs the Host of this condition to avoid having the Host continually issue IN tokens for that pipe.
Channel	A logical channel in an AVCluster. Can be a VideoChannel, AudioChannel, or MetadataChannel.
Controllee	The entity that is controlled by the Controller. The AVFunction always assumes this role.
Controller	The entity that controls the AVFunction. In this version of the specification, the USB Host assumes this role.
Control Sequence	A sequence of a Command and Response Message that is used to convey or retrieve one or more control parameters to or from the AVFunction.
Converter Unit (CU)	Provides the means to transform an incoming VideoTrack and/or AudioTrack into another VideoTrack and/or AudioTrack with different characteristics.
CUD	Converter Unit Description.
Description	Section of the AVDD (AV Description Document) that provides a detailed XML description of a specific Entity or other building block of the AVFunction architecture. Replaces the concept of the class-specific USB Descriptor.
DTS	Digital Theater Systems.
DUD	Metadata Unit Description.
DVD	Digital Versatile Disc.
Effect Unit (EU)	Provides advanced AV manipulation on the incoming logical AudioChannels.
Encoded Audio Bit Stream	A concatenation of a potentially very large number of encoded audio frames, ordered according to ascending time.
Encoded AudioFrame	A sequence of bits that contains an encoded representation of AudioSamples from one or more physical audio channels taken over a fixed period of time.
Encoded VideoFrame	A sequence of bits that contains an encoded representation of VideoSamples from one VideoFrame.
Entity	An addressable logical object inside an AVFunction.
EUD	Effect Unit Description.
Feature Unit (FU)	Provides basic AV manipulation on the incoming logical AV channels.
FUD	Feature Unit Description.
GraphicsEngine Entity (GEE)	An addressable Entity inside the AVCore representing a graphics subsystem of the AVFunction.
H.264	ITU name of ISO MPEG4 part10 Advanced Video Coding standard. See [IEC14496_10].
HE_AAC	High Efficiency Advanced Audio Coding

Term	Description
Header	A collection of SubHeaders present in a SIP, containing Extended Audio Format data.
Input Pin	A logical input connection to an Entity. Carries a single AVCluster.
Input Terminal (IT)	A receptacle for AV information flowing into the AVFunction.
Inter-VideoFrame	A VideoFrame composed of Intra-coded and Inter-coded MacroBlocks.
Intra-VideoFrame	A VideoFrame composed of Intra-coded MacroBlocks only.
ITD	Input Terminal Description.
MacroBlock	An elementary group of VideoSamples considered by an H.264 encoder.
MetadataBundle	The group of all the metadata-only physical channels in an AVBundle.
(Logical) MetadataChannel	A logical transport medium for a single metadata channel. Makes abstraction of the physical Properties and formats of the connection. Is identified by MetadataChannel Type.
MetadataChannel Type	The metadata type of a MetadataChannel. Uniquely identifies the MetadataChannel. Examples are Subtitle (SUB), Commentary (COM), etc.
MetadataCluster	The group of all the metadata-only logical channels in an AVCluster.
MetadataControl	A logical object that is used to manipulate a specific metadata Property. Examples are Font Control, Color Control, etc.
MetadataControl Property	Parameter of a MetadataControl. Examples are Current, Next, Range Properties of a Font Control.
MetadataTrack	A group of MetadataChannels in a MetadataCluster that is associated with a single program.
MetadataTrack Selector	A Control that allows indicating or selecting one MetadataTrack from the MetadataCluster. The selected MetadataTrack is called the Active MetadataTrack.
Metadata Unit (DU)	Provides basic manipulation on the incoming logical MetadataChannels.
Mixer Unit (MU)	Mixes a number of logical input channels into a number of logical output channels.
MLP	Meridian Lossless Packing
MotionVector	A couple of coordinates (x,y) defining an offset from the current MacroBlock position in the current VideoFrame into the previous VideoFrame. The current MacroBlock position is defined by two coordinates (xMB, yMB) defining the position of the upper left pixel of that MacroBlock.
MPEG	Moving Pictures Expert Group.
MUD	Mixer Unit Description.
NAL	Network Abstraction Layer, See [IEC14496_10]
NAL Unit	An integer number of bytes, preceded by a ony-byte NAL Header indicating the type of data in the NAL Unit.
OTD	Output Terminal Description.
Output Pin	A logical output connection to an Entity. Carries a single AVCluster.
Output Terminal (OT)	An outlet for AV information flowing out of the AVCore.
Processing Unit (PU)	Applies a predefined process to a number of logical input channels.
PUD	Processing Unit Description.
Router Unit (RU)	Provides the means to assemble an outgoing AVCluster from multiple incoming AVClusters.
RUD	Router Unit Description.
Selector Unit (SU)	Selects from a number of input AVClusters.
ServiceInterval	A grouping of USB (micro)frames that are related.
ServiceIntervalPacket	A packet that contains all the VideoSamples that are transferred over the bus during a ServiceInterval.
SI	ServiceInterval.
SIP	ServiceIntervalPacket.
SIPDescriptor	A 4-byte field present at the beginning of every SIP, providing information about the layout of the SIP. Only present when transporting Extended Audio Format Type I data.
SOF	Start of Frame ([USB2.0]).

Term	Description
Stereo 3D Video	A video stream consisting of 2 separate video channels, providing a stereoscopic 3-dimensional video experience. One video channel is intended for the left eye, and the other video channel is intended for the right eye.
SubHeader	A header containing additional information, related to the data in the SIP.
SUD	Selector Unit Description.
Terminal	A logical object inside an AVCore that represents a connection to the AVCore's outside world.
Transfer Delimiter	A unique token that indicates an interruption in an isochronous data packet stream. Can be either a zero-length data packet or the absence of an isochronous transfer in a certain USB (micro)frame.
Unit	A logical object inside an AVCore that represents a certain AV functionality.
VideoBundle	VideoChannels are physically organized into VideoTracks, and VideoTracks are then organized into the VideoBundle where each VideoVhannel has a VideoChannel Type associated with it. Very similar to the VideoCluster concept, but the physical characteristics of the video stream, such as VideoFrame Rate are retained in the VideoBundle.
(Logical) VideoChannel	A logical transport medium for a single video channel. Makes abstraction of the physical Properties and formats of the connection. Is identified by VideoChannel Type: i.e. observation location. Examples are Channels for left eye, right eye.
VideoChannel Type	The observation location of a VideoChannel. Uniquely identifies the VideoChannel. Examples are Channels for left eye (OL001), right eye (OL002).
VideoCluster	The group of all the video-only logical channels in an AVCluster.
VideoControl	A logical object that is used to manipulate a specific video Property. Examples are Contrast Control, Zoom Control, etc.
VideoControl Property	A parameter of a VideoControl. Examples are Current, Minimum, Maximum and Resolution Properties of a Brightness Control.
VideoFrame	One of the many still images which compose a complete moving picture or VideoSequence.
VideoParticle	The basic structure used to send video data over USB. Can be one VideoSample, two VideoSamples that share Chrominance information, or individual Luminance and Chrominance components of one VideoSample.
VideoPayload	All data bytes related to one VideoFrame. Includes NAL Headers, InfoBlock Headers, etc. The AVHeader is not part of the VideoPayload.
VideoSample	The basic representation of a video signal, sampled (digitized) at a specific moment in time.
VideoSequence	A sequence of closely related VideoFrames that together make up a movie or video clip.
VideoSlot	A collection of VideoSubSlots, each containing a VideoSample of a different physical video channel, taken at the same moment in time.
VideoStream	A concatenation of a potentially very large number of VideoSlots ordered according to ascending time where the VideoSamples are formatted according to one of the VideoFrame and VideoSample Formats described in this specification and where the video channels are organized in a VideoBundle.
VideoStreamConfig (VideoStream Configuration)	An XML Description that provides all the information necessary to fully characterize a VideoStream. Includes a VideoBundle, VideoFrame, and VideoSample component.
VideoStreamConfigList	A list of VideoStreamConfig Descriptions used to indicate the different VideoStream Configurations an AVData Entity supports.
VideoSubSlot	Holds a single VideoParticle.
Viewpoint	A position from which an event (program) is observed.
WMA	Windows Media Audio.
XML Descriptions	See Descriptions.

2. Introduction

The AVFormat 1 specification defines the following principles for transporting Video data:

- The Video transport structure is described in this document and is built on a VideoFrame by VideoFrame basis.
- The transport structure only supports one VideoTrack.
- Video data is transported using the following formats:
  - Uncompressed video, supporting various RGB and YCbCr color formats, including partial update support. These video stream formats are transported without any compression and therefore without any image quality reduction. These formats can be used when the full frame buffer is available in the source and enough bandwidth is available on the USB.
  - H.264 AVC, Intra or Intra+Inter frame coding, low latency compressed video, YCbCr or RGB color spaces. These formats can also be used when the full frame buffer is available in the source but not enough USB bandwidth is available to support an uncompressed format.
  - Images are transported packed over 1 (2D) or 2 video channels (3D2 or Stereoscopic 3D)

The transport principles described above can be summarized in the following data flow:

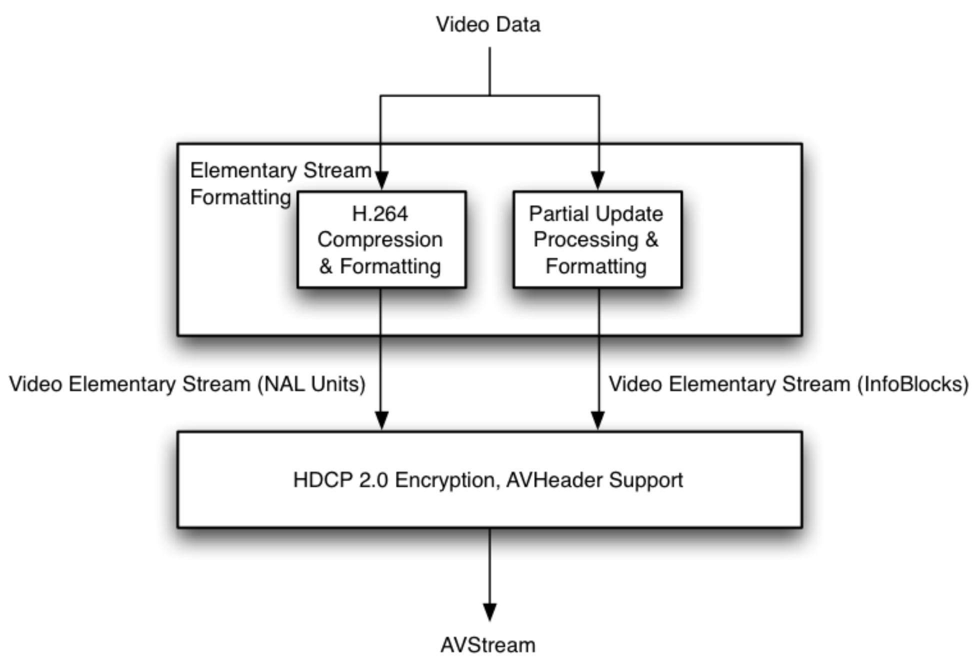


Figure 2-1: AVFormat 1 Data Flow

At the discovery of the available resources in the AVData FrameBuffer Entity, it is the responsibility of the USB AV Class Driver to select one of the VideoStream Configurations to be used during any subsequent data exchange with the AVData FrameBuffer Entity. The decision on which VideoStream Configuration to use is based on the functional features supported by the AVData FrameBuffer Entity as well as the available USB bandwidth. If the AVData FrameBuffer Entity does not support compression and USB bandwidth is limited (e.g., when using High-Speed instead of SuperSpeed), then the VideoFrame size and/or VideoFrame Rate can be adapted accordingly.

It is the responsibility of the source or the sink multi-media management application to define which format is being used for Video Data. Uncompressed or compressed video will be used for the locally constructed frame buffer available in the source.

3. Fundamentals

3.1. Video Transport

3.1.1. Uncompressed Video

When transported over USB, uncompressed VideoSamples are first segmented into VideoParticles and then packetized into an integer number of bytes, called the VideoSubSlot. VideoSubSlots are then further grouped together into VideoSlots and eventually grouped into the VideoBundle.

3.1.1.1. VideoParticle

The basic structure used to represent video data is the VideoParticle.

VideoSamples are packetized into VideoParticles depending on the VideoSample Format (see Section 3.3, "VideoSample Formats":

- For VideoSample Formats that do not involve color space compression, there is a one-to-one relationship between a VideoSample and a VideoParticle. One VideoSample equals one VideoParticle. All Type I VideoSample Formats (see Section 3.3.1.2.1) and the RGB565, RGB888, RGB8888, and IYU2 Type II VideoSample Formats (see Sections 3.3.2.2.1, 3.3.2.2.2, 3.3.2.2.3, and 3.3.2.2.4 respectively) use this (trivial) form of packetizing.
- For VideoSample Formats that involve a form of color space compression, some of the components that make up a VideoSample are shared among VideoSamples:
  - The YUY2, YVYU, and UYVY VideoSample Formats (see Sections 3.3.2.2.5, 3.3.2.2.6, and 3.3.2.2.7 respectively) share one U and V Chrominance sample between two Y Luminance samples; i.e. Y_i and Y_{i+1} share the same U_i and V_i Chrominance samples. One VideoParticle contains all the information needed to describe two VideoSamples: the two Luminance values and the shared U and V Chrominance values for the two VideoSamples.
  - The I420 and YV12 VideoSample Formats (see Sections 3.3.2.2.8 and 3.3.2.2.9 respectively) are planar formats (Luminance and Chrominance values are sent disjointly and chrominance values are shared among four VideoSamples) and therefore, Luminance and Chrominance values are packetized in separate VideoParticles (one VideoParticle contains one component of a VideoSample) so that multiple VideoParticles are required to describe one VideoSample.

In summary, a VideoParticle either contains one VideoSample, two VideoSamples or a Luminance or Chrominance component of a VideoSample.

3.1.1.2. VideoSubSlot

The basic structure used to transport VideoParticles over USB is the VideoSubSlot. All VideoSubSlots shall have the same VideoSubSlot size.

One VideoParticle always occupies one VideoSubSlot.

This specification limits the possible VideoSubSlot sizes ($subSlotSize$) to 1, 2, 3 or 4 bytes per VideoSubSlot. The VideoParticle that occupies the VideoSubSlot is represented using a number of bits ($bitResolution$) less than or equal to the total number of bits available in the VideoSubSlot:

$$bitResolution \leq subSlotSize * 8$$

In case $bitResolution < subSlotSize * 8$, the actual VideoParticle is always left-justified and padded with trailing zero bits to fill the entire VideoSubSlot.

3.1.1.3. VideoSlot

A VideoSlot consists of a collection of VideoSubSlots, arranged according to different interleaving schemes.

3.1.1.3.1. VideoParticle-interleaved VideoSlots

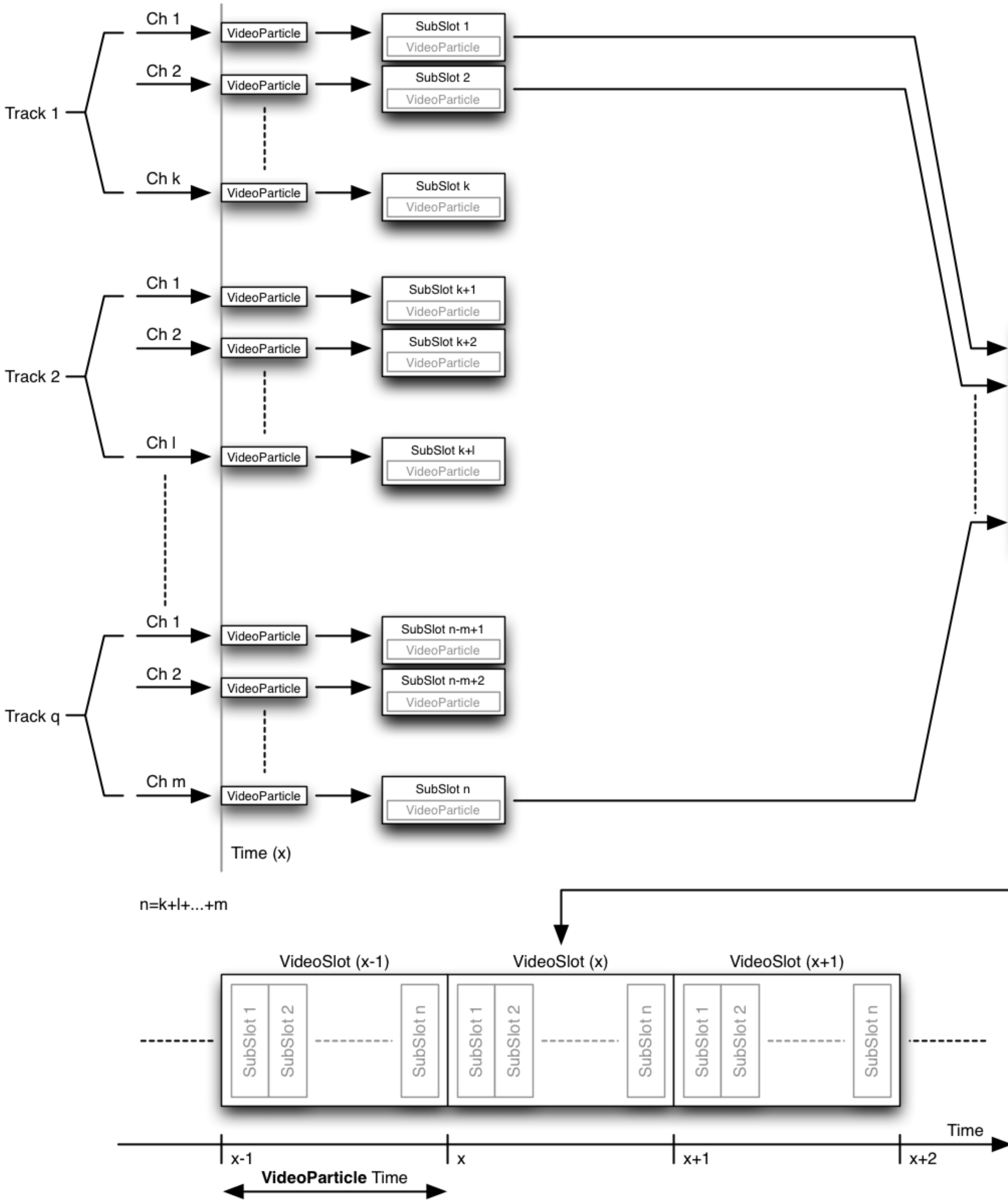


Figure 3-1: VideoParticle Stream

Each VideoSlot groups all the VideoSubslots containing the VideoParticle of a different physical video channel, taken at the same moment in time. The number of VideoSubSlots in a VideoSlot, *n*, equals the total number of video channels in the VideoBundle. The VideoSubSlots in the VideoSlot are ordered starting with the VideoSubslot of the first video channel in the first VideoTrack, followed by the VideoSubSlot of the second video channel in the first VideoTrack and so on until the VideoSubSlot of the last video channel of the last VideoTrack is reached.

There are currently no Frame Assemblies defined that use the VideoParticle-interleaved organization.

3.1.1.3.2. VideoLine-interleaved VideoSlots

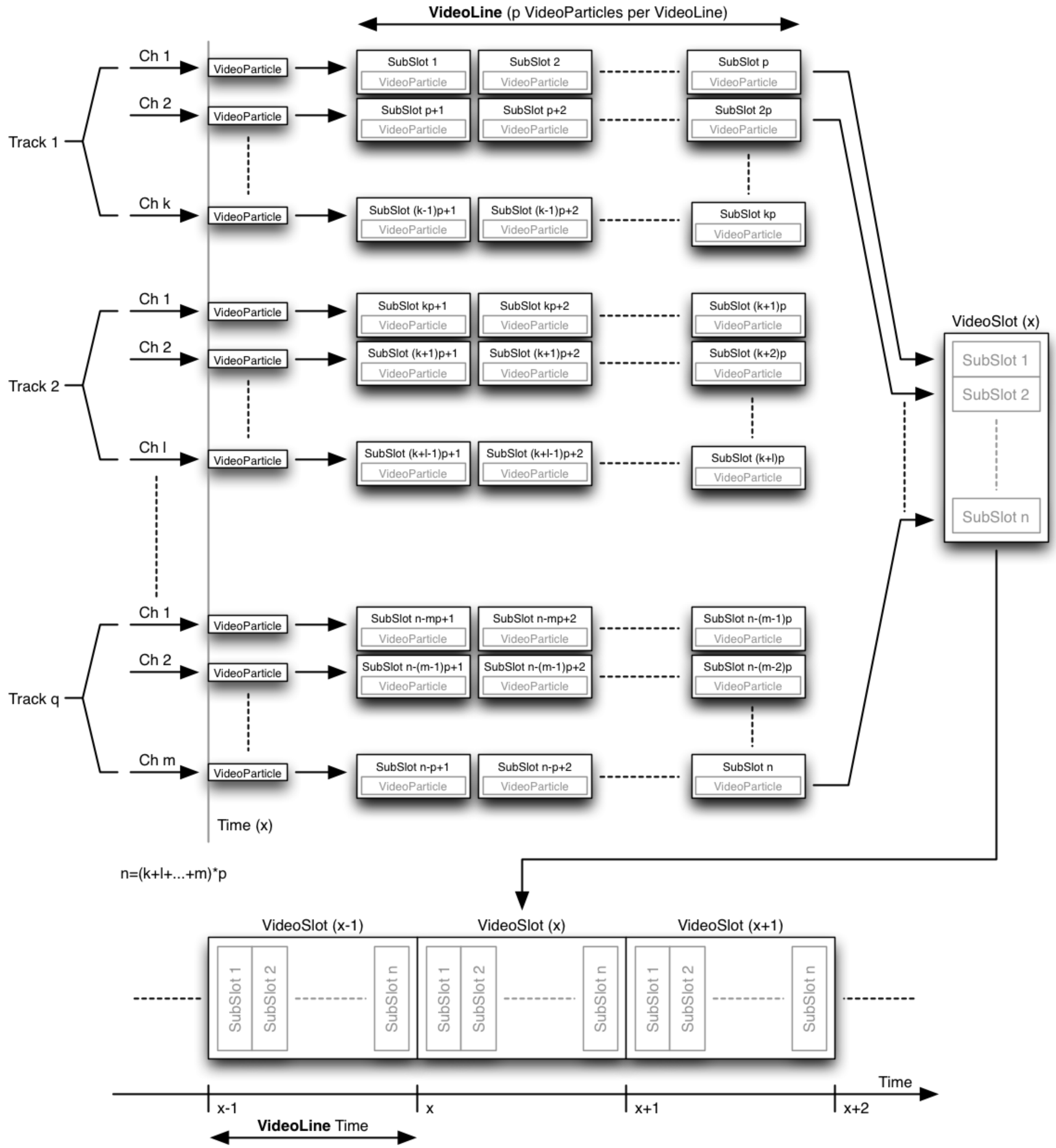


Figure 3-2: VideoLine Stream

Each VideoSlot groups all the VideoSubSlots containing all the VideoParticles that belong to the same VideoLine for all the different physical video channels present in the VideoBundle. The number of VideoSubSlots in a VideoSlot, n, equals the total number of video channels in the VideoBundle times the number of VideoParticles in a VideoLine. The VideoSubSlots in the VideoSlot are ordered starting with all the VideoSubslots of the VideoLine (in timed order) of the first video channel in the first VideoTrack, followed by all the VideoSubSlots of the same VideoLine of the second video channel in the first VideoTrack and so on up to all the VideoSubSlots of the same VideoLine of the last video channel of the last VideoTrack.

All currently defined 3D2 Side-by-Side Frame Assemblies use the VideoLine-interleaved organization.

3.1.1.3.3. VideoFrame-interleaved VideoSlots

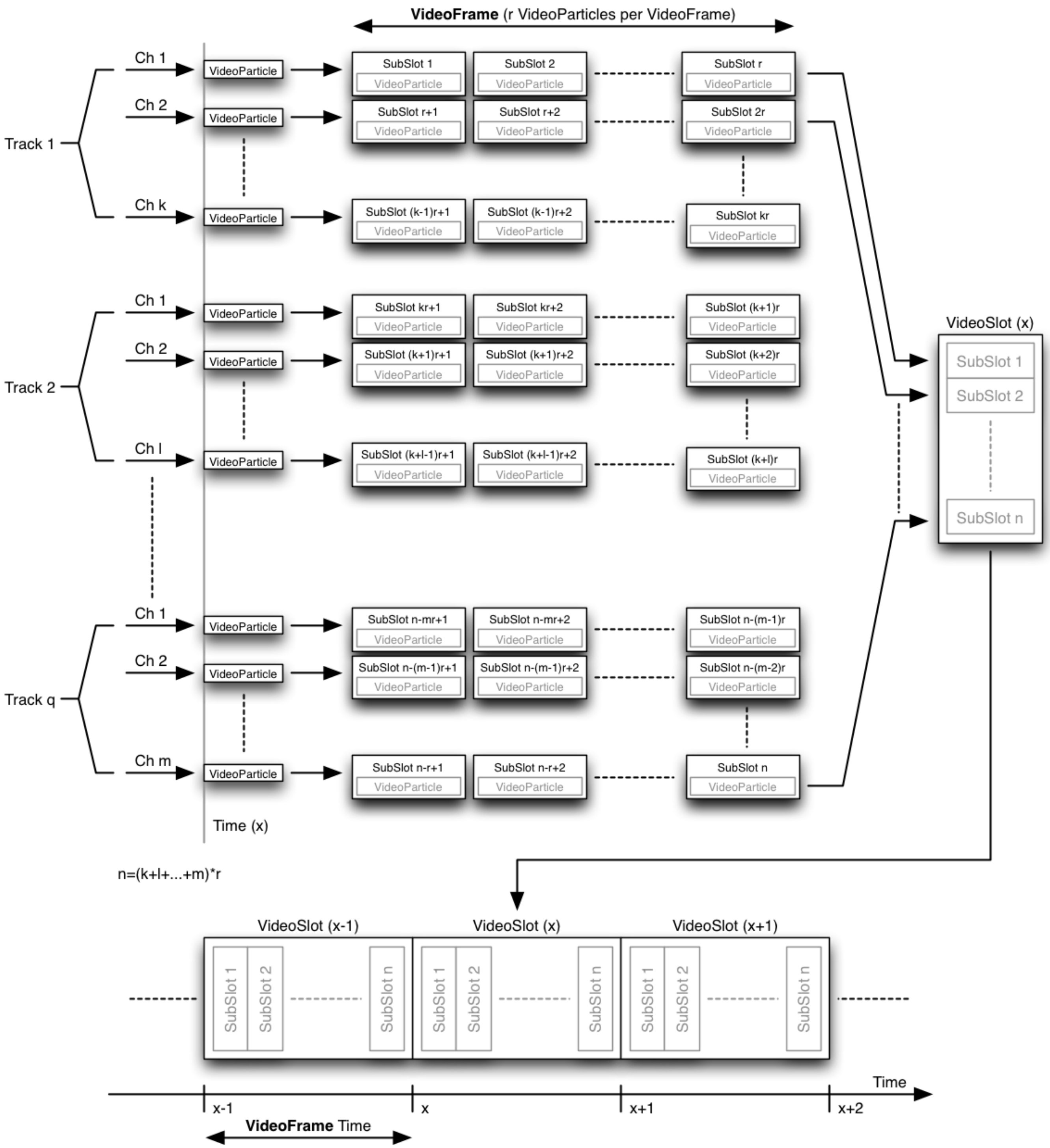


Figure 3-3: VideoFrame Stream

Each VideoSlot groups all the VideoSubSlots containing all the VideoParticles that belong to the same VideoFrame for all the different physical video channels present in the VideoBundle. The number of VideoSubSlots in a VideoSlot, n, equals the total number of video channels in the VideoBundle times the number of VideoParticles in a VideoFrame. The VideoSubSlots in the VideoSlot are ordered starting with all the VideoSubslots of the VideoFrame (in timed order) of the first video channel in the first VideoTrack, followed by all the VideoSubSlots of the same VideoFrame of the second video channel in the first VideoTrack and so on up to all the VideoSubSlots of the same VideoFrame of the last video channel of the last VideoTrack.

All currently defined 3D2 Top-and-Bottom Frame Assemblies and the 2D Frame Assembly use the VideoFrame-interleaved organization.

3.1.2. Compressed Video

When transporting compressed video over USB, only Full Frame compression is considered at this time and the native output of the encoder is encapsulated into NAL Units, packetized and sent over USB as described in subsequent sections of this specification.

3.1.3. VideoBundle

Video channels are transmitted over USB in a VideoBundle. The VideoBundle concept is very similar to the VideoCluster concept as defined in the USB AVCore Definition, except that a VideoBundle retains all of the physical aspects of the video stream, such as VideoSample Rate, bits per VideoSample, etc. Like a VideoCluster, a VideoBundle consists of a number of VideoTracks and each VideoTrack in turn consists of a number of physical video channels. The following figure illustrates the concept.

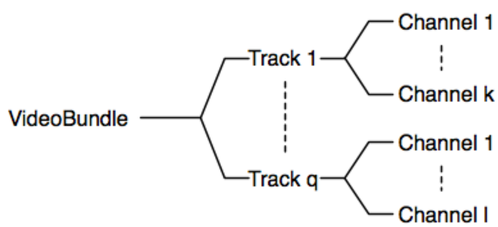


Figure 3-4: VideoBundle

A VideoBundle is a concatenation of a potentially very large number of VideoSlots, ordered according to ascending time. The VideoBundle is sent over USB either as a number of discrete video channels (uncompressed video) or as an encoded bitstream containing all the original video channels in a H.264 coded form. VideoBundles are packetized when transported over USB as described in Section 4, "Video Data Format".

3.1.3.1. VideoBundle Description

The VideoBundle Description is a hierarchical component of the VideoStreams Configuration Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.2. VideoFrame

A VideoFrame is a single picture that is typically shown as part of a larger sequence. Many single VideoFrames are run in succession to produce what appears to be a seamless piece of film or movie. The following sections define a number of concepts related to the VideoFrame.

3.2.1. VideoFrame Rate

The VideoFrame Rates the frequency (rate), expressed in Hertz (Hz), at which an imaging device produces unique consecutive images called VideoFrames. The term applies equally well to computer graphics, video cameras, film cameras, and motion capture systems.

All video-related frequency values shall tolerate at least ±5000 PPM inaccuracy to accommodate sampling clock inaccuracies.

3.2.2. VideoFrame Coordinates

The following VideoFrame Coordinate definitions are used in this document:

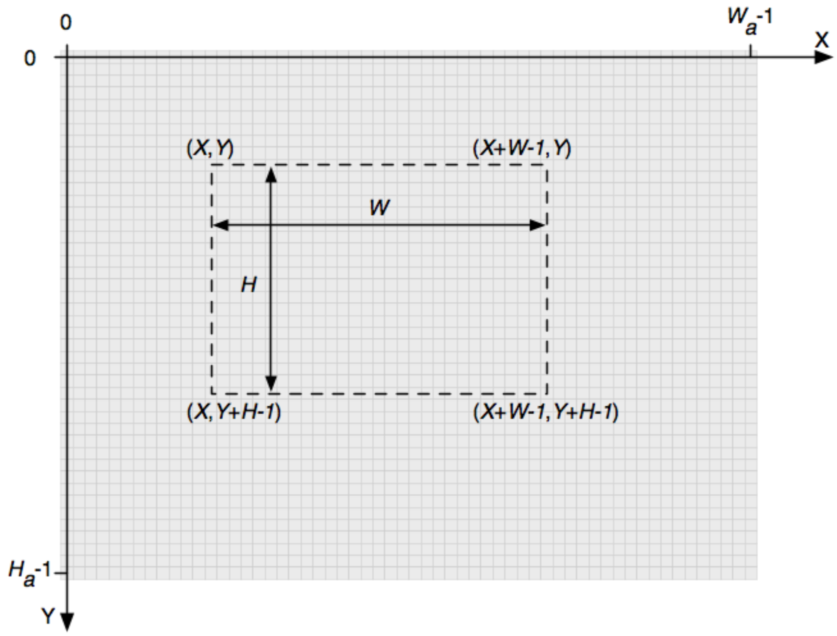


Figure 3-5: VideoFrame Coordinates

- The Width of the VideoFrame in VideoSamples (pixels): W_a
- The Height of the VideoFrame in VideoLines: H_a
- A particular VideoSample in the VideoFrame, indicated by:
  - Its horizontal coordinate: X
  - Its vertical coordinate: Y
- The Origin (0,0) (upper left VideoSample(0) of the VideoFrame) is on the intersection of the first VideoLine (VideoLine 1) and the first VideoColumn (VideoColumn 1) of a VideoSample coordinate system oriented as defined in the picture above.

  Note: VideoLine and VideoColumn numbers are one-based.

3.2.2.1. Subregions

Within this coordinate system, rectangular subregions may be defined as follows:

- Horizontal coordinate of the upper left VideoSample of the subregion in the system above, in VideoSamples: X
- Vertical coordinate of the upper left VideoSample of the subregion in the system above, in VideoLines: Y
- Width of a specific rectangular area within the VideoFrame, in VideoSamples: W
- Height of a specific rectangular area within the VideoFrame, in VideoLines: H

The values X, Y, W, H, W_a, and H_a shall always be multiples of two (even numbers).

Subregions may overlap. Subregions shall be updated in the order they appear in the VideoPayload so that a deterministic result is achieved after all updates are applied.

3.2.3. VideoFrame Spatial Layout and Timing Aspects

The following sections describe the VideoFrame Spatial Layouts for both 2D and 3D2 Frame Organizations (see Section 3.2.4, “VideoFrame Organization” for more information about Frame Organization)

3.2.3.1. 2D VideoFrame Spatial Layout

The following figure describes a typical VideoFrame Spatial Layout for the 2D (one video channel) case:

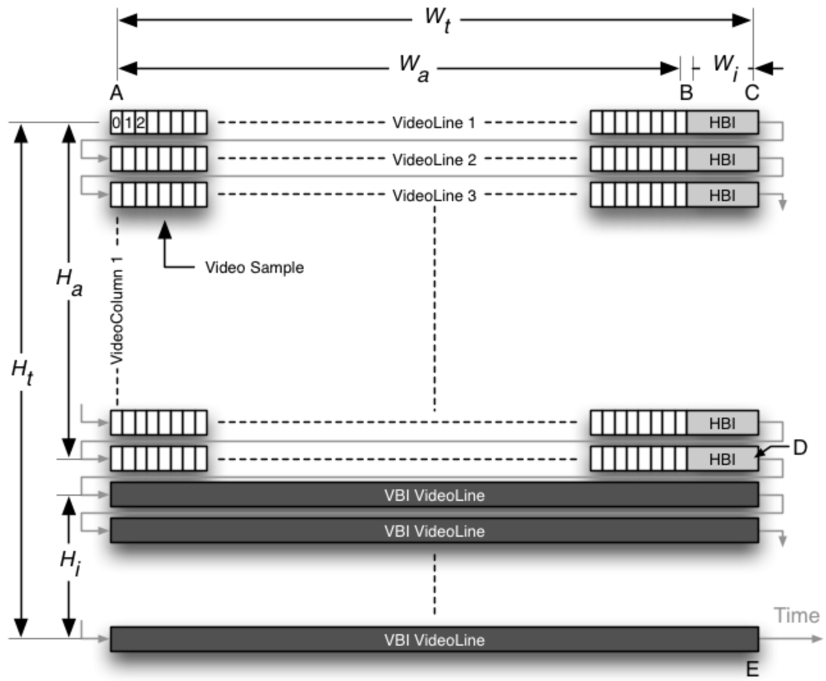


Figure 3-6: 2D VideoFrame Spatial Layout and Timings

A VideoFrame consists of a number of horizontal progressive VideoLines where each one has an active part (for example, the segment between A and B in Figure 3-6), containing the actual VideoSamples (W_a) that make up the VideoLine, and an inactive part, (W_i) (for example, the segment between B and C in Figure 3-6), called the Horizontal Blanking Interval or HBI. The HBI does not contain any VideoSamples. Following the active part of the VideoFrame (H_a, i.e. the part that contains VideoLines with actual VideoSamples, the segment between A and D in Figure 3-6), there are a number of inactive VideoLines (H_i, i.e. the segment between D and E in Figure 3-6) that together constitute the Vertical Blanking Interval or VBI. The VBI does not contain any VideoSamples.

The time difference between B and A is called the Active Line Time, T_{La}, the time difference between C and B is called the Inactive Line Time, T_{Li}, and the time difference between C and A is called the Total Line Time, $T_{Lt} = T_{La} + T_{Li}$.

The time difference between D and A in Figure 3-6 is called the Active VideoFrame Time, T_a, the time difference between E and D is called the Inactive VideoFrame Time, T_i, and the time difference between E and A is called the VideoFrame Time, $T = T_a + T_i$.

VideoSamples are numbered sequentially starting at the top left with VideoSample (0) and ending at the bottom right with VideoSample $(N - 1)$. If W_a is the horizontal resolution (number of VideoSamples in the active part of a VideoLine) and $H_a$is the vertical resolution of the VideoFrame (number of active VideoLines in the VideoFrame) then the total number of VideoSamples in the VideoFrame is:

$$N = (W_a \times H_a)$$

Interlaced formats are not supported.

3.2.3.2. 3D2 VideoFrame Layout

When transporting 3D2 video data, consideration has been given to aligning the 3D2 Layouts, VideoFrame Organizations, and video timings so that they match as closely as possible the 2D Layout, VideoFrame Organization and video timings.

3.2.3.2.1. 3D2 Half Resolution VideoFrame Spatial Layout

All Half Resolution 3D2 VideoFrame Organizations (see Section 3.2.4, “VideoFrame Organization”) use a VideoFrame Spatial Layout that is identical to the Full Resolution 2D VideoFrame Spatial Layout. This is accomplished by subsampling the original Full Resolution Left and Right VideoFrame by a factor of 2, either by dropping every other

VideoLine or every other VideoColumn of the original Left and Right VideoFrames and then recombining the resulting half resolution VideoFrames back into a Full Resolution resulting VideoFrame that has the exact same Spatial Layout, VideoFrame Organization and video timings as one of the original (Left or Right) VideoFrame. See Figure 3-6, “2D VideoFrame Spatial Layout”. All of the Half Resolution Frame Organizations use this Spatial Layout.

3.2.3.2.2. 3D2 Full Resolution VideoFrame Spatial Layout

This specification supports only one Full Resolution Frame Organization (3D2FP, 3D2 Frame Packing – See Section 3.2.4, “VideoFrame Organization”).

In this case, the total amount of actual video data per 3D2 VideoFrame doubles since a Full Resolution Left and Right VideoFrame are combined by packing them together and no subsampling is performed.

To keep the overall VideoFrame timing intact (3D2 VideoFrame Time = Left VideoFrame Time = Right VideoFrame Time), the VideoSample clock is doubled, compared to the 2D Full Resolution case. The area corresponding to the Vertical Blanking Interval of the Left VideoFrame is renamed to Active Space. The Active Space contains exactly the same amount of VideoLines, as does the Vertical Blanking Interval. The active part of the VideoLines in the Active Space shall contain VideoSamples that all have the same (zero) value.

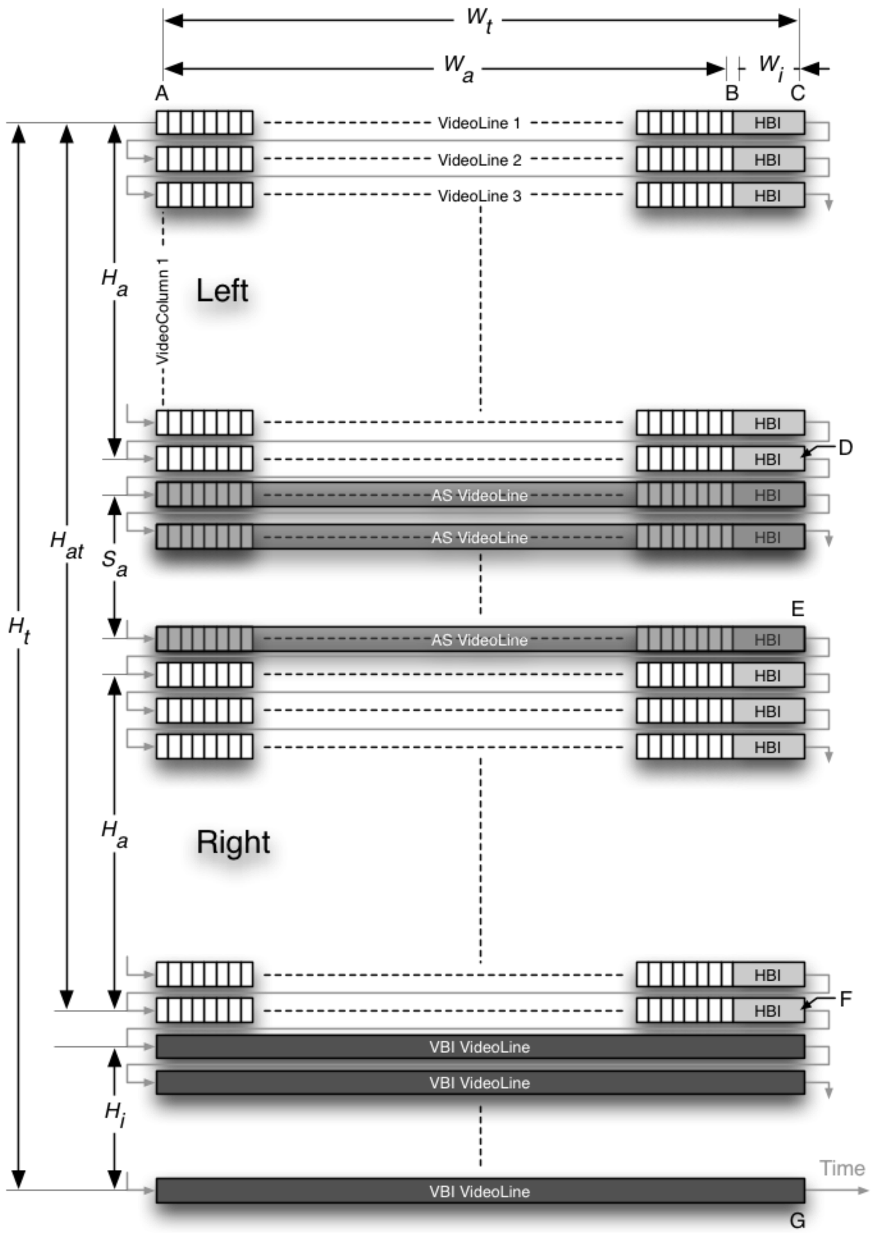


Figure 3-7: 3D2FP VideoFrame Spatial Layout and Timings

In this figure, the horizontal time scale is twice as large as it is in Figure 3-6 so that it takes exactly the same time in both cases from the beginning of the VideoFrame (A) to the end of the VideoFrame (G). As a consequence, all other time values are divided by a factor of 2. The following equations hold:

$$T_{La}^{3D2FP} = \frac{1}{2} * T_{La}^{2D}$$

$$T_{Li}^{3D2FP} = \frac{1}{2} * T_{Li}^{2D}$$

$$T_{Lt}^{3D2FP} = \frac{1}{2} * T_{Lt}^{2D}$$

$$T_{a}^{3D2FP} = \frac{1}{2} * T_{a}^{2D} + \frac{1}{2} * T_{i}^{2D} + \frac{1}{2} * T_{a}^{2D}$$

$$T_i^{3D2FP} = \frac{1}{2} * T_i^{2D}$$

$$T^{3D2FP} = T_a^{3D2FP} + T_i^{3D2FP} = T^{2D}$$

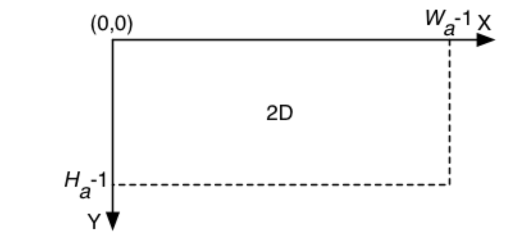
3.2.4. VideoFrame Organization

The AVFormat 1 Definition supports 2D VideoFrames (one video channel) as well as Stereoscopic 3D2 VideoFrames (two video channels) consisting of the assembly of a Left and a Right VideoFrame. The following tables define how the Left and Right VideoFrame can be assembled in different Frame Organizations. Each of the VideoFrame Organizations is assigned a unique VideoFrame Organization ID (FOID) and a unique VideoFrame Organization Legacy View Code (FOLVC) for use while in Legacy View as indicated in the tables. The FOLVC shall also be used as the Frame Packing Arrangement ID (FPA-ID) for use in the H.264 SEI-FPA Message (see Section 3.4.2.3, “Frame Packing Arrangement”). All VideoFrames are progressive scan. Interlaced formats are not supported.

3.2.4.1. 2D

The following table describes the 2D (one video channel) Full Resolution Frame Organization.

Table 3-1: 2D

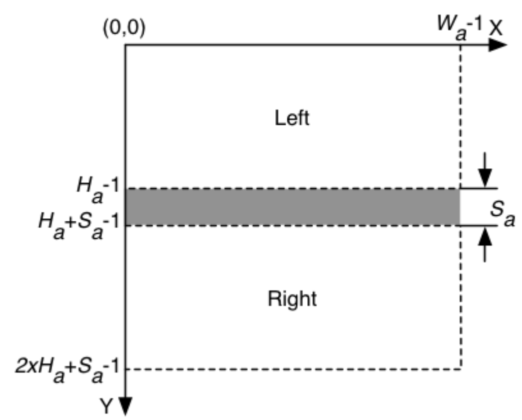


FOID	FOLVC	VideoFrame Organization
2D	0x0001	2D, Full Resolution.

3.2.4.2. 3D2FP

The following table describes the 3D2 (two video channels) Full Resolution Frame Organization. Left and Right VideoFrame are packed into a single resulting VideoFrame. When transported over USB, the VideoSamples in the Active Space (S_a) are either included in the VideoPayload or they are suppressed. When included, they shall all have zero values.

Table 3-2: 3D2FP

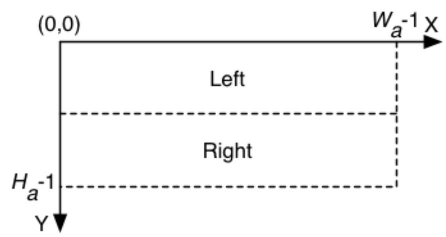


FOID	FOLVC	VideoFrame Organization
3D2FP	0x0002	3D, Full Resolution, Frame Packing. The Active Space (S_a) is either included or suppressed when the VideoFrame is transported over USB.

3.2.4.3. 3D2TAB

The following table describes the 3D2 (two video channels) Half Resolution Top and Bottom Frame Organization. Left and Right VideoFrames are subsampled by discarding every other VideoLine in both the Left and the Right VideoFrame and the subsampled VideoFrames are recombined into a single resulting VideoFrame.

Table 3-3: 3D2TAB

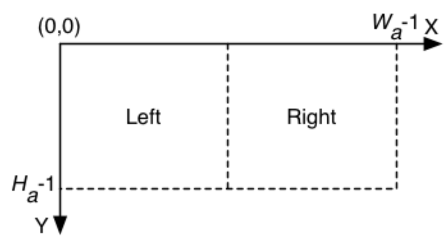


FOID	FOLVC	VideoFrame Organization
3D2TAB	0x0003	3D, Half Vertical Resolution, Odd Line Subsampling for Left and Right VideoFrame.

3.2.4.4. 3D2SHxx

The following table describes the 3D2 (two video channels) Half Resolution Side-by-Side Frame Organizations. Left and Right VideoFrames are subsampled by discarding every other Column in both the Left and the Right VideoFrame and the subsampled VideoFrames are recombined into a single resulting VideoFrame. Four combinations are possible as indicated in the table.

Table 3-4: 3D2SHxx



FOID	FOLVC	VideoFrame Organization
3D2SHOE	0x0004	3D, Half Horizontal Resolution. Odd Column Subsampling for Left and Even Column Subsampling for Right VideoFrame.
3D2SHOO	0x0005	3D, Half Horizontal Resolution. Odd Column Subsampling for Left and Odd Column Subsampling for Right VideoFrame.
3D2SHEO	0x0006	3D, Half Horizontal Resolution. Even Column Subsampling for Left and Odd Column Subsampling for Right VideoFrame.
3D2SHEE	0x0007	3D, Half Horizontal Resolution. Even Column Subsampling for Left and Even Column Subsampling for Right VideoFrame.

Note: “Odd” and “Even” in the tables above indicate the VideoLines or VideoColumns that are *retained* for creating the sub-sampled regions of the Left and Right views. Columns and Lines are numbered as indicated in Figure 3-6, “2D VideoFrame Spatial Layout and Timings”, starting with VideoLine 1 (uppermost line) or VideoColumn 1 (leftmost column). For example, the 3D2SHOE Frame Organization retains the odd VideoColumns 1, 3, 5, etc. from the original Left VideoFrame, dropping the even VideoColumns 2, 4, 6, etc. to create the subsampled Left View and it retains the even VideoColumns 2, 4, 6, etc. from the original Right VideoFrame, dropping the odd VideoColumns 1, 3, 5, etc. to create the subsampled Right View as illustrated in the following figure.

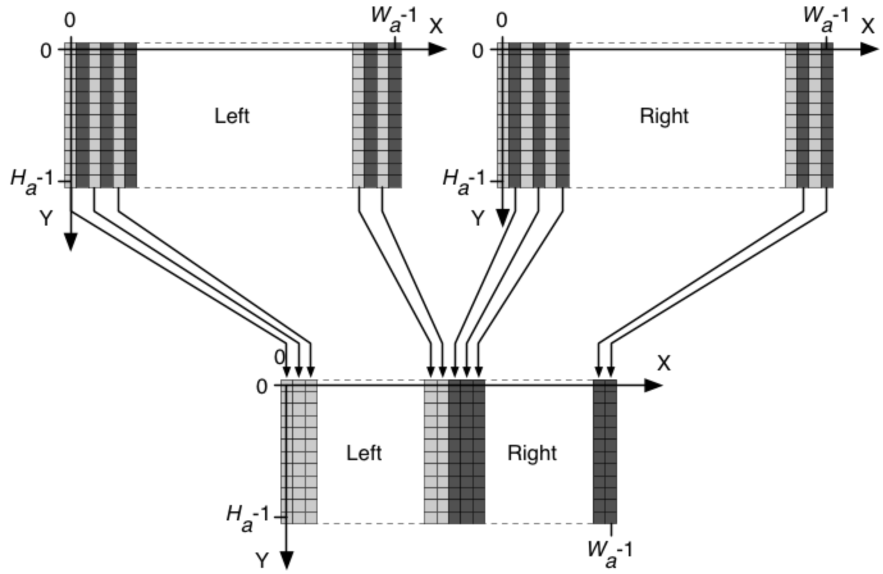


Figure 3-8: 3D2SHOE Frame Organization Subsampling

3.2.4.5. VideoFrame Organization Description

The VideoFrame Organization Description is a hierarchical component of the VideoStreams Configuration Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.2.5. VideoFrame Formats

This specification supports a wide variety of VideoFrame formats, either predefined by this specification or vendor-defined. The VideoFrame Rate, together with the VideoFrame format ID (VFID), fully identifies each VideoFrame format.

The following set of parameters fully defines the (2D or 3D2) VideoFrame:

- F: VideoFrame Rate (in Hz)
- A: The aspect ratio of the VideoFrame
- W_a: Active VideoFrame Width (in VideoSamples)
- H_a: Active VideoFrame Height (in VideoLines)

  Note: for the 3D2FP Frame Organization, H_a indicates the Active VideoFrame Height for both the Left and Right Full Resolution VideoFrame.

- W_t: Total VideoFrame Width (in VideoSamples)
- H_t: Total VideoFrame Height (in VideoLines)
- F_{vs}: VideoSample Rate (pixelclock) (in MHz) – for informational purposes only

For the 3D2FP Full Resolution Frame Organization, the following secondary parameters can be calculated from the previous parameters:

- S_a: Indicates the size of the Active Space (in VideoLines): $S_a = \frac{1}{2} * H_t - H_a$
- H_{at}: Total Active VideoFrame Height (in VideoLines): $H_{at} = 2 * H_a + S_a$

Note that the VideoFrame Rate is an integral part of the VideoFrame Format definition. To fully identify one of the predefined VideoFrame Formats, both the VideoFrame Rate and the VideoFrame ID are required.

This specification defines a number of common predefined VideoFrame formats. Refer to Appendix A, "VideoFrame Format Dimensions and Timings" for a complete list.

3.2.5.1. VideoFrame Format Description

The VideoFrame Format Description is a hierarchical component of the VideoStreams Configuration Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.3. VideoSample Formats

There are currently two VideoSample Format Types defined by this specification:

- VideoSample Format Type I
- VideoSample Format Type II

A unique VideoSample format ID (VSID) identifies each VideoSample Format. In addition, a unique VideoSample Format Legacy View Code (VSLVC) is defined for use while in Legacy View.

3.3.1. Type I Format

VideoSample format Type I groups the VideoSample formats that use one symbol per VideoSample. These formats are usually called RAW data formats. The following sections describe the VideoSample formats that belong to Type I.

3.3.1.1. Type I Format Type Description

The VideoSample Type I Description is a hierarchical component of the VideoStreams Configuration Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.3.1.2. Type I Supported Formats

3.3.1.2.1. RAW Format

The RAW VideoSample format uses a single symbol to represent a VideoSample. To generate a color picture, the VideoSamples need to be further processed, starting from the Bayer pattern (Demosaicing, White Balance, Contrast, Color Saturation, Sharpening, …)

This specification supports a generic RAW format, ranging from 1 up to 32 bits per VideoSample. Each VideoSample is transported in a VideoSubSlot as described above.

The $bitResolution$ and $subSlotSize$ values shall always obey the following equation:

$$subSlotSize = \big((bitResolution - 1)\ DIV\ 8\big) + 1$$

The VSID for the RAW format shall be set to "RAW". The VSLVC shall be set to 0x0001.

3.3.2. Type II Format

VideoSample Type II groups the VideoSample formats that use multiple symbols per VideoSample. The Type II formats are also known as Color Spaces. Typical examples are the RGB formats and the YUV formats. The following sections describe the VideoSample Formats that belong to Type II.

3.3.2.1. Type II Format Type Description

The VideoSample Type II Description is a hierarchical component of the VideoStreams Configuration Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.3.2.2. Type II Supported Formats

The Type II formats use multiple symbols to represent a VideoSample. For some of the Formats, the VideoSample concept does not really map well onto byte boundaries since Chrominance values may be shared among multiple VideoSamples and may even be arranged completely separate from the VideoSample Luminance values (planar arrangements). Therefore, the concept of the VideoParticle was introduced (see Section 3.1.1.1, "VideoParticle". In all cases there is never any padding involved when VideoSamples are first packetized into VideoParticles and then placed into VideoSubSlots since all Luminance and Chrominance values are always byte-aligned.

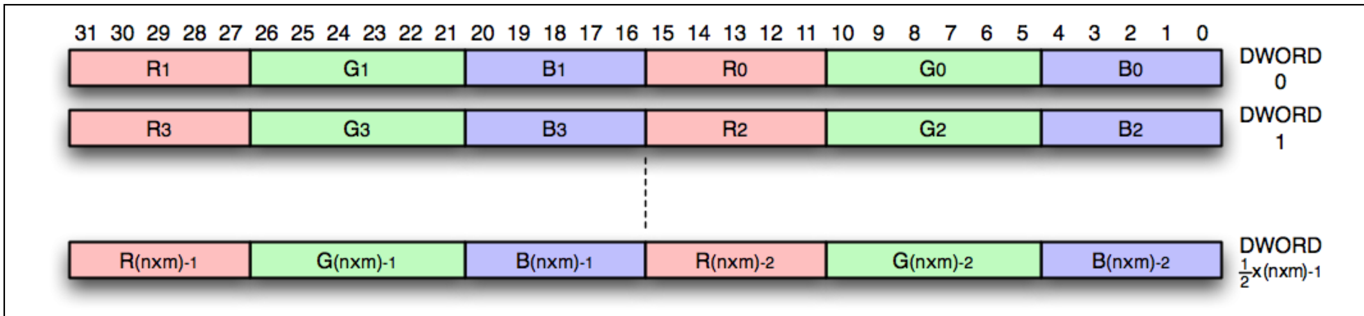
$bitResolution$ and $subSlotSize$ are implied by the VSID and are not reported in the AVDD.

The following Sections specify the VideoSample Formats allowed for use in this specification. The following notation is used in the tables:

- The **VSID** value is used to uniquely identify the VideoSample Format for the purposes of this specification.
- The **VSLVC** value is used to uniquely identify the VideoSample Format while in Legacy View.
- The **FourCC** field indicates the four-character code by which the VideoSample Format (Color Space) is registered at www.fourcc.org.
- The **Type** field specifies the type of the VideoSample Format: RGB, YCbCr, or YUV Planar.
- The **bitResolution** field indicates the number of bits per VideoParticle ($bitResolution$).
- The **subSlotSize** field indicates the number of bytes per VideoSubSlot ($subSlotSize$).
- The **bits/VS** field indicates the (average) number of bits used for one VideoSample (informative).
- In the figures, n is the horizontal resolution W_s and m is the vertical resolution H_s of the VideoFrame.

3.3.2.2.1. RGB565

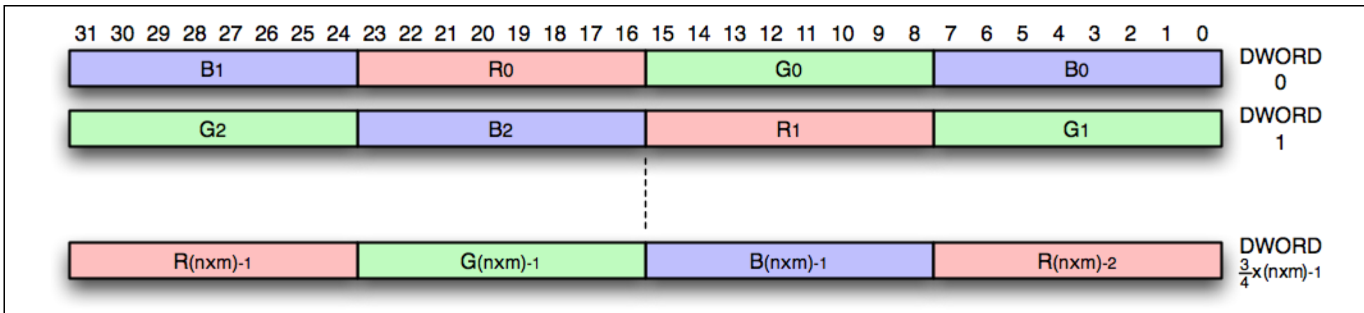
Table 3-5: RGB565 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
RGB565	0x0002	BI_RGB	RGB	16	2	16

3.3.2.2.2. RGB888

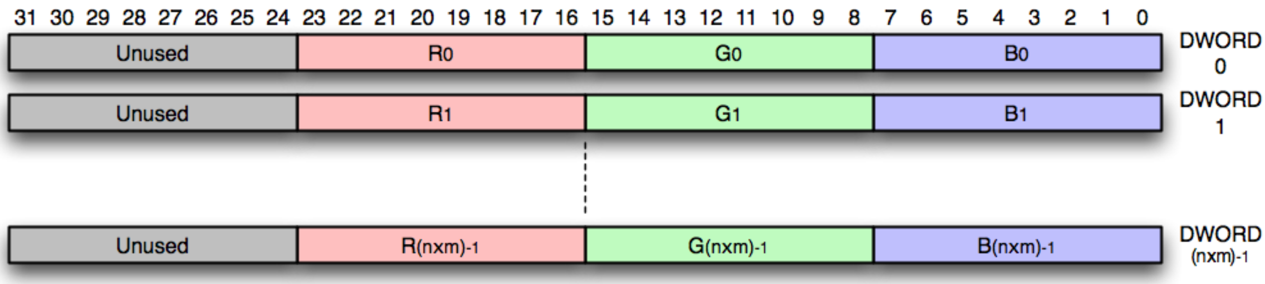
Table 3-6: RGB888 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
RGB888	0x0003	BI_RGB	RGB	24	3	24

3.3.2.2.3. RGB8888

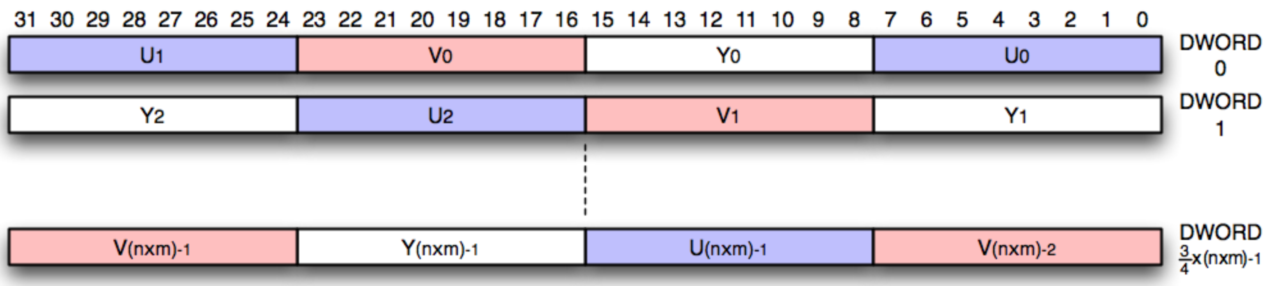
Table 3-7: RGB8888 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
RGB8888	0x0004	BI_RGB	RGB	32	4	32

3.3.2.2.4. IYU2

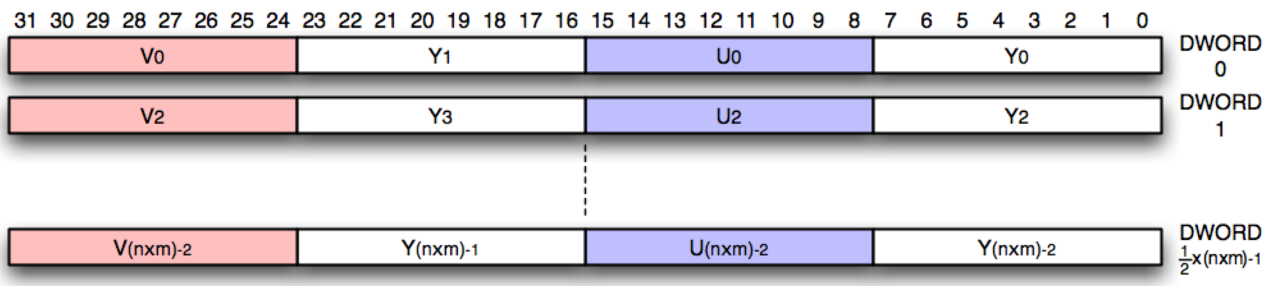
Table 3-8: IYU2 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
IYU2	0x0005	IYU2	YCbCr444	24	3	24

3.3.2.2.5. YUY2

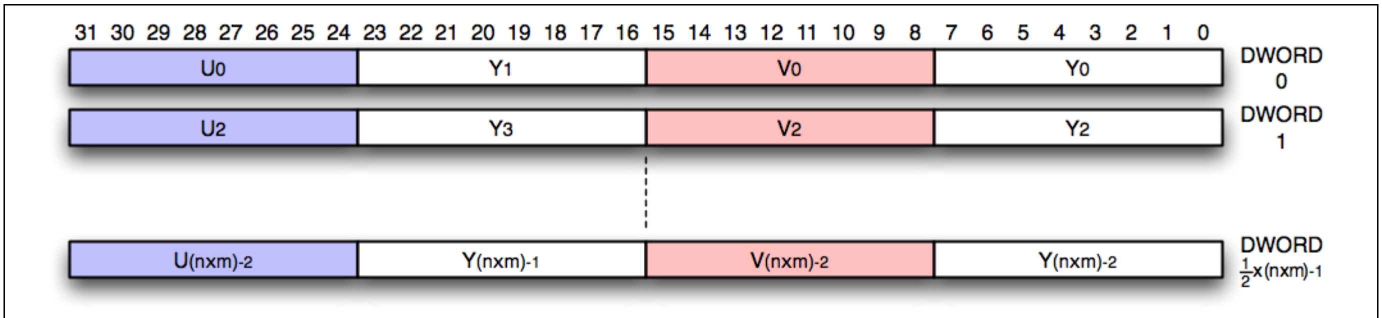
Table 3-9: YUY2 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
YUY2	0x0006	YUY2	YCbCr422	32	4	16

3.3.2.2.6. YVYU

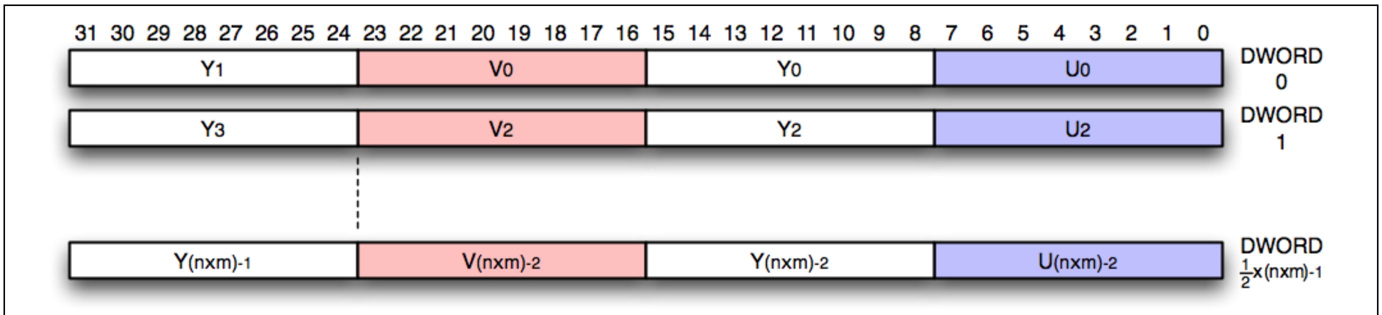
Table 3-10: YVYU VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
YVYU	0x0007	YVYU	YCbCr422	32	4	16

3.3.2.2.7. UYVY

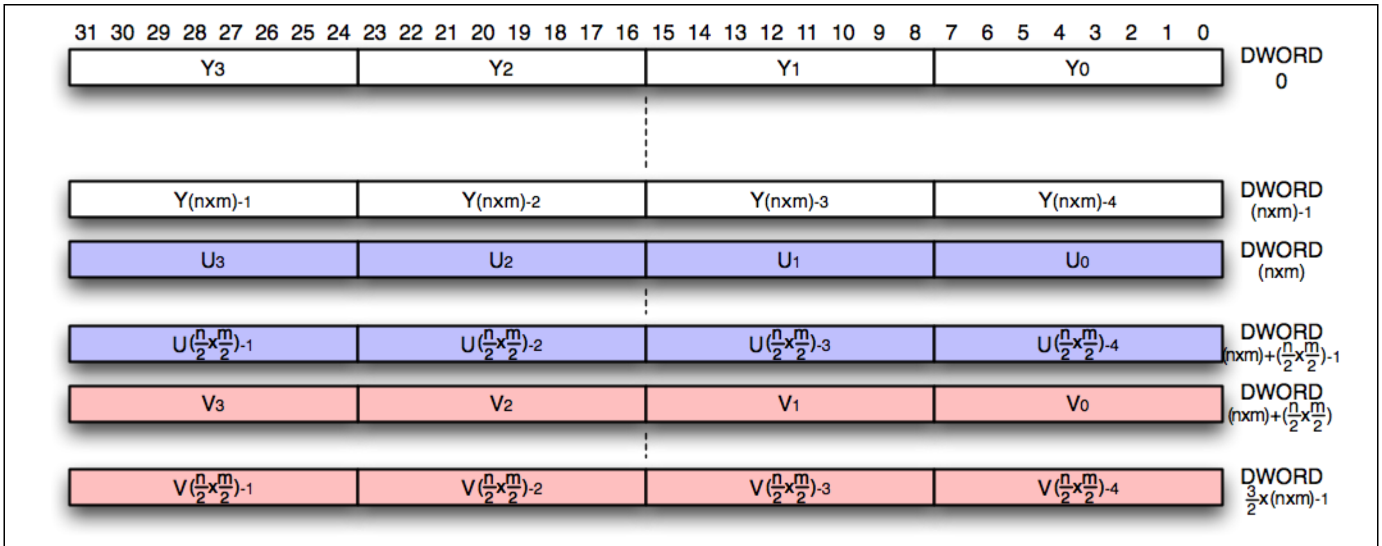
Table 3-11: UYVY VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
UYVY	0x0008	UYVY	YCbCr422	32	4	16

3.3.2.2.8. I420

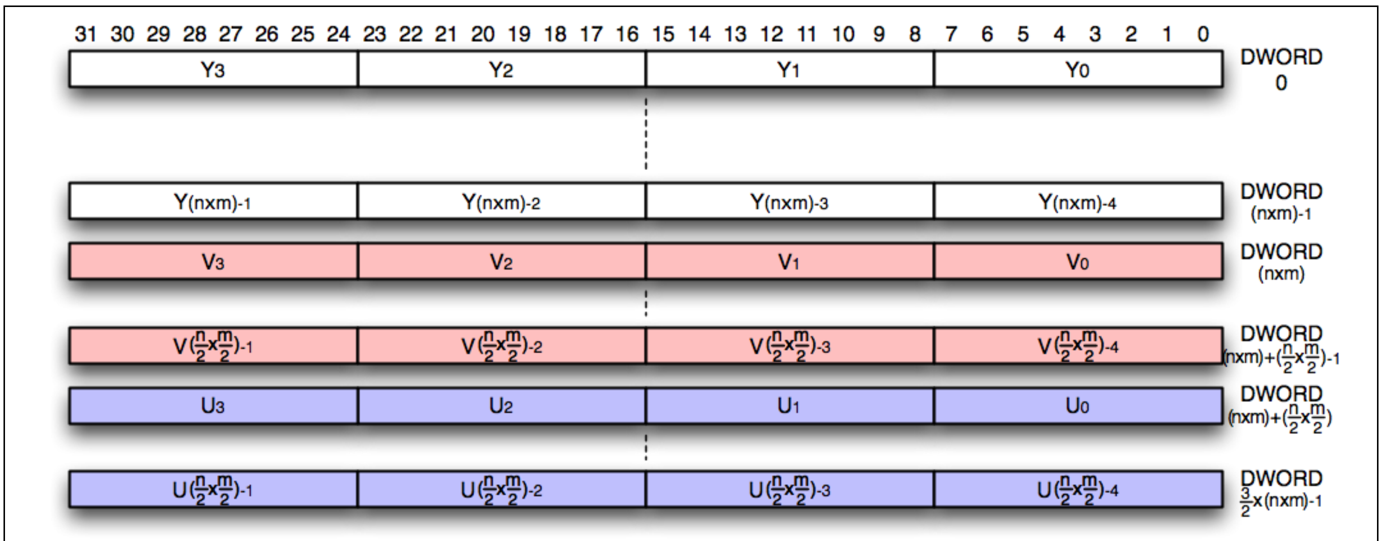
Table 3-12: I420 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
I420	0x0009	I420/IYUV	YUV420 Planar	8	1	12

3.3.2.2.9. YV12

Table 3-13: YV12 VideoSample Format



VSID	VSLVC	FourCC	Type	bitResolution	subSlotSize	bits/VS
YV12	0x000A	YV12	YUV420 Planar	8	1	12

3.4. VideoCompression

This specification defines several methods for compressing video content. A unique Video Compression ID (VCID) identifies each Method. There is also a unique Video Compression Legacy View Code (VCLVC) defined as follows:

Table 3-14: Video Compression Methods

VCID	VCLVC	Method	Description
FULL	0x0001	Uncompressed Full Frame	All the VideoSamples that together constitute a full VideoFrame are transported uncompressed from the video source to the video sink.
PARTIAL	0x0002	Uncompressed Partial Frame	One or more subregions of a VideoFrame (see Section 3.2.2.1, “Subregions”) contain VideoSamples that have changed with respect to the previous VideoFrame. Only these subregions (containing the changed VideoSamples) are transported uncompressed from the video source to the video sink. To reconstruct the full VideoFrame, all VideoSamples outside the changed subregions from the previous VideoFrame are reused.
H264BASEINTRA	0x0003	H.264 Baseline Intra	Full VideoFrames are compressed using H.264 Baseline Profile codec tools and only Intra-coded MacroBlocks. The resulting VideoFrame is called an Intra-VideoFrame.
H264BASEPRED	0x0004	H.264 Baseline Predictive	Full VideoFrames are compressed using H.264 Baseline Profile codec tools and a combination of Intra-coded and Inter-coded MacroBlocks. I-frames are coded using intra-coded MacroBlocks. P-frames use Intra-coded Macroblocks for all areas in the VideoFrame that have changed with respect to the previous VideoFrame. Inter-coded MacroBlocks with Zero-MotionVector Predictor are used for all areas that have not changed (skip mode). The resulting VideoFrame is called an Inter-VideoFrame.
H264HiQINTRA	0x0005	H.264 High Quality Intra	Full VideoFrames are compressed using H.264 High Profile Profile codec tools and only Intra-coded MacroBlocks. The resulting VideoFrame is called an Intra-VideoFrame.
H264HiQPRED	0x0006	H.264 High Quality Predictive	Full VideoFrames are compressed using H.264 High Profile Profile codec tools and a combination of Intra-coded and Inter-coded MacroBlocks. I-frames are coded using intra-coded MacroBlocks. P-frames use Intra-coded Macroblocks for all areas in the VideoFrame that have changed with respect to the previous VideoFrame. Inter-coded MacroBlocks with Zero-MotionVector Predictor are used for all areas that have not changed (skip mode). The resulting VideoFrame is called an Inter-VideoFrame.
VENDOR	0xFFFF	Vendor-defined	Vendor-defined Compression Method. Further details should be provided via the AVDD.

These different compression methods allow for a flexible trade-off between perceived video quality and required USB bandwidth. The following sections provide more details.

3.4.1. Uncompressed Video

The Uncompressed Full Frame Method is a special case of the more general Uncompressed Partial Frame Method. Indeed, the Uncompressed Partial Frame Method degenerates to the Uncompressed Full Frame Method when there is only one subregion (see Section 3.2.2.1, “Subregions”) defined and that subregion encompasses the entire VideoFrame. Therefore, the Uncompressed Full Frame and Partial Frame Methods are described here together.

In general, VideoSamples in each subregion are individually represented using either VideoSample Format Type I (see Section 3.3.1, “Type I Format” or VideoSample Format Type II (see Section 3.3.2, “Type II Format”). In the latter case some form of data compression is accomplished through Color Space data reduction.

In the case of the Uncompressed Partial Frame Method, when there is no change with respect to the previous VideoFrame, and no information contained in the AVHeader needs to be delivered over USB, no Message related to the VideoFrame shall be sent over the USB.

3.4.2. H.264 Compressed Video

All H.264 Compression Methods are based on the same H.264 AVC codec (see [IEC14496_10]). Therefore, they are described here together.

The AVFormat 1 Definition specifies the use of H.264 AVC as the normative codec for compressing video data. Therefore, all compression shall use the H.264 AVC codec with the profiles and levels described in this specification. When compressed video data is packetized, the native data stream produced by the encoder is used. Compression is always applied to full VideoFrames. Hence there is no need for subregion descriptions. The H.264 syntax also clearly defines start and end structures for the stream.

3.4.2.1. Sequence Parameter Set

The Sequence Parameter Set (SPS) contains a number of decoding parameters that apply to a series of consecutive coded video pictures, called a coded video sequence.

Each SPS contains a value in the **seq_parameter_set_id** field that identifies this SPS. The Picture Parameter Set uses this identifier to reference a specific SPS.

The following table outlines the Sequence Parameter Set (SPS) (see [IEC14496_10] for details):

Note 1: Not all fields in this table are present for all Profiles. Refer to [IEC14496_10].

Note 2: The values in this table shall be encoded according to the rules defined in Section 9.1 of [IEC14496_10]).

Table 3-15: Sequence Parameter Set

Field	Value	Description
profile_idc	Value	Profile. Defined at runtime. Depends on configuration.
constraint_set0_flag	0	-
constraint _set1_flag	0	-
constraint _set2_flag	0	-
constraint _set3_flag	0	-
constraint _set4_flag	0	-
constraint _set5_flag	0	-
reserved_zero_2bits	0	-
level_idc	Value	Level. Defined at runtime. Depends on image size.
seq_parameter_set_id	Value	SPS ID.
chroma_format_idc	Value	Defined at runtime. 0, for RAW (considered to be equivalent to monochrome), 1 for YUV4:2:0, 2 for YUV4:2:2, 3 for YUV4:4:4.
separate_colour_plane_flag	Value	Only used when YUV4:4:4 or RGB888.
bit_depth_luma_minus8	0	-
bit_depth_chroma_minus8	0	-
qpprime_y_zero_transform_bypass_flag	0	-
seq_scaling_matrix_present_flag	0	-
log2_max_frame_num_minus4	12	frame_num monitoring support.
pic_order_cnt_type	2	-
max_num_ref_frames	1	-
gaps_in_frame_num_value_allowed_flag	1	Allow frame gaps (like frame drops) and entitle decoder to handle frame buffer accordingly.
pic_width_in_mbs_minus1	Value	(Image Width/16) - 1 (up to 255).
pic_height_in_mbs_minus1	Value	(Image Height/16) - 1 (up to 255).
frame_mbs_only_flag	1	Frame-level coding.
direct_8x8_inference_flag	0	No B frame.
frame_cropping_flag	0	No cropping.
vui_parameters_present_flag	0	Default Display parameters as agreed during pairing process through USB.

3.4.2.2. Picture Parameter Set

The Picture Parameter Set (PPS) contains a number of decoding parameters that apply to one or more individual pictures within a coded video sequence (a series of consecutive coded video pictures).

Each PPS contains a value in the **pic_parameter_set_id** field that identifies this PPS. VCL NAL Units use this identifier to refer to a specific PPS that applies to the VideoPayload contained in the NAL Unit.

Similarly each PPS contains a reference ID to a Sequence Parameter Set in its **seq_parameter_set_id** field that links this PPS to a specific SPS.

The following table outlines the Picture Parameter Set (see [IEC14496_10] for details):

Note 1: Not all fields in this table are present for all Profiles. Refer to [IEC14496_10].

Note 2: The values in this table shall be encoded according to the rules defined in Section 9.1 of [IEC14496_10]).

Table 3-16: Picture Parameter Set

Field	Value	Description
pic_parameter_set_id	Value	PPS ID.
seq_parameter_set_id	Value	Refers to a specific SPS.
entropy_coding_mode_flag	0	CAVLC.
bottom_field_pic_order_in_frame_present_flag	0	-
num_slice_groups_minus1	0	1 slice group for the frame.
num_ref_idx_l0_active_minus1	0	1 reference frame (the previous one, for Inter MB).
num_ref_idx_l1_active_minus1	0	No B frame.
weighted_pred_flag	Value	Defined at runtime (0-MV inter forced for unchanged areas, intra coding otherwise), if sink decoder support. Otherwise (0,0) search area used.
weighted_bipred_idc	0	No B frame.
pic_init_qp_minus26	-16	-
pic_init_qs_minus26	0	-
chroma_qp_index_offset	Value	Defined at runtime.
deblocking_filter_control_present_flag	1	De-blocking.
constrained_intra_pred_flag	0	No constraints in Intra prediction.
redundant_pic_cnt_present_flag	0	-
transform_8x8_mode_flag	Value	Defined at runtime.
pic_scaling_matrix_present_flag	0	-
second_chroma_qp_index_offset	0	-

3.4.2.3. Frame Packing Arrangement

The AVFormat 1 Definition uses the Supplemental Enhancement Information (SEI) Message as defined by [IEC14496_10] to convey within the bit stream the actual Frame Organization (see Section 3.2.4, "VideoFrame Organization") that is used to assemble the VideoFrames. The Frame Packing Arrangement (FPA) Message is used for this purpose. Note that the FPA Message is included for all of the 3D2 Frame Organizations but is explicitly omitted for the 2D Frame Organization.

The FPA Message contains the **frame_packing_arrangement_repetition_period** field that indicates the number of VideoFrames to which this FPA Message applies.

The following table outlines the FPA Message (see [IEC14496_10] for details):

Note: The values in this table shall be encoded according to the rules defined in Section 9.1 of [IEC14496_10]).

Table 3-17: Frame Packing Arrangement Message

Field		Value	Description
frame_packing_arrangement_id		Value	Frame Packing Arrangement ID as defined in Section 3.2.4, "VideoFrame Organization".
frame_packing_arrangement_cancel_flag		0	-
frame_packing_arrangement_type		Value	5 for 3D2FP; 4 for 3D2TAB; 3 for all other 3D2 Frame Organizations.
quincunx_sampling_flag		0	-
content_interpretation_type		1	-
spatial_flipping_flag		0	-
frame0_flipped_flag		0	-
field_views_flag		0	-
current_frame_is_frame0_flag		0	-
frame0_self_contained_flag		0	-
frame1_self_contained_flag		0	-
	frame0_grid_position_x	Value	0 for ID 3D2SHOE and 3D2SHOO, 8 otherwise.
	frame0_grid_position_y	0	-
	frame1_grid_position_x	Value	0 for ID 3D2SHEO and 3D2SHEE, 8 otherwise.
	frame1_grid_position_y	0	-
frame_packing_arrangement_reserved_byte		0	-
frame_packing_arrangement_repetition_period		Value	Defines the number of VideoFrames for which this FPA message is valid.
frame_packing_arrangement_extension_flag		0	-

Note: The FPA Message provides in-band (duplicate) Frame Organization information that is also present in the VideoStreamConfig Description part of the AVDD XML Document. Whenever there is a discrepancy between the FPA Message and the advertised Frame Organization information through the AVDD, the AVDD takes precedence.

3.4.2.4. Video Compression Tools

The following table details the features a codec shall support for the USB AV Compression Profiles during encoding. In the case of

Table 3-18: USB AV Compression Tools

Feature	USB AV Compression Profiles			
	Baseline Intra (H264BASEINTRA)	HighQ Intra (H264HIQINTRA)	Baseline Predictive (H264BASEPRED)	HighQ Predictive (H264HIQPRED)
H.264 Profile	Baseline	High444	Baseline	High444
Level	Up to 5.1	Up to 5.1	Up to 5.1	Up to 5.1
I Slices	Yes	Yes	Yes	Yes
P and B Slices	No	No	Only P	Only P
Multiple Reference frames	No	No	No	No
Arbitrary Frame Ordering	No	No	No	No
In-Loop Deblocking Filter	Yes	Yes	Yes	Yes
Weighted Prediction	No	No	No	Yes
CAVLC Entropy Coding	Yes	Yes	Yes	Yes
CABAC Entropy Coding	No	No	No	No

Feature	USB AV Compression Profiles			
	Baseline Intra (H264BASEINTRA)	HighQ Intra (H264HIQINTRA)	Baseline Predictive (H264BASEPRED)	HighQ Predictive (H264HIQPRED)
FMO/ASO/RS	No	No	No	No
Interlaced coding	No	No	No	No
4:2:0 Chroma Format	Yes	Yes	Yes	Yes
Monochrome Video Format	No	Yes	No	Yes
4:2:2 Chroma Format	No	Yes	No	Yes
4:4:4 Chroma Format	No	Yes	No	Yes
8x8 vs. 4x4 Transform adaptivity	No	Yes	No	Yes
Adaptive Quantization	Yes	Yes	Yes	Yes
Separate Cb and Cr QP control	Yes	Yes	Yes	Yes
Separate Color plane coding	No	Yes	No	Yes
Transform Bypass Coding	No	No	No	No

Note: For MacroBlocks corresponding to unchanged areas of the VideoFrame, the Inter prediction tool with Zero-MotionVector Predictor shall be used. This method introduces image quality loss.

3.4.2.5. VideoCompression Description

The VideoCompression Description is a hierarchical component of the VideoStreams Configuration Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.5. VideoSample Format and VideoCompression Method Dependency

There is a dependency between the VideoSample Format (Color Space) and the VideoCompression Methods that can be used. Indeed, not all VideoSample Formats allow for H.264 compression. The following table indicates which VideoSample Formats can be used for each of the supported VideoCompression Methods:

Table 3-19: VideoSample Format vs. VideoCompression Method

VideoSample Format	Uncompressed Full Frame	Uncompressed Partial Frame	H.264 Baseline Intra	H.264 Baseline Predictive	H.264 HiQ Intra	H.264 HiQ Predictive
RGB565	yes		No		No	
RGB888	yes		No		yes	
RGB8888	yes		No		No	
I420 (4:2:0)	yes		yes		yes	
YV12 (4:2:0)	yes		yes		yes	
YUY2 (4:2:2)	yes		No		yes	
YVYU (4:2:2)	yes		No		yes	
UYVY (4:2:2)	yes		No		yes	
IYU2 (4:4:4)	yes		No		yes	

3.6. VideoStreamConfig

A VideoStream Configuration (VideoStreamConfig) in an AVData FrameBuffer Entity is fully characterized by the following components:

- The VideoBundle (Section 3.1.3)
- The VideoFrame component, consisting of
  - The VideoFrame Rate (Section 3.2.1)
  - The VideoFrame Organization (Section 3.2.4)

    - The VideoFrame format (Section 3.2.5)
- The VideoSample component, consisting of
    - The VideoSample Format (Section 3.3)
    - The VideoCompression (Section 3.4)

3.6.1. VideoStreamConfig Description

The VideoStreamConfig Description is a hierarchical component of the VideoStreamConfigList Description. Refer to Section 3.7.1, "VideoStreamsConfigList Description" for details.

3.7. VideoStreamConfigList

A VideoStreamConfigList is a construct that consists of a list of VideoStreamConfig Descriptions that are supported by a particular AVData FrameBuffer Entity in which the VideoStreamConfigList is advertised.

3.7.1. VideoStreamsConfigList Description

The VideoStreamConfigList Description can be found at: VideoBulkStreamConfigList.

3.8. NAL Units

This specification requires that all compressed VideoStreams be encapsulated using H.264 Network Abstraction Layer (NAL) syntax.

NAL Units are divided into VCL and non-VCL NAL Units where VCL (Video Coding Layer) NAL Units contain the data that represents the values of the VideoSamples in the VideoFrames, and non-VCL NAL units contain any associated additional information such as SPS and PPS parameter sets and Supplemental Enhancement Information (timing information and other supplemental data that may enhance usability of the decoded video signal but are not necessary for decoding the values of the VideoSamples in the VideoFrames). This includes the Frame Packing Arrangement (FPA) Message, used to convey 3D2 Frame Organization information within the VideoStream (in-band).

The following NAL Unit Types are supported:

- VCL NAL Units containing compressed VideoPayloads as defined in Section 3.4.2 of this document
- Non-VCL NAL Units containing SPS data structures as defined in Section 3.4.2.1
- Non-VCL NAL Units containing PPS data structures as defined in Section 3.4.2.2
- Non-VCL NAL Units containing FPA SEI Messages as defined in Section 3.4.2.3

For each NAL Unit Type, the level of criticality of the data being transported (for quality of service management) can be expressed (in the NAL Header).

Each NAL Unit starts with a NAL Header and adds delimiting syntax to the stream as described in the following sections.

3.8.1. NAL header

The NAL Header is one byte in size and the bit allocations within the NAL Header are as follows:

Table 3-20: NAL Header

Bit Offset	Field	Size (#bits)	Value
0	**forbidden_zero_bit**	1	Shall be set to zero (0b0).
1	**nal_ref_idc**	2	Indicates the importance of the NAL Unit content: • SPS, PPS, I-slice: 3 (0b11) critical • Reference P-slice: 1 (0b01) medium • Non-reference P-slice: 0 (0b00) low
3	**nal_unit_type**	5	Indicates the NAL Unit's type: • P-Slice: 1 (0b00001) • I-Slice: 5 (0b00101) • SEI Message: 6 (0b00110) • SPS: 7 (0b00111) • PPS: 8 (0b01000)

3.8.2. NAL Unit Delimiters

NAL Units syntax structures are formatted into a byte stream using the NAL Byte Stream Format as defined in Annex B of [IEC14496_10]. A unique start code prefix (0x000001) precedes each NAL Unit. The use of emulation prevention bytes guarantees that start code prefixes are unique identifiers of the start of a new NAL unit. A unique end code (0x000000 or 0x000001 or the end of the byte stream) follows the NAL Unit. Therefore, searching for the surrounding start code prefix and end code and counting the bytes in between yields the NAL Unit size. The exact algorithms to perform this NAL Unit extraction are described in detail Annex B of [IEC14496_10].

3.8.3. NAL Unit Payload Data

The NAL Unit payload data and payload size varies depending on the NAL Unit Type.

Non-VCL NAL Units contain one SPS, PPS, or SEI.

For compressed video, a VCL NAL Unit typically encapsulates one video slice.

3.9. Current Limitations

At this time, the AVFormat 1 specification only supports VideoBundles consisting of one VideoTrack. The single VideoTrack may contain either one video channel (2D) or 2 video channels (3D2). VideoSlots are either VideoLine-interleaved (Side-by-Side frame organization) or VideoFrame-interleaved (Top-and-Bottom frame organization). VideoParticle-interleaved Streams are not used at this time.

4. Video Data Format

The AVFormat 1 specification uses the Control Bulk Pair (CBP) and the class-defined Control Sequence mechanisms to exchange VideoStreams between the Controller and the AVFunction (see [AVFUNCTION] for a detailed description).

In particular, when sending a VideoStream to the AVFunction, the Controller issues subsequent SET Control Sequences to the SourceData AVControl inside a targeted AVData FrameBuffer-In Entity. Each Command Message (part of the SET Control Sequence – see [AVFUNCTION]) contains all the video data related to exactly one VideoFrame (Full, Partial, or Compressed).

Likewise, when receiving a VideoStream from the AVFunction, the Controller receives subsequent Notify Control Sequences from the SinkData AVControl inside a targeted AVData FrameBuffer-Out Entity. Each Notify Message contains all the video data related to exactly one VideoFrame (Full, Partial, or Compressed).

> Note: AudioStreams are transported separate from any VideoStream, using dedicated isochronous AVData Audio Streaming Interfaces (one per AudioStream). A full description of the supported Audio Formats and all information necessary to implement such AVData Audio Streaming Interfaces can be found in [AVFORMAT_2] and [AVFUNCTION]. To aid in Audio/Video synchronization, timing information may optionally be provided within the VideoStream as described in Section 4.1, "AVHeader".

The following figure illustrates the Command/Notify Message Layout (In the figure, the length of the Control-specific part of the Message (AVHeader + VideoPayload) is chosen arbitrarily):

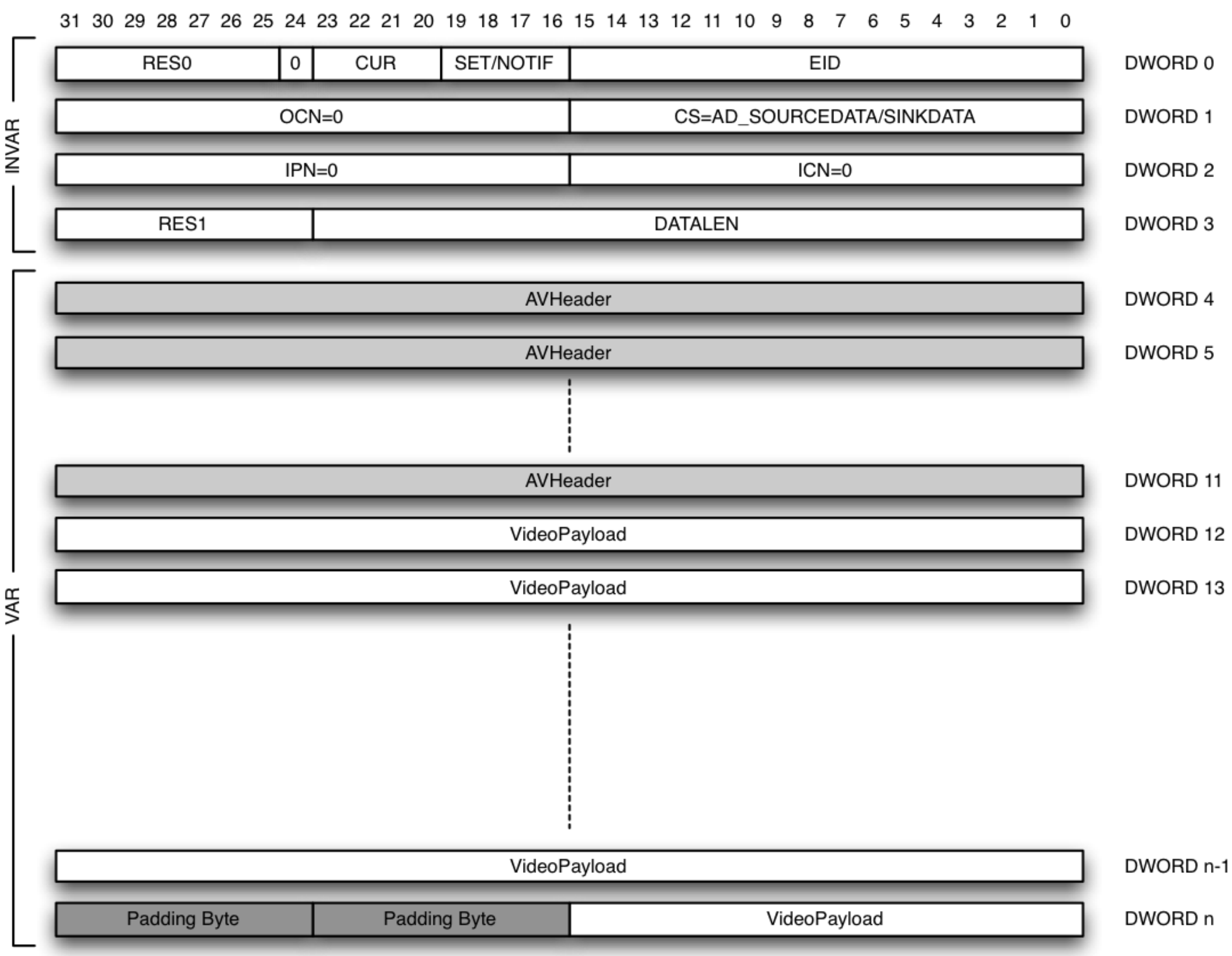


Figure 4-1: Command/Notify Message Layout

The content of the Command or Notify Message is always structured as follows:

- The INVAR part of the Message contains the mandatory fields as defined in [AVFUNCTION].
- The VAR part contains the following fields:
  - A fixed-length AVHeader containing pertinent information about the VideoFrame that follows.
  - A variable-length VideoPayload that contains the actual VideoSample values and some additional video-related information.

The following sections provide more details about the AVHeader and VideoPayload.

4.1. AVHeader

The AVHeader is exactly 32 bytes long and has the following layout:

Table 4-1: AVHeader Layout

Offset	Field	Size	Value	Description
0	**wFlags**	2	Bitmap	D0: VideoFrameIDValid D1: HDCPOn D2: ClockInfoValid D15..3: Reserved. Shall be set to zero.
2	**bVideoFrameID**	1	Number	VideoFrame ID.
3	**Reserved1**	1	Number	Reserved. Shall be set to zero.
4	**dStreamCtr**	4	Number	The streamCtr value as assigned by the HDCP transmitter.
8	**qInputCtr**	8	Number	The inputCtr value associated with the first full 16-byte encrypted data block in the SIP.
16	**dPTSLow**	4	Number	The lower 32 bits of the PTS
20	**dESCRBaseLow**	4	Number	The lower 32 bits of the ESCRBase
24	**wExtensions**	2	Number	D0: Upper bit of the PTS D1: Upper bit of the ESCRBase D10..2: ESCRExt D11..15: Reserved. Shall be set to zero.
26	**Reserved6**	6	Number	Reserved. Shall be set to zero.

The **wFlags** field provides information about the subsequent fields of the AVHeader as follows:

- Bit D0 (VideoFrameIDValid) indicates whether the **bVideoFrameID** field contains a valid value (D0=0b1) or not (D0=0b0). Implementations are allowed to always set the VideoFrameIDValid bit to 0b0, effectively indicating that they do not support a VideoFrame ID.
- Bit D1 (HDCPOn) indicates whether the current video stream is protected and is HDCP-encrypted (D1=0b1) or not (D1=0b0).
- Bit D2 (ClockInfoValid) indicates whether the **dPTSLow, dESCRBaseLow** and **wExtensions** fields contains valid values (D2=0b1) or not (D2=0b0).
- Bits D15..3 are reserved and shall be set to zero.

The **bVideoFrameID** field contains the VideoFrame ID of the current VideoFrame. The VideoFrame ID is a counter value that is incremented by one for every new VideoFrame generated at the source and rolls over to zero (0x00) when the value 255 (0xFF) is reached. If the VideoFrameIDValid bit in the **wFlags** field is not set, then the transmitter should set the **bVideoFrameID** field to zero and the receiver shall ignore the **bVideoFrameID** field.

The **Reserved1** field is reserved and shall be set to zero.

The **dStreamCtr** and **qInputCtr** fields are used by the HDCP protection mechanism. Whenever protected content is transported, the HDCPOn bit in the **wFlags** field shall be set in the AVHeader of every CBP Message that contains a protected VideoFrame. When the HDCPOn bit in the **wFlags** field is not set, then the HDCP-related fields are irrelevant and the transmitter should set them to zero and the receiver shall ignore them.

- The **dStreamCtr** field contains the *streamCtr* value associated with the video stream as assigned by the HDCP transmitter.
- The **qInputCtr** field contains the *inputCtr* value associated with the first 16-byte encrypted data block in the VideoFrame.

The clock information related fields (**dPTSLow**, **dESCRBaseLow** and **wExtensions** fields) shall have valid information when the ClockInfoValid bit in the **wFlags** field is set. When the ClockInfoValid bit in the **wFlags** field is not set, then the clock information related fields are irrelevant and the transmitter should set them to zero and the receiver shall ignore them.

- The PTS is a 33-bit value of which the 32 lower bits are encoded in the **dPTSLow** field and the upper bit is encoded in Bit D0 of the **wExtensions** field. The PTS contains a presentation timestamp as defined in [IEC13818_1], Section 2.4.3.7, Equation 2-11:

$$PTS(i) = \left(\left(SystemClockFrequency * t_F(i)\right) DIV\ 300\right)\ MOD\ 2^{33}$$

  where $SystemClockFrequency$ is the 27 MHz system clock and $t_F(i)$ is the presentation time of VideoFrame (i).

- The $ESCR$ consists of 2 parts, $ESCRBase$ and $ESCRExt$, as defined in [IEC13818_1], Section 2.4.3.7, Equations 2-13, 2-14, and 2-15:

$$ESCR(i) = ESCRBase(i) * 300 + ESCRExt(i)$$

  and

$$ESCRBase(i) = ((SystemClockFrequency * t(i)\ DIV\ 300)\ MOD\ 2^{33}$$

$$ESCRExt(i) = ((SystemClockFrequency * t(i)\ DIV\ 1)\ MOD\ 300$$

  where $SystemClockFrequency$ is the 27 MHz system clock and $t(i)$ is the intended time of arrival of the first byte in VideoFrame(i).

  Note: When using these fields to convey timing information, it is assumed that the synchronization model as defined in [IEC13818_1] is used. This means that the **dPTSLow**, **dESCRBaseLow** and **wExtensions** fields shall be updated at a rate greater than or equal to the update rate required by [IEC13818_1]. In the absence of VideoPayload data (when using the Uncompressed Partial Frame Method), only the AVHeader shall be sent.

The **Reserved6** field is reserved and shall be set to zero.

4.2. VideoPayload

The VideoPayload data organization differs depending on whether the VideoPayload is Uncompressed or Compressed. In all cases, the VideoPayload shall be 16-byte aligned. This is accomplished by zero-padding the actual VideoPayload to the nearest multiple of 16 bytes.

4.2.1.1. Uncompressed Video

The following figure describes the VideoPayload layout used when exchanging Uncompressed Full Frame or Partial Frame VideoPayloads with the SourceData or SinkData AVControl. (The AVHeader is included in the figure for completeness but is not part of the VideoPayload.)

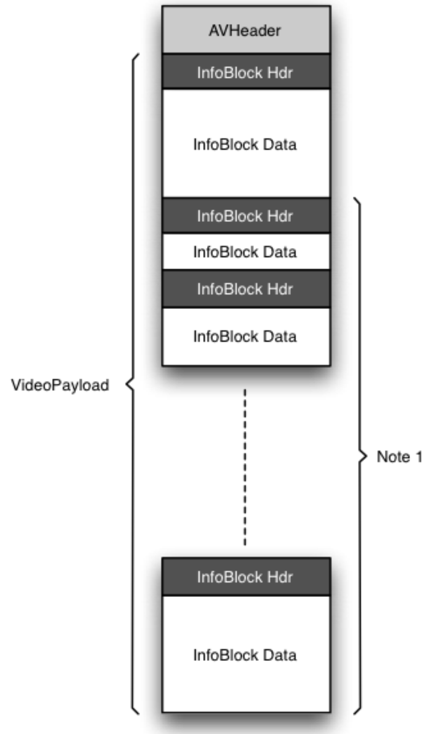


Figure 4-2: VideoPayload Layout for Uncompressed Video (including AVHeader)

Note 1: Only present when more than one subregion is updated. See below for details.

Note 2: Some additional zero-padding bytes may be present at the end of the VideoPayload (not shown here).

The VideoPayload always contains video information related to a single VideoFrame. All video information is organized into one or more **InfoBlock** data structures as described in the following section.

4.2.1.1.1. InfoBlock

The following table describes the InfoBlock layout:

Table 4-2: InfoBlock Layout

Offset	Field	Size	Value	Description
0	**X**	2	Number	X-coordinate of the subregion.
2	**Y**	2	Number	Y-coordinate of the subregion.
4	**W**	2	Number	Width of the subregion.
6	**H**	2	Number	Height of the subregion.
8	**VideoDataLength**	4	Number	Number of bytes of the VideoPayload for the subregion: p.
12	**VideoData**	p	-	Actual VideoPayload for the subregion.

The VideoPayload in the **VideoData** field consists of a set of VideoSamples in a contiguous, rectangular subregion of the VideoFrame, identified by the **X**, **Y**, **W**, and **H** fields. The number of bytes in the VideoPayload is indicated in **VideoDataLength** the field.

There are as many **InfoBlock** data structures present as there are individual subregions to update.

When using the Uncompressed Full Frame Method, there shall only be one **InfoBlock** present, the **X** and **Y** field shall both be set to zero and the **W** and **H** field shall be set to W_a and H_a respectively.

When using the Uncompressed Partial Frame Method, there can be one or more **InfoBlocks** present. It is the responsibility of the Partial Update Definition stage (see Figure , “AVFormat 1 Data Flow”) to create the subregions by determining which VideoSamples have changed between the current VideoFrame and the previous VideoFrame and grouping them into a balanced set of rectangular areas, minimizing the number of redundant (unchanged) VideoSamples included in those rectangular areas but at the same time keeping the number of rectangular areas to a minimum as well. The exact algorithms to achieve this are outside the realm of this specification.

The order in which the **InfoBlock** structures appear in the VideoPayload is important. Updates shall be applied in that exact order so that deterministic results can be achieved.

4.2.1.2. H.264 Compressed Video

The following figure describes the VideoPayload layout used when exchanging Compressed VideoPayloads with the SourceData or SinkData AVControl. (The AVHeader is included in the figure for completeness but is not part of the VideoPayload.)

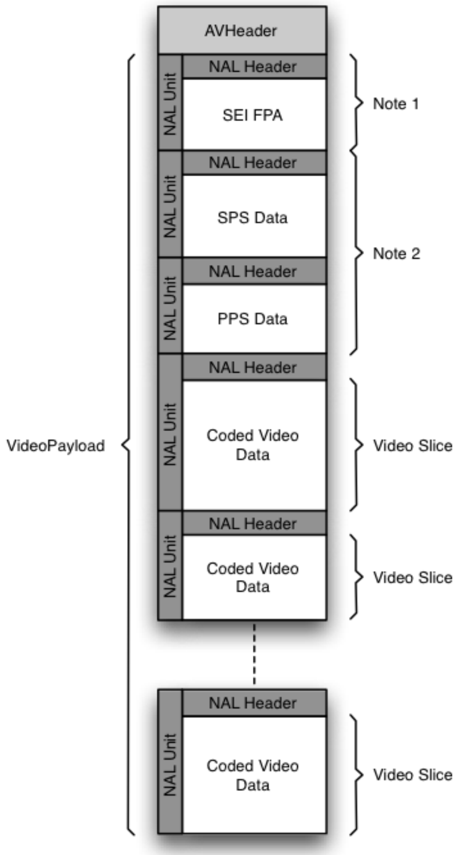


Figure 4-3: VideoPayload Layout for Compressed Video (including AVHeader)

Note 1: Only present in certain VideoPayloads. See (*) below for details.

Note 2: Only present in certain VideoPayloads. See Section 4.2.1.3, "VideoSequence" for details.

Note 3: Some additional zero-padding bytes may be present at the end of the VideoPayload (not shown here).

The VideoPayload always contains video information related to a single VideoFrame. All video information is encapsulated into the proper NAL Units. For Compressed video, the following NAL Unit Types may occur in one VideoPayload:

- One NAL Unit of Type SEI (6) that contains an FPA Message as defined in Section 3.4.2.3, "Frame Packing Arrangement".
  (*) This NAL Unit shall be present for 3D2 Frame Organizations but shall be omitted for the 2D Frame

Organization. The SEI FPA Message shall only be present or repeated as required by Annex D.2.25, "Frame packing arrangement SEI message semantics" of [IEC14496_10].

- One NAL Unit of Type "SPS" (7) containing a Sequence Parameter Set (see Section 3.4.2.1, "Sequence Parameter Set") followed by one NAL Unit of Type "PPS" (8) containing a Picture Parameter Set (see Section 3.4.2.2, "Picture Parameter Set"). These NAL Units are not always present and their presence is governed by the rules set forth in Section 4.2.1.3, "VideoSequence".
- NAL Units of Type "I-Slice" (5) or NAL Units of Type "P-Slice" (1) that contain the coded VideoSamples.

4.2.1.3. VideoSequence

4.2.1.3.1. H.264 Intra Compression Methods

The H.264 Baseline Intra and H.264 High Quality Intra Compression Methods generate Sequences of VideoFrames that are composed only of intra-coded MacroBlocks (Intra-VideoFrames).

In this case, a Sequence Parameter Set (SPS) and a Picture Parameter Set (PPS) NAL Unit shall be inserted in the NAL Unit Stream approximately every 10 seconds, between the FPA SEI NAL Unit (if present) and the actual VCL (Video Coding Layer) NAL Units that contain the coded VideoPayload for the VideoFrame. The SPS NAL Unit repeats elementary configuration parameters for the video sequence (SPS) and the PPS NAL Unit repeats elementary information for each VideoFrame (PPS) to enhance the robustness of the NAL Unit Stream.

As an example of an Intra VideoStream sequence, the following VideoFrame sequence is obtained for the Intra Compression Methods:

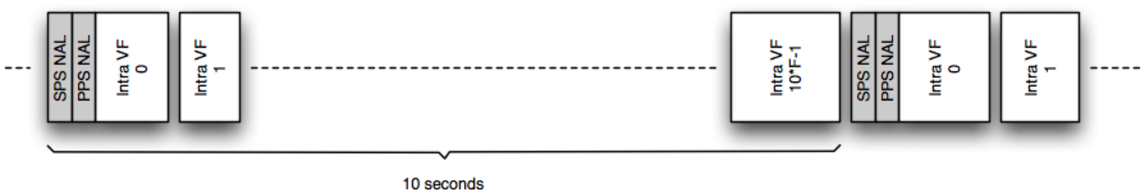


Figure 4-4: Intra VideoStream Sequence

4.2.1.3.2. H.264 Predictive Compression Methods

The H.264 Baseline Predictive and H.264 High Quality Predictive Compression Methods generate two types of VideoFrames in a repeating sequence of approximately 1 second as follows:

- One Intra-VideoFrame composed only of intra-coded MacroBlocks, followed by:
- $F - 1$ Inter-VideoFrames using intra-coded MacroBlocks for all changed areas in the VideoFrame and using inter-coded MacroBlocks with Zero-MotionVector Predictor for all unchanged areas in the VideoFrame.

In this case, a Picture Parameter Set (PPS) and a Sequence Parameter Set (SPS) NAL Unit shall be inserted in the NAL Unit Stream approximately every 10 seconds, between the FPA SEI NAL Unit (if present) and the actual VCL (Video Coding Layer) NAL Units that contain the coded VideoPayload for the VideoFrame. This effectively results in the same 10-second repetition rate for the SPS and PPS information, as is the case for the H.264 Intra Compression Methods.

The H.264 Video Decoder shall parse the SPS fields to understand the VideoSample Format (Color Space) and the PPS fields to understand the VideoStream Frame Organization that are used.

As an example of a Predictive VideoStream sequence, the following VideoFrame sequence is obtained for the Predictive Compression Methods:

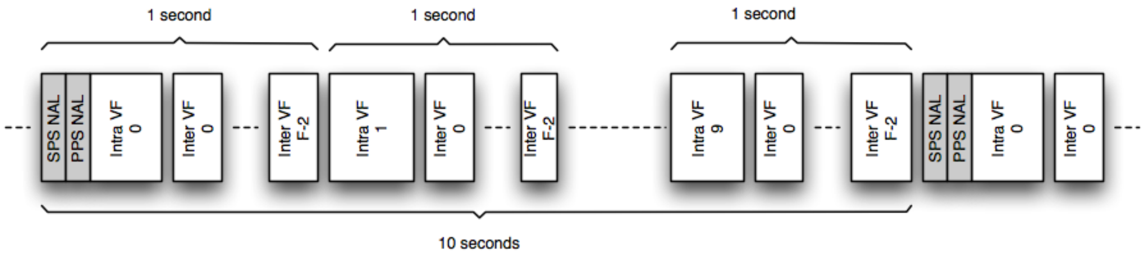


Figure 4-5: Predictive VideoStream Sequence

5. Support for Content Protection

This specification allows the transport of video content that is subject to Digital Right Management rules. These rules imply the existence of robust link protection when this type of content is transported over the USB. The USB AV Device Class Definition relies on the content protection framework protocol provided by the USB Content Security Device Class Definition ([USBCS]) and mandates HDCP 2.1 as the encryption scheme to avoid transmission of content in the clear. The USB version of the HDCP 2.1 implementation is fully defined in the Content Security Method #5 specification ([USBCSM-5]). In-band provisions have been made here to allow compliant deployment of Content Security Method #5 (HDCP 2.1).

When link protection is enabled, only the VideoPayload shall be encrypted using HDCP 2.1 tools, whereas the AVHeader remains in the clear. The AVHeader shall include proper settings for the HDCPOn bit in the **wFlags** field, and the **dStreamCtr** and **qInputCtr** fields, as described in Section 4.1, “AVHeader”.

The HDCP information shall not be included unless it is certain that the receiver of the video content is HDCP-capable. The mechanisms for making this determination over USB are described in [USBCSM-5].

Each VideoStream Configuration (see Section 3.6, “VideoStreamConfig”) reports in its VideoStreamConfig Description whether the HDCP Protocol is supported for that VideoStream Configuration through the `<@hdcp>` attribute.

Appendix A. VideoFrame Format Dimensions and Timings

The following sections summarize all VideoFrame related dimensions and timings that fully describe each VideoFrame Format. For more details, refer to Section 3.2.3, “VideoFrame Spatial Layout and Timing Aspects” and Section 3.2.4, “VideoFrame Organization”.

A.1. DTV Formats

The following tables presents the VideoFrame related dimensions and timings that fully describe the VideoFrame Format as defined by [CEA-861-E]. Note that only a subset of the [CEA-861-E]-defined formats is supported by this specification. All VideoFrame Formats are progressive scan. The Video Identification Code (VIC) as defined in [CEA-861-E] is indicated as a reference.

For the 2D Frame Organization, the VideoFrame Format describes the dimensions and timings of the actual single channel VideoFrame.

For all of the 3D2 Half Resolution Organizations (3D2TAB, 3D2SHOE, 3D2SHOO, 3D2SHEO, 3D2SHEE), the VideoFrame Format describes the dimensions and timings of the resulting VideoFrame after both Left and Right VideoFrame have been subsampled in half by either dropping every other line (3D2TAB) or by dropping every other column (all the 3D2SHxx Frame Organizations) and then merging the 2 subsampled VideoFrames into one resulting VideoFrame. Due to the subsampling by 2 and then the merging of the 2 subsampled VideoFrames into one resulting VideoFrame, this resulting VideoFrame has the exact same dimensions and timings as its corresponding 2D VideoFrame. Therefore, the same table applies to both the 2D Frame Organizations and all of the 3D Half Resolution Frame Organizations.

The following notations are used in the table:

- F: VideoFrame Rate (in Hz)
- VFID: VideoFrame Format ID used to identify the VideoFrame Format
- VFLVC: VideoFrame Format Legacy View Code used to identify the VideoFrame Format while in Legacy View
- VIC: Video Identification Code
- Aspect: The aspect ratio of the VideoFrame
- W_a: Active VideoFrame Width (in VideoSamples)
- H_a: Active VideoFrame Height (in VideoLines)
- W_t: Total VideoFrame Width (in VideoSamples)
- H_t: Total VideoFrame Height (in VideoLines)
- F_{vs}: VideoSample Rate (pixelclock) (in MHz) – for informational purposes only

Table A-1: DTV VideoFrame Dimensions and Timings for 2D and 3D2TAB/3D2SHxx Frame Organizations

F	VFID	VFLVC	VIC	Aspect	W_a	H_a	W_t	H_t	F_{vs}
60	640X480P	0x0002	1	4:3	640	480	800	525	25.200
60	720X480P	0x0004	2	4:3	720	480	858	525	27.027
60	720X480PH	0x0006	3	16:9	720	480	858	525	27.027
120	720X480P	0x0004	48	4:3	720	480	858	525	54.054
240	720X480P	0x0004	56	4:3	720	480	858	525	108.108
50	720X576P	0x0008	17	4:3	720	576	864	625	27.000
100	720X576P	0x0008	42	4:3	720	576	864	625	54.000
200	720X576P	0x0008	52	4:3	720	576	864	625	108.000
24	1280X720P	0x000A	60	16:9	1280	720	3300	750	59.400
25	1280X720P	0x000A	61	16:9	1280	720	3960	750	74.250
30	1280X720P	0x000A	62	16:9	1280	720	3300	750	74.250
50	1280X720P	0x000A	19	16:9	1280	720	1980	750	74.250

F	VFID	VFLVC	VIC	Aspect	W_a	H_a	W_t	H_t	F_{vs}
60	1280X720P	0x000A	4	16:9	1280	720	1650	750	74.250
100	1280X720P	0x000A	41	16:9	1280	720	1980	750	148.500
120	1280X720P	0x000A	47	16:9	1280	720	1650	750	148.500
24	1920X1080P	0x000C	32	16:9	1920	1080	2750	1125	74.250
25	1920X1080P	0x000C	33	16:9	1920	1080	2640	1125	74.250
30	1920X1080P	0x000C	34	16:9	1920	1080	2200	1125	74.250
50	1920X1080P	0x000C	31	16:9	1920	1080	2640	1125	148.500
60	1920X1080P	0x000C	16	16:9	1920	1080	2200	1125	148.500
100	1920X1080P	0x000C	64	16:9	1920	1080	2640	1125	297.000
120	1920X1080P	0x000C	63	16:9	1920	1080	2200	1125	297.000

For the 3D2FP Frame Organization, the situation is different. The overall VideoFrame timing is maintained by doubling the VideoSample Rate to account for the fact that double the number of VideoSamples is present.

There are two possible VideoFormat representations when transporting the 3D2FP Frame Organization over USB:

- The Active Space is suppressed and is not sent over USB ($S_a = 0$).
- The Active Space is included in the VideoPayload sent over USB. The VFIDs reflect this through the –AS suffix.

The following table contains the detailed VideoFrame dimensions and timings. Note that the values for F_{vs} are double those of the previous table.

The following notations are used in the table:

- F: VideoFrame Rate (in Hz)
- VFID: VideoFrame Format ID used to identify the VideoFrame Format
- VFLVC: VideoFrame Format Legacy View Code used to identify the VideoFrame Format while in Legacy View
- VIC: Video Identification Code
- Aspect: The aspect ratio of the VideoFrame
- W_a: Active VideoFrame Width (in VideoSamples)
- H_a: Active VideoFrame Height (in VideoLines) for the Left and the Right VideoFrame
- S_a: Indicates the size of the Active Space (in VideoLines)
- H_{at}: Total Active VideoFrame Height (in VideoLines): $2 * H_a + S_a$
- W_t: Total VideoFrame Width (in VideoSamples)
- H_t: Total VideoFrame Height (in VideoLines)
- F_{vs}: VideoSample Rate (pixelclock) (in MHz) – for informational purposes only

Table A-2: DTV VideoFrame Dimensions and Timings for the 3D2FP Frame Organization

F	VFID	VFLVC	VIC	Aspect	W_a	H_a	S_a	H_{at}	W_t	H_t	F_{vs}
60	640X480P	0x0002	1	4:3	640	480	0	960	800	1050	50.400
60	640X480PAS	0x0003	1	4:3	640	480	45	1005	800	1050	50.400
60	720X480P	0x0004	2	4:3	720	480	0	960	858	1050	54.054
60	720X480PAS	0x0005	2	4:3	720	480	45	1005	858	1050	54.054
60	720X480PH	0x0006	3	16:9	720	480	0	960	858	1050	54.054
60	720X480PHAS	0x0007	3	16:9	720	480	45	1005	858	1050	54.054
120	720X480P	0x0004	48	4:3	720	480	0	960	858	1050	108.108
120	720X480PAS	0x0005	48	4:3	720	480	45	1005	858	1050	108.108
240	720X480P	0x0006	56	4:3	720	480	0	960	858	1050	216.216
240	720X480PAS	0x0007	56	4:3	720	480	45	1005	858	1050	216.216

F	VFID	VFLVC	VIC	Aspect	W_a	H_a	S_a	H_{at}	W_t	H_t	F_{vs}
50	720X576P	0x0008	17	4:3	720	576	0	1152	864	1250	54.000
50	720X576PAS	0x0009	17	4:3	720	576	49	1201	864	1250	54.000
100	720X576P	0x0008	42	4:3	720	576	0	1152	864	1250	108.000
100	720X576PAS	0x0009	42	4:3	720	576	49	1201	864	1250	108.000
200	720X576P	0x0008	52	4:3	720	576	0	1152	864	1250	216.000
200	720X576PAS	0x0009	52	4:3	720	576	49	1201	864	1250	216.000
24	1280X720P	0x000A	60	16:9	1280	720	0	1440	3300	1500	118.800
24	1280X720PAS	0x000B	60	16:9	1280	720	30	1470	3300	1500	118.800
25	1280X720P	0x000A	61	16:9	1280	720	0	1440	3960	1500	148.500
25	1280X720PAS	0x000B	61	16:9	1280	720	30	1470	3960	1500	148.500
30	1280X720P	0x000A	62	16:9	1280	720	0	1440	3300	1500	148.500
30	1280X720PAS	0x000B	62	16:9	1280	720	30	1470	3300	1500	148.500
50	1280X720P	0x000A	19	16:9	1280	720	0	1440	1980	1500	148.500
50	1280X720PAS	0x000B	19	16:9	1280	720	30	1470	1980	1500	148.500
60	1280X720P	0x000A	4	16:9	1280	720	0	1440	1650	1500	148.500
60	1280X720PAS	0x000B	4	16:9	1280	720	30	1470	1650	1500	148.500
100	1280X720P	0x000A	41	16:9	1280	720	0	1440	1980	1500	297.000
100	1280X720PAS	0x000B	41	16:9	1280	720	30	1470	1980	1500	297.000
120	1280X720P	0x000A	47	16:9	1280	720	0	1440	1650	1500	297.000
120	1280X720PAS	0x000B	47	16:9	1280	720	30	1470	1650	1500	297.000
24	1920X1080P	0x000C	32	16:9	1920	1080	0	2160	2750	2250	148.500
24	1920X1080PAS	0x000D	32	16:9	1920	1080	45	2205	2750	2250	148.500
25	1920X1080P	0x000C	33	16:9	1920	1080	0	2160	2640	2250	148.500
25	1920X1080PAS	0x000D	33	16:9	1920	1080	45	2205	2640	2250	148.500
30	1920X1080P	0x000C	34	16:9	1920	1080	0	2160	2200	2250	148.500
30	1920X1080PAS	0x000D	34	16:9	1920	1080	45	2205	2200	2250	148.500
50	1920X1080P	0x000C	31	16:9	1920	1080	0	2160	2640	2250	297.000
50	1920X1080PAS	0x000D	31	16:9	1920	1080	45	2205	2640	2250	297.000
60	1920X1080P	0x000C	16	16:9	1920	1080	0	2160	2200	2250	297.000
60	1920X1080PAS	0x000D	16	16:9	1920	1080	45	2205	2200	2250	297.000
100	1920X1080P	0x000C	64	16:9	1920	1080	0	2160	2640	2250	594.000
100	1920X1080PAS	0x000D	64	16:9	1920	1080	45	2205	2640	2250	594.000
120	1920X1080P	0x000C	63	16:9	1920	1080	0	2160	2200	2250	594.000
120	1920X1080PAS	0x000D	63	16:9	1920	1080	45	2205	2200	2250	594.000

A.2. Additional Formats

The following table presents the VideoFrame related dimensions and timings that fully describe each VideoFrame Format that are commonly used in PC Monitor environments. All VideoFrame Formats are progressive scan.

For the 2D Frame Organization, the VideoFrame Format describes the dimensions and timings of the actual single channel VideoFrame.

For all of the 3D2 Half Resolution Organizations (3D2TAB, 3D2SHOE, 3D2SHOO, 3D2SHEO, 3D2SHEE), the VideoFrame Format describes the dimensions and timings of the resulting VideoFrame after both Left and Right VideoFrame have been subsampled in half by either dropping every other line (3D2TAB) or by dropping every other column (all the 3D2SHxx Frame Organizations) and then merging the 2 subsampled VideoFrames into one resulting VideoFrame. Due to the subsampling by 2 and then the merging of the 2 subsampled VideoFrames into one resulting VideoFrame, this resulting VideoFrame has the exact same dimensions and timings as its corresponding 2D VideoFrame. Therefore, the same table applies to both the 2D Frame Organizations and all of the 3D Half Resolution Frame Organizations.

The following notations are used in the table:

- F: VideoFrame Rate (in Hz)
- VFID: VideoFrame Format ID used to identify the VideoFrame Format
- VFLVC: VideoFrame Format Legacy View Code used to identify the VideoFrame Format while in Legacy View
- Aspect: The aspect ratio of the VideoFrame
- W_a: Active VideoFrame Width (in VideoSamples)
- H_a: Active VideoFrame Height (in VideoLines)
- W_t: Total VideoFrame Width (in VideoSamples)
- H_t: Total VideoFrame Height (in VideoLines)
- F_{vs}: VideoSample Rate (pixelclock) (in MHz) – for informational purposes only

Table A-3: Additional VideoFrame Dimensions and Timings for 2D and 3D2TAB/3D2SHxx Frame Organizations

F	VFID	VFLVC	Aspect	W_a	H_a	W_t	H_t	F_{vs}
100	640X480P	0x0002	4:3	640	480	848	509	43.160
60	768X576P	0x000E	4:3	768	576	976	597	34.960
100	768X576P	0x000E	4:7	768	576	1024	611	62.570
60	800X600P	0x0010	4:3	800	600	1056	628	40.000
100	800X600P	0x0010	4:3	800	600	1072	636	68.180
60	1024X768P	0x0012	4:3	1024	768	1344	806	65.000
100	1024X768P	0x0012	4:3	1024	768	1392	814	113.310
60	1152X864P	0x0014	4:3	1152	864	1520	895	81.620
100	1152X864P	0x0014	4:3	1152	864	1568	915	143.470
60	1280X768P	0x0016	5:3	1280	768	1664	798	79.672
60	1280X800P	0x0018	16:10	1280	800	1680	828	83.460
60	1280X960P	0x001A	16:9	1280	960	1712	994	102.100
60	1280X960P_1	0x001C	16:9	1280	960	1800	1000	108.000
100	1280X960P	0x001A	16:9	1280	960	1760	1017	178.990
60	1280X1024P	0x001E	5:3	1280	1024	1688	1066	108.000
100	1280X1024P	0x001E	5:3	1280	1024	1760	1085	190.960
60	1368X768P	0x0020	16:9	1368	768	1800	795	85.860
60	1400X1050P	0x0022	4:3	1400	1050	1880	1082	122.000
60	1400X1050P_1	0x0024	4:3	1400	1050	1880	1087	122.610
100	1400X1050P	0x0022	4:3	1400	1050	1928	1112	214.390
60	1440X900P	0x0026	16:10	1440	900	1904	932	106.470
60	1600X900P	0x0028	16:9	1600	900	2128	932	118.998
60	1600X1200P	0x002A	4:3	1600	1200	1920	1250	144.000
60	1600X1200P_1	0x002C	4:3	1600	1200	2160	1250	162.000
100	1600X1200P	0x002A	4:3	1600	1200	2208	1271	280.640
60	1680X1050P	0x002E	16:10	1680	1050	2256	1087	147.140
60	1792X1344P	0x0030	4:3	1792	1344	2448	1394	204.800
60	1856X1392P	0x0032	4:3	1856	1392	2528	1439	218.300

F	VFID	VFLVC	Aspect	W_a	H_a	W_t	H_t	F_{vs}
60	1920X1200P	0x0034	16:10	1920	1200	2592	1242	193.160
60	1920X1440P	0x0036	4:3	1920	1440	2600	1500	234.000
*	VENDOR	0xFFFE	*	*	*	*	*	*

* Defined by Vendor

For the 3D2FP Frame Organization, the situation is different. The overall VideoFrame timing is maintained by doubling the VideoSample Rate to account for the fact that double the number of VideoSamples is present.

There are two possible VideoFormat representaions when transporting the 3D2FP Frame Organization over USB:

- The Active Space is suppressed and is not sent over USB ($S_a = 0$).
- The Active Space is included in the VideoPayload sent over USB. The VFIDs reflect this through the –AS suffix.

The following table contains the detailed VideoFrame dimensions and timings. Note that the values for F_{vs} are double those of the previous table.

The following notations are used in the table:

- F: VideoFrame Rate (in Hz)
- VFID: VideoFrame Format ID used to identify the VideoFrame Format
- VFLVC: VideoFrame Format Legacy View Code used to identify the VideoFrame Format while in Legacy View
- Aspect: The aspect ratio of the VideoFrame
- W_a: Active VideoFrame Width (in VideoSamples)
- H_a: Active VideoFrame Height (in VideoLines) for the Left and the Right VideoFrame
- S_a: Indicates the size of the Active Space (in VideoLines)
- H_{at}: Total Active VideoFrame Height (in VideoLines): $2 * H_a + S_a$
- W_t: Total VideoFrame Width (in VideoSamples)
- H_t: Total VideoFrame Height (in VideoLines)
- F_{vs}: VideoSample Rate (pixelclock) (in MHz) – for informational purposes only

Table A-4: Additional VideoFrame Dimensions and Timings for the 3D2FP Frame Organization

F	VFID	VFLVC	Aspect	W_a	H_a	S_a	H_{at}	W_t	H_t	F_{vs}
100	640X480P	0x0002	4:3	640	480	0	960	848	1018	86.320
100	640X480PAS	0x0003	4:3	640	480	29	989	848	1018	86.320
60	768X576P	0x000E	4:3	768	576	0	1152	976	1194	69.920
60	768X576PAS	0x000F	4:3	768	576	21	1173	976	1194	69.920
100	768X576P	0x000E	4:7	768	576	0	1152	1024	1222	125.140
100	768X576PAS	0x000F	4:7	768	576	35	1187	1024	1222	125.140
60	800X600P	0x0010	4:3	800	600	0	1200	1056	1256	80.000
60	800X600PAS	0x0011	4:3	800	600	28	1228	1056	1256	80.000
100	800X600P	0x0010	4:3	800	600	0	1200	1072	1272	136.360
100	800X600PAS	0x0011	4:3	800	600	36	1236	1072	1272	136.360
60	1024X768P	0x0012	4:3	1024	768	0	1536	1344	1612	130.000
60	1024X768PAS	0x0013	4:3	1024	768	38	1574	1344	1612	130.000
100	1024X768P	0x0012	4:3	1024	768	0	1536	1392	1628	226.620
100	1024X768PAS	0x0013	4:3	1024	768	46	1582	1392	1628	226.620
60	1152X864P	0x0014	4:3	1152	864	0	1728	1520	1790	163.240
60	1152X864PAS	0x0015	4:3	1152	864	31	1759	1520	1790	163.240
100	1152X864P	0x0014	4:3	1152	864	0	1728	1568	1830	286.940

F	VFID	VFLVC	Aspect	W_a	H_a	S_a	H_{at}	W_t	H_t	F_{vs}
100	1152X864PAS	0x0015	4:3	1152	864	51	1779	1568	1830	286.940
60	1280X768P	0x0016	5:3	1280	768	0	1536	1664	1596	159.344
60	1280X768PAS	0x0017	5:3	1280	768	30	1566	1664	1596	159.344
60	1280X800P	0x0018	16:10	1280	800	0	1600	1680	1656	166.920
60	1280X800PAS	0x0019	16:10	1280	800	28	1628	1680	1656	166.920
60	1280X960P	0x001A	16:9	1280	960	0	1920	1712	1988	204.200
60	1280X960PAS	0x001B	16:9	1280	960	34	1954	1712	1988	204.200
60	1280X960P_1	0x001C	16:9	1280	960	0	1920	1800	2000	216.000
60	1280X960P_1AS	0x001D	16:9	1280	960	40	1960	1800	2000	216.000
100	1280X960P	0x001A	16:9	1280	960	0	1920	1760	2034	357.980
100	1280X960PAS	0x001B	16:9	1280	960	57	1977	1760	2034	357.980
60	1280X1024P	0x001E	5:3	1280	1024	0	2048	1688	2122	216.000
60	1280X1024PAS	0x001F	5:3	1280	1024	42	2090	1688	2122	216.000
100	1280X1024P	0x001E	5:3	1280	1024	0	2048	1760	2170	381.920
100	1280X1024PAS	0x001F	5:3	1280	1024	61	3009	1760	2170	381.920
60	1368X768P	0x0020	16:9	1368	768	0	1536	1800	1590	171.720
60	1368X768PAS	0x0021	16:9	1368	768	27	1563	1800	1590	171.720
60	1400X1050P	0x0022	4:3	1400	1050	0	2100	1880	2164	244.000
60	1400X1050PAS	0x0023	4:3	1400	1050	32	2132	1880	2164	244.000
60	1400X1050P_1	0x0024	4:3	1400	1050	0	2100	1880	1087	244.320
60	1400X1050P_1AS	0x0025	4:3	1400	1050	37	2137	1880	1087	244.320
100	1400X1050P	0x0022	4:3	1400	1050	0	2100	1928	1112	428.780
100	1400X1050PAS	0x0023	4:3	1400	1050	62	2162	1928	1112	428.780
60	1440X900P	0x0026	16:10	1440	900	0	1800	1904	1864	212.940
60	1440X900PAS	0x0027	16:10	1440	900	32	1832	1904	1864	212.940
60	1600X900P	0x0028	16:9	1600	900	0	1800	2128	1864	237.996
60	1600X900PAS	0x0029	16:9	1600	900	32	1832	2128	1864	237.996
60	1600X1200P	0x002A	4:3	1600	1200	0	2400	1920	2500	288.000
60	1600X1200PAS	0x002B	4:3	1600	1200	50	2450	1920	2500	288.000
60	1600X1200P_1	0x002C	4:3	1600	1200	0	2400	2160	2500	324.000
60	1600X1200P_1AS	0x002D	4:3	1600	1200	50	2450	2160	2500	324.000
100	1600X1200P	0x002A	4:3	1600	1200	0	2400	2208	2542	561.280
100	1600X1200PAS	0x002B	4:3	1600	1200	71	2471	2208	2542	561.280
60	1680X1050P	0x002E	16:10	1680	1050	0	2100	2256	2174	294.280
60	1680X1050PAS	0x002F	16:10	1680	1050	37	2137	2256	2174	294.280
60	1792X1344P	0x0030	4:3	1792	1344	0	2688	2448	2788	409.600
60	1792X1344PAS	0x0031	4:3	1792	1344	50	2738	2448	2788	409.600
60	1856X1392P	0x0032	4:3	1856	1392	0	2784	2528	2878	436.600
60	1856X1392PAS	0x0033	4:3	1856	1392	47	2831	2528	2878	436.600
60	1920X1200P	0x0034	16:10	1920	1200	0	2400	2592	2484	386.320
60	1920X1200PAS	0x0035	16:10	1920	1200	42	2442	2592	2484	386.320
60	1920X1440P	0x0036	4:3	1920	1440	0	2880	2600	3000	468.000

F	VFID	VFLVC	Aspect	W_a	H_a	S_a	H_{at}	W_t	H_t	F_{vs}
60	1920X1440PAS	0x0037	4:3	1920	1440	60	2940	2600	3000	468.000
*	VENDOR	0xFFFE	*	*	*	*	*	*	*	*
*	VENDORAS	0xFFFF	*	*	*	*	*	*	*	*

* Defined by Vendor